



## **Facultad 2**

# **Sistema informático para la gestión de las finanzas del Sindicato de la Educación, la Ciencia y el Deporte**

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas**

**Autor:** Raidel Alejandro Villafaña Fernández.

### **Tutores:**

MSc. Madelis Pérez Gil.

Ing. Roberto Soriano Sifontes.

**La Habana, 2017.**

**“Año 59 de la Revolución.”**

## **Declaración de autoría**

Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Raidel Alejandro Villafaña Fernández.

Firma Autor.

---

Madelis Perez Gil.

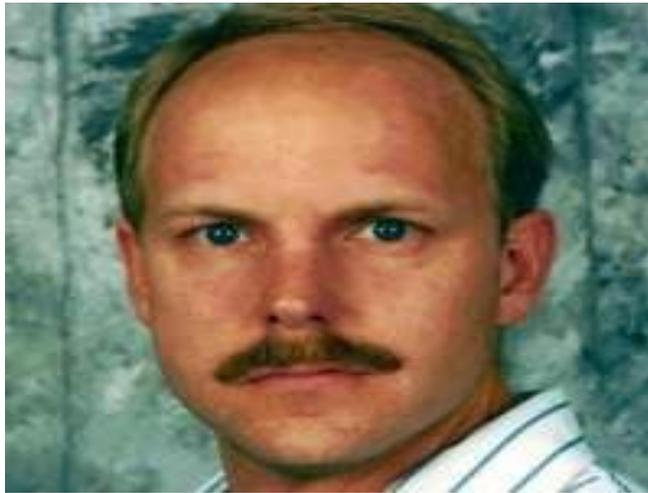
Firma Tutor.

---

Roberto Soriano Sifontes.

Firma Tutor.

Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar



Kent Beck

## **Dedicatoria**

**Dedico esta tesis a las dos personas más importantes de mi vida:**

A mi madre Maricelis Fernández Soriano: Por haberme apoyado en todo momento, por sus consejos, valores, motivación constante que me permitieron ser una persona de bien, pero más que nada, por su amor.

A mi padre Raidel Villafaña Ruiz: Por los ejemplos de perseverancia y constancia que lo caracterizan y que siempre me enseñó, por el valor mostrado para salir adelante y por su amor.

**Todo este trabajo ha sido posible gracias a ellos.**

## **Agradecimientos**

**A Dios:** Porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar.

**A mis padres Maricelis y Raidel:** Quienes a lo largo de toda mi vida han apoyado y motivado mi formación académica, creyeron en mí en todo momento y no dudaron de mis habilidades.

**A la Universidad de las Ciencias Informáticas:** Por abrir sus puertas a jóvenes como nosotros, preparándonos para un futuro competitivo y formándonos como personas de bien.

**A la Rectora Mirian Nicado García:** Primeramente, por sus consejos y aparejado a eso por su acto de humanidad ofreciéndome la oportunidad de reingresar a la universidad en tercer año, cuando pensé que todo estaba perdido.

Este proyecto es el resultado del esfuerzo conjunto de todos los que formamos el grupo de trabajo. Por eso quiero agradecer a mi tutora **Madelis Pérez Gil** por brindarme todo su apoyo y entera confianza cuando aún tenía asignaturas pendientes. A mi tutor **Roberto Soriano Sifontes** por su exigencia, responsabilidad, sabiduría y más que eso por confiar en mí y asignarme el tema de tesis sin apenas conocerme.

A mi oponente Andi por estar dispuesto en todo momento y enseñarme varias cosas de las cuales no tenía conocimiento en cuanto a la aplicación, a Yoiler que independientemente que siempre le gané en el billar, con respecto a este trabajo me brindó muchos consejos y recomendaciones, a Zoila y Lianne por sus críticas constructivas en la predefensa que me sirvieron de mucho, no solo para lograr el objetivo de este trabajo, sino para continuar en la vida como profesional.

A todos los profesores que me impartieron clases y a los que de una u otra forma contribuyeron en mi aprendizaje, a ellos les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza.

A los amigos que construí en el transcurso de la Universidad: Mi amiguis **Laura** que más que mi amiguis es mi hermanita, agradezco todo su cariño. A **Betsy** por su amabilidad prestándome el laboratorio para yo poder adelantar la tesis. Al piquete de las fiestas y también de estudio **Cuber, Ramón, Luki, Elio, Oxford, Chala, Manuel, Pedro, Ángela, Elizabet, Dianelis, Ana Laura, Anamelis**, entre otros, ellos saben quién son. En fin, a todas aquellas personas que de una forma u otra estuvieron en los momentos buenos y malos.

Por último y no menos importante, agradezco a mi novia **Yanixa, Mi Lunita**, por su paciencia y comprensión, por preferir sacrificar su tiempo para que yo pudiera cumplir con el mío. Por su bondad y sacrificio me inspiró a ser mejor para ella, ahora puedo decir que esta tesis lleva mucho de ella, gracias por estar siempre a mi lado.

## **Resumen**

La informatización de procesos manuales o semi-informatizados es una de las tareas que más ha evolucionado el entorno informático durante los últimos años. Para llevar a cabo dicha informatización en Cuba, se le ha asignado la importante tarea a la Universidad de las Ciencias Informáticas (UCI) de producir aplicaciones y servicios informáticos. En la UCI existen secciones sindicales que se encargan de hacer una agrupación de los trabajadores cubanos de la educación, la ciencia y el deporte que por su propia voluntad desean formar parte de la misma, y estas a su vez están dirigidas por secretarios que llevan el control del pago de las cuotas de los afiliados. Actualmente se desarrollan diversos procesos de forma manual, tornándose un poco engorroso y lento, limitando de esta forma la agilidad con que se realizan los mismos. Por lo que se hace necesario facilitar el proceso de gestión de información.

El objetivo de la investigación es desarrollar un sistema que mejora la gestión de la información del Sindicato. Para la confección del mismo se utilizó el lenguaje Java, mediante la herramienta IntelliJ IDEA y se tuvieron en cuenta las fases que ofrece la metodología XP. Al culminar la investigación se obtuvo como resultado un sistema informático para la gestión de las finanzas de la organización del Sindicato, validando la calidad de la solución propuesta con resultados positivos, así como la correspondencia de esta con los requerimientos del cliente.

**Palabras claves:** gestión, información, sindicato, sistema.

## Índice General

<b>Introducción</b> .....	14
<b>Capítulo 1. Fundamentación teórica</b> .....	18
1.1. Introducción.....	18
1.2. Conceptos Asociados.....	18
1.2.1. Gestión de Información.....	18
1.2.2. Sistema de Gestión de Información.....	18
1.2.3. Sistema Informático.....	19
1.2.4. Sistema Informático de Gestión de Información.....	19
1.2.5. Sindicato Nacional de Trabajadores de la Educación, la Ciencia y el Deporte.....	19
1.3. Estudio del Estado del Arte.....	20
1.3.1. Internacional.....	21
1.3.2. Nacional.....	22
1.4. Conclusiones del estudio del estado del arte.....	23
1.5. Metodologías de Desarrollo de Software.....	23
1.5.1. Programación Extrema XP.....	24
1.6. Herramientas y lenguajes informáticos.....	24
1.6.1. Lenguaje de Marcado de Hipertexto (HTML) 5.....	25
1.6.2. Hojas de estilo en cascada (CSS) 3.....	25
1.6.3. JavaScript 1.6.....	26
1.6.4. Groovy 2.0.7.....	26
1.6.5. Servidor Web.....	27
1.6.6. Sistema Gestor de Base de Datos.....	27
1.6.7. PgAdmin III 1.16.1 (18).....	28

1.6.8.	Herramientas CASE.....	28
1.6.9.	Framework.....	29
1.6.10.	Entorno de Desarrollo Integrado (IDE).....	30
1.7.	Conclusiones del Capítulo.....	30
<b>Capítulo 2. Exploración, Planificación y Diseño del Sistema.....</b>		<b>31</b>
2.1.	Introducción.....	31
2.2.	Exploración.....	31
2.2.1.	Historias de Usuario.....	31
2.3.	Planificación.....	37
2.3.1.	Estimación de Esfuerzo.....	37
2.3.2.	Plan de Iteraciones.....	39
2.3.3.	Plan de Entrega.....	42
2.4.	Diseño.....	42
2.4.1.	Patrones de Arquitectura.....	42
2.4.2.	Patrones de Diseño.....	45
2.4.3.	Modelo de Datos.....	46
2.4.4.	Tarjetas Clase Responsabilidad Colaboración (CRC).....	47
2.5.	Conclusiones del capítulo.....	51
<b>Capítulo 3. Implementación y Pruebas.....</b>		<b>52</b>
3.1.	Introducción.....	52
3.2.	Implementación.....	52
3.2.1.	Tareas de ingeniería.....	52
3.3.	Pruebas del Software.....	58
3.3.1.	Pruebas Unitarias.....	59
3.3.2.	Pruebas de Aceptación.....	60
3.4.	Conclusiones del capítulo.....	67

<b>Conclusiones</b> .....	68
<b>Recomendaciones</b> .....	69
<b>Acrónimos</b> .....	70
<b>Referencias bibliográficas</b> .....	71
<b>Bibliografía</b> .....	74

## Índice de figuras

Figura 1. Logotipo del Software Milenium.....	23
Figura 2. Logotipo del Software SoftlandERP.....	24
Figura 3. Logotipo del Software Sofia.....	24
Figura 4. Diagrama del Modelo Vista Controlador.....	55
Figura 5. Diagrama Entidad Relación.....	58
Figura 6. Resultado de la ejecución de las pruebas unitarias.....	72

## Índice de tablas

Tabla 1. Historia de Usuario # 1.....	30
Tabla 2. Historia de Usuario # 2.....	31
Tabla 3. Historia de Usuario # 3.....	32
Tabla 4. Historia de Usuario # 4.....	33
Tabla 5. Historia de Usuario # 5.....	34
Tabla 14. Estimación de esfuerzo.....	35
Tabla 15. Plan de duración de las iteraciones.....	38
Tabla 16. Plan de Entregas.....	40
Tabla 17. Tarjeta CRC # 1.....	45
Tabla 18. Tarjeta CRC # 2.....	46
Tabla 19. Tarjeta CRC # 3.....	46
Tabla 20. Tarjeta CRC # 4.....	46
Tabla 21. Tarjeta CRC # 5.....	47
Tabla 22. Tarjeta CRC # 6.....	47
Tabla 23. Tarjeta CRC # 7.....	47
Tabla 24. Tarjeta CRC # 8.....	48
Tabla 25. Tarjeta CRC # 9.....	48
Tabla 26. Tarjeta CRC # 10.....	48
Tabla 27. Tarea de Ingeniería # 1.....	51
Tabla 28. Tarea de Ingeniería # 2.....	51
Tabla 29. Tarea de Ingeniería # 3.....	52
Tabla 30. Tarea de Ingeniería # 4.....	52
Tabla 31. Tarea de Ingeniería # 5.....	53
Tabla 32. Tarea de Ingeniería # 6.....	53
Tabla 33. Tarea de Ingeniería # 7.....	53
Tabla 34. Tarea de Ingeniería # 8.....	54
Tabla 35. Tarea de Ingeniería # 9.....	54

Tabla 36. Tarea de Ingeniería # 10.....	55
Tabla 37. Tarea de Ingeniería # 11.....	55
Tabla 38. Tarea de Ingeniería # 12.....	55
Tabla 39. Tarea de Ingeniería # 13.....	56
Tabla 40. Prueba de aceptación # 1.....	58
Tabla 41. Prueba de aceptación # 2.....	59
Tabla 42. Prueba de aceptación # 3.....	59
Tabla 43. Prueba de aceptación # 4.....	59
Tabla 44. Prueba de aceptación # 5.....	60
Tabla 45. Prueba de aceptación # 6.....	60
Tabla 46. Prueba de aceptación # 7.....	61
Tabla 47. Prueba de aceptación # 8.....	61
Tabla 48. Prueba de aceptación # 9.....	62
Tabla 49. Prueba de aceptación # 10.....	62
Tabla 50. Prueba de aceptación # 11.....	62
Tabla 51. Prueba de aceptación # 12.....	63
Tabla 52. Prueba de aceptación # 13.....	63

## Introducción

La revolución informática, a pesar de haber iniciado hace más de cuarenta años, se ha intensificado en las últimas décadas mediante la creación de nuevas tecnologías de la información y las comunicaciones. Esto trae consigo que en el mundo actual muchos países se ven inmersos en un proceso de informatización. Cuba, aunque no es un país desarrollado y a pesar de los obstáculos afrontados debido a la imposición del bloqueo económico, ha obtenido resultados satisfactorios en el campo de la informática. En este aspecto se destaca la creación de centros de enseñanza que tengan como base el estudio de esta ciencia, desempeñando un papel protagónico y decisivo.

La Universidad de las Ciencias Informáticas (UCI) es una unidad docente donde se combina el estudio, la producción y la investigación. Su misión es: *“Formar profesionales comprometidos con la Patria y altamente calificados en la rama de la Informática, además de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación”*. (37)

La UCI cuenta con un conjunto de áreas vinculadas a las actividades docentes, productivas, investigativas y de servicios. La cantidad de trabajadores supera los 2 000 distribuidos en diferentes áreas, cada una con funciones específicas para garantizar el éxito de la misión y objeto social. Estos trabajadores como todos los trabajadores del país se encuentran asociados a la organización que los representa, el Sindicato nacional de los trabajadores de la Educación, la Ciencia y el Deporte (SNTECD). EL SNTECD es el encargado de hacer una agrupación de los trabajadores cubanos de la educación, la ciencia y el deporte que por su propia voluntad desean formar parte de la organización. Actualmente existen en la universidad, diversos procesos que realiza esta organización que se desarrollan de manera manual, tornándose un poco engorroso y lento, limitando de esta forma la agilidad con que se realizan los mismos. A continuación, se detallan algunos de ellos:

- Conformar secciones sindicales (hoy la UCI cuenta con 80 secciones sindicales).
- Cambiar escala de la cuota sindical que debe pagar cada uno de los afiliados.
- Depósito realizado por cada uno de los secretarios de las secciones sindicales.
- Controlar el registro de un trabajador a un área.
- Controlar la afiliación de un trabajador a una sección sindical.
- Llevar el control de aspectos tales como: la Sindicalización y el Aporte a la Patria.

- Conocer los trabajadores que fueron dados de alta y de baja.
- Llevar el control del pago de las cuotas correspondientes a la CTC y MTT.
- Llevar el control del atraso en las finanzas y en el Aporte a la Patria.
- Llevar el control de estado de las finanzas y del Aporte a la Patria.
- Registrar el 15 % de las finanzas del sindicato (Es el registro del dinero que se utiliza para hacer compras de cosas que hacen falta para la misma organización).

Teniendo en cuenta lo anterior expuesto se plantea el siguiente **problema a resolver**: las limitaciones existentes en el proceso de gestión de la información de la organización del Sindicato.

Como **objeto de estudio** se tiene: la gestión de la información relacionada con la organización del Sindicato.

Siendo el **objetivo general** de la presente investigación: desarrollar un sistema informático que permita gestionar la información de la organización del Sindicato.

Para dar cumplimiento al objetivo general se desglosaron los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Definir las funcionalidades que debe cumplir el sistema.
- Implementar un sistema informático para la gestión de información que cumpla con los requerimientos definidos.
- Aplicar pruebas al sistema resultante para garantizar la calidad del producto.

Enmarcado en el **campo de acción**: sistema Informático para la gestión de las finanzas del Sindicato.

Con el propósito de dar cumplimiento a lo anteriormente planteado, se elaboran las siguientes **tareas de investigación**:

- Revisar la bibliográfica para generar el marco teórico conceptual en lo referente al desarrollo de los Sistemas de Gestión de Información (SGI).
- Estudiar sistemas homólogos para conocer aspectos regulares en el diseño de los SGI aplicados al proceso de la organización del Sindicato.
- Analizar las herramientas informáticas y metodologías de desarrollo de software existentes para realizar la implementación del sistema.
- Identificar las principales funcionalidades del sistema para la posterior implementación del mismo, teniendo en cuenta los requisitos definidos anteriormente.

- Investigar los diferentes tipos de pruebas de software, para su posterior aplicación sobre el sistema desarrollado.

### **Métodos Científicos de la Investigación**

Los métodos científicos de la investigación son la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados. (1)

### **Métodos de Investigación Teóricos**

Permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crea las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis. (1)

### **Analítico-Sintético (1)**

Análisis: permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio.

Síntesis: establece mentalmente la unión entre las partes previamente analizadas, permite descubrir sus características generales y las relaciones esenciales entre ellas

Este método se aplica para analizar la información obtenida en la organización del Sindicato, esto posibilita identificar aspectos que puedan ser empleados en el desarrollo de la presente investigación. Además, se aprovechó para el análisis de la información existente dentro y fuera del país, acerca de los Sistemas Informáticos (SI) para la gestión de las finanzas del Sindicato de la Educación, la Ciencia y el Deporte. Identificando así, conceptos, definiciones, avances y otros elementos concluyentes de utilidad para la investigación en cuestión.

### **Métodos Empíricos (1)**

Describen y explican las características fenomenológicas del objeto y representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

### **Entrevista (1)**

Es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales

del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que se hace necesaria una buena comunicación.

Método aplicado al secretario de la organización del Sindicato de la Facultad 2 para constatar las condiciones con que cuentan para la gestión de la información del Sindicato. Además, también aplicado a los directivos del Sindicato de la UCI para conocer cómo se gestiona y maneja la información referente a las finanzas del Sindicato.

## **Capítulo 1. Fundamentación teórica**

### **1.1. Introducción**

La generación de tecnologías y herramientas informáticas se transforman constantemente, lo cual implica que los seres humanos actualicen su formación para aplicarlas en los diversos ámbitos en los que se desenvuelven. Las necesidades que se van generando para la formación de las personas en el manejo e interacción de las tecnologías y herramientas informáticas son determinadas e impuestas por las mismas. El presente capítulo está enfocado a brindar una breve descripción de todo lo relacionado con los conceptos asociados a la Gestión de Información (GI) del Sindicato. Se expondrá además un estudio del estado del arte, recorriendo el ámbito internacional y nacional. Finalmente se realiza un estudio para hacer una selección de la metodología de desarrollo de software, las herramientas y lenguaje informático para desarrollar el sistema.

### **1.2. Conceptos Asociados**

#### **1.2.1. Gestión de Información**

La GI es el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta, útil y que esté disponible en el momento que se le necesite. (2)

La finalidad de la GI es ofrecer mecanismos que permitan a la organización adquirir, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, exactitud y actualidad suficiente para servir a los objetivos de la organización. En términos perfectamente entendibles sería conseguir la información adecuada, para la persona que lo necesita, al mejor precio posible para tomar la mejor de las decisiones. (3)

#### **1.2.2. Sistema de Gestión de Información**

Un SGI es un conjunto de sistemas y procedimientos que recopilan información de una variedad de fuentes, la compilan y la presentan en un formato legible. Los SGI se utilizan para crear informes que

proporcionen una visión completa de toda la información que se necesita para tomar decisiones que van desde pequeños detalles diarios hasta una estrategia de nivel superior. (4)

### **1.2.3. Sistema Informático**

Un SI es un conjunto de partes o recursos formados por el hardware, software y las personas que lo emplean, que se relacionan entre sí para almacenar y procesar información con objetivo común. (5)

### **1.2.4. Sistema Informático de Gestión de Información**

Los Sistemas de Informáticos de Gestión de Información (SIGI) son extensamente utilizados debido a que mejora la efectividad en cualquier proceso, reducen costos y tiempos, aumentan la satisfacción de clientes y partes interesadas, logran mejoras continuas y pueden adaptarse a los cambios. Por otro lado, brindan información útil, oportuna, segura e inmediata al solicitante, con el objetivo de satisfacer sus necesidades de forma autónoma y flexible. Están hechos para agilizar el trabajo de las personas que interactúan con ellos, siendo de gran ayuda tanto para el trabajador como para el que solicita la información. (6)

### **1.2.5. Sindicato Nacional de Trabajadores de la Educación, la Ciencia y el Deporte**

El SNTECD es el encargado de agrupar a los trabajadores cubanos de la Educación, la Ciencia y el Deporte que por su propia voluntad desean ingresar en esta organización. Posee un Comité Nacional que es el máximo órgano de dirección, el cual se elige por el voto directo y secreto en cada congreso, éste se encuentra a nivel provincial y municipal. Este comité posee un secretario. En los niveles de dirección del SNTECD de provincia y municipio, también existe un comité y es elegido de igual manera que en el nivel nacional, así como un secretario. (7)

#### **Tareas de la organización (7)**

- Resuelve la contradicción entre masividad y calidad, consiste en coadyuvar a la elevación de la preparación y del nivel de profesionalidad de los docentes, así como poner los mecanismos de trabajo sindical en función del perfeccionamiento del sistema educacional, del fortalecimiento de la escuela como institución esencial y de la integración de diversos factores como son la familia, la comunidad, los medios de difusión masiva y los organismos, entre otros, con las instituciones

educacionales en la labor de formación de las jóvenes generaciones y de elevación del nivel cultural de la población.

- Garantiza la eficiencia de los mecanismos mediante los cuales los trabajadores materializan su participación en la toma de decisiones sobre la proyección, organización y control de la marcha del Proceso Enseñanza Aprendizaje (PEA), los aspectos de la legislación laboral y la creación de condiciones laborales cada vez más adecuadas, de acuerdo con las posibilidades del país.
- Genera motivaciones diversas, propicia un clima de satisfacción, de bienestar, de compromiso, y de acentuar valores éticos, morales, estéticos y convicciones político-ideológicas en cada trabajador y en cada colectivo que estimulen e incentiven los esfuerzos de los educadores por el perfeccionamiento de una educación que, por primera vez en la historia, es patrimonio de todo el pueblo y responde a los intereses de la mayoría de los integrantes de la sociedad.

**Misión:** representar a los trabajadores y movilizarlos alrededor de las tareas vinculadas al impulso de los programas de la Revolución, generando espacios de reflexión, debate, creación y aporte, así como de permanente formación y reforzamiento de valores, en una membresía con posición de vanguardia en la lucha por preservar las conquistas y garantizar la continuidad histórica del socialismo. (7)

**Visión:** ser un Sindicato de vanguardia en el movimiento obrero cubano, cumplidor destacado de las tareas fundamentales, reconocido y admirado en el sector y la sociedad en general, así como con un profundo arraigo en la membresía. (7)

### **1.3. Estudio del Estado del Arte**

Una de las primeras etapas que debe desarrollarse dentro de una investigación es el estudio del estado del arte ya que permite determinar la forma como ha sido tratado el tema, cómo se encuentra el avance del conocimiento en el momento de realizar una investigación y cuáles son las tendencias existentes para el desarrollo de la temática o problemática que se va a llevar a cabo. El estado del arte sirve al investigador como referencia para asumir una postura crítica frente a lo que se ha hecho y lo que falta por hacer en torno a una temática o problemática concreta, para evitar duplicar esfuerzos o repetir lo que ya se ha dicho, y además para localizar errores que ya fueron superados.

### 1.3.1. Internacional

Luego de realizar un estudio de sistemas que poseen características similares a nivel internacional se encontraron los siguientes:

#### **Milenium (8)**

Milenium es un sistema informático capaz de optimizar los procesos empresariales, pues permite:

- El registro, administración, sistematización y consulta de información.
- Generación automática de informes, extractos y consulta.
- Contabilizaciones automáticas.
- Generación de documentos automáticos para cierres contables.
- Depreciación de activos.
- Generación de documentos automáticos para cierres contables.
- Elaboración de planes de pagos según las necesidades de la empresa permitiendo así controlar y programar los pagos sin necesidad de otras herramientas.



Figura 1. Logotipo del Software Milenium

#### **Softland ERP (8)**

Es un sistema que incorpora en su diseño las mejores prácticas de negocio, las cuales ayudan a las empresas a ser más productivas en sus operaciones, al conseguir mayores niveles de eficiencia para competir exitosamente en un mercado cada vez más globalizado.

Opcionalmente se puede adquirir: caja menor, presupuesto financiero, flujo de caja y control presupuestal.

Entre sus principales funciones se tienen:

- Integración.
- Multimoneda.

- Control real en dólares.
- Manejo de unidades estratégicas de negocio o centros de costos.
- Flexibilidad en el análisis.
- Multicompañía.
- Seguridad.



Figura 2. Logotipo del Software SoftlandERP

### **Sofia (8)**

Sistema informático, adaptable a las necesidades específicas de organizaciones públicas o privadas permitiendo optimizar los procesos brindando información en línea. Posee generación automática de transacciones, informes estadísticos e indicadores de gestión.

Permite la definición de perfiles de usuario para restringir el acceso a las diferentes opciones del sistema y el acceso a múltiples usuarios simultáneamente.

Características:

- Versiones en arquitectura web y en arquitectura cliente/servidor.
- Motores de datos de bases relacionales (SYBASE, SQL SERVER, ORACLE).
- Cliente-servidor y web: Lenguaje power builder arquitectura J2EE.



Figura 3. Logotipo del Software Sofia

### **1.3.2. Nacional**

Luego de realizar un estudio de sistemas que poseen características similares a nivel nacional se encontró que no existen antecedentes de estos. Además, en los análisis realizados en la primera Conferencia

Nacional del SNTECD en el año 2015, uno de los temas abordados consistió en realizar el control de la afiliación de manera informatizada o semi-informatizada. Aparejado a esto se han efectuado intercambios con especialistas del Centro de Control de Afiliación del municipio La Lisa al cual pertenece la UCI, estas personas han dado fe de que no existe ningún SI para mejorar el proceso engorroso que se realiza en Sindicato.

#### **1.4. Conclusiones del estudio del estado del arte**

A partir del estudio realizado a los sistemas anteriores se evidenció que se relacionan con actividades de gestión de finanzas, tales como: registro y consulta de información, generación de informes y poseen versiones en arquitectura web y cliente servidor. Se puede concluir que sirven como base para la realización del nuevo sistema informático, pues aportan conocimientos acerca de las funcionalidades a desarrollar, pero se determina no utilizar ninguno de ellos para apoyar la investigación. Los mismos trabajan de manera eficiente y realizan las acciones para lo que fueron creados, pero no cubren en su totalidad con las necesidades específicas detectadas en el Sindicato. Por tanto, no dan solución al problema planteado en la presente investigación. A raíz de lo estipulado anteriormente se dispone implementar un SGI que cubra las necesidades detectadas en el Sindicato, y capaz de satisfacer los requerimientos del negocio exigidos por el cliente.

#### **1.5. Metodologías de Desarrollo de Software**

Para el desarrollo de software, se requiere de diversos elementos que agrupados hacen que el mismo sea o no exitoso. Para esto los desarrolladores se guían por las metodologías de desarrollo de software. Estas se clasifican en tradicionales o ágiles. Las metodologías tradicionales centran su atención en llevar una documentación exhaustiva de todo el proyecto para cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto. Tiene altos costos al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil.

Por otra parte, las ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas; este enfoque muestra su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. Teniendo en cuenta las características de esta investigación donde: es un

proyecto pequeño de poca duración, se tiene un equipo de desarrollo pequeño (1 persona) y el cliente es parte de la investigación, se decide entonces utilizar una metodología ágil para el desarrollo del sistema informático.

### **1.5.1. Programación Extrema XP**

La programación extrema es una metodología ágil, iterativa incremental centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y acciones para enfrentar los cambios. Esta metodología se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes. (9)

**Se selecciona la metodología ágil de desarrollo de software XP porque:**

- El equipo está compuesto por un programador.
- Los requisitos pueden cambiar durante el desarrollo del sistema informático.
- Se cuenta con poco tiempo para el desarrollo del sistema informático.
- El cliente forma parte del equipo de desarrollo, lo cual ayuda a corregir a tiempo los errores que puedan surgir.

Al estudiar las bibliografías de Kent Beck, padre de la metodología de desarrollo de software XP, se pueden definir seis fases para el uso de la misma: exploración, planificación, diseño, implementación y pruebas, mantenimiento, y muerte del proyecto. Por el alcance de esta investigación, la cantidad de integrantes del equipo y atendiendo a la necesidad respecto al tiempo en que se debería construir el sistema esta investigación solo abarca las cuatro primeras fases.

### **1.6. Herramientas y lenguajes informáticos**

Las herramientas y lenguajes informáticos empleados en el desarrollo del sistema, están en correspondencia con el cliente ya que el mismo forma parte del equipo de desarrollo y solicitó que se realizara con las tecnologías y herramientas informáticas que se mencionan posteriormente debido a que en un futuro se le proporcione soporte al sistema de la manera más factible posible.

### **1.6.1. Lenguaje de Marcado de Hipertexto (HTML) 5**

El lenguaje de marcado de hipertexto (HTML) es usado para estructurar y presentar el contenido para la web. Es un sistema que formatea el diseño de las páginas y realiza ajustes a su aspecto. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari saben cómo mostrar una determinada página web, donde están los elementos, donde poner las imágenes y donde ubicar el texto. La diferencia principal, sin embargo, es el nivel de sofisticación del código que se puede construir usando HTML5. (10)

Se decide usar HTML5 ya que éste presenta muchas mejoras en comparación a las demás versiones anteriores de HTML. Se combina con otros lenguajes para producir los sitios que se pueden ver hoy en día y se tienen otras posibilidades para explotar usando menos recursos. Con HTML5, también entra en desuso el lenguaje de marcado de hipertexto extensible (XHTML), dado que ya no sería necesaria su implementación.

### **1.6.2. Hojas de estilo en cascada (CSS) 3**

Las hojas de estilo en cascada (CSS) es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos de lenguaje de marcado extensible (XML). CSS3 se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. El estándar de CSS3 actualmente está bastante estable, los navegadores los soportan. (11)

#### **Ventajas: (12)**

- Se obtiene un mayor control de la presentación del sitio al poder tener todo el código CSS reunido en uno, lo que facilita su modificación.
- Al poder elegir el archivo CSS que se desea mostrar, puede aumentar la accesibilidad ya que se puede asignar un código CSS concreto a personas con diferencias visuales.
- Se consigue hacer mucho más legible el código HTML al tener el código CSS aparte.
- Puede mostrarse distintas hojas de estilo según el dispositivo que se esté utilizando.
- Gracias a la técnica CSS Sprites se puede aligerar la carga del sitio al juntar todas las imágenes en una.

Por tanto, se escoge CSS3 para la aplicación de estilos a los contenidos que se muestran en el sistema informático a desarrollar.

### **1.6.3. JavaScript 1.6**

Es un lenguaje de programación que se utiliza principalmente del lado del cliente permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan en código JavaScript integrado en las páginas web. La ventaja de JavaScript es que al estar alojado en la computadora del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles entre otras. (13) Se escoge JavaScript en su versión 1.6 para el desarrollo de la aplicación informática debido a los siguientes aspectos: (14)

- Ofrece un código estructurado y una larga lista de características que permiten agilizar el proceso de desarrollo desde que se empieza a trabajar con el código e incluso hasta que se realizan las pruebas del sistema informático.
- Tener a disposición una comunidad extensa que apoya JavaScript no solo permite encontrar ayuda y aprender más fácilmente, sino que también puede tener actualizaciones frecuentes y mejoras constantes, de esta manera se puede hacer uso de las últimas tecnologías al momento de realizar el sistema informático.
- Las actualizaciones también implican la corrección de errores y la mejora del funcionamiento, si posee una comunidad extensa, es probable que se notifiquen sobre los errores más rápidamente y la solución que se dé sea de forma casi inmediata.

### **1.6.4. Groovy 2.0.7**

Se escoge Groovy en su versión 2.0.7 como lenguaje de programación del lado del servidor, debido a que el framework seleccionado para el desarrollo del SI está desarrollado sobre el mismo. Además, es un lenguaje potente, opcionalmente digitado y dinámico, con capacidades de compilación estática para la plataforma Java, mejora la productividad de los desarrolladores ya que cuenta con una sintaxis concisa, familiar y fácil de aprender. Se integra perfectamente con cualquier programa Java e inmediatamente

entrega a su aplicación características de gran alcance, incluyendo capacidades de escritura, creación de lenguaje específico de dominio, tiempo de ejecución y meta-programación en tiempo de compilación y programación funcional. (38)

### **1.6.5. Servidor Web**

#### **Apache Tomcat 7.0.68 (15)**

Es un contenedor web desarrollado bajo la licencia de Apache. Fue escogido como servidor web en su versión 7.0.68, debido a que dispone de libre acceso a su código fuente y a su forma binaria. Además, se puede usar en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual de Java.

- Posee mejoras para detectar y prevenir fugas de memoria en las aplicaciones web.
- Realiza una limpieza interna del código.
- Contiene soporte para la inclusión de contenidos externos directamente en una aplicación web.

### **1.6.6. Sistema Gestor de Base de Datos**

Los Sistemas Gestores de Base de Datos (SGBD) proporcionan servicios para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en la base de datos de forma segura y eficiente. Un SGBD actúa como un intermediario entre las aplicaciones y los datos. Los mismos tienen como función esencial proporcionar independencia de los datos y las aplicaciones. Así se puede diseñar la base de datos incluso antes que las aplicaciones, ese diseño permitirá que se desarrollen múltiples aplicaciones posteriormente. (16)

#### **PostgreSQL 9.4.1.1**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia de software libre y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Por lo que un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (16)

Se seleccionó PostgreSQL en su versión 9.4.1 porque:

- Es ampliamente popular e ideal para tecnologías web.

- Fácil de administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Multiplataforma.
- Capacidad de replicación de datos.
- Soporte empresarial disponible. (17)

### **1.6.7. PgAdmin III 1.16.1 (18)**

PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una decena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac, OSX y Solaris. PgAdmin III soporta versiones de servidores 7.3 y superiores.

Ha sido seleccionado PgAdmin III en su versión 1.16.1 dado que es una herramienta de código abierto para la administración de las bases de datos PostgreSQL. El mismo incluye:

- Interfaz administrativa gráfica.
- Herramienta de consulta SQL con explicaciones gráficas.
- Editor de código procedural.
- Agente de planificación SQL/Shell/batch.
- Administración de Slony-L.

### **1.6.8. Herramientas CASE**

Las herramientas de Ingeniería de Software Asistida por Computadora (*CASE*) modelan la información de negocios cuando ésta se transfiere entre distintas entidades organizativas en el seno de una compañía. El objetivo primordial de las herramientas de esta categoría consiste en representar objetos de datos de negocios, sus relaciones y ayuda a comprender mejor la forma en que fluyen estos objetos de datos entre distintas zonas de negocio. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas no satisfacen las necesidades de la organización. (19)

## Visual Paradigm 8.0. (20)

Visual Paradigm es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un software.

Se decide utilizar Visual Paradigm en su versión 8.0 para el modelado UML por las ventajas que ofrece:

- **Dibujo:** facilita el modelado UML, pues proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- **Corrección sintáctica:** controla que el modelado con UML sea correcto.
- **Coherencia entre diagramas:** al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- **Integración con otras aplicaciones:** permite integrarse con otras aplicaciones como herramientas ofimáticas, lo cual aumenta la productividad.
- **Reutilización:** facilita la reutilización, ya que dispone de una herramienta centralizada donde se encuentran los modelos utilizados.
- **Generación de código:** permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación de software.
- **Generación de informes:** permite generar diversos informes a partir de la información introducida por la herramienta.

## 1.6.9. Framework

### Grails 2.2.1

Se escoge Grails en su versión 2.2.1 porque es un framework desarrollado para aplicaciones web, creado en Java y Groovy en 2005. Pero también es una herramienta de construcción de proyectos y pruebas a la vez, que permite, además, añadir plugins desarrollados por terceros. Tiene la peculiaridad de que abarca las tres capas del desarrollo web: acceso a base de datos, capa de negocio y vista. Para conseguir todo esto en un solo producto, Grails se basa en otros frameworks conocidos de fuente abierta: principalmente Spring e Hibernate. La principal característica y que más llama la atención de Grails es que está diseñado

para que se programen las aplicaciones en Groovy, un lenguaje dinámico con la misma sintaxis que Java, pero que añade muchísimas nuevas funcionalidades, permitiendo programar más con menos código. (21)

#### **1.6.10. Entorno de Desarrollo Integrado (IDE)**

##### **IntelliJ IDEA 15.0**

IntelliJ IDEA es un IDE Java comercial desarrollado por JetBrains. Permite escritura de código sin complicaciones. Practica un abordaje no intrusivo e intuitivo para ayudar a escribir, depurar, refactorizar, probar y aprender el código. Crea un entorno adecuado en donde se puede trabajar de manera eficiente. Integración transparente con una amplia variedad de sistemas de control de versiones. Puede coexistir con otros IDEs populares, como Eclipse y herramientas de gestión de proyectos como Maven. Constantemente valida la calidad del código y ofrece soluciones inmediatas para los problemas encontrados en todos los niveles desde la instrucción individual para arquitectura global, utilizando las inspecciones de código avanzado y análisis de matriz de dependencia. Si existe un problema de codificación, problemas potenciales de rendimiento, o el incumplimiento de contrato, IntelliJ IDEA muestra una advertencia y corrige el problema, ayudando a producir de forma limpia, un código de primera línea en menos tiempo. (22)

Por tanto, considerando las características anteriormente planteadas se decide utilizar IntelliJ IDEA en su versión 15.0.

#### **1.7. Conclusiones del Capítulo**

Para llevar a cabo el desarrollo de una aplicación informática se necesita antes, definir un conjunto de aspectos para lograr el objetivo por el cual se realiza. En este capítulo se efectuó un estudio de los conceptos fundamentales que están relacionados con la investigación evidenciándose un mejor entendimiento de la misma. Se realizó además un estudio del estado del arte que arribó a la conclusión de que ninguno englobaba todas las necesidades que requiere el problema que en esta investigación se plantea. Finalmente se seleccionó la metodología de desarrollo de software, las herramientas y el lenguaje informático, que se utilizarán para desarrollar el sistema en cuestión.

## **Capítulo 2. Exploración, Planificación y Diseño del Sistema**

### **2.1. Introducción**

El desarrollo de aplicaciones es un proceso en el cual se definen artefactos que, luego de ser integrados mejoran la comunicación y el entendimiento entre los involucrados en el desarrollo. Este capítulo hace referencia a las fases de Exploración, Planificación y Diseño, propias de la metodología de desarrollo de software escogida. Además, se detallan los artefactos que propone para cada una de las fases mencionadas.

### **2.2. Exploración**

La metodología ágil de desarrollo de software XP comienza con la fase de exploración, en esta fase, los clientes plantean sus necesidades de las cuales se derivan a grandes rasgos las Historias de usuario (HU) que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas informáticas, tecnologías y prácticas que se utilizarán en el proyecto.

Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de Exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tenga el programador con la tecnología. (23)

#### **2.2.1. Historias de Usuario**

Las Historias de usuario (HU) conforman la parte central de la metodología de desarrollo ágil XP. Definen lo que se debe realizar en el proyecto de software, tienen una prioridad asociada definida por el cliente de manera que indica cuáles son las más importantes para el resultado final, las mismas serán divididas en tareas de ingeniería y su tiempo será estimado por los desarrolladores. Las HU son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las HU permiten responder rápidamente a los requisitos cambiantes. (24)

En las HU se consideran los siguientes aspectos:

- **La prioridad en el negocio:**
  - ✓ Alta: cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
  - ✓ Media: cuando el cliente estima que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
  - ✓ Baja: cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.
- **El riesgo en desarrollo:**
  - ✓ Alto: cuando en la implementación de las HU puedes surgir errores que lleven a la inoperatividad del código.
  - ✓ Medio: cuando en la implementación de las HU puedes surgir errores que retrasen la entrega del sistema informático.
  - ✓ Bajo: cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del sistema informático.
- **Un punto de estimación** equivale a una semana de programación la cual corresponde a 40 horas en desarrollo, 8 horas durante 5 días de la semana del calendario normal.

**A continuación, se describen las HU del sistema:**

Tabla 1. Historia de Usuario # 1

Historia de usuario	
<b>Número:</b> 1	<b>Nombre:</b> Crear Estructuras
<b>Usuario:</b> Todos	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo:</b> alto
<b>Puntos estimados:</b> 1.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Raidel Alejandro Villafañá Fernández.	

**Descripción:** permite crear una nueva estructura con niveles y áreas definidos por el usuario.

**Observaciones:** el usuario debe estar autenticado en el sistema.

**Prototipo de interfaz:**

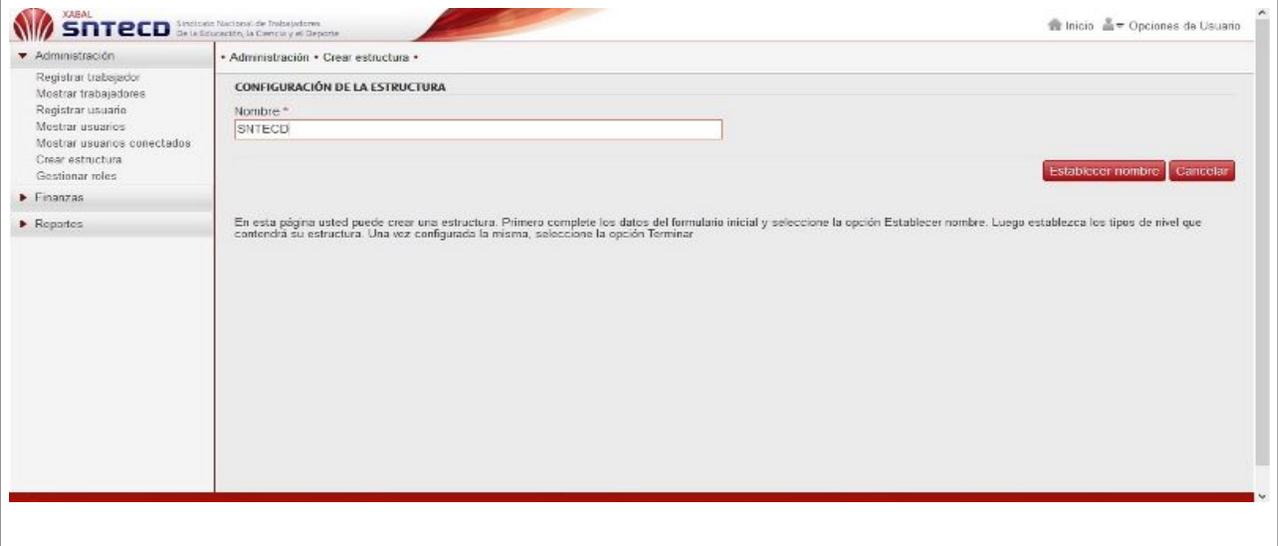


Tabla 2. Historia de Usuario # 2

Historia de usuario	
<b>Número:</b> 2	<b>Nombre:</b> Dar Baja
<b>Usuario:</b> Todos	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo:</b> alto
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> permite darle de baja a un trabajador afiliado a una sección sindical.	

**Observaciones:** debe existir al menos un trabajador afiliado a un área.

**Prototipo de interfaz:**

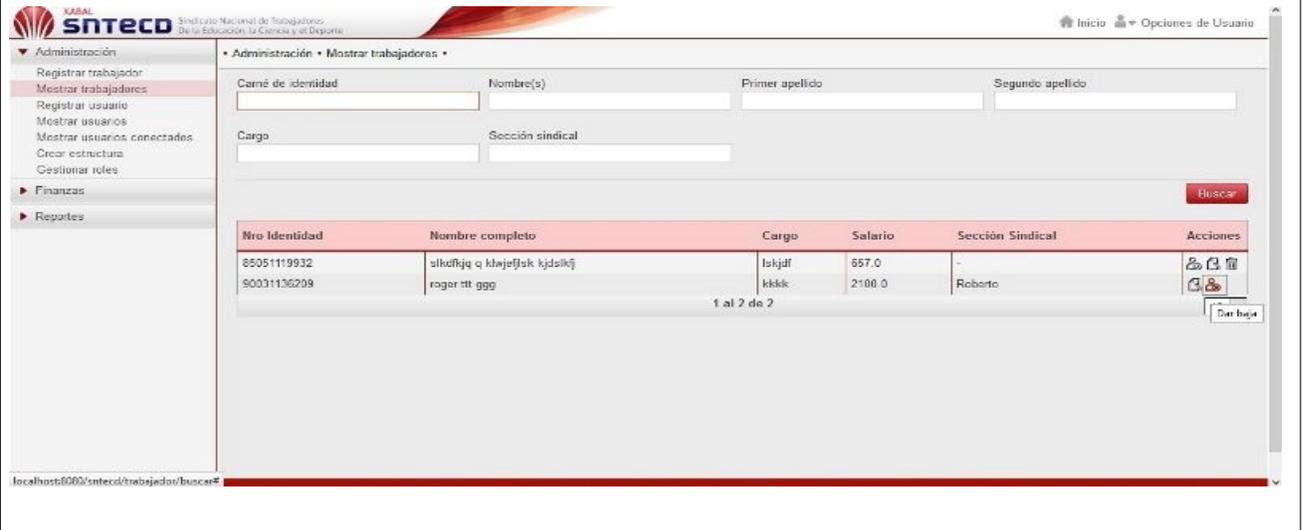


Tabla 3. Historia de Usuario # 3

Historia de usuario	
<b>Número:</b> 3	<b>Nombre:</b> Cambiar Escala
<b>Usuario:</b> Todos	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo:</b> alto
<b>Puntos estimados:</b> 1.0	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> permite cambiar la escala a una cuota sindical.	

**Observaciones:** debe existir al menos una cuota registrada en el sistema.

**Prototipo de interfaz:**



Tabla 4. Historia de Usuario # 4

Historia de usuario	
<b>Número:</b> 4	<b>Nombre:</b> Registrar Pago CTC
<b>Usuario:</b> Todos	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo:</b> medio
<b>Puntos estimados:</b> 1.0	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Raidel Alejandro Villafañá Fernández.	
<b>Descripción:</b> permite registrar el pago mensual de la cuota sindical de un trabajador perteneciente a una sección sindical.	

**Observaciones:** debe existir al menos un trabajador afiliado a un área.

**Prototipo de interfaz:**

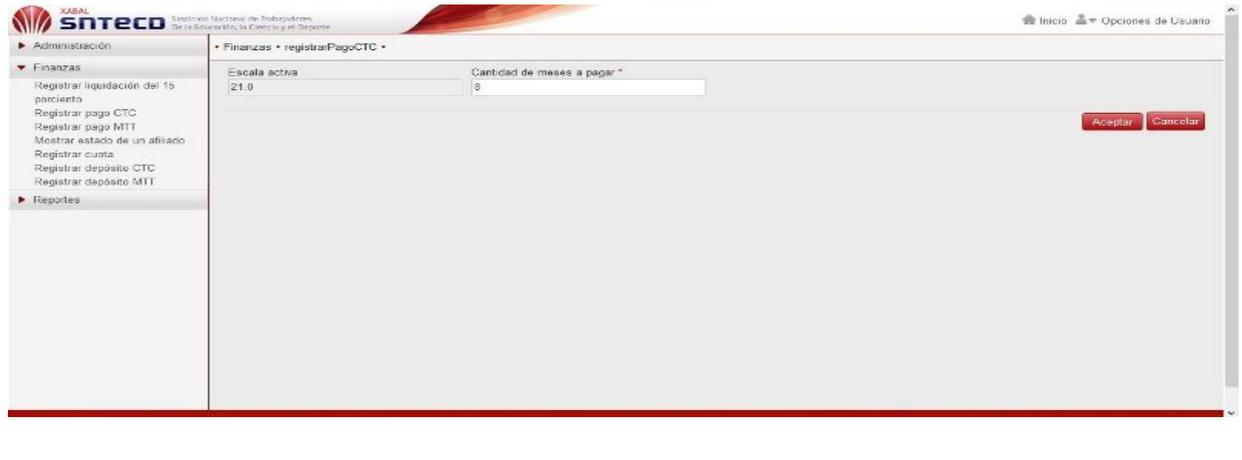
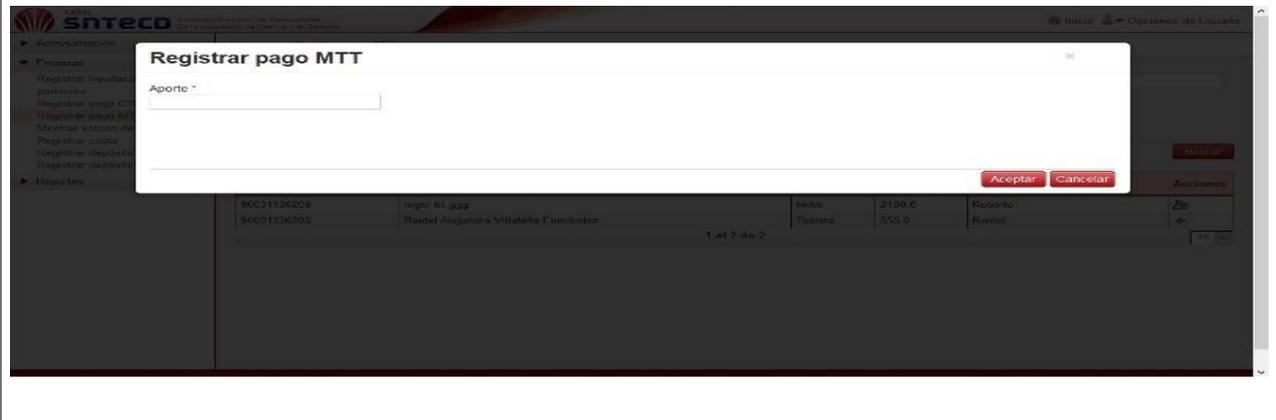


Tabla 5. Historia de Usuario # 5

Historia de usuario	
<b>Número:</b> 5	<b>Nombre:</b> Registrar Pago MTT
<b>Usuario:</b> Todos	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo:</b> medio
<b>Puntos estimados:</b> 1.0	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> permite registrar el pago anual de un trabajador perteneciente a una sección sindical.	
<b>Observaciones:</b> debe existir al menos un trabajador afiliado a un área.	

## Prototipo de interfaz:



El resto de las HU se encuentran en el Anexo B.

## 2.3. Planificación

En esta fase el cliente establece la prioridad de cada HU, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de 3 meses. Esta fase dura unos pocos días. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuantas HU se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de HU. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuantos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las HU seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación. (25)

### 2.3.1. Estimación de Esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto de estimación. Un punto de estimación equivale a una semana de programación. Las HU generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de velocidad de desarrollo, establecida en puntos por iteración, basándose

principalmente en la suma de puntos correspondientes a las HU que fueron terminadas en la última iteración.

Tabla 14. Estimación de esfuerzo

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Crear Estructuras	1.5
	2	Dar Baja	0.5
	3	Cambiar Escala	1.0
	4	Registrar pago CTC	1.0
	5	Registrar Pago MTT	1.0
	6	Registrar Depósito CTC	0.5
	7	Registrar Deposito MTT	0.5
	8	Generar Registro de Control de Sindicalización	1.0
	9	Generar Registro de Comprometido y Plan Mi Aporte a la Patria	1.0
	10	Generar Modelo de Baja	1.0
	11	Generar Modelo de Alta	1.0
	12	Generar Informe de Comprometidos y actividad económica MTT	1.0

	13	Generar Informe de Afiliación y actividad económica de Finanzas	1.0
2	14	Generar Reporte de atrasos en las Finanzas	1.5
	15	Generar Reporte de atrasos de mi Aporte a la Patria	1.5
	16	Generar Reporte de Estado de las Finanzas	1.5
	17	Generar Reporte de Estado de mi Aporte a la Patria	1.5
	18	Mostrar Estado de un Afiliado	1.5
3	19	Registrar Trabajador	1.0
	20	Afiliar Trabajador	1.0
	21	Registrar Cuota	1.5
	22	Registrar liquidación del 15%	1.5
<b>Total</b>			<b>24.5</b>

La estimación del esfuerzo es muy importante tanto para el equipo de desarrollo como para el cliente debido a que de esa manera existe una guía por donde saber qué se realiza en cada momento. A la hora de entregar cada una de las iteraciones dicha estimación de esfuerzo es de vital apoyo pues como están bien definidas las HU que se van a implementar en cada iteración y su tiempo de realización, la calidad, disciplina y cumplimiento son resultados de una buena puesta en práctica de la etapa de planificación dentro de la metodología escogida. Se concluye que las 22 HU agrupadas en 3 iteraciones poseen un tiempo de demora para su realización de 24.5 semanas.

### 2.3.2. Plan de Iteraciones

El proceso de desarrollo está compuesto por varias iteraciones. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las HU que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide que HU se implementarán en cada iteración; esto se realiza para lograr maximizar el valor del negocio. El final de la última iteración el sistema está listo para entrar en producción.

### **Iteración 1**

En dicha iteración se implementarán las HU que el cliente para su impacto en el negocio le asignó nivel de prioridad alta, prioridad que indica que esas HU están relacionadas con la primera etapa del proceso de gestión de información del Sindicato, creando con ellas el funcionamiento base del sistema. Además, se obtendrá una primera versión del producto en la que el cliente podrá probar algunas de las funcionalidades solicitadas como: Crear Estructuras, Dar Baja, Cambiar Escala, Registrar Pago CTC, Registrar Pago MTT, Registrar Depósito CTC, Registrar Depósito MTT, Generar Registro de Control de Sindicalización, Generar Registro de Comprometido y Plan Mi Aporte a la Patria, Generar Modelo de Baja, Generar Modelo de Alta, Generar Informe de Comprometidos y actividad económica MTT, Generar Informe de Afiliación y actividad económica de Finanzas detalladas en las HU 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.

### **Iteración 2**

En dicha iteración se implementarán las HU que el cliente para su impacto en el negocio le asignó nivel de prioridad media, prioridad que indica que esas HU están relacionadas con la segunda etapa del proceso de gestión de información del Sindicato, creando con ellas un funcionamiento más abarcador del sistema. Además, se obtendrá una segunda versión del producto en la que el cliente podrá probar algunas de las funcionalidades solicitadas como: Generar Reporte de atrasos en las Finanzas, Generar Reporte de atrasos de mi Aporte a la Patria, Generar Reporte de Estado de las Finanzas, Generar Reporte de Estado de mi Aporte a la Patria, Mostrar Estado de un Afiliado detalladas en las HU 14, 15, 16, 17, 18.

### **Iteración 3**

En dicha iteración se implementarán las HU que el cliente para su impacto en el negocio le asignó nivel de prioridad baja, prioridad que indica que esas HU están relacionadas con la tercera etapa del proceso de gestión de información del Sindicato, creando con ellas un funcionamiento completo y terminal del sistema.

Además, se obtendrá la tercera y última versión del producto en la que el cliente podrá probar algunas de las funcionalidades solicitadas como: Registrar Trabajador, Afiliar Trabajador, Registrar Cuota, Registrar liquidación del 15% detalladas en las HU 19, 20, 21, 22.

Tabla 15. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Crear Estructuras	3.0
	2	Dar Baja	
	3	Cambiar Escala	
1	4	Registrar pago CTC	3.0
	5	Registrar Pago MTT	
	6	Registrar Depósito CTC	
	7	Registrar Depósito MTT	
1	8	Generar Registro de Control de Sindicalización	3.0
	9	Generar Registro de Comprometido y Plan Mi Aporte a la Patria	
	10	Generar Modelo de Baja	
1	11	Generar Modelo de Alta	3.0
	12	Generar Informe de Comprometidos y actividad económica MTT	
	13	Generar Informe de Afiliación y actividad	

		económica de Finanzas	
2	14	Generar Reporte de atrasos en las Finanzas	3.0
	15	Generar Reporte de atrasos de mi Aporte a la Patria	
	16	Generar Reporte de Estado de las Finanzas	3.0
	17	Generar Reporte de Estado de mi Aporte a la Patria	
	18	Mostrar Estado de un Afiliado	1.5
3	19	Registrar Trabajador	2.0
	20	Afiliar Trabajador	
	21	Registrar Cuota	3.0
	22	Registrar liquidación del 15%	
<b>Total</b>			<b>24.5</b>

### 2.3.3. Plan de Entrega

El plan de entrega no es más que la toma de acuerdo del equipo de desarrollo con el cliente sobre el contenido que posee la primera entrega y determinar un intervalo de tiempo para las restantes entregas del producto. Mediante el mismo el equipo de desarrollo mantiene informado y actualizado al cliente de dicho proceso, funcionamiento y la calidad del sistema.

Tabla 16. Plan de Entregas

Iteración	Fecha de entrega
-----------	------------------

Iteración 1	28 de febrero de 2017
Iteración 2	25 de abril de 2017
Iteración 3	28 de mayo de 2017

## **2.4. Diseño**

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos. (25)

### **2.4.1. Patrones de Arquitectura**

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de subsistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos.

Los patrones arquitectónicos heredan mucho de la terminología y conceptos de patrones de diseño, pero se centran en proporcionar modelos y métodos reutilizables específicamente para la arquitectura general de los sistemas de información. En otras palabras, quiere decir que a diferencia de los patrones de diseño estas son plantillas incompletas y no se pueden aplicar directamente al código con modificaciones meramente contextuales. Los patrones arquitectónicos a su vez se salen del código puro de la aplicación y suben e incluyen software, hardware, redes, incluso las personas. (26)

#### **Modelo Cliente Servidor**

La modalidad Cliente-Servidor es un sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan. Es un modelo que implica productos y servicios enmarcados en el uso de la tecnología de punta y permite la distribución de la información en forma ágil y eficaz a las diversas áreas de una organización. (27)

### **Se decide utilizar la modalidad o arquitectura Cliente-Servidor porque:**

Es en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes: una requiere servicios y la otra los brinda. Pero que, a la vez, pueden realizar tanto actividades en forma conjunta como independientemente. En el caso del cliente es aquel que requiere un servicio del servidor. En esta categoría se realizan funciones de software basándose en el hardware, pero en caso de no tener la capacidad de procesar los datos necesarios recurre al servidor y espera a que éste le brinde los servicios solicitados. El cliente es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor. (28)

Entre las características fundamentales de esta arquitectura se encuentran que tanto el cliente como el servidor pueden realizar tareas en forma conjunta como separada ya que el cliente tiene sus propias aplicaciones, archivos y bases de datos y que, además, pueden estar en la misma plataforma o en plataformas diferentes. Por otra parte el servidor puede brindar varios servicios a la vez, tanto al mismo cliente como a clientes múltiples. (39)

### **Modelo Vista Controlador (29)**

Se propone utilizar el patrón de arquitectura Modelo Vista Controlador (MVC) ya que es un patrón de arquitectura de software que separa los datos, la interfaz y la lógica del sistema.

- El modelo: representa la parte de la aplicación que implementa la lógica del negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos. En el SI el modelo lo representa las clases de dominio que se encuentran ubicadas en el paquete domain.
- La vista: hace una representación de los datos del modelo estando separada de los objetos del modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de representación de cualquier petición que se presente. Por ejemplo, como la capa de modelo devuelve un conjunto de datos, la vista los usaría para hacer una página HTML que los contenga. En el SI la vista es representada por groovy server pages que ubican en el paquete views.
- El controlador: gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto el modelo como de la vista. Los controladores pueden ser vistos como administradores cuidando de que todos los recursos necesarios para completar una tarea se

deleguen a los trabajadores más adecuados. Espera peticiones de los clientes, comprueba su validez de acuerdo a las normas de autenticación o autorización, delega la búsqueda de datos al modelo y selecciona el tipo de respuesta más adecuado según las preferencias del cliente. Finalmente delega este proceso de presentación a la capa de la Vista. En el SI los controladores son representados por clases groovy cuyo nombre finaliza en controller y se ubican en el paquete controllers.

El patrón de arquitectura MVC cumple perfectamente en con el fin particular de cualquier framework de desarrollo, una estructura bien definida que ofrece soporte a un proyecto web, también ayuda a que el proyecto sea organizado y bien desarrollado. Grails trae incluido en su estructura esta arquitectura, por tanto, es el patrón planteado por Grails, framework que se usa para dar solución a esta investigación. Como se muestra en la Figura 4, separa la lógica del negocio y del controlador. Divide una aplicación interactiva en 3 componentes. El “modelo” contiene la información central y los datos. Las “vistas” despliegan información al usuario. Los “controladores” capturan la entrada del usuario.

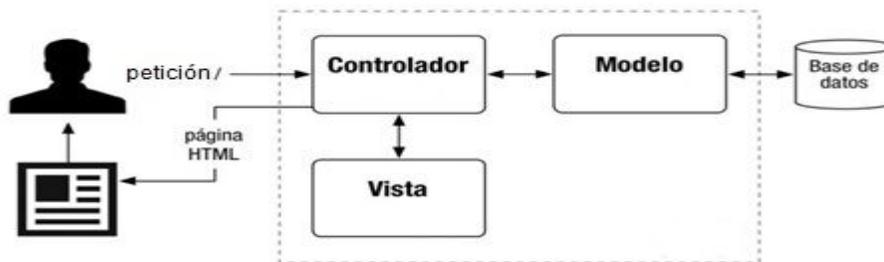


Figura 4. Diagrama del Modelo Vista Controlador  
Modificado por el Autor

## 2.4.2. Patrones de Diseño

Un patrón es una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas, a un problema recurrente dentro de un cierto contexto. El objetivo de los patrones es crear un lenguaje común para los desarrolladores con el objetivo de comunicar

experiencia sobre los problemas y soluciones. Los patrones de diseño expresan esquemas para definir estructuras de diseño con las que se construye un sistema informático.

### **Patrones Grasp**

Los patrones de Principios Generales para Asignar Responsabilidades (GRASP por sus siglas en inglés) representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. (30)

- **Experto en Información:** se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. El SI para la gestión de las finanzas del Sindicato de la Educación, la Ciencia y el Deporte hace uso de ese patrón y se evidencia cuando se desea mostrar el listado de cuotas existentes ya que la única clase con la responsabilidad de conocer esta información es CuotaService. De la misma manera sucede con todas las clases del SI.
- **Creador:** este patrón es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de la clase A. B es un creador de objetos A. Dicho patrón realiza las funciones de insertar, modificar y eliminar datos en la base de datos. Dentro del sistema este patrón se evidencia en los servicios ModeloAltaService y ModeloBajaService los cuales están asociados a cada una de las clases entidades existentes y se encargan de crear y guardar en base de datos los objetos de dichas clases entidades.
- **Alta Cohesión:** asigna una responsabilidad de forma tal que la cohesión siga siendo alta. Este patrón fue utilizado en el sistema en general al agrupar las clases en dependencia de los requisitos. Además, se garantiza en el sistema propuesto, un ejemplo de ello es en el servicio EstadoService, en la funcionalidad obtener estado. La misma necesita conocer los meses que un afiliado a pagado, la responsabilidad de brindar esta información le corresponde a la clase PagoService, evidenciándose la alta cohesión ya que la clase EstadoService no implementa este método, solo hace uso del mismo.
- **Bajo Acoplamiento:** el acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Este patrón permite realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil o que a su vez permitan la reutilización. Este patrón es utilizado

por el framework Grails, y por ende en el sistema, al no asociar las clases del modelo con las de la vista o con el controlador, la dependencia entre las clases, en este caso, se mantiene baja.

- **Controlador:** asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Se evidencia el uso de este patrón, en el SI para la gestión de las finanzas del Sindicato de la Educación, la Ciencia y el Deporte, ya que, para cada petición o evento generado en el mismo, existe un controlador que es el encargado de obtenerla y devolver una respuesta.

### 2.4.3. Modelo de Datos

El modelo de datos se utiliza para representar la base de datos del sistema, permite describir el tipo de datos y sus relaciones, las restricciones de integridad, es decir, las condiciones que deben cumplir los datos para reflejar lo real y las operaciones de manipulación de los datos, modificación, borrado y recuperación. (31)

El modelo de datos ver en la Figura 5, en esta investigación fue el punto de partida en aras de implementar un sistema de gestión de información que solucionara las necesidades específicas de la organización del Sindicato. Al elaborar el modelo de datos resultó de gran ayuda la utilización de las herramientas informáticas especificadas en el Capítulo 1, las cuales se han encargado de facilitarle el trabajo al hombre. Con el modelo de datos realizado, la lógica del negocio está representada a través de la relación entre las tablas y sus dependencias.

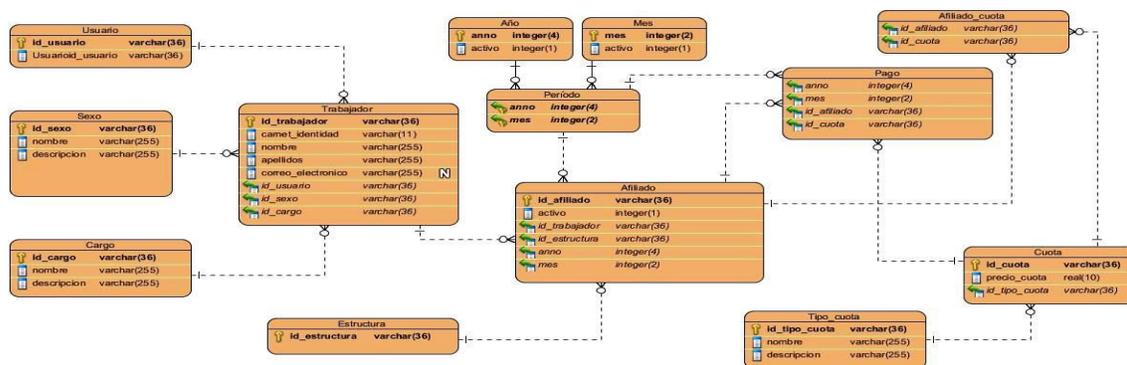


Figura 5. Diagrama Entidad Relación  
Elaboración propia.

### 2.4.4. Tarjetas Clase Responsabilidad Colaboración (CRC)

La utilización de tarjetas CRC (Class-Responsability-Colaboration) es una técnica de diseño orientado a objetos propuesta por Kent Beck padre de la metodología de programación extrema y Ward Cunningham también conocido entre otras muchas materias, por sus aportaciones a dicha metodología. El objetivo de la misma es hacer mediante tarjetas, un intervalo de las clases que se necesitarán para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y la discusión de las mismas por parte de varios actores del equipo de proyecto con el objetivo de que el diseño sea lo más simple posible verificando las especificaciones del sistema. (36)

Un esquema típico de Tarjeta CRC puede ser aquel en el que se indiquen los siguientes datos:

- Nombre de la clase.
- La responsabilidad de la clase.
- Las clases con las que va a colaborar para poder realizar las responsabilidades indicadas. (32)

Tabla 17. Tarjeta CRC # 1

Tarjeta CRC	
<b>Clase:</b> ReporteEstadoService	
Responsabilidad	Colaboración
Esta clase es la encargada de preparar los datos necesarios para realizar el reporte de estado de las finanzas.	EstadoService DNivel DTrabajador

Tabla 18. Tarjeta CRC # 2

Tarjeta CRC
<b>Clase:</b> EstadoService

<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de devolver el estado en que se encuentra un trabajador con respecto al pago de la cuota sindical.	DTrabajador PagoService

Tabla 19. Tarjeta CRC # 3

<b>Tarjeta CRC</b>	
<b>Clase:</b> PagoService	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de manejar los datos asociados a cada tipo de pago.	DTrabajador DPago DRegistroAcciones

Tabla 20. Tarjeta CRC # 4

<b>Tarjeta CRC</b>	
<b>Clase:</b> TrabajadorController	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de realizar las acciones correspondientes a todos los trabajadores registrados en el sistema.	DTrabajador TipoNivelService UsuarioSistemaService

--	--

Tabla 21. Tarjeta CRC # 5

Tarjeta CRC	
<b>Clase:</b> ModeloAltaController	
Responsabilidad	Colaboración
Esta clase es la encargada de gestionar todos los datos necesarios para generar un modelo con los trabajadores dados de alta en el año.	ModeloAltaService TipoNivelService DNivel

Tabla 22. Tarjeta CRC # 6

Tarjeta CRC	
<b>Clase:</b> ModeloBajaController	
Responsabilidad	Colaboración
Esta clase es la encargada de gestionar todos los datos necesarios para generar un modelo con los trabajadores dados de baja en el año.	ModeloBajaService TipoNivelService DNivel

Tabla 23. Tarjeta CRC # 7

Tarjeta CRC	
<b>Clase:</b> ReporteAtrasoMTTService	
Responsabilidad	Colaboración
Esta clase es la encargada de preparar los datos necesarios para realizar el reporte de los afiliados atrasados en el pago de compromiso MTT	DAportePatria DTrabajador

Tabla 24. Tarjeta CRC # 8

Tarjeta CRC	
<b>Clase:</b> ReporteAtrasoFinanzasService	
Responsabilidad	Colaboración
Esta clase es la encargada de preparar los datos necesarios para realizar el reporte de los afiliados atrasados en el pago de la cuota sindical	DTrabajador PagoService

Tabla 25. Tarjeta CRC # 9

Tarjeta CRC	
<b>Clase:</b> ReporteAtrasoMTTController	

<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de gestionar todos los datos necesarios para generar un reporte con los trabajadores atrasados en el pago del compromiso MTT.	TipoNivelService ReporteAtrasoMTTService PDFRenderingService

Tabla 26. Tarjeta CRC # 10

<b>Tarjeta CRC</b>	
<b>Clase: ReporteAtrasoFinanzasController</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
Esta clase es la encargada de gestionar todos los datos necesarios para generar un reporte con los trabajadores atrasados en el pago de la cuota sindical.	TipoNivelService ReporteAtrasoFinanzasService PDFRenderingService

## 2.5. Conclusiones del capítulo

Se abordaron los artefactos que propone la metodología de desarrollo y sirven como eslabón para la siguiente etapa de desarrollo. Se creó el plan de iteraciones agrupando las HU según su prioridad y se realizó el plan de entregas donde se estimó el tiempo de desarrollo en semanas para cada una de las iteraciones. En este capítulo se analizaron los patrones de diseño asociados a la arquitectura propuesta, los cuales darán alto grado de independencia a las clases y facilitarán la implementación. Se realizan las tarjetas CRC para obtener la representación de las principales clases y funcionalidades del sistema. Como parte del diseño del sistema se realizó el diagrama entidad relación de la base de datos que permite almacenar los datos necesarios de la misma.

## **Capítulo 3. Implementación y Pruebas**

### **3.1. Introducción**

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se pone en vigor todas las descripciones y arquitecturas propuestas en la fase de análisis y diseño, es la forma de complementar el trabajo de las fases que lo preceden dentro del proceso de desarrollo de software. La implementación proporciona una materialización precisa de los requisitos. Una de las últimas fases del ciclo de vida antes de entregar un software para su explotación es la fase de pruebas las cuales se realizan conjuntamente con los usuarios finales como forma de validación de la solución desarrollada, con el fin de contar con todas las evidencias necesarias que garanticen el cumplimiento en cuanto a funcionalidad, calidad y alcance esperado por el cliente.

### **3.2. Implementación**

En las metodologías pesadas, el proceso de codificación se alcanza después de largas fases de análisis y diseño las cuales arrojan gran cantidad de documentación a partir del cual el proceso de codificación es relativamente sencillo. En XP este proceso es muy diferente, debido a que desde un principio se inicia con la codificación, favoreciendo el logro del objetivo de estar haciendo entregas al cliente frecuentemente. Algunos de los elementos más importantes en cuanto a la codificación son que, el cliente siempre debe de estar presente en ésta, se debe trabajar en parejas y debe haber una propiedad colectiva del código. Todos estos elementos representan paradigmas nuevos en lo que a la ingeniería de software se refiere, planteando entornos de discusión sobre la convivencia de adoptarlas. (33)

La metodología ágil de desarrollo de software XP está compuesta por fases, la implementación en conjunto con las pruebas juega un papel importante unidas a las demás. En esta fase se realizó la implementación de las HU que fueron seleccionadas por cada iteración.

#### **3.2.1. Tareas de ingeniería**

Las Tareas de Ingenierías (TI) son el artefacto que en la fase de implementación y prueba específicamente en la implementación se encargan de almacenar los detalles para la realización de cada

una de las HU. Una HU puede tener más de una TI lo cual depende de la complejidad de dicha HU. Cuando el equipo de desarrollo conforma las TI deja bien claro en sus campos a qué HU pertenece. Los programadores responsables de implementar dicha funcionalidad, se establecen fecha de inicio y de fin en correspondencia con los datos que aporta la HU en cuestión, los puntos estimados que indican el tiempo real que se cogieron los programadores en realizar la funcionalidad, el tipo de tarea que se le va a realizar, nombre de la misma, y la parte final pero no menos importante la descripción de la tarea; en éste campo como su nombre lo indica se detalla todo lo que se va a realizar y en lo que se va a apoyar para darle solución a la funcionalidad correspondiente. Las TI es el artefacto en el que se apoya el equipo desarrollador para dejar plasmado la manera en que se le da solución a la HU a la que está asignada la TI.

Tabla 27. Tarea de Ingeniería # 1

Tarea	
<b>Número de tarea:</b> 1	<b>Número de Historia de usuario:</b> 1
<b>Nombre de la tarea:</b> Crear Estructuras.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.5
<b>Fecha de inicio:</b> 5 de diciembre de 2016	<b>Fecha de fin:</b> 13 de diciembre de 2016
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> luego de la autenticación del usuario, el sistema le permitirá poder Crear Estructuras especificando un nombre, los niveles y secciones sindicales que forman parte de la misma.	

Tabla 28. Tarea de Ingeniería # 2

Tarea	
<b>Número de tarea:</b> 2	<b>Número de Historia de usuario:</b> 2

<b>Nombre de la tarea:</b> Dar Baja.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha de inicio:</b> 14 de diciembre de 2016	<b>Fecha de fin:</b> 16 de diciembre de 2016
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<p><b>Descripción:</b> luego de la autenticación del usuario, el sistema proporcionará la opción Mostrar Trabajadores, dicha opción brinda la facilidad de filtrar la búsqueda por los siguientes campos: Carnet de Identidad, Nombre(s), Primer Apellido, Segundo Apellido, Cargo y Sección sindical. Como resultado de la búsqueda se muestran los trabajadores registrados en el sistema. Para cada trabajador afiliado se ofrece la opción Dar Baja la cual para poder realizarla se debe dejar plasmado por escrito el motivo por el que causará baja de la sección sindical.</p>	

Tabla 29. Tarea de Ingeniería # 3

Tarea	
<b>Número de tarea:</b> 3	<b>Número de Historia de usuario:</b> 3
<b>Nombre de la tarea:</b> Cambiar Escala.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 19 de diciembre de 2016	<b>Fecha de fin:</b> 23 de diciembre de 2016
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<p><b>Descripción:</b> luego de la autenticación del usuario, el sistema proporcionará la opción Registrar Cuota. Esta opción muestra un listado con las cuotas registradas en el sistema,</p>	

para cada una de estas cuotas se brinda la opción de cambiar escala.

Tabla 30. Tarea de Ingeniería # 4

Tarea	
<b>Número de tarea:</b> 4	<b>Número de Historia de usuario:</b> 4
<b>Nombre de la tarea:</b> Registrar pago CTC.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 26 de diciembre de 2016	<b>Fecha de fin:</b> 30 de diciembre de 2016
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> luego de la autenticación del usuario, el sistema proporcionará la opción Registrar Pago CTC, dicha opción brinda la facilidad de realizar una búsqueda por los siguientes campos: Carnet de Identidad, Nombre(s), Primer Apellido, Segundo Apellido, Cargo y Sección sindical. Como resultado de la búsqueda se obtiene un listado de los trabajadores afiliados a una sección sindical. Para cada uno de esos trabajadores se muestra la opción Registrar Pago la cual luego de seleccionarla brinda la opción de rellenar el campo cantidad de meses a pagar.	

Tabla 31. Tarea de Ingeniería # 5

Tarea	
<b>Número de tarea:</b> 5	<b>Número de Historia de usuario:</b> 5
<b>Nombre de la tarea:</b> Registrar Pago MTT.	

<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 2 de enero de 2017	<b>Fecha de fin:</b> 5 de enero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<p><b>Descripción:</b> luego de la autenticación del usuario, el sistema proporcionará la opción Registrar Pago MTT, dicha opción brinda la facilidad de realizar una búsqueda por los siguientes campos: Carnet de Identidad, Nombre(s), Primer Apellido, Segundo Apellido, Cargo y Sección Sindical. Como resultado de la búsqueda se obtiene un listado de los trabajadores afiliados a una sección sindical. Para cada uno de esos trabajadores se muestra la opción registrar el compromiso, luego de seleccionarla se debe especificar el compromiso y el mes a pagarlo. Una vez que el compromiso es registrado pues se procede a registrar el pago MTT especificando el aporte.</p>	

Tabla 32. Tarea de Ingeniería # 6

Tarea	
<b>Número de tarea:</b> 6	<b>Número de Historia de usuario:</b> 6
<b>Nombre de la tarea:</b> Registrar Depósito CTC.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha de inicio:</b> 6 de enero de 2017	<b>Fecha de fin:</b> 10 de enero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<p><b>Descripción:</b> luego de la autenticación del usuario, el sistema proporcionará la opción Registrar Depósito CTC, para realizar el mismo se debe haber creado la estructura previamente y debe existir al menos un afiliado a una sección sindical. Para registrar el</p>	

depósito se selecciona el nivel y el área correspondiente.

Tabla 33. Tarea de Ingeniería # 7

Tarea	
<b>Número de tarea:</b> 7	<b>Número de Historia de usuario:</b> 7
<b>Nombre de la tarea:</b> Registrar Depósito MTT.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha de inicio:</b> 11 de enero de 2017	<b>Fecha de fin:</b> 13 de enero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> luego de la autenticación del usuario, el sistema proporcionará la opción Registrar Depósito MTT, para realizar el mismo se debe haber creado la estructura previamente y debe existir al menos un afiliado a una sección sindical. Para registrar el depósito se selecciona el nivel y el área correspondiente.	

Tabla 34. Tarea de Ingeniería # 8

Tarea	
<b>Número de tarea:</b> 8	<b>Número de Historia de usuario:</b> 8
<b>Nombre de la tarea:</b> Generar Registro de Control de Sindicalización.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 16 de enero de 2017	<b>Fecha de fin:</b> 20 de enero de 2017

<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.
<b>Descripción:</b> luego de la autenticación del usuario, el sistema brinda la posibilidad de generar en formato PDF el Registro de Control de Sindicalización de los afiliados a una sección sindical pero antes va a tener que seleccionar una estructura y un área a la que pertenece.

Tabla 35. Tarea de Ingeniería # 9

Tarea	
<b>Número de tarea:</b> 9	<b>Número de Historia de usuario:</b> 9
<b>Nombre de la tarea:</b> Generar Registro de Comprometido y Plan Mi Aporte a la Patria.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 23 de enero de 2017	<b>Fecha de fin:</b> 27 de enero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> luego de la autenticación del usuario, el sistema brinda la posibilidad de generar en formato PDF el Registro de Comprometido y Plan Mi Aporte a la Patria de los afiliados a una sección sindical pero antes va a tener que seleccionar una estructura y un área a la que pertenece.	

Tabla 36. Tarea de Ingeniería # 10

Tarea	
<b>Número de tarea:</b> 10	<b>Número de Historia de usuario:</b> 10

<b>Nombre de la tarea:</b> Generar Modelo de Baja.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 30 de enero de 2017	<b>Fecha de fin:</b> 3 de febrero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> luego de la autenticación del usuario, el sistema brinda la posibilidad de generar en formato PDF el Modelo de Baja de los afiliados a una sección sindical pero antes va a tener que seleccionar una estructura y un área a la que pertenece.	

Tabla 37. Tarea de Ingeniería # 11

Tarea	
<b>Número de tarea:</b> 11	<b>Número de Historia de usuario:</b> 11
<b>Nombre de la tarea:</b> Generar Modelo de Alta.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 6 de febrero de 2017	<b>Fecha de fin:</b> 10 de febrero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<b>Descripción:</b> luego de la autenticación del usuario, el sistema brinda la posibilidad de generar en formato PDF el Modelo de Alta de los afiliados a una sección sindical pero antes va a tener que seleccionar una estructura y un área a la que pertenece.	

Tabla 38. Tarea de Ingeniería # 12

Tarea	
<b>Número de tarea:</b> 12	<b>Número de Historia de usuario:</b> 12
<b>Nombre de la tarea:</b> Generar Informe de Comprometidos y actividad económica MTT.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 13 de febrero de 2017	<b>Fecha de fin:</b> 17 de febrero de 2017
<b>Programador responsable:</b> Raidel Alejandro Villafaña Fernández.	
<p><b>Descripción:</b> luego de la autenticación del usuario, el sistema brinda la opción de Registrar Depósito MTT. Un vez registrado dicho depósito se muestra una tabla con los depósitos existentes en el sistema, para cada uno ofrece la opción de Generar en formato PDF el Informe de Comprometidos y actividad económica MTT de los afiliados a una sección sindical</p>	

Tabla 39. Tarea de Ingeniería # 13

Tarea	
<b>Número de tarea:</b> 13	<b>Número de Historia de usuario:</b> 13
<b>Nombre de la tarea:</b> Generar Informe de Afiliación y actividad económica de Finanzas.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1.0
<b>Fecha de inicio:</b> 20 de febrero de 2017	<b>Fecha de fin:</b> 24 de febrero de 2017

**Programador responsable:** Raidel Alejandro Villafaña Fernández.

**Descripción:** luego de la autenticación del usuario, el sistema le brinda la opción de Registrar Depósito CTC. Un vez registrado dicho deposito se muestra una tabla con los depósitos existentes en el sistema, para cada uno ofrece la opción de generar en formato PDF el Informe de Afiliación y actividad económica de Finanzas de los afiliados a una sección sindical

El resto de las TI se encuentran en el Anexo C.

### **3.3. Pruebas del Software**

El proceso de producción de software requiere de diversas pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno cliente. Las pruebas de software por sus siglas en inglés (Software Testing) comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un software, por medio de pruebas sobre el comportamiento del mismo. (34)

#### **La automatización de las pruebas:** (35)

Las principales ventajas de la utilización de pruebas unitarias automáticas en el desarrollo son:

- Fomentan el cambio: facilitan que el programador cambie el código para mejorar su estructura, puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
- Simplifica la integración: permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
- Documenta el código: las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.
- Los errores están más acotados y son más fáciles de localizar: dado que se tienen pruebas unitarias que pueden desenmascararlos.

#### **3.3.1. Pruebas Unitarias**

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. (25)

Las pruebas unitarias, son realizadas desde el punto de vista del programador y sirven, además de probar el código, para poder realizar la refactorización del mismo. Cada programador, antes de comenzar a programar, debe preparar las pruebas unitarias. Esto hace que dichas pruebas estén preparadas para ser corridas durante la codificación y además, hace que al programador le surjan dudas y pueda evacuarlas con el cliente antes de empezar con la codificación. (36)

En la Figura 6 se muestra la interfaz del resultado de la aplicación de estas pruebas que se ejecutaron mediante el framework Grails. Dichas pruebas se le realizan a las clases controladoras y a los servicios, a las clases entidades no se les realizan pruebas debido a que para cada clase entidad existe un servicio asociado a dicha clase que se encargan de crear y guardar en base de datos sus objetos. Como se puede observar en la figura 6, se realizaron 35 pruebas unitarias, las cuales resultaron 100% efectivas.



Figura 6. Resultado de la ejecución de las pruebas unitarias

Elaboración propia.

### 3.3.2. Pruebas de Aceptación

Los programadores escriben las pruebas funcionales para cada HU que deba validarse. Una HU no es aceptada hasta que haya pasado su prueba de aceptación. Las pruebas de aceptación representan algún

tipo de resultado por parte del sistema. Los clientes son los responsables de verificar la exactitud de estas pruebas y de revisar los resultados para poder así priorizar las que fracasaron. Esto significa que en cada iteración se deben realizar nuevas pruebas de aceptación. Las pruebas de aceptación en XP, permiten verificar que las funcionalidades de cada iteración se cumplan correctamente. El probador es el encargado de ayudar al cliente a seleccionar y escribir las pruebas de aceptación para cada HU. Tiene la responsabilidad de ayudar al cliente a tomar las decisiones correctas sobre que significa la calidad para su proyecto. (36)

Tabla 40. Prueba de aceptación # 1

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>Historia de usuario:</b> 1
<b>Nombre:</b> Crear Estructuras.	
<b>Descripción:</b> prueba para la funcionalidad Crear Estructuras.	
<b>Condiciones de ejecución:</b> el Secretario del Sindicato es el único que tiene los permisos para Crear Estructuras con sus Secciones Sindicales correspondientes. Los demás usuarios solo tienen permiso para ver los afiliados y las secciones sindicales a la que pertenecen.	
<b>Pasos de ejecución:</b> el usuario intenta Crear una Estructura con sus Secciones Sindicales en el sistema con los datos válidos.	
<b>Resultados esperados:</b> el usuario crea la Estructura con sus Secciones Sindicales correctamente en el sistema.	

Tabla 41. Prueba de aceptación # 2

Caso de prueba de aceptación
------------------------------

<b>Código:</b> HU2_P2	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Dar Baja.	
<b>Descripción:</b> prueba para la funcionalidad Dar Baja.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir un afiliado a una Sección Sindical.	
<b>Pasos de ejecución:</b> el usuario debe especificar los motivos para darle baja a un afiliado.	
<b>Resultados esperados:</b> el usuario le da baja correctamente a un afiliado.	

Tabla 42. Prueba de aceptación # 3

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3_P3	<b>Historia de usuario:</b> 3
<b>Nombre:</b> Cambiar Escala.	
<b>Descripción:</b> prueba para la funcionalidad Cambiar Escala.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir al menos una cuota registrada en el sistema.	
<b>Pasos de ejecución:</b> el usuario debe seleccionar la opción cambiar escala y luego modificar los datos que existen en el sistema.	
<b>Resultados esperados:</b> el usuario cambia correctamente la escala en el sistema.	

Tabla 43. Prueba de aceptación # 4

Caso de prueba de aceptación	
<b>Código:</b> HU4_P4	<b>Historia de usuario:</b> 4
<b>Nombre:</b> Registrar Pago CTC.	
<b>Descripción:</b> prueba para la funcionalidad Registrar pago CTC.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir al menos un trabajador afiliado a una sección sindical.	
<b>Pasos de ejecución:</b> el usuario debe realizar una búsqueda y posteriormente seleccionar el trabajador al cual le registrará el Pago CTC rellenando el campo Cantidad de Meses a Pagar.	
<b>Resultados esperados:</b> el usuario registra correctamente en el sistema el Pago CTC.	

Tabla 44. Prueba de aceptación # 5

Caso de prueba de aceptación	
<b>Código:</b> HU5_P5	<b>Historia de usuario:</b> 5
<b>Nombre:</b> Registrar Pago MTT.	
<b>Descripción:</b> prueba para la funcionalidad Registrar Pago MTT.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	

Debe existir al menos un trabajador afiliado a una sección sindical.
<b>Pasos de ejecución:</b> el usuario debe realizar una búsqueda y posteriormente seleccionar el trabajador al cual le registrará el Pago MTT. Luego debe Registrar el Compromiso relleno los campos Compromiso y Mes a Pagar. Seguidamente debe realizar nuevamente la búsqueda y Registrar el Pago relleno el campo aporte.
<b>Resultados esperados:</b> el usuario registra correctamente en el sistema el Pago MTT.

Tabla 45. Prueba de aceptación # 6

Caso de prueba de aceptación	
<b>Código:</b> HU6_P6	<b>Historia de usuario:</b> 6
<b>Nombre:</b> Registrar Depósito CTT.	
<b>Descripción:</b> prueba para la funcionalidad Registrar Depósito CTC.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir al menos una Sección Sindical. Debe existir al menos un trabajador afiliado a una Sección Sindical. Debe estar registrado al menos un pago de la cuota sindical de un trabajador.	
<b>Pasos de ejecución:</b> el usuario debe seleccionar una Sección Sindical y presionar en la opción Aceptar.	
<b>Resultados esperados:</b> el usuario registra correctamente en el sistema el Deposito CTC.	

Tabla 46. Prueba de aceptación # 7

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU7_P10	<b>Historia de usuario:</b> 7
<b>Nombre:</b> Registrar Depósito MTT.	
<b>Descripción:</b> prueba para la funcionalidad Registrar Depósito MTT.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir al menos una Sección Sindical. Debe existir al menos un trabajador afiliado a una Sección Sindical. Debe estar registrado al menos un pago del compromiso de un trabajador de mi Aporte a la Patria.	
<b>Pasos de ejecución:</b> el usuario debe seleccionar una Sección Sindical y presionar en la opción Aceptar.	
<b>Resultados esperados:</b> el usuario registra correctamente en el sistema el Deposito MTT.	

Tabla 47. Prueba de aceptación # 8

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU8_P8	<b>Historia de usuario:</b> 8
<b>Nombre:</b> Generar Registro de Control de Sindicalización.	
<b>Descripción:</b> prueba para la funcionalidad Generar Registro de Control de Sindicalización.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado.	

Debe existir al menos una Sección Sindical.
<b>Pasos de ejecución:</b> el usuario debe seleccionar una Sección Sindical a la cual le va a Generar el Registro de Control de Sindicalización.
<b>Resultados esperados:</b> el usuario genera correctamente en el sistema el Registro de Control de Sindicalización.

Tabla 48. Prueba de aceptación # 9

Caso de prueba de aceptación	
<b>Código:</b> HU9_P9	<b>Historia de usuario:</b> 9
<b>Nombre:</b> Generar Registro de Comprometido y Plan Mi Aporte a la Patria.	
<b>Descripción:</b> prueba a la funcionalidad Generar Registro de Comprometido y Plan Mi Aporte a la Patria.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir al menos una Sección Sindical.	
<b>Pasos de ejecución:</b> el usuario debe seleccionar una Sección Sindical a la cual le va a Generar el Registro de Comprometido y Plan Mi Aporte a la Patria.	
<b>Resultados esperados:</b> el usuario genera correctamente en el sistema el Registro de Comprometido y Plan Mi Aporte a la Patria.	

Tabla 49. Prueba de aceptación # 10

Caso de prueba de aceptación
------------------------------

<b>Código:</b> HU10_P10	<b>Historia de usuario:</b> 10
<b>Nombre:</b> Generar Modelo de Baja.	
<b>Descripción:</b> Prueba para la funcionalidad Generar Modelo de Baja.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. El usuario debe seleccionar la Sección Sindical a la cual le va a generar el reporte y posteriormente seleccionar el mes e introducir el año.	
<b>Pasos de ejecución:</b> el usuario intenta generar el Modelo de Baja con los datos válidos.	
<b>Resultados esperados:</b> el usuario genera correctamente en el sistema el Modelo de Baja.	

Tabla 50. Prueba de aceptación # 11

Caso de prueba de aceptación	
<b>Código:</b> HU11_P11	<b>Historia de usuario:</b> 11
<b>Nombre:</b> Generar Modelo de Alta.	
<b>Descripción:</b> prueba para la funcionalidad Generar Modelo de Alta.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. El usuario debe seleccionar la Sección Sindical a la cual le va a generar el reporte y posteriormente seleccionar el mes e introducir el año.	
<b>Pasos de ejecución:</b> el usuario intenta generar el Modelo de Alta con los datos válidos.	

**Resultados esperados:** el usuario genera correctamente en el sistema el Modelo de Alta.

Tabla 51. Prueba de aceptación # 12

Caso de prueba de aceptación	
<b>Código:</b> HU12_P12	<b>Historia de usuario:</b> 12
<b>Nombre:</b> Generar Informe de Comprometidos y actividad económica MTT.	
<b>Descripción:</b> prueba para la funcionalidad Generar Informe de Comprometidos y actividad económica MTT.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado. Debe existir al menos un Depósito MTT.	
<b>Pasos de ejecución:</b> el usuario debe elegir la opción Generar Informe de Comprometidos y actividad económica MTT.	
<b>Resultados esperados:</b> el usuario registra correctamente en el sistema el Informe de Comprometidos y actividad económica MTT.	

Tabla 52. Prueba de aceptación # 13

Caso de prueba de aceptación	
<b>Código:</b> HU13_P13	<b>Historia de usuario:</b> 13
<b>Nombre:</b> Generar Informe de Afiliación y actividad económica de Finanzas.	
<b>Descripción:</b> prueba para la funcionalidad Generar Informe de Afiliación y actividad	

económica de Finanzas.

**Condiciones de ejecución:**

El usuario debe estar autenticado.  
Debe existir al menos un Depósito CTC.

**Pasos de ejecución:** el usuario debe elegir la opción Generar Informe de Afiliación y actividad económica de las Finanzas.

**Resultados esperados:** el usuario registra correctamente en el sistema el Informe de Afiliación y actividad económica de Finanzas.

El resto de las pruebas de aceptación se encuentran en el Anexo D.

A la propuesta de solución se le realizaron 35 pruebas unitarias las cuales las 35 fueron satisfactorias. De los 22 casos de prueba de aceptación fueron resueltas las 4 no conformidades detectadas, y se realizaron los 7 cambios solicitados. Quedando aprobado por el cliente la propuesta de solución final planteada.

### **3.4. Conclusiones del capítulo**

Las pruebas unitarias al igual que las pruebas de aceptación son una actividad fundamental en XP, las primeras deben ser automatizadas y elaboradas antes y durante la implementación de una clase, las segundas jugaron un papel primordial pues es la manera que tiene el cliente de verificar que las HU fueron implementadas correctamente. En éste capítulo se abordaron los temas referidos a la implementación y prueba de la investigación. Se cumplió con cada uno de los objetivos trazados para la validación y prueba del sistema con un todo ya que, cada uno de los resultados obtenidos en las diferentes iteraciones, resultaron de evaluación satisfactoria por las pruebas de aceptación aplicadas. Por tanto, la culminación del presente capítulo representa la garantía y satisfacción conjunta del cliente y el equipo de desarrollo con el producto final.

## Conclusiones

Una vez realizado el desarrollo del sistema se arriba las siguientes conclusiones:

1. Se realizó el estudio del estado del arte permitiendo identificar la necesidad de desarrollar un SI para la gestión de información de las finanzas del Sindicato puesto que ninguno de los estudiados cumplía con las necesidades específicas del área. Se justificaron las herramientas, metodología y tecnologías informáticas que se utilizaron en el desarrollo del trabajo las cuales posibilitaron la implementación del sistema en función de cumplir con el objetivo de la investigación.
2. Se plantearon 22 HU estableciéndose la estimación de esfuerzo para cada una de ellas que arrojó un tiempo a realizarse la investigación de 24.5 semanas (para un aproximado de 6 meses). Se creó el plan de iteraciones en el cual se definieron 3 entregas para el cliente. Se analizaron los patrones de diseño asociados a la arquitectura propuesta los cuales facilitaron la implementación. Como parte del diseño del sistema se realizó el diagrama entidad relación de la base de datos permitiendo almacenar los datos necesarios de la misma y se plasmaron 10 tarjetas CRC para obtener las principales funcionalidades del sistema.
3. Se conformaron 22 TI para poder almacenar los detalles de cada una de las HU. Se implementaron las funcionalidades identificadas y se realizaron las pruebas pertinentes definidas por la metodología de desarrollo de software XP, para la validación de la solución propuesta. Para ello se utilizó el diseño y la aplicación de las pruebas unitarias y de aceptación, validando la calidad de la solución propuesta con resultados positivos.

Atendiendo a lo anteriormente planteado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente. El sistema informático desarrollado contribuirá de manera significativa para la gestión de Información de las finanzas del Sindicato.

## **Recomendaciones**

Los resultados obtenidos en este trabajo sugieren una serie de recomendaciones para seguir perfeccionando la gestión de información del proceso que se realiza en el Sindicato. Por lo que el autor recomienda:

- El despliegue del sistema, para que el proceso de gestión de la información se agilice y se realice de manera informatizada y centralizada.
- Añadirle a la funcionalidad Registrar Trabajador el consumo de los servicios web publicados en la UCI.
- Agregarle al SI una funcionalidad que le notifique a través de correo electrónico a los afiliados que se encuentren atrasados en sus pagos.

## Referencias bibliográficas

1. **Rolando Alfredo Hernández León, Sayda Coello González.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA.* La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2012.
2. La gestión de información y la gestión del conocimiento. *The management of information and the management of knowledge.* [Online] Abril 2015. [Cited: Octubre 16, 2016.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1025-02552015000200002&lng=es&nrm=iso&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1025-02552015000200002&lng=es&nrm=iso&tlng=es). ISSN 1025-0255.
3. *Gestión de la Información, gestión de contenidos y conocimiento.* **Arévalo, Julio Alonso.** Salamanca : s.n., 2007.
4. **Ingram, David.** La Voz de Houston. [Online] [Cited: Noviembre 2, 2016.] <http://pyme.lavoztx.com/qu-es-un-sistema-de-gestin-de-la-informacin-7690.html>.
5. **Alegsa, Leandro.** DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA . [Online] Junio 22, 2016. [Cited: Octubre 31, 2016.] [http://www.alegsa.com.ar/Dic/sistema\\_informatico.php](http://www.alegsa.com.ar/Dic/sistema_informatico.php).
6. **Orlando Yacsell Martínez Batista, Richard Yanes Rapado.** *Sistema de Gestión Académica de Postgrado versión 2.0.* La Habana : s.n., 2014.
7. EcuRed. *Conocimiento con todos y para todos.* [Online] [Cited: Enero 11, 2017.] <https://www.ecured.cu/SNTECD>.
8. GuíasolucionesTIC. [Online] [Cited: Enero 24, 2017.] <http://www.guiadesolucionestic.com/sistemas-de-informacion/gestion-financiera/software-contable/>.
9. Ingeniería de Software. [Online] 2017. [Cited: Enero 15, 2017.] [http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html).
10. **barbarapvn.** Entendiendo HTML5: guía para principiantes. [Online] Mayo 28, 2013. [Cited: Noviembre 7, 2016.] <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>.
11. Introducción a CSS 3. [Online] Junio 9, 2008. [Cited: Noviembre 7, 2016.] <http://www.desarrolloweb.com/articulos/introduccion-css3.html>.
12. Tecnología Innovadora. [Online] Mayo 14, 2012. [Cited: Noviembre 7, 2016.] [http://tecnologiainnovadoraunad.blogspot.com/2012/05/cuales-son-las-ventajas-y-desventajas\\_14.html](http://tecnologiainnovadoraunad.blogspot.com/2012/05/cuales-son-las-ventajas-y-desventajas_14.html).
13. Didáctica y divulgación de la programación. [Online] [Cited: noviembre 7, 2016.] [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=590:i-que-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:i-que-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192).
14. aula formativa. [Online] Febrero 6, 2016. [Cited: 11 7, 2016.] <http://blog.aulaformativa.com/ventajas-desventajas-usar-framework-javascript/>.
15. THE APACHE SOFTWARE FOUNDATION. *Apache Tomcat.* [Online] [Cited: Noviembre 1, 2016.] <http://tomcat.apache.org/>.
16. **Blanco, Cad. Yamila Valdés.** *Módulo de Seguridad para los servicios de la plataforma de telecomunicación PLATEL.* La Habana : s.n., 2014.

17. PostgreSQL. [Online] Noviembre 26, 2011. [Cited: Noviembre 8, 2016.] <http://postgresql-ads.blogspot.com/2011/11/ampliamente-popular-ideal-para.html>.
18. **Reingart, Mariano.** *pgAdmin 3*.
19. ingeniería de software. [Online] Septiembre 9, 2016. [Cited: Noviembre 8, 2016.] [http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html).
20. *Guión Visual Paradigm for UML*. 2014.
21. Tecnología para Desarrollo. [Online] Octubre 16, 2010. [Cited: Noviembre 8, 2016.] <https://www.paradigmadigital.com/dev/se-puede-hacer-una-gran-aplicacion-con-groovy-y-grails/>.
22. software.com.ar. [Online] [Cited: Enero 15, 2017.] <http://www.software.com.ar/p/intellij-idea>.
23. Ciclo de vida de un proyecto XP. *Ciclo de vida de un proyecto XP*. [Online] [Cited: Marzo 5, 2017.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
24. La oficina de proyectos de informática. *La web sobre gerencia de proyectos de informática, software y tecnología*. [Online] 2015. [Cited: Marzo 5, 2017.] <http://www.pmoinformatica.com/>.
25. **Patricio Letelier, M<sup>a</sup> Carmen Penadés.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [Online] Técnica Administrativa, Buenos Aires, Enero 15, 2006. [Cited: Marzo 5, 2017.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. ISSN 1666-1680.
26. Ingenio DS. *Patrones arquitectónicos*. [Online] Septiembre 16, 2016. [Cited: Marzo 6, 2017.] <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
27. EcuRed. *Conocimientos con todos y para todos*. [Online] 2017. [Cited: Marzo 6, 2017.] <https://www.ecured.cu/Cliente-Servidor>.
28. CCM. *Entorno cliente/servidor*. [Online] 2017. [Cited: Marzo 6, 2017.] <http://es.ccm.net/contents/148-entorno-cliente-servidor>.
29. CakePHP. *CakePHP 2.x Cookbook*. [Online] Marzo 13, 2017. [Cited: Marzo 5, 2017.] <https://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>.
30. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN:970-17-0261-1.
31. **Pantaleón, Marta E Zorrilla.** *Modelo de Datos*. s.l. : Universidad de Cantabria, 2011.
32. Gestión de proyectos y desarrollo de software. *Desarrollo de software. Tarjetas CRC*. [Online] Enero 10, 2012. [Cited: Mayo 20, 2017.] <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
33. **Luis Miguel Echeverry Tobón, Luz Elena Delgado Carmona.** *Caso Práctico de la Metodología Ágil XP al Desarrollo de Software*. 2007.
34. La oficina de proyectos de informática. *La web sobre gerencia de proyectos de informática, software y tecnología*. [Online] [Cited: Mayo 21, 2017.] <http://www.pmoinformatica.com/p/pruebas-de-software.html>.

35. **Dayvis Malfará, Diego Cukeman, Fernando Cócar, Juan Pablo Cassinelli, Renzo Séttimo.** *Testing en eXtreme Programming.*
36. **Joel Marcos Balarezo Penadillo, Eveling Giselle Cruz Vasquez, Franshesca Lamadrid Bringas.** Metodologías ágiles. *Programacion Extrema XP.* [Online] Universidad Nacional de Trujillo, 2013. [Cited: Mayo 25, 2017.] <https://es.slideshare.net/EvelingGiselleCruzVS/metodologia-monografia>.
37. Universidad de las Ciencias Informáticas. [Online] [Cited: Junio 13, 2017.] <http://www.uci.cu/>.
38. Apache Groovy. [Online][Cited: Junio 14, 2017.] <http://groovy-lang.org/>

## Bibliografía

1. **Rolando Alfredo Hernández León, Sayda Coello González.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA.* La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2012.
2. La gestión de información y la gestión del conocimiento. *The management of information and the management of knowledge.* [Online] Abril 2015. [Cited: Octubre 16, 2016.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1025-02552015000200002&lng=es&nrm=iso&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1025-02552015000200002&lng=es&nrm=iso&tlng=es). ISSN 1025-0255.
3. *Gestión de la Información, gestión de contenidos y conocimiento.* **Arévalo, Julio Alonso.** Salamanca : s.n., 2007.
4. **Ingram, David.** La Voz de Houston. [Online] [Cited: Noviembre 2, 2016.] <http://pyme.lavoztx.com/qu-es-un-sistema-de-gestin-de-la-informacin-7690.html>.
5. **Alegsa, Leandro.** DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA . [Online] Junio 22, 2016. [Cited: Octubre 31, 2016.] [http://www.alegsa.com.ar/Dic/sistema\\_informatico.php](http://www.alegsa.com.ar/Dic/sistema_informatico.php).
6. **Orlando Yacsell Martínez Batista, Richard Yanes Rapado.** *Sistema de Gestión Académica de Postgrado versión 2.0.* La Habana : s.n., 2014.
7. EcuRed. *Conocimiento con todos y para todos.* [Online] [Cited: Enero 11, 2017.] <https://www.ecured.cu/SNTECD>.
8. GuíasolucionesTIC. [Online] [Cited: Enero 24, 2017.] <http://www.guiadesolucionestic.com/sistemas-de-informacion/gestion-financiera/software-contable/>.
9. Ingeniería de Software. [Online] 2017. [Cited: Enero 15, 2017.] [http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html).
10. **barbarapvn.** Entendiendo HTML5: guía para principiantes. [Online] Mayo 28, 2013. [Cited: Noviembre 7, 2016.] <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>.
11. Introducción a CSS 3. [Online] Junio 9, 2008. [Cited: Noviembre 7, 2016.] <http://www.desarrolloweb.com/articulos/introduccion-css3.html>.
12. Tecnología Innovadora. [Online] Mayo 14, 2012. [Cited: Noviembre 7, 2016.] [http://tecnologiainnovadoraunad.blogspot.com/2012/05/cuales-son-las-ventajas-y-desventajas\\_14.html](http://tecnologiainnovadoraunad.blogspot.com/2012/05/cuales-son-las-ventajas-y-desventajas_14.html).
13. Didáctica y divulgación de la programación. [Online] [Cited: noviembre 7, 2016.] [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=590:ique-es-y-para-que-sirve-](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:ique-es-y-para-que-sirve-)

- javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192.
14. aula formativa. [Online] Febrero 6, 2016. [Cited: 11 7, 2016.] <http://blog.aulaformativa.com/ventajas-desventajas-usar-framework-javascript/>.
  15. THE APACHE SOFTWARE FOUNDATION. *Apache Tomcat*. [Online] [Cited: Noviembre 1, 2016.] <http://tomcat.apache.org/>.
  16. **Blanco, Cad. Yamila Valdés.** *Módulo de Seguridad para los servicios de la plataforma de telecomunicación PLATEL*. La Habana : s.n., 2014.
  17. PostGreSQL. [Online] Noviembre 26, 2011. [Cited: Noviembre 8, 2016.] <http://postgresql-adsi.blogspot.com/2011/11/ampliamente-popular-ideal-para.html>.
  18. **Reingart, Mariano.** *pgAdmin 3*.
  19. ingeniería de software. [Online] Septiembre 9, 2016. [Cited: Noviembre 8, 2016.] [http://www.um.es/docencia/barzana/LAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/LAGP/Enlaces/CASE_principales.html).
  20. *Guión Visual Paradigm for UML*. 2014.
  21. Tecnología para Desarrollo. [Online] Octubre 16, 2010. [Cited: Noviembre 8, 2016.] <https://www.paradigmadigital.com/dev/se-puede-hacer-una-gran-aplicacion-con-groovy-y-grails/>.
  22. software.com.ar. [Online] [Cited: Enero 15, 2017.] <http://www.software.com.ar/p/intellij-idea>.
  23. Ciclo de vida de un proyecto XP. *Ciclo de vida de un proyecto XP*. [Online] [Cited: Marzo 5, 2017.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
  24. La oficina de proyectos de informática. *La web sobre gerencia de proyectos de informática, software y tecnología*. [Online] 2015. [Cited: Marzo 5, 2017.] <http://www.pmoinformatica.com/>.
  25. **Patricio Letelier, M<sup>a</sup> Carmen Penadés.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [Online] Técnica Administrativa, Buenos Aires, Enero 15, 2006. [Cited: Marzo 5, 2017.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. ISSN 1666-1680.
  26. Ingenio DS. *Patrones arquitectónicos*. [Online] Septiembre 16, 2016. [Cited: Marzo 6, 2017.] <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
  27. EcuRed. *Conocimientos con todos y para todos*. [Online] 2017. [Cited: Marzo 6, 2017.] <https://www.ecured.cu/Cliente-Servidor>.
  28. CCM. *Entorno cliente/servidor*. [Online] 2017. [Cited: Marzo 6, 2017.] <http://es.ccm.net/contents/148-entorno-cliente-servidor>.
  29. CakePHP. *CakePHP 2.x Cookbook*. [Online] Marzo 13, 2017. [Cited: Marzo 5, 2017.] <https://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>.
  30. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN:970-17-0261-1.

31. **Pantaleón, Marta E Zorrilla.** *Modelo de Datos*. s.l. : Universidad de Cantabria, 2011.
32. Gestión de proyectos y desarrollo de software. *Desarrollo de software. Tarjetas CRC*. [Online] Enero 10, 2012. [Cited: Mayo 20, 2017.] <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
33. **Luis Miguel Echeverry Tobón, Luz Elena Delgado Carmona.** *Caso Práctico de la Metodología Agil XP al Desarrollo de Software*. 2007.
34. La oficina de proyectos de informática. *La web sobre gerencia de proyectos de informática, software y tecnología*. [Online] [Cited: Mayo 21, 2017.] <http://www.pmoinformatica.com/p/pruebas-de-software.html>.
35. **Dayvis Malfará, Diego Cukeman, Fernando Cócar, Juan Pablo Cassinelli, Renzo Séttimo.** *Testing en eXtreme Programming*.
36. **Joel Marcos Balarezo Penadillo, Eveling Giselle Cruz Vasquez, Franshesca Lamadrid Bringas.** *Metodologías ágiles. Programacion Extrema XP*. [Online] Universidad Nacional de Trujillo, 2013. [Cited: Mayo 25, 2017.] <https://es.slideshare.net/EvelingGiselleCruzVS/metodologia-monografia>.
37. Universidad de las Ciencias Informáticas. [Online] [Cited: Junio 13, 2017.] <http://www.uci.cu/>.
38. ConceptoDefinicion.de. [Online] Noviembre 16, 2014. [Cited: Noviembre 8, 2016.] <http://conceptodefinicion.de/php/>.
39. php. [Online] Mayo 18, 2007. [Cited: Noviembre 8, 2016.] <http://php.net/archive/2007.php>.
40. *Sistema de Gestión de Procesos*.
41. Metodología de la investigación . *Roberto Hernández Sampieri, Carlos Fernández Collado, Pilar Baptista Lucio*. [Online] Abril 2006. [https://competenciashg.files.wordpress.com/2012/10/sampieri-et-al-metodologia-de-la-investigacion-4ta-edicion-sampieri-2006\\_ocr.pdf](https://competenciashg.files.wordpress.com/2012/10/sampieri-et-al-metodologia-de-la-investigacion-4ta-edicion-sampieri-2006_ocr.pdf).
42. Ecured. *Conocimiento con todos y para todos*. [Online] [Cited: Enero 15, 2017.] [https://www.ecured.cu/Programaci%C3%B3n\\_Extrema\\_\(XP\)](https://www.ecured.cu/Programaci%C3%B3n_Extrema_(XP)).
43. Akus.net. *Aprender HTML, Lenguaje de Marcado de Hipertexto*. [Online] [Cited: Noviembre 7, 2016.]
44. Vivir y aprender web. [Online] Abril 4, 2015. [Cited: Noviembre 7, 2016.] <http://viviryaprenderweb.com/introduccion-a-css-hoja-de-estilo-en-cascada/>.
45. MDN.Mozilla Developer Network. *New in Java Script 1.6*. [Online] [Cited: Noviembre 7, 2016.] [https://developer.mozilla.org/en-US/docs/Web/JavaScript/New\\_in\\_JavaScript/1.6](https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/1.6).
46. PHP. [Online] [Cited: Noviembre 8, 2016.] <http://php.net/>.
47. The Apache Software Foundation. *Apache Tomcat*. [Online] [Cited: Noviembre 1, 2016.] <https://tomcat.apache.org/download-70.cgi>.
48. SND. [Online] [Cited: Noviembre 8, 2016.] <https://es.osdn.net/projects/pghintplan/releases/62456>.
49. PgAdmin. [Online] [Cited: Noviembre 9, 2016.] <https://www.pgadmin.org/docs/pgadmin3/1.22/index.html>.

50. Automated Guide-Through Project Management Process. [Online] [Cited: Noviembre 9, 2016.] <https://www.visual-paradigm.com/>.
51. Desarrollo Agil de Aplicaciones Web con Grails framework. [Online] Enero 2012. [Cited: Noviembre 8, 2016.] [https://www.uaeh.edu.mx/nuestro\\_alumnado/icbi/monografias/desarrollo%20agil.pdf](https://www.uaeh.edu.mx/nuestro_alumnado/icbi/monografias/desarrollo%20agil.pdf).
52. JetBrains. *IntelliJ IDEA*. [Online] [Cited: Enero 15, 2017.] <https://www.jetbrains.com/idea/documentation/>.
53. Historias de usuario. *umh2818-TADS*. [Online] [Cited: Marzo 5, 2017.] <http://umh2818.edu.umh.es/wp-content/uploads/sites/884/2016/02/Historias-de-usuario.pdf>.
54. Cliente Servidor. [Online] [Cited: Marzo 6, 2017.] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf).
55. Diseño e implementación de un Framework de Presentación. [Online] 2013. [Cited: Marzo 7, 2017.] <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/18968/9/drodriguezsiTFC0113memoria.pdf>.
56. Patrones de Diseño. [Online] [Cited: Marzo 9, 2017.] <http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>.
57. Biblioteca Nacional de España. [Online] 2016. [Cited: Marzo 10, 2017.] <http://www.bne.es/es/Inicio/Perfiles/Bibliotecarios/DatosEnlazados/Modelos/>.
58. Análisis y Diseño Orientado A Objetos. *TARJETAS CRC*. [Online] [Cited: Mayo 20, 2017.] <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%C3%B1o%20orientado%20a%20objetos/Mcrc.pdf>.
59. ARTEFACTOS XP. [Online] [Cited: Marzo 12, 2017.] <http://www.abacovirtual.edu.pe/chiclayo/filedocente/140318-CEI2012203MB-000029-19072012-173247.pdf>.
60. TESTING Y TDD CON GROOVY Y GRAILS. *XPWeek2011*. [Online] 2011. [Cited: Mayo 26, 2017.] <https://es.slideshare.net/albertovilches/introduccion-al-testing-con-grails-xpweek>.
61. w3ii.com. *Los últimos tutoriales de desarrollo web*. [Online] 2017. [Cited: Mayo 25, 2017.] [http://www.w3ii.com/es/software\\_testing\\_dictionary/acceptance\\_testing.html](http://www.w3ii.com/es/software_testing_dictionary/acceptance_testing.html).