



FACULTAD 1

Módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1

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

Autor: Pedro Luis González Almora

Tutor: Ing. Yaili Ledea Velázquez

La Habana, noviembre del 2022

Año 64 de la Revolución

DECLARACIÓN DE AUTORÍA

Yo Pedro Luis González Almora con número de carnet de identidad: 99051005488 autor del trabajo titulado "Módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1", concedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar me declaro como único autor de su contenido. Para que así conste firman la presente a los ____ días del mes de diciembre del año 2022.

Pedro Luis González Almora

Firma del Autor

Yaili Ledea Velázquez

Firma del Tutor

DATOS DE CONTACTO

Yaili Ledea Velázquez, Profesora Asistente, pertenece al Departamento de Informática en la Facultad 1. Graduada de Ingeniería en Ciencias Informáticas en el 2008. Se desempeña como Profesora Principal de Año Académico y Profesora de Sistemas de Bases de Datos. Tutora de varios trabajos de diploma en pregrado y miembro de tribunales al efecto. Ha cursado e impartido postgrados relacionados con la especialidad.

Correo electrónico: yledea@uci.cu

AGRADECIMIENTOS

A mi novia hermosa Rocío, gracias por permitirme formar parte de tu vida, gracias por tu amor, gracias por ser como eres, gracias por ser la mujer con los mejores sentimientos que he conocido, gracias por presionarme para terminar este trabajo, gracias por ayudarme con las correcciones, gracias por aguantarme, pero sobre todo gracias por enseñarme a creer en mí y motivarme hacer las cosas de la mejor manera.

Gracias por todo.

Quiero agradecer a mi hermana por acompañarme en este duro camino, por apoyarme desde principio hasta fin , por confiar en mí, por ser tan buena y admirable y sobre todo por darme tanto ánimo y tanto amor.

Quiero agradecer a mi papá por haberme dado su apoyo incondicional durante todos estos años, por ser esa razón y el más grande aliciente para el cumplimiento de mis objetivos que significan alegría y orgullo para mí y también para él.

Gracias a mi tío y tía por apoyarme en todo este proceso y en todo el camino recorrido para poder llegar hasta aquí. Me siento afortunado de tenerlos a ustedes en mi vida.

Gracias a mis dos amigos Emmanuel y Leonardo David por estar conmigo en todo momento y apoyarme en todo este largo camino. Gracias por formar parte de mi vida y mi familia.

Gracias a mis amigos de la infancia que de una forma u otra siempre han estado pendiente de todo este duro proceso y me han alentado para salir victorioso en él.

A mi tutora por ayudarme, apoyarme en todo el proceso de mi trabajo de diploma.

Gracias a ti y a tus conocimientos que hicieron posible que hoy yo estuviera aquí defendiendo.

A todas las personas que hicieron posible de una forma u otra que yo esté aquí como a mi padre Rodney. Las nuevas amistades como Yanko, Kalimbo, Estebita, Dalia, Luisa y Liliana que me aconsejaron y apoyaron en todo. A Alejandro y Alito por preocuparse todo el tiempo como iba en el desarrollo de la tesis.

Muchas Gracias.

DEDICATORIA

A mi Abuela, por estar siempre en los momentos importantes de mi vida, por ser el ejemplo para salir adelante y por los consejos que han sido de gran ayuda para mi vida y crecimiento.

Esta tesis es el resultado de lo que me has enseñado en la vida, ya que siempre has sido una persona honesta y entregada a tu trabajo, pero más que todo eso, una gran persona que siempre ha podido salir adelante y ser triunfadora. Es por ello que hoy te dedico este trabajo de tesis.

Gracias por confiar en mí y darme la oportunidad de culminar esta etapa de mi vida.

A mi Madre, por ser la amiga y compañera que me ha ayudado a crecer, gracias por estar siempre conmigo en todo momento. Gracias por la paciencia que has tenido para enseñarme, por el amor que me das, por tus cuidados en el tiempo que hemos vivido juntos, por los regaños que me merecía y que no entendía. Gracias Mamá por estar pendiente durante toda esta etapa. A ti también te dedico este trabajo de diploma.

RESUMEN

La necesidad de mantener el proceso de formación de profesionales competentes de forma coherente con los tiempos que ha tocado vivir, sigue siendo interés común de directivos e investigadores de las ciencias de la educación, en este ámbito de la educación las TIC se han transformado en un pilar de suma importancia. La Universidad de las Ciencias Informáticas promueve el desarrollo personal del estudiante de manera integral a través de todos sus procesos sustantivos, fomentando la activa participación en diferentes eventos académicos, científicos y extensionistas. En el área de trabajo educativo, el proceso de gestión de las actividades y sus evidencias es un proceso manual, por lo que surge la necesidad de desarrollar un módulo informático, capaz de informatizar el proceso y contribuir a la gestión de actividades y evidencias. Para dar solución a lo planteado anteriormente, la presente investigación se organizó a partir de la definición de tres capítulos. Para el desarrollo de los mismos fueron utilizados métodos teóricos y empíricos; los que permitieron estudiar sistemas similares a nivel nacional e internacional en función de agilizar el desarrollo del módulo y como consecuencia se definieron las tecnologías, herramientas y elementos de ingeniería de software más convenientes que permitieron darle solución al problema. Con el fin de obtener un módulo de calidad se realizaron pruebas al módulo que contribuyeron a su validación.

PALABRAS CLAVE: actividades, evidencias, formación, módulo de gestión, trabajo educativo

ABSTRACT

The need to maintain the process of training competent professionals in a way that is consistent with the times in which we live continues to be of common interest to managers and researchers in the educational sciences. In this field of education, TIC have become an important pillar. The University of Informatics Sciences promotes the personal development of the student in an integral way through all its substantive processes, encouraging active participation in different academic, scientific and extension events. In the area of educational work, the process of managing activities and their evidence is a manual process, which is why the need arises to develop a computer module, capable of computerizing the process and contributing to the management of activities and evidence. To solve the above, this research was organized from the definition of three chapters. For their development, theoretical and empirical methods were used; which allowed the study of similar systems at a national and international level in order to speed up the development of the module and as a consequence the

most convenient software engineering technologies, tools and elements were defined that allowed solving the problem. In order to obtain a quality module, tests were carried out on the module that contributed to its validation.

KEYWORDS: *activities, evidence, management module, science, training*

ÍNDICE

INTRODUCCIÓN.....	13
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS DEL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1	19
I.1 Conceptos asociados a la gestión de actividades y evidencias	19
I.2 Análisis de las soluciones existentes	20
I.2.1 Ámbito internacional	20
I.2.2 Ámbito nacional	20
I.2.3 Análisis del estudio de los sistemas	21
I.3 Análisis de las metodologías, herramientas y tecnologías a utilizar para el desarrollo del módulo	23
I.3.1 Metodología de desarrollo de software	23
I.3.2 Herramientas y tecnologías	24
Conclusiones del capítulo	28
CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN: MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS EN EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1	29
II.1 Descripción de la gestión de actividades y evidencias en el área de trabajo educativo de la Facultad 1	29
II.1.2 Modelo Conceptual	29
II.2 Propuesta de solución a la gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la Facultad 1	31
II.3 Especificación de los requisitos del módulo	32
II.3.1 Requisitos Funcionales	32
II.3.2 Requisitos No Funcionales	34
II.4 Modelo de caso de uso del sistema	36
II.4.1 Descripción de los casos de uso	38
II.3.2 Diagramas de secuencia	48
II.5 Estilo arquitectónico	50
II.5.1 Patrón arquitectónico	50
II.5.2 Patrones de diseño	52
II.6 Modelado de datos	54
II.7 Diagrama de despliegue	56
Conclusiones del capítulo	57

CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS EN EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1	58
III.1 Diagrama de componentes	58
III.2 Estándares de codificación.....	62
III.3 Validación de la propuesta de solución.....	65
III.3.1 Pruebas funcionales	66
III.3.2 Pruebas de seguridad.....	73
III.3.3 Pruebas de carga y estrés	74
III.3.4 Prueba de integración	76
III.4 Validación de los resultados de la investigación	77
III.4.1 Selección de expertos	78
III.5.2 Procesamiento de los resultados	78
III.4.3 Método de ladov	80
Conclusiones del capítulo	82
CONCLUSIONES FINALES	83
RECOMENDACIONES	84
REFERENCIAS BIBLIOGRÁFICAS	85
ANEXOS.....	88

ÍNDICE DE TABLAS

TABLA 1: VARIABLE INDEPENDIENTE	15
TABLA 2: VARIABLE DEPENDIENTE	16
TABLA 3: RESUMEN COMPARATIVO DE LAS SOLUCIONES EXISTENTES.....	21
TABLA 4: REQUISITOS FUNCIONALES	32
TABLA 5: DESCRIPCIÓN DE LOS ACTORES DEL MÓDULO	37
TABLA 6: DESCRIPCIÓN DEL CASO DE USO GESTIONAR ACTIVIDAD	38
TABLA 7: DESCRIPCIÓN DEL CASO DE USO GESTIONAR EVIDENCIA	43
TABLA 8: DESCRIPCIÓN DEL DIAGRAMA DE COMPONENTES (GESTIONAR ACTIVIDAD)	60
TABLA 9: DESCRIPCIÓN DEL DIAGRAMA DE COMPONENTES (GESTIONAR EVIDENCIA)	61
TABLA 10. ESTRATEGIA DE PRUEBAS	65
TABLA 11: DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA 1 BASADO EN EL CU-GESTIONAR ACTIVIDAD	66
TABLA 12: CASO DE PRUEBA DEL RF1-CREAR ACTIVIDAD.....	67
TABLA 13: DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA 2 BASADO EN EL CU-GESTIONAR EVIDENCIA.....	68
TABLA 14: CASO DE PRUEBA DEL RF4-CREAR EVIDENCIA	69
TABLA 15: PRUEBA DE CARGA Y ESTRÉS.....	75
TABLA 16: CUADRO LÓGICO DE IADOV	80
TABLA 17: RESULTADOS DE LAS ESCALAS DE SATISFACCIÓN	81

ÍNDICE DE FIGURAS

FIGURA 1. MODELO CONCEPTUAL	30
FIGURA 2. DIAGRAMA DE CASOS DE USO DEL SISTEMA	37
FIGURA 3. DIAGRAMA DE SECUENCIA "INSERTAR ACTIVIDAD"	49
FIGURA 4. DIAGRAMA DE SECUENCIA "EDITAR ACTIVIDAD"	49
FIGURA 5. DIAGRAMA DE SECUENCIA "ELIMINAR ACTIVIDAD"	50
FIGURA 6. MVC EN LARAVEL (2020).....	51
FIGURA 7. MVC EN MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS EN EL STE DE LA FACULTAD 1	52
FIGURA 8. MODELO DE DATOS	55
FIGURA 9. DIAGRAMA DE DESPLIEGUE	56
FIGURA 10. DIAGRAMA DE COMPONENTES DE LA GESTIÓN DE ACTIVIDADES.....	59
FIGURA 11. DIAGRAMA DE COMPONENTES DE GESTIONAR EVIDENCIAS	61
FIGURA 12. RESULTADOS PRUEBAS FUNCIONALES	72
FIGURA 13. RESULTADOS LAS PRUEBAS DE SEGURIDAD	74
FIGURA 14. PRUEBAS DE INTEGRACIÓN	77
FIGURA 15. MEDIA DE LA EVALUACIÓN DE LOS EXPERTOS A LOS INDICADORES PARA VALIDAR LA PROPUESTA DE SOLUCIÓN	79
FIGURA 16. DIAGRAMA DE SECUENCIA INSERTAR EVIDENCIA.....	88
FIGURA 17. DIAGRAMA DE SECUENCIA EDITAR EVIDENCIA	89
FIGURA 18. DIAGRAMA DE SECUENCIA ELIMINAR EVIDENCIA	89

INTRODUCCIÓN

En el transcurso de la historia del hombre, este ha sentido la necesidad de ir simplificando su modo de vida, por lo que ha pasado gran parte de esta buscando la manera de desarrollar mediante los conocimientos matemáticos máquinas capaces de simplificar las tareas de su día a día. Con la aparición de los ordenadores se logró empezar a procesar los datos mediante sistemas de información, los cuales pueden definir las relaciones que guardan los medios, los datos y la información necesaria para llevar a cabo la toma de decisiones.

La necesidad de mantener el proceso de formación de profesionales competentes de forma coherente con los tiempos que ha tocado vivir, sigue siendo interés común de directivos e investigadores de las ciencias de la educación, en este ámbito de la educación las TIC se han transformado en un pilar de suma importancia. La misma brinda a los profesores y estudiantes excelentes herramientas y metodologías que se empeñan en buscar alternativas viables para concretar la intención de mejorar los planes de formación, apoyándose en sistemas de gestión para el trabajo educativo (Peguro, 2021).

Las instituciones de educación superior en Cuba tienen como misión brindar a la sociedad un profesional formado de manera íntegra, profesionalmente competente, con preparación científica para aceptar los retos de la sociedad moderna y con un amplio desarrollo humanístico para vivir en la sociedad de esta época y servirla con sencillez y modestia, con los valores como pilar fundamental de su formación. La formación, la investigación y la extensión universitaria son procesos sustantivos de la universidad, indisolublemente ligados entre sí, y que en su integración aseguran el cumplimiento de la misión de la Educación Superior en la época actual (MES, 2022).

La caracterización de los estudiantes es uno de los procesos de mayor relevancia que se llevan a cabo como parte del trabajo educativo en las Universidades, ya que se considera un instrumento de trabajo para los profesores que contribuye a la toma de decisiones. Según el diccionario de la RAE (Real Academia de la Lengua Española), caracterización es determinar los atributos peculiares de alguien o de algo, de modo que claramente se distinga de los demás.

En correspondencia, la Universidad de las Ciencias Informáticas (UCI) promueve el desarrollo personal del estudiante de manera integral a través de todos sus procesos sustantivos, fomentando la activa participación en diferentes eventos académicos, científicos y extensionistas. La participación en este y otro tipo de actividades es uno de los indicadores a medir en la UCI a la hora de conformar la estrategia

educativa, evaluación integrada del estudiante, de otorgar la repitencia (en caso de que el estudiante lo necesite) y de otorgar el Premio al Mérito Científico a los estudiantes destacados.

Los profesores jefes de año académico que radican en el área de trabajo educativo de la facultad 1 realizan la caracterización de los estudiantes al concluir el año académico. La participación en las actividades y las evidencias de las mismas, son los dos factores por los cuales los profesores se van a guiar para la evaluación de integralidad del estudiante en el año académico vencido.

Actualmente el proceso de recopilación de las evidencias de los eventos o actividades extracurriculares en los que participan los estudiantes presenta las siguientes dificultades:

La información no cuenta con un estándar para su almacenamiento y posterior consulta ya que las fuentes disponibles pueden encontrarse en formato Excel, Word o imágenes, por lo que puede ocasionar una mala gestión del proceso de toma de evidencias. La consulta y análisis se efectúa de forma manual, lo que implica que los resultados de los reportes pueden estar desordenados, presentar errores y puede ocurrir demora en la búsqueda de información. La información no se encuentra centralizada, por lo que se incrementa la morosidad en procesos como la elaboración de la evaluación integrada y la pérdida de evidencias para conformar posteriormente los expedientes de los estudiantes.

Como resultado de la situación anterior se plantea el **problema de investigación** siguiente: ¿Cómo contribuir a la gestión de actividades y sus evidencias para mejorar el trabajo educativo de la Facultad 1 de la UCI?

Se define como **objeto de estudio** de la investigación: la gestión del trabajo educativo de la Facultad 1, enmarcado en el **campo de acción**: la gestión de evidencias y actividades para el trabajo educativo de la Facultad 1.

Con la finalidad de darle solución al problema anterior se trazó como **objetivo general**: Desarrollar un módulo que permita la gestión de las actividades y sus evidencias de los estudiantes para mejorar el trabajo educativo de la Facultad 1 de la UCI.

Para dar cumplimiento al objetivo planteado se trazaron las siguientes **tareas de investigación**:

1. Análisis de los principales conceptos y trabajos relacionados con los sistemas de gestión de actividades y evidencias.
2. Análisis del entorno para el desarrollo del módulo.
3. Estudio de la metodología y las tecnologías a emplear en el proceso de desarrollo de software.

4. Análisis y diseño del módulo para la gestión de las actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1.
5. Implementación del módulo para la gestión de las actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1.
6. Pruebas al módulo para la gestión de las actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1.

Para poder guiar la siguiente investigación se define como **hipótesis**: el módulo para la gestión de las actividades y sus evidencias contribuye a mejorar el trabajo educativo de la Facultad 1.

Variable independiente: módulo informático.

Variable dependiente: gestión.

Tabla 1: Variable Independiente

VARIABLE INDEPENDIENTE	INDICADORES	UNIDAD DE MEDIDA
Módulo Informático	Funcionalidad	Alta: funcionamiento correcto. Media: funcionalidades incompletas. Baja: errores en la implementación.
	Interfaz Gráfica	1- Errores ortográficos 2- No se ajusta a la identidad marcaria de la UCI 3- Botones no coherentes 4- Interfaz acorde a la identidad marcaria de la UCI 5- Botones coherentes
	Satisfacción del usuario	1- Clara satisfacción 2- Más satisfecho que insatisfecho 3- No definida 4- Más insatisfecho que satisfecho 5- Clara insatisfacción 6- Contradictoria

Tabla 2: Variable dependiente

VARIABLE DEPENDIENTE	DIMENSIONES	INDICADORES	UNIDAD DE MEDIDA
Gestión	Organización	Información estructurada	Utilización de identidad marcaria de la UCI
		Solo accede a la información el personal autorizado	Definición de roles
	Centralización	Información almacenada de manera centralizada	Diseño de la base de datos
	Disponibilidad	Tiempo de consulta de la información	Bien: 1 a 3 segundos Regular: 4 a 5 segundos Mal: mayor a 5 segundos

Posibles resultados:

- Módulo de gestión de actividades y evidencias del STE
- Expediente digital de los estudiantes de la Facultad 1 conformado a partir de todas las incidencias registradas en el sistema.

Los **métodos de investigación** son herramientas para la recolección de datos, formular y responder preguntas para llegar a conclusiones a través de un análisis sistemático y teórico aplicado a algún campo de estudio, clasificados en teóricos y empíricos (Shuttleworth, 2019).

Los **métodos teóricos** permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación; de ellos se emplearon los siguientes:

Histórico-Lógico: se emplea para estudiar la evolución histórica y tendencias actuales de la dirección de proyectos extensionistas, para la construcción de los referentes teóricos que sustentan la investigación, el estudio de soluciones similares, así como de las metodologías de desarrollo, las tecnologías, herramientas, *framework* (marcos de trabajo), lenguajes de programación y de modelado utilizados.

Análisis-Síntesis: empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sistema. Así como para sintetizar la información que se obtuvo mediante la entrevista con el cliente de manera que pudiera ser usada en el desarrollo del mismo, además, en la identificación de los elementos del marco teórico de la investigación.

Modelación: para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del sistema, y mediante este, identificar las características y relaciones fundamentales que den cumplimiento a los requisitos funcionales de la solución propuesta.

Los **métodos empíricos** representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a determinada elaboración racional; de ellos se emplearon los siguientes:

Entrevista: empleado en los encuentros con el cliente para obtener la información necesaria que permita determinar las características, cualidades y requisitos con los que debe contar la propuesta de solución.

Observación: se utilizó para identificar buenas prácticas y vulnerabilidades de los sistemas similares para la gestión de procesos del Sistema de Trabajo Educativo de la Facultad 1.

Análisis documental: Para la revisión bibliográfica, la revisión de las fuentes de información, el estudio de documentos clasificados, elementos teóricos que sustentan la concepción sobre la gestión de este proceso.

Los **métodos de validación** son los procesos por el cual se demuestra que los procedimientos analíticos son apto para el uso indicado, de ellos se emplearon los siguientes:

Método de Delphi: se utilizó para validar la pertinencia de la solución planteada al problema de investigación.

Método de ladov: se utilizó para el estudio de la satisfacción de los usuarios con respecto a la solución propuesta.

El presente trabajo de diploma está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones generales, recomendaciones y las referencias bibliográficas empleadas durante el desarrollo de la investigación, y por último para complementar la investigación se presentan una serie de anexos. A continuación, se muestra la descripción de los capítulos:

Capítulo 1. Fundamentación Teórica sobre el Módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1: se realiza un estudio del arte sobre los principales aspectos de la investigación. Se describen el proceso de desarrollo de software, así como las tendencias, técnicas, metodología y tecnologías usadas en la propuesta de solución.

Capítulo 2. Características y diseño del Módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1: se identifican y describen los conceptos asociados al dominio del problema y los procesos relacionados con el negocio teniendo en cuenta la metodología seleccionada en el capítulo anterior. Se definen cuáles son los requerimientos funcionales, no funcionales y el modelo de datos.

Capítulo 3. Implementación y pruebas del Módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1: en este capítulo se define el estándar de codificación que sirve de guía para la implementación de la solución propuesta, así como la estrategia de pruebas a aplicar para lograr un correcto funcionamiento.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS DEL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1

En este capítulo se definen los principales conceptos que sustentan el dominio de la investigación y que posibilitan un entendimiento profundo del tema. Se expone una valoración del estado del arte de la investigación realizada con el estudio de algunas soluciones parecidas y tendencias actuales a la hora de resolver problemas similares. Se describen, además, las principales tecnologías, herramientas y la metodología que se pretende usar para el desarrollo del sistema.

I.1 Conceptos asociados a la gestión de actividades y evidencias

Evidencias

El Diccionario de la Real Academia Española define a la evidencia como “el conjunto disponible de hechos o información que indican si una creencia o proposición es verdad o validez” (Española, 2021).

Evidencia digital

Es un tipo de evidencia física que está construida de campos magnéticos y pulsos electrónicos que pueden ser recolectados y analizados con herramientas y técnicas especiales (CASEY, 2000).

Para la presente investigación se toma como evidencias el documento que acredita o demuestra la participación a eventos científicos, extensionistas y académicos en los distintos niveles y las notificaciones de los profesores guías de la participación de los estudiantes a las actividades políticas. Las evidencias negativas como las notificaciones docentes y las ausencias a la guardia estudiantil.

Actividades

El diccionario de la Real Academia Española define actividad como “[1] la facultad de obrar. [2] Diligencia, eficacia. [3] Prontitud en el obrar. [4] Conjunto de operaciones o tareas propias de una persona o entidad” (Española, 2021).

Para la presente investigación se toma como concepto de actividad todo evento sea Jornada Científica Estudiantil, Bastión Universitario, Guardia Obrera, eventos extensionistas como Festival de Artistas Aficionados, Juegos Mellas entre otros, donde los estudiantes puedan participar y servir de apoyo para su formación académica.

I.2 Análisis de las soluciones existentes

En esta sección se realiza un análisis de soluciones similares a la que se propone, a fin de obtener las características principales de estos sistemas, los elementos claves que no pueden faltar, las arquitecturas más utilizadas y en las condiciones en las que se usan, así como cuestiones específicas de cada solución dentro del campo de acción de la presente investigación.

I.2.1 Ámbito internacional

Sistema web para la gestión estudiantil del colegio particular Pcei Buenas Nuevas

Este sistema automatiza los procesos de matrículas de estudiantes, registro de administrativos y docentes, la gestión de períodos académicos, grados y materias, registros de notas y reportes de calificaciones. Toda la información se almacena en una base de datos relacional, la cual es gestionada mediante el gestor de Base de Datos MySQL. La metodología usada fue SCRUM y como *framework* de desarrollo Laravel. (Pillalaza Quijia, 2020).

Web para la gestión de fichas estudiantiles para la Universidad de Guayaquil

Este sitio web utiliza la herramienta *Open Source* que permite crear un aplicativo web que ayuda a la gestión de fichas estudiantiles, logrando así optimizarlo. En su desarrollo se utiliza la metodología Scrum, la cual a través de sus características garantiza un entorno confiable. Este aplicativo beneficia a los docentes, directivos y alumnos de la Facultad de Arquitectura y Urbanismo de la Universidad de Guayaquil (Alvia Choez, 2018).

I.2.2 Ámbito nacional

Módulo Escalafón de integralidad para el Sistema dataFEU

Módulo que gestiona la confección del escalafón estudiantil universitario para el Sistema dataFEU. En la implementación del módulo se utilizan las herramientas y tecnologías definidas por el Sistema dataFEU con el fin de facilitar la integración. Por tal motivo se emplean el marco de trabajo Symfony 1.4.8 y Ext JS 3.3 empleando MySQL como servidor de base de datos, ajustándose a la soberanía tecnológica por la cual aboga Cuba. Además, como metodología que guía el desarrollo se emplea Proceso Unificado Ágil. Esta investigación arrojó como resultado una solución informática que realiza la confección del escalafón estudiantil universitario reduciendo el tiempo de culminación del proceso (Medero Oliva, 2015).

Sistema de Gestión de la Nueva Universidad (SIGENU)

Sistema informático distribuido del tipo Cliente-Servidor, construido sobre la base de la utilización de software libre, con el uso de Java como lenguaje de programación. Surge en el año 2004 como una iniciativa del MES. Constituye la principal línea de desarrollo para la informatización de la gestión de la información en la Educación Superior en Cuba. Actualmente el SIGENU cuenta con una base de datos de más de 120 tablas donde se almacenan todos los datos necesarios para el funcionamiento del sistema (RAJME & GUERRA).

Actualmente posee seis módulos en funcionamiento (CUJAE, 2022):

- Secretaría para el control docente.
- Estadística para la toma de decisiones.
- Profesor para la gestión de evaluaciones.
- Administrador para la gestión de usuario.
- Archivo Histórico para la gestión de los egresados y bajas.
- Seguridad para el control y seguimiento.

I.2.3 Análisis del estudio de los sistemas

Tabla 3: Resumen comparativo de las soluciones existentes

Sistemas informáticos	Ventajas	Desventajas
Sistema web para la gestión estudiantil del colegio particular Pcei Buenas Nuevas	Utiliza como metodología SCRUM que tiene como base un proceso ágil. Tipo de sistema: Web. Utiliza como <i>framework</i> de desarrollo Laravel. Utiliza como lenguaje de programación php.	Toma solo las actividades curriculares del estudiante, no enmarca las tareas extensionistas que este realiza.
Web para la gestión de fichas estudiantiles para la Universidad de Guayaquil	Optimiza gran parte del proceso de gestionar las fichas estudiantiles.	No realiza un aval del estudiante. La tecnología está desactualizada

	<p>Utiliza SCRUM que se basa en un principio de metodología ágil.</p> <p>Tipo de sistema: Web.</p> <p>Utiliza como lenguaje de programación php.</p>	
<p>Módulo Escalafón de integralidad para el Sistema dataFEU</p>	<p>Utiliza la metodología AUP en su variante UCI.</p> <p>Confecciona el escalafón estudiantil universitario.</p> <p>Tipo de sistema: Web.</p> <p>Utiliza como lenguaje de programación php.</p>	<p>Las tecnologías utilizadas no están actualizadas.</p>
<p>SIGENU</p>	<p>Presenta seis módulos activos los cuales informatizan gran parte del proceso universitario.</p>	<p>No presenta módulos para la gestión de tesis entre otros como la gestión de actividades extensionistas.</p> <p>Tipo de sistema: <i>Desktop</i>.</p> <p>Utiliza como lenguaje de programación Java.</p>

El análisis de los sistemas permitió concluir que:

- La mayoría de los sistemas fueron desarrollados para plataforma web excepto SIGENU que es una aplicación de escritorio.
- Excepto SIGENU que utiliza como lenguaje de programación Java, los otros utilizan php.
- El sistema web para la gestión estudiantil del colegio particular Pcei Buenas Nuevas utiliza como *framework* de desarrollo Laravel.
- No almacenan las evidencias de la participación de los estudiantes en las diferentes actividades que se realizan en el curso académico.

A partir de la comparación establecida anteriormente se puede decir que, aun cuando son semejantes entre sí, no responden a las necesidad del cliente porque no gestionan las actividades extracurriculares

ni las evidencias asociadas a las actividades extracurriculares o no. Utilizan tecnologías las cuales van hacer útiles en la propuesta de solución como es el lenguaje de programación PHP, como *framework* de desarrollo Laravel y que el módulo sea accesible desde la web. Todo lo expuesto anteriormente demuestra que estos sistemas no llegan a ser una solución para nuestro problema u objetivo de la investigación. Independientemente de lo antes mencionado servirán de base para la identificación de una serie de funcionalidades a incluir en la solución.

Para minimizar riesgos e incrementar las posibilidades del éxito en el desarrollo del módulo para la gestión de actividades y sus evidencias se debe utilizar una metodología de desarrollo de software ágil, tema que se abordará en el epígrafe siguiente.

I.3 Análisis de las metodologías, herramientas y tecnologías a utilizar para el desarrollo del módulo para la gestión de actividades y sus evidencias en el sistema de trabajo educativo de la Facultad 1.

I.3.1 Metodología de desarrollo de software

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo ha evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo. Por lo es de gran importancia la realización de un análisis para elegir la que resulte mejor para el proyecto en cuestión (Pacienza, 2015).

Para el desarrollo del módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1, se acordó partir de un enfoque ágil ya que se trata de un proyecto que posee un equipo de desarrollo pequeño, con poca experiencia y poca cultura de trabajo en equipo. Al realizar un estudio de las metodologías ágiles más populares, se decidió hacer uso de la metodología Proceso Unificado Ágil versión UCI (AUP-UCI, por sus siglas en inglés), en su escenario número dos. El escenario número dos garantiza el análisis concreto del negocio y una disminución de las probabilidades de fracaso en el desarrollo del sistema. Permite modelar exclusivamente los conceptos fundamentales del negocio.

I.3.2 Herramientas y tecnologías

Lenguaje de Modelado Unificado

El Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) es un lenguaje estándar para escribir diseños de software, permite visualizar, especificar, construir y documentar los artefactos de un sistema de software. Al igual que los arquitectos de edificios crean planos, los arquitectos de software usando como herramienta las prestaciones de UML, pueden crear diagramas para ayudar a los desarrolladores de software a su construcción. Además de permitir socializar y especificar el diseño de un sistema (Pressman, 2005).

Laravel 9.0

Laravel es un *framework* de los más utilizados y de mayor comunidad en el mundo de Internet. Resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web. Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas (Bautista-Villegas, 2022).

PHP 8.0

PHP es un lenguaje para programar páginas de contenido dinámico. Este es un lenguaje del lado del servidor, el servidor lee el lenguaje php, y manda los datos transformados en lenguaje HTML. El lenguaje PHP se incorpora al lenguaje normal de la página (HTML) mediante scripts. Fue creado originalmente por Rasmus Lerdorf en 1995. Para diseñar páginas con PHP se deben conocer previamente los lenguajes HTML y CSS con los que se crean las páginas web. PHP permite realizar algunas acciones que no pueden realizarse en una página estática: recoger datos del usuario, trabajar con bases de datos, crear sesiones de usuarios, restringir páginas con contraseñas, creación de foros, etc. (Bautista-Villegas, 2022).

Visual Paradigma 8.0

Es una suite de herramientas Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) para el desarrollo de aplicaciones usando UML, perfecto para la construcción de sistemas a gran escala. Aporta al proceso de desarrollo calidad, confiabilidad y estabilidad, así como soporte para

la POO. Es perfecta para la construcción de artefactos, necesarios durante todo el proceso ingenieril del desarrollo, pues permite generar muchos de forma automatizada (Aristigüi, 2017).

Microsoft Visual Studio Code 1.67.1

Es un editor de código ligero y adaptable desarrollado por Microsoft, puede correr sobre Windows, macOS y Linux. Tiene soporte para JavaScript, TypeScript and Node.js, pero a través de su ecosistema de extensiones se puede personalizar a niveles importantes, permitiendo construir un entorno de desarrollo ideal para cualquier programador, en casi cualquier lenguaje de programación, agregándole capacidades que rivalizan con cualquier Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) especializado (Microsoft, 2022).

JavaScript 8.0

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica 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. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez, 2019).

Este lenguaje fue creado con bajas expectativas, sin embargo, en poco tiempo superó a Java como el lenguaje principal para crear páginas web dinámicas. Aunque los primeros 20 años de JavaScript se caracterizaron por fallidos intentos de mejorarlo, rediseñarlo o reemplazarlo, al final de ese período se convirtió en el lenguaje de programación más difundido en el mundo no solo para páginas web, adicionándose la posibilidad de construir aplicaciones web usando Node.js, así como aplicaciones de escritorio y para teléfonos móviles, con el uso de React Native, esto sin dejar de hablar de su uso en robots y sistemas embebidos (WIRFS-BROCK, 2020).

Lenguaje de Marcación de Hipertexto 5.0

Es un lenguaje de marcado o etiquetas que se emplea para darle formato a los documentos o páginas web que se publican en Red Informática Mundial (WWW, por sus siglas en inglés). Fue desarrollado por Tim Berners-Lee basándose en el estándar Lenguaje de marcado generalizado estándar (SGML, por sus siglas en inglés). La organización *World Wide Web Consortium* (W3C, por sus siglas en inglés)

se encarga de su estandarización, para así reducir los problemas de compatibilidad con los navegadores web, que son las aplicaciones que se usan para su interpretación y comprensión humana. En su versión más reciente Leguaje de marcado de hipertexto en su versión 5 (HTML5, por sus siglas en inglés), se le da mucha importancia a la semántica de las etiquetas, ayudando con esto a la una mejor interpretación de los navegadores y los motores de búsqueda en donde se encuentren indexadas (Mora, 2016) y (CONSORTIUM, 1999).

Lenguaje Cascading Style Sheets 3.0

El Hoja de estilo en Cascada (CSS, por sus siglas en inglés) es un lenguaje de estilos estandarizado por W3C y utilizado para definir la presentación, el formato y la apariencia de un documento de marcado como por ejemplo Leguaje de marcado de hipertexto (HTML, por sus siglas en inglés). Nacieron con la necesidad de diseñar la presentación de la información de tal manera que resulte más comprensible, además de permitir facilitar dicho proceso y poderlo hacer para ofrecer distintas presentaciones de la información en dispositivos diferentes con características particulares a cada uno. En su última versión Hoja de estilo en Cascad versión 3 (CSS3, por sus siglas en inglés), se ofrecen nuevas facilidades y herramientas que permiten al diseñador o programador presentar los contenidos de forma más vistosa, en menos tiempo y con menos esfuerzo, por lo que resulta una herramienta muy atractiva (PUIG, 2013).

Bootstrap 5.0

Bootstrap es un *framework* desarrollado por Twitter, dirigido al *frontend* de una aplicación web, por lo que está orientado a la parte que se ejecuta en el cliente, en este caso el navegador web. Utiliza HTML5, CSS y JavaScript, los lenguajes por excelencia para contenido web, para hacer uso de sus librerías que permiten diseñar la web de una manera profesional creando al mismo tiempo un diseño adaptado para dispositivos móviles, con el uso de la filosofía "*Mobile first*" del inglés móvil primero, la cual consiste en enfocarse primero en la vista de teléfonos móviles y luego ir escalando hasta llegar a las vistas de mayor resolución de pantalla. Es código abierto y uno de los proyectos más populares de todos los tiempos en GitHub (GÓMEZ BERMEJO, 2018) y (NISKA, 2014).

JQuery 3.6.1

JQuery es una librería de JavaScript software libre y de código abierto, que permite simplificar el trabajo con JavaScript. Es rápida, sencilla, eficaz y poderosa, se usa para darle dinamismo y funcionalidades al *frontend* de las páginas web, mediante la manipulación del DOM y el manejo de eventos o acciones del usuario, facilita la aplicación de animaciones y la utilización de AJAX para simular el comportamiento

de una aplicación de escritorio. El hecho de simplificar las funcionalidades del lenguaje garantiza lograr mejores resultados con menor tiempo y esfuerzo (VENTURA BAUTISTA, 2021).

MySQL 8.0.27

Utiliza el Lenguaje de Consulta Estructurado (SQL, dado por sus siglas en inglés). Es un producto de código abierto desarrollado bajo la licencia GPL (Licencia Pública General GNU), lo que hace posible que pueda ser modificado por los usuarios que lo utilizan. Es el SGBD más usado y estandarizado para acceder a base de datos relacionales. Facilita la integración con programas desarrollados en C y C++ pues fue desarrollado en este lenguaje. Es un sistema multiplataforma, totalmente funcional en sistemas operativos como Linux, Mac X, UNIX y Microsoft Windows. Permite la creación de base de datos con acceso desde páginas web dinámicas, dando la posibilidad de realizar múltiples y rápidas consultas (Cobo, 2005).

Acunetix 14.8

Acunetix WVS, por sus siglas en inglés (Web Vulnerability Scanner) es una herramienta de seguridad de aplicaciones web automatizada. Es capaz de escanear cualquier sitio o aplicación web que es accesible a través del protocolo HTTP/HTTPS. Sin embargo, no todas las pruebas se pueden realizar de forma automática, por lo tanto, Acunetix WVS proporciona herramientas de penetración manuales para pruebas particulares. Comprueba diferentes vulnerabilidades (por ejemplo, inyección de SQL, Cross Site Scripting). Acunetix comprueba sobre más de 500 tipos diferentes de vulnerabilidades (Baró, 2019), estas vulnerabilidades se identifican tanto en la programación de la página como en la configuración del servidor (Santiesteban Catallops, 2019).

Apache JMeter 5.4.1

La aplicación JMeter de Apache es un software de código abierto, desarrollada en Java y diseñada para cargar el comportamiento funcional de la prueba y medir el rendimiento. Originalmente, fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. JMeter se puede usar para probar el rendimiento tanto en aplicaciones web dinámicas como estáticas. Además, se puede emplear para simular una gran carga en un servidor, grupo de servidores red u objeto para probar su fortaleza o analizar el rendimiento general bajo diferentes tipos de carga. Las características de Apache JMeter incluyen: capacidad para cargar y probar el rendimiento de diferentes aplicaciones, servidores y 21 protocolos. Fácil correlación mediante la capacidad de extraer datos de los formatos de respuestas más populares, HTML, JSON, XML o

cualquier formato de texto, núcleo altamente extensible, almacenamiento en caché y portabilidad completa (Baró, 2019).

Conclusiones del capítulo

- Se constató que es necesario el desarrollo de un módulo de gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la Facultad 1 de la Universidad de las Ciencias Informáticas.
- El análisis de las aplicaciones homólogas permitió identificar características comunes a tener en cuenta para el desarrollo de la propuesta de solución, como la utilización del *framework* Laravel, la utilización de metodologías ágiles como está presente en tres de las aplicaciones homólogas estudiadas.
- Las herramientas y tecnologías seleccionadas para el desarrollo de la propuesta de solución son libres y de licencia pública, de acuerdo con la política de migración al software libre de la Universidad y el país: PHP como lenguaje de programación, Laravel como framework de PHP, MySQL como gestor de base de datos, Bootstrap como framework de CSS y JavaScript, Apache como servidor web, Visual Paradigm como herramienta UML. La metodología seleccionada es AUP-UCI en su escenario No.2 para de esta forma emplearse en los proyectos productivos de la UCI.

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN: MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS EN EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1

El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, así como estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación del sistema. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

II.1 Descripción de la gestión de actividades y evidencias en el área de trabajo educativo de la Facultad 1

La Facultad 1 tiene un departamento de Trabajo Educativo, en el cual radican los profesores principales por año académico. Los cuales a finales de curso realizan la caracterización de los estudiantes. Esta caracterización no es más que la evaluación integral del estudiante en su año académico vencido, teniendo contenida en estas actividades extracurriculares realizadas en el curso (MES, 2022).

Los estudiantes deben presentar evidencias de su participación en las actividades extracurriculares que se desarrollan. Las evidencias pueden ser: diplomas de participantes de algún evento científico, foto en algún evento político como los desfiles del 1ro de mayo o algún premio por participar en alguna actividad extensionista, estas evidencias deben ser archivadas o guardadas para tener constancia de su participación. Esta toma de evidencias son la base para la confección de la Caracterización del estudiante.

II.1.2 Modelo Conceptual

Un modelo de conceptual es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, presentado como uno o más diagramas de clases. Se pueden utilizar para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Los objetos del dominio o clases pueden obtenerse a partir de una especificación de requisitos (GARCERANT, 2008).

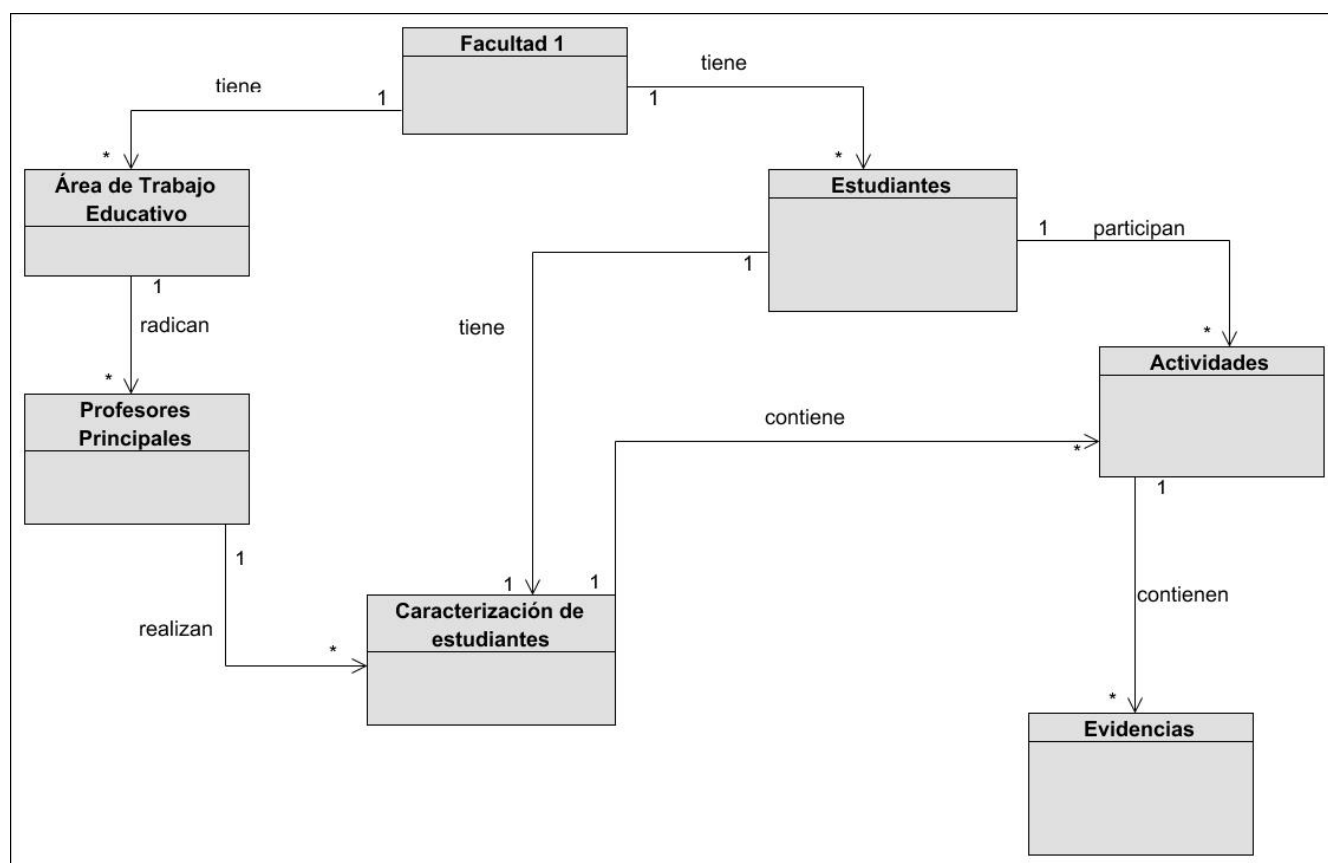


Figura 1. Modelo conceptual

Descripción de las clases del modelo

- Facultad 1: Facultad de la Universidad de las Ciencias Informáticas (UCI).
- Área de Trabajo Educativo: Área donde radican los profesores principales de años académicos.
- Profesores Principales: Son los encargados de coordinar el proceso docente-educativo del año que dirigen.
- Caracterización de estudiantes: Documento que recoge todas las actividades y evidencias de las mismas en las que participó el estudiante en el año académico vencido.
- Actividades: Procesos docentes y de carácter extensionista donde participan los estudiantes para apoyar su formación como futuros profesionales.
- Estudiante: Persona que cursa estudios en la UCI.
- Evidencias: Archivos, fotos, documentos que ratifican la participación de los estudiantes en las actividades.

II.2 Propuesta de solución a la gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la Facultad 1

En consecuencia, de la situación problemática recopilada en el capítulo precedente, la solución propuesta constituye un módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1 para conformar el expediente de los estudiantes. El módulo se desarrollará en una plataforma web, el cual será accesible mediante internet o intranet a través de un ordenador o un dispositivo móvil, el cual tenga en su sistema instalado algún navegador. El módulo debe tener la posibilidad de acceder de manera rápida a la base de datos la cual debe estar centralizada en un mismo servidor para obtener un mejor acceso a la información del Sistema de Trabajo Educativo de la Facultad 1 y una mayor eficiencia y control sobre los datos a gestionar.

El módulo para la gestión de las actividades y sus evidencias permitirá gestionar las actividades de los estudiantes de la facultad 1 de la Universidad de las Ciencias Informáticas. Este módulo asignará roles a usuarios que estén vinculados con la conformación de la caracterización de los estudiantes. Se vincularán usuarios a los subprocesos de gestión de actividades. Gestionará las evidencias que pertenecen a dichas actividades. Permitirá generar informes según actividades y evidencias específicas. Permitirá además generar listas de actividades y evidencias. Notificará al usuario cuando este inserte una actividad o evidencia. Al culminar el proceso de insertar las evidencias por actividades permitirá generar un documento PDF donde estarán organizadas las evidencias según la actividad a la que pertenece.

Roles y permisos

Para garantizar la seguridad del módulo y la protección de la información, se les asignan roles y permisos a los diferentes usuarios, para poder controlar las acciones que se puedan realizar y las funcionalidades que ofrece el módulo en dependencia del usuario. A continuación, se muestran los roles específicos y permisos otorgados para acceder al módulo:

- Profesor Principal: usuario con permiso para gestionar las actividades y las evidencias. Este podrá generar los reportes de las actividades con sus evidencias correspondientes.
- Estudiante: usuario con permiso para gestionar las evidencias. Este usuario también puede ver el listado de las actividades correspondientes.

II.3 Especificación de los requisitos del módulo

Con la finalidad de que el módulo propuesto se desarrolle con las necesidades del cliente, a través del uso de la técnica tormentas de idea, se llegaron a obtener los requisitos funcionales y no funcionales, que no son más que: la condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo (IEEE, 2011).

II.3.1 Requisitos Funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville., 2005)

Teniendo en cuenta las características de la propuesta de solución, en el proceso de levantamiento de requisitos se identificaron un total de 16 requisitos funcionales. A continuación, la tabla 4 muestra los requisitos del módulo.

Tabla 4: Requisitos Funcionales

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Registrar Actividad	El módulo debe permitir al usuario crear una actividad	Alta	Alta
RF2	Modificar Actividad	El módulo debe permitir al usuario modificar una actividad	Alta	Media
RF3	Eliminar Actividad	El módulo debe permitir al usuario eliminar una actividad	Media	Baja
RF4	Registrar Evidencia	El módulo debe permitir al usuario crear una evidencia	Alta	Alta
RF5	Modificar Evidencia	El módulo debe permitir al usuario modificar una evidencia	Alta	Media

RF6	Eliminar Evidencia	El módulo debe permitir al usuario eliminar una evidencia	Media	Baja
RF7	Generar reporte de Actividad	El módulo debe permitir al usuario generar un reporte de la cantidad de actividades según el tipo de actividad	Alta	Alta
RF8	Generar reporte de Evidencia	El módulo debe permitir al usuario generar un reporte de la cantidad de evidencias según el tipo de actividad	Alta	Alta
RF9	Buscar Actividad	El módulo debe permitir al usuario buscar una actividad específica	Media	Baja
RF10	Buscar Evidencia	El módulo debe permitir al usuario buscar una evidencia específica	Media	Baja
RF11	Listar Actividades	El módulo debe permitir al usuario listar todas las actividades almacenadas en la base de datos	Media	Media
RF12	Listar Evidencias	El módulo debe permitir al usuario listar todas las evidencias almacenadas en la base de datos	Media	Media
RF13	Registrar Usuario	El módulo debe permitir registrarse a un usuario para poder acceder a las distintas funcionalidades	Alta	Alta

RF14	Generar reporte de participantes por tipo de actividades	El módulo debe permitir generar un reporte de la cantidad de participantes que hay por cada tipo de actividad	Alta	Alta
RF15	Generar reporte de cantidad de participantes por actividades	El módulo debe permitir generar un reporte en forma tabular con los nombres de las actividades y la cantidad de participantes que tienen	Alta	Alta
RF16	Generar reporte de cantidad de actividades por años	El sistema debe permitir generar un reporte de la cantidad de actividades que hay registrada por años	Alta	Alta

II.3.2 Requisitos No Funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville., 2005).

Dentro del aspecto que debe caracterizar el módulo propuesto se encuentran requisitos de: usabilidad, confiabilidad, portabilidad, eficiencia, funcionalidad y mantenibilidad.

Usabilidad:

- RnF 1: El módulo debe presentar una interfaz agradable y de buen gusto para el usuario.
- RnF 2: El módulo podrá ser visualizado en dispositivos desde las resoluciones 768x1024 y 1366x768.
- RnF 3: El módulo debe gestionar y requerir información de usuarios para su uso.

Confiabilidad:

- RnF 4: El módulo debe mostrar solo la información necesaria, de manera que el usuario logre estar orientado todo el tiempo.

Portabilidad:

- RnF 5: La propuesta de solución se debe ejecutar en los navegadores existentes en la Universidad, así como adaptar su interfaz a cualquier dispositivo que acceda al módulo.

Seguridad:

- RnF 6: La información manejada por el módulo está protegida con el objetivo de evitar accesos no autorizados.
- RnF 7: Ante los posibles errores que puedan ocurrir en el módulo, no se debe mostrar detalles específicos de información que pueda comprometer su seguridad.
- RnF 8: El módulo deberá ser capaz de cerrar la sesión del usuario una vez pasado un determinado tiempo de inactividad.
- RnF 9: Se garantizará la integridad de la información mediante mecanismos de control de acceso, utilizando usuario, contraseña y niveles de accesos para cada usuario de acuerdo a su rol, asegurando que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.

Apariencia o Interfaz externa:

- RnF 10: Empleo de colores claros e interfaces sencillas para el usuario.

Eficiencia:

- RnF 11: El tiempo de demora de una petición al servidor debe ser menor de 8 a 10 segundos aproximadamente.

Mantenibilidad:

- RnF 12: Se debe permitir realizar modificaciones posteriores para mejorar el módulo o en caso de que existan nuevas exigencias por parte del cliente.
- RnF 13: El módulo debe estar bien documentado de manera tal que el tiempo de mantenimiento sea lo más pequeño posible.

Hardware:

- RnF 14: El servidor de base de datos debe poseer una capacidad mínima de 20 GB.
- RnF 15: El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB.
- RnF 16: Los servidores web y de base de datos deben poseer como mínimo 2 GB de memoria RAM.
- RnF 17: Se requiere una tarjeta de red con velocidad de 100 Mbps.
- RnF 18: Se requiere 2 GB de memoria RAM como mínimo.
- RnF 19: Se requiere un procesador de 2 GHZ como mínimo.

Software:

- RnF 20: Un navegador Chrome versión 83.0 o superior; o Mozilla Firefox versión 84.0 o superior.
- RnF 21: Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows 10.
- RnF 22: Servidor Web Apache con PHP 8.0.3
- RnF 23: MySQL 8.0 como gestor de bases de datos.

II.4 Modelo de caso de uso del sistema

Un caso de uso de sistema es una secuencia de acciones que un sistema lleva a cabo que da lugar a un resultado de valor observable para un actor particular (alguien o algo fuera del sistema que interactúa con el sistema) (IBM, 2022).

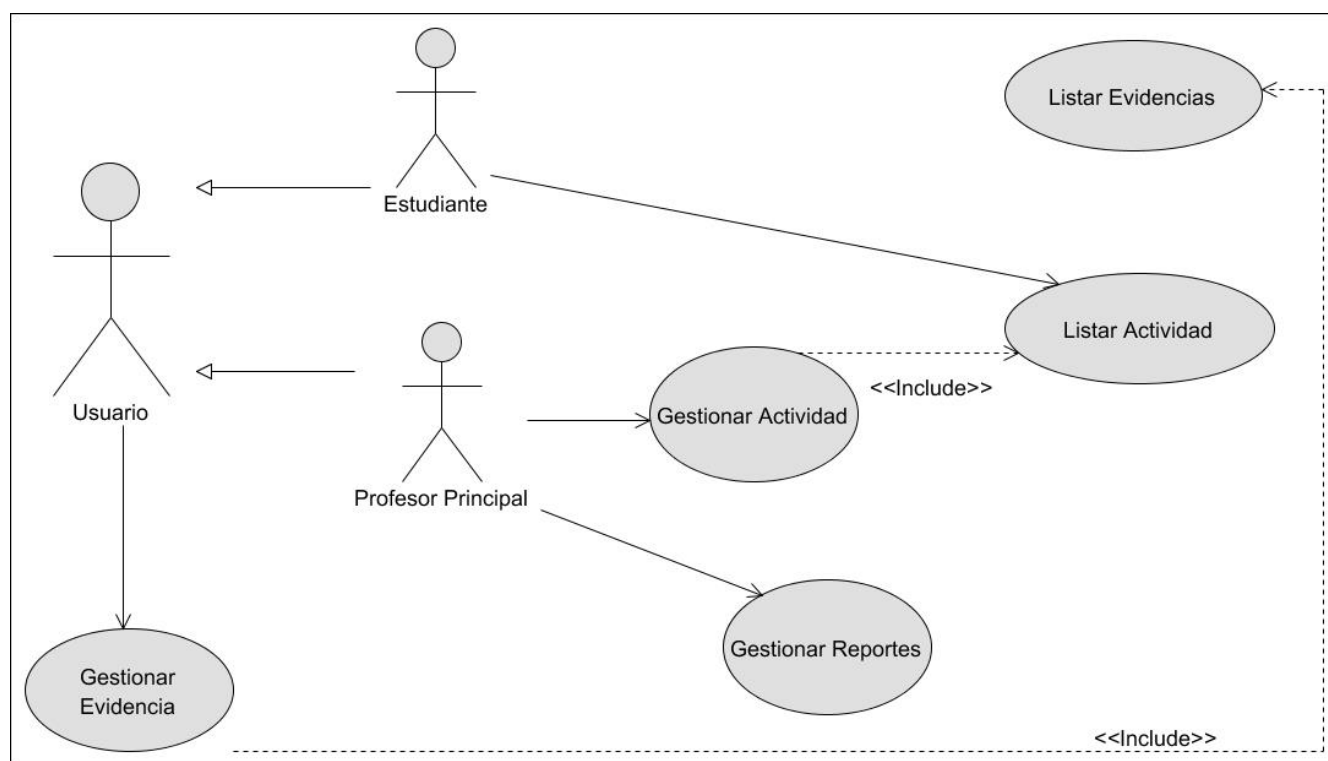


Figura 2. Diagrama de Casos de Uso del Sistema

La figura 2 muestra el diagrama de Casos de Uso diseñado para el módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1. En la tabla 5 se puede apreciar la descripción de los actores del módulo.

Tabla 5: Descripción de los actores del módulo

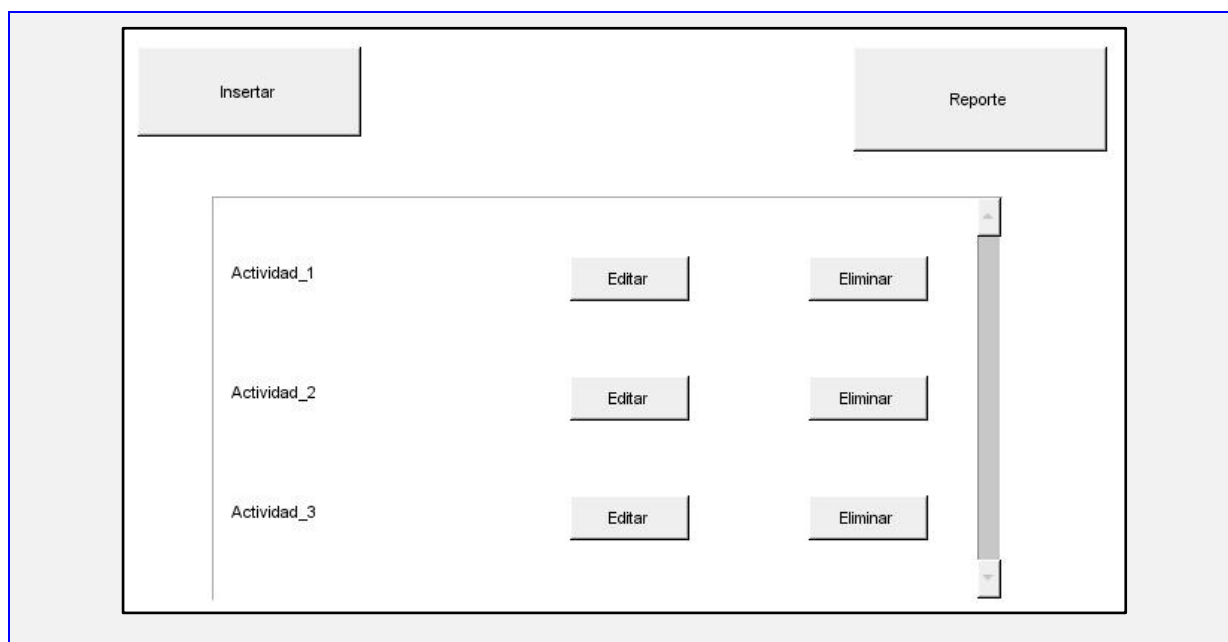
Actores del Sistema	Descripción
Usuario	Usuario que interactúan con el módulo.
Profesor Principal	Gestiona las actividades y las evidencias. Lista las actividades y las evidencias. Genera un PDF de las actividades y sus respectivas evidencias.
Estudiante	Gestiona las evidencias y tiene acceso a listar las actividades.

II.4.1 Descripción de los casos de uso

Caso de Uso No 1. Gestionar Actividad

Tabla 6: Descripción del caso de uso Gestionar Actividad

Objetivo	Permitir registrar, modificar y eliminar datos acerca de una actividad.	
Actores	Profesor Principal	
Resumen	El caso de uso debe permitir al profesor principal del año de insertar, modificar o eliminar una actividad.	
Complejidad	Alta.	
Prioridad	Alto.	
Precondiciones	Profesor principal ya autenticado.	
Poscondiciones	Se registró, modificó o se eliminó actividad(s).	
Flujo de eventos		
Flujo básico: Gestionar actividad		
Actor	Sistema	
1.	Selecciona de la página principal la opción "Actividad".	
2.		Muestra una pantalla con un listado de actividades y permite Registrar, Modificar o Eliminar actividades(s).
Prototipo elemental de interfaz gráfica		



Flujos alternos

2a. Listado de actividades vacío

Actor		Sistema
1.		Carga una pantalla en blanco sin actividades registradas.

Sección 1: “Crear actividad”

Flujo básico: Registrar Actividad

Actor		Sistema
1.	Presiona el botón “Insertar actividad”.	
2.		Muestra un formulario con los siguientes campos a introducir: <ul style="list-style-type: none"> • Nombre de la actividad. • Año. • Tipo de Actividad • Botón cancelar • Botón Insertar Actividad
3.	Introduce los datos y presiona el botón “Insertar Actividad”.	

4.		Verifica que todos los campos estén llenos.
5.		Verifica que los datos introducidos estén correctos.
6.		Almacena los datos de la actividad y muestra el mensaje “La actividad se insertó con éxito”. Finalizando así el caso de uso.

Prototipo elemental de interfaz gráfica

Flujos Alternos

4a. Campos vacíos

Actor		Sistema
1.		Muestra el mensaje “Campos vacíos”.

5a. Datos incorrectos

Actor		Sistema
1.		Muestra el mensaje “Datos incorrectos”.

Sección 2: “Modificar Actividad”

Flujo básico: Editar Actividad		
Actor		Sistema
1.	Marca la opción “Editar” de una de las actividades.	
2.		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> • Nombre de la actividad. • Año. • Tipo de Actividad • Botón Cancelar • Botón Editar
3.	Realiza las actualizaciones deseadas y presiona el botón “Editar Actividades”.	
4.		Verifica que todos los campos estén llenos.
5.		Verifica que los datos introducidos estén correctos.
6.		<p>Actualiza la información incorporada a la actividad y se emite un mensaje “La actividad se modificó con éxito”.</p> <p><i>Finalizando así el caso de uso.</i></p>
Prototipo elemental de interfaz gráfica		

Editar Actividad

Nombre: Tipo Actividad:

Año:

Flujos Alternos

4a. Campos vacíos

Actor		Sistema
1.		Muestra el mensaje “Campos vacíos”.

5a. Datos incorrectos

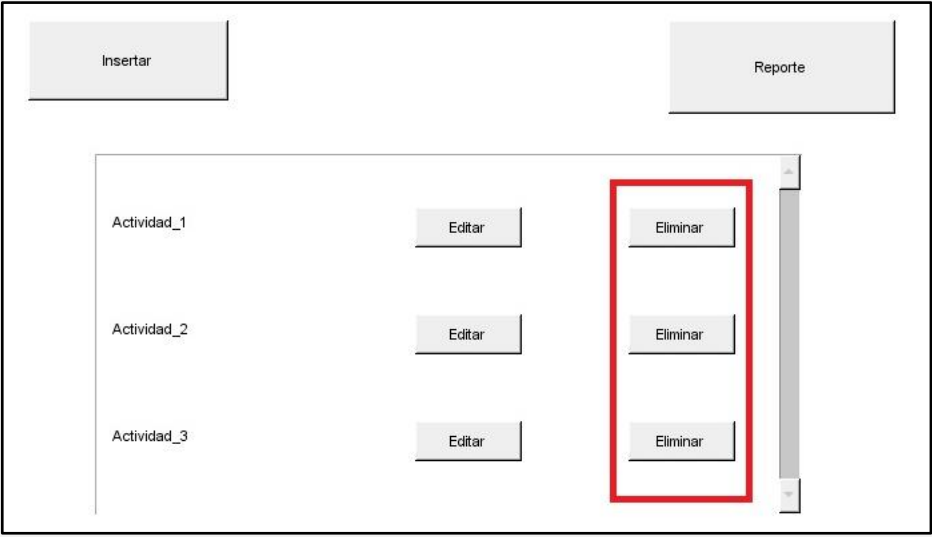
Actor		Sistema
1.		Muestra el mensaje “Datos incorrectos”.

Sección 3: “Eliminar actividad”

Flujo básico Eliminar Actividad

Actor		Sistema
1.	Presiona la opción “Eliminar” de una actividad mostrada.	
		Muestra el mensaje “La actividad se eliminó con éxito”.

Prototipo elemental de interfaz gráfica

		
Relaciones	CU Incluidos	No Aplica
	CU Extendidos	Ninguno
Requisitos no funcionales	RnF 5, RnF 9	
Asuntos pendientes	No Aplica	

Caso de Uso No2. Gestionar Evidencia

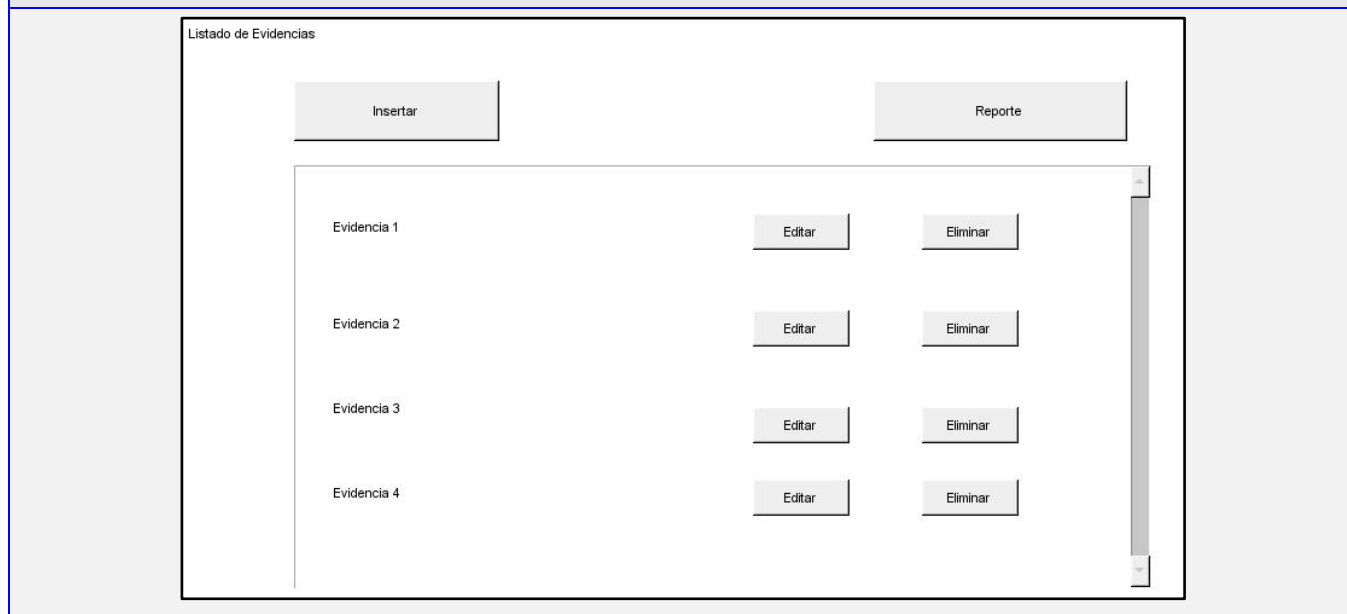
Tabla 7: Descripción del caso de uso Gestionar Evidencia

Objetivo	Permitir registrar, modificar y eliminar datos acerca de una evidencia.
Actores	Profesor Principal, Estudiantes
Resumen	El caso de uso debe permitir al profesor principal del año o a algún estudiante de insertar, modificar o eliminar una evidencia.
Complejidad	Alta.
Prioridad	Alto.
Precondiciones	Profesor principal ya autenticado, estudiante ya autenticado.
Poscondiciones	Se registró, modificó o se eliminó evidencia(s).
Flujo de eventos	

Flujo básico: Gestionar evidencia

Actor		Sistema
3.	Selecciona de la página principal la opción "Evidencia".	
4.		Muestra una pantalla con un listado de evidencias y permite Registrar, Modificar o Eliminar evidencia(s).

Prototipo elemental de interfaz gráfica



Flujos alternos

2a. Listado de evidencias vacío

Actor		Sistema
1.		Carga una pantalla en blanco sin evidencias registradas.

Sección 1: "Crear evidencia"

Flujo básico: Registrar Evidencia

Actor		Sistema
7.	Presiona el botón "Insertar evidencia".	

8.		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> • Descripción. • Año. • Actividad • Archivo • Botón cancelar • Botón Insertar
9.	Introduce los datos y presiona el botón "Insertar".	
10.		Verifica que todos los campos estén llenos.
11.		Verifica que los datos introducidos estén correctos.
12.		<p>Almacena los datos de la evidencia y muestra el mensaje "La evidencia se insertó con éxito".</p> <p><i>Finalizando así el caso de uso.</i></p>

Prototipo elemental de interfaz gráfica

Flujos Alternos

4a. Campos vacíos

Actor		Sistema
1.		Muestra el mensaje “Campos vacíos”.
5a. Datos incorrectos		
Actor		Sistema
1.		Muestra el mensaje “Datos incorrectos”.
Sección 2: “Modificar evidencia”		
Flujo básico: Editar Evidencia		
Actor		Sistema
1.	Marca la opción “Editar” de una de las evidencias.	
2.		Muestra un formulario con los siguientes campos a introducir: <ul style="list-style-type: none"> • Descripción. • Año. • Actividad • Archivo • Botón cancelar • Botón Editar
3.	Realiza las actualizaciones deseadas y presiona el botón “Editar”.	
4.		Verifica que todos los campos estén llenos.
5.		Verifica que los datos introducidos estén correctos.
6.		Actualiza la información incorporada a la evidencia y se emite un mensaje “La evidencia se modificó con éxito”. <i>Finalizando así el caso de uso.</i>
Prototipo elemental de interfaz gráfica		

Editar Evidencia

Descripción:

Año:

Actividad:

Archivo:

Flujos Alternos

4a. Campos vacíos

Actor	Sistema
1.	Muestra el mensaje "Campos vacíos".

5a. Datos incorrectos

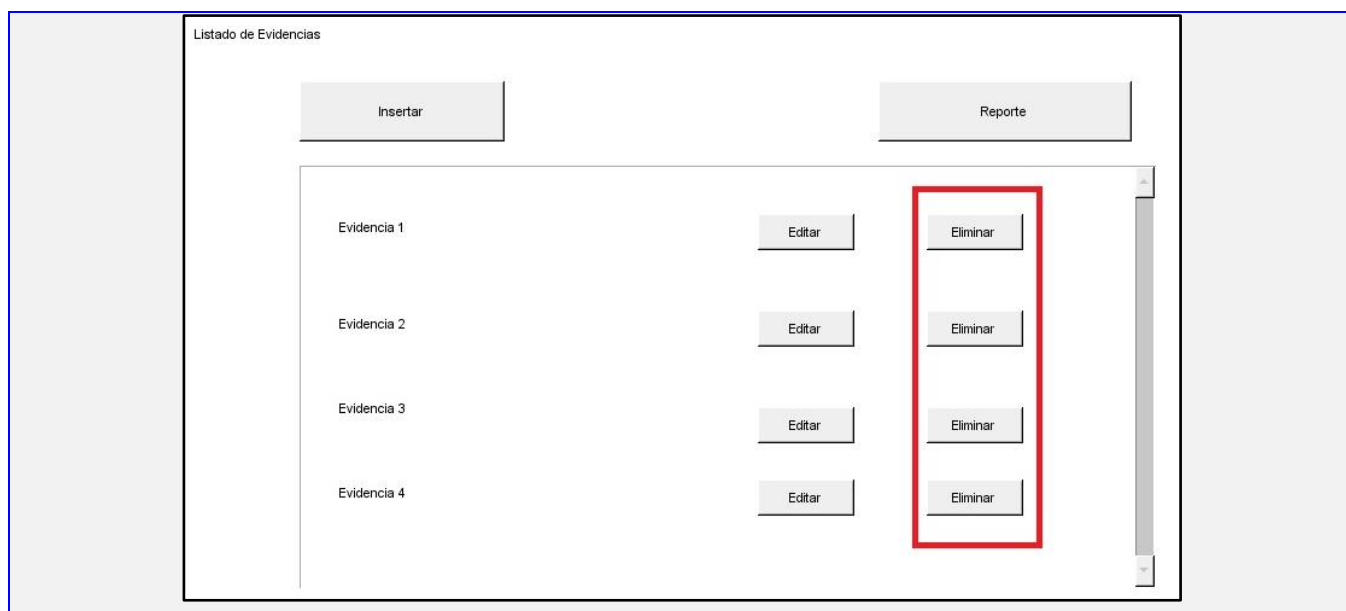
Actor	Sistema
1.	Muestra el mensaje "Datos incorrectos".

Sección 3: "Eliminar evidencia"

Flujo básico Eliminar Evidencia

Actor	Sistema
1. Presiona la opción "Eliminar" de una evidencia mostrada.	
	Muestra el mensaje "La evidencia se eliminó con éxito".

Prototipo elemental de interfaz gráfica



Relaciones	CU Incluidos	No Aplica
	CU Extendidos	Ninguno
Requisitos funcionales	no	RnF 5, RnF 9
Asuntos pendientes		No Aplica

II.5. Diagramas de secuencia

Los diagramas de secuencia son una solución de modelado dinámico popular en UML porque se centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine (Inc, 2022).

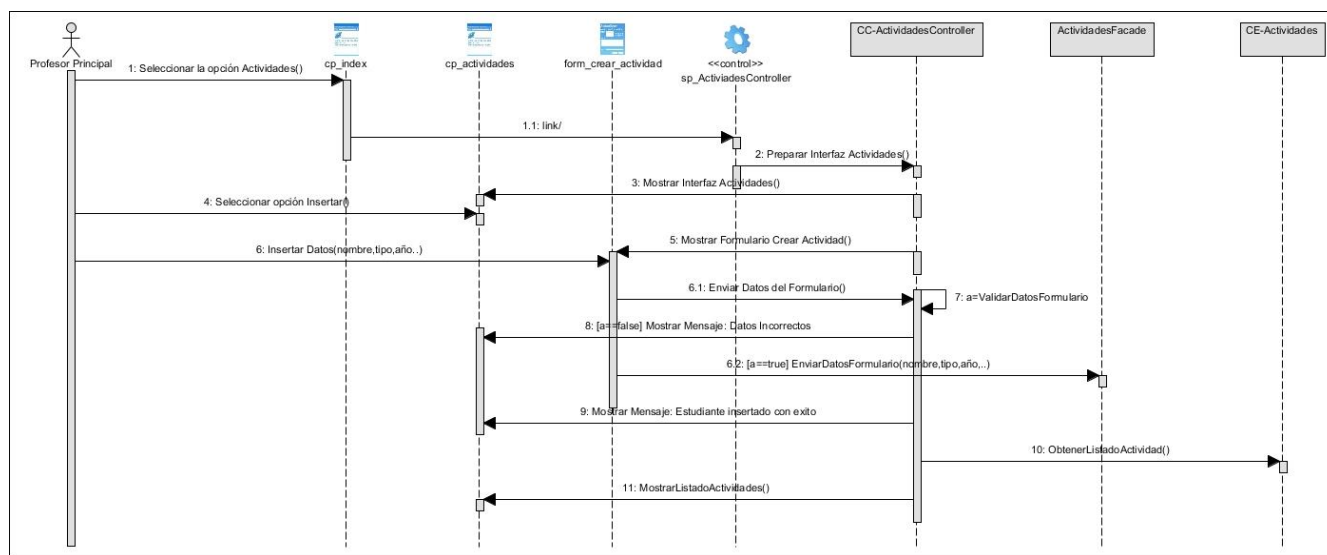


Figura 3. Diagrama de secuencia "Insertar Actividad"

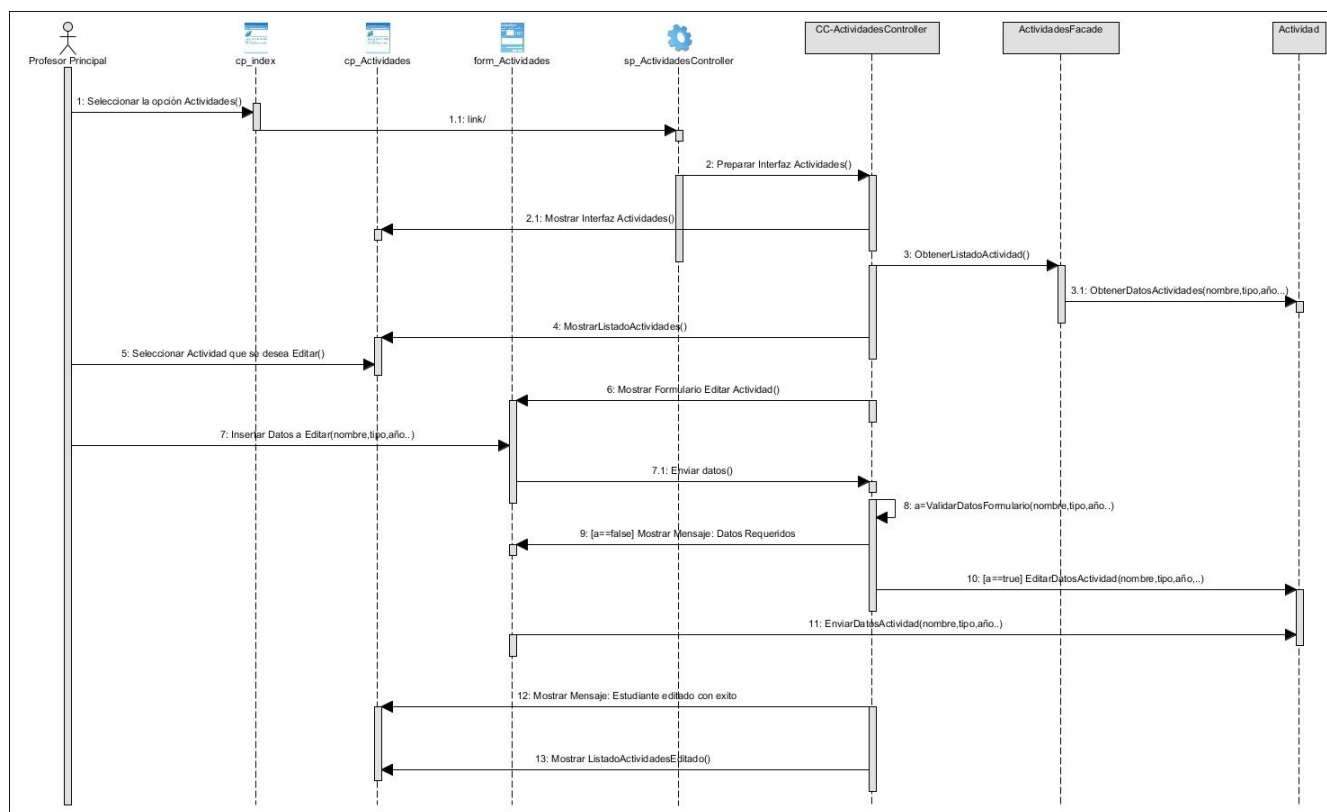


Figura 4. Diagrama de secuencia "Editar Actividad"

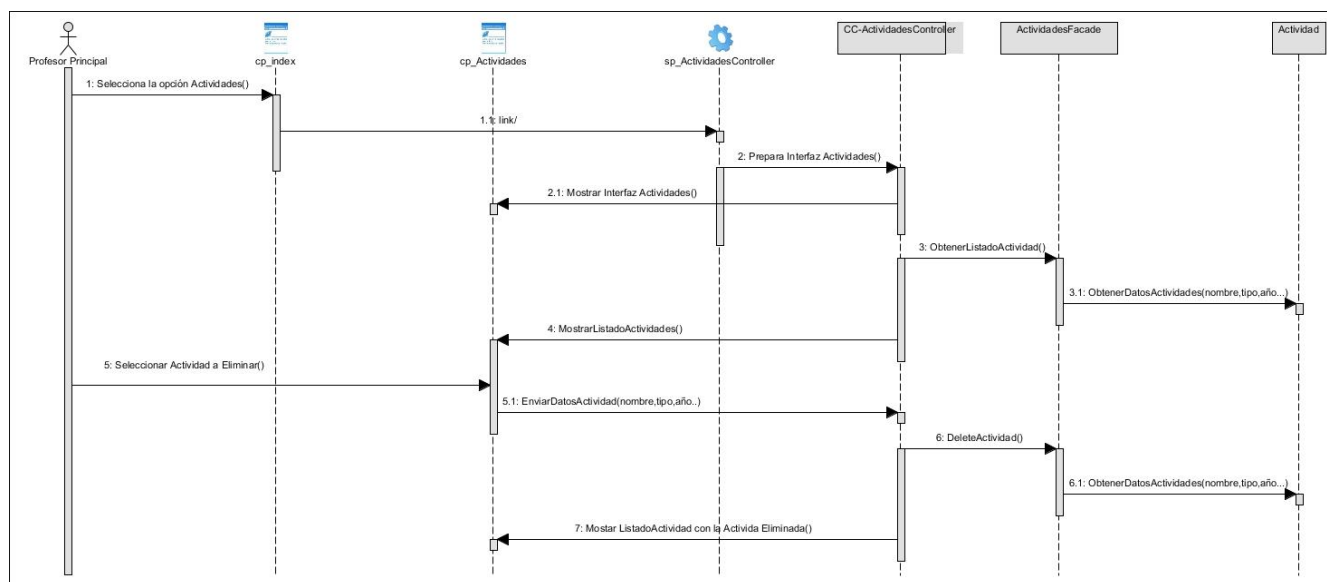


Figura 5. Diagrama de secuencia "Eliminar Actividad"

II.6 Estilo arquitectónico

Es un conjunto de principios que proveen un marco de trabajo abstracto para una familia de sistemas que promueve la reutilización de componentes y diseños mediante soluciones a problemas recurrentes. Es decir, un estilo arquitectónico es un conjunto de decisiones y principios que encajan como solución en determinados tipos de problemas que suelen repetirse en varios sistemas (BARRERA, 2018).

II.6.1 Patrón arquitectónico

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Permiten tener una estructura de código común, por lo que ahorran gran cantidad de tiempo en la construcción de software (Daudinot Hamilton, 2015).

MVC (Modelo – Vista - Controlador) es un patrón arquitectónico de software que separa una aplicación en tres capas descritas como su acrónimo lo indica. Laravel, así como la mayoría de *framework* en PHP, implementa este patrón de diseño en donde cada capa maneja un aspecto de la aplicación.

Modelo: Hace referencia a la estructura de datos de la aplicación. Los datos pueden ser transferidos desde la base de datos, una clase, un servicio, u otros, directamente a la vista o ser transformados en el controlador para ser actualizados nuevamente al origen.

Vista: Es la representación de la información en una interfaz de usuario. Por lo general en interfaces no estáticas se representan los datos que vienen directamente del modelo o estos son transformados en un proceso intermedio en el controlador. En vistas estáticas por lo general no hace falta que las vistas sean renderizadas con datos enviados del controlador.

Controlador: Es el lugar en donde se implementa la lógica de la aplicación, los procedimientos, algoritmos y rutinas que hacen que funcione el software. Actúa como interfaz entre los componentes de modelo y vista aplicando las transformaciones y lógica necesarias.

El MVC en Laravel está implementado de la siguiente manera. En una aplicación web, los controladores estarán situados en la carpeta `app/Http/Controllers`, los modelos directamente en `app` y las vistas en `resources/views`.

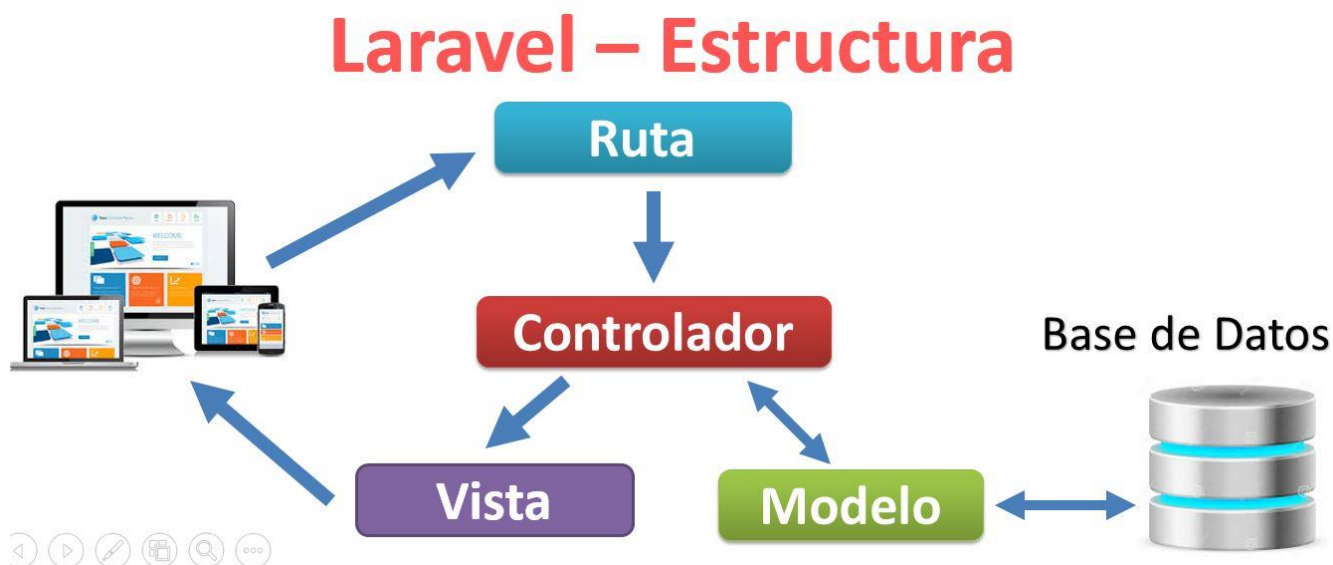


Figura 6. MVC en Laravel (2020)

En el presente módulo se evidencia la utilización del MVC de Laravel de la siguiente forma:

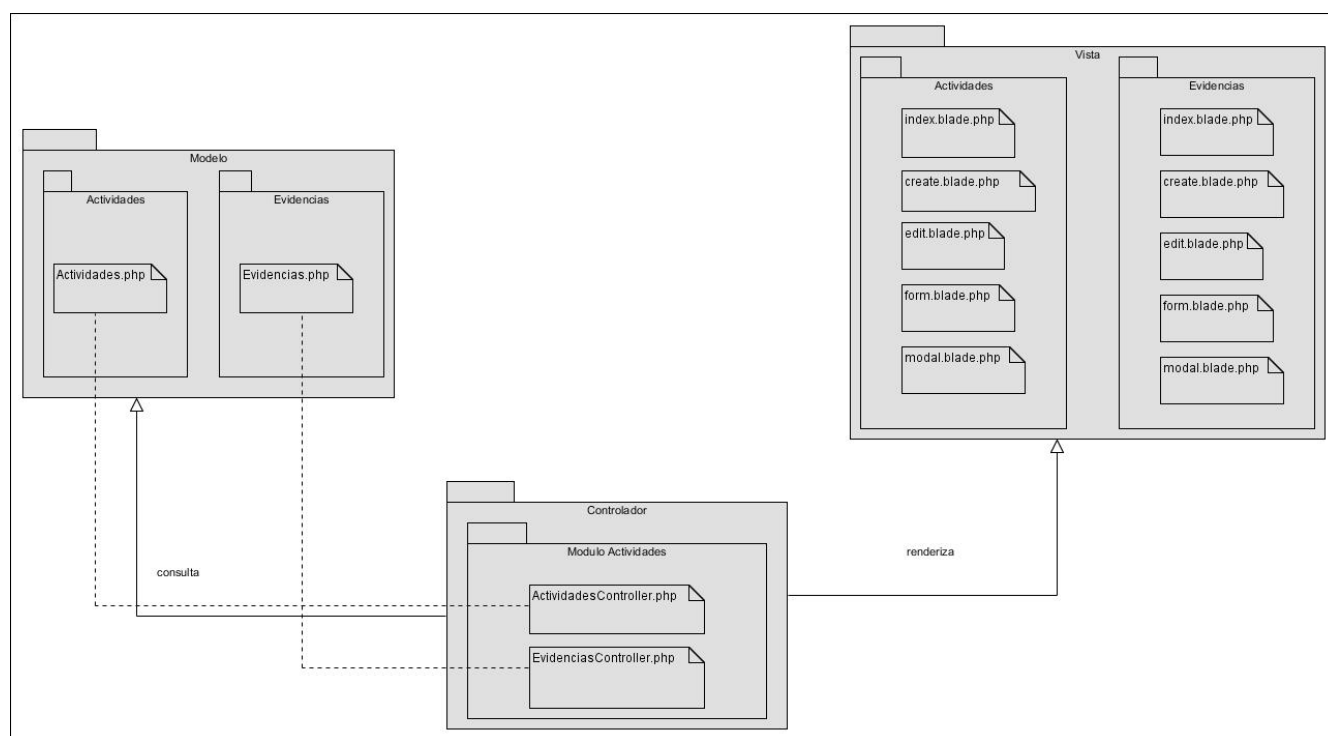


Figura 7. MVC en Módulo para la gestión de actividades y sus evidencias en el STE de la Facultad 1

II.6.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Permiten tener una estructura de código común, por lo que ahorran gran cantidad de tiempo en la construcción de software (Daudinot Hamiltón, 2015).

Patrones Generales de Software para la Asignación de Responsabilidades (GRASP):

Estos patrones se utilizan en la programación orientada a objetos, con lo que el punto de partida para aplicarlos siempre será usar POO. Cómo definir estos objetos/clases será la aplicación de cada patrón. Con el objetivo de alcanzar una solución flexible y reusable, se pusieron en práctica los siguientes patrones GRASP que a continuación se describen:

- **Experto:** El patrón Experto se usa con frecuencia en la asignación de responsabilidades, es un principio básico que se utiliza continuamente en el diseño de objetos. El denominado experto expresa por lo general la intuición común de que los objetos realizan las acciones relacionadas

con la información que poseen. Se puede observar la clase entidad Actividades.php, experta en información relacionada con las actividades.

- **Bajo Acoplamiento:** El bajo acoplamiento entre objetos se refiere a cuánta relación tienen los objetos entre sí. En la presente investigación, por ejemplo, al mostrar la información de las actividades a las que pertenece cada evidencia en el listado de evidencias, se tiene que recorrer la lista de actividades existentes, por lo que entonces aumentamos el acoplamiento entre Actividades.php y Evidencias.php y proporcionando una baja dependencia entre las clases.
- **Alta Cohesión:** Este patrón conduce la asignación de responsabilidades relacionadas con un número relativamente pequeño. Para ello propone asignar a las clases las responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad. En la presente investigación, el patrón se evidencia mayormente en cada clase que realiza una labor única dentro del sistema y colabora con las otras para llevar a cabo tareas específicas, como por ejemplo la clase entidad Actividades.php la cual colabora con la clase evidencias para cumplir sus objetivos.
- **Controlador:** Este patrón define quién debería encargarse de atender un evento del sistema. La solución consiste en asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controladora. En la investigación se ve reflejado por ejemplo ActividadesController.php, ya que contienen todos los métodos necesarios para controlar las actividades que se soliciten con respecto a las actividades.
- **Creador:** Se basa principalmente en la asignación de responsabilidades relacionadas con la creación de objetos. Es el encargado de crear las instancias de los objetos para cada una de las funcionalidades de la aplicación. El patrón se evidencia en las clases controladoras como por ejemplo EvidenciasController.php, el cual se encarga de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento.

Patrones GOF

Los patrones GOF, conocidos como el grupo de los cuatro, se clasifican en tres grupos dependiendo de sus propósitos, Creacional, Estructural y Comportamiento. Estos describen soluciones simples a problemas específicos en el diseño de software orientado a objetos y otros ámbitos referentes al diseño de interfaces (Daudinot Hamilton, 2015).

Patrón decorador: Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir

funcionalidades. Se puede observar el uso de este patrón en la propuesta de solución en todas las vistas, ya que todas heredan de una plantilla padre determinada. En el presente segmento de código fuente se ejemplifica lo antes descrito:

En la vista actividades/ en los archivos *create.blade.php*, *edit.blade.php*, *index.blade.php*, *show.blade.php*, *form.blade.php*:

```
@extends('adminlte::page')
```

Factory: Ayuda a crear instancias de otras clases, con la finalidad de ocultar la complejidad que se requiere para crearlas. Uno de los casos en el cual se debería utilizar Factory en laravel es cuando el objeto que necesitamos crear, puede cambiar y afectar su utilización, por lo que en cada modelo se utiliza el uso de *HasFactory* ayudando a que algún cambio en los atributos de esta clase no afectara en el controlador en el cual será llamado.

II.7 Modelado de datos

El modelado de datos es el proceso de creación de una representación visual o esquema que define los sistemas de recopilación y administración de información de cualquier organización. Este esquema o modelo de datos ayuda a las diferentes partes interesadas, como analistas de datos, científicos e ingenieros, a crear una vista unificada de los datos de una organización. El modelo esboza los datos que recoge la empresa, la relación entre los distintos conjuntos de datos y los métodos que se usarán para almacenarlos y analizarlos (Amazon, 2022).



Figura 8. Modelo de datos

Descripción de las tablas:

- **Tabla Usuario:** Contiene los datos necesarios para poder acceder al sistema.
- **Tabla roles:** Contiene los roles que van a identificar a cada usuario del sistema.
- **Tabla *permissions*:** Es la que tiene los permisos para que los usuarios según el rol puedan gestionar las actividades y evidencias.
- **Tabla *roles_has_permissions*:** Tabla que almacena los roles de los usuarios con sus respectivos permisos asignados.
- **Tabla profesores:** Almacena todos los datos de los profesores que van a intervenir.
- **Tabla Estudiantes:** Contiene todos los datos del estudiante para posteriormente registrarlo en el sistema.
- **Tabla actividades:** Almacena todas las actividades en las cuales los estudiantes participan en el año académico que transitan.

- **Tabla evidencias:** Recoge todas las evidencias que reflejan y demuestran la veracidad de la participación del estudiante en una actividad.

II.8 Diagrama de despliegue

Los elementos del diseño al nivel del despliegue indican cómo se ubicarán la funcionalidad y los subsistemas dentro del entorno computacional físico que soportará al software. En el modelo de despliegue se define la organización física del sistema en términos de nodos de cómputo y se verifica que los casos de uso puedan implementarse como componentes que se ejecutan en esos nodos (González Rodríguez, 2017).

El siguiente diagrama de despliegue describe la arquitectura física del módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1 durante la ejecución, en términos de procesadores, dispositivos y componentes de software. Permite una mejor comprensión entre la correspondencia de la arquitectura de software y la arquitectura de hardware.

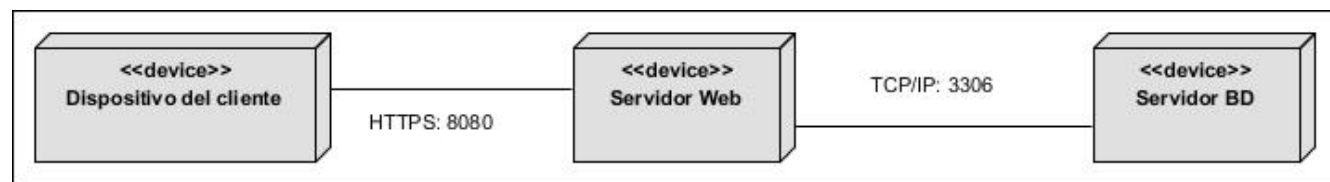


Figura 9. Diagrama de Despliegue

Dispositivo del cliente: Contiene la estación de trabajo cliente que necesita un navegador web que conecte con la aplicación web (Apache) situado en el servidor (computadora personal, Tablet, teléfono celular), el cual utiliza para conectarse el protocolo de comunicación HTTPS (Protocolo Seguro de Transferencia de Hipertexto), el cual debe disponer de un ordenador con sistema operativo Windows y cuyas propiedades mínimas sean un procesador Intel con velocidad de 2.3 GHz, 4GB de memoria RAM y 10240MB de disco duro.

Servidor Web: Aquí se encuentra el código fuente de la aplicación, brinda a los usuarios las interfaces necesarias durante todos los procesos del negocio. Esta estación se comunica con el servidor de base de datos MySQL donde se almacenan los datos de los perfiles de usuarios, realizando la comunicación a través del protocolo TCP.

Servidor de BD: Este nodo es el encargado del almacenamiento de los datos de los perfiles de usuario. Se comunica con el servidor de aplicaciones a través del protocolo TCP, permite el acceso a la información que tanto el usuario como la aplicación pueden manipular, es el resultado almacenado de las iteraciones del software.

Conclusiones del capítulo

En este capítulo fueron descritos los procesos que intervienen en el negocio y se arribó a las siguientes conclusiones:

- Se realizó el levantamiento de requisitos que permitió identificar las funcionalidades que debe tener el sistema.
- Mediante la descripción detallada de los casos de uso se pudieron definir los detalles internos sobre qué debe hacer el sistema como respuesta a las acciones del actor.
- La generación de los artefactos relacionados con el flujo de análisis y diseño, teniendo en cuenta la arquitectura MVC que Laravel establece, permitió obtener una mayor comprensión del sistema y definir los principios que guiarán la implementación y organización de la misma.
- A través del modelado del análisis se obtuvo una visión general conceptual del sistema y mediante el modelado del diseño se obtuvo una abstracción de la implementación del sistema.

Por tanto, se considera que se han creado las condiciones para efectuar la implementación del Módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1.

CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO PARA LA GESTIÓN DE ACTIVIDADES Y SUS EVIDENCIAS EN EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1

En el presente capítulo se describe el proceso de implementación del módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1. Además, se realiza la validación del sistema a través del plan de pruebas. La hipótesis de la investigación es validada a través de diferentes métodos científicos.

En el proceso de trabajo de implementación se desarrolla la arquitectura y el sistema como un todo. Este flujo de trabajo tiene como propósito: definir la organización del código, implementar clases, objetos y subsistemas en términos de componentes y subsistemas de implementación, probar los componentes desarrollados y vincularlos a un sistema funcional. Durante el flujo de trabajo de prueba se verifica que el sistema implementa la verdadera funcionalidad descrita en los casos de uso y que satisface los requisitos del componente, verificando el resultado de la implementación (Suárez Pena, 2016).

III.1 Diagrama de componentes

El diagrama de componentes ilustra la relación que existe entre componentes de software, así como la ubicación de cada uno de ellos dentro del módulo, como se implementan las clases en término de componentes. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponible en el entorno de implementación y en el lenguaje de programación utilizado. Además, muestra las dependencias entre componentes (González Rodríguez, 2017).

A continuación, se pueden observar dos (2) diagramas de componentes correspondiente al módulo para la gestión de actividades y sus evidencias en el STE de la Facultad 1, elaborado y estructurado de acuerdo con el Modelo-Vista-Controlador utilizado por el Framework de desarrollo de PHP llamado Laravel.

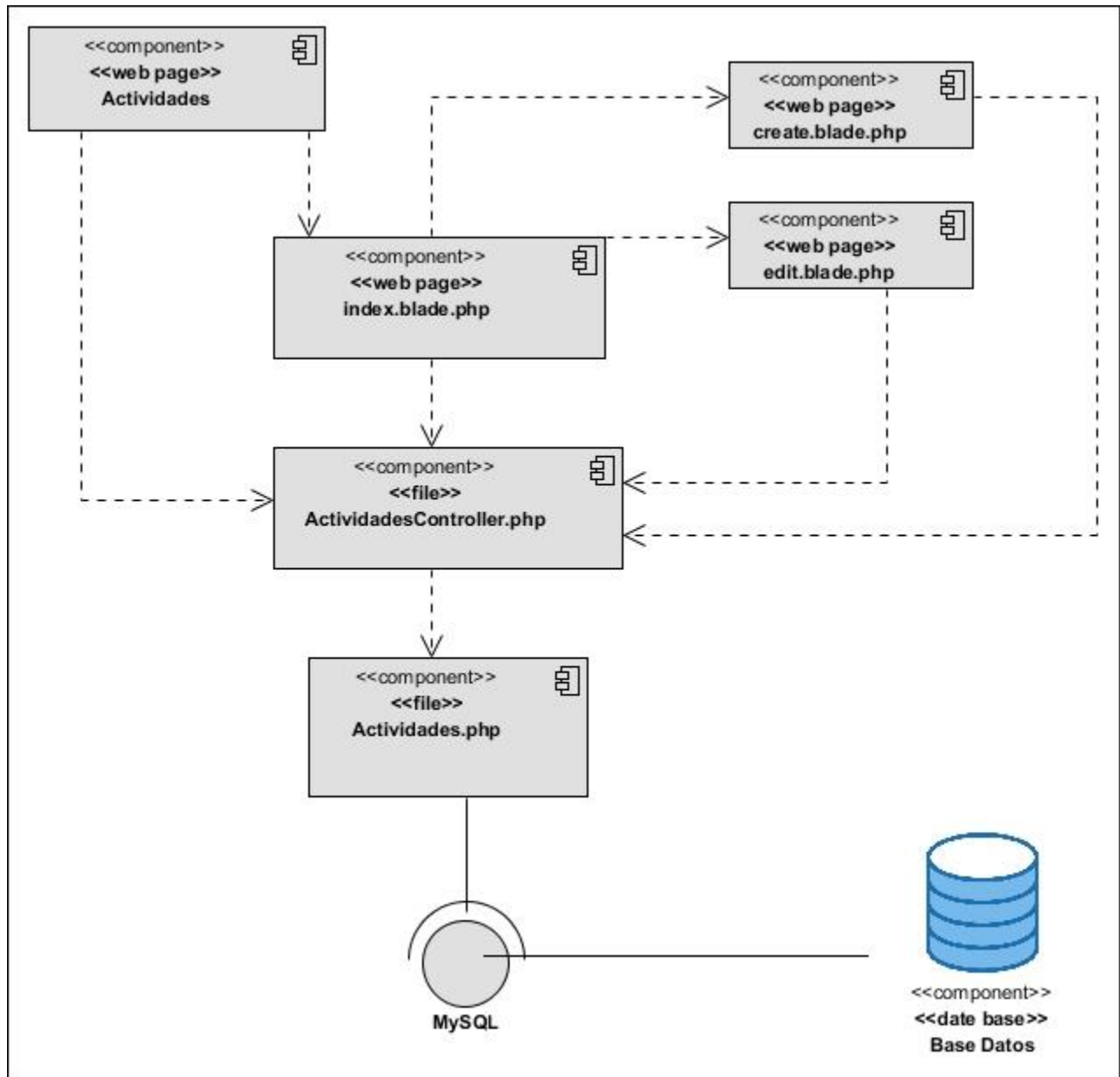


Figura 10. Diagrama de componentes de la gestión de actividades

A continuación, se describen los elementos que componen el diagrama:

Tabla 8: Descripción del diagrama de componentes (Gestionar Actividad)

Componentes		Descripción
Modelo	Actividades.php	Clase entidad que almacena las Actividades en el sistema.
Vistas	Actividades	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las Actividades.
	index.blade.php	Este componente se encarga de mostrar un listado de las Actividades insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar las Actividades.
	create.blade.php	Este componente se encarga de registrar una nueva actividad en la base de datos con los datos necesarios de la misma.
	edit.blade.php	Este componente se encarga de editar una Actividad en la base de datos con los datos necesarios de la misma.
Controlador	ActividadesController.php	Este componente se encarga de realizar todos los métodos relacionados a la gestión de los Estudiantes en el módulo.

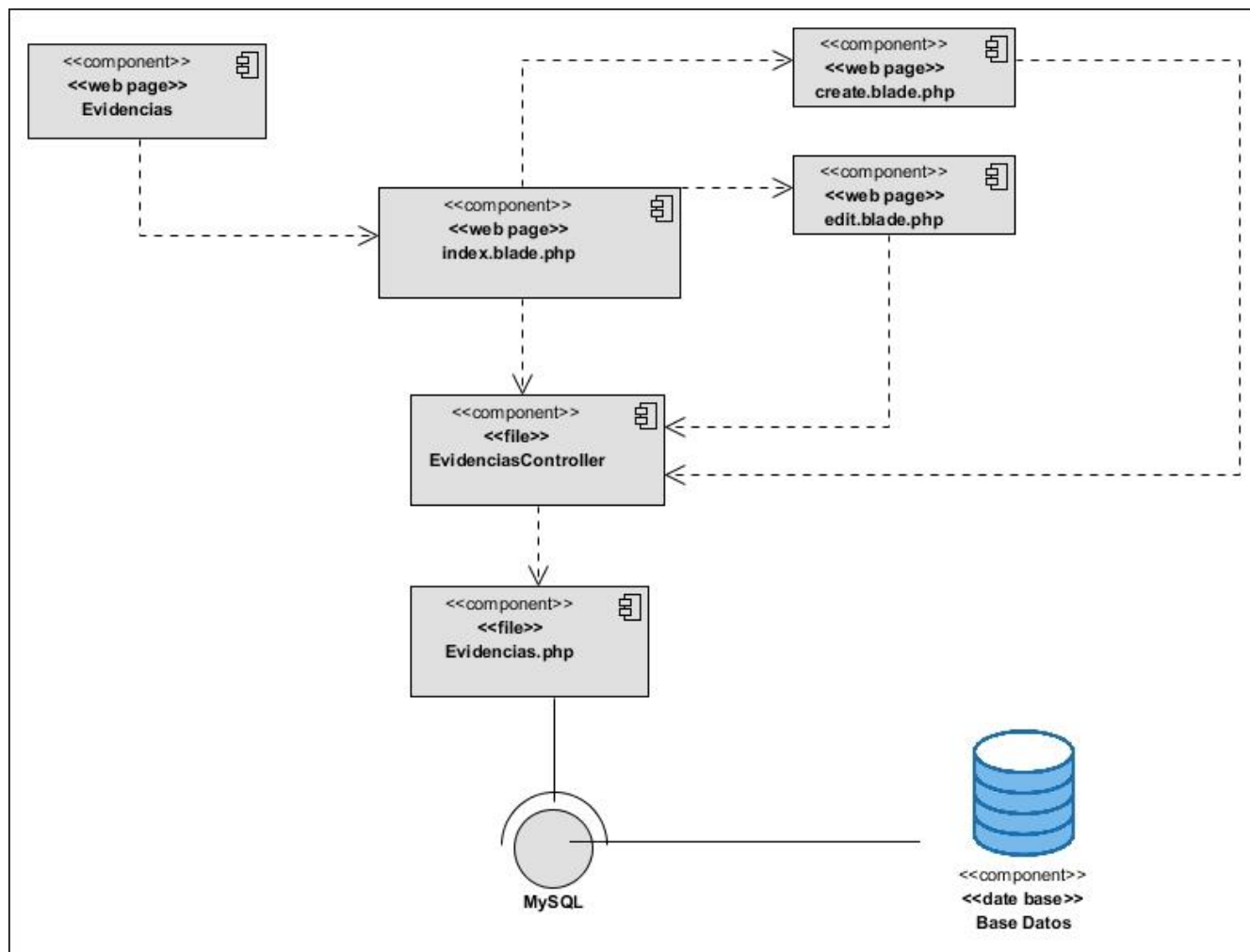


Figura 11. Diagrama de componentes de gestionar evidencias

A continuación, se describen los elementos del diagrama de componentes anteriormente mostrado:

Tabla 9: Descripción del diagrama de componentes (Gestionar Evidencia)

Componentes		Descripción
Modelo	Evidencias.php	Clase entidad que almacena las Evidencias en el sistema.

Vistas	Evidencias	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las Evidencias.
	index.blade.php	Este componente se encarga de mostrar un listado de las Evidencias insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar las Evidencias.
	create.blade.php	Este componente se encarga de registrar una nueva evidencia en la base de datos con los datos necesarios de la misma.
	edit.blade.php	Este componente se encarga de editar una Evidencia en la base de datos con los datos necesarios de la misma.
Controlador	EvidenciasController.php	Este componente se encarga de realizar todos los métodos relacionados a la gestión de las Evidencias en el módulo.

III.2 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento (Guilarte Domínguez, 2019).

Nomenclatura de las clases:

- Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se emplea la notación PascalCasing¹, lo que posibilita que con solo leerlo se entienda el propósito de la misma. Ejemplo: Actividades, en este caso el nombre de la clase está representado por un solo nombre y comienza con la letra mayúscula.

```
class Actividades extends Model
```

Nomenclatura según el tipo de clases:

- Clases controladoras: En estas clases después del nombre le sigue la palabra *Controller*. Ejemplo: *ActividadesController*.

```
class ActividadesController extends Controller
```

- Clases en los modelos: En estas clases que se encuentran dentro del negocio se representan normalmente. Ejemplo: *Evidencias*.

Estándar para comentarios:

- Los comentarios deben ser oraciones completas.
- Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.
- Si un comentario es corto el punto final puede omitirse.
- Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código JavaScript y deben ubicarse en la misma línea que se desea comentar.
- Los comentarios de varias líneas para organización del código aparecerán dentro de los caracteres “/*...*/” en caso de código JavaScript.

```
/*-----Rutas del Modulo Actividades-----*/
```

¹ Esta especifica que los indicadores y nombres de variables, métodos y funciones que están compuestos por múltiples palabras juntas, deben iniciar cada palabra con letra mayúscula.

Estándares para espacios en blanco en expresiones y sentencias:

- Evitar utilizar espacios en blanco en las siguientes situaciones:
 - Inmediatamente dentro de paréntesis, corchetes y llaves.
 - Inmediatamente antes de una coma, un punto y coma o dos puntos.
 - Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función.
- Deben rodearse con exactamente un espacio los siguientes operadores binarios:
 - Asignación (==).

```
@if($evidencia->actividades_id == $actividad->id )
```

- Asignación de aumentación (+=, -=, etc.).
 - Comparación (==, <, >, <=, >=, <>, in, not in, is, is not).
 - Expresiones lógicas (and, or, not).
- Añadir un espacio después de cada delimitador coma ', '.

Otros estándares:

- En los arreglos multilínea, añade una coma al final de cada elemento.
- Añade un salto de línea antes de una sentencia *return*, a menos que el *return* se encuentre solo en un bloque de sentencias, y un salto después de cada llave cierre de sentencia, excepto después de la llave de cierre de clase.

```
$role = Role::all();
$actividades = Actividades::findOrFail($id);

$actividades->delete();

return redirect()->route('actividades.index', compact('role'))->with('info', 'eliminar-actividad');
```

- Utiliza constantes de tipo PHP nativas en minúsculas: *false*, *true* y *null*. Lo mismo aplica para los array().

```
{!! Form::label('imagen', 'Archivo:') !!}  
{!! Form::file('imagen', null, ['class' => 'form-control', 'id' => 'imagen']) !!}
```

- Utilizar los *namespace* para todas las clases excepto en los *Controller*.

```
namespace App\Http\Controllers\Modulo_Actividades;
```

III.3 Validación de la propuesta de solución

La validación del sistema incluye un conjunto de actividades para asegurar que el software desarrollado se corresponda con los requisitos del cliente. Durante la etapa de implementación, pueden ejecutar algunos errores y pueden pasarse por alto algunos elementos que son importantes para el educado funcionamiento del sistema. Por tal motivo, es esencial llevar a cabo la fase de validación, en la cual, a través de varios tipos y métodos de pruebas de software se logra evaluar la calidad y de manera más pragmática, descubrir errores en el sistema. A continuación, se describen las pruebas realizadas al módulo para la gestión de actividades y sus evidencias en el sistema de trabajo educativo de la facultad 1, así como los resultados obtenidos (González Rodríguez, 2017).

Tabla 10. Estrategia de pruebas

Tipo de prueba	Método (técnica) de prueba	Validación
Funcional	Casos de prueba (Caja Negra)	Valida las funcionalidades diseñadas para el módulo.
Seguridad	Software Acunetix	Valida la confidencialidad, integridad y disponibilidad de los datos en el módulo.
Carga y estrés	Software Apache JMeter	Valida el comportamiento del módulo con distintos niveles de usuarios concurrentes y el consumo excesivo de sus recursos.
Integración	Integración de módulos	Consiste en la comprobación de que los elementos del software que interactúan entre sí, funcionan de manera correcta.

III.3.1 Pruebas funcionales

Se denominan pruebas funcionales, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, pues los probadores o analistas de pruebas, no enfocan su atención a cómo se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas (González Herrera, 2017).

Con el propósito de realizar este tipo de pruebas al módulo para la gestión de actividades y evidencias para el STE de la Facultad 1, se diseñó un conjunto de casos de pruebas, referentes a varios casos de uso del negocio del capítulo anterior, pertenecientes a requisitos funcionales de prioridad alta y media, también especificados en dicho capítulo. A continuación, se muestra uno (2) de los casos de prueba mencionados, analizando de la misma varios campos de la tabla para llevar a cabo dicha prueba. En las celdas de la tabla del caso de prueba se pueden encontrar los valores **V**, para datos válidos, **I**, para datos inválidos, y **N/A**, para datos a los que no es necesario proporcionarles un valor.

Tabla 11: Descripción de las variables del caso de prueba 1 basado en el CU-Gestionar Actividad

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	NO	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños.
2	Tipo de Actividad	Campo de texto	NO	Se debe seleccionar uno de los tipos de actividad.
3	Año	Campo de numero	NO	Se debe escribir un número no mayor de 4.

Tabla 12: Caso de prueba del RF1-Crear Actividad

Caso de prueba 1: RF1_Crear Actividad						
Condición de ejecución: el usuario debe tener el permiso de profesor principal						
Escenario	Descripción	1	2	3	Respuesta del sistema	Flujo central
EC 1.1 Registrar actividad de forma correcta	Interfaz con el formulario para llenar los datos de la actividad, si todos son correctos, se registra la actividad en el módulo	V	V	V	Agrega los datos de la actividad y muestra un mensaje de notificación	1- Seleccionar en Actividades la opción de insertar. 2- Rellenar todos los datos del formulario. 3- Seleccionar la opción Insertar
		JICI	Investigativa	2021		
EC 1.2 Registrar actividad con campos vacíos	Interfaz con el formulario para llenar los datos de la actividad, si existe algún campo vacío que no sea requerido, se muestra un mensaje debajo de cada campo señalando	I	I	V	Comprueba si todos los campos están vacíos, si lo están, muestra un mensaje debajo de cada campo señalando que el campo es requerido y que debe de llenarlo	
				2021		

	que el campo es requerido					
EC 1.3	Interfaz con el formulario para llenar los datos de la actividad, si existe algún campo erróneo, se muestra un mensaje debajo de cada campo señalando que el campo tiene un error	V	V	I	Comprueba si los campos tienen errores, si tienen errores muestra un mensaje debajo de los campos diciendo que los campos tienen un error y tienen que ser rellenados correctamente	
Registrar actividad con campos erróneos		JICI	Investigativa	202io		

Tabla 13: Descripción de las variables del caso de prueba 2 basado en el CU-Gestionar Evidencia

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Descripción	Campo de texto	NO	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños.
2	Actividad	Campo de texto	NO	Se debe seleccionar la actividad a la que pertenece.
3	Año	Campo de numero	NO	Se debe seleccionar el año de la actividad a la cual pertenece.

4	Archivo	Campo de tipo file	NO	Seleccionar un archivo de constancia de la participación a la actividad.
---	---------	--------------------	----	--

Tabla 14: Caso de prueba del RF4-Crear Evidencia

Caso de prueba 1: RF4_Crear Evidencia							
Condición de ejecución: el usuario debe tener el permiso de profesor principal o estudiante							
Escenario	Descripción	1	2	3	4	Respuesta del sistema	Flujo central
EC 1.1 Registrar evidencia de forma correcta	Interfaz con el formulario para llenar los datos de la evidencia, si todos son correctos, se registra la actividad en el módulo	V	V	V	V	Agrega los datos de la evidencia y muestra un mensaje de notificación	1- Seleccionar en evidencias la opción de insertar. 2- Rellenar todos los datos del formulario. 3- Seleccionar la opción Insertar
		Diplom a de las JICI	JICI	2021	Diploma.png		

<p>EC 1.2 Registrar evidencia con campos vacíos</p>	<p>Interfaz con el formulario para llenar los datos de la evidencia, si existe algún campo vacío que no sea requerido, se muestra un mensaje debajo de cada campo señalando que el campo es requerido</p>	<p>I</p>	<p>I</p>	<p>V</p>	<p>V</p>	<p>Comprueba si todos los campos están vacíos, si lo están, muestra un mensaje debajo de cada campo señalando que el campo es requerido y que debe de llenarlo</p>	
<p>EC 1.3 Registrar evidencia con campos erróneos</p>	<p>Interfaz con el formulario para llenar los datos de la evidencia, si existe algún campo erróneo, se muestra un mensaje</p>	<p>V</p>	<p>V</p>	<p>I</p>	<p>V</p>	<p>Comprueba si los campos tienen errores, si tienen errores muestra un mensaje debajo de los campos</p>	
				<p>2021</p>	<p>Diploma.png</p>		
		<p>Diplom a de la JICI</p>	<p>JICI</p>	<p>202p</p>	<p>Diploma.pn g</p>		

	debajo de cada campo señalando que el campo tiene un error					diciendo que los campos tienen un error y tienen que ser rellenados correctamente	
--	--	--	--	--	--	---	--

Las pruebas funcionales se realizaron en cinco (5) iteraciones a medida que avanzaba el módulo, las cuales guiaron la calidad del mismo, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando con el ciclo de desarrollo del mismo.

Como resultado final de las pruebas funcionales, se obtuvo, en una primera iteración, un total de dieciséis (16) no conformidades, divididas en nueve (9) de ortografía, tres (3) de redacción, dos (2) de funcionalidad y dos (2) de validación. De estas, se resolvieron doce (12), y cuatro (4) quedaron pendientes. En una segunda iteración, no se identifican nuevas no conformidades y de las cuatro (4) pendientes, solo dos (2) se mantuvieron para la próxima iteración. En una tercera iteración, no se identifican nuevas no conformidades y de las dos (2) pendientes, solo una (1) pasó para la siguiente iteración, donde fue resuelta y no se detectan nuevas no conformidades. En una cuarta iteración no se identifican nuevas inconformidades, obteniendo, de esta manera, resultados satisfactorios. La figura 12, muestra los resultados antes descritos:

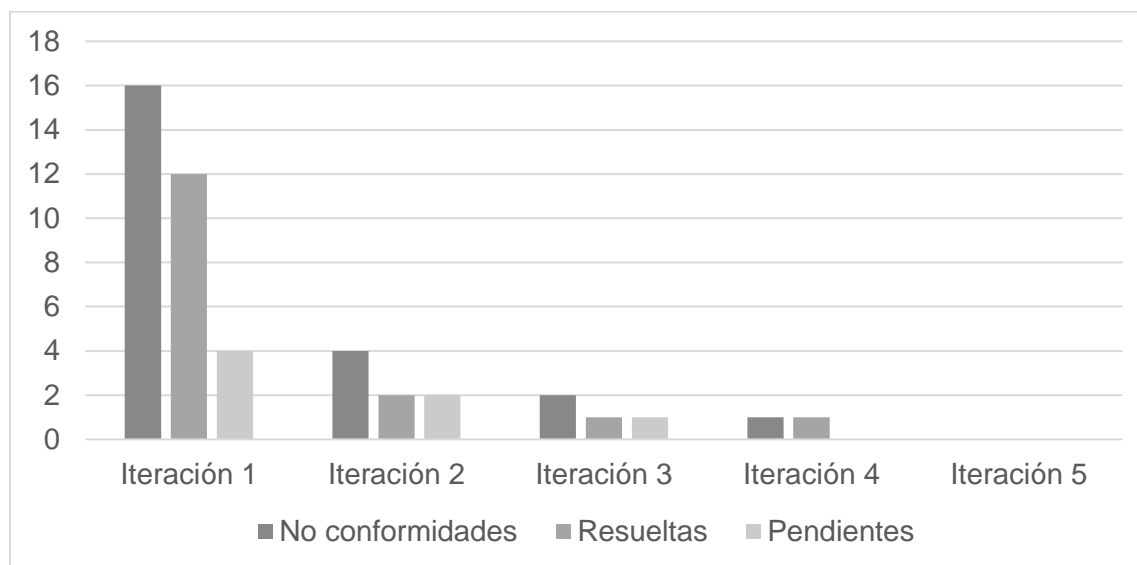


Figura 12. Resultados pruebas funcionales

Las no conformidades de funcionalidad estuvieron relacionadas con registrar datos de una actividad en el módulo y consultar sus datos para visualizar dicha información. En el primer caso, el módulo permitía registrar los datos sin que se le asignara un año a la actividad, el mismo fue resuelto al permitir que el campo año, el cual estaba definido como campo que pudiera tomar valor nulo, fuese requerido. En el segundo caso se intentaba insertar una evidencia, a la cual se le habían asignado datos, pero al presionar el botón no la insertaba en la base de datos, esto se debía a que en el método *store*, encargado de crear y validar los datos de la evidencia no tenía la función *create*, esto se resolvió añadiendo en el controlador, en el método *store* la función *create*.

Las no conformidades de validación fueron detectadas en el formulario para registrar una actividad, donde este permitía introducir caracteres alfanuméricos en el campo año. La solución para los caracteres alfanuméricos fue validar mediante una expresión regular que permite solo el uso de caracteres numéricos. También fueron detectadas en el registrar actividad que el campo nombre permitía registrar números, lo cual se resolvió desde el controlador utilizando la función *validated*, especificando el mismo que dicho campo es *text*, resolviendo de esta forma esta no conformidad.

III.3.2 Pruebas de seguridad

Las pruebas de seguridad permiten realizar una evaluación de los sistemas desde el punto de vista externo y sin conocimiento previo de este. Tienen como objetivo hacer un análisis con el fin de encontrar fallos de seguridad tanto en el diseño como en la implementación de la aplicación. Además, buscan medir la confidencialidad, integridad y disponibilidad de los datos, partiendo de la identificación de amenazas y riesgos en el uso de interfaces de usuarios final. Una vez terminadas las pruebas es posible medir y cuantificar los riesgos a los cuales se ven expuestos aplicativos de la infraestructura interna y externa (Guilarte Domínguez, 2019).

Con el objetivo de realizar una exitosa prueba de seguridad de la propuesta de solución desarrollada, se tuvo en cuenta la herramienta **Acunetix Web Vulnerability Scanner**, herramienta la cual fue descrita con anterioridad en el epígrafe 1.3.2. Durante la primera iteración realizada, la aplicación detectó un total de 14 no conformidades, divididas en cuatro secciones principalmente: nivel alto (0 no conformidades), nivel medio (8 no conformidades), nivel bajo (3 no conformidades) y de carácter informal (3 no conformidades). De las no conformidades de nivel medio, la minoría de ellas son debido al uso de protocolos no seguros para el envío de datos, y el resto se deben al envío de mensajes de error utilizados por Laravel. Las no conformidades de nivel bajo, están relacionadas generalmente a los posibles ataques a la página de actividades y las posibles redirecciones que pueden surgir a partir de la misma una vez que el usuario está dentro del sistema. De tipo Informativo se obtuvo varios errores, relacionados a los campos correspondientes a las descripciones de las evidencias y los campos seleccionar actividad.

Todas las no conformidades o deficiencias encontrados en la primera iteración de la aplicación fueron corregidas, logrando así que en la segunda iteración de la aplicación no se obtuviera ninguna deficiencia en la seguridad del sistema, permitiendo esto poder utilizar la herramienta desarrollada como propuesta de solución de manera segura. A continuación, en la siguiente figura se pueden observar todos los resultados antes descritos:

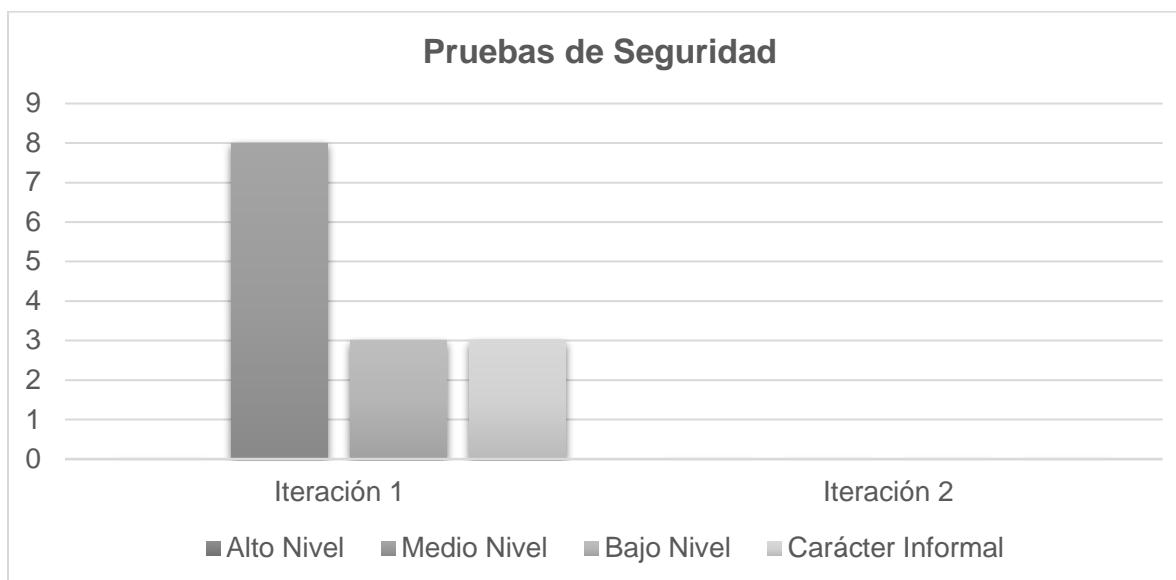


Figura 13. Resultados las pruebas de seguridad

III.3.3 Pruebas de carga y estrés

La prueba de carga y estrés se refiere, generalmente, a la práctica de comprobar el comportamiento de una aplicación mediante cargas o entradas pesadas. Las mismas se realizan con el fin de verificar si el sistema satisface los requisitos de rendimiento para situaciones críticas como pueden ser: la cantidad límite de usuarios accediendo de forma concurrente a los servicios brindados, documentos extremadamente grandes, cantidad de transacciones que se pueden procesar de forma concurrente cada minuto, tiempo de respuesta, entre otros (Guilarte Domínguez, 2019).

Para la realización de esta prueba se utilizó la herramienta **Apache JMeter**, descrita en el epígrafe 1.3.2. Las pruebas se realizaron desde:

Hardware de prueba (PC servidor):

- Sistema Operativo: Windows v10
- Microprocesador: Intel(R) Core (TM) i3-4310u CPU @2.60Ghz
- Memoria RAM: 8.0 GB
- Disco Duro: 500 GB

Hardware de prueba (PC cliente):

- Sistema Operativo: Windows v10
- Microprocesador: Intel(R) Core (TM) i3-4310u CPU @2.60Ghz
- Memoria RAM: 8.0 GB
- Disco Duro: 500 GB
- Tipo de Sistema: Sistema operativo de 64 bits, procesador x64

A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

Muestra: Cantidad de peticiones realizadas para cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min: Tiempo mínimo que demora un hilo en acceder a una página.

Max: Tiempo máximo que demora un hilo en acceder a una página.

Línea 90 %: Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Como se muestra en la tabla 15, se simularon las peticiones realizadas al módulo por un total de 1 y 400 usuarios simultáneamente en cada caso, los cuales realizan hasta 5 peticiones por segundo. Se obtuvieron los siguientes resultados:

Tabla 15: Prueba de carga y estrés

Usuarios	Muestras	Media	Mediana	Min	Max	Línea 90%	%Error	Rend	Kb/s
100	500	1633	1120	96	4501	2802	0.00%	45.6	108.09
500	2500	1262	1084	121	4992	2091	0.50%	226.5	740.58

Las pruebas realizadas muestran que el módulo es capaz de responder a 500 peticiones de 100 usuarios conectados simultáneamente en un tiempo promedio de 1633 milisegundos (1.6 segundos aproximadamente) con 0 % de error, esto evidencia que el módulo puede procesar la carga esperada.

Por otra parte, se realizaron 2500 peticiones iniciadas por 500 usuarios y en este caso el módulo respondió en 1262 milisegundos (1.2 segundos aproximadamente) como tiempo promedio. Esto demuestra que el módulo puede procesar la carga esperada, aunque no fue capaz de responder correctamente el 0.50% de las peticiones realizadas.

III.3.4 Prueba de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y, además, transfieren los datos adecuados en el tiempo preciso y por las vías de comunicación establecidas (González Rodríguez, 2017).

En el caso de la solución desarrollada, para la verificación de una correcta interoperabilidad entre los módulos desarrollados para el Sistema de Trabajo Educativo de la Facultad 1, se llevaron a cabo las acciones siguientes:

- Integración del Módulo de gestión de perfil de usuario para el Sistema de Trabajo Educativo de la Facultad 1.
- Integración del Módulo para la gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la facultad 1.

Se muestra, en la figura 14, los resultados de la ejecución de las dos iteraciones de las pruebas de integración realizadas al sistema:

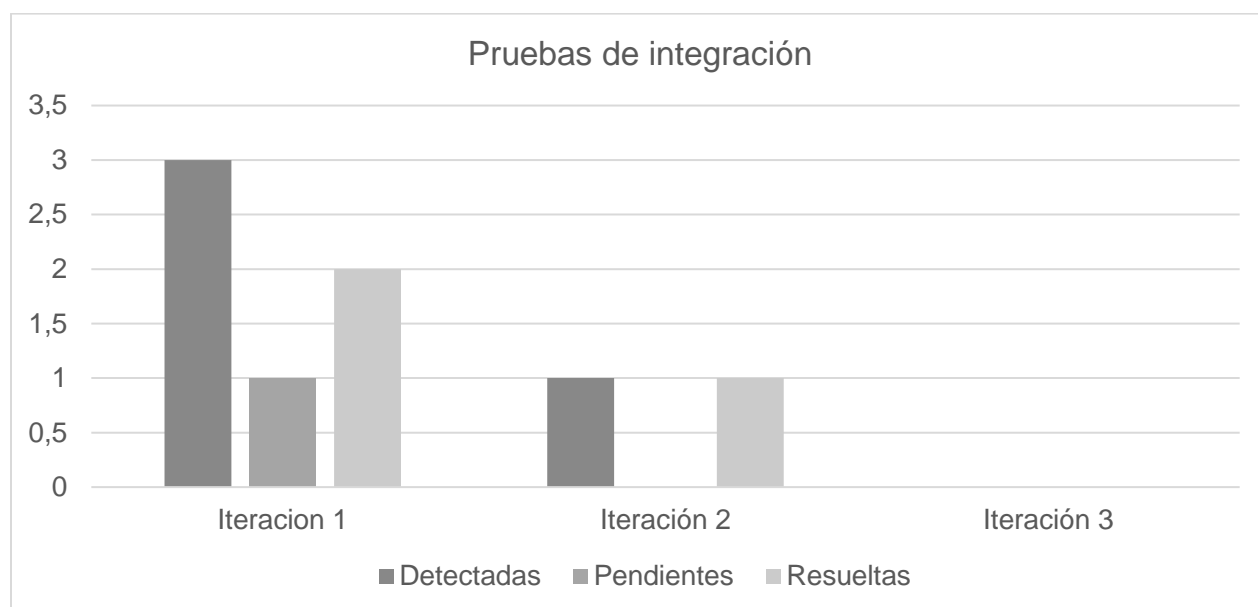


Figura 14. Pruebas de Integración

Mediante la ejecución de las pruebas de integración se logró verificar la operación conjunta del Módulo para la gestión de actividades y sus evidencias para el sistema de trabajo educativo de la Facultad 1 con el Sistema de Trabajo Educativo de la Facultad 1, realizando tres (3) interacciones del sistema en total integración entre estos y comprobando el correcto funcionamiento de los mismos. Durante una primera iteración se encontraron tres no conformidades, de estas dos eran con relación a los roles y permisos para acceder al sistema y una debido a la relación Uno-Mucho de las tablas estudiantes y evidencias. En una segunda iteración se solucionaron dos no conformidades quedando por resolver el problema de la relación Uno-Mucho de las tablas estudiantes y evidencias. En una tercera iteración todas las no conformidades fueron solucionadas logrando una correcta integración entre los componentes internos del sistema.

III.4 Validación de los resultados de la investigación

Desde un principio se determinó como hipótesis de trabajo en esta investigación que el módulo para la gestión de las actividades y sus evidencias contribuye a mejorar el trabajo educativo de la Facultad 1. Partiendo de la premisa de poder gestionar directamente las actividades y las evidencias de los estudiantes que ratifican la participación en esta para luego poder conformar el expediente del

estudiante. El autor del trabajo se decanta por seguir el método de criterios de expertos sometiendo el trabajo al juicio de personas con experiencias en la temática para eliminar aspectos irrelevantes y/o modificar aquellos que lo requieran. Además, se utiliza el método de ladov.

III.4.1 Selección de expertos

Esta es una técnica grupal de análisis de opinión, este consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se ejecutan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. Por lo tanto, la capacidad de predicción de la Delphi se basa en la utilización sistemática de un juicio intuitivo emitido por un grupo de expertos que, con la ayuda de los cuestionarios sucesivos, se obtienen opiniones y eventuales consensos (Jorge, 2019).

Para la selección de los expertos se elaboró un listado preliminar de 14 profesores o personas de la UCI relacionados con el campo; partiendo de que a mayor cantidad de expertos se posea, menor es el error que se incurre en la calidad o resultado de la evaluación. A continuación, se pasó a determinar el coeficiente de competencia de los candidatos a expertos (K) y para ello se tuvo en cuenta:

- Campo de trabajo actual.
- Tiempo que llevan realizando la caracterización de los estudiantes.
- Tiempo que llevan en la dirección y organización de grupos de estudiantes.

De los 14 profesores 13 quedaron con un coeficiente entre 1 y 0.7 aproximadamente (Alto y medio) y se decidió dejar solo a estos como candidatos finales (Ver Anexo 7). Luego de comunicarse con ellos explicando el proceso solo 9 dieron su consentimiento a participar y respondieron el cuestionario (Ver Anexo 5).

III.5.2 Procesamiento de los resultados

Al tabular los resultados se obtiene una tabla con la puntuación dada por cada experto a cada uno de los indicadores. El siguiente gráfico muestra la media de punto otorgado a cada indicador. La tabla completa se encuentra en el anexo 6.

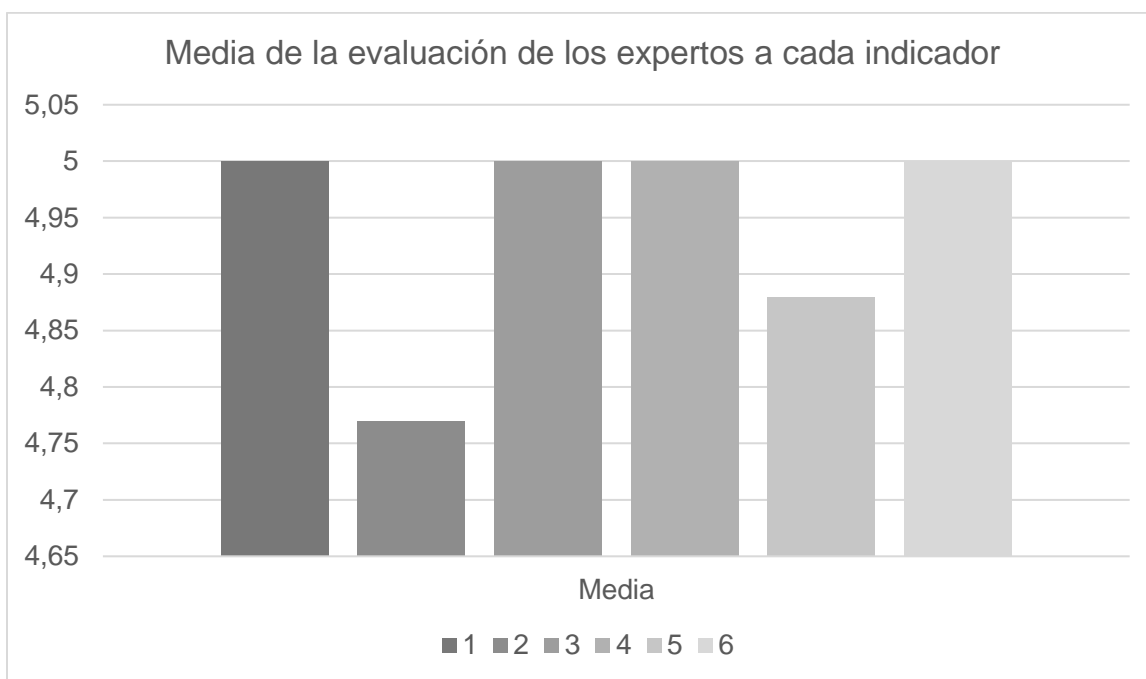


Figura 15. Media de la evaluación de los expertos a los indicadores para validar la propuesta de solución

Como se puede apreciar todos los aspectos fueron evaluados de más de 4 puntos destacándose que solo pueden acceder al módulo los usuarios autorizados, la información se encuentra centralizada y la satisfacción que siente el usuario con el uso del módulo. También todos valoran de forma positiva los demás indicadores que se evaluaron para validar la propuesta de solución.

El aspecto con menor puntuación está referido a la interfaz gráfica, debido al uso incorrecto de íconos y colores en los botones de eliminar y editar. El otro aspecto valorado por debajo de cinco puntos es el tiempo de consulta de la información, debido que a la hora de mostrar los reportes el módulo tarda un aproximado de 5 segundos en responder.

Por último, hay una coincidencia absoluta en que la propuesta ofrece una solución completa al problema de investigación planteado y en que los objetivos de esta investigación se pueden considerar totalmente cumplidos.

III.4.3 Método de ladov

Se constituye en una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario (Verdecia, 2021).

Con el fin de realizar la validación científica se utilizará la técnica de ladov antes descrita. Para llevar a cabo dicha técnica se realizó una encuesta a los profesores jefes de años de la facultad 1 de la universidad de las ciencias informáticas, la cual cuenta de 3 preguntas cerradas y 2 abiertas (Ver anexo 8).

Tabla 16: Cuadro lógico de ladov

3- ¿Se siente satisfecho con el trabajo realizado y los resultados obtenidos en la propuesta de solución?	1- ¿Consideran necesario un módulo para la gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1?								
	Sí			No sé			No		
	2- ¿Considera usted necesario agregarle alguna otra funcionalidad al módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da la mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	3	4
No me gusta nada	6	6	6	6	4	4	3	3	4
No sé decir	2	3	6	3	3	3	6	6	4

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma: Máxima satisfacción (+ 1), Satisfecho (0.5), No definido (0), Insatisfecho (- 0.5) y

Máxima insatisfacción (- 1). A partir de la cantidad de respuestas es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{A(+1) + B(0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

En esta fórmula A, B, C, D y E representan la cantidad de sujetos que interactúan con la aplicación y N representa el total de sujetos. La escala de valores del índice grupal que se toma al aplicar la técnica se encuentra entre:

- 1 y -0.5 Insatisfacción.
- -0.49 y 0.49 Contradicción.
- 0.5 y 1 Satisfacción

Los resultados obtenidos fueron los siguientes:

Tabla 17: Resultados de las escalas de satisfacción

Total, de usuarios(N)	5	Escala
Máxima satisfacción	2	A
Satisfecho	2	B
No definido	1	C
Insatisfecho	0	D
Máxima insatisfacción	0	E

Como resultado del cálculo del ISG, se obtuvo:

$$ISG = \frac{2(+1) + 2(0.5) + 1(0) + 0(-0.5) + 0(-1)}{5} = 0.6$$

La técnica de ladov permitió conocer el índice de satisfacción grupal con respecto al módulo, obteniéndose un resultado de 0.6, valor que se encuentra en el rango de máxima satisfacción definido por la técnica 0.5 a 1. Esto evidencia la aceptación de la propuesta por parte de los encuestados.

Conclusiones del capítulo

- La confección de los diagramas de componentes permite visualizar la integración de los componentes que se encuentran en el módulo de gestión de actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1.
- Con la aplicación de los estándares de codificación se facilita la comprensión y la legibilidad del código, proporcionando un código seguro, confiable, comprobable y mantenible.
- La realización de las pruebas de seguridad, funcionalidad y rendimiento, permitió la obtención de un módulo más robusto y seguro. En las pruebas funcionales se detectaron no conformidades que fueron resueltas satisfactoriamente. En las pruebas de rendimiento se pudo constatar que el módulo de gestión es capaz de responder correctamente las peticiones realizadas. En las pruebas de seguridad se detectaron 14 no conformidades de ellas 8 de nivel medio, 3 de nivel bajo y 3 de nivel informacional, todas resueltas satisfactoriamente. La prueba de integración, permitió una correcta integración entre el módulo desarrollado con el Sistema de Trabajo Educativo de la facultad 1, solucionándose en 2 iteraciones las 3 no conformidades encontradas.
- Mediante el método de Delphi para la selección de expertos se logró un resultado positivo del cumplimiento del objetivo de la investigación. El método de ladov arrojó una máxima satisfacción, evidenciando la aceptación de la propuesta de solución. Al concatenar estos dos resultados obtenidos en la aplicación de estos dos métodos, podemos afirmar que el objetivo de la investigación quedó cumplido satisfactoriamente.

CONCLUSIONES FINALES

Una vez realizada la fundamentación teórica que sustentó este trabajo, definidas las características del módulo, efectuada la implementación y validación del mismo, se obtuvieron resultados que le permiten al autor presentar las siguientes conclusiones:

- ✓ El estudio de los sistemas homólogos permitió identificar la necesidad de desarrollar un nuevo módulo para la gestión de actividades y sus evidencias para el sistema de trabajo educativo de la facultad 1.
- ✓ El análisis de herramientas y tecnologías existentes aportó la base tecnológica para el desarrollo de la solución propuesta.
- ✓ La aplicación de pruebas permitió validar el correcto funcionamiento del módulo desarrollado simulando ambientes reales en diferentes entornos de ejecución.
- ✓ El método de Delphi para la selección de expertos y el método de ladov para ver el índice de satisfacción validó el cumplimiento del objetivo de la investigación.
- ✓ El módulo desarrollado contribuirá a la gestión de las actividades y sus evidencias en el Sistema de Trabajo Educativo de la Facultad 1.

Por todo lo expuesto anteriormente se concluye que el objetivo propuesto para el presente trabajo ha sido cumplido satisfactoriamente.

RECOMENDACIONES

Luego de haber concluido con el desarrollo de la primera versión del módulo para la gestión de actividades y sus evidencias en el sistema de trabajo educativo de la facultad 1, se proponen las siguientes recomendaciones para el futuro:

- Desplegar el módulo implementado en el resto de las áreas de trabajo educativo de las facultades de la UCI, luego de un período de prueba en el área de trabajo educativo de la Facultad 1.

REFERENCIAS BIBLIOGRÁFICAS

(2020). Obtenido de <https://robertomiguelz.blogspot.com/2020/12/mvc-vs-spa.html>

Alvia Choez, A. O. (2018). *Desarrollo de página web para la gestión de fichas estudiantiles para la Universidad de Guayaquil Facultad de Arquitectura y Urbanismo Carrera de Arquitectura para la optimización de los procesos*. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/27231>

Amazon. (13 de Septiembre de 2022). *AWS Amazon*. Obtenido de <https://aws.amazon.com/es/what-is/data-modeling/>

Aprende Web. (s.f.). Obtenido de <https://aprende-web.net/php/objetos/ResumenPHP.pdf>

Aristigüi, L. L. (2017). *Sistema experto basado en modelos de rasgos de contacto bilineales para la predicción de propiedades biológicas en proteínas*. Base Datos. Tesis.Universidad de las Ciencias Informáticas.Facultad 6.

Baró, D. R. (2019). *Sistema para la gestión del banco de problemas científicos de la Facultad 1*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/9998>

BARRERA, G. M. (2018). *Estilo arquitectónico para aplicaciones IoT*. Buenos Aires.

Bautista-Villegas, E. (2022). *Metodologías ágiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel*. Obtenido de <https://doi.org/10.55873/rad.v1i1.168>

CASEY, E. (2000). *Digital Evidence and Computer Crime*. Academic Press.

Cobo, A. G. (2005). *PHP y MySQL Tecnologías para el desarrollo de*. España : Diaz Santos.

CONSORTIUM, W. W. (1999). *HTML 4.01. Specification*.

CUJAE. (2022). *Sigenu - Portal Docente*. Obtenido de <https://sigenu.cujae.edu.cu/sigenu-portal/#modues>

Daudinot Hamiltón, Y. (2015). *Perfil de usuario para el Sistema de Planificación de Actividades SIPAC*. Universidad de las Ciencias Informáticas.

Española, R. A. (2021). *Diccionario de la lengua española*. Obtenido de <https://dle.rae.es/evidencia>

- Fragos, D. J. (s.f.). *Gestión basada en evidencias. ¿Una nueva moda administrativa?* Obtenido de https://academia.uat.edu.mx/pariente/Publicaciones%20recientes/Pariente_Cap%C3%ADtulo_05.pdf
- GARCERANT, I. (10 de Julio de 2008). *Modelo de Dominio. Tecnología y Synergix*. Obtenido de <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>
- GÓMEZ BERMEJO, R. (2018). *Aplicación web para la gestión y búsqueda de eventos*.
- González Herrera, C. Y. (2017). *Módulo para el diseño de modelos entidad relación en la plataforma RDB-Learning*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/8071>
- González Rodríguez, A. (2017). *Subsistema de gestión de perfil de usuario para el buscador Orion*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/7963>
- Guilarte Domínguez, C. R. (2019). *Sistema para la gestión de procesos en la Dirección de Extensión Universitaria*. . Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/10152>
- IBM. (11 de Septiembre de 2022). *IBM*. Obtenido de <https://www.ibm.com>
- IEEE. (2011). *Ingeniería de sistemas y software - Descripción de la arquitectura* .
- Inc, L. S. (13 de Septiembre de 2022). *Lucidchart*. Obtenido de <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>
- Jorge, A. L. (2019). *Método de validación por expertos*. Obtenido de <https://doi.org/10.5195/biblios.2014.180>
- Medero Oliva, D. (Junio de 2015). *Módulo Escalafón de integralidad para el Sistema dataFEU*. La Habana: Universidad de las Ciencias Informáticas. Facultad 3.
- MES. (2022). *Resolución 47*.
- Metodología Gestión de Requerimientos*. (s.f.). Obtenido de <https://sites.google.com/site/metodologiareq/>
- Microsoft. (8 de Junio de 2022). Obtenido de <https://code.visualstudio.com/learn/overview>
- Mora, S. L. (2016). *Programación de aplicaciones web*.
- NISKA, C. (2014). *Extending Bootstrap*.

- Pacienza, J. E. (2015). Metodologías de desarrollo de software. En J. E. Pacienza, *Metodologías de desarrollo de software* (pág. 12).
- Peguero, T. Y. (2021). *Tecnologías en educación superior: Necesidad e impronta ante la covid-19 / Technologies in Higher Education: need and print before the COVID -19*. Revista de Educación.
- Pérez, J. E. (2019). Introducción a JavaScript.
- PILICITA GARRIDO, A., & BORJA LÓPEZ, Y. y. (2021). *Rendimiento de MariaDB y PostgreSQL*.
- Pillalaza Quijia, J. F. (2020). *Desarrollo de un sistema web para la gestión estudiantil del colegio particular Pcei Buenas Nuevas*. Quito: Quito, 2020.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*.
- PUIG, J. C. (2013). *CSS3 y Javascript avanzado*. Obtenido de <https://openlibra.com/es/book/download/css3-y-javascript-avanzado>
- RAJME, C. A., & GUERRA, G. A. (s.f.). *Sistema para la gestión de procesos de educación a distancia en las secretarías docentes*.
- Roger S, P. (1995). *Ingeniería de Software*. Mc Graw Hill.
- Santiesteban Catallops, L. J. (2019). *Sistema de gestión de información tecnológica para la Facultad 1 de la Universidad de las Ciencias Informáticas*. Obtenido de <https://repositorio.uci.cu/jspui/handle/123456789/10067>
- Sommerville. (2005). *Ingeniería de Software—lan Sommerville 7a Edicion*. Obtenido de https://www.academia.edu/15059886/Ingenieria_de_Software_lan_Sommerville_7a_Edicion
- Suárez Pena, W. B. (2016). *Componente para la autorización, la administración de perfiles de usuario y la multi-institución en el marco de trabajo Xalix*. Obtenido de <http://repositorio.uci.cu/jspui/handle/ident/9020>
- VENTURA BAUTISTA, J. N. (2021). *JQUERY*.
- Verdecia, A. &. (2021). *Índice de Satisfacción por la Estrategia Didáctica para la utilización de Recursos Educativos Abiertos en la disciplina Historia de Cuba a través de la técnica ladov*. Obtenido de <https://repositorio.uci>
- WIRFS-BROCK, B. A. (2020). *JavaScript: the first 20 years*.

ANEXOS

Anexo 1: Entrevista a especialistas y profesores relacionados con el área de Trabajo Educativo

La presente entrevista persigue realizar el levantamiento de requisitos y entender mejor los principales problemas que presenta el área donde se desarrollará la investigación.

Entrevistado: _____ Cargo: _____ Fecha: _____

1. ¿Qué es una actividad?
2. ¿Cuáles son los indicadores utilizados en la UCI para definir los tipos de actividades?
3. ¿Qué son las evidencias de actividades?
4. ¿Cuáles son las evidencias que puede presentar un estudiante?
5. ¿Por qué surge la necesidad de gestionar las actividades y sus evidencias en el sistema de trabajo educativo de la Facultad 1?
6. ¿Cuál es la estructura de carpetas que se debe mantener en el servidor donde se almacenan las evidencias?

Anexo 2: Diagrama de secuencia <Insertar Evidencia>

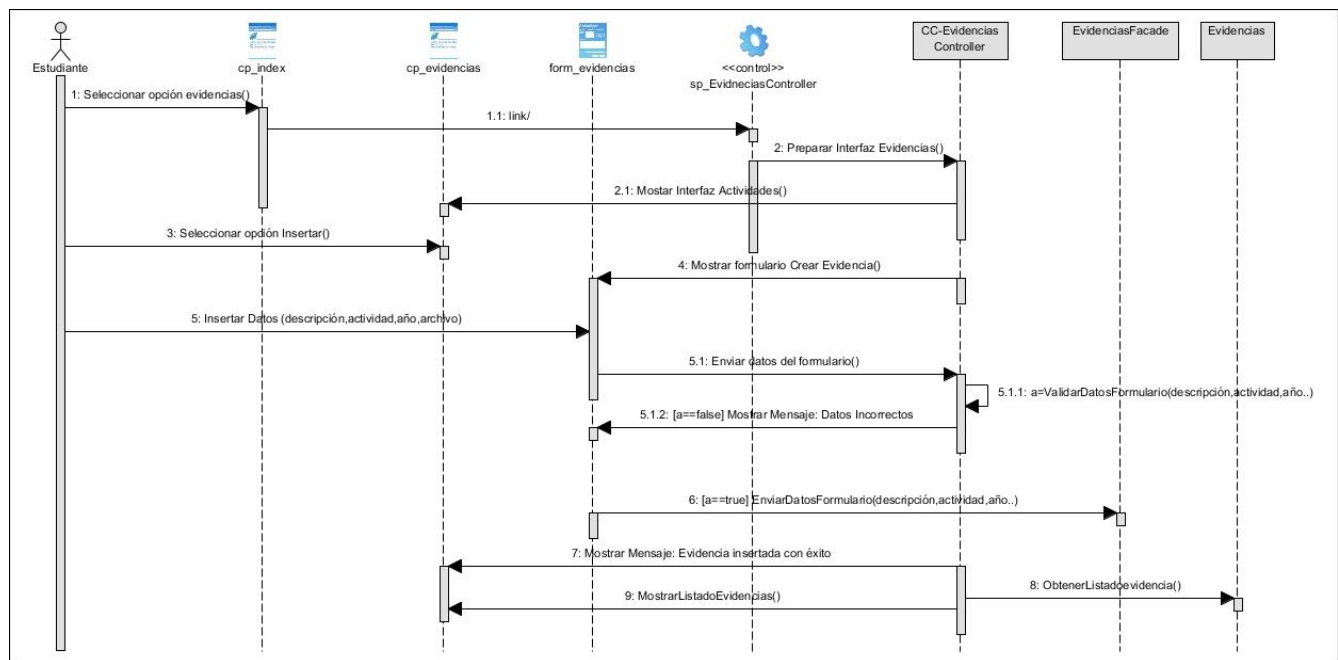


Figura 16. Diagrama de secuencia Insertar Evidencia

Anexo 3: Diagrama de secuencia <Modificar Evidencia>

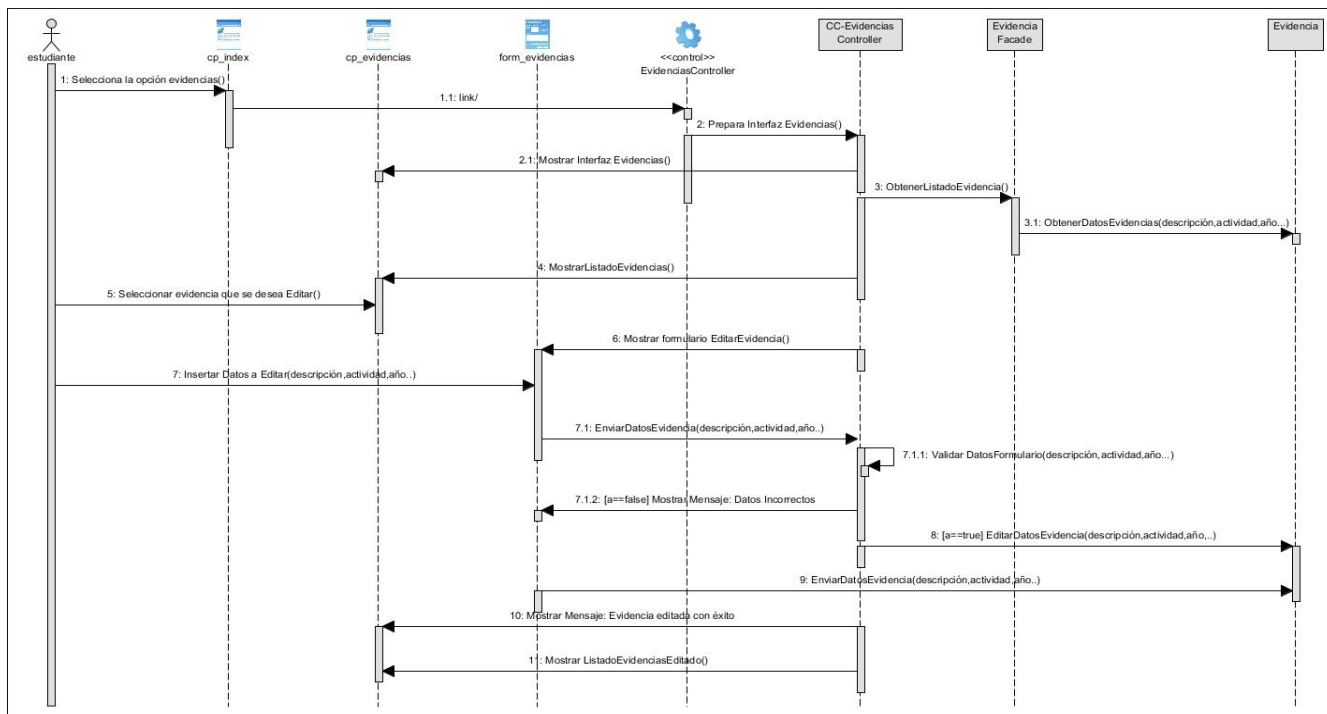


Figura 17. Diagrama de secuencia Editar Evidencia

Anexo 4: Diagrama de secuencia <Eliminar Evidencia>

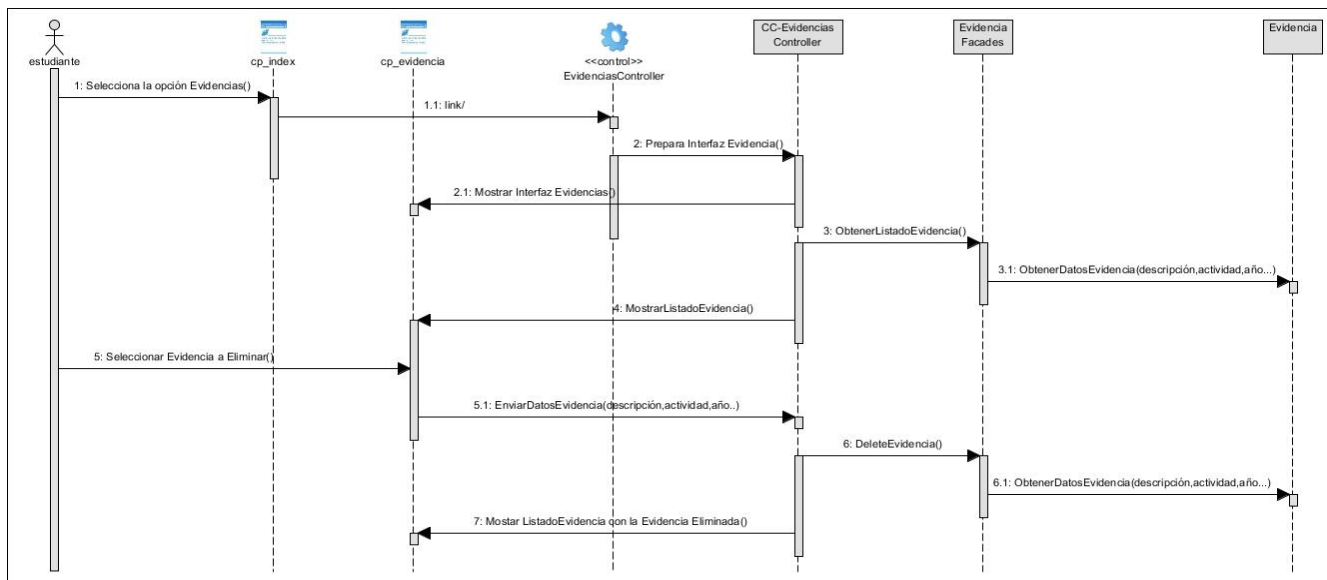


Figura 18. Diagrama de secuencia Eliminar Evidencia

Anexo 5: Modelo encuesta a profesores

Estimado profesor:

Se solicita su valiosa cooperación para la valoración de los resultados del trabajo “Módulo para la gestión de actividades y sus evidencias del Sistema de Trabajo Educativo de la Facultad 1”, el cual constituye el resultado de una investigación dirigida a informatizar el proceso de obtención de evidencias en las actividades que se realizan en la Universidad de las Ciencias Informáticas por parte de los estudiantes de la Facultad 1. Por lo que le pedimos nos ayude a responder el siguiente cuestionario en vistas a evaluar los resultados desde el punto de vista práctico.

Por favor lea detenidamente el informe adjunto y valore los siguientes aspectos en una escala del uno al cinco (donde 1 -Ninguna, 2 - Baja, 3 - Media, 4 - Alta, 5 - Muy alta).

No.	Aspecto a valorar	Evaluación
1	El módulo cumple todas las funcionalidades descritas	
2	Interfaz gráfica del módulo	
3	¿Solo accede a la información el personal autorizado?	
4	¿La información está almacenada de manera centralizada?	
5	Tiempo de consulta de la información	
6	Satisfacción del usuario con el uso del módulo	

Anexo 6: Tabla con los datos resultantes de la encuesta a los expertos

No.	Indicadores/Expertos	1	2	3	4	5	6	7	8	9	MEDIA	MODA
1	El módulo cumple todas las funcionalidades descritas	5	5	5	5	5	5	5	5	5	5	5
2	Interfaz gráfica del módulo	5	5	4	5	4	5	5	5	5	4.77	5
3	¿Solo accede a la información el personal autorizado?	5	5	5	5	5	5	5	5	5	5	5

4	¿La información está almacenada de manera centralizada?	5	5	5	5	5	5	5	5	5	5	5
5	Tiempo de consulta de la información	4	5	5	5	5	5	5	5	5	4.88	5
6	Satisfacción del usuario con el uso del módulo	5	5	5	5	5	5	5	5	5	5	5

Anexo 7: Tabla con los datos del coeficiente de los expertos

$$K = \sum Rd \div Ni \quad \text{Rd: Resultado del indicador}$$

Ni: Número total de indicadores

No. Expertos/ Indicadores	Indicador A	Indicador B	Indicador C	K
1	1	1	1	1
2	1	1	1	1
3	0	1	0	0.33
4	1	1	1	1
5	1	1	1	1
6	1	1	0	0.66 ≈ 0.7
7	1	1	1	1
8	1	1	1	1
9	0	1	1	0.66 ≈ 0.7
10	1	1	0	0.66 ≈ 0.7
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	0	1	0.66 ≈ 0.7

Anexo 8: Encuesta realizada a los Profesores Principales de año encargados de desarrollar el Sistema de Trabajo Educativo de la Facultad 1.

1. ¿Consideran necesario un módulo de gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la Facultad 1?

Sí__ No__ No sé__

2. ¿Considera usted necesario agregarle alguna otra funcionalidad al módulo de gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la Facultad 1?

Sí__ No__ No sé__

3. ¿Se siente satisfecho con el trabajo realizado y los resultados obtenidos en la propuesta de solución?

__ Me gusta mucho

__ Me gusta más de lo que me disgusta

__ Me da lo mismo

__ Me disgusta más de lo que me gusta

__ No me gusta nada

__ No sé decir

4. ¿Qué aspectos usted cree que se puedan mejorar en futuras investigaciones relacionadas con el proceso de gestión de actividades y evidencias en el Sistema de Trabajo Educativo?

5. ¿Considera que algún otro centro pueda hacer uso del módulo de gestión de actividades y sus evidencias para el Sistema de Trabajo Educativo de la Facultad 1?