



Universidad de la Ciencias Informáticas

Facultad 1

Sistema para gestionar la información de los colectivos de años de la Facultad 1

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

Autor:

Bryan Ramírez Díaz

Tutores:

Ing. Yaili Ledea Velázquez

MSc. Hubert Viltres Sala

Ing. Noichel Juan Hernández



Es preciso soñar, pero con la condición de creer en nuestros sueños. De examinar con atención la vida real, de confrontar nuestra observación con nuestros sueños, y de realizar escrupulosamente nuestra fantasía.

Vladimir I. Lenin

Declaración de Autoría

Declaro por este medio que yo Bryan Ramírez Díaz, con carnet de identidad 95091728923, soy el autor principal del trabajo titulado "**Sistema para gestionar la información de los colectivos de años de la Facultad 1**" y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los ____ días del mes de _____ del año 2019.

Autor:

Bryan Ramírez Díaz

Tutores:

Ing. Yaili Ledea Velázquez

MSc. Hubert Viltres Sala

Ing. Noichel Juan Hernández

Resumen

En la Facultad 1 de la Universidad de las Ciencias Informáticas surge la necesidad de mejorar la gestión de las evaluaciones integradas, las evaluaciones del componente laboral de los estudiantes repitentes, actividades estudiantiles, incidencias, estrategias educativas de brigadas y años, estudiantes con arrastres y repitentes, y las cartas a los padres de los estudiantes. Actualmente para ejecutar estos procesos, no cuentan con un sistema informático que gestione la información que manejan, esto constituye a que se vuelva muy engorroso elaborar informes y estos no se ajusten en su totalidad a la realidad objetiva, de ahí la necesidad de desarrollar un sistema que permita gestionar la información de los colectivos de años de la Facultad 1 en la Universidad de las Ciencias Informáticas.

Para la búsqueda y recopilación de información se emplearon los métodos de investigación: observación, entrevista, analítico sintético, histórico lógico y modelación. Se llevó a cabo un estudio de diferentes sistemas de gestión de información, enfocándose fundamentalmente en su estructura y funcionamiento para adquirir una mayor visión a la hora de realizar la propuesta de solución. A partir de la realización de este trabajo se obtuvo como resultado un software que resuelve la problemática antes planteada y constituye a la vez un aporte tecnológico a la Facultad 1 de la UCI.

Palabras clave: colectivos de años, gestión de información, información, sistema informático.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica del sistema de gestión de información de los colectivos de año de la Facultad 1	6
1.1 Gestión de la información	6
1.2 Colectivos de años	8
1.2.1 Principales funciones del colectivo de año	8
1.3 Portales web	9
1.4 Soluciones similares existentes	9
1.4.1 Conclusión de las soluciones similares existentes	11
1.5 Metodología de desarrollo de software	11
1.6 Herramientas y tecnologías	12
1.6.1 Herramienta CASE	12
1.6.2 Marco de trabajo	13
1.6.3 Lenguaje de programación del lado servidor	13
1.6.4 Lenguaje de programación del lado cliente	14
1.6.5 Librerías	15
1.6.6 Entorno Integrado de Desarrollo	15
1.6.7 Servidor de Base de Datos	16
1.6.8 Servidor web	18
Conclusiones del capítulo	18
Capítulo 2: Análisis y diseño del sistema de gestión de información de los colectivos de años de la Facultad 1	20
2.1 Descripción de la propuesta de solución	20
2.2 Modelo conceptual	20

2.3	Requisitos de la propuesta de solución.....	22
2.3.1	Requisitos funcionales.....	23
2.3.2	Requisitos no funcionales.....	26
2.4	Casos de uso del sistema.....	26
2.4.1	Descripción de los casos de uso.....	28
2.5	Arquitectura de software.....	38
2.6	Patrones de diseño.....	40
2.6.1	Patrones Generales de Software para la Asignación de Responsabilidades (GRASP).....	40
2.6.2	Patrones Gang of Four (GOF).....	41
2.7	Diagrama de clases de análisis y diseño.....	42
2.8	Modelo de datos.....	43
2.9	Modelo de despliegue.....	46
	Conclusiones del capítulo.....	47
Capítulo 3: Implementación y validación del sistema de gestión de información de los colectivos de años de la Facultad 1.....		48
3.1	Diagrama de componentes.....	48
3.1.1	Descripción del diagrama de componentes.....	50
3.2	Estándares de codificación.....	50
3.3	Validación de la propuesta de solución.....	54
3.3.1	Pruebas funcionales.....	54
3.3.2	Pruebas de seguridad.....	56
	Conclusiones del capítulo.....	59
Conclusiones.....		60
Recomendaciones.....		61

Bibliografía.....62

Índice de figuras

Figura 1. Modelo Conceptual. Elaboración propia.....	22
Figura 2. Diagrama de casos de uso del sistema. Elaboración propia	27
Figura 3. Arquitectura de software MVC. Fuente. (Santiago, 2018)	39
Figura 4. Ejemplo de uso del patrón Alta Cohesión. Elaboración propia.	41
Figura 5. Ejemplo de uso del patrón Decorador. Elaboración propia.....	42
Figura 6. Diagrama de clases del diseño. Gestionar Evaluaciones Integradas. Elaboración propia.	43
Figura 7. Modelo de datos. Elaboración propia.	44
Figura 8. Diagrama de despliegue. Elaboración propia.	46
Figura 9. Diagrama de componentes. Elaboración propia.....	49
Figura 10. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.	51
Figura 11. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.	51
Figura 12. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.	52
Figura 13. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.	52
Figura 14. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.	52
Figura 15. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.....	53
Figura 16. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.....	53
Figura 17. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.....	53
Figura 18. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.....	54
Figura 19. No conformidades por iteración. Elaboración propia.	56
Figura 20. Reporte de la prueba de seguridad. Fuente. Acunetix.....	58

Índice de tablas

Tabla 1. Requisitos funcionales. Elaboración propia	23
Tabla 2. Descripción del CU-1. Gestionar evaluación integrada. Elaboración propia	29
Tabla 3. Descripción de las Evaluaciones integradas. Elaboración propia	50
Tabla 4. Caso de prueba CU-1_P1. Elaboración propia.....	54
Tabla 5. Caso de prueba CU-1_P2. Elaboración propia.....	55
Tabla 6. Caso de prueba CU-1_P3. Elaboración propia.....	55

Introducción

En la década de 1970, la gestión de la información en gran medida se refería a lo que ahora se denomina gestión de datos: tarjetas perforadas, cintas magnéticas y otros medios de mantenimiento de registros, que implican un ciclo de vida de formatos que requieren origen, distribución, respaldo, mantenimiento y eliminación. El progreso tecnológico va a un ritmo rápido y tiene un alcance universal que produce soluciones adaptativas por parte de las organizaciones. Las nuevas tecnologías cambian los procesos, hacen que operaciones sean más sencillas y generan una transformación en las competencias.

Ante el problema ocasionado por la producción excesiva de documentos a partir del segundo tercio del siglo XIX, DOLLAR (1993), manifiesta el reclamo de las administraciones por la intervención de los archiveros con criterios técnicos, lo que les obligó a introducirse en el ámbito de la gestión de los documentos. Se percataron de lo importante de su intervención desde el mismo instante de la creación de los documentos, pues es en este preciso momento cuando se puede realizar la selección de la documentación y el diseño de sistemas de clasificación que favorezcan la separación de los documentos con valor temporal, susceptibles por consiguiente de ser eliminados, de aquellos otros que tienen un valor permanente y que se deben transferir a los archivos (Aranda, 2013). Resulta esencial para las instituciones y la sociedad en general, la organización de los documentos de archivos y precisamente se logra a través de la gestión documental siendo el conjunto de tareas y procedimientos orientados a lograr una mayor eficacia y economía de los documentos por parte de las organizaciones (Aranda, 2013).

La gestión de la información ocupa, cada vez más, un espacio mayor en la economía de los países a escala mundial, existiendo de esta manera una estrecha relación entre la gestión de la información y el conocimiento y la calidad del quehacer en una organización. Un recurso vital en cualquier organización en la actualidad es la incorporación de las nuevas tecnologías y las herramientas que nos aportan la inteligencia empresarial en la prestación de productos informativos, esto ha constituido un gran reto y a la vez una oportunidad para las ciencias de la información. Se puede afirmar que la web ha evolucionado a grandes pasos, ha dejado de ser un medio para la publicación de información y contenidos, para convertirse en una plataforma de diseño y desarrollo de aplicaciones informáticas distribuidas. El

navegador se ha transformado en un “cliente universal” que facilita el desarrollo de aplicaciones y la adopción de nuevas soluciones tecnológicas, al reducir costos de mantenimiento.

En Cuba el empleo de técnicas modernas de dirección empresarial, adecuadas a nuestras características y basada en las mejores y más avanzadas prácticas contemporáneas, así como el amplio uso de todas las posibilidades de las tecnologías y servicios de información y las telecomunicaciones, deben constituir prioridad del país, a fines de garantizar la mayor eficiencia en la gestión y los procesos productivos. Para ello, es necesario crear los cimientos de una cultura gerencial que reconozca el valor que tienen para una organización, el uso adecuado de la información y la insustituible importancia de utilizar y preservar los conocimientos adquiridos por los trabajadores en el transcurso del tiempo, que son a su vez parte estratégica de su patrimonio.

En la resolución económica del V Congreso del Partido Comunista de Cuba celebrado en 1997 en La Habana se plantea:

“La eficiencia, por tanto, el objetivo central de la política económica pues constituye una de las mayores potencialidades con que cuenta el país. Hacer un mejor uso de los recursos, elevar la productividad del trabajo, alcanzar mejores resultados con menos costos tendrán un efecto positivo en nuestro balance financiero (...)” (Prensa, 2019). Por otra parte, entre los elementos que deben sustentar tal declaración se asegura que: El empleo de técnicas modernas de dirección empresarial (...) así como el amplio uso de todas las posibilidades de las tecnologías y servicios de información y las comunicaciones, deben constituir prioridad del país a fines de garantizar la mayor eficiencia en la gestión y los procesos productivos (Prensa, 2019).

La Universidad de las Ciencias Informáticas, también conocida como la UCI, tiene entre sus más ambiciosos propósitos llevar la Informática a cada rincón del país, y a su vez todo el trabajo que se desarrolla dentro de la misma. Todo este proceso de informatización se lleva a cabo a través de sistemas de gestión entre los cuales se destaca el desarrollo y uso de portales web bajo licencia libre.

En la UCI se gestiona la información de los estudiantes por años académicos y brigadas. La gestión de la información de cada estudiante permite obtener elementos importantes para el trabajo de los profesores guías, colectivos de brigada, colectivos de año, profesor principal y decana. Para poder realizar los

análisis sobre cada estudiante es necesario recopilar la información procedente de varias áreas. En la actualidad existe información almacenada en documentos que no siguen un estándar, en ocasiones la información más relevante está incompleta, no permite realizar una valoración sobre la misma, puede estar duplicada, existe una mayor probabilidad de sufrir pérdidas, el trabajo se hace engorroso y se pierde tiempo en la búsqueda, se necesita un local para almacenar los registros de los documentos. No existe un mecanismo digital que permita generar reportes, documentos oficiales del año y evaluaciones integradas de forma informatizada lo que dificulta el trabajo de los profesores.

Debido a esta problemática, se plantea como **problema de investigación**: ¿Cómo contribuir a la gestión de la información relacionada a los colectivos de años de la Facultad 1?

El **objeto de estudio** está definido por el proceso de gestión de la información asociada a la actividad docente en la Educación Superior.

Objetivo general:

1. Desarrollar un sistema informático que contribuya a la gestión de la información de los colectivos de año de la Facultad 1.

El campo de acción está enmarcado en el proceso de gestión de la información de los colectivos de años en la Facultad 1 de la UCI.

Objetivos específicos:

1. Construir los referentes teóricos fundamentales que sustentan la investigación relacionados con el desarrollo del sistema de gestión de información en la actividad docente educativa.
2. Diagnosticar el estado de la gestión de la información en los años académicos de la Facultad 1.
3. Diseñar las funcionalidades del sistema para gestionar la información de los colectivos de años de la Facultad 1.
4. Implementar las funcionalidades del sistema para gestionar la información de los colectivos de años de la Facultad 1.
5. Validar las funcionalidades del sistema para gestionar la información de los colectivos de años de la Facultad 1.

Idea a defender: El desarrollo de un sistema informático contribuirá a la gestión de la información de los colectivos de años de la Facultad 1 en la Universidad de las Ciencias Informáticas.

Se han empleado como **métodos de la investigación:**

Métodos empíricos:

Observación: Se han visitado diferentes sistemas para observar su funcionamiento y la manera que organizan su información.

Entrevista: Permite consultar expertos para identificar las principales funcionalidades a tener en cuenta en el desarrollo de la propuesta de solución, así como los contenidos a mostrar. Se entrevistaron a los clientes con el fin de obtener la información necesaria para la creación de la aplicación.

Métodos teóricos:

Análítico-sintético: Se utilizó el análisis de la información para arribar a conclusiones prácticas y teóricas de la investigación. Mediante este método se realiza un análisis de los documentos, las publicaciones, bibliografías y en general toda la información relacionada con sistemas de gestión de la información académica de los colectivos de años, para reflejar los elementos más importantes y necesarios en el desarrollo de la solución.

Histórico-lógico: Se ha desarrollado un profundo análisis alrededor de las tecnologías actuales en vista de la selección de cuál sería la adecuada para desarrollar el sistema.

Modelación: Se modelan diagramas para la implementación del sistema.

La **estructura de investigación** del presente trabajo está concebida de la siguiente manera:

Capítulo 1: Fundamentación teórica del sistema de gestión de información de los colectivos de años de la Facultad 1.

En este capítulo quedan plasmados los elementos teóricos que dan soporte a la investigación, así como los principales métodos y herramientas que son empleados para el desarrollo del sistema de gestión de información de los colectivos de años de la Facultad 1.

Capítulo 2: Análisis y diseño del sistema de gestión de información de los colectivos de años de la Facultad 1.

En este capítulo se definen las características del sistema, se determinan los requisitos funcionales y no funcionales que deben estar presentes y se describen las características de la solución propuesta mediante artefactos definidos en el proceso de desarrollo de software.

Capítulo 3: Implementación y validación del sistema de gestión de información de los colectivos de años de la Facultad 1.

En este capítulo se tratan temas referentes a la implementación de la solución y se realizan las pruebas a la aplicación con el propósito de evaluar su correcto funcionamiento, el nivel de seguridad y el porcentaje de usabilidad.

Capítulo 1: Fundamentación teórica del sistema de gestión de información de los colectivos de año de la Facultad 1

En el presente capítulo se tratan un conjunto de aspectos teóricos que permiten fundamentar y dar solidez a la investigación. Se caracterizan los elementos fundamentales para gestionar la información de los colectivos de años mediante un estudio de sistemas similares y los conceptos sobre la información de los colectivos de año analizados por el autor, sus reflexiones y criterios asumidos. Se analizan las herramientas, tecnologías, lenguajes y metodologías de desarrollo de software a utilizar en la investigación.

1.1 Gestión de la información

Proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización. Genera nuevos conocimientos (Díaz Rosabal, y otros, 2017). Gestionar información es ir en busca de nuevos significados, análisis, aplicar el principio de que el todo, es más que la suma de las partes. Es producir un impacto en el ambiente de cualquier organización. Es un proceso que debe estar presente en cada uno de los pasos de la organización, es un proceso y a la vez un subproceso que requiere acción, decisión y evaluación. Mediante la gestión se proporcionan los recursos de información necesarios para una buena toma de decisiones, se desarrollan nuevos conocimientos que posibilitan calidad y eficiencia en los servicios y productos de las organizaciones (wordpress, 2016).

La información es un intercambio de conocimientos entre dos o más personas en una comunidad o sociedad. En otras palabras, se relaciona con el contexto en el que se recogen los datos, su codificación en forma inteligible y, finalmente, el significado atribuido a dichos datos. En informática la información es un conjunto organizado y con sentido de datos. Los ordenadores se han convertido en poderosas herramientas para almacenar, procesar y analizar información. La difusión de Internet como una red global, por otro lado, pone a disposición una gran cantidad de información nunca antes disponible para la humanidad (Alegsa, 2016).

Desde el punto de vista de la tecnología de la información, la información es una relación entre dos datos. Desde este punto de vista permite distinguirse información de datos. Un dato es simplemente un número, una fecha, una palabra, etc. En cambio, el significado que se puede dar a esta figura, relacionándola con uno o más datos o representaciones de conceptos, es información. La información es objeto de estudio y aplicación en diversos campos del conocimiento y la acción humana: en el aspecto técnico es el objeto de estudio de ingeniería de la información; el establecimiento de información de las ciencias sociales es el tema de investigación de ciencias de la comunicación y la sociología en general, con especial atención a los aspectos relacionados con la difusión de los medios de comunicación en la actual sociedad de la información y la era de la información (Alegsa, 2016).

Según (Gonzalez, 2017) considera como funciones de la gestión de información:

1. Determinar las necesidades internas de información (relativas a las funciones, actividades y procesos administrativos de la organización) y satisfacerlas competitivamente.
2. Determinar las necesidades de inteligencia externa de la organización y satisfacerlas competitivamente.
3. Desarrollar la base informacional de la organización y garantizar su accesibilidad.
4. Optimizar el flujo organizacional de la información y el nivel de las comunicaciones.
5. Desarrollar la estructura informacional de la organización y garantizar su operatividad.
6. Manejar eficientemente los recursos institucionales de información, mejorar las inversiones sucesivas en ellos, y optimizar su valor y su aprovechamiento organizacional.
7. Garantizar la integridad y accesibilidad a la memoria corporativa.
8. Evaluar periódicamente la calidad e impacto del soporte informacional para la gestión y el desarrollo de la organización.
9. Optimizar el aprovechamiento de la base y la estructura informacionales de la organización para incrementar su productividad o el rendimiento de la inversión.
10. Establecer, aplicar y supervisar los procedimientos relativos a la seguridad de la información organizacional.
11. Entrenar a los miembros de la organización en el manejo o la utilización, de los Recursos informacionales de la organización.
12. Contribuir a modernizar u optimizar las actividades organizacionales y los procesos administrativos, relacionados con ellas.

13. Garantizar la calidad de los productos informacionales de la organización, y asegurar su dimensión efectiva.

Un Sistema de Gestión de Información permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El Sistema de Gestión de Información aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa (Arévalo, 2007).

1.2 Colectivos de años

El colectivo de año constituye un nivel de dirección atípico en la estructura de las universidades, conducido por el profesor principal de año académico. Este colectivo está integrado por los profesores que desarrollan las asignaturas del año, los profesores guías de cada grupo, los tutores y los representantes de las organizaciones estudiantiles. Tiene como propósito lograr el cumplimiento con calidad de los objetivos instructivos y educativos del año, así como aquellos que complementariamente se hayan concertado para responder a las características propias del grupo y del momento, mediante la implementación de la estrategia educativa del año académico (MES, 2014).

1.2.1 Principales funciones del colectivo de año

Según (MES, 2014) las principales funciones del colectivo de año son:

1. Elaborar, ejecutar y controlar la estrategia educativa del año académico, a partir de la concebida para la carrera, propiciando la integración de las actividades curriculares y extracurriculares que desarrollan los estudiantes; en correspondencia con los objetivos educativos e instructivos de ese año.
2. Elaborar la estrategia específica de comunicación sistemática con la activa participación de los estudiantes.
3. Desarrollar el proceso del diagnóstico integral y de evaluación de los estudiantes.

4. Propiciar la integración de las actividades académicas, laborales e investigativas con las diferentes tareas de impacto social, deportivas, culturales, entre otras que realizan los estudiantes, en correspondencia con los objetivos del año.
5. Dirigir el trabajo metodológico en el año, velando por su rigor y efectividad.
6. Materializar en el año las estrategias curriculares y de orientación profesional propuesta por el colectivo de la carrera, así como el adecuado desarrollo de la asignatura perteneciente a la disciplina principal integradora.
7. Conducir y controlar sistemáticamente la marcha del proceso docente educativo en el año, desarrollando acciones para eliminar las deficiencias detectadas y tomando las medidas que permitan el mejoramiento continuo de la calidad de dicho proceso.
8. Promover la participación generalizada y activa de los estudiantes en tareas económicas y sociales de importancia para la sociedad y la universidad.
9. Promover el diálogo con autoridades políticas y líderes científicos del país y el territorio, acerca de programas de desarrollo y sus resultados.

1.3 Portales web

Un portal web es un sitio web que tiene como propósito centralizar un conjunto de servicios y/o recursos de manera integrada para el usuario, a menudo posibilitando que el mismo se informe, participe, opine o acceda a múltiples aplicaciones desde un mismo lugar (Bembibre, 2009). Un portal web puede tener el objetivo de resolver un tipo de necesidad específica de un grupo de personas o comunidad con intereses en particular.

1.4 Soluciones similares existentes

La evolución en el mundo del software ha llevado a una gran competencia en cuanto a los sistemas informatizados. Los sistemas para la gestión de información académica no están libres de esta competitividad y se han desarrollado de maneras muy diversas, lo que ha permitido la creación de productos de muy buena calidad y un alto grado de compromiso. A continuación se analizan algunas soluciones similares existentes tanto en el ámbito nacional como en el ámbito internacional.

Sistema Integrado de Gestión Académica (SIGA)

Aplicación informática usada por la Universidad de Córdoba para la gestión académica implementada en el lenguaje de programación PHP 5. Abarca desde la gestión del acceso a la universidad hasta la tramitación de los títulos, la matrícula, actas de examen, expedientes, estadísticas, informes, gestión de becas y convalidaciones. Está compuesto por trece módulos (Alumnos, Opciones de alumnos, Diplomas, Profesores, Horario, Inventario, Utilidades, Generador de listados, Económico, Automatización de exámenes, Automatización de encuestas, Automatización de matriculación de alumnos). El sistema SIGA (Sistema Integrado de Gestión Académica) desarrollado en España, está compuesto por diferentes módulos que ayudan a gestionar la información para el proceso de autoevaluación (SIGA, 2018), pero se encuentra desarrollado bajo software propietario y tendría que realizarse un gasto de dinero para su obtención.

Sistema Virtual de Evaluación y Consulta sobre Calidad Académica (SVECCA)

Es un software peruano que fue desarrollado por la Oficina Central de Calidad Académica y Acreditación (OCCAA) de la Universidad Nacional Mayor de San Marcos (UNMSM). El software está disponible para utilizarlo en otras tareas que tengan que ver con la consulta a la comunidad tales como recolectar, vía Internet, la información procedente de la comunidad sanmarquina (estudiantes, docentes, administrativos y egresados), mediante encuestas y cuestionarios diseñados específicamente para esta plataforma virtual. El Sistema Virtual de Evaluación y Consulta sobre Calidad Académica Universitaria (SVECCA) del Perú (UNMSM, 2009), está basado en encuestas desarrolladas a estudiantes y profesores y se debe tener acceso a internet por lo que no tiene la flexibilidad de ser adaptado a todas las áreas necesarias de la Facultad 1 con sus diversas particularidades y distintas maneras de realizar determinados procedimientos.

Sistema de Gestión de la Nueva Universidad (SIGENU)

Es un software desarrollado por la CUJAE (Ciudad Universitaria José Antonio Echeverría), desde un principio fue pensado con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la universidad cubana, finalmente se decidió gestionar la información solo de esa universidad, este sistema ha sido concebido de manera tal que sea capaz de brindar gran seguridad de la información (uclv, 2017), pero no tiene la flexibilidad de ser

adaptado a todas las áreas necesarias de la Facultad 1 con sus diversas particularidades y distintas maneras de realizar determinados procedimientos.

1.4.1 Conclusión de las soluciones similares existentes

A partir del estudio realizado a los sistemas anteriores que de una forma u otra se relacionan con actividades académicas, se determina no utilizar ninguno de ellos por los problemas antes mencionados, sin embargo, el estudio de estos sistemas proporcionó una serie de características que pueden ser agregadas a la investigación, entre las que se encuentran:

- Permitir generar informes de las actividades docentes de los estudiantes, los cuales van a lograr con gran efectividad el control de esta información.
- Permitir registrar las actividades deportivas, culturales e investigativas.

A raíz de lo estipulado anteriormente se dispone a implementar un sistema web de gestión de información que cubra las necesidades detectadas en la Facultad 1, y capaz de satisfacer los requerimientos del negocio exigidos por el cliente.

1.5 Metodología de desarrollo de software

La metodología del Proceso Unificado Ágil (*Agile Unified Process*, AUP), constituye una versión simplificada del Proceso Unificado Racional (*Rational Unified Process*, RUP), desarrollada por Scott Ambler. Esta metodología combina procesos propios del concepto unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad, permitiendo así de manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio. AUP aplica técnicas ágiles, entre las que se incluyen (Rodríguez, 2018):

- El desarrollo dirigido por pruebas.
- El modelado ágil.
- La gestión de cambios ágil.
- La refactorización de bases de datos para mejorar la productividad.

Según (Rodríguez, 2018), al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se decide utilizar una variación de la metodología AUP para la UCI en el escenario 2, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce. Las disciplinas propuestas por AUP son: gestión de configuración, gestión de proyectos y entorno, estas se cubrirán de acuerdo a las áreas de procesos que define CMMI-DEV (*Capability Maturity Model Integration*) en su versión 1.3 quedando de la siguiente forma: gestión de la configuración, la planeación del proyecto, el monitoreo y control de proyecto.

1.6 Herramientas y tecnologías

La propuesta de solución es un sistema web para los colectivos de años de la Facultad 1, al analizar la documentación existente relacionada con este, se identifican las bases tecnológicas y las herramientas para su desarrollo.

1.6.1 Herramienta CASE

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas o programas informáticos destinados a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero (Bry, 2019).

Visual Paradigm es una herramienta UML (Lenguaje Unificado de Modelado) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar, actualizar y compatible entre ediciones (Alonso, 2009). Para la presente investigación, se utiliza esta herramienta en su versión 8.0.

1.6.2 Marco de trabajo

Un marco de trabajo es un conjunto de archivos y pautas que definen la estructura y metodología, sobre cómo hacer el desarrollo de un proyecto software. Se podría decir que es una guía o esquema que nos ayuda a programar de forma sencilla y rápida. Su objetivo es el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya desarrolladas. Principalmente, nos permite centrarnos en el problema, en vez de preocuparnos por implementar funcionalidades que son de uso común en muchas aplicaciones. Generalmente los marcos de trabajos están basados en un lenguaje de programación, aunque no siempre es así.

Para el desarrollo de la aplicación web se utiliza el framework *Symfony* en su versión 3.4 el cual es un marco de trabajo basado en el patrón Modelo-Vista-Controlador (MVC) para el desarrollo de aplicaciones web. Implementado en PHP (del inglés *Hipertext Pre-processor*), con él se pueden crear aplicaciones y sitios web rápidos y seguros de una forma profesional (Symfony, 2019).

Para el diseño de la interfaz web se utiliza el framework CSS (del inglés *Cascading Style Sheets*) *Bootstrap* en su versión 3.3.5, este permite de forma rápida la construcción de interfaces web. Este se adapta a todo tipo de pantallas, es extensible y cuenta con múltiples implementaciones externas (Capote Vázquez, y otros, 2015).

1.6.3 Lenguaje de programación del lado servidor

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo (Ayala, 2017). Para el desarrollo de la aplicación web se emplea el lenguaje PHP (*Hypertext Pre-processor*) en su versión 7.2.9, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (php, 2019). Según (Maria, 2015) las principales ventajas de PHP son:

- Su rapidez, al estar incrustado en el código de HTML, el tiempo de respuesta es más corto.
- Contiene muchas características y funciones especiales necesarias para crear páginas web dinámicas facilitando su uso.

- Se puede ejecutar en la mayoría de los sistemas operativos que son usados en la actualidad.
- Está especialmente diseñado para soportar bases de datos. PHP incluye funcionalidades diseñadas especialmente para interactuar con bases de datos específicas. Esto te permite trabajar sin necesidad de conocer los detalles técnicos que se requieren para comunicarse con una base de datos.
- Es personalizable. Su licencia de código abierto permite a los programadores modificar el software PHP, agregando o modificando sus características según se requiera para ajustarse a los ambientes específicos de cada uno.

1.6.4 Lenguaje de programación del lado cliente

JavaScript se define como un lenguaje de programación interpretado, no requiere la compilación del código de fuente para ejecutarlo. Este trabaja sobre un enfoque útil y práctico hace que el proceso de desarrollo de aplicaciones web dinámicas sea eficiente, está disponible en cualquier navegador web (Caballero, 2017).

Se utiliza el Lenguaje de Marcado de Hipertexto (HTML por sus siglas en inglés) en su versión 5.0. HTML es concebido como un lenguaje para el intercambio de documentos científicos y técnicos adaptado para su uso por no especialistas en tratamiento de documentos. Se sirve de un reducido conjunto de etiquetas estructurales y semánticas apropiadas para la realización de documentos relativamente simples. Pero, además de simplificar la estructura de los documentos soporta el hipertexto (Lapuente, 2018). Es un lenguaje básico predefinido en el marco de trabajo seleccionado.

Para definir la apariencia o estilo de las páginas web escritas en HTML del módulo propuesto se utiliza CSS versión 3.0. Este permite separar el contenido de la forma y proporciona a los diseñadores el mantenimiento de un control más preciso sobre la apariencia de las páginas web. Su novedad más importante consiste en la incorporación de mecanismos para mantener un mayor control sobre el estilo de los elementos que se muestran en la página (Editor, 2016).

1.6.5 Librerías

Una librería es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca. Las librerías utilizadas para el desarrollo de la propuesta de solución son:

JQuery

JQuery es una biblioteca de JavaScript (JS) que simplifica la forma de desarrollar aplicaciones web. Las aplicaciones que utilizan jQuery suelen necesitar menos tiempo y menos código que las aplicaciones hechas con JS puro, permite manipular elementos del DOM (*Document Object Model*) (textos, imágenes, enlaces), cambiar el diseño CSS o realizar peticiones Ajax utilizando instrucciones simples (Domínguez, 2018). Para la implementación de la aplicación web se emplea la librería jQuery en su versión 1.11.2.

Knpsnappy

Knpsnappy es un *wrapper* (empaquetador) para el convertidor *wkhtmltopdf* que permite generar documentos PDF o imágenes haciendo uso del motor *webkit*. Utiliza funciones nativas para las conversiones de html al formato deseado. Tiene buen soporte oficial y en la web y ofrece una variedad de funciones para generar documentos PDF o imágenes de diversas maneras (MENJÍ, 2016). Para la implementación de la aplicación web se emplea *Knpsnappy* en su versión 0.12.5.

1.6.6 Entorno Integrado de Desarrollo

JetBrains PhpStorm es un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) de código abierto y multiplataforma creado por la compañía JetBrains, es un poderoso y completo editor del lenguaje de programación PHP centrado en la productividad del desarrollador. Contiene un editor de código inteligente que analiza sintaxis, configuración de formato del código extendido, comprobación de errores en tiempo real sobre el proceso de implementación, plegado del código en bloques y autocompletado. Brinda soporte a los lenguajes CSS, JavaScript y HTML. Dispone de los sistemas de control de versiones más populares como Git, Subversion, Mercurial y Perforce. Ofrece un formato de código acorde a los estándares de codificación de Symfony con la integración de un módulo para este framework (PhpStorm,

2018). Para el desarrollo de la propuesta de solución se utiliza JetBrains PhpStorm en su versión 2018.2.3.

1.6.7 Servidor de Base de Datos

Un sistema gestor de base de datos (SGBD, en inglés DBMS: *DataBase Management System*) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto. Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas (Bueno, 2017)

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo la licencia BSD (distribución de software Berkeley) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones posee las potencialidades y prestaciones semejantes a otras bases de datos comerciales. PostgreSQL utiliza el modelo cliente-servidor y usa multiprocesos en vez de multihilo para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará al resto y el sistema continuará funcionando (Ferrer Martínez, y otros, 2015).

Características fundamentales de PostgreSQL:

- Es una base de datos 100% ACID (*Atomicity, Consistency, Isolation, Durability*).
- Juegos de caracteres internacionales.
- Múltiples métodos de autenticación.
- Actualización in-situ integrada (pg_upgrade).
- Completa documentación.

- Licencia BSB.
- Multiplataforma.

MySQL

Es un software libre, licenciado bajo la licencia Dual. Es un sistema de administración de bases de datos para bases de datos relacionales. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue escrito en C y C++ y se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más empleados como PHP, Perl y Java, y su adaptación a distintos sistemas operativos (Avitia, 2019).

Es un sistema gestor de bases de datos muy rápido, fiable y fácil de usar. MySQL presenta como desventaja, que una vez eliminada una tabla, no se eliminan automáticamente los privilegios concebidos sobre la misma. Debe usarse explícitamente un comando REVOKE para eliminar los privilegios de una tabla, acción que resulta tediosa para los usuarios con permiso de administración para la base de datos (Avitia, 2019).

Características fundamentales de MySQL:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- La API (*Application Programming Interface*) se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.

- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra “ñ”.
- Es altamente confiable en cuanto a estabilidad se refiere.

Para el desarrollo de la propuesta de solución se selecciona MySQL en su versión 5.7 por su velocidad de procesamiento en operaciones y garantizar un mayor rendimiento cuando se consultan las bases de datos, su bajo consumo ante servidores con bajas prestaciones y su facilidad de uso en la administración.

1.6.8 Servidor web

Un servidor web es el encargado de manejar páginas web y enviarlas a través de la red a quienes lo requieran y tengan los permisos para dichas páginas. Son los principales encargados de generar el tráfico en Internet pues es a través de que se realizan las conexiones a todos los sitios web del mundo, toda página web está almacenada en uno de estos servidores (Carlos, 2014). Como servidor web se emplea Apache en su versión 2.4.34, es un software de código abierto desarrollado por la empresa *Apache Software Foundation* en 1996. Es un servidor web flexible, rápido y eficiente, altamente configurable por su diseño modular, esta característica permite ampliar considerablemente sus capacidades al existir un repositorio extenso completamente gratuito de extensiones y módulos. Es compatible con el lenguaje de programación PHP, comparten muchas de sus características (Foundation, 2017).

Conclusiones del capítulo

En el capítulo se realizó un análisis de los principales conceptos asociados a la investigación teniendo en cuenta la visión de diferentes autores, además de un estudio de los sistemas similares existentes en el mundo y en el país. Finalmente, se llegó a la conclusión de que los mismos no reúnen las condiciones necesarias para mejorar el proceso de gestión de la información de los colectivos de años de la Facultad 1 en la UCI, por lo que surge la necesidad de desarrollar un nuevo sistema que solucionará la problemática identificada.

Para el desarrollo del sistema propuesto se analizaron varias de las metodologías, tecnologías, herramientas y lenguajes existentes y se seleccionó: una variación de AUP para la UCI como metodología para guiar el proceso de desarrollo de software, UML como lenguaje de modelado y Visual Paradigm como herramienta CASE para realizar el modelado del software.

Tras determinarse que la propuesta de solución será una aplicación web, se emplearon tecnologías del lado del cliente y del servidor. En el lado del servidor el lenguaje seleccionado fue PHP con el framework Symfony, mientras que por el lado del cliente se seleccionó HTML para el maquetado, CSS para los estilos visuales a través de Bootstrap y como lenguaje de scripting JavaScript a través de la librería jQuery.

Para el almacenamiento de la información se determinó a MySQL como gestor de bases de datos. El servidor web a utilizar será Apache y por sus disímiles características que lo hacen un potente IDE, fue elegido PHPStorm para desarrollar la propuesta de solución.

Capítulo 2: Análisis y diseño del sistema de gestión de información de los colectivos de años de la Facultad 1

En este capítulo se describen las principales características de la solución propuesta. Se realiza un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que el sistema funcione correctamente. Además, se construyen los artefactos correspondientes al análisis y diseño acorde con la metodología seleccionada en el capítulo anterior.

2.1 Descripción de la propuesta de solución

El Sistema para gestionar la información de los colectivos de años de la Facultad 1(SGICA) permite recopilar información de los colectivos de años de la Facultad 1 y generar reportes, documentos oficiales del año y evaluaciones integradas de forma informatizada. La aplicación propuesta permite administrar las evaluaciones integradas de los estudiantes, las evaluaciones del componente laboral de los estudiantes repitentes (plan de trabajo, evaluación del componente laboral); registrar las actividades de los estudiantes (juegos deportivos, festival de artistas aficionados, guardia estudiantil, eventos colaterales), las incidencias de los estudiantes de la facultad, la estrategia educativa por brigadas y años y los estudiantes promovidos, con arrastre o repitentes. La aplicación permite, además, generar las cartas que se envían a los padres de los estudiantes en determinados momentos del curso, con el objetivo de informarlos sobre la situación académica de los estudiantes.

2.2 Modelo conceptual

A partir del análisis efectuado se realizó un modelo conceptual (Figura 1), el cual describe los procesos importantes y las relaciones entre ellos. En el proceso se describe al Profesor guía, Decana, Tutor de repitentes y Profesor principal de año como principales actores del proceso. Estos son los encargados de gestionar la información de las evaluaciones integradas, evaluaciones del componente laboral y el plan de trabajo, actividades, los estudiantes que están promovidos, arrastrando o repitiendo, las estrategias

educativas por brigadas y años, las incidencias y la carta a los padres de los estudiantes. A continuación, se describen los conceptos que componen el modelo conceptual:

- **Profesor guía:** es un educador por excelencia, que atiende preferentemente las necesidades educativas del grupo de estudiantes que se le ha asignado.
- **Decana:** es la responsable de dirigir la Facultad y representarla legalmente.
- **Tutores de repitentes:** es un profesor de "Tiempo Completo", que atiende a los estudiantes repitentes mediante la observación de su desempeño académico y socio afectivo, dándoles seguimiento, donde los orienta y ayuda para su desarrollo integral.
- **Profesor principal de año:** es un docente de experiencia científica y pedagógica. Es el máximo responsable de la educación de los estudiantes mediante la instrucción, jerarquizando el trabajo político-ideológico en el año, así como el correcto desarrollo del proceso docente educativo.
- **Estudiante:** representa a los estudiantes de la Facultad 1 en la UCI.
- **Evaluación integrada:** información que representa la evaluación integrada de cada estudiante.
- **Evaluación componente laboral:** información que representa el plan de trabajo y la evaluación final de los estudiantes repitentes.
- **Plan de trabajo:** información que representa las actividades planificadas para los estudiantes repitentes.
- **Actividades:** información que representa las actividades que realizan los estudiantes.
- **Juegos Mellas:** información que representa las participaciones en los juegos mellas de los estudiantes.
- **Festivales:** información que representa las participaciones en los festivales de los estudiantes.
- **Guardia:** información que representa las ausencias a las guardias de los estudiantes.
- **Eventos colaterales:** información que representa las participaciones en los eventos colaterales de los estudiantes.
- **Estrategia Educativa Año:** información que representa la estrategia educativa por años.
- **Estrategia Educativa Brigadas:** información que representa la estrategia educativa por brigadas.
- **Incidencia:** información que representa las juntas disciplinarias, comisiones disciplinarias y actas de notificaciones a los estudiantes que incurrir en una incidencia.
- **Promovido:** información que representa a los estudiantes que probaron un año académico.

- **Repitiendo:** información que representa los estudiantes que actualmente están repitiendo un año.
- **Arrastrando:** información que representa los estudiantes que actualmente están arrastrando una o más asignaturas.
- **Carta:** información que representa una carta a los padres de un estudiante.
- **Año:** información que representa los años académicos en la UCI.
- **Grupo:** información que representa las brigadas de la Facultad 1 en la UCI.

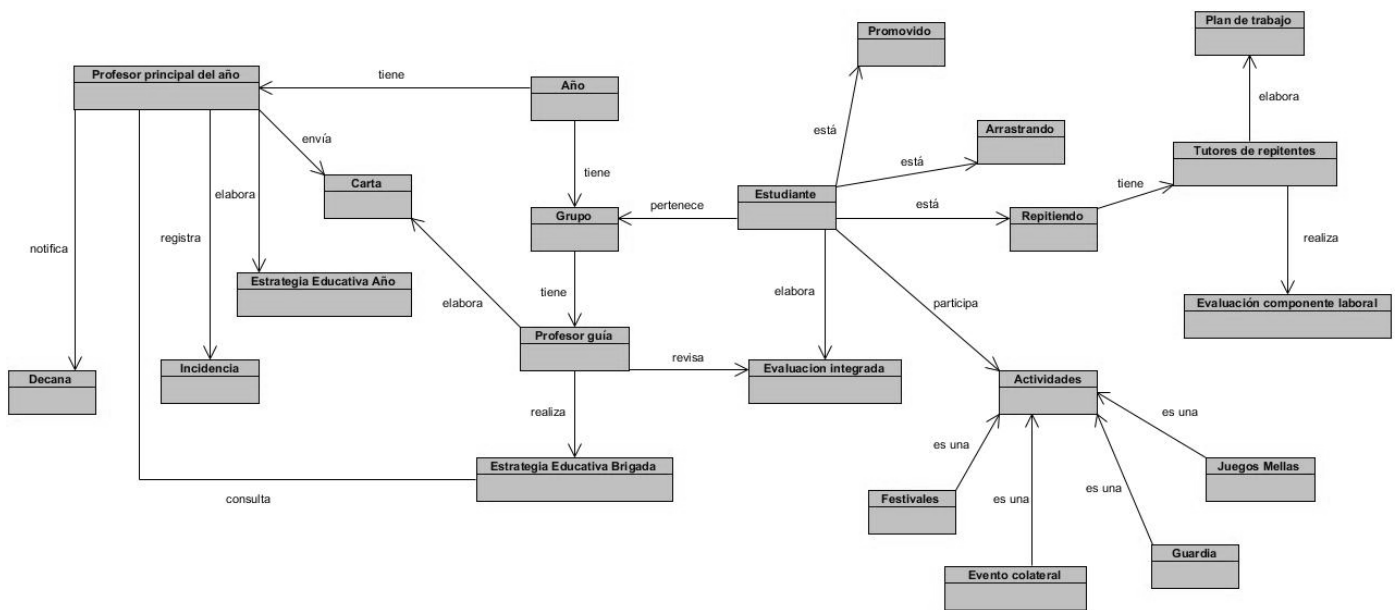


Figura 1. Modelo Conceptual. Elaboración propia

2.3 Requisitos de la propuesta de solución

Los requisitos se utilizan como datos de entrada en la etapa de diseño del producto que se desea lograr. Estos representan que debe hacer el sistema, pero no cómo hacerlo y, además, incorporan las cualidades o propiedades que el mismo debe poseer (Izquierdo, 2016). Los requisitos se pueden clasificar en funcionales y no funcionales.

2.3.1 Requisitos funcionales

Los requisitos funcionales (RF) son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar (PMOinformatica, 2018). Los mismos se especifican en la Tabla 1.

Tabla 1. Requisitos funcionales. Elaboración propia

No.	Requisitos	Descripción	Prioridad
RF 1	Agregar evaluación integrada	El sistema debe permitir agregar una evaluación integrada a cada estudiante.	Alta
RF 2	Listar evaluación integrada	El sistema debe permitir listar las evaluaciones integradas de los estudiantes.	Baja
RF 3	Modificar evaluación integrada	El sistema debe permitir modificar las evaluaciones integradas de los estudiantes.	Media
RF 4	Eliminar evaluación integrada	El sistema debe permitir eliminar las evaluaciones integradas de los estudiantes.	Alta
RF 5	Mostrar plan de trabajo	El sistema debe permitir mostrar el plan de trabajo del estudiante repitente.	Baja
RF 6	Modificar plan de trabajo	El sistema debe permitir modificar el plan de trabajo del estudiante repitente.	Baja
RF 7	Eliminar plan de trabajo	El sistema debe permitir eliminar el plan de trabajo del estudiante repitente.	Alta
RF 8	Agregar plan de trabajo	El sistema debe permitir insertar el plan de trabajo del estudiante repitente.	Media
RF 9	Mostrar evaluación final del componente laboral	El sistema debe permitir mostrar la evaluación del componente laboral del estudiante repitente.	Media
RF 10	Modificar evaluación final del componente laboral	El sistema debe permitir modificar la evaluación del componente laboral del estudiante repitente.	Media
RF 11	Eliminar evaluación final del componente laboral	El sistema debe permitir eliminar la evaluación del componente laboral del estudiante repitente.	Baja
RF 12	Agregar evaluación final del componente laboral	El sistema debe permitir agregar la evaluación del componente laboral del estudiante repitente.	Alta
RF 13	Mostrar guardia	El sistema debe permitir mostrar los estudiantes con ausencias a las guardias y su motivo.	Alta
RF 14	Modificar guardia	El sistema debe permitir modificar un estudiante con ausencia en una guardia.	Media
RF 15	Eliminar guardia	El sistema debe permitir eliminar un estudiante con ausencia en una guardia.	Media

RF 16	Agregar guardia	El sistema debe permitir agregar un estudiante con ausencia en una guardia.	Baja
RF 17	Mostrar actividades estudiantiles	El sistema debe permitir mostrar los estudiantes con participaciones en las actividades estudiantiles.	Baja
RF 18	Modificar actividades estudiantiles	El sistema debe permitir modificar un estudiante que ha participado en una actividad estudiantil.	Media
RF 19	Eliminar actividades estudiantiles	El sistema debe permitir eliminar un estudiante con participación en una actividad estudiantil.	Alta
RF 20	Agregar actividades estudiantiles	El sistema debe permitir agregar un estudiante con participación en las actividades estudiantiles.	Baja
RF 21	Listar incidencia	El sistema debe permitir mostrar los estudiantes con incidencias.	Media
RF 22	Modificar incidencia	El sistema debe permitir modificar un estudiante con incidencias.	Alta
RF 23	Eliminar incidencia	El sistema debe permitir eliminar un estudiante con incidencias.	Media
RF 24	Agregar incidencia	El sistema debe permitir insertar un estudiante con incidencias.	Alta
RF 25	Mostrar estrategia educativa por brigadas	El sistema debe permitir mostrar la estrategia educativa de cada brigada.	Media
RF 26	Modificar estrategia educativa por brigadas	El sistema debe permitir modificar la estrategia educativa de cada brigada.	Media
RF 27	Eliminar estrategia educativa por brigadas	El sistema debe permitir eliminar la estrategia educativa de cada brigada.	Media
RF 28	Crear estrategia educativa por brigadas	El sistema debe permitir insertar la estrategia educativa de cada brigada.	Baja
RF 29	Mostrar estrategia educativa por años	El sistema debe permitir mostrar la estrategia educativa de cada año.	Baja
RF 30	Modificar estrategia educativa por años	El sistema debe permitir modificar la estrategia educativa de cada año.	Alta
RF 31	Eliminar estrategia educativa por años	El sistema debe permitir eliminar la estrategia educativa de cada año.	Baja
RF 32	Crear estrategia educativa por años	El sistema debe permitir crear la estrategia educativa de cada año.	Baja
RF 33	Mostrar arrastres	El sistema debe permitir mostrar los estudiantes con arrastres.	Alta
RF 34	Modificar arrastres	El sistema debe permitir modificar un estudiante con arrastre.	Alta
RF 35	Eliminar arrastres	El sistema debe permitir eliminar un estudiante con arrastre.	Media
RF 36	Agregar arrastres	El sistema debe permitir agregar un estudiante con arrastre.	Media
RF 37	Agregar repitente	El sistema debe permitir agregar un estudiante repitente.	Media
RF 38	Listar repitentes	El sistema debe permitir listar los estudiantes repitentes.	Baja
RF 39	Modificar repitente	El sistema debe permitir modificar un estudiante repitente.	Alta

RF 40	Eliminar repitente	El sistema debe permitir eliminar un estudiante repitente.	Alta
RF 41	Mostrar carta	El sistema debe permitir mostrar las cartas a los padres de los estudiantes.	Media
RF 42	Modificar carta	El sistema debe permitir modificar una carta a los padres del estudiante.	Baja
RF 43	Eliminar carta	El sistema debe permitir eliminar una carta a los padres del estudiante.	Alta
RF 44	Crear carta	El sistema debe permitir crear una carta a los padres del estudiante.	Media
RF 45	Generar reporte de evaluaciones integradas	El sistema debe permitir generar un reporte de la evaluación integrada del estudiante.	Alta
RF 46	Generar reporte de la evaluación del componente laboral	El sistema debe permitir generar un reporte de la evaluación del componente laboral del estudiante repitente.	Media
RF 47	Generar reporte del plan de trabajo	El sistema debe permitir generar un reporte del plan de trabajo del estudiante repitente.	Alta
RF 48	Generar reporte de incidencias	El sistema debe permitir generar un reporte de las incidencias de los estudiantes.	Media
RF 49	Generar reporte de estrategia educativa por brigadas	El sistema debe permitir generar un reporte de la estrategia educativa de cada brigada de la facultad 1.	Baja
RF 50	Generar reporte de estrategia educativa por años	El sistema debe permitir generar un reporte de la estrategia educativa de cada año de la facultad 1.	Media
RF 51	Generar reporte de la carta a los padres del estudiante con problemas en la universidad	El sistema debe permitir generar un reporte de cada carta a los padres de los estudiantes con problemas en la universidad.	Baja
RF 52	Generar reporte de la carta a los padres del estudiante sin problemas en la universidad	El sistema debe permitir generar un reporte de cada carta a los padres de los estudiantes sin problemas en la universidad.	Baja
RF 53	Autenticar usuario	El sistema debe permitir autenticar un usuario.	Alta
RF 54	Crear usuario	El sistema debe permitir que los usuarios con rol ADMIN puedan agregar un usuario.	Alta
RF 55	Editar usuario	El sistema debe permitir editar un usuario.	Alta
RF 56	Eliminar usuario	El sistema debe permitir eliminar un usuario.	Baja
RF 57	Mostrar estudiante	El sistema debe permitir mostrar los datos del estudiante.	Baja
RF 58	Editar estudiante	El sistema debe permitir editar los datos del estudiante.	Media
RF 59	Eliminar estudiante	El sistema debe permitir eliminar un estudiante.	Media
RF 60	Agregar estudiante	El sistema debe permitir agregar un estudiante.	Alta

RF 61	Listar estudiantes	El sistema debe permitir listar todos los estudiantes	Baja
-------	--------------------	---	------

2.3.2 Requisitos no funcionales

Los requisitos no funcionales (RnF) representan características generales y restricciones de la aplicación o sistema que se esté desarrollando (PMOinformatica, 2018). A partir de un análisis se identificaron cuatro (4) RnF, distribuidos en especificaciones de usabilidad, software, hardware y seguridad, los cuales se relacionan a continuación:

RnF1 Usabilidad

- RnF1.1: El sistema debe mantener coherencia y uniformidad en las estructuras y colores de todas las páginas para que el usuario no se desoriente en su navegación.
- RnF1.2: El sistema propuesto podrá ser usado por personas con conocimientos básicos en el manejo de computadoras.

RnF2 Software

- RnF2.1: El servidor web debe ser Apache 2.4 o superior.
- RnF2.2: El servidor web debe tener instalado un intérprete de PHP en su versión 7.2 o superior.
- RnF2.3: El servidor de base de datos debe ser MySQL 5.7 o superior.

RnF3 Hardware

- RnF3.1: El servidor web debe poseer memoria RAM de 2 GB o superior y un procesador i3 a una velocidad de 2.10 GHz o superior.
- RnF3.2: El servidor de la base de datos debe poseer como mínimo 80GB de disco duro.

RnF4 Seguridad

- RnF4.1: La información debe estar protegida con un usuario y una contraseña.
- RnF4.2: El sistema utilizará mecanismos de cifrado para las contraseñas almacenadas en la base de datos.

2.4 Casos de uso del sistema

Los diagramas de casos de uso del sistema sirven para especificar la funcionalidad y el comportamiento de un sistema mediante la relación entre los actores y los casos de uso en un sistema. En el proceso se

describen a los profesores guías, profesores principales del año, tutores de repitentes y decana como actores del proceso y encargados de realizar las acciones en la aplicación. A continuación, se modela el diagrama de casos de uso del sistema para la aplicación:

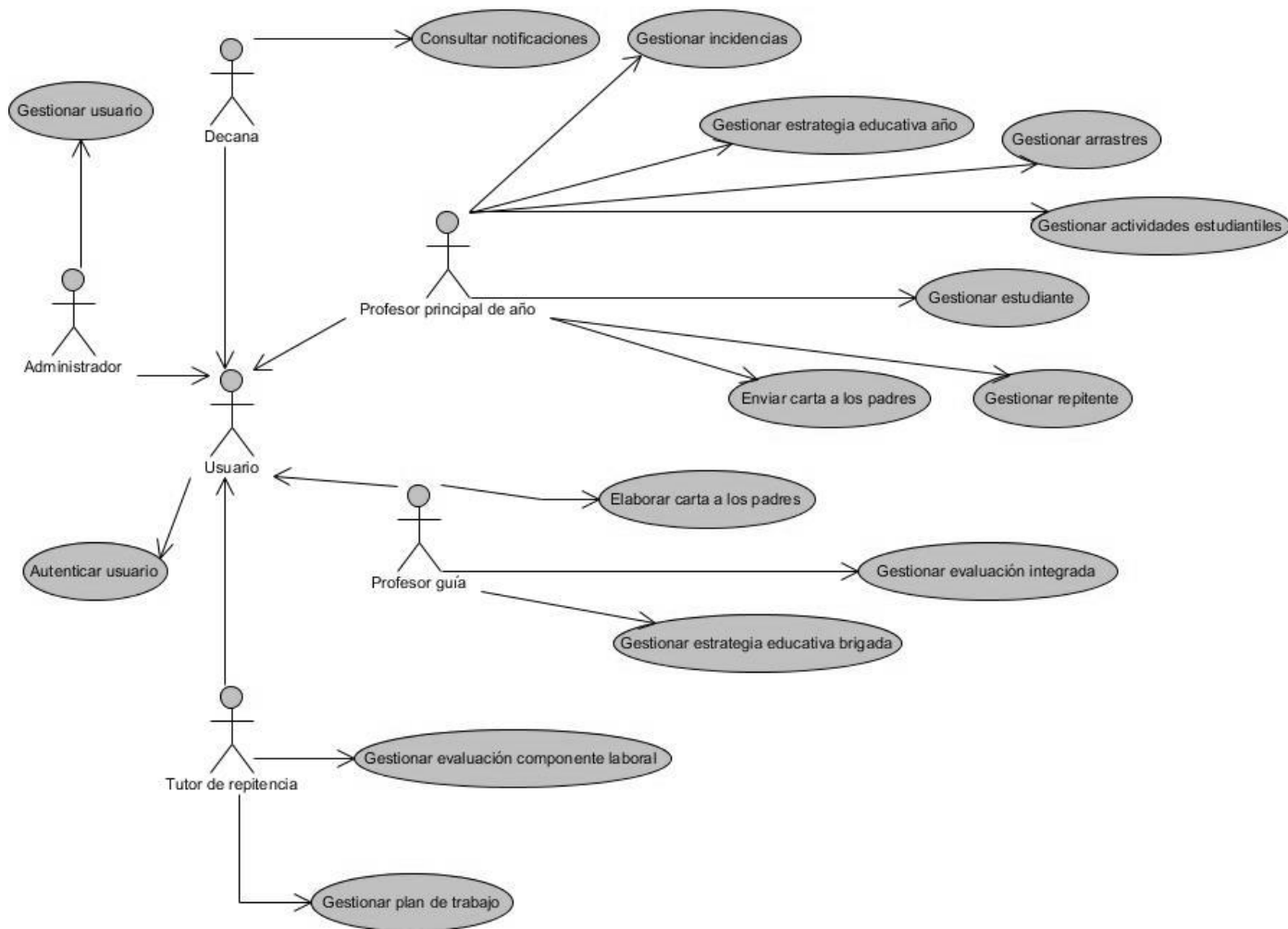


Figura 2. Diagrama de casos de uso del sistema. Elaboración propia

2.4.1 Descripción de los casos de uso

En esta sección se hará una presentación de la descripción del caso de uso del sistema de las evaluaciones integradas, donde han sido asociadas las pantallas correspondientes que muestra el sistema que se obtuvo como resultado de esta investigación.

Tabla 2. Descripción del CU-1. Gestionar evaluación integrada. Elaboración propia

Objetivo	Permitir insertar, modificar y eliminar datos acerca de una evaluación integrada.	
Actores	Profesor guía.	
Resumen	El caso de uso se inicia cuando el administrador del sistema decide insertar, modificar o eliminar datos acerca de una evaluación integrada.	
Complejidad	Alta.	
Prioridad	Alto.	
Precondiciones	Profesor guía ya autenticado.	
Postcondiciones	Se insertó, modificó o se eliminó una evaluación integrada.	
Flujo de eventos		
Flujo básico “Gestionar Evaluaciones Integradas”		
Actor	Sistema	
1.	Selecciona de la página principal la opción “Evaluación integrada”.	
2.		Muestra una pantalla con un listado de evaluaciones integradas y permite insertar, Modificar o Eliminar evaluaciones integradas.
3.	Desea insertar, modificar o eliminar evaluaciones integradas.	
4.		Da la posibilidad de realizar alguna de las siguientes acciones: a) Si decide insertar una evaluación integrada, ir a la sección “Agregar nueva evaluación integrada”. b) Si decide modificar los datos de un usuario, ir a la sección “Editar evaluación integrada”. c) Si decide eliminar una evaluación integrada, ir a la sección “Eliminar evaluación integrada”.
Prototipo elemental de interfaz gráfica		

Evaluaciones integrales

Estudiante	Acciones	Agregar evaluación integrada
Bryan Ramirez Diaz	 	
Ivan Javier Rivero Fernandez	 	
Magdiel Fernandez Santana	 	

Evaluación Integrada

Estudiante	Bryan Ramirez Diaz
Militancia	No
CI	95091728923
Político Ideológico	El estudiante ha participado en la macha de las antorchas, marcha por el fusilamiento de los 8 estudiantes de medicina y primero de mayo. Fue seleccionado como presidente de grupo en tercer año. Participó en la salida al cine yara para ver la película "Inocencia".
Académico	El estudiante tiene como índice académico 3.617 en los 5 años de la carrera . En el transcurso de la misma realice 10 exámenes extraordinarios, tres en primer año (MD1), (M1) para subir nota y (M2), en segundo (Física), (M3) y (FAGO), en tercer año (Ingles 3), (P4) y (SO), en cuarto año Internet de las cosas (IoT) y 2 exámenes mundiales en tercer año (SO) y en cuarto año la optativa (IoT).
Investigativa	El estudiante participó en la Copa Pascal a nivel de facultad los cinco años de la carrera siempre estando entre los tres primeros puestos los cuales permitieron que pasara a la Copa Pascal a nivel UCI los cinco años de la carrera, Participó en tercer año en el DevCap, obtuvo el segundo lugar en la Olimpiada de matemática a nivel UCI en tercer año, obtuvo tercer lugar en Sudoku en cuarto año y segundo lugar a nivel UCI, obtuvo 1er lugar en el concurso de lógica (CMF) en segundo año y 2do lugar en tercer año, participó en la ACM-ICPC en tercer año en el equipo "Sky-Kru", participó en varios eventos colaterales en los cinco años de carrera.
Extensión Universitaria	Participación en los Festivales de Artistas Aficionados (FAA) en 3er año obteniendo 1 plata y en 4to año obteniendo 4 oros y 1 bronce, participó en el FAA a nivel provincial en 4to año, participará en los FAA en 5to año. Participó en los Juegos Mellas en 3er, 4to y 5to año en 100 metros, 4x100, futbol 11 obtenido la medalla de bronce en el 4x100 en 4to año. Participó en el mundialito de futbol de la Facultad 1 obteniendo el 6to lugar. Participó en la competencia de ajedrez interaño obteniendo el 2do lugar.

[Editar](#)[Eliminar](#)[Atras](#)

Flujos Alternos

2a. Listado de evaluaciones integradas vacío

Actor		Sistema
1.		Carga una pantalla en blanco sin evaluaciones integradas registradas.

Sección 1: "Agregar evaluación integrada"

Flujo básico Agregar evaluación integrada

Actor		Sistema
1.	Presiona el botón "Agregar evaluación integrada".	
2.		<p>Muestra una ventana con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> • Estudiante. • CI. • Militancia. • Político Ideológico. • Académico. • Investigativa. • Extensión Universitaria. <p>Y los botones "Crear" y "Cancelar".</p>
3.	Introduce los datos y presiona el botón "Crear".	
4.		Verifica que todos los campos estén llenos.
5.		Verifica que los datos introducidos estén correctos.
6.		Verifica que este usuario no exista.
7.		<p>Almacena los datos de la evaluación integrada.</p> <p>Finalizando así el caso de uso.</p>

Prototipo elemental de interfaz gráfica

Estudiante

CI

Militancia

Político Ideológico:


Académico:

Investigativa:

Extensión Universitaria:

Flujos Alternos

4a. Campos vacíos

Actor		Sistema
1.		Muestra el mensaje "Rellene este campo".
5a. Datos incorrectos		
Actor		Sistema
1.	CI	Muestra el mensaje "Solo puede tener números". Muestra el mensaje "Este valor debe tener exactamente 11 dígitos".
6a. Estudiante existente		
Actor		Sistema
1.		Muestra el mensaje "Este estudiante ya existe".
Sección 2: "Editar evaluación integrada"		
Flujo básico Editar evaluación integrada		
Actor		Sistema
1.	Marca la opción "Editar" ó  de los estudiantes mostrado y presiona el botón "Editar".	
2.		Muestra una ventana con los siguientes campos a introducir: <ul style="list-style-type: none"> • Estudiante. • CI. • Militancia. • Político Ideológico. • Académico. • Investigativa. • Extensión Universitaria. Y los botones "Editar" y "Cancelar".
3.	Realiza las actualizaciones deseadas y presiona el botón "Editar".	
4.		Verifica que todos los campos estén llenos.
5.		Verifica que los datos introducidos estén correctos.
6.		Actualiza la información incorporada a la evaluación integrada. <i>Finalizando así el caso de uso.</i>

Prototipo elemental de interfaz gráfica

Estudiante

bramirez



CI

95091728923

Militancia

Político Ideológico:

El estudiante ha participado en la macha de las antorchas, marcha por el fusilamiento de los 8 estudiantes de medicina y primero de mayo. Fue seleccionado como presidente de grupo en tercer año. Participó en la salida al cine yara para ver la película "Inocencia".

Académico:

El estudiante tiene como índice académico 3.617 en los 5 años de la carrera . En el transcurso de la misma realice 10 exámenes extraordinarios, tres en primer año (MD1), (M1) para subir nota y (M2), en segundo (Física), (M3) y (FAGO), en tercer año (Ingles 3), (P4) y (SO), en cuarto año Internet de las cosas (IoT) y 2 exámenes mundiales en tercer año (SO) y en cuarto año la optativa (IoT).

Investigativa:

El estudiante participó en la Copa Pascal a nivel de facultad los cinco años de la carrera siempre estando entre los tres primeros puestos los cuales permitieron que pasara a la Copa Pascal a nivel UCI los cinco años de la carrera, Participó en tercer año en el DevCap, obtuvo el segundo lugar en la Olimpiada de matemática a nivel UCI en tercer año, obtuvo tercer lugar en Sudoku en cuarto año y segundo lugar a nivel UCI, obtuvo

Extensión Universitaria:

Participación en los Festivales de Artistas Aficionados (FAA) en 3er año obteniendo 1 plata y en 4to año obteniendo 4 oros y 1 bronce, participó en el FAA a nivel provincial en 4to año, participará en los FAA en 5to año. Participó en los Juegos Mallas en 3er, 4to y 5to año en 100 metros, 4x100, futbol 11 obteniendo la medalla de bronce en el 4x100 en 4to año. Participó en el mundialito de futbol de la Facultad 1 obteniendo el 6to

Editar

Cancelar

Flujos Alternos

4a. Campos vacíos

Actor		Sistema
1.		Muestra el mensaje "Rellene este campo".

5a. Datos incorrectos

Actor		Sistema
1.	CI	Muestra el mensaje "Solo puede tener números". Muestra el mensaje "Este valor debe tener exactamente 11 dígitos".

Sección 3: "Eliminar evaluación integrada"**Flujo básico Eliminar evaluación integrada**

Actor		Sistema
1.	Presiona el botón "Eliminar" para una evaluación integrada	
		Muestra el mensaje de confirmación "¿Está seguro que desea eliminar la evaluación integrada del estudiante seleccionado?". Y los botones "Aceptar" y "Cancelar".
2.	Presiona el botón "Aceptar".	
		Elimina la evaluación integrada. Finalizando así el caso de uso.

Prototipo elemental de interfaz gráfica

Evaluación Integrada

Estudiante	Bryan Ramirez Diaz
Militancia	No
CI	95091728923
Político Ideológico	El estudiante ha participado en la macha de las antorchas, marcha por el fusilamiento de los 8 estudiantes de medicina y primero de mayo. Fue seleccionado como presidente de grupo en tercer año. Participó en la salida al cine yara para ver la película "Inocencia".
Académico	El estudiante tiene como índice académico 3.617 en los 5 años de la carrera . En el transcurso de la misma realice 10 exámenes extraordinarios, tres en primer año (MD1), (M1) para subir nota y (M2), en segundo (Física), (M3) y (FAGO), en tercer año (Ingles 3), (P4) y (SO), en cuarto año Internet de las cosas (IoT) y 2 exámenes mundiales en tercer año (SO) y en cuarto año la optativa (IoT).
Investigativa	El estudiante participó en la Copa Pascal a nivel de facultad los cinco años de la carrera siempre estando entre los tres primeros puestos los cuales permitieron que pasara a la Copa Pascal a nivel UCI los cinco años de la carrera, Participó en tercer año en el DevCap, obtuvo el segundo lugar en la Olimpiada de matemática a nivel UCI en tercer año, obtuvo tercer lugar en Sudoku en cuarto año y segundo lugar a nivel UCI, obtuvo 1er lugar en el concurso de lógica (CMF) en segundo año y 2do lugar en tercer año, participó en la ACM-ICPC en tercer año en el equipo "Sky-Kru", participó en varios eventos colaterales en los cinco años de carrera.
Extensión Universitaria	Participación en los Festivales de Artistas Aficionados (FAA) en 3er año obteniendo 1 plata y en 4to año obteniendo 4 oros y 1 bronce, participó en el FAA a nivel provincial en 4to año, participará en los FAA en 5to año. Participó en los Juegos Mellas en 3er, 4to y 5to año en 100 metros, 4x100, futbol 11 obteniendo la medalla de bronce en el 4x100 en 4to año. Participó en el mundialito de futbol de la Facultad 1 obteniendo el 6to lugar. Participó en la competencia de ajedrez interaño obteniendo el 2do lugar.

[Editar](#)[Eliminar](#)[Atras](#)

¿Está seguro que desea eliminar la evaluación integrada de Ivan Javier Rivero Fernandez?

Aceptar

Cancelar

Flujo Alternativo

2a. Cancelar eliminación de usuario

Actor	Sistema
1.	Presiona el botón "Cancelar".
2.	Vuelve al paso 2 del flujo básico "Gestionar Evaluaciones Integradas".

2.5 Arquitectura de software

La arquitectura de software es un conjunto de patrones que definen la organización de un sistema, sus componentes, el ambiente, y los principios que orientan su diseño y evolución. Los patrones arquitectónicos representan la estructura del sistema que comprende a los elementos de software, las propiedades visibles externamente de dichos elementos y las relaciones entre ellos (Naranjo, 2018).

Modelo-Vista-Controlador (MVC)

Para el desarrollo de la aplicación se utilizará el patrón de arquitectura de software Modelo Vista Controlador el cual permite desarrollar la aplicación rápidamente de forma modular y mantenible. Al separar las funciones en modelo, vista, controlador ase que la aplicación se vuelva muy ligera y permite hacer cambios en una parte de la aplicación sin que las demás partes se vean afectadas. Está formado por tres niveles como se muestra en la Figura 3:

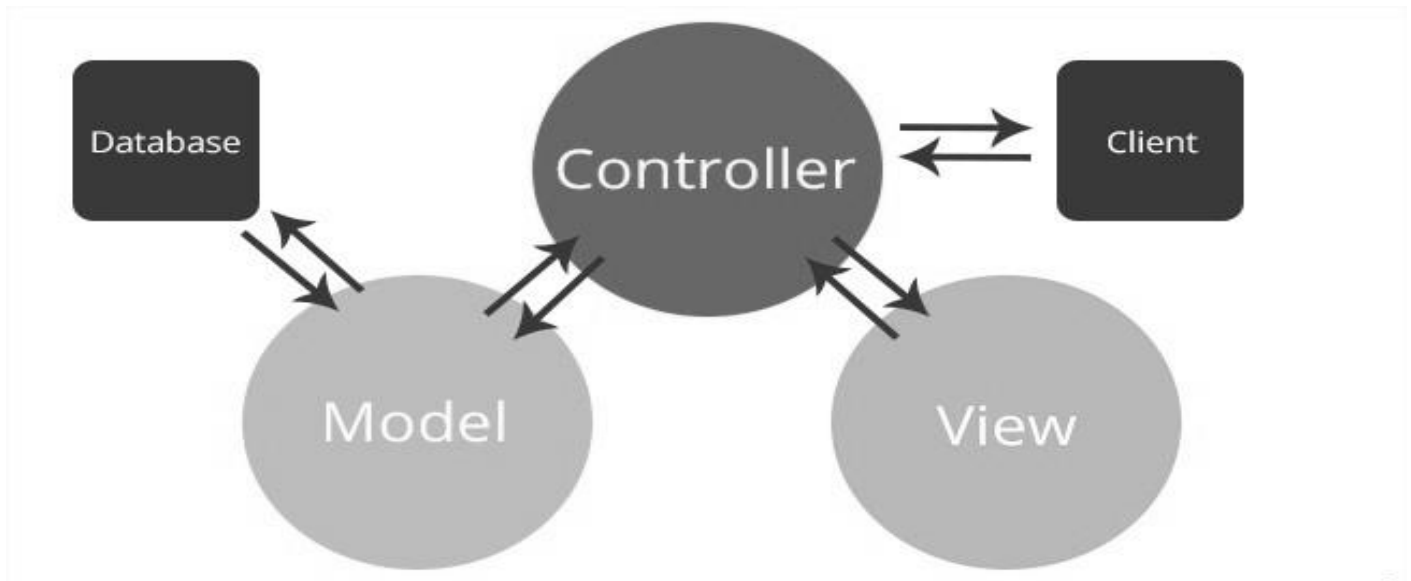


Figura 3. Arquitectura de software MVC. Fuente. (Santiago, 2018)

- **Modelo:** Es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. No tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo. Para trabajar con la base de datos se utiliza el ORM Doctrine con el que trabaja Symfony para la creación de las entidades de la aplicación como son EvaluacionIntegrada.php e Incidencia.php.
- **Vista:** Es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una presentación visual del modelo y muestra los datos al usuario como se evidencian en cada uno de los CRUD (*Create, Read, Update and Delete*) en la aplicación.
- **Controlador:** Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista estos se evidencian en la aplicación en las clases CartaController.php y EstudianteArrastrandoController.php.

2.6 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular, y brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Los patrones son una pareja problema/solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades (Guevara, 2015). Entre los patrones que se utilizarán se encuentran los patrones generales de software para la asignación de responsabilidades (GRASP por sus siglas en inglés *General Responsibility Assignment Software Patterns*) y los patrones GoF (por sus siglas en inglés *Gang Of Four*).

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

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos (Izquierdo, 2016). Los patrones GRASP utilizados en la aplicación fueron:

Creador: Se refleja en las clases controladoras ubicadas en el directorio /src/ProyectoBundle/Controller, donde se encuentran las acciones definidas para las operaciones lógicas del negocio referentes a las entidades y se ejecutan cada una de ellas.

Experto: Se pone en práctica en las clases EstudianteRepitenteController.php y EstudianteArrastrandoController.php, que son las que tienen la información necesaria para cumplir las responsabilidades de generar las correspondientes fichas de los estudiantes con arrastres y repitentes.

Controlador: Permite que a través de un archivo se procesen todas las peticiones de manipulación por analizar a través de un objeto de controlador único. Estas peticiones web son tratadas por un solo controlador frontal que posee el marco de trabajo llamado “app.php” siendo este el único punto de entrada a la aplicación.

Alta Cohesión: Se evidencia al definir una clase controladora para cada entidad, donde cada una maneja la información perteneciente solamente a la entidad correspondiente, de esta forma las

responsabilidades están fuertemente ligadas a la entidad, en un sentido lógico, lo que permitió que estas clases sean fáciles de comprender, reutilizar y conservar.

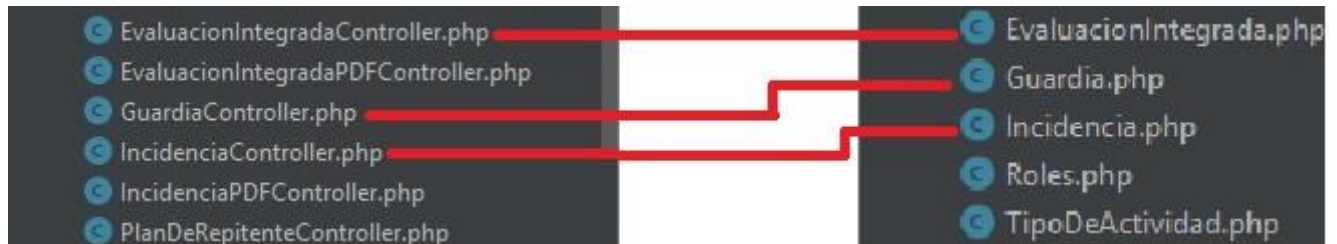


Figura 4. Ejemplo de uso del patrón Alta Cohesión. Elaboración propia.

Bajo Acoplamiento: El uso de este patrón en el diseño facilita la existencia solo de las relaciones entre clases necesarias, de forma tal que pequeños cambios tengan la mínima repercusión en otras clases. También permite que las clases sean fáciles de entender por separado y de reutilización.

2.6.2 Patrones Gang of Four (GOF)

Los patrones GoF describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos y se agrupan en tres grandes categorías: creacionales, estructurales y de comportamiento (PRESSMAN, 2010). El utilizado en el diseño es el siguiente:

Decorador: El motor de plantillas *Twig*, está provisto de un mecanismo de herencia gracias al cual la decoración de plantillas se realiza de una manera flexible y versátil (Domecq Babie, y otros, 2015). Este patrón se observa en el fichero “base.html.twig” ubicados en el directorio app/Resources/views, que es el que contiene el código HTML que es común para todas las páginas, por lo que cada página que se crea heredará de esta.

```
{% extends 'base.html.twig' %}

{% block body %}
    <div class="row">
        <h2 class="sub-header mio" align="center">Evaluaciones integrales</h2>
        <div class="table-responsive">
            <table class="table table-hover">
                <thead>
                <tr>
```

Figura 5. Ejemplo de uso del patrón Decorador. Elaboración propia.

2.7 Diagrama de clases de análisis y diseño

A diferencia del modelo conceptual, un diagrama de clases de análisis y diseño muestra definiciones de entidades software más que conceptos del mundo real (Izquierdo, 2016). Para la presente investigación se generó un diagrama de clases de diseño para cada caso de uso. A continuación, se expone el diagrama de clases de diseño para el CU-1 Gestionar Evaluaciones Integradas.

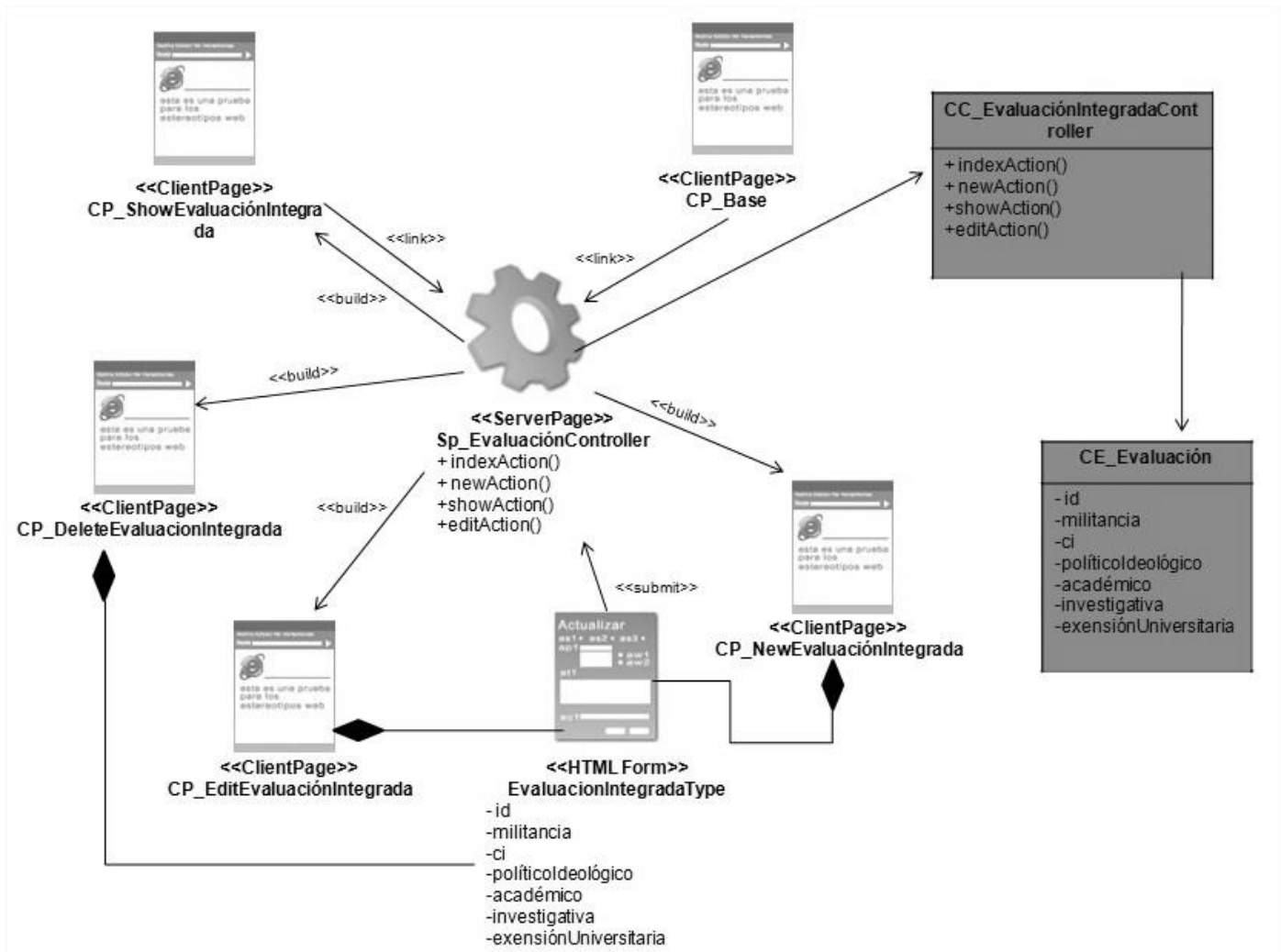


Figura 6. Diagrama de clases del diseño. Gestionar Evaluaciones Integradas. Elaboración propia.

2.8 Modelo de datos

Conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructura y tipos de datos, operaciones y restricciones (Capote Vázquez, y otros, 2015). A continuación, se muestra una propuesta del modelo de datos del Sistema para la gestión de la información de los colectivos de años de la Facultad 1 en la UCI.

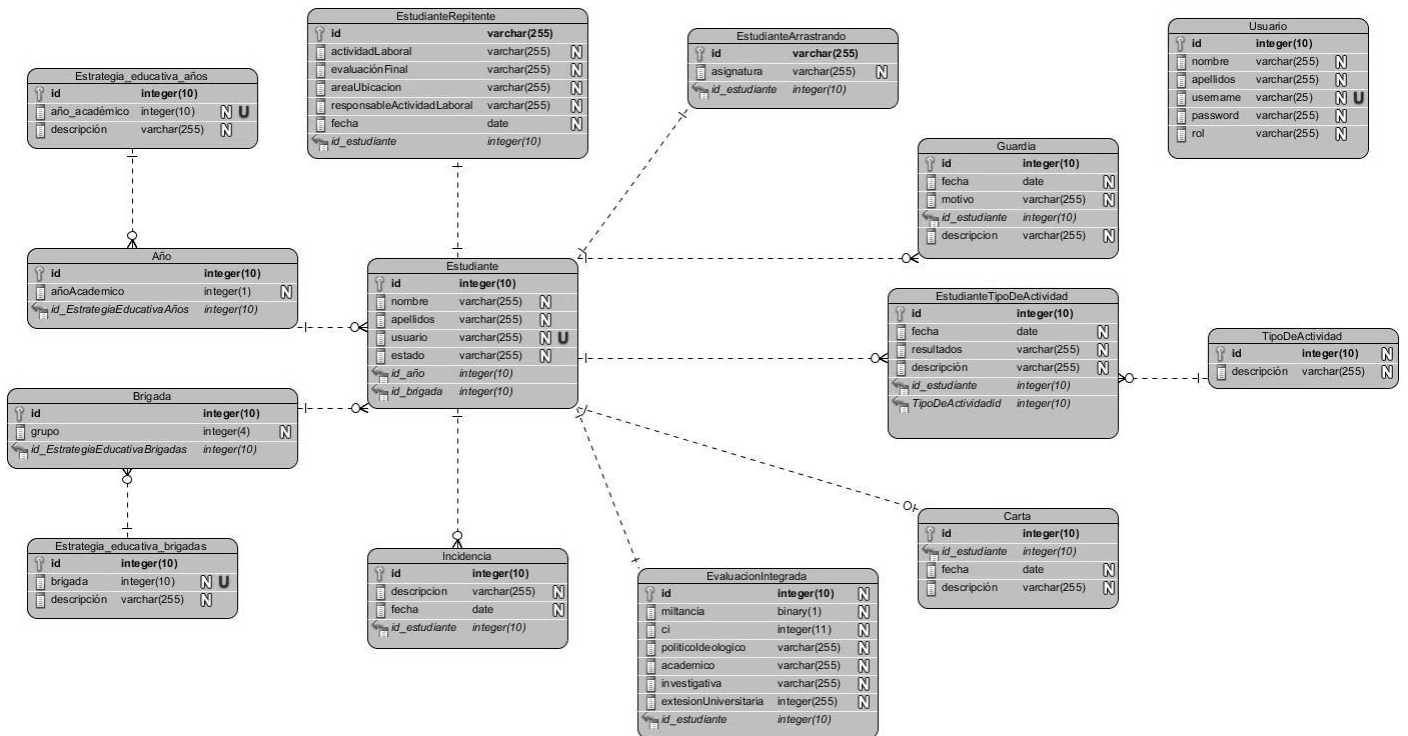


Figura 7. Modelo de datos. Elaboración propia.

Usuario: Registra la información de las personas que trabajan con el sistema. Se almacena el atributo que lo identifica (id), nombre(s), apellidos, username, password y el rol.

Estudiante: Registra la información de los estudiantes de la facultad 1. Se almacena el atributo que lo identifica (usuario), nombre, apellidos, estado, id_año y id_brigada.

Año: Almacena la información de los años académicos de la Facultad 1. Registra el identificador (id), los años académicos y el identificador asociado a la tabla EstrategiaEducativaAños (id_EstrategiaEducativaAños).

Brigada: Almacena la información de las brigadas de la Facultad 1. Registra el identificador (id), las brigadas y el identificador asociado a la tabla EstrategiaEducativaBrigadas (id_EstrategiaEducativaBrigadas).

EvaluacionIntegrada: Almacena la información de las evaluaciones integradas de cada uno de los estudiantes de la Facultad 1. Registra si el estudiante es militante, su carnet de identidad, las categorías político ideológico, académico, investigativa y extensión universitaria y el identificador asociado a la tabla estudiante (id_estudiante).

EstudianteRepitente: Almacena la información de los estudiantes que están repitiendo en la Facultad 1. Registra el identificador (id), actividad laboral, la evaluación final, el área donde va a realizar la actividad laboral, el responsable a cargo de la actividad laboral, la fecha, y el identificador asociado a la tabla estudiante (id_estudiante).

EstudianteArrastrando: Almacena la información de los estudiantes que están arrastrando en la Facultad 1. Registra el identificador (id), las asignaturas que está arrastrando y el identificador asociado a la tabla estudiante (id_estudiante).

Incidencia: Almacena la información de los estudiantes que han cometido incidencias. Registra el identificador (id), una descripción sobre la incidencia y el identificador asociado a la tabla estudiante (id_estudiante).

Carta: Almacena la información de las cartas a los padres de los estudiantes. Registra el identificador (id), la fecha en que se redacta la carta, una descripción y el identificador asociado a la tabla estudiante (id_estudiante).

EstudianteTipoDeActividad: Almacena la información de las actividades culturales, deportivas e investigativas de los estudiantes de la Facultad 1. Registra el identificador (id), la fecha, una descripción dependiendo del tipo de actividad, los resultados obtenidos y los identificadores asociados a las tablas estudiante (id_estudiante) y TipoDeActividad (TipoDeActividadid).

TipoDeActividad: Almacena los tipos de actividades que pueden ser Juegos Deportivos, Festival de Artistas Aficionados ó Eventos colaterales. Registra el identificador (id), y el tipo de actividad.

Guardia: Almacena la información de los estudiantes que no participaron en las guardias estudiantiles. Registra el identificador (id), la fecha, el motivo de la ausencia, una descripción y el identificador asociado a la tabla estudiante (id_estudiante).

Estrategia_educativa_brigadas: Almacena la información de la estrategia educativa por brigadas. Registra el identificador (id), el número de la brigada, una descripción sobre esta.

Estrategia_educativa_años: Almacena la información de la estrategia educativa por años académicos. Registra el identificador (id), el número del año académico y una descripción sobre esta.

2.9 Modelo de despliegue

El modelo de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se relacionan en esos nodos.

En la siguiente figura se muestra de la forma que el usuario tendrá acceso a la aplicación, esto será a través de un navegador web en una PC cliente, con una conexión por HTTPS por el puerto 443 al servidor web Apache. La aplicación se conecta al servidor de base de datos a través del protocolo TCP por el puerto 3306 y realiza las consultas a la base de datos de la aplicación.

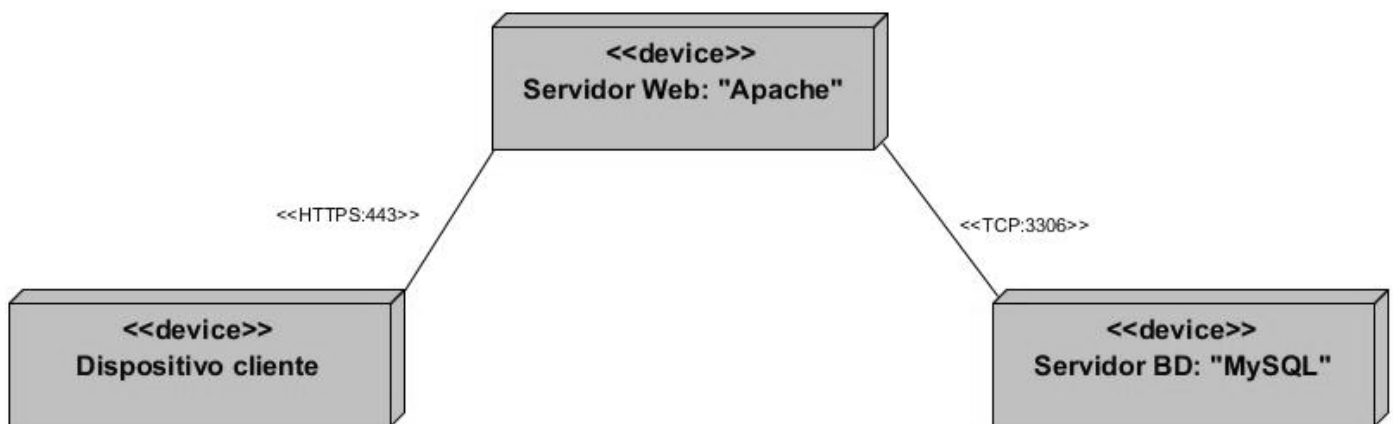


Figura 8. Diagrama de despliegue. Elaboración propia.

La descripción de los nodos de la figura anterior se muestra a continuación:

- **Dispositivo cliente:** Este nodo representa la PC cliente.

- **Servidor Web:** En este nodo se encuentra desplegado o instalado el servidor web apache donde estará desplegada la aplicación.
- **Servidor de Base de Datos:** En este nodo se encuentra desplegado o instalado el servidor de base de datos con el que interactúa el sistema.

Conclusiones del capítulo

En este capítulo se describió la propuesta de solución determinando las actividades que serán informatizadas dentro del proceso de gestión de la información de los colectivos de año de la Facultad 1 en la UCI. La metodología de desarrollo seleccionada para guiar el proceso de software, posibilitó la descripción de 4 actores que interactúan con el sistema. Se definieron 61 requisitos funcionales y 4 requisitos no funcionales. Adoptar la arquitectura de software Modelo-Vista-Controlador propuesta por el framework de desarrollo utilizado, abonó una propicia organización del sistema a implementar.

Capítulo 3: Implementación y validación del sistema de gestión de información de los colectivos de años de la Facultad 1

En el presente capítulo se muestran los artefactos correspondientes a las etapas de implementación y pruebas del sistema, así como los estándares de codificación que debe seguir el equipo de desarrollo para implementar el software. De acuerdo a la metodología que se utiliza, se especifican, de los tipos de pruebas que esta plantea, el que será empleado para validar el sistema.

3.1 Diagrama de componentes

El diagrama de componentes proporciona una visión física de la construcción del sistema. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos (Naranjo, 2018). A continuación, se expone el diagrama de componentes utilizado para las evaluaciones integradas.

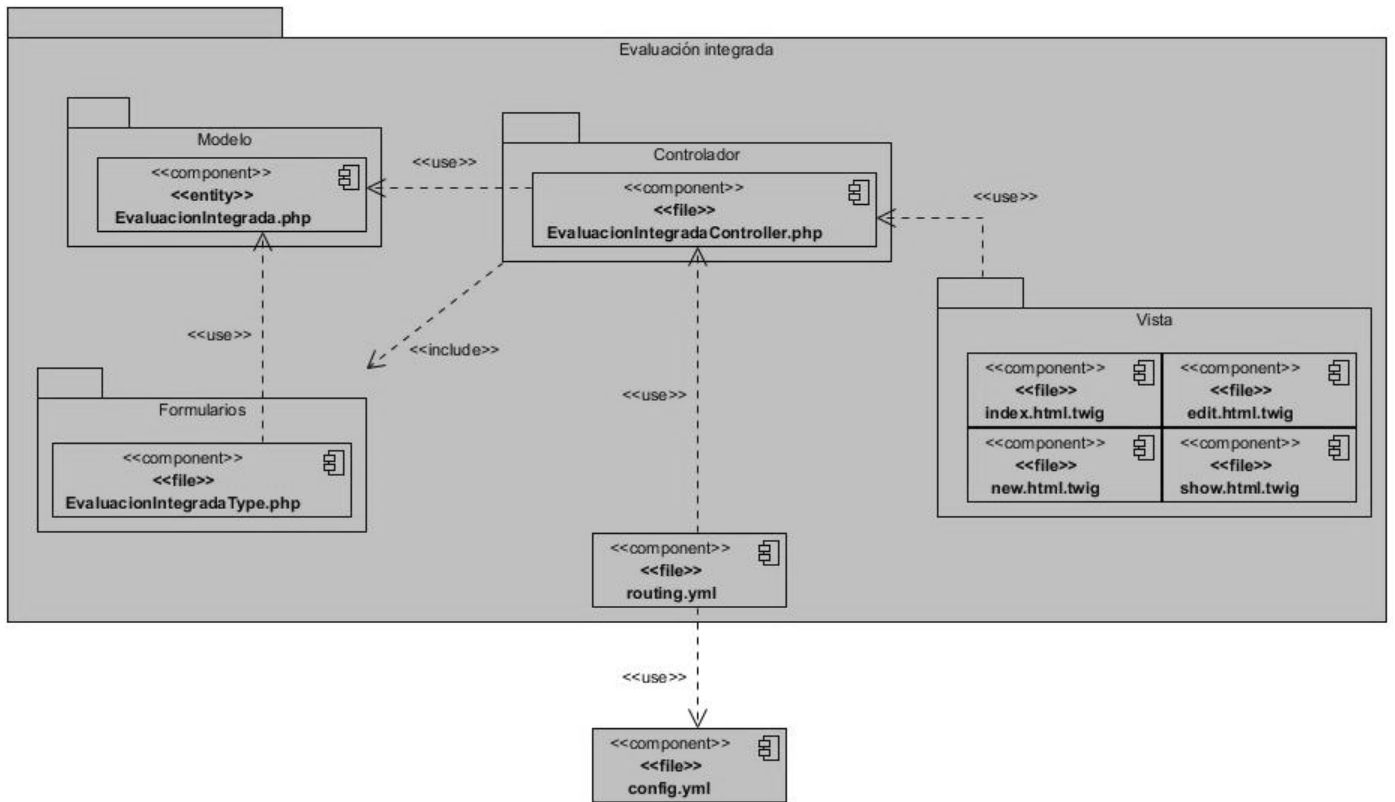


Figura 9. Diagrama de componentes. Elaboración propia.

3.1.1 Descripción del diagrama de componentes

Tabla 3. Descripción de las Evaluaciones integradas. Elaboración propia.

		Componentes	Descripción	
SGICA	Evaluación Integrada	Modelo	EvaluacionIntegrada.php Clase entidad que almacena los datos de las evaluaciones integradas de los estudiantes de la Facultad 1 en la UCI.	
		Controlador	EvaluacionIntegradaController.php Clase controladora que contiene las acciones relacionadas con las evaluaciones integradas de los estudiantes de la Facultad 1 en la UCI. Realiza acciones generales de gestión (insertar, eliminar, mostrar y editar) y genera un reporte.	
		Formulario	EvaluacionIntegradaType.php Clase responsable de construir los formularios relacionados con las evaluaciones integradas.	
		Vista	index.html.twig	Muestra un listado de las evaluaciones integradas de los estudiantes.
			edit.html.twig	Permite editar las evaluaciones integradas de los estudiantes.
			new.html.twig	Permite registrar las evaluaciones integradas de los estudiantes.
			show.html.twig	Muestra la evaluación integrada de un estudiante.
			routing.yml	Contiene el mapa de rutas URL que se enlazan a las acciones de los controladores del módulo.
	config.yml	Contiene las configuraciones generales de la aplicación.		

3.2 Estándares de codificación

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación genera un código de alta calidad. Si se aplica de forma continua un estándar de codificación bien definido existen muchas posibilidades de que el proyecto de software sea un sistema de software fácil de comprender y de mantener. A continuación, se muestra el estilo de código seguido en la aplicación.

1. La apertura de llaves para clases y funciones deben estar en la próxima línea después de su declaración.

```
public function indexAction()
{
    $em = $this->getDoctrine()->getManager();

    $evaluacionIntegradas = $em->getRepository( className: 'EvaluacionIntegrada')->findAll();
    $estudiantes = $em->getRepository( className: 'Estudiante')->findAll();
    return $this->render( view: 'evaluacionintegrada/index', array(
        'evaluacionIntegradas' => $evaluacionIntegradas,
        'estudiante' => $estudiantes,
    ));
}
```

Figura 10. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.

2. Las aperturas de llaves en las estructuras de control se colocan en la misma línea.

```
if ($form->isSubmitted() && $form->isValid()) {
    $em = $this->getDoctrine()->getManager();
    $em->persist($evaluacionIntegrada);
    $em->flush();

    return $this->redirectToRoute( route: '/[id]/show', array('id' => $evaluacionIntegrada->getId()));
}
```

Figura 11. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.

3. El cierre de llaves para clases, funciones y estructuras de control deben estar en la próxima línea después del bloque de código.

```

if ($form->isSubmitted() && $form->isValid()) {
    $em = $this->getDoctrine()->getManager();
    $em->persist($evaluacionIntegrada);
    $em->flush();

    return $this->redirectToRoute( route: '/{id}/show', array('id' => $evaluacionIntegrada->getId()));
}

```

Figura 12. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.

4. La indentación se realiza con tabulación y no con 4 espacios en blanco.

```

public function editAction(Request $request, EvaluacionIntegrada $evaluacionIntegrada)
{
    $deleteForm = $this->createDeleteForm($evaluacionIntegrada);
    $editForm = $this->createForm( type: 'ProyectoBundle\Form\EvaluacionIntegradaType', $evaluacionIntegrada);
    $editForm->handleRequest($request);
}

```

Figura 13. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.

5. Se deberá agregar una línea en blanco delante de la sentencia *return*.

```

if ($editForm->isSubmitted() && $editForm->isValid()) {
    $this->getDoctrine()->getManager()->flush();

    return $this->redirectToRoute( route: '/{id}/edit', array('id' => $evaluacionIntegrada->getId()));
}

```

Figura 14. Fragmento de código fuente de “EvaluacionIntegradaController”. Elaboración propia.

6. Se puede agregar líneas en blanco para mejorar la legibilidad e indicar bloques de códigos relacionados.
7. Debe haber una declaración por línea.

```

public function indexAction()
{
    $em = $this->getDoctrine()->getManager();
    $incidencias = $em->getRepository('className: 'Incidencia')->findAll();

    return $this->render( view: 'incidencia/index', array(
        'incidencias' => $incidencias,
    ));
}

```

Figura 15. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.

8. Los archivos *PHP* deben utilizar solo las etiquetas de apertura `<?php` omitiendo las etiquetas de cierre `?>`.
9. Utilizar la codificación UTF-8.
10. Se debe utilizar un espacio en blanco después de la palabra clave en las estructuras de control.

```

if $form->isSubmitted() && $form->isValid() {
    $em = $this->getDoctrine()->getManager();
    $em->persist($incidencia);
    $em->flush();
}

```

Figura 16. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.

11. No deben existir espacios en la apertura y cierre de los paréntesis.

```

if ($form->isSubmitted() && $form->isValid()) {
    $em = $this->getDoctrine()->getManager();
    $em->persist($incidencia);
    $em->flush();
}

```

Figura 17. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.

12. Debe usarse un espacio en blanco entre los operadores lógicos, aritméticos, de comparación y asignación.

```

if ($form->isSubmitted() && $form->isValid()) {
    $em=$this->getDoctrine()->getManager();
    $em->persist($incidencia);
    $em->flush();
}

```

Figura 18. Fragmento de código fuente de “IncidenciaController”. Elaboración propia.

3.3 Validación de la propuesta de solución

Los procesos de pruebas de software se destacan como el proceso de establecer datos de entrada al producto de software e inspeccionar los resultados obtenidos, permitiendo determinar si el software funciona realmente como se espera. Las pruebas de software responden fundamentalmente a dos interrogantes, ¿se ha obtenido un buen producto?, ¿se ha desarrollado de forma correcta? Este concepto da lugar al proceso de verificación y validación del software (Cañete, 2016).

Según (Cañete, 2016) las pruebas de software se diferencian del resto de las actividades del proceso de desarrollo de software en que estas son un proceso destructivo. El objetivo de quien realiza las pruebas, es descubrir errores que no han sido detectados con anterioridad. Entre los principales tipos de pruebas realizadas a aplicaciones web se encuentran las pruebas funcionales y las pruebas de seguridad.

3.3.1 Pruebas funcionales

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas funcionales se hacen mediante el diseño de casos de prueba con el objetivo de evaluar cada una de las opciones con las que cuenta el paquete informático. A continuación, se describen los casos de prueba para el CU-1 Gestionar evaluaciones integradas.

Tabla 4. Caso de prueba CU-1_P1. Elaboración propia.

Prueba de aceptación	
Código: CU-1_P1	Caso de Uso: Gestionar evaluación integrada
Nombre: Agregar evaluación integrada	
Descripción: Se crea una evaluación integrada con los campos estudiante, CI, militancia, político ideológico, académico, investigativa y extensión universitaria.	
Condiciones de ejecución:	

El usuario debe estar autenticado. El usuario debe tener el rol de ADMIN o PROFESOR.
Entrada/Pasos de ejecución: Una vez autenticado en el sistema, se selecciona en el menú de la izquierda la opción Evaluación integrada, luego selecciona la opción agregar evaluación integrada y seguidamente se llenan todos los campos requeridos del formulario, los campos obligatorios son estudiante, CI, político ideológico, académico, investigativa y extensión universitaria. Ninguno de los campos obligatorios puede quedar en blanco, se debe asegurar que el estudiante seleccionado no tenga una evaluación integrada, el CI solo puede contener números y exactamente 11 dígitos. Luego se presiona el botón "Crear".
Resultado esperado: El sistema muestra la evaluación integral creada.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 5. Caso de prueba CU-1_P2. Elaboración propia.

Prueba de aceptación	
Código: CU-1_P2	Caso de Uso: Gestionar evaluación integrada
Nombre: Editar evaluación integrada	
Descripción: Se edita una evaluación integrada y se verifica que todos los campos requeridos se llenen correctamente.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe tener el rol de ADMIN o PROFESOR.	
Entrada/Pasos de ejecución: Una vez autenticado en el sistema, se selecciona en el menú de la izquierda la opción Evaluación integrada, luego selecciona la evaluación integrada a editar y seguidamente se modifican los campos deseados del formulario, asegurándose que ninguno de los campos obligatorios quede en blanco, que el CI solo contenga números y tenga exactamente 11 dígitos. Luego se presiona el botón "Editar".	
Resultado esperado: El sistema muestra la evaluación integral editada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6. Caso de prueba CU-1_P3. Elaboración propia.

Prueba de aceptación	
Código: CU-1_P3	Caso de Uso: Gestionar evaluación integrada
Nombre: Eliminar evaluación integrada	
Descripción: Se elimina una evaluación integrada.	
Condiciones de ejecución: El usuario debe estar autenticado. El usuario debe tener el rol de ADMIN o PROFESOR.	
Entrada/Pasos de ejecución: Una vez autenticado en el sistema, se selecciona en el menú de la izquierda la opción Evaluación integrada, luego selecciona la opción "mostrar evaluación integrada" de la evaluación integral deseada, luego se selecciona la opción "Eliminar" y seguidamente y seguidamente el sistema muestra un mensaje para confirmar la eliminación del mismo. Luego se selecciona la opción "Aceptar".	
Resultado esperado: El sistema muestra el listado de las evaluaciones integradas sin la eliminada.	
Evaluación de la prueba: Prueba satisfactoria.	

Para ejecutar satisfactoriamente la etapa de prueba se realizaron tres iteraciones. En la primera iteración realizada se detectaron 22 no conformidades que causaban que el sistema no se comportara de manera esperada, de las cuales 9 de errores de ortografía, 5 de funcionalidad y 8 de validaciones, a las cuales se les dio solución en su totalidad. En la segunda iteración de pruebas se encontraron 4 nuevas no conformidades, de las cuales 3 de validaciones y 1 de funcionalidad, a las cuales se les dio solución en su totalidad, y para una tercera iteración no fueron encontradas no conformidades. A continuación, se muestra de manera gráfica el resultado de las pruebas aplicadas:

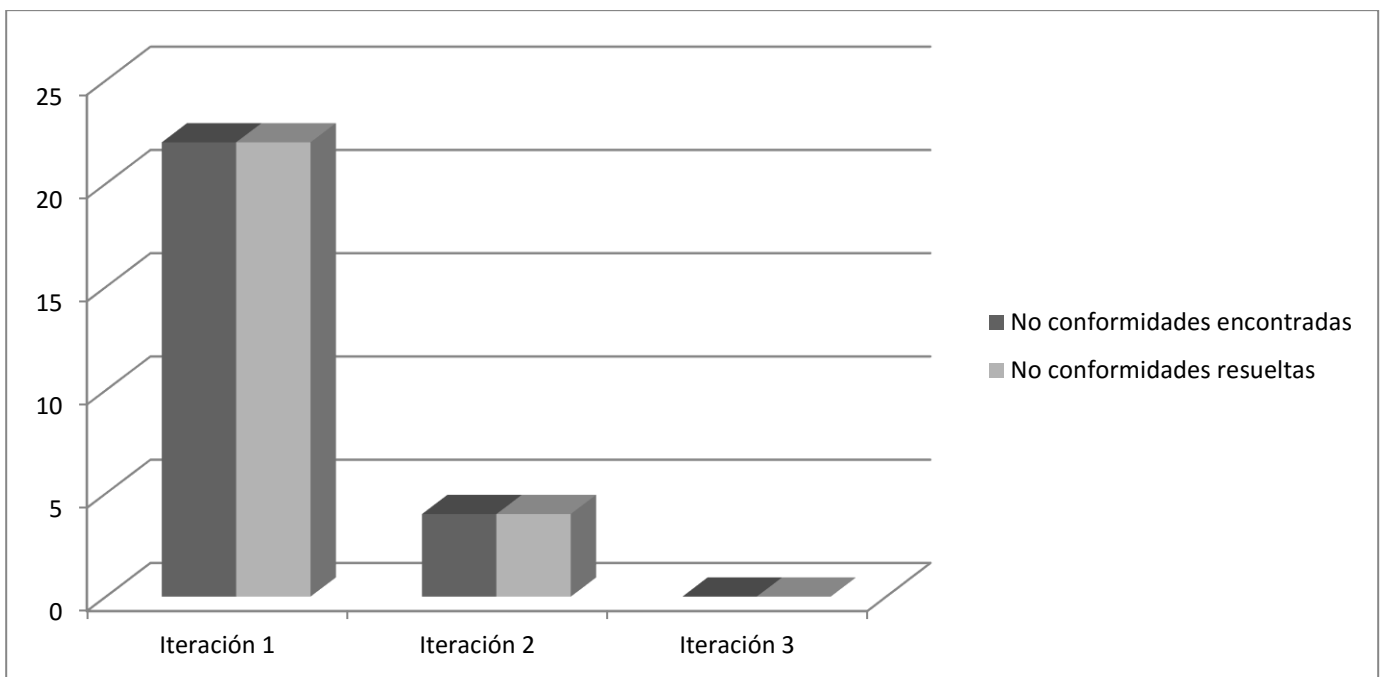


Figura 19. No conformidades por iteración. Elaboración propia.

3.3.2 Pruebas de seguridad

Cualquier sistema basado en computadora que gestione información sensible o cause acciones que puedan dañar (o beneficiar) de manera inadecuada a individuos es un blanco de penetración inadecuada o ilegal. La penetración abarca un amplio rango de actividades: hackers que intentan penetrar en los sistemas por deporte, empleados resentidos que intentan penetrar por venganza, individuos deshonestos

que intentan penetrar para obtener ganancia personal ilícita. La prueba de seguridad intenta verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia (PRESSMAN, 2010).

Para probar la seguridad del sistema se aplicó el método de prueba basada en herramienta para el cual se seleccionó el sistema Acunetix 12.0 el cual luego de haber escaneado el sistema generó un reporte de las vulnerabilidades de seguridad más importantes clasificadas por el impacto que estas pudieran tener en cuanto a los niveles de seguridad, la clasificación se realiza en alta, media, baja e informativa. En una primera iteración se identificaron 12 vulnerabilidades, a las cuales se les dió solución, entre las que se encuentran: formulario HTML sin la protección CSRF (*Cross-site request forgery*), envío de información sensible en texto plano, el método TRACE (*Transition Regional And Coronal Explorer*) está activado, enlaces rotos y el tipo de contraseña de entrada con autocompletado habilitado. De las vulnerabilidades detectadas se encontraron 6 medias, 4 bajas y 2 informativas como se muestra en la Figura 20 a continuación.

Scan details

Scan information	
Start time	31/05/2019, 00:46:13
Start url	http://localhost/portal/web/
Host	localhost
Scan time	206 minutes, 19 seconds
Profile	Full Scan
Server information	Apache/2.4.34 (Win32) OpenSSL/1.1.0i PHP/7.2.9
Responsive	True
Server OS	Windows
Server technologies	PHP

Threat level

Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

Alerts distribution

Total alerts found	12
🔴 High	0
🟡 Medium	6
🟢 Low	4
🔵 Informational	2

Figura 20. Reporte de la prueba de seguridad. Fuente. Acunetix.

Conclusiones del capítulo

A lo largo del proceso de implementación el uso de los estándares de codificación definidos contribuyó a obtener una adecuada uniformidad y legibilidad en el código fuente. El diagrama de componentes dio la posibilidad de ver las dependencias entre los componentes del sistema, además reflejó, de manera más detallada, la ubicación física de los componentes de acuerdo a la arquitectura empleada. La aplicación de pruebas funcionales y de seguridad a la solución desarrollada permitió encontrar errores que afectaban el funcionamiento del sistema, lo que posibilitó corregirlos a tiempo para que el mismo cumpliera totalmente con los requisitos funcionales definidos en la etapa de análisis.

Conclusiones

Con la realización de esta investigación se desarrolló un sistema informático capaz de contribuir al proceso de la gestión de la información de los colectivos de años de la Facultad 1 en la Universidad de las Ciencias Informáticas. Los objetivos planteados fueron cumplidos de manera satisfactoria obteniendo a su vez una serie de resultados:

- El análisis de la estructura de los colectivos de años, permitió definir el flujo lógico de la información, identificar los roles que interactúan con el sistema y ayudó a estructurar una propuesta de solución para el sistema informático desarrollado.
- Se realizó un análisis de los sistemas similares existentes tanto nacional como internacionalmente, de los cuales se obtuvo conocimiento para saber cómo se desarrolla en el mundo el uso de las aplicaciones informáticas dedicadas a la gestión de la información de los colectivos de años en las instituciones de educación superior la cual nos proporcionó una serie de características como generar informes de las actividades docentes de los estudiantes y registrar las actividades deportivas, culturales e investigativas.
- Con la implementación de la aplicación se dio cumplimiento al objetivo general propuesto debido a que permite realizar una valoración más profunda de los estudiantes, disminuye la probabilidad de que la información se encuentre duplicada o sufra pérdidas y disminuye el tiempo de búsqueda de dicha información.
- La validación del sistema a partir de las pruebas funcionales y de seguridad realizadas, permitió comprobar el correcto funcionamiento de la herramienta y la conformidad con los requisitos de software definidos.

Recomendaciones

Durante el desarrollo de la investigación y el desarrollo del sistema surgen aspectos que se deben tener en cuenta para el futuro perfeccionamiento del mismo, los cuales se sugieren a continuación:

- Implementar un componente de notificaciones (vía Email) que permita comunicar a los estudiantes con incidencias la fecha del análisis disciplinario.
- Implementar un componente para el sistema, el cual permita conectarse a la base de datos de la UCI y obtener toda la información relacionada con el proceso académico de los estudiantes de la Facultad 1.

Bibliografía

- Alegsa, Leandro.** alegsa.com.ar. [En línea] 18 de 07 de 2016. <http://www.alegsa.com.ar/Dic/informacion.php>.
- Alonso, Evelyn Menéndez.** Herramientas CASE para el proceso de desarrollo de Software. [En línea] 2009. <https://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
- Aranda, Odalys Font.** *Implementación de un Sistema de Gestión Documental en la Universidad Central "Marta Abreu" de las Villas, Cuba: Facultad de Ciencias de la Información y de la Educación.* 2013.
- Arévalo, Julio Alonso.** *Gestión de la Información, gestión de contenidos y conocimiento.* 2007.
- Avitia, Cristian.** *Comparacion de SGBD.* 2019.
- Ayala, Israel.** es.ccm.net. [En línea] 27 de 03 de 2017. <https://es.ccm.net/contents/304-lenguajes-de-programacion>.
- Bembibre, Victoria.** Definición ABC. [En línea] 2009. <https://www.definicionabc.com/tecnologia/portal.php>.
- Bry, Alej.** academia.edu. [En línea] 2019. https://www.academia.edu/26088366/HERRAMIENTAS_CASE_1_.
- Bueno, Lisbeth Ocaña.** *Gestor de Base de Datos.* 2017.
- Caballero, Julio Giampiere Grados.** *JavaScript el lenguaje más importante y potente en la web.* 2017.
- Cañete, José Enrique Beúnes.** *Portal de Formación y Orientación Profesional para el estudio de la Ingeniería en Ciencias Informáticas en la UCI.* 2016.
- Capote Vázquez, Osmar y Vázquez Leyva, Raúl.** *Sistema de Gestion de Informacion para el Vicedecanato de Investigacion y Postgrado de la Facultad 2.* 2015.
- Carlos.** hostname.cl. [En línea] 26 de febrero de 2014. <https://www.hostname.cl/blog/servidor-web>.
- Díaz Rosabal, Elena María, y otros.** *Las TIC y la gestión del conocimiento.* 2017. ISSN: 2387-0893.
- Domecq Babie, Dayana y Macías Hernández, Juan Darien.** *Sistema para la gestión de información de Investigación, Desarrollo e Innovación Tecnológica del Centro de Informatización de Entidades.* 2015.
- Domínguez, Pablo.** openclassrooms.com. [En línea] 27 de 03 de 2018. <https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery/4309498-que-es-jquery>.
- Editor, Alfaomega Grupo.** *HTML5 y CSS3 - Para diseñadores.* 2016.

Ferrer Martínez, Leonardo y Martínez Rojas, Leonardo. *Sistema de Gestión de la Información para la Autoevaluación de la Facultad 4.* 2015.

Foundation, Apache Software. What is the Apache HTTP Server Project? [En línea] 2017. https://httpd.apache.org/ABOUT_APACHE.html.

Gonzalez, Eric Usgame. <https://es.calameo.com/books/004584916ba12ecb233a6>. [En línea] 2017.

Guevara, Geinel Dorta. *Módulo de contabilidad para el Sistema de Gestión de Información de la asignación, distribución y consumo de Combustible en la UCI.* 2015.

Izquierdo, Rachel Cristina Morales. *Sistema web para la gestión de la información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.* 2016.

—. 2016. *Sistema web para la gestión de la información del Departamento de Cuadros del Comité Central del Partido Comunista de Cuba.* 2016.

Lapiente, María Jesús Lamarca. [hipertexto.info](http://www.hipertexto.info). [En línea] 29 de 07 de 2018. <http://www.hipertexto.info/documentos/html.htm>.

Maria, Luiggi Santa. [staffcreativa.pe](http://www.staffcreativa.pe). [En línea] 2015. <http://www.staffcreativa.pe/blog/ventajas-programacion-php/>.

MENJÍ, CRISIA MABEL. *Diseño y desarrollo de una plataforma de creación de reportes para el Sistema Integrado de Indicadores Gerenciales del Ministerio de Salud.* 2016.

MES. *Perfeccionamiento del sistema de gestión del proceso de formación integral de los estudiantes universitarios en el eslabón de base (segunda parte).* s.l. : Editorial Universitaria Félix Varela, 2014.

Naranjo, Esteban Roberto de León. *MÓDULO PARA LA GESTIÓN DE SERVIDORES PROXY SQUID DEL SISTEMA XILEMA SMART KEEPER.* 2018.

php. php.net. [En línea] 2019. <https://php.net/manual/es/intro-what-is.php>.

PhpStorm. [jetbrains.com](https://www.jetbrains.com/phpstorm/). [En línea] 2018. <https://www.jetbrains.com/phpstorm/?fromMenu>.

PMOinformatica. [pmoinformatica.com](http://www.pmoinformatica.com). [En línea] 30 de 05 de 2018. <http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>.

Prensa, Centro de Información para la. <http://congresopcc.cip.cu/politica-economica>. [En línea] 2019.

PRESSMAN, Roger. *Ingeniería de software.* 2010.

Pressman, Roger S. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO.* s.l. : MC Graw Hill, 2010, pág. 767.

QUALITY. Pruebas de Seguridad. [En línea] 2016. <http://vyvquality.com/pruebas-seguridad/>.

Rodríguez, Yordanis Rodríguez. *UN ACERCAMIENTO AL SISTEMA DE GESTIÓN DE INGRESO PARA LA EDUCACIÓN SUPERIOR DE CUBA.* 2018.

SIGA. datasae.co. [En línea] 2018. <http://www.datasae.co/siga/>.

Symfony. Symfony. [En línea] 2019. <https://symfony.es/pagina/que-es-symfony/>.

uclv. uclv.edu.cu. [En línea] 2017.
[uclv.edu.cu/handle/123456789/1504/browse?type=subject&value=Sistema+de+Gestión+de+la+Nueva+Un
iversidad+\(SIGENU\)](http://uclv.edu.cu/handle/123456789/1504/browse?type=subject&value=Sistema+de+Gestión+de+la+Nueva+Universidad+(SIGENU)).

UNMSM. unmsm.edu.pe. [En línea] 2009. <http://www.unmsm.edu.pe/noticias/ver/2009>.

wordpress. wordpress.com. [En línea] abril de 2016. <https://perikiadgm23.wordpress.com/gestion-de-la-informacion/>.