



Facultad 1

# Módulo de Evaluación integrada del Sistema de gestión de trabajo educativo de la facultad 1

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

**Autor(es):** Stive Hernández Suárez

**Tutor(es):** M. Sc. Disnayle Jorge Chacón

**Co-tutor:** Ing. Leonardo Jiménez Caballero

La Habana, <mes> de <año>

<Año 63 del Triunfo de la Revolución>



## DECLARACIÓN DE AUTORÍA

El (Los) autor(es) del trabajo de diploma con título “**Módulo de Evaluación integrada del Sistema de gestión de trabajo educativo de la facultad 1**” concede(n) a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara(n) como único(s) autores de su contenido. Para que así conste firma(n) la presente a los <día> días del mes de <mes> del año <año>.

< Stive Hernández Suárez>

<nombre del autor>

---

Firma del Autor

---

Firma del Autor

<nombre del tutor>

<nombre del tutor>

---

Firma del Tutor

---

Firma del Tutor

## **DATOS DE CONTACTO**

Disnaye Jorge Chacón: Graduada de Ingeniero en Ciencias informáticas en julio del 2010 en la Universidad de las Ciencias Informáticas (UCI), Profesora del departamento de Informática de la facultad 1. Actualmente se desempeña como profesora principal de 5to año. Ha impartido las asignaturas Proyecto de Investigación y Desarrollo IV, V, VI, VII, Introducción a las Ciencias Informáticas, Introducción a las Ciencias Informáticas I y II, Álgebra Lineal y Matemática I. Máster en Ciencias Matemáticas: Mención Enseñanza de las Matemáticas en la Universidad de la Habana.

## **AGRADECIMIENTOS**

Gracias a todos los profesores que me han impartido clases durante todos estos años, a todos mis compañeros de estudios, a todo el personal de la universidad que de una forma u otra han contribuido a que llegue a este momento. Agradecer a mis tutores, compañeros del tribunal y oponente. ¡Muchas Gracias!

## **DEDICATORIA**

Esta tesis va dedicada a toda mi familia por siempre estar ahí.



## RESUMEN

En la Facultad 1 de la Universidad de Ciencias Informáticas (UCI) se realiza la Evaluación Integral de cada estudiante durante su estancia en la Universidad. Este proceso busca valorar el desempeño de cada alumno en las diferentes esferas de la vida universitaria. Actualmente se realiza de forma manual lo que hace engorroso el control y manejo de esta información, la que se almacena en el mejor de los casos en documento Word, afectándose así la agilidad y calidad del proceso, y que existan duplicados o ausencia de la información, debido a la gestión de grandes volúmenes de información.

Teniendo en cuenta estos aspectos el objetivo general definido para este trabajo es desarrollar un sistema para informatizar la gestión del proceso de evaluación integral en la Facultad 1 de la UCI. Se hace un estudio del estado actual, y se define la propuesta de solución: Módulo para el Sistema de trabajo educativo que permita la gestión de la información en el proceso de Evaluación Integral de los estudiantes de la facultad 1. Como base tecnológica fundamental se usa Laravel 9, PHP 8.0.3 como lenguaje de programación, MySQL 8.0.3 como gestor de base de datos y Apache 2.4.46 como servidor web. La metodología de desarrollo de software empleada es AUP-UCI en el Escenario No.2. La arquitectura de software definida es Modelo-Vista-Controlador. Se muestra, además, el resultado de la estrategia de validación que fue aplicada a la propuesta de solución, que permite obtener un producto eficiente, funcional, seguro y que cumple con sus especificaciones.

## PALABRAS CLAVE

*Evaluación integral, procesos, tesis, sistema de trabajo educativo*

## ABSTRACT

In Faculty 1 of the **Universidad de las Ciencias Informáticas (UCI)** is carried out the integral evaluation of each student during their stay at the University. This process tries to assess their performance in the different spheres of university life. At the current moment is carried out manually, which makes it more cumbersome the handle and control of this information, which is stored in the best of cases in a Word document, thus affecting the agility and quality of the process, and that there are duplicates or absence of information, due to the management of large volumes of information.

Taking these aspects into account, the general objective defined for this work is to develop a system to computerize the management of the integral evaluation process in Faculty 1 of the UCI. A study of the current state is made, and the solution proposal is defined: Module for the Educational Work System that allows the management of information in the process of Integral Evaluation of the students of the faculty 1. As a fundamental technological base Laravel 9 is used, PHP 8.0.3 as programming language, MySQL 8.0.3 as database manager and Apache 2.4.46 as web server. The software development methodology used is AUP-UCI in Scenario No.2. The defined software architecture is Model-View-Controller. It also shows the result of the validation strategy that was applied to the proposed solution, which allows obtaining an efficient, functional, safe product that meets its specifications.

#### **KEYWORDS**

*Integral Evaluation, process, thesis, educational work system*



## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS MÓDULO DE EVALUACIÓN INTEGRADA DEL SISTEMA DE GESTIÓN DE TRABAJO EDUCATIVO DE LA FACULTAD 1 .....	7
I.2 Estudio del estado del arte de los sistemas de gestión del proceso de evaluación integrada .	8
I.3 Herramientas y Tecnologías.....	12
I.3 Metodología de desarrollo de software.....	19
Conclusiones del capítulo.....	21
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO: SISTEMA DE GESTIÓN DEL PROCESO DE EVALUACIÓN INTEGRAL DE LOS ESTUDIANTES DE LA FACULTAD 1. ....	23
II.1 Flujo actual del proceso.....	23
II.2 Modelo conceptual .....	24
II.3 Descripción de las clases del modelo .....	25
II.4 Propuesta de solución .....	25
II.5 Especificación de los requisitos del sistema .....	26
II.6 Modelos de Casos de Usos del Sistema .....	32
II.7 Diagrama de Clases .....	35
II.8 Descripción de la arquitectura de software y los patrones de diseño.....	37
II.9 Diagrama de Modelo de Datos .....	41
II.10 Diagrama de Despliegue .....	43
II.11 Diagrama de Secuencia .....	44
Conclusiones del capítulo.....	47
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA .....	48
III.1 Diagrama de Componentes.....	48
III.2 Estándares de Codificación .....	49
III.2.1 Estándares de Codificación para Laravel:.....	49
III.2.2 Estándares de Codificación para PHP:.....	51

III.3 Verificación del Sistema .....	54
III.3.1 Estrategia de Pruebas .....	54
III.3.2 Pruebas Funcionales .....	55
III.3.3 Pruebas de Rendimiento .....	59
III.3.4 Pruebas de Seguridad .....	61
Conclusiones del capítulo.....	67
CONCLUSIONES FINALES .....	69
RECOMENDACIONES .....	70
REFERENCIAS BIBLIOGRÁFICAS.....	71
ANEXOS.....	76

**ÍNDICE DE TABLAS**

Tabla 1 Comparación de los sistemas homólogos. Fuente: Elaboración Propia.....	10
Tabla 2 Fases de la Variación AUP-UCI. Fuente: Elaboración Propia.....	19
Tabla 3 Requisitos Funcionales. Fuente: Elaboración Propia.....	27
Tabla 4 Actores del Sistema. Fuente: Elaboración Propia.....	32
Tabla 5 Caso de prueba basado en el requisito funcional Crear Comisión. Fuente: Elaboración Propia.....	55
Tabla 6 Aceptación en el caso de prueba Crear Comisión. Fuente: Elaboración Propia.....	56
Tabla 7 Resultado de Pruebas de Carga y Estrés. Fuente: Elaboración Propia.....	60
Tabla 8 Cuadro Lógico de Iadov para usuarios potenciales. Fuente: Elaboración Propia.....	65
Tabla 9 Satisfacción de Usuarios Potenciales. Fuente: Elaboración Propia.....	66
Tabla 10 Escala Numérica para el ISG. Fuente: Elaboración Propia.....	66

## ÍNDICE DE FIGURAS

Ilustración 1 Mapa Conceptual. Fuente: Elaboración Propia.....	24
Ilustración 2 Caso de Uso del Sistema. Fuente: Elaboración Propia. ....	35
Ilustración 3 Diagrama de Clases Grupo. Fuente: Elaboración Propia. ....	36
Ilustración 4 Diagrama de Clases Evaluación. Fuente: Elaboración Propia. ....	37
Ilustración 5 Modelo Vista Controlador. Fuente: Elaboración Propia. ....	39
Ilustración 6 Diagrama de Modelo de Datos Módulo Evaluación Integrada. Fuente: Elaboración Propia.....	42
Ilustración 7 Diagrama Entidad Relación Módulo Evaluación Integrada. Fuente: Elaboración Propia. ....	43
Ilustración 8 Diagrama de Despliegue. Fuente: Elaboración Propia.....	44
Ilustración 9 Diagrama de Secuencia Insertar Evaluación. Fuente: Elaboración Propia. ....	45
Ilustración 10 Diagrama de Secuencia Modificar Evaluación. Fuente: Elaboración Propia.....	46
Ilustración 11 Diagrama de secuencia Eliminar Evaluación. Fuente: Elaboración Propia. ....	47
Ilustración 12 Diagrama de Componentes Grupos. Fuente: Elaboración Propia. ....	48
Ilustración 13 Diagrama de Componentes Evaluación. Fuente: Elaboración Propia. ....	49
Ilustración 14 Resultado de las pruebas funcionales. Fuente: Elaboración Propia. ....	58
Ilustración 15 Resultado de las pruebas de seguridad. Fuente: Elaboración Propia. ....	63
Ilustración 16 Resultado de las Pruebas de Integración.....	64
Ilustración 17 Satisfacción de Usuarios Potenciales. Fuente: Elaboración Propia. ....	67
Ilustración 18 Diagrama de Clases Caracterización .....	76
Ilustración 19 Diagrama de Clases Comisión .....	76
Ilustración 20 Interfaz Crear Grupo .....	77
Ilustración 21 Interfaz Comisión-Evaluaciones.....	77
Ilustración 22 Interfaz Caracterización .....	78

## **OPINIÓN DEL(OS) TUTOR(ES)**

<Contenido de la opinión de los tutores>

## **AVAL DEL CLIENTE**

<Contenido del aval del cliente sobre la solución desarrollada>

## INTRODUCCIÓN

La educación es un derecho primordial y un eslabón clave, como herramienta para cualquier persona, y en el desarrollo del país para alcanzar metas económicas, políticas y sociales propuestas. La educación universitaria ocupa un lugar en la cúspide de esa pirámide educativa, pues permite la formación profesional continua de personas en un cúmulo de diferentes especialidades que brindan posibles soluciones a problemas reales de mundo actual.

La universidad cubana, cuenta con un modelo caracterizado como: humanista, moderna y universal; científica, tecnológica e innovadora; integrada a la sociedad, a la producción, a la localidad; comprometida con la construcción de un socialismo próspero, democrático, soberano y sostenible. Tiene entre sus mayores desafíos el fortalecimiento de una formación integral de los estudiantes, para lograr en ellos una mayor responsabilidad ética, económica, jurídica, social, cultural y ambiental. Se busca que el egresado obtenga, no solo la mejor calificación técnica posible, sino además adquiera cualidades personales y virtudes humanas, para que sea un miembro activo y provechoso de la sociedad.

*“No interesa evaluar los conocimientos del estudiante en términos de saberes, sino de definir campos de competencias más amplios, puestos de manifiesto por el estudiante en situaciones complejas de producción, de resolución de problemas, de investigación, etcétera” (Pacheco Tejeda, 2021).*

Es por ello que las Instituciones de Educación Superior (IES) del presente siglo, orientan sus esfuerzos en promover la formación integral del estudiante desarrollando un pensamiento analítico, crítico, creativo y propositivo.

En Cuba, el sistema educacional es gratuito completamente hasta el nivel superior, y se garantiza el acceso de todas las personas a ese derecho primordial en estado de igualdad. La educación cubana se atiende directamente por el estado con especial interés como lo reafirma su constitución en el artículo 32: El Estado orienta, fomenta y promueve la educación, las ciencias y la cultura en todas sus manifestaciones (Gaceta Oficial de la República de Cuba, 2021).

Dos ministerios: Ministerio de Educación (MINED), que atiende los niveles de enseñanza generales (preescolar, primaria, secundaria, educación técnico profesional y preuniversitaria); y el Ministerio de Educación Superior (MES) que atiende los niveles de enseñanza Universitaria, guían la política educacional. El MES es el organismo encargado de dirigir, proponer, ejecutar y controlar la política del

Estado y el gobierno respecto a la educación superior. Esta responsabilidad estatal la ejecuta a partir de funciones principales que están reflejadas en el Acuerdo No. 4001 del Consejo de Ministros (*goc-2019-02.pdf*, s. f.).

La formación de los profesionales de nivel superior, proceso que se desarrolla en las instituciones de educación superior de todo el país, garantiza la preparación integral de los estudiantes universitarios. Esta se concreta en una sólida formación científico-técnica, humanística y de altos valores ideológicos, políticos, éticos y estéticos. La finalidad es lograr profesionales revolucionarios, cultos, competentes, independientes y creadores, para que puedan desempeñarse exitosamente en los diversos sectores de la economía y de la sociedad en general (Ministerio de Educacion Superior, s. f.).

Un eslabón clave que contribuye a alcanzar ese reto es el proceso de evaluación integral del estudiante. Este proceso ofrece elementos cualitativos y cuantitativos que revelan datos sobre la forma polifacética y armónica de como transcurre su desarrollo y los resultados obtenidos. En la misma medida, le aporta mecanismos para auto reconocer (o que otros reconozcan) tanto las fallas como las potencialidades y oportunidades que lo ayudan a superarlas. Además propone metas, compromisos, tareas y acciones en aras de incrementar su implicación en el proceso de formación (Acevedo et al., 2017).

La Universidad de las Ciencias Informáticas (UCI), como centro docente-productor que forma profesionales altamente calificados y comprometidos en la rama de la informática, también lleva a cabo el proceso de evaluación estudiantil, denominado Proceso de Integralidad para los estudiantes de primero a cuarto año y Proceso de Caracterización para los de quinto. Puede ser descrito como un proceso de profundo análisis e identificación de logros y carencias, de valores, actitudes y potencialidades de futuro.

Se caracteriza de forma acumulativa durante todo el período de formación en el pregrado a partir del control y evaluación de los estudiantes en el ejercicio de sus actividades.

En cada curso escolar el estudiante debe entregar una autoevaluación del período señalado que es tomada como base por la brigada para otorgarle una evaluación. Esta variante a la hora de realizar este proceso trae consigo que la veracidad de esa información se vea comprometida, reflejando poco control de la trayectoria estudiantil ya que las actividades no son registradas en el momento de ocurrencia y dependen de cuan cercano a la realidad se encuentre el resumen realizado por el alumno.



Según datos de los registros de estos procesos en años anteriores, obtenidos en las actas de los Consejos de la Federación Estudiantil Universitaria (FEU), alrededor del 60% de las caracterizaciones de los estudiantes no son entregadas o poseen una calidad deficiente. Lo anterior priva a los jefes de brigada de realizar de forma eficiente su análisis y les obliga a recopilar ellos todas las evidencias necesarias para llevar a cabo esta evaluación, perjudicando los tiempos de envío de la información a organismos superiores y la culminación del proceso (Aguilera Alvarez & Rodríguez Rodríguez, 2020).

Como no existe un estándar en la documentación a almacenar, los formatos en los que cada estudiante envía sus datos son muy diversos: PDF, Microsoft Office u Open Office/Libre Office; y se hacen engorrosos los pasos de unificado y procesamiento de la información. Tampoco se cuenta con un archivo histórico para la consulta de la información de años anteriores.

Una realidad es que en estos momentos todo el proceso se realiza de forma manual en cada una de sus etapas lo que sin dudas complica la labor y genera grandes dificultades en el control de esta información, la que se almacena en el mejor de los casos en documento Word. Luego es enviada a niveles superiores a través de correo electrónico, dispositivos de almacenamiento de datos, etc., lo que resulta en fallas en su manipulación y almacenamiento, afectándose así la agilidad y calidad del proceso. Esto trae consigo la existencia de duplicados o ausencia de la información. Lo anterior evidencia la poca fiabilidad en la información que se recopila, comprometiéndose así la integridad, la calidad y la autenticidad de la misma y, por tanto, del proceso en sí.

Realizar este proceso de evaluación se torna complejo debido a la diversidad de eventos en los que participan los estudiantes. Cada año la responsabilidad de esta actividad recae en el profesor guía y el profesor principal del año. Ellos deben revisar las evidencias con las que cuenta el estudiante e ir evaluando cada rubro para poder elaborar la planilla de Evaluación Integrada del estudiante. Lo anteriormente expuesto implica riesgos de contaminación de la información, ya que no se cuenta con las evidencias que respalden la participación en esos eventos.

Por lo que se plantea el siguiente **problema de la investigación**: ¿Cómo contribuir a la gestión de la información en el proceso de Gestión de Evaluación Integrada de los estudiantes de la Facultad 1?

Se define como **objeto de estudio**: Proceso de Gestión del Trabajo Educativo de la Facultad 1, el cual enmarca al **campo de acción**: Gestión de la Evaluación Integrada de los estudiantes en la Facultad 1 de la UCI.

Para dar respuesta al problema de la investigación se define el siguiente **objetivo general**: Desarrollar un módulo para el Sistema de Trabajo Educativo que contribuya a la gestión de la información en el proceso de Evaluación Integrada de los estudiantes de la Facultad 1.

Las **preguntas científicas** que se definen para guiar la investigación son:

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionados con el proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI?
2. ¿Qué propuesta de solución se define para mejorar la gestión del proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI?
3. ¿Cuáles son las características que debe cumplir el sistema de gestión del proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI?
4. ¿Cómo validar el funcionamiento del sistema gestión del proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI?

Con el propósito de darle cumplimiento al objetivo general y al problema anteriormente planteado se trazaron las siguientes **tareas de la investigación**:

1. Estudio de los referentes teóricos fundamentales que sustentan la investigación relacionados con el proceso de evaluación integral de los estudiantes en la Facultad 1 de la Universidad de las Ciencias Informáticas.
2. Análisis y diseño del sistema para la gestión del proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI.
3. Implementación de las funcionalidades del sistema para la gestión del proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI.
4. Validación de las funcionalidades del sistema para la informatización del proceso de evaluación integral de los estudiantes en la Facultad 1 de la UCI.

Para el cumplimiento de las tareas antes descritas se utilizan los siguientes métodos científicos de investigación:

**Métodos teóricos:**

**Analítico-sintético:** se utiliza para el análisis de las diferentes resoluciones que guían el proceso de evaluación integral, además de las herramientas y tecnologías que se emplean para la definición de sus características y relaciones.

**Modelación:** es necesario para una mejor comprensión del proceso de evaluación integral a través de las herramientas de la ingeniería de software.

**Métodos empíricos:**

**Entrevista:** para el análisis más detallado sobre los procesos asociados a la evaluación integral se entrevistan a los principales profesores que se al ser guías más se involucran en cada etapa.

**Observación:** se utiliza para comprender cómo se lleva a cabo de forma objetiva el proceso de evaluación integral en cada una de sus etapas.

**Capítulo #1. Fundamentación teórica del sistema para la gestión del proceso de evaluación integral de los estudiantes:**

Se hace una revisión de la bibliografía, donde se analizan los principales conceptos. Se analiza del estado del arte de la investigación. Se definen las tecnologías y herramientas a utilizar en la solución propuesta. Se propone la metodología de desarrollo de software para la construcción de la propuesta de solución de la investigación.

**Capítulo #2. Análisis y Diseño de la Solución: Sistema de Gestión del proceso de evaluación integral de los estudiantes de la Facultad 1:**

En este capítulo se definirán las funcionalidades que se van a informatizar, se define el modelado del negocio y los diagramas para lograr una mejor comprensión del mismo. Además, se determinan y especifican los requisitos funcionales y no funcionales, se modela el sistema y se identifican los casos de uso con sus respectivos diagramas para conocer las funcionalidades del sistema. Se especificarán además los patrones utilizados, así como otras características fundamentales como parte del diseño, generándose además un grupo de artefactos teniendo en cuenta la metodología a utilizar.

**Capítulo #3. Implementación y pruebas del Sistema de Gestión del proceso de evaluación integral de los estudiantes de la Facultad 1:**

En este capítulo se describe el proceso de implementación y pruebas de la solución. Se especifican la selección, diseño y aplicación de las pruebas realizadas en correspondencia a la tipología de la solución. Finalmente se muestran los resultados obtenidos una vez concluida la ejecución de pruebas.

## **CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS MÓDULO DE EVALUACIÓN INTEGRADA DEL SISTEMA DE GESTIÓN DE TRABAJO EDUCATIVO DE LA FACULTAD 1**

En este capítulo se abordarán conceptos asociados al dominio de la problemática. Se realiza un estudio del estado del arte de sistemas homólogos existentes, tanto a nivel internacional como nacional. Se explica brevemente la descripción de las tecnologías y herramientas utilizadas para el desarrollo del sistema, así como de la metodología de desarrollo usada, sus principales características y el tipo de arquitectura de software empleada.

### **I.1 Conceptos asociados al dominio del problema**

Para la comprensión del objeto de estudio de esta investigación es de vital importancia definir los siguientes conceptos:

#### **I.1.1 Evaluación integrada**

El concepto de formación tiene relación con el concepto de Paideia de la cultura griega clásica basada en la formación integral del ser humano en todas sus dimensiones: cuerpo (educación física), carácter (educación moral) y conocimientos (educación intelectual); pero su origen se remonta a la mística de la baja edad media referido al proceso mediante el cual cada ser humano se configura según la imagen de Dios que lleva en sí y debe desarrollar. Esta noción fue vertida por la tradición humanista renacentista en términos de formación de los afectos y de la razón, hasta ser asumido y desarrollado por la ilustración del siglo XVIII. El concepto de formación (Bildung) nace en estrecha relación con el concepto de ilustración y de cultura, hasta que en el contexto del romanticismo alemán del siglo XIX se amplía a las esferas de la estética, la pedagogía y aún la política (Alarcón Ortiz et al., 2019).

La formación integral va más allá de la capacitación profesional, aunque la incluye. Es un enfoque o forma de educar. Contribuye a enriquecer el proceso de socialización del estudiante, que afina su sensibilidad mediante el desarrollo de sus facultades artísticas, a su desarrollo moral y abre su espíritu al pensamiento crítico. La educación que brinda la universidad es integral en la medida en que enfoque a la persona del estudiante como una totalidad y que no lo considere únicamente en su potencial cognoscitivo o en su capacidad para el quehacer técnico o profesional. El ámbito de la formación integral es el de una práctica educativa centrada en la persona humana y orientada a cualificar su socialización para que el estudiante

pueda desarrollar su capacidad de servirse en forma autónoma del potencial de su espíritu en el marco de la sociedad en que vive y pueda comprometerse con sentido histórico en su transformación, según Fitché, 1997 (Guzmán Mirás & García González, 2017).

El Enfoque Integral para la Labor Educativa y Política Ideológica en la Universidad es una concepción sistémica que articula los procesos sustantivos universitarios en función de la formación integral de los estudiantes y de la preparación y superación integral de profesores, directivos y trabajadores en general; promueve un ambiente de trabajo educativo en todo el quehacer de la comunidad universitaria a todos los niveles de la estructura en todos los espacios y en todos los momentos; y está presente de modo transversal en los objetivos estratégicos y anuales del sistema de educación superior y de cada una de las universidades.

Especial énfasis dedica el enfoque al eslabón de base de la formación: el año académico; escenario primario en el que profesores y estudiantes trabajan por lograr los objetivos instructivos y educativos correspondientes establecidos en el currículo de cada carrera. El claustro de profesores, los profesores guías, tutores, los estudiantes, agrupados en las células básicas de la FEU (Brigada) y de la UJC (Comité de base), y el Profesor Principal de Año Académico (PPAA) encargado de liderar metodológicamente el trabajo docente educativo conjunto, conforman la Comunidad Universitaria del Año Académico. Operativamente funciona el denominado Colectivo de Año, integrado por los agentes académicos mencionados y los representantes de las organizaciones estudiantiles de base (Alarcón Ortiz et al., 2019).

## **I.2 Estudio del estado del arte de los sistemas de gestión del proceso de evaluación integrada**

En el presente acápite de la investigación se analizan varios sistemas homólogos que gestionan los procesos de ejercicios de culminación de estudios. Se analizaron sistemas informáticos desarrollados para universidades del ámbito nacional e internacional, incluida la UCI. Para la selección de estos sistemas se valoró principalmente su funcionamiento y gestión total o parcial de los procesos de evaluación integral, buscando similitudes con el sistema que se desea implementar.

### **I.2.1 ALEXIA**

Desarrollada por Educaria, empresa dedicada a prestar asistencia y servicios informáticos ubicada en Madrid, España. Es totalmente web y tiene más de 1.200 centros educativos usuarios. Aporta flexibilidad en todas las áreas de gestión: facturación, impagos, ventas, contabilidad, evaluación, programación de

aula, seguimiento del alumno, comunicación, etc. Dispone de cuadro de mando para extraer indicadores a nivel de centro y agrupación. Se integra con G Suite, Office 365, Moodle, y otros aplicativos, y permite incorporar soluciones específicas de horarios, calidad y aprendizaje, con herramientas para que el centro pueda desarrollar su propia estrategia de contenidos. La utilizan docentes, a quienes se les facilita el acompañamiento de sus estudiantes al poder observar calificaciones con el fin de evaluar rendimientos y elaborar hipótesis sobre la ruta de sus aprendizajes. Datos que pueden cruzar, a su vez, con la asistencia y con las herramientas que le ofrece plataforma de enseñanza-aprendizaje Alexia. A través de la misma los centros escolares pueden gestionar con eficacia y rapidez las tareas administrativas de su secretaría, volcar calificaciones, controlar asistencia y comunicarse con las familias según diversos y numerosos objetivos (Aguilera Alvarez & Rodríguez Rodríguez, 2020).

### **I.2.2 AGORA**

Agora es un producto de software estándar de gran calidad para la Gestión de Centros Docentes y Academias de todo tipo. Desarrollado por la empresa española Kherian Soft. Se adapta a cualquier tipo de centro o formación, ya sea de tipo oficial o de carácter libre (academias de enseñanza general, de idiomas, informática, música u oposiciones entre otras), ya se trate de un pequeño centro o una gran empresa de formación con una gestión centralizada de varias sucursales distantes geográficamente. Para ello, la aplicación cuenta con varias versiones y múltiples opciones de organización docente para utilizar en cada caso (Aguilera Alvarez & Rodríguez Rodríguez, 2020).

### **I.2.3 AKADEMOS**

Sistema de Gestión Universitaria de la UCI, conocido como Akademos, es un sistema de planificación empresarial de recursos universitarios que incluye la concepción de varias líneas de desarrollo agrupadas en las áreas de procesos. Fue desarrollado para mejorar la planificación, organización y dirección de los procesos universitarios facilitando su ejecución y control tanto para trabajadores como para estudiantes (Gonzalez Flores et al., 2017).

### **I.2.4 SIGENU**

El Sistema de Gestión de la Nueva Universidad (SIGENU) es un software desarrollado en la Universidad Tecnológica de la Habana “José Antonio Echeverría” (CUJAE) en el 2007. Se perfilaba, en su diseño original, como el resultado de la acción mancomunada de varios IES, en el desarrollo de módulos que auto-

matizarían la gestión de los procesos fundamentales y de apoyo de las universidades: matrícula, proceso docente, postgrado, relaciones internacionales, investigación, etc. No obstante, los esfuerzos realizados por los desarrolladores y la dirección del MES, hasta la actualidad, sólo se ha extendido a todos las IES el módulo de matrícula y promoción (Ruiz Suarez, 2019).

### I.2.5 DATAFEU

Esta es una aplicación informática que asegura la calidad del proceso de evaluación estudiantil y apoya a sus principales dirigentes en el proceso de toma de decisiones. Se centra principalmente en el proceso de integralidad y caracterización de los estudiantes registrando mediante evidencias la trayectoria de cada uno de ellos en un expediente digital. DataFEU fue desarrollado sobre los marcos de trabajo Symfony y Ext JS empleando MySQL como servidor de base de datos, ajustándose a la soberanía tecnológica por la cual aboga Cuba. Cuenta con cinco subsistemas que se integran entre sí y permiten el correcto funcionamiento, integridad y persistencia de los datos que se introducen y procesan. Brinda información a sistemas externos mediante servicios web que garantizan la interoperabilidad del mismo (Aguilera Alvarez & Rodríguez Rodríguez, 2020).

### I.2.6 Comparación entre los sistemas homólogos

En la Tabla 1 se muestra un resumen del estudio de las soluciones informáticas para gestionar la información de los procesos estudiantiles.

*Tabla 1 Comparación de los sistemas homólogos. Fuente: Elaboración Propia*

variables	sistemas	Alexia	Agora	Akaderos	Sigenu	Datafeu
Gestión de tareas administrativas		x	x	x	x	x
Base de datos centralizada			x	x	x	x
Registro de información de las di-		x	x		x	x



versas dimensiones educativas					
Generación automatizada de documentación	x	x			
Carácter propietario	x	x			
Gestión de datos de cultura, deporte y residencia	x	x			x
Accesible desde la web	x	x	x		
Escalabilidad de roles	x	x	x	x	x

Luego de analizar los sistemas homólogos teniendo como puntos fundamentales a las variables definidas en la tabla 1.1 se puede concluir que: las propuestas anteriores no solventan las necesidades de la universidad. No cuentan con la capacidad de llevar el control, mantenimiento y gestión de la información relacionada con la evaluación integral. No permiten controlar el desempeño de los estudiantes en las diversas dimensiones educativas posibles. Presentan además una cierta escalabilidad de roles dependiendo de permisos otorgados por niveles estudiantiles, lo que resulta complicado y extenso para el estudiante. En algunas no se puede acceder al código fuente ya que es privativo. En otro caso su interfaz principal necesita ser instalada en cada computadora que se va a utilizar por ser esta una aplicación de escritorio, lo que ocasiona pérdidas de tiempo y limitaciones en su uso. Tras un análisis exhaustivo de todas las posibilida-

des se concluye que ninguno satisface las necesidades de la problemática, comprobándose de esta forma que ninguno puede ser una solución íntegra. Aún así se identificaron un conjunto de funcionalidades básicas que debe poseer el sistema que se desea construir como son gestión de las evaluaciones de los estudiantes y la gestión de tareas administrativas, presentes en cinco de los sistemas homólogos estudiados.

### **I.3 Herramientas y Tecnologías**

#### **I.3.1 Laravel 9.8.1**

Laravel es un popular marco de trabajo de PHP que permite el desarrollo de aplicaciones web totalmente personalizadas de elevada calidad. Es uno de los más utilizados y de mayor comunidad en el mundo de la Internet. Resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de aplicaciones web.

Laravel pone énfasis en la calidad del código, la facilidad del mantenimiento y escalabilidad, lo que permite realizar proyectos de cualquier magnitud. Trabaja con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos. Por supuesto también dispone de una arquitectura de clases también muy adecuada, que promueve la separación del código por responsables. Su estilo arquitectónico es Modelo-Vista-Controlador (MVC).

Es un marco de trabajo más nuevo que muchos de sus competidores. Ha venido desde atrás consolidándose como el marco de PHP más utilizado. Esta progresión se debe a varios factores, como la cantidad de funcionalidades que nos ofrece ya lista, su facilidad de uso, ya que es uno de los más sencillos de utilizar, el potente sistema de comandos de consola que nos ofrece artisan para resumir tareas tediosas y repetitivas, su solidez que permite que su crecimiento no rompa de manera radical los proyectos que necesitaban ser actualizados(Kinsta, 2022).

#### **1.3.2. PHP 8.1.2**

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de programación de código abierto muy popular. Está especialmente adecuado para el desarrollo web y puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script,

aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, macOS, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos. Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos.(Tapia, 2022).

### **1.3.3. JavaScript:**

JavaScript es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs y crear aplicaciones web. A pesar de la similitud en el nombre, no está relacionado con Java. JavaScript es un lenguaje de alto nivel, dinámico e interpretado. Se define como orientado a objetos, basado en prototipos, imperativo y débilmente tipado. Junto al HTML y a CSS, es una de las tres tecnologías esenciales en la producción mundial webs: la mayoría de los sitios web la emplean y es compatible con todos los navegadores modernos sin necesidad de plugins. JavaScript posee una API para trabajar con texto, arreglos, fechas y expresiones regulares. JavaScript también se utiliza en entornos que no se basan en web, tales como documentos PDF, navegadores específicos del sitio y widgets de escritorio. Máquinas virtuales de JavaScript (VMs) mucho más nuevas y rápidas, así como plataformas construidas sobre este lenguaje, han aumentado la popularidad de JavaScript para aplicaciones web. Por parte del cliente, JavaScript se ha aplicado tradicionalmente como un lenguaje interpretado, pero los navegadores más recientes pueden realizar la compilación al momento. JavaScript es interpretado en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML(Perez Porto & Gardey, 2018).

### **1.3.4. Sistemas gestores de base de datos**

#### **1.3.4.1. MySQL 8.0.3**

Es un sistema gestor de base de datos SQL (Structured Query Language) muy rápido, multiproceso, multiusuario y sólido. MySQL 8.0.3 está diseñado para sistemas de producción de carga pesada y de misión crítica, así como para integrarse en software de implementación masiva. MySQL es el sistema de administración de bases de datos SQL de código abierto desarrollado, distribuido y respaldado por Oracle Corporation.

Está escrito en los lenguajes C y C++ y se destaca por su gran adaptación a varios entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más empleados como PHP, Perl y Java. Este sistema es multiplataforma, es adaptable a distintos sistemas operativos. Permite definir varios usuarios que se autoricen para el manejo de las bases de datos con determinados permisos revocables, garantizando la seguridad de la plataforma. MySQL usa la memoria para búferes internos y caché que respaldan el número de concurrencias de consultas a bases de datos, y se emplean para las respuestas rápidas del sistema de acuerdo a las condiciones del hardware del sistema (Robledano, 2019).

#### **1.3.4.2. PostgreSQL**

PostgreSQL es un poderoso sistema de base de datos relacional de objetos de código abierto que usa y extiende el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. PostgreSQL viene con muchas características destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los datos y a crear entornos tolerantes a fallas, y a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos. Además de ser gratuito y de código abierto, PostgreSQL es altamente extensible. Se pueden crear varios tipos de aplicaciones usando la base de datos PostgreSQL usando lenguajes de programación populares como Java, Python, Ruby, .NET o PHP. PostgreSQL es compatible con ACID (atomicidad, consistencia, aislamiento, durabilidad). Tiene vistas, disparadores y claves externas actualizables y materializadas (Wodyk & Skublewska-Paszkowska, 2020).

#### **1.3.4.3. Selección del sistema gestor de base de datos**

En el caso de bases de datos en las que el número de registros no es demasiado grande (hasta 1000 registros), se utiliza una aplicación web y los parámetros técnicos del dispositivo en el que se ejecuta la base de datos son de clase baja o media, MySQL funciona muy bien. Este hecho confirmaría la gran popularidad de esta solución en el caso de aplicaciones de pequeños consumidores. PostgreSQL logró resultados ligeramente peores para una pequeña cantidad de datos (hasta 1000 registros) en la base de

datos, lo que le otorga el segundo lugar entre las dos soluciones probadas (Wodyk & Skublewska-Paszowska, 2020).

En el caso de las operaciones de agregar, actualizar y eliminar datos de la base de datos para una aplicación de escritorio, el sistema de base de datos PostgreSQL resultó ser casi el doble de eficiente. En el caso de la recuperación de datos de la base de datos, las diferencias fueron significativamente menores en comparación con otras operaciones, pero el sistema MySQL logró mejores resultados. Las grandes diferencias en los tiempos de realización de operaciones particulares a favor del sistema PostgreSQL pueden resultar de su mejor adaptación a la notación JSON. PostgreSQL es más eficiente para la aplicación de escritorio mientras se carga con datos pequeños, en ese caso 1000 consultas. Luego de consultar a varios autores se llega a la conclusión de que MySQL es el sistema de gestor de base de datos indicado para proyectos basados en aplicaciones web que requieran alta velocidad y no necesiten de consultas complejas, como es el caso del sistema de gestión del proceso de evaluación integral en la Facultad 1 de la UCI (Jiménez Caballero & Brito Palacio, 2021).

#### **1.3.4.3.1 phpMyAdmin 5.1.1:**

PhpMyAdmin es una aplicación web que sirve para administrar bases de datos MySQL de forma sencilla y con una interfaz amistosa. Se trata de un software muy popular basado en PHP. La ventaja de usar una aplicación web es que nos permite conectarnos con servidores remotos, a los cuales no siempre se puede acceder usando programas de interfaz gráfica. A través de phpMyAdmin se pueden realizar todas las tareas de administración de un servidor de MySQL. La cantidad de opciones que tenemos disponibles son muy amplias, desde la creación de bases de datos, la creación de tablas, la modificación de su estructura, la búsqueda y edición de los registros, hasta las copias de seguridad o importados (García de Zuñiga, s. f.).

#### **1.3.5. Bootstrap 5**

Bootstrap es un marco de trabajo que ayuda a la creación de las interfaces de usuario mediante la utilización de componentes como CSS y JavaScript. Esto se puede visualizar en cualquier dispositivo porque se ajusta a las proporciones y resolución de la pantalla en la cual se observa, más conocido como diseño responsivo (Responsive Design). Mejora notablemente las proporciones de los CSS de las interfaces de usuario y ayuda al desarrollo web de las aplicaciones; teniendo una interfaz más amigable con el usuario y el incremento en la interacción con la misma desde cualquier lugar (Berrones & Alfonso, 2020).

Bootstrap es un marco de trabajo CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegados, etc. También existe la posibilidad de utilizar extensiones de Javascript adicionales. Fue desarrollado inicialmente por Twitter en 2011 y permite crear interfaces de usuario limpias y compatibles con todo tipo de dispositivos. Entre las ventajas que tiene Bootstrap es que favorece el diseño responsivo, el cual se utiliza para mejorar la experiencia de los usuarios en el sitio web y en consecuencia el posicionamiento (Deymar, 2022).

### **1.3.6. HTML5**

HTML es el lenguaje con el cual se desarrollan las páginas web. Es el lenguaje usado por los navegadores para mostrar las páginas web al usuario, publicando la interfaz más extendida en la red. Permite agrupar textos, sonidos e imágenes y combinarlos a nuestro gusto. Además, nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto. HTML5 es el nuevo estándar para HTML que proporciona a los usuarios y desarrolladores web una funcionalidad mejorada. Las versiones anteriores de HTML, HTML 4.01, que llegaron en 1999, y el desarrollo web han cambiado notablemente desde entonces. Se creó para ofrecer contenido enriquecido sin la necesidad de complementos adicionales ni tecnologías patentadas. El nuevo poder de HTML5 proporciona al usuario todo, desde animación hasta gráficos, música y películas, y también se puede utilizar para crear aplicaciones web complicadas y también es multiplataforma (Fernandez, 2021).

Su nombre, HyperText Markup Language significa literalmente Lenguaje de marcado de hipertexto. El HTML5 es un lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido en los sitios web. Se finalizó y publicó el 28 de octubre 2014 en el World Wide Web Consortium (W3C), siendo esta es la quinta revisión del estándar HTML desde la creación de la World Wide Web. Sus objetivos principales son mejorar el lenguaje dando soporte para los últimos objetos multimedia mientras se mantiene fácilmente legible por los humanos y a su vez ser comprendido constantemente por ordenadores y dispositivos (navegadores web, programas de análisis, etc.). Incluye modelos de procesamiento detallados para apoyar a las implementaciones más interoperables. Se extiende, mejora y racionaliza el margen de beneficio disponible para documentos e introduce un margen de beneficio y las interfaces de programación de aplicaciones (API) para aplicaciones web complejas. Por las mismas razones, HTML 5 también es un candidato potencial para aplicaciones móviles multiplataforma. Muchas

de las características de HTML5 se han diseñado con dispositivos de baja potencia, tales como teléfonos inteligentes y tabletas. En particular, HTML5 añade muchas nuevas características sintácticas. Estos incluyen las nuevas etiquetas video, audio y lienzos, así como la integración de gráficos vectoriales escalables (SVG) (en sustitución de las genéricas etiquetas object) y MathML para fórmulas matemáticas. Estas características están diseñadas para hacer más fácil el incluir y manejar multimedia y contenido gráfico en la web sin tener que recurrir a plugins propietarios y APIs(Fernandez, 2021).

### **1.3.7. Servidor Apache 2.4.52**

Apache es un servidor web gratuito y de código abierto basado en Unix desarrollado por Apache Software Foundation. Apache es liviano, con todas las funciones y más poderoso que otros servidores web basados en Unix. Algunos sondeos realizados demuestran que más del 70% de los sitios web en Internet están manejados por Apache, haciéndolo más extensamente usado que todos los otros servidores web juntos. Apache 2.4.52 es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP. Es multiplataforma, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos. Además gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.(Dinahosting, 2019).

### **1.3.8. Visual Paradigm 8.0**

Visual Paradigm es una herramienta UML. La herramienta está diseñada para una amplia gama de usuarios, incluidos ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en construir de manera confiable sistemas de software a gran escala utilizando un enfoque orientado a objetos. Además, Visual Paradigm admite los últimos estándares de notación UML. Visual Paradigm 8.0 es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite modelar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación.(Pressman, 2002)

### **1.3.9. Apache JMeter 5.4.1**

La aplicación Apache JMeter es un software de código abierto escrito en Java. Está diseñado para probar el rendimiento tanto en recursos estáticos como dinámicos. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su fuerza o para analizar el rendimiento general bajo diferentes tipos de carga. Apache JMeter permite realizar pruebas de carga al sistema que se desarrolla y mide el rendimiento de los recursos web. Toda la información de las pruebas y análisis del rendimiento se muestran a través de las estadísticas del servidor o servicio de destino mediante diagramas y gráficos. Para diseñar un plan de pruebas, JMeter dispone de una interfaz gráfica a modo de diseñador, en la que el probador puede ir agregando componentes de manera visual, y ejecutar los componentes agregados, viendo el resultado. Una vez finalizado el diseño del plan de pruebas, la herramienta permite grabar este como un fichero (.jmx). La propia herramienta permite ejecutar un fichero (.jmx) previamente generado, mediante línea de comandos o la propia interfaz GUI. La ejecución de un fichero (.jmx) realiza peticiones a la aplicación objetivo a testear (JMeter dispone de la posibilidad de generar muchos tipos de peticiones: HTTP, FTP, LDAP, que se especifican al generar el fichero (.jmx)). Para cada petición ejecutada, JMeter recopila ciertos datos. Además, en el fichero (.jmx) se puede especificar el número de usuarios de cada tipo que ejecuta las peticiones contra la aplicación, es decir, el (.jmx) simula una o más comunidades de usuarios trabajando contra la aplicación objetivo (Sheti, 2022).

#### **1.3.10. Acunetix Web Vulnerability Scanner**

Es una solución completa de pruebas de seguridad de aplicaciones web que se puede utilizar tanto de forma independiente como parte de entornos complejos. Ofrece evaluación y gestión de vulnerabilidades integradas, así como muchas opciones para la integración con herramientas de desarrollo de software líderes en el mercado. Al usar Acunetix, aumenta el nivel de ciberseguridad y elimina muchos riesgos de seguridad a un bajo costo de recursos. Esta herramienta permite fortalecer la seguridad del sistema de gestión que se desarrolla mediante una base de pruebas que incluye inyecciones de SQL, XSS, configuraciones incorrectas, pruebas de penetración manuales, contraseñas débiles y bases de datos expuestas. Los resultados de la base de pruebas que aplica la herramienta muestran las vulnerabilidades que deben ser solucionadas para una mayor seguridad de la aplicación (Menejías García et al., 2021).

#### **1.3.11. Visual Studio Code 1.64.2:**

Visual Studio Code es un editor de código fuente desarrollada por Microsoft, ligera pero potente. Se ejecuta en escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para



JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity). Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft. (Documentation for Visual Studio Code, s. f.).

### I.3 Metodología de desarrollo de software

#### AUP-UCI

La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (Jiménez Caballero & Brito Palacio, 2021). En la tabla siguiente se muestra las fases de la metodología AUP-UCI:

*Tabla 2 Fases de la Variación AUP-UCI. Fuente: Elaboración Propia*

<b>Fases AUP</b>	<b>Fases Variación AUP-UCI</b>	<b>Objetivos de las fases (Variación AUP-UCI)</b>
<b>inicio</b>	inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planificación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite

		<p>obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.</p>
<b>elaboración</b>	ejecución	<p>En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.</p>
<b>construcción</b>		
<b>transición</b>		
	cierre	<p>En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto.</p>

La metodología de software AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual). Existen tres formas de encapsular los requisitos, los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso). Surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

- Escenario No. 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No. 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No. 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No. 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Luego del análisis de la investigación realizada, se usa la variante 3 (MC) y el Escenario No. 2 para el desarrollo del sistema. Este proceso es complejo y requiere de un estudio y análisis detallado del proceso de evaluación integrada en la Facultad 1 de la UCI. La identificación de las clases que intervienen en el negocio permite la comprensión de todo el proceso, todas las actividades que se realizan y su relación entre las clases. El Escenario No. 2 garantiza el análisis concreto del negocio y una disminución de las probabilidades de fracaso en el desarrollo del sistema (Jiménez Caballero & Brito Palacio, 2021).

## **Conclusiones del capítulo**

Posterior al análisis de los principales aspectos a tener en cuenta de la gestión de la evaluación integral en la Facultad 1 de La Universidad de las Ciencias Informáticas y estudiar los sistemas con más similitudes usados en el ámbito nacional e internacional, se alcanzó un dominio de los aspectos más importantes a tener en cuenta para lograr la informatización de este proceso de forma exitosa, buscando satisfacer las necesidades del cliente, la Facultad 1.

Luego de un detallado proceso de selección de tecnologías y herramientas a utilizar en el diseño, desarrollo y pruebas del módulo a implementar se obtuvo una base sólida para el inicio del trabajo que se va a realizar. Y la selección de la metodología de desarrollo de software adecuada, AUP-UCI, específicamente su segundo escenario, brinda una mayor rapidez de desarrollo con una baja probabilidad de errores.

Tras el estudio realizado:

- Se constató la necesidad de crear un sistema que gestione la integralidad de los estudiantes en la Facultad 1 de la UCI.
- La investigación de los sistemas homólogos permitió identificar características importantes.
- 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.
- La metodología de desarrollo de software seleccionada es AUP-UCI en el Escenario No.2.

## **CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO: SISTEMA DE GESTIÓN DEL PROCESO DE EVALUACIÓN INTEGRAL DE LOS ESTUDIANTES DE LA FACULTAD 1.**

En el presente capítulo se realiza un análisis de la propuesta de solución: Sistema de Gestión del proceso de evaluación integral de los estudiantes de la Facultad 1; a partir de requisitos funcionales, no funcionales y el diseño de la propuesta de solución. Se presentan los artefactos ingenieriles según la metodología escogida para el desarrollo de la misma. Además, se realiza un análisis de su funcionamiento, teniendo en cuenta estilos arquitectónicos y patrones de diseño aplicables al marco de desarrollo seleccionado: Laravel.

### **II.1 Flujo actual del proceso.**

El proceso de evaluación integral de los estudiantes de la Facultad 1 tiene un nivel de universidad y su organización en la facultad es responsabilidad del decano. En fechas cercanas a la culminación del curso académico corriente, los profesores guías asistidos por los presidentes de grupo indican a los estudiantes que vayan tomando en cuenta todas las actividades en las que participaron y resultados que obtuvieron durante ese curso, y envíen un documento donde presenten esa información anexando a este las evidencias necesarias y suficientes que respalden su participación en los diferentes eventos.

En un día señalado se efectúa una reunión encabezada por el profesor guía y el presidente de grupo en donde además se selecciona a un encargado de redactar el informe del día. Cada alumno le corresponde presentar su resumen con sus evidencias y tras ser escuchado por sus compañeros corresponde a estos presentar sus criterios acerca de cuál debería ser la evaluación otorgada. Se analizan las propuestas y se decide por la más acertada. Al finalizar la reunión se escuchan criterios y opiniones y se toman en cuenta las propuestas.

Para los estudiantes que se encuentran cursando su último año académico en la universidad se les pide además un resumen de los años anteriores de carrera en donde se comprueba nuevamente lo acertado de sus evaluaciones en busca de comprobar además si puede optar por algún premio o reconocimiento especial. En el caso del Premio al Mérito Científico, por año, el estudiante tiene mostrar las actividades que realizó, y que tributan a sus posibilidades de obtención de este reconocimiento. Luego el profesor de

acuerdo a la puntuación que se otorga según los resultados obtenidos y el tipo de evento determina si el estudiante puede o no optar por este reconocimiento.

## II.2 Modelo conceptual

Un modelo conceptual de un sistema software constituye una abstracción externa que describe, mediante diagramas y notaciones con distinto grado de formalidad, el conocimiento que debe poseer una persona acerca de un sistema. Desde el punto de vista del usuario, el modelo conceptual representa la información que cualquier usuario debería tener o adquirir sobre el sistema y está formado por un conjunto de elementos y relaciones entre ellos. El modelo conceptual debe suministrar información al usuario acerca de que hace el sistema y los mecanismos para llevarlo a cabo (Delgado Dapena, 2002).

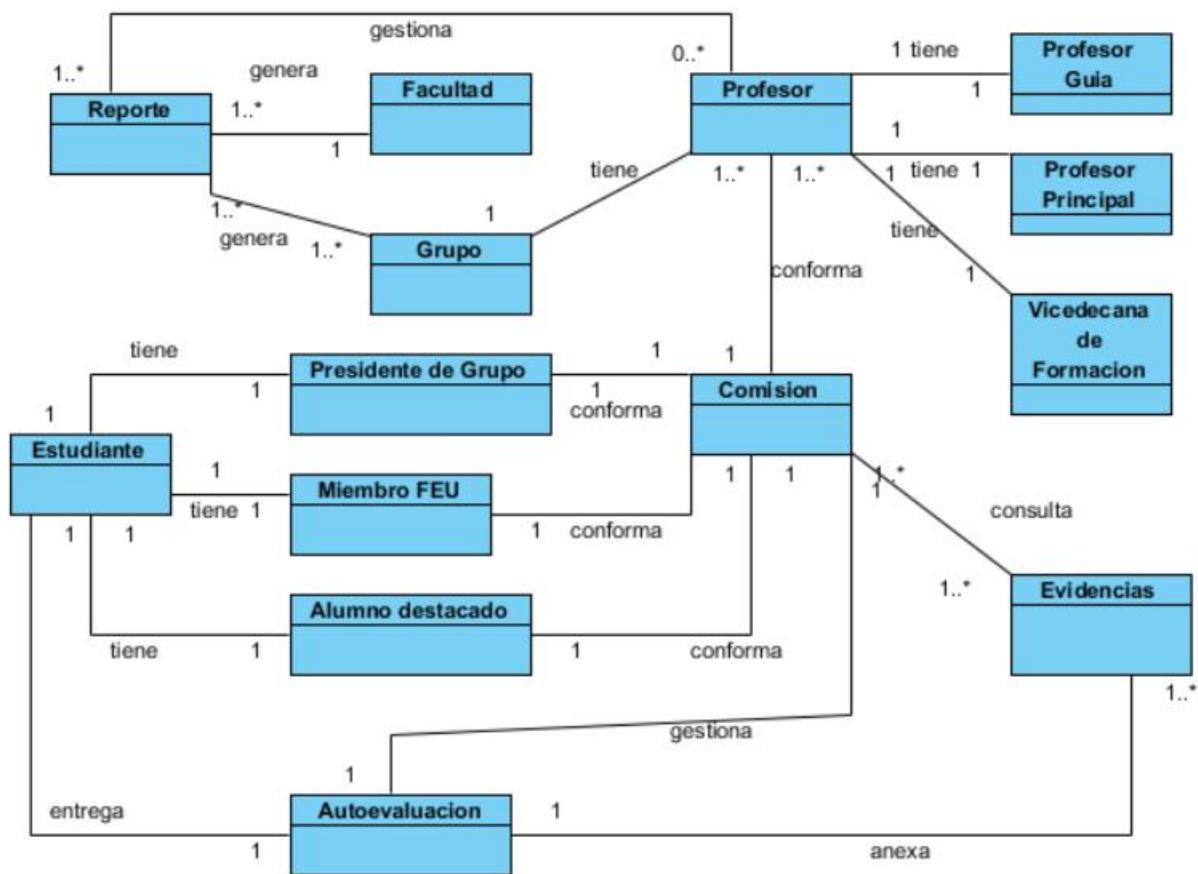


Ilustración 1 Mapa Conceptual. Fuente: Elaboración Propia.

### **II.3 Descripción de las clases del modelo**

Facultad: Facultad 1 de La Universidad de Ciencias Informáticas.

Grupo: Grupos docentes de la Facultad 1.

Reporte: Reporte generado tras la conclusión del proceso de evaluación integral.

Profesor Guía: Profesor encargado de cada grupo docente.

Profesor Principal: Profesor Principal de cada año docente.

Vicedecana de Formación: Encargada de las actividades de formación en la Facultad.

Profesor: Miembro principal encargado de las decisiones finales en la comisión.

Comisión: Conjunto de profesor, presidente de grupo, miembro de la FEU y alumno destacado encargados de la revisión y aprobación de las caracterizaciones presentadas por los estudiantes.

Estudiante: Integrantes de cada grupo docente de la Facultad, encargados de enviar sus caracterización de integralidad y anexar las evidencias que lo respaldan.

Presidente de Grupo: Estudiante designado por sus compañeros como el más capacitado para representarlos.

Miembro FEU: Estudiante con cargo de relevancia en la Federación Estudiantil Universitaria.

Alumno destacado: Estudiante sobresaliente por cualidades y resultados.

Evidencias: Prueba de la participación y de los resultados obtenidos por los estudiantes en las diferentes actividades.

Autoevaluación: Caracterización presentada por el estudiante con sus evidencias anexadas y que debe ser aprobada por la comisión.

### **II.4 Propuesta de solución**

Con el propósito de solucionar la problemática definida en la presente investigación, se desarrollará un sistema que gestione todo el proceso de evaluación integral de la Facultad 1 de la UCI. Como un módulo que formara parte de un sistema de gestión denominado por las siglas SGTE.

Este módulo permitirá gestionar la evaluación integral de todos los estudiantes de la Facultad. Además, permitirá organizar varios de los subprocesos vinculados al proceso de evaluación integral, como el otorgamiento del Premio al Mérito Científico.

Este sistema asignará roles a usuarios que estén vinculados al proceso. De acuerdo con cada rol que sea asignado al usuario se determinarán los subprocesos a los que le corresponda estar vinculado.

El sistema permite la creación de las variables a tomar en cuenta a la hora de determinar la evaluación integral de cada estudiante, así como el otorgamiento de una evaluación por cada uno de los acápites y una evaluación final en correspondencia. También verifica si el estudiante cumple con las condiciones necesarias para optar por el Premio al Mérito Científico. Y permitirá crear informes, en formato PDF, informes de todo lo anteriormente expuesto, en caso de que sea necesario.

## **II.5 Especificación de los requisitos del sistema**

Los requisitos de software son la condición que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema, o una restricción de este. En el otro extremo, es una definición detallada y formal de una función del sistema.

La calidad con la que se realice la captura de los requisitos va a influenciar en todo el proceso de desarrollo del software, repercutiendo en el resto de las fases de desarrollo del mismo. Estos contribuyen a tomar mejores decisiones de diseño y de arquitectura. También le permite al equipo de desarrollo reducir los problemas de mantenimiento.

Estos requisitos son divididos en dos subconjuntos: los requisitos funcionales (RF), que son las condiciones que el sistema debe cumplir, y los requisitos no funcionales (RNF), que representan las cualidades que la solución debe tener. Especificar que el Módulo de Gestión de Evaluación Integrada pertenece al Sistema de Trabajo Educativo de la Facultad, en conjunto con otros módulos, siendo uno de ellos el de Gestión de Perfil de Usuario, del cual se incorporan los requisitos para la creación de usuarios, roles y autenticación necesarios para el funcionamiento correcto del sistema.

### **Requisitos Funcionales**



Los 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 también pueden declarar explícitamente lo que el sistema no debe hacer (García-Holgado et al., 2022).

Estos requerimientos dependen del tipo de software que se desarrolle de los posibles usuarios del sistema y del enfoque general tomado por la organización al redactar requerimientos (Jiménez Caballero & Brito Palacio, 2021).

En el proceso de levantamiento de requisitos para el desarrollo del Sistema de gestión del proceso de evaluación integral de los estudiantes de la facultad 1, fueron detectados un total de 26 requisitos funcionales.

*Tabla 3 Requisitos Funcionales. Fuente: Elaboración Propia.*

No.	Nombre	Descripción	Complejidad	Prioridad del cliente
RF1	Crear autoevaluación	El sistema permite a los usuarios con rol estudiante crear su autoevaluación	Alta	Alta
RF2	Modificar autoevaluación	Permite al usuario con rol estudiante modificar su autoevaluación	Media	Media
RF3	Eliminar autoevaluación	Permite al usuario con rol estudiante eliminar su autoevaluación	Baja	Baja
RF4	Mostrar autoevaluación	Permite al usuario con rol estudiante mostrar su autoevaluación	Media	Media
RF5	Crear reporte	El sistema permitirá al usuario con rol profesor	Media	Media

		crear los diferentes reportes asociados a grupo o facultad		
RF6	Modificar reporte	El sistema permitirá al usuario con rol profesor modificar los reportes	Media	Baja
RF7	Eliminar reporte	El sistema permitirá al usuario con rol profesor eliminar los reportes	Baja	Baja
RF8	Mostrar rol	El sistema permitirá al usuario con rol profesor mostrar los reportes	Media	Media
RF9	Crear lista Premio al Mérito Científico	Permite al usuario con rol de profesor crear un listado para ingresar los estudiantes que optan por el premio	Media	Alta
RF10	Modificar lista Premio al Mérito Científico	Permite al usuario con rol de profesor modificar el listado de los estudiantes que optan por el premio	Media	Alta
RF11	Eliminar lista Premio al Mérito Científico	Permite al usuario con rol de profesor guía la creación de categorías de las variables	Baja	Alta
RF12	Mostrar lista Premio al Mérito	Permite al usuario con rol de	Media	Alta

	Científico	profesor mostrar los estudiantes que optan por el premio		
RF13	Crear comisión	Permite al usuario con rol profesor crear una comisión	Alta	Media
RF14	Modificar comisión	Permite al usuario con rol de profesor modificar una comisión	Media	Media
RF15	Eliminar comisión	Permite al usuario con rol profesor eliminar una comisión	Baja	Baja
RF16	Mostrar comisión	Permite al usuario con rol profesor mostrar una comisión	Media	Alta
RF17	Crear caracterización	Permite al usuario con rol miembro de comisión crear una caracterización	Alta	Alta
RF18	Modificar caracterización	Permite al usuario con rol miembro de comisión modificar una caracterización	Media	Media
RF19	Eliminar caracterización	Permite al usuario con rol miembro de comisión eliminar una caracterización	Baja	Baja
RF20	Mostrar caracterización	Permite al usuario con rol miembro de comisión mostrar una caracterización	Media	Media

RF21	Crear grupo	Permite al usuario con rol profesor crear un grupo para agrupar las autoevaluaciones	Baja	Alta
RF22	Modificar grupo	Permite al usuario profesor modificar un grupo	Media	Alta
RF23	Eliminar grupo	Permite al usuario con rol profesor eliminar un grupo	Baja	Baja
RF24	Mostrar grupo	Permite al usuario con rol profesor y con rol estudiante mostrar un grupo y acceder a la gestión de autoevaluaciones por estudiante	Alta	Alta
RF25	Exportar a PDF las caracterizaciones	Permite al usuario con rol profesor exportar con formato PDF las caracterizaciones	Alta	Alta
RF26	Exportar a PDF los reportes	Permite al usuario con rol profesor exportar en formato PDF los reportes generados por grupo o facultad	Alta	Alta

## **Requisitos No Funcionales**

Requisitos no relacionados directamente con la funcionalidad del sistema. Describen las restricciones del producto a desarrollar. Pueden describir las restricciones externas del sistema. Definen las cualidades globales que el sistema ha de exhibir (García-Holgado et al., 2022).

### **Requisitos de usabilidad:**

**RNF1:** Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.

**RNF2:** El sistema debe poseer interfaces gráficas bien formadas, que sean intuitivas.

**RNF3:** La aplicación web debe poseer un diseño responsivo a fin de garantizar la adecuada visualización en múltiples dimensiones de pantallas de computadoras y dispositivos móviles.

**RNF4:** La interfaz debe ser de color claro con un alto contraste entre el fondo y las opciones del sistema.

**RNF5:** La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.

**RNF6:** El sistema debe proporcionar mensajes de éxito, precaución y error que sean informativos.

**RNF7:** El sistema debe mostrar mensajes de advertencia ante la solicitud de eliminar datos por parte del usuario.

**RNF8:** El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente acceder.

### **Requisitos de Hardware:**

**RNF9:** El servidor debe tener las siguientes características mínimas: 8GB de memoria RAM y 10GB HDD.

### **Requisitos de Seguridad:**

**RNF10:** Los permisos para efectuar cambios en el sistema dependen del rol del usuario, donde el rol corresponde con el área donde puede trabajar el usuario. Aunque todos los roles pueden solicitar la información de otras secciones independientemente de los permisos que tengan sobre esta.

### **Requisitos de diseño e implementación:**

**RNF11:** La lógica del negocio será programada en el lenguaje de programación PHP.

**RNF12:** Las vistas serán programadas en el lenguaje de etiquetado HTML5.

**RNF13:** Para las validaciones de campos será usado el atributo "pattern" de HTML5.

**RNF14:** Para la entrada de los informes de las Evaluaciones Integrales se usará el plugin CKEditor5

**RNF15:** Para los graficar estadísticas se usará el plugin Chart.js

**RNF16:** Las alertas serán lanzadas con el plugin SweetAlert2

**Requisitos de Software:**

**RNF17:** El servidor debe tener los servicios de Apache/2.4.52 (Win64), OpenSSL/1.1.1m, PHP/8.1.2 y MySQL 8.1.2

**RNF18:** El cliente debe contar con un navegador web (Google Chrome 105.0.5195.102 o superior, Mozilla Firefox 104.0 o superior)

**II.6 Modelos de Casos de Usos del Sistema**

El modelo de casos de uso contiene actores, casos de uso y las relaciones que existen entre ellos. Describe el comportamiento del sistema en diferentes condiciones, mientras responde a las peticiones realizadas por los usuarios. Un actor es una entidad externa del sistema que participa en cómo se desarrolla el caso de uso en la aplicación (IONOS Digital Guide, 2022b).

**Actores Del Sistema**

Los actores representan a los usuarios que intercambian información con el sistema. Los actores representan a los usuarios que intercambian información con el Sistema.

*Tabla 4 Actores del Sistema. Fuente: Elaboración Propia.*

Actor del Sistema	Descripción
Estudiante	Gestiona su autoevaluación y muestra grupo
Profesor	Profesor guía, profesor principal y vicedecano de formación. Gestiona la comisión, el premio al Mérito Científico y los reportes
Comisión	Integrada por profesores y estudiantes. gestiona autoevaluaciones, caracterizaciones y consulta evidencias

<b>Objetivo</b>	Insertar Autoevaluación
<b>Actores</b>	Estudiante
<b>Resumen</b>	Debe añadir una autoevaluación a la Evaluación Integrada
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	Usuario autenticado

<b>Postcondiciones</b>	Existencia del Usuario	
<b>Flujo de Eventos</b>		
<b>Flujo Básico&lt;Autoevaluación Insertada Correctamente&gt;</b>		
<b>Actor</b>		<b>Sistema</b>
1	El actor selecciona la opción Insertar Autoevaluación en la tabla de Evaluaciones.	
2		El sistema muestra una ventana modal con los campos a llenar por el actor.
3	El actor llena los campos con la información correspondiente.	
4		Válida que los campos estén correctos.
		Inserta la autoevaluación y retorna un mensaje de éxito.

### Prototipo Elemental de Interfaz Gráfica

Nueva Información del Estudiante
×

---

Estudiante <input type="text"/>	TSU <input type="text"/>	Muy Grave <input type="text"/>
Producción <input type="text"/>	FEU/UJC <input type="text"/>	Repitente <input type="text"/>
Investigación <input type="text"/>	Premio Mella <input type="text"/>	Arrastre <input type="text"/>
Deporte <input type="text"/>	Título de Oro <input type="text"/>	Mundiales <input type="text"/>
Guardia <input type="text"/>	Menos Grave <input type="text"/>	Índice Académico <input type="text"/>
Cultura <input type="text"/>	Grave <input type="text"/>	Otros Indicadores <input type="text"/>
Residencia <input type="text"/>		

### Flujos Alternos

#### Existencia de la Autoevaluación

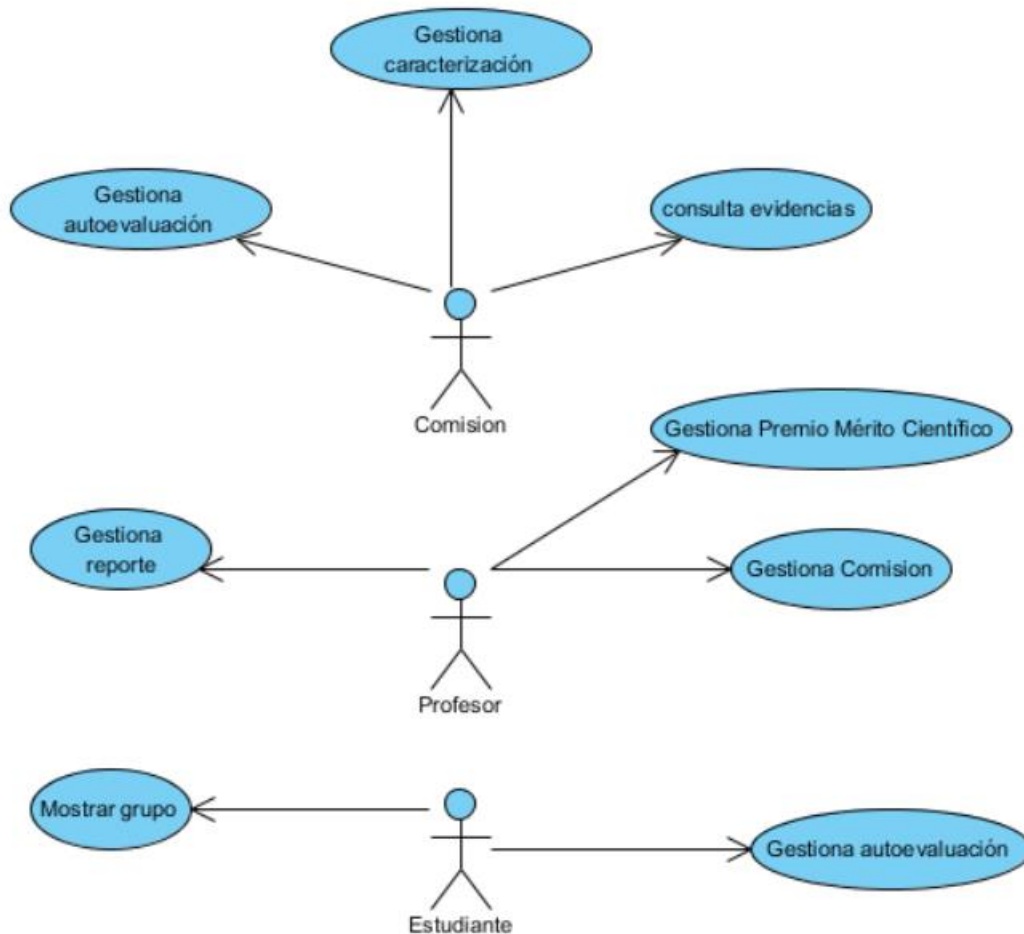
<b>Actor</b>		<b>Sistema</b>
--------------	--	----------------

1	El actor selecciona la actividad Insertar Autoevaluación en la tabla de Evaluaciones.	
2		El sistema muestra una ventana modal con los campos a llenar por el usuario.
3	El actor llena los campos con la información correspondiente.	
4		Válida que los campos estén correctos.
5		Autoevaluación existente retorna con un mensaje de error.

### Diagrama de Caso de Uso del Sistema

Para una mejor comprensión del sistema, se muestra a continuación un diagrama con los casos de usos del sistema.





*Ilustración 2 Caso de Uso del Sistema. Fuente: Elaboración Propia.*

## II.7 Diagrama de Clases

Un diagrama de clases muestra la existencia de clases y sus relaciones en el diseño lógico de un sistema. Un diagrama de clases puede representar todo o parte de la estructura de un sistema. Los diagramas de clase muestran la estructura de un modelo en particular, las entidades que deben existir, su estructura interna y las relaciones con otras clases. Los diagramas de clases no muestran información temporal (Is-las Hernandez, 2021).

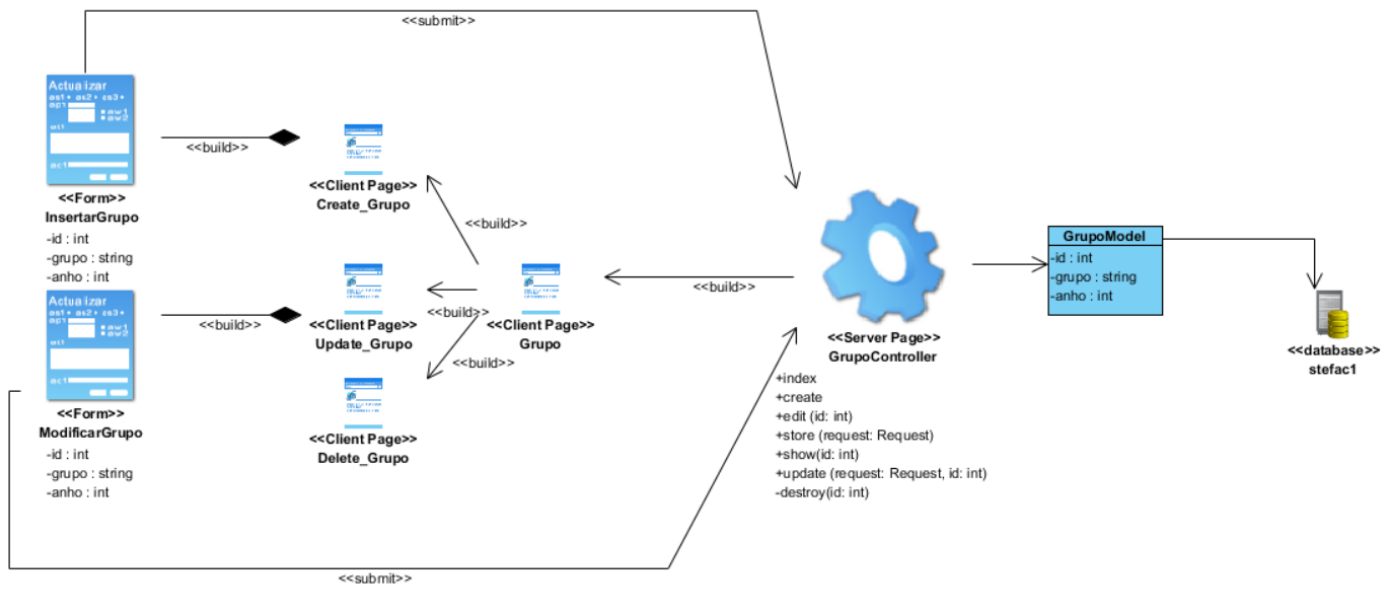


Ilustración 3 Diagrama de Clases Grupo. Fuente: Elaboración Propia.

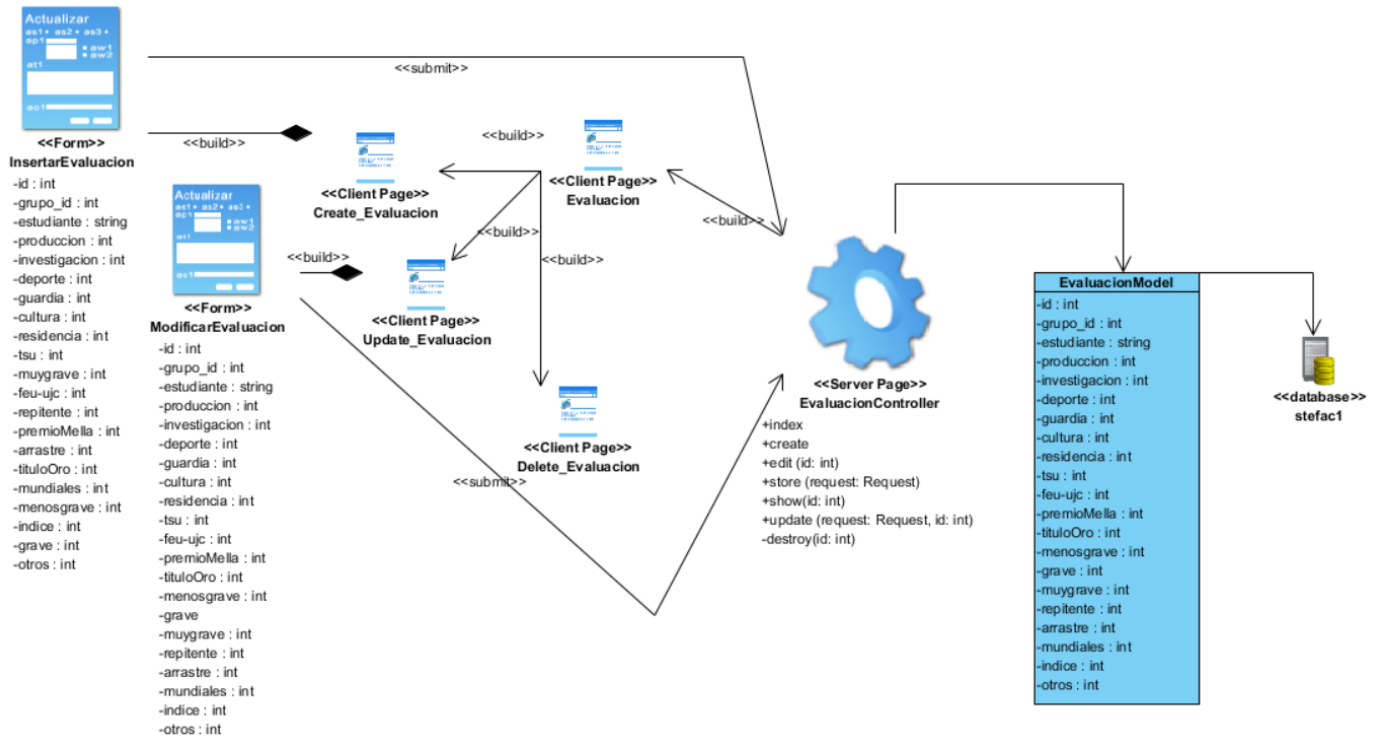


Ilustración 4 Diagrama de Clases Evaluación. Fuente: Elaboración Propia.

## II.8 Descripción de la arquitectura de software y los patrones de diseño

### Arquitectura de Software

Aunque actualmente no existe una definición única para el concepto de arquitectura de software y el termino ha sido abordado por una gran cantidad de autores, se reconoce como la más completa a la dada por el Institute of Electrical and Electronics Engineers (IEEE) Standard 1471-2000 y se refiere a la arquitectura de software como la organización fundamental de un sistema enmarcada en sus componentes, las relaciones entre ellos, y el ambiente, y los principios que orientan su diseño y evolución" (Rodríguez Peña & Silva Rojas, 2016).

Al ser utilizado el marco de trabajo Laravel para el desarrollo del módulo del sistema de gestión para la evaluación integral, la arquitectura y los patrones son heredados del mismo. Laravel utiliza la arquitectura Modelo Vista Controlador (MVC). Es un estilo de arquitectura de software que separa los datos de una

aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. Este sistema es fácilmente escalable y ofrece una multitud de funciones ya integradas (*Modelo vista controlador (MVC)*, s. f.).

**Modelo:** el modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

El modelo es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Además define las reglas del negocio (funcionalidad del sistema). Lleva un registro de las vistas y controladores del sistema y, también, notificará a las vistas los cambios que en los datos pueda producir un agente externo

**Vista:** la vista o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con este.

Las vistas se encargan de recibir los datos del modelo y mostrarlos al usuario. Poseen un registro de su controlador asociado (normalmente porque además lo instancia). Y pueden dar el servicio de actualización para que cuando sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

**Controlador:** los controladores actúan como intermediarios entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El controlador es responsable de recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.). Además contiene las reglas de gestión de eventos, del tipo "si el evento Z entonces acción W" estas acciones pueden suponer peticiones al modelo o a las vistas.

Las ventajas de utilizar el patrón MVC son:

- Permitir la sustitución de las interfaces de usuario.
- Generar componentes de las interfaces.
- Diseñar vistas simultáneas del mismo modelo.
- Aplicar fácilmente cambios en las interfaces.

Además se identificación como problemas:

- La complejidad aumenta rápidamente.
- La vista están muy acopladas.

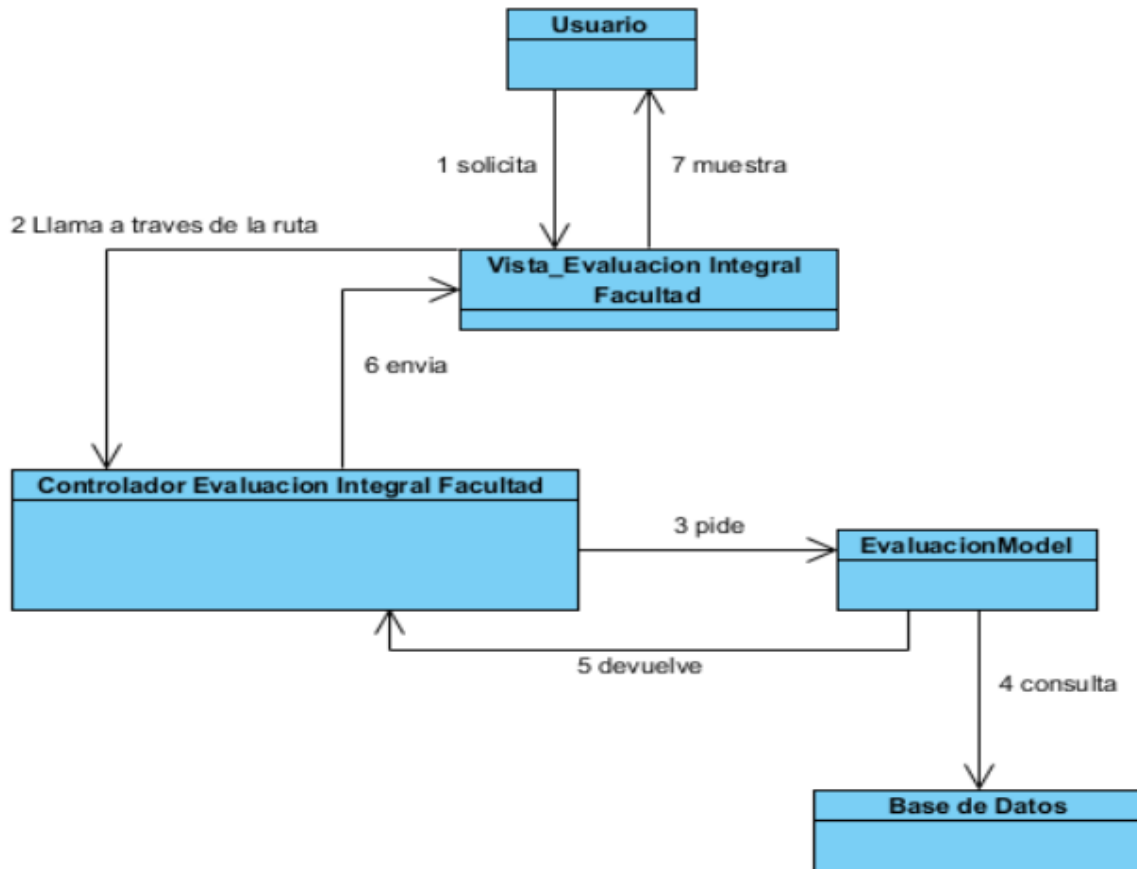


Ilustración 5 Modelo Vista Controlador. Fuente: Elaboración Propia.

### Patrones de Diseño

Los patrones de diseño son principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de software. Es una descripción de un problema y la solución a la que le da el nombre y que se

puede aplicar en nuevos contextos. Muchos patrones ayudan a asignar responsabilidades a los objetos (Olivares Rojas, s. f.).

Los patrones arquitectónicos ofrecen soluciones a problemas de arquitecturas de software dentro de la ingeniería de software. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software que consta de subsistemas, sus responsables e interrelaciones.

### **Patrones GRASP**

Lo esencial de un diseño de objetos lo constituye el diseño de las interacciones de objetos y la asignación de responsabilidades. Las decisiones que se tomen pueden influir profundamente en la extensibilidad, claridad y mantenimiento del sistema de software de objetos, además en el grado y calidad de los componentes reutilizables, por esta razón, durante el diseño se deben realizar los casos de usos con objetos basado en los patrones GRASP (Giraldo et al., 2011).

#### **Patrones GRASP utilizados en el Módulo de Gestión de la Evaluación Integral:**

Tanto en la clase Evaluación como en Grupos, los controladores EvaluacionController y GrupoController son los encargados de crear las instancias relacionadas con ellos (reportes, caracterizaciones, etc...), estando presente el patrón **Creador**. En estos controladores tenemos también el patrón **Experto**, encargado de conocer la cantidad de objetos de la clase y manejar datos que conciernen a estos. Además está presente **Alta Cohesión**, donde la Clase Evaluación está estrechamente relacionada con la clase Grupo, sus elementos están estrechamente relacionados. La clase Grupo es la encargada de gestionar los grupos docentes, que a su vez, contienen las evaluaciones integrales manejadas por la clase Evaluación, que permite la gestión de la autoevaluación de cada estudiante de cada grupo docente del año que le corresponde.

#### **Patrones GOF utilizados en el Módulo de Gestión de la Estrategia Educativa:**

**1-Flyweight:** Reduce la redundancia en gran manera ya que hay muchos objetos que manejan la misma información y la misma forma de procesar los datos.

```

public function store(Request $request)
{
    if($request->anho==1){
        $gruposTotal=GruposModel::all()->where('anho',1);
        for($i=0; $i<$gruposTotal->count(); $i++){
            if($gruposTotal[$i]['grupo']==$request->grupo){
                return back()->with('toast_error','Ya existe el grupo');
            }
        }
        $grupo=new GruposModel();
        $grupo->grupo=$request->grupo;
        $grupo->anho=$request->anho;
        $grupo->save();
        return redirect()->route('index.primer')->with('toast_success','Grupo añadido correctamente');
    }
}

```

El objeto grupo para cada clase, maneja los mismos datos y forma de procesarlos, únicamente se agrega un nuevo parámetro llamado “anho” para poder hacer las consultas para mostrar los grupos correspondientes a cada año docente.

```

$grupo=new GruposModel();
$grupo->grupo=$request->grupo;
$grupo->anho=$request->anho;

```

**2-Observador:** Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él (Giraldo et al., 2011).

```

$grupo=GruposModel::find($id);
$grupo->grupo=$request->grupo;
$grupo->anho=$request->anho;

```

Al actualizar un grupo, todas las actividades que estén vinculadas a él serán actualizadas inmediatamente y se notifica la modificación realizada.

## II.9 Diagrama de Modelo de Datos

La base de datos necesita una definición de su estructura, de manera que permita almacenar datos, reconocer el contenido y recuperar la información. Para el diseño correcto de una base de datos es necesario seguir un conjunto de pasos, comenzando por definir las clases persistentes, luego refinarlas y

clasificarlas junto con sus atributos, para luego realizar el diagrama de entidad-relación (SAP Insights, 2022).

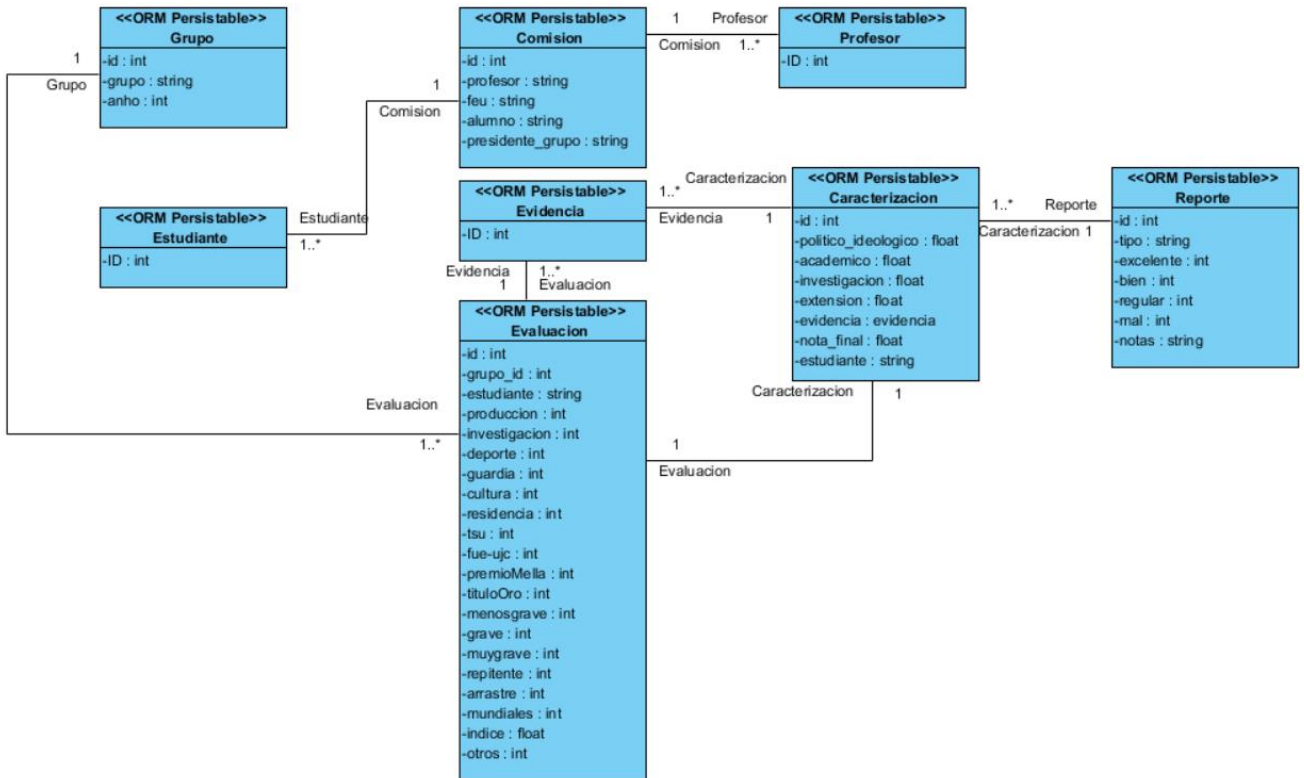


Ilustración 6 Diagrama de Modelo de Datos Módulo Evaluación Integrada. Fuente: Elaboración Propia.



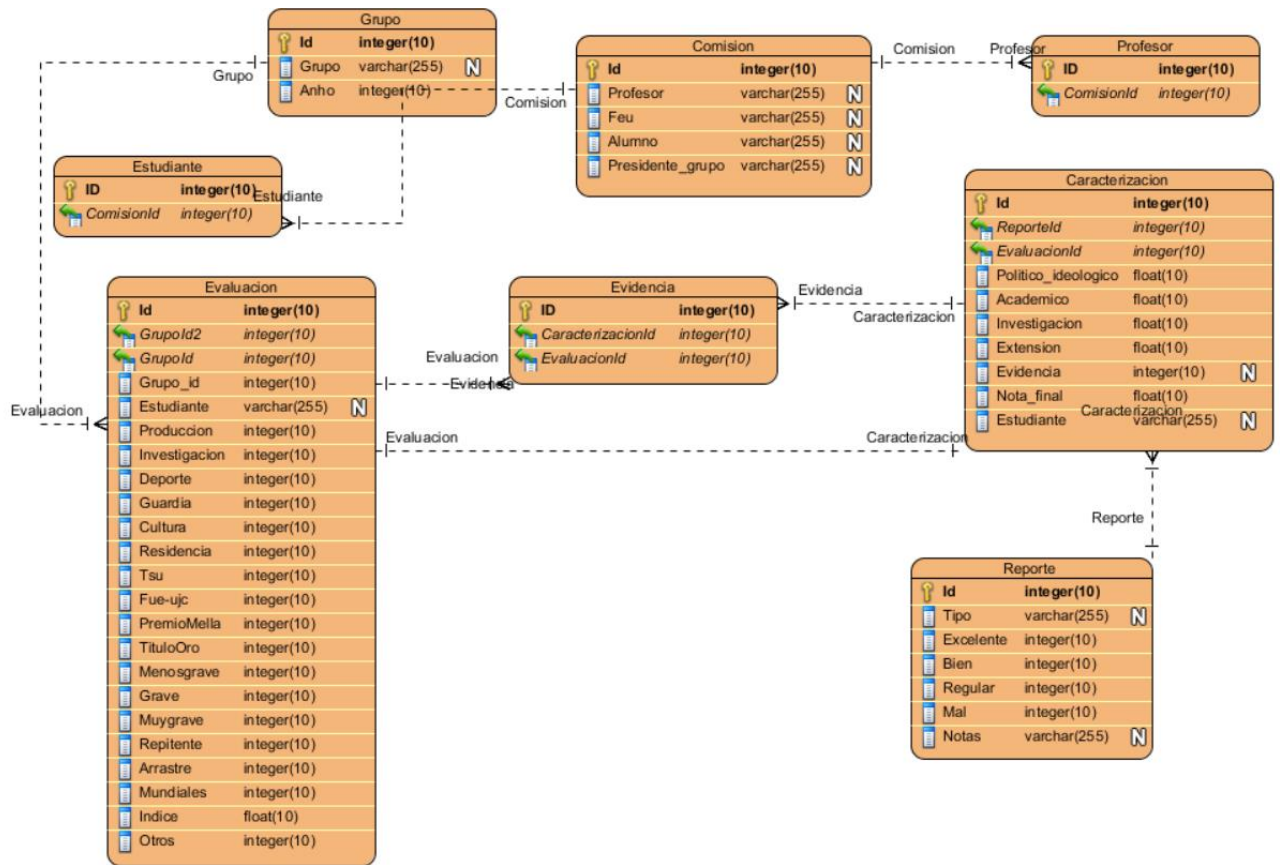
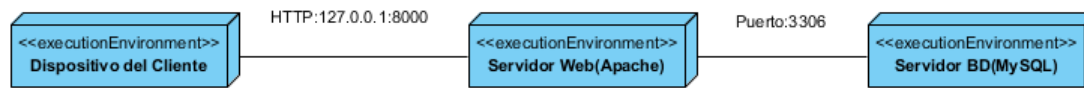


Ilustración 7 Diagrama Entidad Relación Módulo Evaluación Integrada. Fuente: Elaboración Propia.

## II.10 Diagrama de Despliegue

El diagrama de despliegue aparece cuando se modelan aspectos físicos de los sistemas orientados a objetos. Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Son utilizados para modelar la vista de despliegue estática de un sistema. Permite entender como el sistema se desplegará físicamente en el hardware (UNAD, 2022b).



*Ilustración 8 Diagrama de Despliegue. Fuente: Elaboración Propia.*

## II.11 Diagrama de Secuencia

El diagrama de secuencia UML representa los eventos en orden cronológico, razón por la que a veces se le llama diagrama de eventos o escenario de eventos. El orden (es decir, la secuencia exacta) es más importante que los puntos específicos en el tiempo. Sin embargo, es posible añadir restricciones al modelo con el que se está trabajando.

El diagrama de secuencia describe básicamente cómo los objetos (e instancias) intercambian mensajes en un orden determinado. Los objetos son los bloques de construcción básicos de los diagramas UML y representan ciertas características de un elemento del sistema, que varían dependiendo del diagrama. En las interacciones, los objetos son las conocidas como líneas de vida (IONOS Digital Guide, 2022a).

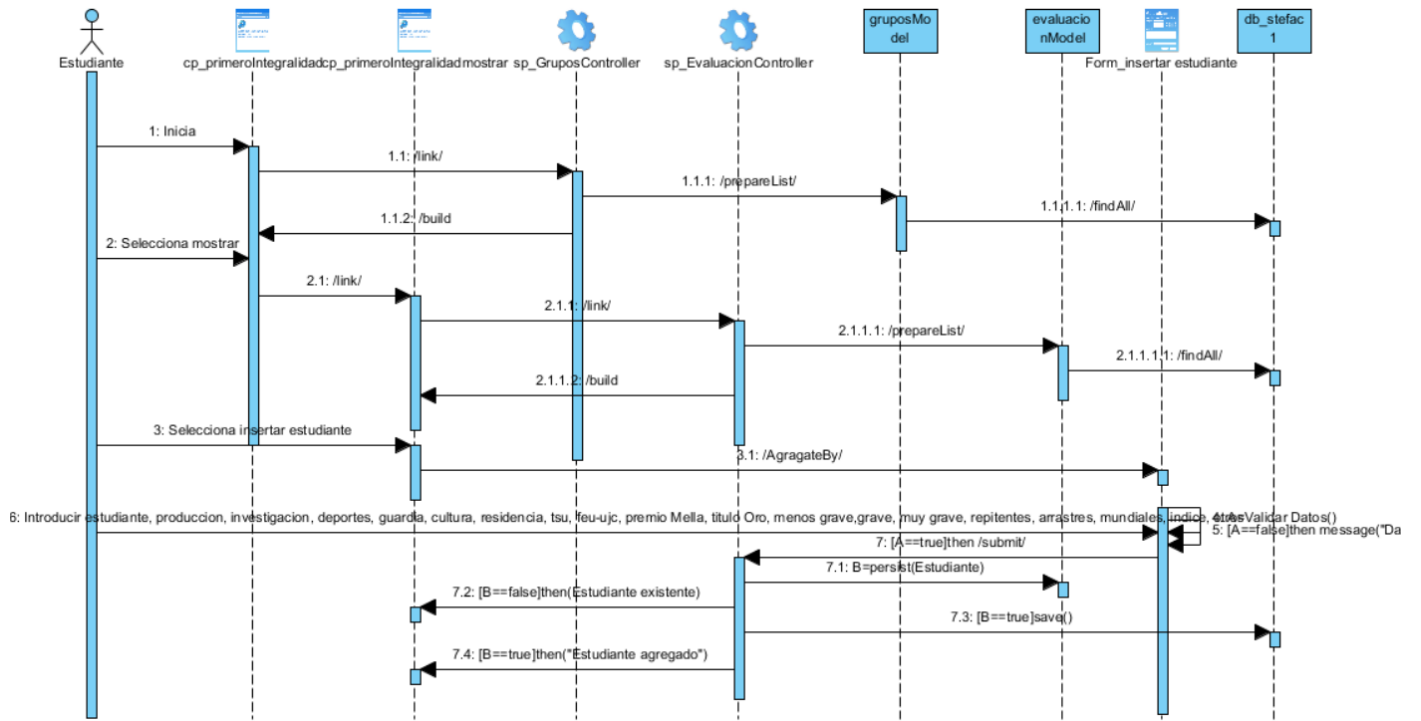


Ilustración 9 Diagrama de Secuencia Insertar Evaluación. Fuente: Elaboración Propia.

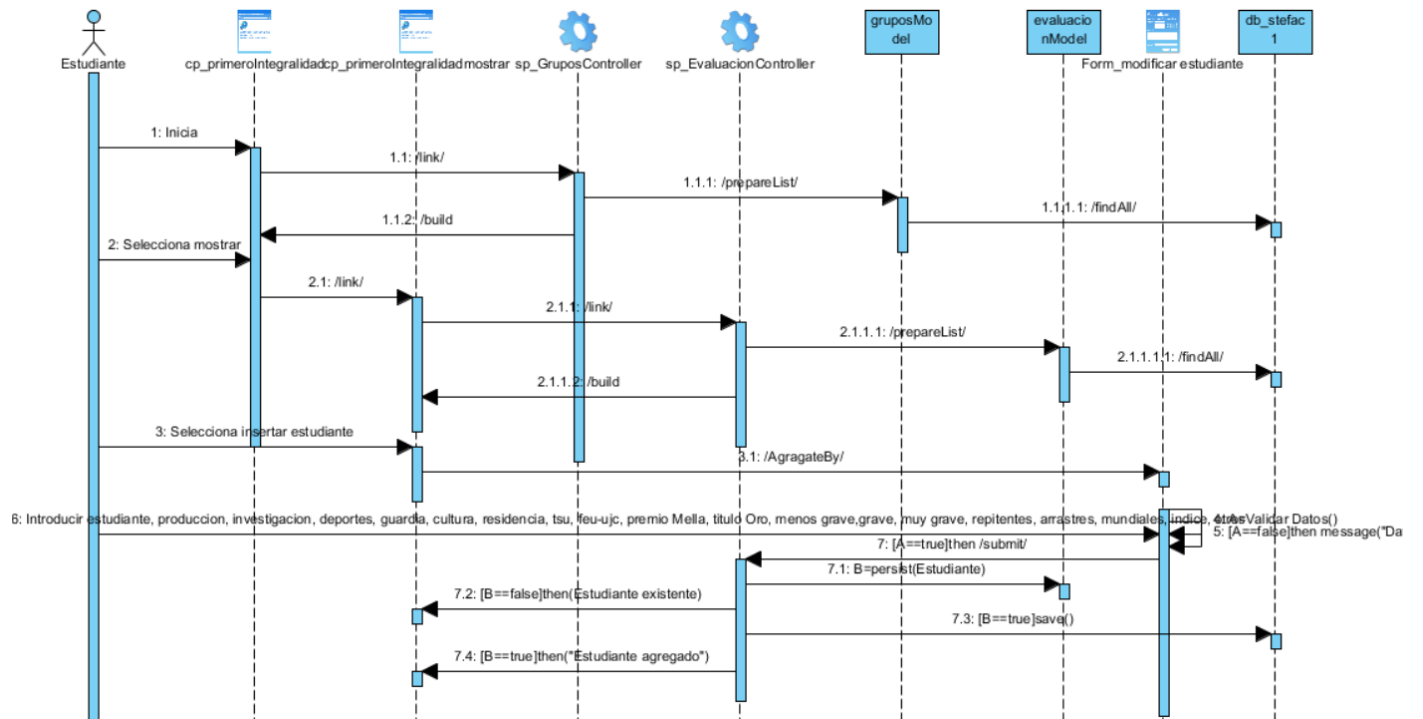


Ilustración 10 Diagrama de Secuencia Modificar Evaluación. Fuente: Elaboración Propia.

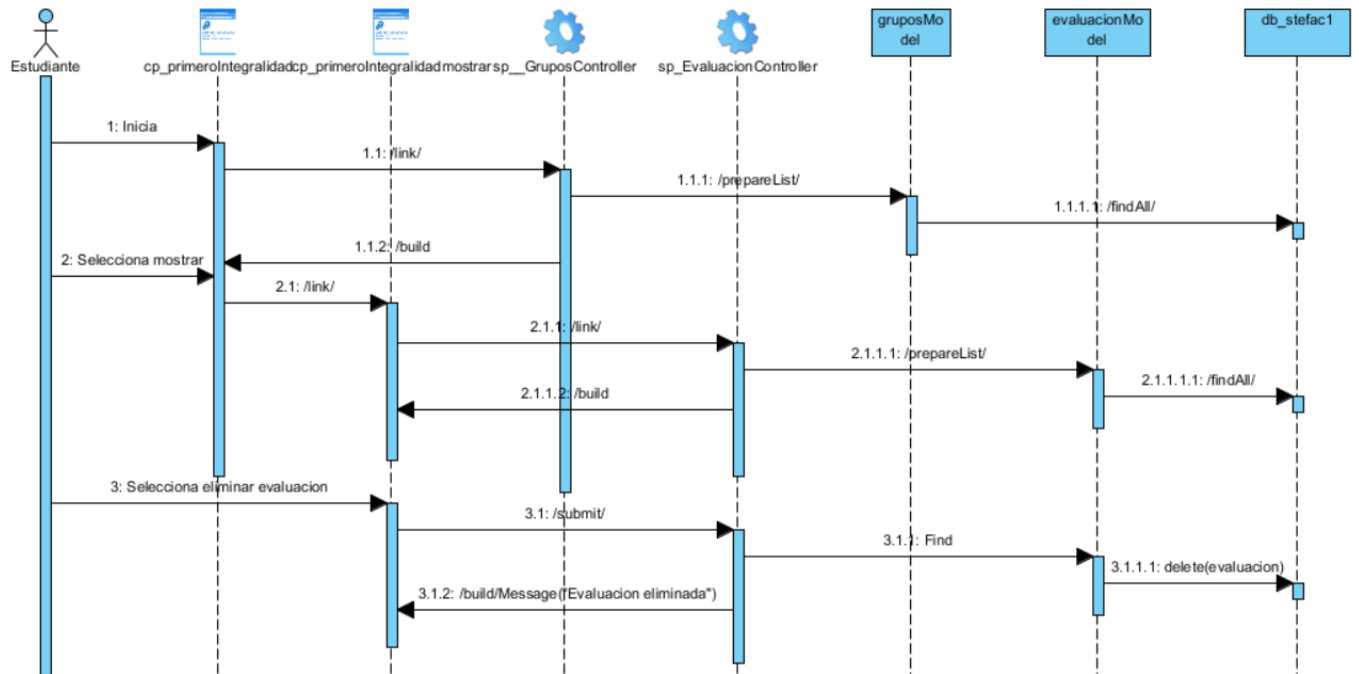


Ilustración 11 Diagrama de secuencia Eliminar Evaluación. Fuente: Elaboración Propia.

## Conclusiones del capítulo

Tras el desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- Los artefactos que se han generado de acuerdo a la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron la guía fundamental para la construcción de la propuesta de solución.
- A partir del proceso de identificación de los requisitos, los requisitos funcionales y no funcionales obtenidos fueron utilizados como guía para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades del cliente.
- Basado en el diagrama de despliegue propuesto se expuso de una manera más clara y precisa la estructura de los elementos de hardware y software del sistema permitiendo obtener una mejor comprensión del mismo

## CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se definen los componentes utilizados en la implementación y los paquetes en los que estará dividido el sistema. Además se exponen las pruebas realizadas al software para comprobar el buen funcionamiento de la aplicación, así como el cumplimiento de los requisitos aceptados por el cliente en las funcionalidades presentes en el sistema. Estas pruebas permiten evaluar la calidad del producto desarrollado y garantizar que el módulo cumpla todas las funcionalidades.

### III.1 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo (UNAD, 2022a).

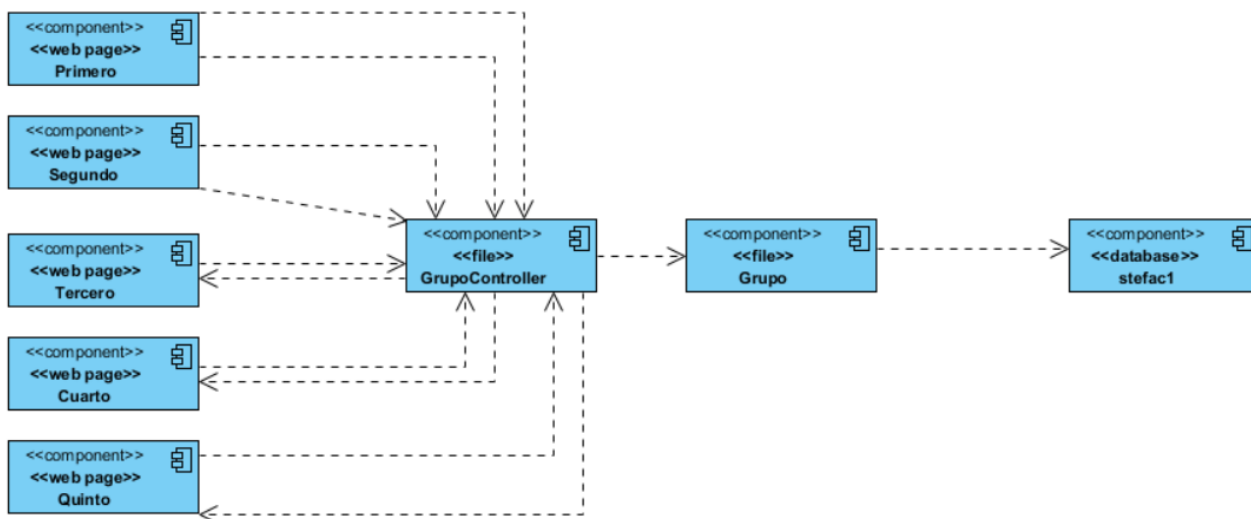


Ilustración 12 Diagrama de Componentes Grupos. Fuente: Elaboración Propia.

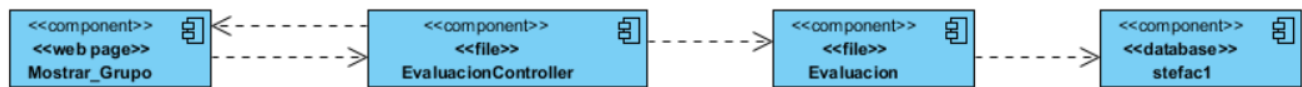


Ilustración 13 Diagrama de Componentes Evaluación. Fuente: Elaboración Propia.

### III.2 Estándares de Codificación

Son un punto de referencia para que todos los programadores dentro de un equipo sigan una misma línea de implementación y, de esta forma mejora la comunicación entre ellos. Estas buenas prácticas posibilitan que el código sea claro y fácil de entender y mantener. Los 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. En el desarrollo de aplicaciones web es importante que el código sea fácil de leer y modificar, para lo que se debe seguir una serie de normas comunes para todos los desarrolladores.

#### III.2.1 Estándares de Codificación para Laravel:

Los estándares de codificación incorporan principios de ingeniería sólidos para la programación en sus respectivos lenguajes y forman la base de cualquier enfoque preventivo (Uniwebsidad, 2022). Los estándares de codificación determinados por los desarrolladores de Laravel para la implementación de un sistema de modo que sea comprensible el código a otros programadores son:

**Controladores:** los nombres de los controladores se derivan del nombre del modelo, añadiendo el sufijo Controller, aplicando el estilo PascalCase.

Ejemplo:

```

<?php
namespace App\Http\Controllers\Modulo_EvaluacionIntegrada;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Modulo_EvaluacionIntegrada\GruposModel;
  
```

```
use App\Models\Modulo_EvaluacionIntegrada\EvaluacionModel;
class EvaluacionController extends Controller
{
```

**Funciones:** las funciones deben ser nombradas aplicando camelCase.

Ejemplo:

```
public function create($id)
{
```

**Modelos:** para nombrar los modelos se toma el nombre de la entidad en singular y se aplica PascalCase.

Ejemplo:

```
<?php

namespace App\Models\Modulo_EvaluacionIntegrada;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class ComisionModel extends Model
{
```

**Rutas:** las rutas deben cumplir el siguiente estándar, aplicando kebab-case.

Ejemplo:

```
Route::post('segundoIntegralidad/crearGrupo/Grupo', [GruposSegundoController::class,
'storeSegundo'])->name('store.segundo');
Route::get('segundoIntegralidad/modificarGrupo/{id}', [GruposSegundoController::class,
'editSegundo'])->name('edit.segundo');
```

**Llaves primarias y foráneas:** si no se aplica ninguna, Laravel asume por defecto que la llave primaria de la tabla es id. Las llaves foráneas se nombran como la entidad en singular, pero añadiendo el sufijo `_id`.

Ejemplo:

```
$table->bigIncrements('id');
$table->bigInteger('grupo_id')->unsigned();
```

**Variables:** las variables deben ser descriptivas. Estas deben aplicar el estilo camelCase.

Ejemplo:

```
$evaluacion
```



**Vistas:** para nombrar a las vistas en Blade se aplica kebab-case. Estas vistas deben terminar en .blade.php.

Ejemplo: Autoevaluacion.blade.php.

### III.2.2 Estándares de Codificación para PHP:

Indentación:

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos. En el siguiente ejemplo se muestra un fragmento de código con las indentaciones realizadas, de 2 espacios cada una, y los saltos de línea o enter al final de cada línea (sin dejar espacios)(PHP - Guía de estilo de codificación, s. f.).

```
public function indexPrimero(){  
  
    $grupos = GruposModel::all()->where('anho',1);  
  
    return view('Modulo_EvaluacionIntegrada.PrimeroCaract', compact('grupos'));  
}
```

Etiquetas de apertura y cierre de PHP

Cuando se utiliza el lenguaje PHP, siempre se deben utilizar las etiquetas <?PHP y ?>, y en ningún caso la versión corta <? y ?>. En general se omite la etiqueta de cierre de PHP (?>) al final de los archivos .module y .inc. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre ?>), que serían identificados como salida HTML y podrían provocar un error muy típico, "Cannot modify header information - headers already sent by..." (PHP - Guía de estilo de codificación, s. f.).

```
<?php
```

```
namespace App\Http\Controllers\Modulo_EvaluacionIntegrada;
```

```
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models Modulo_EvaluacionIntegrada\GruposModel;
```

Operadores:

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, \$numero = 3, en lugar de \$numero=3. Esto se aplica a operadores como +, -, \*, /, =, ==, !=, >, <, . (Concatenación de cadenas),. =, +=, -=, etc. Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, \$numero++(PHP - Guía de estilo de codificación, s. f.).

```
$grupos = GruposModel::all()->where('anho',1);
```

-----  
\$i++

Uso de comillas:

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\$title</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple(PHP - Guía de estilo de codificación, s. f.).

```
return redirect()->route('index.quinto')->with('toast_success','Grupo añadido correctamente');
```

Uso de punto y coma (;) en código PHP:

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo <?PHP print \$title ?>. En Laravel es siempre obligatorio: <?PHP print \$title; ?>(PHP - Guía de estilo de codificación, s. f.).

```
$grupo->save();
```

Estructuras de control:

Con respecto a las estructuras de control(PHP - Guía de estilo de codificación, s. f.), hay que tener en cuenta las siguientes normas:

Debe haber un espacio entre el comando que define la estructura (if, while, for, etc.) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones.

La llave de apertura {se situará en la misma línea que la definición de la estructura, separada por un espacio.

Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).

Las estructuras else y elseif se escribirán en la línea siguiente al cierre de la sentencia anterior.

```
public function storeQuinto(Request $request){
    if ($request->anho==5){
        $gruposTotal=GruposModel::all()->where('anho',5);
        for ($i=0; $i<$gruposTotal->count(); $i++){
            if ($gruposTotal[$i]['grupo']==$request->grupo){
                return back()->with('toast_error','Ya existe el grupo');
            }
        }
        $grupo=new GruposModel();
        $grupo->grupo=$request->grupo;
        $grupo->anho=$request->anho;
        $grupo->save();
        return redirect()->route('index.quinto')->with('toast_success','Grupo añadido correctamente');
    }
}
```

Arrays:

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.

Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos(PHP - Guía de estilo de codificación, s. f.).

Comentar el código

Para la realización de comentarios se emplea /\* \*/ para comentarios en varias líneas y // para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta deberá escribirse en mayúsculas (por ejemplo, TRUE o FALSE)(PHP - Guía de estilo de codificación, s. f.).

```
/**  
 * Store a newly created resource in storage.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @return \Illuminate\Http\Response  
 */
```

### **III.3 Verificación del Sistema**

La verificación y validación es el nombre que se da a los procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes. La V&V es un proceso de ciclo de vida completo. Inicia con las revisiones de los requerimientos y continúa con las revisiones del diseño y las inspecciones del código hasta la prueba del producto (Tes, 2020).

#### **III.3.1 Estrategia de Pruebas**

Según IEEE (2004), las pruebas de software "consisten en verificar el comportamiento de un programa dinámicamente a través de un grupo finito de casos de prueba, debidamente seleccionados del, típicamente, ámbito de ejecuciones infinito, en relación al comportamiento esperado". Contrario a lo que se pueda pensar una prueba exitosa es aquella en la que se consigue que el sistema falle y por tanto, se encuentran errores, aunque estos no sean todos los que posee el producto. De igual forma se puede considerar una prueba exitosa cuando es posible asegurar que no existen más errores en el sistema, cuestión que es mucho más difícil de garantizar. Este es un aspecto del que advierte muy a menudo la teoría de pruebas, ya que no es confiable asumir que se han superado todas las pruebas posibles que pueden efectuarse a un software (Jústiz-Núñez et al., 2014).

En la presente investigación se propone realizar pruebas de funcionalidad, pruebas de rendimiento y pruebas de seguridad. Para las pruebas de funcionalidad se genera como artefacto los Casos de Prueba. Las pruebas de rendimiento se efectúan a través de la herramienta Apache JMeter 5.4.1 se genera un reporte con los resultados de rendimiento. Las pruebas de seguridad se aplican mediante la herramienta Acunetix Web Scanner 11.0 y se genera un reporte con las vulnerabilidades encontradas.

### III.3.2 Pruebas Funcionales

Las pruebas funcionales son aquellas que se aplican al producto final, y permiten detectar en qué puntos el producto no cumple sus especificaciones, es decir, comprobar su funcionalidad. Para realizarlas se debe hacer una planificación que consiste en definir los aspectos a examinar y la forma de verificar su correcto funcionamiento, punto en el cual adquieren sentido los casos de prueba (González Palacio, 2009). Las pruebas de funcionalidad consisten en la revisión de los requisitos aceptados por el cliente contra las funcionalidades presentes en la aplicación. Con el objetivo de evaluar la calidad del producto desarrollado y garantizar que el portal diseñado e implementado cumpla con las funcionalidades, se elaboraron 26 casos de prueba basados en requisitos para realizar pruebas de caja negra. A continuación, se muestra los distintos escenarios del caso de prueba “Añadir Comisión”.

*Tabla 5 Caso de prueba basado en el requisito funcional Crear Comisión. Fuente: Elaboración Propia.*

No	Escenario	Descripción	Profesor	Presidente	Miembro FEU	Alumno Destacado	Respuesta del Sistema	Flujo Central
1	Crear Comisión de forma correcta	El módulo añade una comisión de forma correcta	V Disnayle Jorge Chacón	V Eddie Delgado Alfaro	V Eddie Delgado Alfaro	V Alejandro Santana Rodríguez	El módulo almacena la información y muestra un mensaje de confirmación de la acción	1. El usuario accede a la opción de Insertar Comisión. 2. Se muestra un formulario para que el usua-
2	Crear Comisión de forma	El módulo añade la comisión de	V Introducir	V Javier	V Gabriel	V Alejandro	El módulo no almacena la	

	incorrecta	forma incorrecta	un profesor que no esté registrado	Roque Sañudo	Negrín Yanes	Medero Acosta	información y muestra el mensaje: "El profesor es incorrecto".	rio introduzca la información.  3. El usuario introduce la información y presiona el botón "Guardar".
3	Crear Comisión con campos vacíos	El módulo no crea una comisión con campos obligatorios vacíos	V Aylin Estrada Velazco	V Adrián Lobaina	I Vacío	I Vacío	I Vacía	

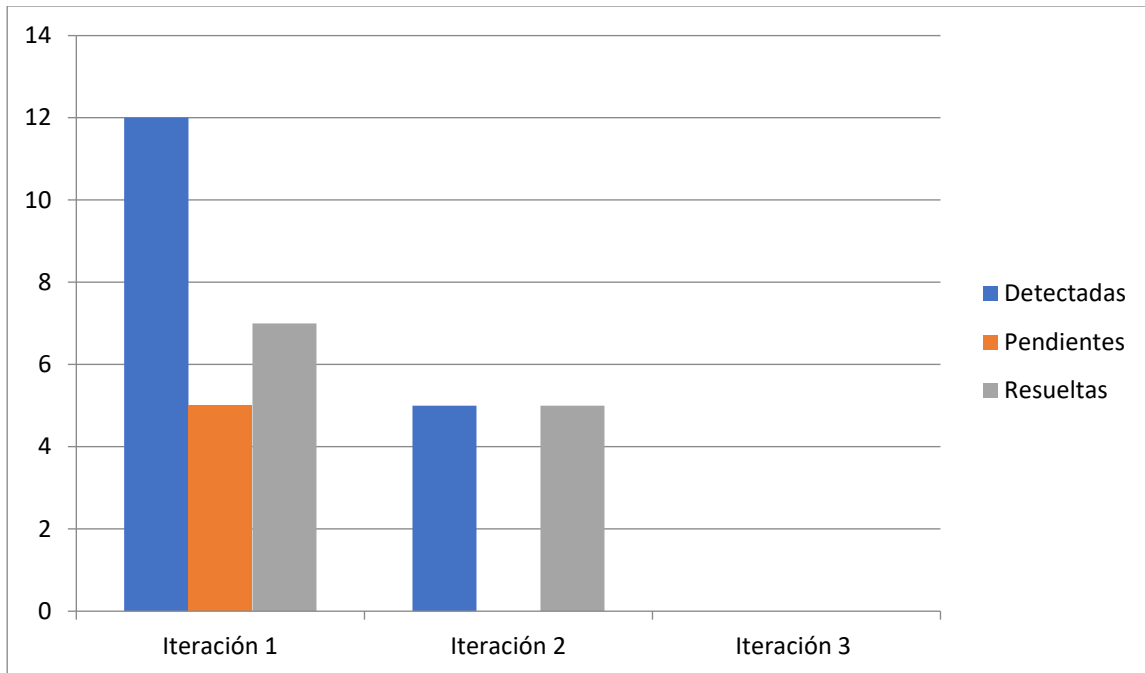
A continuación, se muestra un ejemplo de diseño de casos de prueba de aceptación utilizado para detectar errores en la aplicación y mostrar si cumplía con los requisitos. Se describe el caso de prueba "Añadir Comisión":

*Tabla 6 Aceptación en el caso de prueba Crear Comisión. Fuente: Elaboración Propia.*

**Caso de Prueba Añadir Comisión**

<b>Código de Caso de Prueba: 1</b>	<b>Nombre de Historia de Usuario: Añadir Comisión</b>
<b>Nombre de la persona que realiza la prueba: Stive Hernández Suarez</b>	
<b>Descripción de la prueba: Prueba a la funcionalidad Añadir Comisión</b>	
<b>Entrada-Pasos de Ejecución: La entrada consta de la introducción de los datos en los siguientes campos:</b>	
<ul style="list-style-type: none"> <li>• <b>Profesor: Disnayle Jorge Chacón</b></li> <li>• <b>Presidente: Eddie Delgado Alfaro</b></li> <li>• <b>Miembro FEU: Eddie Delgado Alfaro</b></li> <li>• <b>Alumno Destacado: Alejandro Santana Rodríguez</b></li> </ul>	
<b>Resultado Esperado: Se añade la comisión</b>	
<b>Evaluación de la Prueba: Satisfactoria</b>	

A continuación, se muestran los resultados obtenidos en tres iteraciones de pruebas realizadas al módulo de gestión, la Figura brinda información del total de no conformidades encontradas y las que fueron resueltas por cada iteración. Con un total de 26 requisitos funcionales se detectaron 12 no conformidades en la primera iteración resolviéndose satisfactoriamente 7, en la segunda iteración quedaron pendientes 5 no conformidades resolviéndose satisfactoriamente 5 y en la tercera iteración no se encontraron no conformidades.



*Ilustración 14 Resultado de las pruebas funcionales. Fuente: Elaboración Propia.*

**Entre las no conformidades detectadas en el proceso de pruebas funcionales se encuentran:**

- Errores de estructuración de los contenidos mostrados en las vistas.
- Errores de funcionamiento de algunas rutas: se arreglaron corrigiendo las direcciones de las vistas en los controladores.
- Errores ortográficos.
- Errores de funcionamiento con la conexión de la base de datos: se corrigieron algunos elementos en los modelos y las relaciones entre clases.
- El sistema muestra los mensajes en inglés.



### **III.3.3 Pruebas de Rendimiento**

Las pruebas de rendimiento de software se definen como un análisis de comprobación de funcionamiento del sistema frente a múltiples escenarios de ensayo, con el fin de examinar los componentes de la aplicación. Este tipo de pruebas se basan en garantizar la calidad y operatividad de un determinado sistema(KeepCoding, 2022).

Al módulo se le realizan pruebas de rendimiento de tipo carga y estrés. Se utiliza la herramienta Apache JMeter en su versión 5.4.1, diseñado para pruebas de cargas de comportamientos funcionales y la medición del rendimiento.

#### **Pruebas de carga**

Son un tipo de prueba de rendimiento donde se añaden cargas de trabajo a un sistema para verificar su comportamiento en condiciones de cargas normales y cuando se encuentra cargado al máximo de trabajo(KeepCoding, 2022).

#### **Pruebas de estrés**

Se refieren a las pruebas que miden el rendimiento del software partiendo de la sobrecarga de sus indicadores habituales de funcionamiento, lo que expone al sistema a un alto nivel de cargas de trabajo(KeepCoding, 2022).

Para realizar las pruebas de carga y estrés se utilizaron 2 PC, una como cliente y la otra como servidor, creando así un entorno de trabajo en el cual se efectuó un análisis del tráfico de usuarios en el sistema, estableciéndose como parámetros de muestra el caso crítico de 200 usuarios concurrentes y las siguientes condiciones:

#### **Hardware de prueba (PC servidor):**

- Sistema Operativo: Windows v10
- Microprocesador: Intel(R) Core (TM) i3-8130u CPU @2.20Ghz
- Memoria RAM: 4.0 GB

- Disco Duro: 1TB

**Hardware de prueba (PC cliente):**

- Sistema Operativo: Windows v10
- Microprocesador: Intel(R) Core (TM) i3-8130u CPU @2.20Ghz
- Memoria RAM: 4.0 GB
- Disco Duro: 1TB
- Tipo de Sistema: Sistema operativo de 64 bits, procesador x64

**Software instalado en ambas PC:**

- Servidor de Bases de Datos: MySQL v8.0.3 (PC cliente)
- Tipo de servidor web: Apache v2.4.46 (PC cliente)
- Plataforma: Sistema Operativo Windows (PC cliente y PC servidor)

Habiendo definido el hardware en el que va a estar implementado el módulo, se configuran los parámetros de Apache JMeter.

*Tabla 7 Resultado de Pruebas de Carga y Estrés. Fuente: Elaboración Propia.*

Usuarios	Muestras	Media	Min	Max	Mediana	Linea de 90%	% Error	Rendimiento	Kbs/sec
1	91	364	1	2799	21	3045	0,00%	4,1/sec	73,9
200	3200	9467	2588	43233	165	23523	0,00%	6,2/sec	90,2

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

**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.

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

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

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

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

**Rendimiento (peticiones/segundos):** El rendimiento se mide en cantidad de solicitudes por segundo.

**Kb/s recibidos:** Velocidad de carga de las páginas.

### **Análisis de los resultados de las pruebas de rendimiento**

El informe de Apache JMeter arroja que el módulo es capaz de responder 3200 peticiones de 200 usuarios conectados simultáneamente en un tiempo promedio de 9467 milisegundos con un error de 0.0%. Esto evidencia que el módulo puede responder correctamente las peticiones realizadas.

### **III.3.4 Pruebas de Seguridad**

Las pruebas de seguridad es un tipo de prueba de software que revela vulnerabilidades, amenazas y riesgos en una aplicación de software. El propósito de las pruebas de seguridad es identificar todas las lagu-

nas y debilidades potenciales del sistema que podrían resultar en pérdida de información, ingresos, u otros, así como previene ataques maliciosos de intrusos(Menejías García et al., 2021).

### **Resultados de las pruebas de seguridad**

Con el objetivo de evaluar la seguridad del módulo se emplea la herramienta Acunetix Web Vulnerability Scanner v11.0. Los resultados arrojados fueron los siguientes:

Primera Iteración: 53 no conformidades:

- 14 de nivel medio
- 29 de nivel bajo
- 10 de nivel informacional.

#### **Nivel medio:**

Pérdida del control de acceso: el usuario puede actuar fuera de los roles y permisos preestablecidos al acceder a determinados lugares sin restricciones.

Fallos criptográficos: ciertos datos como credenciales de accesos e información confidencial de los usuarios no se encontraban cifrados. Se hace necesaria la aplicación de un cifrado con algoritmos y protocolos estándares y robustos.

#### **Nivel bajo:**

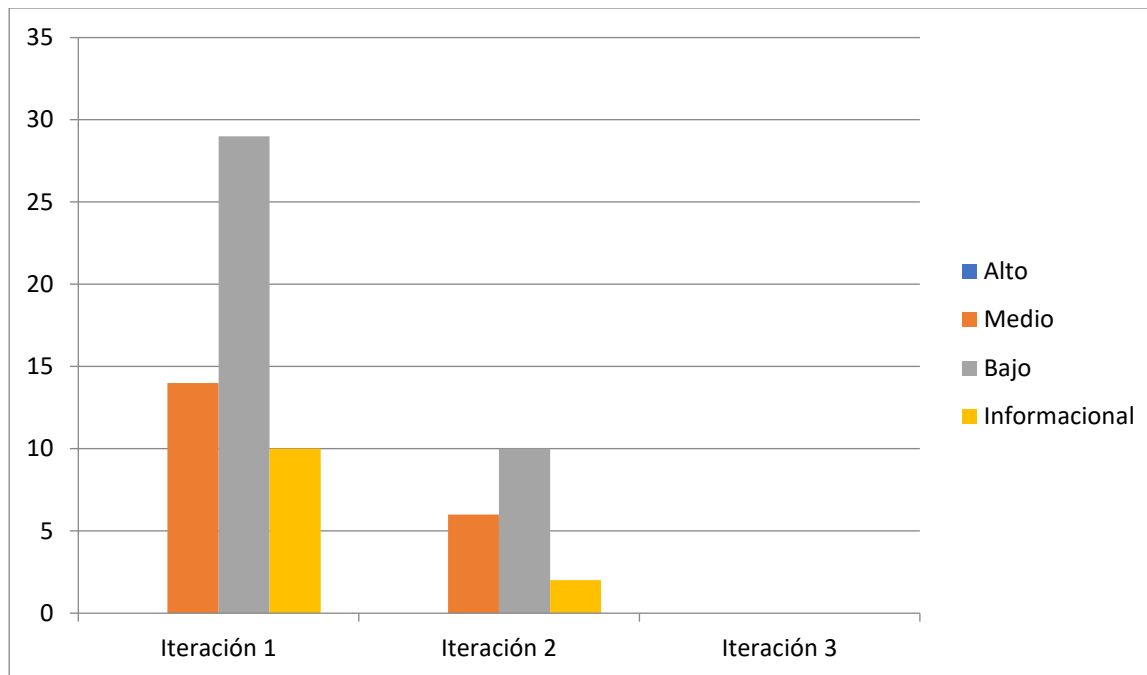
Fallos de identificación y autenticación: posibles ataques de delincuentes cibernéticos debido al no control en las interfaces de acceso del número de intentos de autenticación y a una baja complejidad de las contraseñas.

#### **Nivel informacional:**

Visibilidad de los botones en la página: algunos botones no están en lugares que sean de usual utilización por parte de los usuarios.

Errores ortográficos.

Fueron corregidas en una primera iteración 35 no conformidades de forma satisfactoria, 18 no conformidades en una segunda y en una tercera iteración no se identificaron nuevas no conformidades, garantizando así que el módulo cumpla con los requisitos de seguridad. Se representa a continuación los resultados descritos anteriormente.



*Ilustración 15 Resultado de las pruebas de seguridad. Fuente: Elaboración Propia.*

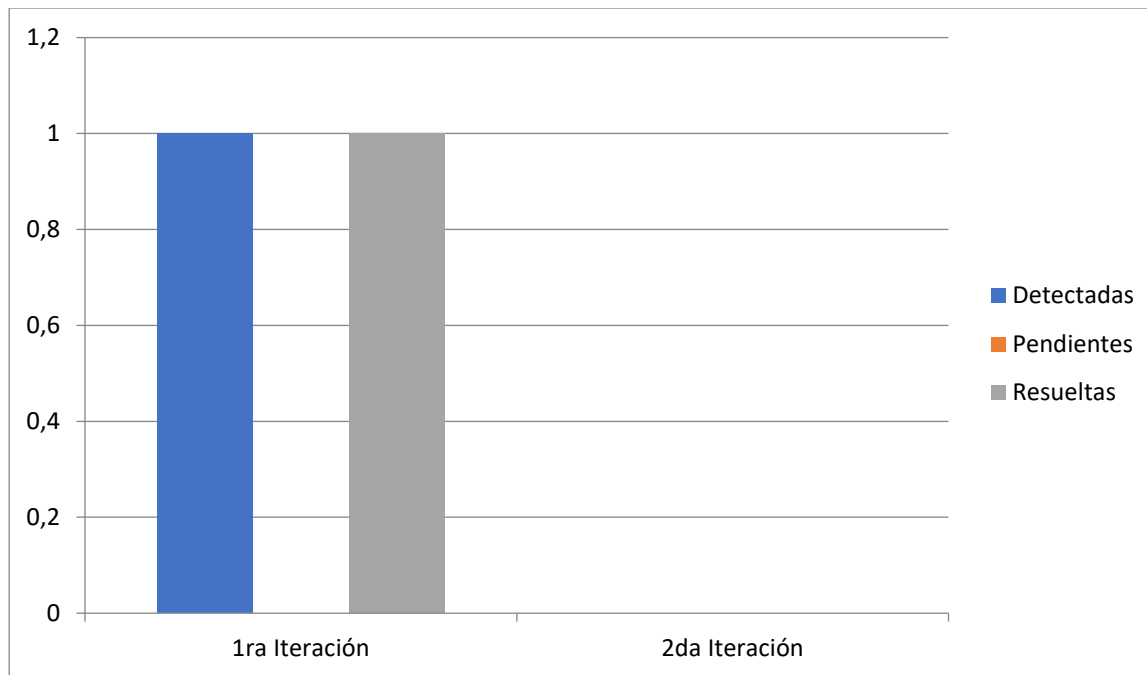
### III.3.5 Satisfacción de usuarios con el módulo

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Estas pruebas validan que todos los componentes funcionen juntos, se comuniquen correctamente y, además, transfieren todos los datos adecuados en el tiempo preciso y por las vías de comunicación establecidas (Jiménez Caballero & Brito Palacio, 2021).

En 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 ejecutaron las siguientes acciones:

- Integración del Módulo de Gestión de Perfil de Usuario
- Integración del Módulo de Generación de Horarios Docentes
- Integración del Módulo de Sistema de Trabajo Educativo
- Integración del Módulo de Evaluación Integrada

En la siguiente gráfica se observan los resultados tras 2 iteraciones de pruebas de integración realizadas al sistema:



*Ilustración 16 Resultado de las Pruebas de Integración*

Tras la conclusión del proceso de pruebas de integración se comprobaron de manera conjunta todos los componentes del Sistema de Trabajo Educativo de la Facultad 1, lo que permitió realizar varias interacciones del sistema en total integración entre los módulos y se comprobó el correcto funcionamiento de los mismos. En estas pruebas se obtuvo una no conformidad debido a que todo el sistema funciona a través de una única base de datos, y por lo tanto tenían que acoplarse al modelo de datos de la presente información.

Con la solución de la no conformidad encontrada se concluye que el proceso de integración transcurrió de forma satisfactoria y el sistema se encuentra funcionando de forma correcta.

### **III.3.6 Satisfacción de usuarios con el módulo**

La técnica de ladov constituye 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 3 preguntas cerradas que se intercalan dentro de un cuestionario, y dos abiertas. Estas 3 preguntas se relacionan a través de lo que se denomina "Cuadro Lógico de ladov" e indica la posición de cada sujeto en la escala de satisfacción (Sarria & Paez, 2021).

En la investigación, la técnica de ladov fue empleada para medir la satisfacción de los usuarios, aquellos que interactúan con el módulo y acceden a sus contenidos y servicios. Para el desarrollo de esta técnica se aplicó una encuesta después de interactuar con el sitio web desarrollado.

*Tabla 8 Cuadro Lógico de ladov para usuarios potenciales. Fuente: Elaboración Propia.*

	¿Considera usted que la Facultad 1 de la UCI necesita un módulo para la gestión de la evaluación integral en sus estudiantes?								
	No			No Sé			Sí		
	¿Utilizaría usted un portal que facilite su proceso de evaluación integral?								
¿Le satisface el módulo desarrollado para la gestión de la información a partir del diseño de interfaz de usuario y la arquitectura de la información?	Sí	No Sé	No	Sí	No Sé	No	Sí	No Sé	Sí
Me satisface mucho	3	1	1	2	1	3	4	3	2
No me satisface tanto	4	3	2	2	4	1	1	3	5
Me da lo mismo	2	4	3	4	4	2	4	1	2
Me insatisface más de lo que me satisface	2	1	2	2	1	2	5	3	2
No me satisface nada	4	2	1	1	1	2	1	4	2
No sé qué decir	5	3	3	3	1	2	1	1	4

La escala de satisfacción se muestra a continuación:

Clara satisfacción

Más satisfecho que insatisfecho

No definida

Mas insatisfecho que satisfecho

Clara insatisfacción

Contradictoria

Para medir el grado de satisfacción se tomó una muestra de 5 usuarios que mantuvieran acceso a los portales web de la UCI. El resultado obtenido se muestra a continuación:

*Tabla 9 Satisfacción de Usuarios Potenciales. Fuente: Elaboración Propia.*

Resultado	Cantidad	Por ciento
Máximo de satisfacción	3	60%
Más satisfecho que insatisfecho	1	20%
No definida	1	20%
Más insatisfecho que satisfecho	0	0%
Clara insatisfacción	0	0%
Contradictoria	0	0%

Con el objetivo de obtener el índice de satisfacción grupal (ISG) 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:

*Tabla 10 Escala Numérica para el ISG. Fuente: Elaboración Propia.*

Escala	Nivel de satisfacción
1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Luego es posible calcular el ISG a través de la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N} \quad (1)$$

Dónde:

A, B, C, D, E, representan el número de sujetos con su índice individual.

N representa el número total de sujetos del grupo.

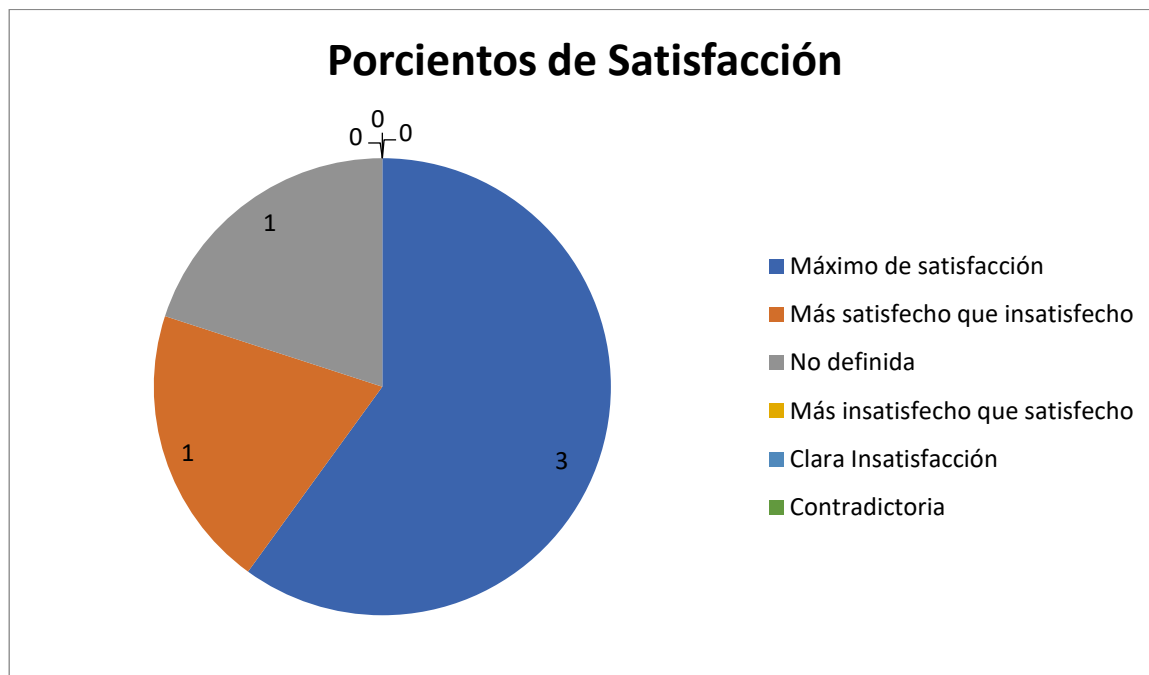
El ISG obtenido para los usuarios potenciales fue el siguiente:

$$ISG = \frac{3(+1) + 1(+0,5) + 1(0) + 0(-0,5) + 0(-1)}{5} = 0,7$$



Como se aprecia, el índice de satisfacción grupal es de lo que significa una clara satisfacción con la propuesta y reconocimiento de su utilidad en el mejoramiento de lo planteado en el problema de la investigación.

A continuación se representan los porcentajes de satisfacción obtenidos luego de calculado el ISG, no encontrándose los 3 últimos colores por estar en 0%.



*Ilustración 17 Satisfacción de Usuarios Potenciales. Fuente: Elaboración Propia.*

### Conclusiones del capítulo

- La confección del diagrama de componentes permite visualizar la integración de los componentes de software del módulo.
- Los estándares de codificación facilitan la comprensión y legibilidad del código, haciéndolo seguro, confiable, permitiendo desarrollar el sistema en un tiempo óptimo y garantizando facilidad y calidad de un futuro mantenimiento.

- La realización de las pruebas de funcionalidad, rendimiento y seguridad permitió la obtención de un sistema más robusto y seguro. Fueron resueltas las no conformidades de las pruebas funcionales, se constató en las pruebas de rendimiento que el sistema es capaz de responder correctamente a las peticiones realizadas y se solucionaron todas las no conformidades que arrojaron las pruebas de seguridad.

## CONCLUSIONES FINALES

- Con la definición de conceptos asociados al dominio del sistema, de los sistemas homólogos nacionales y extranjeros, de las herramientas y tecnologías y de la metodología de desarrollo se obtuvo una base sólida para sustentar la investigación en cuestión.
- Tras el análisis del estudio realizado a los procesos actuales se definió para mejorar la gestión del proceso de evaluación integrada de los estudiantes de la Facultad 1 el desarrollo de un módulo integrado al sistema de Trabajo Educativo.
- Con el proceso de identificación de requisitos funcionales y no funcionales, la definición de la arquitectura, patrones de diseño, artefactos de modelación y estándares de codificación se pudo elaborar una solución que satisface las necesidades del cliente.
- La realización de pruebas de funcionalidad, rendimiento y seguridad permitió obtener un sistema más robusto y seguro, y se solucionaron las no conformidades encontradas de forma satisfactoria, para obtener un producto con la calidad requerida.

## **RECOMENDACIONES**

Después de concluida la investigación y ser desarrollada la propuesta de solución se recomienda al lector seguir investigando acerca del proceso de gestión de la Evaluación Integral en el Ministerio de Educación Superior y en la Universidad de las Ciencias Informáticas para poder explorar en futuras funcionalidades que puedan ser añadidas y así brindar una mejor gestión y seguridad de la información, para aumentar el grado de satisfacción de los usuarios y tener un producto más potente y capaz de dar solución a los problemas que puedan presentarse.

## REFERENCIAS BIBLIOGRÁFICAS

- Acevedo, Ms. I. N. R., Acosta, D. C. A. M., & Nazco, D. C. M. A. (2017). La evaluación integral del estudiante universitario como vía para estimular el protagonismo en su formación. *Universidad y Sociedad*, 9(5), 171-180.
- Aguilera Alvarez, D., & Rodríguez Rodríguez, A. (2020). Sistema para la Evaluación Integral Estudiantil. *Innovación y Software*, 1(2), 52-61.
- Alarcón Ortiz, R. A., Guzmán Mirás, Y., García González, M., Alarcón Ortiz, R. A., Guzmán Mirás, Y., & García González, M. (2019). Formación integral en la educación superior: Una visión cubana. *Revista Estudios del Desarrollo Social: Cuba y América Latina*, 7(3).  
[http://scielo.sld.cu/scielo.php?script=sci\\_abstract&pid=S2308-01322019000300010&lng=es&nrm=iso&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S2308-01322019000300010&lng=es&nrm=iso&tlng=es)
- Berrones, H., & Alfonso, E. (2020). *Desarrollo de una aplicación web con el Framework Bootstrap y el pre-compilador Sass para la gestión de pedidos de productos agrícolas de la Empresa El Chagra*.  
<http://dspace.esPOCH.edu.ec/handle/123456789/13654>
- Delgado Dapena, M. (2002). MODELO CONCEPTUAL DE UN PROYECTO DE SOFTWARE UTILIZANDO EL RAZONAMIENTO BASADO EN CASOS. *Ingeniería Industrial*, 23.
- Deymar, A. (2022, junio 3). *¿Qué es Bootstrap? - Todo lo que necesitas saber*.  
<https://www.hostinger.es/tutoriales/que-es-bootstrap>
- Dinahosting. (2019, enero 8). *¿Qué es Apache y para qué sirve? Ayuda | dinahosting*.  
<https://dinahosting.com/ayuda/que-es-apache-y-para-que-sirve/>
- Documentation for Visual Studio Code*. (s. f.). Recuperado 9 de junio de 2022, de  
<https://code.visualstudio.com/docs>

- Fernandez, Y. (2021, enero 13). *Qué es el HTML5 y qué novedades ofrece*.  
<https://www.xataka.com/basics/que-html5-que-novedades-ofrece>
- Gaceta Oficial de la República de Cuba. (2021, enero 8). *Constitución de la República de Cuba proclamada el 10 de abril de 2019* [Text]. Gaceta Oficial. <https://www.gacetaoficial.gob.cu/es/constitucion-de-la-republica-de-cuba-proclamada-el-10-de-abril-de-2019>
- Garcia de Zuñiga, F. (s. f.). *¿Qué es phpMyAdmin y cómo usarlo? - Blog de arsys.es*. Recuperado 20 de noviembre de 2022, de <https://www.arsys.es/blog/phpmyadmin>
- García-Holgado, A., Vázquez-Ingelmo, A., & García-Peñalvo, F. J. (2022). *Ingeniería de requisitos*.  
<https://repositorio.grial.eu/handle/grial/2515>
- Giraldo, G., Gloria, L., Acevedo, O., Juan, F., Moreno, N., & Rodríguez Peña, A. (2011). Una ontología para la representación de conceptos de diseño de software. 2011. *Goc-2019-o2.pdf*. (s. f.). Recuperado 10 de junio de 2022, de  
<https://www.gacetaoficial.gob.cu/sites/default/files/goc-2019-o2.pdf>
- Gonzalez Flores, I., Cruzata de la Cruz, C. M., & Medina Miguel, V. (2017). *UNA CONTRIBUCIÓN A LA GESTIÓN DE LA INFORMACIÓN DE CIENCIA, TECNOLOGÍA E INNOVACIÓN*.  
<https://www.redalyc.org/journal/5257/525754432004/html/>
- González Palacio, L. (2009). Método para generar casos de prueba funcional en el desarrollo de software. *Revista Ingenierías Universidad de Medellín*, 8(15), 29-36.
- Guzmán Mirás, Y., & García González, M. (2017). La competencia educativa del coordinador de año en el contexto de la nueva universidad cubana. *Revista Cubana de Educación Superior*, 36(1), 127-136.
- IONOS Digital Guide. (2022a). *Diagramas de secuencia: Mostrar interacciones con UML*. IONOS Digital Guide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-secuencia/>

- IONOS Digital Guide. (2022b). *El diagrama de casos de uso en UML*. IONOS Digital Guide.  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>
- Islas Hernandez, L. (2021). 2.4.1 *Diagrama de clases*.  
[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro10/241\\_diagrama\\_de\\_clases.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro10/241_diagrama_de_clases.html)
- Jiménez Caballero, L., & Brito Palacio, Y. (2021). *Sistema de Gestión del proceso de ejercicios de culminación de estudios de la Facultad 1*.
- Jústiz-Núñez, D., Gómez-Suárez, D., & Delgado-Dapena, M. D. (2014). Proceso de pruebas para productos de software en un laboratorio de calidad. *Ingeniería Industrial*, 35(2), 131-145.
- KeepCoding, R. (2022, agosto 8). *¿Qué son las pruebas de rendimiento?* <https://keepcoding.io/blog/que-son-las-pruebas-de-rendimiento/>
- Kinsta. (2022, noviembre 13). *El Framework PHP Laravel—Construcción de Aplicaciones Web para Todos*. Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-laravel/>
- Menejías García, R., Hidalgo Reyes, N. H., Marín Díaz, A., Trujillo Casañola, Y., Menejías García, R., Hidalgo Reyes, N. H., Marín Díaz, A., & Trujillo Casañola, Y. (2021). Procedimiento para evaluar seguridad a productos de software. *Revista Cubana de Ciencias Informáticas*, 15(4), 333-349.
- Ministerio de Educacion Superior. (s. f.). *Resolución No. 2-2018.- Reglamento de Trabajo Docente y Metodológico de la Educación Superior*. VLex. Recuperado 10 de junio de 2022, de <https://cuba.vlex.com/vid/resolucion-no-2-2018-729534989>
- Modelo vista controlador (MVC)*. (s. f.). Recuperado 18 de septiembre de 2022, de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Olivares Rojas, M. C. J. C. (s. f.). *Patrones de Diseño*.
- Pacheco Tejeda, J. (2021, diciembre 9). LA EVALUACIÓN DE LA FORMACIÓN INTEGRAL. *Universidad Católica San Pablo*. <https://ucsp.edu.pe/evaluacion-formacion-integral/>

- Perez Porto, J., & Gardey, A. (2018, septiembre 19). *Definición de JavaScript—Qué es, Significado y Concepto*. <https://definicion.de/javascript/>
- PHP - *Guía de estilo de codificación*. (s. f.). Recuperado 3 de noviembre de 2022, de <http://coppeldev.github.io/php/standards/psr-2.html>
- Pressman, R. S. (2002). *Ingeniería de Software, un enfoque práctico*. [https://www.ecured.cu/Visual\\_Paradigm](https://www.ecured.cu/Visual_Paradigm)
- Robledano, A. (2019, septiembre 24). *Qué es MySQL: Características y ventajas | OpenWebinars*. <https://openwebinars.net/blog/que-es-mysql/>
- Rodríguez Peña, A. D., & Silva Rojas, L. G. (2016). Arquitectura de software para el sistema de visualización médica Vismedic. *Revista Cubana de Informática Médica*, 8(1), 75-86.
- Ruiz Suarez, J. F. (2019). *LAS TIC EN LA GESTIÓN UNIVERSITARIA CUBANA: BARRERAS, PRINCIPIOS, ACCIONES ICT IN CUBAN UNIVERSITY MANAGEMENT: BARRIERS, PRINCIPLES, ACTIONS - PDF Free Download*. <https://docplayer.es/98656314-Las-tic-en-la-gestion-universitaria-cubana-barreras-principios-acciones-ict-in-cuban-university-management-barriers-principles-actions.html>
- SAP Insights. (2022). *¿Qué es el modelado de datos? | Definición, importancia y tipos | SAP Insights*. <https://www.sap.com/latinamerica/insights/what-is-data-modeling.html>
- Sarria, Y. M., & Paez, C. C. T. (2021). Validación del modelo de gestión pública de la calidad de vida en Cuba. *Cooperativismo y Desarrollo*, 9(3), 919-934.
- Sheti, S. (2022, octubre 12). *Apache Jmeter: Todo lo que necesita saber—Geekflare*. <https://geekflare.com/es/apache-jmeter-guide/>
- Tapia, N. (2022, febrero 25). *¿Qué es el lenguaje PHP: Hypertext Preprocessor » BaulPHP*. <https://www.baulphp.com/que-es-el-lenguaje-php-hypertext-preprocessor/>



Tes. (2020, octubre 22). *Verificación vs validación en software integrado*. Parasoft.

<https://es.parasoft.com/blog/verification-vs-validation-in-embedded-software/>

UNAD. (2022a). *Diagramas de Componentes | LENGUAJE DE MODELADO UNIFICADO - UML*.

[https://unadzsurlab.com/UML/U2/diagramas\\_de\\_componentes.html](https://unadzsurlab.com/UML/U2/diagramas_de_componentes.html)

UNAD. (2022b, septiembre 18). *Diagramas de Despliegue | LENGUAJE DE MODELADO UNIFICADO -*

*UML*. [https://unadzsurlab.com/UML/U2/diagramas\\_de\\_despliegue.html](https://unadzsurlab.com/UML/U2/diagramas_de_despliegue.html)

Uniwebsidad. (2022). *Capítulo 2. Guía de Estilo de Código (PHP, la manera correcta)*.

<https://uniwebsidad.com/libros/php-correcto/capitulo-2>

Wodyk, R., & Skublewska-Paszkowska, M. (2020). Performance comparison of relational databases SQL

Server, MySQL and PostgreSQL using a web application and the Laravel framework. *Journal of*

*Computer Sciences Institute*, 17, 358-364. <https://doi.org/10.35784/jcsi.2279>

## ANEXOS

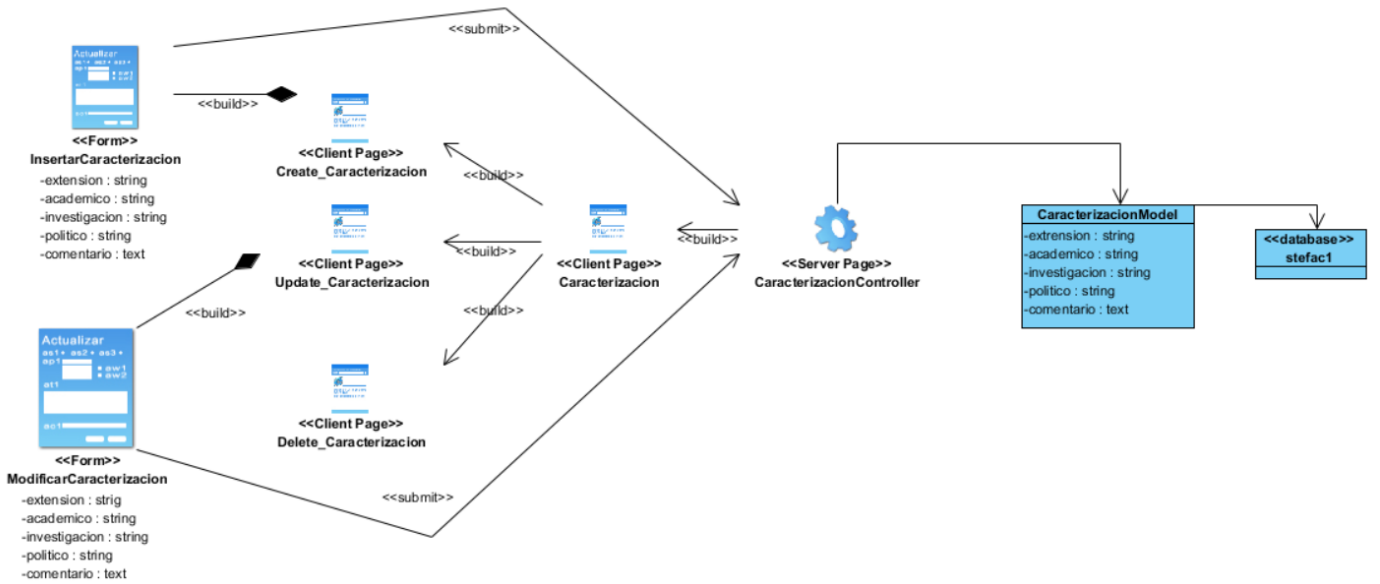


Ilustración 18 Diagrama de Clases Caracterización

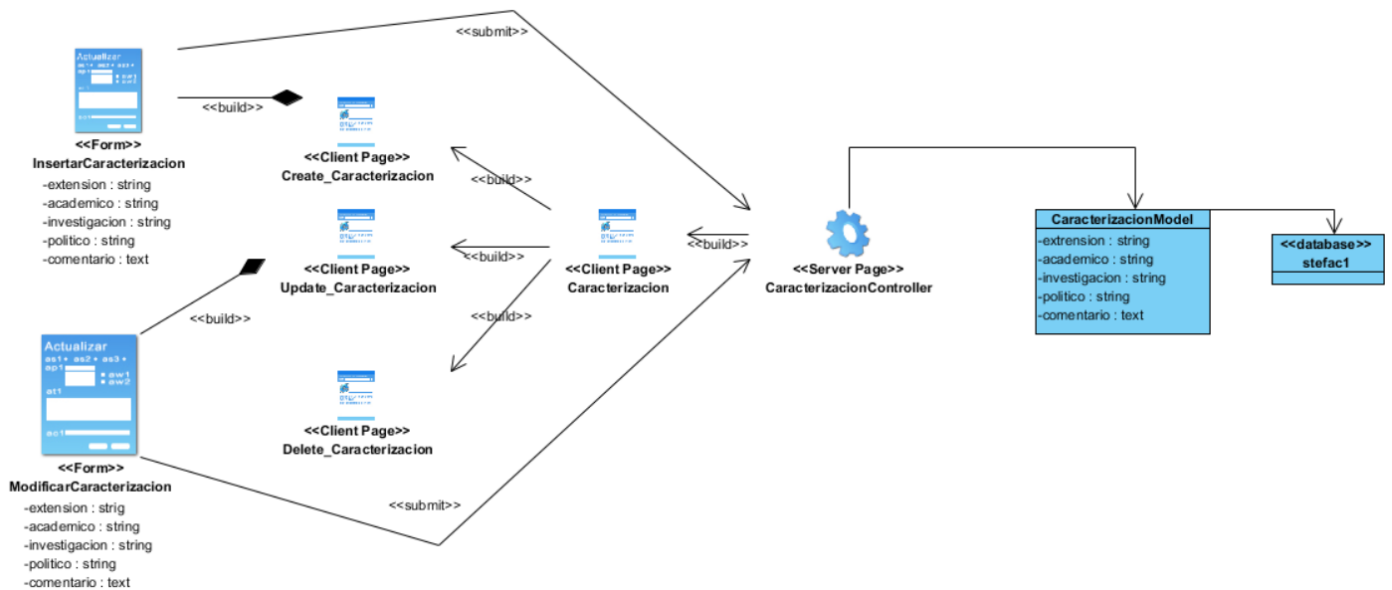


Ilustración 19 Diagrama de Clases Comisión

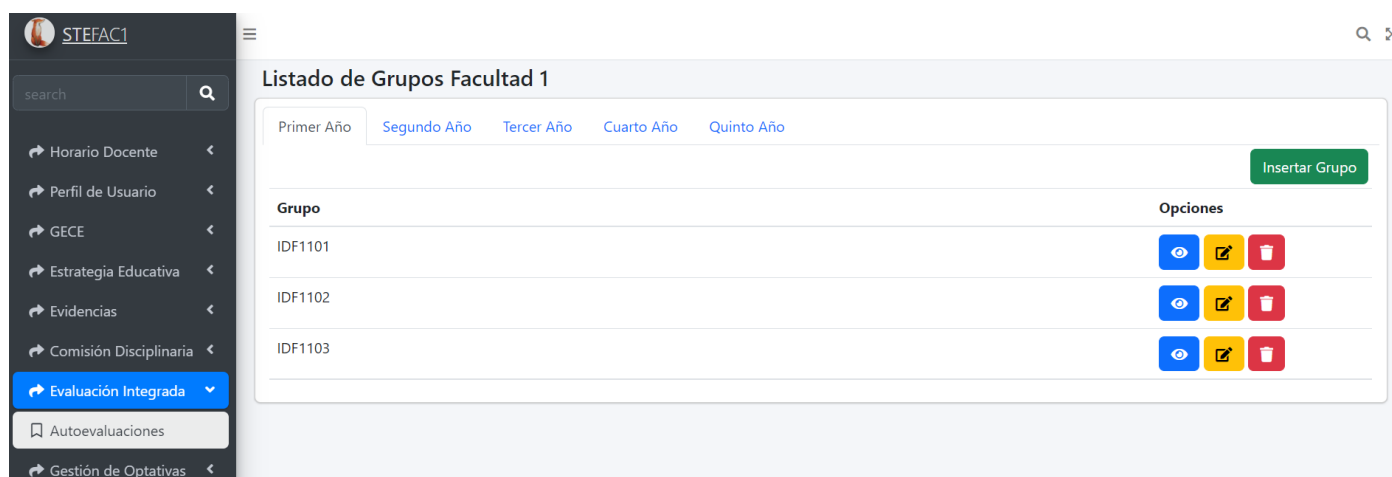


Ilustración 20 Interfaz Crear Grupo

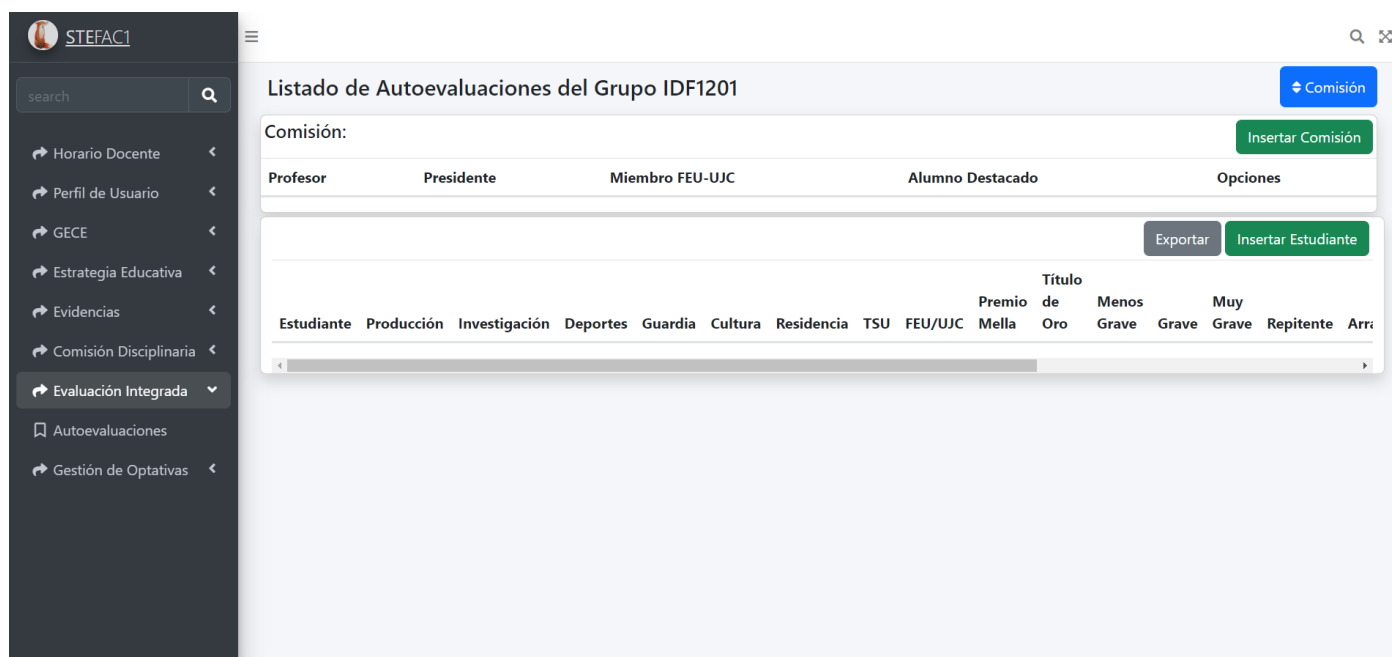


Ilustración 21 Interfaz Comisión-Evaluaciones

The screenshot displays the STEFAC1 user interface for the user Stive Hernandez Suarez. On the left is a dark sidebar menu with the following items: Horario Docente, Perfil de Usuario, GECE, Estrategia Educativa, Evidencias, Comisión Disciplinaria, Evaluación Integrada, and Gestión de Optativas. The main content area is titled 'Stive Hernandez Suarez' and contains five performance cards:
 

- Extensión:** TSU: 2, Deportes: 5, Guardia: 3, Cultura: 2, Residencia: 3, Menos Grave: 4, Grave: 3, Muy Grave: 2. **Calificación: Regular**
- Investigación:** Producción: 4, Investigación: 3. **Calificación: Regular**
- Académico:** Premio Mella: 4, Título de Oro: 5, Repitentes: 3, Arrastres: 4, Mundiales: 5. **Calificación: Bien**
- Político-Ideológico:** UJC-FEU: 3, Otros: 2. **Calificación: Mal**
- Índice Académico:** Índice: 4

 Below these cards is a 'Criterio del Profesor' section with a text input field containing 'fqkfgwhgñq h wrhohfah fkwbajgwbudug eifigr'. To the right of this field is a green 'Insertar Comentario' button and a red trash icon.

Ilustración 22 Interfaz Caracterización