



Facultad 1

**“Módulo del Sistema de Trabajo Educativo de la Facultad 1 para
la gestión de las asignaturas optativas”**

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

Autor:

José Carlos Milanés Anaya

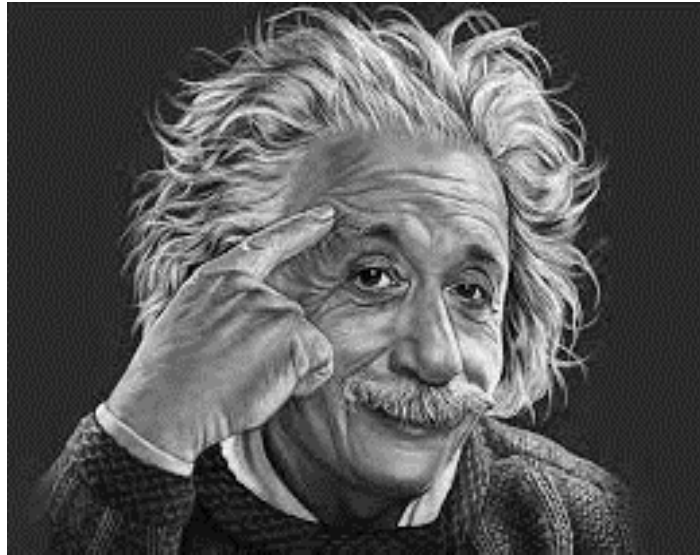
Tutores:

Ing. Yaili Ledea Velázquez

MSc. Disnayle Jorge Chacón

La Habana, noviembre de 2022

Año 64 de la Revolución



"Dos cosas son infinitas: la estupidez humana y el universo; y no estoy seguro de lo segundo".

DECLARACIÓN DE AUTORÍA

El autor José Carlos Milanés Anaya del trabajo de diploma con título “**Módulo del Sistema de Trabajo Educativo de la Facultad 1 para la gestión de las asignaturas optativas**” concede, a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma a los __ días de noviembre del año 2022.

Firma del Autor

Firma del Autor

Firma del Tutor

Firma del Tutor

DATOS DE CONTACTO

Yaili Ledea Velázquez. Ingeniera en Ciencias Informáticas desde 2008. Profesor principal de la Facultad 1 en la Universidad de las Ciencias Informáticas. Consultora de proyectos en temas relacionados con bases de datos y análisis de requisitos. Profesora de escuelas internacionales de postgrado. 12 años de Experiencia profesional en PostgreSQL.

Disnayle Jorge Chacón. Ingeniera en Ciencias Informáticas desde ¿? Profesor principal de la Facultad 1 en la Universidad de las Ciencias Informáticas. Máster ... Profesora de escuelas internacionales de postgrado.

AGRADECIMIENTOS

Gracias a Dios, familiares y a todas las personas que de alguna manera me ayudaron en mi vida ...

DEDICATORIA

A dios, familiares, amigos y demás personas que de alguna manera me han apoyado y ayudado en la vida...

RESUMEN

En la Facultad 1 de La Universidad de las Ciencias Informáticas, se realiza la gestión de toda la información referente al currículo optativo de los estudiantes en su trayecto académico. Durante este proceso se realizan las evaluaciones de los estudiantes tanto cuantitativas como cualitativas de forma manual por lo que en ocasiones hay confusiones y pérdida de la información. Esto evidencia la necesidad de desarrollar un módulo para el Sistema de Trabajo Educativo de la Facultad 1 capaz de gestionar matrículas, evaluaciones, reportes y grupos de los estudiantes de cada año que cursan asignaturas optativas. El módulo se desarrolló utilizando la metodología AUP-UCI en su variante para los proyectos que se desarrollan en la universidad, además de las tecnologías y herramientas *Visual Studio Code*, *Laravel*, *Bootstrap*, el lenguaje JavaScript, el lenguaje de modelado unificado *UML* y *Visual Paradigm*. Se definió y realizó una estrategia de prueba conformada por pruebas de funcionalidad en la que se aplicó el método de caja negra, pruebas de seguridad utilizando la herramienta *Acunetix*, pruebas de carga y estrés haciendo uso de la herramienta apache *JMeter* y pruebas de integración apoyándose en el repositorio de código *github*. Además, se aplicó la técnica de *ladov* para medir el índice de satisfacción grupal del cliente respecto al módulo.

Palabras claves: Asignaturas optativas, Gestionar, Matrículas, Sistema de Trabajo Educativo.

ABSTRACT

In Faculty 1 of the University of Informatics Sciences, the management of all the information regarding the optional curriculum of the students in their academic journey is carried out. During this process, both quantitative and qualitative student evaluations are carried out manually, which is why sometimes there is confusion and loss of information. This evidences the need to develop a module for the Educational Work System of Faculty 1 capable of managing enrollment, evaluations, reports and groups of students each year who take optional subjects. The module was developed using the AUP-UCI methodology in its variant for projects developed at the university, in addition to the technologies and tools Visual Studio Code, Laravel, Bootstrap, the JavaScript language, the UML unified modeling language, and Visual Paradigm. A test strategy was defined and carried out consisting of functionality tests in which the black box method was applied, security tests using the Acunetix tool, load and stress tests using the apache JMeter tool and integration tests based on the github code repository. In addition, the ladov technique was applied to measure the overall customer satisfaction index with respect to the module.

Keywords: Optional subjects, Manage, Enrollment, Educational Work System

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	12
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL PROCESO DE GESTIÓN DE MATRÍCULAS EN ASIGNATURAS OPTATIVAS PARA EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1	16
1.1 Conceptos asociados con el proceso de gestión de matrículas en asignaturas optativas	16
1.2 Análisis y estudio de sistemas homólogos asociados al proceso de gestión de matrículas en asignaturas optativas	17
1.3 Metodología, tecnologías y herramientas.....	20
Conclusiones del capítulo.....	22
CAPÍTULO II: ANALISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE MATRÍCULAS EN ASIGNATURAS OPTATIVAS PARA EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1	23
2.1 Descripción del flujo actual del proceso de gestión de matrícula en las asignaturas optativas.....	23
2.2 Descripción de la propuesta de solución para gestionar las matrículas en las asignaturas optativas	24
2.3 Análisis de la propuesta de solución para gestionar las matrículas en las asignaturas optativas.....	25
2.3.1 Técnicas de captura de requisitos	25
2.3.2 Requisitos funcionales	26
2.3.3 Requisitos no Funcionales	26
2.3.4 Diagrama de caso de uso del sistema	28
2.3.5 Descripción de los Casos de Uso del Sistema	29
2.4 Diseño de la propuesta de solución para gestionar la matrícula en las asignaturas optativas.....	36
2.4.1 Diagrama de clases de diseño con estereotipos web	36
2.4.2 Diagramas de secuencia	38
2.4.3 Modelo de Datos	42
2.4.4 Arquitectura de software.....	44
2.4.5 Patrones de diseño.....	45
2.4.6 Diagrama de despliegue	47
Conclusiones del capítulo.....	48
CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO DE GESTIÓN DE MATRÍCULAS EN ASIGNATURAS OPTATIVAS PARA EL SISTEMA DE TRABAJO EDUCATIVO DE LA FACULTAD 1	49
3.1 Diagrama de Componentes.....	49
3.2 Estándares de Codificación.....	51
3.3 Estrategia de Pruebas	54
3.3.1 Pruebas Funcionales	55

3.3.2 Pruebas de Seguridad	58
3.3.3 Pruebas de Integración	60
3.3.4 Pruebas de Carga y Estrés	63
3.4 Interfaces principales del Módulo de gestión de asignaturas optativas.	64
3.5 Validación Científica	67
Conclusiones del capítulo.....	69
CONCLUSIONES GENERALES	70

ÍNDICE DE TABLAS

TABLA 1. RESUMEN COMPARATIVO DE LOS SISTEMAS HOMÓLOGOS	19
TABLA 2. REQUISITOS FUNCIONALES	26
TABLA 3. CU 1. GESTIONAR OPTATIVA	29
TABLA 4. CU 2. MATRICULAR ESTUDIANTE	35
TABLA 5. DESCRIPCIÓN DEL DIAGRAMA DE COMPONENTES	51
TABLA 6. ESTÁNDARES DE CODIFICACIÓN	52
TABLA 7. RESUMEN DE ESTRATEGIA DE PRUEBAS	54
TABLA 8. DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA 1 PARA EL RF - INSERTAR OPTATIVA	55
TABLA 9. CASO DE PRUEBA DEL RF - INSERTAR OPTATIVA	55
TABLA 10. RESULTADOS DE LAS PRUEBAS DE CARGA Y ESTRÉS	64
TABLA 11. CUADRO LÓGICO DE IADOV	67
TABLA 12. RESULTADOS DE LAS ESCALAS DE SATISFACCIÓN	68

ÍNDICE DE FIGURAS

FIGURA 1. MODELO CONCEPTUAL	24
FIGURA 2. DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	29
FIGURA 3. DIAGRAMA DE CLASES CON ESTEREOTIPOS WEB (GESTIONAR OPTATIVAS).....	37
FIGURA 4. DIAGRAMA DE SECUENCIA (INSERTAR OPTATIVA)	39
FIGURA 5. DIAGRAMA DE SECUENCIA (EDITAR OPTATIVA)	40
FIGURA 6. DIAGRAMA DE SECUENCIA (ELIMINAR OPTATIVA)	41
FIGURA 7. MODELO DE DATOS	43
FIGURA 8. ARQUITECTURA DE SW: MODELO-VISTA-CONTROLADOR	44
FIGURA 9. EMPLEO DEL PATRÓN "EXPERTO"	45
FIGURA 10. USO DEL PATRÓN "FACTORY"	47
FIGURA 11. DIAGRAMA DE DESPLIEGUE	47
FIGURA 12. DIAGRAMA DE COMPONENTES DEL MÓDULO.....	50
FIGURA 13. RESULTADOS DE PRUEBAS DE FUNCIONALIDAD APLICADAS	58
FIGURA 14. RESULTADOS DE PRUEBAS DE SEGURIDAD.....	59
FIGURA 15. RESULTADOS DE LAS PRUEBAS DE INTEGRACIÓN.....	61
FIGURA 16. PRUEBAS DE INTEGRACIÓN	62
FIGURA 17. PANTALLA PRINCIPAL DEL MÓDULO DESARROLLADO	65
FIGURA 18. PANTALLA DE GESTIÓN DE OPTATIVAS	66

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

Las Tecnologías de la información y las comunicaciones (TIC) son el conjunto de medios (radio, televisión y telefonía convencional) de comunicación y las aplicaciones de información que permiten la captura, producción, almacenamiento, tratamiento, y presentación de informaciones en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Incluyen además la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual (Fajardo y Licea, 2022).

Todas las esferas de la sociedad se apoyan en las TIC para alcanzar un alto nivel de desarrollo y encontrar soluciones cada vez más factibles incluyendo el avance tecnológico en las formas y métodos tradicionales de trabajo. La educación es un sector que sosteniéndose en el avance de las tecnologías ha perfeccionado los métodos de enseñanza y aprendizaje, brindando a los profesores y estudiantes una diversidad de herramientas sustentadas y apoyadas en esta base del conocimiento e investigación.

Cuba es un país que, a pesar de no tener un alto nivel de desarrollo en la tecnología, ha sabido encontrar al igual que el resto del mundo los mecanismos necesarios para desarrollar aplicaciones y herramientas de gran utilidad para las instituciones educacionales. Estas aplicaciones y herramientas son construidas con el fin de gestionar la información referente a todo el trayecto académico de los estudiantes, mejorando los resultados en cuanto a mayor eficacia, eficiencia y optimizando recursos.

Las instituciones de Educación Superior en Cuba tienen como misión brindar a la sociedad un profesional formado de manera íntegra, profesionalmente competente, con preparación científica para aceptar los retos de la sociedad moderna y con un amplio desarrollo humanístico para vivir en la sociedad de esta época y servirla con sencillez y modestia, con los valores como pilar fundamental de su formación. Con este propósito todas las carreras universitarias organizan los contenidos que se estudian en tres tipos de currículo: base, propio y optativo/electivo. A partir del currículo base, de los objetivos generales formulados para la carrera y de los programas de las disciplinas, el colectivo de la carrera de cada institución de educación superior decide cómo completar su plan de estudio particular (currículo propio y optativo/electivo), en correspondencia con sus características y la de cada territorio.

Según lo planteado en el Reglamento de Trabajo Docente y Metodológico de la Educación Superior (Res 47/2022): Las asignaturas optativas son aquellas que se incluyen en el plan de estudio y de entre las cuales el estudiante selecciona una cantidad determinada para cursar en forma obligatoria. Los contenidos de estas asignaturas tienen como propósito ampliar y actualizar a los estudiantes sobre temas científicos o tecnológicos relacionados con la profesión.

El plan de estudios “E” de la Universidad de las Ciencias Informáticas (UCI) está concebido siguiendo esta premisa. Específicamente, el currículo optativo-electivo, se diseña de forma tal que complementa la formación del estudiante atendiendo a los intereses específicos del país, del centro de desarrollo de software donde realiza su Práctica Profesional y personales. Este currículo representa el 7.2 % del total de horas, con 5 asignaturas optativas y una electiva distribuidas en el Curso Diurno. Las asignaturas optativas que puede cursar el estudiante a partir del tercer año se ofertan a propuesta de la carrera a partir de un banco de asignaturas que propongan las disciplinas en dependencia de los intereses de la profesión. Este banco de asignaturas optativas se revisa periódicamente. El colectivo de año, conjuntamente con el tutor, deben velar por la elección que efectúe cada estudiante, cuidando que se ajuste al nivel que ha alcanzado y respetando las necesidades de formación individual (*Resolución No.47/2022, 2022*).

Actualmente, en la facultad 1 el proceso de matrícula en estas asignaturas se realiza de forma manual y está propenso a la ocurrencia de errores debido al factor humano, siendo un proceso lento, complejo y monótono. Otra agravante lo constituye el hecho de registrar la matrícula en documentos Word o Excel, lo que provoca que no se pueda comprobar automáticamente si un estudiante ya venció una asignatura en la que está intentando matricular, o si esta se imparte en el mismo horario de otra que también haya matriculado. Este elemento también dificulta la realización de reportes para conocer los estudiantes que faltan por matricular, así como las matrículas en cada asignatura optativa por año académico.

Por lo anteriormente descrito se identificó como **problema de investigación**: ¿Cómo contribuir a la gestión de la matrícula en las asignaturas optativas de los estudiantes de la Facultad 1 de la UCI?

Se define como **objeto de estudio** de la investigación, el proceso de matrícula en asignaturas optativas; enmarcado en el **campo de Acción**: el proceso de matrícula de asignaturas optativas de los estudiantes de la Facultad 1 de la UCI.

En correspondencia se traza como **objetivo general**: desarrollar un módulo para el Sistema de Trabajo Educativo (STE) que permita la gestión de matrícula en las asignaturas optativas de los estudiantes de la Facultad 1 de la UCI.

Con la finalidad de cumplir este objetivo y darle solución al problema que se plantea se formularon las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionados con el proceso de gestión de matrícula en asignaturas optativas?

2. ¿Qué propuesta de solución se define para mejorar la gestión de la matrícula en las asignaturas optativas?
3. ¿Cuáles son las características que debe cumplir el módulo de gestión del proceso de matrícula en las asignaturas optativas de la Facultad 1 de la UCI?
4. ¿Cómo validar el funcionamiento del módulo de gestión del proceso de matrícula en las asignaturas optativas de la Facultad 1 de la UCI y la investigación?

Para dar respuesta a las preguntas científicas se definieron las siguientes **tareas de investigación**:

1. Estudio de los referentes teóricos fundamentales que sustentan la investigación relacionada con el proceso de gestión de matrícula en las asignaturas optativas.
2. Análisis y diseño del módulo para la gestión de la matrícula en las asignaturas optativas.
3. Implementación de las funcionalidades del módulo para la gestión de la matrícula en las asignaturas optativas.
4. Validación de las funcionalidades del módulo para la gestión de la matrícula en las asignaturas optativas y la investigación.

Para dar cumplimiento a los objetivos propuestos, se utilizan diferentes **métodos científicos**. Los principales son (Hernández Sampieri et al., 2014):

Métodos teóricos:

Analítico-sintético: se hace uso de este método para la identificación de conceptos dentro del campo de los servicios web, analizando documentos, para la extracción de los elementos más importantes sobre el proceso de gestión de matrículas en asignaturas optativas.

Modelación: se emplea en la creación de diagramas para representar el proceso de desarrollo mediante el lenguaje de modelado UML, la estructura, relaciones internas y características de la solución.

Métodos empíricos:

Observación: facilita obtener conocimiento acerca del funcionamiento de los sistemas y módulos existentes para el proceso de gestión de matrículas en asignaturas optativas.

Entrevista: se utiliza para recoger la información que tributa al desarrollo del módulo para la gestión de matrícula en las asignaturas optativas. Permite una conversación planificada para obtener

información, mediante reuniones informales con los encargados de hacer este proceso de gestión de matrículas en las asignaturas optativas (ver anexo 1).

Método de Validación:

- **La técnica ladov:** Se constituye en una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario (Verdecia & Augusto, 2021) (Tinajero et al., 2021). Se realizó una encuesta a los profesores principales de cada año con el fin de medir el índice de satisfacción general del módulo de gestión de matrículas en asignaturas optativas (ver anexo 2).

El documento está compuesto por tres capítulos, los cuales se resumen a continuación:

- Capítulo 1. Fundamentos y referentes teórico-metodológicos sobre el proceso de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1: Este capítulo recoge los conceptos fundamentales asociados a la investigación. Incluye un estado del arte del tema a nivel internacional y nacional. Así como fundamentos teóricos de la investigación y conceptos asociados, herramientas y tecnologías a utilizar.
- Capítulo 2. Análisis y diseño del módulo de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1: En este capítulo se describen los requisitos funcionales y no funcionales del sistema para su correcto funcionamiento. Así como el análisis y diseño del mismo empleando los artefactos correspondientes.
- Capítulo 3. Implementación y validación del módulo de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1: En este capítulo se construye la propuesta de solución antes descrita. Se presentan las principales tareas de implementación a realizar y los resultados de las diferentes pruebas de software.

Capítulo I: Fundamentos y referentes teórico-metodológicos sobre el proceso de gestión de matrículas en asignaturas optativas para el sistema de trabajo educativo de la facultad 1

En este capítulo se analizan los conceptos fundamentales asociados a la investigación. Se hace un estudio de las características de la metodología a utilizar, así como las tecnologías, lenguajes y herramientas que se emplearán en la construcción de la aplicación. Se estudian sistemas homólogos existentes en el ámbito nacional e internacional.

1.1 Conceptos asociados con el proceso de gestión de matrículas en asignaturas optativas

A continuación, se muestran los principales conceptos asociados para un mejor entendimiento del objeto de estudio, haciendo énfasis en los conceptos de gestión, matrícula, asignatura y asignaturas optativas.

Gestión: Es el proceso que se realiza con el fin de controlar todos los medios o recursos disponibles para hacer un trabajo o investigación referente a cualquier tema o esfera y garantizar el buen acabado del mismo en cuanto a calidad y tiempo (Cabrera y Noel, 2015).

Existe concordancia con Gutiérrez (2020) cuando plantea que la gestión es entendida como el sistema de actividades estratégicas y planificadas que se llevan a cabo guiadas por instrucciones y técnicas apropiadas para obtener determinados fines o metas. Estos fines, metas u objetivos, pueden ser alcanzados por personas instituciones, empresas tanto públicas como privadas.

Se entiende como **gestión educativa** la capacidad que tiene la institución para dirigir sus procesos, recursos y toma de decisiones, en función a la mejora permanente del proceso de enseñanza-aprendizaje y el logro de la formación integral de todos los estudiantes que atiende (Gutiérrez, 2020).

Matrícula: Es el resultado del proceso de inscripción formal en una institución de educación, a través de la cual las personas dan a conocer oficialmente su participación en el inicio de cada periodo escolar, su condición de estudiante (Cabrera y Noel, 2015). Para los estudiantes del curso diurno y del curso por encuentros de las instituciones cubanas de Educación Superior, la matrícula implica la obligación de cursar todas las asignaturas establecidas en el año académico que le corresponda en el plan de estudio, que incluyen las asignaturas optativas y electivas que seleccione (*Resolución No.47/2022, 2022*). Se debe tener en cuenta que estos estudiantes no pueden matricular asignaturas de años académicos superiores al que le corresponda cursar en su carrera salvo casos excepcionales.

Asignatura: Es una materia que forma parte del plan de estudio de cualquier centro o establecimiento educativo en el momento que un estudiante cursa una carrera o curso (Cabrera y Noel, 2015).

Asignaturas optativas: Son aquellas que se incluyen en el plan de estudio y de entre las cuales el estudiante selecciona una cantidad determinada para cursar en forma obligatoria. Los contenidos de estas asignaturas tienen como propósito ampliar y actualizar a los estudiantes sobre temas científicos o tecnológicos relacionados con la profesión. La cantidad de asignaturas optativas a seleccionar por el estudiante se precisa en el plan del proceso docente aprobado por el rector de la institución de la educación superior (*Resolución No.47/2022, 2022*).

1.2 Análisis y estudio de sistemas homólogos asociados al proceso de gestión de matrículas en asignaturas optativas

Aplicación para la gestión de matrículas y notas de la Escuela de Educación Básica Zoila Rosa García Montenegro:

Es un software desarrollado con el fin de gestionar las matrículas y el registro de notas de los estudiantes. Para el desarrollo del software se utilizó la metodología XP o Programación Extrema que es una metodología ágil y flexible. La aplicación cuenta con varios módulos, entre ellos: módulo de administración, módulo para la gestión de cursos, de alumnos, de materias, de profesores, de matriculación, de notas.

Para el desarrollo del software se utilizaron una serie de herramientas y tecnologías, Microsoft Word como procesador de texto, C# como lenguaje de programación, SQL Server para la base de datos, IIS como servidor de aplicaciones, y *visual studio code* como IDE (Baque y Herrera, 2021).

Aplicación web para matriculación y gestión de notas de La Escuela fiscal N° 331 Esperanza Caputi Olvera de la ciudad de Guayaquil:

Aplicación web desarrollada para automatizar los procesos de gestión de matriculación y gestión de notas que funciona bajo un servidor local. Cuenta con varios módulos y submódulos tales como el de administración que permite gestionar todo lo referente a los alumnos, docentes, cursos, paralelos, materias, usuarios, periodos, tipo de calificación y roles.

El módulo de matriculación permite gestionar la información de los estudiantes, así como registrar y matricular. Posee un módulo de gestión de notas encargado de registrar las notas de los estudiantes y un módulo de reportes que brinda información detallada de las estudiantes en cuanto a su trayectoria de evaluaciones parciales o totales.

Entre las tecnologías y herramientas utilizadas están: Java, Java EE, JSF2, JPA, JavaScript, XML, CSS, HTML, XHTML, *Primefaces*, SQL, Eclipse Mars, MySQL *workbench*, Wildfly, Lenguaje unificado de modelado (UML) y diseño guiado por el dominio (Morales y Navarro, 2019).

Aplicación web para la matriculación y gestión de ingreso de notas académicas de la Escuela de Educación Básica Fiscal Pedro Carbo:

Es una aplicación para automatizar el proceso de matrículas mediante la importación de un archivo Excel, para hacer un fácil ingreso de la información de los estudiantes, gestionar las evaluaciones y generar reportes. Además, envía mensajes a través de correo para notificar a los padres acerca de reuniones, eventos o indisciplinas. Se creó en un ámbito web local, en otras palabras, se puede acceder a ella desde cualquier dispositivo conectado a la red de la institución educativa.

Cuenta con varios módulos, módulo de seguridad, de administración, de matriculación, de ingreso, de notas, de reportes y un módulo de ayuda. Las tecnologías y herramientas utilizadas para el desarrollo del software fueron: Odo, Python, JavaScript, AJAX, CSS, PostgreSQL.

Como metodología se utilizó SCRUM por su flexibilidad, metodología ágil que se centra en el control y planificación de proyectos de corto plazo, permite que el grupo de trabajo esté más implicado, que haya más organización y comunicación y cumplir con mejor calidad los objetivos (Quimis y Figueroa, 2021).

Trabajo de diploma titulado “Sistema informático para la gestión del proceso de matrícula en las asignaturas optativas de la Facultad 1 de la Universidad de las Ciencias Informáticas”:

Desarrollado para agilizar todo el proceso de matriculación por parte de los estudiantes en las asignaturas optativas y tener un mejor manejo de toda la información. La aplicación cuenta con un conjunto de módulos, los cuales se mencionan a continuación: gestión de los usuarios, gestión de matrículas, gestión de facultades, gestión de período lectivo, gestión de turnos docentes, gestión de asignaturas, gestión de disciplinas, gestión de años académicos y gestión de reportes.

Se utilizó la metodología XP por ser un proyecto pequeño, por ser una metodología con el objetivo de promover el trabajo en equipo con la intención de obtener un buen clima de trabajo y buena comunicación entre los integrantes. Es una metodología que se adapta fácilmente a los cambios durante el desarrollo.

Las tecnologías y herramientas utilizadas fueron apache en su versión 2.2.22 como servidor web, PHP por su compatibilidad con las bases de dato más comunes en su versión 5.3.6, HTML versión 5, CSS 3, JavaScript en su versión 1.8, PostgreSQL como gestor de base de datos en la versión 9, pgAdmin

en la versión 1.8 como administrador de bases de datos, UML (*Unified Modeling Language*) o Lenguaje Unificado de Modelado versión 2.0 como lenguaje de modelado, Visual Paradigm en su versión 8.0, como entorno de desarrollo integrado se utilizó Netbeans en la versión 7.4, también se utilizaron *frameworks* como Symfony en su versión 2.6 y Bootstrap en su versión 3 (Cabrera y Noel, 2015).

Tabla 1. Resumen comparativo de los sistemas homólogos

Sistemas analizados \ Aspectos	Gestionar matrículas de estudiantes	Generar reportes	Asignar nota a los estudiantes	Software libre
Aplicación para la Escuela de Educación Básica ZRGM	Sí	No	No	No
Aplicación web para la Escuela fiscal N° 331 E C O G	Sí	No	No	No
Aplicación web para la Escuela de EBFPC	Sí	Sí	No	No
Sistema informático para la Facultad 1 de la UCI	Sí	Sí	No	Sí

La tabla 1 muestra el análisis de los sistemas homólogos estudiados. Luego de un análisis de algunos de los sistemas homólogos existentes en el ámbito nacional e internacional. Se pudo concretar que existen muchas similitudes entre estos sistemas y el objetivo de la investigación. Se identificaron buenas prácticas, nuevas tecnologías y herramientas de gran utilidad para la investigación. Por ejemplo, todos utilizan metodologías ágiles por su gran facilidad de adaptación ante las diferentes situaciones y por ser proyectos pequeños.

Los procesos de gestión definidos en los sistemas homólogos estudiados no entran en especificidades, es decir, solo hablan de la gestión de la información referente a matrículas y evaluaciones y no hacen referencia a si existe algún modelo que gestione información de asignaturas optativas, son sistemas de pago y el trabajo de diploma titulado “Sistema informático para la gestión del proceso de matrícula en las asignaturas optativas de la Facultad 1 de la Universidad de las Ciencias Informáticas” no cuenta con un sistema informático. Para esta investigación se hace necesario saber con precisión y al no encontrarla porque estos sistemas responden a las necesidades de cada institución en particular se hace necesario desarrollar una aplicación de gestión de toda la información referente al proceso de

matrículas de los estudiantes en las asignaturas optativas, atendiendo a las necesidades y formas de proceder de la facultad 1 de La Universidad de las Ciencias Informáticas.

1.3 Metodología, tecnologías y herramientas

Una metodología de desarrollo de software es el entorno guía para gestionar el proceso de desarrollo de un sistema informático (Morales et al., 2019). Tiene entre sus objetivos aumentar la calidad del software que se produce e imponer un proceso disciplinado y eficiente (Rodríguez, 2015).

En la presente investigación se utilizará como metodología de desarrollo AUP en su versión adaptada para los procesos de desarrollo de software de la UCI, en su escenario 2 y se hará uso de los artefactos modelo conceptual y casos de uso del sistema. Se apoyará en el modelo CMMI-DEV v1.3 que es un modelo de mejora del rendimiento de clase mundial para organizaciones competitivas que desean lograr actividades de alto rendimiento y consiste en recolectar las mejores prácticas (Rodríguez, 2015), en el cual la UCI cuenta actualmente con un nivel 2 de madurez.

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que se llamará Ejecución y se agrega la fase de Cierre.

A continuación, se muestran las herramientas y tecnologías asociadas para el desarrollo de la investigación:

Laravel v9

Es un marco de trabajo de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7, desarrollado por Taylor Otwell en 2011 y tiene una gran influencia de *frameworks* como *Ruby on Rails*, *Sinatra* y *ASP.NET MVC* (Espinoza, 2019).

PHP v8

Es un lenguaje de programación de código abierto que se utiliza para el desarrollo web de manera dinámica y puede ser empleado en páginas HTML y ejecutados en un servidor. Permite el desarrollo de aplicaciones web y cuenta con un amplio soporte a diferentes bases de datos, facilitando que los desarrolladores creen sitios sustentados en bases de datos. Basado en el uso de herramientas afectivas para aumentar la productividad en el desarrollo de software, muy ligado a la utilización de buenas prácticas de ingeniería de software (Bautista, 2022).

HTML5 (*Hyper Text Markup Language*) (Lenguaje de marcado de hipertexto)

Fue creado en 1991 por Tim Berners-Lee, su primera versión fue HTML 1.0, se utiliza para la creación, representación y estructuración del contenido de una página web mediante elementos de formato o etiquetas (Rebah et al., 2022). HTML brinda a los autores los medios para publicar documentos en línea con encabezados, texto, tablas, listas, fotos, etc. Recuperar información en línea a través de enlaces de hipertexto, con solo hacer clic en un botón. Diseñar formularios para la realización de transacciones con servicios remotos, para su uso en la búsqueda de información, realización de reservas, pedido de productos, etc. Incluir hojas de cálculo, videoclips, clips de sonido y otras aplicaciones directamente en sus documentos (*HTML & CSS - W3C*, 2022).

CSS3 (*Cascading Style Sheets*)

Es un lenguaje utilizado para construir el diseño y el estilo de las páginas web (Juneau, 2022), incluidos los colores, el diseño y las fuentes. Permite adaptar la presentación a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras. Se puede utilizar con cualquier lenguaje de marcado basado en XML (*HTML & CSS - W3C*, 2022).

Bootstrap v5

Es una biblioteca gratuita y de código abierto para aplicar estilos a los elementos HTML de nuestro sitio web (Juneau, 2022). Presenta diseños basados en HTML y CSS para tipografía. Formulario, botones, taba, navegación, modales, carruseles de imágenes (Burgos y Contreras, 2022).

JavaScript

Es un lenguaje de programación ligero, se encuentra diseñado en un paradigma basado en objetos. Es usado para diseñar y programar el comportamiento de la página web cuando ocurre un evento (Vizcaíno y Paolo, 2022).

MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Constituye un software libre que persigue cumplir el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad (Vivanco et al., 2022).

XAMPP v3.3.0

Es una herramienta que contiene software libre (Apache, MySQL, PHP y Perl). Su implementación permite levantar un servidor local para establecer la conexión con la base de datos (Cevallos y Andres, 2022).

Visual Studio Code v17.2

Es un programa de liberación creativa multiplataforma que puede utilizarse para crear, modificar y depurar código. Orientado a los desarrolladores, el cual soporta diversos lenguajes de programación (Burgos y Contreras, 2022).

Lenguaje Unificado de Modelado (UML)

Es un lenguaje gráfico destinado al modelado de sistemas y procesos (Ramírez, 2022).

Visual Paradigm v8.0

Es una herramienta que soporta el ciclo de vida completo del desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además, posibilita la generación de base de datos, transformación de diagramas de entidad-relación en base de datos, así como ingeniería inversa de base de datos (Hernández et al., 2016).

JMeter v5.4.3: Es una herramienta java que forma parte del proyecto Jakarta, que permite hacer pruebas de rendimiento y pruebas funcionales a Aplicaciones Web (Almenares, 2012).

Conclusiones del capítulo

En el presente capítulo quedó plasmado los fundamentos teórico-metodológicos del proceso de gestión de la matrícula de los estudiantes en las asignaturas optativas. Con el análisis de los conceptos relacionados con la investigación se logró un mejor entendimiento y recopilación de información referente al tema que se aborda. La observación y estudio de los sistemas homólogos llevó a la conclusión de que estos no representan una solución porque no cumplen con las características necesarias para ser utilizados. La descripción y definición tanto de las herramientas, tecnologías, así como la metodología seleccionada para desarrollar la presente investigación, han permitido abordar un poco más sobre las posibles propuestas de solución para dar respuesta al problema planteado.

Capítulo II: Análisis y diseño del módulo de gestión de matrículas en asignaturas optativas para el sistema de trabajo educativo de la facultad 1

A continuación, se realiza una breve descripción del análisis y diseño de la propuesta de solución, mediante los requisitos funcionales y no funcionales, así como los elementos diferenciadores del diseño. A su vez, se pueden apreciar los aspectos esenciales de la arquitectura que se utilizará durante el desarrollo del módulo de gestión de matrículas en las asignaturas optativas y los principales patrones de diseño, acompañados de la descripción de los artefactos generados por la arquitectura para el análisis y diseño.

2.1 Descripción del flujo actual del proceso de gestión de matrícula en las asignaturas optativas

En la Facultad 1 de la Universidad de las Ciencias Informáticas, se realiza el proceso de gestión de matrículas en las asignaturas optativas. Los profesores principales de cada año confeccionan un listado de optativas, el cuales es compartido a los estudiantes para que efectúen las solicitudes de inscripción. La solicitud es evaluada por el Profesor Principal de cada año y luego se le informa al estudiante si su solicitud fue aprobada o rechazada. La información referente al trayecto de los estudiantes en estas asignaturas se registra en documentos Word o Excel, así los profesores encargados gestionan los datos y conocen la cantidad de estudiantes que conforman cada grupo por optativa, las evaluaciones, la cantidad de optativas que un estudiante ha cursado y cuántas le faltan por cursar.

La figura 1 muestra el modelo conceptual referente al proceso de gestión de matrículas en las asignaturas optativas.

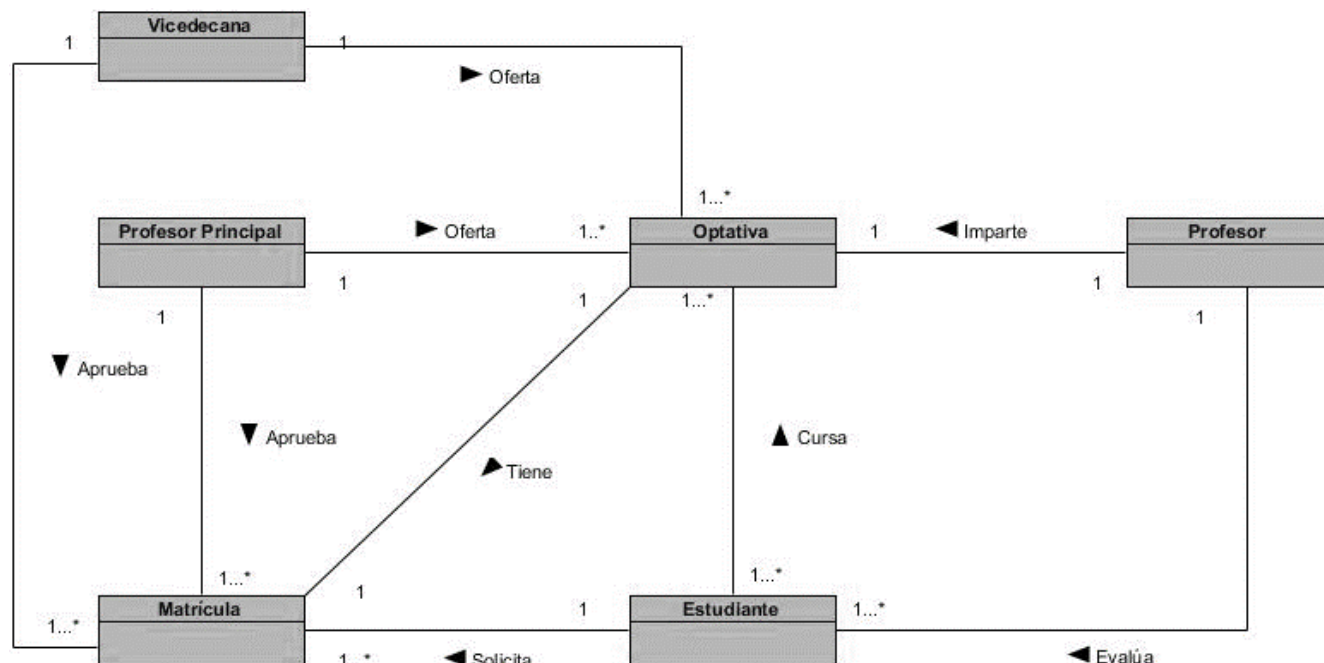


Figura 1. Modelo conceptual

A continuación, se muestra la información detallada de los conceptos representados en el modelo:

- Vicedecana: es el responsable de compartir las asignaturas optativas a los estudiantes.
- Profesor Principal: es el responsable de compartir las asignaturas optativas a los estudiantes.
- Optativa: muestra información detallada de la optativa como el nombre, descripción, profesor que la imparte, año, capacidad, semestre.
- Profesor: es el que imparte la asignatura optativa y evalúa a los estudiantes.
- Estudiante: es quién cursa las asignaturas optativas.
- Matrícula: es la entidad que tiene los datos referentes a los estudiantes que la conforman por optativa.

2.2 Descripción de la propuesta de solución para gestionar las matrículas en las asignaturas optativas

En la Facultad 1 de la Universidad de las Ciencias Informáticas, se lleva a cabo actualmente el proceso de matrículas en las asignaturas optativas, por lo que se torna necesaria la comprensión de las características y conceptos principales asociados al proceso anteriormente mencionado.

El módulo de gestión de matrículas en las asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1 (STE), tiene como objetivo principal la gestión de la información relacionada con el proceso de matrículas de los estudiantes en las asignaturas optativas; la confección de los grupos de estudiantes por optativa con su respectivo profesor; conocer en tiempo real la cantidad de optativas que un estudiante ha cursado; las optativas que actualmente se encuentra cursando y sus respectivas evaluaciones. Además, el estudiante tiene la posibilidad de matricular en la optativa que él desee siempre y cuando exista capacidad en ella y el Profesor Principal lo permita. Con el fin de lograr el pleno funcionamiento del módulo, será integrado al STE de la Facultad 1, para lograr así gestionar toda la información de las matrículas de los estudiantes en las asignaturas optativas desde el propio sistema.

2.3 Análisis de la propuesta de solución para gestionar las matrículas en las asignaturas optativas

En este epígrafe se define la técnica de captura de los requisitos y se identifican los requisitos funcionales y no funcionales del módulo para el STE de la Facultad 1 de la UCI. Además, se hace un análisis y diseño de la propuesta de solución.

2.3.1 Técnicas de captura de requisitos

La fase de captura de requisitos tiene como objetivo descubrir y recoger todos y cada uno de los requisitos funcionales y no funcionales de la aplicación a definir de forma clara y concisa (Catallops y José, 2019).

A continuación, se presentan varios métodos de obtención de requisitos utilizados para el desarrollo de la investigación:

- **Entrevista:** Permitted comprender con mayor claridad el problema que se desea informatizar. Para esto, se realizaron 2 entrevistas a varios Profesores Principales de año encargados de gestionar información referente a las asignaturas optativas en la Facultad 1 de la UCI, y otros especialistas de la facultad (Vicedecana de Formación, Secretaria Docente) encargados de utilizar toda la información referente al proceso de gestión de matrículas en las asignaturas optativas de la Facultad 1 (ver anexo 3).
- **Análisis de los sistemas homólogos:** Fuente para la obtención de requisitos principales del módulo, se encuentra el análisis a fondo de los sistemas homólogos estudiados y antes descritos.

2.3.2 Requisitos funcionales

Los requisitos funcionales son aquellos directamente relacionados con las funciones y las reacciones que el sistema debe proporcionar. Son asignados directamente a elementos o características del sistema de software (Catallops y José, 2019).

Una vez analizado con el cliente, se obtuvo un total de 9 requisitos funcionales, asignándoles una prioridad específica a cada uno de ellos. En la tabla 2 se especifican los requisitos funcionales definidos:

Tabla 2. Requisitos Funcionales

Requisito	Descripción	Prioridad
RF1: Insertar Optativa	Inserta una nueva optativa en la lista de optativas	Alta
RF2: Eliminar Optativa	Elimina una optativa de la lista de optativas	Alta
RF3: Modificar Optativa	Modifica un optativa de la lista	Alta
RF4: Listar Optativa	Lista todas las optativas insertadas	Alta
RF5: Buscar Optativa	Permite buscar una optativa por el nombre	Media
RF6: Matricular estudiante en optativa	Matricula a un estudiante en una optativa determinada	Alta
RF7: Desmatricular estudiante en optativa	Desmatricula a un estudiante de una optativa determinada	Alta
RF8: Listar estudiantes por optativa	Muestra un listado de estudiantes según la optativa en la que están matriculados	Media
RF9: Asignar nota a estudiante	El profesor asigna una nota a cada estudiante matriculado en la optativa	Alta

2.3.3 Requisitos no Funcionales

Un requisito no funcional no es más que un requisito que especifica criterios que se pueden utilizar para determinar el funcionamiento de un sistema, en lugar de comportamientos específicos, permitiendo que se puedan describir como atributos de calidad de un sistema (Martínez y Lugones, 2018).

Usabilidad:

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

Confiabilidad:

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

Portabilidad:

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

Seguridad:

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

Apariencia o Interfaz externa:

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

Eficiencia:

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

Mantenibilidad:

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

Hardware:

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

Software:

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

2.3.4 Diagrama de caso de uso del sistema

La figura 2 muestra el diagrama de caso de uso del sistema.

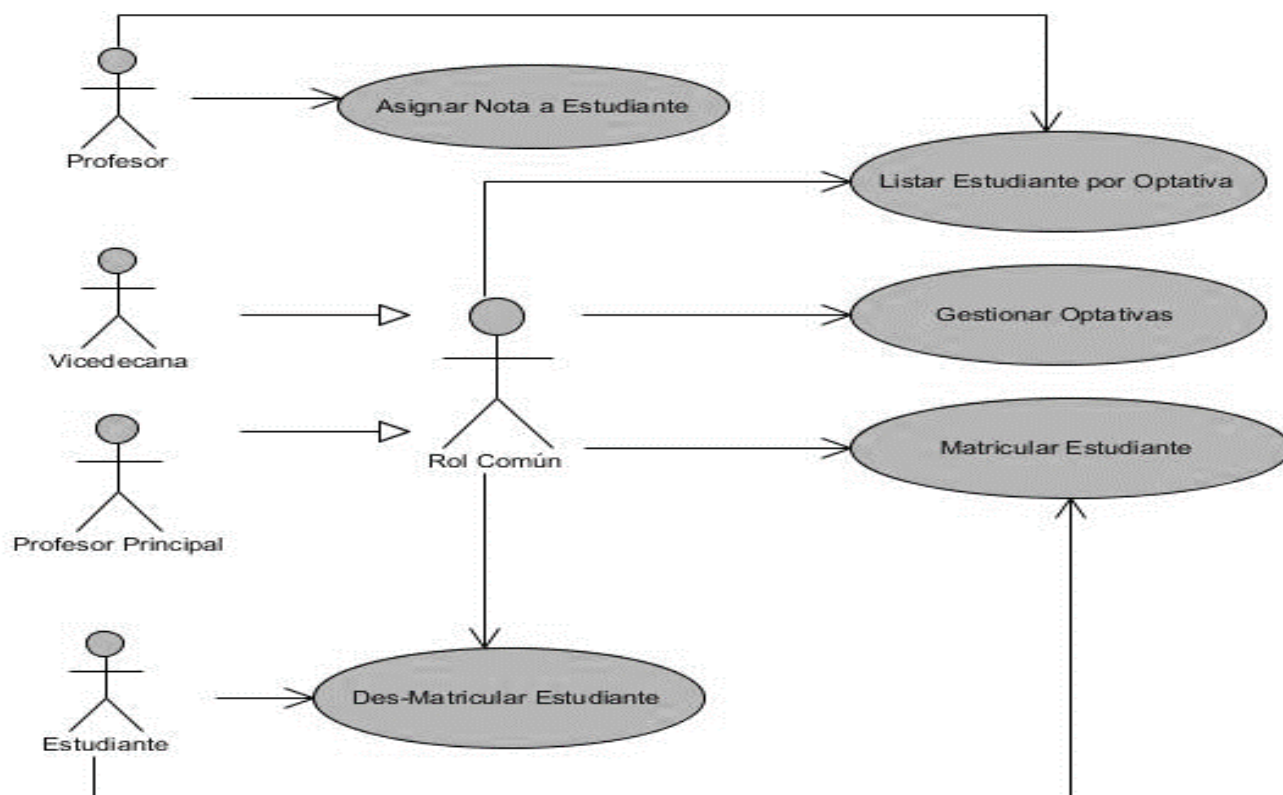


Figura 2. Diagrama de Casos de Uso del sistema

2.3.5 Descripción de los Casos de Uso del Sistema

Tabla 3. CU 1. Gestionar Optativa

Objetivo	Permite insertar, modificar, eliminar y buscar los datos de una asignatura optativa.
Actores	Profesor Principal y Vicedecana.
Resumen	El caso de uso inicia cuando el Profesor Principal y Vicedecana Inserta, modifica y elimina los datos de una asignatura optativa.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Profesor Principal y Vicedecana ya autenticado.
Postcondiciones	Se registró, modificó o eliminó una asignatura optativa.

Flujo de eventos		
Flujo básico < Gestionar Optativa >		
Actor		Sistema
1.	Selecciona la opción "Optativas".	
2.		Muestra en pantalla una lista con las opciones de Insertar, Editar o Eliminar asignatura optativa.
3.	Desea Insertar, Editar o eliminar una asignatura.	
4.		<p>El sistema despliega una pantalla con las opciones para la gestión de asignatura optativa.</p> <ol style="list-style-type: none"> 1. Para insertar asignatura ir a la sección "Insertar asignatura optativa" 2. Para eliminar asignaturas ir a la sección "Eliminar asignatura optativa" 3. Para modificar asignaturas ir a la sección "Editar asignatura optativa" 4. Para mostrar asignaturas ir a la sección "Mostrar asignatura optativa"
Flujos alternos		
2a. Listado de asignatura optativa vacío.		
Actor		Sistema
1.		Muestra en pantalla un listado vacío.
Sección 1: "Insertar asignatura optativa"		
Flujo básico < Insertar asignatura optativa >		
Actor		Sistema

1.	Presiona la opción "Insertar asignatura optativa".	
2.		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> -Nombre -Descripción -Profesor Principal -Profesor Auxiliar -Capacidad -Año Académico -Semestre -Estado <p>Y el botón "Guardar".</p> <p>Y el botón "Cancelar".</p>
3.	Introduce los datos de la asignatura optativa en los campos y presiona la opción "Guardar" o "Cancelar".	
4.		Verifica que los campos estén llenos.
5.		Verifica que la asignatura optativa no esté registrada.
6.		<p>Guarda los datos de la asignatura optativa y muestra un mensaje "Asignatura Optativa registrada".</p> <p>Finalizando así el Caso de Uso.</p>
Flujos alternos		
4a. Campos vacíos.		

Actor		Sistema
1.		Muestra un mensaje “El campo es requerido”.
Flujos alternos		
5a. Asignatura Optativa registrada.		
Actor		Sistema
1.		Muestra un mensaje “Asignatura Optativa registrada ”.
Sección 2: “Editar asignatura optativa”		
Flujo básico < Editar asignatura optativa>		
Actor		Sistema
1.	Presiona la opción “Editar “en la lista de asignaturas optativas mostradas.	
2.		<p>Muestra un formulario con los siguientes campos a modificar:</p> <ul style="list-style-type: none"> -Nombre -Descripción -Profesor Principal -Profesor Auxiliar -Capacidad -Año Académico -Semestre -Estado <p>Y el botón “Editar”.</p> <p>Y el botón “Cancelar”.</p>

3.	Actualiza los datos de la asignatura optativa en los campos y presiona la opción “Guardar” o “Cancelar”.	
4.		Verifica que los campos estén llenos.
5.		Actualiza los datos de la asignatura y muestra un mensaje “Asignatura optativa Editada”. Finalizando así el Caso de Uso.

Flujos alternos

4a. Campos vacíos.

Actor		Sistema
1.		Muestra un mensaje “El campo es requerido”.

Sección 3: “Eliminar asignatura optativa”

Flujo básico <Eliminar asignatura optativa >

Actor		Sistema
1.	Selecciona la asignatura optativa que desea eliminar y presiona el botón ‘Eliminar’.	
2.		Muestra en pantalla un mensaje “Esta seguro que desea eliminar”. -y el botón “Aceptar” -Y el botón “Cancelar”
3.	Presiona el botón “Aceptar”	
4.		Elimina los datos de la asignatura optativa y muestra un mensaje “asignatura optativa eliminada”. Finalizando así el Caso de Uso.

Flujos alternos		
2a. Presiona botón “Cancelar”		
Actor		Sistema
1.		Retornar al paso 2 en la sección “Optativas”
Sección 4: “Buscar Asignatura optativa”		
Flujo básico <Buscar Asignatura optativa >		
Actor		Sistema
1.	Introduce datos de la asignatura que desea buscar y selecciona la opción “Buscar” en la lista de asignaturas optativas mostradas.	
2.		Verifica que los datos introducidos sean correctos
3-		Busca los datos de la asignatura optativa y muestra una nueva lista con los datos de la asignatura encontrada finalizando así el Caso de Uso
Flujos alternos		
2a. “Datos incorrectos”		
Actor		Sistema
1.		Muestra un mensaje “Datos incorrectos”
Relaciones	CU Incluidos	No aplica
	CU Extendidos	No aplica
Requisitos no funcionales	No aplica	
Asuntos pendientes	No aplica	

Tabla 4. CU 2. Matricular Estudiante

Objetivo	Permitir al usuario matricular en una asignatura optativa.	
Actores	Estudiante.	
Resumen	El caso de uso inicia permitiendo a los estudiantes visualizar el listado de asignaturas optativas para matricular.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	Usuarios ya autenticados	
Postcondiciones	Se muestra una pantalla con el listado de asignaturas optativas.	
Flujo de eventos		
Flujo básico <Matricular Estudiante>		
Actor		Sistema
1-	Presiona la opción "Matricular en Asignatura Optativa"	
2-		Muestra en pantalla un listado de asignaturas optativas, con la opción de matricular en caso de no estar matriculado o desmatricular en caso de estar matriculado.
3-	Selecciona la opción de matricular en caso de no estar matriculado o desmatricular en caso de estar matriculado.	
4-		Muestra en pantalla un listado de asignaturas optativas en el cual ya está el estudiante matriculado o desmatriculado. Finaliza así el caso de uso.
Relaciones	CU Incluidos	No aplica

	CU Extendidos	No aplica
Requisitos no funcionales	No aplica	
Asuntos pendientes	No aplica	

2.4 Diseño de la propuesta de solución para gestionar la matrícula en las asignaturas optativas

En esta sección se pueden observar varios diagramas de clases de diseño con estereotipos Web, así como varios diagramas de secuencia correspondientes a los Casos de Usos descritos en el epígrafe anterior.

2.4.1 Diagrama de clases de diseño con estereotipos web

Los diagramas de clases describen de manera general los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir en algún momento entre ellos. Estos diagramas se han convertido a lo largo de los años en la técnica más potente para el modelado conceptual de un sistema de software.

En la presente investigación se crearon un total de 3 diagramas de clases de diseño con estereotipos web, por lo que a continuación se muestra tres de ellos, correspondientes a los requisitos descritos en el sub-epígrafe anterior.

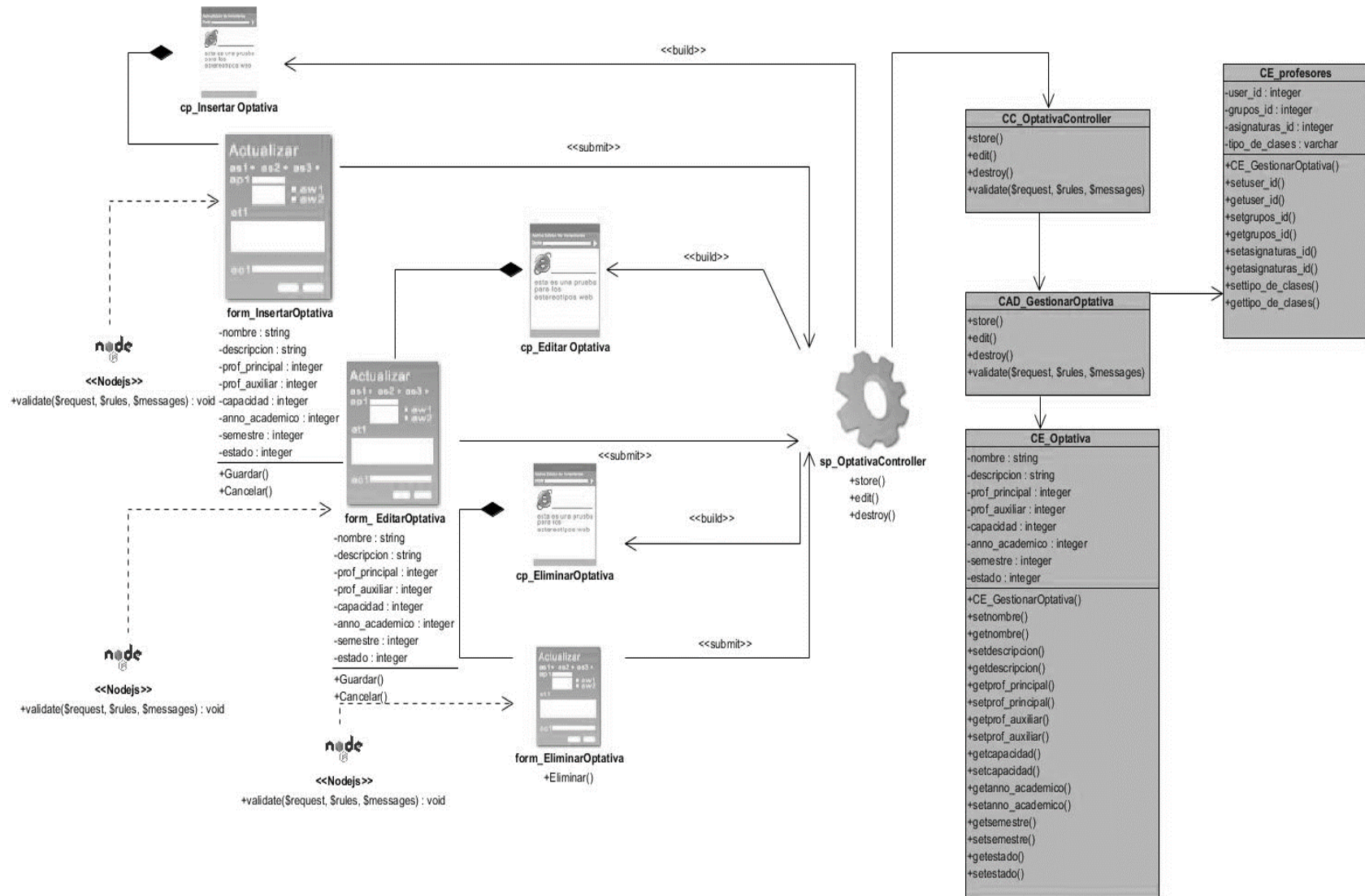


Figura 3. Diagrama de clases con estereotipos Web (Gestionar Optativas)

2.4.2 Diagramas de secuencia

Los diagramas de secuencia muestran de manera general la forma en que un grupo de objetos se comunican entre sí a lo largo de un tiempo específico, además de que permiten observar los objetos de un escenario mediante líneas verticales y los mensajes entre objetos como flechas conectando objetos.

Para la presente investigación se generaron un total de 7 diagramas de secuencias, a continuación, se muestran tres de ellos:

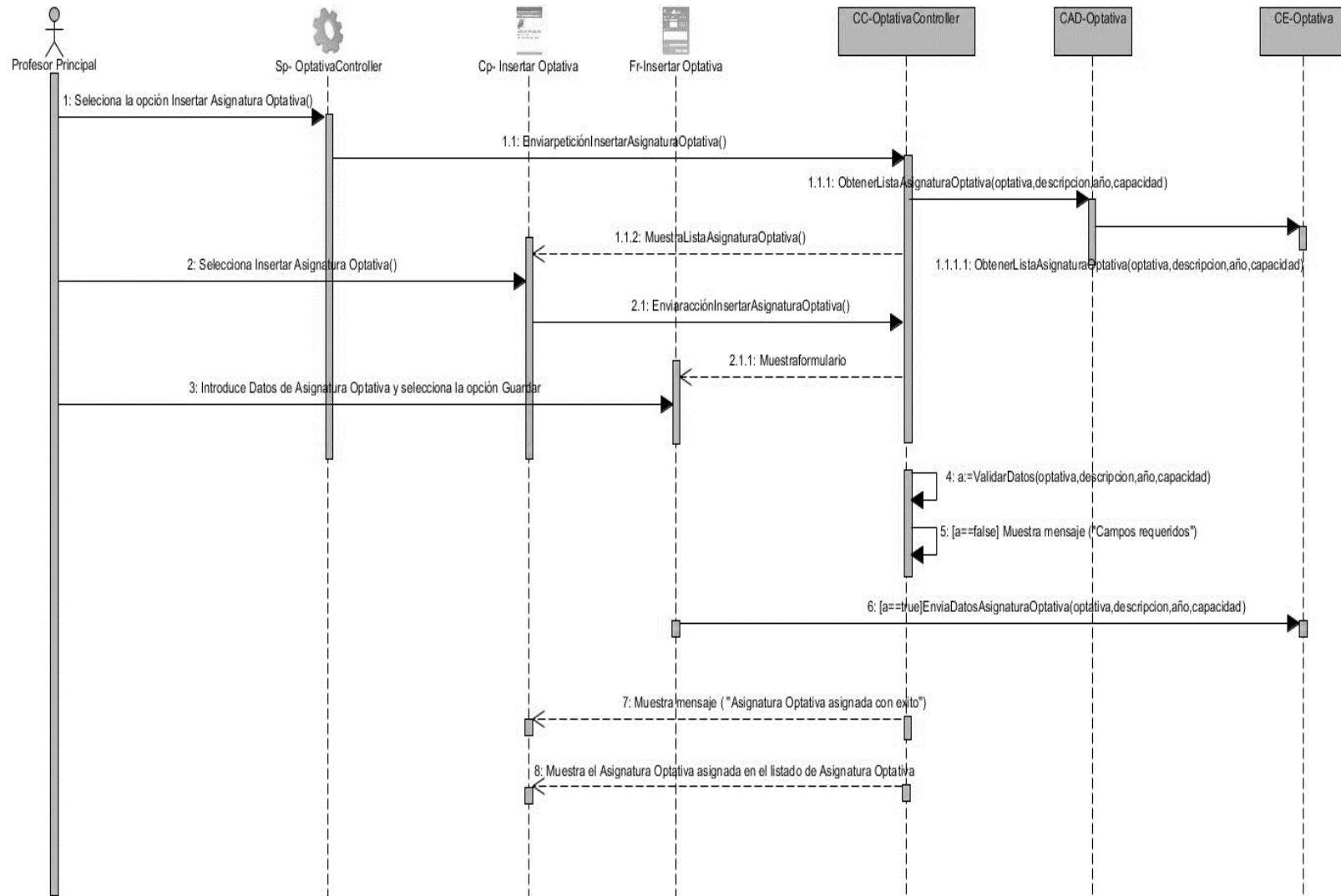


Figura 4. Diagrama de secuencia (Insertar Optativa)

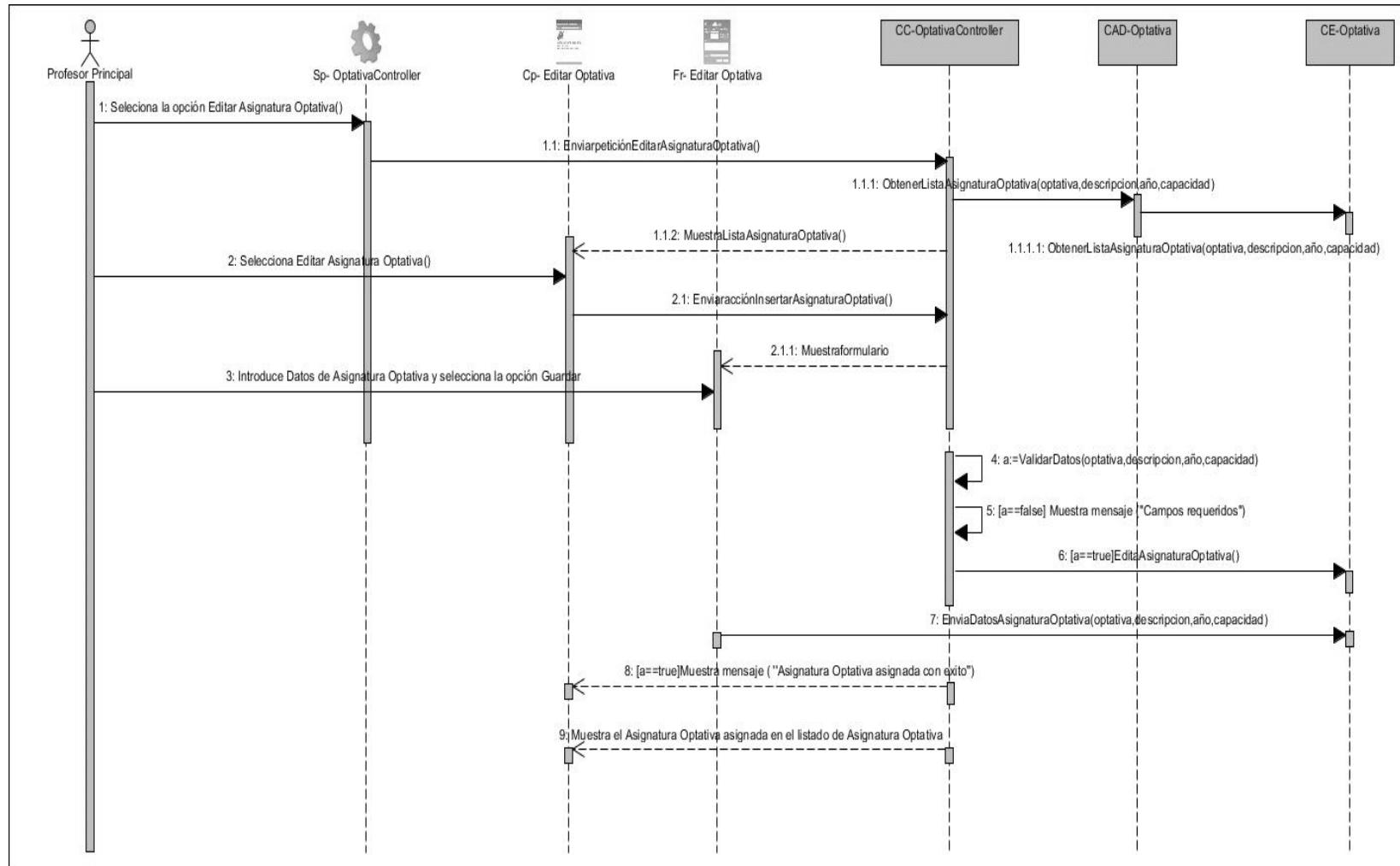


Figura 5. Diagrama de secuencia (Editar Optativa)

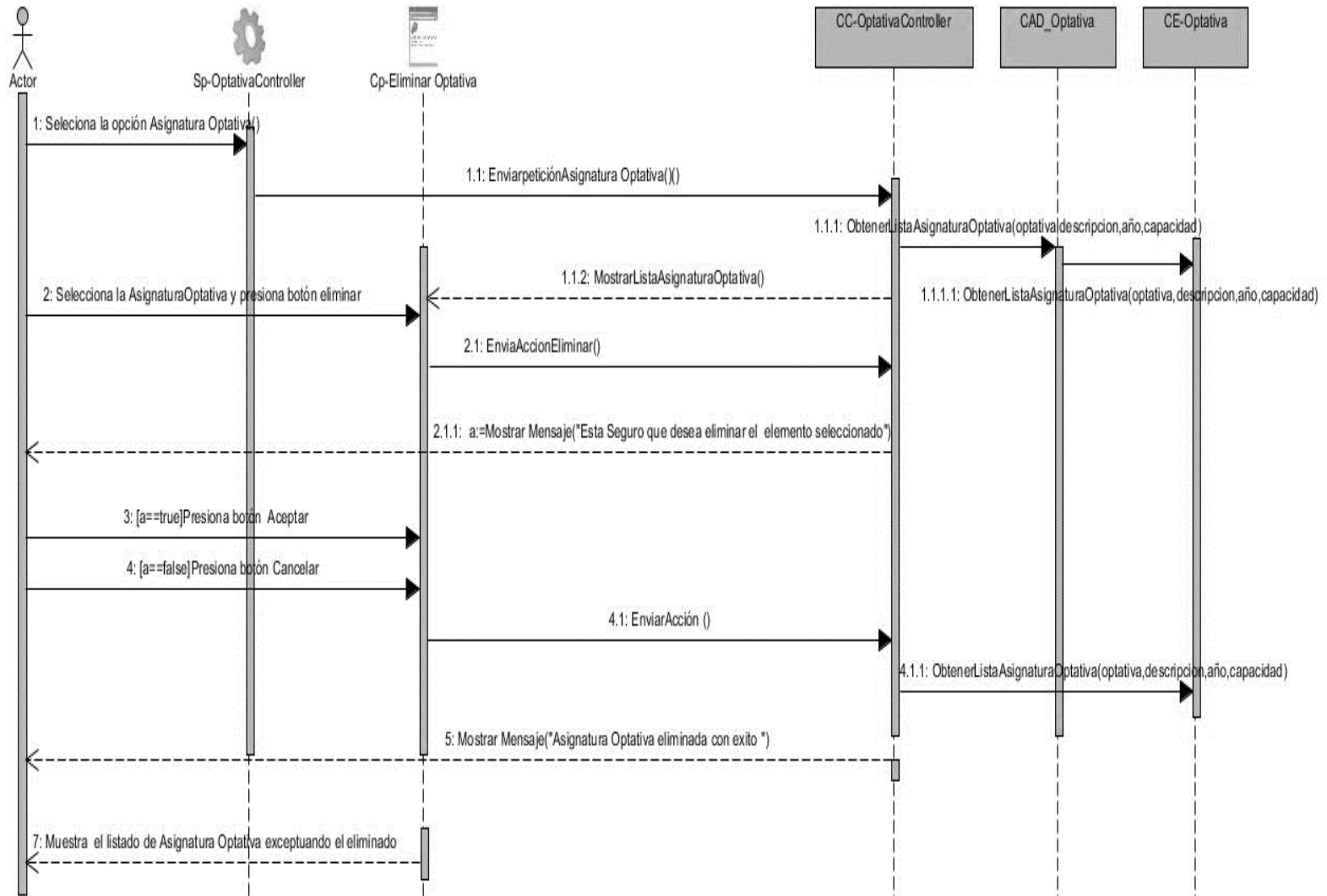


Figura 6. Diagrama de secuencia (Eliminar Optativa)

2.4.3 Modelo de Datos

Un modelo de datos es el proceso de documentar el diseño de un sistema de software cualquiera de manera gráfica, especialmente abarca lo relacionado a la base de datos, algunas tareas son definir los tipos de datos que habrá en una base de datos, identificar relaciones entre entidades, identificar las restricciones de integridad necesarias para evitar pérdida de consistencia. El modelado de datos es usualmente una de las primeras tareas que se deben completar en un proyecto de desarrollo de software, ya que es parte del análisis y diseño de un sistema informático (Pacheco y Subert, 2015).

En la Figura 7 se puede apreciar las relaciones existentes entre las tablas de la base de datos, ejemplo de esto la relación de mucho a mucho existente entre las tablas estudiantes y optativas, creando esto una nueva tabla llamada opt_ests; a su vez se origina una nueva tabla llamada opt_profs, producto de la relación entre las tablas optativas y profesores.

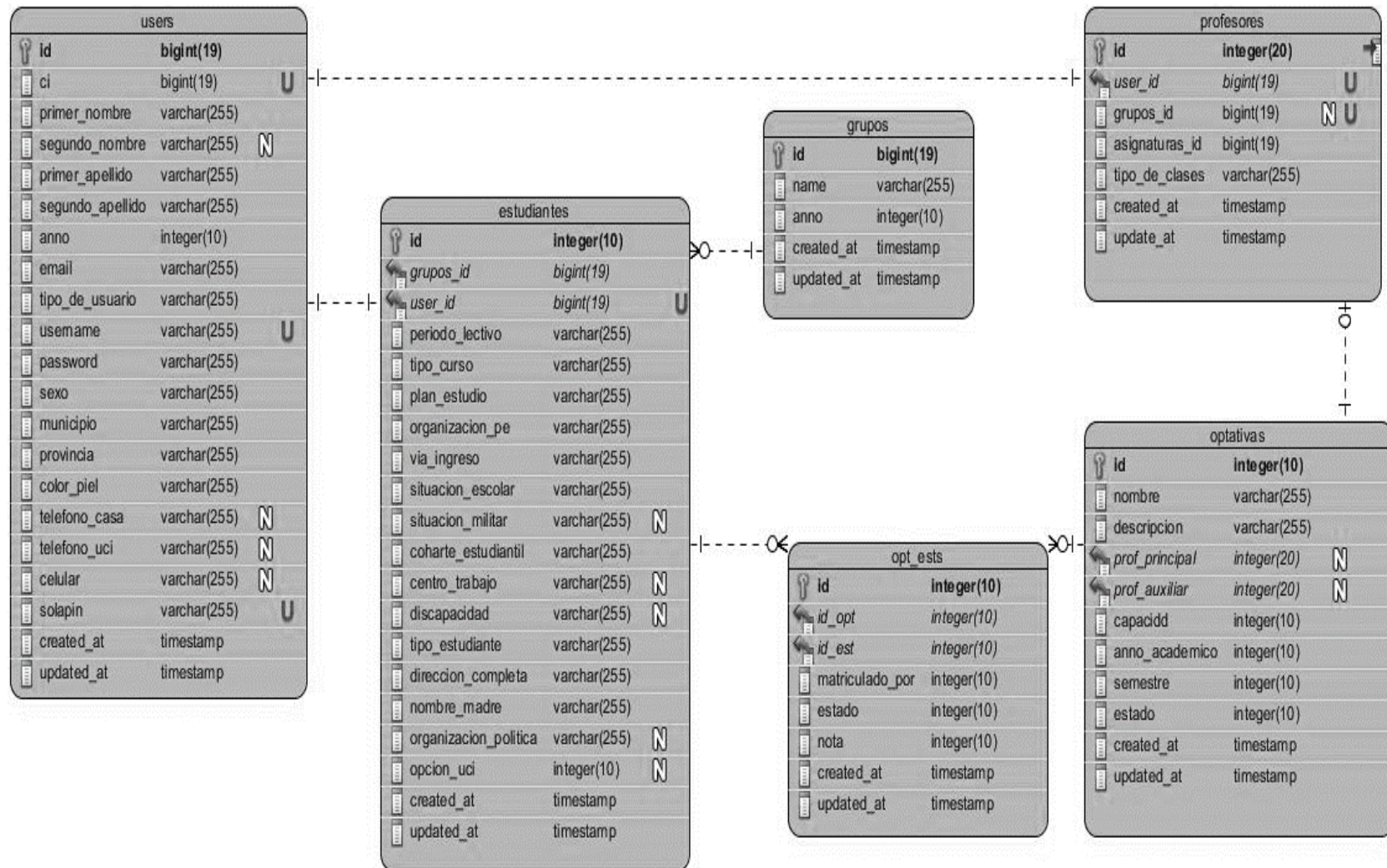


Figura 7. Modelo de datos

2.4.4 Arquitectura de software

El modelo representa la estructura de la base de datos con los atributos de la base de datos. La parte de la vista manejará la interfaz de usuario que se mostrará al usuario. La parte del controlador actuaría como conector entre la vista y el modelo, también maneja la parte de la entrada del usuario a través de la vista. Siendo su propósito esencial cerrar la brecha entre el modelo mental del usuario humano y el modelo digital que existe en la computadora. Además, tiene varios otros componentes útiles como seguridad, generación y validación de formularios, accesos a la base de datos y enrutamiento (Bautista, 2022).

En la Figura 8 se explica el funcionamiento de la arquitectura, y cómo se utilizaría dicha arquitectura en la presente investigación:

1. El cliente envía una petición desde la Vista Optativas.
2. La vista Optativas interactúa con el controlador OptativaController enviando los datos que solicite el cliente.
3. El controlador OptativaController envía al modelo Optativa los datos recibidos de la vista.
4. El modelo Optativa envía la respuesta al controlador OptativaController.
5. El controlador OptativaController envía la respuesta recibida del modelo Optativa hacia la vista Optativas.
6. La vista Optativas envía la respuesta de los datos recibidos del controlador al cliente.

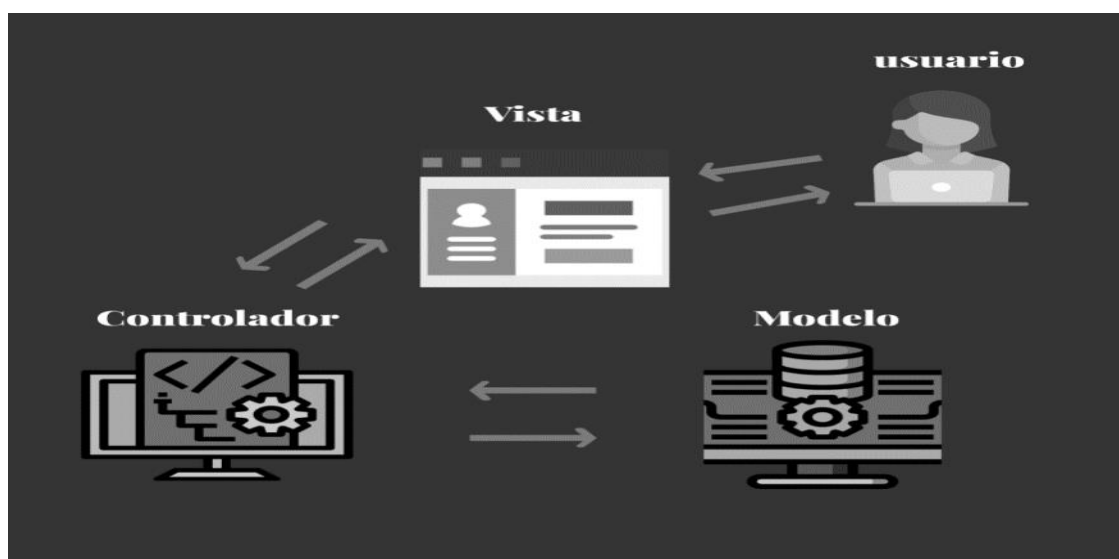


Figura 8. Arquitectura de SW: Modelo-Vista-Controlador

2.4.5 Patrones de diseño

Un patrón de diseño es una descripción o plantilla que generalmente te permite resolver un problema que se puede utilizar en muchas situaciones diferentes, el uso de patrones de diseño permite acelerar el proceso de desarrollo al proporcionar paradigmas probados y comprobados en el diseño de software. Estandarizar las soluciones a problemas comunes permite también mejorar la comunicación entre desarrolladores ya que se emplean nombres bien conocidos y entendidos a la hora de describir un problema de diseño y como abordar su solución (Martínez y Lugones, 2018).

Patrones generales de software para la asignación de responsabilidades

Estos patrones se utilizan generalmente en la programación orientada a objetos (POO). A continuación, se muestran los patrones de diseño GRASP existentes en la presente investigación:

- **Experto:** El patrón Experto se usa con frecuencia en la asignación de responsabilidades, es un principio básico que se utiliza continuamente en el diseño de objetos. En la figura que se muestra a continuación (Figura 9), se puede observar la clase Optativa, experta en información relacionada con las asignaturas optativas.

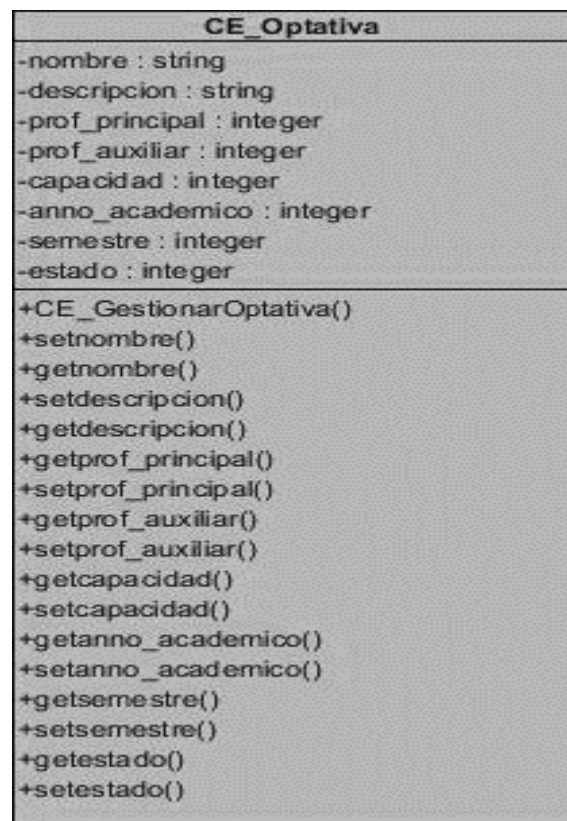


Figura 9. Empleo del patrón "Experto"

- **Alta Cohesión:** La alta cohesión existe cuando los métodos de una clase están bien relacionados entre sí y perfectamente delimitados. Un código altamente cohesionado tiende a ser más autocontenido y con menos dependencias. Mejora la **reutilización de código** y el **mantenimiento del software**. Esto se evidencia en todas las entidades Optativa, OptEst, OptProf.
- **Bajo Acoplamiento:** El bajo acoplamiento existe cuando se tiene poco nivel de relación y dependencias entre clases. Mejora la **mantenibilidad** de los módulos del software y la **reutilización** de las unidades del software, facilita los cambios en el software sin tener que revisar todos los módulos dependientes y las **pruebas unitarias** de cada módulo, al ser más independientes.
- **Controlador:** Es un patrón por el cual definimos objetos llamados controladores que independizan las interfaces con las acciones que haya que hacer. En la investigación se ve reflejado en todas las clases incluidas en la carpeta de *Controllers*, pues contienen todos los métodos necesarios para realizar y de alguna manera controlar los métodos existentes.
- **Creador:** Se basa principalmente en la asignación de responsabilidades relacionadas con la creación de objetos. Es el encargado de crear las instancias de los objetos, él mismo se evidencia en la clase *OptativaController* cuando se crean instancias de la entidad Optativa.

Patrones Gang of Four

- **Factory:** El patrón Factory, o patrón de diseño Factoría, describe un enfoque de programación que sirve para crear objetos sin tener que especificar su clase exacta. En la investigación, el patrón se evidencia en el *Opt_EstController*, siendo más específico en su método *destroy()*, el cual se comporta como una función especializada en dependencia del valor que reciba por parámetro, permitiendo el uso del mismo en diferentes secciones del código sin necesidad de repetir el contenido de la función.
- **Decorador:** Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (González, 2017). Se puede observar el uso de este patrón en la propuesta de solución en todas las vistas, ya que todas heredan de una plantilla padre determinada. En el presente segmento de código fuente se ejemplifica lo antes descrito:

En la vista **optativas** en los archivos *create.blade.php*, *edit.blade.php*, *index.blade.php*, *show.blade.php*: `@extends('adminlte::page')`


```

public function destroy($id)
{
    $value = explode('+', $id);
    $opt_est = DB::table('opt__ests')
->where('id_opt', $value[0])->where('id_est', $value[1])->value('id');
    $opt_est = Opt_Est::find($opt_est);
    $opt_est->delete();

    if ($value[2] == 'show') {
        return redirect('/optativas/' . $value[0]);
    }
    if ($value[2] == 'index') {
        return redirect('/opt_est');
    }
}

```

Figura 10. Uso del patrón "Factory"

2.4.6 Diagrama de despliegue

La figura 11 muestra el diagrama de despliegue que describe la arquitectura física del módulo de gestión de matrículas en las asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1 durante la ejecución, en términos de procesadores, dispositivos y componentes de software. Permite una mejor comprensión entre la correspondencia de la arquitectura de software y la arquitectura de hardware (Martínez y Lugones, 2018).



Figura 11. Diagrama de despliegue

Dispositivo del cliente: Contiene la estación de trabajo cliente que necesita un navegador Chrome o Mozilla Firefox, que conecte con la aplicación web (Apache) situado en el servidor (computadora personal, Tablet, teléfono celular), el cual utiliza para conectarse el protocolo de comunicación HTTPS (Protocolo Seguro de Transferencia de Hipertexto tiene presente un sistema operativo Linux o Windows, requieren de una tarjeta de red con velocidad de 100 Mbps, 2 GB de memoria RAM como mínimo y un procesador de 2 GHZ como mínimo).

Servidor Web: Aquí se encuentra el código fuente de la aplicación, brinda a los usuarios las interfaces necesarias durante todos los procesos del negocio. Esta estación se comunica con el servidor de base de datos MySQL donde se almacenan los datos de los perfiles de usuarios, realizando la comunicación a través del protocolo IP:8000. El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB como mínimo 1 GB de memoria RAM, requieren de una tarjeta de red con velocidad de 100 Mbps, 2 GB de memoria RAM como mínimo, un procesador de 2 GHZ como mínimo y el Servidor Web Apache con PHP 8.0.

Servidor de BD: Este nodo es el encargado del almacenamiento de los datos de los perfiles de usuario. Se comunica con el servidor de aplicaciones a través del protocolo IP, permite el acceso a la información que tanto el usuario como la aplicación pueden manipular, es el resultado almacenado de las iteraciones del software con una capacidad mínima de 20 GB, requieren de una tarjeta de red con velocidad de 100 Mbps, 2 GB de memoria RAM como mínimo, un procesador de 2 GHZ como mínimo y MySQL 8.0 como gestor de bases de datos.

Conclusiones del capítulo

Una vez realizado el análisis y diseño de la propuesta de solución, el estudio de los sistemas homólogos y las entrevistas realizadas en la Facultad 1 los profesores principales de cada año para la obtención de los requisitos funciones y no funcionales del módulo, el desarrollo de los diagramas de clases, de secuencias, de despliegue y el modelo de datos permitieron obtener una perspectiva en cuanto a la estructura y composición del Módulo para desarrollar el módulo.

Capítulo III: Implementación y validación del módulo de gestión de matrículas en asignaturas optativas para el sistema de trabajo educativo de la facultad 1

La implementación de un módulo para un sistema de gestión educativo constituye una de las fases más importantes del desarrollo de software, debido a que en ella se toman los resultados obtenidos en el diseño como punto de partida, con el principal objetivo de lograr obtener el mejor resultado posible durante la implementación del sistema o módulo en términos de componentes como ficheros de código binario, código fuente, scripts y ejecutables. Además, se debe tener presente realizar actividades que garanticen la calidad del módulo en el cual se está trabajando. Las pruebas de validación constituyen una base para garantizar la aceptación favorable de una aplicación informática por parte del usuario. Este capítulo tiene como objetivo, documentar los resultados de las fases de implementación del módulo y de la estrategia de pruebas desarrollada, además de realizar la técnica de ladov para conocer el índice de satisfacción general del módulo.

3.1 Diagrama de Componentes

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico (Martínez y Lugones, 2018).

En la figura 11, se observa el diagrama de componentes correspondiente al Módulo del Sistema de Trabajo Educativo de la Facultad 1 para la gestión de las asignaturas optativas, elaborado y estructurado de acuerdo con la arquitectura Modelo-Vista-Controlador del *Framework* de desarrollo de PHP Laravel. La descripción de cada componente se detalla en la tabla 5.

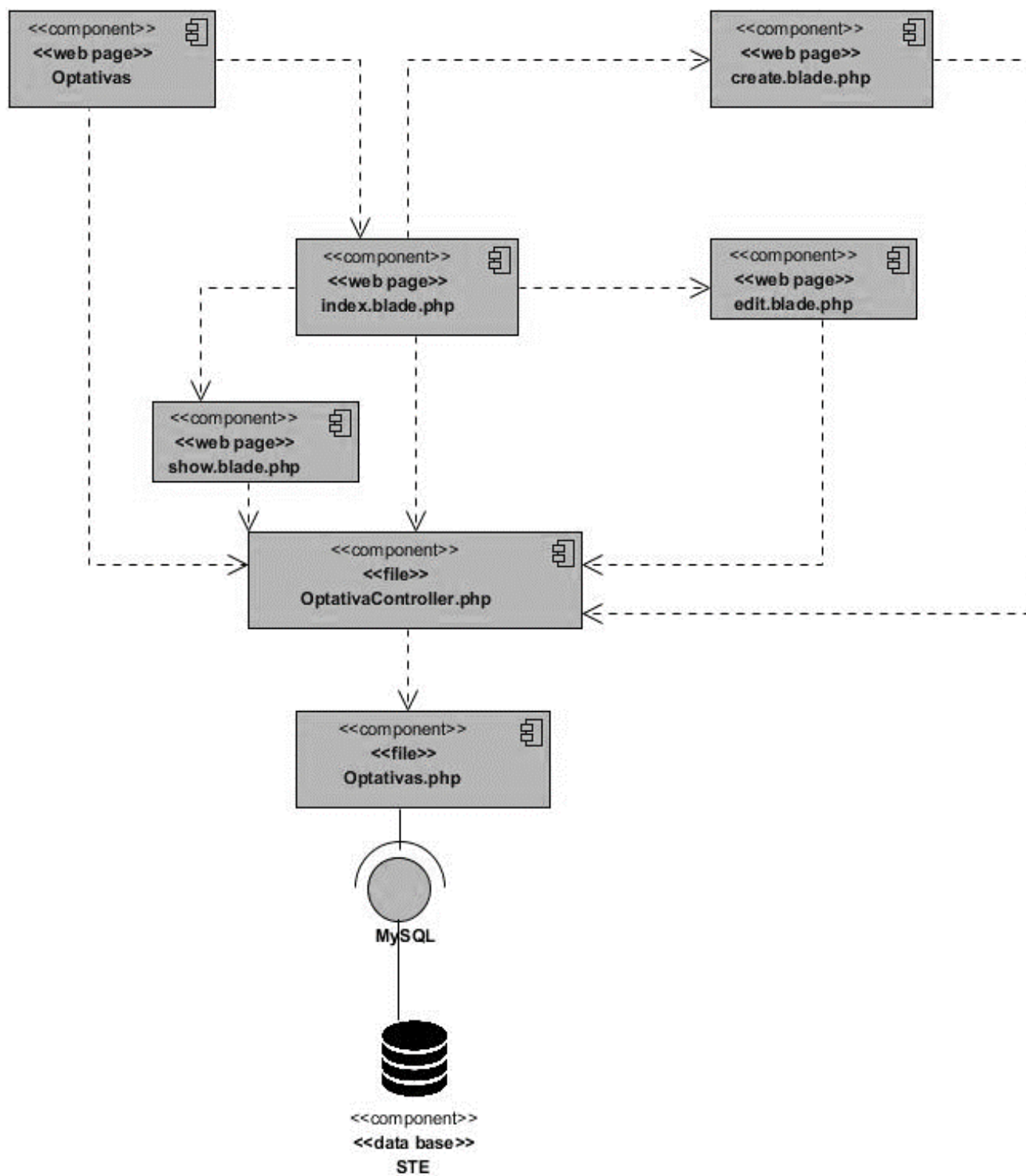


Figura 12. Diagrama de componentes del módulo

Tabla 5. Descripción del diagrama de componentes

Componentes		Descripción
Modelo	Optativa.php	Clase entidad que almacena los datos correspondientes a las Optativas insertadas en la base de datos.
Vistas	Optativa	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las optativas.
	index.blade.php	Este componente se encarga de mostrar un listado de las optativas insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar las optativas.
	create.blade.php	Este componente se encarga de registrar una nueva optativa en la base de datos con los datos necesarios de las mismas.
	edit.blade.php	Este componente se encarga de editar una optativa en la base de datos con los datos necesarios de las mismas.
	show.blade.php	Este componente se encarga de mostrar información detallada de la optativa, estudiantes matriculados profesor que la imparte.
Controlador	OptativaController.php	Este componente se encarga de realizar todos los métodos relacionados a la gestión de las optativas en el módulo.

3.2 Estándares de Codificación

Los estándares de código o *Code Standards* en inglés, son parte de las llamadas buenas prácticas o mejores prácticas. Estas son un conjunto no formal de reglas, que han ido surgiendo en las distintas

comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad del código (Catallops y José, 2019).

Tabla 6. Estándares de codificación

Tipo de estándar	Descripción
Nomenclatura de las Clases	<p>Los nombres de las clases empiezan con mayúsculas y el resto del nombre con minúscula, en caso que sea un nombre compuesto la segunda palabra también comenzará con mayúscula. Ejemplo de esto es la clase <i>Optativa</i>, donde se puede observar que la clase posee un solo nombre y comienza con mayúsculas.</p>
Nomenclatura según el tipo de clases	<p>El nombre de las clases controladoras se encontrará seguido de la palabra <i>Controller</i>. Ejemplo de esto la clase: <i>OptativaController</i>.</p> <p>El nombre de los modelos utilizados dentro de la programación del módulo empieza con letra inicial mayúscula. Ejemplo de esto el modelo <i>OptEst</i>.</p>
Estándares para comentarios	<p>Los comentarios deben ser oraciones completas.</p> <p>Si un comentario es una frase u oración, debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.</p> <p>Los comentarios de una línea se encontrarán seguidos de los caracteres <code>//</code> en caso de código JavaScript, y deben ubicarse en la misma línea.</p> <p>Los comentarios de varias líneas para organización del código aparecerán dentro de los caracteres <code>/** ... */</code> en caso de que se utilice código JavaScript.</p>

<p>Estándares para espacios en blanco en expresiones y sentencias</p>	<p>No utilizar espacios en blancos en los siguientes escenarios:</p> <p>Dentro de paréntesis, corchetes y llaves.</p> <p>Antes de una coma, un punto y coma, o dos puntos.</p> <p>Antes del paréntesis que comienza la lista de argumentos en la llamada a una función.</p> <p>Deben rodearse con un espacio los siguientes operadores binarios:</p> <p>Asignación (=).</p> <p>Asignación de aumentación o de resta (+=, -=).</p> <p>Comparación (===, !==, ==, <, >, >=, <=, !=, <>, <i>in</i>, <i>not in</i>, <i>is</i>, <i>is not</i>).</p> <p>Expresiones lógicas (<i>and</i>, <i>or</i>, <i>not</i>).</p> <p>Añadir un espacio después de cada delimitador “,”.</p>
<p>Otros Estándares</p>	<p>Añadir un salto de línea antes de cada sentencia <code>return ()</code>, a menos que el <code>return ()</code> se encuentre solo en un bloque de sentencias.</p> <p>Añadir un salto de línea después de cada llave cierre de sentencia, excepto después de la llave cierre de clases.</p> <pre> 191 public function destroy(\$id) 192 { 193 \$optativa = Optativa::find(\$id); 194 195 \$optativa->delete(); 196 197 return redirect()->route('optativa.index')->with('info', 'eliminar-afectacion'); 198 } </pre> <p>Utilizar los <i>namespace</i> para los <i>Controller</i> y los <i>Models</i></p> <pre> 2 3 namespace App\Http\Controllers\Modulo_Optativas; 4 </pre>

```

2
3 namespace App\Models\Modulo_Optativas;
4

```

3.3 Estrategia de Pruebas

La fase de pruebas engloba un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Dentro de esta fase pueden desarrollarse varios tipos distintos de pruebas en función de los objetivos de las mismas. Algunos tipos son pruebas funcionales, pruebas de integración, pruebas de seguridad, entre otras (Pérez, 2017).

Tabla 7. Resumen de estrategia de pruebas

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

3.3.1 Pruebas Funcionales

Las pruebas funcionales se centran en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente. Este servicio ayuda a detectar los posibles defectos derivados de errores en la fase de programación (García, 2016).

Con el objetivo de realizar este tipo de pruebas al Módulo del Sistema de Trabajo Educativo de la Facultad 1 para la gestión de las asignaturas optativas, se planificó un conjunto de casos de pruebas, referentes a varios casos de uso del sistema del capítulo anterior, pertenecientes a requisitos funcionales de prioridad alta y media, también especificados en dicho capítulo. A continuación, se muestra uno de los casos de prueba mencionados, analizando de la misma varios campos de la tabla para llevar a cabo dicha prueba. En las celdas de la tabla del caso de prueba se pueden encontrar los valores **V**, para datos válidos, **I**, para datos inválidos, y **N/A**, para datos a los que no es necesario proporcionarles un valor.

Descripción de las variables:

Tabla 8. Descripción de las variables del caso de prueba 1 para el RF - Insertar Optativa

Variable	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	no	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños.
2	Profesor Principal	Campo de selección	sí	Se debe escoger uno de los profesores disponibles.
3	Capacidad	Campo numérico	no	Se debe escribir la capacidad.
4	Estado	Campo de selección	no	Se debe escoger el estado de la optativa.

Tabla 9. Caso de prueba del RF - Insertar Optativa

Caso de prueba 1: RF1_ Insertar Optativa.							
Condiciones de ejecución: El usuario debe tener los permisos necesarios.							
Escenario	Descripción	1	2	3	4	Respuesta del sistema	Flujo Central
o							

EC 1.1	Interfaz con el formulario para llenar los campos de la optativa, si todos son correctos, se agrega la optativa al sistema.	V	V	V	V	Agrega la optativa y muestra un mensaje de notificación.	1. Seleccionar, opción “Gestión de Optativas”. 2. Seleccionar la opción “Insertar Optativa”. 3. Llenar los campos correspondientes en el formulario y seleccionar la opción “Guardar”.
		Programación	Robert Paz	30	Activa		
EC 1.2	Interfaz con el formulario para llenar los datos de la optativa, si existe algún campo vacío, se muestra un mensaje debajo de cada campo señalando que el campo es requerido.	N/A	V	N/A	V	Comprueba si existen campos vacíos, si lo están, muestra un mensaje que debajo de cada campo señalando que el campo es requerido y por lo tanto debe ser llenado.	
			Robert Paz		Activa		
		V	V	I	V		

EC 1.3 Insertar datos de la optativa con campos erróneos.	Interfaz con el formulario para llenar los datos de la optativa, si existe algún campo erróneo, se muestra un mensaje debajo de cada campo señalando que el campo tiene un error.	Programación Orientada a Objetos	Robert Paz Paz	treinta	Activa	Comprueba si los campos tienen errores, si los tienen, muestra un mensaje que debajo de cada campo señalando que el campo tiene un error y por lo tanto debe ser llenado correctamente.	
--	--	---	----------------------	---------	--------	--	--

Las pruebas funcionales se realizaron en 3 iteraciones, permitió conocer la calidad del módulo que se está construyendo. Como resultado final se obtuvo un total de 10 no conformidades.

En la primera iteración se encontraron **4** no conformidades de **ortografía** y **3** no conformidades de **redacción**, las cuales fueron resueltas.

En la segunda iteración se encontraron **3** no conformidades, **2** de **validación** al insertar una optativa y **1** de **funcionalidad** al matricular un estudiante.

No conformidades de **validación**: El sistema permitía insertar una optativa con el campo profesor principal y el campo profesor auxiliar iguales y no permitía insertar una optativa con al menos uno de los campos profesor principal o profesor auxiliar en blanco. Estas no conformidades fueron resueltas haciendo una correcta validación de los campos con JavaScript.

No conformidad de **funcionalidad**: El sistema no permitía matricular un estudiante en una optativa. Esta no conformidad fue resuelta haciendo un nuevo análisis revisión del código hasta encontrar el error funcional en el código.

En la tercera iteración no se detectaron nuevas No conformidades, por lo que una vez finalizadas las pruebas de funcionalidad se puede concluir que el sistema funciona correctamente y realiza todas las tareas para las cuales fue programado. Todos los resultados anteriormente descritos se pueden observar en la Figura 13:

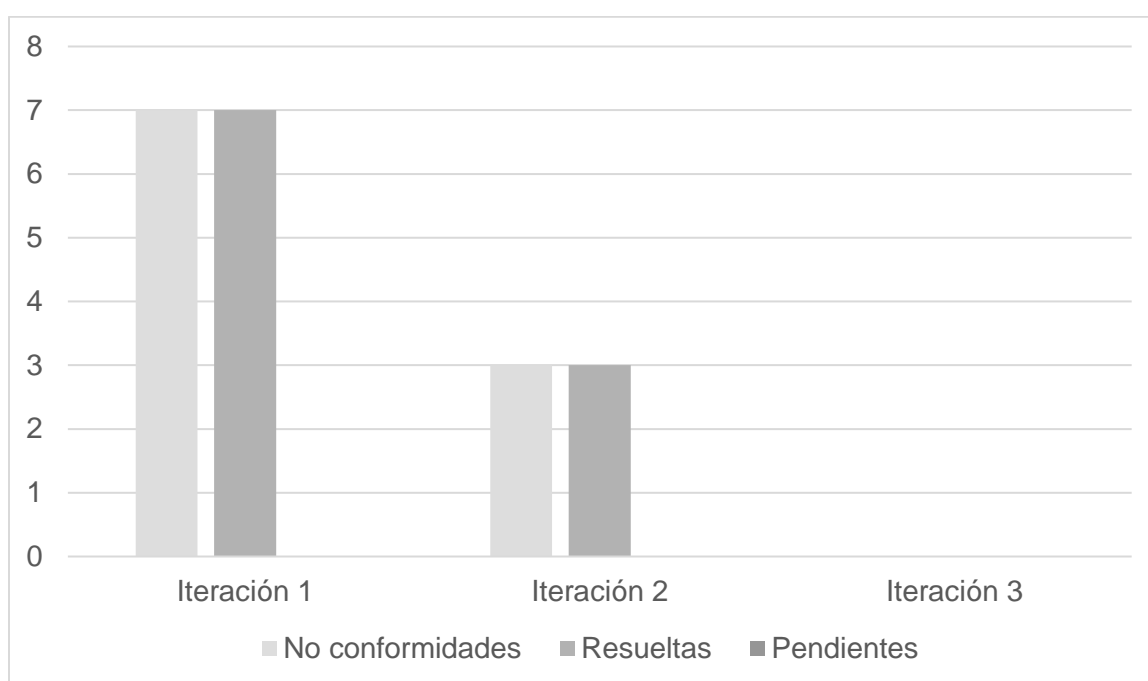


Figura 13. Resultados de pruebas de funcionalidad aplicadas

3.3.2 Pruebas de Seguridad

Las pruebas de seguridad son el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones y sistemas de gestión web, buscando disminuir el impacto de ataques a ellas y pérdida de información sensible para el sistema. La seguridad en aplicaciones web busca asegurar la confidencialidad, disponibilidad e integridad de los datos y funciones que maneja el software (Baró y Carla, 2019).

Con el objetivo de realizar una exitosa prueba de seguridad de la propuesta de solución desarrollada, se tuvo en cuenta la herramienta *Acunetix Web Vulnerability Scanner*, herramienta la cual fue descrita con anterioridad en el epígrafe 1.5. Durante la primera iteración realizada, la aplicación detectó un total de 18 no conformidades, divididas en tres categorías: nivel medio (12 no conformidades), nivel bajo (3 no conformidades) y de carácter informal (3 no conformidades).

Las no conformidades de nivel medio, fueron ocasionadas por el uso de protocolos no seguros para el envío de datos, y el envío de mensajes de error utilizados por Laravel. Las no conformidades de nivel bajo, están relacionadas a los posibles ataques a la página de autenticación y las posibles redirecciones que pueden surgir a partir de la misma una vez que el usuario está dentro del sistema. De tipo Informativo se obtuvo varios errores, relacionados a los campos correspondientes a las contraseñas y los campos repetir contraseña.

Luego de analizados los errores detectados se puede concluir que, tanto los de mensajes, como los de urls, son debido a mensajes de error y de redirección propios del Framework Laravel utilizado en la implementación, los cuales se utilizan para distintas funcionalidades dentro del sistema, por lo que en la segunda iteración de la aplicación no se obtuvo ninguna deficiencia en la seguridad del sistema, permitiendo esto poder utilizar la herramienta desarrollada como propuesta de solución de manera segura. A continuación, se pueden observar los resultados antes descritos:



Figura 14. Resultados de pruebas de seguridad

Una vez finalizadas las pruebas de seguridad se puede concluir que el sistema cumple con todas las normas de seguridad planificadas en los requisitos no funcionales de seguridad descritos en el capítulo 2.

3.3.3 Pruebas de Integración

Las pruebas de integración se pueden definir como un mecanismo de testeo de software, en el cual se realiza un análisis de los procesos relacionados con el ensamblaje o unión de los componentes, sus comportamientos con múltiples partes del sistema o de hardware. Las pruebas de integración están a cargo del examen de las interfaces entre los subsistemas o los grupos de componentes del programa o aplicación que se analiza, lo que contribuye a garantizar su funcionamiento correcto (Baró y Carla, 2019).

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

- Integración del Módulo del Sistema de Trabajo Educativo de la Facultad 1 para la gestión de las asignaturas optativas.

La figura 15 muestra los resultados de la ejecución de las dos iteraciones de las pruebas de integración realizadas al sistema:

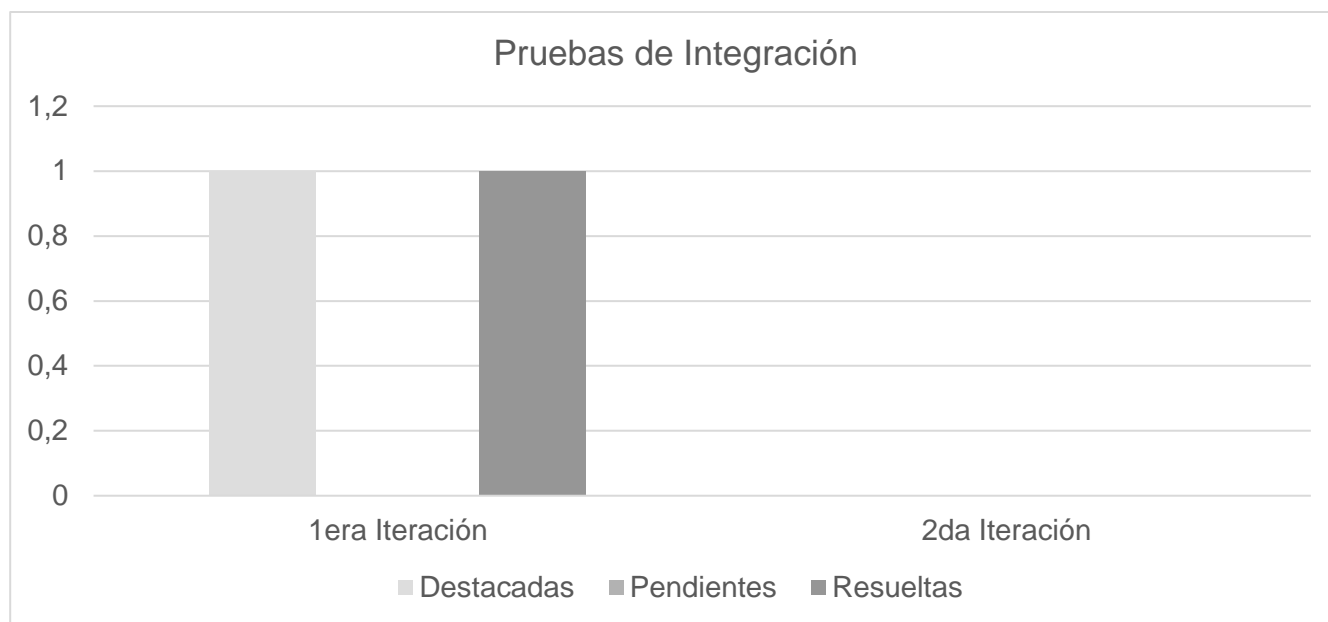


Figura 15. Resultados de las pruebas de integración

Se realizaron 2 iteraciones para lograr integrar el Módulo del Sistema de Trabajo Educativo de la Facultad 1 para la gestión de las asignaturas optativas.

Al intentar integrar el módulo, ocurrió un error porque se hacía referencia a algunas variables que no estaban definidas en el sistema y no existía relación entre los modelos de datos.

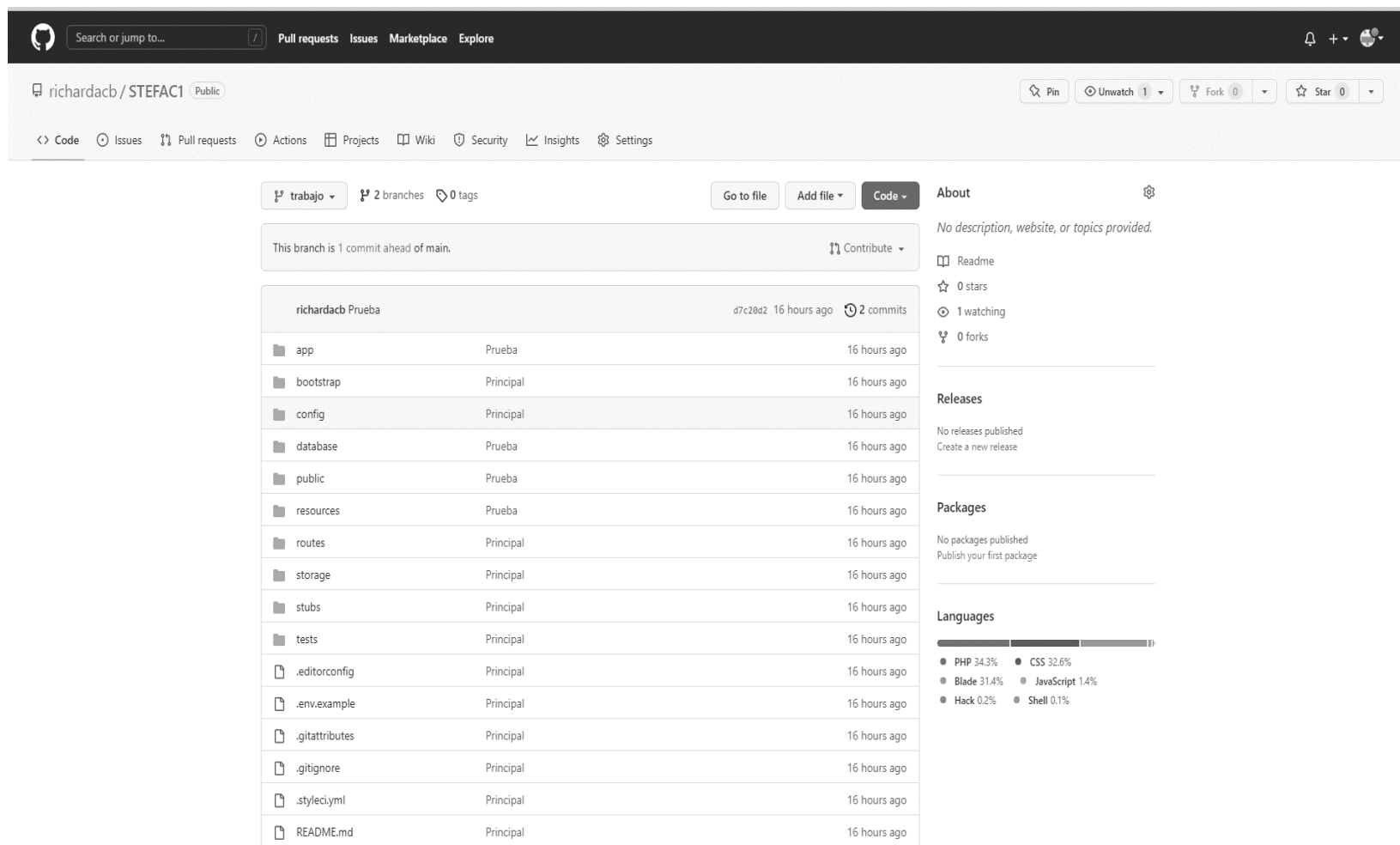


Figura 16. Pruebas de integración

Luego de integrar el Módulo del Sistema de Trabajo Educativo de la Facultad 1 para la gestión de las asignaturas optativas, se pudo comprobar la correcta funcionalidad del mismo con el resto de los módulos que componen el sistema.

3.3.4 Pruebas de Carga y Estrés

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

Para la realización de esta prueba se utilizó la herramienta *Apache JMeter*, descrita en el epígrafe 1.3. Las pruebas se realizaron desde un ordenador con 8GB de RAM, microprocesador Intel Core i5 7th Gen, con 2.00 GHz y sistema operativo Windows 10. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

Muestra: Cantidad de peticiones realizadas para cada URL.

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

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

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

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

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

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

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

Kb/s: Velocidad de carga de las páginas.

En la tabla 10 se muestra información sobre las pruebas de carga y estrés realizadas al módulo, se simuló peticiones para un total de 50, 100 y 150 usuarios simultáneamente.

Tabla 10. Resultados de las pruebas de carga y estrés

Usuarios	Muestras	Media	Mediana	Min	Max	Línea 90%	%Error	Rend	Kb/s
50	132	16 503	16 512.35	65	48466	2902	0.02%	5	6276.7
100	163	22 779	18 542.44	69	48763	2591	0.50%	10	5717.6
150	190	24769	20 745.30	73	48935	2193	1.25%	15	5434.7

Las pruebas realizadas muestran que el módulo es capaz de responder a 132 peticiones de 50 usuarios conectados simultáneamente en un tiempo promedio de 16503 milisegundos (1.6 segundos aproximadamente) con 0.02 % de error, esto evidencia que el módulo puede procesar la carga esperada.

Por otra parte, se realizaron 163 peticiones iniciadas por 100 usuarios y en este caso el módulo respondió en 22779 milisegundos (2.2 segundos aproximadamente) como tiempo promedio. Esto demuestra que el módulo puede procesar la carga esperada, aunque no fue capaz de responder correctamente el 0.50% de las peticiones realizadas.

Por último, y con el objetivo de analizar el comportamiento del módulo en condiciones extremas, se realizó una prueba de estrés para un conjunto de 150 usuarios conectados simultáneamente. En este caso, el módulo responde a las 190 peticiones en un tiempo promedio de 24769 milisegundos (2.4 segundos aproximadamente), pero con un porcentaje de error de 1.25%. Este resultado está estrechamente relacionado al entorno donde se realizó la prueba, el cual no es un servidor dedicado sino un cliente habilitado para realizar la prueba.

Una vez finalizadas las pruebas de carga y estrés realizadas al Sistema de Trabajo Educativo de la Facultad 1, se puede concluir que el sistema actualmente soporta la interacción simultánea de varios usuarios, lo que se considera como un punto positivo que permitirá el correcto uso del sistema.

3.4 Interfaces principales del Módulo de gestión de asignaturas optativas.

Luego de desarrollado el módulo de gestión de asignaturas optativas, en las siguientes figuras se puede visualizar el resultado obtenido una vez terminada la implementación:

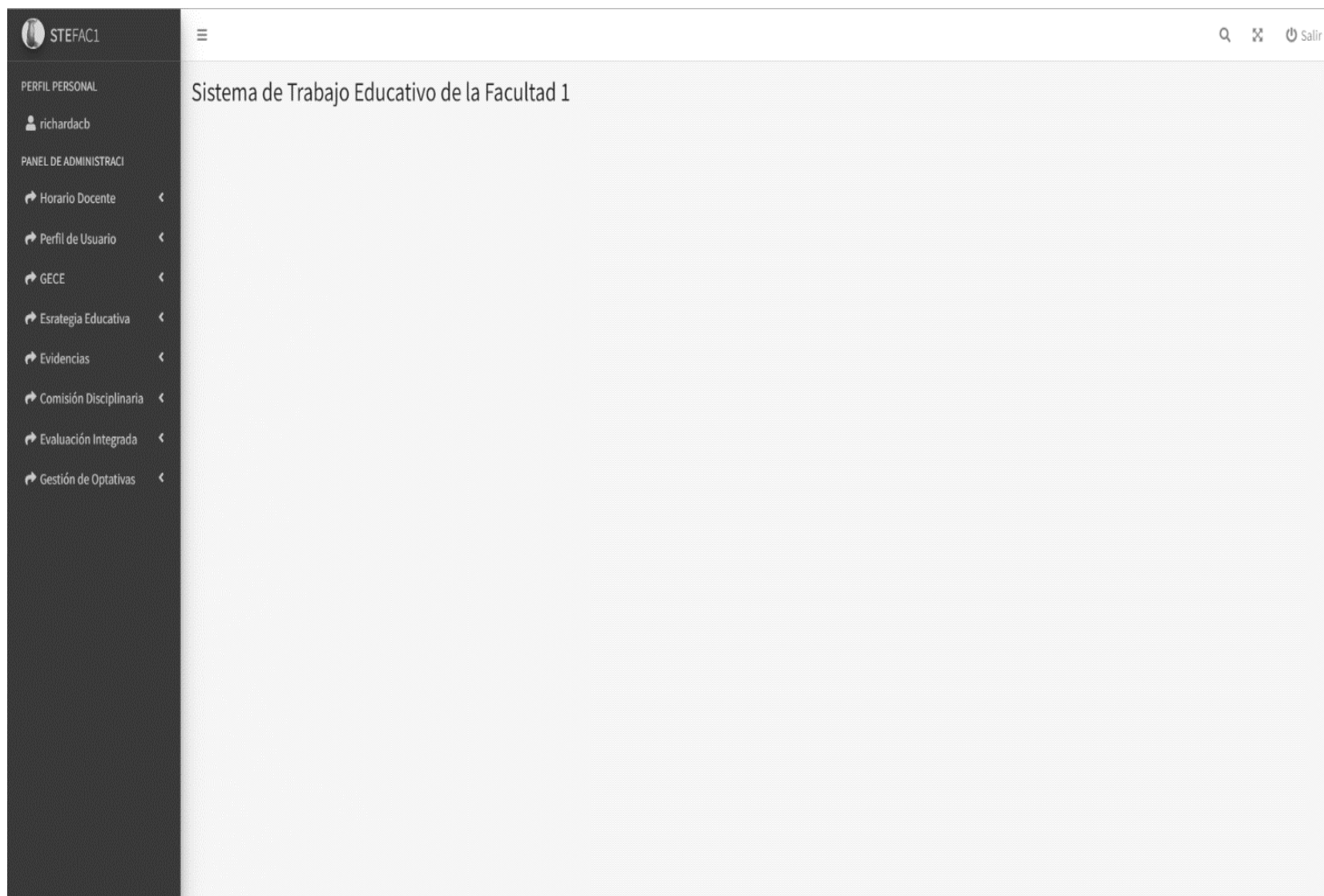


Figura 17. Pantalla principal del módulo desarrollado

PERFIL PERSONAL
richardacb




PANEL DE ADMINISTRACIÓN

- Horario Docente
- Perfil de Usuario
- GECE
- Estrategia Educativa
- Evidencias
- Comisión Disciplinaria
- Evaluación Integrada
- Gestión de Optativas**
 - Optativas
 - Optativas Ofertadas
 - Optativas Asignada

Listado de optativas

Mostrar 5 Entradas Agregar Optativa

Buscar:

Optativa	Descripción	Profesores	Capacidad	Año académico	Semestre	Estado	acciones
GPI	Gestión de Procesos Informáticos	Daniel Jesus Penalver Quintana Jose Carlos Milanes Anaya	20	4	5	1	  

Mostrando 1 a 1 de 1 Entradas Anterior 1 Siguiente

Figura 18. Pantalla de gestión de optativas

3.5 Validación Científica

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

Tabla 11. Cuadro lógico de ladov

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

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1

de la siguiente forma: Máxima satisfacción (+ 1), Satisfecho (0.5), No definido (0), Insatisfecho (- 0.5) y Máxima insatisfacción (- 1). A partir de la cantidad de respuestas es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

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

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

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

Los resultados obtenidos fueron los siguientes:

Tabla 12. Resultados de las escalas de satisfacción

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

Como resultado del cálculo del ISG, se obtuvo:

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

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

Conclusiones del capítulo

En el presente capítulo una vez abordados todo lo relacionado a la implementación, se puede concluir que la elaboración y desarrollo de los diagramas de componentes posibilita adquirir una mejor comprensión del módulo a implementar. Por otra parte, el adecuado uso de los estándares de codificación definidos permite lograr que el código fuente del módulo se encuentre estructurado de una manera legible, lo que a su vez conlleva a la obtención de una aplicación correctamente desarrollada. Por otra parte, la estrategia de pruebas definidos, junto a la validación de las mismas han permitido obtener resultados positivos acerca de la propuesta de solución desarrollada, permitiendo así que el mismo cumpla con los requisitos funcionales y no funcionales definidos junto al cliente.

CONCLUSIONES GENERALES

La presente investigación termina con la confección de un Módulo para la Gestión de Asignaturas Optativas para el Sistema de Trabajo Educativo de la Facultad 1, contribuyendo el mismo al desarrollo del uso de las tecnologías de la información y las comunicaciones en la Facultad. Entre las principales características que se pueden mencionar del módulo desarrollado se encuentran:

1. Con el análisis de los conceptos relacionados con la investigación se logró un mejor entendimiento y recopilación de información referente al tema que se aborda, además, el estudio de los sistemas homólogos llevó a la conclusión de que, a pesar de no resolver el problema, existen sistemas tanto a nivel internacional como nacional que pueden servir de punto de partida.
2. El análisis y elección tanto de las herramientas, tecnologías, así como la metodología escogida para desarrollar la presente investigación, han permitido abordar un poco más sobre las posibles propuestas de solución para dar respuesta al problema planteado.
3. La relación existente entre varias áreas de la programación y la ingeniería de software, permitió realizar de manera exitosa el correcto diseño e implementación del Módulo de Gestión de Asignaturas Optativas de la Facultad 1.
4. La estrategia de pruebas definidos, junto a la validación de las mismas han permitido obtener resultados positivos acerca de la propuesta de solución desarrollada, permitiendo así que el mismo cumpla con los requisitos funcionales y no funcionales definidos junto al cliente. Además, con la validación científica llevada a cabo mediante el uso de la técnica de ladov, se pudo concluir que el módulo desarrollado responde al problema de la investigación.

RECOMENDACIONES

REFERENCIAS BIBLIOGRÁFICAS

- Almenares, L. S. (2012). Prueba Automática de Carga y Estrés. *Serie Científica de la Universidad de las Ciencias Informáticas*, 5(1), Article 1. <https://publicaciones.uci.cu/index.php/serie/article/view/339>
- Baque Jaramillo, E. J., & Herrera Narváez, J. A. (2021). *Desarrollo e Implementación de una aplicación para gestión De matrículas y notas de la Escuela de Educación Básica Zoila Rosa García Montenegro*. <http://dspace.ups.edu.ec/handle/123456789/19757>
- Baró, D. R., & Carla, A. (2019). *Sistema para la gestión del banco de problemas científicos de la Facultad 1*. <https://repositorio.uci.cu/jspui/handle/123456789/9998>
- Bautista-Villegas, E. (2022, enero 25). Metodologías ágiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel. *25/01/2022, Vol. 1 Núm. 1: e168, 7*.
- Burgos Cano, A. L., & Contreras Chacón, M. C. (2022). *Desarrollo de un prototipo de sistema de matriculación para un instituto tecnológico mediante un aplicativo web y un aplicativo móvil*. <http://bibdigital.epn.edu.ec/handle/15000/22196>
- Cabrera, A., & Noel, R. (2015). *Sistema informático para la gestión del proceso de matrícula en las asignaturas optativas de la Facultad 1 de la Universidad de las Ciencias Informáticas*. <https://repositorio.uci.cu/jspui/handle/123456789/7043>
- Catallops, S., & José, L. (2019). *Sistema de gestión de información tecnológica para la Facultad 1 de la Universidad de las Ciencias Informáticas*. <https://repositorio.uci.cu/jspui/handle/123456789/10067>
- Cevallos, M., & Andres, C. (2022). *Desarrollo de aplicación web para la gestión de socios de fundación FUHPED*. <http://bibdigital.epn.edu.ec/handle/15000/22059>

- Espinoza matamoros, J. L. (2019). *Desarrollo e implementación de una aplicación web para la gestión y control en los procesos de adquisiciones del departamento de informática de la autoridad de tránsito municipal*. <http://dspace.ups.edu.ec/handle/123456789/18069>
- Fajardo Quesada, A. J., & Licea González, M. de los Á. (2022, mayo 10). IMPLICACIÓN DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES EN LA ENSEÑANZA- APRENDIZAJE DE LA MEDICINA. *Ciro Redondo 2022*. *Ciro Redondo 2022*. <https://jccredondo2021.sld.cu/index.php/jccredondo/2021/paper/view/73>
- Gaceta Oficial de la República de Cuba. Resolución No.47/2022*. (2022). https://www.gacetaoficial.gob.cu/sites/default/files/goc-2022-o47_0.pdf
- García Gómez, A. (2016). *Versión 2.0 del componente Personal del Sistema de Gestión Universitaria, Akademos*. <https://repositorio.uci.cu/jspui/handle/123456789/7586>
- Gutiérrez, L. V. V. (2020). Gestión educativa y su relación con el desempeño docente. *Ciencia y Educación - Revista Científica*, 1(2), 18-28. <https://doi.org/10.48169/Ecuatesis/0102202008>
- Hernández, L. R. B., Peña, D. M., Valdés, O. R., & Cornelio, O. M. (2016). Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso. *Serie Científica de la Universidad de las Ciencias Informáticas*, 9(7), Article 7. <https://publicaciones.uci.cu/index.php/serie/article/view/822>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación* (6ta edición). Mc Graw Hill Education. <https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>
- HTML & CSS - W3C*. (2022, mayo 18). <https://www.w3.org/standards/webdesign/htmlcss>
- Juneau, C. (2022). Cook-it!: A Web Application for Easy Meal Planning. *Senior Theses*, 1-16.

- Martínez Álvarez, A. C., & Lugones Iznaga, L. M. (2018). *Módulo de Gestión de Transporte para la Cadena de Suministros en el sistema de gestión DISTRA*. <https://repositorio.uci.cu/jspui/handle/123456789/9917>
- Morales, D. B., Borrell, J. B., & Armas, L. J. (2019). Aplicación móvil para el análisis de la información captada en SIGEv3.0. *Serie Científica de la Universidad de las Ciencias Informáticas*, 12(6), 55-71.
- Morales Yela, K. E., & Navarro Castellanos, C. J. (2019). *Desarrollo de aplicación web para matriculación y gestión de notas de la escuela fiscal N° 331 Esperanza Caputi Olvera de la ciudad de Guayaquil*. <http://dspace.ups.edu.ec/handle/123456789/16997>
- Pacheco, S. C., & Subert, M. (2015). *Sistema informático para la gestión de la información de los procesos de las Brigadas Técnicas Juveniles*. <https://repositorio.uci.cu/jspui/handle/123456789/7458>
- Pérez Polo, M. (2017). *Módulo de gestión Comercial para el Sistema de Asesoría de la Agencia de Supervisión e Inspección de Cargas* [BachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1]. <https://repositorio.uci.cu/jspui/handle/123456789/7988>
- Quimis Baque, Y. M., & Figueroa Smith, D. J. (2021). *Desarrollo e implementación de una aplicación web para la matriculación y gestión de ingreso de notas académicas de la Escuela de Educación Básica Fiscal Pedro Carbo*. <http://dspace.ups.edu.ec/handle/123456789/20581>
- Ramírez Guardado, J. J. (2022). *Estudio de factibilidad para el desarrollo de un sistema de control del personal de limpieza e insumos para la Asociación Serviplus de la ciudad de Babahoyo*. <http://dspace.utb.edu.ec/handle/49000/11650>
- Rebah, H. B., Boukthir, H., & Chedebois, A. (2022). *Website Design and Development with HTML5 and CSS3*. John Wiley & Sons.
- Rodríguez Sánchez, T. (2015). *Metodología de desarrollo para la Actividad productiva de la UCI*.

- Tinajero, M., Catota, V., & Catota, E. (2021). LA TÉCNICA DE IADOV. NIVELES DE SATISFACCIÓN DEL CLIENTE EN RM LATACUNGA – MALTERÍA PLAZA AÑO 2019. *Prospectivas UTC «Revista de Ciencias Administrativas y Económicas»*, 4(1), 110-120.
- Verdecia, A., & Augusto, L. (2021, octubre). *Repositorio Digital:Índice de Satisfacción por la Estrategia Didáctica para la utilización de Recursos Educativos Abiertos en la disciplina Historia de Cuba a través de la técnica Iadov*. <https://repositorio.uci.cu/handle/123456789/9789>
- Vivanco, M. T., Gutiérrez, A. C., Nuez, L. Á., & Zaez, J. C. Q. (2022). APLICACIÓN WEB PARA LA GESTIÓN DEL PLAN DE TRABAJO MENSUAL DE LOS DOCENTES UNIVERSITARIOS / WEB APPLICATION FOR THE MANAGEMENT OF THE MONTHLY WORK PLAN OF UNIVERSITY TEACHER. *Universidad & Ciencia*, 11(1), 25-39.
- Vizcaíno, R., & Paolo, E. (2022). *Desarrollo de una aplicación prototipo para la obtención de la zona de cobertura de la red Sigfox en el interior de edificaciones*. <https://doi.org/10/CD>

ANEXOS

Anexo 1: Entrevista realizada al personal encargado de desarrollar la gestión de matrículas en asignaturas optativas con el objetivo de conocer el flujo actual del proceso.

1. ¿Cómo se realiza el proceso de gestión de matrículas en asignaturas optativas?
2. ¿Cuáles son los actores que participan en la gestión de matrículas en asignaturas optativas?
3. ¿Qué tan complejo es el proceso de gestión de matrículas en asignaturas optativas?

Anexo 2: Encuesta realizada a los profesores principales de cada año para validar la propuesta de solución.

1. ¿Consideran necesario un módulo de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1?
 si__ no__ no sé__
2. ¿Considera usted necesario agregarle alguna otra funcionalidad al módulo de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1?
 si__ no__ no sé__
3. ¿Se siente satisfecho con el trabajo realizado y los resultados obtenidos en la propuesta de solución?
 me gusta mucho
 me gusta más de lo que me disgusta
 me da lo mismo
 me disgusta más de lo que me gusta
 no me gusta nada
 no se decir
4. ¿Qué aspectos usted cree que se puedan mejorar en futuras investigaciones relacionadas con el proceso de gestión de matrículas en asignaturas optativas?
5. ¿Considera que algún otro centro pueda hacer uso del módulo de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1?

Anexo 3: Entrevista realizada al personal encargado de desarrollar el proceso de gestión de matrículas en asignaturas optativas para el Sistema de Trabajo Educativo de la Facultad 1.

1. ¿Cuál es la información que necesitan para los reportes?
2. ¿Cuál es la información que necesitan de los estudiantes?
3. ¿Cuál es la información que necesitan recopilar de las asignaturas optativas?
4. ¿Cuál es el personal autorizado en la Facultad 1 para gestionar la información de las matrículas en las asignaturas optativas?