



Facultad 1

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

Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación

Autor: Leandro Valdivia Martínez

Tutor(es):

Ing. Jeanne Morejón Cruz

M Sc. Alexander Rodríguez Rabelo

La Habana, septiembre de 2020

Año 62 del Triunfo de la Revolución

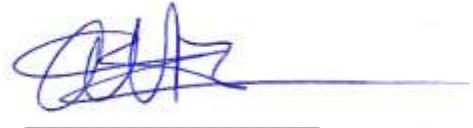
Declaración de autoría

Yo Leandro Valdivia Martínez declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 14 días del mes de septiembre del año 2020.



Leandro Valdivia Martínez



Ing. Jeanne Morejón Cruz

M Sc. Alexander Rodríguez Rabelo

Resumen

La Práctica Profesional del Docente, constituye un proceso complejo en el que confluyen múltiples factores que van a incidir en la concreción de teorías, lineamientos, políticas; es decir, en el logro de los fines educativos. En tal sentido, el docente como centro del proceso de la Práctica, debe lidiar con múltiples y simultáneos elementos en su hacer pedagógico. En el Ministerio de Educación de la República de Cuba se limita la correcta gestión de la información por parte de los profesores, la toma de decisiones y el uso efectivo de los recursos humanos en función de un mayor aprovechamiento debido a la insuficiente introducción de nuevas tecnologías que lo propicien.

La presente investigación tiene como objetivo desarrollar un módulo para el Sistema de Gestión Académica que gestione la información de la práctica docente de los profesores en el Ministerio de Educación. Para ello, Se definió como metodología de desarrollo de software AUP en su variante UCI. Soportado en Angular 8 como framework para aplicaciones web, Visual Paradigm para UML 15.0 como herramienta de modelado, como Servidor de aplicaciones web Nginx 1.14, PgAdmin III como aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. Como lenguajes a emplear se definieron CSS 3, HTML 5 y PHP 7.2.

A partir del ambiente de desarrollo anterior y la aplicación de métodos científicos y técnicos se obtuvo el análisis y diseño del Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación, el cual propiciará una mejor gestión de la información académica.

Palabras clave: gestión de la información, Práctica Docente, MINED, software.

Índice

Declaración de autoría.....	I
Resumen	II
Índice.....	III
Introducción	1
Capítulo 1: Fundamentos teóricos de la investigación	6
1.1. Marco conceptual de la investigación	6
1.2. Sistemas homólogos para la informatización de la Práctica Docente	10
1.3. Ambiente de desarrollo.....	13
1.4. Metodología de desarrollo de software	16
1.5. Conclusiones parciales.....	17
Capítulo 2: Análisis y diseño de propuesta de solución.....	19
2.1. Modelo de Negocio.....	19
2.2. Requisitos del sistema.....	21
2.3. Descripción de la arquitectura	28
2.4. Diagrama de Clases del Diseño.....	32
2.5. Diagrama de paquetes	33
2.6. Conclusiones parciales.....	34
Capítulo 3: Implementación y validación de la propuesta de solución.....	36
1.1 Estándares de codificación.....	36
1.2 Tratamiento de errores	38
1.3 Pruebas internas	38
1.4 Conclusiones parciales.....	40
Conclusiones generales.....	41
Recomendaciones	42

Referencias Bibliográficas43

Anexos46

Introducción

La educación, proceso inexcusablemente requerido en la natural evolución del ser humano, permanece siempre incompleta, siempre falible en su esencial pretensión de encontrar mejores formas de vida para la humanidad. No pueden las personas eludir la actividad educativa. Es la esencia de su permanencia en el universo, la responsable de su actual realidad, la precursora de su proyección futura.

Los cambios vertiginosos que se viven en la actualidad con respecto al sistema educativo y el perfil de sus maestros, demandan que estos cada vez estén mejor preparados en su práctica profesional. Las escuelas han venido transformándose a lo largo del tiempo y han atendido la formación de los docentes en los diferentes ámbitos de la educación básica; con ello se hace necesario que los alumnos se desarrollen en sus competencias profesionales y genéricas, así como el que adquieran experiencias necesarias para el trabajo docente (Mercado & Rockwell, 1988).

La Práctica Docente, se caracteriza por la complejidad, singularidad y simultaneidad de las interacciones que en ella se suscitan. Sin duda, un proceso de esta naturaleza está rodeado de diversas concepciones, dilemas, y obstáculos sobre los que con poca frecuencia se reflexiona y sobre los que cada practicante actúa con el repertorio de potencialidades que posea (Achili, 1988).

Al respecto Cifuentes (1999), señala dos complejidades de la Práctica: a) La complejidad epistemológica referida al qué se pretende conseguir con las prácticas de enseñanza y afirma que lógicamente se pretende que el estudiante se inicie en la profesión del profesor, pero esto es más que adquirir conocimientos y destrezas. Se trata también de la adquisición de intereses, valores y actitudes de los profesores, aprender las características, significado y función social de la profesión. b) La complejidad organizativa derivada de la relación entre dos instituciones bastante diferenciadas, universidad y centros de enseñanza no universitaria, y la particularidad de la Práctica Profesional que trasciende lo disciplinar y requiere de una organización interdisciplinar. Aunado a esto se encuentra el hecho de la dificultad de unificar criterios y conceptos sobre realidades tan complejas.

La acción tutorial viene siendo muy diferente en una escuela alemana, donde se forja un compromiso esencial entre profesores y alumnos en la generación de un conocimiento científico, que, en la universidad francófona, orientada al desarrollo de las profesiones, o que, en las universidades inglesas, como la de Oxford, donde el tutor es una figura docente reconocida en el ámbito universitario por su capacidad para desarrollar el pensamiento reflexivo y crítico en los alumnos. De esta forma, se priva del suficiente detenimiento al desarrollo de una práctica que, necesariamente unida a la que

tiene lugar en las aulas, logre dar continuidad al trabajo docente de los futuros profesores (Gaitán et al., 2005).

El tutor durante los diferentes semestres apoya a que los estudiantes se aproximen a la realidad de las aulas, propiciando que se desenvuelva de manera autónoma y construya una responsabilidad real a partir de las situaciones que se les presentan.

Zabalza y Cid (1998) señalan que el papel del tutor y del asesorado va mucho más allá del apoyo para planificar o de la mera revisión de la práctica, su rol consiste en que el alumno se apropie de las bases conceptuales y coadyuve en el proceso de formalizar las competencias que tienen que ver con el ejercicio profesional. Es decir que además de acompañar, sea un guía y un modelo en su proceso de formación, para que así el estudiante pueda afianzar más las bases teóricas en referencia con las prácticas.

Aguilera (2015) menciona que: “del espacio y tiempo donde confluyen el tutor y tutorado, de su relación y roles, es decir de su relación formativa, emergen ciertos saberes, tradiciones y valores que van conformando los rasgos identitarios y conocimiento de los maestros” (p.6), lo que apoya a lo antes mencionado, donde se destaca el papel del educador en la formación de los alumnos, para la construcción de su identidad docente.

La Revolución cubana, ha dedicado una parte importante de sus recursos a la informatización de diferentes esferas de la sociedad, siendo una de ellas el sector de la educación, controlado a la vez por el Ministerio de Educación (MINED). Para el MINED el sistema de formación práctico-docente constituye vía por excelencia para garantizar la vinculación de la teoría con la práctica y la preparación integral de los futuros educadores, así como para elevar la motivación de los estudiantes por su profesión. Debe orientarse a la formación en valores del estudiante, a la asimilación y consolidación de conocimientos y habilidades profesionales que propicien la solución creativa de las tareas del ejercicio de la profesión y al cumplimiento de forma responsable, de la disciplina laboral y de las funciones que desarrolla el personal de una institución educativa.

La práctica está centrada en los problemas profesionales que enfrenta el estudiante en su contexto laboral y social y las tareas están diseñadas para que sean desarrolladas en los escenarios escolar, familiar y comunitario, concretando así el vínculo con la realidad social del país y de cada territorio en particular. La formación práctico-docente constituye la columna vertebral del plan de estudio, alrededor de la cual se integran todas las asignaturas y los diferentes procesos que tienen lugar en las instituciones. En este sector se han desarrollado diversas soluciones informáticas con apoyo de

otras instituciones como CINESOFT, DESOFT o la Universidad de las Ciencias Informáticas (UCI) con el objetivo de facilitar el trabajo y minimizar el tiempo de determinadas tareas dentro del proceso práctico docente.

La Universidad de ciencias informáticas UCI, se dedica al desarrollo de software, en la misma confluyen varios centros entre los que se encuentra el centro FORTES que se dedica entre otras cosas al desarrollo de aplicaciones de apoyo a la docencia.

Una de las necesidades que presenta el MINED es gestionar la información generada en la práctica docente de los estudiantes y profesores que, así como de los profesores de la tarea “Educando por Amor” y alumnos ayudantes de las distintas universidades que imparten su clases en los distintos niveles del sistema educativo cubano (primaria, secundaria y preuniversitaria), tal como las asignaturas que imparten, los años de práctica, las formas de evaluación, entre otros aspectos que resulten de interés a las instituciones de este sector.

Actualmente los documentos asociados al proceso se gestionan de forma manual mediante la utilización de herramientas ofimáticas, provocando que muchas veces pueda perderse o deteriorarse, debido a la poca fiabilidad en el archivado de esta documentación. Este tipo de práctica dificulta el mantenimiento en el tiempo de la información histórica, No propicia el análisis de los datos para la obtención de conocimientos sobre el actuar de los profesionales. Dificulta la correcta ejecución del proceso de la práctica docente, que puede repercutir en la calidad de las actividades en ejecución. Además, limita la correcta gestión de la información por parte de los profesores, la toma de decisiones y el uso efectivo de los recursos humanos en función de un mayor aprovechamiento.

Partiendo de la anterior situación problemática se plantea el siguiente **problema de investigación**: ¿Cómo mejorar la gestión de la información de los profesores y estudiantes en el proceso de práctica docente del MINED?

Para el desarrollo de la investigación se enmarca como **objeto de estudio** el proceso de gestión de la práctica docente en el Ministerio de Educación y se delimita como **campo de acción** a la gestión de la información de la práctica docente en el Ministerio de Educación.

Se define como **objetivo general** desarrollar un módulo para el Sistema de Gestión Académica que gestione la información de la práctica docente de los profesores en el Ministerio de Educación.

Se tuvieron en cuenta para el desarrollo de la presente investigación las siguientes **preguntas científicas**:

- ¿Cuáles son los referentes teórico-metodológicos a tener en cuenta para abordar la solución del problema planteado relacionado con la gestión de la práctica docente?
- ¿Qué propuesta de solución se define para desarrollar un módulo de gestión para de la práctica docente?
- ¿Cuáles son las características que debe cumplir el módulo de gestión de práctica docente para el Sistema de Gestión Académica del Ministerio de Educación?
- ¿Cómo se valida el funcionamiento del módulo de gestión de la práctica docente para el Sistema de Gestión Académica del Ministerio de Educación?

Para la realización del presente trabajo el objetivo general antes mencionado se divide en las siguientes tareas de investigación:

- Caracterizar los fundamentos teórico-metodológicos que permitan el desarrollo de la solución propuesta
- Diseñar la propuesta de solución teniendo en cuenta la definición del proceso a informatizar y la metodología de desarrollo de software definida para la investigación.
- Validar funcionalmente la solución desarrollada a partir de las pruebas de software.

Los **métodos científicos de investigación** utilizados para la realización de la investigación son los siguientes:

- **Métodos teóricos:**
 - **Analítico-sintético:** se utilizó este método pues permite dividir el fenómeno a estudiar y unir las partes previamente analizadas. Se analizaron los aspectos más importantes para llevar a cabo la realización de este módulo. Se tuvo en cuenta para esto, los aspectos más importantes para llevar a cabo este módulo para el Ministerio de Educación y de ahí se sintetizó como realizar la propuesta planteada
 - **Histórico-lógico:** se evidencia en el estudio de los sistemas de gestión de información de alumnos ayudantes existente en la Universidad de Ciencias Informáticas, así como en el análisis de sus características y objetivos, permitiendo entender su funcionamiento y definir sus objetivos.
 - **Modelado:** Se aplica con el objetivo de ayudar a comprender las teorías relacionadas con la gestión del proceso de práctica docente que llevan a cabo los estudiantes y profesores de las diferentes instituciones. También se ha utilizado para modelar todos los diagramas

correspondientes a la etapa de análisis, diseño e implementación del módulo a desarrollar en la presente investigación.

- **Métodos empíricos:**

- **Entrevista:** se utiliza para lograr un mejor entendimiento sobre las características y distintas etapas en el funcionamiento del proceso de gestión de la práctica docente llevado a cabo en el Ministerio de Educación.
- **Observación:** Se vio aplicado en el diagnóstico del problema que se investiga, a fin de obtener los elementos necesarios para desarrollar el módulo para la gestión de la práctica docente del Sistema de Gestión Académica para el Ministerio de Educación.
- **Documental:** Se utilizó para consultar bibliografía en fuentes de carácter documental tales como artículos, revistas científicas, libros y documentales.

El documento de investigación está estructurado de la siguiente forma:

Capítulo 1. Fundamentos teóricos del Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación: Este capítulo abarca la fundamentación teórica que sustenta el desarrollo de la investigación. Se profundiza en temas relacionados como: marco conceptual del objeto de estudio y el campo de acción, características y funciones asociados a los sistemas de gestión de la práctica docente y la definición del ambiente de desarrollo.

Capítulo 2. Análisis y diseño de la propuesta del módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación Se describe la propuesta de solución enmarcada en el proceso actual de gestión de la información de la práctica docente existente en el Ministerio de Educación. Se realiza un análisis del proceso que conforma el negocio y se presenta una descripción detallada de los requisitos con los que debe cumplir la propuesta de solución, así como los elementos del diseño y la arquitectura identificadas durante el estudio de artefactos obtenidos en los flujos de trabajos: Requisitos, Análisis y Diseño del Sistema.

Capítulo 3. Implementación y pruebas del módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación: Se detalla la construcción de la solución propuesta teniendo en cuenta las técnicas de programación, estrategias de codificación y la integración con otras aplicaciones, este capítulo se muestran los resultados que permiten validar la propuesta desarrollada.

Capítulo 1: Fundamentos teóricos del Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación

El fundamento teórico es una de las partes más importantes en el desarrollo de un trabajo de diploma pues propicia un marco de referencia para la adecuada interpretación de los resultados que se recogen. En el presente capítulo se analizan y exponen un conjunto de enfoques teóricos e investigativos que han sido objeto de publicación en el área del conocimiento donde se ubica la investigación de este trabajo. Además, se realiza un estudio de los diferentes conceptos, objetivos y características fundamentales asociados a el módulo que se estará desarrollando.

1.1. Marco conceptual de la investigación

Para alcanzar una mejor comprensión y ampliación de las diferentes temáticas que se estarán desarrollando en el presente trabajo de diploma, se mostrarán un grupo de términos identificados durante la investigación realizada que corresponde al módulo de práctica docente que pertenecerá al Ministerio de Educación.

1.1.1. Gestión de la enseñanza en Cuba

Como parte de las medidas organizativas que permiten atender la diversidad de intereses y posibilidades en el campo educativo, se cuenta con niveles de enseñanza bien diferenciados en sus objetivos específicos y metodología, pero comunes en su empeño de proporcionar instrucción y cultura a todos los ciudadanos.

La Educación Primaria es de carácter obligatorio y universal y tiene como principio enseñar, atender y educar, por igual, a todos los niños comprendidos en el grupo de edades de 6 a 11 años y garantizar su formación integral. La meta superior es elevar la calidad en el proceso de enseñanza-aprendizaje, de ahí que esté en un proceso continuo de mejoramiento, a partir de la aplicación de un nuevo modelo pedagógico (García-Ruiz & Sánchez, 2006).

Cuba cuenta con una red de escuelas primarias distribuidas por todo el territorio nacional, en el sector urbano y en el rural, incluyendo multigrados en las zonas más intrincadas del país, de forma que se abarca a la totalidad de los niños en edad escolar. Los estudios de Secundaria Básica, también de carácter obligatorio y universal, agrupan a los alumnos comprendidos entre los 12 y 14 años de edad, y se realizan en dos tipos de centros: las escuelas secundarias básicas urbanas (ESBU) y las

escuelas secundarias básicas en el campo (ESBEC); estas últimas con régimen interno (García-Ruiz & Sánchez, 2006).

También existen las escuelas vocacionales de Arte (EVA), y las escuelas de iniciación deportiva (EIDE). Su objetivo es sentar las bases del desarrollo integral de la personalidad del adolescente, proporcionando la adquisición de conocimientos básicos y el desarrollo de habilidades propias de la Lengua Materna, la Matemática, las Ciencias Naturales y Sociales, el conocimiento del idioma inglés, además de contribuir a la educación estética de los alumnos, la formación de una cultura física y al dominio elemental de los principios de la técnica, estrechamente vinculados con la actividad productiva.

La Enseñanza Pre Universitaria se distribuye en Institutos Preuniversitarios cuyos objetivos y fines responden a diferentes perfiles de orientación vocacional y cuya matrícula es el resultado de un proceso de selección en correspondencia con las aspiraciones y aptitudes profesionales de los educandos. Así es posible identificar, desde los Pre Universitarios en el Campo, Urbanos, Vocacionales de Ciencias Exactas, Vocacionales Pedagógicos, de Perfeccionamiento Atlético, de Iniciación Artística, hasta los militares. En todos ellos, se trata de brindar orientación vocacional para motivar la futura incorporación a las aulas universitarias, si bien es una tarea que adolece de gran disparidad en su ejecución y por ende en sus resultados (Furió, Vilches & Romo, 2001).

Estos niveles de enseñanza se complementan con un bien estructurado sistema de Educación Técnica y Profesional, que atiende la formación de técnicos de nivel medio y obreros altamente calificados, además de los que se preparan en oficios propios de la demanda territorial. Se cuenta, además, con las escuelas de Educación Especial cuya matrícula abarca el 100% de los necesitados de la misma, y su fin fundamental es que los niños, adolescentes y jóvenes que son atendidos en las diferentes instituciones creadas para este fin, puedan acceder a niveles superiores de desarrollo dentro de sus posibilidades reales, para lograr una adecuada incorporación a la vida social y laboral (Iglesias, Lozano & Pastor, 2011).

El sistema cierra con la Educación de Adultos que recorre todos los niveles de enseñanza. En el curso 2001-2002, se introdujo en esta enseñanza un nuevo programa dirigido a jóvenes de 18 a 29 años desvinculados del estudio y del trabajo. El impacto social de este programa en la familia, la comunidad y en los propios estudiantes ha sido notable por los cambios experimentados por los jóvenes, y aporta un nuevo concepto en la literatura pedagógica, a saber, el estudio como empleo, que ha beneficiado especialmente a los trabajadores del sector azucarero (Sacasas, 2013).

Para referirse al empleo de las TIC en el sistema, hay que señalar que, desde la enseñanza primaria, hasta el preuniversitario, se hace un uso sistemático de las Teleclases, las Video clases, y la Computación. Y que, además, hay un programa de enseñanza de la computación a toda la población que desee aprender, a través de locales habilitados con el equipamiento necesario en cada municipio y localidades con alta población (Estepa, et al., 2004; Cedeño & Figueroa, 2013).

1.1.2. Práctica docente

La práctica docente es el estudio de la didáctica, y el espacio empírico para contrastar las teorías sobre las prácticas de aula y a su vez, corresponde realizar desde ellas una permanente reflexión. El docente se presenta como un profesional con dominio de saberes complejos capacitado para tomar decisiones en un contexto concreto a través de distintos elementos en su práctica docente. Una práctica, no se limita a su dimensión operatoria y ésta tampoco se limita a la obtención de un resultado, sino que supone igualmente la identificación de elementos que conforman la producción de ese resultado, el proceso de transformación de lo real. (García Cabrero, Loredo Enríquez, and Carranza Peña, 2008)

La práctica profesional se entiende como el conjunto de acciones, estrategias e intenciones, que un sujeto pone en juego para intervenir y transformar su realidad (SEP, 2012a, p. 7); es en ella donde los estudiantes ponen en juego sus creencias, conocimientos, métodos e ideas acerca de lo que es y hace el docente. Parte de esto es transmitido de los maestros con los que han convivido y tenido relación a lo largo de su formación.

La Práctica Profesional del Docente, constituye un proceso complejo en el que confluyen múltiples factores que van a incidir en la concreción de teorías, lineamientos, políticas; es decir, en el logro de los fines educativos. En tal sentido, el docente como centro del proceso de la Práctica, debe lidiar con múltiples y simultáneos elementos en su hacer pedagógico. Por una parte, debe demostrar dominio conceptual sobre los contenidos de cada una de las áreas curriculares del nivel donde se desempeñará; liderazgo, traducido en autoridad moral y cognitiva que genere el trabajo cooperativo; el respeto hacia los otros y disposición para la toma de decisiones conjunta. Al mismo tiempo, debe evidenciar el dominio de estrategias, técnicas y habilidades que favorezcan los procesos de enseñanza, aprendizaje y evaluación; así como un desempeño personal y profesional sustentado en valores éticos y morales (Insuasty & Castillo, 2011).

Del docente se espera que evidencie actualidad didáctica, comunicación asertiva, creatividad, capacidad para reflexionar sobre su hacer y constituirse en aprendiz permanente, por lo que debe ser

un investigador de su propia acción, de tal manera que pueda generar transformaciones en la realidad en la que intervenga. Además, debe controlar sus emociones, sentimientos y afectos de tal forma que pueda equilibrar la subjetividad e intersubjetividad propia de la dinámica del aula y de la escuela (Cuevas & Uribe, 2013).

1.1.3. Sistemas de Gestión Académica

Se define como Sistema de Gestión Académica a una herramienta que gestiona los procesos de formación docente, las funcionalidades que incluye permiten realizar las siguientes tareas:

- Gestión de estudiantes y estructuras.
- Control y desarrollo del proceso docente, lo que incluye asistencia y evaluaciones
- Registro de las asignaturas por cada disciplina dentro de una carrera.

(Cabello Landeaux, 2016)

Un sistema de gestión es un grupo de varias reglas y principios relacionados entre sí de forma ordenada, para contribuir a la gestión de procesos específicos o generales de una organización. Permite establecer una política, unos objetivos y alcanzar dichos objetivos (Passoni, 2005).

1.1.4. Tecnologías de la información en el MINED

La Dirección de Tecnología Educativa del MINED (DTE) creada en octubre del 2003, es una entidad que tiene como objeto social el desarrollo de servicios telemáticos y la creación de recursos metodológicos para el proceso de enseñanza aprendizaje. Sustenta sus actividades a través del portal CUBAEDUCA y de la Red Informática del Ministerio de Educación (RIMED), que está destinada a brindar servicios telemáticos a escuelas, estudiantes, profesores y entidades educativas del país.

El uso de recursos informáticos y audiovisuales de carácter educativo, desde una visión tecnológica, pedagógica, metodológica, científica e investigativa y de gestión de información y del conocimiento, de apoyo al proceso docente educativo en todos los niveles de enseñanza y la formación inicial y permanente de los docentes, bajo los preceptos de la estrategia nacional de informatización de la sociedad cubana, constituye uno de los objetivos de la Dirección de Tecnología Educativa.

En este sentido se evidencia la carencia de software de gestión para automatizar los procesos sustantivos de las instituciones, organismos y estructura del MINED, siendo el enfoque actual las

aplicaciones del proceso enseñanza-aprendizaje. Por tal motivo se desarrolla un análisis del estado del arte enfocado a las herramientas, plataformas, software y aplicaciones de gestión.

1.2. Sistemas homólogos para la informatización de la Práctica Docente

Como parte de la investigación realizada se caracterizaron varios sistemas informáticos que brindan un conjunto de funcionalidades relacionadas con la práctica docente en instituciones educacionales con el objetivo de analizar el proceso en sí y la gestión de su información. A continuación, se mencionan algunos de estos:

1.2.1. Soluciones en el entorno internacional

Portal de Facultad de Medicina de la Universidad Chile

El Portal de la Facultad de Medicina de la Universidad de Chile gestiona las actividades docentes del centro. Comprende distintas áreas como son la de pregrado, postgrado, investigaciones y extensión. Cuenta con una sección de alumnos ayudantes que contiene distintos documentos (resoluciones y normas) de tipo informativos, brindando la posibilidad de descargarlos para que los estudiantes soliciten, cambien o se reincorporen a la ayudantía. Los tutores pueden acceder al modelo de evaluación, planilla disponible para su confección y su posterior entrega al finalizar el año académico. Dicho sistema brinda datos sobre los tipos de alumnos ayudantes, sus funciones y un listado de los alumnos ayudantes del presente curso académico. (Goic, 2003)

Sistema Académico para Facultades. Universidad Nacional de Asunción. Paraguay.

El objetivo principal del sistema es proveer a la universidad de mecanismos automatizados que faciliten la planificación, organización, gestión y control académico. La arquitectura del sistema está pensada para su uso en un entorno multiusuario dando alcance a los diferentes sectores que componen la organización de la universidad. Entre los objetivos del sistema está registrar, controlar y emitir los resultados de cursos y exámenes de las asignaturas curriculares de todas las carreras de la universidad y apoyar la evaluación de las actividades educativas desarrolladas por la universidad mediante varios informes estadísticos. Cuenta con varias funcionalidades y módulos, dentro de ellos está el módulo de Plan de estudio, que constituye la estructura de información principal del sistema, en base a él giran gran parte de los controles que se realizan. Cuenta con los módulos “Estudiantes” y “Docentes” los cuales se encargan de la gestión del personal docente y los estudiantes respectivamente (Soto, 2015).

DSR Gestión del Personal

Es un software desarrollado por *Dataproces Soluciones Registradas (DSR)* que proporciona medios fáciles y flexibles para organizar y controlar los datos del empleado. En su base de datos la información almacenada en cada una de las fichas de empleado contiene, tanto los datos personales del empleado como varios códigos de estados administrativos. Entre estos códigos existe uno para contratos de empleo, que puede utilizar para asignar niveles o contratos de empleo estandarizados (DSR, 2019). El sistema realiza el proceso de selección e integración de personal definiendo el perfil a seleccionar como primer paso, luego se recogen todos los currículums del personal que solicita el puesto de trabajo y se realiza una valoración automática de las calificaciones de los aspirantes al puesto. Este proceso, no sólo puede realizar selecciones de personal externo de la empresa, sino que también puede buscarse entre los empleados de la misma, bien porque haya solicitado participar o bien porque el proceso de selección los haya considerado adecuados. Al finalizar el proceso de selección con la elección del candidato a entrar en la empresa, se pasan automáticamente sus datos de la ficha de aspirante a la de empleado (DSR, 2019).

1.2.2. Software desarrollado en Cuba

Akademos

La Universidad de las Ciencias Informáticas (UCI) cuenta con una herramienta informática llamada Sistema de Gestión Universitaria (AKADEMOS) que da solución a la mayoría de los problemas existentes en el MINED, sin embargo, este sistema fue desarrollado solo para la educación superior. Es un sistema Web que brinda una interfaz común para todos sus usuarios. Además, ofrece la información actualizada en tiempo real de las facultades de la universidad, es una herramienta que mantiene informado a los estudiantes sobre su desempeño académico, además de llevar el control del desarrollo del proceso docente, notas, asistencia, lleva a cabo la gestión de la práctica docente de los alumnos ayudantes que pertenecen a la universidad, siendo este el principal módulo a investigar y analizar dentro del sistema AKADEMOS que presenta la Universidad de Ciencias Informáticas. (Cabello Landeaux, 2016)

Sigenu

El Sistema de Gestión de la Nueva Universidad (SIGENU) es una herramienta que se desarrolla con la finalidad de gestionar toda la información académica que se vincula con la educación superior en Cuba. En correspondencia con su carácter nacional y la diversidad de sistemas de la enseñanza

superior con que cuenta la universidad cubana, este se concibe de manera que brinda gran seguridad e integridad de la información, y a la vez, es tan flexible que permita ser adaptado a todos los centros de educación superior del país con sus diversas particularidades y distintas maneras de realizar determinados procedimientos. (Castillo et al., 2015)

Gestacad

Es una aplicación web que permite organizar y sistematizar el proceso de ajuste de plan de estudio en la Universidad de Matanzas, de forma tal que la experiencia de los casos existentes puede ser aprovechada y se logra una adecuada y más rápida toma de decisiones. Este sistema permite obtener una alternativa basada en la web, aptada al contexto de la Universidad de Matanzas, lo que permitirá facilitar el ajuste de planes de estudio para estudiantes que cambian el mismo. (Mondéjar, 2015)

1.2.3. Análisis de los sistemas descritos

Durante el estudio de los diversos sistemas homólogos existentes que manejan información asociada a los procesos de práctica docente, se pudo constatar que poseen una variedad de funcionalidades y en dependencia del área donde son aplicados. La aplicación SIGENU, aunque gestiona información de la práctica docente. No provee una plataforma para desarrollar los procesos de solicitudes y propuestas de ingreso. Tampoco contempla el sistema de evaluación del desempeño de los alumnos que realizan sus prácticas docentes, así como la confección de su plan de trabajo. El Portal de la Facultad de Medicina de la Universidad de Chile, aunque es posible su uso tecnológicamente, es de carácter informativo, Dada la incompatibilidad de las tecnologías es imposible de migrar, dado también a que este software es de carácter privativo.

En el caso de AKADEMOS se puede decir que es muy similar a lo que se desea desarrollar en este módulo, pero posee algunos inconvenientes como el que fue desarrollado con una tecnología distinta a la que se necesita en este proyecto y no da solución a la situación problemática planteada a la introducción de este trabajo. Teniendo en cuenta los inconvenientes presentados por algunas de las herramientas antes mencionadas y sus aportes surge la presente investigación para desarrollar una solución informática integrada al Sistema de Gestión del MINED que se está llevando a cabo, esta solución gestionará la información de la práctica docente en función de elevar la calidad del proceso docente-educativo en la institución. De los sistemas estudiados ninguno cumple la totalidad de estas condiciones o características, no obstante, algunos de ellos aportan gran valor a la investigación y pudieran ser utilizadas ciertas herramientas, lenguajes y metodologías en la realización de la propuesta de solución.

1.3. Ambiente de desarrollo

Las herramientas son un punto importante en la elaboración de una aplicación, son los programas que se reutilizan para automatizar las actividades definidas en el proceso de desarrollo de software; permiten crear y darle soporte al mismo, muchas veces haciendo el trabajo más factible y sencillo. En este epígrafe se especifican las herramientas, tecnologías y lenguajes definidas por el proyecto Sistema de Gestión Académica y utilizadas para el desarrollo de la propuesta de solución.

1.3.1. Herramientas informáticas

Angular 8

Es un framework para aplicaciones web de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Angular se puede combinar con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador, en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles. Se utilizará Angular para el fronted(interfaz). (Han et al. 2002)

Herramienta de modelado: Visual Paradigm para UML 15.0

Es una suite de aplicaciones, herramienta profesional que soporta el ciclo completo de desarrollo del software a través de la representación de varios tipos de diagramas. Propicia un conjunto de ayudas para el desarrollo de programas informáticos, incluyendo las fases de planificación, el análisis, diseño, la generación del código fuente de los programas y la documentación. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos. (Visual Paradigm, 2009)

Servidor de aplicaciones web Nginx 1.14

Es un servidor web ligero de código abierto. Es multiplataforma, por lo que corre en sistemas tipo Unix, GNU/Linux, BSD, Solaris, MacOS, Windows, entre otros. Conocido por su alto rendimiento, estabilidad, configuración simple y bajo consumo de recursos. A diferencia de los servidores tradicionales, nginx no se basa en subprocesos para manejar las solicitudes. En su lugar, utiliza una arquitectura más escalable basada en eventos (asíncrona). Esta arquitectura utiliza cantidades pequeñas, pero más importantes, predecibles de memoria bajo carga. Nginx fue inicialmente desarrollado con el fin explícito de superar el rendimiento ofrecido por el servidor web Apache.14. Como servidor de archivos estáticos, nginx usa menos memoria que Apache, y puede manejar más

solicitudes por segundo. Se decidió utilizar este servidor debido a que optimiza las consultas más rápido que otros cuando existe una gran cantidad de información.(Hurtado and Sarango 2017)

PgAdmin III

Es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. La aplicación se puede utilizar para manejar PostgreSQL 7.3 y superiores y funciona sobre casi todas las plataformas. Este software fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL7 a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración. La aplicación también incluye un editor de la sintaxis SQL, un editor de código del lado del servidor. La conexión del servidor se puede realizar mediante TCP/IP y puede ser cifrado mediante SSL por seguridad. No se requieren controladores adicionales para comunicarse con la base de datos del servidor.(González Collado and Valdés Marrero 2013)

PostgreSQL 10.4

Es un Sistema de Gestión de Bases de Datos Objeto-Relacional. Pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, PostgreSQL incluye características de la orientación a objetos como son: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, además de otras específicas del gestor, como lo son: un mejor soporte para sub-selects, triggers, vistas y procedimientos almacenados.(González Collado and Valdés Marrero 2013)

1.3.2. Lenguajes de desarrollo

CSS 3

Las hojas de estilo en cascada (Cascading Style Sheets, por sus siglas en inglés). CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características de presentación, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo.css, y reducir la complejidad y la repetición de código en la estructura del documento. Es un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. El W3C5 es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores. La idea que se

encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.(González Collado and Valdés Marrero 2013)

HTML 5

Lenguaje de marcado de hipertexto (*HyperText Markup Language*, por su nombre en inglés), predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con imágenes y otros objetos. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>); también puede describir, hasta un cierto punto, la apariencia de un documento; es el lenguaje de publicación de la *World Wide Web*. (*World Wide Web Consortium*, 1999)

Lenguaje de Modelado Unificado (UML) 2.5

Unified Model Language (UML, por sus siglas en inglés, traducido al español como Lenguaje de Modelado Unificado) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio y otros sistemas. Un modelo UML está compuesto por tres clases de bloques de construcción; los elementos que son abstracciones de cosas reales o ficticias (objetos, acciones, etc.) de los cuales existen cuatro tipos de elementos, estructurales, ambientales, grupales y de anotación; las relaciones que son las que relacionan los elementos entre sí, ejemplo: dependencia, asociación, generalización y comprensión; y los diagramas que son las colecciones de elementos con sus relaciones.(Rumbaugh, Booch, and Jacobson 2000)

PHP 7.2

El acrónimo de "*PHP: Hypertext Preprocessor*", es un lenguaje interpretado de "código abierto", de alto nivel. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. Está diseñado especialmente para desarrollo web. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los

sistemas operativos y plataformas sin costo alguno. Uno de sus inconvenientes es que es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código, pero no necesariamente impide que el código sea examinado.(González Collado and Valdés Marrero 2013)

1.4. Metodología de desarrollo de software

El desarrollo de software, es uno de los sectores tecnológicos más competitivos, sin embargo, ha tenido una evolución constante en lo que se refiere a las metodologías con el objetivo de mejorar, optimizar procesos y ofrecer una mejor calidad. Son un enfoque estructurado para el desarrollo de software que incluyen modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos (Modelos Y Metodologías Para El Desarrollo De Software, 2018). Estas metodologías se clasifican en dos grandes grupos: tradicionales y ágiles. Las metodologías tradicionales se caracterizan por:

- Documentación exhaustiva de todo el proyecto.
- Los costos son altos al implementar un cambio.
- No es una buena solución para entornos volátiles.
- El equipo de desarrollo debe ser grande.

Mientras las metodologías ágiles se definen por:

- Buena solución para proyectos a corto plazo.
- Facilidad de respuesta a cambios repentinos en el desarrollo.
- Equipos de desarrollo pequeños.
- Ciclos de desarrollo cortos.

1.4.1. Metodología AUP-UCI

En el presente trabajo de llevar a cabo un enfoque ágil debido a que permite la realización de aplicaciones basándose en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos organizados y multidisciplinarios.

Para el desarrollo de este trabajo se determinó usar la metodología "Proceso Unificado Ágil" (AUP por sus siglas en inglés). Al no existir una metodología de software universal, debido a que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide utilizar una variación de la

metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI, en específico del sistema en cuestión.

De las cuatro fases que propone AUP-UCI: Inicio, Elaboración, Construcción y Transición, se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola que es Ejecución y se agrega la fase de Cierre (Garay, 2016).

AUP propone siete disciplinas: Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno, por lo que se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también (Sánchez, 2014). A partir de que la disciplina de Modelado de negocio propone tres variantes a utilizar en los proyectos: casos de uso del negocio, descripción de proceso de negocio y modelo conceptual, y existen tres formas de encapsular los requisitos: casos de uso del sistema, historias de usuario y descripción de requisitos por proceso, surgen cuatro escenarios para modelar el sistema en los proyectos, quedando de la siguiente forma:

Escenario 1: proyectos que modelen el negocio con casos de uso del negocio solo pueden modelar el sistema con casos de uso del sistema.

Escenario 2: proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de uso del sistema.

Escenario 3: proyectos que modelen el negocio con descripción de proceso de negocio solo pueden modelar el sistema con descripción de requisitos por proceso.

Escenario 4: proyectos que no modelen negocio solo pueden modelar el sistema con HU.

En la presente investigación se utilizará AUP-UCI como metodología de desarrollo de software en su escenario 3 debido a las características del problema de investigación y su entorno de solución, el alcance del proyecto y cantidad de recursos humanos. Además, se tiene en cuenta la estandarización y experiencia productiva en la universidad aplicando esta metodología.

1.5. Conclusiones parciales

El proceso de gestión de información en la práctica docente en el Ministerio de Educación de la República de Cuba es complejo, dinámico y conlleva muchos factores a tener en cuenta.

El análisis de los sistemas existentes propició identificar las tendencias en el proceso de gestión de información en la práctica docente, sin embargo, estos no se adecuan a las características

funcionales y técnicas requeridas para su uso en instituciones cubanas.

Las herramientas, tecnologías y lenguajes definidos para desarrollar la solución se corresponden con las definidas para el proyecto AKADEMOS para facilitar su integración; además cumplen con las políticas de informatización del país sobre la soberanía tecnológica y el uso de software libre.

Capítulo 2: Análisis y diseño del Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación

Introducción

En este capítulo se abordan las características generales del sistema a desarrollar, primeramente, es vital el análisis del modelo de negocio para una mejor comprensión del entorno de la investigación. Luego se describe la propuesta de solución y se representa mediante un mapa conceptual. Se explican los requisitos funcionales y no funcionales, así como las técnicas para su obtención. Se describe la arquitectura, el diseño del sistema y los patrones utilizados. Se muestra el diagrama lógico de datos, las principales vistas del sistema.

2.1. Modelo de Negocio

El modelo del negocio describe cada proceso del negocio, especificando sus datos, actividades, roles y las reglas del mismo. Facilita un entendimiento entre clientes y desarrolladores y se enfoca en comprender los problemas actuales de la organización e identifica mejoras potenciales. (Pressman, 2005).

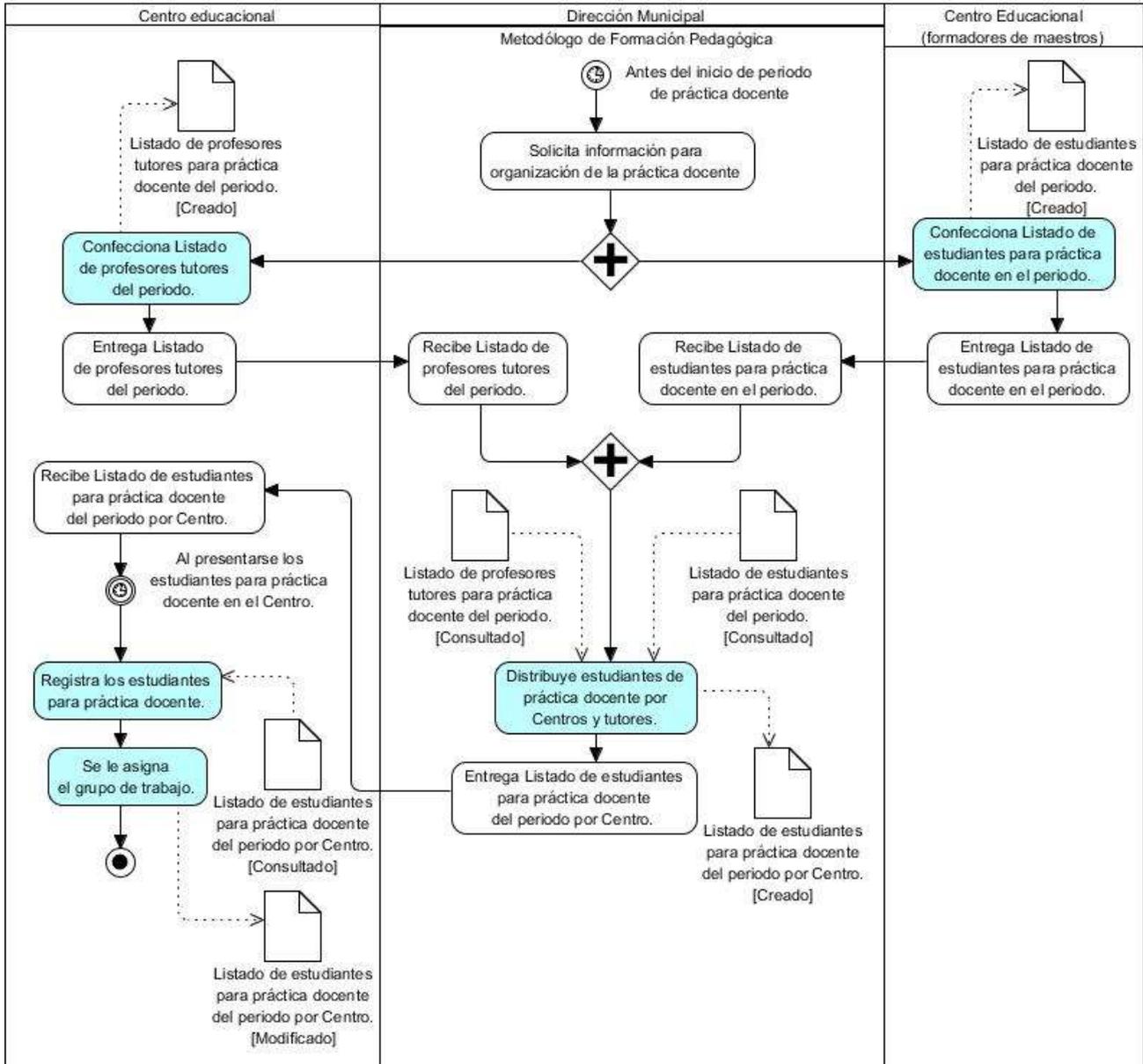
2.1.1. Identificación de roles del entorno del negocio

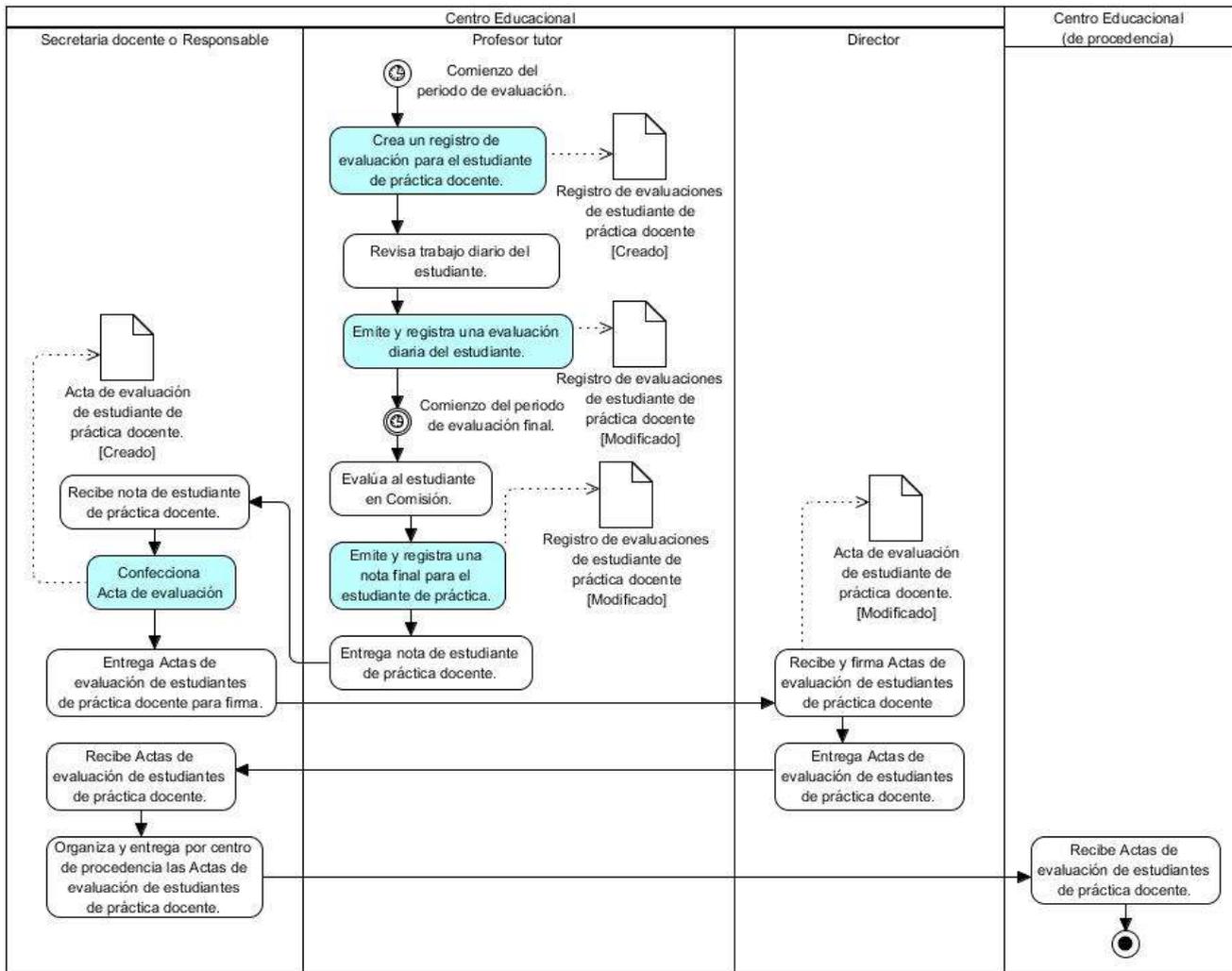
Una vez identificados los procesos de negocio, es preciso encontrar los involucrados en su realización. A continuación, se muestran los roles que se identificaron en la solución propuesta:

Tabla 1. Actores del negocio. Fuente: Elaboración propia.

Actores del negocio	Descripción
Secretaria docente	Gestiona y confecciona actas de evaluaciones de estudiantes en práctica docente.
Director	Firma y aprueba evaluaciones de estudiantes de práctica docente.

2.1.2. Diagrama del proceso del negocio





2.2. Requisitos del sistema

Los requisitos para un sistema de software determinan lo que debe hacer el sistema y definen las restricciones en su funcionamiento e implementación, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo. (Sommerville, 2005)

2.2.1. Técnicas para la obtención de requisitos

Para identificar y extraer los requisitos se utilizan técnicas que ofrece la ingeniería de requisitos, dentro de las que se encuentran: puntos de vista, entrevistas, escenarios, casos de uso, prototipos, entre otros. De estas técnicas se utilizaron en la investigación las siguientes:

Entrevista: La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o

descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Debe quedar claro que no basta con hacer preguntas para obtener toda la información necesaria. Es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista.

Análisis documental: Varios tipos de documentación, como manuales y reportes, pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones. La documentación difícilmente refleja la forma en que realmente se desarrollan las actividades, o donde se encuentra el poder de la toma de decisiones. Sin embargo, puede ser de gran importancia para introducir al analista al dominio de operación y el vocabulario que utiliza.

Prototipos: permite obtener una pequeña muestra de lo que sería el producto final. Se diseñaron un grupo de prototipos y se mostraron al cliente, para de esta forma conocer su opinión y rectificar algunos aspectos que desee modificar antes de llegar al producto final.

Además, se realizaron reuniones con la participación del cliente y representantes del proyecto donde se definieron aspectos que eran de gran importancia para la solución. Esto ayudó a definir algunas funcionalidades que requería la aplicación con el apoyo y críticas de los participantes en las reuniones.

El descubrimiento de requisitos es el proceso de recoger información sobre el sistema propuesto y los existentes, extraer los requisitos del usuario y del sistema de esta información.(Sommerville, 2005)

2.2.2. Requisitos funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Estos requisitos dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización al redactar el requisito. Los requisitos funcionales del sistema describen con detalle la función de este, sus entradas, salidas y excepciones.(Sommerville, 2005) Para el desarrollo del sistema se identificaron los siguientes requisitos funcionales:

Tabla 2. Descripción de requisitos funcionales. Fuente: Elaboración propia.

RF	Nombre	Complejidad	Prioridad
RF1	Listar periodos de práctica docente en curso académico.	Media	Baja
RF2	Registrar periodo de práctica docente en curso académico.	Media	Alta
RF3	Modificar periodo de práctica laboral en curso académico.	Baja	Baja
RF4	Eliminar periodo de práctica docente en curso académico.	Baja	Baja
RF5	Asociar tutor de estudiantes de práctica docente.	Media	Media
RF6	Generar reporte sobre tutores de estudiantes en práctica docente.	Media	Alta
RF7	Registrar tipos de evaluación para estudiantes de práctica docente.	Media	Alta
RF8	Registrar formas de evaluación para estudiantes de práctica docente.	Media	Media
RF9	Registrar estudiante de práctica docente.	Media	Media
RF10	Modificar estudiante de práctica docente	Baja	Media
RF11	Buscar estudiante de práctica docente	Baja	Baja
RF12	Filtrar estudiante de práctica docente	Baja	Baja
RF13	Listar estudiante de práctica docente	Media	Baja
RF14	Eliminar estudiante de práctica docente.	Baja	Baja
RF15	Asignar estudiante de práctica docente a centro educacional.	Media	Media
RF16	Eliminar estudiante de práctica docente a centro educacional.	Baja	Baja

RF17	Imprimir listado de estudiantes de práctica docente ubicados en centro educacional.	Baja	Baja
RF18	Registrar estudiante de práctica docente como estudiante de práctica del centro educacional.	Media	Media
RF19	Eliminar estudiante de práctica docente como estudiante de práctica del centro educacional.	Baja	Baja
RF20	Imprimir estudiantes de práctica docente como estudiante de práctica del centro educacional.	Media	Baja
RF21	Registrar evaluación del estudiante de práctica docente en un período.	Media	Baja
RF22	Modificar evaluación del estudiante de práctica docente en un período.	Media	Baja
RF23	Reporte de evaluaciones de los estudiantes de práctica docente por centro de procedencia	Media	Media
RF24	Reporte de evaluaciones de los estudiantes de práctica docente por centro de ubicación.	Media	Media
RF25	Reporte cuantitativo de estudiantes por centros educacionales, enseñanzas y asignaturas.	Media	Media

2.2.3. Requisitos no funcionales

Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Define las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. (Sommerville, 2005) El sistema se realizará bajo los siguientes requisitos no funcionales:

Tabla 3. Descripción de requisitos no funcionales. Fuente: Elaboración propia.

Usabilidad
RnF 1. La aplicación informática es una aplicación web integrada a un sistema gestión.
RnF 2. Las vistas del sistema deben indicar en cada momento la acción que se está realizando, así como los íconos deben estar representados por una imagen acorde a la acción que se realiza.
RnF 3. Para el uso del sistema se requiere como mínimo una PC cliente con el navegador web Mozilla Firefox 17
RnF 4. El sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo para usuarios sin experiencia y el acceso de manera rápida y efectiva a la información buscada.
RnF 5. El sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a mayor comprensión por parte del cliente de la herramienta de trabajo
Confiabilidad
RnF 6. Cuando ocurre una excepción el sistema mostrará un mensaje explicativo del error ocurrido y permanecerá en el mismo estado sin realizar ninguna otra operación.
RnF 7. Solo el personal autorizado para interactuar con la aplicación podrá utilizarla. Para lograr esto se necesitará de un nombre de usuario único y una contraseña definida anteriormente.
Eficiencia
RnF 8. El sistema debe tener un tiempo de respuesta corto.
RnF 9. El sistema debe soportar la conexión simultánea de todos los posibles usuarios.
Soporte
RnF 10. El sistema contará con un grupo de soporte y asesoría al cliente que proporcionará soporte al sistema siempre que sea necesario.

RnF 11. El sistema debe brindar como apoyo un manual de usuario, en el cual se refleja detalladamente la explicación de cada una de sus funcionalidades, brindándole al usuario una mayor experiencia del trabajo con el mismo.
RnF 12. El componente contará con toda la documentación definida en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.
RnF 12. Se precisa que la documentación del sistema este actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
Restricciones de diseño e implementación
RnF 13. El sistema debe cumplir con la arquitectura de información definida para el Sistema de Gestión Académica.
RnF 14. El sistema debe ser desarrollado con las herramientas y tecnologías definidas por el Centro de Formación para las Tecnologías.
RnF 15. El sistema debe ser desarrollado con herramientas y tecnologías de código abierto.

2.2.4. Descripción de requisitos

La descripción de los requisitos es el producto del trabajo final que genera la ingeniería de requisitos. Describe la función y el desempeño de un sistema basado en computadoras y las restricciones que regirán su desarrollo. (Pressman, 2005)

A continuación, se muestra la descripción del requisito funcional Listar periodos de práctica docente en curso académico:

Tabla 4. Descripción del RF: Listar periodos de práctica docente en curso académico. Fuente: Elaboración propia.

Precondiciones	Debe existir al menos un periodo de práctica docente creado.
Flujo de eventos	
Flujo básico Listar periodos de práctica docente en curso académico.	
1.	El requisito inicia cuando el actor accede a la sección Periodo de Práctica Docente.
2.	El sistema muestra un listado con los periodos de práctica docente existentes. Cada periodo de práctica docente muestra la siguiente información: <ul style="list-style-type: none"> • Dia, mes y año de inicio del periodo • Dia, mes y año de fin del periodo

	<ul style="list-style-type: none"> Se muestran opciones de cada periodo la cuales serán modificar o eliminar dicho periodo 																
Flujos alternativos																	
1.	N/A																
Pos-condiciones																	
1.	Se muestran un listado con los periodos de práctica docente existentes.																
Validaciones																	
1.	Marco conceptual de la investigación																
Conceptos	Práctica Docente	Marco conceptual de la investigación															
Restricciones del sistema	<ul style="list-style-type: none"> Las opciones Modificar periodo de práctica docente y Eliminar periodo de práctica docente solo aparecen si existen periodos de práctica docentes creadas anteriormente. 																
Requisitos especiales	N/A																
Asuntos pendientes	N/A																
Prototipo elemental de interfaz gráfica de usuario																	
 <p>The screenshot shows a window titled "Periodo de Práctica Docente". At the top right is a "Registrar" button. Below it is a search area with a text input field, a "Buscar" button, a dropdown menu, and a "Filtrar" button. The main area contains a table with the following data:</p> <table border="1"> <thead> <tr> <th>No</th> <th>Curso</th> <th>Inicio</th> <th>Fin</th> <th>Opciones</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2019-2020</td> <td>15 - 1 - 2019</td> <td>30 - 1 - 2019</td> <td>Modificar Eliminar</td> </tr> <tr> <td>2</td> <td>2019-2020</td> <td>1 - 2 - 2019</td> <td>15 - 2 - 2019</td> <td>Modificar Eliminar</td> </tr> </tbody> </table> <p>At the bottom of the window are "Cancelar" and "Aceptar" buttons.</p>			No	Curso	Inicio	Fin	Opciones	1	2019-2020	15 - 1 - 2019	30 - 1 - 2019	Modificar Eliminar	2	2019-2020	1 - 2 - 2019	15 - 2 - 2019	Modificar Eliminar
No	Curso	Inicio	Fin	Opciones													
1	2019-2020	15 - 1 - 2019	30 - 1 - 2019	Modificar Eliminar													
2	2019-2020	1 - 2 - 2019	15 - 2 - 2019	Modificar Eliminar													

2.3. Descripción de la arquitectura

La arquitectura utilizada para la propuesta de solución es la Arquitectura Cliente-Servidor, un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas. El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen, cosa que ocurriría si ellos mismos desempeñan las funciones que le son proporcionadas de forma directa y transparente. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes. (Provoste, 2019). La figura muestra la representación de la arquitectura:

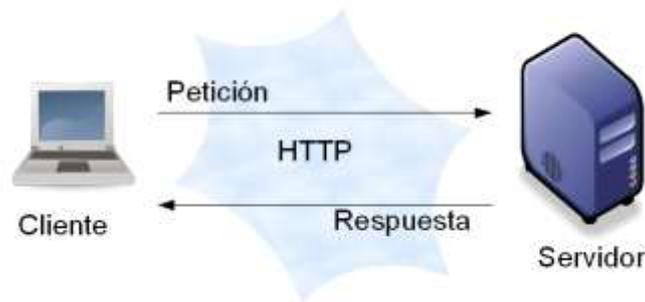


Figura 5. Arquitectura Cliente-Servidor

- Cliente:** es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador. La misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (Luján-Mora, 2001).
- Servidor:** es un programa que está esperando permanentemente las solicitudes de conexión por parte de los clientes web. La parte servidor de las aplicaciones web está formada por páginas estáticas que siempre muestran el mismo contenido y por programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas. La salida de este script suele ser una página HTML estándar que se envía al navegador del cliente (Luján-Mora, 2001).

2.3.1. Patrón arquitectónico

La arquitectura de la solución propuesta está basada en el patrón arquitectónico Modelo Vista Controlador (MVC). Este patrón garantiza la reducción del esfuerzo de programación, la cual es necesaria en la implementación de sistemas múltiples. Permite separar cada una de las capas del sistema: el modelo, donde se encuentran los datos y las reglas del negocio; la vista, que muestra la información del modelo al usuario; y el controlador, que gestiona las entradas del usuario.

A partir de la descripción realizada de las tecnologías en el anterior acápite, se evidencia la utilización del MVC de la siguiente manera:

- **El Modelo:** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **La Vista:** o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste.
- **El Controlador:** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Este modelo de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones API (*Application Programming Interface*) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.



Figura 6. Arquitectura MVC. Fuente: elaboración propia.

2.3.2. Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema. Esta solución provee una forma confiable, segura y probada para resolver problemas recurrentes en el diseño de software. Los patrones de diseño tienen como finalidad precisar en detalle los subsistemas y componentes de la aplicación. (Visconti y Astudillo, 2015)

Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. Ayudan a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento (Tabares, 2014). A continuación, se describen los patrones GRASP utilizados en el desarrollo de la propuesta de solución:

Experto: propone asignar una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso del patrón se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades

sobre los elementos de negocio. Un ejemplo del uso del patrón en la propuesta de solución es la librería encargada de la información de las licencias de maternidad, la que posee las responsabilidades relacionadas con la información de las licencias de maternidad.

Creador: expresa la asignación a una clase la responsabilidad de crear una instancia de otra. El uso del patrón se evidencia en la clase *Init* que se encarga de cargar los elementos del marco de trabajo dígame, librerías, modelos. En la propuesta de solución cada clase controladora crea una instancia de las librerías que necesita para su funcionamiento a partir de la clase *Init*. En la Figura 8 se pone de manifiesto en la clase *Init*, pues esta es la responsable de la creación (o instanciación) de nuevos objetos o clases.

Controlador: define que se le debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. El uso del patrón se evidencia en las clases controladoras que se encargan de obtener los datos y enviarlos a las librerías y las vistas, así como de manejar los posibles errores o mensajes que se muestran. En la Figura 9, el menú de la izquierda define cual es la clase la controladora de la propuesta de solución licencia de maternidad, dicha controladora es la que recibirá las peticiones asociadas a esta funcionalidad.

Alta cohesión: la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una clase con baja cohesión hace muchas tareas no afines o un trabajo excesivo.

Patrones GoF

Los patrones Banda de los Cuatro (*Gang-of-Four*) describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para ayudarnos a realizar tareas complejas. Existen tres tipos de patrones: de creación, estructurales y de comportamiento. Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas (Peña, 2016). Los patrones *GoF* que se utilizaron en el desarrollo del componente son:

Patrón de comportamiento: comprenden la asignación de responsabilidades entre objetos y algoritmos. Estos no solo conciernen a los objetos y las clases sino también la comunicación entre estas, además caracterizan flujos de control complejos que son difíciles de seguir en tiempo de

ejecución. (Romero y González, 2014)

Mediador: define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Los tipos de clases librerías implementadas en GUUD funcionan como mediadoras entre las clases controladoras y las modelos o acceso a datos.

Instancia única (*Singleton*): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto (Peña, 2016).

2.4. Diagrama de Clases del Diseño

El diagrama de clases del diseño especifica la estructura de clases del sistema con relaciones entre clases y estructuras de herencia. Es desarrollado buscando una solución ideal (Sellarès, 2011). A continuación, se muestra un fragmento del diagrama de clases del diseño del CU Gestionar evaluación de estudiantes en práctica docente.

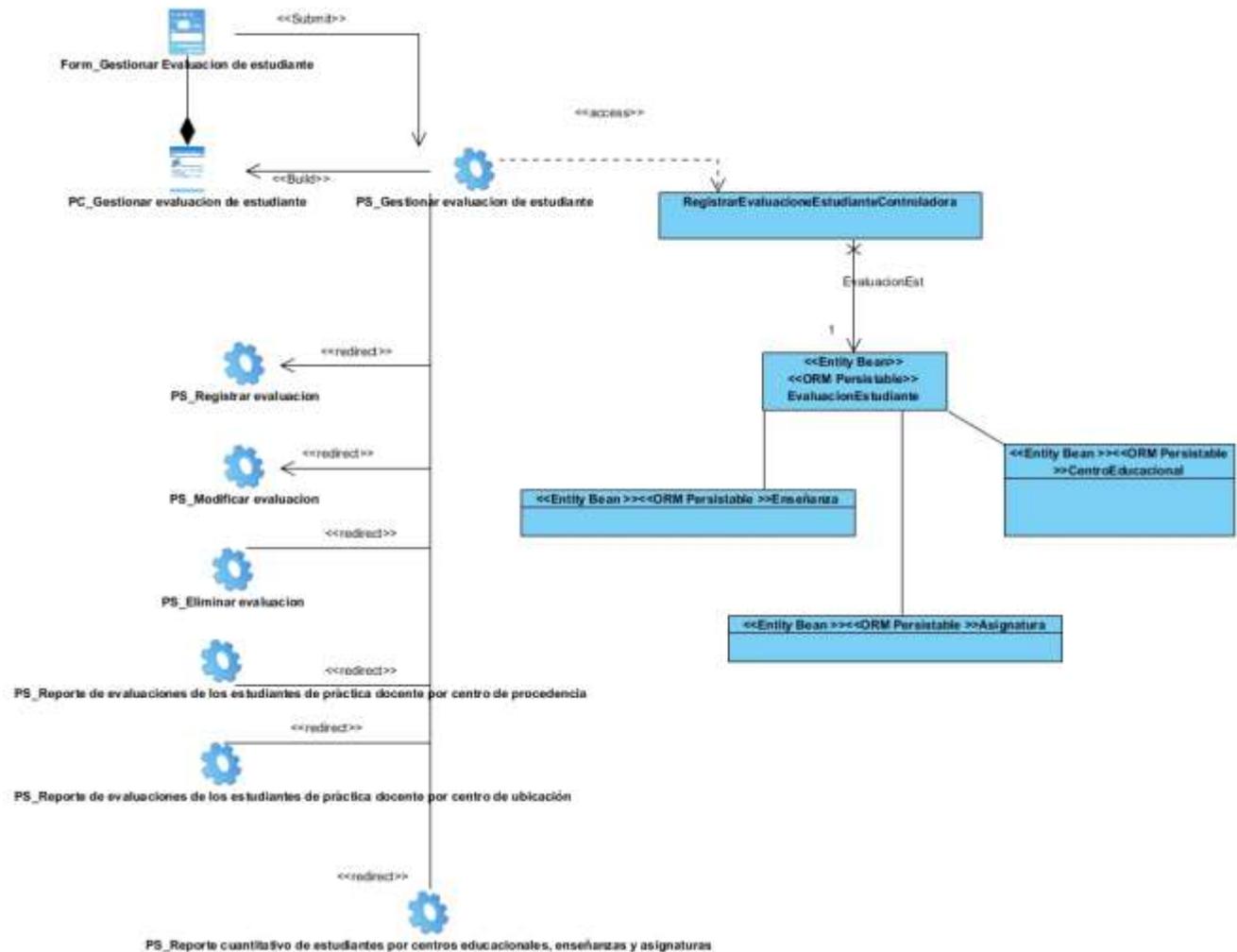


Figura. 11. Diagrama de clases del diseño del CU Gestionar evaluación de estudiantes en práctica docente. Fuente: elaboración propia.

2.5. Diagrama de paquetes

Para la confección del modelo del diseño, se plantea una estructura de paquetes que sea manejable para la implementación. Cada uno de estos paquetes está compuesto por diversos subpaquetes que a su vez contienen los diagramas de clases del diseño. Todas las clases están agrupadas en el paquete Repositorio de clases. En Sesiones se encuentran todas las clases controladoras agrupadas en paquetes, donde un paquete tiene las controladoras autogeneradas, otro las personalizaciones que se hacen sobre las controladoras autogeneradas y uno para las controladoras propias del proceso. El paquete Entidades contiene a su vez otros paquetes con las entidades autogeneradas y personalizadas. Todas las vistas están contenidas en el paquete Vistas. Estos paquetes se relacionan

entre ellos ya que las vistas consultan y actualizan las entidades e invocan a las controladoras y estas modifican las entidades.

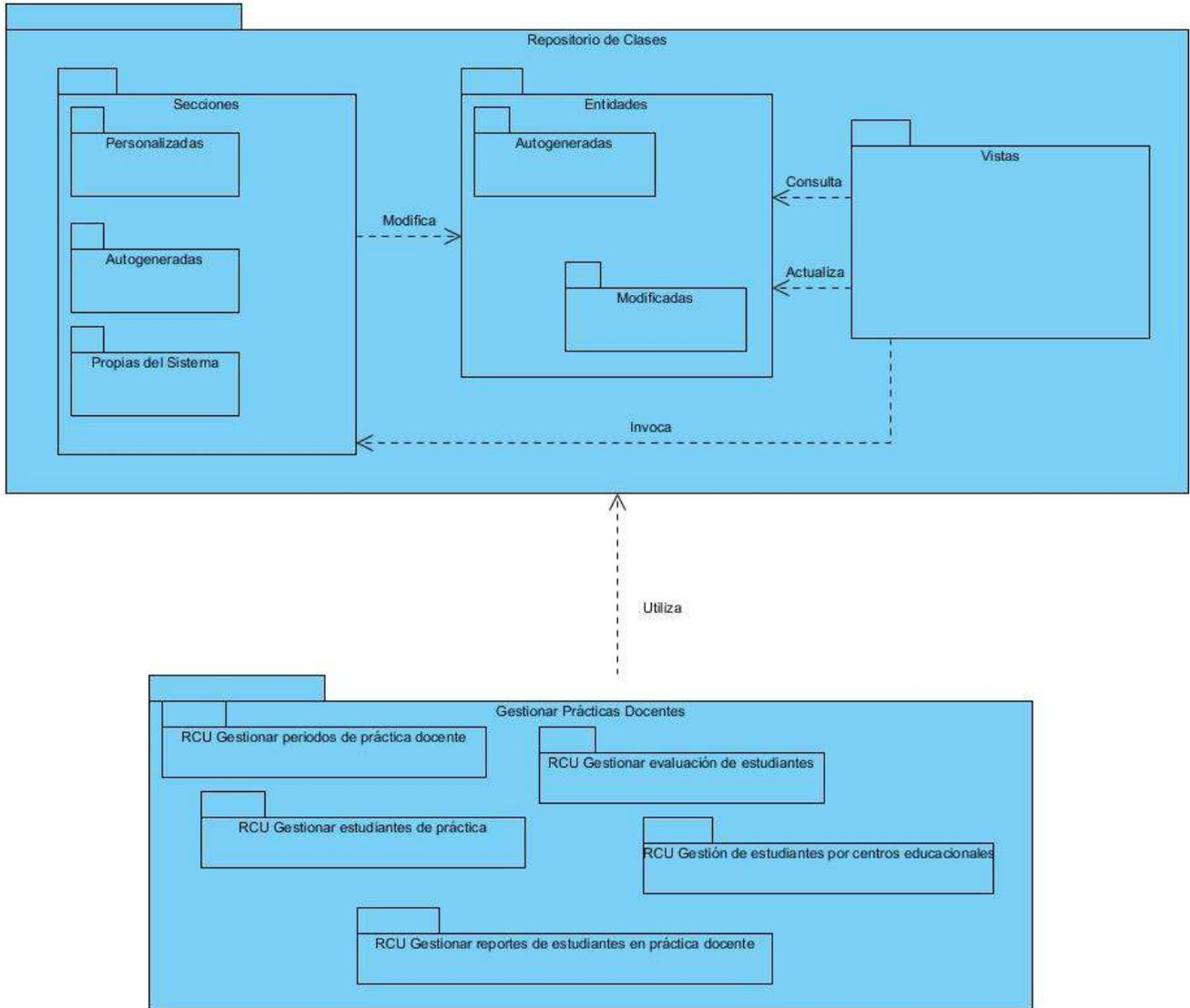


Figura. 14. Diagrama de paquetes. Fuente: Elaboración propia.

2.6. Conclusiones parciales

El proceso de gestión de información en la práctica docente en el Ministerio de Educación de la República de Cuba es complejo, dinámico y conlleva muchos factores a tener en cuenta.

El análisis de los sistemas existentes propició identificar las tendencias en el proceso de gestión de

información en la práctica docente, sin embargo, estos no se adecuan a las características funcionales y técnicas requeridas para su uso en instituciones cubanas.

Las herramientas, tecnologías y lenguajes definidos para desarrollar la solución se corresponden con las definidas para el proyecto AKADEMOS para facilitar su integración; además cumplen con las políticas de informatización del país sobre la soberanía tecnológica y el uso de software libre.

Capítulo 3: Implementación y validación del Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación

En el presente capítulo después del diseño realizado en el capítulo anterior, se presentan los estándares de codificación y los elementos de seguridad a tener en cuenta. Se presentan las pruebas correspondientes para la validación del buen funcionamiento de la aplicación a desarrollar y mostrar la calidad del resultado.

1.1 Estándares de codificación.

Los estándares de codificación son reglas que se aplican para lograr uniformidad en el código producido por un grupo de desarrollo de un sistema. Estos reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran, lo que permite obtener un código de alta calidad.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación, es de gran importancia para la calidad del software. La aplicación de estándares de codificación además posibilita que el software que se obtiene sea fácil de comprender y de mantener en el tiempo. El uso de los mismos tiene ventajas tales como:

- Asegurar la legibilidad del código entre distintos programadores, facilitando la depuración del mismo.
- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

A continuación, se presentan algunos de los estándares de codificación utilizados en la solución propuesta. Se debe utilizar como idioma el español, las palabras no se acentuarán. Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas. Se deben usar siempre dos líneas en blanco en las siguientes circunstancias:

- ✓ Entre las secciones de un fichero fuente.
- ✓ Entre las definiciones de clases e interfaces.

Se debe usar siempre una línea en blanco en las siguientes circunstancias: Entre métodos, entre las variables locales de un método y su primera sentencia. Además, antes de un comentario de bloque o de un comentario de una línea y entre las distintas secciones lógicas de un método para facilitar la

lectura. Se debe dar un espacio en blanco entre una palabra clave del lenguaje y un paréntesis.

Respecto a las normas de inicialización, declaración y colocación de variables, constantes, clases y métodos:

- ✓ Todas las instancias y variables de clases o métodos empezarán con minúscula. Las palabras internas que lo forman, si son compuestas, empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "_" o signo de peso "\$", aunque ambos están permitidos por el lenguaje.
- ✓ Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúscula. Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas.

Respecto a la indentación y longitud de la línea se deben emplear cuatro espacios como unidad de indentación. Los tabuladores deben ser exactamente cada 8 espacios. Y evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

Importaciones:

- Las importaciones deben estar en líneas separadas.
- Siempre deben colocarse al comienzo del archivo.
- Deben quedar agrupadas de la siguiente forma:
 - Importaciones de la librería estándar.
 - Importaciones terceras relacionadas.
 - Importaciones locales de la aplicación/librerías.
- Cada grupo de importaciones debe estar separado por una línea en blanco.

Comentarios:

- Los comentarios deben ser oraciones completas.
- Si un comentario es una frase u oración, su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.

Si un comentario es corto, el punto final puede omitirse.

1.2 Tratamiento de errores

Durante el tiempo de ejecución de un sistema pueden fracasar diferentes rutinas; es esto a lo que comúnmente se le llama excepción. Mediante el tratamiento de excepciones se restaura a un estado en el que la rutina pueda seguir la ejecución, lo que permite obtener un sistema robusto y fiable.

En el sistema propuesto, el control de las excepciones se lleva a cabo a toda porción de código donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos. También se controlan los errores en la validación de datos provenientes de la interfaz de usuario, puesto que encierran una lógica compleja en cierta medida.

Para el manejo de las excepciones o errores, en las clases controladoras de procesos, se utiliza el bloque *try* para detectar cuándo ocurra algún fallo y un bloque *catch* donde se manejarán dichas excepciones, mediante mensajes que se muestran en la interfaz de usuario.

Existe un archivo denominado *page.xml*, que engloba la configuración de todos los mensajes que se deben mostrar por cada tipo de excepción, así como la página a la que el sistema redirecciona en caso de la aparición de un error sorpresivo.

1.3 Pruebas internas

Durante esta etapa se prueban los componentes del producto con el objetivo de medir la calidad del software. El proceso de pruebas está encaminado a medir el cumplimiento de las funcionalidades establecidas por el cliente, reduciendo de esta manera el número de errores no detectados. Entre las pruebas internas se decide hacer pruebas unitarias, de caja blanca y de camino básico. También se emplea la herramienta *JUnit* que permite correr un conjunto de pruebas de forma automática e informa de aquellas que han fallado (Montes Ramírez et al. 2016).

Las pruebas unitarias o prueba de unidad enfocan los esfuerzos de verificación en la unidad más pequeña del diseño de software: el componente o módulo de software. Al usar la descripción del diseño de componente como guía, las rutas de control importantes se prueban para descubrir errores dentro de la frontera del módulo. La relativa complejidad de las pruebas y los errores que descubren están limitados por el ámbito restringido que se establece para la prueba de unidad. Las pruebas de unidad se enfocan en la lógica de procesamiento interno y de las estructuras de datos dentro de las fronteras de un componente. Este tipo de pruebas puede realizarse en paralelo para múltiples

componentes. (Sommerville 2015)

Las pruebas realizadas mediante el método de caja blanca se basan en el examen cercano de los detalles de procedimiento. Las rutas lógicas en el software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles. En ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. (Pressman 2010)

La prueba de camino básico permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. (Sommerville 2015) La complejidad ciclomática se aplica a todos los métodos de la herramienta, pero en la investigación se analizará la funcionalidad de evaluación del profesor. Al calcular la complejidad ciclomática a partir del grafo realizado del método en cuestión, se llegó a la conclusión de que se necesitan 3 casos de prueba para validar todas las posibles variantes del método. El algoritmo de evaluación del profesor posee poco riesgo debido a que el resultado arrojado por la métrica pertenece al intervalo entre 1-6.

Después de aplicar esta complejidad ciclomática se procede a la realización de la técnica de prueba automatizada, pues esto facilita identificar funciones que no ofrecen una salida acorde con la lógica que se deseaba implementar.

Una vez realizada la corrección de los errores detectados, se verifica si los métodos de cada clase se ejecutaron de forma correcta, y si obtienen los resultados esperados. Se toman precauciones para evitar errores similares y se definen nuevos casos de prueba para verificar el correcto desempeño de las funcionalidades.

También se emplean pruebas de caja negra para probar las funcionalidades tomando como unidad a cada interfaz. Aquí no se centra en cómo se generan las respuestas del sistema, solo se analizan los datos de entrada y los resultados obtenidos.

Este método contiene la técnica de partición de equivalencia que consiste en clasificar las entradas de datos del sistema en grupos que presentan un comportamiento similar, por lo cual serán procesados de la misma forma. Se pueden definir particiones tanto para valores válidos como inválidos.

1.4 Conclusiones parciales

Los estándares de codificación, el tratamiento de errores y los elementos de seguridad informática propiciarán la aplicación de buenas prácticas en el desarrollo de software y la obtención de un producto con calidad.

Las pruebas a realizar tanto al código como a las funcionalidades validan que el sistema funciona correctamente, determinado por los resultados obtenidos en su ejecución.

Conclusiones generales

Una vez concluida la presente investigación se puede afirmar que se desarrolló satisfactoriamente el análisis y diseño del Módulo de “Práctica Docente” para el Sistema de Gestión Académica del Ministerio de Educación. Durante su realización se arribaron a las siguientes conclusiones:

La elaboración del marco teórico metodológico permitió crear las bases del desarrollo de la investigación y constatar su novedad de la investigación.

Las herramientas, tecnologías y lenguajes de programación seleccionados permitirán obtener una solución robusta, con estándares de codificación y buenas prácticas de desarrollo, las cuales propician la extensibilidad de la aplicación.

El análisis y diseño aplicando la metodología definida permitieron identificar los requisitos funcionales y no funcionales, así como los artefactos ingenieriles que respaldan la solución propuesta, lo cual permitió cumplir con las expectativas del usuario.

A partir de la aplicación de pruebas definidas y descritas, la solución a las no conformidades detectadas y las recomendaciones realizadas durante el proceso se obtendrá una solución robusta.

Recomendaciones

Desarrollar un componente de análisis para la recomendación de forma automática, basada en los datos almacenados en el sistema, las mejores variantes de asignación de estudiantes y profesores en la práctica docente.

Referencias Bibliográficas

Achilli, E. L. (1988). La práctica docente: una interpretación desde los saberes del maestro. Cuadernos de Antropología social, (2), 5-18.

Cabello Landeaux, Roberto. 2016. "Módulo de Gestión de La Información Del Departamento de Docencia Y Capacitación Del Centro Nacional de Electromedicina."

Cañedo Ortiz, T. D. J., & Figueroa Rubalcava, A. E. (2013). La práctica docente en educación superior: una mirada hacia su complejidad. Sinéctica, (41), 2-18.

Castillo, Vanessa Danae Muñoz, Hilda García Barrios, Orlando Rubiera Hernández, Carlos Ramón López Paz, and Ingrid Wilford Rivera. 2015. "SIGENU-DSS-LITE: Nuevas Capacidades de Integración de Información Docente En Instituciones de Educación Superior En Cuba." Ciencias de La Información 46 (2): 3–8.

Cuevas, M. I. C., & Uribe, R. M. (2013). Caracterización de la práctica docente en ambientes virtuales de aprendizaje.

Estepa Murillo, P., Mayor Ruiz, C. M., Hernández de la Torre, E., Sánchez Moreno, M. R., Rodríguez López, J. M., Altopiedi, M., & Torres Gordillo, J. J. (2004). Las necesidades formativas docentes de los profesores universitarios. Revista Fuentes, 6, 74-95.

Furió, C., Vilches, A., Guisasola, J., & Romo, V. (2001). Finalidades de la enseñanza de las ciencias en la secundaria obligatoria. ¿ Alfabetización científica o preparación propedéutica?. Enseñanza de las Ciencias, 19(3), 365-376.

Gaitán, C., Martínez, D., Gaetán, S., Romero, J., Saavedra, E., & Alvarado, P. E. (2005). Caracterización de la práctica docente universitaria. Colombia (Tesis de pregrado). Pontificia Universidad Javeriana: Bogotá. Recuperada en: <https://bit.ly/35YAAAt9> (01/06/2016).

García Cabrero, Benilde, Javier Loredó Enríquez, and Guadalupe Carranza Peña. 2008. "Análisis de La Práctica Educativa de Los Docentes: Pensamiento, Interacción Y Reflexión." Revista Electrónica de Investigación Educativa 10 (SPE): 1–15.

García-Ruiz, M., & Sánchez Hernández, B. (2006). Las actitudes relacionadas con las ciencias naturales y sus repercusiones en la práctica docente de profesores de primaria. Perfiles educativos, 28(114), 61-89.

Goic, Alejandro. 2003. "Facultad de Medicina de La Universidad de Chile: 170 Años Al Servicio Del País." Revista Médica de Chile 131 (4): 355–58.

González Collado, Yarelis, and Israel Valdés Marrero. 2013. "Implementación Del Proceso de Culminación de Estudios de Pregrado Como Un Evento Dentro Del Sistema de Gestión Universitaria".

Han, Sangdon, D Ryan Anderson, Andrew D Bond, Hiufung V Chu, Raymond L Disch, Daniel Holmes, Jerome M Schulman, Simon J Teat, K Peter C Vollhardt, and Glenn D Whitener. 2002. "Total Syntheses of Angular [7]-, [8]-, and [9] Phenylene by Triple Cobalt-Catalyzed Cycloisomerization: Remarkably Flexible Heliphenes." *Angewandte Chemie International Edition* 41 (17): 3227–30.

Hurtado, Mario E Cueva, and Diego Javier Alvarado Sarango. 2017. "Análisis de Certificados SSL/TLS Gratuitos Y Su Implementación Como Mecanismo de Seguridad En Servidores de Aplicación." *Enfoque UTE* 8: 273–86.

Iglesias Martínez, M. J., Lozano Cabezas, I., & Pastor Verdú, F. R. (2011). Los discursos del profesorado novel universitario: Un estudio sobre su práctica docente.

Insuasty, E. A., & Castillo, L. C. Z. (2011). Caracterización de los procesos de retroalimentación en la práctica docente. *Revista Entornos*, (24), 73-86.

Mercado, R., & Rockwell, E. (1988). La práctica docente y la formación de maestros. *Revista Investigación en la Escuela*, 4, 65-78.

Mondéjar, Abel González. 2015. "Facultad de Ciencias Económicas E Informática."

Passoni, Lucía I. 2005. "Gestión Del Conocimiento: Una Aplicación En Departamentos Académicos." *Gestión Y Política Pública* 14 (1): 57–73.

Peña, Ayala A. 2016. *Ingeniería de Software: Una Guía para Crear Sistemas de Información*. México : s.n., 2016.

Pressman. 2005. *Modelo de Dominio*.

Rodríguez Sánchez, Tamara. 2015. "Metodología de Desarrollo Para La Actividad Productiva de La UCI."

Rumbaugh, James, Grady Booch, and Ivar Jacobson. 2000. *El Lenguaje Unificado de Modelado: Manual de Referencia*. Addison Wesley,.

Sacasas, J. Á. F. (2013). El principio rector de la Educación Médica cubana. Un reconocimiento a la doctrina pedagógica planteada por el profesor Fidel Ilizástigui Dupuy. *Revista Cubana de Educación Médica Superior*, 27(2), 239-248.

Sommerville. 2005. *Requisitos*.

Vega, Ignacio U González. 2011. "MÓDULO V."

World Wide Web Consortium. 1999. "HTML 4.01 Specification."

Yukavetsky, Gloria. 2003. "La Elaboración de Un Módulo Instruccional." Puerto Rico: Universidad de Puerto Rico En Humacao 5.

Anexos

Este prototipo de interfaz de usuario muestra un cuadro de diálogo con el título "Listado de período para asignar profesores". El cuadro contiene una tabla con los siguientes datos:

No	Curso	Inicio	Fin	Opciones
1	2019-2020	15 - 1 - 2019	30 - 1 - 2019	Asociar profesor
2	2019-2020	1 - 2 - 2019	15 - 2 - 2019	Asociar profesor

Debajo de la tabla, hay tres botones: "Cancelar", "Aceptar" y "Exportar".

Figura. 15. Prototipo de Interfaz de usuario del requisito funcional: Asociar tutor de estudiantes de práctica docente. Fuente: Elaboración propia.

Este prototipo de interfaz de usuario muestra un cuadro de diálogo con el título "Registrar período". El cuadro contiene tres campos de entrada de tipo lista desplegable:

- Curso
- Fin
- Inicio

Debajo de los campos, hay dos botones: "Cancelar" y "Aceptar".

Figura. 16. Prototipo de Interfaz de usuario del requisito funcional: Registrar periodo de práctica docente en curso académico. Fuente: Elaboración propia.

Período de Práctica Docente

No	Curso	Inicio	Fin	Opciones
1	2019-2020	15 - 1 - 2019	30 - 1 - 2019	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
2	2019-2020	1 - 2 - 2019	15 - 2 - 2019	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Figura. 17. Prototipo de Interfaz de usuario del requisito funcional: Listar periodos de práctica docente en curso académico. Fuente: Elaboración propia.

Asociar estudiantes de practica a centros

No	Centro	Periodo	Opciones
1	EP Julio Diaz	Noviembre	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/> <input type="button" value="Asoc estudiante"/>
2	ESBU Jose Marti	Diciembre	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/> <input type="button" value="Asoc centro"/>

Figura. 18. Prototipo de Interfaz de usuario del requisito funcional: Asignar estudiante de práctica docente a centro educacional. Fuente: Elaboración propia.

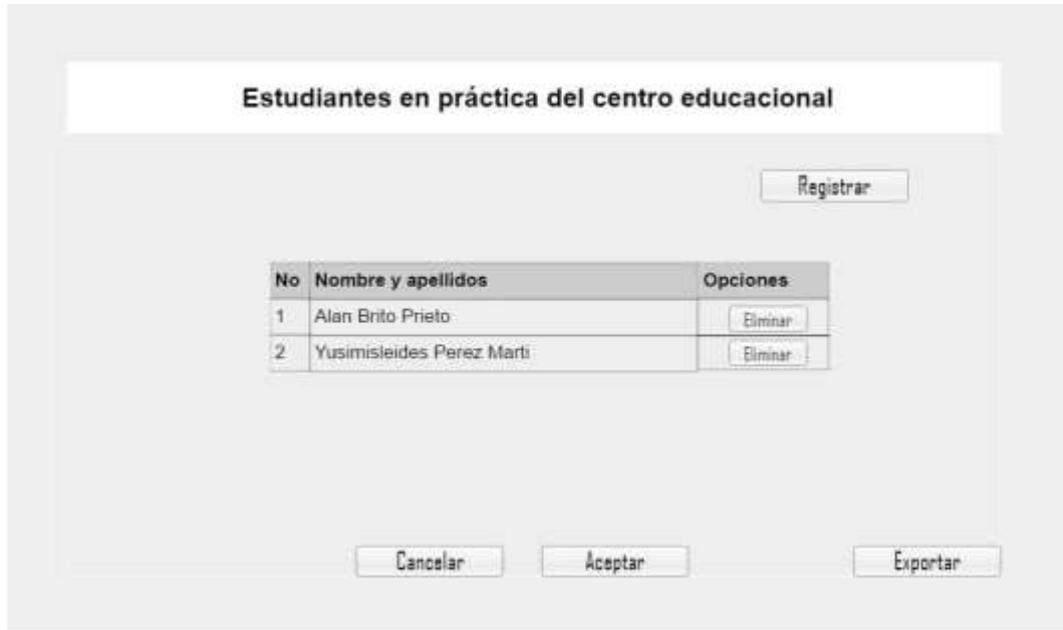


Figura. 19. Prototipo de Interfaz de usuario del requisito funcional: Listar estudiante de práctica docente.
Fuente: Elaboración propia.

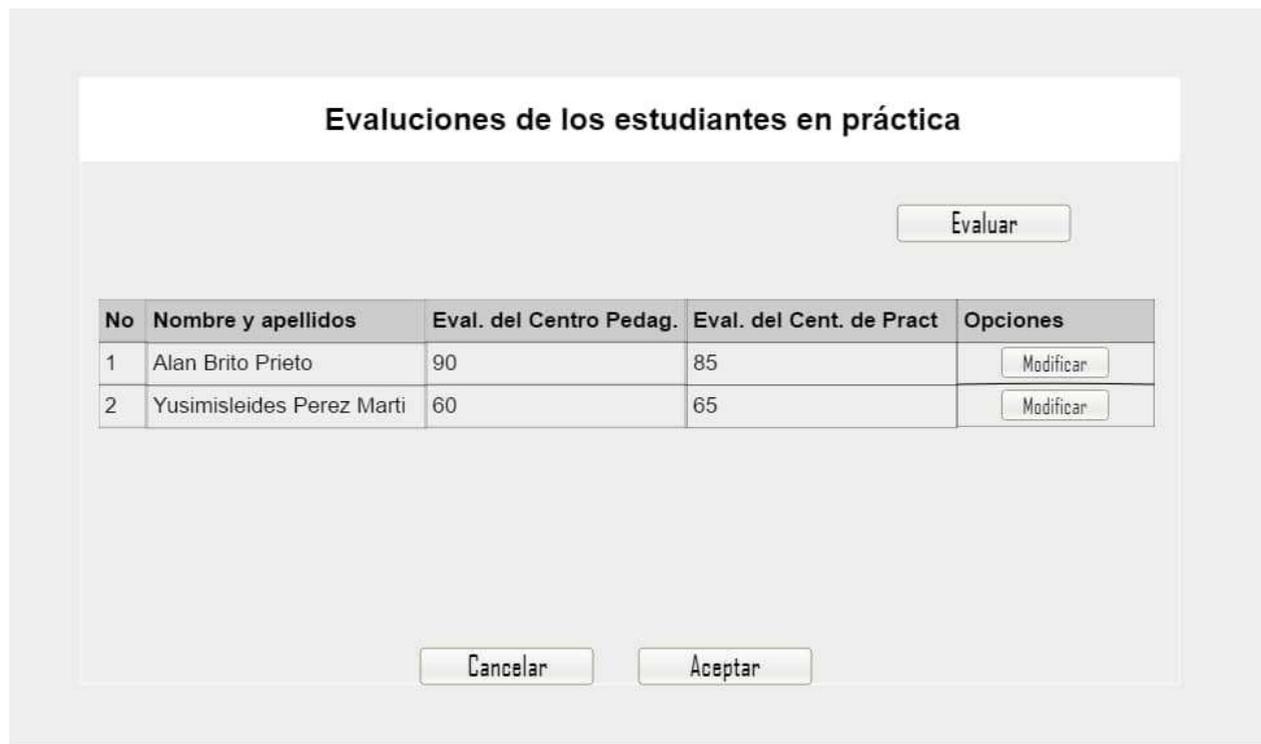


Figura. 20. Prototipo de Interfaz de usuario del requisito funcional: Reporte de evaluaciones de los estudiantes de práctica docente por centro de procedencia: Elaboración propia.