



**TECNOLÓGICO NACIONAL DE MÉXICO**  
**INSTITUTO TECNOLÓGICO DE APIZACO**  
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“METODOLOGÍA PARA EL DESARROLLO GLOBAL DE SOFTWARE EN  
APLICACIONES WEB”

**T E S I S**

QUE PARA OBTENER EL GRADO DE:  
MAESTRA EN SISTEMAS COMPUTACIONALES

**PRESENTA:**

LIC. LORENA ZAMORA VELÁZQUEZ

**DIRECTORES:**

M. C. JOSÉ JUAN HERNÁNDEZ MORA

M. C. MARIA GUADALUPE MEDINA BARRERA

APIZACO, TLAXCALA 2016



Apizaco, Tlax., 22 de agosto de 2016

No. de Oficio: DEPI/285/16

ASUNTO: Se Autoriza Impresión de Tesis de Grado.

LIC. LORENA ZAMORA VELÁZQUEZ,  
CANDIDATA AL GRADO DE MAESTRA  
EN SISTEMAS COMPUTACIONALES  
No. de Control: **M08370722**  
PRESENTE.

Por este medio me permito informar a usted, que por aprobación de la Comisión Revisora asignada para valorar el trabajo, mediante la Opción: **I Tesis de Grado por Proyecto de Investigación**, de la **Maestría en Sistemas Computacionales**, que presenta con el tema: **"METODOLOGÍA PARA EL DESARROLLO GLOBAL DE SOFTWARE EN APLICACIONES WEB"** y conforme a lo establecido en el Procedimiento para la Obtención del Grado de Maestría en el Instituto Tecnológico, la División de Estudios de Posgrado e Investigación a mi cargo le emite la:

#### AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar un ejemplar del mismo debidamente encuadernado y seis copias en CD en formato PDF, para presentar su Acto de Recepción Profesional a la brevedad.

Sin otro particular por el momento, le envío un cordial saludo.

**A T E N T A M E N T E**  
*PENSAR PARA SERVIR, SERVIR PARA TRIUNFAR®*

  
DR. JOSÉ FEDERICO CASCO VÁSQUEZ  
JEFE DE LA DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN.



Secretaría de Educación Pública  
Instituto Tecnológico de Apizaco  
División de Estudios de Posgrado  
e Investigación

C.p.- Expediente.

JFCV/MJSH+mebr





Apizaco, Tlax., 11 de agosto de 2016

ASUNTO: Aprobación del trabajo de Tesis de Maestría.

**DR. JOSE FEDERICO CASCO VASQUEZ**  
JEFE DE LA DIVISION DE ESTUDIOS DE  
POSGRADO E INVESTIGACION,  
PRESENTE.

Por este medio se le informa a usted, que los integrantes de la **Comisión Revisora** para el trabajo de tesis de maestría que presenta la **LIC. LORENA ZAMORA VELÁZQUEZ**, con número de control **M08370722**, candidata al grado de **Maestra en Sistemas Computacionales** y egresada del **Instituto Tecnológico de Apizaco**, cuyo tema es **"METODOLOGIA PARA EL DESARROLLO GLOBAL DE SOFTWARE EN APLICACIONES WEB"**, fue:

**APROBADO**

Lo anterior, al valorar el trabajo profesional presentado por la candidata y constatar que las observaciones que con anterioridad se le marcaron así como correcciones sugeridas para su mejora ya han sido realizadas.

Por lo que se avala se continúe con los trámites pertinentes para su titulación.

Sin otro particular por el momento, le envié un cordial saludo.

LA COMISION REVISORA

M.C. JOSE JUAN FERNANDEZ MORA

M.D.S. HIGINIO NAVA BAUTISTA

M.C. MARIA GUADALUPE MEDINA BARRERA

M.C.C. MARIA JANA SANCHEZ HERNANDEZ

C. p.- Interesada.



*Gracias Dios por permitirme  
alcanzar una meta más en mi vida.*

*Dedico este trabajo a mis padres,  
Luis y Leticia,  
y a mis hermanos:  
Daniel y Diego*

## **Agradecimientos**

Gracias al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico, para mi preparación.

Gracias a mis profesores por su dedicación, compromiso y labor realizada que contribuyó en mi formación académica a lo largo de estos años.

Quiero agradecer al Instituto Tecnológico de Apizaco por proporcionar los medios y herramientas necesarias que me permitieron realizar este trabajo de tesis.

Gracias a mi comité tutorial por el tiempo dedicado para el desarrollo adecuado de este proyecto.

Agradezco a mis compañeros y amigos con quienes compartí esta experiencia, gracias porque siempre que los necesité conté con su apoyo.

Deseo agradecer especialmente a Dios por bendecirme con una maravillosa familia, sin ella nada de esto sería posible. Gracias a mi madre por animarme y alentarme en los momentos más difíciles, a mi padre por brindarme al máximo su apoyo y por creer en mí, a mi hermano Danny por cuidarme y por estar conmigo en todas las adversidades, a mi hermanito Diego por ser el más pequeñito y la alegría de la casa.

## **Resumen**

El presente trabajo de tesis, tiene como finalidad la propuesta de una metodología para el Desarrollo Global de Software (GDS, por sus siglas en inglés) enfocada a la construcción de aplicaciones web, con el objetivo de que pueda ser empleada por equipos de trabajo localizados en puntos geográficos distintos, dicha metodología se fundamenta en los principios del desarrollo ágil e ingeniería web, tomando en consideración las buenas prácticas para su elaboración y otorgando los instrumentos necesarios para cumplir con el propósito de la investigación.

Para la creación de la metodología primero se identificaron los elementos primordiales a considerar a partir de algunos trabajos que se analizaron, dichos trabajos se realizaron con el uso de metodologías ágiles y técnicas GDS, este análisis se hizo para poder establecer los criterios a tener en cuenta. También, se llevó a cabo una revisión de los modelos existentes para el desarrollo de aplicaciones basadas en la web, al igual que las metodologías de desarrollo de software tradicional y ágil, realizando una evaluación, selección y compatibilidad entre las características más relevantes de éstas, para finalmente establecer las bases que originaron la metodología propuesta. Para comprobar la aplicación adecuada de la metodología que se propone se utiliza un caso de estudio, para el cual participaron dos equipos de trabajo quienes desarrollaron un sistema web en conjunto, dichos equipos colaboraron separados geográficamente. El resultado del caso de estudio se refleja al generar un producto software a pesar de las dificultades que surgieron por la distancia, que fueron superadas por medio de diferentes estrategias. Además, se aplicó una evaluación de la calidad de la aplicación web, hecha por los equipos que emplearon la metodología propuesta, para comprobar que es posible la producción de software con la calidad requerida.

## **Abstract**

This thesis, aims to propose a methodology for Global Software Development (GDS, for its acronym in English) focused on building web applications, with the aim that can be used by teams located in different geographical locations, this methodology is based on the principles of agile development and web engineering, taking into consideration the best practices to develop and provide the tools necessary to accomplish the purpose of the investigation.

For the creation of the methodology, first the primary elements to consider from some studies were analyzed and identified. Such work was carried out with the use of agile methodologies and GDS techniques, this analysis was to establish the criteria taken into account. Also, there was a review conducted of existing models for the development of web-based applications, like traditional agile software methodology development, making an evaluation, selection and compatibility among the most important characteristics of these applications, to finally establish the foundations that led to the proposed methodology. To verify the proper application of the methodology proposed in the case study, which involved two teams who developed a web system as a whole; these geographically separated teams collaborated separately. The result of the case study is reflected by generating a software product despite the difficulties encountered by the distance, which were overcome through different strategies. In addition, an assessment of the quality of the web application, made by the teams that used the proposed methodology, to verify that software production is possible with the required quality that was applied.



## Índice general

Índice general.....	I
Lista de tablas .....	V
Lista de figuras .....	VII
Capítulo 1 Introducción.....	1
1.1 Descripción del problema.....	1
1.2 Justificación .....	2
1.3 Objetivo general .....	3
1.4 Objetivos específicos.....	3
1.5 Metas .....	4
1.6 Pregunta de investigación.....	4
1.7 Organización de la tesis.....	4
1.8 Análisis del estado del arte .....	4
1.8.1 Modelos y metodologías de ingeniería de software entre lo tradicional, lo ágil y lo global .....	5
1.8.2 Sistemas de información para la gestión de documentos digitales.....	10
1.8.3 Tecnologías de la información para el desarrollo de sistemas de información .....	12
Capítulo 2 Marco teórico.....	14
2.1 Metodologías de desarrollo y de ingeniería del software.....	14
2.1.2 Desarrollo tradicional .....	15
2.1. 2 Desarrollo ágil .....	20
2.1.2.1 Programación extrema (XP) .....	25
2.1.2.2 SCRUM .....	27
2.1.2.3 Crystal.....	30

2.1.2.4 TDD (Test Driven Development).....	31
2.2 Desarrollo global de software (GSD) .....	32
2.2.1 Prácticas ágiles en el GSD.....	33
2.2.2 Desafíos dentro del GSD .....	33
2.3 Ingeniería web .....	34
2.3.1 Métodos de la ingeniería web .....	35
2.3.2 Aplicaciones web.....	38
2.3.2.1 Arquitectura cliente/servidor .....	40
Capítulo 3 Marco metodológico y desarrollo de la propuesta.....	42
3.1. Prospección de los modelos y metodologías existentes .....	42
3.1.1 Metodologías de desarrollo e ingeniería del software .....	42
3.1.2 El trabajo colaborativo como factor de éxito en DGS.....	44
3.1.2.1 Las reuniones en DGS .....	46
3.1.2.2 Tecnologías de la Información y Comunicación (TIC) para la colaboración a través de internet.....	47
3.1.3 Metodologías de desarrollo e ingeniería web .....	50
3.1.4 Herramientas para el desarrollo de aplicaciones web.....	51
3.1.4.1 Herramientas para la lógica de presentación (lado del cliente) .....	52
3.1.4.2 Herramientas de la lógica de negocio.....	54
3.1.4.3 Herramientas para la lógica de datos (lado del servidor) .....	56
3.5 Descripción de la metodología propuesta.....	57
3.5.1 Los participantes.....	61
3.5.2 Reuniones y etapas .....	62
3.5.3 Instrumentos de apoyo.....	68

Capítulo 4 Aplicación de la metodología propuesta.....	85
4.1 Desarrollo del caso de estudio .....	85
4.1.2 El plan de comunicación .....	86
4.1.3 Desarrollo de los ciclos.....	91
4.1.3.1 Ciclo 1. Módulo: Diseño y desarrollo de la base de datos .....	91
4.1.3.2 Ciclo 2. Módulo: Acceso al sistema .....	96
4.1.3.3 Ciclo 3. Módulo: Gestión de usuarios .....	101
4.1.3.4 Ciclo 4. Módulo: Gestión de materiales .....	105
4.1.3.5 Ciclo 5. Módulo: Consultas .....	110
4.1.3.6 Ciclo 6. Módulo: Reportes.....	117
4.1.3.7 Ciclo 7. Módulo: Exportar e importar la base de datos .....	123
4.1.3.8 Ciclo 8. Módulo: Sección de ayuda.....	126
4.1.4 Pruebas y liberación del SARB .....	130
4.1.5 Manual de usuario .....	131
Capítulo 5 Evaluación y resultados .....	132
5.1 Evaluación de la metodología en cuanto a agilidad .....	132
5.3 Pruebas generadas en cada ciclo.....	137
5.2 Pruebas generales del software.....	143
Capítulo 6 Conclusiones y trabajos futuros .....	145
6.1 Conclusiones.....	145
6.2 Trabajos futuros .....	149
Referencias bibliográficas .....	150
Anexos .....	156
Anexo 1. Manual de usuario.....	156

Anexo 2. Publicaciones .....	175
Anexo 3. Cartas .....	189
3.1 Carta de satisfacción.....	189
3.2 Carta de término de estancias .....	190
Anexo 4. Cuestionarios .....	191
Anexo 5. Itinerarios, reportes y formatos de sugerencias y cambios .....	193

## Lista de tablas

Tabla 2.1 Comparación entre metodologías ágiles y tradicionales (Tinoco, Rosales, & Salas, 2010).....	15
Tabla 2.2 Terminología SCRUM. ....	29
Tabla 3.1 Comparativa de Metodologías de Desarrollo de Software (Kumar & Kumar, 2014). .....	43
Tabla 3.2 Problemas en la comunicación y alternativas de solución. ....	45
Tabla 3.3 Comparativa de metodologías de desarrollo web.....	50
Tabla 3.4 Entradas, salidas y participantes de la metodología propuesta.....	60
Tabla 4.1 Plan de actividades del ciclo no. 01, módulo: diseño y desarrollo de la base de datos. .....	92
Tabla 4.2 Plan de actividades del ciclo no. 02, módulo: acceso al sistema.....	97
Tabla 4.3 Interacción entre niveles: Acceso al sistema. ....	100
Tabla 4.4 Plan de actividades del ciclo no. 03, módulo: gestión de usuarios.....	101
Tabla 4.5 Interacción entre niveles: Gestión de usuarios. ....	104
Tabla 4.6 Plan de actividades del ciclo no. 04, módulo: gestión de materiales. ....	105
Tabla 4.7 Interacción entre niveles: Gestión de materiales. ....	109
Tabla 4.8 Plan de actividades del ciclo no. 05, módulo: consultas. ....	110
Tabla 4.9 Interacción entre niveles: Consultas. ....	113
Tabla 4.10 Plan de actividades del ciclo no. 06, módulo: reportes. ....	117
Tabla 4.11 Interacción entre niveles: Reportes. ....	119
Tabla 4.12 Plan de actividades del ciclo no. 07, módulo: exportar e importar la base de datos. .....	123
Tabla 4.13 Interacción entre niveles: Exportar e importar la base de datos. ....	125
Tabla 4.14 Plan de actividades del ciclo no. 08, módulo: sección de ayuda. ....	127
Tabla 4.15 Interacción entre niveles: Sección de ayuda.....	129
Tabla 5.1 Aplicación de la evaluación de acuerdo al primer postulado. ....	133
Tabla 5.2 Aplicación de Evaluación de acuerdo al segundo postulado.....	134

Tabla 5.3 Aplicación de Evaluación de acuerdo al tercer postulado.....	135
Tabla 5.4 Aplicación de Evaluación de acuerdo al cuarto postulado.....	135
Tabla 5.5 Resultados al aplicar el framework a la metodología propuesta. ....	136
Tabla 5.6 Caso de prueba ciclo no. 02, módulo acceso al sistema. ....	137
Tabla 5.7 Caso de prueba ciclo no. 03, módulo gestión de usuarios.....	138
Tabla 5.8 Caso de prueba ciclo no.04, módulo gestión de materiales.....	138
Tabla 5.9 Caso de prueba ciclo no. 05, módulo consultas.....	139
Tabla 5.10 Caso de prueba ciclo no.06, módulo reportes.....	140
Tabla 5.11 Caso de prueba ciclo no. 07, módulo exportar e importar la base de datos. ....	141
Tabla 5.12 Caso de prueba ciclo no. 08, módulo sección de ayuda. ....	141
Tabla 5.13 Pruebas de integración al SARB. ....	142
Tabla 5.14 Resultados de las pruebas al SARB. ....	143

## Lista de figuras

Figura 2.1 Modelo de cascada.....	17
Figura 2.2 Modelo en espiral.....	17
Figura 2.3 Rational Unified Process (Pilemalm, Lindell, Hallberg, & Eriksson, 2007).....	18
Figura 2.4 Modelo en V.....	19
Figura 2.5 El proceso de programación extrema.....	26
Figura 2.6 Metodología SCRUM.....	28
Figura 2.7 Modelo Test Driven Development.....	32
Figura 2.8 Separación de funciones.....	40
Figura 3.1 Trabajo colaborativo en Google drive (Ortiz, 2016).....	48
Figura 3.2 Trabajo colaborativo con Office 2016 (Microsoft, s.f.).....	48
Figura 3.3 Ejemplo de grupo en Facebook.....	49
Figura 3.4 Ejemplo de video llamada con Skype (Microsoft, Skype, s.f.).....	49
Figura 3.5 Almacenamiento y distribución de archivos con Dropbox (Dropbox, s.f.).....	50
Figura 3.6 Patrón Cliente/Servidor.....	52
Figura 3.7 Esqueleto básico de un documento HTML (Luján, 2001).....	54
Figura 3.8 Metodología propuesta.....	59
Figura 3.9 Criterios para el plan de comunicación.....	60
Figura 3.10 Plan de trabajo del ciclo.....	68
Figura 3.11 Plan de comunicación (Sección 1).....	69
Figura 3.12 Plan de comunicación (Sección 2).....	70
Figura 3.13 Plan de comunicación (Sección 3).....	71
Figura 3.14 Plan de comunicación (Sección 4).....	72
Figura 3.15 Plan de comunicación (Sección 5).....	72
Figura 3.16 Plan de comunicación (Sección 6).....	73
Figura 3.17 Plan de comunicación (Sección 7).....	74
Figura 3.18 Itinerario.....	75
Figura 3.19 Reporte de reunión.....	77
Figura 3.20 Recolección de requisitos.....	78

Figura 3.21 Plantilla de análisis de requisitos. ....	79
Figura 3.22 Diseño de la propuesta. ....	80
Figura 3.23 Formato para la parte de elaboración de la etapa 3, interacción entre niveles. ....	81
Figura 3.24 Formato para el caso de prueba. ....	82
Figura 3.25 Documento de validación de liberación del ciclo. ....	82
Figura 3.26 Formato de sugerencias o cambios. ....	83
Figura 4.1 Equipos de trabajo para el desarrollo del caso de estudio. ....	85
Figura 4.2 Plan de comunicación (Sección 1) SARB. ....	86
Figura 4.3 Plan de comunicación (Sección 2) SARB. ....	87
Figura 4.4 Plan de comunicación (Sección 3) SARB. ....	88
Figura 4.5 Plan de comunicación (Sección 4) SARB. ....	88
Figura 4.6 Plan de comunicación (Sección 5) SARB. ....	89
Figura 4.7 Plan de comunicación (Sección 6) SARB. ....	89
Figura 4.8 Plan de comunicación (Sección 7) SARB. ....	90
Figura 4.9 Continuación del plan de comunicación (Sección 7) SARB. ....	91
Figura 4.10 Formato de recolección de requisitos SARB-Ciclo 01. ....	93
Figura 4.11 Formato de análisis de requisitos SARB-Ciclo 01. ....	93
Figura 4.12 Diseño de la propuesta SARB-Ciclo 01. ....	94
Figura 4.13 Diseño de la base de datos del sistema. ....	94
Figura 4.14 Diagrama entidad-relación de la base de datos SARB. ....	95
Figura 4.15 Interfaz general del SARB (sección que no requiere autenticación). ....	96
Figura 4.16 Interfaz general del SARB (sección que requiere autenticación). ....	96
Figura 4.17 Formato de recolección requisitos SARB-Ciclo 02. ....	97
Figura 4.18 Formato de análisis de requisitos SARB-Ciclo 02. ....	98
Figura 4.19 Diagrama del caso de uso SARB-Ciclo 02. ....	98
Figura 4.20 Diseño de la propuesta SARB-Ciclo 02. ....	99
Figura 4.21 Interfaz de acceso al sistema. ....	100
Figura 4.22 Interfaz de bienvenida. ....	101
Figura 4.23 Formato de recolección requisitos SARB-Ciclo 03. ....	102
Figura 4.24 Formato de análisis de requisitos SARB-Ciclo 03. ....	102

Figura 4.25 Diagrama del caso de uso SARB-Ciclo 03. ....	103
Figura 4.26 Diseño de la propuesta SARB-Ciclo 03.....	103
Figura 4.27 Formulario para agregar un nuevo usuario. ....	104
Figura 4.28 Interfaz para la administración de usuarios.....	105
Figura 4.29 Formato de recolección requisitos SARB-Ciclo 04.....	106
Figura 4.30 Formato de análisis de requisitos SARB-Ciclo 04.....	106
Figura 4.31 Diagrama del caso de uso SARB-Ciclo 04. ....	107
Figura 4.32 Diseño de la propuesta SARB-Ciclo 04.....	108
Figura 4.33 Interfaz para agregar una nueva tesis.....	109
Figura 4.34 Interfaz que permite la administración de materiales. ....	110
Figura 4.35 Formato de recolección requisitos SARB-Ciclo 05.....	111
Figura 4.36 Formato de análisis de requisitos SARB-Ciclo 05.....	111
Figura 4.37 Diagrama del caso de uso SARB-Ciclo 05. ....	112
Figura 4.38 Diseño de la propuesta SARB-Ciclo 05.....	113
Figura 4.39 Menú del sistema con los distintos filtros para las consultas.....	114
Figura 4.40 Interfaz para la consulta por carrera.....	114
Figura 4.41 Interfaz para la consulta por fecha. ....	115
Figura 4.42 Interfaz para la consulta por asesor / director. ....	115
Figura 4.43 Interfaz para la consulta por autor.....	115
Figura 4.44 Interfaz para la consulta por título. ....	116
Figura 4.45 Interfaz para la consulta por número de descargas. ....	116
Figura 4.46 Interfaz para la consulta en todo el contenido disponible.....	116
Figura 4.47 Formato de recolección requisitos SARB-Ciclo 06.....	117
Figura 4.48 Formato de análisis de requisitos SARB-Ciclo 06.....	118
Figura 4.49 Diagrama del caso de uso SARB-Ciclo 06. ....	118
Figura 4.50 Diseño de la propuesta SARB-Ciclo 06.....	119
Figura 4.51 Adición del botón de reporte en la búsqueda por carrera.....	120
Figura 4.52 Adición del botón de reporte en la búsqueda por fecha. ....	120
Figura 4.53 Adición del botón de reporte en la búsqueda por asesor / director. ....	121
Figura 4.54 Adición del botón de reporte en la búsqueda por autor. ....	121

Figura 4.55 Adición del botón de reporte en la búsqueda por título. ....	122
Figura 4.56 Adición del botón de reporte en la búsqueda por número de descargas. ....	122
Figura 4.57 Adición del botón de reporte en la búsqueda en todo el contenido. ....	122
Figura 4.58 Formato de recolección requisitos SARB-Ciclo 07. ....	123
Figura 4.59 Formato de análisis de requisitos SARB-Ciclo 07. ....	124
Figura 4.60 Diagrama del caso de uso SARB-Ciclo 07. ....	124
Figura 4.61 Diseño de la propuesta SARB-Ciclo 08. ....	125
Figura 4.62 Interfaz para exportar e importar la base de datos. ....	126
Figura 4.63 Formato de recolección requisitos SARB-Ciclo 08. ....	127
Figura 4.64 Formato de análisis de requisitos SARB-Ciclo 08. ....	128
Figura 4.65 Diagrama del caso de uso SARB-Ciclo 08. ....	128
Figura 4.66 Diseño de la propuesta SARB-Ciclo 08. ....	129
Figura 4.67 Sección de ayuda: Manual de usuario. ....	130
Figura 4.68 Aprobación de liberación del SARB. ....	130
Figura 5.1 Formulación de Framwork de la metodología propuesta en cuanto agilidad (Mendes, Estevez, & Fillotrani, 2010). ....	132
Figura 5.2 Resultados de la evaluación general del SARB. ....	144

## Capítulo 1 Introducción

En la actualidad existe la tendencia a desarrollar software a través del uso de metodologías ágiles y el desarrollo global de software (DGS o GSD, por sus siglas en inglés Global Software Development). Estas herramientas se consideran ideales para la construcción de una aplicación web, el cual tenga fundamento en dichas tendencias y de esta manera reducir la distancia entre equipos de trabajo, que se encuentran localizados en sitios remotos, así como la entrega rápida de software incremental. Sin embargo el verdadero problema está en lograr resolver los problemas que surgen por la distancia (comúnmente de comunicación).

Un ejemplo de aplicación web es un Sistema de Administración de Recursos Bibliográficos (SARB), que además permite lograr una más amplia difusión del material bibliográfico hacia un mayor número de usuarios en línea, así como una gestión eficiente en forma remota de dichos documentos que conforman el recurso bibliográfico. Un sistema de administración de documentos digitales permite un ahorro en papel, pues al estar de manera digital no solo contribuye al medio ambiente, también contribuye al ahorro de espacio, independencia de los documentos, acceso eficiente a los archivos, integridad y seguridad de la información y una mejor administración. Por lo cual al desarrollar este software es indispensable hacer uso de los elementos y herramientas necesarias para su implementación, que den la confianza de que el trabajo realizado sea de calidad, cumpliendo con los plazos establecidos en tiempo y forma, correcta administración del recurso, realizando cada etapa de manera satisfactoria, asegurar que se cumple con los requisitos del cliente y que este quede conforme con el trabajo entregado.

### 1.1 Descripción del problema

Puesto que actualmente se tiene la necesidad de trabajar en equipos para el desarrollo de sistemas de información pero con las nuevas sociedades del conocimiento, que pueden estar en lugares remotos, surge la necesidad de crear metodologías de ingeniería del software para poder sincronizar y colaborar con equipos que no estén en un mismo lugar físico. Es por ello

que se propone implementar una metodología de ingeniería del software que permita desarrollar un sistema web en equipos de trabajo separados físicamente. Y para la demostración del funcionamiento de la metodología propuesta se plantea crear un sistema automático de bibliografía digital para poner documentos en línea como pueden ser tesis de los institutos tecnológicos federales del estado, que presenta la siguiente problemática:

La mayoría de las tesis están en forma impresa y solo algunas están en digital, esto hace que la consulta dependa mucho de la ubicación física, luego entonces hay pocas consultas y sólo son de manera local. Además el uso de las tesis puede ser insuficiente para toda la población estudiantil debido a que la institución sólo cuenta con un tomo por tesis; también al ser impresas ocupan mucho espacio físico para su almacenamiento y no se contribuye a la conservación del medio ambiente.

## **1.2 Justificación**

Actualmente las empresas desarrollan sistemas en lugares o países diferentes debido a la globalización. Esta manera de producir de software es mejor conocida como desarrollo global de software (GSD, *Global Software Development*), lo cual está adquiriendo gran importancia, esto debido a que desde una perspectiva mundial se requiere ser una empresa rentable y competitiva para poder mantenerse dentro del mercado. Para lograr esto, las empresas pueden beneficiarse del surgimiento de las grandes fuerzas de trabajo que cuentan con habilidades múltiples a un menor costo en lugares donde la producción es más barata, esto gracias a las tecnologías de la información y comunicación. Por medio de estas el producto software (código) se puede enviar de manera rápida a los diferentes lugares de desarrollo. Vemos el caso específico de India y China, quienes ofrecen grandes fuerzas de trabajo con múltiples habilidades a un costo muy bajo si se compara con Estados Unidos y Europa Occidental. Estos países no son los únicos, también están Brasil, Europa del Este y Rusia, Malasia y Vietnam (Ågerfalk, Fitzgerald, Holmstrom, & Ó Conchúir, ICSP 2008).

Ahora bien, considerando que no todos los Tecnológicos Federales del estado de Tlaxcala tienen personal específico para el desarrollo de software, pero si cuentan con profesionales del área, quienes pueden colaborar de manera remota. Como es el caso particular del Instituto Tecnológico de Apizaco (ITA) y del Instituto Tecnológico del Altiplano de Tlaxcala (ITAT). Estas instituciones se ubican en puntos geográficamente separados, en las cuales se pueden crear equipos de desarrollo de software con profesores investigadores, quienes pueden trabajar en el proyecto desde sus respectivos planteles. Desde esta perspectiva se observa un área de oportunidad para probar una metodología de desarrollo de software con equipos de trabajo físicamente separados en las instituciones antes mencionadas.

### **1.3 Objetivo general**

Implementar una metodología de ingeniería del software que permita desarrollar un sistema de información para la web, con equipos de trabajo localizados en ubicaciones separadas, cumpliendo con la calidad requerida.

### **1.4 Objetivos específicos**

- Hacer el análisis del estado del arte de la ingeniería del software para equipos separados.
- Realizar el planteamiento de la metodología de ingeniería del software.
- Utilizar la metodología propuesta para desarrollar un Sistema de Administración de Recursos Bibliográficos con equipos de trabajo separados.
- Análisis y diseño del sistema automático para el manejo de información bibliográfico digital con la metodología planteada.
- Desarrollar e implementar el sistema en un Tecnológico Federal del estado de Tlaxcala.
- Realizar pruebas del desarrollo de software y su implementación.

### **1.5 Metas**

- Una metodología de ingeniería del software que permita desarrollar un sistema de información con equipos de trabajo en lugares diferentes.
- Un prototipo de un sistema de información desarrollado con la metodología planteada.
- Dos publicaciones.

### **1.6 Pregunta de investigación**

¿Es posible crear una metodología de Ingeniería de Software que permita desarrollar sistemas de información de calidad y seguros, con equipos de trabajo en lugares diferentes?

### **1.7 Organización de la tesis**

La estructura de este trabajo de investigación se describe a continuación: El Capítulo 1 describe de manera general, el problema en cuestión, la justificación, los objetivos tanto generales como específicos, alcances, limitaciones y la organización que tiene el documento, por último se trata el estado del arte de los trabajos relacionados a este. El Capítulo 2 contempla el marco teórico, el cual se refiere a todos los conceptos que permiten al lector entender el trabajo. El Capítulo 3 desarrolla los procedimientos que permitieron definir la solución más viable para elaborar la metodología junto con el desarrollo de la propuesta que hace referencia al objetivo general de la investigación. El Capítulo 4 comprende la descripción de la aplicación de la metodología propuesta en un caso de estudio, los resultados se detallan en el Capítulo 5 describiendo el proceso de evaluación que verifica que la metodología cumpla con los fines propuestos. El Capítulo 6 cierra la investigación con las conclusiones y las recomendaciones, y finalmente se presentan los anexos.

### **1.8 Análisis del estado del arte**

En esta sección se describen las investigaciones más recientes sobre los trabajos relacionados a esta investigación. Se analizan los modelos y metodologías de ingeniería del software

tradicionales, ágiles y globales, al igual que el desarrollo de aplicaciones web para la gestión de documentos y las tecnologías de la información como herramienta de apoyo en dichos proyectos.

### **1.8.1 Modelos y metodologías de ingeniería de software entre lo tradicional, lo ágil y lo global**

En un estudio realizado para analizar los modelos de ingeniería de software partiendo de lo tradicional a las metodologías modernas, Kumar & Kumar (2014) indican que su *investigación tiene que ver con las metodologías que analizan el ciclo de vida de software a través de los modelos de desarrollo, que se conocen como el ciclo de vida de desarrollo de software*. Y que considerando que dichos modelos tienen tanto ventajas como desventajas, el objetivo principal su trabajo fue la representación de los distintos modelos de desarrollo de software, exponiendo para cada uno tanto las buenas como las malas prácticas. Para esto realizan un análisis comparativo de las metodologías tradicionales y modernas. Estas metodologías fueron examinadas en el ciclo de vida y estudiadas por las dos categorías mencionadas: tradicionales y modernas. Dentro de las metodologías tradicionales se estudiaron fueron el modelo de cascada, el modelo en espiral, el modelo de desarrollo rápido de aplicaciones (RAD), modelo del proceso racional unificado (RUP) y el modelo V; para las metodologías modernas se consideraron el modelo basado en componentes y los modelos ágiles (Programación extrema, Scrum, Crystal, Desarrollo Basado en Funciones, Desarrollo Guiado por Pruebas). Las ventajas y desventajas que se encontraron de los modelos anteriores son:

- Métodos tradicionales: se ocupan en proyectos altamente críticos en los cuales los requerimientos no cambian frecuentemente, y tanto los requerimientos como sus características son limitados, con un equipo de desarrollo grande.
- Metodologías modernas: estas se ocupan en proyectos que tienen un nivel crítico un poco bajo, pero al contrario de los métodos tradicionales los requerimientos si cambian más frecuentemente, el diseño es maleable, la calidad puede ser mejorada, los avances se ven reflejados como entregas que se realizan de manera iterativa e incremental, mayor rendimiento, se pueden detectar fallas con una fácil capacidad.

Con lo anterior se llega a la conclusión que para seleccionar adecuadamente un modelo de desarrollo de software es importante que el equipo de trabajo elija aquel que se adapte mejor al proyecto a desarrollar, considerando lo que cada metodología ofrece (Kumar & Kumar, 2014).

Centrándose en el caso del desarrollo de software en México, una investigación realizada por Jiménez y Orantes (2012) dice que las compañías que se dedican a producir software son candidatas a utilizar una nueva tendencia que surge en el campo de la ingeniería del software conocida como metodologías híbridas. Esto lo consiguieron en un estudio que se realizó con empresas mexicanas desarrolladoras de software. Para este fin, igual se hizo uso de resultados estadísticos concernientes a las prácticas de ingeniería de software que se usan en estas empresas.

Existen, como ya se vio anteriormente, dos tipos principales de metodologías que son las tradicionales y ágiles. Pero actualmente son las metodologías híbridas las que están creando una nueva tendencia en la ingeniería del software, puesto que combina lo mejor de ambas metodologías, aprovechando de esta manera las ventajas que cada una ofrece.

Los resultados de este estudio se obtuvieron a partir de una prueba de hipótesis de manera formal. Lo que se ocupó para poder realizar el estudio fueron encuestas, datos estadísticos del INEGI (Instituto Nacional de Estadística y Geografía). Las encuestas fueron aplicadas a 86 empresas de la lista que proporciona el INEGI seleccionándolas de manera aleatoria. El resultado que arroja este trabajo es que las empresas mexicanas de desarrollo de software son candidatas para utilizar metodologías híbridas, además de que seguramente al ocupar dichas metodologías lograrán mejores resultados.

Jiménez y Orantes (2012) dicen que *es recomendable que se diseñe y utilice una metodología híbrida para el desarrollo de software, que integre prácticas de Ingeniería de Software de las tres metodologías más usadas por las empresas en México: RUP, XP y Scrum*, seleccionando aquellas características que otorguen mayores ventajas, tomando en cuenta el software que se va a desarrollar, el equipo desarrolladores, los recursos hardware y el tiempo establecido para concluir el proyecto, entre otros factores.

Los equipos de desarrollo de software en México comúnmente están conformados por 10 integrantes aproximadamente; para el desarrollo el tiempo estimado es de alrededor de 2 a 3 meses, y lo que generalmente se producen son aplicaciones web. Según la investigación estos elementos son los que permiten hacer uso de una metodología híbrida. Lo anterior no quiere decir que grandes empresas de desarrollo de software no puedan emplear este tipo de metodologías, puesto que al considerar las características de cada empresa así como al equipo de trabajo, es posible aplicar una metodología híbrida que muy probablemente proporcionará resultados buenos.

La mayoría de las empresas dedicadas a desarrollar software en México no llevan a cabo un estudio de factibilidad y tampoco hacen uso de un modelo de gestión y aseguramiento de la calidad. De acuerdo a la investigación dichas empresas podrían crear mejores planes para el desarrollo de software a partir del análisis, discusión y diseño adecuado de una metodología híbrida para seleccionar aquellas características que representen una mayor ventaja de los diferentes modelos de desarrollo de software.

Puede resultar de gran ayuda una metodología híbrida para aquellas empresas de software en México, pero no solo en este país sino también en aquellos países de Latinoamérica que tienen gran similitud, en los cuales se pueden obtener los mismos resultados, tomando en cuenta las diferencias propias de estos países, de sus empresas de software y de los equipos de desarrollo.

En conclusión en este trabajo de investigación se demuestra que es posible aplicar una metodología híbrida en el desarrollo de software en México. Igualmente, provee información que puede llegar a ser de gran ayuda al momento de diseñar dicha metodología, o en el caso de prácticas existentes en las empresas que se dedican al desarrollo de software. Esta información puede ser utilizada en otras investigaciones afines a la ingeniería de software en México (Jiménez & Orantes, 2012).

En el trabajo realizado por Paasivaara & Laseenius (2006) se argumenta que en una primera impresión los métodos de desarrollo de software ágil y global pudieran parecer que no son

compatibles, puesto que para los métodos ágiles es de suma importancia la comunicación presencial, es decir, cara a cara, en cambio para el desarrollo global de software la comunicación es el mayor problema al que se enfrenta. Uno de los principales obstáculos a vencer es el poder emplear prácticas ágiles en aquellos ambientes en los cuales no es posible una interacción continua cara a cara.

Más sin embargo, se han empleado métodos ágiles en proyectos distribuidos con éxito, lo que demuestra que es posible que beneficien al desarrollo global de software. La investigación analiza los beneficios y retos potenciales al incluir los métodos ágiles dentro del desarrollo global de software (Paasivaara & Lassenius, 2006).

La comunicación, como ya se mencionó, es uno o quizá el mayor problema dentro del desarrollo global de software. Minoli (2010) propone la introducción de IR (Ingeniería de requisitos) en A-GSD (Desarrollo Global de Software Ágil) con lo cual se pretende reducir la distancia y conseguir de esta manera mejorar la comunicación entre los integrantes de los equipos distribuidos.

Para el desarrollo de esta investigación se seleccionó un conjunto de técnicas conocidas, dichas técnicas pueden contribuir al dar valor a procesos de desarrollo distribuido ágil, también se presenta cómo estas técnicas se pueden emplear.

Puede parecer en un principio que la introducción de estas técnicas es algo contrario a los métodos ágiles, pero en opinión de Minoli (2010) es que complementan adecuadamente a los métodos ágiles en equipos distribuidos. Y al contrario como ocurre con equipos de trabajo ágiles, los equipos A-GSD requieren de una mayor y mejor documentación debido a que no se encuentran ubicados en el mismo lugar. Todas las herramientas propuestas requieren ser probadas en entornos de desarrollo reales.

El desarrollo global de software es una tendencia en aumento, por lo tanto (Avritzer, y otros, 2014) se necesitan proyectos de software en los que su diseño este realizado con técnicas de

desarrollo global de software para implementar procesos y herramientas que permitan apoyar la colaboración a través de grandes áreas geográficas. De una manera más precisa, los proyectos de este tipo requieren el desarrollo de procesos y herramientas que den soporte a la administración de proyectos, a la comunicación así como a la gestión de riesgos. A diferencia del desarrollo tradicional de software el cual puede aplicar procesos y herramientas estándar para apoyar la comunicación y la colaboración, el desarrollo global de software en cambio necesita contar con una administración de proyectos unificada e integral, un proceso de desarrollo, colaboración, y un enfoque de comunicación considerando las zonas horarias, el número de sitios y la diversidad cultural.

En el trabajo realizado por Avritzer et al. (2014), se presenta un enfoque nuevo para el modelado y cuantificación de los marcos de la ingeniería del software. Para la realización de la propuesta se utilizaron métricas de supervivencia transitorios para apoyar el diseño de los proyectos globales de ingeniería de software. Su enfoque combina el análisis de supervivencia y el análisis de los marcos de ingeniería de software global. En este enfoque, aplicaron las métricas de supervivencia transitorias para apoyar el diseño de proyectos de ingeniería de software a nivel mundial. Por lo tanto combina el análisis de supervivencia y análisis global marcos de ingeniería de software. Se ocupó el tiempo requerido para poder recuperarse de un desastre dentro un proyecto de software como métrica de supervivencia en un marco de ingeniería de software global dado. La estructura del modelo de ingeniería de software global que se empleó se compone de modelos de apoyo a la evaluación de las herramientas de comunicación, los procesos de desarrollo de software y la gestión de la diversidad cultural. En este trabajo se demostró la aplicación del enfoque propuesto empleándolo al análisis de un ejemplo procedente de un proyecto real de desarrollo de software global. Los resultados que se obtuvieron muestran que la combinación de análisis de supervivencia y el modelado de los marcos globales de ingeniería de software pueden proporcionar una perspectiva interesante en el diseño de los marcos globales de ingeniería de software.

Lo que finalmente se obtuve de este trabajo fue la propuesta de una nueva teoría, el cual es el primero en tratar de cuantificar la ruta a la supervivencia en términos de esfuerzo / tiempo del

proyecto. Se agregaron en los resultados del modelo de supervivencia de la evaluación en el marco de las siguientes dimensiones: 1) la comunicación, 2) la diversidad cultural y 3) el proceso. El enfoque del modelo identifica algunos de los factores más importantes que contribuyen a los problemas relacionados con la comunicación, la distancia cultural y el proceso. Asimismo, prescribieron intervenciones apropiadas para mitigar estos problemas. Una vez que se aplican estas intervenciones, se está en condiciones de evaluar en una escala normalizada (0-1) la eficacia de la comunicación y el proceso. Estas evaluaciones se aplicaron aun proyecto de software industrial grande para calcular métricas de supervivencia tales como el tiempo medio para recuperar el proyecto de un desastre y la probabilidad de recuperación completa del proyecto a tiempo. Los resultados muestran el impacto de diferentes parámetros del sistema en el tiempo medio para recuperarse de un desastre.

Avritzer et al. (2014) asegura que el enfoque del modelo introducido en este documento puede ser utilizado como una herramienta de evaluación de bajo nivel para ayudar a evaluar alternativas de inversión para la construcción de los marcos de desarrollo de software global. La métrica de supervivencia se puede calcular de manera eficiente, ya que se evalúa analíticamente. Por lo tanto, la heurística de inversión teniendo en cuenta el impacto de supervivencia y el costo asociado de las intervenciones correspondientes se puede desarrollar con gran precisión y un buen rendimiento.

En concreto, se propone recoger datos empíricos sobre diferentes marcos de proyectos de desarrollo de software global para desarrollar enfoques para mejorar la supervivencia. (Avritzer et al., 2014).

### **1.8.2 Sistemas de información para la gestión de documentos digitales**

En el artículo de Fu, Zhang, & Hu (2011) se dice que las bibliotecas digitales son una tendencia y que una Biblioteca Digital Móvil es una extensión y expansión de estas. Va dirigido a dispositivos móviles, debido a la gran popularidad de su uso entre la población, puesto que lo hace fácil de llevar a cualquier lado, almacenar, leer los documentos en

cualquier lugar en cualquier momento, buscar nuevos contenidos, entre otras ventajas. En este trabajo se presenta como llevar a cabo la realización de una biblioteca digital móvil basada en papel electrónico, la cual es una combinación de los libros en papel electrónicos y recursos digitales. En base a esto, las bibliotecas pueden ampliar el modo de servicio y el campo, ofrecer a los lectores una nueva plataforma de servicios de lectura móvil. Para la realización de una biblioteca digital móvil primero son necesarios los requisitos como es la conexión a internet, los tipos de formato soportado para los libros de papel electrónico (TXT, HTML, PNG, JPG, PDF, MP3, MTXT), las búsquedas en el texto con alta precisión a través de palabras claves. Posteriormente los requisitos del entorno para lograr la biblioteca digital móvil:

1. El medio ambiente de los recursos digitales
2. Un entorno multi-colaborativo
3. El entorno de protección de copyright
4. El entorno de la red inalámbrica

Finalmente se concluye que la tecnología suele causar cambios positivos en el servicio de bibliotecas, ya que el uso de papel electrónico modifica las formas de lectura tradicionales. Por lo cual las bibliotecas digitales móviles serán una aplicación común entre la sociedad en un futuro cercano.

Este futuro cercano no está tan lejos como pudiéramos imaginar, un claro ejemplo de una biblioteca digital en la cual incluyen un apartado de tesis digitales es en la biblioteca central de la UNAM (Ordoñez, 2003). Esta biblioteca proporciona una herramienta de suma importancia que permite preservar, conservar y difundir todas las tesis que se producen en esta institución cada año. En este trabajo se justifica y se da a conocer los motivos del sistema de las tesis digitales en la UNAM. El sistema emerge como un nuevo servicio de recursos electrónicos dentro de Biblioteca Central.

El servicio de las tesis digitales ofrece al usuario rapidez en el acceso y consulta, así como brindar el acceso multiusuario a un mismo título, característica ausente en las tesis impresas y microfilmadas.

Dado que cada año se genera gran cantidad de tesis en la UNAM, las tesis digitales son una solución al problema del espacio físico.

Una de las ventajas de las tesis digitales es que el usuario a través de las tecnologías de la información ya no tiene que recurrir como antes a la versión impresa como única opción para acceder a la información.

### **1.8.3 Tecnologías de la información para el desarrollo de sistemas de información**

Existe un trabajo que describe una metodología desarrollada en base a las tecnologías de la información con el título “Gestión y Desarrollo de Proyectos de Software: Metodología Ágil basada en Telecomunicaciones MATE”. Esta metodología como su nombre lo indica se basa en telecomunicaciones la cual se orienta al desarrollo de proyectos de software en el que participan grupos de desarrollo pequeños que cuentan con pocos recursos, aprovechando la tecnología existente y ahorrando costos que se generan en una empresa de software convencional, al ser un ambiente completamente virtual. El trabajo en esta metodología es semi-sincrónico, esto quiere decir que muchas de las actividades se realizan en correspondencia con otros colaboradores en ciertos momentos que se requiere juntar el trabajo realizado, revisar avances y acordar nuevas entregas. Se hicieron uso de elementos de otras metodologías como Scrum y la programación en parejas XP, pero considerando que las reuniones se realizan de manera virtual y las parejas de programadores residen en lugares diferentes. El principal objetivo de esta metodología es que permita el desarrollo de software con los miembros del equipo de desarrollo trabajando a distancia.

En general los recursos necesarios para poner en marcha un proyecto de desarrollo de software son: a) recursos humanos, b) ámbito de trabajo, c) hardware de desarrollo y comunicaciones,

d) software, e) capital, f) tiempo y g) una metodología apta para el trabajo distribuido. En base a esto y a las metodologías antes mencionadas se propone Metodología Ágil basada en Telecomunicaciones (MATE) compuesta por las fases: 1) Ciclo analítico, 2) Acuerdo VIP (Valor Individual Ponderado de las historias de cada EVA), 2) Ronda grande, 3) Ronda media (semanal o quincenal), 4) Ronda diaria, 5) Evaluación EVA.

Para comprobar la funcionalidad de esta metodología el equipo de desarrollo se conformó por miembros que radicaban en distintos países con el mismo huso horario, confirmando el potencial de esta metodología para el trabajo distribuido, desarrollando como producto final. El sistema e-MATE que es una herramienta que ayuda a la organización y planificación de proyectos de desarrollo de sistemas, tanto para el uso de la metodología MATE como Scrum (Tumino, Bournissen, Bracalenti, Schlemper, & Kucharski, 2013).

Al finalizar este capítulo se han analizado trabajos relacionados a lo que se pretende realizar en este proyecto de investigación, puesto que se llevó a cabo la revisión del estado del arte. De igual manera se presentó la descripción del problema, la justificación, la cual indica el por qué se debe desarrollar una metodología para el DGS. Se mencionaron los objetivos de forma general y específica, entre otros aspecto relevantes a tomar en cuenta.

## Capítulo 2 Marco teórico

En este capítulo se encuentran los conceptos teóricos precisos que permitirán aclarar el resto de los procesos que se realizaron en este trabajo. Aquí se describen todos estos conceptos utilizados en el desarrollo de la metodología del proyecto, como son las principales metodologías de desarrollo y de ingeniería del software tradicional y ágil, al igual que conceptos relacionados a aplicaciones web y el desarrollo global de software.

### 2.1 Metodologías de desarrollo y de ingeniería del software

El desarrollo de software no es algo sencillo por lo que las metodologías son un aspecto fundamental para formalizar y optimizar este proceso, así como generar un producto con de calidad, en el tiempo y los costos establecidos.

El concepto “ingeniería de software” se propuso originalmente en 1968, en una conferencia realizada para discutir lo que entonces se llamaba la “crisis del software”. Se volvió claro que los enfoques individuales al desarrollo de programas no escalaban hacia los grandes y complejos sistemas de software. Éstos no eran confiables, costaban más de lo esperado y se distribuían con demora. (Sommerville, 2011).

Diversos autores coinciden en señalar algunos requisitos que deben tener las metodologías de desarrollo (Tinoco, Rosales, & Salas, 2010):

- Visión del producto.
- Vinculación con el cliente.
- Establecer un modelo de ciclo de vida.
- Gestión de los requisitos.
- Plan de desarrollo.
- Integración del proyecto.
- Medidas de progreso del proyecto.
- Métricas para evaluar la calidad.

- Maneras de medir el riesgo.
- Como gestionar los cambios.
- Establecer una línea de meta.

En tiempos recientes, han surgido las metodologías ágiles, como una alternativa, una reacción a las metodologías tradicionales y principalmente a su burocracia.

Tinoco, Rosales y Salas (2010) muestra en resumen las características de las dos metodologías en la siguiente tabla:

**Tabla 2.1 Comparación entre metodologías ágiles y tradicionales (Tinoco, Rosales, & Salas, 2010)**

Metodologías ágiles	Metodologías tradicionales
Se basan en heurísticas provenientes de prácticas de producción de código.	Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo.
Preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente por el equipo.	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso muy controlado, numerosas normas.
Contrato flexible o incluso inexistente.	Contrato prefijado.
El cliente es parte del desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10).	Grupos grandes.
Pocos artefactos.	Más artefactos.
Menos énfasis en la arquitectura del software.	La arquitectura de software es esencial.

### 2.1.2 Desarrollo tradicional

Las metodologías de desarrollo tradicional de software son conducidas a través de una planeación.

El desarrollo de un proyecto en estas metodologías comienza con un estricto proceso de elicitación de requerimientos, el cual se realiza antes de las etapas de análisis y diseño. Con lo

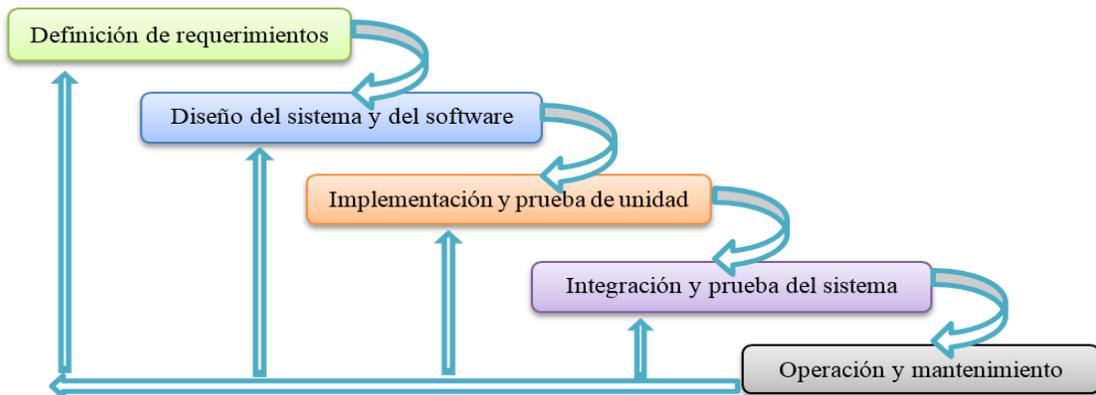
cual se pretende garantizar como resultado un producto software de alta calidad en tiempo y forma de acuerdo al calendario establecido.

Sólo es posible realizar un proyecto a la vez al emplear las metodologías tradicionales, es difícil imaginar el realizar otros proyectos simultáneamente, puesto que por lo general son proyectos de grandes dimensiones y estructuras bien definidas, además de que el proceso que se sigue es secuencial hacia una sola dirección y sin opción de retorno. Esto que indica que es inflexible y sin modificaciones; una vez que los requerimientos son establecidos y acordados estos permanecerán de esta manera, es decir, sin cambios a lo largo del proyecto. Dada esta postura se requiere de periodos de tiempos prolongados de planeación previa, y cuando finalmente se concluye dicha planeación el cliente deja de tener una participación importante (Navarro, Fernández, & Morales, 2013). Entre las metodologías tradicionales podemos mencionar:

- Cascada
- Espiral
- RUP (Rational Unified Process)
- El modelo en V

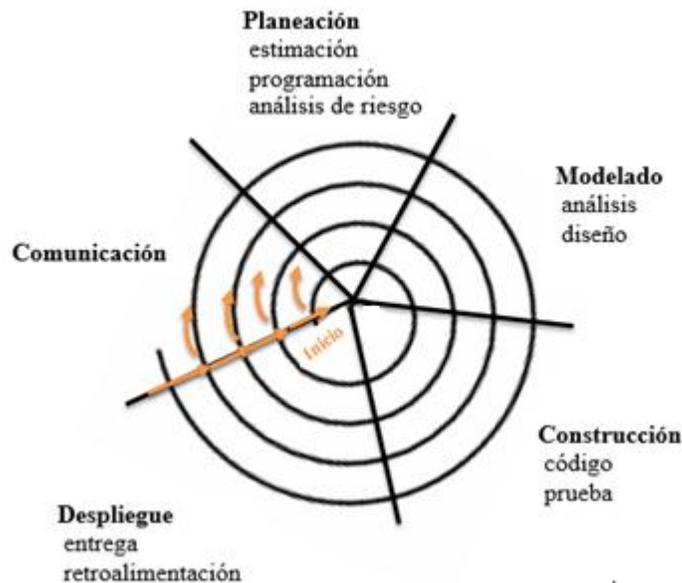
**Modelo de cascada.** Desarrollado por Royce en 1970 y llamado de esta manera por su gráfico que se parece a la caída de una cascada, el cual se muestra en la figura 2.1, donde se aprecian las etapas de este modelo. Dicho modelo es fácil de entender y poner en práctica, además de que es uno de los modelos más antiguos. El modelo de cascada sirve como punto de referencia para muchos otros modelos de ciclo de vida. Este modelo refuerza la noción de definir antes de diseñar, y diseñar antes de comenzar con el código (Kumar & Kumar, 2014).

El modelo de cascada es un claro ejemplo de un proceso conducido por un plan; en principio, se debe planear y programar todas las actividades del proceso, antes de comenzar a trabajar con ellas. (Sommerville, 2011).



**Figura 2.1 Modelo de cascada.**

**Modelo en espiral.** Este es otro modelo clásico como el de cascada, y de los más populares. Con el empleo del modelo espiral (figura 2.2), el software se desarrolla en una serie de entregas evolutivas. Este modelo es dividido por el equipo de software en un conjunto de actividades estructurales (Pressman, 2010).

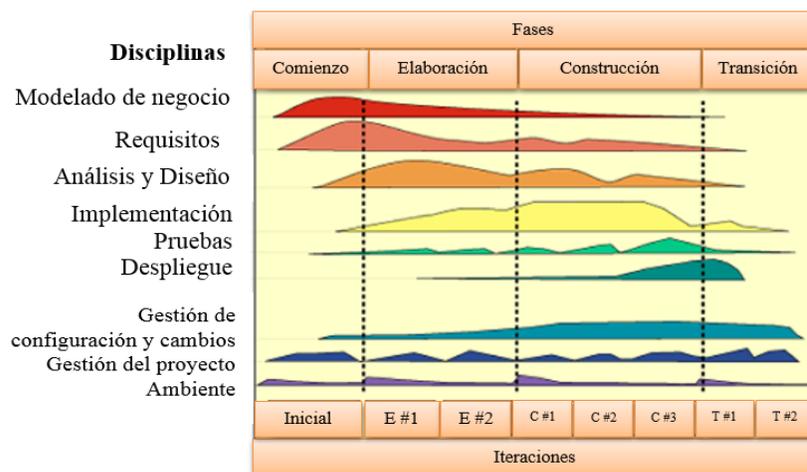


**Figura 2.2 Modelo en espiral.**

**RUP (Rational Unified Process).** Es un proceso de ingeniería de software destinado a guiar a las organizaciones de desarrollo de software en sus esfuerzos para crear un software sólido.

Según la metodología RUP, la vida de un sistema se describe como un número finito de ciclos de desarrollo. Cada ciclo de desarrollo se divide en las cuatro fases del proyecto: inicio, elaboración, construcción y transición. Estas fases, a su vez, se dividen en un número de iteraciones, dependiendo de las necesidades y el tamaño del proyecto. RUP incluye nueve disciplinas que se ejecutan de forma iterativa durante las diferentes fases. Las disciplinas se dividen en disciplinas técnicas y de apoyo. Las disciplinas técnicas incluyen el modelado de negocios, requisitos, análisis y diseño, implementación, pruebas y despliegue. Las disciplinas de apoyo incluyen la gestión de proyectos, gestión de configuración y cambios, y ambiente. Juntos estos últimos proporcionan la infraestructura que cada proyecto necesita para proceder sin problemas en el trabajo de desarrollo. Sin embargo, mientras RUP sugiere un proceso para desarrollar software, los aspectos que rodean las cuestiones de organización y la participación activa del usuario están menos cubiertos (Pilemalm, Lindell, Hallberg, & Eriksson, 2007).

En la figura 2.3 se observa un ejemplo de las relaciones entre las disciplinas y fases como se describe en la versión 2003.06.00 RUP. Las diferentes disciplinas reciben diferente cantidad de atención en función de la fase del proyecto actual.



**Figura 2.3 Rational Unified Process (Pilemalm, Lindell, Hallberg, & Eriksson, 2007).**

**El modelo en V.** Puede ser considerado como una extensión del modelo de cascada, como se muestra en la figura 2.4. En lugar de moverse hacia abajo de manera lineal, las etapas del

procedimiento están doblados hacia arriba después de la fase de codificación, para formar la típica V. El modelo en V establece las relaciones entre cada fase del ciclo de vida del desarrollo y su fase asociada de la prueba. Los ejes horizontales y verticales representan el tiempo o la integridad del proyecto (de izquierda a derecha) y el nivel de abstracción, respectivamente. El modelo en V es simple y fácil de usar. Cada fase tiene resultados específicos. Tiene mayor posibilidad de éxito sobre el modelo de cascada debido al desarrollo temprano de los planes de prueba durante el ciclo de vida. Funciona bien para pequeños proyectos en donde los requisitos son fáciles de entender. Algunas de las deficiencias del modelo en V son (Kumar & Kumar, 2014):

- Muy rígido como el modelo de cascada.
- Poca flexibilidad y el alcance de ajuste es difícil y costoso.
- El software se desarrolla durante la fase de ejecución, por lo que no se producen prototipos tempranos de software.
- Este modelo no proporciona un camino claro para los problemas encontrados durante las fases de prueba.

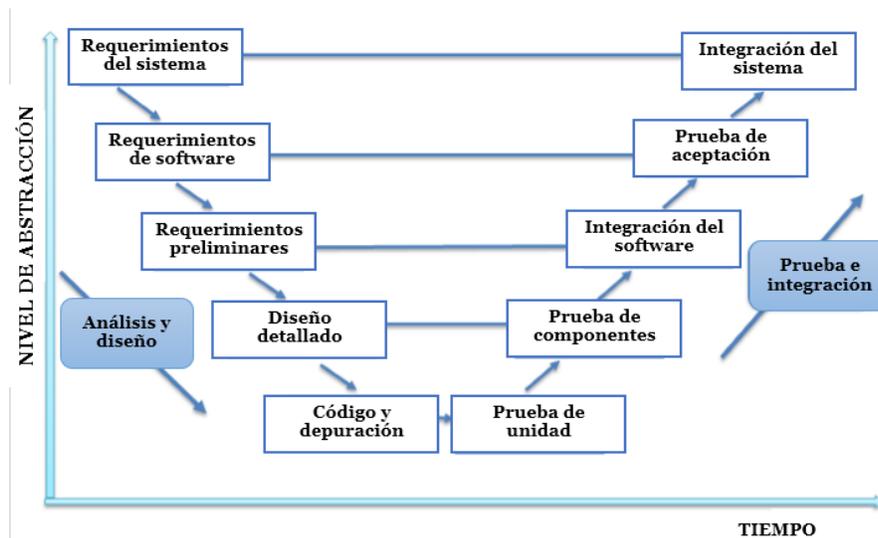


Figura 2.4 Modelo en V.

### 2.1. 2 Desarrollo ágil

Las metodologías ágiles son una opción distinta a los métodos tradicionales, en los cuales los equipos de desarrollo pueden producir software de manera iterativa e incremental. Con esta forma de desarrollo es posible incorporar cambios a los requerimientos a largo del proyecto (Mendes, Estevez, & Fillottrani, 2010).

El “Manifiesto ágil para el desarrollo de software” es un documento en el cual se resume en cuatro postulados y doce principios las mejores prácticas para desarrollar software de forma rápida pero conservando su calidad. Los cuatro postulados principales en que el manifiesto ágil hace énfasis son (Herrera & Valencia, 2007):

**a. Los individuos e interacciones por encima de los procesos y las herramientas:** para garantizar una mayor productividad, las metodologías ágiles valoran el recurso humano como el principal factor de éxito. Reconocen que contar con recurso humano calificado con capacidades técnicas adecuadas, facilidades para adaptarse al entorno, trabajar en equipo e interactuar convenientemente con el usuario, da mayor garantía de éxito que contar con herramientas y procesos rigurosos.

Las metodologías ágiles reconocen que es más importante construir un buen equipo de trabajo que las herramientas y procesos. Procura primero conformar el equipo y que éste defina el entorno más conveniente de acuerdo con las necesidades y las circunstancias.

**b. Software funcionando por encima de la documentación:** los profesionales relacionados con el desarrollo de software, aunque no es su fuerte producir documentos, reconocen su importancia, al igual que reconocen el tiempo y costo de mantener una documentación completa y actualizada. En este sentido, las metodologías ágiles respetan la importancia de la documentación como parte del proceso y del resultado de un proyecto de desarrollo de software, sin embargo, con la misma claridad hacen énfasis en que se deben producir los documentos estrictamente necesarios; los documentos deben ser cortos y limitarse a lo fundamental, dando prioridad al contenido sobre la forma de presentación.

La documentación, en las metodologías ágiles procura mecanismos más dinámicos y menos costosos como son la comunicación personal, el trabajo en equipo, la auto-documentación y los estándares.

- c. La colaboración del cliente por encima de la negociación del contrato:** clásicamente el usuario o cliente es quien solicita e indica qué debe hacer el software, y espera los resultados de acuerdo con sus exigencias o expectativas, en los plazos establecidos. Con frecuencia las dos partes, cliente y equipo de desarrollo, asumen posiciones distantes, con ingredientes de rivalidad y prevención al punto de tener que dedicar tiempo valioso a la tarea de redactar, depurar y firmar el contrato. En este sentido, y complementando el valor que se da al trabajo en equipo, las metodologías ágiles incluyen de manera directa y comprometida al cliente o usuario en el equipo de trabajo. Es un ingrediente más en el camino al éxito en un proyecto de desarrollo de software. Más que un ambiente de enfrentamiento en el cual las partes buscan beneficio propio, evadiendo responsabilidades y procurando minimizar sus riesgos, bajo la filosofía de las metodologías ágiles se pretende lograr el beneficio común, el del equipo de desarrollo y el del cliente. La participación del cliente debe ser constante, desde el comienzo hasta la culminación del proyecto, y su interacción con el equipo de desarrollo debe ser de excelente calidad. Es el cliente quien sabe qué es lo que necesita o desea, el más indicado para corregir o hacer recomendaciones en cualquier momento del proyecto.
- d. La respuesta al cambio por encima del seguimiento de un plan:** dada la naturaleza cambiante de la tecnología y la dinámica de la sociedad moderna, un proyecto de desarrollo de software se enfrenta con frecuencia a cambios durante su ejecución. Van desde ajustes sencillos en la personalización del software hasta cambios en las leyes, pasando por la aparición de nuevos productos en el mercado, comportamiento de la competencia, nuevas tendencias tecnológicas, etc. En este sentido, las metodologías pesadas con frecuencia caen en la idea de tener todo completo y correctamente definido desde el comienzo. No se cuenta entre sus fortalezas la habilidad para responder a los cambios. Por el contrario, en las metodologías ágiles la planificación no debe ser estricta, puesto que hay muchas variables en juego, debe ser flexible para

poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para los siguientes meses.

Los doce principios del manifiesto ágil, que se deben tomar en cuenta en el proceso de desarrollo de software, son las características que diferencian un proceso ágil de uno (Mendes, Estevez, & Fillottrani, 2010):

1. La prioridad es satisfacer al cliente mediante entregas de software tempranas y continuas.
2. Los cambios en los requerimientos son aceptados.
3. El software que funcione se entrega frecuentemente, con el menor intervalo posible entre entregas.
4. El cliente y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. El proyecto se construye en base a individuos motivados.
6. El dialogo cara a cara es el método más eficiente y efectivo para comunicar información dentro del equipo.
7. El software que funcione es la medida principal del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requerimientos y diseños surgen de equipos auto-organizados.
12. El equipo reflexiona en cómo ser más efectivos, y ajusta su comportamiento en consecuencia.

Sommerville (2011) nos dice que hay distintos enfoques para el desarrollo de software rápido, pero que sin embargo tienen en común las siguientes características fundamentales:

1. **Los procesos de especificación, diseño e implementación están entrelazados.** No se cuenta con una descripción a detalle del sistema, y la documentación respecto al diseño

es reducida o generada de manera automática por el entorno de programación que se usa para la implementación del sistema. Tan sólo se documentan las características más significativas del sistema, que son las que conformaran los requerimientos del usuario.

2. **El sistema se desarrolla en diferentes versiones.** Tanto los desarrolladores como el usuario o los usuarios finales son parte importante del equipo de trabajo a lo largo del proyecto, quienes participan se encargan de especificar y evaluar de cada versión. Los integrantes del equipo de desarrollo pueden presentar posibles cambios al software y agregar otros requerimientos que podrían verse implementados en siguientes versiones.
3. **Las interfaces de usuario del sistema se desarrollan usando con frecuencia un sistema de elaboración interactivo, que permita que el diseño de la interfaz se cree rápidamente en cuanto se dibujan y colocan iconos en la interfaz.** En este caso, es posible que el sistema genere tanto una interfaz basada en la Web que va dirigida a un navegador así como una interfaz para una plataforma en específico.

En cuanto a los equipos de desarrollo ágil se refiere, Pressman (2010) indica que es el proceso el que se debe adaptar a las necesidades de los integrantes y del equipo, y no al contrario. Al igual se hace mención que entre los integrantes del equipo debe haber factores claves presentes en un equipo ágil:

- **Competencia.** Esta característica involucra el talento natural, las destrezas con que se cuenta para desarrollar el software y el conocimiento general acerca del proceso elegido por el equipo de trabajo para ser aplicado. Es importante el considerar para todos los integrantes las destrezas así como el conocimiento del proceso.
- **Enfoque común.** Las distintas actividades y tareas que realice cada integrante del equipo de desarrollo deben de ir hacia un mismo fin que es del de entregar al cliente un incremento de software que sea funcional, con la calidad y tiempos establecidos. Para que esto sea, se deben realizar ajustes continuos al proceso, estos pueden ser grandes o pequeños, para que este se adapte a las necesidades del equipo.

- **Colaboración.** La colaboración entre todos los miembros del equipo hace posible la transmisión de información relevante y crucial para el desarrollo del proyecto. Entre las actividades que se logran por medio de la colaboración esta la que corresponde a la ingeniería del software para evaluar, analizar y utilizar la información que se da al equipo de software; la creación de información útil que ayude a cada uno de los integrantes a comprender el trabajo del equipo; y proporcionar al cliente información significativa que le aporte valor del negocio.
- **Habilidad para tomar decisiones.** en el equipo de desarrollo de software debe existir la autonomía y capacidad de tomar decisiones que le den el control del rumbo a seguir en cuanto a asuntos técnicos como del proyecto se refiere.
- **Capacidad para resolver problemas difusos.** El dirigente del proyecto debe considerar que existe una continua imprecisión y cambios a lo cual el equipo ágil debe enfrentarse constantemente. En algunas situaciones, es necesario que el equipo se capacite de comprender y aceptar que el problema que se planteaba en un principio no sea el mismo más adelante. A pesar de lo malo que pudiera parecer esto, el aprendizaje obtenido a partir de la experiencia de cualquier actividad que tenga que ver con solucionar problemas, aun cuando se trate de la resolución del problema incorrecto, será favorecedor en el futuro.
- **Confianza y respeto mutuos.** Para que exista una integración firme entre los miembros del equipo debe haber una atmosfera de confianza y respeto, de esta forma es posible lograr que todos se dirijan hacia una misma meta.
- **Organización propia.** En cuanto a organización propia se refiere y en relación al desarrollo ágil el equipo debe organizarse a sí mismo para la realización del trabajo, establecer y regular el proceso más adecuado al problema a resolver, llevar a cabo la programación de las tareas a realizar con el objetivo de que estas conduzcan a la entrega rápida del incremento de software de manera satisfactoria.

El desarrollo ágil actualmente es importante porque a menudo uno de los requisitos más importantes es el rápido desarrollo y entrega de software. El principal objetivo de estos

métodos ágiles es reducir radicalmente el tiempo de entrega del trabajo de sistemas de software y ser capaces de responder rápidamente a las necesidades cambiantes (Sommerville, 2011).

#### 2.1.2.1 Programación extrema (XP)

XP es la metodología ágil más conocida. Fue desarrollada por Kent Beck buscando guiar equipos de desarrollo de software pequeños o medianos, entre dos y diez desarrolladores, en ambientes de requerimientos imprecisos o cambiantes (Navarro, Fernández, & Morales, 2013). La programación extrema consiste en los siguientes cinco valores (Fojtik, 2011):

1. **Comunicación:** Un gran número de problemas de desarrollo se encuentran en la comunicación incorrecta, no sólo entre los miembros del equipo, sino también con el cliente. Si se usa XP, en varios equipos es común la asignación de un rol especial, normalmente llamado coach, quien detecta fallas en la comunicación y asegura que esta sea correcta. Sin embargo, los equipos pequeños no deben subestimar lo bueno de la comunicación continua. La comunicación frecuente es importante tanto entre los desarrolladores como para el cliente.
2. **Simplicidad:** La metodología pretende el desarrollo de un software tan fácil como sea posible, y no tratar con funcionalidades que no sean actualmente importantes y que podrían ser utilizados en el futuro. La metodología XP dice que no se debe crear una arquitectura más robusta de lo necesario por el momento.
3. **Retroalimentación (Feedback):** Este aspecto es muy importante para el correcto desarrollo. Se ejecuta en varios niveles. Uno de ellos es la prueba, que debe realizarse en todas las etapas de desarrollo y no después de la etapa de implementación.
4. **Coraje:** Un valor muy importante de XP es el valor de corregir y eliminar los errores a toda costa. Incluso puede significar la eliminación de una gran parte del código o hasta fundamental volver a realizar el diseño arquitectónico. Los desarrolladores pueden llegar a creer que al eliminar una gran parte del código son señales de fracaso y por lo que son menos propensos a probar más, superar estas cuestiones sobre todo de orgullo es muestra de coraje.

5. **Respeto:** Los miembros del equipo deben estar interesados en el trabajo de sus colegas. En el caso de las personas que prefieren trabajar solos, sin una estrecha relación con los demás, XP será inutilizable. Este valor XP corresponde estrechamente con el énfasis en la comunicación.

Fojtik (2011) menciona que la metodología XP es flexible y que consta de las actividades básicas que se describen a continuación (Pressman, 2010) y se pueden ver en la figura 2.5:

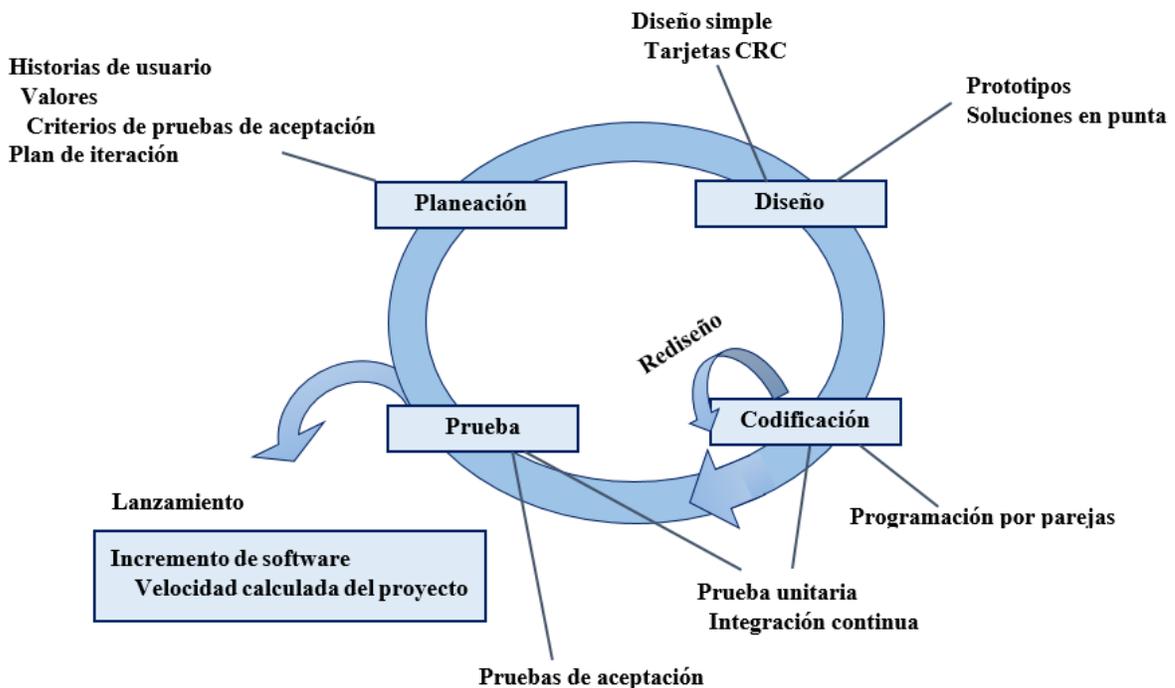


Figura 2.5 El proceso de programación extrema.

1. **Planificación y Gestión.** Esta actividad tiene que ver con la obtención de requerimientos, los cuales son necesarios para comprender el contexto en que se va a desarrollar el software, así el equipo será capaz de generar las salidas solicitadas que cumplan con las características y funcionalidades requeridas.
2. **Diseño.** El diseño debe ser sencillo y es preferible en lugar de uno complejo, debido a que la metodología XP sigue muy estrictamente el principio MS (mantenlo sencillo), esto es porque el diseño es el que conduce la implementación de una historia de

acuerdo a como ésta se vaya escribiendo. No se apoya el diseñar funcionalidades extras, puesto que el desarrollador considera que se necesitarán más adelante.

3. **Codificación.** Una vez desarrolladas las historias, concluido el diseño previo y antes de comenzar con la codificación primero se desarrollan un conjunto de pruebas unitarias para cada una de las historias que serán incluidas en los incrementos de software. El elaborar las pruebas unitarias permite al desarrollador entender mejor lo que debe realizar para la implementación, pues conoce lo que se requiere para pasar la prueba. Nada fuera de lo establecido debe ser añadido, recordando el principio MS. Finalmente, cuando la codificación es concluida, es necesario el revisarla a través de la prueba unitaria, así se consigue retroalimentación al momento, útil para los desarrolladores.
4. **Prueba.** Como se mencionó en el punto anterior, el generar pruebas unitarias es un factor clave en la metodología XP. Dichas pruebas tienen que ser implementadas haciendo uso de una estructura que pueda automatizarlas, con el objetivo de que se ejecuten en reiteradas ocasiones de forma fácil. Con esto se induce hacia una estrategia de pruebas de regresión siempre que se cambie el código.

### 2.1.2.2 SCRUM

Es un método de desarrollo ágil y quizás uno de los más utilizados también. Se argumenta que Scrum funciona mejor para los equipos de 5-7 personas, pero la mayoría de las técnicas se pueden utilizar para desarrolladores individuales igual (Blom, 2010). El desarrollo dentro de la metodología Scrum se caracteriza por ser simple pero que al mismo tiempo se necesita trabajo duro, puesto que la gestión está basada en la adaptación continua con respecto a las circunstancias de la evolución del proyecto y no a través de un plan (Palacio, 2007). La diferencia principal entre el enfoque definido (cascada, espiral e iterativo) y el enfoque empírico (SCRUM) es que el enfoque de SCRUM asume que el análisis, el diseño y los procesos de desarrollo en la fase de Sprint son impredecibles (figura 2.). Un mecanismo de control se utiliza para gestionar la imprevisibilidad y controlar el riesgo. Flexibilidad, capacidad de respuesta, y la fiabilidad son los resultados. SCRUM tiene los siguientes grupos de fases (Schwaber, 1995):

**1. Antes del juego (Pregame)**

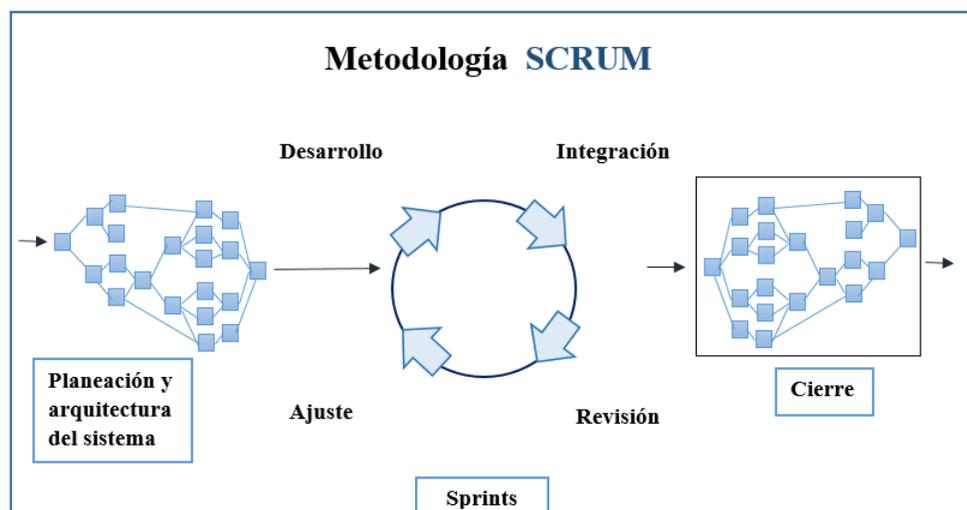
- Planificación: Definición de una nueva versión basada en el actual backlog, junto con una estimación de su horario y costo. Si un nuevo sistema está siendo desarrollado, esta fase consiste tanto en la conceptualización como en el análisis. Si un sistema existente se está mejorando, entonces esta fase consiste en un análisis limitado.
- Arquitectura: Es el diseño de cómo se implementarán los elementos del backlog. Esta fase incluye la modificación de la arquitectura del sistema y un diseño de alto nivel.

**2. El juego (Game)**

- Desarrollo de los Sprints: Se trata del desarrollo de nuevas funcionalidades para liberación, con el constante respeto a las variables de tiempo, a los requisitos, la calidad, el coste y la competencia. La interacción con estas variables define el final de esta fase. Existen múltiples sprints iterativos de desarrollo, o ciclos, que se utilizan para evolucionar el sistema.

**3. Después del juego (Postgame)**

- Cierre: Preparación para la liberación, incluyendo la documentación final, efectuado pruebas, y la liberación preliminares.



**Figura 2.6 Metodología SCRUM.**

En la siguiente tabla se muestra en resumen los términos más importantes de esta metodología, con las que se comprenderá mejor esta metodología (Sommerville, 2011).

**Tabla 2.2 Terminología SCRUM.**

Termino Scrum	Definición
<b>Equipo de desarrollo</b>	Grupo auto-organizado de desarrolladores de software, no mayor a 7 personas, quienes son responsables de desarrollar el software y otros documentos esenciales del proyecto.
<b>Producto incremental potencialmente entregable</b>	Una reunión diaria del equipo de Scrum que revisa el progreso y prioriza el trabajo a realizar ese día. Idealmente, esto debería ser una reunión corta, cara a cara, que incluya a todo el equipo.
<b>Producto backlog</b>	Es una lista de lo que se debe hacer, y que el equipo Scrum debe abordar. Pueden ser definiciones de características o requisitos de software, historias de usuario o descripciones de tareas complementarias necesarias, como la definición de la arquitectura o la documentación del usuario
<b>Dueño del producto</b>	Un individuo (o quizá un grupo pequeño) cuyo trabajo es identificar las características o requisitos del producto, dar prioridad a éstos para el desarrollo y revisar continuamente el producto backlog para garantizar que el proyecto satisface las necesidades críticas del negocio. El dueño del producto puede ser un cliente, o podría ser un gerente del producto en una empresa de software u otro representante de las partes interesadas.
<b>Scrum</b>	Una reunión diaria del equipo de Scrum que revisa el progreso y prioriza trabajo a realizar ese día. Idealmente, esto debería ser una reunión corta, cara a cara, que incluye todo el equipo.
<b>ScrumMaster</b>	Es el responsable de asegurar que el proceso de Scrum se siga y guía al equipo en el uso efectivo de Scrum. Es responsable de la interconexión con el resto de la compañía y de asegurar que el equipo Scrum no es desviado por interferencia exterior. Los desarrolladores de Scrum insisten en que el ScrumMaster no debe ser considerado como un jefe de proyecto. Para otros, sin embargo, no siempre resulta fácil ver la diferencia.
<b>Sprint</b>	Es una iteración de desarrollo. Sprints son generalmente de 2-4 semanas.
<b>Velocidad</b>	Una estimación de la cantidad de esfuerzo de producto backlog que un equipo puede cubrir en un solo sprint. La comprensión de la velocidad de un equipo ayuda a estimar la prontitud en que puede ser cubierto un sprint y proporciona una base para medir la mejora del rendimiento.

### 2.1.2.3 Crystal

Crystal es una metodología de ágil de software desarrollada por Alistair Cockburn en el que el equipo está más enfocado al desarrollo de software en lugar de centrarse en las herramientas o procesos. Los métodos crystal son un conjunto de herramientas de elementos de la metodología para adaptarse a los distintos proyectos. Los proyectos de gran tamaño o de seguridad crítica requieren más elementos de la metodología que los proyectos pequeños no críticos. Con los métodos de crystal, las organizaciones sólo desarrollan y utilizan la metodología tanto como demandan sus necesidades de negocio. Básicamente esto se utiliza para los equipos pequeños y pequeños proyectos que no tienen una vida crítica (Kumar&Kumar, 2012).

No existe una única metodología Crystal puesto que su creador, Alistair Cockburn, cree que tipos diferentes de proyectos requieren tipos diferentes de metodologías; y que cada equipo debe, a partir de estas características comunes, crear nuevos miembros de la familia (Barroso, Oliveros, García, Y., & Coello, 2008).

Las siguientes son características de Crystal (Ramírez & Flórez, 2014):

- La documentación tiene menos relevancia en comparación con otras metodologías más tradicionales.
- Busca que las versiones del proyecto que se desarrollen sean probadas.
- Se encuentra fundamentada en SCRUM.
- Las pruebas son un punto muy importante.

Las propiedades o valores de Crystal son siete y se mencionan a continuación (Garzías, 2012):

1. **Entregas frecuentes**, realizadas a partir de iteraciones e incrementos. Los tiempos de entrega pueden variar desde semanales o trimestrales dependiendo de proyecto del que se trate.
2. **Mejora reflexiva**. Se refiere a una mejora continua. Esta se logra en base a las iteraciones, las cuales permiten ir ajustando el proyecto, lo que conduce hacia la mejora.

3. **Comunicación osmótica.** Todo el equipo reunido en un mismo sitio, de esta manera se logra una comunicación directa, es decir, cara a cara.
4. **Seguridad personal.** Libertad de expresar opiniones sin miedo alguno, y que estas sean tomadas en cuenta.
5. **Enfoque.** A partir de objetivos y prioridades bien definidas y claras, establecer periodos en los que no se interrumpa al equipo para que lleven a cabo la realización de actividades concretas en un entorno físico adecuado.
6. **Fácil acceso a usuarios expertos.** En comparación con otras metodologías, como es el caso de XP, en Crystal no se requiere que el cliente este constantemente con el equipo de desarrollo, así que sólo es necesario que se encuentre disponible en las reuniones semanales que como mínimo deben realizarse.
7. **Entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua.** Varios equipos ágiles compilan e integran varias veces al día.

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

Test Driven Development (Desarrollo Dirigido por Pruebas) es una disciplina iterativa e incremental de diseño y programación, en la cual se escribe una línea nueva de código como respuesta a una prueba no aprobada o fallida que se ha programado previamente. Lo que se persigue es generar código adecuado funcional (Araújo, 2007).

Esta metodología, como se aprecia en la figura 2.7, es un enfoque para el desarrollo de software en el que se escriben primero las pruebas y luego el desarrollo real se hace en iteraciones sobre la base de dichas pruebas. Cada iteración produce código necesario para pasar las pruebas de esa iteración. Por último, el programador o equipo refactoriza el código para acomodar cualquier cambio (Kumar & Kumar, 2014).

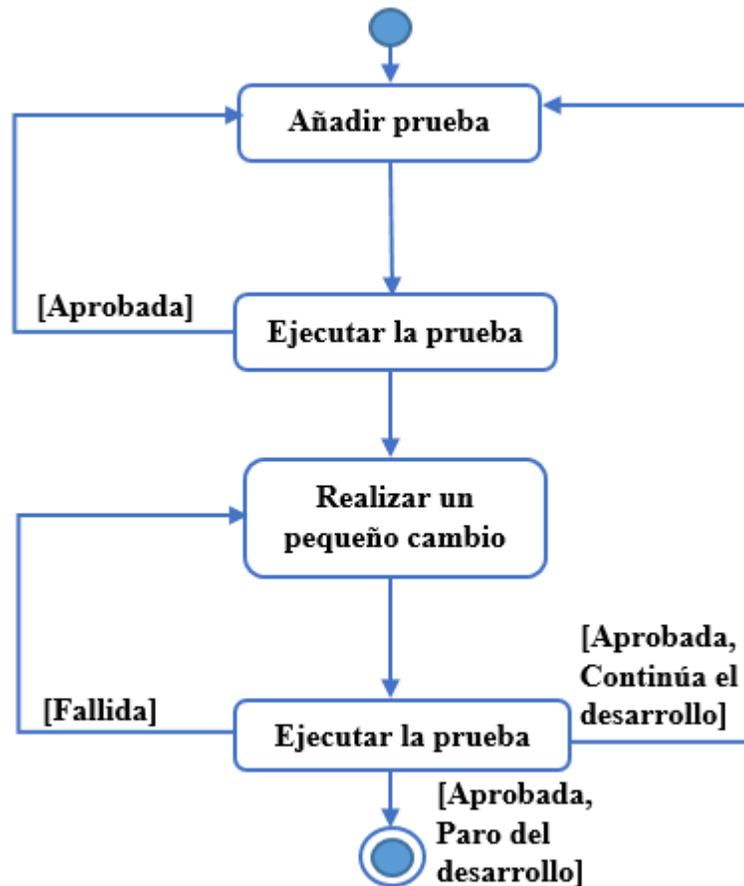


Figura 2.7 Modelo Test Driven Development.

## 2.2 Desarrollo global de software (GSD)

El Desarrollo Global de Software es una tendencia en aumento en los últimos años, donde las fuerzas de trabajo se encuentran distribuidas, con lo que se produce software total o parcialmente en ubicaciones distintas de donde se localiza el cliente final, según lo menciona en su trabajo Minoli (2010). Esta tendencia (DGS) se consolidó como uno de los aspectos más relevantes en la investigación y práctica de la Ingeniería del Software en la década de 2010. Apareció hace un poco más de 20 años con las primeras prácticas de outsourcing, pero se oficializa como tal en 2006 con la celebración de la primera conferencia internacional sobre desarrollo global ICGSE (IEEE International Conference on Global Software Engineering) (Vizcaíno, García, & Piattini, Visión General del desarrollo Global de Software, 2014).

### 2.2.1 Prácticas ágiles en el GSD

Aunque los métodos ágiles son muy adecuadas cuando los clientes y desarrolladores están localizados en un lugar común y hay una interacción frecuente entre ellos, varias organizaciones de software han publicado su experiencia con éxito de la incorporación ágil de desarrollo de software distribuido. Sin embargo, existen desafíos asociados con esta combinación, y para conseguir que funcione con eficacia es necesario un esfuerzo considerable. Las mayores dificultades se resumen en lo relacionado con la comunicación, el personal, la cultura, los diferentes husos horarios, la confianza y la gestión del conocimiento. Sin embargo, diversas tácticas y soluciones también son reportadas por diferentes organizaciones de software para mitigar estos desafíos. (Jalali & Wohlin, 2010)

En el trabajo realizado por Paasivaara & Lassenius (2006) se presentan dos combinaciones entre métodos ágiles y el desarrollo distribuido, indicando la manera en que podrían emplearse:

- **Programación extrema distribuida.** Los principios de XP podrían ser útiles en proyectos distribuidos, a pesar de que se necesita una gran cantidad de trabajo en el diseño de cómo aplicar dichos principios en entornos distribuidos.
- **Scrum distribuido.** Puede llegar a ser posible el aplicar principios de Scrum a proyectos distribuidos cuando se presta suficiente atención en la planificación de cómo organizar una comunicación frecuente con eficiencia.

### 2.2.2 Desafíos dentro del GSD

Existen tres tipos de distancias que afectan directamente al DGS, en los cuales podemos agrupar los principales obstáculos a los que se enfrenta (Holmstrom, Fitzgerald, Agerfalk, & Conchuir, 2006):

- **Distancia temporal.** Hace referencia al tiempo. La distancia temporal es difícil cuando se trata de controlar los proyectos que constituyen diferentes sitios, debido a los distintos husos horarios. Además de los problemas de control, la distancia temporal desafía la comunicación y coordinación dentro y entre los equipos. En particular, los

retrasos de respuesta pueden ser vistos como frustraciones. Claramente, la comunicación y la coordinación son desafíos causados por la distancia temporal. El principal problema es la demora en las respuestas. Además, el control del proyecto es más difícil cuando la reducción de tiempo se superpone.

- **Distancia geográfica.** Se trata del espacio físico existente entre distintas ubicaciones. A pesar de que hay una ventaja con respecto a la fuerza intelectual (es decir, la capacidad de reclutar al mejor personal de distintos lugares donde la educación y el empleo es más competitivo), las empresas experimentan problemas relacionados con la distancia geográfica. El establecimiento de un sentimiento de confianza y pertenencia, es decir, de integración al equipo, puede ser difícil. A pesar de las tecnologías de comunicación como el correo electrónico, las reuniones virtuales, las redes sociales, el teléfono, entre otras aún puede haber la sensación de ser equipos diferentes.
- **Distancia sociocultural.** La distancia sociocultural es una dimensión compleja que implica la cultura organizacional, la cultura nacional y el lenguaje, la política, las motivaciones individuales y la ética de trabajo. Además, las diferencias culturales, políticas y religiosas pueden cuestionar el trabajo dentro proyecto. Claramente, la distancia sociocultural es una dimensión compleja que puede desencadenar malentendidos y confusión como resultado de problemas de lenguaje e interpretación. Esto tiene implicaciones para la comunicación, la coordinación y el control y hace que sea un reto real para crear un entendimiento mutuo entre los equipos.

### 2.3 Ingeniería web

En 1998, se estableció la ingeniería Web como una nueva disciplina, junto con Yogesh Deshpande y Steve Hansen de la Universidad de Western Sydney, Australia (Ginige & Murugesan, 2001).

La Ingeniería Web se refiere a la aplicación de enfoques sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de aplicaciones basadas en Web. (Deshpande, y otros, 2002).

### 2.3.1 Métodos de la ingeniería web

En esta sección revisaremos algunas metodologías para la construcción de software basado en la web:

**UWE (UML-based Web Engineering).** Es una metodología enfocada al desarrollo de aplicaciones que se fundamenta en la utilización de UML (Unified Modeling Language) para sus modelos y métodos en el proceso de desarrollo. Esta metodología especifica de manera clara cada uno de los elementos del modelo. UWE considera las etapas y modelos siguientes (Nieves, Ucán, & Menéndez, 2014):

- Análisis de requisitos. Se definen los requerimientos funcionales para la aplicación web a través del empleo de un modelo de casos de uso.
- Modelo de contenido. Especifica los conceptos correspondientes a la aplicación de forma detallada, utilizando para este fin un diagrama de clases.
- Modelo de navegación. Muestra la navegación de los objetos en la aplicación, así como una serie de estructuras (índices, menús y consultas).
- Modelo de presentación. Permite mostrar las interfaces de usuario a través de vistas abstractas
- Modelo de proceso. Proporciona una representación de la apariencia de las actividades que se conectan con cada clase de proceso

**Modelo RRM (Relationship Management Methodology).** En Lamarca (2013) se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia. Esta metodología es apropiada para dominios con estructuras regulares (es decir, con clases de objetos bien definidas, y con claras relaciones entre esas clases).

El modelo propone un lenguaje que permite describir en un diagrama Entidad-Relación los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipermedia de la aplicación. Los objetos del dominio se definen con la ayuda de entidades, atributos y

relaciones asociativas. El modelo introduce el concepto de *slice* (trozo) con el fin de modelar los aspectos unidos a la presentación de las entidades. Un *slice* corresponde a un subconjunto de atributos de una misma entidad destinados a ser presentados de forma agrupada. La navegación se modela con la ayuda de primitivas de acceso, enlaces estructurales (unidireccional y bidireccional) que permiten especificar la navegación entre *slices*, y visita guiada condicional, índice condicional y agrupación, que permiten especificar la navegación entre entidades. El esquema completo del dominio y de la navegación de la aplicación se denomina esquema RMDM<sup>1</sup> y se obtiene como resultado de las tres primeras etapas del método. Las etapas son (Lamarca, 2013):

- **Primera etapa:** Representar los objetos del dominio con la ayuda del modelo Entidad-Relación ampliado con relaciones asociativas (aquéllas que permiten representar caminos navegacionales entre entidades puestos en evidencia en la fase de análisis).
- **Segunda etapa:** Determinar la presentación del contenido de las entidades de la aplicación así como su modo de acceso. El esquema obtenido como resultado de esta etapa se denomina esquema E-R+. Se trata de un esquema Entidad-Relación en el que cada entidad ha sido reemplazada por su esquema de entidad. Un esquema de entidad está constituido por nodos (los trozos o *slides*) unidos por relaciones estructurales.
- **Tercera etapa:** Definir los caminos de navegación inducidos por las relaciones asociativas del esquema E-R+. A continuación, es posible definir estructuras de acceso de alto nivel (agrupaciones), lo que permite dotar a la aplicación de accesos jerárquicos a niveles diferentes de los trozos de información. El esquema RMDM resultante se obtiene añadiendo al esquema E-R+ las agrupaciones y caminos navegacionales definidos en esta etapa.

Las cuatro etapas restantes consisten en:

- Definición del protocolo de conversión de elementos del diagrama RMDM en objetos de la plataforma de desarrollo.
- Concepción del interfaz usuario.

---

<sup>1</sup>Relationship Management Data Model

- Concepción del comportamiento en ejecución.
- Construcción del sistema y test.

**Modelo OOHDM (Oriented Hypermedia Design Methodology).** Desde que la web está basada en el principio de diseño de hipertexto, podemos utilizar las directrices de diseño basados en la web para apoyar la hipermedia y las funcionalidades de navegación. Al separar las distintas partes del desarrollo basado en la web, podemos reducir la complejidad del problema, mejorar la capacidad de reutilización, y acortar el periodo de desarrollo. De acuerdo con OOHDM, el desarrollo de aplicaciones hipermedia se divide en cuatro pasos, incluyendo el diseño conceptual, diseño de navegación, diseño de interfaces abstractas e implementación. Estas actividades se llevan a cabo como una mezcla de métodos iterativos e incrementales (Karimpoor, Isazadeh, & Sadighi, 2008):

- **Diseño conceptual.** Durante esta actividad, construimos un esquema conceptual que representa los objetos, sus relaciones y colaboraciones existentes en el dominio de destino. Tenemos el modelo del dominio de destino utilizando el Lenguaje de Modelado Unificado (UML), una inherentemente herramienta de modelado orientado a objetos, en forma de clases y asociaciones entre ellos. El modelado orientado a objetos utiliza la agregación, la generalización / especialización, y un concepto de embalaje, llamados subsistemas, para mejorar la reutilización y acortar el tiempo de desarrollo.
- **Diseño de navegación.** La navegación es un aspecto importante en el desarrollo de aplicaciones basadas en web, por lo que el usuario debe ser capaz de llegar a la parte deseada del programa con sólo tres clics. Por desgracia, la mayoría de las organizaciones no prestan suficiente atención a las funciones de navegación e hipermedia al desplegar sus aplicaciones web y esto confunde a los usuarios, por lo que no pueden alcanzar los objetivos deseados. Y en algunas aplicaciones, a pesar de que los desarrolladores han considerado para la navegación una estructura de navegación coherente no se aplica a toda la aplicación, por lo que el sitio se convierte en un enorme conjunto de enlaces y nodos despeinados.

- En OOHD, la navegación se construye como una vista de un modelo conceptual, lo que permite la construcción de diferentes modelos de acuerdo a los perfiles de los diferentes usuarios. El diseño de navegación se expresa en dos esquemas, el esquema de la clase de navegación y el esquema de contexto navegacional. En OOHD, hay un conjunto de tipos predefinidos de clases de navegación: nodos, enlaces y estructuras de acceso. Los nodos de navegación, que se construyen como vistas de los objetos conceptuales, se denominan objetos de navegación. Si algunas de las clases en el modelo conceptual no están relacionadas con la navegación, se pueden eliminar en los objetos de navegación o ser convertidas en atributos en otras clases.
- El diseño de navegación debe considerar la forma en que el usuario explora el espacio hipertexto. Por desgracia, los nodos y enlaces no son suficientes para cumplir con este objetivo, este es el punto en el que aparecen los contextos de navegación. El propósito de la definición de contextos es hacer posible el clasificar los objetos de navegación de muchas maneras diferentes.
- **Diseño de interfaces abstractas.** Con el tiempo, lo que el usuario ve es la interfaz. Incluso los objetos de navegación no son visibles directamente por el usuario, pero se presentan al usuario mediante la interfaz abstracta. Para un solo modelo de navegación, podemos crear diferentes interfaces, por lo que los usuarios pueden acceder a la aplicación desde diferentes dispositivos, como teléfonos móviles, o diversos tipos de navegadores.
- **Implementación.** Cualquier modelo específico de la aplicación, como el diseño OOHD, debe ser finalmente implementado utilizando una tecnología de aplicación.

### 2.3.2 Aplicaciones web

Una aplicación web es un programa de software que se ejecuta en un servidor web. A diferencia de las aplicaciones de escritorio tradicionales, que son lanzadas por el sistema operativo, las aplicaciones web deben ser accesibles a través de un navegador web. Las aplicaciones web tienen varias ventajas sobre las aplicaciones de escritorio (Christensson, 2014):

- Se ejecutan dentro de los navegadores web, los desarrolladores no necesitan desarrollar aplicaciones web para múltiples plataformas.
- Los desarrolladores no necesitan para distribuir actualizaciones de la aplicación web a los usuarios. Mediante la actualización de la aplicación en el servidor, todos los usuarios tienen acceso a la versión actualizada.
- Desde el punto de vista del usuario, una aplicación web puede proporcionar una interfaz de usuario más consistente a través de múltiples plataformas, a causa de esto la apariencia depende del navegador en lugar del sistema operativo.
- Los datos que se introducen en una aplicación web se procesan y se guardan de forma remota. Esto le permite acceder a los mismos datos desde múltiples dispositivos, en lugar de transferir archivos entre sistemas informáticos.

Mientras que las aplicaciones web ofrecen varias ventajas, Christensson (2014) menciona que tienen algunas desventajas en comparación con las aplicaciones de escritorio:

- Dado que no se ejecutan directamente desde el sistema operativo, tienen un acceso limitado a los recursos del sistema, tales como la CPU, memoria y el sistema de archivos. Por lo tanto, los programas de gama alta, tales como la producción de video y otras aplicaciones de medios funcionan mejor como aplicaciones de escritorio.
- Las aplicaciones web también dependen completamente del navegador web. Si el navegador se bloquea, por ejemplo, es posible que se pierda el progreso no guardado. Además, las actualizaciones del navegador puede que no sean compatibles con las aplicaciones web, creando problemas inesperados.

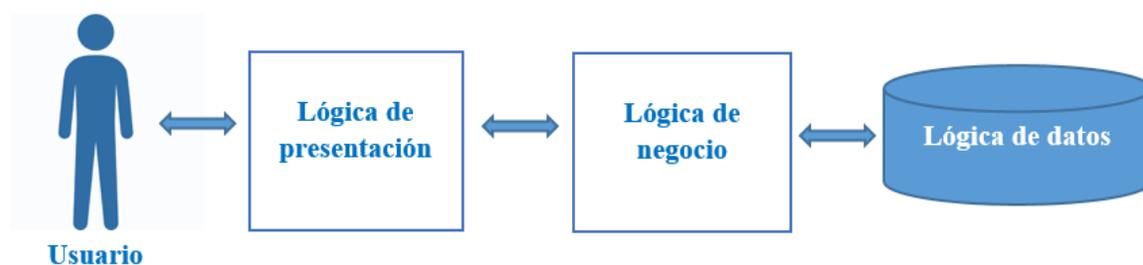
Algunas personas prefieren las aplicaciones de escritorio, mientras que otros prefieren las aplicaciones web. Por lo tanto, muchas compañías de software ofrecen ahora dos versiones de escritorio y web de sus programas más populares (Christensson, 2014).

### 2.3.2.1 Arquitectura cliente/servidor

Las aplicaciones web, Luján (2001) nos dice que tratan de un tipo especial de aplicaciones cliente/servidor, que es no es otra cosa que una arquitectura. La arquitectura cliente/servidor es una arquitectura de red, donde cada ordenador o proceso dentro de la red puede ser cliente o tratarse de un servidor. Comúnmente, si nos referimos a servidores estamos hablando de aquellos ordenadores que están dedicados a la gestión de recursos; ahora bien, en cuanto de clientes se trata estos son normalmente computadores menos potentes en comparación con los servidores, los cuales hacen uso de los recursos que ofrecen los dichos servidores. Esta arquitectura involucra la presencia de una relación dada entre los procesos que piden servicios (clientes) y los procesos que dan respuesta a estos servicios (servidores). Los dos procesos mencionados pueden ser ejecutados en el mismo procesador o en diferentes.

En la arquitectura cliente/servidor es posible la separación de funciones entre niveles, la figura 2.8 representa tal separación (Luján, 2001):

- **Lógica de presentación.** La presentación de los datos es una función independiente del resto.
- **Lógica de negocio (o aplicación).** Los flujos de trabajo pueden cambiar según las necesidades existentes de un procesador a otro.
- **Lógica de datos.** La gestión de los datos debe ser independiente para poder ser distribuida según las necesidades de la empresa en cada momento.



**Figura 2.8 Separación de funciones.**

Los conceptos que se analizaron en este capítulo sirven como referencia crucial para la definición de la metodología propuesta, aun así en el capítulo 3 son retomados de nuevo para realizar un análisis, el cual representa un aporte para al fin propuesto.

## Capítulo 3 Marco metodológico y desarrollo de la propuesta

En este capítulo se pretende realizar una metodología para el desarrollo global de software en aplicaciones web, que permita comprender los elementos que debe contener una aplicación web desarrollada por equipos que se encuentran geográficamente distantes. Se basa en procesos de desarrollo del software, en herramientas que posibilitan la comunicación a través de internet y técnicas de desarrollo global de software.

En base a las necesidades que se tienen para el desarrollo de la metodología propuesta se definieron las etapas siguientes:

- Prospección de los modelos y metodologías existentes.
  - Metodologías de desarrollo e ingeniería del software
  - El trabajo colaborativo como factor de éxito en DGS:
    - Reuniones distribuidas.
    - Tecnologías de la información y comunicación para la comunicación a través de internet.
  - Metodologías de desarrollo e ingeniería web.
  - Herramientas para el desarrollo de aplicaciones web.
- Desarrollo de la metodología propuesta.

### 3.1. Prospección de los modelos y metodologías existentes

En esta sección se lleva a cabo una revisión y análisis de las metodologías ágiles de ingeniería del software y de las metodologías de ingeniería web. También se revisan algunas herramientas de comunicación que permitan la colaboración entre equipos de trabajo separados geográficamente, así como herramientas para el desarrollo de aplicaciones web y técnicas que permitan el trabajo colaborativo.

#### 3.1.1 Metodologías de desarrollo e ingeniería del software

Podemos apreciar una comparación entre las metodologías ágiles y tradicionales que hace Kumar & Kumar (2014) en la tabla 3.1, las cuales fueron descritas a detalle en el Capítulo 2.

**Tabla 3.1 Comparativa de Metodologías de Desarrollo de Software (Kumar & Kumar, 2014).**

Parámetro de la metodología	Metodologías	
	Tradicionales	Ágiles
Requerimientos	Definidos antes de la implementación	Adquiridos de forma iterativa
Costos de re-trabajo	Alto	Bajo
Flexibilidad del diseño	Difícil de lograr	Fácilmente alcanzable
Dirección del desarrollo	Fijo	Flexible
Detección de fallas	Problemático	Fácil
Pruebas	Después de la codificación	Sobre cada iteración
Implicación del cliente	Bajo	Alto
Habilidad de los desarrolladores	Nada en particular	Las habilidades interpersonales y conocimientos básicos del negocio
Tipo de proyecto	A gran escala	Baja y mediana escala
Reusabilidad	Difícil de lograr	Fácilmente alcanzable
Satisfacción del cliente	Bajo	Alto
Desempeño	No es alto	Alto

Es importante hacer mención de que existen características deseables en una metodología, las cuales son consideradas por ciertos autores (Reyes, 2007):

- Existencia de reglas predefinidas.
- Cobertura total del ciclo de desarrollo.
- Verificaciones intermedias.
- Planificación y control.
- Comunicación efectiva.
- Utilización sobre un abanico amplio de proyectos.
- Fácil formación.
- Herramientas CASE<sup>2</sup>.
- Actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento.
- Soporte de la reutilización de software.

<sup>2</sup>*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora: Son diversas aplicaciones informáticas destinados a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero.

### 3.1.2 El trabajo colaborativo como factor de éxito en DGS

El trabajo colaborativo se refiere al que se efectúa a través de la cooperación voluntaria que se da entre dos o más personas o grupos de personas quienes realizan un esfuerzo en conjunto para alcanzar un objetivo común, lo que proporciona beneficios para todos los involucrados. Aunque dichos beneficios no sean de igual proporción debido a las diferencias y desigualdades que existen entre los participantes. Con el uso de las TIC es viable trabajar de manera no presencial, haciendo uso de dispositivos conectados a internet, con lo que es posible el compartir conocimientos y cooperar en la producción de bienes y servicios. Los dispositivos que se conectan a internet cuentan con cualidades que proporcionan un gran potencial de uso colaborativo (Silva & Reygadas, 2013):

- Tienen la capacidad de manejar grandes volúmenes de información con un costo considerablemente bajo.
- Facilitan la distribución de cualquier tipo de información.
- Difunden la información a diferentes personas, quienes interactúan con ella al recibirla.
- Permiten que la información sea dirigida a varias direcciones.
- Otorgan flexibilidad y acortan las distancias geográficas.

El término que engloba al trabajo colaborativo es *groupware*, el cual surge en los años 80, para referirse al entorno de trabajo en que los integrantes en grupo colaboran y se apoyan para alcanzar un objetivo, haciendo uso de las tecnologías que ofrecen las TIC (Honmy, 2008).

La comunicación entre los integrantes del *groupware*, siendo ésta un elemento clave que determina el éxito o fracaso de un proyecto como lo menciona Piorun & Giusti (2001), se encuentra en una situación vulnerable ante el factor distancia que se presenta al desarrollar software de manera distribuida.

Es a través de la comunicación cómo las personas se relacionan entre sí, esta actividad intervienen tres elementos principales que son a) emisor, b) receptor y c) canal. Cualquier interferencia que haga que el proceso de comunicar no se efectúe satisfactoriamente se le

conoce como ruido, los cuales pueden generarse por distintas razones que pueden ser desde problemas relacionados con el emisor, el canal o el receptor. En cuanto de proyectos se trata los ruidos pueden incrementarse, ya que lo que se busca comunicar involucra cambios que perjudican la tarea laboral y personal de la gente. Las personas muestran desagrado a aquello que desconocen y será con los esfuerzos hechos por la comunicación del proyecto el poder conseguir que el desagrado se lo menor posible.

Cuando se comunica una información es común que todos los integrantes tengan una versión distinta de la realidad, lo que da comienzo a una serie de conflictos produciendo problemas en la comunicación, como se resume en la tabla 3.2.

**Tabla 3.2 Problemas en la comunicación y alternativas de solución.**

Problema	Solución
<b>La gente no escucha</b>	Mecanismos de ida y vuelta. Implican el reafirmar lo escuchado, lo que tiene valor tanto para el emisor como el receptor. Cerrar los aspectos específicos de lo cordado.
<b>Comunicación no verbal. Dejar de lado la carga emotiva de los participantes.</b>	Se debe aprender a <i>escuchar sentimientos y emociones</i> , para tal fin es posible realizar ejercicios de empatía para ahondar en la causas de la motivación negativa.
<b>Diferentes lenguajes. Distintos significados para un mismo concepto lo que desencadena una barrera de comunicación profunda.</b>	Manifiestar de manera clara al inicio del intercambio, los objetivos comunes a lograr y un pequeño grupo de códigos comunes consiguiendo así un microlenguaje acorde a la situación, generando a su vez un sentimiento de pertenencia y motivación grupal.
<b>Abundancia de detalles (en muchos casos no es la mejor información)</b>	Establecer un plan de comunicaciones. Definir qué información requiere cada persona interna o externa. El mayor impacto en la gente no es por medio de las palabras sino con los hechos.

Al realizar el plan de comunicación se deben tomar en cuenta los siguientes puntos (Piorun & Giusti, 2001):

- Definir a quiénes van dirigidas las comunicaciones.
- Definir qué es lo que se comunica a cada uno.

- Definir cuándo se realiza cada comunicación.
- Definir como se realiza la comunicación. (memo, mail, circular, etc.).
- Definir origen de datos para el armado de cada comunicación.
- Definir la estructura que tendrá cada una.
- Definir la forma de distribución más confiable, directa y rápida.
- Definir los responsables del equipo para la ejecución de este plan.

### 3.1.2.1 Las reuniones en DGS

Las reuniones distribuidas dentro del DGS se pueden realizar de diferentes maneras, si es los horarios de los quipos no coinciden, como expresa Vizcaíno, García, & Piattini (2014) en el libro Desarrollo Global de Software:

- Reuniones mediante comunicación: En este método de reunión deben asistir la mayor parte de los miembros a una hora en que la mayoría pueda, todo debe quedar muy bien documentado para aquellos que no puedan asistir, así se enteren de lo que trató la reunión. Se considera la forma menos efectiva de manejar las reuniones distribuidas
- Enfoque de enlace: Para llevar a cabo este tipo de reuniones se requiere de un vínculo, pues se realizan dos reuniones en horarios distintos y el vínculo debe distribuir la información entre los participantes de las dos zonas.
- Alterar el estado de las reuniones: Como el nombre lo dice se alternan las reuniones en base a los distintos horarios de los equipos, es decir, que en ocasiones las reuniones se realizaran en el horario de trabajo de un equipo, y en otras ocasiones en los horarios de otro equipo.
- “Compartir el dolor”: Establecer una hora en la que debe llevarse a cabo la reunión la cual sea conveniente para todos. Se trata de llegar a un acuerdo para que así cada equipo cuente con tiempo flexible de trabajo.
- “Sentir el dolor”: Se refiere a que las reuniones se programan fuera del horario de trabajo correspondiente al de uno de los equipos o de una de las partes del equipo, por eso el nombre de *sentir el dolor*, puesto que un equipo o parte del equipo tendrá que sacrificar tiempo extra en el trabajo.

### 3.1.2.2 Tecnologías de la Información y Comunicación (TIC) para la colaboración a través de internet

Las Tecnologías de la Información y la Comunicación, que suelen ser nombradas TIC, son tecnología que favorece el tratamiento de la información y el proceso de comunicación. Tienen gran importancia en la sociedad actual debido a que se hace uso de ellas en diversas actividades de la vida diaria satisfaciendo a muchas necesidades (Baelo & Cantón, 2009).

En el caso particular de la propuesta metodológica, que es el objetivo principal de este trabajo, la cual va dirigida al desarrollo global de software, que como se describió en el Capítulo 2, se basa en el desarrollo de software con equipos de trabajo colaborando en sitios físicos remotos. Por lo cual para poder realizar dicha colaboración es necesario establecer una estrategia de comunicación, a través del uso de la tecnología disponible (como el internet). En esta parte es donde entran las TIC, centrándose especialmente en aquellas herramientas que permitan el trabajo en grupo.

Actualmente existen diversos servicios de internet que van enfocados al desarrollo y gestión de tareas e información de manera remota. A continuación se citan algunas herramientas que existen en la red que nos permiten crear grupos de trabajo, colaborar o compartir documentos entre varias personas de forma remota:

**Google Drive.** (Figura 3.1) Es un servicio que ofrece Google que permite almacenar archivos en la nube<sup>3</sup>, compartir dichos archivos o carpetas con otros usuarios quienes pueden verlos, editarlos, comentarlos o descargarlos sin que sea requerido enviar archivos adjuntos por correo electrónico (Google, s.f.).

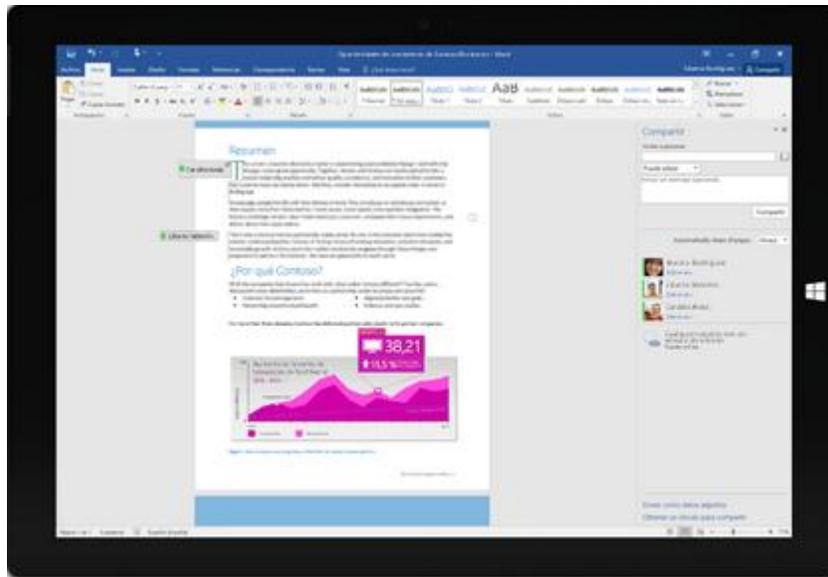
---

<sup>3</sup> Servicio de almacenamiento en internet.



**Figura 3.1 Trabajo colaborativo en Google drive (Ortiz, 2016).**

**Office 2016.** (Figura 3.2) Permite una colaboración en tiempo real, donde se puede editar al mismo tiempo desde múltiples usuarios y múltiples dispositivos. Con esta herramienta igual es posible el comentar documentos y contestarlos sin tener que salir del modo de edición o tener que recargar el documento, existiendo para tal acción un nivel de contestación como si se tratara de un blog o foro de internet (Merino, 2015). Además de la creación conjunta de documentos permite el uso de voz y video (Microsoft, Pagina oficial de Microsoft. Recuperado el 2016, s.f.).



**Figura 3.2 Trabajo colaborativo con Office 2016 (Microsoft, s.f.).**

**Facebook groups.** Este servicio de redes sociales tiene la opción de crear grupos (ejemplo de grupo en Facebook figura 3.3) en los que las personas integrantes de dicho grupo pueden compartir contenidos, comunicarse y colaborar en documentos del grupo (Facebook, s.f.). Una de las desventajas es que no permite video llamadas entre varias personas y no se puede editar documentos en tiempo real.



**Figura 3.3 Ejemplo de grupo en Facebook.**

**Skype.** Con esta herramienta se puede llevar a cabo la realización de video llamadas entre varias personas, así como mensajería instantánea, uso compartido de pantalla y el uso compartido de archivos (Microsoft, Skype, s.f.). Ver figura 3.4



**Figura 3.4 Ejemplo de video llamada con Skype (Microsoft, Skype, s.f.).**

**Dropbox.** (Figura 3.5) Es una herramienta para almacenar distintos tipos de archivos que se puede vincular a distintos dispositivos como computadoras, teléfonos e incluso desde el sitio web de Dropbox. Lo interesante de este servicio es que se pueden compartir archivos con otras personas lo que facilita la realización de proyectos en equipos (Dropbox, s.f.).



**Figura 3.5 Almacenamiento y distribución de archivos con Dropbox (Dropbox, s.f.).**

### 3.1.3 Metodologías de desarrollo e ingeniería web

En la tabla 3.3 se presenta un análisis comparativo de las metodologías de desarrollo web que fueron mencionadas en la sección 2.3, en la cual se pueden apreciar de manera clara y concisa las diferencias entre cada metodología.

**Tabla 3.3 Comparativa de metodologías de desarrollo web.**

	UWE	RMM	OOHDM
<b>Etapas</b>	<ol style="list-style-type: none"> <li>1. Análisis de requisitos</li> <li>2. Modelo de contenido</li> <li>3. Modelo de navegación</li> <li>4. Modelo de presentación</li> <li>5. Modelo de proceso</li> </ol>	<ol style="list-style-type: none"> <li>1. Representación E-R<sup>4</sup></li> <li>2. Esquema E-R+</li> <li>3. Definición de los caminos de navegación</li> <li>4. Definición del protocolo de conversión</li> <li>5. concepción de la interfaz de usuario</li> <li>6. concepción del comportamiento en ejecución</li> <li>7. Construcción y pruebas del sistema</li> </ol>	<ol style="list-style-type: none"> <li>1. Diseño conceptual</li> <li>2. Diseño de navegación</li> <li>3. Diseño de interfaces abstractas</li> <li>4. Implementación</li> </ol>

<sup>4</sup> Entidad-Relación

Continuación de la tabla 3.3:

	UWE	RMM	OOHDM
<b>Técnica de modelado</b>	AOM <sup>5</sup>	E-R	OO <sup>6</sup>
<b>Representación gráfica</b>	Diagramas UML: de clases, de actividades, de secuencia de colaboración, entre otras.	<ul style="list-style-type: none"> <li>▪ Diagrama E-R</li> <li>▪ Diagrama E-R+</li> <li>▪ Diagrama RMDM</li> </ul>	<ul style="list-style-type: none"> <li>▪ Diagrama de clases</li> <li>▪ Esquema de clase de navegación</li> <li>▪ Esquema de contexto navegacional</li> <li>▪ Diagrama de configuración</li> </ul>
<b>Notación</b>	UML	E-R propio	OMT <sup>7</sup> /UML Propio ADVs <sup>8</sup>
<b>Herramienta de soporte</b>	UWE UML	RMCASE	OOHDM-Web

Con la tabla 3.3 se puede contrastar las características propias de cada una de las metodologías revisadas, para finalmente seleccionar aquella que mejor se adapte a las necesidades del proyecto.

### 3.1.4 Herramientas para el desarrollo de aplicaciones web

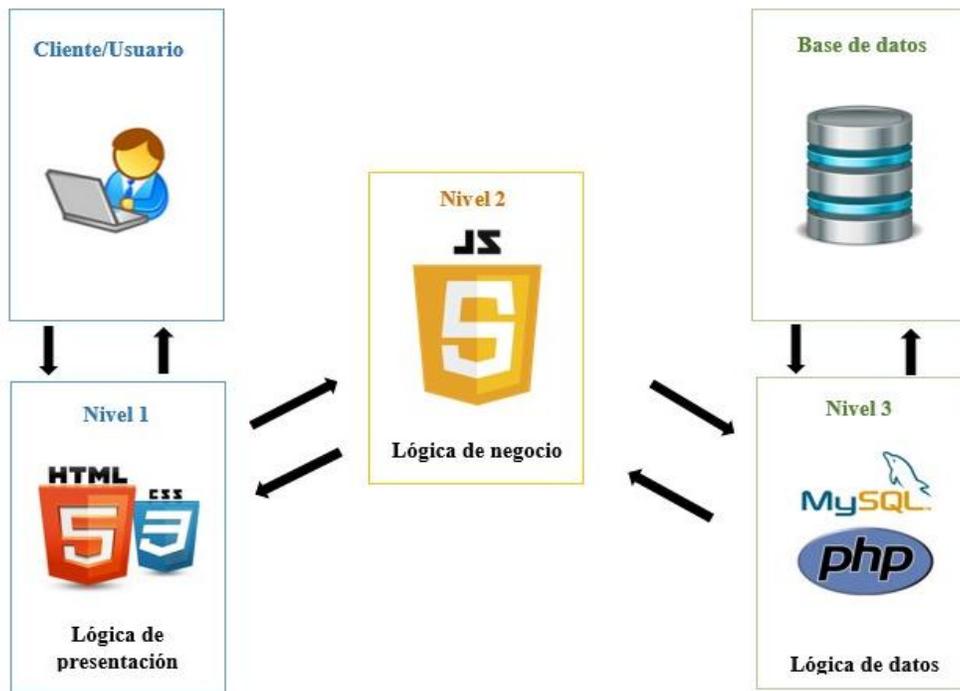
Al momento de seleccionar las herramientas que a utilizar y teniendo claro lo que se va a realizar para el desarrollo de un sistema basado en la web es necesario considerar si las herramientas en cuestión permiten todo aquello que se requiere lograr. En esta sección se hace una revisión de las herramientas tanto del lado del cliente como del lado del servidor, al igual que las de la lógica de negocio con las cuales es posible el desarrollo de aplicaciones web, considerándola arquitectura cliente servidor como se puede observar en la figura 3.6.

<sup>5</sup> Modelamiento Orientadas por Aspectos

<sup>6</sup> Orientado a Objetos

<sup>7</sup> Técnica de Modelado de Objetos (Object-Modeling Technique)

<sup>8</sup> Vistas Abstractas de Datos (Abstract Data Views)



**Figura 3.6 Patrón Cliente/Servidor.**

#### 3.1.4.1 Herramientas para la lógica de presentación (lado del cliente)

**Lenguaje de Marcación de Hipertexto (HTML).** En el trabajo realizado para el desarrollo de aplicaciones web de Aguilar, J.M. (2006), indica que HTML<sup>9</sup> hace referencia a un lenguaje utilizado para el desarrollo de páginas y aplicaciones Web y documentos. No se trata de un lenguaje de programación como en el caso de C++ o Java, realmente se trata de un lenguaje de marcado que implementó las especificaciones de SGML<sup>10</sup> conforme al estándar internacional ISO 8879.

Los documentos de HTML están formados por un conjunto de etiquetas especialmente intercaladas escritas en texto plano (ASCII). Son interpretadas por los navegadores Web que formatean y muestran los documentos. Un conjunto de documentos HTML pueden estar entrelazados mediante enlaces (en inglés conocidos como hyperlinks o links). Los documentos

<sup>9</sup> Es el acrónimo inglés de *Hyper Text Markup Language* (Lenguaje de Marcado de Hipertexto).

<sup>10</sup> Standard Generalized Markup Language.

HTML son almacenados en la computadora con la extensión .html o .htm. La estructura básica de un documento HTML contiene tres etiquetas principales (Aguilar, 2006) (Luján, 2001):

1. Comienzo y fin de un documento (**<html>**): En la figura 3.7 las líneas 3 y 10 son las que indican el inicio y fin de la página. Ésta etiqueta es obligatoria, pues indica que el documento es del tipo HTML.
2. Encabezado de la página (**<head>**): Las líneas 4 y 6 (figura 3.7) en conjunto, marcan el comienzo y fin del encabezado de la página. La etiqueta **<head>** da más información al navegador web acerca del documento, dicha información no se muestra al usuario, sólo es procesada por el navegador.
3. Cuerpo de la página (**<body>**): Con el uso de ésta etiqueta se podrá incluir la mayor parte de la información que contendrá el documento, la cual se visualizará en el navegador web, tal como: texto imágenes, texto, tablas, formularios, entre otros elementos (figura 3.7, líneas 7 y 9).

EJEMPLO: Estructura básica de una página	
1	Tipo de documento
2	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
3	<HTML>
4	<HEAD>
5	Encabezado de la página
6	</HEAD>
7	<BODY>
8	Cuerpo de la página
9	</BODY>
10	</HTML>

**Figura 3.7 Esqueleto básico de un documento HTML (Luján, 2001).**

**Hoja de Estilo en Cascada (CSS).** CSS u Hoja de estilo (Cascading Style Sheets), se trata de un lenguaje que proporciona a los elementos del documento HTML estilos visuales como es el tamaño, color, fondo, borde, etc. Con el uso de las hojas de estilo, se busca el superar las limitantes y reducir la complejidad de HTML, de esta manera es posible separar la estructura de la presentación (Gauchat, 2012).

#### **3.1.4.2 Herramientas de la lógica de negocio**

**JavaScript.** Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Cuando se habla de una página web dinámica, se hace referencia a aquella que integra efectos como son el texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario (Eguiluz, s.f.).

Este lenguaje de programación permite el script<sup>11</sup> de eventos, clases y acciones para el desarrollo de aplicaciones en internet entre el cliente y el servidor. Con JavaScript los usuarios no sólo leerán las páginas sino que además las páginas adquieren un carácter interactivo. Dentro de las características más relevantes de JavaScript encontramos que: (Alonso, Serrano, & Calzada, 2004):

- Se interpreta por el ordenador que recibe el programa, no se compila.
- Tiene una programación orientada a objetos. El código de los objetos está predefinido y es expandible. No usa clases ni herencia.
- El código está integrado (incluido) en los documentos HTML.
- Trabaja con los elementos del HTML.
- No se declaran los tipos de variables.
- Ejecución dinámica: los programas y funciones no se chequean hasta que se ejecutan.
- Los programas de JavaScript se ejecutan cuando sucede algo, a ese algo se le llama evento.

Según Alonso, Serrano & Calzada (2004) los elementos dentro de una página tales como un botón o un cuadro de selección, pueden originar un evento que ejecutará una acción. Cuando ocurre alguno de estos eventos se ejecuta una función en JavaScript. Esta función está compuesta de varias sentencias que examinan o modifican el contenido de la página Web, o hacen otras tareas para dar respuesta de algún modo al evento. Por lo general, los comandos de un programa en JavaScript se dividen en 5 categorías:

- Variables y sus valores.
- Expresiones, que manipulan los valores de las variables.
- Estructuras de control, que modifican cómo las sentencias son ejecutadas.
- Funciones, que ejecutan un bloque de sentencias.

---

<sup>11</sup> Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

### 3.1.4.3 Herramientas para la lógica de datos (lado del servidor)

**Pre-procesador de Hipertexto (PHP).** Es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web, el cual puede ser incluido en HTML (The\_PHP\_Group, s.f.).

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre otras funciones.

Éste lenguaje permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, DB2, Microsoft SQL Server, entre otros; lo cual permite la creación de Aplicaciones Web muy robustas. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI<sup>12</sup>, módulo para Apache (Comas, 2004).

**Sistema de gestión de base de datos MySQL.** Se trata de un manejador de base de datos que administra, registra, controla y manipula datos (Aguilar, 2006). En el manual de referencia de MySQL se encuentra que es el sistema más popular de gestión de base de datos SQL Open Source, el cual es desarrollado, distribuido, y apoyado por Oracle Corporation. Se dice que es de Open Source (código abierto), porque es posible para cualquiera el usar y modificar el software. Quien sea puede descargar el software de MySQL desde internet y usarlo sin pagar nada.

La página oficial de MySQL dice que este software usa el GPL (GNU General Public License), para definir lo que se puede y que no se puede hacer con el software en diferentes situaciones. El código puede ser adaptado según sean las necesidades que se tenga.

---

<sup>12</sup> Interfaz de Entrada Común (en inglés Common Gateway Interface)

Las bases de datos dentro de MySQL son relacionales. Una base de datos relacional almacena datos en tablas separadas en lugar de colocarlos todos juntos como si fuera un gran almacén. Esta propiedad proporciona velocidad y flexibilidad al momento de almacenar, organizar y acceder a los datos. La parte de SQL de MySQL representa a Structure Query Language. SQL es el lenguaje estandarizado más común para acceder a las bases de datos (ORACLE, s.f.).

Con los aspectos vistos en esta sección se puede inferir que con el empleo de las TIC dentro del desarrollo global de software, así como también el trabajo colaborativo es posible el superar uno de los mayores obstáculos que se enfrentan debido a la distancia en la que se ubican las personas o grupos de desarrolladores dentro de un proyecto, que es la comunicación, en torno a la cual depende todo el proyecto a lo largo del ciclo de vida.

### **3.5 Descripción de la metodología propuesta**

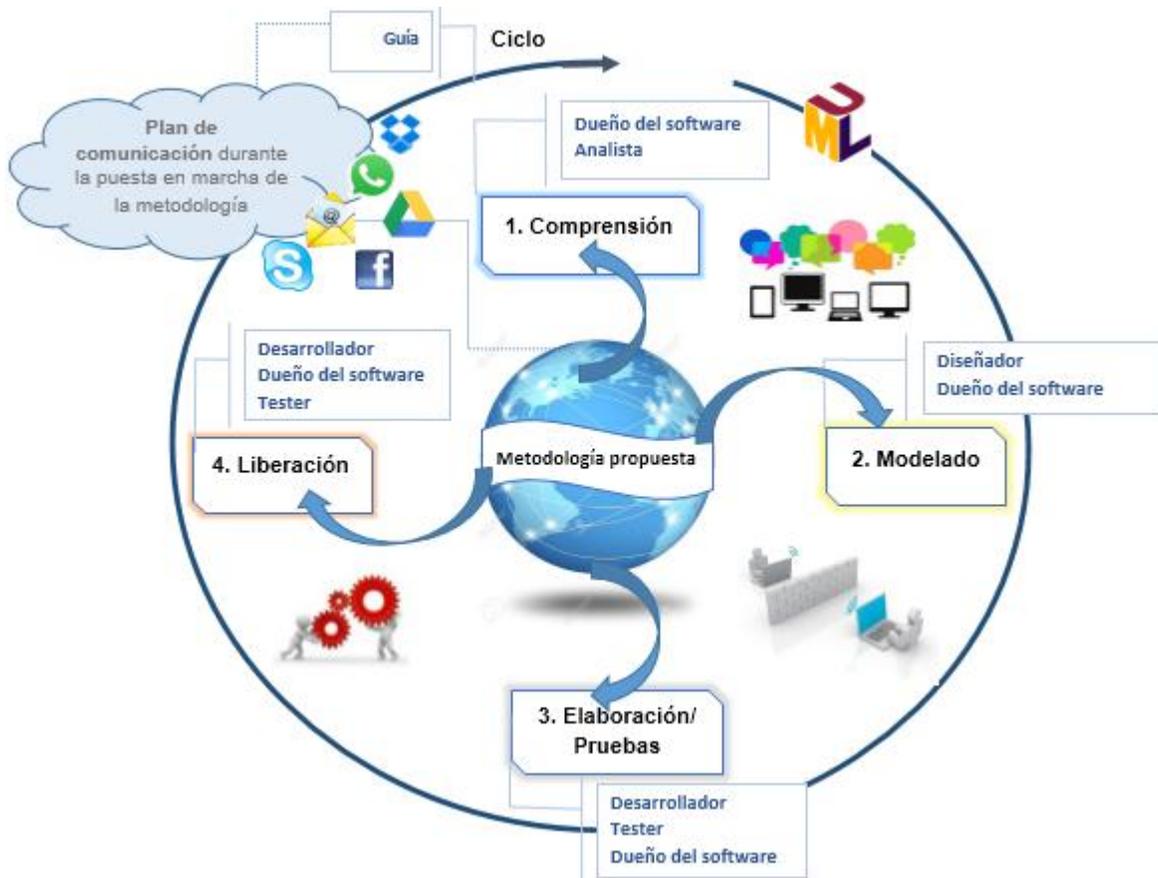
Este trabajo va enfocado a la creación de un modelo que sea ideal para la construcción de software, donde los colaboradores, quienes integran el equipo de trabajo (conformado por sub-equipos distribuidos), se localizan en puntos geográficamente separados. Para cual se requiere llevar de forma paralela en todo el desarrollo del proyecto un plan de comunicación, con la finalidad de superar las barreras que implica la distancia, y así conseguir la creación de software para la web. Para tal finalidad, las bases para la metodología que se propone se sustentan en principios metodológicos y de ingeniería, de tal manera se proporciona sustento firme para el desarrollo de aplicaciones web. Se inicia con el establecimiento de los elementos necesarios para la construcción de un sistema orientado a internet. Una vez establecida la parte del producto software que se va a elaborar se procede a la determinación y elaboración del modelo de producción para el DGS, el cual se fundamenta en las metodologías ágiles de ingeniería de software y en las estrategias colaborativas para garantizar la calidad de la comunicación.

Se determina que para el modelo de producción de aplicaciones web se tomarán en cuenta algunas especificaciones de UWE, puesto que hace uso de UML el cual al ser un lenguaje gráfico es más fácil de comprender y así poder producir un diseño con el que se elaborara cada

módulo que formará parte del sistema, para lo cual se emplea el diagrama de caso de uso. De igual manera se consideran aspectos de la metodología Scrum. La elaboración del plan de comunicación se realiza en base a las premisas a cumplir según sugiere Piorun & Giusti (2001) incluyendo los tres tipos de distancias que afectan al DGS.

Dentro del plan de comunicación, que es la parte que interesa para el DGS, se considera de suma importancia una documentación de todos los procesos que se realicen. Esta documentación debe estar disponible para cada uno de los miembros del equipo en conjunto, lo que asegura que todos se dirijan en una misma dirección y se cumplan los objetivos sin complicaciones.

La metodología que se propone se puede ver en la figura 3.8, la cual es una estructura que consta de cinco etapas en las que en cada una se especifican los participantes que intervienen. Esto no quiere decir que durante el tiempo que se utilice la metodología para un proyecto se debe realizar la misma función limitada por el rol asignado, se puede participar en distintas tareas correspondientes a otras funciones durante el ciclo de vida, a excepción del guía. Es importante indicar que todos los miembros participan en las decisiones que se requieran tomar a lo largo del empleo de la metodología propuesta.



**Figura 3.8 Metodología propuesta.**

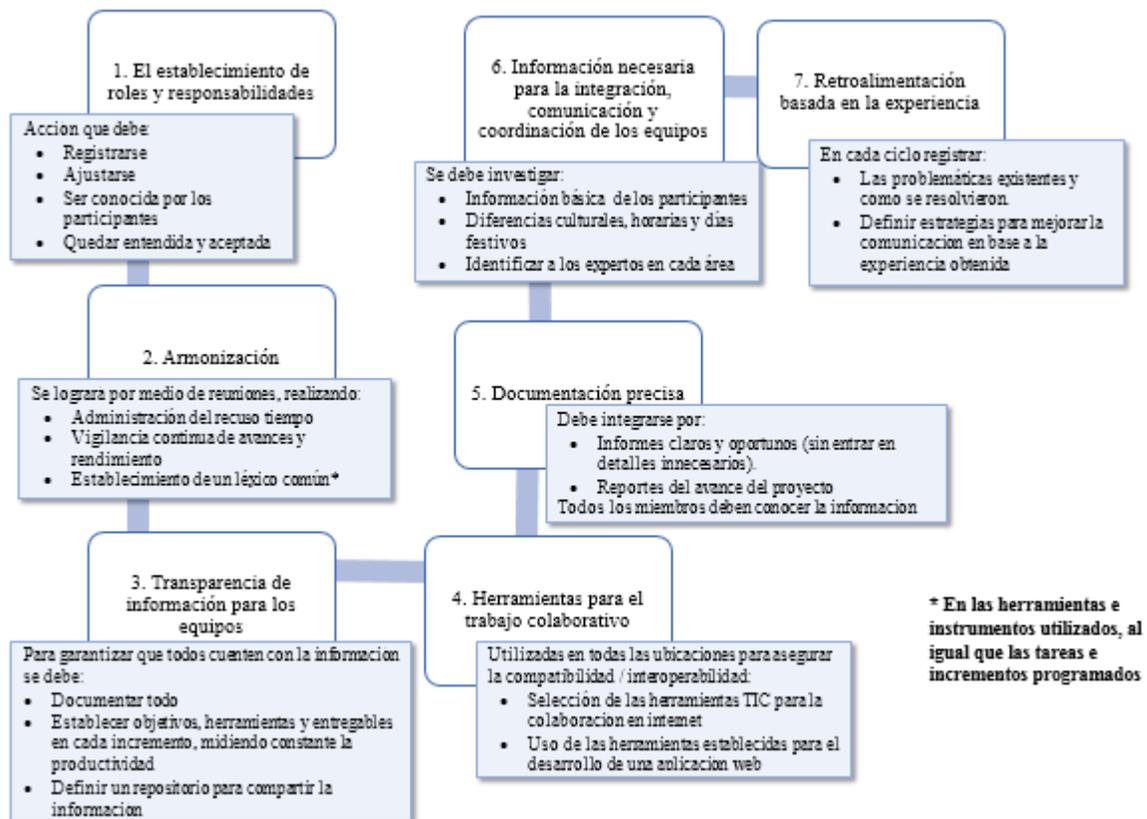
La metodología que se desarrolla se realiza por ciclos lo que significa que es iterativa y que cada ciclo produce un incremento de software. Puesto que esta perfilada hacia lo ágil, es posible que se susciten cambios a lo largo del proyecto, por la que la metodología establecida debe ser flexible para admitir las variaciones que se puedan originar.

Las entradas, salidas y participantes de las etapas que componen cada ciclo se muestran en la tabla 3.4.

**Tabla 3.4 Entradas, salidas y participantes de la metodología propuesta.**

Etapa	Entradas	Salidas	Participantes
<b>Comprensión</b>	Obtención y análisis de los requerimientos	Clasificación de requerimientos: funcionales y no funcionales Plan de trabajo Diagrama UML de caso de uso	Dueño del software Analista
<b>Modelado</b>	Casos de uso	Propuesta del incremento Evaluación y retroalimentación del diseño	Diseñador Dueño del software
<b>Elaboración/ Pruebas</b>	Diseño del incremento Caso de prueba	Determinación de cambios y correcciones Incremento	Desarrollador Tester Dueño del software
<b>Liberación</b>	Nuevo incremento	Integración e implementación del incremento, pruebas de integración al finalizar el proyecto.	Desarrollador Dueño del software Tester

En la figura 3.9 se puede ver los criterios a tomar en cuenta para el plan de comunicación.



**Figura 3.9 Criterios para el plan de comunicación.**

Seguir un plan de comunicación en la metodología propuesta es el elemento que diferencia la metodología que en esta sección se establece de las existentes, y la especializa al desarrollo global de software al crear dicho plan como un mecanismo que permita superar los desafíos que por la distancia se presentan. Este mecanismo se puede ir mejorando y ajustándose con base a la experiencia que se va acumulando, lo que permite adaptarlo a distintas circunstancias.

### 3.5.1 Los participantes

**Guía.** Es la persona a cargo de la supervisión de que se cumpla y se lleve adecuadamente el empleo de la metodología, se asegura de que el entorno de desarrollo sea propicio para que todo transcurra debidamente. Se trata de quien marca el camino a seguir, en ningún momento intervendrá en el trabajo que realicen los demás participantes de los equipos, su función se limita de proveer las herramientas necesarias para regular la comunicación (lleva el plan de comunicación). Se sugiere que este participante no tenga otra función dentro del proyecto, puesto que no se trata de un líder, sino de un guía, de allí su nombre, y de esta manera evitar confusiones respecto al papel que desempeña.

**Dueño del software.** Este participante representa al dueño principal, y a los usuarios finales del sistema. El dueño del software es quien solicita el sistema de información y determina finalmente lo que se desarrollará, en otras palabras decide lo que se hace, tiene la última palabra. Al representar a los usuarios finales debe considerar en todo momento las prioridades de estos, quienes después de todo son los que harán uso del sistema, es decir, para quienes va dirigida la aplicación web. Los requerimientos en base a los usuarios finales son los que tendrán peso en las decisiones del dueño del software.

El equipo general de desarrolladores (integrado por los equipos distribuidos) deben de estar formados por integrantes que abarquen o tengan nociones de todas las funciones, esto para que en algún momento puedan ayudar o aportar una idea o solución a alguna problemática que se presente. Además de que si por alguna cuestión algún miembro falta o se ausenta no afecte al

proceso de desarrollo, y así evitar retrasos en las entregas de los incrementos. Los roles considerados necesarios para poder desarrollar un sistema de información son:

- **Analista.** Se trata de quien realiza la recolección y clasificación de los requerimientos, así como el análisis de los mismos.
- **Diseñador.** Es quien realiza la propuesta de la vista del incremento o funcionalidad a realizar correspondiente a cada ciclo, a través de un diseño de forma gráfica.
- **Desarrollador.** Realiza la codificación para el nuevo incremento funcional potencialmente entregable.
- **Tester.** Lleva a cabo tanto la propuesta del plan de pruebas como la aplicación del mismo.

### 3.5.2 Reuniones y etapas

Antes de comenzar la descripción de las etapas, se definirán las distintas reuniones virtuales que se deben y pueden llevarse a cabo en la metodología que se describe, con la finalidad de que sean conocidas y sea más claro cuando se les haga mención dentro de cada etapa. Las estrategias de las reuniones, así como las herramientas a utilizar deben definirse en el *Plan de comunicación*. Los tipos de reuniones se especifican a continuación:

**Reunión informal.** Se realiza antes de comenzar el ciclo con la intención de que se creen vínculos sociales entre los participantes, generando de tal manera equipos cohesivos para que trabajen mejor en conjunto. Se trata de que todos se conozcan entre sí para crear el sentido de pertenencia, en otras palabras que cada persona dentro del proyecto se sienta integrada al equipo como si se tratara de uno solo ubicado en un mismo lugar y no de varios dispersos, con lo cual la realización del trabajo se desarrollará en armonía, produciendo satisfacción con el resultado obtenido, lo que se traduce en motivación. Este tipo de reunión no requiere ser documentada, aunque puede contribuir para en la retroalimentación de para mejorar el plan de comunicación.

**Reunión previa.** Se lleva a cabo antes de comenzar un ciclo, deben participar todos los miembros del equipo, aquí se determinan las tareas a realizar para el nuevo ciclo, la duración en días y el porcentaje que representa en el proyecto total, con lo que se va construyendo el *mapa general del plan de comunicación*. Esta reunión tiene un carácter formal, y se realiza antes de comenzar el ciclo. Aquí el guía, a través de un análisis previo en base a las habilidades de los integrantes que conforman los equipos (registrado en el plan de comunicación), se definen las responsabilidades que cada uno tomará en el desarrollo de ciclo actual. El guía prepara, presenta y comparte en el *plan de comunicación* que será actualizado en esta reunión, el cual contiene los mecanismos y herramientas tecnológicas para que el proceso de comunicación se realice de manera adecuada. El documento obtenido de la *reunión previa* a parte del *plan de comunicación* actualizado, será el reporte de esta reunión. Se podrá medir el avance del proyecto que se obtendrá al finalizar el ciclo. Antes de realizarla se debe llenar y distribuir el formato del *itinerario*, que indica los punto a abordar.

Conforme los ciclos avanzan se refina y se ajusta, como se ha mencionado, el *plan de comunicación* (en base a experiencias en ciclos anteriores), para adaptar el proceso al nuevo ciclo antes de que éste comience. Todo documento generado se agregará o actualizará en el *repositorio*.

**Reunión tipo A.** Son reuniones de alto nivel, dirigidas a los participantes que intervienen en una etapa específica, con la intención de que les permita coordinar sus actividades si así lo requieren. Es el único tipo de reunión en el que no es necesaria la dirección del guía, puesto que se centra en el proceso de organizar el trabajo a realizar entre los desarrolladores del incremento. Sólo en casos muy particulares que se crea necesaria la intervención del guía, éste puede intervenir para manejar algún conflicto que surja. Normalmente lo que se obtiene de este tipo de reuniones es una parte del incremento que debe irse integrando en el *repositorio*.

Las *Reuniones tipo A* no tienen un límite de uso, pueden ocuparse cada vez que se consideren necesarias y no requieren ser documentadas.

**Reunión tipo RC.** Es una reunión donde se presentan los resultados al finalizar cada etapa, y donde el dueño del software es pieza importante para evaluar lo producido en el ciclo. Asisten todos los participantes, con el propósito de supervisar los avances, realizar observaciones, correcciones, aclarar posibles dudas o conflictos, apoyar en trabajos atrasados, dar opciones de solución a los problemas que se presenten o hasta cambiar o adiconas requerimientos. El documento generado es el formato de liberación del ciclo. Se mide el porcentaje alcanzado y se define la fecha para la siguiente *reunión previa* donde se planificará el ciclo que continua.

En las reuniones que se definieron es el guía quien juega un papel importante como moderador, pues es quien marcará el rumbo que deben seguir estas reuniones con la finalidad del lograr el objetivo que persiguen. Es importante que para cada *reunión previa* el guía realice antes un *itinerario* que contemple, como lo menciona Piorun & Giusti (2001), lo siguiente:

- Elaborar un programa de los puntos a abordar en cada tipo de reunión
- Distribuir el programa entre todos los miembros que participaran en la reunión con un tiempo considerable antes de que ésta se realice.
- Hacer un sondeo al inicio con los integrantes, esto con el objetivo evitar en la medida de lo posible las sorpresas.
- Documentar los puntos más relevantes que se traten durante la reunión.
- Asignar y repartir a cada tema e integrante un tiempo especial.
- A no ser la *Reunión previa*, debe evitarse que los temas se salgan fuera de contexto y se tornen personales, buscar la concentración y evitar las distracciones.
- Al finalizar la reunión deben darse a conocer las conclusiones por cada punto tratado, así como indicar a cada miembro aquello que falta por realizar.
- El *Reporte de la reunión* debe ser compartido entre todos los participantes, para lo cual debe agregarse al repositorio. Debe quedar establecida la fecha para la próxima reunión.
- Entre una reunión y otra, hacer un seguimiento de cada punto (un contraste).

En cada etapa se hace mención de los participantes principales en cada una de ellas, debido a la relevancia de su función dentro de la fase, puesto que se determinaron como pieza clave por la cantidad de trabajo que realizan y la importancia que tienen para el desarrollo de la etapa. Cabe señalar que tanto el dueño del software como los integrantes de los equipos pueden y deben involucrarse en todas las etapas. El guía vigila el proceso a lo largo del ciclo de vida del proyecto.

Un ciclo consta de alrededor de 4 semanas, es preferible no rebasar este tiempo para que sea un proceso ágil, donde las entregas de incrementos son rápidas.

**Etapa 1: Comprensión.** Etapa en que el analista obtiene los requisitos pertenecientes al módulo definido que se va a desarrollar de acuerdo al *mapa general*. Los requerimientos son obtenidos a partir del dueño del software a través de la *reunión previa*, en este punto el analista procede a clasificar y analizar los requerimientos para refinarlos y detectar información faltante necesaria para el correcto desarrollo del sistema. De ser requerido se lleva a cabo una reunión de tipo A para aclarar dudas y obtener más información, realizada entre el dueño del software y el analista.

El objetivo de esta etapa es que el resto del equipo sea capaz de comprender lo que se debe hacer. El analista debe hacer uso del *formato de recolección de requerimientos* con el cual los podrá clasificar en funcionales y no funcionales, con lo cual se podrán identificar riesgos y prioridades, también se elabora la descripción textual de los casos de uso así como su diagrama UML, que serán incluidos en el *formato de análisis de requerimientos*. Estas actividades y el resto para las siguientes etapas se realizan en base *plan de trabajo del ciclo*. Todos los documentos generados deben ser compartidos en el *repositorio* establecido en el *plan de comunicación* para que todos los participantes tengan acceso a la información. El guía debe asegurarse que la etapa se lleve a cabo satisfactoriamente, que se cumpla con el *plan de comunicación*, que todos reciban la información de manera oportuna por medio de la estrategia que haya elegido para tal fin.

**Etapa 2: Modelado.** Una vez identificados y analizados los requerimientos se realiza el modelado o diseño de lo que será el incremento del sistema, específicamente para la interfaz. Si se trata del primer ciclo se lleva a cabo el análisis y diseño de la arquitectura de la base de datos el cual lo realiza el diseñador. Este participante del equipo también crea el bosquejo preliminar del sistema de forma textual, así como el mapa de navegación de cada ciclo con lo cual el diseñador realiza el modelado, ambos elementos se en el *diseño de la propuesta*, el diseño de las pantallas se comparte en el repositorio pero no es necesario incluirlas en la documentación formal.

Los participantes para coordinarse en su trabajo pueden hacer uso de *reuniones tipo A* o compartir la información requerida a través del *repositorio* con documentación clara y precisa. Una vez concluido el *diseño de la propuesta*, se realiza una *reunión tipo RC* para darlo a conocer al resto de los participantes, en esta reunión se evalúa la propuesta y se da una retroalimentación, donde el punto de vista más importante a considerar es el del dueño del software. Todo se documenta y se comparte en el repositorio donde queda al alcance de todos. El guía debe asegurarse de que se sigan todos los lineamientos y que todos reciban la información de manera precisa y puntual para evitar atrasos y confusiones.

**Etapa 3: Elaboración/Pruebas.** Una vez que ya se cuenta con el diseño previo aceptado, se puede comenzar con la codificación del sistema por parte de los desarrolladores, por cada trabajo concluido el tester aplicará sus casos de prueba. Al comienzo de la etapa el tester diseñará las pruebas con el *formato de caso de prueba*, basándose en el *análisis y diseño de requerimientos* y en el *diseño de propuesta*. Los elementos mencionados deben encontrarse previamente a través del repositorio para que esté disponible, sobre todo para los desarrolladores, quienes de esta manera puedan determinar cómo realizar mejor su trabajo y pasar los casos de prueba de manera satisfactoria. Una vez concluida la parte de la programación se realiza inmediatamente la evaluación. Si el tester encuentra fallas o posibles mejoras lo hace saber a los desarrolladores con el *formato de sugerencias y correcciones*. Para la coordinación del trabajo entre los participantes de la etapa se recomienda el uso de las *reuniones tipo A*.

Para la metodología descrita se definieron como herramientas de construcción de software, considerando que va enfocado a la web, para la parte de la lógica de presentación el uso de HTML, para la lógica de negocio JavaScript, y en el caso de la lógica de datos PHP y MySQL (para la base de datos). Se determinó de esta manera, de acuerdo a los criterios del *plan de comunicación*, para tener un léxico común entre los desarrolladores en cuanto a las herramientas a utilizar en la construcción de la aplicación web, evitando así conflictos en el trabajo colaborativo.

Finalizada la parte de codificación y pruebas, realizada entre los desarrolladores y el tester, se procede a una *reunión tipo RC* para evaluar junto al dueño del software el nuevo incremento (para cambios usar el *formato de sugerencias y correcciones*). Si fuera necesario realizar más evaluaciones debido a modificaciones o correcciones seguir el protocolo de *reunión tipo RC*. Todos los documentos generados se agregan en el *repositorio* tan pronto como sea posible. El guía vigila el correcto desarrollo de esta fase, y como en las etapas anteriores debe de cerciorarse de que la información se encuentre disponible para todos los participantes en el momento oportuno.

**Etapa 4: Liberación.** Integración del incremento dentro del sistema. Se realiza por parte de los desarrolladores, quienes al momento de la integración realizaran los ajustes necesarios para lograr esta tarea con éxito. Realizada la integración se presenta el sistema con el nuevo incremento al dueño del software en una *reunión del tipo RC*. El tester elabora una evaluación/pruebas para el sistema en conjunto con el nuevo módulo implementado. Si surgen nuevos requerimientos o cambios estos se consideran para un nuevo ciclo registrándolos en el *formato de sugerencias y correcciones*.

Al finalizar la etapa el guía se hace una recapitulación del ciclo a través de los problemas surgidos y cómo se enfrentaron, haciendo uso del *formato de retroalimentación* (que forma parte del *plan de comunicación*: sección 7). También el final de un ciclo representa una parte más que se construye del *manual de usuario*. Todo queda registrado en el repositorio, el guía

verifica que esta acción se lleve a cabo, y da soporte a lo largo de la etapa. Se firma el documento referente a la liberación, que valida que se ha aceptado el nuevo incremento (todo documento se comparte en el *repositorio* en el lugar asignado).

### 3.5.3 Instrumentos de apoyo

**Plan de trabajo del ciclo.** Almacena las actividades a realizar durante el ciclo para completar un incremento. El instrumento se presenta en la imagen 3.10.

<i>Plan de actividades del ciclo</i>			
Ciclo no.		Módulo a desarrollar:	
	Tiempo estimado		
	Objetivo del ciclo		
<i>Actividades</i>			<i>Responsable</i>

**Figura 3.10 Plan de trabajo del ciclo**

Descripción de los datos que requiere el *plan de trabajo del ciclo*:

- Ciclo no.: número del ciclo de que corresponde el plan de actividades.
- Módulo a desarrollar: Representa al incremento que se pretende alcanzar al finalizar el ciclo.
- Tiempo estimado: El cálculo del tiempo que tomará el realizar la actividad.
- Actividades: Las actividades a realizar (de manera sencilla, no entrando en detalles) en el ciclo correspondiente para lograr de manera satisfactoria el desarrollo del módulo.
- Responsables: Participantes quienes responderán por la realización de la actividad.

**Plan de comunicación.** En éste de determinan las herramientas y estrategias e utilizar en el proceso de comunicación de acuerdo a los criterios definidos en la introducción de la sección

3.6 del capítulo actual, vistos en la figura 3.9. El formato del documento que se propone para esta finalidad se muestra por secciones.

En la figura 3.11 se muestra la *sección 1* para el establecimiento de funciones.

<i>Plan de comunicación : Sección 1</i>						
<i>1. Establecimiento de funciones</i>		Roles: 01 Guía, 02 Dueño del software, 03 Analista, 04 Diseñador, 05 Desarrollador, 06 Tester				
<i>Distribución de los participantes</i>						
Concepto			Etapas			
Nombre	Firma	Asignación de roles	E1	E2	E3	E4

**Figura 3.11 Plan de comunicación (Sección 1).**

Campos a llenar de la plantilla del Plan de comunicación (Sección 1):

- Concepto
  - Nombre: Correspondiente al integrante de los equipos.
  - Asignación de roles: La función que desempeñara el integrante dentro del proyecto.
  - Firma: En esta sección los distintos miembros de los equipos firman de conocimiento y aceptación de las funciones asignadas.
- Mn (Número del módulo del que se trata)
  - E1, E2, E3, E4: Hacen referencia a la etapa de 1) Comprensión, 2) Modelado, 3) Elaboración / Pruebas, respectivamente. Se marca las casilla dela etapa en la que la persona será responsable.

En cuanto a la armonización se refiere, se presenta el formato en figura 3.12 que incluye el mapa general del proyecto.

<i>Plan de comunicación: Sección 2</i>						
<i>2. Armonización: Mapa general</i>						
<i>Nombre del sistema</i>						
<i>Descripción breve del sistema</i>						
<i>Objetivo general del proyecto</i>						
<i>Tiempo estimado de terminación</i>						
<i>Módulos del proyecto</i>						
<i>Nombre del módulo</i>	<i>Descripción</i>	<i>Tiempo estimado</i>	<i>Prioridad</i>	<i>Porcentaje</i>	<i>Avance al finalizar el módulo</i>	<i>Objetivo</i>

**Figura 3.12 Plan de comunicación (Sección 2).**

Datos que se requieren en el Plan de comunicación (Sección 2):

1. En primera parte de este formato los datos a llenar son:
  - Nombre del proyecto: Nombre que tendrá el sistema de información a realizar.
  - Descripción breve del sistema: Se expone de manera resumida de lo que tratara el sistema.
  - Objetivo general del proyecto: Indica la dirección del proyecto de manera global.
  - Tiempo estimado de terminación: Lapso que se considera para finalizar el trabajo total.
  
2. En la segunda parte del formato los datos que se solicitan son:
  - Nombre del módulo: Los módulos identificados en que se puede dividir el sistema, los cuales se realizarán en un ciclo a la vez.
  - Descripción: Especificación del módulo correspondiente
  - Tiempo estimado: Tiempo en que se espera concluir y entregar del módulo funcional en cuestión.
  - Prioridad: Importancia de realización del módulo del que se trate.

- **Porcentaje:** Corresponde al porcentaje que representa el módulo concluido con respecto al sistema total.
- **Avance al finalizar el módulo:** Mide el progreso del sistema, sumando el porcentaje que se logró al término del ciclo anterior (si es que hay) con el que se obtuvo en el último realizado.
- **Objetivo:** Resultado que se espera lograr con la realización del módulo.

En la figura 3.13 se muestra el formato para la definición del *Repositorio*, por el medio del cual se compartirá la información para todos los equipos.

<i>Plan de comunicación: Sección 3</i>	
<i>3. Transparencia de la información para los equipos</i>	
<i>Nombre del Repositorio</i>	
<i>Carpetas que integran el Repositorio</i>	
<i>Nombre</i>	<i>Descripción</i>

**Figura 3.13 Plan de comunicación (Sección 3).**

Información necesaria en el Plan de comunicación (Sección 3):

- **Nombre del Repositorio:** Nombre con el cual se definirá la herramienta a través de que será posible el compartir información a todos los participantes.
- **Carpetas que integran el Repositorio:**
  - **Nombre:** Denominación del medio para archivar los documentos creados.
  - **Descripción:** Definir el tipo de documentos que contendrá.

La figura 3.14 corresponde al establecimiento de las herramientas destinadas al trabajo colaborativo.

<i>Plan de comunicación: Sección 4</i>	
<i>4. Herramientas para el trabajo colaborativo</i>	
<i>Razón</i>	<i>Herramienta seleccionada</i>
<i>Reuniones:</i>	
<i>Repositorio:</i>	
<i>Localización personal:</i>	

**Figura 3.14 Plan de comunicación (Sección 4).**

Los datos a llenar del Plan de comunicación (Sección 4) son:

- Herramienta seleccionada, para:
  - Reuniones: Determinación de la herramienta de TIC a emplear en la realización de reuniones.
  - Repositorio: Determinación de la herramienta de TIC a emplear en para establecer el repositorio.
  - Localización personal: Determinación de la herramienta de TIC que se usará para localizar a un miembro de forma específica.

La figura 3.15, representa el formato que contendrá cada documento, con la finalidad de asegurar de que todos visualicen la información, con lo que se confirma que es aceptada.

<i>Plan de comunicación: Sección 5</i>		
<i>5. Documentación precisa y oportuna</i>		
<i>Estrategia:</i>		
<i>Formato</i>		
<i>Nombre</i>	<i>Función</i>	<i>Firma</i>

**Figura 3.15 Plan de comunicación (Sección 5).**

Descripción de los campos del Plan de comunicación (Sección 5):

- Estrategia: Acción que se propone para saber que a todos los integrantes llega la información.
- Formato
  - Nombre: Nombre del participante.
  - Función: Rol actual que desempeña.
  - Firma: Cuando el participante conoce y acepta la información agrega su firma.

El formato para la recolección de información relevante de los integrantes de los equipos se puede ver en la figura 3.16.

Plan de comunicación: Sección 6				
6 Información concerniente a los miembros de los equipos				
Equipo 1:				
Nombre				
Funcion				
Telefono movil				
e-mail				
Habilidad identificada				
Zona horaria				
Dias festivos				
Equipo 2:				
Nombre				
Funcion				
Telefono movil				
e-mail				
Habilidad identificada				
Zona horaria				
Dias festivos				
Equipo n:				
Nombre				
Funcion				
Telefono movil				
e-mail				
Habilidad identificada				
Zona horaria				
Dias festivos				

**Figura 3.16 Plan de comunicación (Sección 6).**

Los datos requeridos para llenar la información básica de cada integrante, perteneciente a cada equipo clasificados según su ubicación en el Plan de comunicación (Sección 6) son:

- Equipo n: se ingresa la ubicación, es decir, el lugar en el que se localiza un grupo de participantes.

- Nombre: Nombre del integrante.
- Función: Responsabilidades (rol) dentro del proyecto.
- Teléfono móvil: Número celular del miembro de los equipos.
- E-mail: Correo electrónico.
- Habilidad identificada: Área en que mejor se desempeña.
- Zona horario: Referente al horario que se maneja según la ubicación del equipo.
- Días festivos: Aquellos días que no son laborables, dependiendo la ubicación en la que encuentre el equipo.

Cubriendo el último criterio del plan de comunicación se presenta en la figura 3.17 el formato para la retroalimentación por ciclo

<i>Plan de comunicación: Sección 7</i>			
<i>7. Formato de retroalimentación</i>			
Impacto: G-grave, M-medio, B-bajo			
Ciclo 1			
Problemática	Afectación	Impacto	Cómo se solucionó
Ciclo 2			
Problemática	Afectación	Impacto	Estrategia de solución
Ciclo n			
Problemática	Afectación	Impacto	Cómo se solucionó

**Figura 3.17 Plan de comunicación (Sección 7).**

La información que se necesita para el Plan de comunicación (Sección 7) son los siguientes:

- Ciclo n:
  - Problemática: Dificultades presentadas durante el ciclo.
  - Afectación: Lugar o parte que en que afectó el problema.
  - Impacto: Nivel de efecto de la problemática.
  - Cómo se solucionó: Acciones realizadas para contrarrestar el daño.

**Itinerario.** Permite programar los temas a tratar en una reunión, la duración de la misma, el objetivo que se busca lograr, el tiempo establecido para cada tema y quienes participaran en dicha reunión, la plantilla para el *Itinerario* se aprecia en la figura 3.18.

<i>Itinerario</i>		
Fecha		Hora
<i>Tipo de reunión</i>		
<i>Tiempo estimado de duración</i>		
<i>Ciclo</i>		
<i>Módulo</i>		
Objetivo		
Puntos a abordar		
No.	<i>Especificación</i>	<i>Tiempo asignado</i>
1		
2		
3		
4		
5		
n		
Participantes de la reunión		
<i>Nombre</i>	<i>Función</i>	<i>Firma de enterado</i>

**Figura 3.18 Itinerario.**

Descripción de los datos del Itinerario:

- Fecha: Data en que se planea la reunión.
- Hora: La hora en que se tiene programada la reunión.
- Tipo de reunión: Reunión de la que tratará el itinerario.
- Tiempo estimado de duración: Intervalo de tiempo que se determina se llevará acabo la reunión.
- Ciclo: Número de ciclo al que corresponde la reunión que se planea.
- Módulo: El nombre del módulo que se está elaborando en el ciclo.
- Objetivo: Dirección que sigue la reunión.
- Puntos a abordar
  - Especificación: Determinación del punto a tratar.
  - Tiempo asignado: El tiempo que se estipula necesario para tratar el tema.
- Participantes de la reunión
  - Nombre: Se refiere al nombre de la persona que estará en la reunión.
  - Función: Rol que desempeña en el ciclo actual.
  - Firma de enterado: Una vez que el participante recibe el itinerario agrega su firma, con la cual se comprueba que conoce la información.

**Reporte de reunión.** Recupera la información relevante vista en una reunión dada. El instrumento que permite tal acción se presenta en la figura 3.19.

<b>Reporte de reunión</b>				
Fecha de la reunión:			Duración:	
Tipo de reunión:		Ciclo:	Módulo:	
Participante	Asistencia	Causa de la falta		
1.				
2.				
3.				
4.				
5.				
n				
Puntos del itinerario				
No.	Especificación	Abordado	Causa	Conclusiones
1				
2				
3				
4				
5				
n				
<i>Se cumplió el objetivo de la reunión:</i>				
<i>Causa:</i>				

**Figura 3.19 Reporte de reunión.**

Datos requeridos en el *Reporte de reunión*:

- Fecha de la reunión: La fecha en que se realizó la reunión.
- Duración: Tiempo que duró la reunión.
- Tipo de reunión: El tipo de reunión que se efectuó.
- Ciclo: El número de ciclo en que se hizo la reunión.
- Módulo: El nombre del módulo que producirá el incremento correspondiente al ciclo.
- Participante: Nombre del integrante que debe participar en la reunión.
- Asistencia: Marcar si el participante estuvo presente en la reunión.
- Cusa de la falta: Si es que el participante no asistió mencionar el motivo.
- Puntos del itinerario
  - Especificación: Punto del itinerario.
  - Abordado: Confirmación de que se vio el punto.
  - Causa: Si es que no se abordó el punto indicar la razón.
  - Conclusiones: Las conclusiones a las que se llegó después de analizado en punto.

- Se cumplió el objetivo de la reunión: Determinación si la causa por la que se hizo la reunión realmente se cumplió.
- Causa: Si es que no se cumplió anotar que originó esta situación.

**Recolección y de requisitos.** La platilla que se propone para esta finalidad, permite registrar los requerimientos del sistema, clasificarlos en funcionales y no funcionales.

La figura 3.20 muestra el patrón que se sugiere para la recolección de requisitos.

<i>Recolección de requisitos</i>	
<i>Formato del ciclo no.</i>	<i>Módulo</i>
Requerimientos funcionales	
Requerimientos no funcionales	

**Figura 3.20** Recolección de requisitos.

El formato presentado de Recolección de requisitos, requiere lo siguiente:

- Formato del ciclo no. contiene: El número del ciclo actual.
- Módulo: Nombre del módulo que se desarrolla en el ciclo.
- Requerimientos funcionales: Los requerimientos identificados y clasificados en requerimientos funcionales.
- Requerimientos no funcionales: Los requerimientos identificados y clasificados en requerimientos no funcionales.

**Análisis de requisitos.** La plantilla que permite esta actividad (figura 3.21), presenta el análisis hecho sobre los requerimientos obtenidos, por medio de la especificación textual de los casos de uso y el diagrama que los representa.

<b>Análisis de requisitos</b>	
<i>Formato del ciclo no.</i>	<i>Módulo</i>
Especificación textual de los casos de uso	
<i>Caso de uso: Consultas</i>	
<i>Actores</i>	
<i>Resumen</i>	
<i>Secuencia</i>	
<i>Precondiciones</i>	
<i>Post-condiciones</i>	
<i>Comentarios</i>	
Diagrama de caso de uso	

**Figura 3.21 Plantilla de análisis de requisitos.**

Los datos que se requieren para llenar la *plantilla de análisis de requisitos* son:

- Formato del ciclo no. contiene: El número del ciclo actual.
- Módulo: Nombre del módulo que se desarrolla en el ciclo.
- Especificación textual de los casos de uso.
  - Actores: Quienes están interactuando en el caso de uso.
  - Resumen: La síntesis del caso de uso.
  - Secuencia: Orden de las acciones que seguirá el sistema en la parte que corresponde al caso de uso.
  - Precondiciones: Los requisitos previos para la interacción en materia concerniente al caso de uso.
  - Post-condiciones: Los requisitos posteriores para la interacción en materia concerniente al caso de uso.
  - Comentarios: Información adicional que se requiera agregar.
- Diagramas de casos de uso: Gráfico UML correspondiente al caso de uso.

**Diseño de la propuesta.** Instrumento dirigido para la etapa de *modelado*. Su objetivo es la descripción de manera textual del diseño que se propone, así como también la imagen de la misma y su mapa de navegación.

<i>Diseño de la propuesta</i>	
<i>Formato del ciclo no.</i>	Módulo
<i>Objetivo</i>	
<i>Composición</i>	
<i>Respuesta</i>	
Figura	
Mapa de navegación	

**Figura 3.22 Diseño de la propuesta.**

En la figura 3.22 se puede ver el formato para el Diseño de la propuesta. Los datos que pide son:

- Formato del ciclo no. contiene: El número de ciclo correspondiente.
- Módulo: El nombre del módulo perteneciente al ciclo actual.
- Objetivo: Finalidad que persigue el diseño propuesto.
- Composición: Elementos que contendrá la interfaz (sólo en caso de que aplique).
- Respuesta: La respuesta que se espera obtener.
- Figura: Imagen de la propuesta.

- Mapa de navegación: Estructura que tendrá la interfaz web (no aplica en todos los casos).

**Interacción entre niveles:** Para la documentación correspondiente a la etapa 3 de la metodología, específicamente para la parte de elaboración se debe llenar este formato (figura 3.23 como se indica a continuación).

Interacción entre niveles (Arquitectura cliente-servidor)		
Formato del ciclo no.	Módulo	
Lógica de presentación	Lógica de negocio	Lógica de datos

**Figura 3.23 Formato para la parte de elaboración de la etapa 3, interacción entre niveles.**

Manera en que se llena el formato:

- Formato del ciclo no. contiene: Número del ciclo actual.
- Módulo: Nombre del módulo actual que se desarrolla en el ciclo.
- Lógica de presentación: La forma en que se comportará la interfaz.
- Lógica de negocio: La manera en que se realizaran las peticiones al sistema.
- Lógica de datos: El comportamiento que presentará la base de datos.

**Caso de prueba** Cada en cada ciclo se debe realizar un plan de pruebas por parte del tester, mediante casos de prueba, correspondientes a la etapa de *Elaboración y Pruebas*. El instrumento para elaborar el *Caso prueba* se muestra en la figura 3.24.

<i>Caso de prueba</i>		
Formato del ciclo	Módulo	
Caso de prueba	Resultado esperado	Resultado

**Figura 3.24 Formato para el caso de prueba.**

Datos para alimentar el instrumento de *caso de prueba*:

- Formato del ciclo no. contiene: Número del ciclo actual.
- Módulo: Nombre del módulo actual que se desarrolla en el ciclo.
- Caso de prueba: Descripción del caso de prueba a realizar.
- Resultado esperado: Es el resultado que se espera obtener al aplicar el caso de prueba.
- Resultado: Es el resultado, producto de comparar el caso de prueba con el resultado después de realizarlo, pudiendo adquirir el valor de correcto si lo que se esperaba es lo que se obtuvo, o incorrecto, si lo que se esperaba no coincide con lo que se obtuvo.

**Liberación del ciclo:** para la última etapa del ciclo se debe firmar junto con el dueño del producto la aprobación del ciclo (figura 3.25).

<i>Liberación del ciclo</i>	
Formato del ciclo no.	Módulo
Integrante:	Firma de aprobación
Dueño del software:	
Analista:	
Diseñador:	
Desarrollador:	
Tester:	
<i>Firma de conocimiento del guía</i>	

**Figura 3.25 Documento de validación de liberación del ciclo.**

Los datos a ingresar son:

- Formato del ciclo no. contiene: El número del ciclo en el que se elabora la solicitud.
- Módulo: Nombre del módulo en desarrollo dentro del ciclo actual.

- Integrante: Enfrente del rol se indica el nombre de la persona que desempeña esa función.
- Firma de aprobación: Se incluye la firma del participante que tiene asignado ese rol, concluyendo con la firma de conocimiento del guía, que más bien como un testigo en esta situación.

**Formato de sugerencias y correcciones.** Cuando algún miembro de los equipos desee exponer una sugerencia o solicitar un cambio puede hacer uso de la plantilla de la figura 3.26.

Formato de sugerencias y correcciones	
Fecha:	
Ciclo:	
Módulo:	
Tipo de sugerencia o cambio: 01 Comunicación, 02 Proceso de desarrollo, 03 Otro	
En caso de ser del tipo 03 especificar	
Propuesta de sugerencia o cambio	
Razon de la sugerencia o cambio	
Elementos involucrados	

**Figura 3.26 Formato de sugerencias o cambios.**

Datos de la plantilla *formato de sugerencias o cambios*:

- Fecha: La fecha en que se realiza la solicitud.
- Ciclo: El número del ciclo en el que se elabora la solicitud.
- Módulo: Nombre del módulo en desarrollo dentro del ciclo actual.
- Tipo de sugerencia o cambio: Elegir el tipo de sugerencia o cambio que se desee hacer.
- En caso de ser del tipo 03 especificar: Si el tipo de sugerencia o cambio pertenece al tipo *03 Otro*, aclarar de qué se trata.

- Propuesta de sugerencia o cambio: Espacio para hacer la descripción de la sugerencia o cambio de la que se trata.
- Elementos involucrados: A qué afecta o repercute la solicitud.

En este capítulo se establecieron los lineamientos de la metodología propuesta, la cual se empleará y comprobará el funcionamiento en el desarrollo del capítulo siguiente, con la elaboración de un caso de estudio para generar una aplicación web que administre recurso bibliográfico (tesis).

## Capítulo 4 Aplicación de la metodología propuesta

Para probar la funcionalidad de la metodología propuesta en el Capítulo 3, ésta se aplica a un caso de estudio para desarrollar un sistema de información basado en la web, específicamente para el manejo automático de bibliografía digital. Para el desarrollo del proyecto se ha contado con dos equipos de trabajo separados geográficamente: el primer equipo localizado en el Instituto Tecnológico de Apizaco (ITA), ubicado en Apizaco, Tlaxcala; el segundo equipo localizado en el Instituto Tecnológico del Altiplano de Tlaxcala (ITAT), ubicado en San Diego, Xocoyucan, Tlaxcala. Tal proyecto consiste en desarrollar una aplicación web que administre las tesis, que son el producto académico generado por los alumnos de las instituciones mencionadas. El sistema va enfocado a la consulta a través de internet del material bibliográfico (tesis), con lo que se pretende que el acceso a este recurso no dependa de su ubicación física (impresa o la versión digital almacenada en CD<sup>13</sup>), de esta forma un mayor número de usuarios pueden tener acceso a esta información.

### 4.1 Desarrollo del caso de estudio

Para el desarrollo del caso de estudio se contó con la participación de dos equipos de trabajo ubicados cada uno en puntos geográficamente distantes dentro del estado de Tlaxcala, en dos Institutos Tecnológicos como se puede apreciar en la figura 4.1.



**Figura 4.1 Equipos de trabajo para el desarrollo del caso de estudio.**

<sup>13</sup> Disco compacto, conocido como CD por sus siglas en inglés de Compact Disc.

**4.1.2 El plan de comunicación**

Siendo el plan de comunicación un elemento primordial que se siguió en el trayecto del desarrollo del proyecto, se muestra a continuación los formatos que conformaron dicho plan para la construcción de SARB.

En primer lugar se tiene la sección 1 que corresponde al establecimiento de funciones, como se puede ver en la figura 4.2.

<i>Plan de comunicación : Sección 1</i>						
<i>1. Establecimiento de funciones</i>		Roles: 01 Guía, 02 Dueño del software, 03 Analista, 04 Diseñador, 05 Desarrollador, 06 Tester				
<i>Distribución de los participantes</i>						
Concepto			Etapas			
Nombre	Firma	Asignación de roles	E1	E2	E3	E4
José Juan Hernández Mora	<i>Vo.Bo.</i>	02	x	x		x
Astrid Ariadna Torres Fernández	<i>Astrid A. F. T.</i>	02	x	x		x
Lorena Zamora Velázquez	<i>L. Zamora</i>	01	x	x	x	x
Rocío Hernández Acoltzi	<i>R. Hernández</i>	03	x			
Lidia Cortés Ruiz	<i>L. Cortés</i>	04		x		
René López Caballero	<i>R. López</i>	05			x	x
Said Pérez Flores	<i>S. Pérez</i>	06			x	x

**Figura 4.2 Plan de comunicación (Sección 1) SARB.**

Finalizada la asignación de roles, se procede a la elaboración del *Mapa General*, que corresponde a la estrategia de armonización, así los integrantes del equipo de desarrollo tienen un panorama general del proyecto y saben hacia donde se dirigen. Este mapa hace referencia a la sección 2 del plan de comunicación. La figura 4.3 muestra el *Mapa General* que se llevó a cabo para el SARB, el cual se fue completando en cada ciclo, hasta la conclusión del sistema con la elaboración del último módulo.

Plan de comunicación: Sección 2						
2. Armonización: Mapa general						
Nombre del sistema	Sistema de Administración de Recursos Bibliográficos (SARB)					
Descripción breve del sistema	El sistema a desarrollar tiene como finalidad el almacenar, preservar y difundir el producto académico realizado por alumnos del Instituto Tecnológico del Altiplano de Tlaxcala (ITAT), específicamente tesis en formato digital.					
Objetivo general del proyecto	Crear un sistema que le permita al ITA e ITAT poder administrar en forma digital sus productos académicos bibliográficos (tesis) y ponerlos en línea.					
Tiempo estimado de terminación	14 semanas (70 días)					
Módulos del proyecto						
Nombre del módulo	Descripción	Tiempo estimado	Prioridad	Porcentaje	Avance al finalizar el módulo	Objetivo
1. Diseño y desarrollo de la base de datos	Se obtendrá como resultado la base de datos en MySQL.	5 días	Alta	10%	10%	Diseñar y desarrollar la base de datos del SARB con MySQL.
2. Acceso al sistema	Este módulo permitirá tener el acceso al sistema.	8 días	Alta	10%	20%	Realizar el acceso al sistema para los usuarios.
3. Gestión de usuarios	Se obtendrá el control sobre los usuarios.	10 días	Media	15%	35%	Realizar el módulo de gestión de usuarios.
4. Gestión de materiales	El resultado es la sección de gestión de materiales funcionando correctamente.	10 días	Media	15%	50%	Realizar el módulo de gestión de materiales.
5. Consultas	Los filtros de las distintas consultas a realizar deben arrojar resultados correctos.	18 días	Media	20%	70%	Realizar la consulta de tesis por diferentes filtros.
6. Reportes	Se deben poder generar reportes a partir de las consultas posibles.	8 días	Baja	10%	80%	Agregar a la parte de consultas la sección de reportes.
7. Exportar e importar	Este módulo debe ser capaz de respaldar y recuperar la base de datos.	5 días	Baja	10%	90%	Realizar el módulo de Exportar/Importar para la tabla de usuarios y tesis.
8. Sección de ayuda	Se agregará al sistema el manual de usuario para el módulo de ayuda.	2 días	Baja	10%	100%	Realizar la sección de ayuda.

**Figura 4.3 Plan de comunicación (Sección 2) SARB.**

Para la sección 3 del plan de comunicación el resultado final se puede apreciar en la figura 4.4. Este es el resultado al término de los ciclos, puesto que se fue ajustado a las necesidades y la experiencia obtenida en cada ciclo, en otras palabras lo que mejor convino al equipo. De tal manera se cubre el objetivo de hacer transparente la información para todos los integrantes, puesto que con el repositorio se pone al alcance de cada uno de los participantes dentro del equipo de desarrollo toda la documentación generada en cada etapa de cada ciclo. Como se observa en la figura 4.4 se colocaron las subcarpetas que contiene cada carpeta principal, con la intención de un mayor control y organización de los documentos.

<i>Plan de comunicación: Sección 3</i>	
<i>3. Transparencia de la información para los equipos</i>	
<i>Nombre del Repositorio</i>	SARB ITA-ITAT
<i>Carpetas que integran el Repositorio</i>	
<i>Nombre</i>	<i>Descripción</i>
Comprensión: ▪ <i>Ciclo no. ____</i>	La carpeta de <i>Comprensión</i> contiene todos los documentos y formatos realizados en esta etapa, los cuales se organizan en subcarpetas con el número del ciclo al que corresponden.
Modelado: ▪ <i>Ciclo no. ____</i>	La carpeta de <i>Modelado</i> contiene todos los documentos y formatos realizados en esta etapa, los cuales se organizan en subcarpetas con el número del ciclo al que corresponden.
Elaboración: ▪ <i>Ciclo no. ____</i>	La carpeta de <i>Elaboración</i> contiene todos los documentos y formatos realizados en la etapa 3, pero solo los concernientes a la elaboración (codificación), los cuales se organizan en subcarpetas con el número del ciclo al que corresponden.
Pruebas: ▪ <i>Ciclo no. ____</i>	La carpeta de <i>Pruebas</i> contiene todos los documentos y formatos realizados en la etapa 3, pero solo los concernientes a pruebas, los cuales se organizan en subcarpetas con el número del ciclo al que corresponden.
Liberación: ▪ <i>Ciclo no. ____</i> ○ <i>Formatos</i>	La carpeta de <i>Liberación</i> contiene todos los documentos y formatos realizados en esta etapa, los cuales se organizan en subcarpetas con el número del ciclo al que corresponden.
Reuniones ▪ <i>Ciclo no. ____</i>	La carpeta de <i>Reuniones</i> contiene todos los documentos y formatos generados a partir de las reuniones realizadas, los cuales se organizan en subcarpetas con el número del ciclo al que corresponden.

**Figura 4.4 Plan de comunicación (Sección 3) SARB.**

Como en el caso de la sección 3 del plan de comunicación, para la sección 4 las herramientas ocupadas para la comunicación entre los miembros de los equipos distribuidos se ajustaron a lo largo del proyecto, quedando como resultado la figura 4.5.

<i>Plan de comunicación: Sección 4</i>	
<i>4. Herramientas para el trabajo colaborativo</i>	
<i>Razón</i>	<i>Herramienta seleccionada</i>
<i>Reuniones:</i>	Skype
<i>Repositorio:</i>	Google Drive
<i>Localización personal:</i>	Teléfono móvil

**Figura 4.5 Plan de comunicación (Sección 4) SARB**

Se determinó como estrategia la firma electrónica para asegurar que todos conocieran y revisaran los documentos que se fueran generando, agregando el formato de la figura 4.6 a cada documento, con la que el guía pudo constatar que todos tenían la información.

<i>Plan de comunicación: Sección 5</i>		
<i>5. Documentación precisa y oportuna</i>		
<i>Estrategia:</i>	<i>Firma digital</i>	
<i>Formato</i>		
<i>Nombre</i>	<i>Función</i>	<i>Firma</i>
José Juan Hernández Mora	Dueño del software	<i>Vo.Bo.</i>
Astrid Ariadna Torres Fernández	Dueño del software	<i>Astrid A. F. T.</i>
Lorena Zamora Velázquez	Guía	<i>LZamora</i>
Rocío Hernández Acoltzi	Analista	<i>RH Acoltzi</i>
Lidia Cortés Ruiz	Diseñador	<i>LC Ruiz</i>
René López Caballero	Desarrollador	<i>RLC</i>
Said Pérez Flores	Tester	<i>Said P. Flores</i>

**Figura 4.6 Plan de comunicación (Sección 5) SARB.**

La figura 4.7 representa la sección 6 del plan de comunicación, con este formato se recolectaron los datos más significativos de cada persona que integra a los equipos distantes.

<i>Plan de comunicación: Sección 6</i>					
<i>6. Información concerniente a los miembros de los equipos</i>					
<i>Equipo 1: ITAT</i>					
Nombre	Astrid Ariadna Torres Fernández		Lorena Zamora Velázquez		
Función	Dueño del producto		Guía		
Teléfono móvil	241*****		749*****		
e-mail	astrid_00@live.com.mx		lorena-zamora@hotmail.com		
Habilidad identificada	Docente en el área de sistemas y computación		Conocimiento de la metodología		
Zona horaria	Tiempo del centro		Tiempo del centro		
Días festivos	16 de septiembre: aniversario de la independencia de México, 01 y 02 de noviembre: día de muertos, 20 de noviembre aniversario de la revolución mexicana		16 de septiembre: aniversario de la independencia de México, 01 de octubre: aniversario del ITA, 01 y 02 de noviembre: día de muertos, 20 de noviembre aniversario de la revolución mexicana		
<i>Equipo 2: ITA</i>					
Nombre	José Juan Hernández Mora	Rocío Hernández Acoltzi	Lidia Cortés Ruiz	René López Caballero	Said Pérez Flores
Función	Dueño del software	Analista	Diseñador	Desarrollador	Tester
Teléfono móvil	241*****	241*****	951*****	241*****	248*****
e-mail	jjhmora@terra.com.mx	rocio978@hotmail.com	lidicortezruiz@gmail.com	renelc@gmail.com	said_tmd@hotmail.com
Habilidad identificada	Docente en el área de sistemas y computación	Responsabilidad, gran capacidad de organización.	Creatividad, experiencia en interfaces web.	Buen programador	Experiencia como tester
Zona horaria	Tiempo del centro	Tiempo del centro	Tiempo del centro	Tiempo del centro	Tiempo del centro
Días festivos	16 de septiembre: aniversario de la independencia de México, 01 de octubre: aniversario del ITA, 01 y 02 de noviembre: día de muertos, 20 de noviembre aniversario de la revolución mexicana				

**Figura 4.7 Plan de comunicación (Sección 6) SARB.**

Finalmente el plan de comunicación se concluye con la sección 7 que hace referencia a la retroalimentación obtenida a través de los ciclos realizados, como se ve en la figura 4.8 y 4.9, lo que permitió mejorar la comunicación entre los integrantes del equipo de trabajo.

<i>Plan de comunicación: Sección 7</i>			
7. Formato de retroalimentación			
Impacto: G-grave, M-medio, B-bajo			
Ciclo 1			
Problemática	Afectación	Impacto	Cómo se solucionó
Desorganización	G	Al buen desarrollo de las tareas.	Haciendo mas reuniones, repitiendolas o alargandolas en duración.
Desinterés	G	A la motivación para desempeñar correctamente las funciones asignadas.	Contactando a los participantes constantemente para que cumplieran con sus tareas.
Conflictos interpersonales	M	Al desempeño en conjunto.	Escuchando las problemáticas existentes e intentando llegar a buenos términos.
Desconocimiento del proyecto	M	A la visión del resultado que se pretende lograr.	Documentando más a detalle todas las acciones.
Ciclo 2			
Problemática	Afectación	Impacto	Estrategia de solución
Desorganización	G	Al trabajo en equipo, exceso de individualismo.	Con ayuda de los miembros con mas experiencia para recobrar el rumbo.
Desinterés	G	Incumplimiento de las actividades y funciones, ausentismo.	Trabajar para que cada integrante se sienta parte importante del equipo y adquiera el sentido de pertenencia.
Tecnológicos	M	La información no es conocida por todos de manera oportuna.	Utilizar varios medios por los cuales compartir la información.
Conflictos interpersonales	M	Falta de integración del equipo de trabajo.	Aumentar el tiempo de duración de las reuniones informales.
Ciclo 3			
Problemática	Afectación	Impacto	Cómo se solucionó
Desorganización/ Desinformación	G	Retraso en el tiempo establecido para concluir el módulo.	Estableciendo un solo medio para compartir el repositorio, con carpetas mejor definidas. Documentar menos pero mas claro y preciso para no saturar a los miembros del equipo.
Desinterés	M	Ausentismo, y sobre carga de trabajo para otros integrantes.	El guía hace saber a cada persona la importancia de su función dentro del equipo y trata de acercarse a los que se ausentan por medio de los integrantes que han hecho mejor vinculo con ellos.
Conflictos interpersonales	B	No se logra una cohesión completa del equipo	Dejar de darle importancia a los problemas que surgen siempre y cuando no se tornen graves.
Ciclo 4			
Problemática	Afectación	Impacto	Cómo se solucionó
Desinformación	G	Tareas repetidas, desintegradas y sin cumplir con lo establecido.	Se establece la firma electronica como estrategia para asegurar que todos visualicen los documentos y con ello tengan la información necesaria.
Desinterés	B	Ausentismo, y sobre carga de trabajo para otros integrantes.	Se ponen "sanciones" a quienes faltan, tales como: apoyar en las tareas de otros en el siguiente ciclo, apoyar al guía en la planificación de las reuniones, o tomar la función de alguien en específico, entre otros.
Conflictos interpersonales	B	No se logra una cohesión completa del equipo	Dejar de darle importancia a los problemas que surgen siempre y cuando no se tornen graves.

**Figura 4.8 Plan de comunicación (Sección 7) SARB.**

Ciclo 5			
Problemática	Afectación	Impacto	Cómo se solucionó
Desinformación	B	Re- trabajo y tiempo al limite	El guía revisa constantemente que todo firmen los documentos que se van generando
Desinterés	B	Ausentismo, y sobre carga de trabajo para otros integrantes.	Se continua con la estrategia de los castigos
Conflictos interpersonales	B	No se logra una cohesión completa del equipo	Dejar de darle importancia a los problemas que surgen siempre y cuando no se tornen graves.
Ciclo 6			
Problemática	Afectación	Impacto	Cómo se solucionó
Desinformación	B	Re- trabajo y tiempo al limite	El guía revisa constantemente que todo firmen los documentos que se van generando
Desinterés	B	Ausentismo, y sobre carga de trabajo para otros integrantes.	Se continua con la estrategia de la "sanción"
Conflictos interpersonales	B	No se logra una cohesión completa del equipo	Dejar de darle importancia a los problemas que surgen siempre y cuando no se tornen graves.
Ciclo 7			
Problemática	Afectación	Impacto	Cómo se solucionó
Conflictos interpersonales	M	Unos a otros se cumplan de los problemas.	Enfrentarse y decir claramente cual es el problema y dar una solución, para que de esta forma se centre la atención en el objetivo común a lograr y obtener los resultados esperados.
Poco tiempo	G	Se genera estrés entre los integrantes.	Trabajar horas de calidad, evitar el abarcar mucho con impacto negativo en la calidad, seguir la idea de "poco pero bien hecho".
Ciclo 8			
Problemática	Afectación	Impacto	Cómo se solucionó
Poco tiempo	M	Se genera estrés entre los integrantes.	Trabajar horas de calidad, evitar el abarcar mucho con impacto negativo en la calidad, seguir la idea de "poco pero bien hecho".

**Figura 4.9 Continuación del plan de comunicación (Sección 7) SARB.**

### 4.1.3 Desarrollo de los ciclos

En las subsecciones siguientes se presenta el desarrollo de cada uno de los ciclos, los que se desarrollaron para la elaboración total del proyecto con la aplicación de la metodología propuesta.

#### 4.1.3.1 Ciclo 1. Módulo: Diseño y desarrollo de la base de datos

Antes de comenzar el ciclo se llevó a cabo la *Reunión Previa*, en la que se obtuvo como resultado el *Plan de actividades del ciclo* (tabla 4.1). De este ciclo se obtiene el diseño de la base de datos y su desarrollo en MySQL, basado en los requerimientos generales para el sistema SARB, nombre que fue definido en la reunión previa al ciclo.

**Tabla 4.1 Plan de actividades del ciclo no. 01, módulo: diseño y desarrollo de la base de datos.**

<i>Plan de actividades del ciclo</i>			
Ciclo no.	01	Módulo a desarrollar:	Diseño y desarrollo de la base de datos
Tiempo estimado		5 días	
Objetivo del ciclo		Obtener el diseño y desarrollo de la base de datos para el sistema SARB en MySQL	
<i>Actividades</i>			<i>Responsable</i>
1. Obtención de los requerimientos generales			Analista
2. Diseño de la base de datos			Diseñador
3. Desarrollo de la base de datos en MySQL			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

La información que se encuentra en la tabla 4.1 se va agregando al Mapa general, que es parte del plan de comunicación.

A continuación se describe el desarrollo del ciclo a través de las distintas etapas, mostrando los documentos más relevantes que se generaron en cada una:

**Etapas 1. Comprensión:** Para la etapa de comprensión se obtuvieron los documentos de la figura 4.10 y 4.11, correspondiente al formato de *Recolección de requisitos* y al de *Análisis de requisitos* respectivamente, para el módulo actual. El mapa de navegación que aparece es en base a los módulos iniciales que se determinaron serán los que constituirán el sistema, y serán desarrollados en cada ciclo. Los requerimientos aquí obtenidos son los esenciales para el SARB, igual se presenta un esquema del caso de uso que engloba al todo el sistema.

Recolección de requisitos	
Formato del ciclo no.	01
Módulo	Diseño y desarrollo de la base de datos
Requerimientos funcionales	
<ul style="list-style-type: none"> <li>RC01.1 Se requiere que cada usuario se identifique dentro del sistema con nombre de usuario y contraseña, con cuatro niveles de usuarios: Administrador, Jefe del Centro de Información (Jefe del C.I.), Operador y Visitante. Cada usuario con distinto tipo de acceso al sistema. El visitante no requiere identificarse dentro del sistema</li> <li>RC01.2 Administrar usuarios (altas, modificación, bajas).</li> <li>RC01.3 Administrar tesis (altas, modificación, bajas).</li> <li>RC01.4 Realizar consultas del material bibliográfico.</li> <li>RC01.5 Generar diversos reportes por: consultas realizadas y material bibliográfico.</li> <li>RC01.6 Recuperar y respaldar base de datos</li> <li>RC01.7 Contar con una sección de ayuda</li> </ul>	
Requerimientos no funcionales	
<ul style="list-style-type: none"> <li>RNF-C01.1 Como lo propone el plan de comunicación que los instrumentos y herramientas a utilizar deben ser estándar con la finalidad de manejar un léxico común, se hace uso de las que propone la metodología que son PHP, MySQL JavaScript y HTML.</li> <li>RNF-C01.2 La interfaz contará con colores claros, de preferencia con tonos azules y blancos. Deberá ser simple y de acceso fácil e intuitivo a las distintas secciones.</li> </ul>	

Figura 4.10 Formato de recolección de requisitos SARB-Ciclo 01.

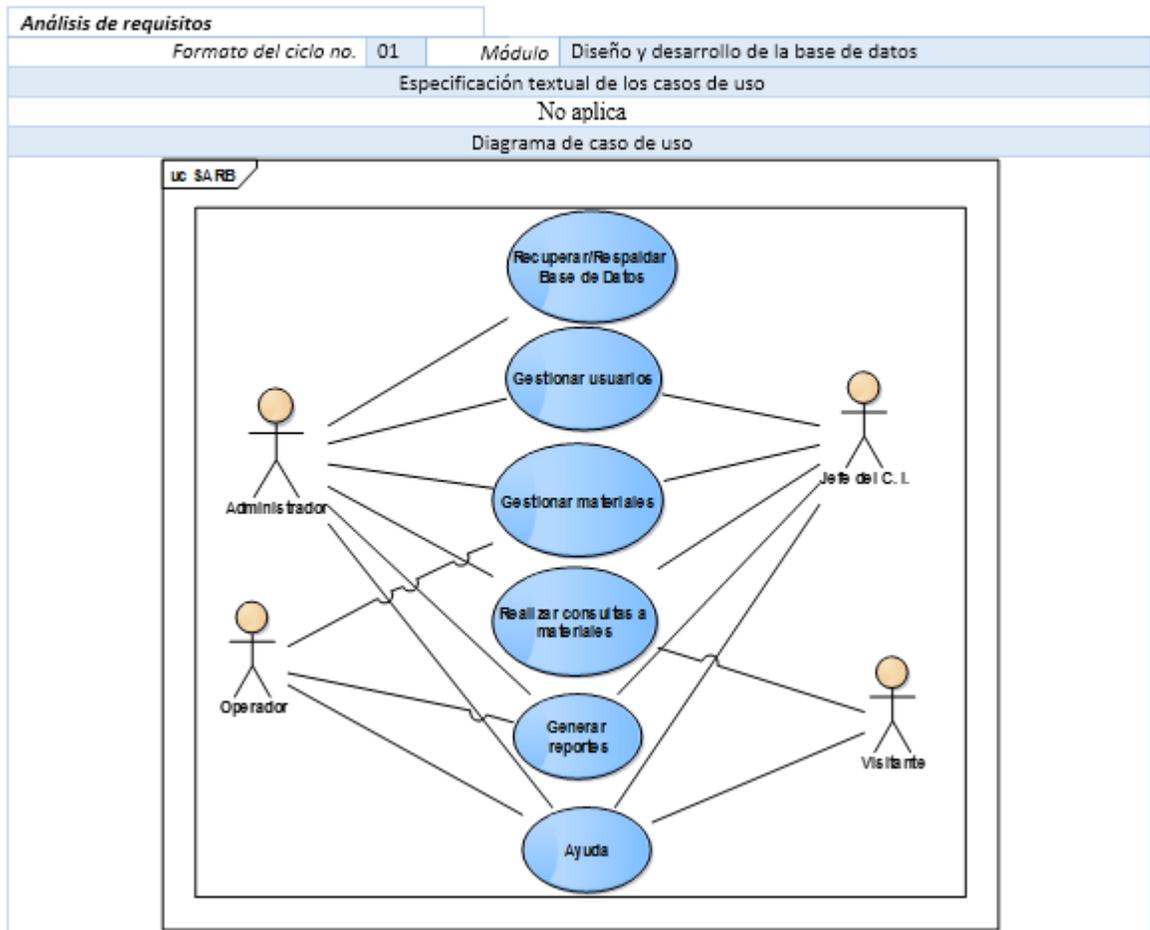
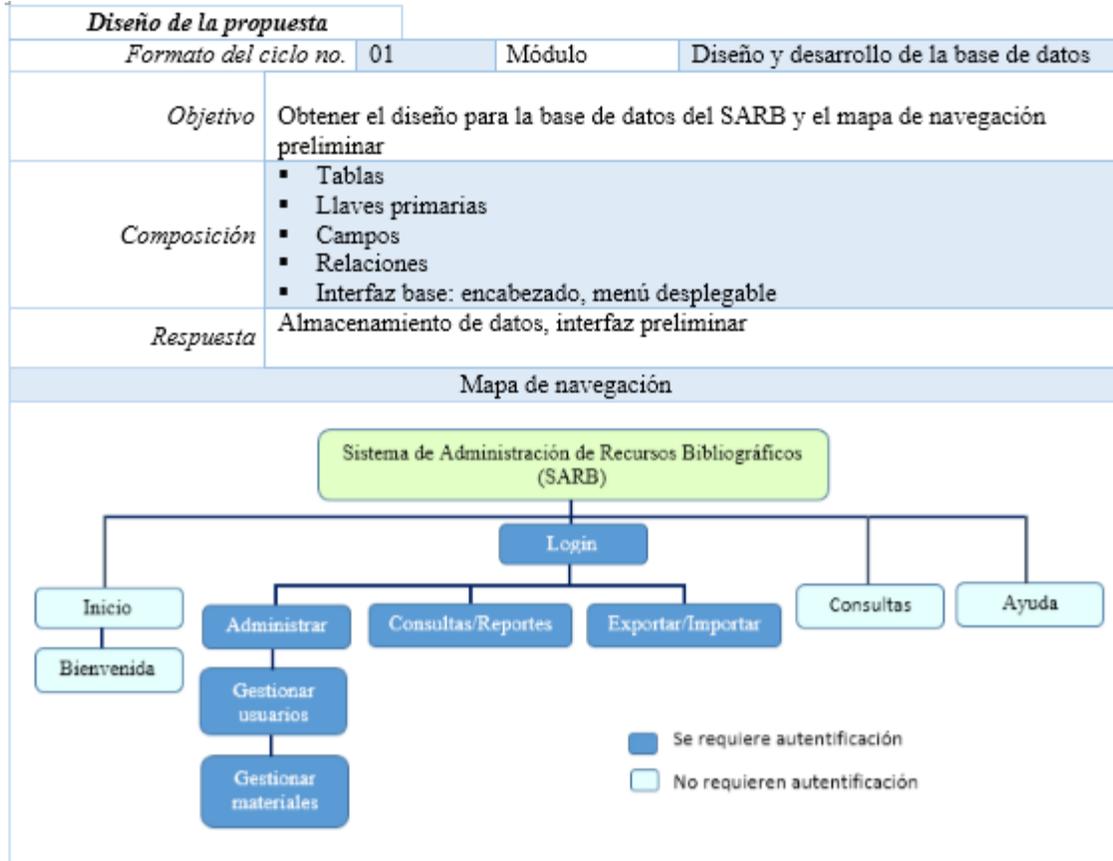


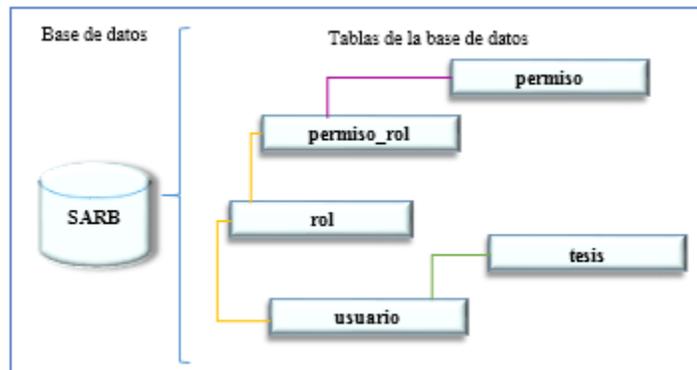
Figura 4.11 Formato de análisis de requisitos SARB-Ciclo 01.

**Etapa 2. Modelado:** En la figura 4.12 se encuentra el formato que contiene el diseño realizado para la base de datos del sistema.



**Figura 4.12** Diseño de la propuesta SARB-Ciclo 01.

El diseño de la base de datos quedó como se ve en la figura 4.13, compuesta de 5 tablas.



**Figura 4.13** Diseño de la base de datos del sistema.

**Etapa 3. Elaboración:** El resultado de esta etapa se muestra en la figura 4.14, el cual sigue el patrón propuesto en el modelado. Igual se obtiene la interfaz general o base sobre la cual se desarrollaran los módulos.



**Figura 4.14 Diagrama entidad-relación de la base de datos SARB.**

A partir del diagrama entidad-relación (ver figura 4.14) y los diagramas de caso de uso se obtiene la interfaz general del sistema (SARB), como se puede observar en las figuras 4.15 y 4.16. Este diseño se seguirá el desarrollo de cada ciclo hasta concluir el sistema.



Figura 4.15 Interfaz general del SARB (sección que no requiere autenticación).



Figura 4.16 Interfaz general del SARB (sección que requiere autenticación).

#### 4.1.3.2 Ciclo 2. Módulo: Acceso al sistema

El desarrollo del ciclo 2 incluye el acceso o autenticación dentro del sistema y la pantalla de bienvenida. En la tabla 4.2 se pueden ver las actividades correspondientes al ciclo. A continuación se presenta los resultados de este ciclo dividido en las etapas según la metodología.

**Tabla 4.2 Plan de actividades del ciclo no. 02, módulo: acceso al sistema.**

<i>Plan de actividades del ciclo</i>			
Ciclo no.	02	Módulo a desarrollar:	Acceso al sistema
	Tiempo estimado	8 días	
	Objetivo del ciclo	Realizar el acceso al sistema para los diferentes usuarios	
<i>Actividades</i>			<i>Responsable</i>
1. Obtención y análisis de los requerimientos para este módulo			Analista
2. Diseño de la interfaz de acceso y bienvenida			Diseñador
3. Desarrollo del módulo			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

**Etapa 1. Comprensión:** La figura 4.17 muestra el documento generado para esta etapa, correspondiente al formato de recolección de requisitos y la figura 4.18 al análisis de requisitos, para el módulo actual.

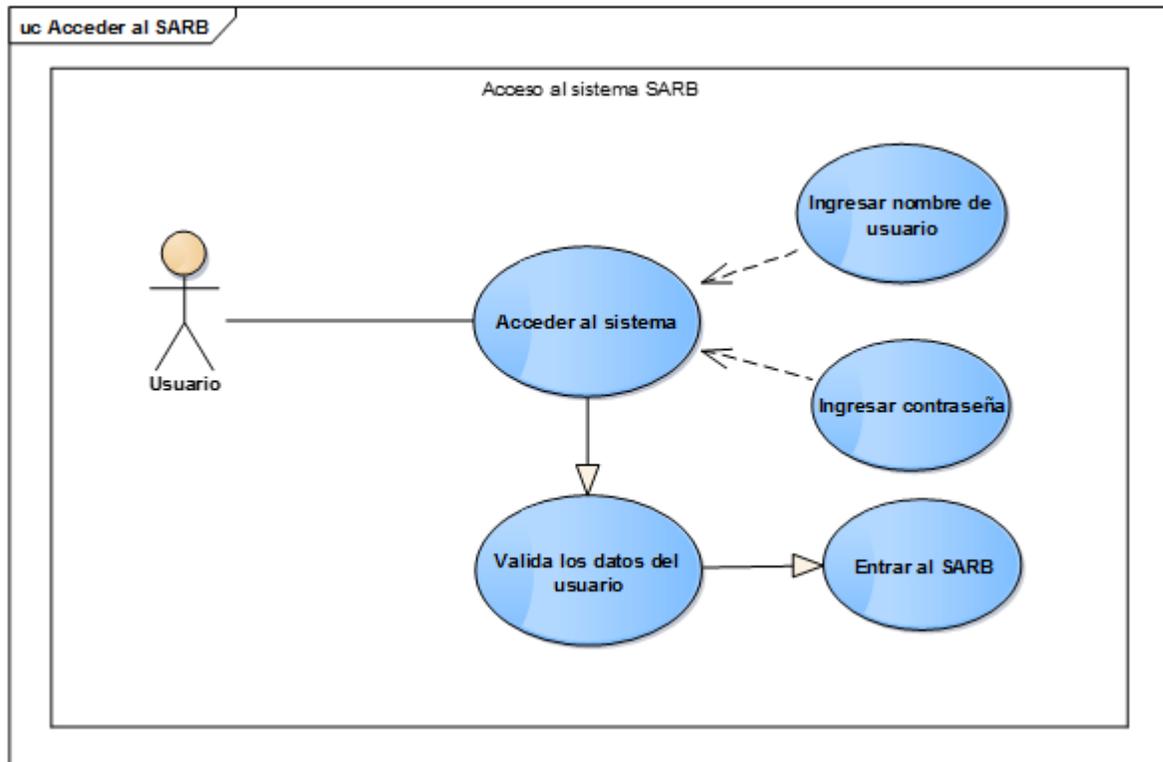
<i>Recolección requisitos</i>			
<i>Formato del ciclo no.</i>	02	<i>Módulo</i>	Acceso al sistema
<i>Requerimientos funcionales</i>			
<ul style="list-style-type: none"> <li>RC02.1 Los usuarios del sistema deberán identificarse para tener acceso al resto de las funciones, esto dependiendo de los permisos con que se cuente, a través de un nombre de usuario y una contraseña, con estos datos el usuario se identifica como administrador, jefe del CI, u operador. En el caso del visitante, éste no tendrá que identificarse (no tendrá un usuario y contraseña) puesto que no necesitará funciones administrativas.</li> </ul>			
<i>Requerimientos no funcionales</i>			
<ul style="list-style-type: none"> <li>RNF-C02.1 Cada vez que se ingresen incorrectamente los datos (usuario y/o contraseña) el sistema debe ser capaz de evitar el acceso a las funciones limitadas por permisos, definidos según el nivel de usuario del que se trate, cuando esta situación se presente no se puede visualizar otra interfaz hasta que los datos sean los correctos.</li> <li>RNF-C02.2 El manual de usuario debe contener el proceso para ingresar al sistema a través de un usuario y una contraseña.</li> </ul>			

**Figura 4.17 Formato de recolección requisitos SARB-Ciclo 02.**

<b>Análisis requisitos</b>	
Formato del ciclo no.	02
Módulo	Acceso al sistema
Especificación textual de los casos de uso	
Caso de uso: Acceso al sistema	
Actores	Administrador, Jefe del C.I., Operador
Resumen	Para poder realizar cualquier acción del sistema los actores deben acceder al sistema por medio de un nombre de usuario y una contraseña.
Secuencia	<ol style="list-style-type: none"> <li>1. El sistema solicita datos de usuario: nombre y contraseña</li> <li>2. El actor introduce los datos y selecciona la opción de <u>ingresar</u></li> <li>3. El sistema valida los datos</li> <li>4. Si los datos no son válidos el sistema regresa al paso 1</li> <li>5. Si los datos son válidos el sistema permite el acceso a la funciones correspondientes según los permisos del nivel de usuario del que se trate</li> </ol>
Precondiciones	El actor cuenta con un usuario y contraseña, el usuario debe abrir la url del sistema
Post-condiciones	El actor introduce su usuario y contraseña y el sistema lo identifica con un rol, mostrando el resultado de la acción
Comentarios	Solo el visitante no requiere autenticarse

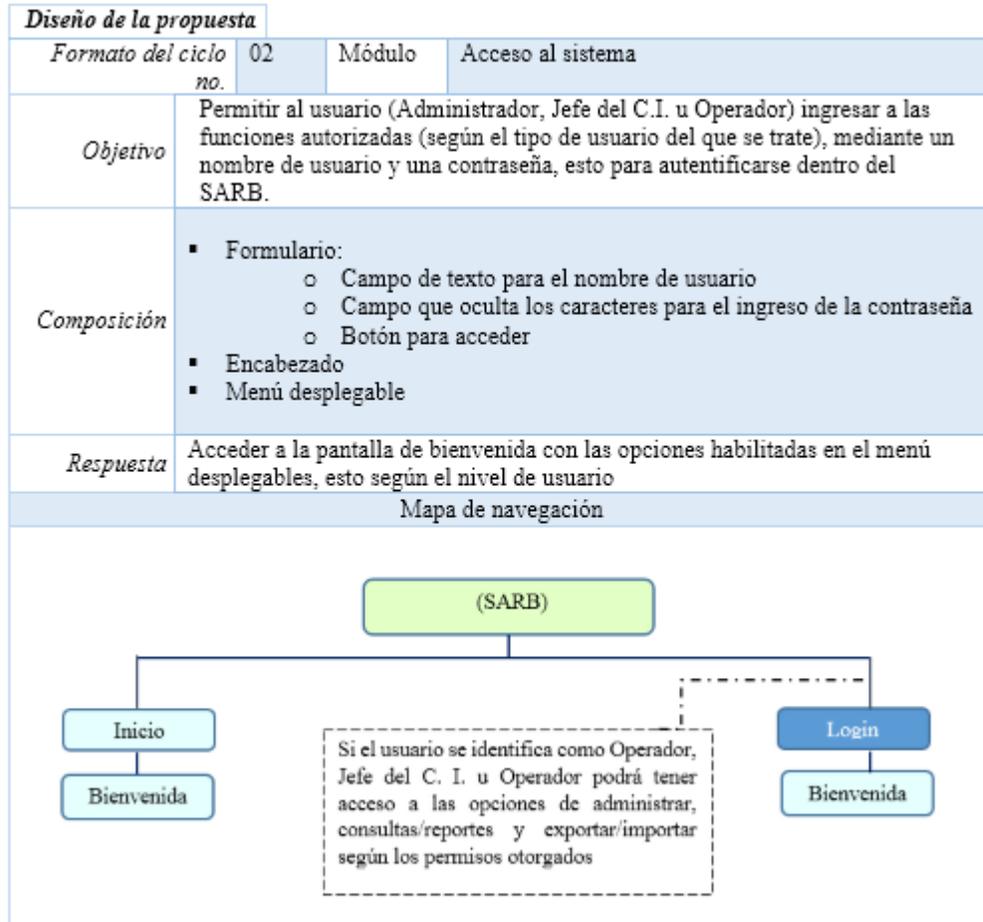
**Figura 4.18** Formato de análisis de requisitos SARB-Ciclo 02.

El diagrama del caso de uso, que es parte del formato de análisis de requisitos se muestra en la figura 4.19.



**Figura 4.19** Diagrama del caso de uso SARB-Ciclo 02.

**Etapa 2. Modelado:** El diseño del módulo de ciclo 2 se describe en la figura 4.20, con el formato para esta etapa.



**Figura 4.20** Diseño de la propuesta SARB-Ciclo 02.

**Etapa 3. Elaboración:** El resultado de la codificación, parte de la etapa 3 para el módulo 2, se aprecia en las figuras 4.21 y 4.22, las cuales corresponden a la pantalla de bienvenida y de acceso al sistema (SARB). La tabla 4.3 contiene la interacción entre niveles de este módulo.

**Tabla 4.3 Interacción entre niveles: Acceso al sistema.**

<i>Interacción entre niveles (Arquitectura cliente-servidor)</i>			
Formato del ciclo no.	02	Módulo	Acceso al sistema
Lógica de presentación	Lógica de negocio		Lógica de datos
En el menú desplegable, desde la interfaz general, aparece la sección SARB, en la que se debe ingresar el nombre de usuario y la contraseña.	Cuando los datos han sido ingresados se llama al archivo login.php, desde el cual se hace una consulta a la tabla <i>usuarios</i> , con el fin de validar los datos introducidos y determinar de qué tipo de usuario se trata. Al momento de entrar con permisos se accede primero a una pantalla de bienvenida.		Los datos ingresados (nombre de usuario y contraseña) se verifican en la base de datos Repositorio (no se lleva a cabo ningún cambio)
Interfaz del Administrador	Una vez validados los datos (nombre de usuario y contraseña), se habilitan todas las funciones.		
Interfaz del Jefe del C.I.	Una vez validados los datos (nombre de usuario y contraseña), se habilitan las funciones de gestión (de usuarios y de materiales) y reportes.		
Interfaz del Operador	Una vez validados los datos (nombre de usuario y contraseña), se habilitan las funciones agregar material y reportes		
Interfaz del Visitante	No requiere que se validen nombre de usuario y contraseña, sólo tiene habilitado las funciones por default (consultas, login, ayuda, página de inicio)		



**Figura 4.21 Interfaz de acceso al sistema.**



**Figura 4.22 Interfaz de bienvenida.**

#### 4.1.3.3 Ciclo 3. Módulo: Gestión de usuarios

Del ciclo 3 para el desarrollo del módulo de gestión de usuarios se obtendrá como resultado la sección de agregar usuario y administrar usuarios, en esta última se podrá modificar o eliminar información acerca de un usuario en específico. La tabla 4.4 muestra las actividades correspondientes al ciclo. Posteriormente se describe el desarrollo del módulo en las etapas principales de la metodología.

**Tabla 4.4 Plan de actividades del ciclo no. 03, módulo: gestión de usuarios.**

<i>Plan de actividades del ciclo</i>			
Ciclo no.	03	Módulo a desarrollar:	Gestión de usuarios
	Tiempo estimado	10 días	
	Objetivo del ciclo	Realizar el módulo de gestión de usuarios con las funciones de agregar, modificar y eliminar.	
<i>Actividades</i>			<i>Responsable</i>
1. Obtención y análisis de los requerimientos para este módulo			Analista
2. Diseño de la interfaz de agregar y administrar usuarios			Diseñador
3. Desarrollo del módulo			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

**Etapa 1. Comprensión:** En la figura 4.23 se encuentra los requerimientos recabados para la realización del módulo. En base al análisis de los requerimientos se puede ver la descripción textual del caso de uso en la figura 4.24 y el diagrama del mismo en la figura 4.25.

Recolección de requisitos			
Formato del ciclo no.	03	Módulo	Gestión de usuarios
Requerimientos funcionales			
<ul style="list-style-type: none"> <li>▪ RC03.1 Realizar altas, modificaciones y bajas de los diferentes tipos de usuario (Administrador, Jefe de CI y Operador), dichas acciones podrán ser realizadas por el Administrador y por el Jefe del C.I dentro de la opción de agregar usuario.</li> <li>• RC03.1.1 Cada vez que se ingrese un nuevo usuario se deberán proporcionar los siguientes datos: nombre usuario, contraseña, nombre, apellido paterno, apellido materno y tipo de usuario (puesto) realizando una confirmación de alta.</li> <li>• RC03.1.2 Cada vez que se realice una baja de algún usuario se deberá identificar éste en la tabla de usuarios, eligiendo posteriormente la opción eliminar, realizando una confirmación de baja.</li> <li>• RC03.1.3 Cada vez que acceda a modificar un usuario el sistema listará los datos correspondientes a éste (nombre de usuario, nombre, apellido paterno, apellido materno y tipo de usuario (puesto)), posteriormente el actor podrá modificarlos, realizando una confirmación para guardar los cambios.</li> </ul>			
Requerimientos no funcionales			
<ul style="list-style-type: none"> <li>▪ RNF-C03.1 El sistema debe mostrar un mensaje de confirmación cuando se elimine un usuario.</li> <li>▪ RNF-C03.2 Cada vez que se agregue, modifique o elimine a un usuario debe verse reflejada la acción en la lista de usuarios.</li> <li>• RNF-C03.3 El proceso de gestión de usuarios debe tener un apartado dentro del manual de usuarios.</li> </ul>			

**Figura 4.23 Formato de recolección requisitos SARB-Ciclo 03.**

Análisis de requisitos			
Formato del ciclo no.	03	Módulo	Gestión de usuarios
Especificación textual de los casos de uso			
Caso de uso: Gestión de usuarios			
Actores	Administrador, Jefe del C.I.		
Resumen	El Administrador/Jefe del C.I puede dar de alta, eliminar o modificar uno o varios usuarios, a través del módulo de gestión de usuarios, introduciendo los datos necesarios mediante un formulario o seleccionando la opción según corresponda (eliminar, modificar), el sistema finalmente muestra el resultado de acuerdo a la operación realizada.		
Secuencia	<ol style="list-style-type: none"> <li>1. El actor selecciona el módulo Administrar</li> <li>2. El actor selecciona una opción correspondiente: agregar usuario o administrar usuarios (modificar, eliminar)</li> <li>3. Si se eligió <u>agregar</u> usuario, se podrá realizar el registro correspondiente al nuevo usuario mediante un formulario</li> <li>4. Si se eligió <u>administrar</u> usuarios, (modificar, eliminar) el sistema muestra una tabla con los usuarios y sus datos.</li> <li>5. El actor selecciona un usuario</li> <li>6. El actor selecciona la opción eliminar o modificar</li> <li>6.1. Si se seleccionó <u>eliminar</u>, el sistema solicita la confirmación de la acción</li> <li>6.2. Si se seleccionó <u>modificar</u>, el sistema muestra los datos del usuario y el actor procede a modificar los datos</li> <li>7. Si el usuario <u>cancela</u> la acción elegida, el sistema muestra la tabla de los usuarios existentes</li> <li>8. Si el actor <u>confirma</u> la acción, el sistema actualiza la BD, agregando un nuevo usuario, borrando o modificando el registro (muestra la tabla con los usuarios)</li> </ol>		
Precondiciones	Los actores de este caso de uso deben estar identificados dentro del sistema.		
Post-condiciones	Las altas, modificaciones, eliminaciones de los registros deben quedar reflejados en la base de datos.		
Comentarios			

**Figura 4.24 Formato de análisis de requisitos SARB-Ciclo 03.**

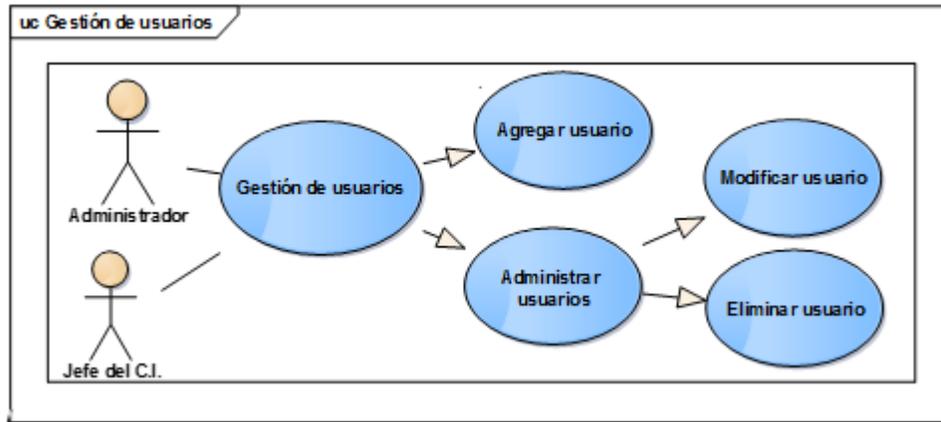


Figura 4.25 Diagrama del caso de uso SARB-Ciclo 03.

**Etapa 2. Modelado:** Para el diseño del módulo del ciclo 3 en la figura 4.26 se presenta la propuesta.

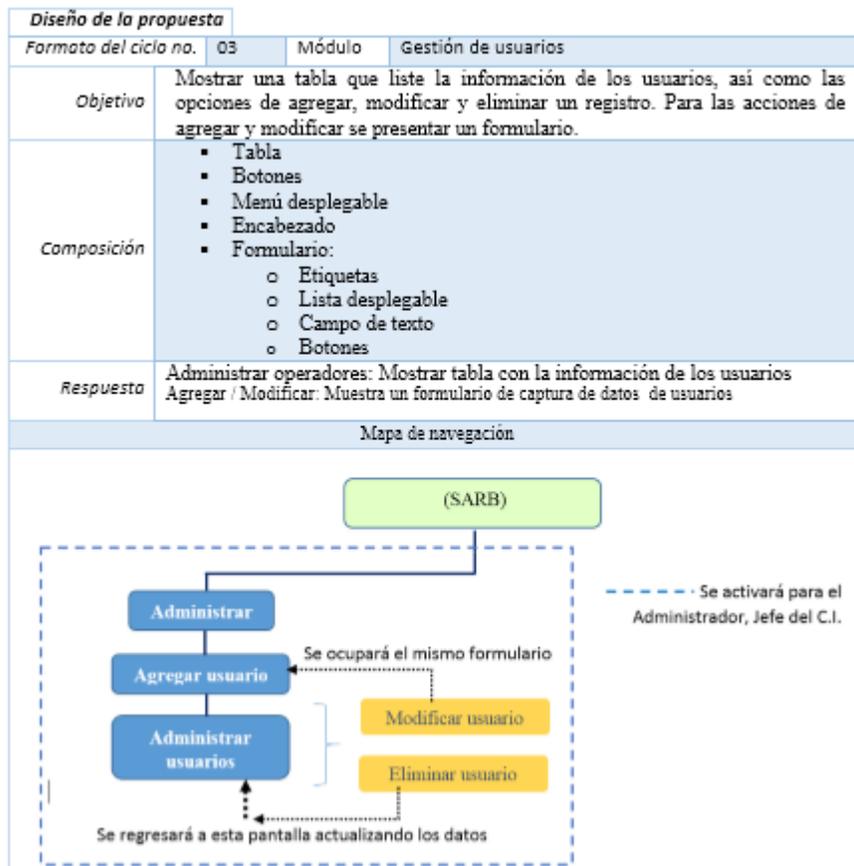


Figura 4.26 Diseño de la propuesta SARB-Ciclo 03.

**Etapa 3. Elaboración:** La tabla 4.5 hace referencia a la interacción entre niveles correspondiente al módulo de gestión de usuarios, del ciclo número tres. En la figura 4.27 y 4.28 se observan las interfaces resultado de la codificación.

**Tabla 4.5 Interacción entre niveles: Gestión de usuarios.**

<i>Interacción entre niveles (Arquitectura cliente-servidor)</i>			
Formato del ciclo no.	03	Módulo	Gestión de usuarios
Lógica de presentación	Lógica de negocio		Lógica de datos
<b>Administrar:</b> 1. Agregar usuario: Muestra el formulario mediante el cual es posible dar de alta un nuevo usuario. 2. Administrar usuarios: Esta interfaz presenta la tabla con los datos de los usuarios con las opciones de eliminar y modificar.	1. Si se llenan adecuadamente los campos del formulario, los datos son enviados a usuarios.php, donde se muestra la tabla con los usuarios y los valores del nuevo registro. 2. La acción elegida actualiza la tabla desde usuarios.php. Para el caso de modificar, envía al formulario de agregar.php (cambiando únicamente el título) con los datos del usuario seleccionado.		La tabla de <i>usuarios</i> refleja las acciones realizadas.

The image shows a web form titled "Agregar nuevo usuario". It has the following fields and controls:

- Usuario \*:** A text input field with the placeholder text "Usuario".
- Contraseña:** A text input field with the placeholder text "Clave".
- Puesto:** A dropdown menu with "Administrador" selected.
- Nombre:** A text input field with the placeholder text "Nombre".
- Apellido Paterno:** A text input field with the placeholder text "Apellido Paterno".
- Apellido Materno:** A text input field with the placeholder text "Apellido Materno".
- Buttons:** Two buttons at the bottom: "Guardar" (blue) and "Cancelar" (white).

**Figura 4.27 Formulario para agregar un nuevo usuario.**



**Figura 4.28 Interfaz para la administración de usuarios.**

**4.1.3.4 Ciclo 4. Módulo: Gestión de materiales**

El plan de actividades del ciclo número cuatro, concerniente al módulo de gestión de materiales se observa en la tabla 4.6.

**Tabla 4.6 Plan de actividades del ciclo no. 04, módulo: gestión de materiales.**

<i>Plan de actividades del ciclo</i>	
Ciclo no.	04
Módulo a desarrollar:	Gestión de materiales
Tiempo estimado	10 días
Objetivo del ciclo	Realizar el módulo de gestión de materiales con las funciones de agregar, modificar y eliminar.
<i>Actividades</i>	
Responsable	
1. Obtención y análisis de los requerimientos para este módulo	Analista
2. Diseño de la interfaz de agregar y administrar materiales	Diseñador
3. Desarrollo del módulo	Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño	Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo	Dueño del software
6. Asegurar el buen cumplimiento de la metodología	Guía

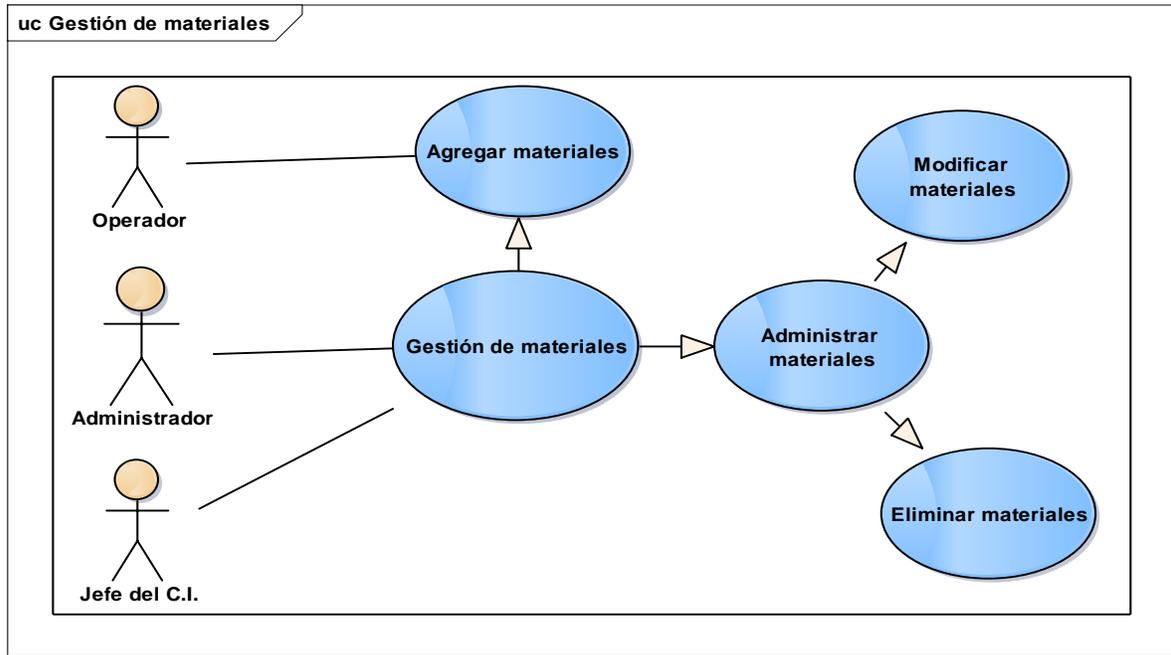
**Etapa 1. Comprensión:** En la imagen 4.29 se pueden ver los requerimientos necesarios para el ciclo número cuatro, el respectivo análisis de estos requerimientos se muestran en la imagen 4.30 al igual que el diagrama del caso de uso en la figura 4.31.

<i>Recolección de requisitos</i>		
<i>Formato del ciclo no.</i>	04	<i>Módulo</i> Gestión de materiales
Requerimientos funcionales		
<ul style="list-style-type: none"> <li>▪ RC04.1 Realizar la administración del material bibliográfico (agregar, modificar o eliminar)                             <ul style="list-style-type: none"> <li>○ RC04.1.1 Cada vez que se ingrese un nuevo material, se deberán proporcionar los siguientes datos: título, autor, director/asesor, carrera, fecha de publicación y el documento en formato PDF Se realizará una confirmación de alta</li> <li>○ RC04.1.2 Cada vez que se modifique un material, se deberán mostrar los siguientes datos: título, autor, director/asesor, carrera, fecha de publicación y documento PDF.</li> <li>○ RC04.1.3 Cada vez que se realice una baja de un material se podrá efectuar proporcionando el título, autor, director/asesor o fecha de publicación realizando una confirmación de baja.</li> </ul> </li> </ul>		
Requerimientos no funcionales		
<ul style="list-style-type: none"> <li>▪ RNF-C04.1 El sistema debe mostrar un mensaje de confirmación cuando se elimine un material (tesis).</li> <li>▪ RNF-C04.2 Cada vez que se agregue, modifique o elimine un material debe verse reflejada la acción en la lista de materiales.</li> <li>▪ RNF-C04.3 El proceso de gestión de materiales debe tener un apartado dentro del manual de usuarios.</li> </ul>		

**Figura 4.29 Formato de recolección requisitos SARB-Ciclo 04.**

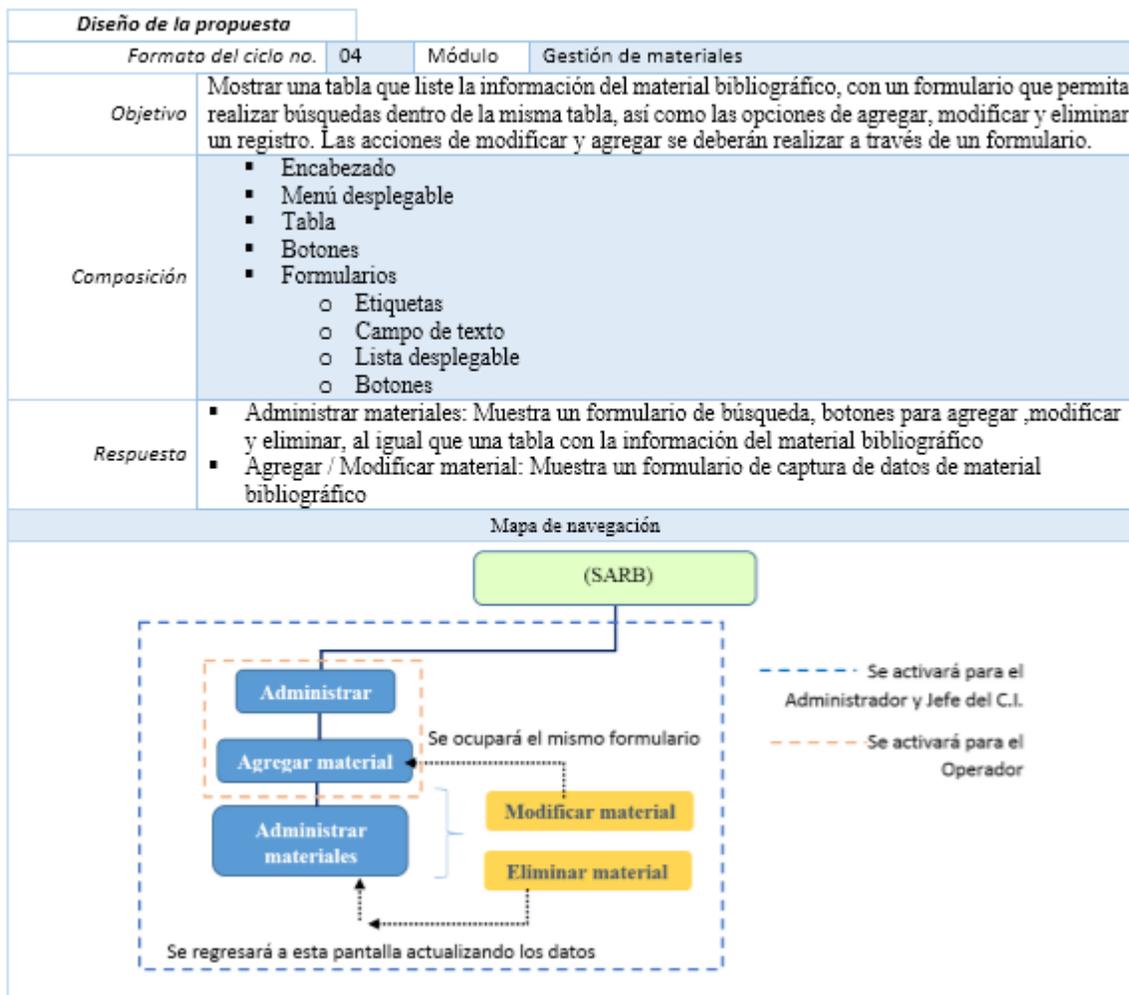
<i>Análisis de requisitos</i>	
<i>Formato del ciclo no.</i>	04
<i>Módulo</i>	Gestión de materiales
Especificación textual de los casos de uso	
<i>Caso de uso:</i> Gestión de materiales	
<i>Actores</i>	Operador, Administrador, Jefe CI
<i>Resumen</i>	El actor puede agregar, modificar o eliminar el material bibliográfico en el sistema a través del módulo de gestión de material (agregar material o administrar materiales), introduciendo los datos necesarios a través de un formulario o seleccionando la opción correspondiente (modificar, eliminar), el sistema muestra el resultado de esta operación.
<i>Secuencia</i>	<ol style="list-style-type: none"> <li>1. El actor selecciona el módulo de gestión de materiales</li> <li>2. El actor selecciona una opción correspondiente: agregar material o administrar materiales (modificar, eliminar)</li> <li>3. Si se eligió <u>agregar</u>, se podrá realizar el registro correspondiente al nuevo material mediante un formulario</li> <li>4. Si se eligió <u>administrar</u> materiales, (modificar, eliminar) el sistema muestra una tabla con los materiales y sus datos, con la opción de búsqueda general.</li> <li>5. El actor busca un material</li> <li>6. El actor selecciona la opción eliminar o modificar                             <ol style="list-style-type: none"> <li>6.1. Si se seleccionó <u>eliminar</u>, el sistema solicita la confirmación de la acción</li> <li>6.2. Si se seleccionó <u>modificar</u>, el sistema muestra los datos del material y el actor procede a modificar los datos</li> </ol> </li> <li>7. Si el usuario <u>cancela</u> la acción elegida, el sistema muestra la tabla de los materiales existentes</li> <li>8. Si el actor <u>confirma</u> la acción, el sistema actualiza la base de datos, agregando un nuevo material, borrando o modificando el registro (la interfaz muestra el listado de los materiales existentes)</li> </ol>
<i>Precondiciones</i>	El actor debe estar identificado en el sistema
<i>Post-condiciones</i>	Las acciones realizadas al material bibliográfico en el sistema se reflejan en la base de datos.
<i>Comentarios</i>	El Operador sólo puede agregar materiales

**Figura 4.30 Formato de análisis de requisitos SARB-Ciclo 04.**



**Figura 4.31 Diagrama del caso de uso SARB-Ciclo 04.**

**Etapa 2. Modelado:** El diseño, basado en el análisis realizado en la etapa previa, se realiza con el establecimiento del objetivo a alcanzar, la estructura o composición de la interfaz que empleará el usuario y la respuesta que debe emitir en su correcto funcionamiento, para tal efecto y como elemento que traza la organización de las partes que conforman el sistema web se presenta el mapa de navegación específico para este módulo. Ver figura 4.32.



**Figura 4.32** Diseño de la propuesta SARB-Ciclo 04.

**Etapa 3. Elaboración:** Para la elaboración se realiza la tabla 4.7 con la que se puede determinar la interacción entre niveles en cuanto a la arquitectura cliente servidor, de esta forma a la hora de programar el módulo de gestión de materiales se sabe lo que realizará cada parte. Para el usuario tiene mayor relevancia la interfaz, pues es con la que interactúa para hacer uso del sistema, así que al momento de ingresar a las opciones de gestión de materiales será de la manera más sencilla, con lo que se evita tener que trasladarse a distintas pantallas que pudieran resultar confusas, ocasionando que no se realice la tarea objetivo.

**Tabla 4.7 Interacción entre niveles: Gestión de materiales.**

<i>Interacción entre niveles (Arquitectura cliente-servidor)</i>			
Formato del ciclo no.	04	Módulo	Gestión de materiales
Lógica de presentación	Lógica de negocio	Lógica de datos	
<b>Administrar:</b> 1. Agregar material: Muestra el formulario mediante el cual es posible dar de alta un nuevo material (tesis). 2. Administrar materiales: Esta interfaz presenta la tabla con los datos de los materiales con las opciones de eliminar y modificar. Contiene un formulario para realizar una búsqueda específica de un material.	1. Si se llenan adecuadamente los campos del formulario, los datos son enviados a detalles.php, donde se muestran los valores del nuevo registro. 2. La acción elegida actualiza la tabla de la interfaz desde tesis.php. Para el caso de modificar, envía al formulario de tesis.php?agregar (cambiando únicamente el título) con los datos del material seleccionado.	La tabla de tesis refleja las acciones realizadas.	

El resultado de la codificación aparece en la figura 4.33 y 4.34.

**Figura 4.33 Interfaz para agregar una nueva tesis.**



**Figura 4.34** Interfaz que permite la administración de materiales.

**4.1.3.5 Ciclo 5. Módulo: Consultas**

Las actividades que se llevaron a cabo para la realización de este ciclo se presentan en la tabla 4.8 (plan de actividades).

**Tabla 4.8** Plan de actividades del ciclo no. 05, módulo: consultas.

<i>Plan de actividades del ciclo</i>			
Ciclo no.	05	Módulo a desarrollar:	Consultas
	Tiempo estimado	18 días	
	Objetivo del ciclo	Realizar el módulo de consulta de tesis por diferentes filtros	
<i>Actividades</i>			<i>Responsable</i>
1. Obtención y análisis de los requerimientos para este módulo			Analista
2. Diseño de la interfaz de para cada tipo de consulta			Diseñador
3. Desarrollo del módulo			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

**Etapa 1. Comprensión:** La figura 4.35 contiene la recolección de requisitos hecha para el módulo de consultas, y la figura 4.36 muestra el análisis realizado.

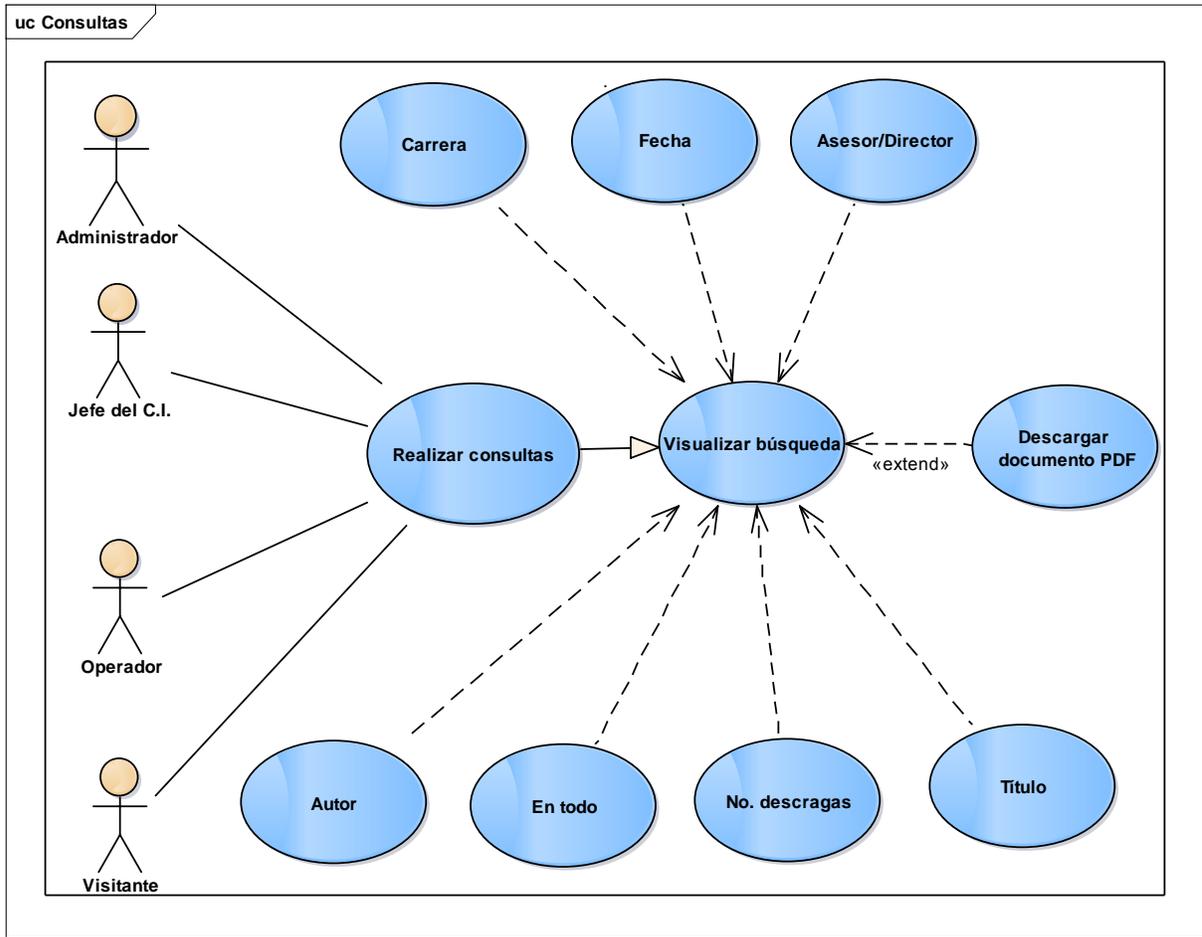
<i>Recolección de requisitos</i>		
<i>Formato del ciclo no.</i>	05	<i>Módulo</i> Consultas
<b>Requerimientos funcionales</b>		
<ul style="list-style-type: none"> <li>▪ RC05.1 Consultar las tesis almacenadas en la base de datos. La consulta de material podrá realizarse por: título, autor, asesor/director, fecha, carrera, todo o número de descargas</li> <li>▪ RC05.2 El resultado de la consulta deberá mostrar una lista de los materiales en una tabla con los siguientes datos: título, autor, asesor/director, fecha, carrera, PDF (documento) y número de descargas</li> </ul>		
<b>Requerimientos no funcionales</b>		
<ul style="list-style-type: none"> <li>▪ RNF-C05.1 Cada tipo de consulta debe tener un apartado propio.</li> <li>▪ RNF-C05.2 El sistema debe mostrar los resultados correspondientes a la búsqueda realizada.</li> <li>▪ RNF-C05.3 El proceso de consulta a los materiales debe tener un apartado dentro del manual de usuarios.</li> </ul>		

**Figura 4.35 Formato de recolección requisitos SARB-Ciclo 05.**

<i>Análisis de requisitos</i>	
<i>Formato del ciclo no.</i>	05 <i>Módulo</i> Consultas
<b>Especificación textual de los casos de uso</b>	
<i>Caso de uso:</i> Consultas	
<i>Actores</i>	Administrador, Jefe del C.I., Operador y Visitante
<i>Resumen</i>	Los actores de este caso de uso pueden consultar y descargar el material bibliográfico, el cual estará disponible en el módulo de consultas, estas consultas se pueden hacer por diferentes filtros y el sistema mostrará el resultado de la operación en una tabla.
<i>Secuencia</i>	<ol style="list-style-type: none"> <li>1. El actor selecciona el módulo de consultas</li> <li>2. <u>Elige</u> la opción de búsqueda: título, director/asesor, autor, título, todo, descargas o carrera e ingresa el valor solicitado</li> <li>3. El sistema solicita el dato de búsqueda</li> <li>4. El sistema realiza la búsqueda del material el dato ingresado</li> <li>5. El sistema muestra los resultados de la consulta, mostrados en una tabla con los datos siguientes: título, director/asesor, autor, fecha, no. descargas, carrera y PDF (documento)</li> </ol>
<i>Precondiciones</i>	Para la consulta de tesis no se requiere estar identificado dentro del sistema, sólo se ingresa a éste sin identificarse y se elige un filtro
<i>Post-condiciones</i>	Se muestra la lista de ejemplares de acuerdo al filtro elegido y dato ingresado
<i>Comentarios</i>	La consulta por número de descargas sólo interesa para el Administrador y el Jefe del C.I.

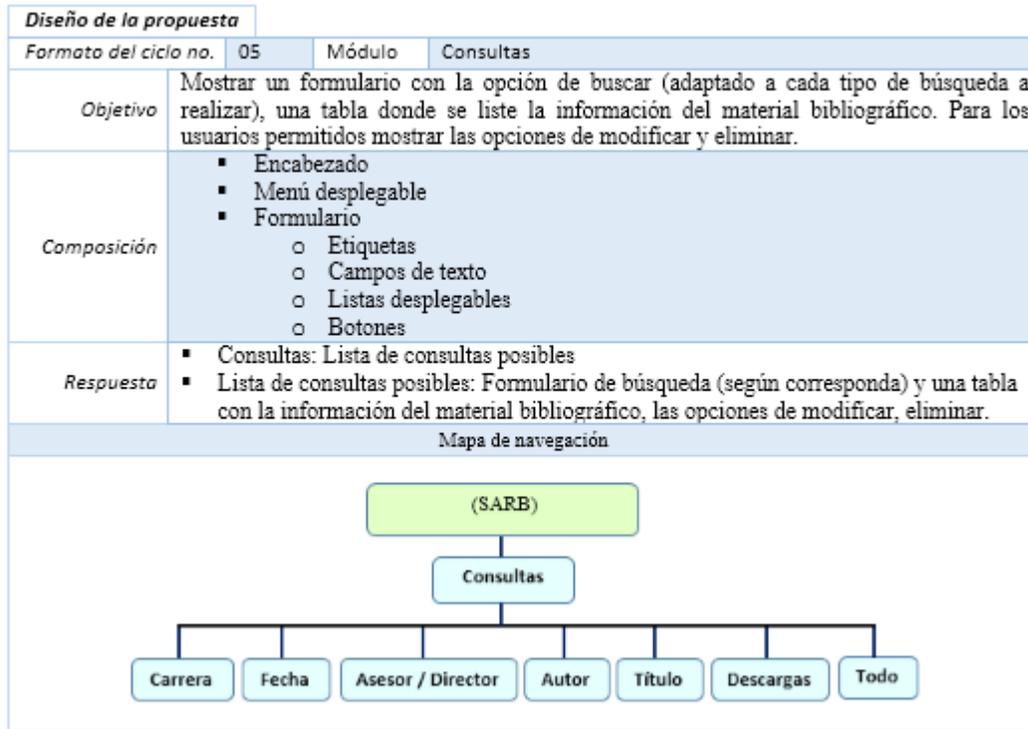
**Figura 4.36 Formato de análisis de requisitos SARB-Ciclo 05.**

El diagrama del caso de uso del módulo consultas se presenta en la figura 4.37.



**Figura 4.37 Diagrama del caso de uso SARB-Ciclo 05.**

**Etapa 2. Modelado:** El formato elaborado para esta etapa aparece en la figura 4.38, en el mapa de navegación que integra el formato de diseño de la propuesta se aprecia con claridad los distintos tipos de consultas que se pueden realizar dentro del sistema, lo cuales son los filtros que se ocuparan para recuperar la información que requiera el usuario.



**Figura 4.38** Diseño de la propuesta SARB-Ciclo 05.

**Etap 3. Elaboración:** La tabla 4.9 muestra la interacción entre niveles para el ciclo número cinco, con el que se obtendrá el desarrollo del módulo de consultas.

**Tabla 4.9** Interacción entre niveles: Consultas.

Interacción entre niveles (Arquitectura cliente-servidor)			
Formato del ciclo no.	05	Módulo	Consultas
Lógica de presentación	Lógica de negocio		Lógica de datos
Consultas: Esta sección muestra las opciones de filtro siguientes: <ul style="list-style-type: none"> <li>▪ Carrera</li> <li>▪ Fecha</li> <li>▪ Asesor/Director</li> <li>▪ Autor</li> <li>▪ Título</li> <li>▪ Descargas</li> <li>▪ Todo</li> </ul> Cada uno cuenta con un formulario a través del cual se ingresa el criterio de búsqueda.	El sistema realiza la búsqueda de acuerdo al filtro elegido y el dato ingresado, mostrando los resultados a través de una tabla con los materiales (tesis) encontrados y sus principales características. El material se puede descargar. Desde aquí igual se puede hacer parte de la gestión de materiales, pues se agregan las opciones de eliminar y modificar (solo para los actores con estos permisos). Esto se realiza desde materiales.php		La consulta se lleva a cabo dentro de la tabla <i>tesis</i> .

Las figuras 4.39 a la 4.46 muestran los resultados que se obtuvieron de la codificación que se realizó para este ciclo.



**Figura 4.39** Menú del sistema con los distintos filtros para las consultas.



**Figura 4.40** Interfaz para la consulta por carrera.

The screenshot shows a web interface with a blue header containing 'Inicio' with a home icon, 'Consultar Tesis' with a dropdown arrow, 'SARB', and 'Ayuda'. Below the header, the text 'Visualizar por fecha de publicación:' is displayed. There are two search options: 1) 'Ingresar año:' followed by a text input field containing 'e.g. 2016' and a blue 'Consulta' button. 2) 'O ingresar mes y año:' followed by a text input field containing 'e.g. 2016', a dropdown menu labeled 'Elegir mes', and a blue 'Consulta' button.

**Figura 4.41** Interfaz para la consulta por fecha.

The screenshot shows a web interface with a blue header containing 'Inicio' with a home icon, 'Consultar Tesis' with a dropdown arrow, 'SARB', and 'Ayuda'. Below the header, the text 'Visualizar por asesor/director:' is displayed. There is a row of 26 blue buttons, each containing a letter from 'A' to 'Z'. Below this row, the text 'O introducir las primeras letras:' is followed by a text input field and a blue 'Consulta' button.

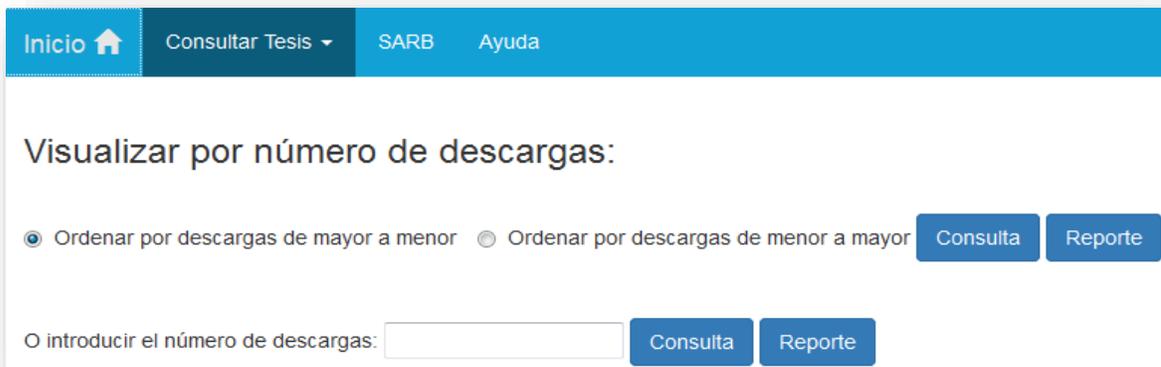
**Figura 4.42** Interfaz para la consulta por asesor / director.

The screenshot shows a web interface with a blue header containing 'Inicio' with a home icon, 'Consultar Tesis' with a dropdown arrow, 'SARB', and 'Ayuda'. Below the header, the text 'Visualizar por autor:' is displayed. There is a row of 26 blue buttons, each containing a letter from 'A' to 'Z'. Below this row, the text 'O introducir las primeras letras:' is followed by a text input field and a blue 'Consulta' button.

**Figura 4.43** Interfaz para la consulta por autor.



**Figura 4.44** Interfaz para la consulta por título.



**Figura 4.45** Interfaz para la consulta por número de descargas.



**Figura 4.46** Interfaz para la consulta en todo el contenido disponible.

**4.1.3.6 Ciclo 6. Módulo: Reportes**

La finalidad de desarrollar el ciclo número seis es el realizar la parte del sistema que genere los reportes que requiere el usuario, para lograrlo se planearon las actividades que conducirían a obtener lo esperado, como se puede ver en la tabla 4.10.

**Tabla 4.10 Plan de actividades del ciclo no. 06, módulo: reportes.**

<i>Plan de actividades del ciclo</i>			
Ciclo no.	06	Módulo a desarrollar:	Reportes
Tiempo estimado		8 días	
Objetivo del ciclo		Agregar a la sección de consultas la parte de reportes	
<i>Actividades</i>			<i>Responsable</i>
1. Obtención y análisis de los requerimientos para este módulo			Analista
2. Diseño de la interfaz para la generación de reportes			Diseñador
3. Desarrollo del módulo			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

**Eta** **1. Comprensión:** La obtención de requerimientos para el módulo de reportes se observa en la figura 4.47.

<i>Recolección de requisitos</i>			
<i>Formato del ciclo no.</i>	06	<i>Módulo</i>	Reportes
<i>Requerimientos funcionales</i>			
<ul style="list-style-type: none"> <li>▪ RC06.1 Generar reportes del material bibliográfico a partir de una consulta. El reporte podrá realizarse por: título, autor, asesor/director, fecha, carrera, todo o número de descargas</li> <li>▪ RC06.2 El resultado del reporte deberá mostrarse en un documento PDF en una lista de materiales con los siguientes datos: título, autor, asesor/director, fecha, carrera, PDF (documentos) y número de descargas</li> </ul>			
<i>Requerimientos no funcionales</i>			
<ul style="list-style-type: none"> <li>▪ RNF-C06.1 Cada tipo de reporte corresponde a un tipo de consulta, dicho esta manera, puede agregarse solo un apartado u opción de generar reporte a partir de una consulta.</li> <li>▪ RNF-C06.2 El sistema debe mostrar el reporte correspondiente en un documento PDF, en base los resultados de la consulta realizada.</li> <li>▪ RNF-C06.3 El proceso de generar reportes debe tener un apartado dentro del manual de usuarios.</li> </ul>			

**Figura 4.47 Formato de recolección requisitos SARB-Ciclo 06.**

El formato de análisis de requisitos empleado para el ciclo seis con la información correspondiente se puede ver en la figura 4.48 y su diagrama de caso de uso en la figura 4.49.

Análisis de requisitos	
Formato del ciclo no.	06
Módulo	Reportes
Especificación textual de los casos de uso	
Caso de uso: Reportes	
Actores	Administrador, Jefe del C.I. y Operador
Resumen	Los actores de este caso de uso pueden generar reportes del material bibliográfico, esta acción estará disponible junto con el módulo de consultas, estos reportes se pueden hacer por en base a las diferentes filtros de consulta y el sistema mostrará el resultado de la operación en un listado presentado en un documento PDF.
Secuencia	<ol style="list-style-type: none"> <li>1. El actor selecciona la sección de consultas/reportes</li> <li>2. Se elige la opción de búsqueda: título, director/asesor, autor, título, todo, descargas o carrera. Se ingresa el dato requerido y se selecciona la opción de generar reporte</li> <li>3. El sistema solicita el dato de búsqueda</li> <li>4. El sistema realiza la búsqueda del material(es) del dato ingresado</li> <li>5. El sistema muestra los resultados de la consulta y genera un reporte en un documento en formato PDF con los datos siguientes: título, director/asesor, autor, fecha, no. Descargas y carrera, incluyendo la fecha en que se realizó y el título correspondiente al reporte.</li> </ol>
Precondiciones	Se requiere estar identificado dentro del sistema
Post-condiciones	Se genera el reporte de acuerdo al filtro elegido y el dato ingresado
Comentarios	

Figura 4.48 Formato de análisis de requisitos SARB-Ciclo 06.

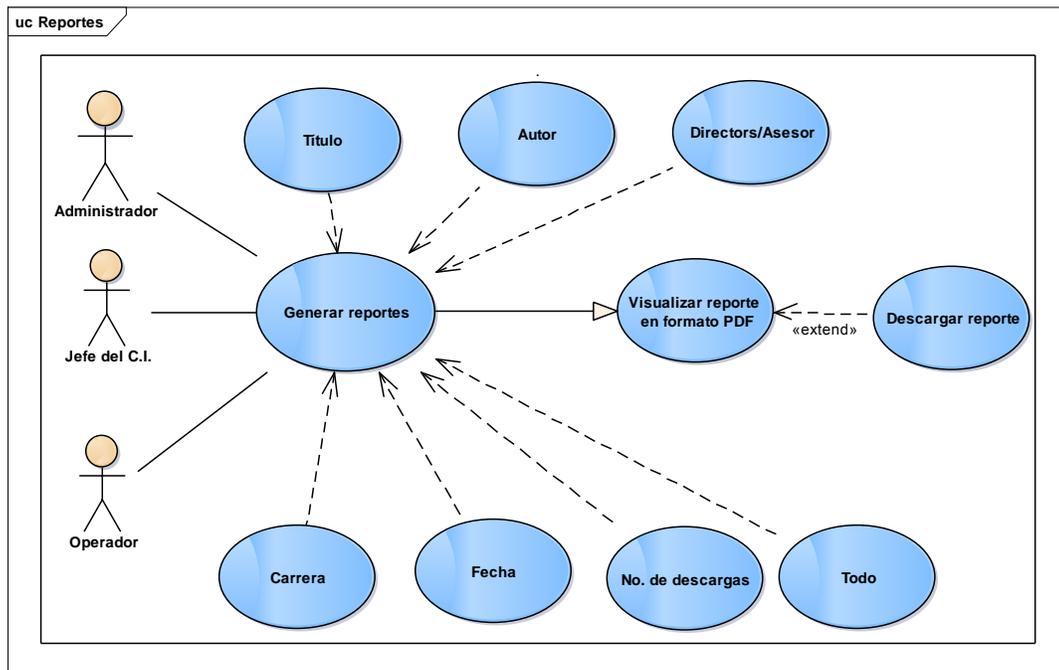
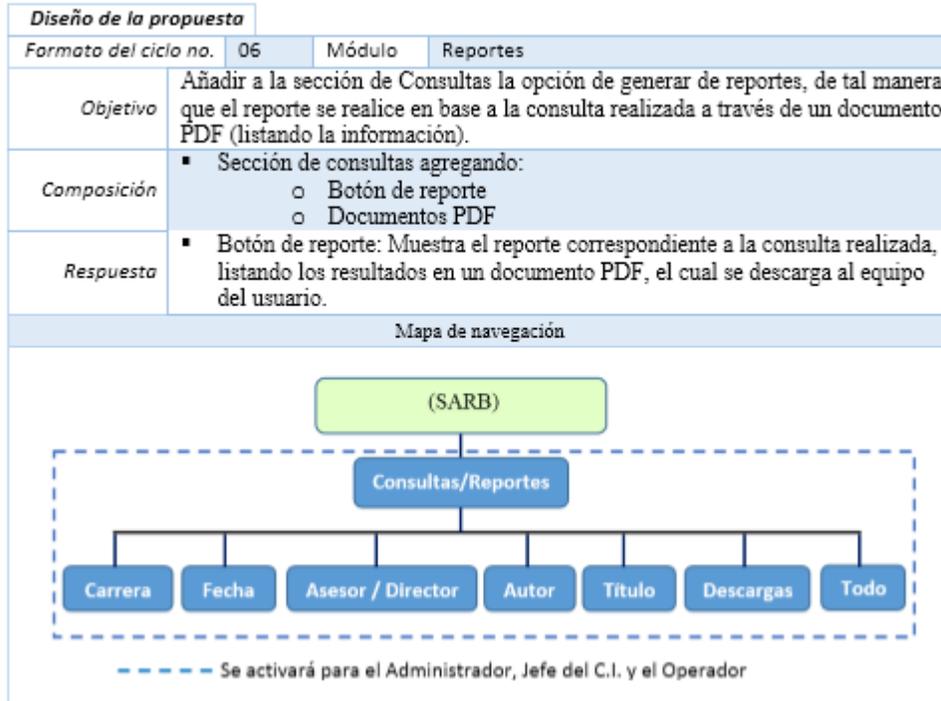


Figura 4.49 Diagrama del caso de uso SARB-Ciclo 06.

**Etapa 2. Modelado:** La figura 4.50 muestra el diseño de la propuesta con su descripción.



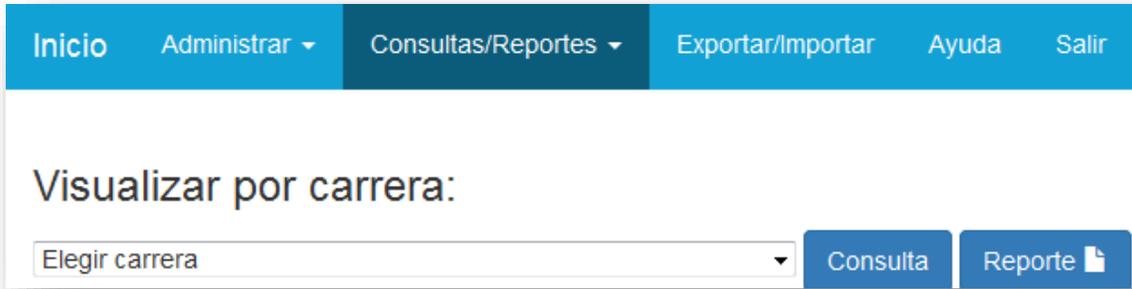
**Figura 4.50** Diseño de la propuesta SARB-Ciclo 06.

**Etapa 3. Elaboración:** La tabla 4.11 indica la interacción entre niveles para el módulo de reportes.

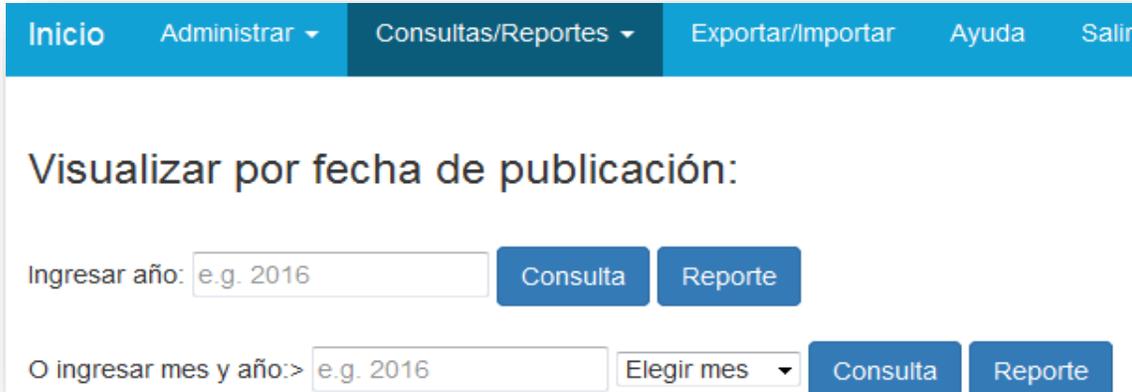
**Tabla 4.11** Interacción entre niveles: Reportes.

Interacción entre niveles (Arquitectura cliente-servidor)			
Formato del ciclo no.	06	Módulo	Reportes
Lógica de presentación	Lógica de negocio		Lógica de datos
Consultas/Reportes: Dentro de la sección de consultas se agrega un botón que permite generar un reporte a partir de una consulta hecha, mostrándolo en un documento PDF. Es posible obtener reportes por los siguientes conceptos: <ul style="list-style-type: none"> <li>▪ Carrera</li> <li>▪ Fecha</li> <li>▪ Asesor/Director</li> <li>▪ Autor</li> <li>▪ Título</li> <li>▪ Descargas</li> <li>▪ Todo</li> </ul>	El sistema realiza la búsqueda de acuerdo al filtro elegido y el dato ingresado, mostrando los resultados a través de un reporte, que los lista en un documento con extensión .pdf, mostrando el título del reporte y la fecha de elaboración Esto se realiza desde materiales.php		La consulta para generar el reporte se realiza dentro de la tabla <i>tesis</i> .

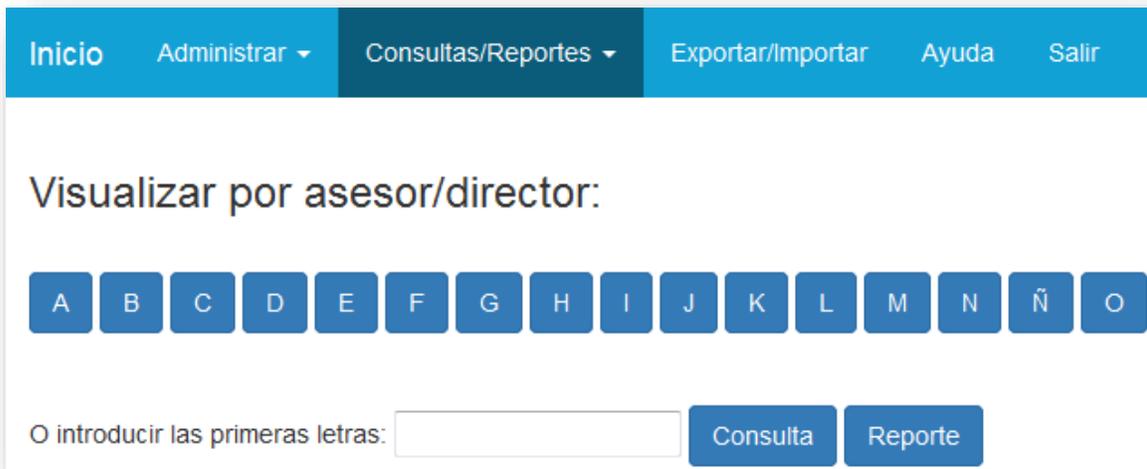
Como se indicó, a la parte de consultas que requiere autenticación se le agregó un botón para generar un reporte en base al filtro seleccionado y los datos ingresados, generando así un documento PDF con los datos obtenidos, las figuras 4.51 a la 4.57 muestran los resultados obtenidos de este ciclo.



**Figura 4.51 Adición del botón de reporte en la búsqueda por carrera.**



**Figura 4.52 Adición del botón de reporte en la búsqueda por fecha.**



**Figura 4.53** Adición del botón de reporte en la búsqueda por asesor / director.



**Figura 4.54** Adición del botón de reporte en la búsqueda por autor.



**Figura 4.55** Adición del botón de reporte en la búsqueda por título.



**Figura 4.56** Adición del botón de reporte en la búsqueda por número de descargas.



**Figura 4.57** Adición del botón de reporte en la búsqueda en todo el contenido.

**4.1.3.7 Ciclo 7. Módulo: Exportar e importar la base de datos**

Con la elaboración de este ciclo se busca que el usuario cuente con una forma de respaldar y recuperar la información más importante que alimenta a la base de datos y que hace funcionar al sistema. Para desarrollar el módulo de exportar e importar la base de datos, con lo que se logrará el objetivo del ciclo, el plan de actividades que se determinó se presenta en la tabla 4.12.

**Tabla 4.12 Plan de actividades del ciclo no. 07, módulo: exportar e importar la base de datos.**

<i>Plan de actividades del ciclo</i>			
Ciclo no.	07	Módulo a desarrollar:	Exportar e importar la base de datos
	Tiempo estimado	5 días	
	Objetivo del ciclo	Realizar el módulo de exportar / importar de la base de datos las tablas de usuarios y tesis	
<i>Actividades</i>			<i>Responsable</i>
1. Obtención y análisis de los requerimientos para este módulo			Analista
2. Diseño de la interfaz donde se podrá exportar e importar			Diseñador
3. Desarrollo del módulo			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

**Etapa 1. Comprensión:** Para esta etapa se requiere de la recolección de requisitos (figura 4.58) y el análisis de los mismos (figura 4.59) con el empleo de los formatos definidos. El diagrama del caso de uso para el módulo de exportar e importar la base de datos de muestra en la figura 4.60.

<i>Recolección de requisitos</i>			
<i>Formato del ciclo no.</i>	07	<i>Módulo</i>	Exportar / Importar la base de datos
<i>Requerimientos funcionales</i>			
<ul style="list-style-type: none"> <li>▪ RC07.1 Exportar o importar la información (registros) almacenada dentro de la base de datos referente a los usuarios y materiales (tesis). La finalidad de exportar, es la de mantener un respaldo de la información relevante que se encuentra dentro de la base de datos, de esta forma se pueda recuperar (importar) dicha información en un caso dado.</li> </ul>			
<i>Requerimientos no funcionales</i>			
<ul style="list-style-type: none"> <li>▪ RNF-C07.1 El tipo de archivo requerido para este caso de uso es de extensión .csv</li> <li>▪ RNF-C07.2 El proceso exportar/importar debe tener un apartado dentro del manual de usuarios.</li> </ul>			

**Figura 4.58 Formato de recolección requisitos SARB-Ciclo 07.**

Recolección de requisitos	
Formato del ciclo no.	07
Módulo	Exportar / Importar la base de datos
Especificación textual de los casos de uso	
Caso de uso: Exportar / Importar base de datos	
Actores	Administrador
Resumen	El administrador debe exportar/importar la base de datos (registros correspondientes a usuarios y materiales) cada vez que lo considere pertinente, el sistema debe reflejar esta operación.
Secuencia	<ol style="list-style-type: none"> <li>1. El actor selecciona el módulo de exportar/importar</li> <li>2. Selecciona que registros desea respaldar o recuperar: usuarios o materiales</li> <li>3. El actor elige la acción a realizar (exportar o importar).</li> <li>4. El sistema realiza la acción correspondiente: si se exportó se genera un archivo con los datos; si se importó, se actualiza la base de datos con los datos importados</li> </ol>
Precondiciones	Se requiere estar identificado dentro del sistema Si se elige importar, se debe seleccionar el archivo que contenga los registros correspondientes
Post-condiciones	Se descarga el archivo donde se hizo el respaldo, si se eligió exportar El sistema debe reflejar los datos actualizados, si es que se llevó a cabo una recuperación (importar)

Figura 4.59 Formato de análisis de requisitos SARB-Ciclo 07.

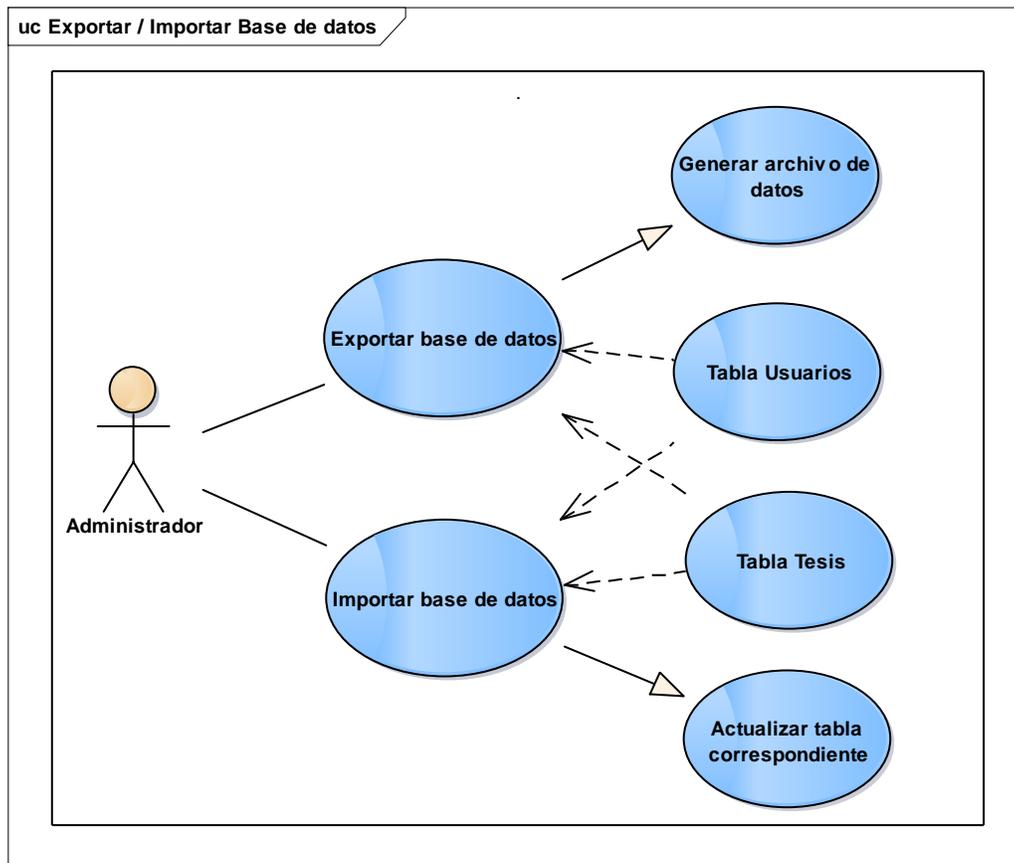


Figura 4.60 Diagrama del caso de uso SARB-Ciclo 07.

**Etapa 2. Modelado:** Con el formato de diseño de la propuesta que se utiliza en la etapa de modelado, se presenta el correspondiente al ciclo seis en la figura 4.61.



**Figura 4.61** Diseño de la propuesta SARB-Ciclo 08.

**Etapa 3. Elaboración:** Para la parte de codificación se puede apreciar de manera descrita la interacción entre niveles que tendrá el módulo de exportar e importar la base de datos en la tabla 4.13.

**Tabla 4.13** Interacción entre niveles: Exportar e importar la base de datos.

Interacción entre niveles (Arquitectura cliente-servidor)		
Formato del ciclo no.	07	Módulo Exportar e importar base de datos
Lógica de presentación	Lógica de negocio	Lógica de datos
Exportar/Importar: En el módulo de exportar/importar se agrega una sección para cada acción, cada una contará con botones de radio para elegir sobre qué datos se va a trabajar (tesis o usuarios), así como también el	<ul style="list-style-type: none"> <li>▪ Importar: Una vez que el usuario marque el botón radio ya sea de usuarios o tesis, al igual que haya elegido el archivo con extensión .cvs que contenga los datos a importar presiona el botón importar para completar la operación.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Importar: los datos obtenidos del archivo .cvs se agregan a la tabla correspondiente (tabla <i>usuarios</i> o <i>tesis</i>), en la cual los datos existentes son remplazados por los del archivo.</li> </ul>

Continuación de la tabla 4.13.

<p>botón correspondiente para efectuar la tarea (importar o exportar), según sea el caso. Para importar se añade un botón que permita buscar y agregar el archivo que contenga los datos.</p>	<ul style="list-style-type: none"> <li>▪ Exportar: el sistema genera un archivo con extensión .csv, con los datos de la opción elegida (tesis o usuarios), el cual se descarga en la computadora del usuario, con el nombre y la fecha correspondientes cuando haya presionado el botón exportar.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Exportar: Se obtienen todos los datos de la tabla usuarios o tesis, dichos datos se almacenan en un archivo csv.</li> </ul>
---	--	--

Las interfaces obtenidas resultado del desarrollo del módulo para exportar e importar se encuentran en la figura 4.62.



**Figura 4.62 Interfaz para exportar e importar la base de datos.**

#### 4.1.3.8 Ciclo 8. Módulo: Sección de ayuda

El último ciclo desarrollado para el módulo de la sección de ayuda como en los anteriores realizó un plan de actividades, para alcanzar el objetivo de realizar la parte del sistema que brinde al usuario la información necesaria para manipular el sistema en cualquier dificultad que se le presente. La tabla 4.14 muestra el plan de actividades del ciclo.

**Tabla 4.14 Plan de actividades del ciclo no. 08, módulo: sección de ayuda.**

<i>Plan de actividades del ciclo</i>			
Ciclo no.	08	Módulo a desarrollar:	Sección de ayuda
Tiempo estimado		2 días	
Objetivo del ciclo		Realizar la sección de ayuda	
<i>Actividades</i>			Responsable
1. Obtención y análisis de los requerimientos para este módulo			Analista
2. Diseño de la interfaz de para la sección de ayuda			Diseñador
3. Desarrollo del módulo			Desarrollador
4. Revisar que el resultado del desarrollo siga el diseño			Tester
5. Revisar que se obtenga el incremento al finalizar el ciclo			Dueño del software
6. Asegurar el buen cumplimiento de la metodología			Guía

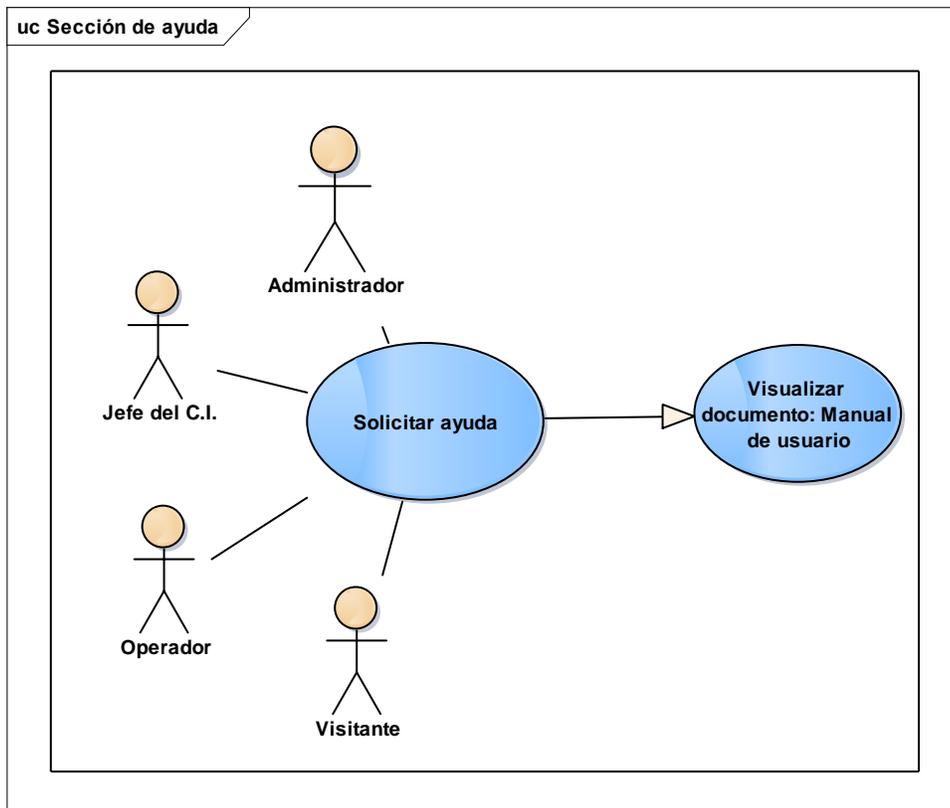
**Etap 1. Comprensión:** El empleo del formato para la recolección de requisitos para el ciclo ocho se observa en la figura 4.63, así como el correspondiente al análisis en la figura 4.64 y en la figura 4.65 el diagrama del caso de uso.

<i>Recolección de requisitos</i>			
<i>Formato del ciclo no.</i>	08	<i>Módulo</i>	Sección de ayuda
Requerimientos funcionales			
<ul style="list-style-type: none"> <li>▪ RC08.1 Mostrar el manual de usuario en caso de surgir alguna duda en cuanto al funcionamiento del sistema</li> </ul>			
Requerimientos no funcionales			
<ul style="list-style-type: none"> <li>▪ RNF-C08.1 El manual de usuario debe mostrar de manera gráfica cada procedimiento</li> </ul>			

**Figura 4.63 Formato de recolección requisitos SARB-Ciclo 08.**

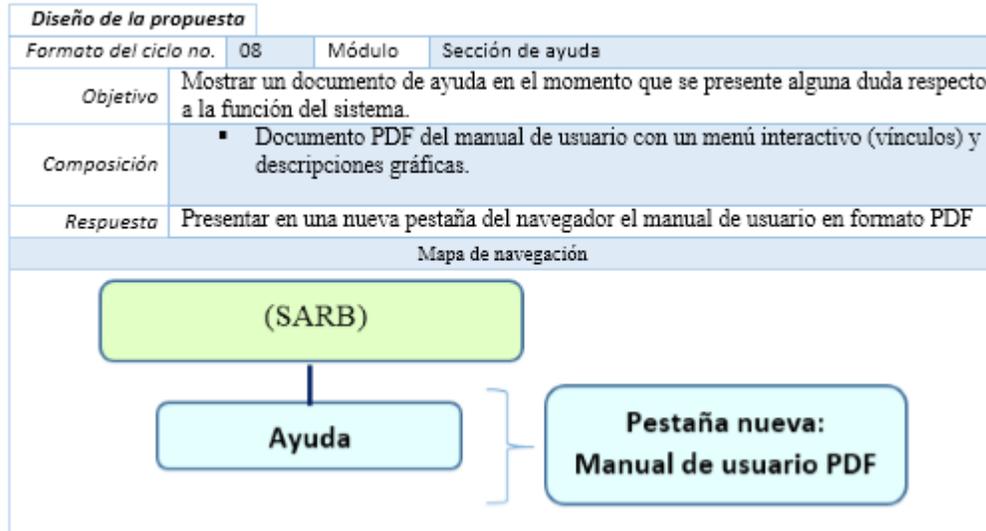
<b>Análisis de requisitos</b>	
Formato del ciclo no.	08
Módulo	Sección de ayuda
Especificación textual de los casos de uso	
Caso de uso: Sección de ayuda	
Actores	Administrador, Jefe del C.I., Operador, Visitante
Resumen	Todos los actores pueden consultar el manual de usuario cuando lo requieran
Secuencia	<ol style="list-style-type: none"> <li>1. El actor selecciona la opción de ayuda</li> <li>2. Se muestra en pantalla en documento en PDF del manual de usuario</li> <li>3. Del índice elige el proceso que desea conocer o consultar</li> <li>4. El documento se muestra en la sección elegida</li> </ol>
Precondiciones	No se requiere estar identificado dentro del sistema
Post-condiciones	El sistema sólo muestra la manera en cómo debe realizarse cada acción
Comentarios	

**Figura 4.64** Formato de análisis de requisitos SARB-Ciclo 08.



**Figura 4.65** Diagrama del caso de uso SARB-Ciclo 08.

**Etapa 2. Modelado:** El formato de diseño (figura 4.66) de la propuesta contiene una descripción textual y de manera gráfica el mapa de navegación, elementos que muestran la estructura que tendrá el sistema al finalizar la construcción del módulo.



**Figura 4.66** Diseño de la propuesta SARB-Ciclo 08.

**Etapa 3. Elaboración.** La interacción entre niveles, así como se ha realizado para ciclos anteriores, para la sección de ayuda, siendo ésta la última en desarrollarse no es la excepción para realizarse y la información se muestra en la tabla 4.15.

**Tabla 4.15** Interacción entre niveles: Sección de ayuda.

Interacción entre niveles (Arquitectura cliente-servidor)		
Formato del ciclo no.	08	Módulo Sección de ayuda
Lógica de presentación	Lógica de negocio	Lógica de datos
Ayuda: Sólo se requiere presionar esta opción en la barra de menú.	El sistema desplegará en una nueva pestaña y en formato PDF el manual de usuario.	

En la figura 4.67 es posible ver el resultado después de la codificación de la sección de ayuda.



Figura 4.67 Sección de ayuda: Manual de usuario.

#### 4.1.4 Pruebas y liberación del SARB

Las pruebas realizadas, se llevaron a cabo para cada ciclo, tal y como se determinó al establecer la metodología propuesta en el capítulo 3 mediante casos de prueba, diseñados en base a cada módulo desarrollado. La descripción detallada de éstas se presenta como parte del capítulo 5.

La implementación se sistema final de una versión preliminar del sistema fue aprobada haciendo uso del formato de liberación de ciclo, adaptándolo para tal efecto. Cada liberación correspondiente a cada ciclo fue aprobado, pero siendo la aprobación general del SARB la más relevante se muestra en la figura 4.68.

Liberación del SARB		
No. de ciclos incluidos.	08	Módulo integrados
<ol style="list-style-type: none"> <li>1. Desarrollo de la base de datos</li> <li>2. Acceso al sistema</li> <li>3. Gestión de usuarios</li> <li>4. Gestión de materiales</li> <li>5. Consultas</li> <li>6. Reportes</li> <li>7. Exportar e importar base de datos</li> <li>8. Sección de ayuda</li> </ol>		
Integrante:		Firma de aprobación
Diseño del software 1: José Juan Hernández Mora		<i>Va. B.</i>
Diseño de software 2: Astrid Ariadna Torres Fernández		<i>Astrid A. F. T.</i>
Analista: Rocío Hernández Acoltzi		<i>[Firma]</i>
Diseñador: Lidia Cortés Ruiz		<i>[Firma]</i>
Desarrollador: René López Caballero		<i>[Firma]</i>
Tester: Said Pérez Flores		<i>[Firma]</i>
Firma de conocimiento del guía: Lorena Zamora Velázquez		<i>[Firma]</i>

Figura 4.68 Aprobación de liberación del SARB.

### **4.1.5 Manual de usuario**

El manual de usuario se fue desarrollando en cada ciclo, conforme se construía el módulo se realizaba su parte correspondiente de funcionamiento y utilización dentro de este manual, el cual se puede consultar en el Anexo 1.

En este capítulo se hizo uso de la metodología propuesta, con la que se construyó el SARB a través de ciclos para la elaboración de cada módulo que integra al sistema, lo que representa una iteración que da como resultado un incremento, lo cual va desarrollando a la aplicación web hasta concluir con lo que es una primera versión. Los itinerarios y reportes de las reuniones, al igual que los formatos de sugerencias y cambios se encuentran en el anexo 5.

## Capítulo 5 Evaluación y resultados

En este capítulo se describe la evaluación aplicada a la metodología propuesta en cuanto a agilidad, al igual que las pruebas al caso de estudio para la construcción del SARB (para el que se empleó la metodología planteada en el capítulo 3). Se presentan los resultados obtenidos para los casos de prueba aplicados a cada uno de los ciclos, las pruebas de integración al concluir el desarrollo de cada módulo (estando ya todos unidos en un único sistema), finalizando con las pruebas generales del SARB.

### 5.1 Evaluación de la metodología en cuanto a agilidad

La metodología que en este trabajo se propone está orientada al desarrollo ágil, por lo que se hace uso de un *Framework para Evaluación de Metodologías Ágiles* (Mendes, Estevez, & Fillotrani, 2010) para evaluarla, esta herramienta se considera adecuada debido a que mide que tanto las metodologías cumplen con los postulados declarados en el Manifiesto Ágil.

El Framework utilizado establece mediciones para cada uno de los cuatro postulados, las cuales satisfacen la Teoría Representacional de la Medición y se definen usando escala de intervalos. Para lo anterior se fórmula lo siguiente (figura 5.1):

$$m(P_i) = m(P_{i,1}) + m(P_{i,2}) \quad i = 1..4$$

**Figura 5.1 Formulación de Framework de la metodología propuesta en cuanto agilidad (Mendes, Estevez, & Fillotrani, 2010).**

En el que la medida de cada postulado  $m(P_i)$  es igual a la suma de las medidas de los atributos relacionados  $m(P_{i,1})$  y  $m(P_{i,2})$ , en donde  $(P_i, i=1..4)$  fueron expresados como valoración dos atributos  $(P_{i,1})$  y  $(P_{i,2})$ . El atributo que los principios intentan enfatizar (atributo positivo) se mide en una escala de 0 a 5, y el otro atributo (atributo negativo) en una escala de -5 a cero. De este modo, cada principio podría obtener una medida entre -5 en el caso que ambos

atributos tomen el peor valor (-5 el atributo negativo y 0 el atributo positivo), y 5 en el caso que ambos atributos tomen el mejor valor (0 el atributo negativo y 5 el atributo positivo). Si se obtiene un valor cero o cercano a cero, significa que la metodología no valora significativamente el atributo positivo por sobre el negativo, en cuyo caso, el principio del Manifiesto Ágil no se satisface plenamente (Mendes, Estevez, & Fillotrani, 2010). La aplicación de este framework a la metodología propuesta se presenta a continuación. Con respecto al primer postulado se muestra la tabla 5.1 con el formato del Framework para la Evaluación.

**Tabla 5.1 Aplicación de la evaluación de acuerdo al primer postulado.**

<b>P1 Valorar al individuo y las interacciones del equipo por sobre el proceso y las herramientas.</b>			
<b>P1.1</b>	<b>Valorar al individuo y las interacciones.</b>	<b>P1.2</b>	<b>Valorar el proceso y las herramientas.</b>
<b>valor</b>	<b>Descripción</b>	<b>valor</b>	<b>Descripción</b>
4	No define roles para individuos.	-3	Define actividades, entregables, herramientas de desarrollo y de gestión.
5	Clara definición de roles para individuos.	-4	Define actividades, entregables y herramientas de desarrollo.
5	Clara definición de roles y responsabilidades.	-2	Define actividades y entregables.
4	Clara definición de roles, responsabilidades y conocimientos técnicos.	-4	Define actividades para cada iteración.
4	Clara definición de roles, responsabilidades, conocimientos técnicos e interacciones entre miembros del equipo de trabajo.	-3	Define actividades para el proyecto pero no a nivel de cada iteración.
Promedio		4	Promedio
			-3

**Postulado P1.** La metodología obtiene para el atributo P1.1 un valor de 4, por lo que se deduce que valora al individuo, define roles y responsabilidades. Para el atributo p1.2 se obtiene un valor de -3, ya que debido al plan de comunicación que describe las herramientas a utilizar con la finalidad de construir un léxico común entre los participantes, la definición de actividades sólo de manera general en cada iteración, la determinación de entregables y el plan de gestión con el que cuenta el guía, todo esto genera el puntaje, lo que lo reduce es que no se lleva a cabo de manera detallada.

En la tabla 5.2 se muestra la evaluación para el segundo postulado, el cual se refiere a Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva.

**Tabla 5.2 Aplicación de Evaluación de acuerdo al segundo postulado.**

P2 Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva.			
P2.1	Valorar desarrollo de software que funcione.	P2.2	Valorar documentación exhaustiva.
valor	Descripción	valor	Descripción
4	Generar entregable al finalizar el proyecto.	-3	Requiere documentación detallada al comienzo del proyecto.
5	Generar entregable con testing satisfactorio al finalizar cada iteración.	-2	Requiere sólo documentación para comenzar a implementar la funcionalidad incluida en una iteración.
4	Generar entregable con testing satisfactorio e integrado con el resto de las funciones al finalizar la iteración.	-1	No requiere documentación para comenzar a implementar la funcionalidad incluida en una iteración.
Promedio		Promedio	
4		-2	

**Postulado P2.** Obtiene 4 puntos en el atributo P2.1, lo que significa que al concluir un ciclo (iteración) se debe producir un entregable, que forma parte del sistema final, el cual incluye las pruebas con resultados satisfactorios. Se alcanza un valor de -2 para el atributo P2.2 ya que para se requiere documentación (plan de comunicación) al comienzo del proyecto que consta de varias secciones aunque no es tan detallada, y debido a que los integrantes se encuentran distribuidos el documentar es una actividad importante para la comunicación, pero no de forma excesiva puesto que esto satura y confunde a los desarrolladores.

En la tabla 5.3 se muestra la evaluación para el tercer postulado, el cual se refiere a valorar la colaboración con el cliente por sobre la negociación contractual.

**Tabla 5.3 Aplicación de Evaluación de acuerdo al tercer postulado.**

<b>P3 Valorar la colaboración con el cliente por sobre la negociación contractual.</b>			
<b>P3.1</b>	<b>Valorar la colaboración con el cliente.</b>	<b>P3.2</b>	<b>Valorar la negociación contractual.</b>
<b>valor</b>	<b>Descripción</b>	<b>valor</b>	<b>Descripción</b>
5	Cliente colabora a demanda del equipo.	-1	Existe contratación detallada al inicio y no se aceptan cambios.
5	Cliente es parte del equipo, responde consultas y planifica las iteraciones.	-3	La contratación exige contemplar cambios durante el proyecto.
4	Cliente es parte del equipo, responde consultas, planifica iteraciones, y colabora en la escritura de requerimientos y pruebas.	0	El contrato por la construcción del producto no aporta valor al producto.
Promedio		5	
Promedio		-1	

**Postulado P3.** La metodología obtiene un valor de 5 en el atributo P3.1 pues se considera al cliente como miembro del equipo, el cual está involucrado tanto en la parte de planificación de cada ciclo como en la pruebas, aunque no escribe los requerimientos si los revisa. Se evalúa el atributo P3.2 con valor de -1 puesto que aunque no se exige contemplar cambios durante el proyecto si se considera.

En la tabla 5.4 se muestra la evaluación para el tercer postulado, el cual se refiere a Valorar la respuesta al cambio por sobre el seguimiento de un plan.

**Tabla 5.4 Aplicación de Evaluación de acuerdo al cuarto postulado.**

<b>P4 Valorar la respuesta al cambio por sobre el seguimiento de un plan.</b>			
<b>P4.1</b>	<b>Valorar la respuesta al cambio.</b>	<b>P4.2</b>	<b>Valorar el seguimiento de un plan.</b>
<b>valor</b>	<b>Descripción</b>	<b>valor</b>	<b>Descripción</b>
5	No prevé incorporar cambios durante la ejecución del proyecto.	-2	Define plan detallado al inicio del proyecto.
4	Prevé introducir sólo cambios de alta prioridad.	-1	Define un plan detallado de iteraciones, no acepta cambios durante una iteración.
4	Permite la evolución y el cambio, pero no es recomendable en la iteración en curso.	-1	Define un plan para cada iteración, que puede ser modificado.
1	Permite introducir cambios en la iteración en curso.	0	No define planificación alguna.
Promedio		4	
Promedio		-1	

**Postulado P4.** En el atributo P4.1 se observa un valor promedio de 4, esto debido aunque se contemplan cambios durante la ejecución del proyecto no se recomienda realizarlos durante la iteración en curso, puesto que tendría que modificarse la planificación del ciclo y reorganizar las actividades. Se logra un valor de -1 en P4.2, pues al principio del proyecto aunque no existe un plan detallado para la construcción del software, si existe uno para el plan de comunicación, la ventaja es que la mayor parte de dicho plan se va elaborando y adaptando a lo largo del proyecto.

Por último se concentran los resultados de la aplicación del Framework en la tabla 5.5, haciendo uso de la plantilla proporcionada por éste.

**Tabla 5.5 Resultados al aplicar el *framework* a la metodología propuesta.**

Postulados											
Primer postulado			Segundo Postulado			Tercer postulado			Cuarto postulado		
P1.1	P2.2	Total	P2.1	P2.2	Total	P3.1	P3.2.	Total	P4.1	P4.2	Total
4	-3	1	4	-2	2	5	-1	4	4	-1	3

Analizando los resultados obtenidos:

- *Valor para el postulado 1:* Se obtiene un total de 1 punto.
- *Valor para el postulado 2:* Se obtiene un total de 2 puntos.
- *Valor para el postulado 3:* Se obtiene un total de 4 puntos.
- *Valor para el postulado 4:* Se obtiene un total de 3 puntos.

En base a la evolución realizada se llega a la conclusión de que la metodología propuesta satisface los postulados ágiles, esto se determinó ya que el valor de cada postulado debe ser mayor a cero para que se considere como una metodología ágil.

### 5.3 Pruebas generadas en cada ciclo.

Como parte de la etapa tres de la metodología propuesta, correspondiente a elaboración y pruebas, en esta sección se presentan los resultados de las pruebas que se realizaron en cada ciclo con el formato correspondiente.

La tabla 5.6 describe el caso de prueba que se realizó para el ciclo número dos del módulo acceso a usuarios.

**Tabla 5.6 Caso de prueba ciclo no. 02, módulo acceso al sistema.**

<i>Caso de prueba</i>			
Formato del ciclo	02	Módulo	Acceso al sistema
<b>Caso de prueba</b>	<b>Resultado esperado</b>		<b>Resultado</b>
Validar el acceso al sistema del al administrador.	Al momento en que accede al sistema el administrador se muestra el menú con todas las funciones.		Correcto
Validar el acceso al sistema al jefe del centro de información.	Al momento en que accede al sistema el jefe del centro de información se muestra el menú con las opciones de gestión de usuarios, gestión de materiales, consultas, generar reportes y ayuda.		Correcto
Validar el acceso al sistema al operador.	Al momento en que accede al sistema el operador se muestra el menú con las opciones de gestión de materiales pero sólo para agregar tesis, permitiéndosele eliminar o modificar el último material que haya introducido, al igual que las funciones de consultas, generar reportes y ayuda.		Correcto
Acceso al sistema por parte del visitante.	Al momento que el visitante ingresa al sistema no requiere autenticarse, tiene sólo acceso a consultas y ayuda.		Correcto
Identificar datos incorrectos.	Al no ingresar los datos correctos de nombre de usuario y contraseña estos no coinciden, por lo que no se muestran las funciones restringidas y se permanece en la página de inicio se sesión.		Correcto

En la tabla 5.7 se muestran los casos de prueba para la gestión de usuarios con resultados satisfactorios

**Tabla 5.7 Caso de prueba ciclo no. 03, módulo gestión de usuarios.**

<i>Caso de prueba</i>			
Formato del ciclo	03	Módulo	Gestión de usuarios
<b>Caso de prueba</b>	<b>Resultado esperado</b>		<b>Resultado</b>
Agregar usuario	Agregar un usuario únicamente por el administrador o jefe del centro de información.		Correcto
Modificar usuario	Poder realizar modificaciones de los usuarios existentes.		Correcto
Eliminar usuario	Eliminar al usuario seleccionado de la base de datos.		Correcto
Duplicación de usuarios	Comprobar que no se agregue un usuario existente.		Correcto
Validación de datos	Considerando que todos los campos se requieren al agregar un usuario se debe asegurar que no quede ninguno vacío.		Correcto
Administrar usuarios	Mostrar todos los usuarios introducidos en la base de datos y revisar que como se vayan agregando se visualicen en la tabla de administrar usuarios.		Correcto
Confirmación de datos	Cuando se requiere eliminar, agregar o modificar un usuario el sistema debe solicitar confirmación de la acción antes de llevarla a acabo.		Correcto
Validación de tipo dato.	El usuario en el campo de tipo puesto no podrá ingresar alguno inexistente.		Correcto

Para el ciclo de gestión de materiales en la tabla 5.8 se describen los casos de prueba.

**Tabla 5.8 Caso de prueba ciclo no.04, módulo gestión de materiales.**

<i>Caso de prueba</i>			
Formato del ciclo	04	Módulo	Gestión de materiales
<b>Caso de prueba</b>	<b>Resultado esperado</b>		<b>Resultado</b>
Agregar tesis.	Agregar una tesis únicamente por el administrador, jefe del centro de información u operador.		Correcto
Modificar tesis.	Poder realizar modificaciones de las tesis existentes.		Correcto
Eliminar tesis.	Eliminar la tesis seleccionado de la base de datos.		Correcto
Duplicación de materiales.	Comprobar que no se agregue un usuario existente.		Correcto
Validación de datos.	Considerando que todos los campos se requieren al agregar una tesis se debe asegurar que no quede ninguno vacío.		Correcto
Administrar materiales.	Mostrar los materiales introducidos en la base de datos y revisar que como se vayan agregando se actualice la tabla de administrar materiales.		Correcto
Confirmación de datos.	Cuando se requiere eliminar, agregar o modificar una tesis el sistema debe solicitar confirmación de la acción antes de llevarla a acabo.		Correcto
Validación de rango de dato.	El usuario en el campo de tipo fecha no podrá introducir una mayor a la actual.		Correcto

La tabla 5.9 corresponde al caso de prueba para el módulo de consultas, en la que es posible ver que los resultados han sido satisfactorios.

**Tabla 5.9 Caso de prueba ciclo no. 05, módulo consultas.**

<i>Caso de prueba</i>			
Formato del ciclo	05	Módulo	Consultas
<b>Caso de prueba</b>	<b>Resultado esperado</b>		<b>Resultado</b>
Seleccionar un tipo de consulta.	El ingreso a esta función no requiere autenticación, por lo que pueden acceder el administrador, el jefe del centro de información, el operador y el visitante.		
Realizar consulta de materiales por carrera.	Mostrar una tabla con el listado de los resultados obtenidos de la búsqueda por la carrera seleccionada, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Realizar consulta de materiales por fecha ya sea ingresando sólo año o año y mes.	Mostrar una tabla con el listado de los resultados obtenidos de la búsqueda por fecha ingresada, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Realizar consulta de materiales por asesor/director.	Mostrar una tabla con el listado de los resultados obtenidos de la búsqueda por asesor/director ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Realizar consulta de materiales por autor.	Mostrar una tabla con el listado de los resultados obtenidos de la búsqueda por autor ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Realizar consulta de materiales por título.	Mostrar una tabla con el listado de los resultados obtenidos de la búsqueda por título ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Realizar consulta de materiales por descargas.	Mostrar una tabla con el listado de los resultados obtenidos de la búsqueda por número de descargas ingresado u ordenado el número de descargas de mayor a menor o viceversa, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Realizar consulta de materiales buscando en todo el contenido.	Mostrar una tabla con el listado de los resultados obtenidos del dato de búsqueda ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha y el documento descargable.		Correcto
Confirmación de datos.	Puesto que para realizar una consulta se requiere de ingresar un dato, antes de ejecutar la acción se debe pedir el dato de acuerdo al tipo de filtro necesario para cada tipo de consulta.		Correcto
Descarga de tesis.	De la lista obtenida de resultados obtenida de la consulta debe ser posible la descarga del material bibliográfico (tesis) en formato PDF.		Correcto

La tabla 5.10 para el módulo de reportes, contiene la descripción de los casos de prueba.

**Tabla 5.10 Caso de prueba ciclo no.06, módulo reportes.**

<i>Caso de prueba</i>			
Formato del ciclo	06	Módulo	Reportes
Caso de prueba	Resultado esperado		Resultado
Seleccionar un tipo de reporte.	Para tener acceso a esta función se requiere estar identificado dentro del sistema, por lo que sólo pueden acceder el administrador, el jefe del centro de información y el operador. Se debe encontrar la función de reportes agregada a la de consultas.		Correcto
Generar reporte de materiales por carrera.	Se debe crear un documento PDF con el listado de los resultados obtenidos de la búsqueda por la carrera seleccionada, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Generar reporte de materiales por fecha ya sea ingresando sólo año o año y mes.	Se debe crear un documento PDF con el listado de los resultados obtenidos de la búsqueda por fecha ingresada, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Generar reporte de materiales por asesor/director.	Se debe crear un documento PDF con el listado de los resultados obtenidos de la búsqueda por asesor/director ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Generar reporte de materiales por autor.	Se debe crear un documento PDF con el listado de los resultados obtenidos de la búsqueda por autor ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Generar reporte de materiales por título.	Se debe crear un documento PDF con el listado de los resultados obtenidos de la búsqueda por título ingresado, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Generar reporte de materiales por descargas.	Se debe crear un documento PDF con el listado de los resultados obtenidos de la búsqueda por número de descargas ingresado u ordenado el número de descargas de mayor a menor o viceversa, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Generar reporte de materiales buscando en todo el contenido.	Se debe crear un documento PDF con el listado de todos los materiales existentes, visualizando por material los datos de título, autor, asesor/director, carrera, fecha, teniendo por encabezado el título del reporte y el número de resultados.		Correcto
Confirmación de datos.	Puesto que para realizar un reporte se requiere de ingresar un dato, antes de ejecutar la acción se debe pedir el dato de acuerdo al tipo de filtro necesario para cada tipo de reporte.		Correcto
Descarga de tesis.	De la lista obtenida de resultados obtenida de la consulta debe ser posible la descarga del material bibliográfico (tesis) en formato PDF.		Correcto

En la tabla 5.11 se muestran los casos de prueba para el ciclo número 07 del módulo exportar e importar la base de datos.

**Tabla 5.11 Caso de prueba ciclo no. 07, módulo exportar e importar la base de datos.**

<i>Caso de prueba</i>			
Formato del ciclo	07	Módulo	Exportar e importar la base de datos
<b>Caso de prueba</b>	<b>Resultado esperado</b>		<b>Resultado</b>
Ingresar a la sección de exportar / importar.	Para tener acceso a esta función se requiere estar identificado dentro del sistema como administrador.		Correcto
Importar tabla tesis.	Cuando se importa la tabla de tesis se debe ver reflejada la acción en la base de datos (tabla tesis) y el sistema (administración de materiales y consultas / reportes).		Correcto
Importar tabla usuarios.	Cuando se importa la tabla de tesis se debe ver reflejada la acción en la base de datos (tabla usuarios) y el sistema (administración de usuarios).		Correcto
Exportar tabla tesis.	Cuando se exporta la tabla tesis se debe generar un archivo con extensión .cvs que contenga los datos referentes a esta tabla: título, autor, asesor/director, carrera, ruta del material, fecha y número de descargas.		Correcto
Exportar tabla usuarios.	Cuando se exporta la tabla usuarios se debe generar un archivo con extensión .cvs que contenga los datos referentes a esta tabla: nombre de usuario, contraseña, puesto, nombre, apellido paterno, apellido materno.		Correcto
Confirmación de datos.	Puesto que para exportar o importar se requiere saber de qué tabla se trata, antes de ejecutar la acción se debe pedir el nombre de la tabla, para el caso de importar se debe solicitar el documento que contenga los datos.		Correcto

Por último tenemos la tabla 5.12 con los casos de prueba para la sección de ayuda.

**Tabla 5.12 Caso de prueba ciclo no. 08, módulo sección de ayuda.**

<i>Caso de prueba</i>			
Formato del ciclo	08	Módulo	Sección de ayuda
<b>Caso de prueba</b>	<b>Resultado esperado</b>		<b>Resultado</b>
Solicitar ayuda.	Para tener acceso a esta función no se requiere estar identificado dentro del sistema, tiene acceso el administrador, el jefe del centro de información, el operador y el visitante.		Correcto
Mostrar ayuda.	Al momento de seleccionar la opción de ayuda se mostrará el documento correspondiente al manual de usuario, el cual se desplegará en una pestaña nueva del navegador en formato PDF.		Correcto

Como se observó en las tablas de los casos de prueba de cada ciclo los resultados obtenidos fueron satisfactorios. Finalmente al tener unidos todos los módulos que conforman el SARB se realizaron las pruebas de integración, que se describen en la tabla 5.13.

**Tabla 5.13 Pruebas de integración al SARB.**

Pruebas de integración	
Prueba de integración	Resultado
1. Realizar la instalación del sistema en el equipo del cliente en el servidor WAMP (que implica el manejador de base de datos MySQL, Apache y PHP).	El proceso de instalación fue correcto.
2. Ingresar a la página de bienvenida que no requiere autenticación (incluye las opciones de consultas, ayuda y acceder a funciones específicas del SARB).	Ingresando al sistema desde un navegador con la dirección del SARB, fue posible acceder a la página de bienvenida del sistema.
3. Ingresar como administrador.	Validación de datos correctos (nombre de usuario y contraseña) e inicio de sesión con perfil de administrador.
4. Ingresar como jefe del centro de información.	Validación de datos correctos (nombre de usuario y contraseña) e inicio de sesión con perfil de jefe del centro de información.
5. Ingresar como operador.	Validación de datos correctos (nombre de usuario y contraseña) e inicio de sesión con perfil de operador.
6. Dar de alta a un usuario desde el perfil del administrador y del jefe del centro de información.	Se dio de alta a un usuario desde cada perfil en la base de datos.
7. Verificar el usuario nuevo desde la tabla de administrar usuarios.	Fue posible validar que el usuario ingresado se encontraba ahora en la tabla de administrar usuarios.
8. Realizar una modificación al usuario agregado y posteriormente eliminar el registro.	La modificación y eliminación del registro se vio reflejada en la tabla de administrar usuarios, con lo que se comprobó que la acción se realizó.
9. Dar de alta 10 tesis de la siguiente manera: 3 desde el perfil del operador, 3 desde el perfil del jefe del centro de información y 4 desde el perfil del administrador.	Se dieron de alta satisfactoriamente todas las tesis.
10. Realizar una modificación a un material existente y posteriormente eliminar el registro desde cada perfil permitido.	La modificación y eliminación de cada registro seleccionado se vio reflejado en el sistema (administración de materiales), con lo que se comprobó que la acción se realizó.
11. Consultar las tesis desde el perfil de cada tipo de usuario y por cada filtro: carrera, fechas, asesor/director, autor, título, descargas y en todo. Descargar un documento de cada tipo de consulta.	Las consultas y descargar a documentos han sido exitosas.
12. Generar reportes de cada tipo establecido: carrera, fechas, asesor/director, autor, título, descargas y de todo.	La creación de reportes fue satisfactoria.
13. Exportar la tabla de usuarios y tesis.	Se generó por cada tabla el documento con extensión .csv con los datos contenidos en la base de datos. La exportación fue exitosa.
14. Importar el documento que contienen la información de usuarios y el documento que contiene la información de tesis.	Se importaron los datos de cada tabla (usuarios, tesis) dentro de la base de datos. La importación fue exitosa.
15. Ingresar a la parte de ayuda.	Se abrió una pestaña nueva dentro del navegador con el manual de usuario en formato PDF.

## 5.2 Pruebas generales del software

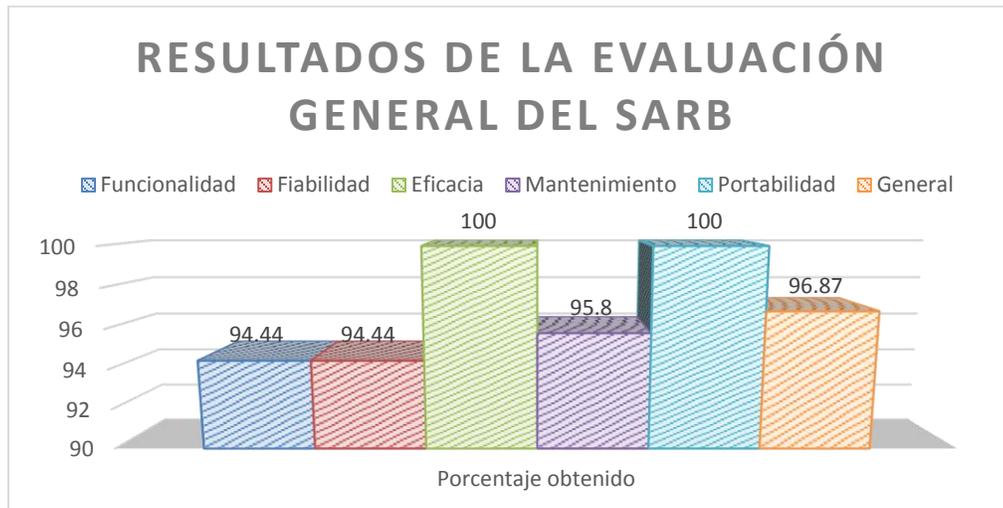
En base a la norma ISO/IEC 9126, con el cual es posible evaluar la calidad del software (Borbón, 2013), se llevaron a cabo las pruebas generales del SARB, a través de un cuestionario y una matriz de evaluación de software, la cual propone Duran, Ruiz, & Rivodó (Recuperado en 2016) que se generó después de analizar aspectos necesarios para realizar dicha evaluación de forma objetiva considerando la norma (Duran, Ruiz, & Rivodó, 2016).

Se utilizó el cuestionario empleado por Gutiérrez (2013) en su trabajo de tesis, haciendo las modificaciones pertinentes para ajustarlo al caso de estudio. Las métricas ocupadas para responder cada pregunta fueron las mismas que en el trabajo mencionado: ‘Si’ sí se obtuvo el resultado esperado, ‘medio’ si se obtuvo pero no del todo y ‘No’ si no se obtuvo o fue muy poco el resultado final, asignando pesos de la siguiente manera: Si=3, medio=2 y No =1; así al sumar el puntaje obtenido se multiplica por 100 y se divide entre el puntaje total para sacar el porcentaje de cumplimiento de cada aspecto. Los resultados de las pruebas se encuentran en la tabla 5.14.

**Tabla 5.14 Resultados de las pruebas al SARB.**

Característica	Sub-característica	Pregunta	Si	Medio	No	Total	%
<b>Funcionalidad</b>	Adecuación	¿Cuenta con las funciones necesarias para cada tarea?	6				
	Exactitud	¿Realiza lo que fue determinado de la manera que se espera y correctamente?	3	2			
	Seguridad	¿El acceso a las funciones restringidas del SARB está limitado a usuarios registrados autorizados?	6			17	94.44
<b>Fiabilidad</b>	Madurez	¿Presenta fallos con frecuencia?	6				
	Recuperabilidad	¿El sistema es capaz de recuperar los datos de un fallo?	6				
	Tolerancia a fallas	¿El sistema puede detectar datos erróneos al momento de introducirlos?	3	2		17	94.44
<b>Eficiencia</b>	Comportamiento en tiempos	¿El tiempo de respuesta para un proceso es adecuado?	6				
	Utilización de los recursos	¿El sistema funciona con una velocidad aceptable aun cuando se tienen varios programas en ejecución?	6			12	100
<b>Mantenibilidad</b>	Capacidad de ser analítico	¿Es fácil diagnosticar o identificar partes a modificar?	6				
	Estabilidad	¿No existen riesgos o efectos inesperados cuando se realizan cambios?	3	2			
	Facilidad de prueba	¿Las modificaciones son fáciles de validar?	6				
	Confiabilidad	¿Es fácil de modificar y adaptar?	6			23	95.8
<b>Portabilidad</b>	Adaptabilidad	¿Es posible utilizar la aplicación web con cualquier navegador?	6				
	Instalabilidad	¿Es fácil de instalar?	6				
	Coexistencia	¿Comparte con facilidad recursos con otro software o dispositivo?	6				
	Remplazabilidad	¿Es fácil de usarlo en lugar de otro software?	6			24	100
<b>Total</b>						93	96.87

Para interpretar de una mejor manera los datos obtenidos en la tabla 5.5 se elaboró una gráfica, la cual se presenta en la figura 5.2.



**Figura 5.2 Resultados de la evaluación general del SARB.**

En la gráfica es posible apreciar que el resultado global fue de 96.87%, con lo que se determina que se alcanzó un nivel satisfactorio. Lo anterior señala que se cumplió con los requerimientos por parte de los clientes, los resultados son en gran medida por el empleo de la metodología propuesta para el desarrollo del SARB. Se demuestra de esta forma que se cumple con el objetivo de generar un sistema de calidad.

Los resultados presentados en este capítulo muestran la evaluación de la metodología propuesta, que concluye que se puede definir como ágil, así como también los obtenidos del caso de estudio que se utilizó para comprobar la funcionalidad de dicha metodología. Se lograron para cada caso resultados que han sido satisfactorios (casos de prueba para cada ciclo, pruebas de integración y pruebas generales del SARB). Queda demostrado con los resultados que es posible utilizar la metodología definida en el capítulo 3 para generar de manera adecuada producto software en tiempo y forma, cumpliendo con la calidad requerida.

## Capítulo 6 Conclusiones y trabajos futuros

En este capítulo se describen las conclusiones generales después de haber desarrollado la metodología y aplicado la misma al caso de estudio en el sistema planteado en este trabajo de tesis. Finalmente se da a conocer una propuesta de trabajos futuros con el objetivo de mejorar lo aquí propuesto con otros elementos que beneficien el trabajo desarrollado.

### 6.1 Conclusiones

Retomando en este punto la pregunta de investigación “¿Es posible crear una metodología de Ingeniería de Software que permita desarrollar sistemas de información de calidad y seguros, con equipos de trabajo en lugares diferentes?”, que fue planteada al principio de este trabajo (Capítulo 1 sección 1.6), en base a los resultados obtenidos del caso de estudio analizado en esta investigación es posible generar una respuesta.

Con la experiencia obtenida al elaborar este trabajo, se detectó que uno de los principales problemas presentados al usar la metodología planteada para desarrollar el SARB, fue la comunicación en los equipos de trabajo distribuidos. Este factor se vio afectado por la distancia en la que se encontraban los participantes, a pesar de tratarse de tan sólo dos ubicaciones dentro de un mismo estado y manejando el mismo horario. Como se describe en el capítulo 4, los factores que actúan negativamente sobre la comunicación son principalmente la desorganización, el desinterés, conflictos interpersonales, y la desinformación. Lo anterior genera para el guía el empleo de un esfuerzo mayor para poder integrar a los miembros del equipo, lo que se describe más adelante.

La idea de la metodología para trabajar con los distintos equipos es importante para la persona que tiene la función de guía, ya que le permite determinar la ubicación de los integrantes. Cabe destacar que para el momento del desarrollo del proyecto se maneja un único equipo (sin importar la ubicación de los participantes), de esta forma se evita divisiones y la falta de cohesión entre los desarrolladores. A pesar de esta característica que plantea la metodología no se elude del todo la fragmentación de los equipos. Para abordar esta problemática fue de gran

utilidad el plan de comunicación, mostrado en la sección 7 (formato de retroalimentación) que permitió el contar con un registro de los conflictos presentados en cada ciclo, así como el impacto de cada uno, el nivel de afectación y la propuesta de solución. Con empleo de esta estrategia fue posible mejorar las acciones a realizar para eliminar o reducir en la medida de lo posible las problemáticas presentadas y su grado de impacto.

Los conflictos interpersonales, que se presentaron de manera repetida en cada ciclo, se atacaron tomando distintas medidas, ya que creaban el desinterés de los integrantes al provocar que no todos los miembros se sintieran pertenecientes al equipo, sumándole la falta de convivencia y la interacción cara a cara. Lo anterior provocaba un ambiente hostil, percibiéndose que las distintas personalidades de los participantes no podían adaptarse unas a otras. Para lo cual en ciclos intermedios se determinó en reuniones rápidas limar asperezas que aquejaban al equipo, hablar claro de lo que molestaba a cada uno, y exponer aquellas situaciones que habían sido expresadas abiertamente. El problema no se eliminó por completo, pero las tensiones se aliviaron, fue posible enfocarse en los objetivos del proyecto y al ver los avances que se habían logrado a la mitad del proyecto, se originó una razón que motivó a que cada persona a realizar su trabajo, participar en las actividades y reuniones. Se acordó que en caso de que no se efectuara lo acordado, la persona que incumpliera tenía que apoyar a sus compañeros en otras tareas, lo que fomentó a que todos se dieran cuenta de la importancia de la función que tenía asignada cada quien, además de que se requirió una mayor interacción entre los participantes. Se obtuvo entre los miembros, en cierta medida, un sentido de pertenencia y por ende el surgimiento del interés, dando como resultado que disminuyeran los conflictos interpersonales y la desorganización.

Otro problema que se presentó en los equipos fue la desinformación. Esta situación se contempló dentro del plan de comunicación al determinar los medios electrónicos para solucionar esta dificultad, contando con un mapa general, haciendo uso de las mismas herramientas para la codificación, con la intención de manejar un mismo léxico. De igual manera se determinó una estrategia de comunicación, con la finalidad de tener una documentación precisa y oportuna. Pero sin duda alguna fue el abusar del uso de varios

medios que proporcionan las TIC (a través de los cuales se puede compartir los documentos, o contactar a los desarrolladores del equipo) lo que contribuyó más a la desinformación (se produjo un efecto contrario al que se creía obtener). Por tal motivo, no es algo recomendable, pues satura a los miembros, los confunde, provoca que olviden que documentos que ya revisaron o la omisión de algunos que nunca vieron, ocasionando que cuenten con información parcial de lo que se está haciendo y lo que se realizará. La consecuencia de esta situación genera retrasos en los tiempos de entrega, duplicidad de trabajo, incompatibilidad al momento de integrar, como si se encontraran en otra sintonía muy distinta a los objetivos planteados. Para dar solución a la desinformación se determinó en el ciclo 4 (ver sección 7 del plan de comunicación, capítulo 4) la firma electrónica, pues antes de utilizarla se empleó la firma en papel, pero se tenía que imprimir el formato, firmarlo, escanearlo o sacarle foto con algún dispositivo móvil y subirlo al repositorio, lo que resultaba engorroso, poco profesional, originando tedio y que no todos revisaran la información. Debido a esta circunstancia se determinó un solo medio para el repositorio con el cual compartir los documentos entre todo el equipo, teniendo cada integrante conocimiento de que sólo allí se podrían revisar los documentos generados. El repositorio fue seccionado en carpetas bien definidas, los documentos compartidos contaban con información precisa y clara.

En lo referente a los tiempos programados en el plan de actividades existieron retrasos lo que fue inevitable, pues debido a todos los problemas mencionados, apenas dio tiempo de concluir en el límite del plazo de cada ciclo. A partir de la iteración número 5 el trabajo de los participantes dentro del equipo mejoró, con lo que se pudo presentar e instalar en el equipo personal del cliente el SARB, faltando sólo la instalación en un servidor y aun así, el sistema entregado cumplió con los requerimientos solicitados por el cliente en tiempo y forma, con la calidad requerida. Esto puede ser constatado con resultados satisfactorios en las pruebas vistas en el capítulo 5, superando de manera grata los problemas propios que surgen de trabajar con equipos de desarrollo a distancia.

Luego entonces se puede concluir que es posible trabajar con integrantes ubicados en puntos geográficos separados para generar producto software de calidad, y que el tiempo es un

elemento clave para que el equipo vaya madurando, integrándose mejor y reduciendo los problemas que se presentan. La interacción hace posible la cohesión, se puede pensar que conforme se van realizando más proyectos o ciclos, y la retroalimentación en base a la experiencia se hace más rica los resultados serán mejores.

Como se dijo en el capítulo 2, actualmente existe una tendencia que va en aumento en cuanto al desarrollo a distancia de software fusionado con las metodologías ágiles. Sin embargo no se trata de crear solo una metodología y seguirla como una receta, esto pudiera crear la ilusión de que es algo muy sencillo de llevar a la práctica. La realidad es que la comunicación y la organización de las personas que integran los equipos de trabajo, son factores que se ven afectados por la distancia, lo cual repercute directamente a los plazos establecidos. Tal como sucedió en el caso de estudio planteado para probar la metodología propuesta, en el que se atrasaron las fechas a cumplir debido a problemas en la comunicación y organización. Por lo que esta metodología no sólo es una serie de pasos para generar un producto software, sino más bien propone un plan para superar los obstáculos producidos por la distancia y evitar en lo posible atrasos en los objetivos y metas establecidos.

Con lo que respecta al desarrollo de la aplicación web, fue muy notorio la gran facilidad que ofrecieron las herramientas de desarrollo PHP, MySQL y HTML para la creación de SARB. La razón de que el sistema estuviera basado en dichas herramientas es por el recurso disponible en internet de licencia GNU, lo que permite el reutilizar código, reduciendo así los tiempos de desarrollo, pero principalmente, como se ha dicho en repetidas ocasiones a lo largo de este trabajo fue el utilizar un mismo lenguaje entre los participantes del equipo.

Finalmente y a manera de conclusión general, la metodología propuesta en el presente trabajo de investigación pudo combinar métodos, modelos y principios del desarrollo ágil con técnicas de desarrollo global de software y un plan de comunicación. De tal manera que fue posible construir una metodología que hace frente a situaciones que sufre el desarrollo global de software, que permite desarrollar sistemas de información y cualquier producto software con calidad, donde los equipos de trabajo pueden estar distribuidos geográficamente.

## 6.2 Trabajos futuros

Los siguientes puntos son algunos de los posibles trabajos futuros proyectados con la finalidad de darle continuidad al trabajo realizado:

- Desarrollar una herramienta de comunicación creada específicamente para la colaboración a través de internet para el desarrollo de software global-ágil en función de la metodología propuesta.
- Probar la metodología para equipos de desarrollo de software ubicados en mayores distancias, empezando por equipos de trabajo ubicados en distintos estados, y luego ubicados países distintos que hablen un mismo idioma. Lo cual mejorara la metodología de desarrollo planteada enriqueciéndola con mayor experiencia al ser aplicada en otros contextos.
- Crear un indicador para medir la calidad de la comunicación entre los desarrolladores de software, que permita detectar los puntos fuertes y sobre todo los vulnerables, además de que considere la colaboración a distancia, de esta manera tomar las mejores acciones posibles para atacar esos problemas.
- Crear una comunidad virtual exclusiva para el desarrollo de aplicaciones en entornos distribuidos donde se comparta experiencias, consejos, enlaces a artículos científicos, se resuelvan dudas y se oriente a los desarrolladores, proporcionando información confiable y actualizada.
- Mejorar el sistema web desarrollado siguiendo la metodología establecida, agregando otras funciones para el manejo no sólo de tesis, sino también de artículos, proyectos, ensayos entre otros, desarrollados por la comunidad académica de instituciones educativas de nivel superior, para difundir el trabajo realizado en estas instituciones.

### Referencias bibliográficas

- Ågerfalk, P., Fitzgerald, B., Holmstrom, H., & Ó Conchúir, E. (ICSP 2008). Benefits of Global Software Development: The Known and Unknown. *D Pfahl, and D.M.Raffo (Eds.): Making Globally Distributed Software a Success Story* (págs. 1-9). Berlin Heidelberg: Springer-Verlag.
- Aguilar, J. M. (Octubre de 2006). Guía para el desarrollo de aplicaciones Web con Apache, PHP y MySQL. *Trabajo escrito en la modalidad de seminario o diplomado de actualización y capacitación profesional*. Edo.de México, México: Universidad Nacional Autónoma de México. Facultad de Estudios Superiores Aragón.
- Alonso, F., Serrano, N., & Calzada, S. (2004). *Informática III*. San Sebastián, España: tecnun.
- Araújo, A. (2007). Test driven development :fortalezas y debilidades. *Reportes Técnicos* , 07-13.
- Avritzer, A., Beecham, S., Kroll, J., Menasché, D., Noll, J., & Paasivaara, M. (2014). Survivability Models for Global Software Engineering. *IEEE 9th International Conference on Global Software Engineering* , 100 - 109 .
- Baelo, R., & Cantón, I. (2009). Las tecnologías de la información y la comunicación en la educación superior. Estudio descriptivo y de revisión. *Revista Iberoamericana de Educación*.
- Barroso, R., Oliveros, Y., García, L., Y., Á., & Coello, J. (2008). Aplicación de la metodología ágil Crystal Clear a un caso de estudio. *Convención Científica de Ingeniería y Arquitectura*.
- Blom, M. (2010). Is Scrum and XP suitable for CSE Development? *Procedia Computer Science* 1, 1511-1517.

- Borbón, N. (12 de Marzo de 2013). *Evaluacion de software*. Obtenido de NORMA DE EVALUACIÓN ISO/IEC 9126: <http://actividadreconocimiento-301569-8.blogspot.mx/2013/03/norma-de-evaluacion-isoiec-9126.html>
- Christensson, P. (. (17 de Febrero de 2014). *TechTerms*. Obtenido de Web Application Definition: <http://techterms.com>
- Comas, A. (2004). Java o PHP. *Revista Digital Universitaria*, Vol. 5, No. 7 [en línea]. Disponible en Internet: <<http://www.revista.unam.mx/vol.7/num12/art104/int104.htm>>.
- Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., & White, B. (2002). WEB ENGINEERING. *Journal of Web Engineering*, Vol. 1, No.1, 03-17.
- Dropbox. (s.f.). *Dropbox*. Obtenido de Visita guiada de Dropbox (Recuperado el 2016): <https://www.dropbox.com/>
- Duran, G., Ruiz, I., & Rivodó, Y. (29 de Julio de 2016). *microevolution*. Obtenido de ¿Qué es una Matriz de Evaluación de Software?: <http://microevolution.wikispaces.com/2.-+%C2%BFQu%C3%A9+es+una+Matriz+de+Evaluaci%C3%B3n+de+Software%3F>
- Eguiluz, J. (s.f.). *LIBROSWEB*. Recuperado el 23 de 06 de 2016, de Introducción a JavaScript: <http://librosweb.es/libro/javascript/>
- Escribano, A. (2009). *Ingeniería web dirigida por modelos*. España: Escuela Superior de Informática. Universidad de Castilla-La Mancha.
- Facebook. (s.f.). *Servicio de Ayuda*. Recuperado el 2016. Obtenido de Conceptos básicos sobre los grupos.
- Fojtik, R. (2011). Extreme Programming in development of specific software. *Procedia Computer Science* 3, 1464–1468.

- Fu, Y., Zhang, X., & Hu, X. (2011). Realization of Electronic Paper-Based mobile digital library. *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on* , 6719 - 6722 .
- Garzás, J. (25 de Septiembre de 2012). *Las metodologías Crystal. Otras metodologías ágiles que, quizás, te puedan encajar más que Scrum*. Obtenido de javiergarzas.com: <http://www.javiergarzas.com/2012/09/metodologias-crystal.html>
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona, España: MARCOMBO.
- Ginige, A., & Murugesan, S. (2001). Web Engineering: An Introduction. *IEEE MultiMedia*, 14-18.
- Google. (s.f.). *Google Drive*. Obtenido de Página oficial (Recuperado el 2016): [https://www.google.com/intl/es\\_mx/drive/](https://www.google.com/intl/es_mx/drive/)
- Gutiérrez, L. (Noviembre de 2013). Implementación de la lógica de negocio de un sistema informático para la administración de documentos del sistema de calidad de una empresa fabricante de papel filtro, usando metodologías ágiles. Apizaco, Tlaxcala, México.
- Herrera, E., & Valencia, L. E. (2007). Del manifiesto ágil sus valores y principios. *Scientia Et Technica, vol. XIII, núm. 34*, 381-386.
- Holmstrom, H., Fitzgerald, B., Agerfalk, P., & Conchuir, E. (2006). Agile Practices Reduce Distance in Global Software Development. *Information Systems Management, Vol 23, No. 3*, 7-18.
- Honmy, R. (2008). La web. Heramienta detrabajo colaborativo. "Experiencia en la Universidad de Carabobo". *Pixel-Bit. Revista de Medios y Educación, núm. 31*, 131-139.

- Jalali, S., & Wohlin, C. (2010). Agile Practices in Global Software Engineering – A Systematic Map. *2010 5th IEEE International Conference on Global Software Engineering* , 45-54.
- Jiménez, E., & Orantes, S. (2012). METODOLOGÍAS HÍBRIDAS PARA DESARROLLO DE SOFTWARE: UNA OPCIÓN FACTIBLE PARA MÉXICO. *Revista Digital Universitaria, Volumen 13 No. 1*.
- Karimpoor, H., Isazadeh, A., & Sadighi, M. (2008). Object-Oriented Hypermedia Design and J2EE Technology for Web-based Applications. *Issues in Informing Science and Information Technol, Vol. 5*, 729-740.
- Kumar, G., & Kumar, P. (2012). Impact of Agile Methodology on Software Development Process. *International Journal of Computer Technology and Electronics Engineering*, 46-50.
- Kumar, G., & Kumar, P. (2014). Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies. *Fourth International Conference on Advanced Computing & Communication Technologies*, 189 - 196.
- Lamarca, M. J. (08 de Diciembre de 2013). *HiperTexto*. Obtenido de Hipertexto: El nuevo concepto de documento en la cultura de la imagen: <http://www.hipertexto.info/index.htm>
- Luján, S. (2001). *Programación en Internet: Clientes WEB*. Alicante, España: Editorial Club Universitario.
- Mendes, K., Estevez, E., & Fillotrani, P. (2010). Evaluación de Metodologías Ágiles para Desarrollo de Software. *XII Workshop de Investigadores en Ciencias de la Computación* , 455-459.
- Merino, I. (22 de Septiembre de 2015). *GENBETA*. Obtenido de Las 14 novedades más importantes de Office 2016: <http://www.genbeta.com/ofimatica/las-14-novedades-mas-importantes-de-office-2016>

- Microsoft. (s.f.). *Página oficial de Microsoft. Recuperado el 2016*. Obtenido de Productos Office.
- Microsoft. (s.f.). *Skype*. Obtenido de Página oficial de Skype (Recuperado el 2016): <https://www.skype.com/es/>
- Minoli, M. (2010). Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de ingeniería de requisitos. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 66-75.
- Navarro, A., Fernández, J. D., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospect. Vol. 11, No. 2*, 30-39.
- Nieves, C., Ucán, J., & Menéndez, V. (2014). UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio. *Revista Latinoamericana de Ingeniería* , 137-143.
- ORACLE. (s.f.). *MySQL*. Recuperado el 23 de Junio de 2016, de Reference Manual. 1.3.1 What is MySQL: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- Ordoñez, I. E. (2003). Las tesis digitales en la Biblioteca Central de la UNAM. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal Sistema de Información Científica*.
- Ortiz, V. (2016). *Test blendspace*. Obtenido de Docs en Google Drive: <https://www.tes.com/lessons/rvgP3zHqij5-uQ/doc-en-google-drive>
- Paasivaara, M., & Lassenius, C. (2006). Could Global Software Development Benefit from Agile Methods? *IEEE International Conference on Global Software Engineering*.
- Palacio, J. (2007). Flexibilidad con Scrum. *Navegapolis*.
- Pilemalm, S., Lindell, P., Hallberg, N., & Eriksson, H. (2007). Integrating the rational unified process and participatory design for development of socio-technical systems: A user participative approach. *Design Studies*, 263–288.

- Piorun, D., & Giusti, J. L. (2001). *Liderando proyectos*. Buenos Aires, Argentina: Macchi.
- Pressman, R. S. (2010). *Ingeniería del Software. Un enfoque práctico*. México: McGraw-Hill.
- Ramírez, L. C., & Flórez, A. S. (2014). Buenas prácticas , una solución de para un mejor desarrollo de software. *Revista Mundo FESC*, 37-45.
- Reyes, M. (Octubre de 2007). Metodologías para el desarrollo de software educativo. *Monografía*. Pachuca de Soto, Hidalgo, México: Universidad Autónoma de Hidalgo.
- Schwaber, K. (1995). *Scrum development process, in: Proceedings of the OOPSLA'95 Workshop on Business Object Design and Implementation*. Austin, Texas, United States.
- Silva, D., & Reygadas, L. (2013). Tecnología y trabajo colaborativo en la sociedad del conocimiento. *Alteridades, vol. 23, núm. 45*, 107-122.
- Sommerville, I. (2011). *Ingeniería de Software*. México: PEARSON EDUCACIÓN.
- The\_PHP\_Group. (s.f.). *php*. Recuperado el 23 de 06 de 2016, de Manual de PHP. Conceptos básicos: <http://php.net/manual/es/getting-started.php>
- Tinoco, O., Rosales, P., & Salas, J. (2010). Criterios de selección de metodologías de desarrollo de software. *Revista de la Facultad de Ingeniería Industrial*, 70-74.
- Tumino, M., Bournissen, J., Bracalenti, C., Schlemper, E., & Kucharski, S. (2013). Gestión y Desarrollo de Proyectos de Software: Metodología Ágil Basada en telecomunicaciones MAte. *Revista Latinoamericana de ingeniería de Software*, 253-258.
- Vizcaíno, A., García, F., & Piattini, M. (2014). *Desarrollo Global de Software*. España: Ra-Ma.
- Vizcaíno, A., García, F., & Piattini, M. (2014). Visión General del desarrollo Global de Software. *International Journal of Information Systems and Software Engineering for Big Companies (IJISEBC)*, Vol. 1, Num. 1, 8-22.

**Anexos**

**Anexo 1. Manual de usuario**



---

# MANUAL DE USUARIO DEL SARB

---

(Sistema de Administración de Recurso Bibliográfico)



INSTITUTO DEL ALTIPLANO DE TLAXCALA

## Tabla de contenido

Descripción.....	2
Entorno del sistema .....	2
1. Página de inicio .....	2
2. Logearse en el SARB .....	3
3. Administrar .....	3
3.1 Agregar material .....	3
3.2 Administrar materiales .....	5
3.2.1 Eliminar tesis .....	5
3.2.2 Editar/modificar tesis.....	6
3.3 Agregar usuario .....	6
3.4 Administrar operador.....	7
3.4.1 Eliminar un usuario .....	8
3.4.2 Editar/modificar un usuario .....	8
4. Realizar consultas/reportes .....	9
4.1 Por carrera.....	9
4.2 Por fecha .....	10
4.3 Por asesor/director .....	12
4.4 Por Autor .....	13
4.5 Por título .....	14
4.6 Consulta y reporte a partir de todo el material bibliográfico.....	15
5. Exportar e importar base de datos .....	16
5.1 Exportar .....	16
5.2 Importar .....	17
6. Salir del sistema.....	18
7. Ayuda .....	18

### Descripción

Este sistema tiene el objetivo de almacenar, preservar y difundir el producto académico realizado por alumnos del Instituto Tecnológico de Apizaco (ITA), específicamente tesis en formato digital. El sistema de Administración de Recursos Bibliográficos (SARB) pretende, a través del uso de la tecnología, poner el acceso a la literatura científica (propia del ITA) de manera abierta en la red de internet, lo que permite dar a conocer el trabajo realizado por la comunidad institucional, así como también una gestión eficiente de dichos documentos que conforman el recurso bibliográfico.

### Entorno del sistema

Barra de navegación completa del sistema



#### 1. Página de inicio

Esta es la página de bienvenida, la cual aparece cada vez que se ingresa al sistema:



## 2. Logearse en el SARB

En la barra de navegación seleccionar SARB:



Una vez dentro del apartado SARB ingrese su usuario y contraseña en los espacios indicados, posteriormente presiona el botón entrar:



## 3. Administrar

**Importante:** Para poder tener acceso a esta opción debe estar logeado dentro del sistema (Jefe del centro de información, Administrador u Operador)

### 3.1 Agregar material

De la barra de navegación selecciona la opción "Administrar", y del menú desplegable selecciona "Agregar material":



A continuación se mostrará el formulario correspondiente a la opción “Agregar Material” en donde podrá ingresar los datos correspondientes al nuevo material bibliográfico (tesis), una vez ingresados dichos datos da clic en el botón guardar para agregar la nueva tesis o en cancelar para evitar guardar los datos:

### Agregar tesis

**Título \*** Tesis ejemplo 1. Ingresar el Título de la tesis  
**Autor \*** Autor Ejemplo 2. Ingresar el Autor de la tesis  
**Asesor/Director** Asesor Ejemplo 3. Ingresar el Asesor/Director de la tesis  
**Carrera** Maestría en Sistemas Computacionales 3. Elegir una carrera  
**Fecha Edición** 01-04-2000 3. Seleccionar una fecha (a. Año. b. Mes. c. Día)  
**Archivo \*** Examinar... Tesis.pdf 3. Seleccionar el archivo pdf de la tesis  
1. Presionar guardar o cancelar

Al presionar el botón de guardar aparecerá en pantalla el detalle de la tesis guardada, en esta parte puede modificar la nueva tesis, eliminarla u agregar otra:

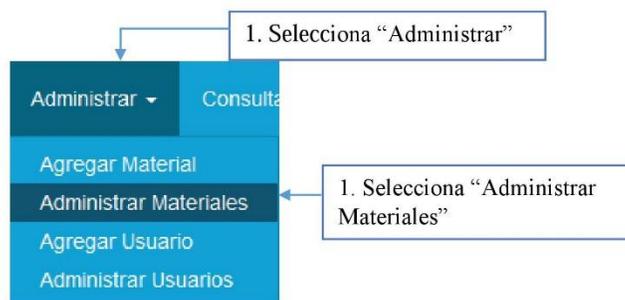
### Detalle material bibliográfico

Botón para editar/modificar la tesis agregada    Botón para agregar otra tesis  
Botón para eliminar la tesis agregada

Atributo	Valor
Título	Tesis ejemplo
Autor(es)	Autor Ejemplo
Tutor	Asesor Eje
Fecha	01-04-2000
Carrera	MSC
PDF	

### 3.2 Administrar materiales

De la barra de navegación selecciona la opción “Administrar”, y del menú desplegable selecciona “Administrar Materiales”:



En esta sección se muestra una tabla con todas las tesis existentes, en la última columna hay dos botones los cuales permiten editar/modificar o eliminar el material bibliográfico, en cadena a buscar se puede ingresar el criterio de la tesis de la cual se quiere administrar.

#### Tesis

1. Ingresar el criterio de la tesis a buscar

2. Presionar botón: actualizar/editar o eliminar

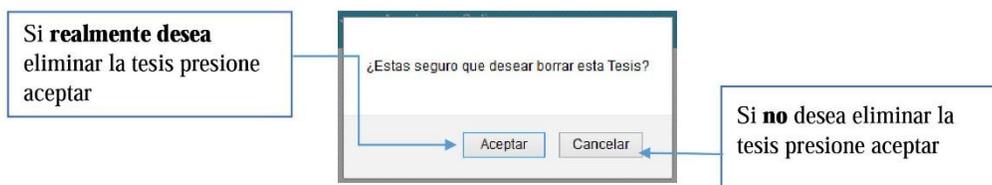
Buscar en el contenido de una tabla

Cadena a buscar:

Título	Autor	Director	Carrera	PDF	Fecha	
Metodología para el desarrollo de objetos de aprendizaje implementada a dispositivos móviles	Maria Guadalupe Perez Lopez	M.C. Jose	MSC		2015-12-28	<input type="button" value="actualizar/editar"/> <input type="button" value="eliminar"/>
Metodología para el desarrollo de objetos de aprendizaje implementada a dispositivos móviles	Maria Guadalupe Perez Lopez	M.C. Jose	MSC		2015-12-29	<input type="button" value="actualizar/editar"/> <input type="button" value="eliminar"/>
Metodología para el desarrollo de objetos de aprendizaje implementada a dispositivos móviles	Ing. Lucila	Jose	singer		2016-01-02	<input type="button" value="actualizar/editar"/>

#### 3.2.1 Eliminar tesis

Si presiona el botón de eliminar solo tendrá que confirmar que realmente desea borrar la tesis:



### 3.2.2 Editar/modificar tesis

Si presiona el botón de editar/modificar aparecerán por default los datos de la tesis seleccionada con opción para modificar aquellos que desee, una vez terminados los cambios puede dar clic en guardar, si no desea realizar ningún cambio presione cancelar:



### 3.3 Agregar usuario

Para agregar un nuevo usuario en la barra de navegación selecciona “Administrar” y del menú desplegable selecciona “Agregar Usuario”:



Inserta los datos que solicita el formulario, al finalizar presiona guardar si desea crear el nuevo usuario o presiona cancelar si no desea crear otro usuario:

### Agregar nuevo usuario

Usuario \*

1. Ingresar un nombre de usuario

Contraseña

2. Ingresar una contraseña

Puesto

3. Seleccionar puesto

Nombre

4. Ingresar nombre del usuario

Apellido Paterno

5. Ingresar apellido paterno

Apellido Materno

5. Ingresar apellido materno

6. Si desea crear el usuario presione guardar

7. Si no desea crear el usuario presione cancelar

#### 3.4 Administrar operador

Cuando requiera editar/modificar o eliminar algún usuario seleccione de la barra de menú “Administrar” y de la barra desplegable elija “Administrar Usuarios”:



Le mostrará una tabla con los usuarios existentes y en la última columna verá dos botones con los cuales podrá editar/modificar o eliminar un usuario:

Administración de operadores

1. Botón para editar/modificar

1. Botón para eliminar

+ Agregar

Usuario	Nombre	Apellido Paterno	Apellido Materno	Puesto	
Astrid	Astrid	Torres	Fernandez	Administrador	 
Daniel	Daniel	Zamora	Velazquez	Administrador	 
Elena	Elena	Rodriguez	Juárez	Jefe de Cl	 
Lorena	Lorena	Zamora	Velazquez	Administrador	 
Mónica	Mónica	Herrera	Manzano	Operador	 

### 3.4.1 Eliminar un usuario

Para eliminar un usuario (seguir antes los pasos para Administrar Operador) presiona el botón eliminar:

1. Presionar el botón de eliminar del usuario que desea quitar

Usuario	Nombre	Apellido Paterno	Apellido Materno	Puesto	
Astrid	Astrid	Torres	Fernandez	Administrador	 
Daniel	Daniel	Zamora	Velazquez	Administrador	 

Finalmente debe confirmar su decisión de borrar al usuario:

¿Estas seguro que desea borrar este operador?

2. Presionar aceptar **si** desea eliminar al usuario

3. Presionar cancelar **si no** desea eliminar al usuario

Aceptar Cancelar

### 3.4.2 Editar/modificar un usuario

Para poder modificar/editar (seguir antes los pasos para Administrar Operador) los datos de un usuario existente debe seleccionar el botón de modificar/editar de la tabla de usuarios:

3. Presionar el botón de modificar/editar

Usuario	Nombre	Apellido Paterno	Apellido Materno	Puesto	
Astrid	Astrid	Torres	Fernandez	Administrador	 
Daniel	Daniel	Zamora	Velazquez	Administrador	 

Podrá visualizar los datos correspondientes al usuario seleccionado y los podrá modificar, una vez finalizada la edición/modificación puede presionar el botón guardar para almacenar los cambios o en cancelar para evitar hacer cualquier modificación en los datos del usuario:

El formulario muestra los datos de un usuario con los siguientes campos:

- Usuario \*: Astrid
- Contraseña: Clave
- Puesto: Administrador
- Nombre: Astrid
- Apellido Paterno: Torres
- Apellido Materno: Fernandez

En la parte inferior del formulario hay dos botones: "Guardar" (en azul) y "Cancelar" (en gris). Dos cuadros de texto con flechas azules indican las acciones:

- Un cuadro a la izquierda dice: "4. Presionar el botón 'Guardar' para guardar cambios". Una flecha apunta desde este cuadro al botón "Guardar".
- Un cuadro a la derecha dice: "5. Presionar el botón 'Cancelar' para no guardar cambios". Una flecha apunta desde este cuadro al botón "Cancelar".

#### 4. Realizar consultas/reportes

En esta sección podrá realizar reportes y consultas de los materiales bibliográficos (tesis).

##### 4.1 Por carrera

En la barra de menú selecciona "Consultas/Reportes" y "Carrera":



Elije una opción en "Elegir carrera" y si solo desea una consulta presiona "Consulta" y si desea un reporte presiona "Reporte":

Visualizar por carrera:

Elegir carrera	Consulta	Reporte		Autor	Asesor/Director	Carrera	PDF	Fecha	
Elegir carrera Maestría en Sistemas Computacionales Maestría en Ingeniería Administrativa									
Metodología para el desarrollo de objetos de aprendizaje implementada a dispositivos móviles				Maria Guadalupe Perez Lopez	M.C. Jose	MSC		2015-12-28	 
Diseño de un sistema de almacenamiento digital para la transmisión visualización y descarga de imágenes radiológicas				Ricardo R.A.	M.C. Jose	MSC		2014-12-01	 

Si lo que presionó fue “Reporte” le aparecera un recuadro para guardar o abrir el reporte en formato pdf, presiona “Aceptar” para visualizar el contenido:



#### 4.2 Por fecha

En la barra de menú selecciona “Consultas/Reportes” y “Fecha”:

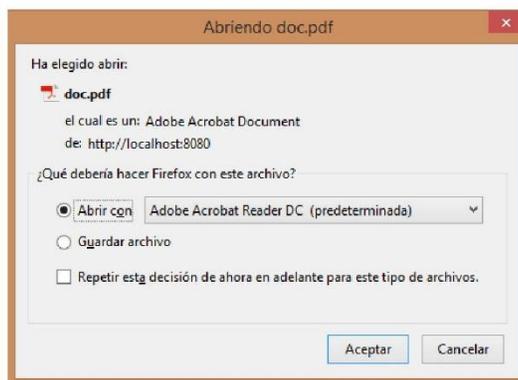


Puede realizar la búsqueda o el reporte ingresando solo el año en “Ingresar año”, y presiona “Consultar” si solo desea visualizar los resultados, si desea reporte presione el botón de “Reporte”:

### Visualizar por fecha de publicación:

Ingresar año:

Si presionó “Reporte” aparecerá un recuadro para visualizar o guardar el archivo pdf solo presione “Aceptar”:



Puede realizar la búsqueda o el reporte ingresando el año y mes en “O ingresar mes y año”, y presiona “Consultar” si solo desea visualizar los resultados, si desea reporte presione el botón de “Reporte”:

O ingresar mes y año:>

Si presionó “Reporte” aparecerá un recuadro para visualizar o guardar el archivo pdf solo presione “Aceptar”:



#### 4.3 Por asesor/director

En la barra de menú selecciona “Consultas/Reportes” y “Asesor/Director”:



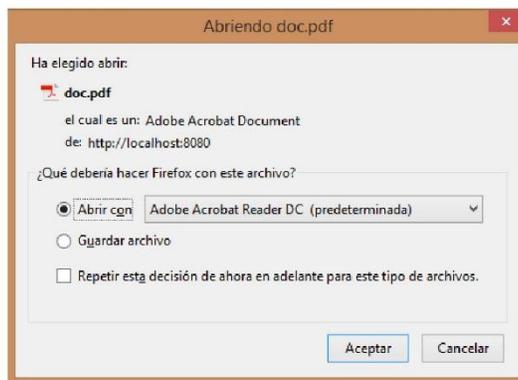
Si desea consultar todos los asesores/directores que inicien con una letra en específico seleccione el botón con la letra, o si conoce el título o como inicia puede teclearlo en “O introducir las primeras letras” y presiona “Consulta” o “Reporte”:

Visualizar por asesor/director:



O introducir las primeras letras:

Si lo que presionó fue “Reporte” le aparecerá un recuadro para guardar o abrir el reporte en formato PDF, presiona “Aceptar” para visualizar el contenido:



#### 4.4 Por Autor

En la barra de menú selecciona “Consultas/Reportes” y “Autor”:



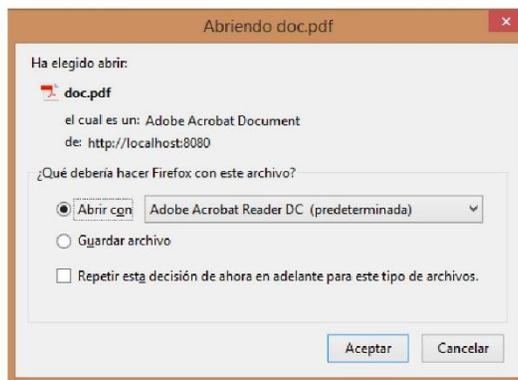
Si desea consultar todos los autores que inicien con una letra en específico seleccione el botón con la letra, o si conoce el título o como inicia puede teclearlo en “O introducir las primeras letras” y presiona “Consulta” o “Reporte”:

Visualizar por autor:



O introducir las primeras letras:

Si lo que presionó fue “Reporte” le aparecerá un recuadro para guardar o abrir el reporte en formato PDF, presiona “Aceptar” para visualizar el contenido:



#### 4.5 Por título

En la barra de menú selecciona “Consultas/Reportes” y “Título”:

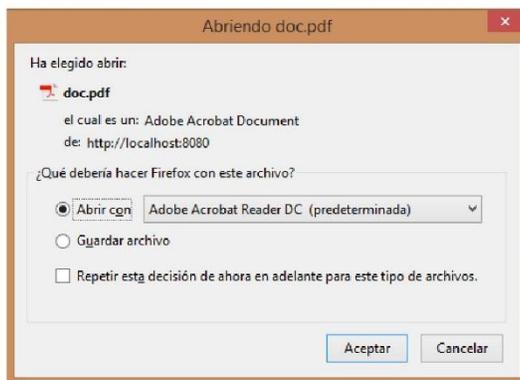


Si desea consultar todos los títulos que inicien con una letra en específico seleccione el botón con la letra, o si conoce el título o como inicia puede teclearlo en “O introducir las primeras letras” y presiona “Consulta” o “Reporte”:

Visualizar por título:



Si lo que presionó fue “Reporte” le aparecerá un recuadro para guardar o abrir el reporte en formato PDF, presiona “Aceptar” para visualizar el contenido:



*4.6 Consulta y reporte a partir de todo el material bibliográfico*

En la barra de menú selecciona “Consultas/Reportes” y “Todo”:



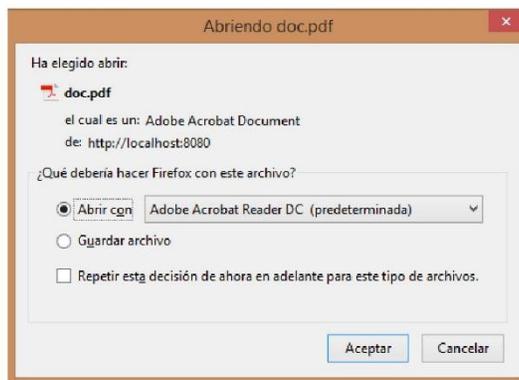
Tesis

+ Agregar Reporte

Buscar en el contenido de una tabla

Cadena a buscar

Titulo	Autor	Director	Carrera	PDF	Fecha	
Metodologia para el desarrollo de objetos de aprendizaje implementada a dispositivos moviles	Maria Guadalupe Perez Lopez	M.C. Jose	MSC		2015-12-28	 
Metodologia para el desarrollo de objetos de aprendizaje implementada a dispositivos moviles	Maria Guadalupe Perez Lopez	M.C. Jose	MSC		2015-12-29	 



## 5. Exportar e importar base de datos

Esta opción permite exportar e importar los datos de los usuarios y las tesis del sistema.

### 5.1 Exportar

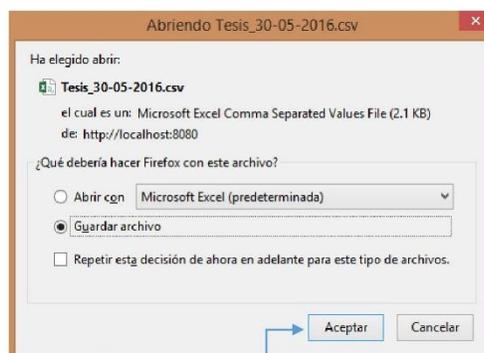
Para exportar los datos de los usuarios o de las tesis, deberá primero seleccionar de la barra de menú "Recuperar" y después "Exportar/Importar":



En la sección de "Exportar" marca el círculo de la opción que desea exportar (Tesis o Usuarios) y finalmente presiona el botón "Exportar":



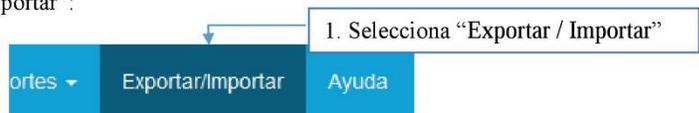
Una vez hecho esto se mostrará un cuadro para guardar el archivo, presiona aceptar para guardarlo en su equipo de cómputo:



4. Presione “Aceptar” para guardar el archivo

### 5.2 Importar

Para importar los datos de los usuarios o de las tesis, deberá primero seleccionar de la barra de menú “Exportar / Importar”:



Marca el círculo de lo que quiera importar usuarios o tesis:

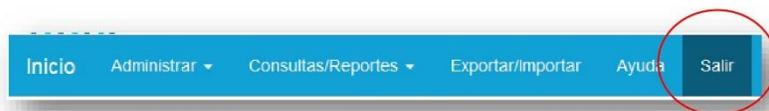


Elije el archivo en la opción de “Examinar” del cual va a importar y finalmente da clic en “Exportar”:



#### 6. Salir del sistema

Para salir del sistema seleccione de la barra de menú “Salir”, el cual cerrará su sesión y lo regresará a Login:



#### 7. Ayuda

Para poder visualizar este material puede presionar la opción Ayuda de la barra de menú:



## Anexo 2. Publicaciones

Memorias del Congreso Internacional  
de Investigación Academia Journals  
Tlaxcala 2016

Copyright 2016 Academia Journals

Tlaxcala, Tlaxcala, México  
17 y 18 de marzo de 2016

### Administración de recursos bibliográficos utilizando tecnologías web para bibliotecas de instituciones educativas de nivel superior

Lic. Lorena Zamora Velázquez<sup>1</sup>, M. en C. José Juan Hernández Mora<sup>2</sup>, M. en C. María Guadalupe Medina Barrera<sup>3</sup>,  
Mtra. Astrid Ariadna Fernández Torres<sup>4</sup>

**Resumen**—En este trabajo se presenta una metodología ágil para equipos de trabajo en cual los integrantes se encuentran en puntos geográficamente distantes, haciendo uso de las tecnologías de la información y comunicación para desarrollar un sistema de información. En el funcionamiento de la metodología propuesta se realiza el diseño e implementación de un sistema para la administración de recursos bibliográficos digitales en plataforma web; el propósito de este software es facilitar la consulta, administración y distribución de información de documentos digitales, incluyendo la gestión de tesis en formato PDF. Para realizar las pruebas de funcionalidad del sistema se cuenta con el apoyo de los Institutos Tecnológicos de Apizaco y del Altiplano ubicados en el Estado de Tlaxcala.

**Palabras clave**—Metodología Ágil, Desarrollo Global de Software, Sistemas de información digital.

#### Introducción

En la actualidad existe la tendencia a desarrollar software a través del uso de metodologías ágiles y el desarrollo global de software (GSD, Global Software Development), las cuales se consideran ideales para la construcción de un Sistema de Administración de Recursos Bibliográficos (SARB), el cual tenga fundamento en dichas tendencias y de esta manera reducir la distancia entre el equipos de trabajo localizados en sitios remotos, así como la entrega rápida de software incremental. Sin embargo el verdadero problema está en lograr resolver los problemas que surgen por la distancia (comúnmente de comunicación). Un sistema de administración de documentos digitales permite un ahorro en papel, pues al estar estas de manera digital no solo contribuye al medio ambiente, también contribuye al ahora de espacio, independencia de los documentos, acceso eficiente a los archivos, integridad y seguridad de la información y mejor administración.

Una aplicación web para la administración además permite lograr una más amplia difusión del material bibliográfico hacia un mayor número de usuario en línea, así como una gestión eficiente en forma remota de dichos documentos que conforman el recurso bibliográfico. Al desarrollar un software es indispensable hacer uso de los elementos y herramientas necesarias para su implementación, que den la confianza de que el trabajo realizado sea de calidad, cumpliendo con los plazos establecidos en tiempo y forma, correcta administración del recurso, realizando cada etapa de manera satisfactoria, se asegura que se cumple con los requisitos del cliente y que quede conforme con el trabajo entregado.

#### Antecedentes

##### *Desarrollo Global de Software Ágil (A-GSD)*

Con el uso de las Tecnologías de la Información y Comunicación (TIC) en tiempos actuales el desarrollo de actividades de un grupo de trabajo no se ven afectadas por la distancia. Con el surgimiento de Internet y otros factores económicos dieron el ambiente idóneo para que fuera posible que se proporcionaran servicios desde puntos distantes de donde fueron solicitados. En cuanto a desarrollo de software se refiere este es uno de los campos en el que ahora es más usual esta manera de trabajo entre los desarrolladores. Actualmente no es raro ver que varias empresas importantes desarrolladoras de software generan de manera completa o en parte su software en lugares distantes de donde se localiza el cliente final. A este estilo de trabajo se le conoce como Desarrollo Global de Software (GSD) (Minoli, de Castro, & Garzías, 2010).

Al mismo tiempo, las Metodologías Ágiles es otra tendencia que provee una serie de técnicas y métodos para la gestión de proyectos de software. El objetivo de estas metodologías es generar de manera rápida software funcional.

<sup>1</sup> Lic. Lorena Zamora Velázquez estudiante de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, [Lorena-zamora@hotmail.com](mailto:Lorena-zamora@hotmail.com)

<sup>2</sup> M. en C. José Juan Hernández Mora profesor investigador de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, [jjhora@itapizaco.edu.mx](mailto:jjhora@itapizaco.edu.mx)

<sup>3</sup> M. en C. María Guadalupe Medina Barrera profesora investigadora de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, [lupita\\_medina@hotmail.com](mailto:lupita_medina@hotmail.com)

<sup>4</sup> Mtra. Astrid Ariadna Fernández Torres investigadora del Instituto Tecnológico del Altiplano de Tlaxcala, [astrid\\_00@live.com.mx](mailto:astrid_00@live.com.mx)

El software se va produciendo a través de incrementos con nuevas funcionalidades, cada vez en periodos cortos de tiempo (Sommerville, 2011). La idea de combinar estas dos tendencias, es normal dado que se están empleando cada vez con más frecuencia en el desarrollo de software, pero esto no es algo sencillo, pues se deben tener en cuenta que al trabajar con un grupo de personas no solo se trata de seguir una serie de lineamientos y pasos de una metodología, sino otros factores que se deben considerar para generar un ambiente en el que se dé una buena comunicación, que es el elemento clave para la productividad de los integrantes del equipo (Minoli, de Castro, & Garzías, 2010).

#### *Retos en el Desarrollo global de software (GSD)*

El desarrollo global de Software nos presenta varios desafíos, estos son algunos (Fryer & Mats, 2008):

- Se requieren modelos de procesos que permitan colaborar a varias organizaciones y no solo a una sobre un mismo proyecto proporcionando formas de organización, colaboración y coordinación, esto se traduce actualmente en una baja de hasta el 50% en la productividad.
- Establecer una buena comunicación es algo complejo aun pensando en un solo grupo, pero si este se divide en equipos ubicados en lugares diferentes el problema de la comunicación se hace más grande originando información incorrecta, exclusión de cosas relevantes, fallas, lo que da lugar a realizar doble trabajo.
- El coordinar las tareas para varios equipos distribuidos geográficamente es muy difícil, ya que si no se lleva a cabo adecuadamente las consecuencias se reflejan en retraso en la entrega de resultados.
- La información no llega de manera oportuna a todos los integrantes de los equipos distribuidos por distintos factores como los husos horarios de los lugares donde pueden estar localizados.
- Definir las métricas en equipos que colaboran de manera distante, para evaluar el avance, productividad, el tiempo, entre otros aspectos puede resultar complejo si se consideran las desigualdades en el desarrollo o infraestructuras diversas, las cuales ocasionan información un tanto incoherente complicando la evaluación.
- Proteger los derechos de propiedad intelectual es algo de suma importancia, y sobre todo si alguno de los equipos de trabajo se encuentra en algún país donde sus leyes respecto a este punto son muy suaves.
- Asegurar que las herramientas que se empleen para el desarrollo y la comunicación sean robustas para evitar la incompatibilidad que trae consigo problemas de integración.

#### *Aplicaciones Web*

Para comprender que es una aplicación web (web-based application) hay que tener en claro que se trata de una aplicación del tipo cliente/servidor, en el que el cliente (navegador o explorador de internet) al igual que el servidor (el servidor web) y el protocolo a través del cual se comunican (HyperText Transfer Protocol (HTTP)) se encuentran normalizados y no tienen que ser hechos por el programador de aplicaciones (Luján, 2001).

#### *Tecnologías de la Información y Comunicación (TIC)*

Las Tecnologías de la Información y la Comunicación, que suelen ser nombradas TIC, son tecnología que favorece el tratamiento de la información y el proceso de comunicación, que tienen gran importancia en la sociedad actual debido a que se hace uso de ellas en diversas actividades de la vida diaria satisfaciendo a muchas necesidades (Baelo & Cantón, 2009).

### **Metodología Propuesta**

La metodología propuesta que se emplea para el desarrollo de este software se muestra en la figura 1. En dicho modelo el proceso de producción de software se realiza en iteraciones (bloques temporales) e incrementos, cada iteración representa un avance en el proyecto (un incremento). En este modelo se toman en consideración cuatro etapas por iteración para el desarrollo general del sistema, la primera etapa se toman en cuenta los requisitos del cliente los cuales se analizan y comprenden para saber lo que se debe hacer, al igual que se establecen las herramientas de comunicación a través de internet que se emplearán para la comunicación de los equipos de trabajo remotos, la segunda etapa se refiere al diseño de esa iteración, la tercer etapa se refiere a la realización y pruebas del código, en la última etapa se realizan la integración de la iteración y su implementación dentro del sistema.



Figura 1. Metodología propuesta

Para el desarrollo de este sistema se cuenta con dos equipos de trabajo ubicados en puntos geográficamente separados dentro del estado de Tlaxcala, como se presenta en la figura 2:



Figura 2. Equipos de trabajo para el desarrollo del sistema

A pesar de encontrarse estos dos equipos separados, fue importante la existencia de una única reunión inicial cara a cara, para conocer a todos los integrantes y así como los requisitos del cliente, como punto de partida, todo quedó registrado en documentos que se compartieron a través de internet hacia todos los integrantes. Los requisitos por parte del cliente los cuales se obtuvieron a partir de una entrevista de manera general se mencionan a continuación:

1. Se requiere que cada usuario se identifique dentro del sistema con nombre de usuario y contraseña, con cuatro niveles de usuarios: Administrador, Jefe del centro de información, Operador y Visitante. Cada usuario con distinto tipo de acceso al sistema.
2. Administrar (altas, modificación, bajas) de usuarios.
3. Administrar (altas, modificación, bajas) de tesis.
4. Realizar consultas del material bibliográfico.
5. Generar diversos reportes por: consultas realizadas y material bibliográfico.

**Plan para la Administración del Proyecto**

La figura 3 se muestra el plan de administración del proyecto, que será tomado a lo largo del desarrollo del sistema, con la finalidad de asegurar la buena comunicación entre los equipos de trabajo, puesto que este es el verdadero reto a superar ya que no solo se trata de seguir una serie de pasos de una metodología, sino de una verdadera integración por parte de los desarrolladores.



Figura 3. Plan de administración para el proyecto

**Comprensión**

Para el análisis se emplearon diagramas UML en los que de manera gráfica se visualizan las funciones del sistema, la figura 4 es un diagrama de casos de uso que nos da un panorama general del sistema en el cual podemos identificar de manera sencilla a cada uno de los usuarios con sus actividades correspondientes, esto a partir de los requerimientos generales del cliente.

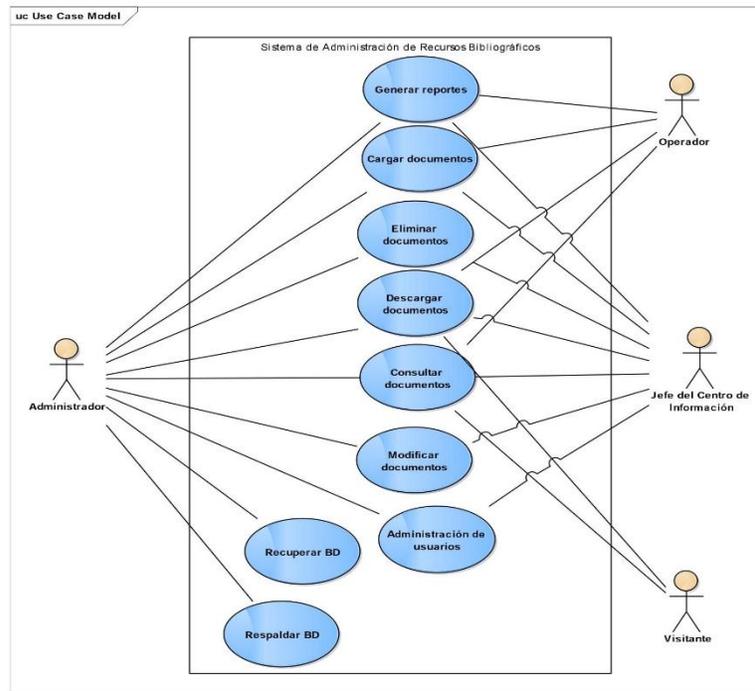


Figura 4. Panorama general del sistema

**Modelado**

En la figura 5 podemos visualizar el mapa de navegación del sistema, a través del mapa podemos ver de manera esquemática la estructura de la aplicación web, esto da la pauta para hacer el diseño de la interfaz de usuario (Figura 6) siguiendo este esquema como guía.

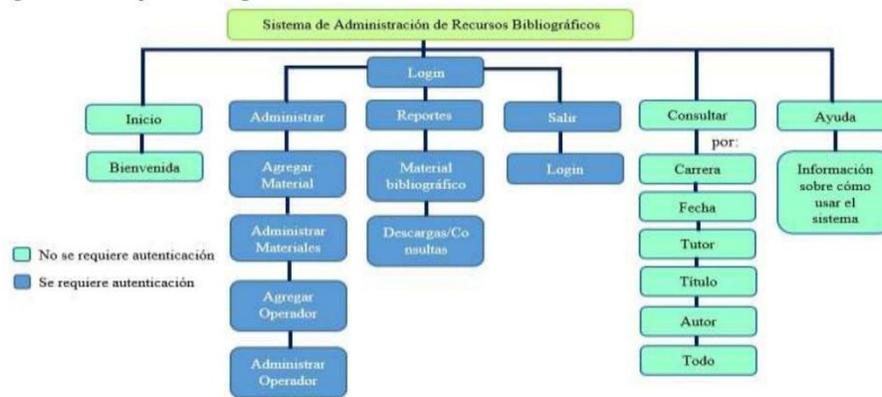


Figura 5. Mapa de navegación de sistema



Figura 6. Diseño general de la interfaz de usuario

### Conclusiones

Se utiliza la metodología propuesta en este trabajo para el desarrollo de un sistema como prueba, al inicio del desarrollo en las iteraciones se demostró que es posible trabajar con equipos separados geográficamente, con lo cual se obtuvo la versión preliminar del proyecto. El principal obstáculo a vencer, como se había previsto fue la mala comunicación, afectando directamente a la coordinación y organización de los equipos y tareas a realizar, lo que ocasionó que existiera un desfase en el calendario de las actividades, y que el trabajo que se estaba realizando no fuera compatible entre los equipos, por tal motivo se recomienda, una primera y única reunión presencial incluyendo al cliente, de no poderse efectuar por cuestiones de la distancia, haciendo uso de la tecnología realizar la reunión a través de una videoconferencia. Esta primera reunión además de proporcionar los requerimientos generales del cliente, es una buena oportunidad para conocer a los integrantes de los equipos de desarrollo y tratar de establecer buenas relaciones entre ellos. Definiendo de manera correcta las funciones de los equipos de trabajo. Para solucionar el problema de la mala comunicación entre los equipos se estableció un solo medio de comunicación virtual (común entre todos los integrantes) en el cual se determinan los estándares y formatos para el intercambio de información, como el compartir toda la información a través de carpetas con nombres claros y bien definidos, al inicio de las videoconferencias para revisión de avances y definición de actividades se comenzó con charlas informales para centrar la atención, posteriormente se establecieron reuniones virtuales formales con compromisos y metas que se les da seguimiento en cada reunión, logrando acuerdos y resultados que enfocaron a los equipos en los objetivos del proyecto como fin común dejando de lado las diferencias personales. Terminando el proyecto en tiempo y forma, y con la calidad requerida.

### Referencias

- Baelo, R., & Cantón, I. (2009). Las tecnologías de la información y la comunicación en la educación superior. Estudio descriptivo y de revisión. *Revista Iberoamericana de Educación*.
- Fryer, K., & Mats, G. (15 de Enero de 2008). developerWorks. Obtenido de IBM developerWorks: <http://sg.com.mx/content/view/1038>
- Luján, S. (2001). *Programación en Internet: Clientes Web*. España: Editorial Club Universitario.
- Minoli, M., de Castro, V., & Garzàs, J. (2010). Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de Ingeniería de requisitos. *Revista Española de Innovación, Calidad e Ingeniería de Software*, 66-75.
- Sommerville, I. (2011). *Ingeniería de Software*. México: Pearson Educación.

ACADEMIA JOURNALS



Primer Congreso Internacional de Academia Journals  
en Educación Superior Tecnológica Pública

Tlaxcala 2016

*Certificado*

*otorgado a*

Lic. Lorena Zamora Velázquez  
M. C. José Juan Henández Mora  
M. C. Ma. Guadalupe Medina Barrera  
Mtra. Astrid Ariadna Fernández Torres

*por su artículo intitulado*

Administración de recursos bibliográficos utilizando tecnologías web para bibliotecas de instituciones  
educativas de nivel superior

(Artículo No. Tlax002)

el cual fue presentado en el Congreso desarrollado del 16 al 18 de marzo 2016 en Tlaxcala, México,  
y publicado en el portal de Internet *AcademiaJournals.com*,  
con número de registro **ISSN 1946-5351, VOL. 8, NO. 2**  
y en el libro electrónico online con **ISBN 978-1-939982-21-6** intitulado  
*Compendio de Investigación Academia Journals Tlaxcala 2016*

Dr. Rafael Moras  
Editor, Academia Journals  
Profesor de Ing. Industrial, St. Mary's University



## Diseño y desarrollo de una base de datos para un sistema de administración de recursos bibliográficos en instituciones educativas de nivel superior

Lic. Lorena Zamora Velázquez<sup>1</sup>, M. en C. José Juan Hernández Mora<sup>2</sup>, M. en C. María Guadalupe Medina Barrera<sup>3</sup>,  
Mtra. Astrid Ariadna Fernández Torres<sup>4</sup>

**Resumen**— Este trabajo tiene como objetivo el diseño y desarrollo de una base de datos para un Sistema de Administración de Recursos Bibliográficos dirigido a instituciones educativas de nivel superior, específicamente tesis creadas por alumnos de estas instituciones. Puesto que no se tiene un sistema automatizado para el préstamo o publicación de tesis, y debido a la tendencia actual que tiene la información de ser en forma digital, surge la necesidad de crear un sistema que le permita a dichas instituciones poder administrar en forma digital sus productos académicos bibliográficos y ponerlos en línea. Para esto antes se requiere el diseño y desarrollo de la base de datos, la cual se hizo con un sistema de base de datos a través de una metodología ágil. Esta base de datos permite que se almacenen los recursos bibliográficos, lo que mejorará el control y distribución de los mismos y se almacenaran los usuarios con sus respectivos datos y contraseñas que les permitirán tener acceso al sistema.

**Palabras clave**— Sistema de base de datos, Sistemas de información digital, Ingeniería del software

### Introducción

Actualmente, las bases de datos son parte esencial de distintas áreas en las que se requiere manejar grandes volúmenes de datos de forma organizada y computarizada, puesto que estas permiten tanto registrar gran cantidad de datos así como la consulta de los mismos de manera rápida. Crear una base de datos para un sistema de administración de documentos digitales permite un ahorro en papel, pues al estar estas de manera digital no solo contribuye al medio ambiente, también contribuye al ahorro de espacio, independencia de los documentos, acceso eficiente a los archivos, integridad y seguridad de la información y mejor administración.

Una base de datos que para una aplicación web diseñada para la administración permite una más amplia difusión del material bibliográfico hacia un mayor número de usuario en línea, así como una gestión eficiente en forma remota de dichos documentos que conforman el recurso bibliográfico. Al desarrollar un software es indispensable hacer uso de los elementos y herramientas necesarias para su implementación, que den la confianza de que el trabajo realizado sea de calidad, cumpliendo con los plazos establecidos en tiempo y forma, correcta administración del recurso, realizando cada etapa de manera satisfactoria, se asegura que se cumple con los requisitos del cliente y que quede conforme con el trabajo entregado.

### Antecedentes

#### *Ingeniería del software*

La ingeniería de software engloba procesos, métodos y herramientas, lo que permite obtener sistemas complejos basados en computadora en el tiempo establecido y con la calidad requerida. Desarrollar software consta de 5 actividades estructurales: comunicación, planeación, modelado, construcción y despliegue. Las actividades mencionadas se emplean en todos los proyectos de software. Llevar a cabo la práctica de la ingeniería de software es idónea para resolver problemas, puesto que sigue una serie de principios fundamentales (Sommerville, 2011).

#### *Aplicaciones Web*

Para comprender que es una aplicación web (web-based application) hay que tener en claro que se trata de una aplicación del tipo cliente/servidor, en el que el cliente (navegador o explorador de internet) al igual que el servidor

<sup>1</sup> Lic. Lorena Zamora Velázquez estudiante de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, [Lorena-zamora@hotmail.com](mailto:Lorena-zamora@hotmail.com)

<sup>2</sup> M. en C. José Juan Hernández Mora profesor investigador de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, [jhmora@itapizaco.edu.mx](mailto:jhmora@itapizaco.edu.mx)

<sup>3</sup> M. en C. María Guadalupe Medina Barrera profesora investigadora de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, [lupita\\_medina@hotmail.com](mailto:lupita_medina@hotmail.com)

<sup>4</sup> Mtra. Astrid Ariadna Fernández Torres investigadora del Instituto Tecnológico del Altiplano de Tlaxcala, [astrid\\_00@live.com.mx](mailto:astrid_00@live.com.mx)

(el servidor web) y el protocolo a través del cual se comunican (HyperText Transfer Protocol (HTTP)) se encuentran normalizados y no tienen que ser hechos por el programador de aplicaciones (Luján, 2001).

**Sistema gestor de bases de datos**

Un sistema gestor de base de datos (SGBD), no es otra cosa que una colección de datos relacionados entre sí, así como una serie de programas que permiten el acceso a estos datos. La colección de datos, generalmente llamada base de datos, contiene información de suma importancia para una empresa u organización. El objetivo principal de un SGBD es el de otorgar una manera de recopilar y recobrar la información de una base de datos de tal forma que sea tanto práctica como eficiente. Los sistemas de bases de datos se crean para administrar enormes volúmenes de información. La administración de los datos involucra tanto la definición de estructuras para recopilar información como el suministro de mecanismos para el manejo de información almacenada, aún en las caídas del sistema o de los intentos de acceso no autorizados. A pesar de que los datos vayan a compartirse entre diferentes usuarios, el sistema debe impedir en la medida de lo posible resultados extraños. (Silberschatz, Korth, & Sudarshan, 2006)

**Tecnologías de la Información y Comunicación (TIC)**

Las Tecnologías de la Información y la Comunicación, que suelen ser nombradas TIC, son tecnología que favorece el tratamiento de la información y el proceso de comunicación, que tienen gran importancia en la sociedad actual debido a que se hace uso de ellas en diversas actividades de la vida diaria satisfaciendo a muchas necesidades (Baelo & Cantón, 2009).

**Metodología Propuesta**

La metodología propuesta que se emplea para el desarrollo de este software se muestra en la figura 1. En dicho modelo el proceso de producción de software se realiza en iteraciones (bloques temporales) e incrementos, cada iteración representa un avance en el proyecto (un incremento). En este modelo se toman en consideración cuatro etapas por iteración para el desarrollo general del sistema, la primera etapa se toman en cuenta los requisitos del cliente los cuales se analizan y comprenden para saber lo que se debe hacer, al igual que se establecen las herramientas de comunicación a través de internet que se emplearán para la comunicación de los equipos de trabajo remotos, la segunda etapa se refiere al diseño de esa iteración, la tercer etapa se refiere a la realización y pruebas del código, en la última etapa se realizan la integración de la iteración y su implementación dentro del sistema.



Figura 1. Metodología propuesta

**Desarrollo de la metodología**

A pesar de encontrarse geográficamente separados los dos (Baelo & Cantón, 2009) equipos que participaron en de desarrollo, fue importante la existencia de una única reunión inicial cara a cara, para conocer a todos los integrantes y así como los requisitos del cliente, como punto de partida, todo quedó registrado en documentos que se compartieron a través de internet hacia todos los integrantes. Los requisitos por parte del cliente los cuales se obtuvieron a partir de una entrevista de manera general se mencionan a continuación:

1. Se requiere que cada usuario se identifique dentro del sistema con nombre de usuario y contraseña, con cuatro niveles de usuarios: Administrador, Jefe del centro de información, Operador y Visitante. Cada usuario con distinto tipo de acceso al sistema.
2. Administrar (altas, modificación, bajas) de usuarios.
3. Administrar (altas, modificación, bajas) de tesis.
4. Realizar consultas del material bibliográfico.
5. Generar diversos reportes por: consultas realizadas y material bibliográfico.

**Etapa 1. Comprensión**

En esta etapa para la parte de análisis se utilizaron diagramas UML en los que de forma gráfica se presentan las funciones del sistema, la figura 2 muestra un diagrama de casos de uso que nos proporciona un panorama general del sistema en el cual podemos identificar de manera sencilla a cada uno de los usuarios con sus actividades correspondientes, esto en base a los requerimientos generales del cliente.

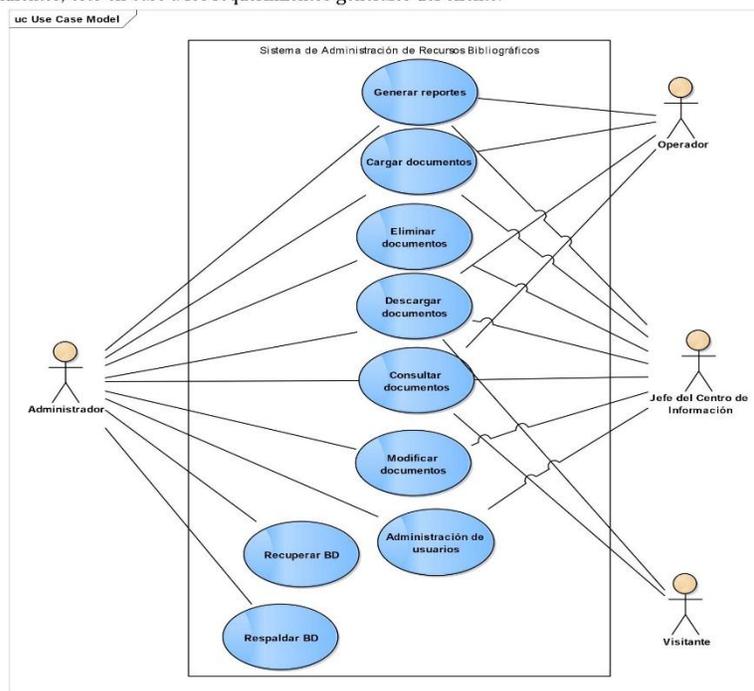


Figura 2. Panorama general del sistema

**Etapa 2. Modelado**

Tomando en cuenta que la base de datos a desarrollar corresponde a una aplicación web, se propone la arquitectura cliente/servidor, puesto que nos permite la separación de funciones.

La arquitectura cliente/servidor se describe en la figura 3. Se divide en 3 niveles definidos en 1: Lógica presentación, 2: Lógica de negocio y 3: Lógica de datos (Luján, 2001).

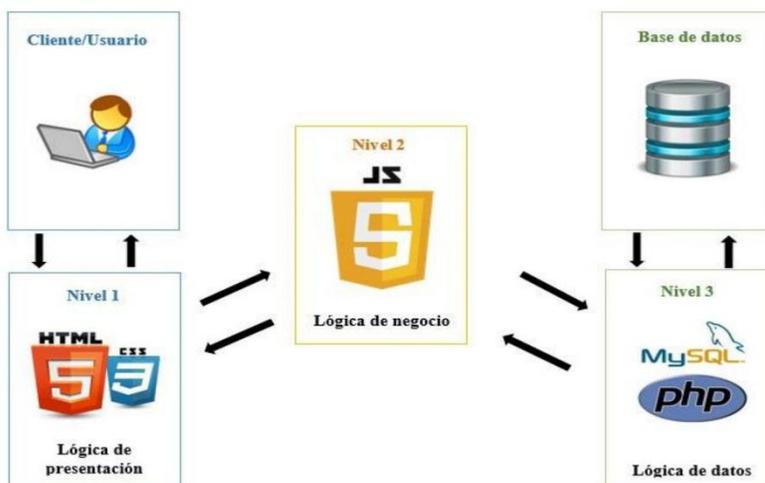


Figura 3. Patrón cliente/servidor

El nivel 1 del sistema es la parte que se presenta al usuario final, esto es la lógica de presentación en la cual se muestran menús, formularios, la presentación de datos, esta parte facilita la interacción del usuario con el sistema. Se tienen 4 menús según el nivel de acceso:

- Administrador
- Jefe del centro de información
- Operador
- Visitante

En el nivel 2 que es la lógica de negocio en la cual están todos los programas que serán ejecutados, estos reciben las peticiones de los usuarios y mandan las respuestas al nivel 1: lógica de presentación.

La lógica de datos es la encargada de llevar a cabo las transacciones con la base de datos, este nivel involucra:

- Altas, bajas, modificaciones.
- Consultas

**Nivel 1: Lógica de presentación al usuario**

Por medio de un diseño de interfaz de usuario se propone el diseño de la figura 4.



Figura 4. Diseño general de la interfaz de usuario

**Nivel 2: La lógica de negocio.**

En base a los requerimientos del cliente, en este nivel se encuentran las distintas funciones que contiene el sistema considerando los diferentes tipos de usuarios, las cuales se pueden apreciar en la Tabla 1.

Funciones	Niveles			
	Administrador	Jefe del C.I.	Operador	Visitante
<b>Administración de usuarios</b>				
F1. Generar altas	✓	✓		
F2. Generar bajas	✓	✓		
F3. Generar modificaciones	✓	✓		
F4. Generar consultas	✓	✓		
<b>Tesis</b>				
F5. Generar altas tesis	✓	✓	✓	
F6. Generar bajas tesis	✓	✓		
F7. Generar modificaciones tesis	✓	✓		
F8. Realizar consultas tesis	✓	✓	✓	✓
F9. Realizar descargas tesis	✓	✓	✓	✓
F10. Generar reportes	✓	✓	✓	
<b>Seguridad</b>				
F11. Recuperar base de datos	✓			
F12. Respaldar base de datos	✓			

Tabla 1. Funciones generales del sistema.

**Nivel 3: La lógica de datos.**

El nivel 3, que es la lógica de datos en la cual se llevan a cabo las transacciones a las tablas de la base de datos, el diseño general se puede visualizar en la figura 5.

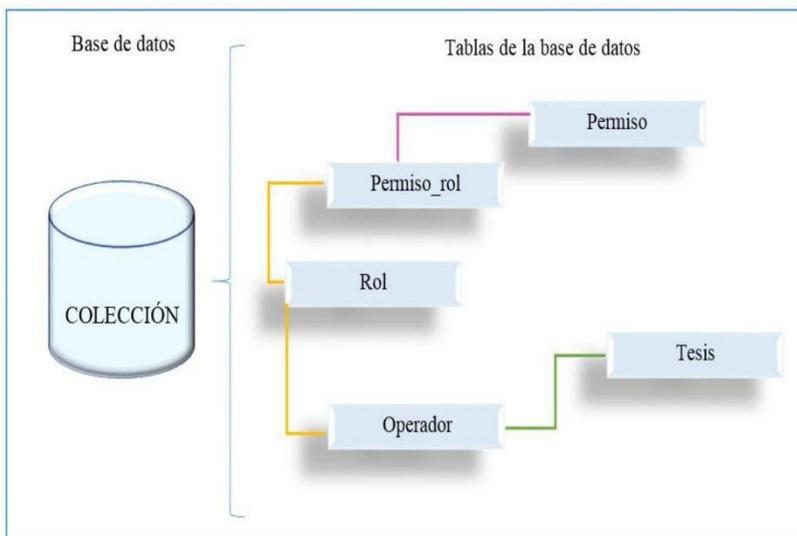


Figura 5. Base de datos COLECCIÓN

Las transacciones que se efectúan en las tablas de la base de datos son: altas, bajas, modificaciones y consultas, por medio de dichas transacciones se da respuesta a la capa de gestión de peticiones.

### Conclusiones

El diseño y desarrollo de esta base de datos ha permitido satisfacer los requerimientos del cliente para el sistema de administración de recursos bibliográficos en instituciones educativas de nivel superior, el cual ayuda a administrar de manera eficiente las tesis realizadas por el Instituto Tecnológico de Apizaco y del Instituto Tecnológico del Altiplano de Tlaxcala, hasta este punto es una primera versión de lo que será el sistema final, en sus posteriores iteraciones. La aplicación de la metodologías propuesta y de las diferentes tecnologías web hacen viable que el sistema se vaya desarrollando de manera propicia, es importante en todo momento tomar en cuenta con sumo cuidado y atención los requerimientos por parte del cliente, y que este se mantenga en comunicación con los equipos de desarrollo a lo largo del proceso de elaboración del sistema, lo que asegura un producto de software final que cumpla con la función para la cual es desarrollado.

### Referencias

Baelo, R., & Cantón, I. (2009). Las tecnologías de la información y la comunicación en la educación superior. Estudio descriptivo y de revisión. Revista Iberoamericana de Educación.

Luján, S. (2001). Programación en Internet: Clientes Web. España: Editorial Club Universitario.

Silberschatz, A., Korth, H., & Sudarshan, S. (2006). Fundamentos de base de datos. Quinta Edición. España: McGraw- Hill.

Sommerville, I. (2011). Ingeniería de Software. México: Pearson Educación.



**BUAP** / **INGENIERÍA**  
FACULTAD

ACADEMIA JOURNALS



Congreso Internacional de Investigación  
Academia Journals Puebla 2016

AcademiaJournals.com

# Certificado

otorgado a

Lic. Lorena Zamora Velázquez  
M. C. José Juan Henández Mora  
M. C. Ma. Guadalupe Medina Barrera  
Mtra. Astrid Ariadna Fernández Torres

por su artículo intitulado

**Diseño y desarrollo de una base de datos para un sistema de administración de recursos bibliográficos en instituciones educativas de nivel superior (Artículo No. Pue461)**

el cual fue presentado en el Congreso desarrollado los días 9 y 10 de junio 2016 en la ciudad de Puebla, México, e incluido en las publicaciones del congreso, mismas que incluyen un volumen online con número ISSN 1946-5351, Vol. 8 No. 3, 2016 y un libro digital (ebook), mismo que tiene ISBN 978-1-939982-22-3 con código de barras.

**Dr. Rafael Moras**

Editor, Academia Journals  
Profesor de Ing. Industrial, St. Mary's University



**M. en I. Fernando Daniel Lázcano Hernández**  
Director de la Facultad de Ingeniería

**M. H. D. Julia I. Rodríguez Morales**  
Coordinadora General del Congreso

Pue461

## Anexo 3. Cartas

## 3.1 Carta de satisfacción



TECNOLÓGICO NACIONAL DE MÉXICO  
Instituto Tecnológico del Altiplano de Tlaxcala

San Diego, Xocoyucan, Tlaxcala, 08 de junio de 2016

**ASUNTO: Carta de satisfacción**

**MTRO. FELIPE PASCUAL ROSARIO AGUIRRE**  
**DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO**  
**P R E S E N T E:**

A través de este medio informo a usted que la Lic. Lcena Zamora Velázquez alumna de la **Maestría en Sistemas Computacionales**, con No. de control **M08370722** terminó de forma satisfactoria el proyecto que llevaba a cabo titulado **"Diseñar e implementar un Sistema de Gestión de Producto Académico Bibliográfico en Instituciones de Nivel Superior, para el manejo automático de bibliografía digital"**. El proyecto en mención se desarrolló durante su estancia técnica en el Centro de Información del Instituto Tecnológico del Altiplano de Tlaxcala, donde fungió como asesor externo y responsable la **Mtra. Astrid Ariadna Torres Fernández, Jefa del Centro de Información** durante el periodo comprendido del **17 de Agosto al 11 de diciembre del 2015**, cumpliendo con los horarios asignados los días miércoles y jueves de 8:00 a 16:00 hrs.

Sin otro particular por el momento reciba un cordial saludo.

**ATENTAMENTE**  
*Vida, tierra, ciencia®*



**M. I. ELIUBÍ ECHEVERRÍA LANDÍN**  
**DIRECTOR DEL ITAT**

TECNOLÓGICO NACIONAL  
DE MÉXICO  
CLAVE: 29DIT0001F  
I. TEC. DEL ALTIPLANO DE TLAXCALA  
DIRECCIÓN



Km 7.5 Carretera Federal San Martín Tez-Tlaxcala  
San Diego Xocoyucan, Tlax. C.P. 90122  
Tel. 012484617247, Tel y Fax: 012484613972, e-mail: dir\_altiplano@tecnm.mx  
www.italtiplanotlaxcala.edu.mx



## 3.2 Carta de término de estancias



TECNOLÓGICO NACIONAL DE MÉXICO  
Instituto Tecnológico del Altiplano de Tlaxcala

San Diego, Xocoyucan, Tlaxcala, 08 de junio de 2016

**ASUNTO: Carta de liberación**

**MTRO. FELIPE PASCUAL ROSARIO AGUIRRE**  
**DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO**  
**PRESENTE:**

La presente es para informar que la estudiante de maestría **Lic. Lorena Zamora Velázquez**, del Instituto Tecnológico de Apizaco realizó una estancia en el Centro de Información del Instituto Tecnológico del Altiplano de Tlaxcala, con el proyecto titulado **“Diseñar e implementar un Sistema de Gestión de Producto Académico Bibliográfico en Instituciones de Nivel Superior, para el manejo automático de bibliografía digital”**. La estancia tuvo un periodo comprendido del **17 de agosto al 11 de diciembre de 2015**, donde fungió como asesor externo y responsable la **Mtra. Astrid Ariadna Torres Fernández**, Jefa del Centro de Información durante el periodo comprendido. La estancia y trabajo de la estudiante fue satisfactoria.

Sin otro particular por el momento reciba un cordial saludo.

**ATENTAMENTE**

*Vida, tierra, ciencia®*

**M.I. ELIUBI ECHEVERRÍA LANDÍN**  
**DIRECTOR DEL ITAT**



TECNOLÓGICO NACIONAL  
DE MÉXICO  
CLAVE: 28DIT0001F  
I. TEC. DEL ALTIPLANO DE TLAXCALA  
DIRECCIÓN



*L*

Km 7.5 Carretera Federal San Martín Tex-Tlaxcala  
San Diego Xocoyucan, Tlax. C.P. 90122

Tel. 012484617247, Tel y Fax 012484613072, e-mail: [dir\\_altiplano@tecnm.mx](mailto:dir_altiplano@tecnm.mx)  
[www.italtiplanotlaxcala.edu.mx](http://www.italtiplanotlaxcala.edu.mx)



## Anexo 4.Cuestionarios

**Proyecto:** Sistema de Administración de Recursos Bibliográficos

PRUEBAS GENERALES DEL SISTEMA			
<b>Pruebas de funcionalidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿Tiene las funciones para las tareas que fueron requeridas?	X		
2. ¿Hace lo que fue acordado de forma esperada y correcta?		X	
3. ¿El acceso a las funciones restringidas del SARB está limitado a usuarios registrados autorizados?	X		
<b>Pruebas de fiabilidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿El sistema funciona sin presentar fallos?	X		
2. ¿El sistema es capaz de recuperar los datos de un fallo?	X		
3. ¿El sistema puede detectar datos erróneos al momento de introducirlos?	X		
<b>Pruebas de eficiencia</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿El tiempo de respuesta para un proceso es adecuado?	X		
2. ¿El sistema funciona con una velocidad aceptable aun cuando se tienen varios programas en ejecución?	X		
<b>Pruebas de mantenibilidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿Es fácil diagnosticar o identificar partes a modificar?	X		
2. ¿No existen riesgos o efectos inesperados cuando se realizan cambios?	X		
3. ¿Las modificaciones son fáciles de validar?	X		
4. ¿Es fácil de modificar y adaptar?	X		
<b>Pruebas de portabilidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿Es posible utilizar la aplicación web con cualquier navegador?	X		
2. ¿Es fácil de instalar?	X		
3. ¿Comparte con facilidad recursos con otro software o dispositivo?	X		
4. ¿Es fácil de usarlo en lugar de otro software?	X		

*Vo.Bo*

---

FIRMA CLIENTE 1  
JOSÉ JUAN HERNÁNDEZ MORA

**Proyecto:** Sistema de Administración de Recursos Bibliográficos

PRUEBAS GENERALES DEL SISTEMA			
<b>Pruebas de funcionalidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿Tiene las funciones para las tareas que fueron requeridas?	X		
2. ¿Hace lo que fue acordado de forma esperada y correcta?	X		
3. ¿El acceso a las funciones restringidas del SARB está limitado a usuarios registrados autorizados?	X		
<b>Pruebas de fiabilidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿El sistema funciona sin presentar fallos?	X		
2. ¿El sistema es capaz de recuperar los datos de un fallo?	X		
3. ¿El sistema puede detectar datos erróneos al momento de introducirlos?		X	
<b>Pruebas de eficiencia</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿El tiempo de respuesta para un proceso es adecuado?	X		
2. ¿El sistema funciona con una velocidad aceptable aun cuando se tienen varios programas en ejecución?	X		
<b>Pruebas de mantenibilidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿Es fácil diagnosticar o identificar partes a modificar?	X		
2. ¿No existen riesgos o efectos inesperados cuando se realizan cambios?		X	
3. ¿Las modificaciones son fáciles de validar?	X		
4. ¿Es fácil de modificar y adaptar?	X		
<b>Pruebas de portabilidad</b>			
<i>Pregunta</i>	<i>Si</i>	<i>Medio</i>	<i>No</i>
1. ¿Es posible utilizar la aplicación web con cualquier navegador?	X		
2. ¿Es fácil de instalar?	X		
3. ¿Comparte con facilidad recursos con otro software o dispositivo?	X		
4. ¿Es fácil de usarlo en lugar de otro software?	X		

*Astrid A. F. T.*

---

FIRMA CLIENTE 2

ASTRID ARIADNA TORRES FERNÁNDEZ

## Anexo 5. Itinerarios, reportes y formatos de sugerencias y cambios

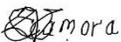
<b>Itinerario</b>			
Fecha	17 de agosto de 2015	Hora	2:00 pm
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 1
Módulo	Diseño y desarrollo de la base de datos		
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de diseño y desarrollo de la base de datos del ciclo 1. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía		
Rocío Hernández Acoltzi	Analista		
Lidia Cortés Ruiz	Diseñador		
René López Caballero	Desarrollador		
Said Pérez Flores	Tester		

<b>Itinerario</b>			
Fecha	25 de agosto de 2015	Hora	2:00
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 2
Módulo	Acceso al sistema		
Objetivo			
Definir el plan de actividades para el desarrollo del módulo de acceso al sistema del ciclo 2. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía		
Rocío Hernández Acoltzi	Analista		
Lidia Cortés Ruiz	Diseñador		
René López Caballero	Desarrollador		
Said Pérez Flores	Tester		

<b>Itinerario</b>			
Fecha	07 de septiembre de 2015	Hora	2:00 pm
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 3
	Módulo	Gestión de usuarios	
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de gestión de usuarios del ciclo 3. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía		
Rocío Hernández Acoltzi	Analista		
Lidia Cortés Ruiz	Diseñador		
René López Caballero	Desarrollador		
Said Pérez Flores	Tester		

<b>Itinerario</b>			
Fecha	23 de septiembre de 2015	Hora	2:00 pm
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 4
	Módulo	Gestión de materiales	
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de gestión de materiales del ciclo 4. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía		
Rocío Hernández Acoltzi	Analista		
Lidia Cortés Ruiz	Diseñador		
René López Caballero	Desarrollador		
Said Pérez Flores	Tester		

<b>Itinerario</b>			
Fecha	09 de octubre 2015	Hora	2:00 pm
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 5
	Módulo	Consultas	
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de consultas del ciclo 5. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía		
Rocío Hernández Acoltzi	Analista		
Lidia Cortés Ruiz	Diseñador		
René López Caballero	Desarrollador		
Said Pérez Flores	Tester		

<b>Itinerario</b>			
Fecha	06 de noviembre de 2015	Hora	2:00
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 6
	Módulo	Reportes	
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de diseño y reportes del ciclo 6. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía		
Rocío Hernández Acoltzi	Analista		
Lidia Cortés Ruiz	Diseñador		
René López Caballero	Desarrollador		
Said Pérez Flores	Tester		

<b>Itinerario</b>			
Fecha	19 de noviembre de 2015	Hora	2:00 pm
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 7
Módulo	Exportar e importar base de datos		
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de exportar e importar base de datos del ciclo 7. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Determinar las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía	L. Zamora	
Rocío Hernández Acoltzi	Analista	R. Hernández	
Lidia Cortés Ruiz	Diseñador	L. Cortés	
René López Caballero	Desarrollador	R. López	
Said Pérez Flores	Tester	S. Pérez	

<b>Itinerario</b>			
Fecha	30 de noviembre de 2015	Hora	2:00 pm
		Tipo de reunión	Previa
		Tiempo estimado de duración	2 horas
		Ciclo	No. 8
Módulo	Sección de ayuda		
Objetivo			
Establecer el plan de actividades para el desarrollo del módulo de la sección de ayuda del ciclo 8. Actualizar y presentar el plan de actividades.			
Puntos a abordar			
No.	Especificación	Tiempo asignado	
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	30 min.	
2	Determinar el número de días requeridos para concluir el módulo.	30 min.	
3	Medir el alcance que tendrá el módulo.	15 min.	
4	Obtener requerimientos.	30 min.	
5	Presentar el plan de comunicación actualizado	15 min.	
Participantes de la reunión			
Nombre	Función	Firma de enterado	
José Juan Hernández Mora	Dueño del software	Vo.Bo.	
Astrid Ariadna Torres Fernández	Dueño del software	Astrid A. F. T.	
Lorena Zamora Velázquez	Guía	L. Zamora	
Rocío Hernández Acoltzi	Analista	R. Hernández	
Lidia Cortés Ruiz	Diseñador	L. Cortés	
René López Caballero	Desarrollador	R. López	
Said Pérez Flores	Tester	S. Pérez	

<b>Reporte de reunión</b>				
Fecha de la reunión:		17 de agosto de 2015		Duración: 3 horas
Tipo de reunión:	Ciclo: 1	Módulo:	Diseño y desarrollo de la base de datos	
Participante	Asistencia	Causa de la falta		
1. José Juan Hernández Mora	Presente			
2. Astrid Ariadna Torres Fernández	Presente			
3 Lorena Zamora Velázquez	Presente			
4. Rocío Hernández Acoltzi	Presente			
5. Lidia Cortés Ruiz	Ausente	Desinformación		
6. René López Caballero	Presente			
7. Said Pérez Flores	Ausente	Desinformación		
Puntos del itinerario				
No.	Especificación	Abordado	Causa	Conclusiones
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.
4	Obtener requerimientos.	Parcial	Desintegración.	Requerimiento del cliente.
5	Presentar el plan de comunicación actualizado	Parcial	Inicio del proyecto.	Elementos de organización.
Se cumplió el objetivo de la reunión:			En un 90%	
Causa:			Falta de integración de los integrantes.	

<b>Reporte de reunión</b>				
Fecha de la reunión:		25 de agosto de 2015		Duración: 2 horas 20min.
Tipo de reunión:	Ciclo: 2	Módulo:	Acceso al sistema	
Participante	Asistencia	Causa de la falta		
1. José Juan Hernández Mora	Presente			
2. Astrid Ariadna Torres Fernández	Presente			
3 Lorena Zamora Velázquez	Presente			
4. Rocío Hernández Acoltzi	Ausente	Cuestiones académicas.		
5. Lidia Cortés Ruiz	Ausente	Cuestiones académicas.		
6. René López Caballero	Ausente	Cuestiones académicas.		
7. Said Pérez Flores	Presente			
Puntos del itinerario				
No.	Especificación	Abordado	Causa	Conclusiones
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.
4	Obtener requerimientos.	Si		Requerimiento del cliente.
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.
Se cumplió el objetivo de la reunión:			Se cumplió al 80%	
Causa:			Inasistencia de algunos integrantes.	

<b>Reporte de reunión</b>					
Fecha de la reunión:		07 de septiembre de 2015		Duración:	1 hora 50 min.
Tipo de reunión:		Ciclo:	3	Módulo:	Gestión de usuarios
Participante	Asistencia	Causa de la falta			
1. José Juan Hernández Mora	Ausente	Cuestiones laborales.			
2. Astrid Ariadna Torres Fernández	Ausente	Cuestiones laborales.			
3 Lorena Zamora Velázquez	Presente				
4. Rocío Hernández Acoltzi	Presente				
5. Lidia Cortés Ruiz	Presente				
6. René López Caballero	Ausente	Desinformación.			
7. Said Pérez Flores	Ausente	Desinformación.			
Puntos del itinerario					
No.	Especificación	Abordado	Causa	Conclusiones	
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.	
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.	
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.	
4	Obtener requerimientos.	Parcial	Inasistencia de los clientes	Requerimiento del cliente.	
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.	
<i>Se cumplió el objetivo de la reunión:</i>			En un 80%.		
<i>Causa:</i>			Inasistencia de algunos integrantes.		

<b>Reporte de reunión</b>					
Fecha de la reunión:		23 de septiembre de 2015		Duración:	2 hrs. 35 min.
Tipo de reunión:		Ciclo:	4	Módulo:	Gestión de materiales
Participante	Asistencia	Causa de la falta			
1. José Juan Hernández Mora	Ausente	Cuestiones laborales.			
2. Astrid Ariadna Torres Fernández	Presente				
3 Lorena Zamora Velázquez	Presente				
4. Rocío Hernández Acoltzi	Ausente	Razones de salud.			
5. Lidia Cortés Ruiz	Presente				
6. René López Caballero	Presente				
7. Said Pérez Flores	Presente				
Puntos del itinerario					
No.	Especificación	Abordado	Causa	Conclusiones	
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.	
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.	
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.	
4	Obtener requerimientos.	Si		Requerimiento del cliente.	
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.	
<i>Se cumplió el objetivo de la reunión:</i>			En un 97 %		
<i>Causa:</i>			Ausencia en la reunión de un cliente.		

<b>Reporte de reunión</b>					
Fecha de la reunión:		09 de octubre 2015		Duración:	3 horas
Tipo de reunión:		Ciclo:	5	Módulo:	Consultas
Participante	Asistencia	Causa de la falta			
1. José Juan Hernández Mora	Presente				
2. Astrid Ariadna Torres Fernández	Ausente	Cuestiones laborales.			
3 Lorena Zamora Velázquez	Presente				
4. Rocío Hernández Acoltzi	Presente				
5. Lidia Cortés Ruiz	Presente				
6. René López Caballero	Ausente	Razones de salud.			
7. Said Pérez Flores	Presente				
Puntos del itinerario					
No.	Especificación	Abordado	Causa	Conclusiones	
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.	
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.	
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.	
4	Obtener requerimientos.	Si		Requerimiento del cliente.	
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.	
Se cumplió el objetivo de la reunión:			En un 94 %		
Causa:			Ausencia de dos integrantes.		

<b>Reporte de reunión</b>					
Fecha de la reunión:		06 de noviembre de 2015		Duración:	45 minutos.
Tipo de reunión:		Ciclo:	6	Módulo:	Reportes
Participante	Asistencia	Causa de la falta			
1. José Juan Hernández Mora	Presente				
2. Astrid Ariadna Torres Fernández	Presente				
3 Lorena Zamora Velázquez	Presente				
4. Rocío Hernández Acoltzi	Presente				
5. Lidia Cortés Ruiz	Ausente	Razones de salud.			
6. René López Caballero	Presente				
7. Said Pérez Flores	Ausente	Cuestiones académicas.			
Puntos del itinerario					
No.	Especificación	Abordado	Causa	Conclusiones	
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.	
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.	
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.	
4	Obtener requerimientos.	Si		Requerimiento del cliente.	
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.	
Se cumplió el objetivo de la reunión:			En un 94%		
Causa:			Ausentismo en la reunión.		

<b>Reporte de reunión</b>				
Fecha de la reunión:		19 de noviembre de 2015		Duración: 2 horas 10 min.
Tipo de reunión:		Ciclo: 7	Módulo:	Exportar e importar base de datos
Participante	Asistencia	Causa de la falta		
1. José Juan Hernández Mora	Presente			
2. Astrid Ariadna Torres Fernández	Presente			
3 Lorena Zamora Velázquez	Presente			
4. Rocío Hernández Acoltzi	Presente			
5. Lidia Cortés Ruiz	Presente			
6. René López Caballero	Presente			
7. Said Pérez Flores	Presente			
Puntos del itinerario				
No.	Especificación	Abordado	Causa	Conclusiones
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.
4	Obtener requerimientos.	Si		Requerimiento del cliente.
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.
Se cumplió el objetivo de la reunión:			Al 100%	
Causa:				

<b>Reporte de reunión</b>				
Fecha de la reunión:		30 de noviembre de 2015		Duración: 38 minutos
Tipo de reunión:		Ciclo: 8	Módulo:	Sección de ayuda
Participante	Asistencia	Causa de la falta		
1. José Juan Hernández Mora	Presente			
2. Astrid Ariadna Torres Fernández	Presente			
3 Lorena Zamora Velázquez	Presente			
4. Rocío Hernández Acoltzi	Presente			
5. Lidia Cortés Ruiz	Presente			
6. René López Caballero	Presente			
7. Said Pérez Flores	Presente			
Puntos del itinerario				
No.	Especificación	Abordado	Causa	Conclusiones
1	Definir las tareas a realizar y las responsabilidades para el nuevo ciclo.	Si		Se llenó la tabla del plan de actividades.
2	Determinar el número de días requeridos para concluir el módulo.	Si		Tiempo establecido para concluir el módulo.
3	Medir el alcance que tendrá el módulo.	Si		El porcentaje que representa.
4	Obtener requerimientos.	Si		Requerimiento del cliente.
5	Presentar el plan de comunicación actualizado	Si		Elementos de organización.
Se cumplió el objetivo de la reunión:			Al 100%	
Causa:				

<i>Formato de sugerencias y correcciones</i>	
<i>Fecha:</i>	08 de octubre de 2015
<i>Ciclo:</i>	04
<i>Módulo:</i>	Gestión de materiales
<i>Tipo de sugerencia o cambio: 01 Comunicación, 02 Proceso de desarrollo, 03 Otro</i>	02
En caso de ser del tipo 03 especificar	
Propuesta de sugerencia o cambio	
Cambiar los colores del calendario, correspondiente al campo de fecha de publicación, del formulario para agregar nueva tesis.	
Razón de la sugerencia o cambio	
Los colores son oscuros y los valores no son visibles, debido a que el texto tiene casi el mismo color.	
Elementos involucrados	
La parte de diseño (la interfaz correspondiente a gestión de materiales: agregar nueva tesis)	

<i>Formato de sugerencias y correcciones</i>	
<i>Fecha:</i>	04 de noviembre de 2015
<i>Ciclo:</i>	02
<i>Módulo:</i>	Acceso al sistema
<i>Tipo de sugerencia o cambio: 01 Comunicación, 02 Proceso de desarrollo, 03 Otro</i>	02
En caso de ser del tipo 03 especificar	
Propuesta de sugerencia o cambio	
El encabezado debe ser por cada una de las instituciones.	
Razón de la sugerencia o cambio	
Se había hecho la propuesta y diseño de un encabezado para las dos instituciones, pero debía ser un encabezado por institución.	
Elementos involucrados	
La parte de análisis y diseño	