

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4

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

**Módulos Finanzas y Emulación para el sistema de
Gestión Sindical de la Universidad de las Ciencias
Informáticas**

Autora: Lisandra Duany López

Tutor: Ing. Arcadio Abad Márquez

Co-tutor: Ing. Diego Miguel Torres Salazar

La Habana, junio 2014

“Año 56 de la Revolución”



Declaración de autoría:

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ de 2014.

Firma de Autora

Lisandra Duany López

Firma del Tutor

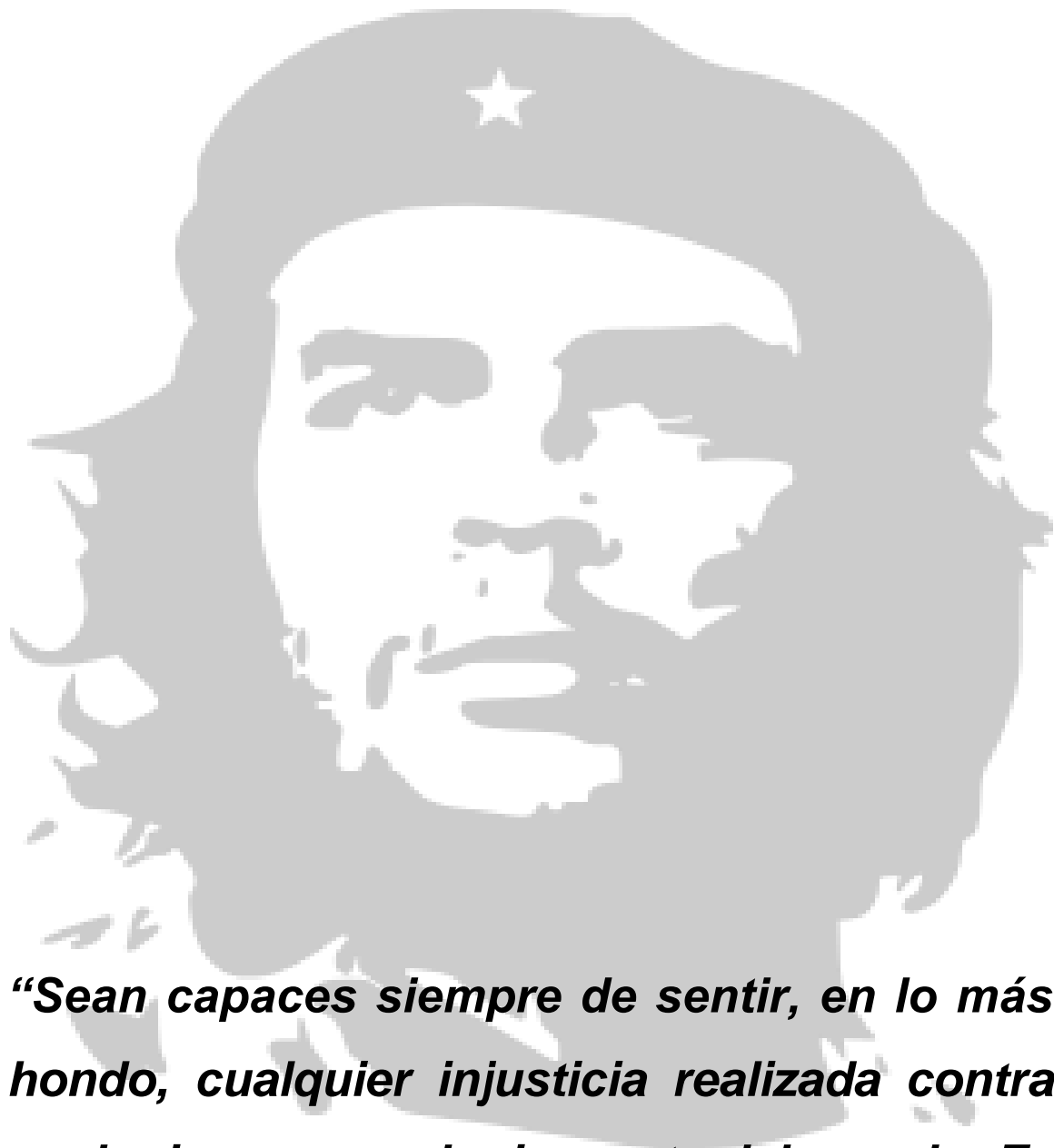
Ing. Arcadio Abad Márquez

Firma del Co-tutor

Ing. Diego Miguel Torres Salazar



Agradecimiento



“Sean capaces siempre de sentir, en lo más hondo, cualquier injusticia realizada contra cualquiera, en cualquier parte del mundo. Es la cualidad más linda del revolucionario.”



Ernesto Che Guevara

Dedicatoria

Dedico mi tesis a la persona más importante en mi vida, mi madre, porque ha sido, es y siempre lo será mi guía a seguir, por la que he logrado todas mis metas y sueños en la vida. Porque ha sido siempre madre y padre para mí, me ha apoyado en todo y ha dedicado su vida a mí. Me enseñó que para lograr algo en la vida primero hay que tener un sueño y después luchar para conseguirlo, todo requiere esfuerzo. A mis amigas que siempre confiaron en mí, y me dedico esta tesis a mí, porque es mi resultado de varios años de mucho sacrificio.



Agradecimientos

Primero que todo quiero agradecer a la revolución que me dio la oportunidad de estudiar en esta universidad, donde hoy terminé mis estudios como Ingeniera en Ciencias Informáticas, y donde pude conocer personas maravillosas a las que les quiero agradecer por formar parte de mi vida.

A mi madre en especial, que es la que ha estado en todo momento conmigo, cuando estaba triste por sacar una mala calificación o cuando estaba feliz por obtener mejores resultados. Por confiar en mí en todo momento cuando otros pensaron que yo no podía, ella siempre supo que sí.

A mi familia que me ha apoyado siempre desde que comencé a estudiar, en especial a mi hermana que siempre ha dicho desde niña que quería ser como yo, a mi abuela Blanca, a mi tía Mavis a mi tío Misael y a mis tías y abuela Milagros, Odalis y Manin, a mi tío manolito que siempre ha estado pendiente de mis estudios. A alguien que no es mi padre biológico pero para mí es mi padre porque es el que conozco y siempre se ha preocupado por mí y me ha dado ejemplo de padre, a Alexander y a su madre Beba que es como una abuela para mí.

A mi novio Salvatore, que me ha apoyado mucho desde que empezamos y en especial en estos últimos días que ha demostrado su apoyo hasta el final.

A la FEU, de la cual aprendí mucho y a todos los dirigentes con los que compartí experiencias a Ulises, Aldo, Yusdel, Carlos Arce, Rosa, Arcadio, José Augusto, Kenier, Uffo, Liurver, Yordanis Rodríguez.

A todos los que de una forma u otra han compartido conmigo en todas las etapas de estudio, y en esta universidad a personas muy especiales como son Yan que es como mi hermano, aunque no pueda estar aquí está presente en mi corazón, al flaco Yasmany que aunque se graduó y se perdió hoy le doy gracias porque compartimos juntos muchas cosas malas y buenas y esa tesis yo sé que él hubiese querido ser parte de ella, a otra persona que sé que si no menciono me mata, al Fide que desde que lo conocí puedo decir que es una de las mejores personas que he conocido en esta universidad.

A todos mis compañeros de grupos del 8106, 4303 y del 5503, que me hicieron sentir muy bien junto a ellos todos los años de la universidad. A todos los profesores que tuve a lo largo de los años que estudié en la UCI, a Lisandra Guibert, a Antonio el vicedecano y a toda mis amigas las secretarias que siempre han estado conmigo de forma incondicional, a mi gordita Elibetsy, a Leyvis, a Nivia y a Nilda, a todas las quiero mucho.



A todos mis amigos el negro Marlon, Osmel, Sol, William DJ, que los quiero mucho, a un amigo que también significó mucho para mí y que siempre ha estado conmigo apoyándome en los momentos malos y buenos a Manuel (manolo).

A todas mis amigas a Lisbet Alvarez, Rosalía, Roselía, Mara, Iraini, Maydalís. En esta universidad he conocido personas excepcionales y he tenido amigas que nunca olvidaré, que para mí no son amigas son hermanas, Alice, Olga, Edelin, Licet, Liena, a todas las quiero y les agradezco por formar parte de mi vida porque siempre han estado conmigo para reír, para llorar para todo, y de verdad que no tengo palabras para decírles todo lo que quisiera.

Agradezco a mi tutor Arcadio porque independientemente de la tesis siempre me ha apoyado y ayudado desde que entré en esta universidad y siempre hemos compartido muchas cosas juntas.

A todos los que de una forma u otra estuvieron pendiente a mi tesis, y me preguntaban, cómo va la tesis, a todos muchas gracias.



Resumen

La Central de Trabajadores de Cuba es la organización de masas que agrupa, organiza y representa a los trabajadores del país. La Universidad de las Ciencias Informáticas cuenta con un buró sindical, el cual está compuesto por varias esferas entre las que se encuentran: Finanza y Emulación. Por la necesidad que existe de tener mayor control y que el manejo de la información de los procesos que se generan en estas esferas sea más ágil, se decide realizar una herramienta informática; para esto se hace un estudio de los procesos a informatizar así como de herramientas similares seleccionando algunas funcionalidades para la propuesta de solución, agregándole las necesidades del cliente. Se seleccionaron a través del estudio de tendencias de desarrollo *web*: las herramientas, metodología y tecnologías para el desarrollo de la solución propuesta. El proceso de desarrollo de *software* fue guiado por la metodología Programación Extrema, generando los artefactos propios así como toda la documentación necesaria como evidencia del proceso de desarrollo y como base para futuras modificaciones en el sistema. Se desarrollaron pruebas de calidad que permitieron validar la funcionalidad del sistema y corregir deficiencias propias de la programación. La propuesta de solución contribuirá a un mejor control y manejo de los procesos de las esferas de Finanzas y Emulación, por parte de los dirigentes sindicales tanto de las Secciones Sindicales como del Buró de la universidad.

Palabras claves: emulación, finanzas, organizaciones de masas, sindicatos, sistema de gestión de información.



Índice

| | |
|---|----|
| Introducción..... | 1 |
| Capítulo 1 Fundamentación Teórica..... | 6 |
| 1.1. Introducción | 6 |
| 1.2. Conceptos Fundamentales | 6 |
| Gestión de la Información..... | 6 |
| Finanzas..... | 7 |
| Emulación..... | 8 |
| 1.3. Análisis de Soluciones Similares..... | 8 |
| Aportes del estudio de las aplicaciones a la propuesta de solución..... | 11 |
| 1.4. Metodologías de Desarrollo de <i>Software</i> | 12 |
| Programación Extrema (XP)..... | 12 |
| SCRUM | 13 |
| 1.5. Sistemas Gestores de Bases de Datos (SGBD) | 14 |
| MySQL | 14 |
| PostgreSQL..... | 15 |
| Selección del Gestor de Base Datos | 15 |
| 1.6. Servidor web Apache 2.2..... | 16 |
| 1.7. Entorno de Desarrollo Integrado (IDE)..... | 16 |
| Eclipse..... | 16 |
| Zend Studio..... | 16 |
| NetBeans | 17 |



| | |
|---|----|
| Selección del IDE | 17 |
| 1.8. Lenguajes de Programación | 17 |
| 1.8.1. Lenguaje del lado del cliente | 17 |
| JavaScript..... | 18 |
| Las Hojas de Estilo en Cascada (CSS 3) | 18 |
| XHTML | 18 |
| 1.8.2. Lenguaje del lado del servidor | 19 |
| PHP | 19 |
| Páginas de Servidor Java (JSP) | 20 |
| Selección del Lenguaje de Programación..... | 20 |
| 1.9. Sistemas Gestores de Contenidos (CMS)..... | 20 |
| Joomla..... | 21 |
| WordPress..... | 21 |
| Drupal..... | 21 |
| Selección del CMS a utilizar | 22 |
| 1.10. Conclusiones parciales | 23 |
| Capítulo 2 Propuesta de solución..... | 24 |
| 2.1. Introducción | 24 |
| 2.2. Propuesta de sistema..... | 24 |
| 2.3. Aspectos no funcionales del sistema..... | 24 |
| Usabilidad..... | 25 |
| Disponibilidad | 25 |



| | |
|--|----|
| Seguridad | 25 |
| Confidencialidad | 25 |
| 2.4. Usuarios del Sistema | 26 |
| 2.5. Fase de Exploración | 27 |
| Historias de Usuarios | 27 |
| 2.6. Fase de Planificación | 29 |
| Plan de Iteraciones | 29 |
| Plan de Duración de las Iteraciones | 30 |
| Estimación de esfuerzos por historia de usuario | 31 |
| Plan de Entregas | 32 |
| 2.7. Diseño | 34 |
| Tarjetas Contenido, Responsabilidad y Colaboración (CRC) | 34 |
| 2.8. Conclusiones Parciales | 35 |
| Capítulo 3 Implementación y pruebas | 37 |
| 3.1. Introducción | 37 |
| 3.2. Implementación | 37 |
| Tareas de Ingeniería | 37 |
| 3.3. Estructura de Drupal | 40 |
| 3.4. Creación de las tablas de la base de datos | 42 |
| 3.5. Módulos desarrollados para el Sistema de Gestión Sindical de la UCI | 42 |
| 3.6. Patrones de diseño utilizados | 44 |
| 3.7. Instalación de los módulos | 45 |



| | |
|--|----|
| 3.8. Interfaz de usuario de los módulos..... | 45 |
| 3.9. Pruebas..... | 47 |
| Pruebas unitarias..... | 47 |
| Pruebas de aceptación..... | 48 |
| Resultado de las pruebas..... | 52 |
| 3.10. Validación de la investigación..... | 53 |
| 3.11. Conclusiones parciales..... | 54 |
| Conclusiones generales..... | 55 |
| Recomendaciones..... | 56 |
| Bibliografía..... | 57 |
| Glosario de términos..... | 61 |
| Anexos..... | 62 |
| Anexo 1. Fases de XP..... | 62 |
| Anexo 2. Historias de usuario..... | 62 |
| Anexo 3. Tareas de ingeniería..... | 69 |
| Anexo 4. Estructura de capas y datos de Drupal..... | 77 |
| Anexo 5. Tablas de base datos..... | 78 |
| Anexo 6. Interfaces de Usuario..... | 79 |
| Anexo 7. Casos de prueba..... | 87 |
| Anexo 8. Preguntas para la encuesta de validación de la investigación..... | 95 |

Índice de figuras

| | |
|---|-----------|
| Figura 1: Ubicación de los módulos en la estructura de archivos de Drupal..... | 43 |
|---|-----------|



| | |
|--|----|
| Figura 2: Prototipo de interfaz de usuario correspondiente a crear nueva actividad | 46 |
| Figura 3: Prototipo de interfaz de usuario correspondiente a gestionar trabajadores | 47 |
| Figura 4: Resultado de las pruebas de aceptación..... | 52 |
| Figura 5. Estructura de capas y datos de Drupal..... | 78 |
| Figura 6: Prototipo de interfaz de usuario correspondiente a adicionar trabajador de forma manual | 79 |
| Figura 7: Prototipo de interfaz de usuario correspondiente a dar baja de una SS. | 79 |
| Figura 8: Prototipo de interfaz de usuario correspondiente a eliminar trabajador. | 80 |
| Figura 9: Prototipo de interfaz de usuario correspondiente a bajas | 80 |
| Figura 10: Prototipo de interfaz de usuario correspondiente a editar SS..... | 81 |
| Figura 11: Prototipo de interfaz de usuario correspondiente a hacer cambio de escala..... | 82 |
| Figura 12: Prototipo de interfaz de usuario correspondiente a modificar datos de usuario..... | 82 |
| Figura 13: Prototipo de interfaz de usuario correspondiente al potencial financiero | 83 |
| Figura 14: Prototipo de interfaz de usuario correspondiente a actualizar estado financiero..... | 83 |
| Figura 15: Prototipo de interfaz de usuario correspondiente a reporte del pago CTC | 84 |
| Figura 16: Prototipo de interfaz de usuario correspondiente a reporte del pago Día de la Patria..... | 85 |



| | |
|--|----|
| Figura 17: Prototipo de interfaz de usuario correspondiente a ver estado financiero | 85 |
| Figura 18: Prototipo de interfaz de usuario correspondiente a ver reporte del estado financiero | 85 |
| Figura 19: Prototipo de interfaz de usuario correspondiente a ver actividad | 86 |
| Figura 20: Prototipo de interfaz de usuario correspondiente a ver trazabilidad de actividades | 86 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Comparación entre las metodologías XP y SCRUM | 13 |
| Tabla 2: Descripción de usuarios del sistema | 26 |
| Tabla 3: HU Gestionar trabajador | 28 |
| Tabla 4: HU Generar potencial financiero | 28 |
| Tabla 5: Plan de duración de las iteraciones | 30 |
| Tabla 6: Estimación de esfuerzos por historia de usuario | 31 |
| Tabla 7: Plan de entregas | 33 |
| Tabla 8: Tarjeta CRC 1 | 34 |
| Tabla 9: Tarjeta CRC 2 | 35 |
| Tabla 10: TI Generar potencial financiero | 37 |
| Tabla 11: TI Actualizar estado financiero de las SS | 38 |
| Tabla 12: TI Ver estado financiero CTC | 39 |
| Tabla 13: TI Ver estado financiero del Día de la Patria | 40 |
| Tabla 14: Caso de prueba realizado a la HU Gestionar trabajador | 49 |
| Tabla 15: Caso de prueba realizado a la HU Actualizar estado financiero | 50 |



| | |
|---|-----------|
| Tabla 16: Caso de prueba realizado a la HU Ver estado financiero | 51 |
| Tabla 18: HU Gestionar actividad | 62 |
| Tabla 19: HU Gestionar avisos | 63 |
| Tabla 20: HU Gestionar áreas | 63 |
| Tabla 21: HU Gestionar SS | 64 |
| Tabla 22: HU Dar baja del potencial | 64 |
| Tabla 23: HU Hacer cambio de escala..... | 65 |
| Tabla 24: HU Ver estado financiero | 65 |
| Tabla 25: HU Actualizar estado financiero..... | 66 |
| Tabla 26: HU Generar reportes financieros | 66 |
| Tabla 27: HU Listar actividades próximas | 67 |
| Tabla 28: HU Generar reportes emulación..... | 67 |
| Tabla 29: HU Ver trazabilidad de actividades | 68 |
| Tabla 30: HU Modificar datos usuario | 68 |
| Tabla 31: TI Adicionar nuevo trabajador | 69 |
| Tabla 32: TI Modificar datos del trabajador | 69 |
| Tabla 33: TI Eliminar trabajador | 70 |
| Tabla 34: TI Mostrar trabajador nuevo | 71 |
| Tabla 35: TI Adicionar áreas | 71 |
| Tabla 36: TI Modificar áreas..... | 72 |
| Tabla 37: TI Mostrar áreas | 72 |
| Tabla 38: TI Eliminar áreas | 73 |



| | |
|---|-----------|
| Tabla 39: TI Dar baja a trabajadores del potencial | 74 |
| Tabla 40: TI Hacer cambio de escala en el pago de los trabajadores | 74 |
| Tabla 41: TI Generar reportes financieros | 75 |
| Tabla 42: TI Generar Reportes Emulación | 75 |
| Tabla 43: TI Modificar datos de usuario autenticado | 76 |
| Tabla 44: TI Ver trazabilidad de actividades | 76 |
| Tabla 45: CP Gestionar áreas | 87 |
| Tabla 46: Gestionar SS..... | 88 |
| Tabla 47: Gestionar actividad | 88 |
| Tabla 48: Gestionar avisos | 89 |
| Tabla 49: Dar baja del potencial | 90 |
| Tabla 50: Hacer cambio de escala..... | 91 |
| Tabla 51: Generar reporte financiero | 91 |
| Tabla 52: Generar reporte emulación..... | 92 |
| Tabla 53: Modificar datos de usuario..... | 93 |
| Tabla 54: Listar próximas actividades | 93 |
| Tabla 55: Ver trazabilidad de actividades | 94 |



Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) tienen un papel creciente en las diferentes esferas de la sociedad, su auge ha estado acompañado por un gran avance en la gestión de la información y el conocimiento. La evolución de las tecnologías de la información tiende a incorporar dentro de sus estructuras informáticas la gestión de procesos que generan gran número de información, en su mayoría en formato duro.

La Central de Trabajadores de Cuba (CTC) es la organización de masas que agrupa, organiza y representa a los trabajadores del país. La misma está compuesta por 16 sindicatos nacionales, entre los que se encuentra el Sindicato Nacional de Trabajadores de la Educación, la Ciencia y el Deporte (SNTCED), al cual pertenecen los trabajadores de la Universidad de las Ciencias Informáticas (UCI). El Buró Universitario Sindical UCI (BUS UCI) está compuesto por 11 personas, tiene 93 secciones sindicales (SS) que agrupan a más de 4700 trabajadores. La gestión de toda la información que generan las distintas esferas de trabajo sindical se hace muy engorrosa, debido a la complejidad de estos procesos, los que hasta la fecha se gestionan de forma manual.

En el trabajo sindical son muy importantes las esferas de Finanzas y Emulación, ambas necesitan de un robusto sistema de control para garantizar el correcto funcionamiento de las mismas y la eficiencia en todos los procesos que ellas generan. En la universidad este control se realiza mediante el trabajo de los activistas de finanzas y emulación a diferentes niveles, utilizando extensas tablas impresas y enviando información generalmente por correo electrónico.

Cada SS cuenta con un activista de finanzas que es el encargado de la gestión del proceso de cobro de las finanzas y todo lo que ella genera. Una vez al mes debe ir a la oficina de recaudación y hacer el depósito, actualizando el estado de cada trabajador de su SS en cuanto al pago de las Finanzas y el Día de la Patria. En el BUS UCI existe un miembro encargado de controlar este proceso en la universidad, pero no tiene la posibilidad de llevar el control de toda la información de la Finanza y el Día de la Patria



de las distintas SS, se hace complejo verificar a la hora de confeccionar los potenciales¹ que se hayan diseñado usando escalas correctas.

En la esfera de Emulación también se cuenta con un responsable que lleva el control de las actividades, pero no se controla de manera efectiva la realización de las actividades orientadas desde el BUS o las que se realizan en la propia SS. Al acumularse gran cantidad de actividades en la que los trabajadores de la SS participan, no se puede manejar fácilmente la relación de los trabajadores participantes en dichas actividades por los miembros del BUS UCI. Cuando se solicita alguna información referente a estos procesos, la recopilación se hace por correo haciéndolo engorroso y poco eficiente.

La universidad cuenta con el personal en cada SS encargado de estos procesos y la posibilidad de disponer de herramientas informáticas en función de los mismos. Sin embargo, no emplean las potencialidades de las TIC, puestas a su disposición para facilitar los procesos realizados en las esferas sindicales Finanza y Emulación del BUS UCI y sus SS. La situación descrita permite plantear el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión de la información de los procesos de Finanza y Emulación del BUS UCI?

Objetivo general: Desarrollar los módulos para la gestión de la información generada por los procesos que realizan las esferas de Finanza y Emulación del BUS UCI y de las SS, que permita agilizar el trabajo y tener mayor control de la información gestionada en dichos procesos.

El **objeto de investigación** lo constituyen los procesos de gestión de información de las organizaciones de masas.

El **campo de acción** se centra en los procesos de gestión de información de organizaciones de masas, particularizado en las esferas Finanzas y Emulación del BUS UCI.

¹ Tabla que relaciona a todos los afiliados de una SS con sus respectivos salarios y cuotas a cotizar.



Objetivos específicos:

- ❖ Elaborar el marco teórico de la investigación.
- ❖ Seleccionar a partir del análisis de las tendencias de herramientas, tecnologías y metodologías más utilizadas en el desarrollo de este tipo de sistemas, las que se utilizarán en la solución que se propone.
- ❖ Elaborar la propuesta de solución a partir de los módulos de gestión de información que cumpla con los requerimientos definidos.
- ❖ Validar la solución propuesta a través de pruebas de aceptación y unitarias a los módulos obtenidos.

Como **hipótesis** de la investigación se tiene, que el desarrollo de los módulos Finanza y Emulación contribuirá a un mayor control por parte de los dirigentes sindicales de la UCI.

El presente estudio se desarrolla con el empleo de la combinación dialéctica de los Métodos Teóricos y Empíricos utilizados en la investigación. Entre los Métodos Científicos Teóricos se encuentran:

- ❖ **Método Analítico-Sintético:** el empleo de este método se evidencia cuando se realiza un análisis de toda la teoría y documentación actual, que permiten la extracción de los elementos fundamentales relacionados con el objeto de estudio, para la elaboración del marco conceptual que sustenta la investigación.
- ❖ **Método Inductivo-Deductivo:** se empleó durante todo el proceso investigativo con el objetivo de prever y premeditar posibles causas, consecuencias y resultados de los elementos que relacionan los procesos de la investigación.

Entre los Métodos Empíricos se empleó:

- ❖ **Entrevista de tipo abierta** (fundamentalmente al cliente): utilizado para obtener información verbal y real de los procesos sindicales a informatizar, para delimitar las funcionalidades a desarrollar.



Para dar cumplimiento a los objetivos trazados se definieron las siguientes **tareas de investigación**:

- ❖ Análisis de los fundamentos teóricos y metodológicos que sustentan la propuesta de solución.
- ❖ Estudio de soluciones similares existentes a nivel nacional e internacional.
- ❖ Selección de la metodología y herramientas de desarrollo de *software* a utilizar para la construcción de la propuesta de solución.
- ❖ Definición de las funcionalidades que deben cumplir los módulos.
- ❖ Definición de las prioridades de las funcionalidades identificadas.
- ❖ Selección de los patrones de diseño y arquitectónicos a utilizar en la solución.
- ❖ Desarrollo de la solución propuesta al problema planteado.
- ❖ Realización y análisis de pruebas que se definan y sean necesarias para garantizar el cumplimiento de los objetivos.
- ❖ Corrección de las no conformidades detectadas como resultado de las pruebas aplicadas.

Variables

Independientes:

- ❖ Módulo de Finanzas del BUS UCI.
- ❖ Módulo de Emulación del BUS UCI.

Dependientes:

- ❖ Mayor control de los procesos en cuestión.

El documento está organizado en 3 capítulos de la siguiente forma:

Capítulo 1: Fundamentación Teórica.



En este capítulo se abordan los conceptos necesarios para sustentar la investigación realizada. Se realiza un estudio del arte referente a sistemas similares que puedan contribuir a solucionar el problema planteado así como un estudio de las tendencias, herramientas, metodologías y tecnologías que permiten seleccionar las que faciliten el desarrollo de la solución.

Capítulo 2: Propuesta de solución.

En este capítulo se expone la propuesta de solución al problema antes planteado. Se inicia el desarrollo de la solución propuesta haciendo énfasis en las fases iniciales de la metodología de *software* seleccionada para sustentar el proceso así como la definición de las funcionalidades, diseño y estructuración de la misma. Al finalizar el capítulo queda determinada la estimación de tiempo de construcción, así como las entregas al cliente del producto.

Capítulo 3: Implementación y pruebas.

En el tercer capítulo, se plantea el proceso seguido para la construcción e implementación de la herramienta, haciendo énfasis en las características esenciales de la misma y la forma de dar cumplimiento a las funcionalidades descritas. Se realiza conjuntamente con los usuarios finales pruebas como forma de validación de la solución desarrollada, con el fin de contar con todas las evidencias necesarias que garanticen el cumplimiento en funcionalidad, calidad y alcance esperado por el cliente.

Al final del documento se incluyen las referencias bibliográficas, un glosario de términos y los anexos referenciados.



Capítulo 1 Fundamentación Teórica

1.1. Introducción

En el presente capítulo se elabora la fundamentación teórica de la investigación, realizando un análisis de los principales conceptos relacionados con el problema existente. Se lleva a cabo un estudio de algunos sistemas similares existentes vinculados al objeto de estudio. Finalmente se describe el análisis de las herramientas, lenguajes, tecnologías y metodologías a emplear en la creación de la propuesta de solución.

1.2. Conceptos Fundamentales

Para el desarrollo y mayor entendimiento de la presente investigación, es necesario puntualizar algunos conceptos fundamentales que son referidos en el marco metodológico.

Gestión de la Información

En la bibliografía consultada se define la gestión de la información con diferentes enfoques, según el concepto establecido por la Universidad de California un Sistema Gestión Información (SGI) es la aplicación de la tecnología de la información para apoyar las principales funciones y actividades de un negocio. De igual forma apoyan el proceso de recolección, manipulación, almacenamiento, distribución y utilización de los recursos de información de una organización. (1)

En el Instituto Tecnológico de Sonora se define como "... un conjunto de componentes relacionados que recolectan, procesan, almacenan y distribuyen información para apoyar a la toma de decisiones y el control de una organización". (2)

La autora de esta investigación coincide con la Lic. Lourdes Aja Quiroga que define la gestión de la información como "*... parte esencial de la infraestructura para la gestión del conocimiento, suministran información, impulsan la generación del conocimiento para la búsqueda de soluciones a los problemas que enfrentan las organizaciones, analizan su impacto sobre los resultados de las empresas e influyen en el comportamiento de los*



individuos ante la información” en otro momento refleja que “La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas de las compañías en materia de desempeño y de calidad”. (3)

Para el desarrollo de la presente investigación, también se hizo necesario analizar otros conceptos relacionados con el dominio del problema entre los que figura el de finanzas y emulación.

Finanzas

Alfonso Ortega Castro define a las Finanzas como: *“La disciplina que, mediante el auxilio de otras, tales como la contabilidad, el derecho y la economía, trata de optimizar el manejo de los recursos humanos y materiales de la empresa, de tal suerte que, sin comprometer su libre administración y desarrollo futuros, obtenga un beneficio máximo y equilibrado para los dueños o socios, los trabajadores y la sociedad”. (4)*

Los estatutos de la afiliación a los sindicatos, derechos y deberes de los afiliados plantean en el artículo 6 inciso c, que *“Todo afiliado a una sección sindical tiene el deber de abonar y mantener al día el pago de la cuota sindical.”*

El Manual de Normas y Procedimientos para el trabajo de los Centros de Recaudación del movimiento sindical precisa en el Capítulo 4 lo siguiente:

“La Cotización de los afiliados es un principio organizativo establecido en los Estatutos y en los Reglamentos Sindicales que viabilizan su aplicación y refleja un profundo sentido político, no se efectúa por presiones organizativas o de otra índole que violenten de alguna forma la motivación individual del afiliado. No forma parte de los indicadores de la emulación, ni colectiva ni individualmente, ni en ningún nivel de dirección sindical.”

Mantener al día la cotización sindical significa que el afiliado debe, desde el mes de enero, abonar puntualmente lo establecido en la escala acordada según el ingreso total que recibe. Tal definición implica que sistemáticamente las organizaciones de base y los



organismos de dirección deben relacionarse con los trabajadores para poder transmitir e intercambiar sobre los resultados que se van alcanzando en las diferentes tareas, sin olvidar las relacionadas con la eficiencia financiera.

Emulación

Los estatutos de la afiliación a los sindicatos, derechos y deberes de los afiliados plantean en el artículo 6 inciso f, que *“Todo afiliado a una sección sindical tiene el deber de participar en las actividades enmarcadas dentro de la Emulación Socialista.*

En la universidad existe un sistema de emulación sindical en que se relacionan todas las actividades que del BUS se planifican, emulando en ellas todas las SS. Cada SS puede implementar su propio sistema de emulación interna en el que se recogen todas las actividades convocadas, por el buró o la administración, y la participación de los afiliados en ellas. En el orden del día de las asambleas de afiliados existe un punto fijo: “Proceso político por una educación de calidad”, en el que se reconoce mensualmente a los trabajadores que más destaquen entre sus compañeros, trimestralmente se realiza un corte de emulación y anualmente un balance de emulación, en los que se reconocen las SS más destacadas.

Al finalizar el curso escolar se seleccionan a los Educadores Ejemplares y Trabajadores Destacados por la Calidad, reconocimientos a la labor como educadores y seleccionados por sus propios compañeros. (5)

1.3. Análisis de Soluciones Similares

Para la propuesta de solución se hizo necesario un análisis de la documentación y organizaciones existentes a nivel nacional e internacional, referente al desarrollo de aplicaciones de sistemas de gestión de información que permitan agilizar el trabajo y tener mayor control de la información gestionada en dichos procesos. Después de realizado el estudio de la bibliografía se pudo obtener algunos sistemas informáticos destinados a la gestión de la información en organizaciones sindicales, entre los que se encuentran los siguientes:



Sindicato Único de Admisión y Permanencia de Trabajadores de la República de Argentina

Este sistema de gestión de información lleva el control de todo lo relacionado con una organización de masas en Argentina donde, mediante esta aplicación web se maneja el proceso de afiliación de los trabajadores. Recoge aspectos como son las asignaciones que se refieren a un pago de una suma fija, que puede ser mensual o por única vez, que es abonado al trabajador ante determinadas circunstancias de su vida: cuando se casa, cuando espera un bebé, durante el período de licencia por maternidad, cuando nace un hijo o decide adoptar uno o cuando su hijo va a la escuela. Mediante este sistema efectúa los controles de derecho sobre las asignaciones familiares de los trabajadores y las hace efectivas, en caso de corresponder. (7)

Dicho sistema fue analizado, pero se encontró que no es útil para darle solución al problema planteado ya que no tiene ningún vínculo directo con las finanzas o con la emulación. Se puede reflejar el trabajo que hace para dicha organización de masas pero en forma general no maneja la información real del estado financiero de los trabajadores, no realiza reportes, no controla de ninguna forma las actividades en las que participan los trabajadores así como ninguna información relacionada con ese aspecto.

Gestión Integral para Sindicatos de la República de Argentina

Este *software* permite cubrir las necesidades de información en cuanto a gestión de beneficiarios (afiliados y cargas de familia), empresas adheridas, obra social (gestión de turnos, bonos de atención médica), agenda de contactos, liquidación de sueldos y gestión administrativa: inventario de bienes, salida de correspondencia, órdenes de pago y servicios especiales prestados a los afiliados. Este *software* internacional no se puede extrapolar, ya que las características de la organización en el socialismo cubano difieren del funcionamiento sindical en otros sistemas sociales. No se evidenció tratamiento de una emulación sindical, no hay registros de actividades por lo que una parte importante del problema no sería resuelta con esta aplicación.



Herramienta web para automatizar la gestión de la información y la toma de decisiones en el Sindicato de Administración Pública

En el ámbito nacional, en la UCI se desarrolló una herramienta web para automatizar la gestión de la información y la toma de decisiones en el Sindicato de Administración Pública. La misma permite informatizar todo el proceso de gestión de la información relacionada con los modelos que son generados en cada una de las estructuras del sindicato, desde el municipio hasta la máxima instancia, que es la CTC Nacional. Este modelo es la vía para conocer toda la información de los trabajadores del país, donde se refleja el total de centros de trabajos, secciones sindicales y otros datos referidos a la organización. Constituye el modelo fundamental de datos del movimiento obrero, y su elaboración data desde casi la propia creación de la CTC en Cuba, aunque no en la forma que se realiza actualmente.

A pesar de ser una solución cubana, contempla la informatización de procesos propios del Sindicato de la Administración Pública no incluyendo las esferas de Finanzas y Emulación, por lo que no se puede utilizar para solucionar el problema de la investigación.

Una solución informática para el control de las finanzas de las secciones sindicales de la Universidad de Las Tunas

En la Universidad de Las Tunas “Vladimir Ilich Lenin”, en el curso 2011-2012, se desarrolló un *software* titulado “Una solución informática para el control de las finanzas de las secciones sindicales de la Universidad de Las Tunas”, encaminado al control de las finanzas y los reportes que de este se derivan, pero no cumplen con la totalidad de las necesidades de esta investigación, ya que se necesita, además de las finanzas, garantizar todo el proceso realizado dentro de la emulación. Esta aplicación se limita solo a controlar y hacer reportes.

La herramienta que se analiza fue desarrollada como una aplicación de escritorio ya que las principales funcionalidades fueron diseñadas para el control de las finanzas solamente en la SS. No se toma como solución al problema planteado ya que en la UCI se necesita que exista comunicación constante entre las SS y el BUS.



Sistema informático para la Gestión de la Información en el Sindicato Nacional de Trabajadores de la Educación, la Ciencia y el Deporte (SNTECD) en la Universidad de Las Tunas “Vladimir Ilich Lenin”

Esta herramienta surge como mejora de algunas funcionalidades implementadas en “Una solución informática para el control de las finanzas de las secciones sindicales de la Universidad de Las Tunas”, con el objetivo de eliminar las deficiencias existentes. Entre los aspectos que se pueden señalar están:

- ❖ Existe desigual funcionamiento e inestabilidad en la permanencia de los dirigentes.
- ❖ Los dirigentes sindicales en su mayoría no están bien preparados.
- ❖ No se puede disponer de mucha información referente a procesos sindicales.
- ❖ La información con la que se cuenta en ocasiones no es fiable.

Con el desarrollo de este sistema se desea facilitar la gestión de la información del sindicato para alcanzar mejores resultados en el control y mayor efectividad en la toma de decisiones. Este *software* no se escoge como solución, ya que el BUS UCI no presenta problemas con el personal que se encarga de manejar todos los procesos. Se necesita que se aprovechen los recursos informáticos que se tienen para manejar estos procesos que son tan engorrosos, y esta propuesta sigue siendo desarrollada como una aplicación de escritorio.

Aportes del estudio de las aplicaciones a la propuesta de solución

Después del análisis de varios sistemas similares se pudo constatar que según las tendencias actuales los sistemas de gestión de información, de las organizaciones de masas, están encaminados al desarrollo web. Dichos sistemas no poseen funcionalidades que engloben los procesos de Finanzas y Emulación en un Buró Sindical, por esa razón son descartados y se decide realizar los módulos para la gestión de la información de estos procesos, sin embargo, los sistemas analizados aportan una serie de elementos significativos para la futura propuesta de solución tales como:



- ❖ Visualización de la información.
- ❖ Actualización de la información.
- ❖ Análisis de la forma en la que generan los reportes.

1.4. Metodologías de Desarrollo de Software

Existen dos grandes enfoques en lo que respecta a las metodologías de desarrollo, el tradicional pensado en una serie de pasos y etapas que generan el uso exhaustivo de documentación durante todo el ciclo del proyecto y en el caso de las metodologías ágiles es de vital importancia la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y mantener una buena relación con el cliente. (6)

Teniendo en cuenta que el equipo de desarrollo es pequeño, el sistema a desarrollar no es complejo y se cuenta con poco tiempo para la implementación del mismo, se decide seleccionar una metodología de desarrollo de *software* ágil, entre los principales métodos ágiles se tiene a Programación Extrema (XP), *Scrum*, *Iconix*, *Cristal*, *Methods Ágil Unified Process* (AUP), se hizo un análisis de las metodologías XP y *Scrum* teniendo en cuenta los siguientes indicadores:

- ❖ Experiencia del equipo de desarrollo.
- ❖ Adecuada para equipos de desarrollos pequeños.
- ❖ Adecuada para realizar proyectos pequeños.
- ❖ Adecuada para agilizar el proceso de desarrollo.
- ❖ La documentación generada es suficiente para comprender todo el proceso de desarrollo.
- ❖ Flexibilidad ante los cambios.
- ❖ Necesaria una comunicación fluida entre el cliente y el equipo de desarrollo.

Programación Extrema (XP)

Es una metodología ágil para el desarrollo de software, se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo de manera que se pueda lograr, un



producto de calidad en poco tiempo. Se define especialmente como adecuada para proyectos con requisitos imprecisos y muy cambiantes. Simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (7)

Se basa en las pruebas realizadas a los principales procesos, de tal manera que en un futuro se puedan hacer pruebas de las fallas que pudieran ocurrir para así obtener los posibles errores. También añade funcionalidad con retroalimentación continua y no introduce funcionalidades antes que sean necesarias. Empieza en pequeño, el manejo del cambio se convierte en parte sustantivo del proceso y el costo del cambio no depende de la fase o etapa. El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. (7) **(Ver Anexo 1)**

SCRUM

Está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos, la innovación, la competitividad y la productividad son fundamentales. El desarrollo de *software* se realiza mediante iteraciones, denominadas *sprint*², con una duración de 30 días, se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Esta metodología tiene algunas desventajas, por ejemplo, genera muy poca documentación en comparación con otras metodologías, y en muchas ocasiones es necesario completarla con otros procesos de XP. (8)

Tabla 1: Comparación entre las metodologías XP y SCRUM (fuente: creado por la autora)

| Indicadores para la selección de la metodología | XP | SCRUM |
|---|----|-------|
|---|----|-------|

²Sprint: Iteración de duración de SCRUM prefijada durante la cual el equipo trabaja para convertir las historias a las que se ha comprometido, en una nueva versión del *software* totalmente operativo.



| | | |
|--|------|------|
| Experiencia del equipo de desarrollo | alta | baja |
| Adecuada para equipos de desarrollos pequeños | si | si |
| Adecuada para realizar proyectos pequeños | si | si |
| Adecuada para agilizar el proceso de desarrollo | si | si |
| La documentación generada es suficiente para comprender el proceso de desarrollo | si | no |
| Flexibilidad ante los cambios | si | si |
| Necesaria una comunicación fluida entre el cliente y el equipo de desarrollo | si | si |

Luego de realizar el análisis de estas metodologías, teniendo en cuenta los indicadores seleccionados para establecer la comparación entre ellas, se decide seleccionar XP para guiar todo el proceso de desarrollo de la solución que se propone.

1.5. Sistemas Gestores de Bases de Datos (SGBD)

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. Hay muchos otros servicios que habitualmente proporcionan los SGBD, entre ellos se encuentran los servicios de copia de seguridad, la compresión de datos o la flexibilidad en añadir nuevos tipos de datos. (9)

MySQL

Es un sistema de bases de datos relacional, multihilo y multiusuario, es rápido en la lectura pero puede provocar problemas de integridad en entornos de alta concurrencia en



la modificación. Funciona en diferentes plataformas, en todos los navegadores modernos que tengan un compilador de C++. Es relativamente sencillo añadir otro sistema de almacenamiento, esto es útil si desea añadir una interfaz SQL para una base de datos propia. Está escrito en C y C ++ pero tiene la desventaja de ser una herramienta propietaria. (10)

PostgreSQL

Soporta el almacenamiento de objetos binarios grandes (gráficos). Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. Es fácil de administrar, su sintaxis SQL es estándar y fácil de aprender, y es multiplataforma. La documentación está muy bien organizada, pública y libre, con comentarios de los propios usuarios. Altamente adaptable a las necesidades del cliente. Soporte nativo para los lenguajes más populares del medio como PHP.

Las características que más se han tenido en cuenta desde su creación han sido, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. PostgreSQL es un SGBD objeto-relacional, funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Utiliza un modelo cliente/servidor y usa multiprocesos. Utiliza numerosos tipos de datos y da la posibilidad de definir nuevos tipos. (11)

Selección del Gestor de Base Datos

Se decide utilizar PostgreSQL en su versión 9.1 como SGBD ya que permite alcanzar escalabilidad ajustando el número de procesadores y la cantidad de memoria, es capaz de soportar una mayor cantidad de peticiones simultáneas, es un gestor multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso. Mientras que MySQL presenta características muy similares pero es propietario lo que dificulta obtener su licencia.



1.6. Servidor web Apache 2.2

Para el desarrollo de la aplicación será utilizado Apache HTTP Server como servidor web. Es una tecnología gratuita de código fuente abierta. Es un servidor altamente configurable de diseño modular y trabaja con PHP. Es configurable, multiplataforma y fácil de usar. Posee abundante documentación, permitiendo a cualquier usuario implementar módulos para una función específica. Para la instalación de varios *framework* como Drupal es necesario tener instalado un servidor web, pero se recomienda la utilización de Apache para obtener un mejor rendimiento y soporte de la comunidad. (12)

1.7. Entorno de Desarrollo Integrado (IDE)

Los Entornos de Desarrollo Integrado de *software* (IDE por sus siglas en inglés), son herramientas que ayudan a los programadores a desarrollar *software* de forma más amigable. Es decir, aquellos en los que el programador puede acceder con el menor esfuerzo a diferentes recursos como editores, compiladores y herramientas de análisis. (13)

Eclipse

Es un IDE de código abierto, mediante un sistema de *plugins* se le puede añadir nuevas funcionalidades a la aplicación; entre ellas soportes adicionales para programación en otros lenguajes, por ejemplo, PHP Eclipse, que proporciona un entorno cómodo para el desarrollo de aplicaciones PHP. Otro lenguaje es Subclipse, que le ofrece una interfaz aceptable para trabajar con proyectos remotos colaborativos mediante subversión. (14)

Zend Studio

Zend Studio es un Ambiente Integrado de Desarrollo (IDE PHP) disponible para los desarrolladores profesionales que ofrecen las capacidades necesarias para desarrollar aplicaciones de negocio. Posee características como refactorización, generación de código, asistente de código y análisis semántico. Permite implementar, sin problemas de código en la nube, mediante un entorno de ejecución Zend servidor diseñado para desarrolladores. Sin embargo es descartado por ser *software* propietario. (15) (16)



NetBeans

Al IDE NetBeans es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores en todo el mundo. Permite fácilmente y muy rápido desarrollar aplicaciones web, es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar sus programas. (17)

Selección del IDE

Se selecciona NetBeans en su versión 7.3 por todas las características que este posee, entre ellas, se puede decir que es más fácil de utilizar, tiene versiones en español, utiliza plataformas libres, permite integrarse con subversión, permite generar la documentación del código fuente y se puede programar en PHP. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código. Este IDE ofrece formato de código acorde a los estándares de código de Drupal, y reconocimiento a los principales tipos de archivos específicos de este CMS, tales como (*.info*, *.module* *.install*).

1.8. Lenguajes de Programación

Un lenguaje de programación consiste en un conjunto de órdenes o comandos, reglas, símbolos y palabras especiales que permiten construir un programa y describir el proceso deseado. Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que se combinan para formar los programas de cómputo. Un lenguaje de programación es un lenguaje especial, no natural, diseñado con un vocabulario, morfología y sintaxis muy simples, rígidas y orientadas a la programación de instrucciones elementales cuya ejecución por un determinado sistema físico da lugar a la realización de una tarea. Los lenguajes de programación no son aplicaciones, sino herramientas que permiten construir y adecuar aplicaciones. (18)

1.8.1. Lenguaje del lado del cliente

Un lenguaje del lado del cliente es totalmente independiente del servidor, están incluidos dentro de las páginas HTML, lo cual permite que la página pueda ser albergada en cualquier sitio y se ejecutan dentro del ámbito del buscador. (19)



JavaScript

Es un lenguaje abierto, se puede programar sin necesidad de crear clases, tal como se realiza en los lenguajes de programación estructurada. Posee una sintaxis semejante a la del lenguaje Java y C, aunque a diferencia de éstos no requiere compilación. Una de sus funciones es la de crear contenidos dinámicos, elementos de la página que tengan movimiento, cambien de color, crear efectos especiales en las páginas o definir interactividad con el usuario. Sus programas se interpretan en forma directa por el navegador. (14) (20)

Las Hojas de Estilo en Cascada (CSS)

Las Hojas de Estilo en Cascada (CSS por sus siglas en inglés) son un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla o cómo se va a imprimir. La forma que utiliza para la descripción de estilos, ofrece a los desarrolladores el control total sobre el estilo y el formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación, definen la forma de mostrar los elementos y permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas al mismo tiempo. CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. (21)

XHTML

XHTML es un lenguaje extensible de marcado de hipertexto (*Extensible Hypertext Markup Language*). Es el lenguaje de programación pensado para sustituir a HTML. XHTML es la versión XML de HTML con las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. (22)

Para la implementación se ha decidido utilizar JavaScript en su versión 1.8.5 porque es un lenguaje de programación sencillo y pensado para hacer las cosas con rapidez. Se resalta la compatibilidad con la mayoría de los navegadores modernos. Se utilizará para la implementación de algunos aspectos dinámicos como por ejemplo la validación de



formularios y la creación de ventanas emergentes con el propósito de mostrar algún tipo de información. El lenguaje CSS 3 permite controlar el estilo a utilizar de las páginas y define la forma de mostrar los elementos.

1.8.2. Lenguaje del lado del servidor

La programación web del lado del servidor es una tecnología en la cual una petición del usuario es atendida por la ejecución directa en el servidor web de una aplicación, para generar páginas web dinámicas. Es comúnmente usada para dar interactividad a los sitios web que se conectan a bases de datos u otros almacenes de datos. Una de las principales ventajas de la programación del lado del servidor es la habilidad de preparar la respuesta basada en los requerimientos del usuario, derechos de acceso, o consultas en la base de datos.

Por lo tanto es el servidor el que maneja toda la información de las bases de datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página web con los resultados de todas las operaciones. Para escribir páginas dinámicas del lado del servidor existen varios lenguajes de programación, como son: *Common Gateway Interface (CGI)* comúnmente escritos en Perl, *Active Server Pages (ASP)*, *Hipertext Preprocessor (PHP)*, y *Java Server Pages (JSP)*. (23)

Para la investigación se analizaron con mayor profundidad los lenguajes PHP y JSP por ser de los más utilizados y por el desarrollar tener conocimiento previo de ellos.

PHP

PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo de páginas web dinámicas y que puede ser incrustado en HTML. Está enfocado principalmente a la programación de scripts del lado del servidor. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas, es gratuito y rápido, con una gran librería de funciones y mucha documentación. (14) (24) (19)



Páginas de Servidor Java (JSP)

Es un lenguaje script, basado en la tecnología Java. No es necesario que el usuario disponga de la máquina virtual de Java, ya que esta se encuentra en el servidor que crea las páginas HTML. (19) (23)

Selección del Lenguaje de Programación

Se ha escogido PHP en su versión 5.4.16 principalmente por su condición de *software* libre, así como por la amplia documentación que existe sobre este lenguaje. También se destaca su soporte para la programación orientada a objetos y su manejo de excepciones. La capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad como es el caso de PostgreSQL que es el que se utilizará en la propuesta de solución.

1.9. Sistemas Gestores de Contenidos (CMS)

Los Sistemas de Gestión de Contenidos (*Content Management Systems* o CMS) son herramientas que se utilizan principalmente para crear y mantener con facilidad la gestión de páginas webs. Para la creación del sitio propiamente dicho, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente.

Los CMS proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios. Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado. Puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización le permite adaptarse al idioma, sistema de medidas y cultura del visitante. (25)



Joomla

Joomla es un CMS de código abierto construido con PHP. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL. Tiene características como: hacer caché de páginas para mejorar el rendimiento, indexado web, versiones imprimibles de páginas, flash con noticias, blogs, foros, calendarios, búsqueda en el sitio web e internacionalización del lenguaje.

Está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel administrador de Joomla se podrá crear, editar y borrar las secciones y categorías del sitio de la manera en que más se desee. (26)

WordPress

WordPress es un sistema de gestión de weblogs desarrollado con PHP y MySQL, bajo licencia GPL, se ha convertido en un *software* muy popular por su licencia, su facilidad de uso y sus características como gestor de contenidos. Algunas de sus características fundamentales son: separa el contenido y el diseño en XHTML y CSS, posee una fácil instalación, actualización y personalización. (27)

Drupal

Drupal es un CMS modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web y un énfasis especial en la usabilidad y consistencia de todo el sistema.

Su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web. Es un sistema de ayuda en línea y páginas de ayuda para los módulos, tanto para usuarios como para administradores. Todo el contenido en Drupal es totalmente indexado



en tiempo real y se puede consultar en cualquier momento. Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo. Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles.

El sistema de control de versiones permite seguir y auditar totalmente las sucesivas actualizaciones del contenido. También permite mantener comentarios o eliminar cambios recuperando una versión anterior. Todo el contenido creado en Drupal tiene un enlace permanente asociado a él, sin temor de que el enlace falle en el futuro.

Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones, por ejemplo incorpora soporte para otras bases de datos, entre ellas PostgreSQL. Este CMS ha sido diseñado desde el principio para ser multiplataforma, puede funcionar con Apache como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X, al estar implementado en PHP, es totalmente portable.

La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún *software* adicional. Toda la actividad y los sucesos del sistema son capturados en un registro de eventos, que puede ser visualizado por un administrador. (28)

Selección del CMS a utilizar

Después de analizar varios de los CMS conocidos y utilizados en la universidad, se escoge Drupal en su versión 7.26 por las características que presenta, que son de gran utilidad para la realización de la propuesta de solución, que no las tienen los demás CMS estudiados. Se pueden mencionar algunas como son la facilidad que tiene para manejar datos con el uso de las taxonomías, las vistas y los tipos de contenidos. Está también el uso de los hooks y de funciones que ya vienen predefinidas, también es bueno destacar que en la UCI existe una comunidad de usuarios de Drupal que tienen su propio sitio, en el cual se puede consultar tanto información referente a cualquier tema del CMS, como dar respuesta a las dudas existentes. En el Centro de Tecnologías para la Formación (FORTES), existe un grupo de desarrolladores capacitados en Drupal, por lo que sería



mucho más fácil el intercambio a la hora de aclarar alguna duda o necesitar algún complemento.

1.10. Conclusiones parciales

El estudio realizado a partir de los métodos de investigación científica permitió elaborar el marco teórico de la investigación; de esta forma se lograron definir los principales conceptos que complementan la investigación. Los sistemas similares estudiados permitieron obtener funcionalidades útiles para el desarrollo de la propuesta de solución. El estudio de las principales características de las metodologías, herramientas y lenguajes utilizados actualmente para el desarrollo de sistemas de gestión de información de las organizaciones de masas, permitió seleccionar las que se van a utilizar para darle cumplimiento al objetivo general.



Capítulo 2 Propuesta de solución

2.1. Introducción

Una vez realizado el análisis del estado del arte, elegidas las herramientas, lenguajes y la metodología de desarrollo a utilizar, se está en condiciones de plantear una solución al problema descrito.

En este capítulo se hace alusión a las fases de Exploración, Planificación y Diseño, pertenecientes a la metodología de desarrollo XP. El objetivo principal de estas es: conocer el alcance del producto a desarrollar, así como la estrategia a seguir para la implementación, adecuándose a los plazos de tiempo impuestos por los usuarios y estimar los tiempos de entrega de cada versión.

2.2. Propuesta de sistema

Para solucionar el problema identificado se decide realizar dos módulos, que posteriormente serán integrados al Sistema de Gestión Sindical de la UCI; que actualmente se ha desarrollado aunque no se encuentra publicado en espera de los módulos de Finanzas y Emulación. Con la creación de estos módulos se pretende primeramente resolver todos los problemas de gestión de información, con que cuenta dicha organización de masa. Organizar y almacenar de manera precisa toda la información y documentación, que se genera durante el proceso de pago de las Finanzas y la del Día de la Patria, y las actividades que se realizan en el BUS y las SS. Los usuarios en dependencia del rol que tengan podrán interactuar de manera efectiva para poder conocer el estado financiero o de emulación de su SS o de manera general.

2.3. Aspectos no funcionales del sistema

Cada sistema presenta distintas características no funcionales que, a la par de las funciones que realiza, lo haga marcar la diferencia entre un producto de alta aceptación y uno de poca aceptación, estas características en conjunto crean un sistema capaz de



proporcionarle al cliente un atractivo visual, usabilidad, rápido desempeño y que llene sus expectativas.

Usabilidad

El sistema debe presentar un acceso fácil y rápido a los componentes para facilitar el uso del mismo a los usuarios con pocos conocimientos en el campo de la informática. Debe tenerse en cuenta también el tiempo de respuesta de la aplicación a las acciones del usuario y se debe acceder a todas las funcionalidades con la menor cantidad de accesos a funcionalidades.

Disponibilidad

La aplicación siempre estará disponible para todos los usuarios que pertenecen a la red de la universidad, por lo que con solo autenticarse, siempre que los datos de autenticación sean correctos, podrá acceder al sistema.

Seguridad

Se plantea una aplicación cuya seguridad sea orientada a roles lo que permite asignar los permisos de distintas acciones a distintos tipos de roles y asignar a estos los usuarios que necesitará el sistema. Es necesario destacar que el usuario anónimo o no autenticado solo podrá acceder al formulario de acceso, solo los usuarios autenticados en el sistema serán capaces de hacer uso de las funcionalidad del sistema.

El sistema de autenticación se implementó haciendo uso del sistema de seguridad que brinda el *CMS Drupal* en su versión 7.26 el cual permite, a determinados roles acceder o no a distintas funciones que presentará la aplicación.

Confidencialidad

Una de las principales características de un sistema basado en roles es la confidencialidad: permite que un usuario no sea capaz de ver información a la que no tenga permiso.



2.4. Usuarios del Sistema

Los usuarios del sistema o usuarios clientes son personas que se conectan al sistema, para hacer uso de los servicios que este les proporciona. Obtienen un resultado a partir de los procesos que se ejecutan, incluyendo los que mantienen la aplicación actualizada y en correcto funcionamiento. Dentro de los usuarios del sistema se pueden distinguir diferentes roles, y dependiendo de dicho rol poseerá más o menos privilegios en su estancia dentro del sistema. También se debe añadir que un usuario dentro del sistema puede pertenecer a diferentes roles, siempre teniendo en cuenta las restricciones que existen para cada uno. En la presente investigación se definieron los siguientes usuarios:

Tabla 2: Descripción de usuarios del sistema

| Usuarios del sistema | Descripción |
|----------------------|--|
| Anónimo | Este usuario es el que navega dentro de la aplicación, sin tener una cuenta creada, puede acceder a informaciones públicas como historias, documentos, noticias, entre otras. |
| Autenticado | Es el usuario que ya teniendo una cuenta creada dentro de la aplicación puede realizar algunas acciones relacionadas como: ver los datos de su SS, datos personales, diferentes tipos de avisos, entre otras. |
| Financiero | Tiene todos los privilegios de un usuario autenticado y los suyos como financiero, puesto que puede gestionar toda la información de las finanzas, dígame actualizar, modificar, eliminar e incluir nuevas informaciones como: generar potencial, actualizar estado financiero, dar alta o baja, generar reportes entre otras. |
| Emulación | Tiene todos los privilegios de un usuario autenticado y los suyos como emulador, puesto que puede gestionar toda la información de la emulación dígame: actualizar, modificar, |



| | |
|----------------------|---|
| | eliminar e incluir nuevas informaciones como actividades de las SS, participantes, avisos sobre emulación, entre otros. |
| Secretario SS | Tiene los privilegios de los usuarios autenticados, los financieros y los de emulación, el mismo puede gestionar toda la información de la SS, tanto de finanzas como de emulación. Es el encargado de asignar el rol de financiero y de emulador en la SS. |
| Miembro BUS | Tiene los privilegios de los usuarios mencionados anteriormente, el mismo puede gestionar avisos de las diferentes esferas, publicar información, generar reportes, asigna roles tanto del Buró como de las SS. |
| Administrador | Es el que gestiona todo lo referente al funcionamiento de la aplicación. |

2.5. Fase de Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. (7)

Historias de Usuarios

Las Historias de Usuarios (HU) son la técnica utilizada para especificar los requisitos del *software*. Son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración. Se puede considerar que las HU en XP juegan un papel similar a los casos de uso en otras metodologías, pero en realidad son muy diferentes, solo muestran la silueta de una tarea a realizarse. Son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar las entregas, en las que se puede desarrollar una o varias HU, esto depende del tiempo que demore la implementación de cada una de las mismas. (7) (29)



A continuación se muestran algunas de las HU que representan las funcionalidades implementadas para la propuesta de solución. Las restantes se encuentran en los anexos. **(Ver Anexo 2)**

Tabla 3: HU Gestionar trabajador

| Historia de Usuario | |
|---|-------------------------------------|
| No: 3 | Nombre: Gestionar trabajador |
| Usuario: Financiero | |
| Prioridad: Alta | Estimación: 1 |
| Nivel de Complejidad: Alta | Iteración: 1 |
| Descripción: el usuario debe tener la posibilidad de gestionar la información de los trabajadores. Esto incluye almacenar información de los mismos (nombre, apellidos, área, teléfono, categoría científica y salario). | |
| Observación: gestionar incluye: adicionar, modificar, ver y eliminar los datos de cada trabajador. | |

Tabla 4: HU Generar potencial financiero

| Historia de Usuario | |
|----------------------------|---|
| No: 4 | Nombre: Generar potencial financiero |
| Usuario: Financiero | |
| Prioridad: Alta | Estimación: 0.3 |



| | |
|---|---------------------|
| Nivel de Complejidad: Alta | Iteración: 1 |
| Descripción: El usuario debe tener la posibilidad de gestionar el potencial para llevar el control del pago de las Finanzas y Día de la Patria, teniendo en cuenta cuánto debe pagar cada trabajador según su salario. | |
| Observación: Deben de haberse incluido los trabajadores y sus salarios respectivamente. Gestionar incluye: adicionar, modificar, ver y eliminar el potencial. | |

2.6. Fase de Planificación

En esta fase se establece la prioridad de cada una de las HU y se realiza una estimación de esfuerzos necesarios por parte del desarrollador. Se toman acuerdos en conjunto con el cliente para realizar el cronograma de entregas. (7)

Plan de Iteraciones

En el plan de iteraciones se especifica de manera detallada el orden de desarrollo de cada una de las HU. Al concluir cada iteración, se obtendrá como resultado la implementación de algunas de las funcionalidades que permitirán darle cumplimiento a los objetivos propuestos (30)

Iteración 1: Esta iteración enmarca las HU que tienen una complejidad alta, para el desarrollo de la aplicación. Las que se caracterizan por solicitarle datos al cliente para introducirlos en el sistema, siendo estas:

- ❖ Gestionar áreas
- ❖ Gestionar SS
- ❖ Gestionar trabajador
- ❖ Gestionar potencial financiero
- ❖ Gestionar actividad



- ❖ Gestionar avisos

Iteración 2: Esta iteración es para las HU restantes de prioridad alta que son importantes para el usuario, las cuales son:

- ❖ Dar baja del potencial
- ❖ Hacer cambio de escala
- ❖ Actualizar estado financiero
- ❖ Generar reporte financiero
- ❖ Generar reporte emulación
- ❖ Modificar datos de usuario

Iteración 3: Esta iteración tiene como objetivo la codificación de HU de prioridad baja, las que permiten al usuario solamente consultar la información, las cuales tendrán como funcionalidad:

- ❖ Ver estado financiero
- ❖ Listar actividades próximas
- ❖ Ver trazabilidad de actividades
- ❖ Buscar trabajador

Plan de Duración de las Iteraciones

En la metodología XP, la creación del sistema se divide en etapas para facilitar su realización. La duración ideal de una iteración es de una a tres semanas. Para cada iteración se define un conjunto de historias que se van a implementar, que deben superar las pruebas de aceptación que establece el cliente, al final de cada etapa de trabajo. (29)

Tabla 5: Plan de duración de las iteraciones

| Iteración | Historia de Usuario | Duración Total |
|-----------|---------------------|----------------|
|-----------|---------------------|----------------|



| | | |
|----------|---|-----------|
| 1 | Gestionar áreas Gestionar SS Gestionar trabajador Gestionar potencial financiero Gestionar actividad Gestionar avisos | 3 semanas |
| 2 | Dar baja del potencial Hacer cambio de escala Actualizar estado financiero Generar reportes financieros Generar reportes emulación Modificar datos usuario | 3 semanas |
| 3 | Buscar trabajador Ver estado financiero Listar actividades próximas Ver trazabilidad de actividades | 3 semanas |

Estimación de esfuerzos por historia de usuario

Tabla 6: Estimación de esfuerzos por HU

| No | Historia de Usuario | Estimación(semanas) |
|----|----------------------|---------------------|
| 1 | Gestionar trabajador | 0.3 |



| | | |
|-----------|---------------------------------|------------|
| 2 | Gestionar áreas | 0.3 |
| 3 | Gestionar SS | 0.2 |
| 4 | Generar potencial financiero | 1 |
| 5 | Dar baja del potencial | 0.5 |
| 6 | Hacer cambio de escala | 0.5 |
| 7 | Ver estado financiero | 0.5 |
| 8 | Actualizar estado financiero | 0.5 |
| 9 | Generar reportes financieros | 0.5 |
| 10 | Gestionar actividad | 0.5 |
| 11 | Listar actividades próximas | 0.5 |
| 12 | Generar reportes emulación | 0.5 |
| 13 | Gestionar avisos | 0.2 |
| 14 | Ver trazabilidad de actividades | 0.5 |
| 15 | Modificar datos usuarios | 0.5 |
| 16 | Buscar trabajador | 0.5 |

Plan de entregas

En el plan de entregas se realiza un cronograma donde el cliente establece la prioridad de cada HU, cuáles serán agrupadas para conformar una entrega y el orden de las



mismas. En correspondencia, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. (31)

Tabla 7: Plan de entregas

| No | Historia de Usuario | Primera Iteración | Segunda Iteración | Tercera Iteración |
|----|--------------------------------|-------------------|-------------------|-------------------|
| 1 | Gestionar trabajador | V 1.0 | Finalizado | Finalizado |
| 2 | Gestionar potencial financiero | V 1.0 | Finalizado | Finalizado |
| 3 | Gestionar avisos | V 1.0 | Finalizado | Finalizado |
| 4 | Gestionar actividad | V 1.0 | Finalizado | Finalizado |
| 5 | Gestionar áreas | V 1.0 | Finalizado | Finalizado |
| 6 | Gestionar SS | V 1.0 | Finalizado | Finalizado |
| 7 | Dar baja del potencial | - | V 1.0 | Finalizado |
| 8 | Hacer Cambio de Escala | - | V 1.0 | Finalizado |
| 9 | Actualizar estado financiero | - | V 1.0 | Finalizado |
| 10 | Generar reportes financieros | - | V 1.0 | Finalizado |
| 11 | Generar reportes emulación | - | V 1.0 | Finalizado |
| 12 | Modificar datos usuario | - | V 1.0 | Finalizado |
| 13 | Buscar trabajador | - | - | Finalizado |
| 14 | Ver estado financiero | - | - | Finalizado |



| | | | | |
|----|---------------------------------|---|---|------------|
| 15 | Listar actividades próximas | - | - | Finalizado |
| 16 | Ver Trazabilidad de Actividades | - | - | Finalizado |

2.7. Diseño

Tarjetas Contenido, Responsabilidad y Colaboración (CRC)

La metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Cada tarjeta CRC representa una clase, objeto, módulo, paquete, etc. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior izquierda están descritas las responsabilidades y a la derecha las clases que le sirven de soporte. (29) (32)

A continuación se muestran las tarjetas CRC utilizadas para de la propuesta de solución:

Tabla 8: Tarjeta CRC 1

| Módulo Finanzas | |
|---------------------|---|
| Responsabilidades | Colaboraciones |
| Gestionar áreas | Module CoreOptions Field SQL storage |
| Gestionar SS | Module CoreOptions Field SQL storage |
| Listar trabajadores | Gestion Laboral Chaos tolos Views |



| | |
|-------------------------------------|-------------|
| Agregar trabajador | |
| Gestionar trabajadores | Chaos tools |
| Pago CTC | Chaos tools |
| Pago MTT | Chaos tools |
| Consultar monto a pagar por mes CTC | Chaos tools |
| Consultar monto a pagar por mes MTT | Chaos tools |

Tabla 9: Tarjeta CRC 2

| Módulo Emulación | |
|--------------------------|-----------------------|
| Responsabilidades | Colaboraciones |
| Gestionar actividad | Chaos tools |
| Listar actividades | Chaos tools Views |
| Generar reporte | Chaos tools |
| Gestionar avisos | Chaos tools |

2.8. Conclusiones Parciales

Los artefactos elaborados permitieron realizar una descripción de la propuesta de solución, lo cual posibilitó organizar todo el proceso de desarrollo y servirá posteriormente



como guía de la implementación. El estudio de la audiencia a la que va dirigida la solución permitió delimitar la existencia de 6 tipos de usuarios que interactúan con el sistema, con diferentes niveles de acceso a las distintas funcionalidades.

El seguimiento de las fases que propone la metodología XP permitió definir un total de dieciséis HU, se precisó la prioridad de cada una de ellas para el negocio, permitiendo de esta forma establecer el orden de su implementación y el número de iteraciones en que serán desarrolladas. Estas, a su vez, permitieron construir el plan de entrega, que abarcará un total de nueve semanas, durante las cuales se desarrollarán las mismas. También fueron elaboradas las tarjetas CRC, lo que permitió definir la relación que existe entre cada clase y otras clases colaboradoras permitiendo una mejor organización a la hora de implementar para dar paso a la fase de Implementación.



Capítulo 3 Implementación y pruebas

3.1. Introducción

En el presente capítulo se detallan las tres iteraciones abarcadas en el proceso de construcción del sistema, se exponen las tareas generadas por cada HU. Para concluir se analizan los resultados de las pruebas diseñadas para cada una de las HU que se proponen en la fase de Exploración y Planificación.

3.2. Implementación

Al definir las iteraciones, se comienza a realizar la implementación de las HU seleccionadas para realizarse en cada una de las mismas, iniciando con una revisión del plan de iteraciones el cual puede ser modificado en caso de ser necesario. El principal objetivo de este plan es descomponer cada HU en tareas de desarrollo, asignando a un grupo de desarrollo o persona, la responsabilidad de implementación. Estas pueden ser escritas en lenguaje técnico y no es necesario que el cliente las entienda.

En el presente trabajo se llevaron a cabo 3 iteraciones de desarrollo sobre la aplicación, permitiendo obtener un producto con todas las características necesitadas por el cliente. A continuación se muestran cada una de las iteraciones.

Tareas de Ingeniería

Para llevar a cabo la correcta implementación de las HU se deben definir por parte del equipo de desarrollo las Tareas de Ingeniería (TI) que se realizarán en cada una de las iteraciones. A continuación se descomponen las HU en TI, de esta forma se facilita el trabajo.

Iteración 1

En esta iteración se implementarán las HU desde la 1 hasta la 6, se muestra de ejemplo la tarea generada por la HU “Generar potencial financiero”, en los anexos puede consultar las restantes TI (**Ver Anexo 3**)

Tabla 10: TI Generar potencial financiero



| Tarea de ingeniería | |
|---|------------------------------|
| Número tarea: 1 | Número historia: 4 |
| Nombre tarea: Generar potencial financiero | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 17/2/2014 | Fecha fin: 22/2/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: Implementar la funcionalidad adicionar trabajador haciendo uso de function gestion_laboral_generar_potencial (\$form, &\$form_state, \$term, \$no_js_use = FALSE){} | |
| Genera una vista con formato de planilla, para luego exportarla con los datos de los trabajadores agrupados por cuota sindical. | |

Iteración 2

En esta iteración se implementarán las HU desde la 7 hasta la 12, se muestra como ejemplo la tarea generada por la HU “Actualizar estado financiero de las SS”, para consultar las restantes TI puede acceder a los anexos. **(Ver Anexo 3)**

Tabla 11: TI Actualizar estado financiero de las SS

| Tarea de ingeniería | |
|---|------------------------------|
| Número tarea: 1 | Número historia: 9 |
| Nombre tarea: Actualizar estado financiero de las SS | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.5 |



| | |
|---|-----------------------------|
| Fecha inicio: 12/3/2014 | Fecha fin: 18/3/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: Implementar la funcionalidad actualizar el estado financiero de las SS utilizando</p> <pre>function gestion_laboral_form_actualizar_estado (\$form, &\$form_state, \$edit = array()) {}</pre> <p>Esta funcionalidad permite editar la entidad (potencial), recibe un arreglo u objeto por parámetros y esto permite modificar sus campos correspondientes.</p> | |

Iteración 3

En esta iteración se implementarán HU desde la 13 hasta la 16, como ejemplo se muestran las tareas generadas de las HU “Ver estado financiero CTC” y “Ver estado financiero del Día de la Patria”, para consultar las restantes TI acceder a los anexos. **(Ver Anexo 3)**

Tabla 12: TI Ver estado financiero CTC

| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 1 | Número historia: 13 |
| Nombre tarea: Ver estado financiero CTC | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.5 |
| Fecha inicio: 8/4/2014 | Fecha fin: 10/4/2014 |
| Programador responsable: Lisandra Duany | |



Descripción: Implementar la funcionalidad estado financiero utilizando `gestion_laboral_estado_financiero_ctc($form, &$form_state, $term, $no_js_use = FALSE){}`

Esta función ofrece una vista y un formulario con los cumplidores e incumplidores por mes y año completo del pago de la CTC, y permite ver el historial de pagos de cada trabajador.

Tabla 13: TI Ver estado financiero del Día de la Patria

| Tarea | |
|--|------------------------------|
| Número tarea: 2 | Número historia: 13 |
| Nombre tarea: Ver estado financiero del Día de la Patria | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.5 |
| Fecha inicio: 8/4/2014 | Fecha fin: 10/4/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: Implementar la funcionalidad estado financiero utilizando <code>gestion_laboral_estado_financiero_mtt(\$form, &\$form_state, \$term){}</code></p> <p>Esta función ofrece una vista y formulario con los cumplidores e incumplidores por mes del pago de Día de la Patria, y permite ver el historial de los pagos de cada trabajador.</p> | |

3.3. Estructura de Drupal

Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menús*), permisos de usuario y



plantillas (templates). (33) Para consultar la estructura de capas y datos de Drupal acceder a los anexos. **(Ver Anexo 4)**

Los nodos (*nodes*) son los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio web crece, lo va haciendo el número de nodos los cuales van formando un “depósito de nodos” cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal la forma este “depósito” de nodos.

Los módulos (*modules*) son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal, permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. Son extensiones que se instalan en el sitio web proporcionándole nuevas funcionalidades.

La siguiente capa de Drupal la constituyen los bloques y menús (*blocks & menús*). Estos permiten estructurar y organizar los contenidos en la página web. Es decir que son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

La siguiente capa importante en Drupal es la de control de usuarios y permisos. Actualmente, la mayor parte de los sitios web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario. Es decir que las operaciones que se pueden realizar sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación) se encuentran limitadas por la capa de control de usuarios y permisos de Drupal.

La última capa, es la capa de temas (*themes*) y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se quiere es renovar la apariencia de un sitio web. (33)



Drupal se estructura en un conjunto agrupado de carpetas que contienen el código PHP, imágenes y ficheros de configuración necesarios para su funcionamiento. Cada vez que algún usuario accede al sitio web, el servidor web accede a esta carpeta, interpreta el código y realiza las acciones pertinentes.

Los ficheros `install.php` y `update.php` que se hallan en la raíz, realizan las tareas de instalación y actualización del sitio web. También en la raíz aparece el fichero `cron.php` este contiene el código encargado de la ejecución de las tareas periódicas de mantenimiento del sitio. Se encuentran las carpetas *modules* y *themes* en las que Drupal guarda los módulos y temas que trae por defecto. Allí se podría guardar también los módulos y temas que se descarguen para el sitio, pero a pesar de que funcionaría, no es lo correcto ya que Drupal limpia esas carpetas cada vez que se actualiza. Lo correcto es crear dos carpetas diferenciadas (*themes* y *modules*) en “`../sites/all`” lo que evitará que se puedan borrar cada vez que se actualice el *core*³. Esto ayudará a mantener los ficheros de los módulos y temas más ordenados y controlados. (33)

3.4. Creación de las tablas de la base de datos

Por defecto, la instalación de Drupal crea 74 tablas que son utilizadas por el CMS para almacenar y manejar datos de nodos, usuarios, bloques, menús y contenidos. El diseño de la estructura de las tablas que se agregaron a la base de datos de Drupal es el primer paso para comenzar la codificación de la solución informática. Teniendo en cuenta las nuevas funcionalidades, se hace necesario crear las siguientes tablas. **(Ver anexo 5)**

3.5. Módulos desarrollados para el Sistema de Gestión Sindical de la UCI

En una primera etapa de la implementación se creó el archivo con la información del módulo (`.info`) y posteriormente otro archivo (`.install`) para crear la estructura de las tablas asociadas. La implementación de las funcionalidades se llevó a cabo en un tercer archivo

³ *Core*:(núcleo) son los módulos provistos por Drupal al instalarse, algunos de ellos fueron contribuciones de la comunidad de Drupal que se incorporaron.

(.module). La estructura de módulo se encuentra en el directorio sites/all/modules/custom desde donde se verificó su ejecución.

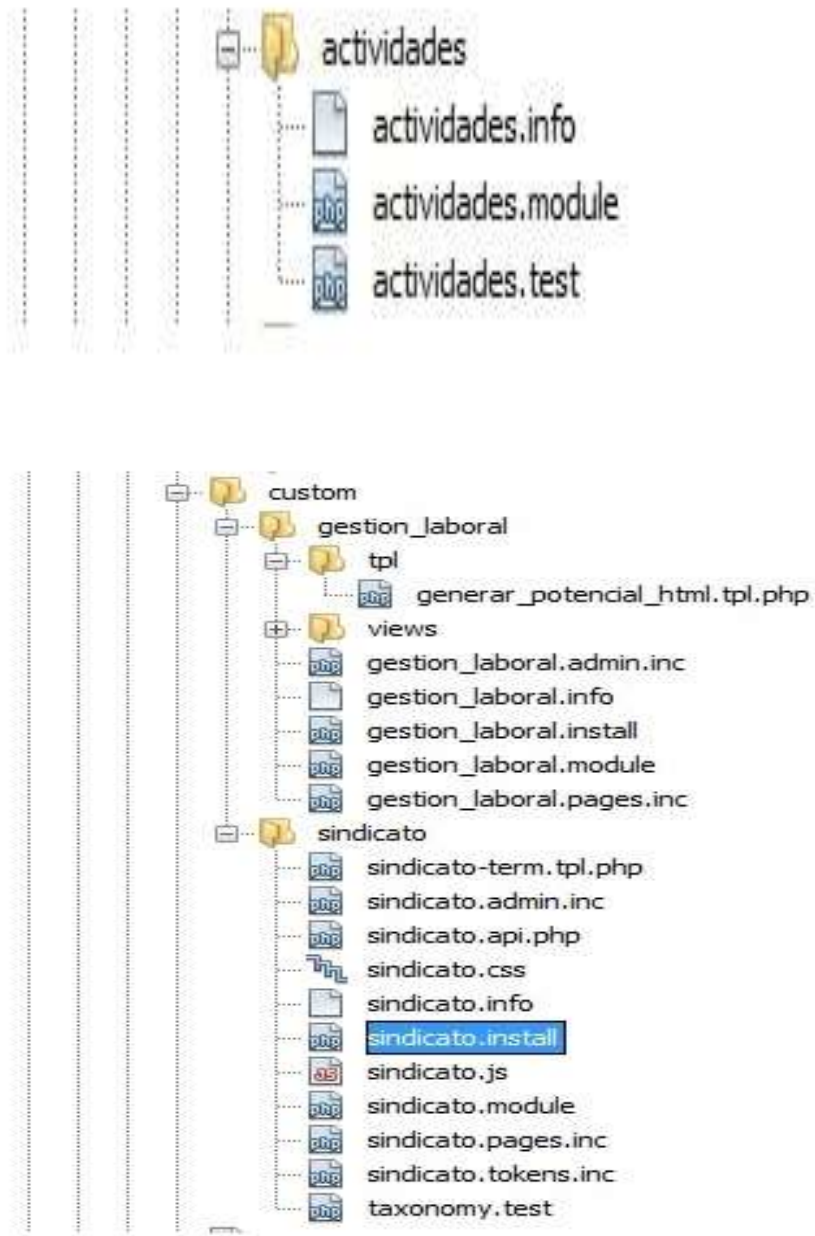


Figura 1: Ubicación de los módulos en la estructura de archivos de Drupal



3.6. Patrones de diseño utilizados

Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo. Por tanto está basado en la recopilación del conocimiento de los expertos en desarrollo de *software*. (34)

Dentro del *core* de Drupal se hace uso de multitud de patrones de diseño, entre ellos se pueden citar los siguientes:

***Inversion of Control* (Inversión de control)**

Este patrón de diseño consiste en un cambio en el flujo de ejecución de un programa en el que en lugar de especificar el flujo de la información se especifica la respuesta esperada de una operación. (34)

Se utiliza para ser habilitada o deshabilitada la funcionalidad que proporcionan los módulos a través de la correspondiente página de administración.

***Singleton* (Instancia única)**

Este patrón se evidencia si se asocian los módulos y los temas como objetos, cada uno de ellos en si no almacena datos, sino que representan un conjunto de funciones que extienden el núcleo para añadirle funcionalidades al mismo, cada uno por separado. De esta forma se pueden considerar estos como una clase con una instancia *singleton*. (34)

Dentro del *core* de Drupal se utiliza este patrón de diseño en diversas tareas como la gestión del pool de conexiones (1 conexión permite hacer varias peticiones a la base de datos) con la base de datos y pensando en los módulos y temas de Drupal como objetos para llevar a cabo la gestión de dichos elementos.

***Decorator* (Decorador)**

El patrón *Decorator* responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la



primera. Este patrón de diseño es utilizado en el *core* de Drupal para facilitar la extensión de sus componentes. (34)

Observer (Observador)

Define una dependencia de uno-a-muchos entre objetos de forma que, cuando un objeto cambia de estado, se notifica a los objetos dependientes para que se actualicen automáticamente. (35)

Patrones Arquitectónicos

Drupal y su arquitectura no corresponden puramente al patrón Modelo Vista Controlador (MVC) ni a ningún otro. Sin embargo su flexible estructura separa por completo el desarrollo del aspecto a través de Temas (Vista), las interfaces para trabajar con la base de datos vienen en el núcleo de Drupal (Modelo) y básicamente se programa solo controladores a la hora de extender el núcleo con los módulos contribuidos. Un desarrollador tiene que saber que cuando desarrolla para Drupal ya está desarrollando en MVC. Aunque no esté haciendo la separación de sus archivos modelo, vista o controlador en diferentes carpetas, se están programando por separado estos aspectos. (36)

3.7. Instalación de los módulos

Para instalar un módulo en el sistema lo primero que se debe realizar es copiar la carpeta contenedora de los archivos al directorio `sites/all/modules` con el nombre definido. Luego, en la interfaz administrativa se visita el vínculo “Módulos” en el menú superior del sistema mostrando una lista de los módulos disponibles. Para realizar la activación de un módulo basta con seleccionarlo y guardar los cambios. La interfaz se actualiza y se activan los cambios realizados.

3.8. Interfaz de usuario de los módulos

A continuación se muestran las interfaces de usuarios generadas luego de implementar todas las HU definidas por el cliente, sin interfaces propias del *framework* Drupal ya que



son solamente los módulos creados, el portal que soportará estos módulos y otros no ha sido terminado, pero mediante las interfaces que se obtuvieron se pudo verificar su correcto funcionamiento y satisfacción del cliente. Las restantes interfaces del sistema se encuentran en los anexos. **(Ver Anexo 6)**

Inicio » Agregar contenido

Sección Sindical
Administre su SS

Navegación

- ▼ Agregar contenido
 - Actividades
 - Calendar
 - Field Example
 - Reporte CTC
 - Reporte MIT

Management

Mi cuenta Cerrar sesión

Crear Actividades

Title *

Fecha

Show End Date

| Date | Time |
|---|------------------------------------|
| <input type="text" value="06/06/2014"/> | <input type="text" value="15:00"/> |
| E.g., 06/07/2014 | E.g., 15:00 |

Tipo de actividades

Figura 2: Prototipo de interfaz de usuario correspondiente a crear nueva actividad



Figura 3: Prototipo de interfaz de usuario correspondiente a gestionar trabajadores

3.9. Pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas, anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. (37)

Pruebas unitarias

Las pruebas unitarias o pruebas de unidad deben ser construidas antes que el código, permitiéndole a los programadores tener máxima claridad de lo que van a programar antes de hacerlo, así como conocer cada uno de los casos de prueba que deberán pasar, lo que optimizará el trabajo y el código será de mayor calidad. Estas pruebas son el arte



y la práctica de tomar una pequeña porción de código, una unidad, y someterlo a pruebas de programación para probar su corrección. Las unidades más pequeñas en el código PHP son normalmente las funciones; en las pruebas unitarias se ejecutarán las funciones con conjuntos de parámetros controlados, y se realizan afirmaciones acerca de los valores de retorno de la función, o sobre el estado de la aplicación antes y después de que la función se ha ejecutado. (37)

Para la realización de las pruebas unitarias en el CMS Drupal, se utilizó el módulo *unit testing* el cual es una práctica que consiste en tomar una pequeña porción de código y someterlo a unas pruebas de programación para probar su corrección. Por lo general se someten a prueba, funciones o pequeñas funcionalidades específicas con un conjunto de parámetros controlados que servirán para hacer las comprobaciones necesarias en función de los valores de retorno de la funcionalidad testada. Dentro de sus utilidades se pueden observar:

- ❖ Ayuda a producir un código de mayor calidad.
- ❖ Detección rápida de errores cuando se programan nuevas funcionalidades o se realizan cambios en el código.
- ❖ Sirve como pequeña fuente de documentación sobre qué es lo que se espera que haga el código.

Realizar *unit tests* obliga al programador a escribir el código en pequeñas porciones con el fin de que puedan ser probadas independientemente.

Luego de realizadas las pruebas del código se obtuvieron resultados satisfactorios, para un total de 15 pruebas realizadas no se encontraron errores en el código.

Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las HU, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una HU ha sido correctamente implementada.



Las pruebas de aceptación son consideradas como “pruebas de caja negra”(“*Black box system tests*”). Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos, así mismo en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución.

Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. (37)

Existen en la actualidad muchas herramientas para desarrollar pruebas de aceptación, lo suficientemente sencillas para que un cliente pueda manejarlas. Sin embargo estas herramientas son inútiles si el cliente es incapaz de diseñar un conjunto completo de pruebas que verifiquen toda la funcionalidad del sistema y no solo una parte o con unos valores concretos. (37)

Para consultar los restantes casos de pruebas realizados para las pruebas de aceptación puede remitirse a los anexos. **(Ver Anexo 7)**

Casos de prueba realizados para las HU de la iteración 1

Tabla 14: Caso de prueba realizado a la HU Gestionar trabajador

| Caso de prueba | |
|--|----------------------------|
| Número caso prueba: 1 | Número historia usuario: 3 |
| Nombre caso prueba: Gestionar trabajador | |
| Descripción: Prueba para la funcionalidad que le permite al usuario ver el estado financiero de los trabajadores de la SS. | |
| Condiciones de ejecución: El usuario debe tener rol de financiero o secretario. | |



| |
|---|
| Entradas: |
| <ul style="list-style-type: none">❖ El usuario después de haberse autenticado selecciona la opción administrar mi sección sindical para adicionar los trabajadores puede acceder al <i>link</i> Agregar trabajador, a continuación muestra una vista con todos los trabajadores selecciona los que desea agregar y procede a Enviar.❖ Para realizar la acción de modificar los trabajadores accede al <i>link</i> Editar el cual mostrará una vista con los dato del trabajador, donde podrá realizar las modificaciones❖ Para realizar la acción de eliminar los trabajadores accede al <i>link</i> Eliminar el cual mostrará eliminará el trabajador agregado de su SS. |
| ❖ Resultado esperado: Permite ver cómo está el estado financiero de los trabajadores de la SS. |
| Evaluación: Prueba realizada satisfactoriamente |

Casos de prueba para las HU de la iteración 2

Tabla 15: Caso de prueba realizado a la HU Actualizar estado financiero

| Caso de prueba | |
|---|-----------------------------------|
| Número caso prueba: 2 | Número historia usuario: 9 |
| Nombre caso prueba: Actualizar estado financiero | |
| Descripción: Prueba para la funcionalidad que le permite al usuario ver el estado financiero de los trabajadores de la SS. | |
| Condiciones de ejecución: El usuario debe tener rol de financiero o secretario. | |



| |
|---|
| Entradas: El usuario después de haberse autenticado selecciona la opción administrar mi sección sindical para actualizar el estado financiero se selecciona el trabajador y se accede al <i>link</i> efectuar pago, en dependencia del pago que desee realizar, tanto el de las finanzas como el del Día de la patria, a continuación muestra una vista donde podrá realizar el pago, y de esta manera se actualiza el estado de su finanza. |
| Resultado esperado: Permite actualizar el estado financiero de los trabajadores de la SS. |
| Evaluación: Prueba realizada satisfactoriamente |

Casos de prueba para las HU de la iteración 3

Tabla 16: Caso de prueba realizado a la HU Ver estado financiero

| Caso de prueba | |
|---|------------------------------------|
| Número caso prueba: 3 | Número historia usuario: 13 |
| Nombre caso prueba: Ver estado financiero | |
| Descripción: Prueba para la funcionalidad que le permite al usuario ver el estado financiero de los trabajadores de la SS. | |
| Condiciones de ejecución: El usuario debe tener rol de financiero o secretario. | |
| Entradas: El usuario después de haberse autenticado selecciona la opción administrar mi sección sindical y escoge la pestaña estado financiero, puede ver el estado tanto de las finanzas como del Día de la patria, existen dos pestañas una para cada estado, en el cual se muestra una vista con los trabajadores filtrados por | |

cumplidores e incumplidores en dependencia de la categoría que quiera seleccionar, por mes por curso, y además ahí da la posibilidad de observar el historial de cada trabajador.

Resultado esperado: Permite ver cómo está el estado financiero de los trabajadores de la SS.

Evaluación: Prueba realizada satisfactoriamente

Resultado de las pruebas

Luego de realizadas las pruebas que fueron ejecutadas al final de cada iteración según las funcionalidades que abarcaban, el gráfico que se presenta corresponde a los resultados por iteraciones de la realización de las pruebas de aceptación y consisten en 13 no conformidades para la 1ra iteración, 9 para la segunda y 5 para la última iteración para un total de 27 no conformidades, donde las principales no conformidades significativas encontradas fueron errores de validación y las no significativas fueron errores ortográficos.

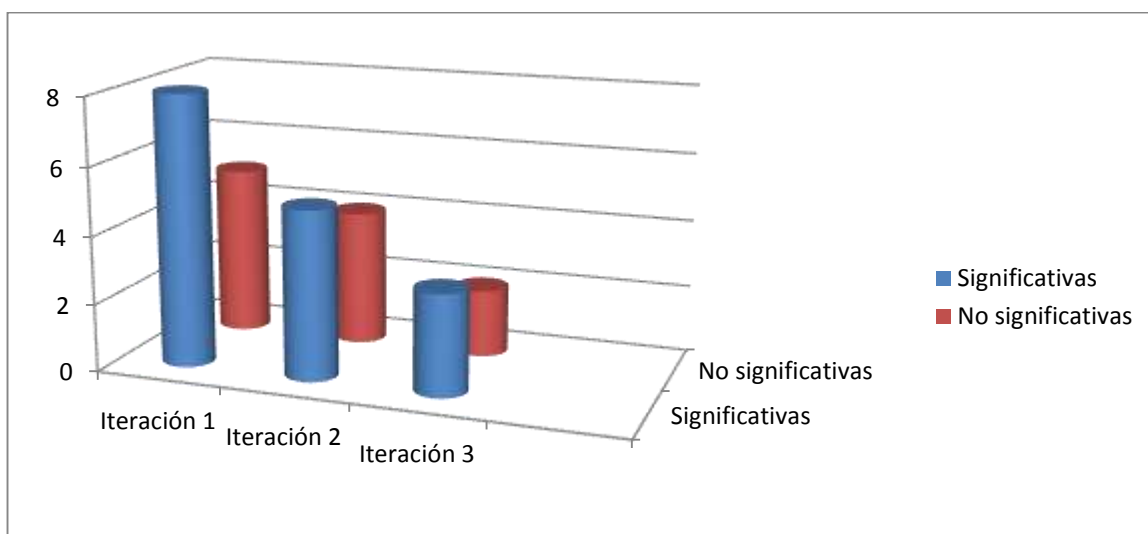


Figura 4: Resultado de las pruebas de aceptación



3.10. Validación de la investigación

Para validar la aplicación fue necesario comprobar la hipótesis planteada, donde la variable que interviene en la hipótesis redactada es, el mayor control de los procesos en cuestión.

La necesidad de un mayor control está determinada por el hecho de que existe, gran cúmulo de información en los procesos y esto causa pérdida de información en el envío, a través de correo electrónico o el manejo en formato duro.

El método escogido para realizar la validación de la investigación fue la encuesta. La misma fue aplicada a un grupo de futuros usuarios del sistema. **(Ver Anexo 8)**

Resultados de la encuesta

Se encuestó una muestra de 13 dirigentes sindicales que serán futuros usuarios del sistema para validar que cumpla con los objetivos propuestos en la investigación. Las personas encuestadas fueron 7 financieros y 6 emuladores de diferentes SS de la universidad. Los resultados obtenidos gracias a la aplicación de la encuesta validaron la hipótesis planteada en la investigación, demostrando que el sistema es lo suficientemente ágil, para los usuarios que lo utilizarán, ya que 12 usuarios lo consideran ágil mientras que una persona consideró que podía ser más ágil en la gestión de la información. De igual forma quedó demostrado que con el uso de estos módulos, se logrará tener mayor control de la información gestionada en dichos procesos, debido a que el 100% de los usuarios lo consideran así. La familiarización de los usuarios con los módulos es buena, un 84,61% de los usuarios están de acuerdo con este planteamiento, no se retrasará el tiempo para llevar a cabo el proceso del cobro de las finanza y es fácil de utilizar. En general los usuarios coinciden en que, el uso de estos módulos para el Sistema de Gestión de Información del BUS UCI, permitirá agilizar el trabajo y tener mayor control de la información gestionada en dichos procesos.



3.11. Conclusiones parciales

En el presente capítulo se abordaron todos los temas referentes a la implementación de la solución y a las pruebas a seguir durante la elaboración del mismo. Se detallaron las TI que fueron definidas a partir de las HU, lo que permitió organizar mejor el trabajo a la hora de programar y darle solución a cada una de ellas. Fueron aplicadas las pruebas de aceptación para comprobar que las funcionalidades implementadas cumplen con las necesidades del cliente y así verificar el cumplimiento de los objetivos de la investigación a través de la validación de la solución propuesta.



Conclusiones generales

- ❖ Los métodos científicos utilizados para investigar el objeto de estudio posibilitaron identificar la teoría y los conceptos que sustentan la presente investigación.
- ❖ La selección de la metodología, herramientas y tecnologías a utilizar posibilitaron cumplir con los objetivos propuestos para la investigación y así satisfacer las necesidades del cliente.
- ❖ Se realizaron todas las tareas de investigación, logrando de esta forma obtener los módulos Finanza y Emulación para mejorar la gestión de información de los procesos realizados.
- ❖ Los módulos realizados pueden ser adaptables a otros medios, por lo que se podrían utilizar en otras instituciones sin efectuarles cambios significativos.
- ❖ Las pruebas de aceptación realizadas permitieron verificar el correcto funcionamiento de las funcionalidades definidas en la investigación y eliminar los errores encontrados.
- ❖ La utilización de los módulos realizados pertenecientes al Sistema de Gestión Sindical de la UCI servirá de apoyo a los dirigentes sindicales de la universidad, pues su desarrollo evitará la gran acumulación de información, ya sea en formato duro o a través de correo electrónico y evitará la pérdida de información.



Recomendaciones

- ❖ Crear nuevos reportes como:
 - Secciones sindicales más destacadas en el pago de la CTC y del día de la Patria haciendo uso de porcentajes.
 - Comparar por año las secciones sindicales más destacadas en el pago de la CTC y el Día de la Patria.
 - Trabajadores que más participan en las actividades convocadas por su sección sindical o el buró UCI.
- ❖ Se recomienda extender el uso de esta herramienta hacia otros centros del país.



Bibliografía

1. **California, University of.** Technology and Information Management. *What is Information Systems Management (ISM)?* [Online] 2014. [Cited: 01 12, 2014.] <http://tim.soe.ucsc.edu/undergraduates>.
2. **Instituto Tecnológico de Sonora.** Introducción a los Sistemas de Informacón. *Introducción a los Sistemas de Informacón.* [Online] 2013. [Cited: 03 19, 2014.] http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm.
3. **Quiroga, Lic. Lourdes Aja.** Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. [Online] [Cited: 12 9, 2013.]
4. **Castro, Alfonso Ortega.** Finanza. [Online] [Cited: 12 9, 2013.] <http://articulosFinanzas.blogspot.com/2009/01/Finanzas.html>.
5. **CUBA. Secretariado Nacional De la CTC.** La Habana : s.n., 2006.
6. **Roberth G. Figueroa, Camilo J. Solís y Armando A. Cabrera.** Metodologías Tradicionales vs Metodologías Ágiles . [Online]
7. **José H. Canós, Patricio Letelier yM^a Carmen Penadés.** *Metodologías Ágiles en el Desarrollo de Software.*
8. **Rodolfo Villarroel Acevedo, Cristián Rioseco Reinoso.** Revista Cubana de Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas.* [Online] 2011. [Cited: 12 9, 2013.] <http://rcci.uci.cu/index.php/rcci/article/view/86..>
9. **Connexions.** Funciones de los Sistemas Gestores de Bases de Datos. [Online] [Cited: Febrero 28, 2014.]
10. **MySQL.** [Online] [Cited: Febrero 24, 2014.] <http://dev.mysql.com/doc/refman/5.0/es/roadmap.html>.



11. Sitio Oficial de Postgres. [Online] [Cited: Enero 28, 2014.]
<http://www.postgresql.org.es>.
12. Sitio Oficial Apache. [Online] [Cited: Enero 21, 2013.]
<http://www.apache.org>.
13. Blanco, Carlos. carlosblanco.pro. entornos desarrollo integrado-introduccion. *carlosblanco.pro. entornos desarrollo integrado-introduccion*. [Online] 2012. [Cited: 01 24, 2014.] [http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/..](http://carlosblanco.pro/2012/04/entornos-desarrollo-integrado-introduccion/)
14. LENGUAJE DE PROGRAMACIÓN 1. [Online] [Cited: Febrero 25, 2014.]
http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf.
15. Zen, The Company PHP. [Online] [Cited: abril 1, 2014.]
<http://www.zend.com/en/products/studio/>.
16. Zend Studio - The Leading PHP IDE. [Online] [Cited: marzo 29, 2014.]
<http://www.zend.com/en/products/studio/>.
17. Netbeans. [Online] [Cited: Enero 24, 2014.] <https://www.netbeans.org>.
18. Aranda, Vicente Trigo. ACTA Autores científico-técnicos y académicos.
19. Mendoza, Esteban Sánchez. Programación Web. p. 29.
20. *Maestría en tecnología de informacion aplicaciones web*. Lado Servidor programacion web usuarios .
21. World Wide Web Consortium. [Online] [Cited: Febrero 28, 2014.]
<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
22. Pergamino Virtual. [Online] [Cited: Febrero 28, 2014.]
<http://www.pergaminovirtual.com.ar/definicion/XHTML.html>.



23. Sitio Oficial de PHP. [Online] [Cited: Febrero 25, 2014.]
<http://www.php.net>.
24. Lenguajes de Programación. [Online] [Cited: Febrero 25, 2014.]
<http://upcommons.upc.edu/pfc/bitstream/2099.1/5837/5/part%204.pdf>.
25. X Cuerda, J Minguilló. Introducción a los Sistemas de Gestión de
Contenidos (CMS) de código abierto. [Online] 2004.
<http://www.itescam.edu.mx>.
26. Toro, Miguel Perez-David. slideshare. [Online] [Cited: Marzo 31, 2014.]
<http://www.slideshare.net/miguelgonzalez/joomla-cms-presentation-831179>.
27. Claudio Escobar Arraigada, Joel Lagos Llancao. *'WordPress' y la
creación de un sitio Web dinámico.* Serie Bibliotecología y Gestión de
Información. 2006. ISSN 0718-1701.
28. Comunidad de usuarios de Drupal. [Online] [Cited: Febrero 20, 2014.]
<http://drupal.org>.
29. Carmona, Luis Miguel Echeverry Tobón-Luz Elena delgado. *Tesis Caso
Práctico de la Metodología Ágil XP al desarrollo de Software.* PEREIRA,
UNIVERSIDAD TECNOLÓGICA DE PEREIRA FACULTAD DE INGENIERÍA:
ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA
COMPUTACIÓN INGENIERÍA DE SISTEMAS PEREIRA : s.n., 2007.
30. Soriano, Fernando. *Ciclo de Ingeniería de Software. Desarrollo Iterativo
de Software.* Universidad FASTA. 2008.
31. Bajo de Luque, M.J. y et.al. Metodologías ágiles. *Agile Alliance.* [Online]
2011. [Cited: 2 25, 2014.]
<http://metologiasagiles.herobo.com/index.php/2011-12-05-16-09-55/metologia-xp>.



-
32. Metodologías ágiles. Agile Alliance. [Online] [Cited: Febrero 25, 2014.] <http://metologiasagiles.herobo.com/index.php/2011-12-05-16-09-55/metologia-xp>.
 33. Tutorial Drupal. [Online] [Cited: Abril 12, 2014.] <http://www.cursosdrupal.com/content/arquitectura>.
 34. Tejada, David Hernández. *Guía de patrones de diseño*. 2002.
 35. Sánchez, Ester Guerra. *Técnicas de programación*. Madrid : s.n., 2008/2009.
 36. Koalasot. [Online] [Cited: marzo 25, 2014.] http://www.Drupal_KOALASOFT.htm.
 37. joskovich, jose.



Glosario de términos

Código abierto: es la expresión con la que se conoce al software distribuido y desarrollado libremente.

Core: son los módulos provistos por Drupal al instalarse, algunos de ellos fueron contribuciones de la comunidad de Drupal que se incorporaron.

Multihilo: es aquel sistema operativo que posee varios hilos de ejecución, es una característica que le permite a la aplicación realizar varias tareas a la vez.

Multiusuario: en general se le llama multiusuario a la característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente.

Multiprogramado: es aquel que se caracterizan por tener múltiples programas activos compitiendo por los recursos del sistema: procesador, memoria, dispositivos periféricos.

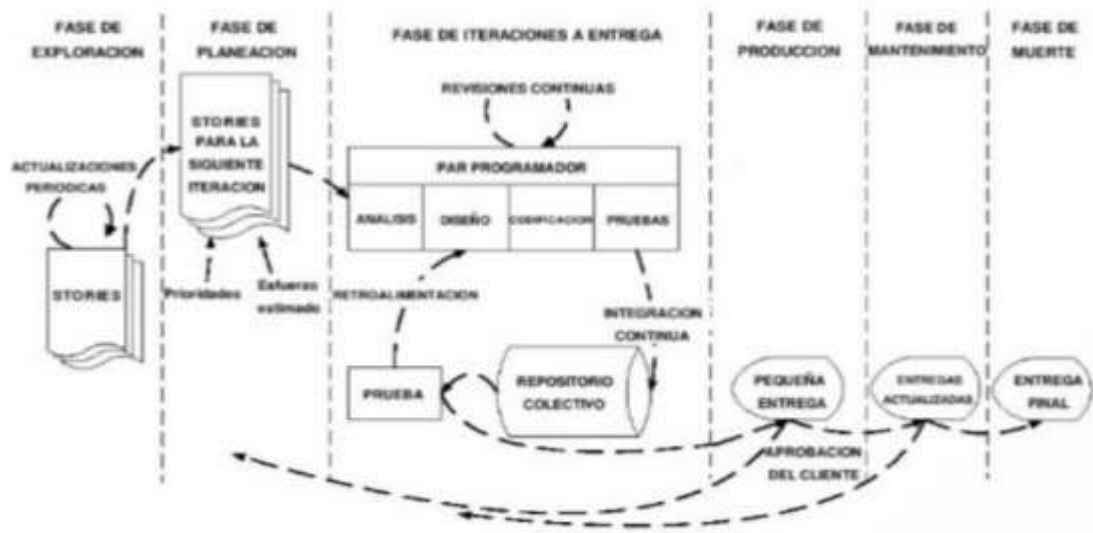
Potencial: tabla que relaciona a todos los afiliados de una SS con sus respectivos salarios y cuotas a cotizar.

Sprint: iteración de duración de SCRUM prefijada durante la cual el equipo trabaja para convertir los requisitos de los usuarios, en una nueva versión del *software* totalmente operativo.

Weblogs: es un sitio *web* personal, constituido por noticias y reflexiones, con un formato que facilita las actualizaciones.

Anexos

Anexo 1. Fases de XP



Anexo 2. Historias de usuario

Tabla 17: HU Gestionar actividad

| Historia de Usuario | |
|---|------------------------------------|
| No: 5 | Nombre: Gestionar actividad |
| Usuario: Emulación | |
| Prioridad: Alta | Estimación: 0.3 |
| Nivel de Complejidad: Alta | Iteración: 1 |
| Descripción: el usuario puede gestionar la información de las actividades que se realizan, tanto en la SS como en el BUS. Esto incluye almacenar información de cada actividad (nombre, fecha, hora, lugar, listado de participantes). | |



Observación: gestionar incluye adicionar, modificar, ver y eliminar los datos de cada trabajador.

Tabla 18: HU Gestionar avisos

| Historia de Usuario | |
|--|---------------------------------|
| No: 6 | Nombre: Gestionar avisos |
| Usuario: Miembro BUS | |
| Prioridad: Alta | Estimación: 0.2 |
| Nivel de Complejidad: Alta | Iteración: 1 |
| Descripción: el usuario puede gestionar avisos en los diferentes procesos que se realizan, tanto en los de Emulación como en los de Finanzas. | |
| Observación: gestionar incluye adicionar, modificar, ver y eliminar todos los avisos que sean emitidos por el BUS o por las SS. | |

Tabla 19: HU Gestionar áreas

| Historia de Usuario | |
|-----------------------------------|--------------------------------|
| No: 1 | Nombre: Gestionar áreas |
| Usuario: Miembro Buro | |
| Prioridad: Alta | Estimación: 1 días |
| Nivel de Complejidad: Baja | Iteración: 1 |



Descripción: permite gestionar las áreas a las que posteriormente serán agregadas las SS y los trabajadores.

Observación: gestionar incluye insertar, ver, modificar y eliminar las áreas.

Tabla 20: HU Gestionar SS

| Historia de Usuario | |
|---|-----------------------------|
| No: 2 | Nombre: Gestionar SS |
| Usuario: Miembro Buro | |
| Prioridad: Alta | Estimación: 1 días |
| Nivel de Complejidad: Baja | Iteración: 1 |
| Descripción: permite gestionar las SS. | |
| Observación: gestionar incluye insertar, ver, modificar y eliminar las SS. | |

Tabla 21: HU Dar baja del potencial

| Historia de Usuario | |
|-----------------------------------|---------------------------------------|
| No: 7 | Nombre: Dar baja del potencial |
| Usuario: Financiero | |
| Prioridad: Alta | Estimación: 1 días |
| Nivel de Complejidad: Alta | Iteración: 2 |



Descripción: permite eliminar algún trabajador del potencial.

Observación: debe incluirse la fecha y la razón por la cual ha causado baja.

Tabla 22: HU Hacer cambio de escala

| Historia de Usuario | |
|---|---------------------------------------|
| No: 8 | Nombre: Hacer cambio de escala |
| Usuario: Financiero | |
| Prioridad: Alta | Estimación: 1 día |
| Nivel de Complejidad: Alta | Iteración: 2 |
| Descripción: permite cambiar la escala de pago de un trabajador en caso de que haya tenido un aumento y reducción del salario. | |
| Observación: | |

Tabla 23: HU Ver estado financiero

| Historia de Usuario | |
|-----------------------------------|--------------------------------------|
| No: 6 | Nombre: Ver estado financiero |
| Usuario: Autenticado | |
| Prioridad: Alta | Estimación: 1 día |
| Nivel de Complejidad: Baja | Iteración: 3 |



Descripción: permite al usuario ver su estado financiero.

Observación: debe haberse realizado la HU 3

Tabla 24: HU Actualizar estado financiero

| Historia de Usuario | |
|---|---|
| No: 7 | Nombre: Actualizar estado financiero |
| Usuario: Financiero | |
| Prioridad: Alta | Estimación: 1 día |
| Nivel de Complejidad: Alta | Iteración: 2 |
| Descripción: el usuario selecciona los datos que desea modificar de uno o varios trabajadores y actualiza el estado del trabajador seleccionado. | |
| Observación: debe haberse realizado la HU 3 | |

Tabla 25: HU Generar reportes financieros

| Historia de Usuario | |
|-----------------------------------|---|
| No: 8 | Nombre: Generar reportes financieros |
| Usuario: Financiero | |
| Prioridad: Alta | Estimación: 2 días |
| Nivel de Complejidad: Alta | Iteración: 2 |



Descripción: el usuario selecciona el trabajador o la SS del cual quiere ver el reporte. El sistema muestra el reporte final de las finanzas.

Observación: esto se realiza dependiendo del rol, solo lo realiza el financiero, el secretario o un miembro del BUS.

Tabla 26: HU Listar actividades próximas

| Historia de Usuario | |
|--|--|
| No: 10 | Nombre: Listar actividades próximas |
| Usuario: Miembro del BUS | |
| Prioridad: Alta | Estimación: 1 día |
| Nivel de Complejidad: Baja | Iteración: 3 |
| Descripción: permite al usuario ver el listado con todas las actividades incluidas. | |
| Observación: debe haberse incluido al menos una actividad. | |

Tabla 27: HU Generar reportes emulación

| Historia de Usuario | |
|-----------------------------------|---|
| No: 12 | Nombre: Generar reportes emulación |
| Usuario: Emulación | |
| Prioridad: Alta | Estimación: 2 días |
| Nivel de Complejidad: baja | Iteración: 2 |



Descripción: el usuario selecciona el trabajador o la SS del cual quiere ver el reporte. El sistema muestra el reporte final de la emulación.

Observación: esto se realiza dependiendo del rol, solo lo realiza el de emulación, el secretario o un miembro del BUS.

Tabla 28: HU Ver trazabilidad de actividades

| Historia de Usuario | |
|---|--|
| No: 14 | Nombre: Ver trazabilidad de actividades |
| Usuario: Autenticado | |
| Prioridad: Alta | Estimación: 1 día |
| Nivel de Complejidad: Baja | Iteración: 3 |
| Descripción: permite ver la trazabilidad de todas las actividades en las que ha participado un trabajador determinado o varios trabajadores. | |
| Observación: | |

Tabla 29: HU Modificar datos usuario

| Historia de Usuario | |
|-----------------------------|--|
| No: 17 | Nombre: Modificar datos usuario |
| Usuario: Autenticado | |
| Prioridad: Alta | Estimación: 1 día |



| | |
|---|---------------------|
| Nivel de Complejidad: Alta | Iteración: 2 |
| Descripción: permite que el usuario pueda modificar sus datos personales | |
| Observación: | |

Anexo 3. Tareas de ingeniería

Tabla 30: TI Adicionar nuevo trabajador

| Tarea de ingeniería | |
|---|-----------------------------|
| Número tarea: 1 | Número historia: 1 |
| Nombre tarea: Adicionar nuevo trabajador | |
| Tipo tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 22/2/2014 | Fecha fin: 26/2/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: implementar la funcionalidad adicionar trabajador haciendo uso de <code>trabajador_form_adicionar(\$form, &\$form_state, \$edit = array()) {}</code> Esta funcionalidad permite adicionar un nuevo ítem entidad (trabajador) | |

Tabla 31: TI Modificar datos del trabajador

| Tarea de ingeniería | |
|------------------------|---------------------------|
| Número tarea: 2 | Número historia: 1 |



| | |
|--|-----------------------------|
| Nombre tarea: Modificar datos del trabajador | |
| Tipo tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 22/2/2014 | Fecha fin: 26/2/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: Implementar la funcionalidad modificar datos del trabajador</p> <p>trabajador_form_modificar (\$form, &\$form_state, \$edit = array()) {}</p> <p>Esta funcionalidad permite editar la entidad (trabajador), recibe un arreglo u objeto por parámetros y esto permite modificar sus campos correspondientes.</p> | |

Tabla 32: TI Eliminar trabajador

| Tarea de ingeniería | |
|--|-----------------------------|
| Número tarea: 3 | Número historia: 1 |
| Nombre tarea: Eliminar trabajador | |
| Tipo tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 22/2/2014 | Fecha fin: 26/2/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: implementar la funcionalidad eliminar trabajador</p> <p>function trabajador_form_eliminar(\$form, &\$form_state, \$edit = array()) {}</p> <p>Esta misma función se utilizó para editar la entidad (trabajador), tiene la opción de eliminar la entidad y se valida en la misma función con la confirmación de dicha</p> | |



acción, demás se comprueba dicha acción en la función
`functionsindicato_form_vocabulary_submit($form, &$form_state) {}` donde se captura
 la acción del usuario, si accedió a editar y guardó sus datos o a eliminar la entidad.

Tabla 33: TI Mostrar trabajador nuevo

| Tarea de ingeniería | |
|---|-----------------------------|
| Número tarea: 4 | Número historia: 1 |
| Nombre tarea : Mostrar trabajador nuevo | |
| Tipo tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 22/2/2014 | Fecha fin: 26/2/2014 |
| Programador responsable: LisandraDuany | |
| Descripción: implementar la funcionalidad mostrar trabajador nuevo <code>function trabajador_overview_terms(\$form, &\$form_state, \$vocabulary) {}</code> Esta funcionalidad lista la entidad (trabajador) con las entidades hijas o las entidades que pertenecen a la misma. | |

Tabla 34: TI Adicionar áreas

| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 1 | Número historia: 1 |
| Nombre tarea: Adicionar las áreas | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |



| | |
|---|----------------------------|
| Fecha inicio: 27/2/2014 | Fecha fin: 1/3/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: implementar la funcionalidad adicionar áreas utilizando</p> <pre>function sindicato_form_vocabulary(\$form, &\$form_state, \$edit = array()) {}</pre> <p>Esta funcionalidad permite adicionar un nuevo ítem entidad (área) y por parámetros se maneja la acción que desea el usuario ya que dicha función se utiliza para el <i>CRUD</i> en general de la entidad.</p> | |

Tabla 35: TI Modificar áreas

| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 2 | Número historia: 1 |
| Nombre tarea: Modificar las áreas | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 27/2/2014 | Fecha fin: 1/3/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: implementar la funcionalidad buscar trabajador utilizando</p> <pre>function sindicato_form_vocabulary(\$form, &\$form_state, \$edit = array()) {}</pre> <p>Esta funcionalidad permite editar la entidad (área), recibe un arreglo u objeto por parámetros y esto permite modificar sus campos correspondientes.</p> | |

Tabla 36: TI Mostrar áreas



| Tarea de ingeniería | |
|---|------------------------------|
| Número tarea: 3 | Número historia: 1 |
| Nombre tarea: Mostrar las áreas | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 27/2/2014 | Fecha fin: 1/3/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: implementar la funcionalidad buscar trabajador utilizando function sindicato_overview_terms(\$form, &\$form_state, \$vocabulary) {}</p> <p>Esta funcionalidad lista la entidad (área) con las entidades hijas o las entidades que pertenecen a la misma, en este caso sección sindical (ss), se le pasa por parámetro la entidad (área) y busca la entidad hija correspondiente.</p> | |

Tabla 37: TI Eliminar áreas

| Tarea de ingeniería | |
|---|------------------------------|
| Número tarea: 4 | Número historia: 1 |
| Nombre tarea: Eliminar las áreas | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 27/2/2014 | Fecha fin: 1/3/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: implementar la funcionalidad buscar trabajador utilizando | |



```
function sindicato_form_vocabulary($form, &$form_state, $edit = array()) {}
```

Esta misma función se utilizó para editar la entidad (área), tiene la opción de eliminar la entidad y se valida en la misma función con la confirmación de dicha acción, demás se comprueba dicha acción en la función `functionsindicato_form_vocabulary_submit($form, &$form_state) {}` donde se captura la acción del usuario, si accedió a editar y guardó sus datos o a eliminar la entidad.

Tabla 38: TI Dar baja a trabajadores del potencial

| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 1 | Número historia: 7 |
| Nombre tarea: Dar baja a trabajadores del potencial | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.2 |
| Fecha inicio: 2/3/2014 | Fecha fin: 4/3/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: implementar la funcionalidad dar baja del potencial <pre>function gestion_laboral_bajas_form(\$form, &\$form_state, \$edit = array()) {}</pre> <p>Esta misma función se utilizó para editar la entidad (trabajador), tiene la opción de eliminar la entidad y pasarlo a otra.</p> | |

Tabla 39: TI Hacer cambio de escala en el pago de los trabajadores

| Tarea de ingeniería | |
|------------------------|---------------------------|
| Número tarea: 1 | Número historia: 8 |



| | |
|---|------------------------------|
| Nombre tarea: Hacer cambio de escala en el pago de los trabajadores | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 4/3/2014 | Fecha fin: 8/3/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: implementar la funcionalidad hacer cambio de escalas en el salario de los trabajadores utilizando:</p> <pre>function trabajador_form_edit(\$form, &\$form_state, \$edit = array()) {}</pre> <p>Esta misma función se utilizó para editar la entidad (trabajador), tiene la opción de modificar el salario de los trabajadores.</p> | |

Tabla 40: TI Generar reportes financieros

| Tarea de ingeniería | |
|---|------------------------------|
| Número tarea: 1 | Número historia: 10 |
| Nombre tarea: Generar reportes financieros de las SS | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 9/3/2014 | Fecha fin: 11/3/2014 |
| Programador responsable: Lisandra Duany | |
| <p>Descripción: implementar la funcionalidad que permita generar los reportes en las finanzas.</p> | |

Tabla 41: TI Generar Reportes Emulación



| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 1 | Número historia: 10 |
| Nombre tarea: Generar reportes de emulación | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 9/3/2014 | Fecha fin: 11/3/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: implementar la funcionalidad que permita generar los reportes en la emulación. | |

Tabla 42: TI Modificar datos de usuario autenticado

| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 1 | Número historia: 12 |
| Nombre tarea: Modificar datos usuario | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 19/3/2014 | Fecha fin: 22/3/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: implementar la funcionalidad para modificar los datos del usuario autenticado. | |

Tabla 43: TI Ver trazabilidad de actividades

| Tarea de ingeniería | |
|--|------------------------------|
| Número tarea: 1 | Número historia: 13 |
| Nombre tarea: Ver trazabilidad de actividades | |
| Tipo tarea: Desarrollo | Puntos estimados: 0.3 |
| Fecha inicio: 25/3/2014 | Fecha fin: 30/3/2014 |
| Programador responsable: Lisandra Duany | |
| Descripción: Implementar la funcionalidad ver trazabilidad de actividades utilizado <code>functionactividades_node_info() {}</code> Esta función le indica al sistema de una nueva entidad llamada actividades, en este caso al creación de un nodo. | |

Anexo 4. Estructura de capas y datos de Drupal

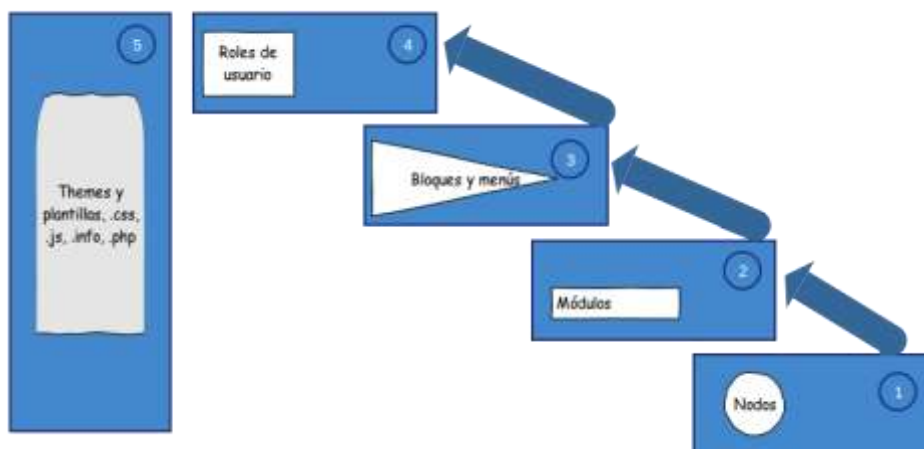
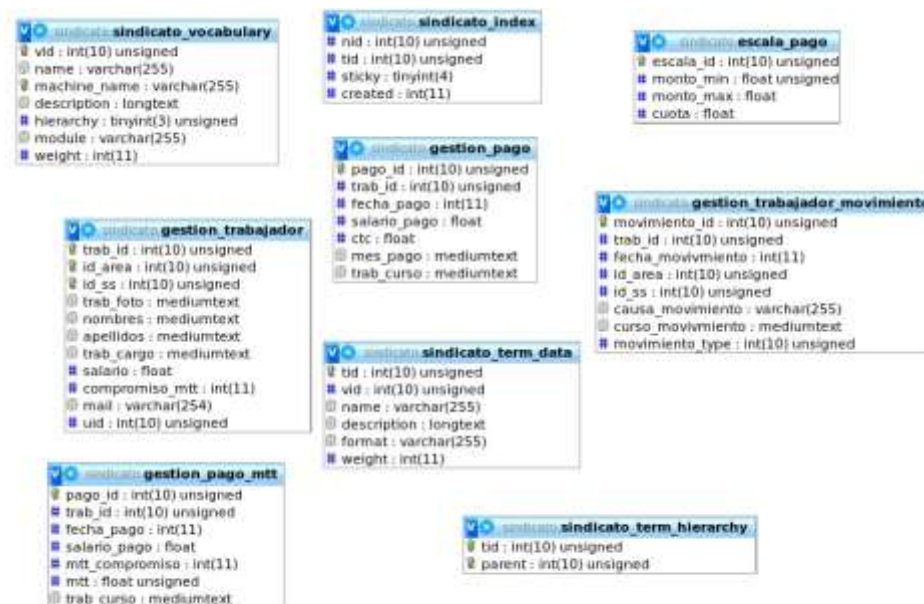




Figura 5. Estructura de capas y datos de Drupal

Anexo 5. Tablas de base datos





Anexo 6. Interfaces de Usuario

The screenshot shows a web interface titled "Estructura". On the left, there are three vertical navigation panels: "Sección Sindical" with "Administre su SS", "Navegación" with links for "Agregar contenido", "Calendar", "Field Example", "Reporte CTC", and "Reporte MTT", and "Management" with a dropdown for "Administración" containing "Estructura" (with sub-items "Gestión Laboral (Bajas)" and "Configuración"). The main content area has tabs for "Vista", "Editar", "Estado financiero CTC", "Estado financiero MTT", and "Potencial". Below the tabs is a dropdown menu labeled "Adicionar por/o" with "Trabajador" selected. The form contains three input fields: "Nombres" (with label "Nombres del trabajador"), "Apellidos" (with label "Apellidos del trabajador"), and "Cargo" (with label "Cargo del trabajador"). An "Agregar" button is at the bottom.

Figura 6: Prototipo de interfaz de usuario correspondiente a adicionar trabajador de forma manual

The screenshot shows a web interface titled "Sindicato" with a blue header bar containing a logo and the text "Sindicato". Below the header, there is a navigation panel on the left with "Inicio" and the same "Sección Sindical", "Navegación", and "Management" panels as in Figure 6. The main content area displays a confirmation dialog: "Estás seguro deseas dar baja al trabajador Yaset Lorenzo Garmendía ?". Below the dialog is a large empty text area labeled "Causa". At the bottom, there is a warning "Esta acción no se puede deshacer." and two buttons: "Baja" and "Cancelar".

Figura 7: Prototipo de interfaz de usuario correspondiente a dar baja de una SS

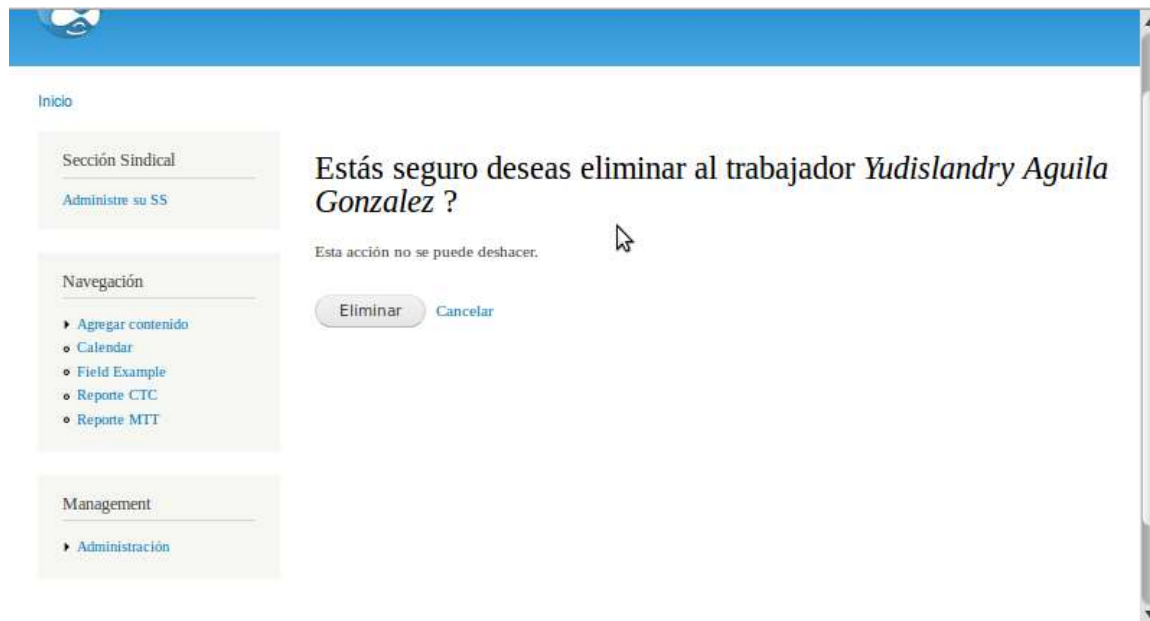


Figura 8: Prototipo de interfaz de usuario correspondiente a eliminar trabajador

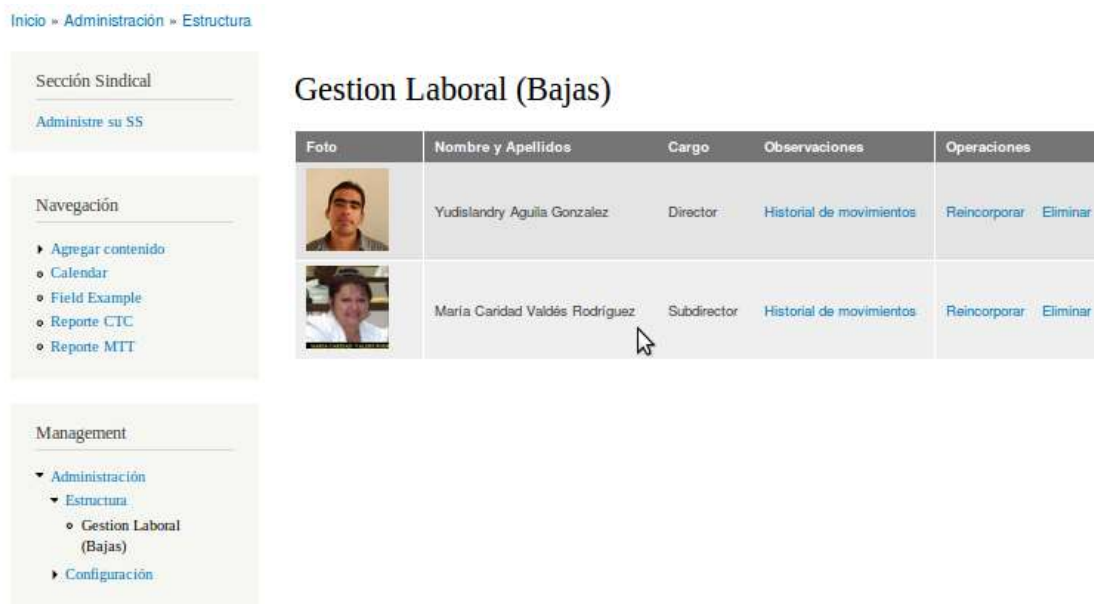


Figura 9: Prototipo de interfaz de usuario correspondiente a bajas



Figura 10: Prototipo de interfaz de usuario correspondiente a editar SS

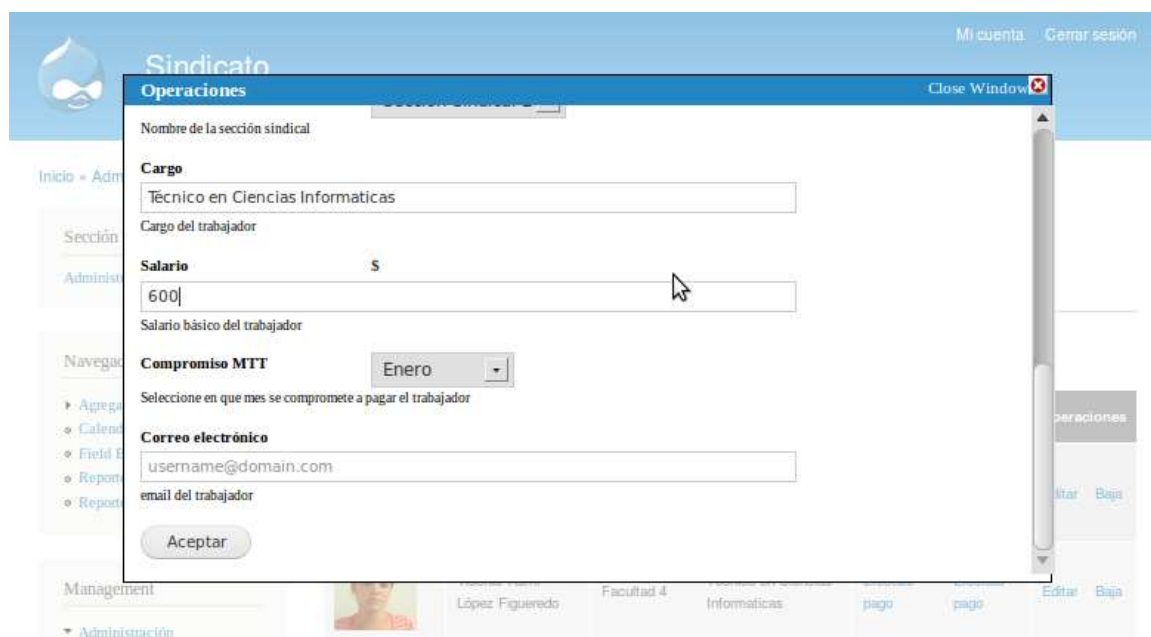




Figura 11: Prototipo de interfaz de usuario correspondiente a hacer cambio de escala

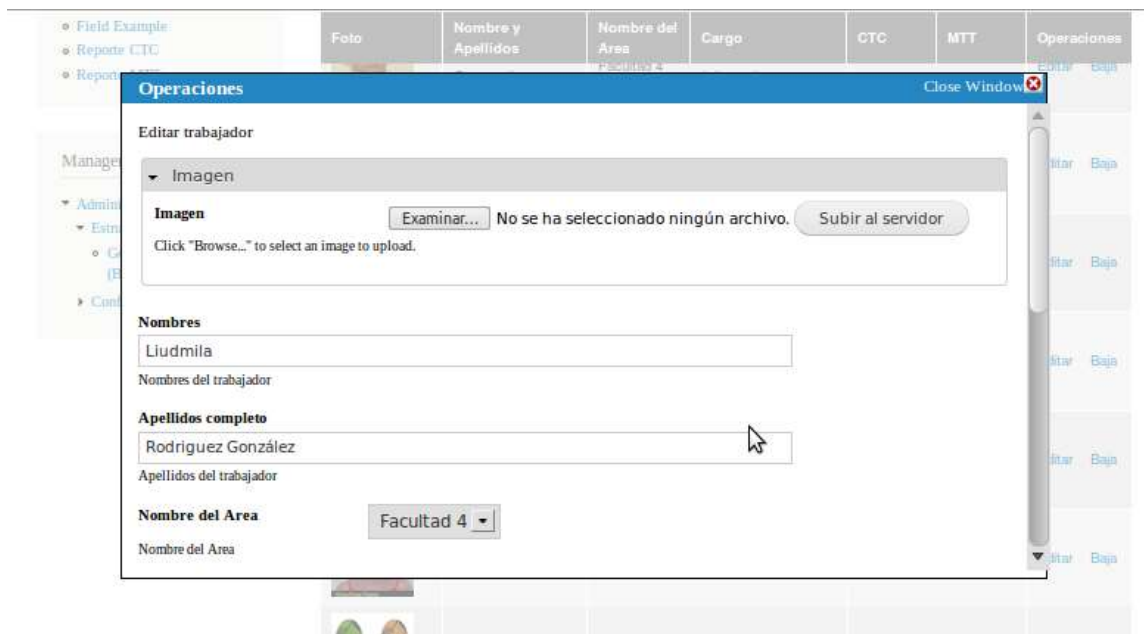


Figura 12: Prototipo de interfaz de usuario correspondiente a modificar datos de usuario



Inicio » Administración » Estructura

Sección Sindical
Administre su SS

Navegación

- Agregar contenido
- Calendar
- Field Example
- Reporte CTC
- Reporte MTT

Management

- ▼ Administración
 - ▼ Estructura
 - Gestion Laboral (Bajas)
 - Configuración

Estructura

Vista Editar Estado financiero CTC Estado financiero MTT Potencial

| Escala | Nombre y Apellidos | Salario | Ene. | Feb. | Mar. | Abr. | Mayo | Jun. | Jul. | Agos. | Sept. | Oct. | Nov. | Dic. |
|--------|-------------------------------|---------|------|------|------|------|------|------|------|-------|-------|------|------|------|
| 2.00 | Yeset Lorenzo Garmendia | 0.00 | | | | | | | | | | | | |
| 2.00 | Yisenia Taimi López Figueredo | 0.00 | | | | | | | | | | | | |
| 2.00 | Elier Rosales García | 0.00 | | | | | | | | | | | | |
| 2.00 | Leandro Pérez Solenzal | 0.00 | | | | | | | | | | | | |

Figura 13: Prototipo de interfaz de usuario correspondiente al potencial financiero

Sección Sindical
Administre su SS

Sección Sindical 2

Operaciones Close Window

Formulario CTC

Salario
Salario básico del trabajador

Monto a pagar
Monto a pagar dado el salario básico anterior del mes

Curso
Curso escolar escolar actual

Meses

Enero
Febrero
Marzo
Abril

Monto a pagar dada la cantidad de meses es: 30

Figura 14: Prototipo de interfaz de usuario correspondiente a actualizar estado financiero

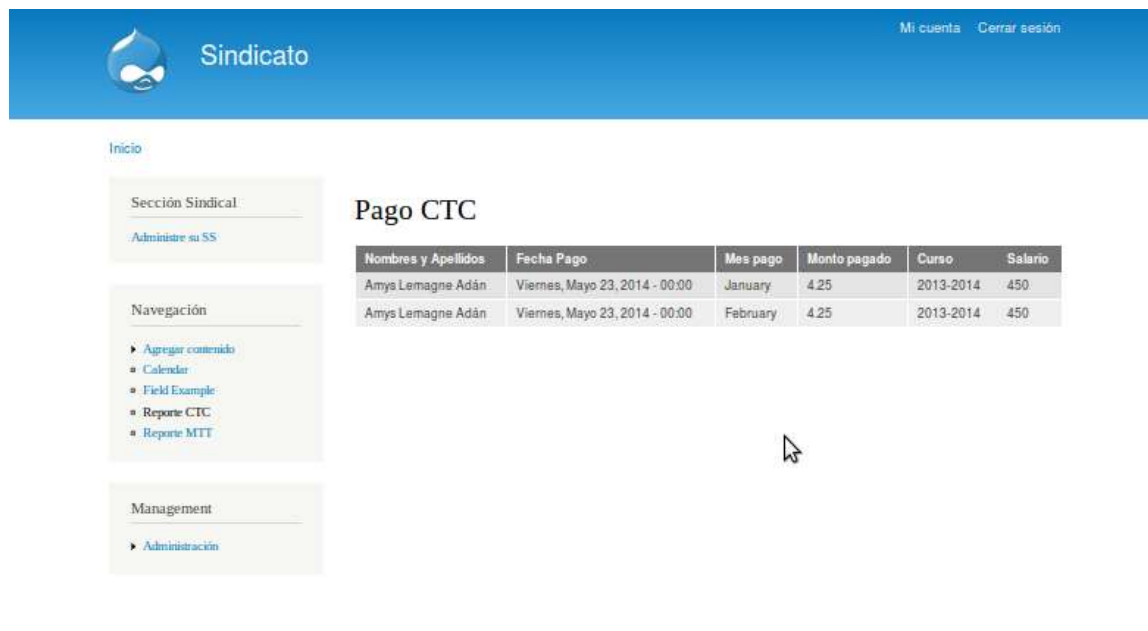


Figura 15: Prototipo de interfaz de usuario correspondiente a reporte del pago CTC

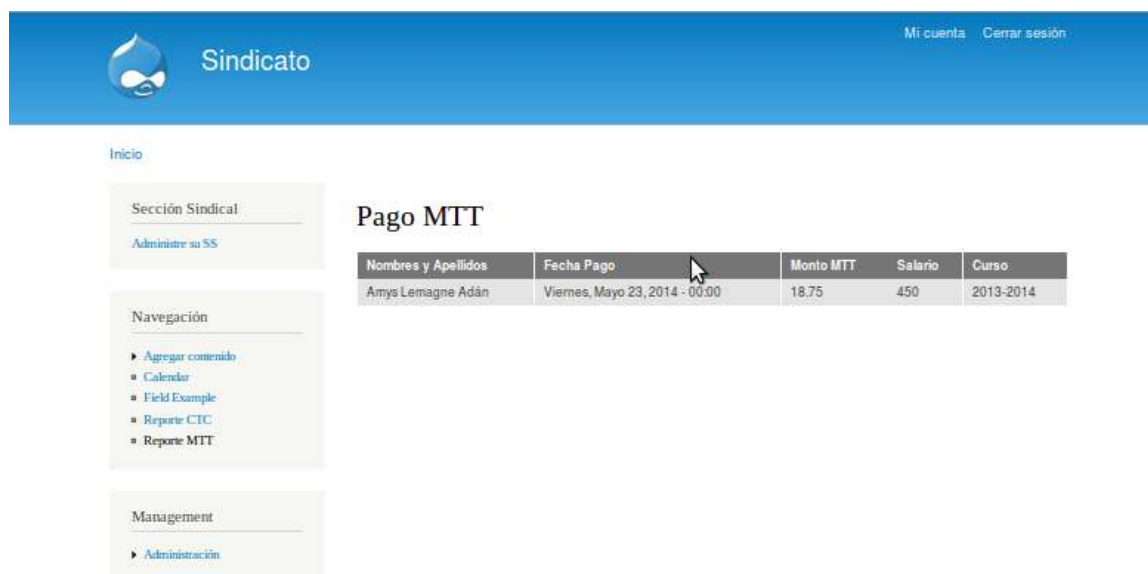


Figura 16: Prototipo de interfaz de usuario correspondiente a reporte del pago Día de la Patria



Figura 17: Prototipo de interfaz de usuario correspondiente a ver estado financiero

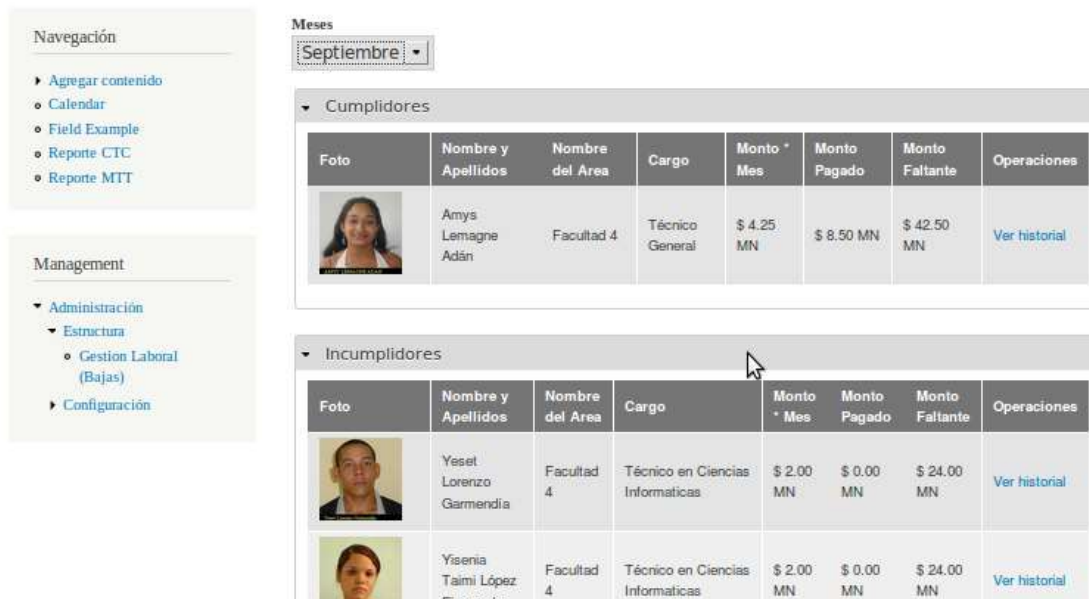


Figura 18: Prototipo de interfaz de usuario correspondiente a ver reporte del estado financiero

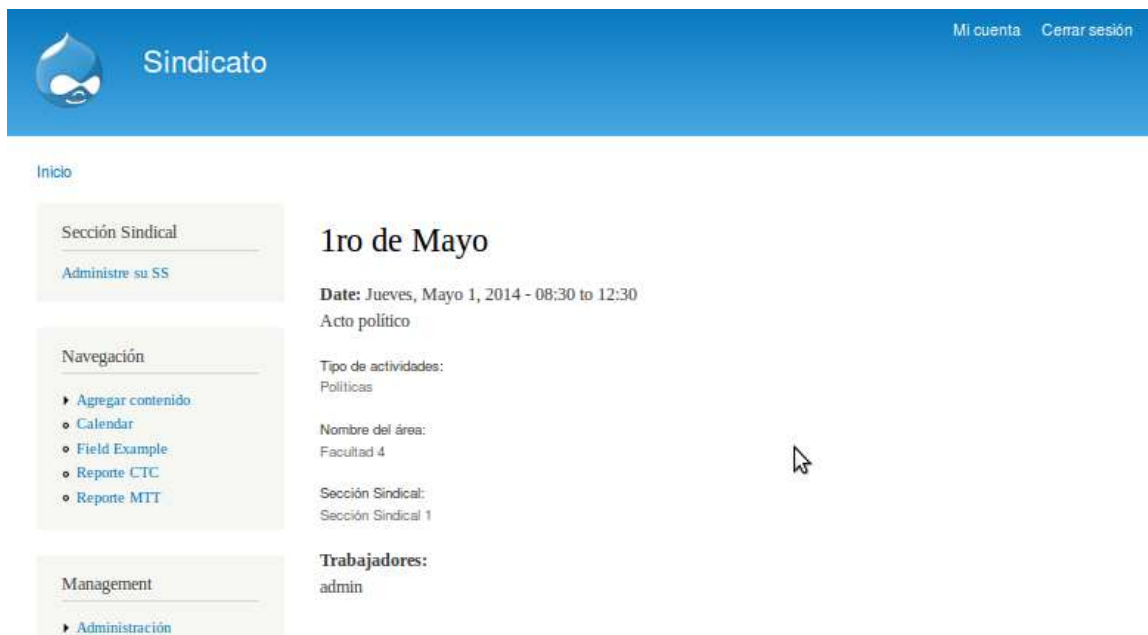


Figura 19: Prototipo de interfaz de usuario correspondiente a ver actividad

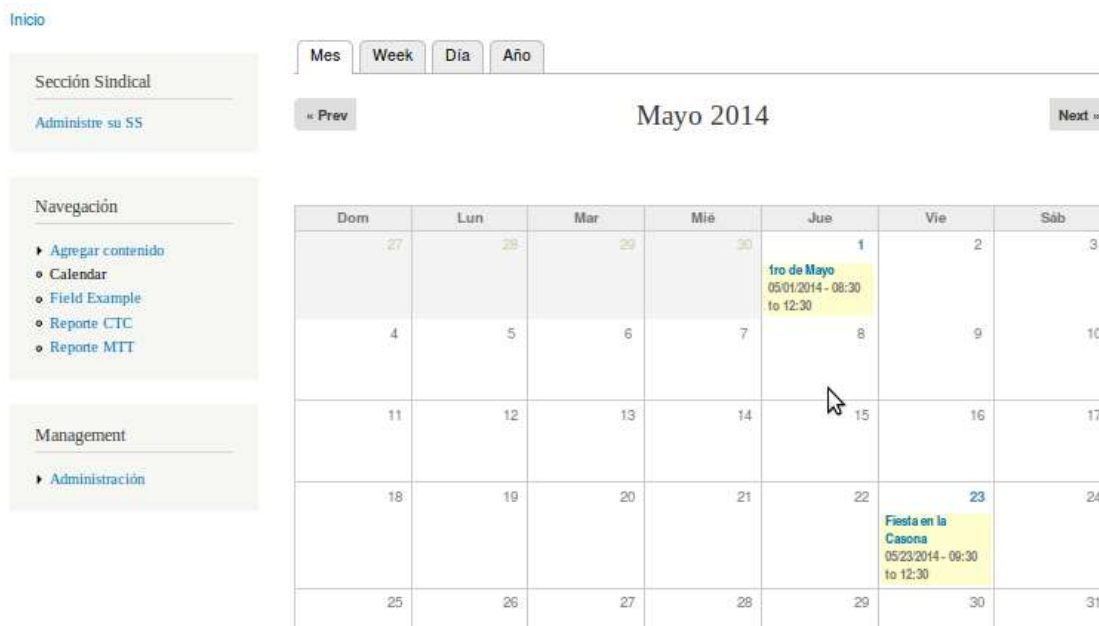


Figura 20: Prototipo de interfaz de usuario correspondiente a ver trazabilidad de actividades



Anexo 7. Casos de prueba

Tabla 44: CP Gestionar áreas

| Caso de prueba | |
|---|---------------------------|
| Número caso prueba:1 | Número historia usuario:1 |
| Nombre caso prueba: Gestionar áreas | |
| Descripción: prueba para la funcionalidad que le permite al usuario adicionar, | |
| Condiciones de ejecución: el usuario debe tener rol de financiero o secretario. | |
| Entradas: <ul style="list-style-type: none">❖ El usuario después de haberse autenticado selecciona la opción estructura/sindicatos/áreas y muestra el listado de todas las áreas existentes. En el caso de que no se haya agregado ninguna área, selecciona Adicionar área.❖ Para insertar nuevas áreas selecciona Adicionar área, inserta los datos requeridos (nombre y descripción) y procede a Enviar.❖ Para modificar el usuario selecciona el área y en operaciones escoge el <i>link</i> editar área, introduce los datos nuevos y procede a Enviar.❖ El usuario para realizar la acción de eliminar, selecciona editar área y luego procede a Eliminar. | |
| Resultado esperado: se puede adicionar, modificar o eliminar las áreas | |
| Evaluación: prueba realizada satisfactoriamente | |



Tabla 45: Gestionar SS

| Caso de prueba | |
|--|---------------------------|
| Número caso prueba:6 | Número historia usuario:2 |
| Nombre caso prueba: Gestionar SS | |
| Descripción: prueba para la funcionalidad que le permite al usuario adicionar, modificar, eliminar y mostrar las áreas. | |
| Condiciones de ejecución: el usuario debe estar autenticado | |
| Entradas: <ul style="list-style-type: none">❖ El usuario después de haberse autenticado selecciona la opción estructura/sindicatos/áreas inserta los datos requeridos y procede a Enviar❖ Para modificar el usuario selecciona el área y puede modificar sus datos | |
| Resultado esperado: los datos son registrados y el área se ha creado satisfactoriamente. | |
| Evaluación: prueba realizada satisfactoriamente | |

Tabla 46: Gestionar actividad

| Caso de prueba | |
|---|---------------------------|
| Número caso prueba: 7 | Número historia usuario:5 |
| Nombre caso prueba: Gestionar actividad | |



| |
|---|
| <p>Descripción: prueba para la funcionalidad que le permite al usuario adicionar, modificar, eliminar y mostrar las actividades.</p> <p>Prueba para la funcionalidad que le permite al usuario adicionar, modificar, eliminar y mostrar las áreas.</p> |
| <p>Condiciones de ejecución: el usuario debe estar autenticado</p> |
| <p>Entradas:</p> <ul style="list-style-type: none"> ❖ El usuario después de haberse autenticado selecciona la opción estructura/sindicatos/áreas inserta los datos requeridos y procede a Enviar ❖ Para modificar el usuario selecciona el área y puede modificar sus datos. |
| <p>Resultado esperado: los datos son registrados y el área se ha creado satisfactoriamente.</p> |
| <p>Evaluación: prueba realizada satisfactoriamente</p> |

Tabla 47: Gestionar avisos

| Caso de prueba | |
|--|-----------------------------------|
| Número caso prueba: 8 | Número historia usuario: 6 |
| Nombre caso prueba: Gestionar avisos | |
| Descripción: prueba para la funcionalidad que le permite al usuario adicionar, modificar, eliminar y mostrar las áreas. | |
| Condiciones de ejecución: el usuario debe estar autenticado | |



| |
|---|
| <p>Entradas:</p> <ul style="list-style-type: none"> ❖ El usuario después de haberse autenticado selecciona la opción estructura/sindicatos/áreas inserta los datos requeridos y procede a Enviar ❖ Para modificar el usuario selecciona el área y puede modificar sus datos. |
| <p>Resultado esperado: los datos son registrados y el área se ha creado satisfactoriamente.</p> |
| <p>Evaluación: prueba realizada satisfactoriamente.</p> |

Tabla 48: Dar baja del potencial

| Caso de prueba | |
|--|-----------------------------------|
| Número caso prueba: 1 | Número historia usuario: 1 |
| Nombre caso prueba: Dar baja del potencial | |
| Descripción: prueba para la funcionalidad que le permite al usuario dar baja del potencial a uno o varios trabajadores. | |
| Condiciones de ejecución: el usuario debe tener rol de financiero o secretario | |
| Entradas: el usuario después de haberse autenticado, selecciona administrar su SS, aparecen todos los trabajadores de dicha SS y ahí se escoge dar baja en el trabajador que desee. Luego te muestra una nueva vista donde se debe introducir fecha y motivo de la baja, muestra un mensaje de confirmación si está seguro que desea darle baja a ese trabajador, posteriormente se procede a enviar. | |



| |
|--|
| <p>Resultado esperado: se elimina el trabajador de la SS y se guarda para posteriormente ser utilizada la información que se necesite, se puede reincorporar o eliminar definitivamente de la SS.</p> |
| <p>Evaluación: prueba realizada satisfactoriamente</p> |

Tabla 49: Hacer cambio de escala

| Caso de prueba | |
|--|-----------------------------------|
| Número caso prueba: 9 | Número historia usuario: 8 |
| Nombre caso prueba: Hacer cambio de escala | |
| Descripción: prueba para la funcionalidad que le permite al usuario cambiar la escala del salario | |
| Condiciones de ejecución: el usuario debe estar autenticado con el rol de financiero o secretario. | |
| Entradas: el usuario selecciona, administrar su SS, escoge el trabajador que desea modificar y en operaciones escoge, editar, luego en el campo salario, introduce el nuevo salario del trabajador. | |
| Resultado esperado: los datos se modifican satisfactoriamente | |
| Evaluación: prueba realizada satisfactoriamente | |

Tabla 50: Generar reporte financiero

| Caso de prueba |
|----------------|
|----------------|



| | |
|---|------------------------------------|
| Número caso prueba: 10 | Número historia usuario: 10 |
| Nombre caso prueba: Generar reporte financiero | |
| Descripción: prueba para la funcionalidad que le permite al usuario generar el reporte financiero | |
| Condiciones de ejecución: el usuario debe estar autenticado y tener el rol de financiero o secretario. | |
| Entradas: | |
| Resultado esperado: se muestra el reporte de las finanzas. | |
| Evaluación: prueba realizada satisfactoriamente. | |

Tabla 51: Generar reporte emulación

| Caso de prueba | |
|---|------------------------------------|
| Número caso prueba: 10 | Número historia usuario: 10 |
| Nombre caso prueba: Generar reporte emulación | |
| Descripción: | |
| Condiciones de ejecución: el usuario debe estar autenticado | |
| Entradas: | |
| Resultado esperado: los datos son registrados y el área se ha creado satisfactoriamente. | |



Evaluación: prueba realizada satisfactoriamente.

Tabla 52: Modificar datos de usuario

| Caso de prueba | |
|--|------------------------------------|
| Número caso prueba: 10 | Número historia usuario: 12 |
| Nombre caso prueba: Modificar datos de usuario | |
| Descripción: prueba para la funcionalidad que le permite al usuario modificar los datos de un trabajador. | |
| Condiciones de ejecución: el usuario debe estar autenticado y tener el rol de financiero o secretario. | |
| Entradas: el usuario después de haberse autenticado selecciona la opción "Administrar su SS", selecciona el trabajador y escoge la opción "Editar", ahí puede modificar toda la información del trabajador. | |
| Resultado esperado: los datos son modificados satisfactoriamente. | |
| Evaluación: prueba realizada satisfactoriamente | |

Tabla 53: Listar próximas actividades

| Caso de prueba | |
|--|------------------------------------|
| Número caso prueba: 10 | Número historia usuario: 14 |
| Nombre caso prueba: Listar próximas actividades | |



| |
|--|
| Descripción: prueba para la funcionalidad que le permite al usuario ver todas las actividades próximas. |
| Condiciones de ejecución: el usuario debe estar autenticado. |
| Entradas: el usuario después de haberse autenticado selecciona la opción en el menú “Calendario” y selecciona por día, mes, año, para ver todas las próximas actividades. |
| Resultado esperado: se muestran todas las actividades. |
| Evaluación: prueba realizada satisfactoriamente |

Tabla 54: Ver trazabilidad de actividades

| Caso de prueba | |
|--|------------------------------------|
| Número caso prueba: 10 | Número historia usuario: 15 |
| Nombre caso prueba: Ver trazabilidad de actividades. | |
| Descripción: prueba para la funcionalidad que le permite al usuario mostrar las actividades que se han realizado. | |
| Condiciones de ejecución: el usuario debe estar autenticado. | |
| Entradas: el usuario después de haberse autenticado selecciona la opción “Calendario” y selecciona de que forma desea ver las actividades si por mes o por año. | |
| Resultado esperado: se muestran todas las actividades realizadas hasta la fecha. | |



Evaluación: Prueba realizada satisfactoriamente

Anexo 8. Preguntas para la encuesta de validación de la investigación

Preguntas que conforman la encuesta:

1. ¿Considera usted que los módulos agilizarán el trabajo forma tal que sea mucho más rápida la gestión de la información de los procesos?

___ Si ___ No

2. ¿Cree usted que con el uso de estos módulos se logrará tener mayor control de la información gestionada en dichos procesos?

___ Si ___ No

3. ¿Considera usted que la familiarización con los módulos puede retrasar el tiempo para llevar a cabo del proceso del cobro de las finanzas?

___ Si ___ No

4. ¿Usted cree que los módulos son fáciles de utilizar?

___ Si ___ No

5. ¿Considera usted que el uso de estos módulos para el Sistema de Gestión de Información del BUS UCI permita agilizar el trabajo y tener mayor control de la información gestionada en dichos procesos?