



Herramienta informática para la gestión del plan de trabajo metodológico (SiGPTM)

**Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas**

Autores:

Dayana de la Caridad Capestany Varona

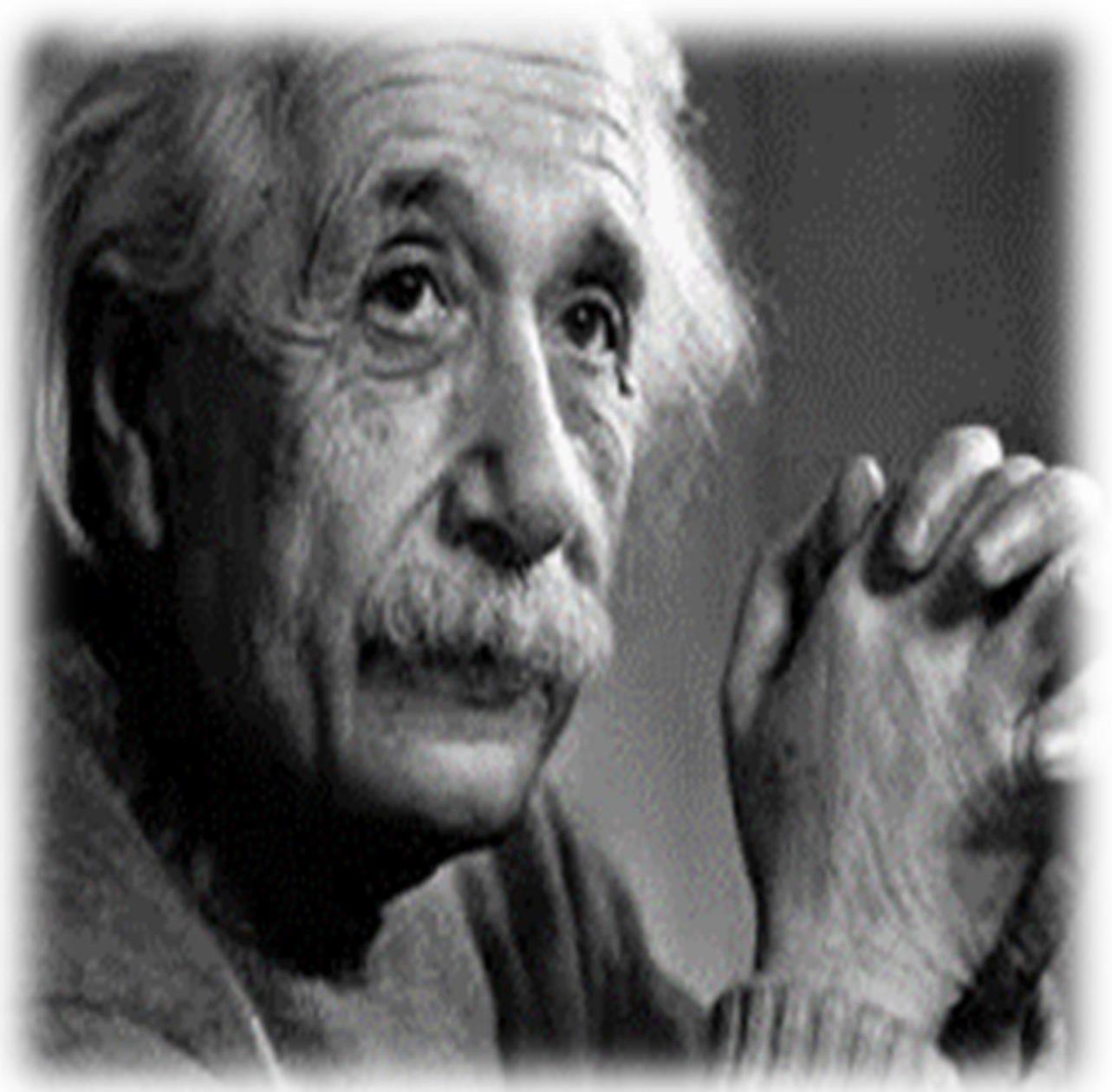
Darlyn Díaz Morales

Tutores:

MsC. Elizabeth Rodríguez Stiven

Dr.C Yoan Martínez Márquez

MsC. Yalice Gámez Batista



*"Nunca consideres el estudio como una obligación, sino
como una oportunidad para penetrar en el bello y
maravilloso mundo del saber"*

Albert Einstein

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos el presente a los ____ días del mes de _____ del año _____.

Autores:

Dayana de la Caridad Capestany Varona

Darlyn Díaz Morales

Tutores:

MsC. Elizabeth Rodríguez Stiven

Dr.C Yoan Martínez Márquez

MsC. Yalice Gámez Batista

Dedicatoria

Agradecimientos

Resumen

En la actualidad las instituciones educacionales a nivel mundial están obligadas al mejoramiento continuo debido a los constantes cambios que operan en la sociedad, la rapidez con que cambian las tecnologías, y el incremento significativo de la información. Dada la necesidad de elevar la calidad educacional en las instituciones educativas es importante expresar el papel que juega el trabajo metodológico en las mismas, pues va dirigido a cambiar puntos de vista, estilos de trabajo y modos de actuación de los implicados. En la Universidad de las Ciencias Informáticas la gestión del trabajo metodológico es llevado a cabo por los directivos docentes, quienes son los encargados de ejecutar las funciones del trabajo metodológico en su área. Cada curso académico las acciones a realizar se plasman en un plan de trabajo metodológico las que podrán adecuarse para los períodos lectivos, en correspondencia con el diagnóstico y los resultados que se vayan alcanzando. Este proceso presenta insuficiencias debido a que la información está dispersa por las diversas áreas administrativas, se genera en distintos formatos y no se garantiza el acceso a los datos generados de cursos anteriores.

La presente investigación propone la herramienta SiGPTM que permite gestionar el plan de trabajo metodológico contribuyendo a obtener la información organizada, estructurada y disponible. En el desarrollo de la aplicación se emplearon herramientas y tecnologías de código abierto. La calidad fue constatada a través de pruebas que validan el funcionamiento del sistema asegurándose que el mismo cumpla con las normas vigentes y responda a los requerimientos del cliente además se verificó la viabilidad de la herramienta mediante una encuesta de satisfacción con la aplicación de la técnica ladov.

Palabras claves: sistema de gestión, trabajo metodológico, plan de trabajo metodológico

Índice de contenido

Introducción	1
Capítulo 1: Fundamentación Teórica.....	5
1.1 Marco teórico de la investigación.....	5
1.2 Análisis de los Sistemas de Gestión Universitaria.....	8
1.3 Tecnologías y herramientas.....	16
1.3.1 Lenguajes de Programación	17
1.3.2 Marco de trabajo.....	18
1.3.3 Entorno de Desarrollo Integrado	19
1.3.4 Gestor de Base de datos	19
1.3.5 Herramienta de modelado.....	19
1.4 Metodología de desarrollo	20
1.5 Conclusiones parciales.....	23
Capítulo 2: Herramienta para la gestión del plan de trabajo metodológico (SiGPTM).....	24
2.1 Descripción del sistema	24
2.2 Desarrollo de la herramienta.....	26
2.3 Fase de Planificación.....	27
2.3.1 Especificación de requisitos del sistema	27
2.3.1.1 Roles del sistema.....	27
2.3.1.2 Requisitos funcionales	28
2.3.1.3 Requisitos no funcionales	31
2.3.2 Historias de Usuario.....	32
2.3.3 Plan de iteraciones	33
2.3.4 Plan de entrega	36
2.4 Fase de Diseño.....	37
2.4.1 Modelo de datos	37
2.4.1.2 Patrones utilizados en el diseño del modelo de datos.....	38

2.4.2 Tarjetas CRC	38
2.4.3 Arquitectura de desarrollo	39
2.4.4 Patrones de diseño	41
2.4.5 Métricas para la validación del diseño	42
2.5 Fase de Desarrollo	47
2.5.1 Estándares de codificación	47
2.5.2 Interfaces del sistema	48
2.6 Conclusiones parciales	50
Capítulo 3: Validación de la calidad de la herramienta SiGPTM	51
3.1 Fase de Pruebas	51
3.2 Pruebas unitarias	52
3.3 Pruebas de integración	54
3.4 Pruebas de sistema	55
3.4.1 Pruebas de liberación	55
3.5 Pruebas de aceptación	56
3.6 Valoración del aporte práctico de la investigación	57
3.7 Conclusiones parciales	60
Conclusiones generales	61
Recomendaciones	62
Bibliografía	63

Índice de Tablas

Tabla 1: Análisis de los Sistemas de Gestión Universitaria (SGU). Elaboración Propia.....	16
Tabla 2: Usuarios del sistema. Elaboración Propia	28
Tabla 3.Requisitos funcionales. Componente Ejecución. Elaboración Propia	28
Tabla 4.Requisitos funcionales. Componente Planificación. Elaboración Propia	29
Tabla 5.Requisitos funcionales. Componente Control. Elaboración Propia	30
Tabla 6.Requisitos funcionales. Componente Análisis. Elaboración Propia.....	30
Tabla 7.Historia de usuario: Crear acta de actividad metodológica. Elaboración Propia.....	32
Tabla 8.Plan de iteraciones. Elaboración Propia	34
Tabla 9. Plan de entrega. Elaboración Propia	36
Tabla 10.Tarjeta CRC de la clase acta_actividad_metodologica. Elaboración Propia	38
Tabla 11: Criterio de evaluación de la métrica TOC	43
Tabla 12: Criterios de evaluación d a métrica RC.....	45
Tabla 13: Caso de prueba de caja blanca correspondiente al camino 5. Elaboración Propia	54
Tabla 14: Caso de prueba del escenario crear acta de actividad metodológica. Elaboración Propia	55
Tabla 15: Caso de prueba de aceptación de la HU "Crear PTM". Elaboración Propia.....	57
Tabla 16: Cuadro lógico de ladov. Elaboración Propia	59
Tabla 17: Resultados de la escala de satisfacción. Elaboración Propia.	60

Índice de Figuras

Figura 1: Diferencias entre las metodologías ágiles y las tradicionales. Elaboración Propia.....	21
Figura 2: Desarrollo del proyecto basado en XP. Elaboración Propia.....	22
Figura 3: Descripción del sistema. Elaboración Propia.....	24
Figura 4: Modelo de datos. Elaboración Propia.....	37
Figura 5: Ejemplo del patrón llave subrogada. Elaboración Propia.....	38
Figura 6: Interacción entre los componentes MTV. Elaboración Propia.....	40
Figura 7: Código de patrón Experto. Elaboración Propia.....	41
Figura 8: Código de patrón Creador. Elaboración Propia.....	41
Figura 9: Código de patrón Controlador. Elaboración Propia.....	42
Figura 10: Código de patrón Decorador. Elaboración Propia.....	42
Figura 11: Métrica TOC correspondiente al atributo Responsabilidad. Elaboración Propia.....	43
Figura 12: Métrica TOC correspondiente al atributo Complejidad. Elaboración Propia.....	44
Figura 13: Métrica TOC correspondiente al atributo Reutilización. Elaboración Propia.....	44
Figura 14: Métrica RC correspondiente al atributo Acoplamiento. Elaboración Propia.....	45
Figura 15: Métrica RC correspondiente al atributo Cantidad de pruebas. Elaboración Propia.....	46
Figura 16: Métrica RC correspondiente al atributo Reutilización. Elaboración Propia.....	46
Figura 17: Nomenclatura de los comentarios y declaración de variables. Elaboración Propia.....	47
Figura 18: Declaración de funciones. Elaboración Propia.....	48
Figura 19: Nomenclatura de las clases. Elaboración Propia.....	48
Figura 20: Interfaz Autenticar Usuario. Elaboración Propia.....	49
Figura 21: Interfaz del sistema correspondiente a la página inicial. Elaboración Propia.....	49
Figura 22: Interfaz correspondiente a Crear acta de actividad metodológica. Elaboración Propia.....	50
Figura 23: Estrategia para la verificación de la viabilidad de la solución. Elaboración Propia.....	51
Figura 24: Código Fuente loadProfesor. Elaboración Propia.....	53
Figura 25: Grafo de flujo. Elaboración Propia.....	53
Figura 26: Resultados de la prueba de liberación. Elaboración Propia.....	56
Figura 27: Niveles de satisfacción. Elaboración Propia.....	58
Figura 28: Rango de valores para el cálculo de satisfacción grupal. Elaboración Propia.....	59

Introducción

En la actualidad las instituciones educacionales a nivel mundial están obligadas al mejoramiento continuo debido a los constantes cambios que operan en la sociedad, la rapidez con que cambian las tecnologías, y el incremento significativo de la información. El contexto educativo cubano no está ajeno a este fenómeno; es por ello que se trabaja intensamente con el objetivo de utilizar las Tecnologías de la Información y las Comunicaciones (TICs) para apoyar la gestión y mejorar la calidad de los procesos en las instituciones educacionales [1].

En este proceso de mejora continua las universidades juegan un papel fundamental por su alta responsabilidad formadora e innovadora. Al respecto Fidel Castro expresó: *“Las universidades deben convertirse en centros donde verdaderamente el espíritu académico, el espíritu de investigación y los métodos pedagógicos estén realmente a la altura de la patria que estamos forjando”* [2]. Consecuente con ello, Díaz-Canel planteó: *“La universidad con el uso de la gestión universitaria conserva, memoriza, integra, ritualiza una herencia cultural de conocimientos, ideas, valores, actualizándola, transmitiéndola. Además, esta es conservadora, regeneradora, generadora. La educación de las próximas generaciones debe ser de mayor calidad, por ende, debemos esforzarnos en optimizar los recursos y hacer una gestión universitaria de mayor calidad y eficiencia”* [1].

Dada la necesidad de elevar la calidad educacional en las instituciones educativas es importante expresar el papel que juega el trabajo metodológico en las mismas, pues va dirigido a cambiar puntos de vista, estilos de trabajo y modos de actuación de los implicados, con el fin de obtener mayor eficiencia en su práctica pedagógica. Tiene un efecto multiplicador de las experiencias y permite resolver problemas que se presentan en el proceso docente educativo, ofreciendo el modo de proceder [3].

Según Horruitiner, la actividad metodológica caracteriza el sistema de trabajo del claustro en correspondencia con las exigencias y necesidades de la sociedad en la formación de los profesionales de nivel superior, manifiesta en los objetivos de los planes de estudio, que para su desarrollo considera la adecuada integración entre lo académico, lo investigativo y lo laboral [4]. Plantea además, que la contradicción a resolver está precisamente en lograr que la estructuración del trabajo metodológico en colectivos, guiados por coordinadores en cada nivel, contribuya a la labor de dirección del proceso docente, cuya responsabilidad corresponde a los decanos y a los jefes de departamento. Las funciones principales del trabajo metodológico son la planificación, la

organización, la regulación y el control del proceso docente educativo. La adecuada gestión de estas funciones, garantiza el eficiente desarrollo del proceso docente educativo [5].

En la Universidad de las Ciencias Informáticas la gestión del trabajo metodológico se realiza por los directivos docentes, quienes son los encargados de ejecutar estas funciones en su área. Se sigue una estructura jerárquica que parte desde la Vicerrectoría de Formación, Dirección de Formación, Vicedecanato de Formación, Colectivos de Años, hasta Departamentos Docentes (estas últimas tres estructuras a nivel de facultad). El flujo de información se realiza ascendente y descendentemente a partir del cual se generan orientaciones centrales desde la Vicerrectoría. Para establecer nuevas líneas de trabajo cada área se retroalimenta del nivel superior. Los problemas detectados durante el proceso docente-educativo se toman como punto de partida para establecer los objetivos a alcanzar con el trabajo metodológico en el próximo ciclo. Estos objetivos se concretan en acciones específicas a desarrollar, utilizando las formas y tipos que se establecen en el reglamento. Las acciones a realizar se plasman en un plan de trabajo metodológico (en lo adelante PTM) para cada curso académico. Los PTM de los colectivos y niveles de dirección deben estar elaborados al inicio del curso académico y podrán adecuarse para cada período lectivo, en correspondencia con el diagnóstico y los resultados que se vayan alcanzando.

El proceso antes descrito presenta un conjunto de insuficiencias, las cuales fueron identificadas a partir de la aplicación de una entrevista estructurada (consiste en proporcionar un número fijo de preguntas cerradas de selección múltiple) a los directivos docentes de la facultad 3. Las deficiencias más relevantes se relacionan a continuación:

1. La información está dispersa por las diversas áreas administrativas generando pérdida, duplicidad e inconsistencia de la información que se obtiene durante el control y seguimiento de los PTM.
2. Los datos generados en el trabajo metodológico se guardan en diversos formatos lo que dificulta la realización de futuros análisis de opiniones o señalamientos.
3. A pesar de la existencia de un repositorio en la facultad, la forma en que se almacenan las evidencias del trabajo metodológico que están dispersas por las distintas áreas, no garantiza el acceso a los datos generados en cursos anteriores.

Dada la situación problemática anterior se plantea como **problema a resolver**:

¿Cómo gestionar el PTM de forma que contribuya a obtener información organizada, estructurada y disponible?

Teniendo en cuenta el problema antes expuesto se define como **objeto de estudio**:

Los Sistemas de Gestión Universitaria (SGU).

Enmarcándose en el **campo de acción:**

Sistemas de Gestión Universitaria que incluyen el trabajo metodológico.

A su vez se determina como **objetivo general:** Desarrollar una herramienta informática para la gestión del PTM que contribuya a la obtención de la información organizada, estructurada y disponible.

A partir del objetivo general se desglosan los siguientes **objetivos específicos:**

1. Elaborar el marco teórico de la investigación mediante el estudio de los referentes teóricos asociados a los sistemas de gestión universitaria.
2. Modelar la gestión del trabajo metodológico para obtener una organización, estructuración y disponibilidad de los datos que permita futuros análisis.
3. Implementar una herramienta informática que soporte el modelo teórico para la gestión del PTM.
4. Valorar la solución propuesta para la gestión del PTM mediante la viabilidad del software y el nivel de aceptación del cliente.

Teniendo en cuenta el problema a resolver se formuló la siguiente **idea a defender:**

El desarrollo de una herramienta informática para la gestión del PTM contribuirá a la obtención de información organizada, estructurada y disponible.

Para el desarrollo y cumplimiento de la presente investigación fueron utilizados los siguientes métodos:

Métodos Teóricos:

Histórico-Lógico: este método contribuyó a la realización de un estudio de los sistemas de gestión universitaria relevantes, con el fin de conocer la evolución y desarrollo de los mismos y el tratamiento que estos dan al trabajo metodológico.

Análisis-síntesis: el empleo de este método permitió realizar un estudio y análisis de la bibliografía consultada, tomando lo que aporta en mayor medida a la propuesta de solución.

Modelación: permitió representar modelos y diagramas de cada una de las fases de desarrollo para apoyar la investigación.

Métodos Empíricos:

Entrevista: se realizó una entrevista a los directivos docentes de la facultad 3: vicedecano de formación, jefes de departamento y profesores principales de años académicos. La entrevista permitió identificar las principales deficiencias existentes en la gestión del trabajo metodológico y la necesidad de contar con una herramienta informática para solucionar el problema.

Medición: se utilizó para obtener información numérica de la entrevista realizada, a partir de la misma se realizaron estimaciones y comparaciones cuantitativas.

El documento se encuentra estructurado de la siguiente manera:

Capítulo 1: Referentes teóricos.

En este capítulo se describe el marco teórico de la investigación, abordando los principales conceptos asociados a la gestión universitaria y el papel de las TICs en la gestión del trabajo metodológico. Además, se analizan un conjunto de sistemas de gestión universitaria en el ámbito internacional y nacional, tomando como referencia los siguientes indicadores: universidades que lo utilizan, funcionalidades, ámbito al que va dirigido (pregrado o posgrado), gestión del trabajo metodológico, análisis de los datos almacenados en el sistema que apoye la toma de decisiones (distinto del estadístico), así como el acceso al código fuente (open source o libre acceso).

Capítulo 2: Propuesta de solución.

En este capítulo se describen los elementos esenciales que componen la propuesta de solución. Se detallan los requisitos funcionales y no funcionales que se tienen en cuenta para su implementación. Además, siguiendo la metodología XP se describen las fases de desarrollo de la misma, generándose un conjunto de artefactos. Finalmente, se especifica el modelo de datos creado, la arquitectura de solución y los patrones de diseño empleados.

Capítulo 3: Verificación de la viabilidad de la propuesta de solución.

En el presente capítulo se valida la calidad de la herramienta de gestión del plan de trabajo metodológico (SiGPTM) siguiendo la cuarta fase de la metodología XP: Fase de Pruebas. Se validan los niveles de pruebas mediante las pruebas unitarias realizadas por el equipo de desarrollo, las pruebas de liberación realizadas por el grupo de calidad de software del Centro de Gobierno Electrónico (CEGEL) y las pruebas de aceptación realizadas por el cliente. Se verifica la viabilidad de la propuesta de solución mediante la aplicación de la técnica ladov para medir el nivel de satisfacción de los usuarios respecto a los tres indicadores: organización, estructuración y disponibilidad.

Capítulo 1: Fundamentación Teórica

En este capítulo se describe el marco teórico de la investigación, abordando los principales conceptos asociados a la gestión universitaria y el papel de las TICs en la gestión del trabajo metodológico. Además, se analizan un conjunto de sistemas de gestión universitaria en el ámbito internacional y nacional, tomando como referencia los siguientes indicadores: universidades que lo utilizan, funcionalidades, ámbito al que va dirigido (pregrado o posgrado), gestión del trabajo metodológico, análisis de los datos almacenados en el sistema que apoye la toma de decisiones (distinto del estadístico), así como el acceso al código fuente (open source o libre acceso).

1.1 Marco teórico de la investigación

Cada día la sociedad ejerce mayor presión sobre las instituciones, y sobre todo el sistema educativo, con el fin de lograr niveles de calidad de la educación cada vez mayor. En este sentido, son varias las definiciones que se dan en la literatura asociadas al concepto de calidad de la educación. García Hoz sostiene que está determinada por dos componentes, por una parte, la disposición, en la que se incluyen las condiciones de integridad y coherencia y, por la otra, la eficacia. En tal sentido define la calidad educativa como *"el modo de ser de la educación que reúne las condiciones de integridad, coherencia y eficacia. En otras palabras, una educación tiene calidad en la medida en que es completa, coherente y eficaz"* [6].

Por su parte Díaz Canel plantea que para lograr el mejoramiento continuo de la calidad de la educación superior es preciso hacer un aprovechamiento óptimo de la tecnología digital que favorezca la gestión de la información y el conocimiento durante toda la vida universitaria. Resaltando en su discurso la importancia de la gestión universitaria en la actualidad cuando dice: *"La universidad con el uso de la gestión universitaria conserva, memoriza, integra, ritualiza una herencia cultural de conocimientos, ideas, valores, actualizándola, transmitiéndola. Además, esta es conservadora, regeneradora, generadora. La educación de las próximas generaciones debe ser de mayor calidad, por ende, debemos esforzarnos en optimizar los recursos y hacer una gestión universitaria de mayor calidad y eficiencia"* [1].

Gestión Universitaria

La gestión universitaria o también conocida como gestión académica es el proceso mediante el cual se controlan, organizan y dirigen todas las entidades y actores que intervienen en la labor docente. Engloba otros subprocesos como la definición de los planes de estudio, el control de la matrícula administrativa, el expediente académico y personal de los estudiantes, el control de la asistencia y las evaluaciones de los mismos. En esta tarea participan directivos, personal de

secretaría y profesores [7]. Además, es un proceso fundamental referente a las actividades de formación, donde participan diversas áreas administrativas como son: Pregrado, Posgrado, Producción, Investigación, Ingreso, Ubicación Laboral, Residencia, Extensión Universitaria, entre otras [8]. El área de pregrado tiene como objetivo dirigir metodológicamente la elaboración, desarrollo y actualización permanente de los fundamentos teóricos relacionados con la formación integral de los futuros profesionales en las universidades cubanas [9]. Uno de los procesos fundamentales que da cumplimiento a ese objetivo es el trabajo metodológico encargado de la superación y actualización del conocimiento del claustro de profesores.

Trabajo Metodológico

Según la resolución número 210-07 publicada en la Gaceta Oficial de la República de Cuba en el año 2007 define el Trabajo Metodológico (TM) como la labor que, apoyados en la didáctica, realizan los sujetos que intervienen en el proceso docente educativo, con el propósito de alcanzar óptimos resultados en dicho proceso, jerarquizando la labor educativa desde la instrucción, para satisfacer plenamente los objetivos formulados en los planes de estudio [5].

De igual manera en la resolución número 119-08 aparece como el sistema de actividades que de forma permanente y sistemática se ejecuta en los diferentes niveles de educación, con el objetivo de elevar la preparación político ideológico, pedagógica-metodológica y científica de los cuadros y de los docentes graduados y en formación, para garantizar las transformaciones dirigidas a la conducción eficiente del proceso docente educativo, y que, en combinación con las diferentes formas de la superación profesional y postgraduada, permitan alcanzar el nivel de idoneidad que se requiere [13].

Consecuentemente, en la resolución número 268-91 se conceptualiza como el trabajo de dirección del proceso docente-educativo que se desarrolla en la educación superior, con el objetivo de garantizar el cumplimiento de las exigencias y necesidades de nuestra sociedad en la formación de profesionales del nivel superior, concretado en los objetivos que se establecen en los planes y programas de estudio [14].

En las resoluciones anteriores también se hace referencia a las formas organizativas, los métodos, los medios y la evaluación del aprendizaje. Las formas organizativas del trabajo metodológico también llamadas tipos de actividades metodológicas son [5]:

- La **reunión metodológica** es el tipo de trabajo docente metodológico que viabiliza el análisis, debate y toma de decisiones acerca de temas vinculados al proceso docente educativo para su mejor desarrollo.

- La **clase metodológica** es el tipo de trabajo docente-metodológico que, mediante la demostración, la argumentación y el análisis, orienta a los profesores sobre algunos aspectos de carácter metodológico que contribuyen a su preparación para la ejecución del proceso docente educativo. La clase puede tener carácter demostrativo o instructivo, y responderá a los objetivos metodológicos formulados.
- La **clase abierta** es el tipo de trabajo docente-metodológico que permite, mediante el análisis de una actividad docente prevista para los estudiantes, la orientación a los profesores de un departamento docente o de un colectivo metodológico, sobre algunos aspectos de carácter metodológico que contribuyen a su preparación para la ejecución del proceso docente educativo.
- La **clase de comprobación** es el tipo de trabajo docente-metodológico que tiene como objetivo comprobar la preparación del profesor mediante la observación de una clase desarrollada por él ante otros miembros del departamento docente.
- El **taller metodológico** es el tipo de trabajo docente-metodológico que tiene como objetivo debatir acerca de una problemática relacionada con el proceso de formación y en el cual los profesores presentan experiencias relacionadas con el tema tratado. Se proyectan alternativas de solución a dicho problema a partir del conocimiento y la experiencia de los participantes.
- El control de la actividad docente (**controles a clases**) es aquel que se realiza a una de las formas organizativas del proceso docente educativo, previsto en el horario de clases de los estudiantes, y estará dirigido a comprobar el logro de los objetivos propuestos para dicha actividad. Los resultados del control se recogerán en un documento que será firmado por el docente controlado, como constancia de que fue informado de sus resultados. El control de la actividad docente se calificará utilizando las categorías de Excelente, Bien, Regular o Mal.

Cada una de las actividades metodológicas generan un acta que recoge el debate del intercambio entre los profesores y las acciones a realizar durante el ciclo de trabajo metodológico.

Papel de las TICs en la gestión del trabajo metodológico

La introducción y el desarrollo de las TICs constituye un reto importante para la universidad. El valor estratégico que la revolución tecnológica concede a la educación universitaria en particular y a la aparición de nuevos trabajos a los que la universidad debe hacer frente son elementos que amplifican la importancia de la integración de estas tecnologías en la universidad [15].

Desarrollar la gestión del trabajo metodológico de manera que tenga un impacto en la formación integral del profesional que requiere la sociedad, es una de las vías más pertinentes que tiene la universidad actual y en particular la universidad cubana [16]. En este contexto, el uso apropiado y pertinentes de la TICs contribuye a garantizar el éxito, la calidad del proceso de formación y permite a los directivos docentes trazar estrategias de dirección efectivas. Consecuentemente con ello, es necesario incorporar a través de las TICs, los procedimientos que respondan al modo de actuación de cada proceso metodológico. Como resultado de lo anterior, se encuentran los sistemas de gestión universitaria.

1.2 Análisis de los Sistemas de Gestión Universitaria

Para Hernández un sistema de gestión es un conjunto de reglas y principios relacionados entre sí de forma ordenada, para contribuir a la gestión de procesos generales o específicos de una organización. Permite establecer una política, unos objetivos y alcanzar dichos objetivos [17]. Este proceso establece cuatro etapas que hacen de un sistema un proceso circular, pues en la medida que el ciclo se repita recurrente y recursivamente, se logrará en cada ciclo obtener una mejora.

Baños Martínez propone 4 etapas para el desarrollo de sistemas de gestión, las cuales comparten los autores [18]:

1. Etapa de Ideación

El objetivo de esta etapa es trabajar en la idea que guiará los primeros pasos del proceso de creación de los sistemas de gestión.

2. Etapa de Planeación

En esta etapa, se definen las estrategias que se utilizarán, la estructura organizacional que se requiere, el personal que se asigna, el tipo de tecnología que se necesita, el tipo de recursos que se utiliza y la clase de controles que se aplican en todo el proceso.

3. Etapa de Implementación

El propósito de esta etapa está definido por la administración, sustentada en los mecanismos o instrumentos administrativos (estrategias, tácticas, procedimientos, presupuestos, etc.), que están sistémicamente relacionados y que se obtienen del proceso de planificación.

4. Etapa de Control

El objetivo principal en esta etapa es verificar o evaluar, si el elemento seleccionado ya sea la actividad, proceso, unidad o sistema, está cumpliendo sus objetivos o a alcanzando los resultados que se esperan.

En la bibliografía consultada se analizaron siete Sistemas de Gestión Universitaria (SGU), cuatro en el ámbito internacional y tres en el ámbito nacional. El estudio de estos sistemas está basado en seis indicadores fundamentalmente: universidades que lo utilizan, funcionalidades, ámbito al que va dirigido (pregrado o posgrado), gestión del trabajo metodológico, análisis de los datos almacenados en el sistema que apoye la toma de decisiones y el acceso al código fuente (open source o libre acceso).

SIU-Guaraní: Sistema de Gestión Académica

Es un sistema de gestión académica de libre acceso desarrollado en Argentina por la Facultad de Informática de la Universidad Nacional de La Plata. Entre las universidades que lo utilizan están: Universidad de Moreno, Universidad de Avellaneda, Universidad Nacional de San Juan, Universidad de Córdoba. El diseño del sistema fue genérico para adaptarse a las características disímiles de distintas universidades nacionales [19]. Permite la gestión de los alumnos de forma segura con la finalidad de obtener información consistente para los niveles operativos y directivos (decano y secretario académico). Incorpora la planificación anual del calendario académico incluyendo comisiones y turnos de exámenes. También brinda la posibilidad de que el alumno realice por sí mismo operaciones tales como la inscripción a cursos y materias y la consulta de su situación académica. Los autores refieren que esto representa una disminución de la carga operativa ya que el propio alumno es quien realiza parte de la operación. Además, permite a los alumnos de posgrado la inscripción a materias, consulta de notas y solicitar certificados. También, realiza análisis estadísticos de la información almacenada. Los módulos o subsistemas que ofrece este sistema son [20]:

- Gestión de Matrículas: permite la inscripción y admisión, la reinscripción del alumno, el tratamiento de sanciones y su cambio de plan.
- Gestión de Cursado: administra los actos por los cuales un alumno selecciona las materias a cursar, el seguimiento de las actuaciones académicas de los alumnos durante el cursado de la materia y la registración del resultado de dicha cursada en las actas correspondientes.
- Gestión de Aulas: permite la asignación de aulas a la comisión para cursado de materias y a las mesas para el tomado de exámenes.

- Gestión de Exámenes: administra los actos por los cuales un alumno selecciona las materias a rendir y la registración del resultado de dichos exámenes en las actas correspondientes.
- Gestión de Equivalencias: permite definir política de equivalencias y gestionar las equivalencias otorgadas a los alumnos.
- Gestión de Egresados: administra el egreso de alumnos y su título. Incluye también el manejo de padrones y certificados
- Estadísticas Generales: permite generar información estadística asociada al alumno, censados, ingresantes y egresados. Debe generar, también, un archivo de interface que actualice los datos del sistema.
- Gestión de Carreras y Planes: permite el registro de las carreras y planes de estudio que establecen las resoluciones.
- Información General: permitirá a los niveles directivos de la universidad explorar la información de manera tal de ayudar en la toma de decisiones.
- Planificación: permite definir el calendario académico anual con sus respectivos periodos lectivos, comisiones, turnos de exámenes y las mesas de exámenes correspondientes.

Sin embargo, a pesar de la cantidad de servicios que brinda este sistema, no se ajusta a los intereses de la investigación, pues no gestionan evidencias de las actividades metodológicas. Es un sistema diseñado para ser usado por el decano, secretario académico y estudiantes, los directivos docentes como jefes de departamento no son usuarios del sistema.

SIGA: Sistema Integrado de Gestión Académica

Sistema comercial desarrollado en España, utilizado en la Universidad de Piura. Es de libre acceso que puede ser utilizado en muchos centros educacionales, no solo universitarios, sino también en conservatorios, academias, colegios, centro de formación de empresas, maestrías, posgrados, entre otros. Además, realiza análisis estadísticos de la información almacenada. Está compuesto por 13 módulos que cubren en su gran mayoría las necesidades de cualquier institución. Entre sus principales módulos están [21]:

- Módulo Alumno: gestión integral de alumnos matriculados en el centro, de tutorías, asistencias o faltas al aula, calificaciones y mensajerías.
- Generador de diplomas: para la realización propia y personalizada de diplomas y certificados del centro. Diversidad de formatos a definir por el usuario.

- Profesores: alta, baja, consulta y modificación de profesores. Cada profesor da clases en un curso/asignatura/grupo/turno con control de número de horas. Definición de clases a impartir. Gestión de incidencias de cualquier tipo, faltas. Historial. Listados.
- Horarios: para realizar los cuadros horarios de las clases semanales/ mensuales/ anuales. Con control de incompatibilidad de horarios y profesores. Ejemplo: avisa cuando a un profesor se le intenta programar clases en dos aulas distintas a la misma hora. Como en todos los módulos los diversos listados y horarios se imprimen en pantalla e impresora.

Las versiones que dispone este sistema en estos momentos son: SIGA Profesional y SIGA Lite. Ambas versiones tienen las mismas funciones y opciones, únicamente se diferencian en que SIGA Lite tiene limitaciones en el número de alumnos matriculados (Hasta 1000 alumnos).

Sin embargo, a pesar de la cantidad de funcionalidades que brinda este sistema, no se ajusta a los intereses de la investigación, pues no existen evidencias de la gestión de actividades metodológicas.

openSIS: Sistema de Gestión Académica de libre acceso

Es un sistema de acceso libre para estudiantes de escuelas públicas o privadas construido por profesionales de la educación. Está basado en la web, de código abierto. Se utiliza en la Universidad de Zúrich. Presenta características que incluyen información demográfica sobre los estudiantes, planificación, libro de notas, atención a estudiantes, reportes estadísticos, elegibilidad, portal para los padres y portal para los estudiantes.

Cuando se implementa openSIS en conjunto con openIntel, la administración de la escuela toma visibilidad de los datos transaccionales en la forma de conocimiento. Esto permite que sea posible realizar análisis predictivos e individualizar el aprendizaje lo cual redundará en una mejoría a profesores y alumnos. OpenIntel es un sistema de gestión académica que permite construir rápidamente portales de datos abiertos e impactantes soluciones con énfasis en análisis espacial, interacción de datos y visualización [22].

Entre las versiones que tiene se encuentran **openSIS Pro** (Profesional) y **openSIS CE** (Central). Se diferencian en que openSIS CE no incluye las funcionalidades siguientes [23]:

- Planes de lecciones para maestros: permite que los maestros puedan adjuntar planes de lecciones a los calendarios para que los padres puedan ver los temas que se tratarán en clase.
- Portal de clase - páginas web para profesores: permite a los profesores gestionar el horario de las clases y actualizar las notas.

A modo de conclusión, de este sistema se destaca la diversidad de análisis de datos que realiza en función de brindar conocimiento para las tomas de decisiones. Sin embargo, no se ajusta a las necesidades de la investigación, pues no existen evidencias de la gestión de actividades metodológicas, siendo este un aspecto relevante para la misma.

SIGMA Gestión Universitaria AIE (Agrupación de Interés Económico)

Organización sin ánimo de lucro fundada en 1996 e integrada por las universidades Pompeu Fabra, Alcalá, Autónoma de Barcelona, Carlos III, Castilla-La Mancha, Pública de Navarra, Valladolid, Zaragoza y Autónoma de Madrid. Sistema Integrado de Gestión Académica en Web (gestión virtual) es una aplicación que admite una gestión académica no presencial. SIGMA Academic (Académico) permite definir los parámetros necesarios para realizar una implantación adaptada y estructurada en función de los procesos de gestión que la universidad necesita. La solución SIGMA Academic ofrece grandes ventajas para las instituciones universitarias ya sea en la formación de pregrado o posgrado. Además, realiza análisis estadísticos de la información almacenada. Las principales universidades que utilizan este sistema son: Universidad de Deusto, Fundación universitaria de Bages y el Centro universitario La Selle. Entre sus módulos están [24]:

- Planificación académica: gestión de los recursos que la universidad pone a disposición de los estudiantes, titulaciones, personal docente, aulas.
- Planificación docente: proyección de la oferta realizada por parte de la institución en función del tiempo y recursos disponibles (profesores, aulas, horarios, asignaturas, grupos).
- Gestión de accesos: gestión de las pruebas de acceso a la universidad y de los diferentes procesos de admisión, así como la gestión de asignación de plazas.
- Gestión de grado y máster: gestión completa del ciclo de vida de los estudios de grado y máster.
- Gestión de doctorado: gestión completa del ciclo de vida de los expedientes de estudios de doctorado, incluyendo la gestión de documentos de actividades, plan de investigación, figura del tutor/director.
- Gestión de formación continua: gestión de cursos cortos, cursos de expertos, teniendo en cuenta las particularidades de este tipo de formación.
- Gestión de títulos: gestión de la tramitación de títulos para estudios oficiales y propios, así como la emisión de suplemento europeo al título.

- Análisis: solución que ofrece cuadros de mandos, informes y paneles de control en tecnología Business Intelligence (Inteligencia de Negocios).

Después de realizado el estudio se concluye que a pesar de la cantidad de servicios que brinda este sistema, no se ajusta a la investigación, pues no existen evidencias de la gestión de actividades metodológicas. Además, el sistema no es de libre acceso.

GestAcad: Sistema para la Gestión Académica

Sistema creado en Cuba por un grupo de jóvenes desarrolladores de la Universidad de Matanzas Camilo Cienfuegos en un acercamiento a la solución del problema de la gestión de la información docente en las instituciones de educación superior cubana. Su principal característica es que permite llevar el control de la academia de enseñanza o centro de estudios de manera fácil y fiable. También facilita la actualización y el procesamiento de informaciones docentes de posgrado. Sus datos se muestran en la INTRANET de la Universidad Matanzas en formato Web [25].

El sistema brinda funcionalidades que responden a la búsqueda de un alumno, proporciona listados de estudiantes por grupo, reportes dinámicos de la información existente, tablas con los resultados docentes de un grupo en un semestre y de un estudiante en toda su carrera. Además, consta de los siguientes componentes [26]:

- Módulo de Administración para la gestión de las tablas del sistema vía Web, así como agregar nuevas consultas al sitio oficial y establecer los distintos niveles de acceso a estas.
- Módulo Web para las Secretarías Docentes para la Gestión de Estudiantes que permite hasta el momento la realización de acciones generales comunes en una Secretaría Docente, así como la obtención de reportes oficiales.
- Módulo Web para los Jefes de Departamentos docentes donde se incluyen acciones relativas como la asignación de la carga docente y el control sobre los profesores del Dpto.
- Módulo Web para los Profesores donde estos pueden llevar el control docente de sus estudiantes, el control de las evaluaciones, así como reportes relativos a su carga docente.
- Sitio Web con reportes en línea con la utilidad del registro docente para los profesores además de la búsqueda de estudiantes la cual devuelve, además de algunos datos personales del estudiante, su ubicación según el horario docente detallando aula, asignatura y tipo de clases que está recibiendo además de su estado si se ha pasado asistencia en el turno de clase.

Se identificó que el sistema no cubre aún todas las esferas de la gestión universitaria necesarias para un mejor desarrollo del proceso docente-educativo. Además, no se ajusta a los intereses de la investigación, pues no existen evidencias de que se gestione alguna actividad del trabajo metodológico.

SIGENU: Sistema de Gestión de la Nueva Universidad

Es un sistema de libre acceso desarrollado en Cuba por el Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE) con el propósito de automatizar los procesos vinculados a la gestión docente de todos sus centros adscritos, el cual pretende controlar y motorizar el proceso de gestión académica a nivel nacional. Ha sido implementado en los Centros de Educación Superior del país. Permite el apoyo a la toma de decisiones acorde a los principales procesos docentes como matrícula, bajas y graduados. SIGENU está compuesto por cuatro elementos fundamentales:

- Base de Datos: donde se almacena la información del sistema.
- Servidor de Aplicaciones: actúa como intermediario entre la Base de Datos y las aplicaciones clientes y permite dar servicio a través de la red.
- Aplicación Cliente de Administración: permite la inserción y actualización de los usuarios y contiene las funcionalidades que permiten a los administradores monitorear el sistema.
- Aplicación Cliente (SIGENU): aplicación de escritorio que permite la inserción y actualización de toda la información que se registra en el sistema y la obtención de reportes.

Consta de los siguientes módulos:

- Codificadores: contiene toda la información con que debe contar el sistema y que es provista por el Ministerio de Educación Superior (MES).
- Matrícula: permite realizar el proceso de matrícula a través del cual los estudiantes pasarán a ser registrados en el sistema como estudiantes de Educación Superior.
- Control de estudiantes: permite buscar un estudiante registrado en el sistema, modificar los datos de un estudiante tanto personales como docentes, ubicar a un estudiante en un grupo o cambiarlo de grupo, realizar el pase de estudiantes a otros años de estudio y definir los que serán repitentes, así como dar baja a un estudiante del centro ya sea por licencia de matrícula, resolución o traslado.
- Plan de Estudio: permite la creación de los planes de estudio para las diferentes carreras del centro, así como realizar ajustes y modificaciones a un plan de estudio específico.

- Evaluaciones: permite registrar, modificar o eliminar las evaluaciones de los estudiantes registrados en el sistema, así como premios y bonificaciones.
- Reportes: permite obtener diversos reportes con los que se puede recuperar toda la información necesaria del sistema [27].

El sistema, a pesar de contener el módulo Sistema de Información Docente de la Educación Superior Cubana, que se encarga del proceso de toma de decisiones a nivel del ministerio, no aplica análisis sobre la información almacenada. Además, no existen evidencias de la gestión de actividades metodológicas por lo que no se ajusta a la investigación.

AKADEMOS: Sistema de Gestión de la Universidad

Es un sistema Web cubano desarrollado en la UCI. Brinda una interfaz común para todos sus usuarios. Además, realiza la gestión de toda la información referente a la formación de pregrado de un estudiante. Su principal misión es el control de procesos que intervienen en la gestión académica de un centro de estudios universitarios. Surge como respuesta a la necesidad de sustentar y dar soporte en la UCI a la labor del personal de secretaría. Asimismo, tiene como visión obtener un producto genérico, capaz de ser aplicable y adaptable en cualquier centro que implemente el control docente universitario. Los primeros módulos liberados fueron: Matrícula y Plan de Estudio, a pocos meses de haber comenzado su desarrollo y, posteriormente, se fueron incorporando los demás módulos que conforman el sistema: Expediente, Profesor, Estudiante, Control Docente y Reportes. A continuación, se nombra y describen algunos de estos módulos:

- Módulo Matrícula: permite el control de los datos de los estudiantes que van a comenzar a ser parte de la Universidad, así como la gestión de los movimientos a que estos son sometidos en su paso por la misma.
- Módulo Profesor: da la posibilidad de planificar la carga docente de los profesores pues es el encargado de asignar al mismo a un departamento y además asignarle grupos. Los profesores son la materia fundamental con que cuenta un centro de estudios para desarrollar con éxito su labor formativa.
- Módulo Estudiantes: brinda a los estudiantes, principales protagonistas del proceso docente, un espacio destinado a brindarles información referida a su desempeño académico en su paso por la universidad.

Cuenta con doce áreas de procesos identificadas (Pregrado, Posgrado, Cooperación, Residencia, Teleformación, Biblioteca, Desarrollo, Tecnologías, Investigación, Extensión, Organizaciones y

Egreso) [28]. Los avances obtenidos con su despliegue en la UCI son palpables, por ejemplo, en todo momento, el sistema está disponible para la consulta y actualización de la información, ya sea por directivos, profesores y estudiantes, lo que agilizó la gestión, y eliminó la necesidad de intermediarios entre la información y los que la generan o necesitan [29]. De igual forma, incluye la gestión de horarios docentes.

Del punto de vista metodológico el sistema solo gestiona el plan de controles a clases por áreas docentes, en este caso los directivos expresan que la asignación de controles por profesor es realizada de forma automática sin posibilidad de cambios. Los controles a clases almacenados siguen una estructura y formato estándar para todos los profesores que son controlados en la universidad, cuyo formato fue de mucha utilidad para la investigación.

Después del estudio y análisis realizado a los distintos sistemas de gestión universitaria, se concluye que resuelven parte de los problemas que existen en las universidades, ya que están concebidos para una mejor gestión, organización y control de los procesos docentes. Sin embargo, exceptuando el caso de Akademos, que solo se centra en la gestión del plan de los controles a clases, el resto de los sistemas no gestionan el trabajo metodológico. Para un mayor entendimiento del análisis de los indicadores por sistemas estudiados, se muestra Tabla 1.

Tabla 1: Análisis de los Sistemas de Gestión Universitaria (SGU). Elaboración Propia

Indicadores SGU	Universidad que lo utiliza	Pregrado o posgrado	Gestión de TM	Existe análisis de datos almacenados	Open Source
SIU-Guaraní	Universidad de Córdoba	ambos	No	No	Si
SIGA	Universidad de Piura	ambos	No	No	Si
openSIS	Universidad de Zúrich	ambos	No	SI	Si
SIGMA	Universidad de Zaragoza	ambos	No	No	Si
GestAcad	Universidad de Matanzas	ambos	No	No	Si
SIGENU	CUJAE	ambos	No	No	Si
AKADEMOS	UCI	ambos	Si (Controles a clases)	No	No

1.3 Tecnologías y herramientas

A continuación, se describen las tecnologías, herramientas y metodología utilizadas para el análisis, diseño e implementación de la propuesta de solución.

1.3.1 Lenguajes de Programación

Python 3.5

Lenguaje interpretado que tiene una sintaxis simple, clara y sencilla. Python es de tipado dinámico, pues no es necesario declarar el tipo de datos que va a contener una determinada variable. Para el desarrollo de la aplicación se utilizó Python debido a las facilidades que proporciona. Además, es multiplataforma y tiene gran cantidad de bibliotecas disponibles [30]. Se utilizan las siguientes bibliotecas:

- Pillow: es una biblioteca gratuita para el lenguaje de programación Python que agrega soporte para abrir, manipular y guardar muchos formatos de archivos de imágenes diferentes. Está disponible para Windows, Mac OS X y Linux.
- Rest framework: es una aplicación Django que permite construir proyectos software bajo la arquitectura REST (por sus siglas en inglés Transferencia de Estado Representacional). Esta se emplea, cuando los servicios web utilizan interfaces de programación de aplicaciones (API). REST es una arquitectura de software para sistemas distribuidos. Los recursos de un servicio REST pueden ser considerados como los sustantivos de esta arquitectura, para operar sobre ellos se tienen los comandos HTTP: GET, PUT, POST y DELETE.
- Model utils: es una aplicación que nos facilita el trabajo con modelos en Django. Se utiliza para manejar modelos de tiempo, estado, comportamiento, opciones, así como clases abstractas. Soporta las versiones de Django desde 1.8 y Python 2.7, 3.3, 3.4, 3.5.
- Pip: es un administrador de paquetes de Python. Es útil en la instalación de las bibliotecas a utilizar y del propio Django [31].

HTML 5

HTML 5 es un lenguaje de marca usado para estructurar y presentar el contenido para la web. Es una nueva versión con elementos, atributos, comportamientos y un conjunto amplio de tecnologías que permiten crear sitios web y diversas aplicaciones de gran alcance. [32]. En el presente trabajo se usa para la creación de las interfaces web que son mostradas al usuario.

CSS 3

Las hojas de estilo en cascada (CSS por sus siglas en inglés) son las que ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web. Permiten definir de manera eficiente la representación de las páginas, además es

uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo [32].

JavaScript

JavaScript es un lenguaje que puede ser utilizado por profesionales y por quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, pues los navegadores son los encargados de interpretar estos códigos. Es utilizado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador web, así como para validar datos de entrada en las interfaces de las aplicaciones [32]. La biblioteca utilizada de JavaScript en este trabajo es:

- **jQuery:** biblioteca de JavaScript en su versión 1.8.2 que permite programar páginas dinámicas compatibles con todos los navegadores. Facilita el desarrollo de aplicaciones enriquecidas del lado del cliente, en JavaScript, compatible con todos los navegadores [33].

1.3.2 Marco de trabajo

Un marco o ambiente de trabajo está orientado a la reutilización de componentes permitiendo el desarrollo rápido de aplicaciones. Los marcos de trabajo a emplear para la implementación del sistema son: Django 2.0.3 y Materialize 1.0.0.

Django 2.0.3

Es un marco de trabajo web sencillo que se empezó a desarrollar en el año 2002 y fue publicado en el año 2005. Fue escrito en el lenguaje de programación Python. Este fomenta el desarrollo rápido y diseño limpio. Permite construir aplicaciones web con menos código [34]. Entre sus principales ventajas están:

- La utilización de una modificación de la arquitectura Modelo-Vista-Controlador (MVC) que se nombra Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático.
- Se basa en la rapidez, la calidad, la sostenibilidad y sobre todo la sencillez.
- Permite la reutilización del código.

Materialize 1.0.0

Es un marco de trabajo web moderno y adaptativo basado en Material Design, este último es un lenguaje de diseño creado por Google que combina el principio clásico de un diseño exitoso a través de la innovación y las tecnologías. Además, permite crear interfaces agradables [35].

1.3.3 Entorno de Desarrollo Integrado

Pycharm 2017.2.3

Es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Posee un elevado número de ventajas como son: el autocompletado, herramienta de análisis y refactorización, resaltador de sintaxis para código Python, HTML, CSS, JavaScript, así como para las plantillas de Django [36].

1.3.4 Gestor de Base de datos

SQLite Admin

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID (transacciones de base de datos atómicas, consistencia de base de datos, aislamiento, y durabilidad), contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público [37]. El Administrador SQLite es una herramienta poderosa para crear, diseñar o administrar fácilmente archivos de base de datos SQLite. El editor de código SQL le ayuda a escribir rápidamente consultas SQL con características tales como la finalización y el resaltado del código. Ambas versiones principales de los archivos de base de datos SQLite son compatibles con el Administrador de SQLite [38]. Python incluye soporte para SQLite nativamente desde la versión 2.5 incorporado en la Biblioteca Estándar.

1.3.5 Herramienta de modelado

Visual Paradigm para UML 8.0

Es una herramienta profesional que soporta el ciclo completo del desarrollo de software: análisis y diseño orientados a objetos, construcción y despliegue a través de la representación de varios tipos de diagramas. Es una herramienta de modelado, multiplataforma y cuenta con una versión libre para la comunidad. Genera la documentación del sistema en los formatos PDF, HTML y el formato de documentos de Microsoft Word [39].

UML 2.0

Lenguaje de Modelado Unificado (UML por sus siglas en inglés), es el lenguaje de modelado más conocido y utilizado en la actualidad para especificar o para describir métodos o procesos. Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio y otros sistemas. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes

de programación, esquemas de bases de datos y componentes reutilizables. Diseñado para aplicarse con todos los métodos de desarrollo, etapas de ciclo de vida, dominios de aplicación y medios. Se puede emplear en el desarrollo de software brindando gran variedad de formas para dar soporte a una metodología de desarrollo, pero no especifica en sí mismo que metodología o proceso usar [40].

1.4 Metodología de desarrollo

El desarrollo de software es un proceso complejo, que puede tomar varias formas y estrategias. En el transcurso del mismo existen patrones, modalidades y sistemas para obtener los resultados de una fase o iteración y guiar el proceso de obtención de un artefacto durante la construcción del producto final.

Este proceso es una de las tareas más difíciles de la ingeniería; prueba de ello es la existencia de numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo [41]. Las metodologías de desarrollo se pueden enmarcar en dos grandes grupos, las tradicionales y las ágiles. Las tradicionales enfatizan en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto. Son recomendadas para proyectos de grandes dimensiones y con grandes equipos de desarrollo. Las metodologías ágiles dan importancia a la capacidad de respuesta a los cambios, se enfatiza en la satisfacción del cliente y promueven el trabajo en equipo [42].

La Figura 1 recoge esquemáticamente las principales diferencias de las metodologías ágiles con respecto a las tradicionales [41].

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Figura 1: Diferencias entre las metodologías ágiles y las tradicionales. Elaboración Propia

Una buena elección de la metodología garantiza la calidad del producto y el buen desarrollo del proceso del software. Luego de la evaluación de las metodologías de desarrollo y el estudio de sus características, se determina utilizar una metodología ágil. Como una alternativa acertada de este tipo de metodología, se identificó a la Programación Extrema (XP). Se asume esta última por ser una metodología flexible ante los cambios de requisitos, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, formada por un pequeño equipo de desarrollo donde todas las personas tienen algo que ver con el proyecto. Gracias a la filosofía del “pair programming” (programación en parejas) se consigue que los desarrolladores apliquen las buenas prácticas, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios [43].

La Figura 2 muestra el desarrollo del proyecto basado en la metodología XP. Donde se evidencia las diferentes etapas y artefactos que se generan durante la evolución del mismo.

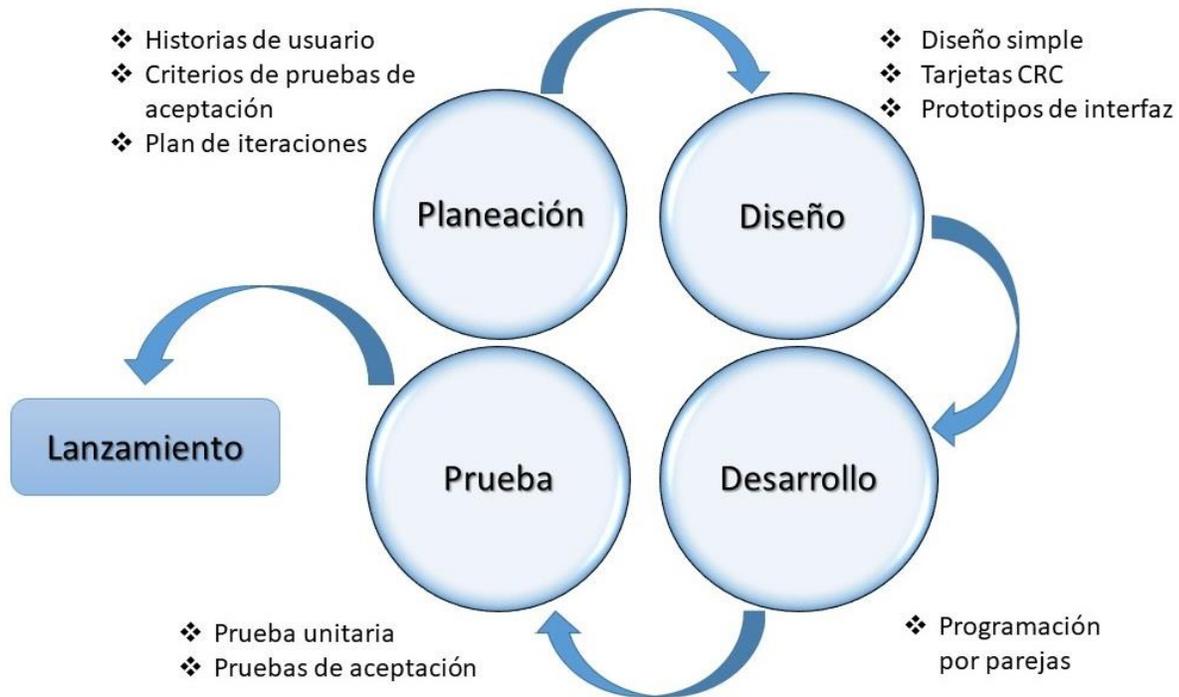


Figura 2: Desarrollo del proyecto basado en XP. Elaboración Propia

XP consta de cuatro fases para el desarrollo [44] [45]:

1. **Planificación:** es la fase donde se realiza una recopilación de todos los requerimientos del proyecto, se crean las Historias de Usuarios (HU), el plan de iteraciones y el plan de entregas.
2. **Diseño:** la metodología XP hace especial énfasis en los diseños simples y sencillos, con el objetivo de procurar hacerlo todo lo menos complicado posible para el cliente o usuario, se crean las tarjetas CRC (Clase-Responsabilidad-Colaboración), las cuales definen una clase expresando sus funcionalidades y las otras clases con las que colabora.
3. **Desarrollo:** esta fase consiste en establecer una buena comunicación entre el equipo y el cliente, para que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere.
4. **Pruebas:** se basa en la comprobación del funcionamiento de la codificación que se halla implementado, garantizando la evaluación de las distintas tareas en las que ha sido divididas las HU de manera que si se encuentran errores darles solución para prever que no vuelvan a ocurrir.

1.5 Conclusiones parciales

Como resultado del estudio del capítulo se arribó a las siguientes conclusiones parciales:

- Los sistemas de gestión universitaria estudiados no cumplen con todas las características necesarias para el desarrollo de la propuesta. Sin embargo, el estudio de los sistemas aportó significativamente a la investigación ya que, a partir de la estructuración, el formato y la manera de guardar los controles a clases de AKADEMOS se generaron las demás actividades metodológicas.
- El estudio de las metodologías y tecnologías utilizadas en el desarrollo de software, permitió ampliar los conocimientos para su posterior uso.

De esta forma se demuestra que, para dar solución al problema planteado, es preciso desarrollar una herramienta informática para la gestión del trabajo metodológico que contribuya a la obtención de la información organizada, estructurada y disponible. Dando cumplimiento así, al primer objetivo específico definido en la investigación.

Capítulo 2: Herramienta para la gestión del plan de trabajo metodológico (SiGPTM)

En este capítulo se describen los elementos esenciales que componen la propuesta de solución. Se detalla la visión general de la herramienta para la gestión del plan de trabajo metodológico (SiGPTM). Siguiendo la metodología XP se describen las fases de desarrollo, especificando descripción de requisitos funcionales y no funcionales, el modelo de datos, la arquitectura de desarrollo y los patrones de diseño empleados como artefactos fundamentales.

2.1 Descripción del sistema

A partir de las etapas analizadas anteriormente para el desarrollo de un sistema de gestión, se formuló la propuesta de solución. Esta consta de cuatro componentes fundamentales: Planificación, Ejecución, Control y Análisis. Mediante estos componentes el sistema permitirá contribuir con la organización del trabajo metodológico del área, pues los directivos docentes de la facultad podrán tener en un mismo formato la documentación y materiales asociados. Además, ayudará a la estructuración de las actas de las actividades metodológicas, informes de controles a clases, así como otras evidencias. Permitirá también que estas se encuentren disponibles para ser consultadas, por todo el que tenga acceso, en cualquier momento. La adecuada gestión de estas funciones, a través del sistema, ayudará en el desarrollo del proceso docente educativo. En la Figura 3 se muestra la descripción del sistema, luego se detallan los componentes como parte del proceso de gestión del PTM.

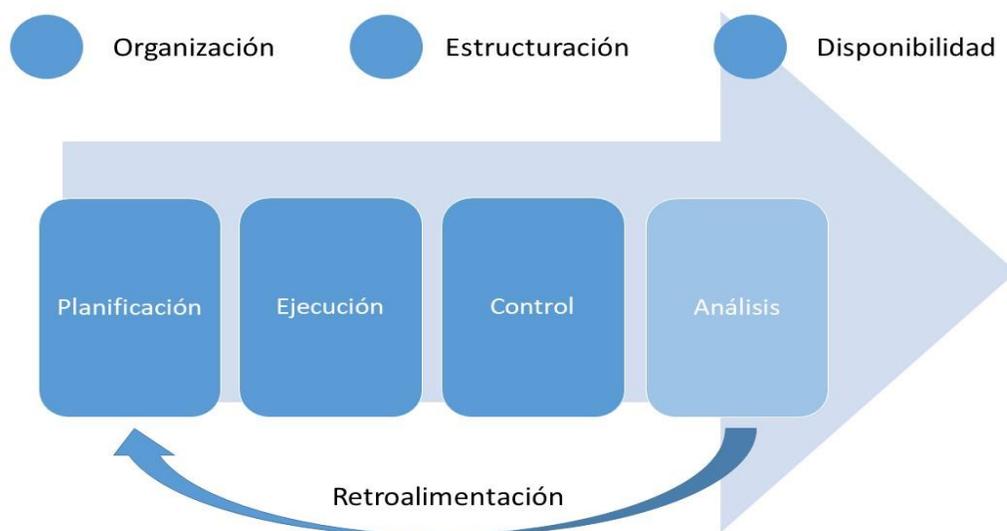


Figura 3: Descripción del sistema. Elaboración Propia

Para la gestión del PTM en la facultad existen tres estructuras organizativas:

- Un vicedecanato de formación de la facultad dirigido por el vicedecano de formación.
- Cuatro departamentos docentes (Ciencias Básicas, Técnica de Programación, Ciencias Sociales y Humanísticas e Ingeniería de Software) dirigido cada uno por un jefe de departamento docente.
- Cuatro colectivos de años académicos (segundo año, tercer año, cuarto año y quinto año) dirigido cada uno por un profesor principal de años académicos.

La gestión del PTM se realiza siguiendo un flujo de información ascendente y descendente. El vicedecano de formación será el encargado de iniciar y cerrar el proceso para cada curso académico. Para ello debe realizar la configuración inicial del sistema, la cual consiste en habilitar el sistema introduciendo el curso académico correspondiente, las asignaturas por semestre y departamento docente, los problemas metodológicos y las líneas de trabajo metodológico con las cuales se trabajarán en la facultad. Una vez realizada esta acción en el sistema el resto de los directivos de las restantes estructuras organizativas podrán comenzar la gestión de su plan de trabajo metodológico.

Componente de Planificación

En la planificación cada estructura organizativa declara (del PTM) la fundamentación, caracterización del claustro de profesores, los problemas metodológicos, indicadores metodológicos para evaluar la planificación y las líneas de trabajo metodológico con las que se trabajará. Estas últimas son definidas en el sistema inicialmente por el vicedecano de formación, donde cada departamento y año académico las asume, y pueden además agregar alguna si lo considera necesario. Durante la planificación también se establece el sistema de actividades metodológicas que pueden estar asociadas a una línea o no. Estas acciones son realizadas por todos los jefes de estructuras. Además, se crea el plan de controles a clases donde se especifican las semanas y la cantidad de controles que se le realizarán al profesor durante el semestre. Este será planificado solo por el jefe de departamento. La planificación realizada puede ser actualizada al finalizar el primer semestre. Como salida se obtiene el PTM por cada una de las estructuras y en el caso de los departamentos docentes el plan de controles a clases por semestre. A partir del desarrollo de este componente se contribuye a la organización de los datos.

Componente de Ejecución

El componente de ejecución, recibe como entradas el PTM y el plan de controles a clases que constituyen las salidas de la planificación. El usuario notificará al sistema que se ejecutó una actividad cuando se crea el acta y se le asocia un conjunto de materiales. Para garantizar la estructuración de la información, el acta sigue una estructura estándar que consta de los metadatos

del profesor que imparte o dirige y de la actividad metodológica, los principales planteamientos del debate y acuerdos. Los materiales se guardan en la base de datos, para permitir la recuperación de ellos en cualquier momento como un fichero único de hasta cuatro mega byte. Esta funcionalidad puede ser realizada por cualquier jefe de estructura. También, se notificará al sistema la ejecución de un control a clases cuando se crea un informe. Para asegurar la estructuración de la información, el mismo está conformado por los metadatos asociados al profesor visitado, metadatos de los profesores controladores, metadatos al control, principales logros, señalamientos y recomendaciones. Esta funcionalidad será ejecutada por los jefes de departamentos docentes. Ambos documentos pueden ser generados en formato PDF. Como salida de este componente se tienen almacenados en la base datos las evidencias del cumplimiento del PTM y el plan de controles a clases.

Componente de Control

En el control se tiene como entradas, las salidas de los componentes planificación y ejecución, mediante estas se conoce lo planificado y lo ejecutado hasta el momento. El sistema notifica mensualmente a los jefes de estructura el cumplimiento de las actividades metodológicas planificadas a través de reportes estadísticos o reportes de seguimiento. Además, el sistema notifica a los jefes de departamentos y al vicedecano de formación los reportes estadísticos del cumplimiento de los controles y solamente al jefe de departamento los reportes de seguimiento de los controles. Se alerta de forma semanal cuando se incumple o se acerca una actividad metodológica o un control a clases.

Componente de Análisis

En el análisis se tiene como entrada la salida del componente ejecución para realizar búsquedas y reportes estadísticos por curso, semestre y estructura organizativa. Esto permite la disponibilidad de la información. Una vez que la información esté organizada, estructurada y disponible desde este componente se pueden aplicar técnicas de minería de texto para futuros análisis, que faciliten mejores decisiones sobre el proceso docente metodológico. Concluida esta etapa se contribuirá a la retroalimentación del proceso inicial para un nuevo ciclo de planificación.

2.2 Desarrollo de la herramienta

Para llevar a cabo el desarrollo de la herramienta SiGPTM, se siguen las cuatro fases de la metodología XP, en este capítulo se describen tres de ellas (Planificación, Diseño y Desarrollo). En la fase planificación se realiza el levantamiento de los requisitos, identificando los requerimientos funcionales y no funcionales con los que debe cumplir el sistema. Estos requisitos son especificados mediante las historias de usuarios. A partir de estas se genera el plan de iteraciones y el plan de entrega que determinan la duración estimada del desarrollo de la

herramienta. Además, se definen las funcionalidades de cada uno de los roles en el sistema. En la fase de diseño se presenta el modelo de base de datos, la aplicación del patrón arquitectónico Modelo-Plantilla-Vista, así como los Patrones de Software de Asignación de Responsabilidad General (GRASP) y los conocidos como la Pandilla de los Cuatro (GoF), se genera el conjunto de tarjetas Clase-Responsabilidad-Colaborador (CRC), y finalmente se aplican las métricas Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC) para validar el diseño. En la fase desarrollo se muestra el estándar de codificación implementado para el desarrollo de la propuesta de solución.

2.3 Fase de Planificación

La metodología XP define como fase inicial del desarrollo de software la planificación. A continuación, se describen los principales artefactos que se generan durante esta etapa en el desarrollo de la herramienta SiGPTM.

2.3.1 Especificación de requisitos del sistema

El flujo de trabajo de requerimientos es uno de los más importantes, pues en este se establece qué debe hacer exactamente el sistema que se construye [46]. Para el levantamiento de los requisitos del sistema se realizó una entrevista abierta entre los desarrolladores y el cliente. En ella se identificaron los requisitos funcionales y no funcionales, Además se especificó el nivel de prioridad de cada uno de ellos. Para la validación de estos requisitos se presentaron al cliente los prototipos de interfaces de la aplicación para su aprobación. Las funcionalidades del sistema serán ejecutadas dependiendo del rol que desempeñe el usuario autenticado. Por lo que fue necesario especificar los roles que se desempeñarán en el sistema y las funcionalidades asociadas a cada uno. Luego se definen las historias de usuario por iteraciones y se especifica el tiempo de duración de cada una.

2.3.1.1 Roles del sistema

Con el objetivo de restringir los permisos a cada una de las vistas que presenta el sistema, en la Tabla 2 se definen los siguientes roles:

Tabla 2: Usuarios del sistema. Elaboración Propia

Roles	Descripción
Profesor (P)	El profesor tiene acceso a la visualización del PTM, así como a la búsqueda de los materiales almacenados en las actividades metodológicas.
Profesor Principal de Año Académico (PPAA)	El profesor principal de año accede a la gestión de las actividades metodológica, visualiza el PTM y consulta los reportes que le son necesarios para su trabajo.
Jefe de Departamento(JDD)	El jefe de departamento es el encargado de planificar el PTM de su área. Realiza el plan de controles a clases y actualiza su cumplimiento. Puede generar y buscar las evidencias asociadas a las actividades metodológicas y consultar reportes para el seguimiento y control de la ejecución de estas actividades.
Vicedecano de Formación (VDF)	El vicedecano de formación tiene acceso a gestionar el PTM de la facultad, visualizar todos los PTM generados por las distintas áreas y realizar los reportes de las actividades.
Administrador	Encargado de administrar los roles.

2.3.1.2 Requisitos funcionales

Los requisitos funcionales son condiciones o capacidades que se necesita para resolver un problema y lograr un objetivo determinado desde el punto de vista del usuario. En las siguientes tablas, se representan los requisitos funcionales obtenido luego de aplicada la entrevista. Los mismos se encuentran dividido por los componentes del sistema. También se especifican el acceso de los diferentes roles a las funcionalidades del sistema.

Tabla 3.Requisitos funcionales. Componente Ejecución. Elaboración Propia

Requisitos funcionales	Acceso				
	P	PPAA	JDD	VDF	ADMIN
COMPONENTE EJECUCIÓN					
RF31-Crear PTM por estructura	no	no	si	si	no
RF32-Modificar PTM	no	si	si	si	no
RF33-Visualizar el PTM en formato PDF	si	no	si	si	no
RF34-Planificar controles a clases por profesor	no	no	si	no	no
RF35-Crear el plan de controles a clases por departamento	no	no	si	no	no
RF36-Modificar plan de controles a clases	no	no	si	no	no
RF37-Visualizar la planificación de controles a clases en formato PDF	no	no	si	no	no
RF38-Crear acta de actividad metodológica	no	si	si	si	no
RF39-Modificar acta de actividad metodológica	no	si	si	si	no
RF40-Adjuntar materiales	no	si	si	si	no
RF41-Visualizar acta de actividad metodológica en formato PDF	no	si	si	si	no
RF42-Crear informe de control a clase	no	no	si	no	no
RF43-Modificar informe de control a clase	no	no	si	no	no
RF44-Visualizar informe de control a clase en formato PDF	no	no	si	no	no

Tabla 4.Requisitos funcionales. Componente Planificación. Elaboración Propia

Requisitos funcionales	Acceso				
	P	PPAA	JDD	VDF	ADMIN
COMPONENTE PLANIFICACIÓN					
RF1-Autenticar usuario	si	si	si	si	si
RF2-Asignar roles	no	no	no	no	si
RF3-Quitar asignación de roles	no	no	no	no	si
RF4-Adicionar curso académico	no	no	no	si	no
RF5-Modificar curso académico	no	no	no	si	no
RF6-Adicionar profesor	no	no	si	no	no
RF7-Modificar profesor	no	no	si	no	no
RF8-Eliminar profesor	no	no	si	no	no
RF9-Adicionar asignatura	no	no	no	si	no
RF10-Modificar asignatura	no	no	no	si	no
RF11-Eliminar asignatura	no	no	no	si	no
RF12-Adicionar línea de trabajo metodológico	no	si	si	si	no
RF13-Modificar línea de trabajo metodológico	no	si	si	si	no
RF14-Eliminar línea de trabajo metodológico	no	si	si	si	no
RF15-Agregar objetivo de línea de trabajo metodológico	no	si	si	si	no
RF16-Adicionar actividad metodológica	no	si	si	si	no
RF17-Modificar actividad metodológica	no	si	si	si	no
RF18-Eliminar actividad metodológica	no	si	si	si	no
RF29-Adicionar otras actividades metodológicas	no	si	si	si	no
RF20-Modificar otras actividades metodológicas	no	si	si	si	no
RF21-Eliminar otras actividades metodológicas	no	si	si	si	no
RF22-Adicionar problemas metodológicos	no	si	si	si	no
RF23-Modificar problemas metodológicos	no	si	si	si	no
RF24-Eliminar problemas metodológicos	no	si	si	si	no
RF25-Adicionar indicadores metodológicos	no	si	si	si	no
RF26-Modificar indicadores metodológicos	no	si	si	si	no
RF27-Eliminar indicadores metodológicos	no	si	si	si	no
RF28-Adicionar orientaciones de los controles a clases	no	no	si	si	no
RF29-Modificar orientaciones de los controles a clases	no	no	si	si	no
RF30-Eliminar orientaciones de los controles a clases	no	no	si	si	no

Tabla 5.Requisitos funcionales. Componente Control. Elaboración Propia

Requisitos funcionales	Acceso				
	P	PPAA	JDD	VDF	ADMIN
COMPONENTE CONTROL					
RF45- Generar reporte del cumplimiento mensual del PTM de la facultad	no	no	si	si	no
RF46-- Generar reporte del cumplimiento mensual del PTM del departamento	no	no	si	si	no
RF47-Generar reporte del cumplimiento mensual del PTM de los años académicos	no	si	no	si	no
RF48-Generar reporte mensual de los resultados de los controles a clase por facultad	no	no	no	si	no
RF49-Generar reporte mensual de los controles planificados por categoría docente en la facultad	no	no	no	si	no
RF50-Generar reporte mensual de los controles realizados por categoría docente en los departamentos	no	no	si	si	no
RF51-Generar reporte mensual de los controles planificados por categoría docente en los departamentos	no	no	si	si	no
RF52-Generar reporte mensual de los controles realizados por categoría docente en los departamentos	no	no	si	si	no
RF53-Generar reporte mensual de los resultados de los controles a clase por asignaturas/departamentos	no	no	si	si	no
RF54-Generar reporte mensual de los resultados de los controles a clase por tipología de clase	no	no	si	si	no
RF55-Generar reporte del seguimiento del PTM por actividades metodológicas	no	si	si	si	no
RF56-Generar reporte del seguimiento del PTM por controles a clase	no	no	si	si	no

Tabla 6.Requisitos funcionales. Componente Análisis. Elaboración Propia

Requisitos funcionales	Acceso				
	P	PPAA	JDD	VDF	ADMIN
COMPONENTE ANÁLISIS					
RF57-Generar reporte estadístico del cumplimiento del PTM por curso académico	no	si	si	si	no
RF58-Generar reporte estadístico del cumplimiento del PTM por semestre	no	si	si	si	no
RF59-Generar reporte estadístico del cumplimiento del PTM del por curso académico	no	no	si	si	no
RF60-Generar reporte estadístico del cumplimiento del PTM del por semestre	no	no	si	si	no
RF61-Generar reporte estadístico del cumplimiento del PTM de los años académicos por curso académico	no	si	no	si	no
RF62-Generar reporte estadístico del cumplimiento del PTM de los años académicos por semestre	no	si	no	si	no
RF63-Generar reporte estadístico de los resultados de los controles a clase por año académico	no	no	no	si	no
RF64-Generar reporte estadístico de los resultados de los controles a clase por año semestre	no	no	no	si	no
RF65-Generar reporte estadístico de los controles planificados por categoría docente por curso académico	no	no	no	si	no

RF66-Generar reporte estadístico de los controles planificados por categoría docente por curso semestre	no	no	no	si	no
RF67-Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por curso académico	no	no	si	si	no
RF68-Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por semestre	no	no	si	si	no
RF69-Generar reporte estadístico de los controles planificados por categoría docente en los departamentos por curso académico	no	no	si	si	no
RF70-Generar reporte estadístico de los controles planificados por categoría docente en los departamentos por curso académico	no	no	si	si	no
RF71-Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por curso académico	no	no	si	si	no
RF72-Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por semestre	no	no	si	si	no
RF73-Generar reporte estadístico de los resultados de los controles a clase por asignaturas/departamentos por curso académico	no	no	si	si	no
RF74-Generar reporte estadístico de los resultados de los controles a clase por asignaturas/departamentos por semestre	no	no	si	si	no
RF75- Generar reporte estadístico de los resultados de los controles a clase por tipología de clase por curso académico	no	no	si	si	no
RF76-Generar reporte estadístico de los resultados de los controles a clase por tipología de clase por semestre	no	no	si	si	no
RF77- Generar reporte estadístico del PTM por actividades metodológicas por curso académico	no	si	si	si	no
RF78-Generar reporte estadístico del PTM por actividades metodológicas por semestre	no	si	si	si	no
RF79- Generar reporte del estadístico del PTM por controles a clase por curso académico	no	no	si	si	no
RF80-Generar reporte del estadístico del PTM por controles a clase por semestre	no	no	si	si	no
RF81-Buscar materiales metodológicos	si	si	si	si	no

2.3.1.3 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son esas características que posibilitan que el producto sea atractivo, usable, rápido y confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto.

Requisitos de Usabilidad

RNF 1- La herramienta debe contar con una interfaz amigable y de fácil entendimiento.

Requisitos de Disponibilidad

RNF 2- La aplicación deberá estar disponible siempre, asegurando el acceso desde cualquier lugar de red a los usuarios autorizados.

Requisitos de Software

RNF 3- Para un funcionamiento óptimo del sistema, se recomienda el uso del navegador Mozilla Firefox en una versión superior a la 30.0.

RNF 4- La herramienta informática a desarrollar será una aplicación web.

Requisitos de Hardware

Servidor:

RNF 5- La memoria RAM debe ser igual o superior a 2 GB

RNF 6-El disco duro debe tener una capacidad de almacenamiento igual o superior a 250 Gb

RNF 7-El procesador debe de tener una frecuencia igual o superior a 3.00 GHz

Requisitos de Seguridad

RNF 8- El sistema debe garantizar el control de acceso a través de la autenticación de usuarios para la administración del mismo.

2.3.2 Historias de usuario

Las historias de usuario (HU) son la técnica utilizada en XP para especificar los requisitos del software. En estas, el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [47]. Se identificaron un total de 81 HU, de ellas 30 de prioridad Alta y 14 de prioridad media y 37 de prioridad baja. A continuación, se muestra HU correspondiente al requisito funcional Crear acta de actividad metodológica:

Tabla 7.Historia de usuario: Crear acta de actividad metodológica. Elaboración Propia

Historia de Usuario	
Número: 38	Nombre del requisito: Crear acta de actividad metodológica
Programador: Darlyn Díaz Morales	Iteración Asignada: 2
Prioridad: Media	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real: 4 días
Descripción: Permite crear el acta de la actividad metodológica.	
Observaciones:	

Prototipo elemental de interfaz gráfica de usuario:

Acta de actividad Metodológica

Departamento	Fecha	
<input type="text"/>	<input type="text"/>	
Profesor que Imparte/Dirige		
<input type="text"/>		
Categoría Docente	Categoría Científica	Cargo
<input type="text"/>	<input type="text"/>	<input type="text"/>
Tipo de Actividad Metodológica	Título	
<input type="text"/>	<input type="text"/>	
Objetivos		
<input type="text"/>		
Principales Planteamientos		
<input type="text"/>		
Acuerdos		
<input type="text"/>		

2.3.3 Plan de iteraciones

Para lograr una mayor organización del trabajo se crea el plan de iteraciones. Este tiene como objetivo mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuarios [48].

En la Tabla 8 se muestra el plan de iteraciones propuesto para el desarrollo del sistema. El mismo está compuesto por 3 iteraciones en correspondencia con cada uno de los componentes antes expuestos. Donde en una primera iteración se implementaron las historias de usuarios que tienen prioridad alta en el componente de planificación, representando estas las de mayor relevancia para el avance de la aplicación. En la segunda iteración se toman las historias de usuarios de prioridad media que hacen alusión a la ejecución y en la tercera la perteneciente al componente de control y análisis con prioridad baja.

Tabla 8. Plan de iteraciones. Elaboración Propia

Iteración	HU	Prioridad	Duración de la HU (días)	Duración Total (semanas)
1 (Planificación)	Autenticar usuario	Media	3	8 semanas (56 días)
	Asignar roles	Alta	2	
	Quitar asignación de roles	Alta	2	
	Adicionar profesor	Alta	3	
	Modificar profesor	Alta	2	
	Eliminar profesor	Alta	1	
	Adicionar asignatura	Alta	3	
	Modificar asignatura	Alta	2	
	Eliminar asignatura	Alta	1	
	Adicionar línea de trabajo metodológico	Alta	3	
	Modificar línea de trabajo metodológico	Alta	2	
	Eliminar línea de trabajo metodológico	Alta	1	
	Agregar objetivo de línea de trabajo metodológico	Alta	2	
	Adicionar actividad metodológica	Alta	3	
	Modificar actividad metodológica	Alta	2	
	Eliminar actividad metodológica	Alta	1	
	Adicionar otras actividades	Alta	3	
	Modificar otras actividades	Alta	2	
	Eliminar otras actividades	Alta	1	
	Adicionar problemas metodológicos	Alta	3	
	Modificar problemas metodológicos	Alta	2	
	Eliminar problemas metodológicos	Alta	1	
	Adicionar indicadores metodológicos	Alta	3	
	Modificar indicadores metodológicos	Alta	2	
	Eliminar indicadores metodológicos	Alta	1	
Adicionar orientaciones de los controles a clases	Alta	3		
Modificar orientaciones de los controles a clases	Alta	2		
Eliminar orientaciones de los controles a clases	Alta	1		
2da (Ejecución)	Crear PTM por estructura	Media	5	6 Semanas (40 días)
	Modificar PTM	Media	3	
	Visualizar el PTM en formato PDF	Media	3	
	Planificar controles a clases por profesor	Media	4	
	Crear el plan de controles a clases por departamento	Media	3	
	Modificar plan de controles a clases			
	Visualizar la planificación de controles a clases en formato PDF	Media	3	
	Crear acta de actividad metodológica	Media	4	
	Modificar acta de actividad metodológica	Media	4	
	Adjuntar materiales	Media	2	
	Visualizar acta de actividad metodológica en formato PDF	Media	3	
	Crear informe de control a clase	Media	3	
	Modificar informe de control a clase	Media	2	

	Visualizar informe de control a clase en formato PDF			
3 (Control y Análisis)	Generar reporte del cumplimiento mensual del PTM de la facultad	Baja	1	5 semana (37 días)
	Generar reporte del cumplimiento mensual del PTM del departamento	Baja	1	
	Generar reporte del cumplimiento mensual del PTM de los años académicos	Baja	1	
	Generar reporte mensual de los resultados de los controles a clase por facultad	Baja	1	
	Generar reporte mensual de los controles planificados por categoría docente en la facultad	Baja	1	
	Generar reporte mensual de los controles realizados por categoría docente en los departamentos	Baja	1	
	Generar reporte mensual de los controles planificados por categoría docente en los departamentos	Baja	1	
	Generar reporte mensual de los controles realizados por categoría docente en los departamentos	Baja	1	
	Generar reporte mensual de los resultados de los controles a clase por asignaturas/departamentos	Baja	1	
	Generar reporte mensual de los resultados de los controles a clase por tipología de clase	Baja	1	
	Generar reporte del seguimiento del PTM por actividades metodológicas	Baja	1	
	Generar reporte del cumplimiento del PTM por curso académico	Baja	1	
	Generar reporte estadístico del cumplimiento del PTM por curso académico	Baja	1	
	Generar reporte estadístico del cumplimiento del PTM por semestre	Baja	1	
	Generar reporte estadístico del cumplimiento del PTM del por curso académico	Baja	1	
	Generar reporte estadístico del cumplimiento del PTM del por semestre	Baja	1	
	Generar reporte estadístico del cumplimiento del PTM de los años académicos por curso académico	Baja	1	
	Generar reporte estadístico del cumplimiento del PTM de los año académicos por semestre	Baja	1	
	Generar reporte estadístico de los resultados de los controles a clase por año académico	Baja	1	
	Generar reporte estadístico de los resultados de los controles a clase por año semestre	Baja	1	
	Generar reporte estadístico de los controles planificados por categoría docente por curso académico	Baja	1	
	Generar reporte estadístico de los controles planificados por categoría docente por curso semestre	Baja	1	
Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por curso académico	Baja	1		

Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por semestre	Baja	1
Generar reporte estadístico de los controles planificados por categoría docente en los departamentos por curso académico	Baja	1
Generar reporte estadístico de los controles planificados por categoría docente en los departamentos por curso académico	Baja	1
Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por curso académico	Baja	1
Generar reporte estadístico de los controles realizados por categoría docente en los departamentos por semestre	Baja	1
Generar reporte estadístico de los resultados de los controles a clase por asignaturas/departamentos por curso académico	Baja	1
Generar reporte estadístico de los resultados de los controles a clase por asignaturas/departamentos por semestre	Baja	1
Generar reporte estadístico de los resultados de los controles a clase por tipología de clase por curso académico	Baja	1
Generar reporte estadístico de los resultados de los controles a clase por tipología de clase por semestre	Baja	1
Generar reporte estadístico del PTM por actividades metodológicas por curso académico	Baja	1
Generar reporte estadístico del PTM por actividades metodológicas por semestre	Baja	1
Generar reporte del estadístico del PTM por controles a clase por curso académico	Baja	1
Generar reporte del estadístico del PTM por controles a clase por semestre	Baja	1
Buscar material metodológico	Baja	1

2.3.4 Plan de entrega

El plan de entrega se encarga de definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una de ellas como se aprecia en la Tabla 9.

Tabla 9. Plan de entrega. Elaboración Propia

	Iteración 1	Iteración 2	Iteración 3
Cantidad de HU	30	14	37
Fecha de entrega	30/3/2018	16/5/2018	11/6/2018

2.4.1.2 Patrones utilizados en el diseño del modelo de datos

Para el diseño del modelo de datos se utiliza el patrón llaves subrogadas el cual plantea: que se genere una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. Una clave subrogada es un identificador único que se asigna a cada registro de una tabla. Esta clave, generalmente, no tiene ningún sentido específico de negocio. Permite que las tablas sean más fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas. La utilización de este patrón se evidencia al asignarle a cada entidad un "id" para su identificación. Se puede observar un ejemplo en la Figura 5 correspondiente a las entidades *acta_actividad_metodologica*, *actividad_metodologica_linea* y *rol* con *idActa*, *idActividadMetodologica* y *idRol* respectivamente.

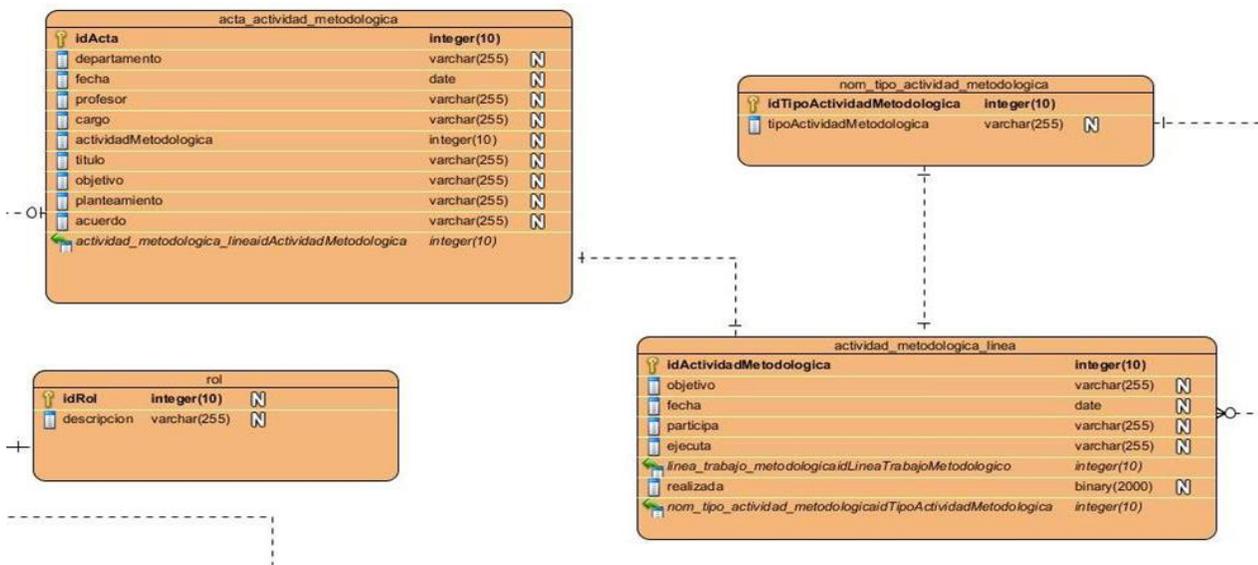


Figura 5: Ejemplo del patrón llave subrogada. Elaboración Propia

2.4.2 Tarjetas CRC

Las tarjetas CRC (Class-Responsibility-Collaboration) son una técnica de diseño orientado a objetos. El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objeto de que el diseño sea lo más simple posible verificando las especificaciones del sistema [49]. Se obtuvo un total de 23 tarjetas CRC. A continuación, se muestra la tarjeta correspondiente a la clase *acta_actividad_metodologica*:

Tabla 10. Tarjeta CRC de la clase acta_actividad_metodologica. Elaboración Propia

TARJETA CRC	
Clase: acta_actividad_metodologica	
Atributos	
Nombre	
idActa	
departamento	
fecha	
profesor	
cargo	
titulo	
objetivo	
planteamiento	
acuerdo	
Responsabilidades:	Colaboraciones:
	Actividad_metodologica_linea
	material_metodologico

2.4.3 Arquitectura de desarrollo

El marco de trabajo utilizado para el desarrollo de la propuesta de solución, está basado en el patrón de arquitectura de software Modelo-Plantilla-Vista (Model-Template-View o MTV), el cual es una modificación de la arquitectura Modelo-Vista-Controlador (MVC). En el MTV [50]:

- **Modelo:** es la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos. El modelo define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.
- **Plantilla:** la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación. Responde a cómo algunas cosas son mostradas sobre una página web u otro tipo de documento. La plantilla es básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solamente crea contenido en HTML (también XML, CSS, Javascript).

- Vista:** la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada, o sea, es como un puente entre el modelo y las plantillas. La vista se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados, se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.

La interacción de los componentes MTV inicia al usuario hacer una petición del navegador a la URL, encontradas en **url.py** el cuál es el responsable de mapear las rutas en el proyecto. Esta manda la petición a la vista que tiene configurada en el **view.py**, que es el archivo donde se maneja el ciclo de solicitudes/respuestas de la aplicación web. La vista le manda la petición al modelo ubicado en el **models.py** donde se definen las entidades de la aplicación y son traducidos automáticamente por Django a tablas de base de datos. Seguidamente el modelo interactúa con la base de datos, y esta responde al modelo. Luego la vista le envía la respuesta al código HTML y finalmente la plantilla muestra los datos en el navegador. En la Figura 6, se expone el ejemplo del requisito funcional adicionar profesor.

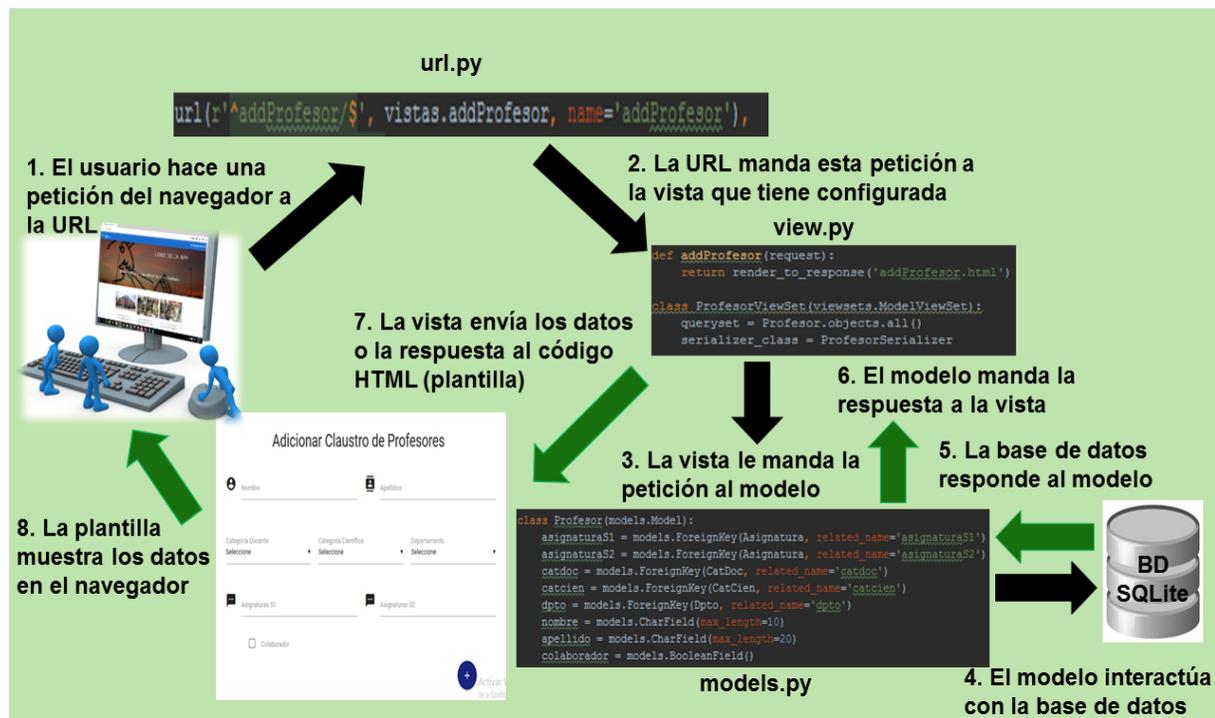


Figura 6: Interacción entre los componentes MTV. Elaboración Propia

2.4.4 Patrones de diseño

Un patrón de diseño constituye un esquema para refinar subsistemas o componentes. Este describe una estructura común y recurrente de componentes interrelacionados, que resuelve un problema general de diseño dentro de un contexto particular, dividiéndose en dos grandes grupos: Patrones de Software de Asignación de Responsabilidad General (GRASP) y Grupo de cuatro (GoF).

Los patrones GRASP nos dan principios generales para asignar responsabilidades y se utiliza sobre todo en la realización de diagramas de interacción.

Los utilizados en la solución son:

- **Experto:** necesario para asignar a una clase la información imprescindible para cumplir su responsabilidad. En este caso en view.py se encuentra la clase *ProfesorViewSet* que contiene toda la información que se necesita visualizar (ver Figura 7).

```
class ProfesorViewSet(viewsets.ModelViewSet):  
    queryset = Profesor.objects.all()  
    serializer_class = ProfesorSerializer
```

Figura 7: Código de patrón Experto. Elaboración Propia

- **Creador:** conveniente para guiar la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que deba conectar con el objeto producido en cualquier evento, ejemplo de esto en la clase *Profesor* (ver Figura 8).

```
class Profesor(models.Model):  
    asignaturaS1 = models.ForeignKey(Asignatura, related_name='asignaturaS1')  
    asignaturaS2 = models.ForeignKey(Asignatura, related_name='asignaturaS2')  
    catdoc = models.ForeignKey(CatDoc, related_name='catdoc')  
    catcien = models.ForeignKey(CatCien, related_name='catcien')  
    dpto = models.ForeignKey(Dpto, related_name='dpto')  
    nombre = models.CharField(max_length=10)  
    apellido = models.CharField(max_length=20)  
    colaborador = models.BooleanField()
```

Figura 8: Código de patrón Creador. Elaboración Propia

- **Controlador:** patrón utilizado para el manejo de los eventos del sistema, que no pertenece a la interfaz del usuario. El patrón controlador se evidencia en el uso de las clases controladoras que se implementan con la aplicación, como es el caso de las clases contenidas en view.py (ver Figura 9).

```

class CatDocViewSet(viewsets.ModelViewSet):
    queryset = CatDoc.objects.all()
    serializer_class = CatDocSerializer

class CatCienViewSet(viewsets.ModelViewSet):
    queryset = CatCien.objects.all()
    serializer_class = CatCienSerializer

class DptoViewSet(viewsets.ModelViewSet):
    queryset = Dpto.objects.all()
    serializer_class = DptoSerializer

class TipoAMViewSet(viewsets.ModelViewSet):
    queryset = TipoAM.objects.all()
    serializer_class = TipoAMSerializer

class AnnoAcadViewSet(viewsets.ModelViewSet):
    queryset = AnnoAcad.objects.all()
    serializer_class = AnnoAcadSerializer

class PlanJAnnoViewSet(viewsets.ModelViewSet):
    queryset = PlanJAnno.objects.all()
    serializer_class = PlanJAnnoSerializer

```

Figura 9: Código de patrón Controlador. Elaboración Propia

Los patrones GOF describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. Estos se dividen en: estructurales, creacionales y de comportamiento [51]. Los patrones GOF utilizados durante el desarrollo del sistema fueron:

- **Estructurales**

Decorador: ayuda a añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Un ejemplo del código donde se evidencia este patrón se representa a continuación:

```

@detail_route(methods=['delete'], url_name='plan_delete', url_path='plan_delete')
def plan_delete(self, request, *args, **kwargs):
    mes = kwargs['pk']
    planes = PlanJAnno.objects.filter(mes=mes)
    for p in planes:
        p.delete()

```

Figura 10: Código de patrón Decorador. Elaboración Propia

2.4.5 Métricas para la validación del diseño

Un aspecto importante a tener en cuenta en la evaluación del diseño, es la creación de métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objetos. Para validar el diseño realizado se tendrán en cuenta distintos atributos de calidad como: la responsabilidad de las clases, la reutilización, la complejidad de implementación y mantenimiento, el bajo acoplamiento y la cantidad de pruebas unitarias. Las métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC) permiten evaluar estos atributos.

Tamaño Operacional de Clase (TOC)

El tamaño operacional de una clase está dado por el número de métodos asignados a una clase. Para ello se miden los atributos de calidad representados en los siguientes gráficos. Al analizar las 23 clases de la herramienta se obtuvo un valor aproximado de 5.409 como promedio de la cantidad de procedimientos, el cual se utilizó para asignar las categorías de baja, media y alta a los atributos representados teniendo en cuenta los criterios mostrados en la Tabla 11.

Tabla 11: Criterio de evaluación de la métrica TOC

	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio.
	Media	Entre Promedio y $2 \times$ Promedio.
	Alta	$> 2 \times$ Promedio.
Complejidad implementación	Baja	\leq Promedio.
	Media	Entre Promedio y $2 \times$ Promedio.
	Alta	$> 2 \times$ Promedio.
Reutilización	Baja	$> 2 \times$ Promedio.
	Media	Entre Promedio y $2 \times$ Promedio.
	Alta	\leq Promedio.

Luego de aplicar los criterios de medición correspondientes a la métrica se obtienen los siguientes resultados mostrados en los que se representan a continuación:

Responsabilidad:

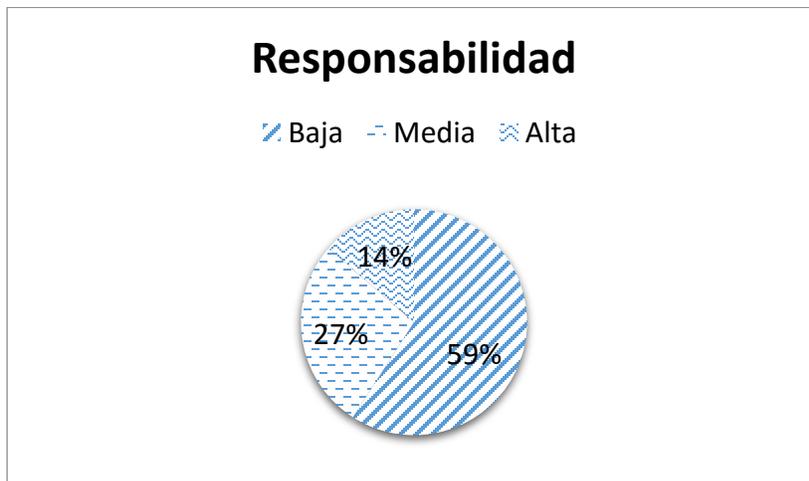


Figura 11: Métrica TOC correspondiente al atributo Responsabilidad. Elaboración Propia

Este atributo se refiere a la responsabilidad asignada a cada clase y se mide a partir del número de métodos que contenga. En la Figura 11 se muestra el resultado de la aplicación de la métrica: el 59% de las clases evaluadas con baja responsabilidad, el 27% con responsabilidad media y el

14% con alta responsabilidad. Más del 50% de las clases fueron evaluadas de baja responsabilidad lo cual indica que el diseño realizado requiere de pocos esfuerzos en las tareas de mantenimiento.

Complejidad:

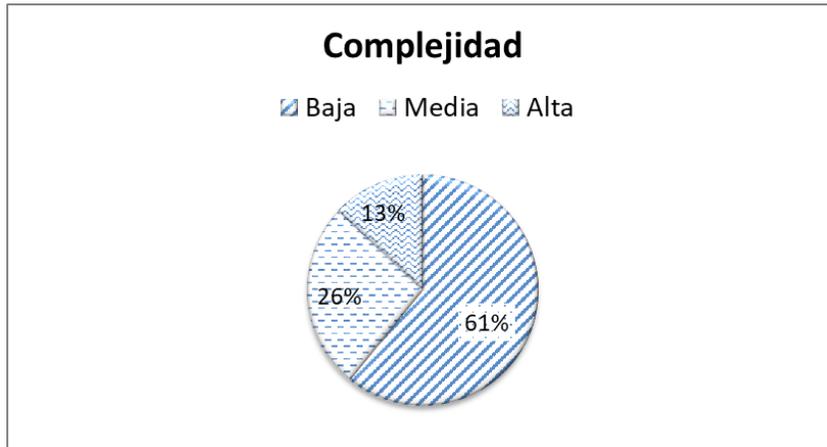


Figura 12: Métrica TOC correspondiente al atributo Complejidad. Elaboración Propia

El atributo complejidad mide el grado de dificultad que tiene implementar un diseño de clases determinado. En la Figura 12 se muestra el resultado de la aplicación de la métrica: el 61% de las clases evaluadas con baja complejidad, el 26% con complejidad media y el 13% con complejidad alta. Más del 60% de las clases fueron evaluadas de baja complejidad lo que indica que el diseño realizado requiere de poco esfuerzo para realizar arreglos o mejoras.

Reutilización:



Figura 13: Métrica TOC correspondiente al atributo Reutilización. Elaboración Propia

Este atributo mide el grado de reutilización presente en una clase. En la Figura 13 se muestra el resultado de la aplicación de la métrica: el 61% de las clases evaluadas con una alta reutilización, el 26% con reutilización media y el 13% con reutilización baja.

Al aplicar esta métrica se comprobó que el diseño de la propuesta de solución presenta un bajo nivel de responsabilidad, logrando un porcentaje bajo de complejidad de implementación. Esto implica un menor grado de dificultad en el desarrollo de la herramienta.

Relaciones entre clases (RC)

Esta métrica está dada por el número de relaciones de uso de una clase con otra y evalúa atributos de calidad como: acoplamiento, complejidad de mantenimiento, reutilización y cantidad de pruebas. Analizadas las 23 clases se obtuvo el valor aproximado de 2.72 de promedio de asociación de uso, el cual se tomó como referencia para asignar las categorías de baja, media y alta a los atributos representados teniendo en cuenta los criterios mostrados en la Tabla 12.

Tabla 12: Criterios de evaluación de la métrica RC

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Reutilización	Baja	>2* Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	<= Promedio.
Cantidad de Pruebas	Baja	<= Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.

Luego de aplicar los criterios de medición correspondientes a la métrica se obtienen los siguientes resultados mostrados en los gráficos que se representan a continuación:

Acoplamiento:

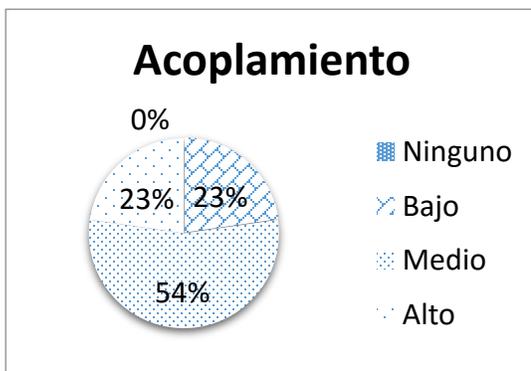


Figura 14: Métrica RC correspondiente al atributo Acoplamiento. Elaboración Propia

El atributo acoplamiento consiste en el grado de dependencia o interconexión de una clase con otras. En la Figura 14 se muestra los resultados de la aplicación de la métrica: el 23% de las clases evaluadas con bajo acoplamiento, el 54% con acoplamiento medio y el 23% con alto acoplamiento. Más del 50% de las clases fueron evaluadas de medio acoplamiento lo cual indica que el grado de dependencia o interconexión de una clase con otras es medio.

Cantidad de pruebas:

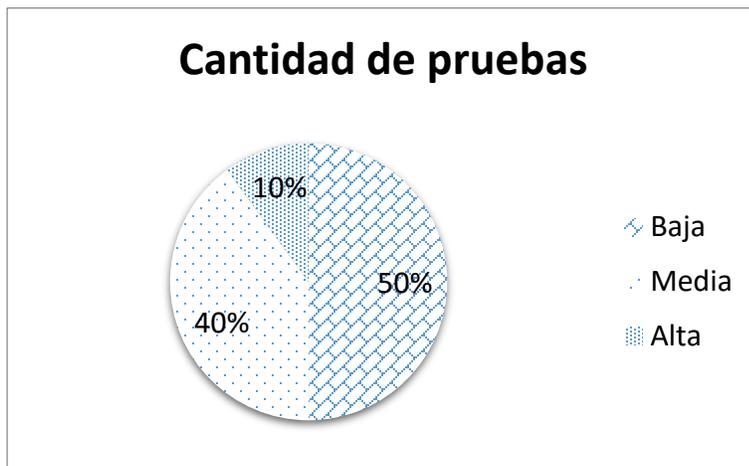


Figura 15: Métrica RC correspondiente al atributo Cantidad de pruebas. Elaboración Propia

El atributo cantidad de pruebas consiste en el número de esfuerzo para realizar las pruebas de calidad. En la Figura 15 se muestra el resultado de la aplicación de la métrica: el 50% de las clases evaluadas con baja cantidad de pruebas, el 40% con media y el 10% con alta cantidad de pruebas. El 50% de las clases fueron evaluadas con bajo esfuerzo, lo cual indica que el diseño realizado requiere un bajo grado de esfuerzo a la hora de realizar cambios, rectificaciones y pruebas de software.

Reutilización:

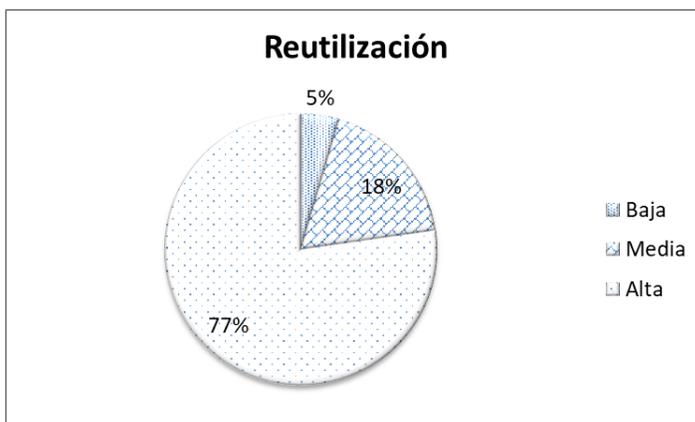


Figura 16: Métrica RC correspondiente al atributo Reutilización. Elaboración Propia

Este atributo mide el grado de reutilización presente en una clase. En la Figura 16 se muestra el resultado de la aplicación de la métrica: el 70% de las clases evaluadas con una alta reutilización, el 18% con reutilización media y el 5% con reutilización baja.

Según lo analizado en las figuras antes expuestas, los valores de RC se comportan de forma satisfactoria. Estos resultados expresan que las clases del diseño de la herramienta presentan bajo acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas son bajas y en consecuencia el grado de reutilización es alto.

2.5 Fase de Desarrollo

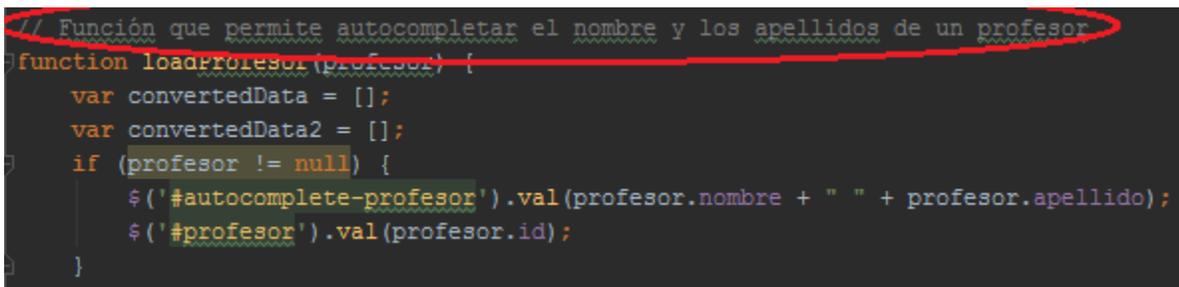
En la fase de desarrollo está incluida toda la programación por parte de los desarrolladores del proyecto. Para realizar esta fase se definieron estándares de codificación ya que programar bajo ellos mantiene el código consistente y facilita su comprensión y escalabilidad. A continuación, se presentan los utilizados en la solución de la herramienta:

2.5.1 Estándares de codificación

Los estándares de codificación, son parte de las llamadas buenas prácticas, estas son un conjunto no formal de reglas que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad del código. A continuación, se describe la nomenclatura seguida en la propuesta de solución.

Nomenclatura de los comentarios y declaración de variables

Los comentarios deben añadir claridad al código y ser precisos, con el propósito de entender que es lo que se está desarrollando. Siempre se pondrán encima de la función a la cual se hace alusión dicho comentario. Las variables se escriben con la primera letra minúscula.



```
// Función que permite autocompletar el nombre y los apellidos de un profesor
function loadprofesor(profesor) {
  var convertedData = [];
  var convertedData2 = [];
  if (profesor != null) {
    $('#autocomplete-profesor').val(profesor.nombre + " " + profesor.apellido);
    $('#profesor').val(profesor.id);
  }
}
```

Figura 17: Nomenclatura de los comentarios y declaración de variables. Elaboración Propia

Declaración de funciones

No existe espacio entre el nombre del método, el paréntesis y la lista de parámetros. Se abre la llave "{" al final de la misma línea que la declaración. La llave de "}" debe aparecer en línea aparte con la misma indentación que el método o clase que cierra.

```

function initDpto(val) {
    $.ajax({
        type: "GET",
        url: "/api/dpto",
    }).done(function (data) {
        $('#option-dpto').remove();
        $.each(data, function (k, v) {
            $('#dpto').append($('', {
                class: 'option-dpto',
                value: v.id,
                text: v.nombre
            }));
        });
        $('#dpto').material_select('destroy');
        $('#dpto').val(val).change();
        $('#dpto').material_select();
    });
}

```

Figura 18: Declaración de funciones. Elaboración Propia

Nomenclatura de las clases

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, exceptuando el caso de los nombres compuestos que cada palabra comienza con letra mayúscula.

```

class ActMet(models.Model):
    tipoAM = models.ForeignKey(TipoAM, on_delete=models.CASCADE, blank=True)
    objetivo = models.TextField(blank=True)
    fecha = models.CharField(max_length=100, blank=True, default='2018')
    participa = models.CharField(max_length=100, blank=True)
    ejecuta = models.CharField(max_length=100, blank=True)
    plan = models.IntegerField(blank=True, default=0)
    real = models.IntegerField(blank=True, default=0)
    visitadas = models.IntegerField(blank=True, default=0)
    linea = models.ForeignKey(LineaTM, on_delete=models.CASCADE, related_name='linea', null=True)

```

Figura 19: Nomenclatura de las clases. Elaboración Propia

2.5.2 Interfaces del sistema

A continuación, se muestran algunas interfaces del sistema, así como la descripción de sus elementos y funcionalidades.

En la interfaz mostrada en la Figura 20 el usuario se autentica y en dependencia del rol se le mostrará la pantalla inicial.



Figura 20: Interfaz Autenticar Usuario. Elaboración Propia

La Figura 21 corresponde a la interfaz inicial de la herramienta donde se muestran los cuatro componentes antes descritos (Planificación, Ejecución, Control y Análisis), en este caso para el jefe de departamento.

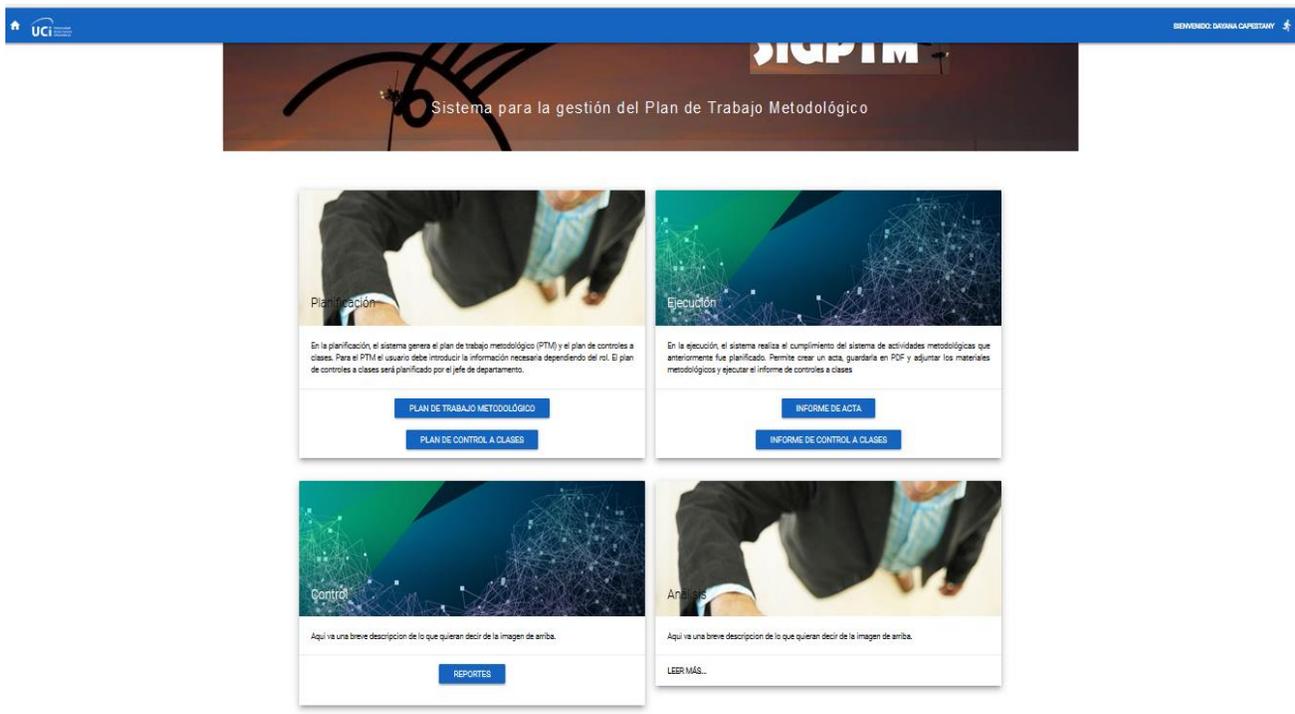


Figura 21: Interfaz del sistema correspondiente a la página inicial. Elaboración Propia

La Figura 22 corresponde al requisito funcional Crear acta de actividad metodológica. Esta interfaz se muestra al vicedecano de formación, jefe de departamento y profesor principal de año académico.

The screenshot shows a web form titled "Acta de Actividad Metodológica". The form contains several input fields and buttons:

- Departamento:** A dropdown menu with the text "Seleccione".
- Fecha:** A date selection field with a calendar icon.
- Profesor que Imparte/Dirige:** A text input field with a person icon.
- Categoría Docente:** A dropdown menu with the text "Seleccione".
- Categoría Científica:** A dropdown menu with the text "Seleccione".
- Cargo:** A text input field with a person icon.
- Tipo de Actividad Metodológica:** A dropdown menu with the text "Seleccione".
- Título:** A text input field with a document icon.
- Objetivos:** A text input field with a list icon.
- Principales Planteamientos:** A text input field with a list icon.
- Acuerdos:** A text input field with a list icon.
- ADJUNTAR MATERIALES:** A blue button with the text "Upload one or more files".
- CREAR ACTA:** A blue button with a right-pointing arrow.

Figura 22: Interfaz correspondiente a Crear acta de actividad metodológica. Elaboración Propia

2.6 Conclusiones parciales

Como resultados del desarrollo de la herramienta para la gestión PTM se obtuvo:

- Un modelo de la gestión del PTM que consta de cuatro componentes (Planificación Ejecución Control y Análisis) y sigue la estructura organizativa del proceso que permite la organización, estructuración y disponibilidad de la información.
- Una herramienta para la gestión del PTM que soporta el modelo anterior cuyo desarrollo fue guiado por la metodología XP, generando los artefactos correspondientes en cada una de sus fases.

De esta manera se le da cumplimiento al segundo y tercer objetivo específico.

Capítulo 3: Validación de la calidad de la herramienta SiGPTM

En el presente capítulo se valida la calidad de la herramienta para la gestión del plan de trabajo metodológico (SiGPTM) según lo indica la cuarta fase de la metodología XP: Fase de Pruebas. Se validan los niveles de pruebas mediante las pruebas unitarias realizadas por el equipo de desarrollo, las pruebas de liberación realizadas por el grupo de calidad de software del centro CEGEL y las pruebas de aceptación realizadas de conjunto con el cliente. Además, se verifica la viabilidad de la propuesta de solución mediante la aplicación de la técnica ladov con el objetivo de medir el nivel de satisfacción de los usuarios respecto a los tres indicadores: organización, estructuración y disponibilidad. En la Figura 14 se representa la estrategia utilizada para la verificación de la viabilidad de la solución.

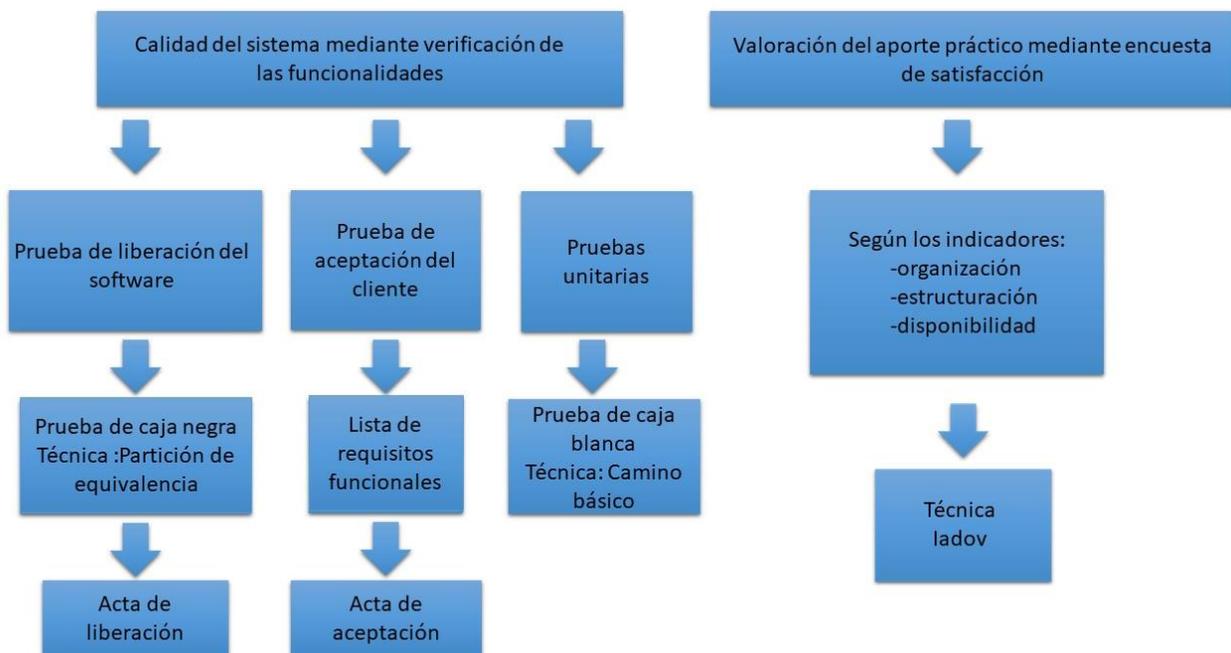


Figura 23: Estrategia para la verificación de la viabilidad de la solución. Elaboración Propia

3.1 Fase de Pruebas

En la metodología XP, la fase de pruebas es un elemento crítico para garantizar la calidad del software y representa una revisión final de las especificaciones de los requisitos, del diseño y de la codificación. Estas pruebas tienen como objetivo encontrar y corregir el mayor número posible de errores con la menor cantidad de esfuerzo y tiempo. Deben conducirse sistemáticamente, y los casos de prueba deben diseñarse utilizando técnicas de caja blanca y negra [52]. El software debe probarse desde dos perspectivas diferentes:

1. Diseño de casos de prueba de caja blanca: se basa en la lógica interna del programa.

2. Diseño de casos de prueba de caja negra: realiza pruebas sobre los requisitos externos del software.

La validación de la herramienta se realizó transitando por los cuatro niveles de prueba: unidad, integración, sistema y aceptación. En el nivel de unidad se aplicaron las pruebas de caja blanca utilizando el método de camino básico, en el de integración las técnicas que prevalecieron fueron las de diseño de casos de prueba de caja negra. En el nivel de sistema, se emplearon las pruebas de liberación por el grupo de calidad del centro CEGEL. Las pruebas de aceptación fueron realizadas con el cliente mediante la revisión de los requisitos.

3.2 Pruebas unitarias

Las pruebas unitarias se encuentran en el nivel de unidad, estas son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada. Hacen un uso intensivo de las técnicas de prueba de caja blanca, ejercitando los caminos específicos.

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Tiene como objetivo examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado. Al utilizar sus métodos garantiza que se ejercita, al menos una vez, todos los caminos independientes de cada módulo y las estructuras internas de datos para asegurar su validez [52].

Al sistema desarrollado se le aplicó la técnica de camino básico. Esta última permite obtener una medida de la complejidad lógica del diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye un grafo de flujo asociado y se calcula su complejidad ciclomática. A continuación, se describen los pasos para ejecutar dicha técnica tomando como ejemplo el método *loadProfesor*.

```

function loadProfesor(profesor) {
  var convertedData = [];
  var convertedData2 = [];
  if (profesor != null) {
    $('#autocomplete-profesor').val(profesor.nombre + " " + profesor.apellido);
    $('#profesor').val(profesor.id);
  }
  $.ajax({
    type: "GET",
    url: "/api/profesor"
  }).done(function (data) {
    $.each(data.data, function (index, value) {
      var nombre = value.nombre + " " + value.apellido;
      convertedData[nombre] = null;
      convertedData2[value.id] = {tag: value.nombre + " " + value.apellido, id: value.id};
    });
    $('#autocomplete-profesor').autocomplete({
      data: convertedData,
      minLength: 0,
      onAutocomplete: function (val) {
        $.each(data.data, function (index, value) {
          if (value.nombre + " " + value.apellido == val) {
            $('#profesor').val(value.id);
          }
        });
      }
    });
  });
}

```

Figura 24: Código Fuente loadProfesor. Elaboración Propia

1. A partir del diseño o del código fuente, se genera el grafo de flujo asociado como se muestra en la Figura.

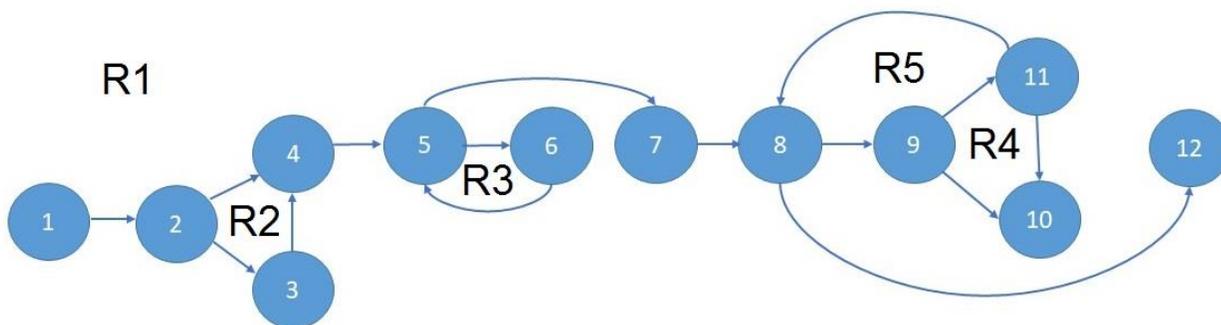


Figura 25: Grafo de flujo. Elaboración Propia

2. Se calcula la complejidad ciclomática del grafo. Existen tres formas para calcularla, donde G: Grafo, R: Regiones, A: Aristas, N: Nodos, P: Nodos Predicados.

➤ $V(G) = R = 5$

$$\text{➤ } V(G) = A - N + 2 = 14 - 11 + 2 = 3 + 2 = 5$$

$$\text{➤ } V(G) = P + 1 = 4 + 1 = 5$$

Si el resultado de la complejidad ciclomática por las 3 formas es el mismo, entonces podemos considerar el grafo de flujo correcto.

3. Se determina un conjunto básico de caminos independientes.

Camino 1: 1, 2, 3, 4, 5, 6, 5, 7, 8, 9, 10, 11, 8, ...

Camino 2: 1, 2, 4, 5, 7, 8, 9, 11, 8, 12

Camino 3: 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 8, ...

Camino 4: 1, 2, 4, 5, 6, 5, 7, 8, 9, 11, 8, 12

Camino 5: 1, 2, 3, 4, 5, 7, 8, 9, 11, 12

Los puntos suspensivos (...) que siguen a los caminos 1 y 3 indican que cualquier camino del resto de la estructura de control es aceptable.

4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico. A continuación, se muestra uno de los casos de prueba que satisfacen el conjunto básico previamente descrito correspondiente al camino 5.

Tabla 13: Caso de prueba de caja blanca correspondiente al camino 5. Elaboración Propia

Descripción del método	Permite autocompletar el nombre y apellidos de un profesor
Condición de ejecución	1) Si el profesor en la base de datos es distinto de nulo. 2) Iterar por cada uno de los profesores en la base de datos. 3) Iterar por el profesor que se entró por parámetro. 3.1) Si el nombre y el apellido del profesor entrado es igual al valor encontrado en 2.
Entrada	Profesor
Resultado esperado	Autocompletado el nombre y apellido del profesor.

3.3 Pruebas de integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la integración. El objetivo es coger los módulos probados mediante las pruebas unitarias y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño a partir de los casos de prueba de caja negra.

Las pruebas de caja negra, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente

todos los requisitos funcionales de un programa. Estas pruebas intentan descubrir errores enmarcados en las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, en estructuras de datos, de rendimiento, de inicialización y terminación [52].

A continuación, se presenta el escenario que permite crear un acta de actividad metodológica.

Tabla 14: Caso de prueba del escenario crear acta de actividad metodológica. Elaboración Propia

Descripción	Variable 1	Respuesta del sistema	Flujo central
Permite crear un acta de actividad metodológica.	V (introducir valores en los campos)	El sistema muestra el mensaje "acta creada CORRECTAMENTE".	1. Seleccionar la opción Crear Acta Metodológica . 2. Introducir los valores en los campos. 3. Seleccionar la opción Aceptar . 4. El sistema agrega el acta a la base de datos.
	I (introducir un valor numérico)	El sistema muestra el mensaje "Existen parámetros con errores".	
	I (dejar campos vacíos)	El sistema muestra el mensaje "Existen campos vacíos".	

3.4 Pruebas de sistema

Este tipo de pruebas deben ser ejecutadas idealmente por un equipo de pruebas ajeno al equipo de desarrollo, una buena práctica en este punto corresponde a la tercerización de esta responsabilidad. Por tal razón, fueron realizadas las pruebas de liberación por el grupo de calidad del centro CEGEL. La obligación de este grupo, consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a las especificaciones definidas en el proyecto.

3.4.1 Pruebas de liberación

Para llevar a cabo la prueba de liberación del software, se aplicó el método de prueba de caja negra por CEGEL. Con el objetivo de comprobar que las funcionalidades de la herramienta se realizaron correctamente y responden a las necesidades del cliente, el método se aplicó en tres iteraciones como se muestra en la Figura 23.

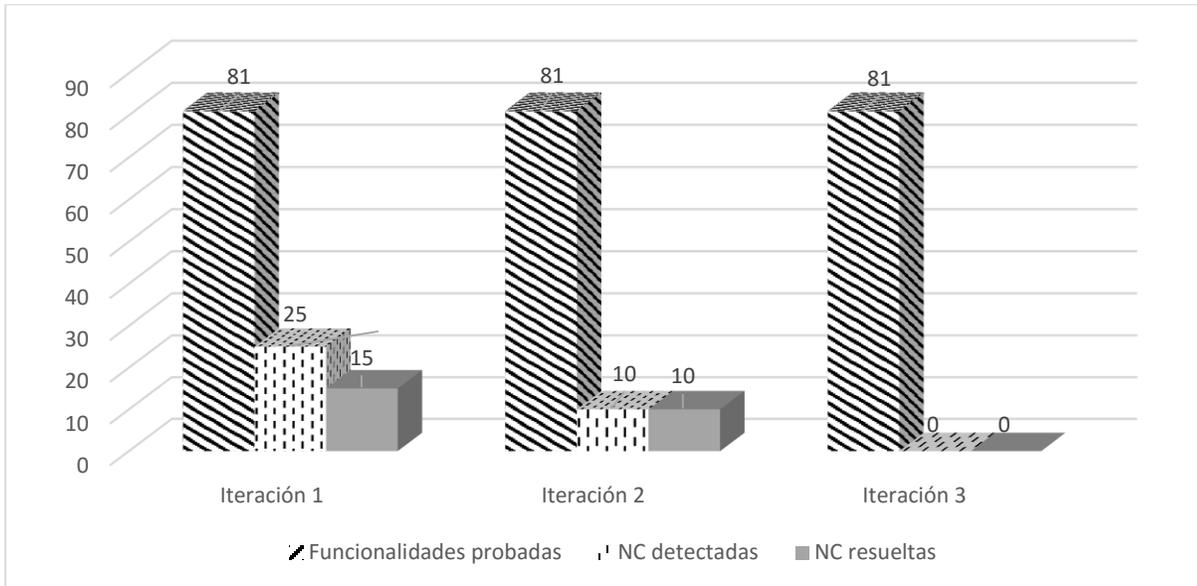


Figura 26: Resultados de la prueba de liberación. Elaboración Propia

En la primera iteración se detectaron un total de 25 No Conformidades (NC), predominando entre ellas las de tipo funcionalidad y de interfaz, de prioridad alta, al finalizar la iteración quedaron resueltas 15 NC. En la segunda iteración los resultados mejoraron al disminuir a 10 NC de importancia media y de tipo ortografía quedando todas resueltas, ya en la tercera iteración se obtuvieron 0 NC. La imagen anterior ilustra los resultados de aplicar el método de caja negra, teniendo en cuenta los tipos de NC identificados (funcionalidad, ortografía, interfaz).

3.5 Pruebas de aceptación

La prueba de aceptación es indispensable en este proceso. Generalmente es desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar cómo el sistema satisface sus criterios de aceptación, validando los requisitos que han sido levantados para el desarrollo, incluyendo la documentación y procesos de negocio. Está considerada como la fase final del proceso, para crear un producto confiable y apropiado para su uso [52].

Para la aplicación de estas pruebas se confeccionó un caso de prueba de aceptación por cada HU. Seguidamente se muestra el caso de prueba de aceptación de la HU "Crear PTM".

Tabla 15: Caso de prueba de aceptación de la HU "Crear PTM". Elaboración Propia

Caso de prueba de aceptación		
Código de caso de prueba: 1		Nombre de HU: Crear PTM por estructura
Nombre de la persona que realizó el caso de prueba: Dayana Capestany Varona Darlyn Díaz Morales		
Descripción de la prueba: revisar a través de la herramienta el correcto funcionamiento del requisito funcional Crear PTM.		
Condiciones de ejecución: el usuario debe haber introducido el curso académico.		
Entrada/Pasos de ejecución		
Acción:	Entrada:	Resultados esperados:
Se selecciona la opción de Crear PTM	<ol style="list-style-type: none"> 1. Fundamentación. 2. Caracterización del claustro de profesores 3. Problemas metodológicos 4. Líneas de trabajo metodológico. 5. Objetivo general de líneas metodológicas. 6. Objetivos específicos de líneas metodológicas. 7. Indicadores metodológicos. 8. Sistema de actividades metodológicas por líneas metodológicas. 9. Otras actividades metodológicas. 10. Orientaciones de controles a clases. 	El sistema debe mostrar el PTM creado.
Evaluación de prueba: Satisfactorio		

Se realizó un encuentro con la MsC. Elizabeth Rodríguez Stiven, con el objetivo de revisar las funcionalidades de la herramienta, teniendo en cuenta los casos de pruebas definidos. Los resultados que se obtuvieron se calificaron como satisfactorios, avalados por la especialista entrevistada, definiendo la aplicación como un importante aporte para la gestión del PTM en la UCI.

3.6 Valoración del aporte práctico de la investigación

Para valorar la satisfacción de los usuarios finales sobre el sistema se utilizó la técnica ladov. Esta técnica está conformada por cinco preguntas: tres cerradas y 2 abiertas con el objetivo de verificar si el sistema contribuye con la organización, estructuración y disponibilidad de la información. Las preguntas abiertas permiten profundizar en la naturaleza de las causas que originan los diferentes

niveles de satisfacción. Los criterios que se utilizan en ladov se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov". Para la realización de esta técnica se seleccionaron 10 usuarios que interactúan directamente con la aplicación SiGPTM, entre ellos están jefes de departamentos, profesores principales de años académicos, vicedecano de formación y profesor. A estos se les aplicó la encuesta para determinar el grado de satisfacción con la propuesta de solución.

El número resultante de la interrelación de las tres preguntas nos indica la posición de cada sujeto en la escala de satisfacción.

La escala de satisfacción es la siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

+1	Máximo de satisfacción
0,5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Figura 27: Niveles de satisfacción. Elaboración Propia

La satisfacción grupal se calcula mediante la fórmula:

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

En esta fórmula *A, B, C, D, E*, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5 y donde *N* representa el número total de sujetos del grupo.

El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

Estos valores representados gráficamente en un eje, se aprecian de la forma siguiente

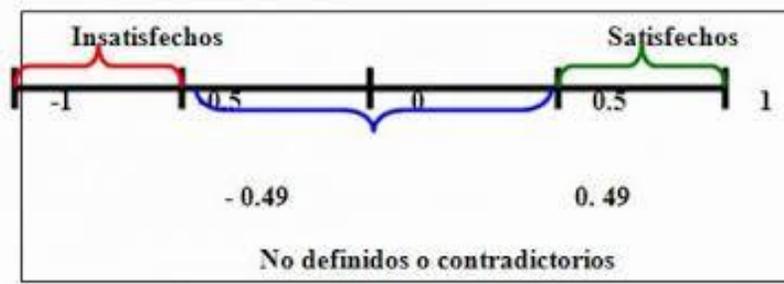


Figura 28: Rango de valores para el cálculo de satisfacción grupal. Elaboración Propia

Preguntas:

1. ¿Le gusta la forma en que se diseñó el sistema SiGPTM?
2. ¿Cree usted que con la herramienta SiGPTM se logra una organización, estructuración y disponibilidad de la información?
3. ¿Siente usted que la herramienta SiGPTM le es útil para su trabajo?
4. ¿Utilizando la herramienta SiGPTM es complejo realizar el seguimiento y control de las actas de actividades metodológicas?
5. ¿Se siente satisfecho con el resultado obtenido con la aplicación del SiGPTM propuesto para contribuir con la organización, estructuración y disponibilidad de la información?

Tabla 16: Cuadro lógico de ladov. Elaboración Propia

5. ¿Le gusta la forma en que se diseñó el sistema SiGPTM?	1. ¿Se siente satisfecho con el resultado obtenido con la aplicación del SiGPTM propuesto para contribuir con la organización, estructuración y disponibilidad de la información?								
	No			No sé			Si		
	3. ¿Siente usted que la herramienta SiGPTM le es útil para su trabajo?								
	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Al aplicar la encuesta se obtuvieron los siguientes resultados:

Tabla 17: Resultados de la escala de satisfacción. Elaboración Propia.

Total de encuestados(N)		Escala
Máximo de satisfacción	6	A
Más satisfecho que insatisfecho	4	B
No definido y contradictorio	0	C
Mas insatisfecho que satisfecho	0	D
Máxima insatisfacción	0	E
Contradictoria	0	F

Resultados obtenidos:

$$ISG = \frac{7(+1) + 3(+0.5) + 0(0) + 0(0.5) + 0(-1)}{10}$$

$$ISG = 0.8$$

Como resultado del cálculo se obtuvo un Índice de Satisfacción Grupal (ISG) con alto valor, que refleja aceptación de la propuesta y un reconocimiento a su utilidad. El cuestionario contribuyó a verificar el desarrollo de las nuevas funcionalidades. Por lo que se puede afirmar que con la solución propuesta se logró desarrollar una herramienta informática para la gestión del plan de trabajo metodológico (SiGPTM) contribuyendo a la obtención de la información organizada, estructurada y disponible.

3.7 Conclusiones parciales

- La calidad de la herramienta fue validada en los cuatro niveles de prueba obteniendo como resultado la carta de aceptación del cliente y el acta de liberación emitida por el centro CEGEL certificando así la correcta funcionalidad del mismo.
- La viabilidad de la propuesta fue probada con resultados satisfactorios mediante el uso de la técnica de ladov. Lo cual confirma que la herramienta desarrollada permite la organización, estructuración y disponibilidad de los datos.

Conclusiones generales

Al finalizar el presente trabajo se arribó a las siguientes conclusiones:

- El estudio de los conceptos asociados a la gestión universitaria permitió definir buenas prácticas, analizar tecnologías para seleccionar las más acorde a las características de la herramienta.
- El análisis de las etapas de los sistemas de gestión permitió modelar el PTM obteniendo mayor organización, estructuración y disponibilidad de los datos.
- El desarrollo de la herramienta SiGPTM permite gestionar el PTM contribuyendo a obtener la información organizada, estructurada y disponible.
- La aplicación de métricas de validación del diseño, pruebas unitarias y de aceptación a la herramienta SiGPTM, certifican la obtención de una aplicación informática que responde a los requerimientos del cliente, avalado por las actas de liberación y aceptación emitidas.

Con el empleo de la herramienta desarrollada, se contribuye a la organización, estructuración y disponibilidad de la información para la gestión del PTM.

Recomendaciones

Durante la realización de la presente investigación surgieron ideas que pueden servir como recomendación para el perfeccionamiento de la herramienta:

- Aplicar técnicas de minería de texto que permitan futuros análisis sobre opiniones o señalamientos.

Bibliografía

- [1] Díaz-Canel, Miguel I, «Cuba superó el millón de graduados universitarios. En Conferencia Especial en Pedagogía 2011,» *Revista de Educación*, pp. 81-110, 2011.
- [2] Castro Ruz, Fidel, «Discurso en la universidad de La Habana dado por nuestro Comandante en Jefe: Fidel Castro Ruz,» *Granma*, 11 05 1959.
- [3] Veitía Arrieta, DrC. Isabel Julia, Machado Bravo, DrC. Ena Margarita y Campos Maura, DrC. Eraidá, «Trabajo metodológico y su enfoque de sistema en la educación superior pedagógica,» vol. III, nº 1, 2014.
- [4] Silva Horrúitiner, Pedro, «El trabajo metodológico. Una concepción desde la Vicerrectoría Académica.,» *Pedagogía Universitaria*, vol. III, nº 1, 1998.
- [5] Ministerio de Educación Superior *Resolucion No 210-07 Reglamento para el Trabajo Docente Metodologico Educacion Superior Cap 2*, La Habana: Gaceta Oficial de la República de Cuba, julio 2007.
- [6] García Hoz, Víctor, *Calidad de educación, trabajo y libertad*, Madrid: Dossat, 1982.
- [7] Hernandez, Andry Suarez, «Arquitectura para Akademos 2.0,» La Habana, junio 2008.
- [8] González, Ana Maria Sánchez, «Sistema de Gestión Universitaria,» [En línea]. Available: <http://www.monografias.com/trabajos92/gestion-universitaria/gestion-universitaria.shtml>.
- [9] Ministerio de Educación Superior , «Ministerio de Educación Superior,» [En línea]. Available: <http://www.mes.gob.cu/es/pregrado>.
- [10] Bates, Tony, *Cómo gestionar el cambio tecnológico. Estrategias para los responsables de centros universitarios*, 2001.
- [11] Cela, Jose, Sogues, Montserrat y Flores, Óscar, *Competencias profesionales de un perfil psicopedagógico en el proceso de incorporación de las Tecnologías de la Información y la Comunicación (TIC) a la docencia universitaria*, 2002.
- [12] Salinas, Jesús, «Innovación docente y uso de las TIC en la enseñanza universitaria,» *Universities and Knowledge Society Journal*, vol. 1, nº 1, 2004.

- [13] Meneses Benítez, Gerardo, «Universidad: NTIC, interacción y aprendizaje,» *Eduotec. Revista Electrónica de Tecnología Educativa*, nº 20, 2006.
- [14] Ministerio de Educación Superior, *Resolucion No119-08*.
- [15] Ministerio de Educación Superior, *Resolucion No 268-91*.
- [16] Sobrino Pontigo, Elena y Echemendía Gallego, J, «Visión de la labor y algunos resultados concretos de la aplicación del trabajo metodológico,» 2012.
- [17] Hernández, Tatiana, Godínez, Isaac y Cira, Lidia, «Procedimiento para el diseño e implantación de un sistema de gestión integrado,» *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal*, vol. XXVIII, nº 2, pp. 27-33, 2007.
- [18] Baños Martínez, Marcos Antonio, «Procedimiento para la mejora del proceso de Formación del Profesional del Sistema de Gestión de la Cujae,» 2015.
- [19] Gurmendi, María de Luján, «Sistema de información universitaria-Programa SIU,» Brasil, 2000.
- [20] Diaz, Javier, y otros, «Accesibilidad Web en la Práctica Inicativas Académicas en Informática,» *Anales Simposio sobre la Sociedad de la Información*, vol. 40, 2011.
- [21] Dara Informática, «SIGA Principal Sistema Integrado de Gestión Académica,» [En línea]. Available: <http://www.dara.es/siga> .
- [22] Rodríguez Guerra, Claudia Amelia y Salgueiro, Armando Plasencia, «Sistemas de Gestión Académica de libre acceso,» *Revista de Control, Cibernética y Automatización*, 2015.
- [23] OpenSIS, «OpenSIS,» [En línea]. Available: <http://www.opensis.com/>.
- [24] SIGMA, «SIGMA Academic,» [En línea]. Available: <http://www.sigmaaie.org/es/soluciones/sigma-academic?id=1&t=1>.
- [25] Calderín Delgado, Yanoski, «GESTACAD. Sistema para la gestión académica».
- [26] Calderín Delgado, Yanoski y Telot González, Julio A, «Gestacad.Sistema para a gestión académica,» *Departamento de Informática.Universidad de Matanzas "Camilo Cienfuegos"*, p. 66, 2004.

- [27] Muñoz Castillo, Vanessa Danae, y otros, «SIGENU-DSS-LITE: Nuevas capacidades de integración de información docente en Instituciones de Educación Superior en Cuba,» *Ciencias de la Información*, vol. 46, nº 2, pp. 3-8, 2015.
- [28] González, Ana Maria Sánchez, «Sistema de Gestión universitaria,» La Habana.
- [29] Yordanis Camejo Valdivia, Grisel Infante Costa, Darién Cepero Rojas, Frank Benavides Dalmendray, «AKADEMOS, UN SISTEMA AUTOMATIZADO PARA LA GESTIÓN ACADÉMICA,» *Serie Científica. Universidad de las Ciencias Informáticas*, vol. 1, nº 1, 2007.
- [30] Duque, Raúl González, Python para todos, 2015.
- [31] Python Software Foundation (US), «Python,» 2016. [En línea]. Available: <https://www.python.org>.
- [32] Duque, Raúl González, El gran libro de HTML5, CSS3 y Javascript, marcombo, ediciones técnicas, 2012.
- [33] Osmani, Addy, Learning JavaScript Design Patterns: A JavaScript and jQuery Developer's Guide, O'Reilly Media, Inc, 2012.
- [34] Django, «Django en español,» 2016. [En línea]. Available: <http://django.es>.
- [35] Materialize, «materializecss.com,» [En línea]. Available: <https://materializecss.com>.
- [36] Islam, Quazi Nafiul, Mastering PyCharm. Use PyCharm with fluid efficiency, 2015.
- [37] Owens, Michael y Allen, Grant, The Definitive Guide to SQLite, 2010.
- [38] SQLite Administrator, «SQLite Administrator,» [En línea]. Available: <https://sqliteadmin.orbmu2k.de/>.
- [39] «Visual Paradigm,» 5 diciembre 2016. [En línea]. Available: <http://www.visual-paradigm.com/aboutus/newsreleases/vp122.jsp>.
- [40] Eriksson, Hans-Erik, y otros, UML 2 toolkit, John Wiley & Sons, 2003.
- [41] H, Canós José, Letelier, Patricio y M.Carmen., «Metodologías ágiles en el desarrollo de software,» 2003.

- [42] Arias Figueroa, D., y otros, «Metodologías ágiles y desarrollo basado en el conocimiento evaluación cuantitativa de F/OSS para la reutilización, normas ISO y su aplicación en centros educativos,» *XIV Workshop de Investigadores en Ciencias de la Computación*, 2012.
- [43] Orjuela Duarte, Ailin y Rojas, Mauricio, «Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo,» *Avances en Sistemase Informatica*, vol. 5, nº 2, 2008.
- [44] D. Bustamante y J. C. Rodríguez, «Metodología Actual: Metodología XP,» 2014.
- [45] M. Mendoza Sánchez, «Metodologías de desarrollo de software,» 2004.
- [46] . Beck, Kent y Martínez Zapata, Francisco Javier, «Una explicación de la programación extrema: aceptar el cambio,» *Revista Técnica García Molina*, 2002.
- [47] Pérez Virgen, Henry León y Salamando Mejía, Carlos Alberto, Levantamiento de requerimientos basados en el conocimiento del proceso, Univeridad Tecnológica de Pereira, 2012.
- [48] Jeffries, R., Anderson, A., Hendrickson, C, *Extreme Programming Installed*, Addison-Wesley Professional, 2001.
- [49] Joskowicz, José., «Reglas y prácticas en eXtreme Programming,» *Universidad de Vigo*, vol. 22, 2008.
- [50] K. Beck, «Embracing change with extreme programming,» vol. 32, nº 10, pp. 70-77, 1999.
- [51] Condori Ayala, José Luis, «Phython-Django. Framework de desarrollo web para perfeccionistas. Basado en el Modelo MTV,» *Revista de Información, Tecnología y Sociedad*, p. 36, 2012.
- [52] Guerrero, Carlos A., Suárez, Johanna M. y Gutiérrez, Luz E, «Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web,» *Información tecnológica*, vol. IV, nº 3, pp. 103-114, 2013.
- [53] Pressman, Roger S, *Ingeniería del software. Un enfoque práctico*, La Habana: Félix Varela, 2005.

