

Universidad de las Ciencias Informáticas



Carretera a San Antonio de los Baños Km 2½, Rpto. Torrens, La Habana

Facultad de Ciencias y Tecnologías Computacionales

Sistema Informático para la gestión de los indicadores asociados a la emulación en la UCI

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

Autor: Diana Loinaz Velázquez

Tutora: Ing. Mairena Arzuaga Limonta

La Habana, noviembre de 2023

“Año 65 de la Revolución”

DECLARACIÓN DE AUTORÍA

La autora del trabajo de diploma con título ***“Sistema Informático para la gestión de los indicadores asociados a la emulación en la UCI”*** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como únicos autores de su contenido.

Para que así conste firman la presente a los ____ días del mes de _____ del año _____.

Diana Loinaz Velázquez

Ing. Mairena Arzuaga Limonta

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Mairena Arzuaga Limonta, ingeniera en Ciencias Informáticas, profesor asistente, centro de Representación y análisis de datos. mlimonta@uci.cu

AGRADECIMIENTOS

Primero que todo quiero agradecer a mi familia que han formado parte muy importante en todo el transcurso de mi carrera. Sobre todo, a mi mamá por confiar en mí, por todo el amor y por todos los esfuerzos que hizo para que yo pudiera concluir mi carrera, por las noches de desvelo, por los días de preocupación y a mi papá por darme todo incluso cuando no podía, por todo lo que ha hecho por mí estos años. Mi hermana que la amo con la vida gracias por estar ahí en todo momento, junto a mis sobrinas Daniela y Dianela que las adoro. A mi tía que es como mi segunda madre, muchas gracias por todo, a mis primos que desde la distancia han formado parte muy importante en estos años. Mis abuelos que los amo, siempre están presente en mi corazón, a mi abuelo Pedro que no me pudo verme graduada en parte todo es por ti, a ti que en tu lecho de muerte me tuviste presente, aunque no estuve en ese momento. A mi padrino por su amor y preocupación.

A mi pareja por toda su ayuda, amor y por siempre estar ahí para mí en los buenos y malos momentos, juntos comenzamos este camino y juntos lo terminamos. Gracias por toda la ayuda y por todos estos años que compartimos tanto en la universidad como fuera. Gracias por todas las noches de explicaciones, por las peleas para que estudiara, por siempre ayudarme a encontrar la luz cuando todo se oscurecía en mi vida, cuando ya no creía que podía ahí estabas tú para hacerme ver lo contrario. Tengo mucho que agradecerte y lo que ponga aquí se queda corto con todo lo que has hecho por mí.

A mis amigos, Daymara, Michel, Basulto, Álvaro, Josman, ..., que más que amigos se han convertido en toda una familia, muchas gracias por todos estos años, los llevo en el corazón.

RESUMEN

El sindicato es la organización que agrupa a los trabajadores en Cuba. Específicamente en la Universidad de las Ciencias Informáticas (UCI) se encuentra afiliada al sindicato Nacional de Trabajadores Nacional de la Educación, la Cultura y el Deporte. Para llevar a cabo actividades sindicales en la universidad se cuenta con una estructura organizativa representada por el Buró Sindical Universitario (BUS). Debido a que entre los objetivos de la UCI se encuentra estimular a los trabajadores destacados, el BUS ha trazado una estrategia de emulación con el propósito de evaluar a los trabajadores con la mayor equidad posible. En esta estrategia se tienen en cuenta varios indicadores como la participación en las actividades convocadas. Pero medir estos indicadores resulta trabajoso ya que en ocasiones la asistencia a las actividades no es almacenada o no se lleva un control de los indicadores a medir en la emulación. La presente investigación tiene como objetivo el desarrollo de una aplicación web para la gestión de indicadores asociados a la emulación sindical en la UCI. La investigación está regida por la metodología AUP-UCI en su escenario cuatro: historias de usuario. Para el desarrollo del sistema se seleccionaron varias herramientas y tecnologías como: CSS3, JavaScript, HTML5, Node.js y Express. Las pruebas aplicadas al sistema arrojaron resultados satisfactorios y se usó como herramienta de prueba Jasmine. El sistema va a permitir la gestión de los indicadores asociados a la emulación en la UCI.

PALABRAS CLAVE

CTC, Emulación, Indicadores, Sindicato, Secciones Sindicales

ABSTRACT

The union is the organization that brings together workers in Cuba. Specifically in the University of Computer Sciences (UCI) is affiliated with the National Union of National Workers of Education, Culture and Sports. To carry out union activities at the university, there is an organizational structure represented by the University Union Bureau (BUS). Because one of the UCI's objectives is to encourage outstanding workers, the BUS has outlined an emulation strategy with the purpose of evaluating workers with the greatest possible equity. In this strategy, several indicators are taken into account, such as participation in the planned activities. But measuring these indicators is difficult since sometimes attendance at activities is not stored or control of the indicators to be measured in the emulation is not kept. The objective of this research is the development of a web application for the management of indicators associated with union emulation in the UCI. The research is governed by the AUP-UCI methodology in its scenario four: user stories. For the development of the system, several tools and technologies were selected such as: CSS3, JavaScript, HTML5, Node.js and Express. The tests applied to the system gave satisfactory results and Jasmine was used as a testing tool. The system will allow the management of the indicators associated with emulation in the UCI.

KEYWORDS

CTC, Emulation, Indicators, syndicate, Union Sections

ÍNDICES

INTRODUCCIÓN	12
CAPÍTULO I: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio	17
I.1 Sistema web para la gestión de indicadores asociados a la emulación sindical en la UCI.	17
I.2 Gestión de indicadores asociados a la emulación sindical.	17
I.2.1 Sistemas de gestión de la información.	18
I.2.2 Sistemas de gestión de la información de la emulación.	18
I.3 Análisis de soluciones similares.	20
I.3.1 Soluciones similares a nivel internacional.	20
I.3.2 Soluciones similares a nivel nacional.	20
I.4 Metodología de desarrollo de software.	21
I.5 Herramientas y tecnologías.	23
Conclusiones del capítulo	29
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO	30
II.1 Propuesta de solución.	30
II.2 Modelo de Dominio.	30
II.3 Descripción del modelo de dominio.	31
II.4 Levantamiento de requisitos	32
II.4.1 Requisitos funcionales.	32
II.3.2 Requisitos no funcionales.	36
II.4 Historias de Usuarios	36
II.5 Arquitectura del Software.	37
II.6 Patrones de Diseño.	39
II.6.1 Patrones GRASP.	39
II.6.2 Patrones GOF.	40
II.7 Diagramas de Clases de Diseño.	40
II.8 Modelo de datos.	41
Conclusiones del capítulo	42
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS.	43

III.1 Estándares de codificación.	43
III.3 Pruebas	43
III.3.1 Pruebas unitarias	44
III.3.2 Pruebas de validación	46
III.3.2 Pruebas de aceptación	50
Conclusiones del capítulo	51
CONCLUSIONES FINALES	52
RECOMENDACIONES	53
REFERENCIAS BIBLIOGRÁFICAS	54
ANEXOS	57

ÍNDICE DE TABLAS

<i>Tabla 1: Descripción del modelo de dominio.</i>	<i>32</i>
<i>Tabla 2: Requisitos Funcionales.</i>	<i>35</i>
<i>Tabla 3: Historia de usuario autenticar usuario</i>	<i>37</i>
<i>Tabla 13: Caso de prueba de partición de equivalencia del RF1 Autenticar Usuario</i>	<i>48</i>
<i>Tabla 1: Descripción del modelo de dominio.</i>	<i>57</i>
<i>Tabla 2: Requisitos Funcionales.</i>	<i>60</i>
<i>Tabla 3: Historia de usuario autenticar usuario</i>	<i>61</i>
<i>Tabla 4: Historia de usuario gestionar quejas</i>	<i>62</i>
<i>Tabla 5: Historia de usuario gestionar respuesta a nivel UCI</i>	<i>63</i>
<i>Tabla 6: Historia de usuario gestionar trabajador</i>	<i>63</i>
<i>Tabla 7: Historia de usuario gestionar indicadores</i>	<i>63</i>
<i>Tabla 8: Historia de usuario gestionar emulación por trabajador</i>	<i>64</i>
<i>Tabla 9: Historia de usuario gestionar cargo</i>	<i>64</i>
<i>Tabla 10: Historia de usuario gestionar sección sindical</i>	<i>65</i>
<i>Tabla 11: Historia de usuario gestionar emulación por sección sindical</i>	<i>65</i>
<i>Tabla 12: Historia de usuario gestionar actividades</i>	<i>66</i>
<i>Tabla 13: Caso de prueba de partición de equivalencia del RF1 Autenticar Usuario</i>	<i>67</i>
<i>Tabla 14: Caso de prueba de partición de equivalencia del RF10 Registrar trabajador</i>	<i>69</i>

ÍNDICE DE FIGURAS

<i>Figura 1: Modelo de dominio</i>	31
<i>Figura 2: Historia de usuario autenticar usuario</i>	37
<i>Figura 2.1: Arquitectura Modelo-Vista-Controlador.</i>	38
<i>Figura 6: Diagrama clase de diseño</i>	41
<i>Figura 7: Modelo de datos</i>	42
<i>Figura 8: Prueba unitaria Jasmine con errores</i>	45
<i>Figura 9: Prueba unitaria Jasmine con resultados correctos</i>	45
<i>Figura 10: Gráfica de pruebas unitarias</i>	46
<i>Figura 11: Gráfica de pruebas funcionales.</i>	49
<i>Figura 2: Historia de usuario autenticar usuario</i>	61
<i>Figura 3: Historia de usuario gestionar quejas</i>	62
<i>Figura 4: Historia de usuario gestionar respuesta a nivel UCI</i>	62
<i>Figura 5: Historia de usuario gestionar trabajador</i>	63

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada

INTRODUCCIÓN

El uso de las TIC se ha convertido en un elemento indispensable para establecer las líneas de desarrollo de la sociedad cubana, buscando dar solución a los problemas del hombre. Han transformado la manera de trabajar, optimizando recursos y haciéndolos más productivos. Las TIC comprenden el conjunto de recursos y soluciones tecnológicas que posibilitan la recopilación, el procesamiento y transmisión de datos. (1.1)

Cuba ha identificado desde muy temprano la necesidad de introducir en la práctica social las TIC y lograr una cultura digital, por lo que se crea la Universidad de las Ciencias Informáticas (UCI). En la UCI se vinculan la docencia, producción e investigación, con el objetivo de desarrollar software de alta calidad, lo que facilitará que la sociedad logre acercarse más hacia el objetivo de un desarrollo justo. Su misión es "Formar profesionales comprometidos con la Patria y altamente calificados en la rama de la Informática, además de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación". (1)

Los trabajadores de la UCI, como casi todos los trabajadores del país, pertenecen al movimiento sindical cubano. El sindicato tiene como misión por una parte organizar a los trabajadores preparándolos para movilizarlos en el cumplimiento de sus deberes laborales y sociales en la construcción socialista y, por otra parte, representar y defender sus legítimos derechos. Los trabajadores tienen su máxima representación en la Central de Trabajadores de Cuba (CTC) que es una organización de masas, integrada por los sindicatos y sus afiliados, organizados sindicalmente de forma voluntaria para defender sus intereses.

La CTC tiene como principal característica su arraigado sentido de unidad de los trabajadores, su esencia democrática y su participación en la lucha por los intereses de los trabajadores y la defensa de sus derechos laborales, sociales, culturales, civiles y políticos. Así como la reafirmación de la defensa de la Patria y la integridad e identidad nacional, plasmado en sus Estatutos y Resoluciones.

Para estimular al trabajador la CTC utiliza como instrumento la Emulación Socialista, un método que permite estimular el esfuerzo para alcanzar en el trabajo cotidiano resultados superiores. La organización de la Emulación Socialista en el centro de trabajo debe tener en cuenta tres momentos: emular, evaluar y estimular. (2)

Para ejercer una labor sindical en la universidad, se dispone de una estructura organizativa compuesta por un Buró Sindical Universitario (BUS) como máxima representación. El cual atiende varias esferas donde cada miembro del Buró se especializa en una esfera diferente, pero todos colaboran entre sí para trabajar en función de los trabajadores. Una de estas esferas es la esfera de la Emulación, donde el buró universitario ha adoptado una estrategia de emulación para fomentar la competitividad y motivación de los trabajadores, promoviendo así un ambiente laboral más productivo y eficiente.

La estrategia de la emulación tiene en cuenta una serie de indicadores, como son: los resultados laborales del trabajador, la realización de la guardia obrera, participación a las asambleas de las secciones sindicales, participación en las actividades convocadas por el CTC y por la universidad y las donaciones de sangre. Partiendo de este análisis se otorgan las categorías emulativas como: cumplidor, destacado o vanguardia. Es por ello que se debe almacenar toda la información asociada a ello.

La emulación en las secciones sindicales es dirigida y controlada por los secretarios generales de dicha sección sindical, ellos son los que deben evaluar los indicadores a medir en la emulación sindical. Es muy común que se olvide en qué actividades participó cada trabajador, si realizó o no la guardia obrera, ya que esta información no es almacenada y a la hora de realizar el chequeo de emulación tienden hacer uso excesivo de la memoria.

Lo anteriormente planteado puede provocar que se sea injusto a la hora de seleccionar a un trabajador destacado en relación a otro compañero. Por esto existe la necesidad de tener un mayor control y manejo de la información para garantizar que los dirigentes sindicales realicen una emulación coherente, ya que puede darse el caso que no se reconozca a nivel de universidad a los trabajadores que más lo merezcan, trayendo consigo el resultado contrario al deseado.

El Buró Sindical Universitario para evaluar los indicadores se almacena su evidencia, ya sean fotos, información solicitada o actas. Estas evidencias son enviadas por correo o por un grupo en WhatsApp creado por los secretarios. Estos dos medios de comunicación tienen limitaciones que hacen que la recepción de la información se dificulte. Por ejemplo, mediante el correo solo se pueden enviar archivos de hasta 10 mb y el correo del sindicato tiene una capacidad máxima de almacenamiento de 250 mb que se llena muy rápido y los miembros del buró se ven en la necesidad de eliminar los correos anteriores pudiendo eliminar información importante. Cuando existe demasiada información en el grupo de WhatsApp resulta engorroso encontrar la información que se está buscando.

Para encontrar cada uno de los indicadores a evaluar en la emulación es necesario revisar cada una de estas evidencias una por una para obtener el listado de trabajadores que cumplen un requisito específico o para seleccionar a los trabajadores destacados. Esto implica que el miembro del buró encargado de la emulación ocupe demasiado tiempo en la búsqueda de esta información. Además, para encontrar el nivel de participación de los trabajadores en una actividad en específico se hace muy difícil porque al igual tendría que buscarlas en cada una de estas evidencias.

Considerando la problemática anteriormente descrita, se plantea como **problema de investigación**: ¿Cómo apoyar el proceso de emulación sindical a través de la gestión de indicadores en la Universidad de las Ciencias Informáticas?

Para la realización de la investigación se define como **objeto de estudio** gestión de indicadores asociados a la emulación sindical, enmarcándose en el **campo de acción** Sistema Informático para la gestión del proceso de emulación sindical de los trabajadores basado en las tecnologías web.

Objetivo general: Desarrollar un sistema web para la gestión de indicadores para el apoyo del proceso de emulación sindical en la UCI mediante el uso de tecnologías web.

Para solucionar la problemática existente el objetivo general ha sido desglosado en los siguientes **objetivos específicos**:

- Elaborar un marco teórico de la investigación.

- Realizar una propuesta de solución a partir del análisis de la problemática dada en la gestión de indicadores asociados a la emulación sindical.
- Definir las funcionalidades y el diseño que debe cumplir el sistema.
- Implementar una aplicación web para la gestión de información que cumpla con los requerimientos definidos.
- Realizar pruebas a la aplicación web implementada para validar que las funcionalidades cumplan con los requerimientos definidos.

Para el cumplimiento de los objetivos propuestos se utilizarán los siguientes **métodos de la investigación**:

Métodos teóricos:

- Analítico-Sintético: se utilizó este método para el estudio de diferentes bibliografías sobre gestión de indicadores asociados a la emulación sindical apoyado en los indicadores correspondientes.
- Inducción-Deducción: se utilizó para recopilar datos, identificar patrones y en la toma de decisiones basadas en evidencias.
- Histórico-Lógico: se utilizó para realizar un estudio de las soluciones similares empleadas por otras instituciones. Permitió conocer la evolución y desarrollo del tema de investigación.
- Modelación: Permitió la creación de modelos. Los artefactos generados constituyen una representación visual del sistema a desarrollar.

Métodos empíricos:

- Entrevista: se realizaron entrevistas a los principales miembros del sindicato de la Universidad de las Ciencias Informáticas para investigar datos fundamentales sobre la organización.

- Análisis documental: consiste en examinar y analizar diferentes tipos de documentos para extraer información útil para el aporte a la investigación.
- Tormenta de ideas: busca analizar y mejorar los detalles, esto implica revisar y pulir los elementos claves de la solución, considerando su viabilidad, eficacia y eficiencia.

El enfoque central de la investigación se centra en analizar cómo gestionar los indicadores asociados a la emulación sindical en la UCI y proponer soluciones. A lo largo de los siguientes capítulos, se presentará un análisis detallado de los aspectos clave relacionados con el tema, brindando una visión integral para comprender y abordar el problema planteado.

Capítulo 1: Fundamentos y referentes teóricos-metodológico sobre el objeto de estudio.

En este capítulo se describe los conceptos fundamentales para la realización de la investigación. Se identifican las metodologías, tecnologías y herramientas utilizadas para facilitar el desarrollo de la solución.

Capítulo 2: Diseño de la solución al problema a resolver.

Este capítulo se centra en presentar la propuesta de solución al problema de investigación planteado, se realiza el modelo de dominio con su descripción, el levantamiento de requisitos funcionales y no funcionales y se presentan las diferentes historias de usuarios con cada una de sus descripciones. También se realiza un breve estudio de la arquitectura del software y se muestran los diferentes patrones de diseño que se van a utilizar.

Capítulo 3: Implementación y pruebas

Este capítulo se enfoca en describir el proceso de pruebas realizado para evaluar y validar la solución propuesta. Se describen cómo se llevaron a cabo las pruebas, la ejecución de las mismas, se presentan los resultados obtenidos de las pruebas realizadas y se realiza un análisis de los resultados de las pruebas, evaluando si la solución propuesta cumple con los requisitos establecidos y si se ha alcanzado los objetivos de pruebas.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO

En el presente capítulo se abordan aspectos relacionados con el objeto de estudio definido para el problema planteado. Se hace referencia a conceptos relacionados al tema, se realiza un análisis de las diferentes metodologías, herramientas y leguajes a utilizar para la realización de la aplicación web para dar solución al problema ante descrito. Realizar una revisión de diversas aplicaciones y sitios web relacionados con instituciones que presentan semejanzas entre sí.

I.1 Sistema web para la gestión de indicadores asociados a la emulación sindical en la UCI.

Para lograr un mayor entendimiento del objeto de estudio se definen diferentes conceptos para obtener información con el fin de comprender mejor su importancia y utilidad en la investigación:

Sindicato: es una asociación de trabajadores que se establecen en dicha agrupación para velar por diferentes intereses, beneficios y procedimientos de tipo económico, profesional y social que afectan a los trabajadores. (3)

Emulación Socialista: es un instrumento para desarrollar el trabajo político ideológico, un método que permite estimular el esfuerzo para alcanzar en el trabajo cotidiano resultados superiores; lograr que cada trabajador sea más eficiente y productivo en su puesto de labor y lo realice con entrega, sacrificio y compromiso revolucionario, consciente de que solo trabajando duro saldremos adelante y podremos impulsar la economía del país. (4)

I.2 Gestión de indicadores asociados a la emulación sindical.

La gestión de indicadores asociados a la emulación sindical es un tema importante en el ámbito laboral. Se refiere a la implementación de estrategias y medidas para promover la participación activa de los sindicatos. Los indicadores juegan un papel fundamental, son herramientas de medición que permiten evaluar el desempeño y los resultados obtenidos por los trabajadores.

I.2.1 Sistemas de gestión de la información.

Una adecuada gestión de la información, posibilita reducir los riesgos en la administración de la organización, como son la toma de decisiones apresuradas, tardías o inconsistentes, la entrada al mercado con productos no competitivos, entre otros, que ocasionan pérdidas y reducen su competitividad en el mercado. Obtener la información necesaria, con la calidad requerida, es una premisa indispensable para la supervivencia de las empresas, si se considera que las organizaciones acortan cada vez más sus ciclos estratégicos y que la toma de decisiones, así como el cambio es continua. (5)

I.2.2 Sistemas de gestión de la información de la emulación.

La emulación socialista en Cuba se enfoca en la competencia sana entre empresas y organizaciones para lograr un mayor desarrollo económico y social dentro del marco del socialismo. Este sistema se ha implementado en Cuba durante muchos años y ha sido clave para el desarrollo de la economía del país. (6)

Según la revista Emulación socialista y gestión empresarial en Cuba, la emulación socialista tiene varios lineamientos que se encuentran en una revisión constante para adaptarse a las necesidades y desafíos del país. Algunos de estos lineamientos son:

- Establecimiento de objetos claros y específicos para las empresas u organizaciones.
- Utilización de incentivos y reconocimientos públicos para motivar a los trabajadores y mejorar la productividad.
- Competencia saludable y cooperación entre las empresas y organizaciones para fomentar la innovación y el desarrollo tecnológico.
- Supervisión y control para garantizar el cumplimiento de los objetivos y la calidad en la producción. (7)

Estos lineamientos, aunque no se mencionan directamente en el Código del Trabajo, son promovidos como una herramienta para mejorar la productividad y la eficiencia en el trabajo.

El Código del Trabajo es una ley fundamental, ya que establece los derechos y deberes de los trabajadores, así como las normas y regulaciones para el funcionamiento de las relaciones laborales. En su Capítulo 2 hace referencia a las Organizaciones Sindicales, donde su Artículo 12 dice: *“ El Estado reconoce y estimula a las organizaciones sindicales que agrupan en su seno a los trabajadores de los diferentes sectores y ramas de la economía y representan sus derechos e intereses específicos, con independencia de la naturaleza o características de su relación de trabajo”*. También hace referencia a impulsar y estimular la creatividad de los colectivos laborales, organizar y dirigir la emulación entre los trabajadores para así lograr un incremento en los niveles de eficiencia e incentivar la estimulación a trabajadores y entidades en su correspondencia con los resultados alcanzados en su trabajo. (8)

En la UCI, el Código del Trabajo se aplica para proteger los derechos laborales y promover el desarrollo profesional de los trabajadores y la emulación se fomenta como una práctica para mejorar la calidad y eficiencia en la investigación y el desarrollo tecnológico.

Existen varios sindicatos en Cuba, entre ellos está el sindicato nacional de los trabajadores, la cultura y el deporte al cuál se asocia la UCI.

Entre los objetivos de la UCI se encuentra lograr mediante la emulación un ambiente de motivación, entusiasmo, dinamismo y competencia fraternal entre los trabajadores, que permita la eficiencia económica, la productividad, el ahorro, la disciplina y la cultura por la calidad del trabajo. El sistema de emulación en la UCI está integrado por dos componentes fundamentales: emulación individual y emulación colectiva. En cada una de ellas se tendrán en cuenta diferentes aspectos o indicadores a medir, que deben cumplir los trabajadores o áreas. (4)

En la universidad se realiza la emulación individual donde se lleva a cabo cuatro cortes en el año donde en cada corte se seleccionan los trabajadores cumplidores e incumplidores, después se realiza la emulación anual donde se seleccionan a los trabajadores vanguardias que hayan salido como cumplidores en los cuatro trimestres. También se realiza la emulación por secciones sindicales donde se tienen en cuenta varios indicadores como son la

realización de actividades, las iniciativas realizadas por cada sección sindical y si envían en tiempo la información solicitada por el Buró Sindical.

También se realiza la emulación de cada uno de los trabajadores donde se tienen en cuenta varios indicadores entre los que se encuentran, la realización de la guardia obrera, la participación en las actividades convocadas por la universidad y por la CTC, si es donante de sangre y los resultados labores obtenidos.

Actualmente en la esfera de la emulación de la universidad se pasa mucho trabajo a la hora de buscar los trabajadores destacados debido que las áreas no llevan una debida gestión de los indicadores, ya que casi nunca tienen toda la información asociada a las actividades realizadas y convocadas, por lo tanto, es muy difícil llegar a final de año con el propósito que se quiere, que es seleccionar los trabajadores vanguardias.

I.3 Análisis de soluciones similares.

A continuación, se realiza un estudio de las diferentes soluciones existentes para conocer sus diferentes características y funcionalidades.

I.3.1 Soluciones similares a nivel internacional.

UGT (Unión General de Trabajadores de España): portal web diseñado para fomentar la actividad sindical en España, está compuesto por varias funcionalidades para facilitar a los usuarios su interacción, tiene acceso a revistas, la prensa, vídeos, fotos relacionados con la actividad sindical. (9)

I.3.2 Soluciones similares a nivel nacional.

SNTECD (Sindicato Nacional de trabajadores de la educación, el deporte y la salud): Es el de mayor membresía en Cuba, llamado a reconocer, estimular y enaltecer la labor de sus afiliados. El portal web tiene gran cantidad de documentos relacionados a la historia, objetivos, eventos que se realizan, también ofrece noticias nacionales como internacionales. (10)

Sistema de Gestión de la Central de Trabajadores de Cuba (SigestCTC): Es una aplicación web que permite la gestión de la información y obtener datos sobre los comités y burós

municipales, posibilita la realización de reportes de bajas, gestionar datos de los municipios y provincias. Tiene representación en cada una de las provincias y municipios del país, lo que permite obtener una estructura organizativa. (11)

Sistema de gestión para la sección sindical Vicerrectoría Primera de la UCI: Esta aplicación web permite mejorar y agilizar el trabajo de los sindicalistas y la calidad de la información. El funcionamiento del sistema permite insertar, modificar, gestionar afiliado y gestionar la Dirección Administrativa y Comité Sindical. Entre los procesos que analiza se encuentra el pago de finanzas, y registro de opinión de cada afiliado de los cuáles no se hace el proceso de divulgación de la información. (12)

Portal Web para la CTC de la Facultad 6: Este portal web permite gestionar los procesos vinculados a la gestión financiera, emulación, información de actividades que se vayan a realizar y las secciones sindicales. Este portal no realiza el proceso de divulgación de la información. (13)

Estos son algunos de los sistemas de gestión sindical existentes, pero estos presentan algunos impedimentos para su uso ya que no se encuentran actualizados por lo que no se puede realizar una gestión efectiva. No se tiene acceso al código fuente de los sistemas de gestión sindical existentes lo que tiene gran importancia ya que esto permitiría su actualización y mejora continua. Además de los inconvenientes mencionados, los sistemas existentes también presentan problemas al momento de emular un trabajador, es decir, cuando se necesita realizar un seguimiento detallado de las acciones realizadas por cada trabajador. Por lo tanto, el nuevo sistema de gestión sindical debe ser capaz de emular a un trabajador de manera eficiente y precisa, para poder brindar un mejor servicio.

I.4 Metodología de desarrollo de software.

Las metodologías de desarrollo de software se utilizan en el ámbito de la programación con el objetivo de trabajar en equipo de manera más organizada, se clasifican en dos grupos: ágiles y tradicionales. (14)

Las metodologías ágiles se basan en dos aspectos puntuales, el retrasar las decisiones y la planificación adaptativa, permitiendo aún más el desarrollo de software a gran escala. También se basa en una metodología incremental, en la que cada ciclo de desarrollo se va agregando nuevas funcionalidades a la aplicación final, los ciclos son mucho más cortos y rápidos. (14)

Las metodologías tradicionales se focalizan en documentación, planificación y procesos y los ciclos de desarrollo son poco flexibles. Estas metodologías no se adaptan nada bien a los cambios, centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto en la fase inicial del desarrollo del proyecto. (14)

La UCI desarrolló una versión de la metodología AUP, con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. (16)

Las fases de la metodología AUP-UCI son:

- Inicio
- Ejecución
- Cierre

La metodología AUP-UCI tiene 4 escenarios:

- Escenario 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

- Escenario 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU (17)

Después del estudio de los diferentes tipos de metodologías ágiles y tradicionales y realizar un estudio de los diferentes tipos de metodologías, se decide hacer uso de la metodología AUP-UCI en su escenario 4. Esta metodología consiste en centrar la atención en actividades que son esenciales para el desarrollo, no en todas las que forman parte del proyecto. Con el uso de esta metodología se puede utilizar cualquier conjunto de herramientas. Además porque emplea un equipo de desarrollo pequeño, esta es una metodología híbrida, es decir, no va hacer robusta ni ágil y al estar en un punto intermedio permite generar documentación adecuada, posibilitando hacer un sistema como desee el usuario y el desarrollador.

1.5 Herramientas y tecnologías.

A continuación, se describen las herramientas y tecnologías a utilizar para el desarrollo de la Aplicación Web para la gestión de indicadores asociados a la emulación sindical de la UCI, entre ellas se encuentran las siguientes:

Visual Paradigm: Para el modelado se utiliza el Visual Paradigm en su versión 15.1. Es un modelador que permite diseñar todo tipo de diagramas UML, el equipo de desarrollo de software puede realizar análisis y diseño de sistemas con facilidad. Ofrece un modelado visual completo y cuenta con un equipo de desarrollo y soporte dedicado a proporcionar actualizaciones regulares, lo que garantiza que la herramienta se mantenga actualizada. ES una potente herramienta que genera código a partir de diagramas y genera documentación. (18)

Ventajas que proporciona Visual Paradigm:

- Dibujo.
- Corrección sintáctica.
- Coherencia entre diagramas.
- Integración con otras aplicaciones.

- Trabajo multiusuario.
- Reutilización.
- Generación de código.
- Generación de informes. (18.1)

CSS3 (Cascading Style Sheets u Hojas de Estilo en Cascada): tiene el fin de separar la estructura de la presentación. Es un lenguaje constituido a base de selectores, propiedades y valores que tiene como objetivo dar aspecto visual al código HTML. Algunas de las características que presenta CSS3 son:

- Capacidad para diseñar y estilizar páginas web de manera avanzada.
- Es compatible con múltiples dispositivos, tiene capacidad para crear animaciones y transiciones, como transformaciones 2D y 3D, rotaciones y efectos de sombra.
- Control de la apariencia de las fuentes.
- Control avanzado sobre los bordes y las esquinas de los elementos. (19)

JavaScript: Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas, que es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. JavaScript es un lenguaje de programación basado en prototipos y se ejecuta en el lado del cliente de la web. Presenta varias características como son: lenguaje de alto nivel, dinámico, multi paradigma, sensible a mayúsculas y minúsculas, no necesita los puntos y comas al final de cada línea. Es utilizado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. (20)

A continuación, se describen algunas de las características que presenta JavaScript:

- Lenguaje del lado del cliente.

- Lenguaje orientado a objetos: utiliza clases y objetos como estructuras que permiten organizarse de forma simple y son reutilizables durante el desarrollo.
- No es necesario aclarar el tipo de dato al declarar una variable. Esta característica supone una gran ventaja a la hora de ganar rapidez programando.
- Lenguaje interpretado: utiliza un intérprete que permite convertir las líneas de código en el lenguaje de la máquina. Esto tiene un gran número de ventajas como la reducción del procesamiento en servidores web al ejecutarse directamente en el navegador del usuario, o que es apto para múltiples plataformas permitiendo usar el mismo código.
- Muy utilizado por desarrolladores: en la actualidad es uno de los lenguajes más demandados por su versatilidad y su infinita capacidad para crear plataformas cada vez más atractivas.

HTML5 (Hyper Text Markup Language): Lenguaje de Marcado de Hipertexto, es utilizado para la creación de páginas web estáticas. Consiste en una serie de elementos que se utilizan para encarar diferentes partes del contenido para que se vean y se comporten de una manera determinada. Permite describir hipertexto y tiene un despliegue rápido. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones. Es la 5ta versión más importante del estándar e introduce algunos elementos que constituyen ventajas de su empleo y hacen que se ajuste a las necesidades actuales de la web. Presenta soporte para contenidos multimedia que permite la reproducción de contenidos multimedia directamente del navegador. Algunas de las principales características de HTML5 son:

- Sirven para organizar y estructurar una web.
- Permiten definir jerarquías o niveles de orden.
- Son compatibles con todo tipo de dispositivos. (21)

Sistema de gestión de base de datos: PostgreSQL: Un sistema de gestión de base de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos. PostgreSQL es un sistema de gestión de base de datos objeto-relacional. Es uno de los sistemas de gestión de base de datos de código abierto más potente del mercado. Ofrece las siguientes características:

- Estabilidad y confiabilidad.
 - Potencia.
 - Robustez.
 - Facilidad de administración e implementación de estándares.
 - Funciona muy bien con grandes cantidades de datos.
 - Alta concurrencia de usuarios accediendo a la misma vez.
 - Ofrece fiabilidad y estabilidad y es capaz de manejar grandes volúmenes de datos.
- (22)

Dentro de las principales ventajas se encuentran:

- Soporta distintos datos.
- Es extensible a través de su código fuente, disponible sin costos adicionales.
- Es multiplataforma, disponible en Linux, Unix y Windows, entre otros.

PgAdmin: está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Su interfaz gráfica soporta todas las características de PostgreSQL y facilita de gran manera la administración ya que permite desde hacer búsqueda SQL hasta desarrollar toda la base de datos de forma fácil e intuitiva: directamente desde la interfaz gráfica. PgAdmin permite

acceder a todas las funcionalidades de la base de datos, consulta, manipulación y gestión de datos.

Principales características:

- Ofrece una interfaz gráfica intuitiva y fácil de usar
- Permite administrar usuarios, roles y permisos.
- Permite ejecutar consultas SQL en base de datos PostgreSQL
- Permite obtener copias regulares de la base de datos o la ejecución de scripts SQL en momentos específicos. (23)

Node.js: es un entorno de ejecución de código JavaScript en el lado del servidor que permite a los desarrolladores crear aplicaciones altamente escalables y de alto rendimiento. Permite a los desarrolladores mejorar la eficiencia, gestionar dependencias, automatizar tareas y realizar pruebas para crear aplicaciones más eficientes y robustas. Algunas de las características claves de node.js son:

- Velocidad: las aplicaciones de node.js pueden manejar grandes cantidades de solicitudes entrantes con tiempos de respuesta rápidos.
- Escalabilidad: es altamente escalable y se puede utilizar para construir aplicaciones de alto rendimiento que pueden manejar grandes volúmenes de tráfico.
- Plataforma cruzada: es compatible con varias plataformas, incluyendo Windows, Linux y macOS, lo que hace muy flexible y fácil de utilizar en diferentes entornos de desarrollo.
- Comunidad activa: tiene una gran y activa comunidad de desarrolladores que contribuyen con nuevas características, bibliotecas y soluciones a problemas comunes. (24)

Express: es un framework web de código abierto para Node.js, es conocido por su simplicidad, permite crear rápidamente aplicaciones web y APIs, proporcionando una sintaxis clara y un conjunto de funcionalidades esenciales. Express proporciona una base sólida para desarrollar aplicaciones web de una sola página, sitios web e incluso aplicaciones más complejas con la ayuda de bases de datos. Existen varias razones por las que usar express como son:

- Sencillez y facilidad de uso: Su estructura de código es intuitiva y fácil de entender.
- Popularidad y documentación: Cuenta con una amplia comunidad de desarrolladores.
- Versatilidad: Puede adaptarse a diferentes proyectos y necesidades. (25)

Visual Studio Code: es un software libre y multiplataforma, cuenta con soporte para depuración de código y dispone de un sinnúmero de extensiones que da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. Tiene gran rendimiento y compatibilidad con múltiples plataformas. Es un editor de código fuente altamente eficiente y rápido y es ligero en cuanto consumo de recursos. Una de las características más importantes de este software, se puede actualizar con frecuencia y rapidez de actualización de la extensión. Visual Studio Code no sólo es una herramienta de código, se pueden instalar muchas extensiones para compilar, depurar y embellecer el código. (26)

Características de Visual Studio Code:

- Hay gran cantidad de extensiones con la posibilidad de actualizaciones automáticas.
- Las actualizaciones del programa son de forma rápida y ordenada.
- Código de autocompleto.
- Texto terminal de comandos.
- Capacidad para utilizar el control de código fuente.

Conclusiones del capítulo

El análisis realizado a partir de los métodos de investigación científica ha proporcionado una definición clara de los conceptos asociados al tema. El análisis de soluciones similares ha permitido comprender mejor las funcionalidades para el desarrollo de la propuesta de solución. La correcta selección de metodología, herramientas y tecnologías va permitir desarrollar una solución adecuada al problema de la investigación determinar las más adecuadas.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

En este capítulo se realizará una breve descripción de las características y diseño que presentará el sistema. Se realizará la especificación de los requisitos funcionales y no funcionales del sistema informático, estos requisitos son fundamentales para definir y especificar las funcionalidades y características que debe cumplir el sistema, tanto en términos de comportamiento como también los aspectos de calidad que debe satisfacer.

II.1 Propuesta de solución.

La propuesta de solución consiste en desarrollar un sistema de gestión de indicadores que integre diferentes funcionalidades. El sistema debe contar con mecanismos para recopilar datos relevantes, como registro de actividades, reuniones y otros indicadores, estos datos se pueden ingresar manualmente. El sistema debe permitir realizar un seguimiento continuo de los indicadores. Debe garantizar la seguridad, el acceso seguro a la información, restringiendo los permisos según el rol y responsabilidades de cada usuario.

II.2 Modelo de Dominio.

El modelado de dominio o conceptual consiste en identificar y definir los conceptos claves, sus relaciones y las reglas de negocio que rigen el funcionamiento del sistema. También puede servir como base para el diseño y la implementación de la arquitectura del software, ya que proporciona una visión general de las entidades y las interacciones claves dentro del sistema. (27)

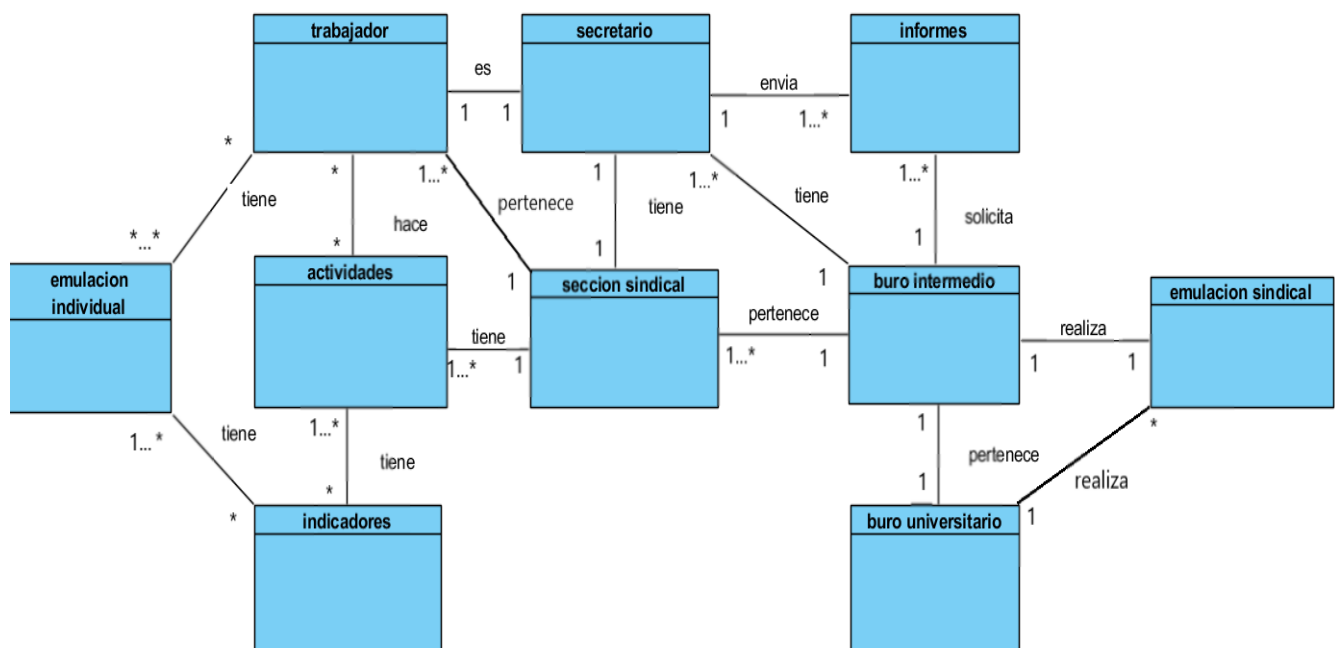


Figura 1: Modelo de dominio

II.3 Descripción del modelo de dominio.

Conceptos	Descripción
Trabajador	Es una persona que realiza una actividad laboral en la universidad.
Secretario	Persona que brinda apoyo administrativo y de gestión en la organización.
Sección Sindical	Actúa como órgano representativo de los trabajadores.
Buró Intermedio	Actúa entre dos o más partes facilitando la comunica-

	ción, coordinación y colaboración entre ellas.
Buró Sindical Universitario	Es un órgano de representación y organización sindical conformados por los trabajadores de la universidad.
Actividades	Acciones que se llevan a cabo con un propósito específico.
Emulación individual	Actúa como fuente de motivación y crecimiento en el ámbito profesional.
Emulación sindical	Promueve el intercambio de conocimientos entre los sindicatos, contribuyendo a fortalecer el movimiento sindical.
Indicadores	Son medidas o variables para medir el desempeño y los resultados alcanzados.

Tabla 1: Descripción del modelo de dominio.

II.4 Levantamiento de requisitos

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Es la fase donde los desarrolladores y el cliente trabajan juntos para identificar y documentar los requisitos del proyecto. (28)

II.4.1 Requisitos funcionales.

Los requisitos funcionales describen con detalles las funcionalidades del software, sus entradas, salidas y excepciones. Al establecer los requisitos funcionales de manera clara y precisa, se facilita la comunicación y colaboración entre los involucrados en el proyecto, ya que proporcionan una guía clara durante todas las etapas del ciclo de vida del proyecto, desde el diseño inicial hasta la implementación y pruebas. (28)

N°	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar Usuario	El sistema debe permitir autenticar al usuario que tenga los permisos adecuados para acceder.	Alta	Alta
RF2	Gestionar quejas UCI			
2.1	Registrar quejas UCI	El sistema debe permitir registrar una queja.	Alta	Media
2.2	Modificar quejas UCI	El sistema debe permitir modificar una queja.	Media	Baja
2.3	Listar quejas UCI	El sistema debe permitir listar de las quejas.	Alta	Baja
2.4	Eliminar quejas UCI	El sistema debe permitir eliminar las quejas.	Baja	Baja
RF3	Gestionar respuesta de las quejas.			
3.1	Registrar respuesta de las quejas	El sistema debe permitir registrar las respuestas de las quejas.	Alta	Media
3.2	Modificar respuesta de las quejas	El sistema debe permitir modificar las repuestas de las quejas.	Media	Baja
3.3	Listar respuesta de las quejas	El sistema debe permitir listar las respuestas de las quejas.	Alta	Baja
3.4	Eliminar respuesta de las quejas	El sistema debe permitir eliminar las respuestas de las quejas.	Baja	Baja
RF4	Gestionar trabajador.			
4.1	Registrar trabajador	El sistema debe permitir registrar a los trabajadores.	Alta	Media
4.2	Modificar trabajador	El sistema debe permitir modificar a los trabajadores.	Media	Media
4.3	Listar trabajador	El sistema debe permitir listar a los trabajadores.	Alta	Baja

4.4	Eliminar trabajador	El sistema debe permitir eliminar a los trabajadores.	Baja	Baja
RF5	Gestionar indicadores.			
5.1	Registrar indicadores	El sistema debe permitir registrar los indicadores.	Alta	Media
5.2	Modificar indicadores	El sistema debe permitir modificar los indicadores.	Media	Media
5.3	Listar indicadores	El sistema debe permitir listar los indicadores.	Alta	Baja
5.4	Eliminar indicadores	El sistema debe permitir eliminar los indicadores.	Baja	Baja
RF6	Gestionar emulación de trabajador.			
6.1	Registrar emulación de trabajador	El sistema debe permitir registrar la emulación de los trabajadores.	Media	Media
6.2	Modificar emulación de trabajador	El sistema debe permitir modificar la emulación de los trabajadores.	Baja	Media
6.3	Listar emulación de trabajador	El sistema debe permitir listar la emulación de los trabajadores.	Alta	Baja
6.4	Eliminar emulación de trabajador	El sistema debe permitir eliminar la emulación de los trabajadores.	Baja	Baja
RF7.	Gestionar cargo.			
7.1	Registrar cargo	El sistema debe permitir registrar los cargos de los trabajadores.	Alta	Baja
7.2	Modificar cargo	El sistema debe permitir modificar los cargos de los trabajadores.	Media	Media
7.3	Listar cargo	El sistema debe permitir listar los cargos de los trabajadores.	Alta	Baja
7.4	Eliminar cargo	El sistema debe permitir eliminar los cargos de los trabajadores.	Baja	Baja

RF8.	Gestionar sección sindical.			
8.1	Registrar sección sindical	El sistema debe permitir registrar la sección sindical.	Alta	Media
8.2	Modificar sección sindical	El sistema debe permitir modificar la sección sindical.	Media	Media
8.3	Listar sección sindical	El sistema debe permitir listar la sección sindical.	Alta	Baja
8.4	Eliminar sección sindical	El sistema debe permitir eliminar la sección sindical.	Baja	Baja
RF9	Gestionar emulación por sección sindical.			
9.1	Registrar emulación por sección sindical	El sistema debe permitir registrar la emulación por sección sindical.	Media	Alta
9.2	Modificar emulación por sección sindical	El sistema debe permitir modificar la emulación por sección sindical.	Baja	Media
9.3	Listar emulación por sección sindical	El sistema debe permitir listar la emulación por sección sindical.	Alta	Baja
9.4	Eliminar emulación por sección sindical	El sistema debe permitir eliminar la emulación por sección sindical.	Baja	Baja
RF10	Gestionar actividades.			
10.1	Registrar actividades	El sistema debe permitir registrar las actividades.	Alta	Baja
10.2	Modificar actividades	El sistema debe permitir modificar las actividades.	Baja	Media
10.3	Listar actividades	El sistema debe permitir listar las actividades.	Alta	Baja
10.4	Eliminar actividades	El sistema debe permitir eliminar las actividades.	Baja	Baja

Tabla 2: Requisitos Funcionales.

II.3.2 Requisitos no funcionales.

Los requisitos no funcionales son características y restricciones que describen como debe ser el sistema informático en términos de calidad, seguridad, usabilidad y otros aspectos.

(28)

1: Requisitos de usabilidad:

1.1: El sistema debe ser intuitivo y fácil de usar para los usuarios finales, facilitando la realización de las tareas.

1.2: La interfaz debe ser clara.

1.3: Debe brindar comentarios adecuados para que los usuarios comprendan el progreso de las acciones realizadas.

2: Requisitos de seguridad

2.1: El sistema debe cumplir con los estándares de seguridad establecidos garantizando la confidencialidad, integridad y disponibilidad de los datos almacenados.

2.2: Protección contra accesos no autorizados.

3: Disponibilidad:

3.1: El sistema estará disponible para los usuarios que pertenecen a la red de la universidad.

4: Confidencialidad:

4.1: El usuario solo podrá visualizar la información a la que tenga permiso.

II.4 Historias de Usuarios

Las historias de usuarios (HU) son descripciones que resumen la necesidad al desarrollar un producto o un servicio, es una técnica utilizada para especificar los requisitos del software.

Número 1	Nombre de la Historia de usuario: Autenticar usuario.	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 1	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: ingresar un nombre de usuario y una contraseña para validar las credenciales del usuario y garantizar que tenga los permisos adecuados para acceder.		
Observaciones:		
Prototipo:  <i>Figura 2: Historia de usuario autenticar usuario</i>		

Tabla 3: Historia de usuario autenticar usuario

II.5 Arquitectura del Software.

El patrón arquitectónico Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implantación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. (29)

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los Controladores o de las Vistas. Es el propio sistema que tiene recomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas y notificar a las Vistas cuando cambia el Modelo. (29)

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia del propio Modelo. (29)

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por las alteraciones de la Vista. (29)

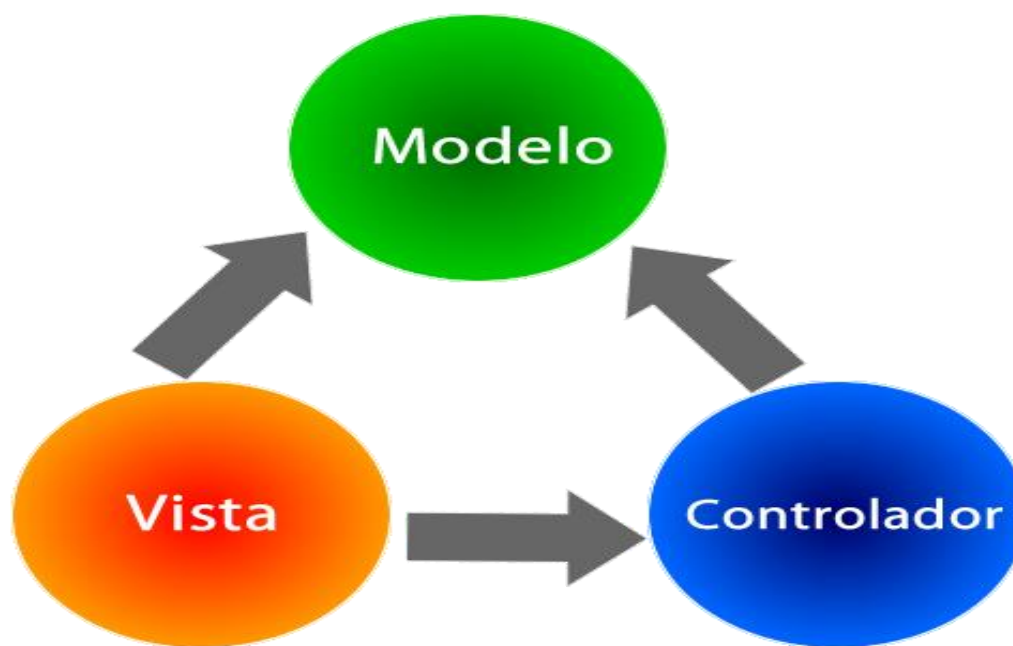


Figura 2.1: Arquitectura Modelo-Vista-Controlador.

II.6 Patrones de Diseño.

Los patrones de diseño son una solución comprobada y reusable para problemas recurrentes en el diseño. Estos patrones dan soluciones generales que se aplican a situaciones específicas para abordar problemas comunes en el diseño de software. Un patrón de diseño nomina, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que hace más útiles para crear un diseño orientado a objetos reusable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones y la distribución de las responsabilidades. (30)

II.6.1 Patrones GRASP.

GRASP son patrones generales de software para asignación de responsabilidades. Se considera que más que patrones son una serie de “ buenas prácticas” de aplicación recomendable en el diseño de software. (30)

Experto: es el principio básico de asignación de responsabilidades. Nos indica que la responsabilidad de la creación de un objeto o la implantación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada. Se evidencia al momento de gestionar un usuario a la hora de asignar los roles y los permisos.

Creador: Ayuda a identificar quién debe ser el responsable de la creación de nuevos objetos o clases. Se evidencia al momento de creación y asignación de actividades.

Controlador: es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que lo implementa, de tal forma que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Se evidencia a la hora de asignar tareas a las secciones sindicales, controlando así el funcionamiento de las mismas.

Alta Cohesión: dice la información que almacena una clase debe ser coherente y debe estar en la medida de lo posible relacionada con la clase. Se evidencia en el Buró Sindical Universitario donde asigna actividades a las secciones sindicales y ellas a su vez controlan que sus trabajadores cumplan con estas actividades.

Bajo Acoplamiento: es la idea de tener las clases lo menos ligadas entre sí. De tal forma que, en caso de producirse una modificación en algunas de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. Se evidencia en la gestión de la información de los trabajadores ya que posee una estructura y debe encargarse de gestionarla

II.6.2 Patrones GOF.

Los patrones Gang of Four (GOF) constituyen una herramienta fundamental para cualquier programador. Se utilizan para solucionar problemas de creación de instancias, ayudando a encapsular y abstraer dicha creación. Los patrones GOF utilizados son:

Observador: este patrón define una dependencia de uno-a-muchos, de forma que cuando un objeto cambie de estado se notifica y se actualiza automáticamente a todos los objetos que dependan de él. (30)

II.7 Diagramas de Clases de Diseño.

Un diagrama de clases es una representación visual de los objetos de clases en un sistema de modelo, clasificados por tipos de clases. Cada tipo de clases es representado como un rectángulo con tres comportamientos para el nombre de la clase, los atributos y las operaciones. A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. (31)

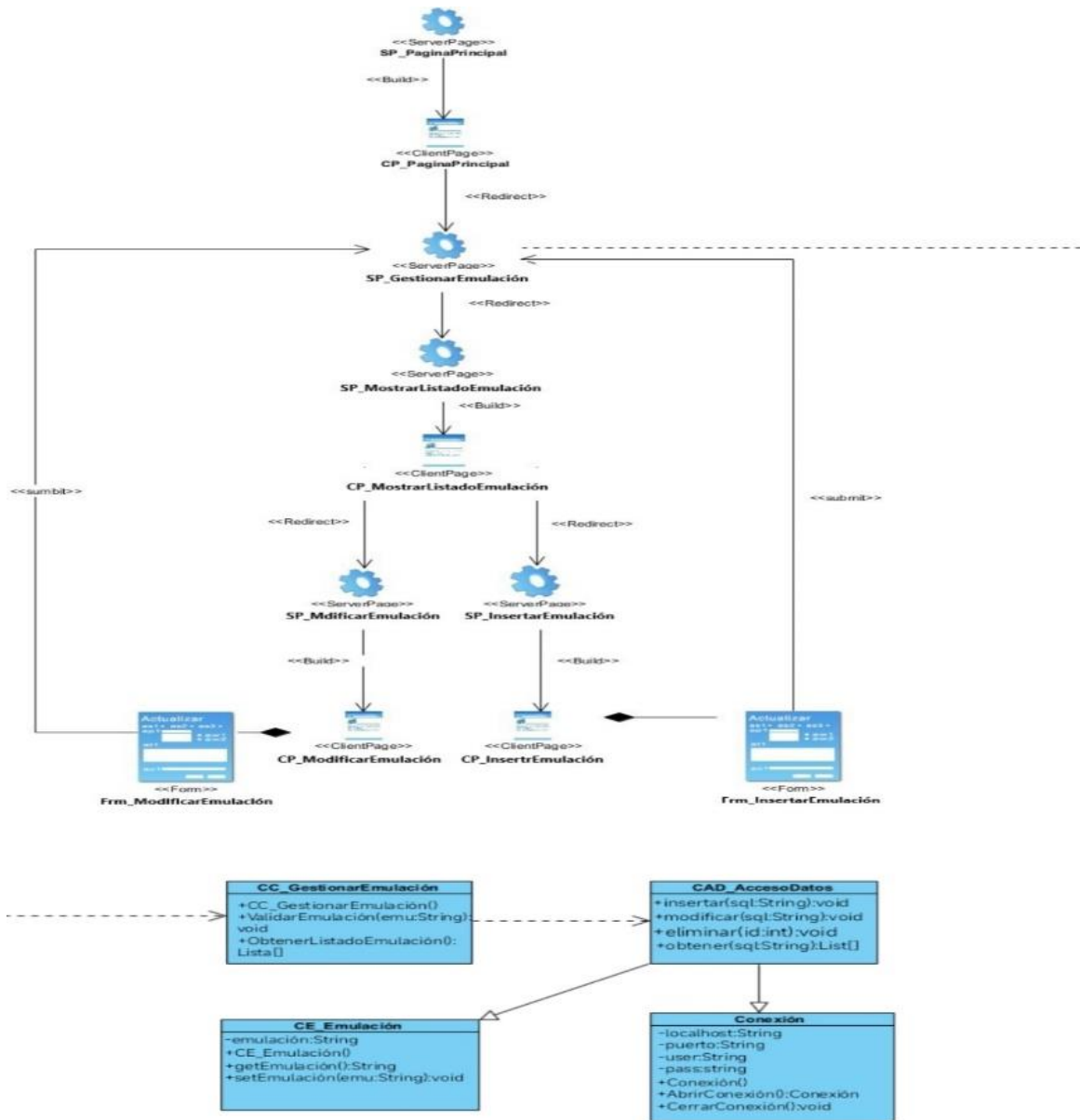


Figura 6: Diagrama clase de diseño

II.8 Modelo de datos.

Un modelo de datos es básicamente una " descripción " de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar dato de esos contenedores. Los modelos de datos no son cosas físicas, son abstracciones que permiten la implementación de un sistema eficiente de base de datos. (32)

El modelo entidad-relación (MER) es un modelo de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados entidades y relaciones, incorporando una representación visual conocida como diagrama entidad-relación. (32)

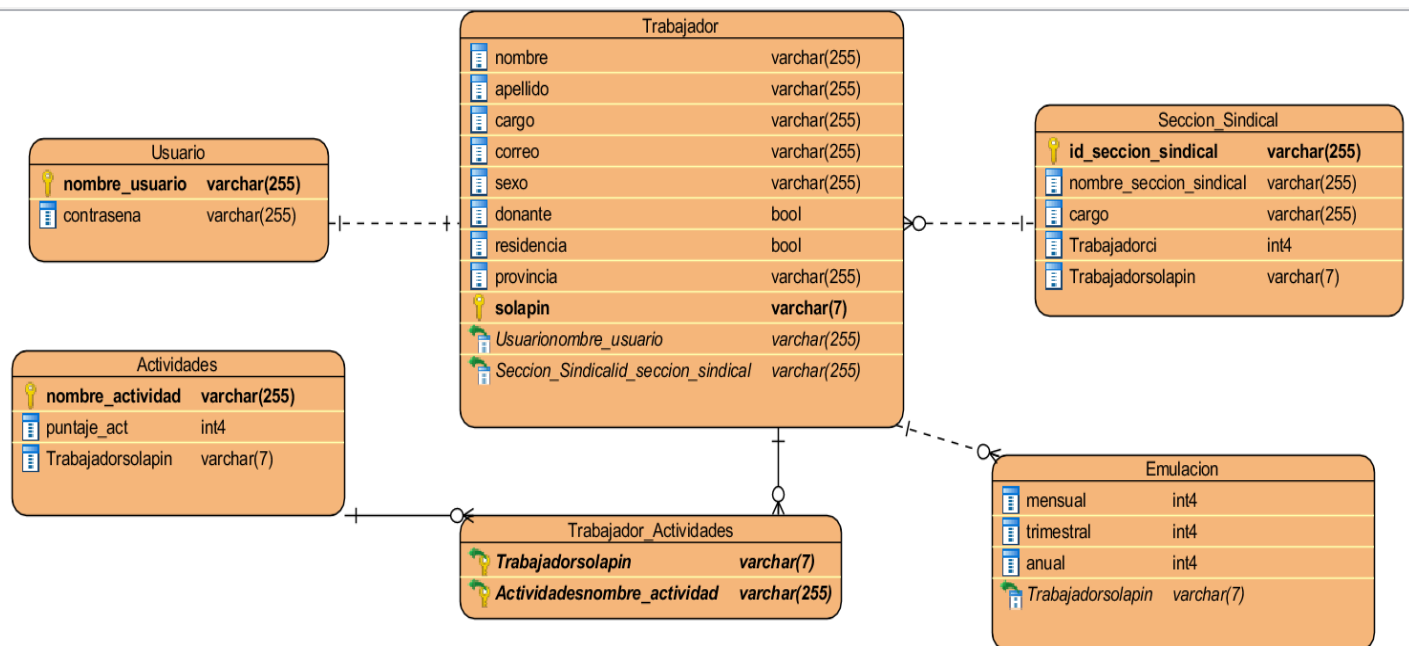


Figura 7: Modelo de datos

Conclusiones del capítulo

En este capítulo apoyándose en la metodología AUP-UCI en su escenario 4, se han definido las bases para el desarrollo del sistema. Mediante el levantamiento de requisitos se identificaron 10 requisitos funcionales y 4 no funcionales. A partir del análisis y comprensión de los requisitos funcionales se elaboraron 10 historias de usuario, lo que garantiza que la propuesta de solución cumpla con las necesidades del cliente. También se determinaron los patrones de diseño GRAF y GOF, que son herramientas valiosas para mejorar la calidad del software.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS.

En el presente capítulo se evalúa el grado de calidad y fiabilidad de los requisitos obtenidos en el desarrollo de la investigación. Este capítulo ofrece una visión de cómo se llevó a cabo la implementación de la aplicación web y se realizarán las pruebas para evaluar la funcionalidad y de validación. También se analizarán los posibles problemas encontrados durante las pruebas.

III.1 Estándares de codificación.

Los estándares de codificación son patrones o modelos que comprenden todos los aspectos de la generación del código. Son un conjunto de reglas establecidas para el formato y la estructura del código fuente, para asegurar la legibilidad, la consistencia y la mantenibilidad del mismo. Los estándares de codificación facilitan la comprensión del código y se reduce los errores. El objetivo de los estándares de codificación de software, es incluir buenas prácticas de programación que conduzcan a un código seguro. (33)

Nomenclatura de las variables:

- Se deben evitar los nombres de difícil comprensión o que no tenga ningún sentido.
- Los nombres de las variables se escriben con la primera letra minúscula.
- Si el nombre es compuesto se separan con un guión bajo “_”.

Nomenclatura de las clases:

- Si el nombre es compuesto se separan con un guión bajo “_”.

III.3 Pruebas

Las pruebas de software permiten controlar la calidad y funcionalidad de cualquier producto que se desarrolle; son la mejor garantía de que este no presente fallos y se comporta adecuadamente. Deben aplicarse en todas las fases del desarrollo, de manera que puedan

descubrirse fallos aislados, esto aporta varios beneficios que incluyen la solución de errores en una fase temprana. (34)

III.3.1 Pruebas unitarias

Las pruebas unitarias comprueban que cada una de las piezas o unidades más pequeñas del software en el que está trabajando funcione correctamente. Estas pruebas se aplican de manera individual y son las primeras que deben realizarse durante todo el proceso de desarrollo.

Es necesario aislar esas unidades, que pueden ser fragmentos de código, para verificar su comportamiento. Al trabajar con unidades tan pequeñas es posible testear el proyecto por partes, sin necesidad de que esté terminado. Son muy rápidas y facilitan la comprobación de la legibilidad del código.

Entre las ventajas que está el ahorro de tiempo y de dinero, ya que permiten detectar errores al principio y evita seguir avanzando arrastrando este fallo que va a condicionar el resto del proyecto.

Jasmine es uno de los frameworks de desarrollo dirigido por comportamiento para código Java Script, es uno de los más utilizados para realizar pruebas unitarias, se basa en Behavior Driven Development, permite realizar test tanto en el cliente como en el servidor y es un framework de desarrollo que no depende de ninguna otra librería JavaScript. Jasmine realiza pruebas al código, para así asegurar su estabilidad.

A continuación, se muestra las pruebas realizadas al sistema con la herramienta Jasmine:

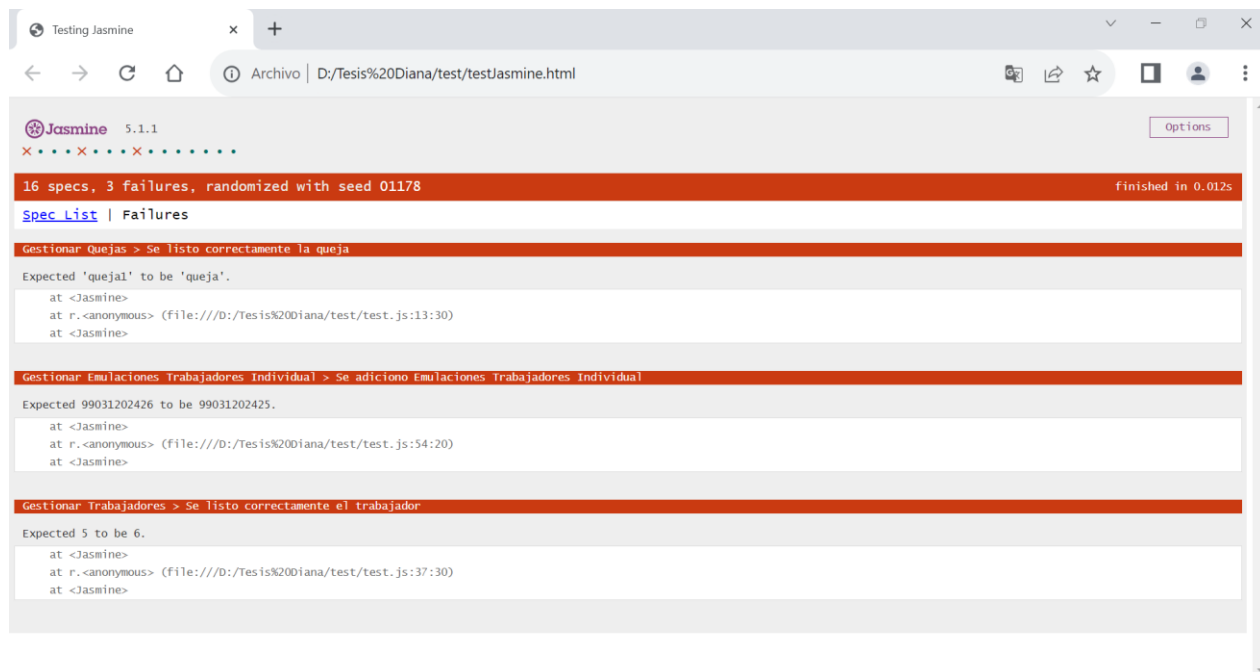


Figura 8: Prueba unitaria Jasmine con errores

En la figura 8, se muestra el resultado de la primera prueba da un total de 3 errores.

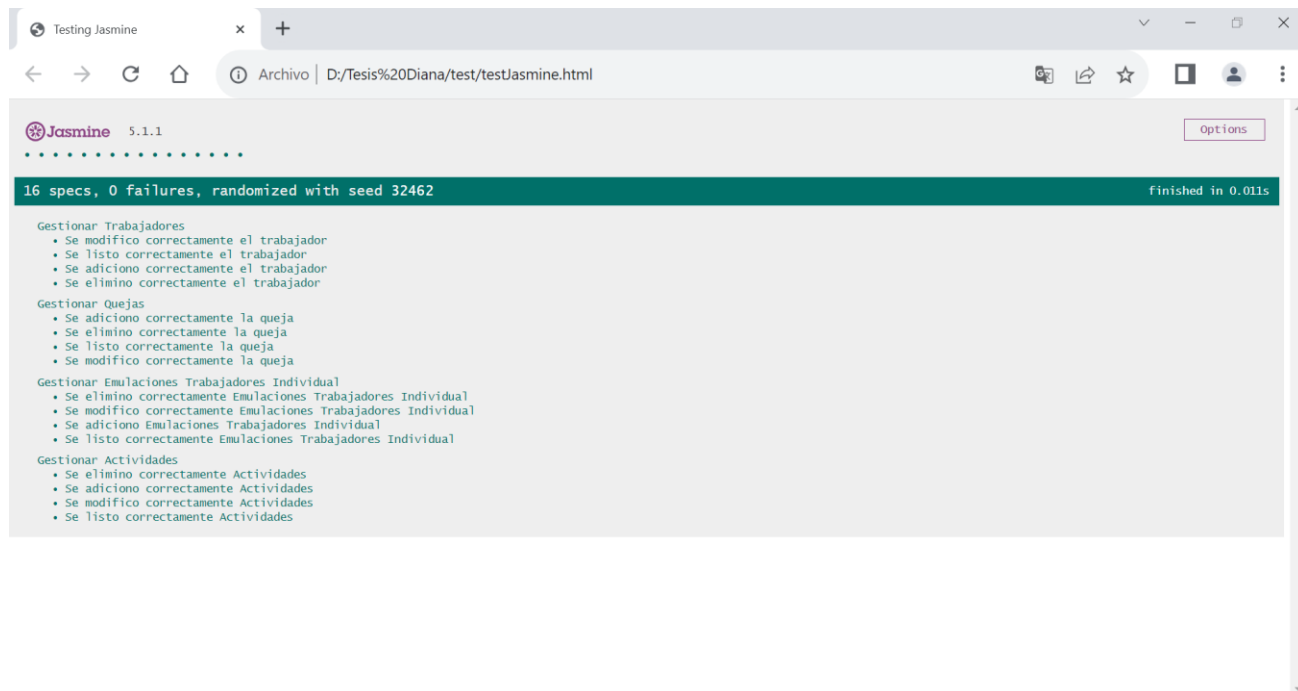


Figura 9: Prueba unitaria Jasmine con resultados correctos

En la figura 9 se muestran los resultados correctos proporcionados por la aplicación con la herramienta de prueba.

A continuación, se muestra una gráfica con las pruebas unitarias realizadas al sistema.

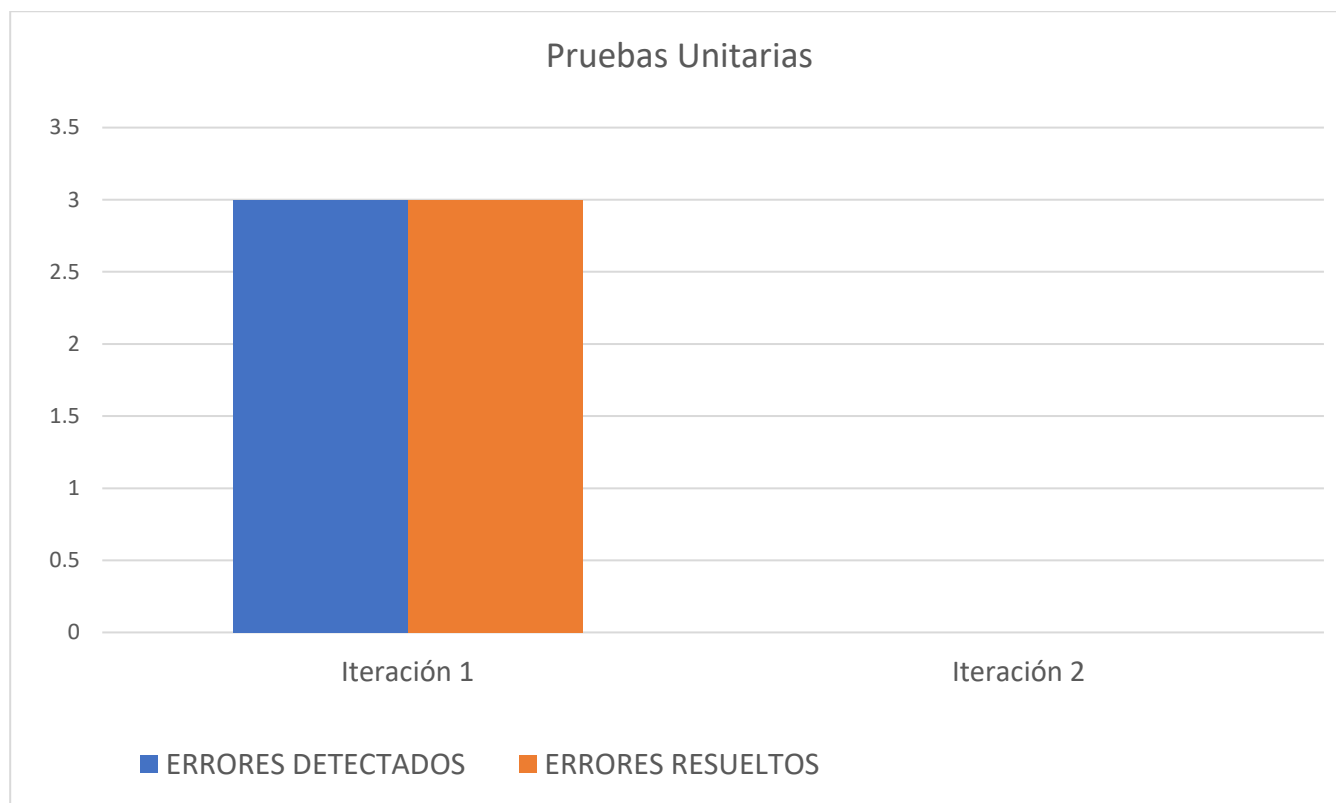


Figura 10: Gráfica de pruebas unitarias

En la figura 10 se evidencian la cantidad de pruebas realizadas con la herramienta Jasmine donde se le hicieron 16 pruebas, de los cuales se obtuvieron 3 errores. Posteriormente, se muestra la cantidad de errores resueltos.

III.3.2 Pruebas de validación

La validación del software es el proceso de comprobar que el sistema está acorde a las especificaciones y que cumple con las necesidades reales de los usuarios del sistema. Las

pruebas consisten en un proceso crítico para asegurar que el software sea entregado al cliente libre de defectos. (35)

La partición equivalente es una técnica de prueba de Caja Negra que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de pruebas. Tiene un conjunto de clases equivalentes donde el conjunto de datos define entradas válidas y no válidas al sistema, donde las entradas válidas generan un valor esperado y las entradas no válidas generan un valor inesperado (excepciones). (36)

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo Central
EC 1.1 Insertar correctamente un usuario y una contraseña.	El usuario inserta correctamente el nombre de usuario y la contraseña.	Admin	Admin	Se accede de manera correcta al sistema.	_Insertar un usuario correctamente. _Insertar una contraseña correctamente. _Seleccionar la opción autenticar
EC 1.2 Insertar de manera incorrecta un usuario o una contraseña.	El usuario inserta de manera incorrecta un usuario o una contraseña.	admin	Admin987	Muestra un mensaje “Las credenciales son incorrectas”.	_Insertar un usuario correctamente. _Insertar una contraseña de manera incorrecta. _Seleccionar la opción autenticar.

		Admin91	Admin		_Insertar un usuario de manera incorrecta. _Insertar una contraseña de manera correcta. _Seleccionar la opción autenticar.
EC 1.3 Campo vacío usuario o contraseña	El usuario no inserta el nombre del usuario o no inserta la contraseña del usuario.	admin	-	Muestra un mensaje "Campo Vacío"	_Insertar un usuario correctamente. _Dejar el campo de la contraseña vacío. _Seleccionar la opción autenticar.
		-	Admin		_Dejar el campo de usuario vacío. _Insertar de manera correcta la contraseña. _Seleccionar la opción autenticar.

Tabla 13: Caso de prueba de partición de equivalencia del RF1 Autenticar Usuario

Al aplicar las pruebas de partición de equivalencia en las tablas 13 y 14 se identificaron grupos de datos de entrada que comparten características en términos de su comportamiento esperado. Al probar cada grupo de datos de entrada respectivo, se puede garantizar que se cubran todas las posibles situaciones y se compruebe la funcionalidad de los requisitos, se han cumplido los criterios de partición de equivalencia establecidos, se puede concluir que el sistema cumple con los requisitos y funciona correctamente.

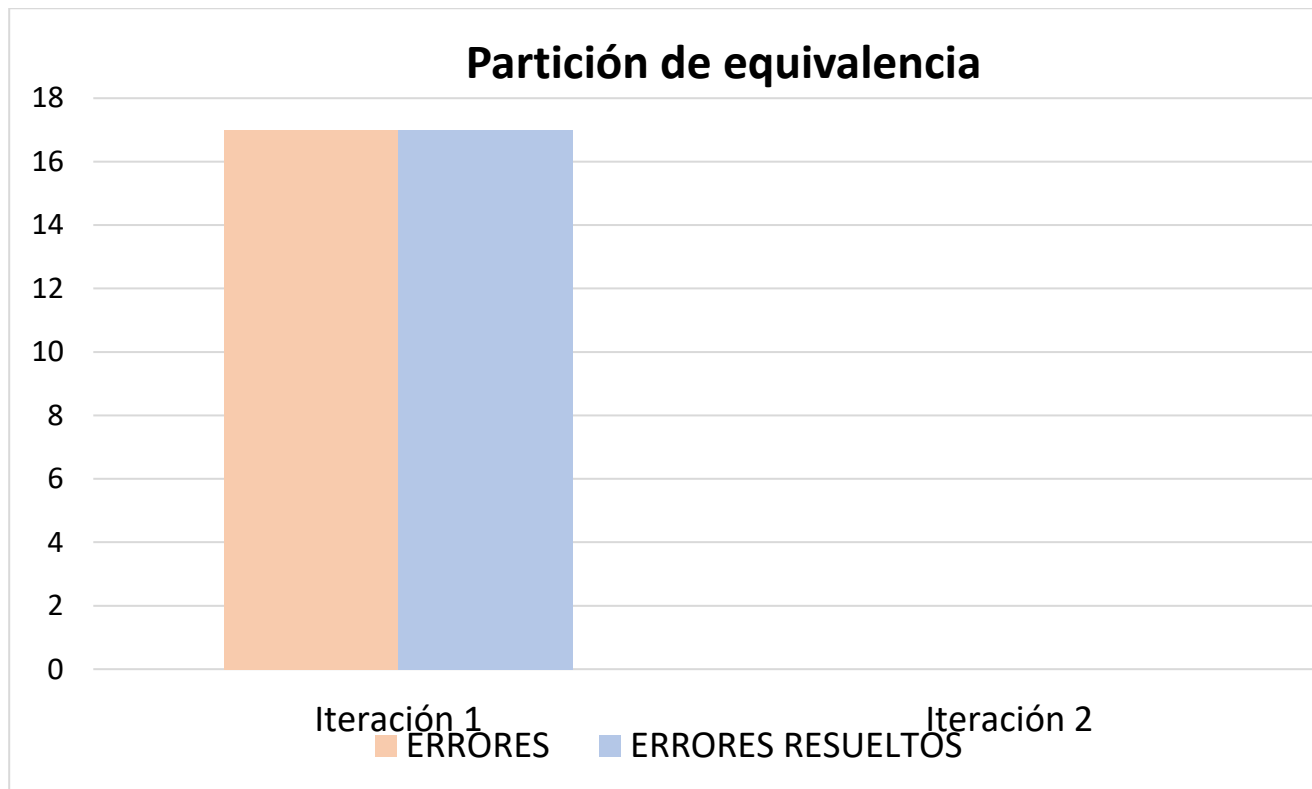


Figura 11: Gráfica de pruebas funcionales.

En la figura 11 se hace evidencia de las pruebas de caja negra mediante la técnica de partición de equivalencia, donde se obtuvieron en la primera iteración 17 no conformidades y en la segunda se obtuvo un sistema libre de no conformidades.

III.3.2 Pruebas de aceptación

Las pruebas de aceptación o pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados en las historias de usuarios. Son básicamente pruebas funcionales sobre el sistema completo, buscan una cobertura de la especificación de requerimientos y del manual del usuario. El cliente registra cada uno de los problemas que encuentra durante las pruebas y los informa al desarrollador. (37)

Requisito de aceptación: Los usuarios deben poder autenticarse en el sistema.

Caso de prueba: Autenticación de usuario.

- 1- Abrir el navegador web e ingresar la URL de inicio de sesión del sistema.
- 2- Insertar el nombre del usuario y la contraseña en los campos correspondientes.
- 3- Hacer clic en el botón autenticar.
- 4- Verificar que el sistema envíe correctamente al usuario al panel de control.
- 5- Cerrar sesión desde el panel de control y verificar que el sistema redirija al usuario a la página de inicio de sesión.

Resultado obtenido:

- El usuario puede autenticarse correctamente utilizando sus credenciales de inicio de sesión.
- El usuario es redirigido al panel de control personalizado.
- Las funcionalidades solo son accesibles para usuarios autenticados.
- Al cerrar sesión, el usuario es redirigido a la página de inicio de sesión.

Las pruebas de aceptación se realizan de manera satisfactoria, se concluye que el sistema cumple con los requisitos establecidos por los usuarios. También se revelaron áreas de mejoras que se abordaron y corrigieron para mejorar la calidad y la satisfacción del producto.

Conclusiones del capítulo

Este capítulo resume los resultados de las pruebas realizadas de caja negra y caja blanca para evaluar la funcionalidad del sistema. Destaca los aspectos positivos y negativos identificados durante las pruebas, así como cualquier error o problema encontrado, lo que ha permitido mejorar la calidad y el rendimiento del sistema. Evalúa si el sistema cumple con los requisitos establecidos en las fases de diseño y análisis. Las pruebas han permitido realizar ajustes en el sistema para que los usuarios tengan una buena experiencia durante su uso y ha permitido validar la propuesta de solución. La implementación permitió desarrollar un sistema funcional.

CONCLUSIONES FINALES

Una vez concluido con la investigación y analizado los resultados obtenidos, se obtienen las siguientes conclusiones:

- El desarrollo de una aplicación web para la gestión de los indicadores asociados a la emulación sindical permitió tener un control de la emulación en la UCI.
- La selección adecuada de herramientas y tecnologías permitió la creación de una aplicación web acorde con los requisitos del cliente.
- La utilización de patrones arquitectónicos aceleró el proceso de desarrollo la aplicación web para la gestión de los indicadores asociados a la emulación sindical.
- Mediante la realización de las pruebas de software se obtuvo una aplicación web funcional y sin errores.

RECOMENDACIONES

Se recomienda:

- Informatizar los procesos sindicales asociados a las esferas de Cotización y Organización.

REFERENCIAS BIBLIOGRÁFICAS

- 1.1. EDUTEC. Revista Electrónica de Tecnología Educativa . ISSN 1135-9250 Núm. 55
1. Universidad de las Ciencias Informáticas. [Online] [Cited: Junio 13, 2017.] <http://www.uci.cu/>.
2. Mesa, Salvador Valdés. mayo-junio 2013. *Contribución al Examen Periódico Universal, (EPU), de Cuba sobre el cumplimiento de los Derechos Económicos, Sociales y Culturales en Cuba, del 2009 al 2012.* mayo-junio 2013.
3. García, A. (2019). El papel de los sindicatos en la actualidad. Revista de Trabajo.
4. UCI, Buró Sindical Universidad de las Ciencias Informáticas. *Estrategia de trabajo de la esfera Emulación Socialista.*
5. Aja Quiroga L. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. Acimed 2002;9(1).
6. Pérez, LA (2018). Cuba: Entre la Reforma y la Revolución. Prensa de la Universidad de Oxford.
7. García, M. (2019). Emulación socialista y gestión empresarial en Cuba. Revista
8. Ley NO.166 del Código del Trabajo de la República de Cuba (2014). La Habana, Cuba.
9. Unión General de Trabajadores. [En línea] www.ugt.es.
10. SNTECD. [En línea] <http://www.sindicatoeducacion.cu/>.
11. Meriño Menadier, Dennis. "Sistema de Gestión del Sindicato Nacional de Administración Pública". s.l. : Universidad de las Ciencias Informáticas, 2007.
12. Medina Soca, Yoalys Miriam y Muñoz Delgado, Yamisleidys. "Análisis y Diseño del Proceso de Funcionamiento para la Sección Sindical Vicerrectoría Primera en la UCI. s.l. : Universidad de las Ciencias Informáticas, 2009.
13. Moya Alcalá, Daymaris. Portal Web para la CTC de la Facultad 6. s.l. : Universidad de las Ciencias Informáticas, 2011
14. Roberth G. Figueroa, Camilo J. Solís y Armando A. Cabrera. Metodologías Tradicionales vs Metodologías Ágiles . [Online]

15. Navarro Cadavid, Andrés, Fernández Martínez, Juan Daniel y Morales Vélez, Jonathan. julio-diciembre, 2013. *Revisión de metodologías ágiles para el desarrollo de software PROSPECTIVA*. Colombia : s.n., julio-diciembre, 2013.
16. Yero Tarancón, Yixander . 2015. Xabal Excriba. Gestor de Documentos Administrativos. [En línea]. [Citado el: 18 de 11 de 2018.]
17. Sánchez, Tamara Rodríguez. 2015. Metodología de desarrollo para la Actividad productiva de la UCI. L a Habana, Cuba : s.n., 2015.
18. RONDÓN, Y.Q., DOMÍNGUEZ, L.C. y BERENGUER, A.D., 2011. Diseño de la base de datos para sistemas de digitalización y gestión de medias., vol. 8. ISSN 1667.
- 18.1. Guión Visual Paradigm four UML.Ingeniería del Software. Curso2013-2014. 2013.
19. BARCIA, D. 2003. Qué es CSS? 2003. Dirección: <http://www.maestrosdelweb.com/introcss/i>.
20. HICKSON, I. y HYATT, D. 2009. HTML5. 2009. Dirección: <http://dev.w3.org/html5/spec/Overview.html#html-vs-xhtml>.
21. J, Eguíluz Pérez. Introducción a JavaScript . [En línea] 2009. www.librosweb.es.
22. Sitio Oficial de Postgres. [Online] [Cited: Enero 28, 2014.] <http://www.postgresql.org.es>.
23. PgAdmin.2007.Dirección: <http://www.purosoftware.com/programacion-bases-de-datos/11-pg-admin-3.html>.
24. MARDAN, Azat, 2018. Using Express.js to create Node.js web apps. En: Practical Node.js. Sprin-ger. pp. 51-87.
25. MARDAN, Azat, 2014. Express.js Guide: The Comprehensive Book on Express.js. Azat Mardan.
26. Flores, F. (2022). Qué es Visual Studio Code y qué ventajas ofrece. OpenWebinars.
27. PRESSMAN, Roger S., 2005. *Software engineering: a practitioner's approach*. Palgrave macmillan. ISBN 0-07-301933-X.
28. RS, Pressman. Ingeniería de software: Un enfoque práctico. Nueva York, EUA : s.n., 2007.
29. González, Y. F. (2012). Patrón Modelo-Vista-Controlador. *Revista Digital de las Tecnologías de la Informacion y las Comunicaciones* , 47-57.
30. Vlissides, E. G. (s.f.). *Patrones de Diseño Elementos de software orientado a objetos reutilizable*.

31. C, Larman. UML y Patrones: Introducción al análisis y diseño orientado a objetos. Mexico : s.n., 1999.
32. M, Blázquez Ochando. Modelo entidad-relación. Fundamentos y Diseño de Bases de Datos. 2014.
33. **DJ, Salas Álvarez.** Estándares de codificación Java . [En línea] 2016.
34. La importancia de las pruebas de software . (2022). INGENIERÍA Y TECNOLOGÍA.
35. Bernal, S. M. (2017). Verificación, validación y pruebas de software. *Killkana Técnica* , 25-32.
36. JOSÉ LUIS ARISTEGUI. Test cases in software test. 2010.
37. LATIN AMERICAN CONGRESS ON REQUIREMENTS ENGINEERING & SOFTWARE TESTING LASCREST MEDELLÍN 2012. July 13-14, Medellín, Colombia.

ANEXOS

Conceptos	Descripción
Trabajador	Es una persona que realiza una actividad laboral en la universidad.
Secretario	Persona que brinda apoyo administrativo y de gestión en la organización.
Sección Sindical	Actúa como órgano representativo de los trabajadores.
Buró Intermedio	Actúa entre dos o más partes facilitando la comunicación, coordinación y colaboración entre ellas.
Buró Sindical Universitario	Es un órgano de representación y organización sindical conformados por los trabajadores de la universidad.
Actividades	Acciones que se llevan a cabo con un propósito específico.
Emulación individual	Actúa como fuente de motivación y crecimiento en el ámbito profesional.
Emulación sindical	Promueve el intercambio de conocimientos entre los sindicatos, contribuyendo a fortalecer el movimiento sindical.
Indicadores	Son medidas o variables para medir el desempeño y los resultados alcanzados.

Tabla 1: Descripción del modelo de dominio.

N°	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar Usuario	El sistema debe permitir autenticar al usuario que tenga los permisos adecuados para acceder.	Alta	Alta
RF2	Gestionar quejas UCI			
2.1	Registrar quejas UCI	El sistema debe permitir registrar una queja.	Alta	Media
2.2	Modificar quejas UCI	El sistema debe permitir modificar una queja.	Media	Baja
2.3	Listar quejas UCI	El sistema debe permitir listar de las quejas.	Alta	Baja
2.4	Eliminar quejas UCI	El sistema debe permitir eliminar las quejas.	Baja	Baja
RF3	Gestionar respuesta de las quejas.			
3.1	Registrar respuesta de las quejas	El sistema debe permitir registrar las respuestas de las quejas.	Alta	Media
3.2	Modificar respuesta de las quejas	El sistema debe permitir modificar las repuestas de las quejas.	Media	Baja
3.3	Listar respuesta de las quejas	El sistema debe permitir listar las respuestas de las quejas.	Alta	Baja
3.4	Eliminar respuesta de las quejas	El sistema debe permitir eliminar las respuestas de las quejas.	Baja	Baja
RF4	Gestionar trabajador.			
4.1	Registrar trabajador	El sistema debe permitir registrar a los trabajadores.	Alta	Media
4.2	Modificar trabajador	El sistema debe permitir modificar a los trabajadores.	Media	Media
4.3	Listar trabajador	El sistema debe permitir listar a los	Alta	Baja

		trabajadores.		
4.4	Eliminar trabajador	El sistema debe permitir eliminar a los trabajadores.	Baja	Baja
RF5	Gestionar indicadores.			
5.1	Registrar indicadores	El sistema debe permitir registrar los indicadores.	Alta	Media
5.2	Modificar indicadores	El sistema debe permitir modificar los indicadores.	Media	Media
5.3	Listar indicadores	El sistema debe permitir listar los indicadores.	Alta	Baja
5.4	Eliminar indicadores	El sistema debe permitir eliminar los indicadores.	Baja	Baja
RF6	Gestionar emulación de trabajador.			
6.1	Registrar emulación de trabajador	El sistema debe permitir registrar la emulación de los trabajadores.	Media	Media
6.2	Modificar emulación de trabajador	El sistema debe permitir modificar la emulación de los trabajadores.	Baja	Media
6.3	Listar emulación de trabajador	El sistema debe permitir listar la emulación de los trabajadores.	Alta	Baja
6.4	Eliminar emulación de trabajador	El sistema debe permitir eliminar la emulación de los trabajadores.	Baja	Baja
RF7.	Gestionar cargo.			
7.1	Registrar cargo	El sistema debe permitir registrar los cargos de los trabajadores.	Alta	Baja
7.2	Modificar cargo	El sistema debe permitir modificar los cargos de los trabajadores.	Media	Media
7.3	Listar cargo	El sistema debe permitir listar los cargos de los trabajadores.	Alta	Baja
7.4	Eliminar cargo	El sistema debe permitir eliminar los cargos de los trabajadores.	Baja	Baja

RF8.	Gestionar sección sindical.			
8.1	Registrar sección sindical	El sistema debe permitir registrar la sección sindical.	Alta	Media
8.2	Modificar sección sindical	El sistema debe permitir modificar la sección sindical.	Media	Media
8.3	Listar sección sindical	El sistema debe permitir listar la sección sindical.	Alta	Baja
8.4	Eliminar sección sindical	El sistema debe permitir eliminar la sección sindical.	Baja	Baja
RF9	Gestionar emulación por sección sindical.			
9.1	Registrar emulación por sección sindical	El sistema debe permitir registrar la emulación por sección sindical.	Media	Alta
9.2	Modificar emulación por sección sindical	El sistema debe permitir modificar la emulación por sección sindical.	Baja	Media
9.3	Listar emulación por sección sindical	El sistema debe permitir listar la emulación por sección sindical.	Alta	Baja
9.4	Eliminar emulación por sección sindical	El sistema debe permitir eliminar la emulación por sección sindical.	Baja	Baja
RF10	Gestionar actividades.			
10.1	Registrar actividades	El sistema debe permitir registrar las actividades.	Alta	Baja
10.2	Modificar actividades	El sistema debe permitir modificar las actividades.	Baja	Media
10.3	Listar actividades	El sistema debe permitir listar las actividades.	Alta	Baja
10.4	Eliminar actividades	El sistema debe permitir eliminar las actividades.	Baja	Baja

Tabla 2: Requisitos Funcionales.

Número 1	Nombre de la Historia de usuario: Autenticar usuario.	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 1	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: ingresar un nombre de usuario y una contraseña para validar las credenciales del usuario y garantizar que tenga los permisos adecuados para acceder.		
Observaciones:		
Prototipo:  <i>Figura 2: Historia de usuario autenticar usuario</i>		

Tabla 3: Historia de usuario autenticar usuario

Número 2	Nombre de la Historia de usuario: Gestionar quejas UCI	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 1	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar las quejas UCI _ Modificar las quejas UCI _ Listar las quejas UCI _ Eliminar las quejas UCI		

Observaciones: las quejas planteadas deben ser a nivel de universidad

Presentar una Queja ×

Nombre * **Apellidos ***

Correo *

Teléfono opcional **Tipo de Queja ***

Descripción de la Queja *

Act
Ve a

Prototipo:

Figura 3: Historia de usuario gestionar quejas

Tabla 4: Historia de usuario gestionar quejas

Número 3	Nombre de la Historia de Usuario: Gestionar respuesta a nivel UCI	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 1	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar las respuestas _ Modificar las respuestas _ Listar las respuestas _ Eliminar las respuestas		
Observaciones:		
Prototipo Respuesta a Quejas Figura 4: Historia de usuario gestionar respuesta a nivel UCI		

Tabla 5: Historia de usuario gestionar respuesta a nivel UCI

Número 4	Nombre de la Historia de Usuario: Gestionar trabajador	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 1	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar los datos de los trabajadores _ Modificar los datos de los trabajadores _ Listar los datos de los trabajadores _ Eliminar los datos de los trabajadores		
Observaciones:		
Trabajadores Prototipo <i>Figura 5: Historia de usuario gestionar trabajador</i>		

Tabla 6: Historia de usuario gestionar trabajador

Número 5	Nombre de la Historia de Usuario: Gestionar indicadores	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 2	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar indicadores _ Modificar indicadores _ Listar indicadores _ Eliminar indicadores		
Observaciones:		
Prototipo:		

Tabla 7: Historia de usuario gestionar indicadores

Número 6	Nombre de la Historia de Usuario: Gestionar emulación por trabajador	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 2	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar emulación _ Modificar emulación _ Listar emulación _ Eliminar emulación		
Observaciones:		
Prototipo:		

Tabla 8: Historia de usuario gestionar emulación por trabajador

Número 7	Nombre de la Historia de Usuario: Gestionar cargo	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 2	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar cargo _ Modificar cargo _ Listar cargo _ Eliminar cargo		
Observaciones:		
Prototipo		

Tabla 9: Historia de usuario gestionar cargo

Número 8	Nombre de la Historia de Usuario: Gestionar sección sindical
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta

Iteración asignada: 2	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar sección sindical _ Modificar sección sindical _ Listar sección sindical _ Eliminar sección sindical		
Observaciones:		
Prototipo		

Tabla 10: Historia de usuario gestionar sección sindical

Número 9	Nombre de la Historia de Usuario: Gestionar emulación por sección sindical	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 3	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar emulación _ Modificar emulación _ Listar emulación _ Eliminar emulación		
Observaciones:		
Prototipo		

Tabla 11: Historia de usuario gestionar emulación por sección sindical

Número 10	Nombre de la Historia de Usuario: Gestionar actividades.	
Programador: Diana Loinaz Velázquez	Prioridad en negocio: Alta	
Iteración asignada: 3	Riesgo en Desarrollo: Alto	Tiempo estimado:
Descripción: el sistema debe ser capaz de: _ Registrar actividades _ Modificar actividades _ Listar actividades _ Eliminar actividades		

Observaciones:
Prototipo

Tabla 12: Historia de usuario gestionar actividades

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo Central
EC 1.1 Insertar correctamente un usuario y una contraseña.	El usuario inserta correctamente el nombre de usuario y la contraseña.	Admin	Admin	Se accede de manera correcta al sistema.	_Insertar un usuario correctamente. _Insertar una contraseña correctamente. _Seleccionar la opción autenticar
EC 1.2 Insertar de manera incorrecta un usuario o una contraseña.	El usuario inserta de manera incorrecta un usuario o una contraseña.	admin	Admin987	Muestra un mensaje “Las credenciales son incorrectas”.	_Insertar un usuario correctamente. _Insertar una contraseña de manera incorrecta. _Seleccionar la opción autenticar.

		Admin91	Admin		_Insertar un usuario de manera incorrecta. _Insertar una contraseña de manera correcta. _Seleccionar la opción autenticar.
EC 1.3 Campo vacío usuario o contraseña	El usuario no inserta el nombre del usuario o no inserta la contraseña del usuario.	admin	-	Muestra un mensaje "Campo Vacío"	_Insertar un usuario correctamente. _Dejar el campo de la contraseña vacío. _Seleccionar la opción autenticar.
		-	Admin		_Dejar el campo de usuario vacío. _Insertar de manera correcta la contraseña. _Seleccionar la opción autenticar.

Tabla 13: Caso de prueba de partición de equivalencia del RF1 Autenticar Usuario

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 10.1 Insertar correctamente los datos de un trabajador.	Se inserta de manera correcta los datos de un trabajador.	Se guarda de manera correcta los datos del trabajador.	_Se inserta de manera correcta todos los campos con la información del trabajador. _Se selecciona la opción aceptar.
EC 10.2 Campo vacío	Se deja vacío algún campo obligatorio	Muestra un mensaje “Campo Vacío”	_Se deja algún campo vacío. _Se selecciona la opción aceptar. _Muestra un mensaje “Campo Vacío”
EC 10.3 Insertar incorrectamente los datos de un trabajador.	Se intenta insertar datos numéricos en campos que solo deben contener caracteres alfabéticos.	Se muestra un mensaje de error indicando que se ha ingresado caracteres inválidos.	_Se inserta valores numéricos en el nombre de un trabajador. _Se inserta la información del trabajador en los demás campos. _Se selecciona la opción aceptar.

			_Muestra un mensaje de error.
--	--	--	-------------------------------

Tabla 14: Caso de prueba de partición de equivalencia del RF10 Registrar trabajador