

Desarrollo del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba

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

Autor:

Juan Pablo León Bencomo

Tutor(es):

Ing. Alexander Rodríguez Mompié

Ing. Mairelys Boeras Velázquez

La Habana, septiembre de 2020

“Año 62 del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Juan Pablo León Bencomo, soy el autor principal del trabajo final de tesis de pregrado que se titula: "Desarrollo del módulo Control Docente para el Sistema de Gestión Académica del Ministerio de Educación de la República de Cuba". El cuál ha sido desarrollado como parte del trabajo del Centro de Tecnologías para la Formación. Autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana, a los _____ días del mes de _____ del año 2020.



Juan Pablo León Bencomo

Autor

Ing. Mairelys Boeras Velázquez

Tutora

Ing. Alexander Rodríguez Mompíe

Tutor

RESUMEN

La presente investigación pretende perfeccionar todo lo referente al proceso de Control Docente de los niveles de enseñanza pertenecientes al Ministerio de Educación de la República de Cuba. La entidad no cuenta con una manera eficiente de realizar y controlar el proceso dado que en la actualidad mediante la utilización de herramientas ofimáticas. Es por ello, que se decide como propuesta de solución el desarrollo del módulo Control Docente, el cual formará parte de un conjunto de módulos relacionados entre sí que permitirán cumplir las funcionalidades necesarias para proveer a los distintos centros educativos asociados al Ministerio de Educación de la República de Cuba de un sistema que permita la gestión de sus procesos. Para su desarrollo se estudiaron los procesos llevados a cabo en el Instituto Central de Ciencias Pedagógicas y en sistemas de gestión académicas, tanto nacionales como internacionales, relacionados con el control de las actividades docentes; además de estudiar a profundidad las herramientas, tecnologías, metodologías y lenguajes que fueron definidos por el Centro de Tecnologías para la Formación para diseñar el sistema. Se realizó el análisis y diseño del módulo, generando los artefactos y la documentación que propone la metodología de desarrollo empleada.

Palabras clave: control docente, módulo, desarrollo

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación teórica del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.	6
1.1. Conceptos asociados al problema	6
1.2. Análisis de soluciones similares.....	8
1.3. Tecnologías	14
1.4. Conclusiones del capítulo	19
CAPÍTULO 2: Análisis y diseño del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.....	20
2.1. Descripción de la propuesta de solución.....	20
2.2. Diagrama de Procesos de Negocio	20
2.3. Análisis de los requisitos	20
2.4. Requisitos de software	21
2.5. Descripción de requisitos por procesos	28
2.6. Descripción de la arquitectura	28
2.7. Diagramas de clases de diseño.....	34
2.8. Modelo de despliegue.....	36
2.9. Conclusiones del capítulo	36
CAPÍTULO 3: Implementación y pruebas del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.	38
3.1. Diagrama de componentes.....	38
3.2. Estándares de codificación.....	40
3.3. Estrategia de Pruebas	41
3.4. Estrategia de Validación	46
3.5. Conclusiones del capítulo	48
Conclusiones Generales.....	49
Recomendaciones	50
Bibliografía.....	51

INTRODUCCIÓN

Con el vertiginoso avance alcanzado por las Tecnologías de la Informática y las Comunicaciones (TIC) en estas últimas décadas queda en evidencia el éxito alcanzado por las empresas, fábricas e instituciones que despliegan las TIC como eslabón fundamental en su desarrollo y perfeccionamiento, permitiendo el aumento de la eficiencia, la eficacia, la productividad y otros factores que resultan indispensables para el logro de sus metas.

Cuba es una nación que intenta mantenerse a la vanguardia con el uso de las tecnologías pese a las limitaciones que presenta. Uno de los sectores que intenta mantenerse al compás del uso de las tecnologías es el sector de la educación ya que es uno de los de mayor importancia en este país. Desde el triunfo de la Revolución se han creado centros de educación de todos los niveles y formas de enseñanza permitiendo así lograr que la población cubana haya sido alfabetizada en un 99,8%[1]. Es por esto que el Departamento de Educación de la UNESCO señala que Cuba dispone de la tasa de analfabetismo más baja y de la tasa de es-colarización más alta del continente [2].

En la actualidad existe un gran auge en la incorporación de las Tecnologías de la Información y las Comunicaciones como herramienta de apoyo al desarrollo de los procesos. El sector de la educación en Cuba ha tenido la necesidad del uso de las mismas para aprovechar las potencialidades que brindan como medio para apoyar la gestión de sus procesos y contribuir al aprendizaje de los estudiantes.

Los centros educacionales se enfrentan hoy en día a la necesidad de adquirir, implementar y abrazar las nuevas tecnologías, en particular las de aprendizaje, permitiendo a las distintas instituciones de enseñanza vincular de manera dinámica, el aprendizaje con los nuevos medios de comunicación, la gestión de tareas y recursos disponibles en la institución, así como divulgar y promover información, una de las organizaciones que se encarga de llevar el control de todos estos procesos es el Ministerio de Educación (MINED).

El MINED tiene como misión esencial, dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en la actividad educacional, excepto en la Educación Superior; o sea, para los niveles de enseñanza: Círculos Infantiles, Educación Pre-escolar, Primaria, Secundaria básica, Preuniversitaria, Educación Técnica Profesional, Educación de adultos y la Educación especial. Dentro de estos niveles de enseñanza se desarrollan un conjunto de procesos que elevan la calidad de la educación[3].

Dentro de estos procesos y con carácter muy importante se encuentran el control y seguridad de las evaluaciones docentes de los estudiantes, además del control de los registros de asistencia y el control de los datos emitidos por los reportes estadísticos. Cada centro educacional es responsable de controlar y resguardar las evaluaciones de sus alumnos, así como los registros de asistencia;

además de notificar a las estructuras educacionales, dígase la estructura Municipal, Provincial y Nacional de Educación a la que se subordina, del avance de sus escolares.

El Departamento de Servicios de Información Estadísticas de la Dirección de Organización, Planificación e Información del MINED, informó en su anuario estadísticos del curso 2018 – 2019, que Cuba contaba con un total de 1 791 143 estudiantes distribuidos en 10 795 centros de educación[4].

El MINED, es el encargado de contribuir a elevar la calidad del proceso de formación de los estudiantes y la superación permanente de su claustro. Para esto sirve como base el estudio estadístico y cualitativo de las calificaciones de los estudiantes en todos los centros de educación.

Hoy en día este proceso está centrado en el correo electrónico y la entrega personal, siendo estos los medios de colaboración-comunicación que se emplea para el envío de la información. Además, debido a que la forma de controlar, monitorizar, archivar y consultar la información que se genera es de forma manual, se acumula una gran cantidad de documentos, resultando engorroso la correcta gestión de la información para las diferentes estructuras que la consolidan, haciendo que llevar ese procesamiento en los diferentes niveles hasta llegar al Nacional se convierta en un proceso lento y difícil.

Debido a la falta de informatización y el gran cúmulo de información se ve afectada la obtención de reportes a partir del análisis cualitativo y estadístico, que muestren los resultados con diferentes grados de exactitud y detalles del procesamiento de dicha información permitiendo fortalecer las bases del análisis estadístico. Al ser el trabajo manual la principal forma de trabajo surge la necesidad de un sistema informático centralizado que permita validar pautas generales para la entrega de la información en todos los centros de educación y la centralización de la misma de forma automatizada garantizando su veracidad, completitud y confiabilidad.

Esta problemática ocasiona que el MINED solo cuente con los promedios de las evaluaciones finales de cada estudiante y el registro de asistencia al concluir un nivel de enseñanza no siendo posible realizar estudios detallados de sobre esa información, para realizar reportes, los cuales permitirían controlar el rendimiento de los estudiantes permitiendo trazar nuevas estrategias educativas en caso que sea necesario.

A partir de la situación existente se plantea el siguiente **problema de investigación**: ¿Cómo ampliar la gestión del proceso de control docente en los centros educacionales del País, y para todos los niveles de la estructura educacional del Ministerio de Educación de la República de Cuba?

Para dar solución al problema en cuestión se propone como **objeto de estudio** los procesos de gestión académica en los organismos formadores y como **campo de acción** el proceso de Control

Docente para los niveles de enseñanzas pertenecientes al Ministerio de Educación de la República de Cuba.

Siguiendo como **objetivo general** desarrollar un módulo para el Sistema de Gestión Académica que permita ampliar la gestión del proceso de control docente en los centros educacionales del País, y para todos los niveles de enseñanza del Ministerio de Educación de la República de Cuba.

Las **preguntas científicas** que guían y orientan el desarrollo del proceso investigativo son las siguientes:

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

Para lograr el cumplimiento de los objetivos se hace necesario desarrollar las siguientes **tareas de investigación**:

- ✓ Estudio del estado del arte relacionado con la gestión de los procesos de control docente en los sistemas de gestión académica.
- ✓ Análisis de las tecnologías, herramientas y lenguajes necesarios para desarrollar la solución propuesta.
- ✓ Análisis del diseño y los requisitos para crear los artefactos necesarios para darle cumplimiento al objetivo general de la investigación.
- ✓ Definición de una estrategia de pruebas diseñadas para medir el nivel de calidad de la solución propuesta.

En la presente investigación científica se emplearon los siguientes métodos teóricos y empíricos. Como método teórico se utiliza el método **Histórico-Lógico** en la fundamentación teórica, haciendo un estudio de sistemas similares desarrollados en el país y en el mundo, así como de la gestión de las actividades del proceso de gestión del control docente en el MINED y las regulaciones para este proceso emitidas por dicha organización. Se utiliza el método **Analítico-Sintético** para hacer un análisis exhaustivo de los procesos de negocio y de diseño descomponiéndolo en sus partes

fundamentales. Se utiliza además el método **Sistémico** tomando el evento de control docente un elemento más dentro del Sistema de Gestión Académica, en este caso como otro de sus eventos. Se tiene en cuenta todas y cada una de las relaciones que se establecen dentro de este proceso. El método **Modelación** permitirá descubrir y estudiar las relaciones entre los procesos de gestión de información, mediante la realización de modelos con vista a investigar la realidad y comprender el objeto de estudio en su totalidad. Además, permite representar lo que se quiere estudiar de una forma más simple.

Como método **empírico** se utiliza la **Observación** el cual permite la comprensión de la problemática de la investigación ya que posibilita la investigación de los procesos en su manifestación externa, posibilitando así el planteamiento del problema científico y el **Experimento** en los casos de prueba que se le realizan a la aplicación en su primera fase de implementación permitiendo corregir las no conformidades arrojadas luego de estas pruebas, se utiliza la Revisión de documentos al revisar la bibliografía existente y al consultar información.

Con la realización de esta investigación se espera obtener como **resultado** un módulo para el Sistema de Gestión Académica del MINED que permita:

1. Llevar el control de las evaluaciones de los estudiantes y profesores y registro de asistencia de los estudiantes para todos los niveles de enseñanza del MINED.
2. Consultar información referente a las evaluaciones de los estudiantes y profesores en los diferentes niveles de la estructura del Ministerio para la toma de decisiones.
3. Contar con reportes de consolidación de información relacionado con las evaluaciones y asistencia de los estudiantes en los diferentes niveles de la estructura educacional del MINED: en centros educacionales, nivel municipal, provincial y nacional.
4. Dotar al MINED de la información necesaria relacionada con las evaluaciones docentes y registro de asistencias de los estudiantes para contribuir a la toma de decisiones en todos los niveles de la estructura y con ello la mejora de la educación en el País.

El presente trabajo cuenta con tres capítulos. A continuación, se hace una breve descripción de cada uno de ellos:

CAPÍTULO 1. Fundamentación teórica del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba: En este capítulo se definen algunos conceptos importantes para la comprensión del objeto de estudio de la investigación. Se abordan los temas relacionados con el marco teórico de la investigación, es decir, se hace referencia al estado del arte del tema tratado en el ámbito nacional e internacional. Además, se realiza una descripción de las herramientas y tecnologías empleadas en el desarrollo de la solución.

CAPÍTULO 2. Propuesta de solución del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba: En este capítulo se describen las

actividades que forman parte del proceso de control docente, se muestra el objeto de informatización y la descripción general de la propuesta. Se definen los requisitos funcionales y no funcionales del módulo a desarrollar. Por último, se describe la arquitectura, los patrones de diseño y se muestra el modelo de la base de datos.

CAPÍTULO 3. Implementación y pruebas del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba: En este capítulo se presentan los estándares de codificación a seguir para la implementación del módulo, así como las relaciones entre componentes que lo forman. Además, se definen la estrategia de pruebas para validar las funcionalidades del módulo y se define la estrategia de validación a utilizar para validar los posibles resultados de las pruebas.

CAPÍTULO 1: Fundamentación teórica del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.

En el presente capítulo se realiza un estudio del estado del arte referente al problema de investigación. Como parte de dicho estudio se analizan diferentes sistemas que permiten la gestión de evaluaciones. Además, se describen herramientas, metodologías y tecnologías empleadas para el desarrollo del módulo.

1.1. Conceptos asociados al problema

Control Docente:

Para comprender en que consiste el control docente se analizan dos conceptos fundamentales: Control y Docente.

Control:

Hace referencia al dominio, mando y preponderancia, o a la regulación sobre un sistema[5].

Docente:

Es aquella persona que se dedica de forma profesional a la enseñanza. La docencia es una profesión cuyo objetivo principal es transmitir la enseñanza a otras personas, se puede hablar en un marco general de enseñanza o sobre un área en específico[6].

Gestión

Hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera[7].

Sistema de evaluación

Para comprender en que consiste un sistema de evaluación se analizan dos conceptos fundamentales: Sistema y Evaluación.

Sistema

Un sistema es un conjunto de elementos organizados en cierta forma y que pueden estar más o menos delimitados tanto respecto a ellos como respecto a otros sistemas coexistentes como parte de un sistema más amplio[8].

Un sistema es una interrelación permanente entre sus elementos o partes consecuencia de un proceso permanente de interdependencia entre ellos, y entre ellos y el medio. Cada elemento del

sistema constituye una representación simplificada de alguna característica del mismo y permite ser tomado como objeto de estudio aisladamente. Sin embargo, cada elemento adquiere significado sólo en la medida que constituye una parte integrante del todo[9].

Evaluación

La tarea de una evaluación es iluminar el raciocinio que dirige el desarrollo de un programa y su evolución, identificar factores históricos y contextuales que lo influyen y facilitar el examen crítico de estos aspectos dentro y fuera de la comunidad sobre la cual actúa el programa (...) Puede reunir información relevante para el programa, sus objetivos y aspiraciones, y puede someterla a un examen profundo. Puede obtener, articular y competir la comprensión de por qué el proceso es como es por referencia a los objetivos de quienes en él participan, los condicionamientos circunstanciales y las oportunidades disponibles, y reflexionar sobre esta comprensión a la luz del contexto más amplio de la experiencia del programa como un todo[10].

La evaluación es un proceso mediante el cual se realiza una “cuidadosa valoración retrospectiva de los méritos, importancia y valor de la aplicación, productividad y resultados de las intervenciones gubernamentales, que se pretende desempeñe un papel en futuras situaciones y acciones prácticas[11].

Se denomina evaluación al proceso dinámico a través del cual, e indistintamente, una empresa, organización o institución académica puede conocer sus propios rendimientos, especialmente sus logros y flaquezas y así reorientar propuestas o bien focalizarse en aquellos resultados positivos para hacerlos aún más rendidores[12].

A modo de conclusión se puede llegar que un sistema de evaluación es el conjunto ordenado de normas y procedimientos que regulan el proceso de actividad educativa y que permite evaluar el desempeño de los estudiantes en un periodo determinado y en un nivel de enseñanza específico.

Centralización

Se define la centralización como la reunión de elementos bajo una misma dirección o un mismo individuo que se encargará de su gestión. La centralización de la información surge debido a la existencia de varias entidades dispersas geográficamente que deben justificar los trabajos realizados mediante documentación y se crea la necesidad de tener toda esa información guardada y clasificada en un sistema accesible por todos de forma rápida y en cualquier momento necesario. Permite mejorar el flujo de la documentación, mejora la calidad de los datos, aumenta la seguridad y facilita el seguimiento y control de la información[13].

En lo concerniente a los planes de estudio, la centralización implica reunir y ordenar todos los datos en una única fuente o servidor central para ser gestionados y monitorizados por el Ministerio de Educación de la República de Cuba de manera más eficiente.

Estandarización

El término estandarización proviene del término estándar, aquel que refiere a un modo o método establecido, aceptado y normalmente seguido para realizar determinado tipo de actividades o funciones. Tiene como connotación principal la idea de seguir el proceso estándar a través del cual se tiene que actuar o proceder. Al mismo tiempo, esta idea supone la de cumplir con reglas que, si bien en ciertos casos pueden estar implícitas, en la mayoría de las oportunidades son reglas explícitas y de importante cumplimiento a fin de que se obtengan los resultados esperados y aprobados para la actividad en cuestión[13].

La estandarización de la información asociada a los planes de estudio permite establecer una guía que define el qué y el cómo deben estar estructurados los datos para asegurarse de que sean útiles y fácilmente enlazables y utilizables por el Ministerio de Educación de la República de Cuba.

1.2. Análisis de soluciones similares

Desde los primeros años del siglo XXI se hace evidente, para las organizaciones o entidades dedicadas a la educación, la necesidad de manejar su información mediante la gestión de procesos que faciliten su competitividad. Dicha necesidad demanda el uso de herramientas tecnológicas que le permita a tales entidades estar a la vanguardia. La continua evolución tecnológica exige a las organizaciones, alinear sus procesos administrativos con la gestión tecnológica, facilitando muchas de las tareas inherentes a los diferentes procesos y funciones, lo cual representa optimización en costos y tiempo.

A medida que estas instituciones se desarrollan, su directorio activo de dominio crece exponencialmente, según comienzan nuevos cursos se incorporan nuevos estudiantes y trabajadores. Con una gran cantidad de estudiantes pertenecientes a diferentes tipos de enseñanza, puede ser todo un reto mantener una visión general de sus evaluaciones y progreso. Muchos centros de educación ya optan por la solución de un sistema de gestión académica, el análisis de estos sistemas nos permitirá conocer las tendencias actuales en la gestión de control docente.

1.2.1. Ambiente Internacional

Sistema automatizado de control de matrículas y notas para la unidad educativa González Suárez

Este sistema es desarrollado por la estudiante Gladys Inés Martínez Barreno como trabajo para optar por el título de Ingeniería en Sistemas Informáticos y Comunicaciones, en la Universidad Técnica de Ambato, Facultad de Ingeniería en Sistemas, el año 2016. El lenguaje de programación utilizado para la implementación fue Visual Basic 6.0. Implementada con el objetivo de agilizar el proceso de gestión académica tanto para el personal administrativo como para los alumnos, estudiantes, padres y profesores, a la vez indica que es necesario la automatización de procesos académicos mediante un software que optimice las tareas con fácil acceso y control adecuado de datos, concluye además que se debe trabajar con una base de datos para mantener la integridad de los datos[14].

Este sistema posee un módulo llamado docencia que permite a los estudiantes, padres y profesores dar seguimiento a las evaluaciones y registro de asistencia de sus estudiantes, pero es el módulo secretaria el que permite registrar y modificar esta información.

ACADEMIC XXI

El sistema ACADEMIC XXI ha sido concebido para acoplarse con facilidad al modelo de estudio de todas las instituciones educativas de nivel primario y medio. Esto se ha logrado ya que se le han incluido diferentes características. Es un sistema multiplataforma. Cuenta con un módulo "NOTAS XXI" para el registro de notas desde terminales instaladas para profesores. El sistema cuenta con módulos como docente, estudiante, administrativo, matrículas, eventos y cartelera donde se gestiona toda la información referente a la docencia[15].

Los procesos relacionados al control docente se gestionan en los módulos docente, el módulo docente se encarga de la gestión de planes de estudio, pero también se encarga de la gestión de las evaluaciones de los profesores, mientras que el módulo estudiante permite el control y gestión de las evaluaciones de los estudiantes.

1.2.2. Ambiente Nacional

SIGENU

El Sistema de Gestión de la Nueva Universidad (SIGENU) es un sistema que se desarrolla con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la universidad cubana, este sistema se concibe de manera tal que sea capaz de brindar gran seguridad e integridad de la información, y a la vez, ser

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[16].

El sistema está basado en un servidor de aplicaciones que no es más que el componente de tiempo de ejecución principal en todas las configuraciones y donde una aplicación se ejecuta. Para el despliegue del sistema se utiliza como servidor web el Tomcat y servidor de aplicaciones el jboss.

Este sistema está compuesto por 8 elementos fundamentales base de datos, cliente secretaria, cliente estadística, cliente profesor, cliente administrador, cliente archivo, cliente tracking, cliente servicios.

De estos elementos el proceso de gestión del control docente se encuentra desglosado dentro de varios componentes como son el de secretaria, el de profesor y el de estudiantes en el módulo estudiantes se realizan todas las operaciones con los estudiantes del sistema, o sea esta es la parte del sistema que permite el trabajo cotidiano de la secretaría docente en el transcurso del curso escolar.

La aplicación de secretaria es al encarga de registrar la información en el sistema, a las ves de obtener reportes necesarios en la educación superior. El profesor también dispone del módulo de profesor donde realiza el seguimiento del alumnado en su asignatura, realiza la inserción de la evaluación cualitativa de los cortes, la asistencia en línea permite el seguimiento de asistencia a un estudiante en el sistema de igual forma tanto las evaluaciones periódicas y finales, lo que permite registrar notas solamente al grupo al que es asignado.

GESTACAD

El sistema de gestión académica GESTACAD es un software en explotación en la Universidad de Matanzas “Camilo Cienfuegos”. Este sistema inicialmente fue creado sobre Microsoft ACCESS 2000 y sus consultas visualizadas vía Web usando la tecnología ASP de Microsoft. En la actualidad, la versión del sistema en ejecución incluye un software cliente elaborado sobre el lenguaje de programación Borland Delphi, el que tiene acceso a una base de datos construida sobre InterBase Firebird Server, cuyos datos se muestran en la INTRANET de la Universidad en formato Web utilizando el lenguaje Server Side PHP para la elaboración de las consultas Web a la Base de Datos[17].

El sistema consta con:

- Un 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.
- Un 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

- Un 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 departamento.
- Un Módulo Web para la Gestión de la Matrícula.
- Un 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.
- Un 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.

El modulo profesores posee como principales funcionalidades el reporte de notas por asignatura y grupo, reporte de los resultados académicos de un estudiante en toda su carrera, reportes de asistencia y evaluaciones de los estudiantes y la gestión de las evaluaciones docentes de un estudiante.

AKADEMOS

El sistema fue creado por la Dirección de Informatización de la Universidad de las Ciencias Informáticas. Integra a su vez los marcos de trabajo CodeIgniter en su versión 1.7.3 y JQuery 1.7.1 en una sola infraestructura, razón por la que posee las características de ambos. En la integración se incluyen mejoras y modificaciones hechas específicamente al CodeIgniter, además de creación de plugins y componentes de interfaz de usuario en JQuery[18].

Este sistema gestiona los procesos académicos en la UCI, para las modalidades y tipos de cursos que se definan. El sistema está separado por módulos entre los que se encuentran. El módulo carrera, planificación docente, horario docente, personal y secretaría, control docente, estudiante, materiales docentes, ayudantía, tesis y títulos, archivo y reportes.

El módulo control docente maneja los procesos asociados a la docencia de **pregrado**. Gestiona los grupos docentes por asignaturas, asignando estudiantes y profesores. Permite registrar evaluaciones, asistencia, bonificaciones y premios, permitiendo llevar el control del comportamiento docente del estudiante.

Se pueden generar los documentos del proceso docente como: Certificación de notas, Certificación de calificaciones de curso académico, Certificación de bonificaciones, Certificación de examen de premio, Acta de comparecencia a examen final, Modelo de corte evaluativo y el Registro de asistencias y evaluaciones.

1.2.3. Análisis de las soluciones:

El análisis realizado a los sistemas de Gestión Académica permitió elaborar un resumen en forma de tabla teniendo en cuenta los siguientes parámetros:

- Procedencia: Referente a la procedencia del producto. Puede tomar valores nacional o internacional.
- Gestión de evaluaciones: Indica si el sistema puede gestionar las evaluaciones docentes.
- Registro de asistencia: Indica si el sistema permite el llevar el control del registro de asistencia.
- Enseñanzas pertinentes: referente a si el sistema esta implementado para las enseñanzas pertenecientes a la estructura del MINED o no.
- Privativo: Hace referencia a si el software es adquirido mediante pago o no.

Tabla 1. Análisis de los sistemas estudiados

Sistemas	Procedencia	Gestión de evaluaciones	Registro de asistencia	Enseñanzas pertinentes.	Privativo
Sistema automatizado de control de matrículas y notas	Internacional	Si	Si	No	No
Academic XXI	Internacional	Si	Si	Si	Si
Gestacad	Nacional	Si	Si	No	No
SIGENU	Nacional	Si	Si	No	No
Akademios	Nacional	Si	Si	No	No

A partir del estudio de las soluciones, se arriba las siguientes impresiones:

- **Sistema automatizado de control de matrículas y notas:** es un sistema que no está implementado para los niveles de enseñanza pertenecientes al MINED no permitiendo que sea utilizado como solución para la problemática planteada, pero evidencia y documenta la necesidad del uso de una base de datos para garantizar la integridad de la información.
- **Academic XXI:** Cumple con casi todos los principales requisitos del sistema, exceptuando la realización de reportes y a pesar de estar correctamente documentado no se cuenta con el código base de esta aplicación web, por lo que el uso de esta herramienta entra en contradicción con el lineamiento 108 del Partido Comunista de Cuba referente al desarrollo de soluciones propias para garantizar la soberanía tecnológica[19].

- **Gestacad:** Garantiza las funcionalidades imprescindibles para la solución, pero está desarrollado para las enseñanzas universitarias implicando que no pueda ser utilizado como solución, debido a que la solución requiere que las enseñanzas gestionadas pertenezcan a la estructura del MINED. Demuestra la necesidad de utilizar una herramienta para optimizar el tiempo de respuesta de las búsquedas en la Base de datos cuando se tiene un gran volumen de información.
- **SIGENU:** Garantiza las principales funcionalidades para la solución, pero está desarrollado para las enseñanzas universitarias implicando que no pueda ser utilizado como solución, debido a que la solución requiere que las enseñanzas gestionadas pertenezcan a la estructura del MINED. Evidencia la necesidad de separar la interfaz del acceso a los datos para garantizar el rendimiento del sistema en cuanto a velocidad de respuesta.
- **Akademos:** Es la solución más adecuada para el sistema a desarrollar, pero no cumple con el requisito de estar implementada para las enseñanzas pertenecientes a la estructura del MINED. Posee una estructura robusta que puede ser utilizada como base para el sistema a desarrollar ya que permite manejar en un solo módulo todos los datos referentes a la gestión de evaluaciones de profesores y estudiantes, registro de evaluaciones y emisión de reportes.

Por todo lo antes expuesto, se puede resumir que los sistemas señalados no cumplen con el objetivo de la presente investigación. Ninguna reúne las funcionalidades necesarias y suficientes para ser utilizada directamente como propuesta de solución. Sin embargo, cada una contribuyó a una mejor comprensión de las principales características que debe cumplir un sistema de este tipo.

1.2.4. Conclusiones del análisis de soluciones similares

Tras el análisis de los sistemas homólogos se ha arribado a las siguientes conclusiones:

- A modo general el sistema debe permitir llevar el control de las actividades docentes.
- El sistema debe separar la interfaz del acceso a los datos para mejorar su velocidad de respuesta.
- Se debe hacer uso de una base de datos que permita garantizar la integridad de la información.
- El sistema debe tener la capacidad de gestionar las evaluaciones.
- El sistema debe permitir registrar la asistencia y puntualidad.
- Dado que AKADEMOS es el sistema más completo y que más se adapta a las necesidades existentes su estructura puede ser utilizada como base para el desarrollo de la solución informática.

1.3. Tecnologías

Como entorno de desarrollo se escogió el framework Django 3.0 para el trabajo con backend y para el manejo del fronted se escogió el framework Angular 8.2. Para la selección de estas tecnologías se tuvo en cuenta lo que aportaban al proyecto, además de la facilidad de integración con el módulo principal ya que estas son las tecnologías propuestas por la dirección del Centro de Tecnologías para la Formación (FORTES) para el desarrollo del Sistema de Gestión Académica para el MINED el cual tendrá contenido un módulo de control docente.

Se escoge **Django 3.0** ya que es un framework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables. Está escrito en Python y tiene una comunidad muy amplia, que está en continuo crecimiento sus principales ventajas sobre otros framework son:

- Rapidez: Con Django se puede construir una aplicación muy buena en poco tiempo.
- Varias funcionalidades: Posee varias funcionalidades y módulos que ya están implementados, ya sea porque hay módulos de la comunidad, por cualquier paquete Python o las propias aplicaciones que Django trae.
- Seguridad: Django implementa por defecto algunas medidas de seguridad, las más clásicas, para que no haya SQL Injection, no haya Cross Site Request Forgery (CSRF) o no haya Clickjacking por JavaScript siendo todo esto manejado de forma sencilla.
- Escalabilidad: Se puede pasar desde muy poco a una aplicación enorme perfectamente, una aplicación que sea modular, que funcione rápido y sea estable.
- Versatilidad: En un principio Django comenzó siendo un Framework para almacenar noticias por sitios de prensa, blogs y este estilo de webs, pero con el tiempo ha ganado tanta popularidad que puede ser utilizado para cualquier propósito[20].

Angular 8.2 es un framework JavaScript moderno de código abierto diseñado para construir aplicaciones web, sucesor de AngularJS 1, uno de los mejores frameworks de JavaScript para crear aplicaciones web del lado del cliente. Aunque es considerado la versión siguiente, consiste en una reescritura completa de su predecesor. Está construido sobre los últimos estándares web (componentes web, observables y decoradores), la curva de aprendizaje es mínima y el rendimiento es mejor. Sigue un enfoque basado en componentes y se puede usar en combinación con cualquier framework de aplicaciones web del lado del servidor, como ASP.NET y Node.js. A diferencia de AngularJS 1 que era un conjunto de controladores y vistas individuales, en Angular la aplicación se trata como un árbol de componentes. En una aplicación angular, un árbol de componentes tendrá un componente raíz que actuará como el punto de entrada de la aplicación. Todos los demás componentes que forman parte de la aplicación se cargarán dentro del componente raíz, y se pueden

anidar de cualquier forma que necesitemos. Angular también tiene el concepto de módulos, que se utilizan para agrupar componentes con una funcionalidad similar[21].

1.3.1. Arquitectura del sistema

A continuación, se hace una descripción de las herramientas y los lenguajes que serán empleados para el desarrollo del módulo, los cuales fueron definidos por el Centro de Tecnologías para la Formación.

Lenguajes de programación

Lenguaje: HTML 5

Descripción: Hypertext Markup Language (HTML, por sus siglas en inglés) traducido al español como Lenguaje de Marcas de Hipertexto, es el lenguaje de marcado 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. 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[22].

Lenguaje: Python 3.0

Descripción: es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma[23].

Lenguaje: CSS3

Descripción: las hojas de estilo en cascada (*Cascading Style Sheets*, CSS por sus siglas en inglés) 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[24].

Lenguaje: Lenguaje de Modelado Unificado (UML)

Descripción: *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[25].

Lenguaje: *Typescript 2.0*

Descripción: es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un súper conjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases[26].

Lenguaje: *Json*

Descripción: acrónimo de JavaScript *Object Notation* (Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, aunque hoy se considera un formato de lenguaje independiente. JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia[27].

Lenguaje: *PostgreSQL 10.0*

Descripción: es un Sistema Gestor de Bases de Datos Objeto-Relacional (de sus siglas en inglés ORDBMS). Se distribuye bajo la licencia *PostgreSQL*, una licencia de código abierto liberal. *PostgreSQL* incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es una herramienta de diseño y gestión de bases de datos[28].

Herramientas de desarrollo

Nombre: *nginx 1.14*

Descripción: 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[29].

Nombre: *PgAdmin 3.0*

Descripción: es la plataforma de desarrollo y administración de código abierto más popular y rica en funciones para *PostgreSQL*. Puede ejecutarse como una aplicación web o de escritorio. Es una herramienta multiplataforma capaz de ejecutarse en sistemas con Microsoft Windows, Linux y

MacOS. Está diseñado para múltiples versiones de *PostgreSQL* y sus derivados. Presenta una extensiva documentación y una comunidad virtual[30].

Nombre: Visual Paradigm 8.0

Descripción: software de modelado UML eficiente, rápido y motiva el trabajo de forma colaborativa. Permite realizar Diagramas de procesos de negocios, Modelado UML, Diagramas de casos de usos, Diagramas de actividad, de interacción, de bases de datos, de entidad-relación. Además, posee integración para varios IDE, puede realizar Ingeniería de Código y también generar documentación, entre otras cosas; todo bajo un modelo colaborativo. Visual Paradigm soporta todas las necesidades de diseño y modelado a lo largo del ciclo de vida de desarrollo de software, es una herramienta que ayuda a construir aplicaciones de calidad, de manera más rápida, óptima y más barata[31].

Nombre: PyCharm 11.0.5

Descripción: PyCharm es uno de los entornos de desarrollo más completos para Python. Es parte de la suite de herramientas de programación ofrecidas por JetBrains, que cuenta con entornos para construir código en distintos idiomas como PHP y Ruby[32].

1.3.2. Metodología de desarrollo de software

Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y tradicionales (o robustas)[33].

Metodología ágil: Se basa en una fuerte y constante interacción, donde clientes y desarrolladores trabajan juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no a la documentación. Exige que el proceso sea adaptable, permitiendo realizar cambios de último momento.

Metodología tradicional: Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo.

Para la implementación del módulo se decide utilizar una metodología ágil. Éstas permiten entregar productos de calidad con los costes y tiempos pactados, y las metodologías tradicionales ya no bastan para este cometido, no se adaptan a las nuevas expectativas de los usuarios y a las exigencias del mercado. Las metodologías ágiles permiten centrarse en la correcta obtención del módulo sin importar cuán documentado esté, enfatizando en la rapidez y la calidad con que se desarrolle el mismo.

Como guía en el proceso de desarrollo del módulo de plan de estudio para el sistema de gestión académica del Ministerio de Educación de la República de Cuba se propone por el Centro de Tecnologías para la Formación utilizar la metodología AUP en su variante UCI. La variación AUP para la UCI surge a partir de la necesidad de que los proyectos productivos de la UCI se realicen con la utilización de una sola metodología, ya que, al no existir una metodología de software universal, toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable[34].

Debido a esto se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, nombrada Ejecución y se agrega la fase de Cierre quedando estructurado de la siguiente manera.

1. **Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
2. **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
3. **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Esta metodología posee cuatro escenarios para modelar el sistema en los proyectos, el utilizado para el desarrollo del módulo control docente es el escenario 3.

Escenario No3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP. Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.



Figura 1. Escenario 3 de la metodología AUP-UCI

Para la descripción del negocio se utilizará Diagrama de Procesos de Negocio(DPN) con el fin de obtener un buen entendimiento del funcionamiento de todos los subprocesos que se derivan del proceso de Control Docente.

1.4 Conclusiones del capítulo

En el anterior capítulo se abordaron elementos referentes al caso de estudio en cuestión, en tal sentido se puede asumir que:

- El análisis de sistemas homólogos, permitió identificar las tendencias en el desarrollo de sistemas informáticos para la gestión de evaluaciones, registro de asistencia y emisión de reportes.
- Se comprende la necesidad de desarrollar un módulo adaptado a las necesidades específicas de la institución.
- Quedan definidas las tecnologías y herramientas a utilizar, así como la metodología que servirá como guía para el ciclo de vida del proyecto.

CAPÍTULO 2: Análisis y diseño del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.

En el presente capítulo se hace una descripción del módulo a desarrollar. Se presentan los artefactos obtenidos, una vez aplicada metodología AUP-UCI. Se presentan artefactos obtenidos de la metodología. Se establece el entendimiento entre el cliente y el equipo de trabajo en relación con los objetivos a lograr, realizando un correcto análisis y diseño del módulo.

2.1. Descripción de la propuesta de solución

El sistema a desarrollar es el módulo Control Docente para el Sistema de Gestión y Secretaria Docentes XAUCE Akademos para el Ministerio de Educación este módulo permite:

1. Llevar el control de evaluaciones y asistencia para todas las enseñanzas pertenecientes a este organismo.
2. Llevar el control de los premios y bonificaciones asignadas a los estudiantes.
3. Calcular promedios de índice académico para todas las enseñanzas y estudiantes en cualquier punto de su carrera académica.
4. Generar reportes de promoción al final de cualquier año para cualquier enseñanza.
5. Generar un resumen de asistencia para cualquier centro educacional.

2.2. Diagrama de Procesos de Negocio

Un diagrama de procesos es una representación gráfica de los pasos que se siguen en una secuencia de actividades, y que son identificados mediante símbolos determinados, incluyendo toda la información que sería necesaria para un posterior análisis. Próximamente, se mostrarán los diagramas pertenecientes a los procesos del negocio. Dirigirse a [Anexo 1](#).

2.3. Análisis de los requisitos

El análisis de requisitos permite al ingeniero de sistemas especificar las características operacionales del software (función, datos y rendimientos), indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.

2.3.1. Técnicas de obtención de requisitos

Existe un gran número de técnicas para obtener requerimientos. A continuación, se describen las técnicas utilizadas en este proyecto.

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.

Estudio de documentación:

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.

2.4. Requisitos de software

La especificación de requisitos de software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software[35].

2.4.1. Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Describen lo que el sistema debe hacer y dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer[35].

Tabla 2. Especificación de los Requisitos

N°	Nombre	Descripción	Prioridad	Complejidad
RF1	Registrar Asistencia	Permite registrar las ausencias de los estudiantes en horas lectivas en un día de clases asignando un valor al campo asistencia para cada estudiante.	Media	Baja
RF2	Modificar Asistencia	Permite modificar el valor registrado de asistencia de los estudiantes en un día de clase modificando el valor del campo asistencia para un estudiante determinado.	Baja	Baja
RF3	Listar asistencia	Permite obtener un listado de las ausencias a horas lectivas en un día de clase en un centro educacional. En el listado se deben mostrar los siguientes datos: <ul style="list-style-type: none"> • Asistencia • Grupo • Grado • Fecha • Nombre 	Media	Baja
RF4	Registrar evaluación	Permite registrar los resultados obtenidos por un estudiante en una evaluación asignándole un valor al campo evaluación.	Alta	alta
RF5	Modificar evaluación	Permite modificar el valor del campo evaluación dependiendo del resultado alcanzado los resultados obtenidos por un estudiante en una evaluación.	Baja	Media
RF6	Listar evaluaciones	Permite obtener un listado de las evaluaciones de un grupo de estudiantes en una asignatura determinada. Mostrando los valores de los campos: <ul style="list-style-type: none"> • Evaluación 	Media	Media

		<ul style="list-style-type: none"> • Nombre • Grupo • Asignatura 		
RF7	Crear grupo docente	<p>Crea un grupo docente que agrupa los estudiantes por asignatura modificando los valores de los campos:</p> <ul style="list-style-type: none"> • Nombre • Asignatura • Habilitado 	Baja	Baja
RF8	Modificar grupo docente	<p>Permite modificar los valores de los valores de los campos pertenecientes a un grupo docente como son:</p> <ul style="list-style-type: none"> • Nombre • Asignatura 	Baja	Baja
RF9	Eliminar grupo docente	Inhabilita temporalmente un grupo docente cambiando el valor del campo habilitado.	Baja	Baja
RF10	Listar grupos docente	<p>Permite obtener un listado de los grupos docentes existentes en el sistema mostrando los valores de los campos:</p> <ul style="list-style-type: none"> • Nombre • Asignatura • Habilitado 	Baja	Baja

RF11	Listar evaluaciones por estudiantes	<p>Obtiene un listado de las evaluaciones obtenidas por un estudiante desde que ingresa a la escuela y todo el proceso hasta salir de las enseñanzas pertenecientes al MINED mostrando los valores de los campos:</p> <ul style="list-style-type: none"> • Evaluación • Año • Asignatura 	Media	Media
RF12	Generar promedio de evaluaciones por año académico	<p>Genera el promedio de un estudiante en un año académico listando los valores de los siguientes campos:</p> <ul style="list-style-type: none"> • Evaluación • Asignatura 	Baja	Baja
RF13	Generar promedio de evaluaciones por enseñanza	<p>Genera el promedio de un estudiante en una enseñanza. Mostrando los valores de los campos:</p> <ul style="list-style-type: none"> • Evaluaciones • Año • Asignatura 	Baja	Baja
RF14	Generar promedio de evaluaciones por nivel.	<p>Genera el promedio de un estudiante en un nivel mostrando el valor obtenido.</p>	Baja	Baja
RF15	Generar resumen de asistencia.	<p>Genera un resumen de las ausencias de los estudiantes en un centro educacional en un período de tiempo determinado mostrando los valores de los campos:</p>	Baja	Media

		<ul style="list-style-type: none"> • Asignatura • Fecha 		
RF16	Generar resumen de evaluaciones por grupo docente.	<p>Genera un resumen de evaluaciones de todos los estudiantes pertenecientes a un grupo docente listando todos los valores de los campos:</p> <ul style="list-style-type: none"> • Evaluación • Nombre Estudiante 	Baja	Baja
RF17	Crear Premios	<p>Crea un premio asignando valores a los campos:</p> <ul style="list-style-type: none"> • Nombre • Descripción • Habilitado • Tipo 	Baja	Baja
RF18	Eliminar Premios	Inhabilita un premio cambiando el valor del campo habilitado.	Baja	Baja
RF19	Modificar Premios	<p>Modifica los atributos de un premio al modificar el valor de uno de los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre • Descripción • Tipo 	Baja	Baja
RF20	Listar premios	Devuelve un listado con los premios existentes mostrando los valores de los campos:	Baja	Media

		<ul style="list-style-type: none"> • Nombre • Descripción • Tipo 		
RF21	Asignar un premio a un estudiante	Asigna un premio a un estudiante.	Media	Media
RF22	Genera un reporte de promoción	<p>Genera un reporte de la promoción obtenida por los estudiantes de un centro hasta un momento determinado listando los valores de los campos:</p> <ul style="list-style-type: none"> • Nombre • Promedio 		

2.4.2. Requisitos no funcionales

Los requisitos no funcionales, como su nombre sugieren, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen 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. A menudo se aplican al sistema en su totalidad y normalmente apenas se aplican a características o servicios individuales del sistema[35].

El sistema propuesto debe contar con los siguientes requisitos no funcionales:

RnF1-Interfaz: El sistema debe seguir las pautas y restricciones de diseño pertenecientes a la marca Xauce.

RnF2-Funcionalidad: El módulo soportará la conexión simultánea de todos los posibles usuarios habituales, un promedio de 9000 y un máximo de 20000.

RnF3-Confiability/ Fiabilidad: Garantizar la integridad y confidencialidad de la información mediante mecanismos de seguridad.

RnF4-Confiability: Mostrar la información necesaria del error.

RnF5-Usabilidad: Mostrar elementos de apoyo a la navegación.

RnF6-Usabilidad/ Operatividad: Mostrar funcionalidades de acuerdo a los privilegios.

RnF7-Disponibilidad: Disponibilidad del módulo las 24 horas del día.

RnF8-Hardware: El Servidor de Bases de Datos y Servidor Web debe tener: Microprocesador de 8 núcleos, 8GB de memoria RAM, 250 GB de disco duro y una fuente de 800 W como mínimo. o El cliente debe tener: Microprocesador de 1GHz, 512 Mbs de memoria RAM, 1GB de disco duro, Mozilla Firefox 36.0 como mínimo.

RnF9-Soporte: El módulo cumplirá con las normas de codificación, conversiones para nomenclatura, bibliotecas de clase definidas para el Sistema de Gestión y Secretarías Docentes Xauce Akademos para el MINED.

RnF10-Soporte: El módulo contará con toda la documentación definida en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.

RnF11-Restricciones de diseño: El módulo deberá ser desarrollado empleando las herramientas y la arquitectura de información definida para Sistema de Gestión y Secretarías Docentes Xauce Akademos para el MINED.

2.5. Descripción de requisitos por procesos

La descripción de requisitos por procesos describe las interacciones que existirán entre el actor del proceso y el software, permite facilitar la comprensión del comportamiento del sistema a desarrollar. Posteriormente se describirá, en las tablas que se encuentran en el [anexo 2](#) algunos requisitos funcionales pertenecientes al proceso de gestión del control docente.

2.6. 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. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes[36].

En la siguiente imagen se muestra la representación de la arquitectura:

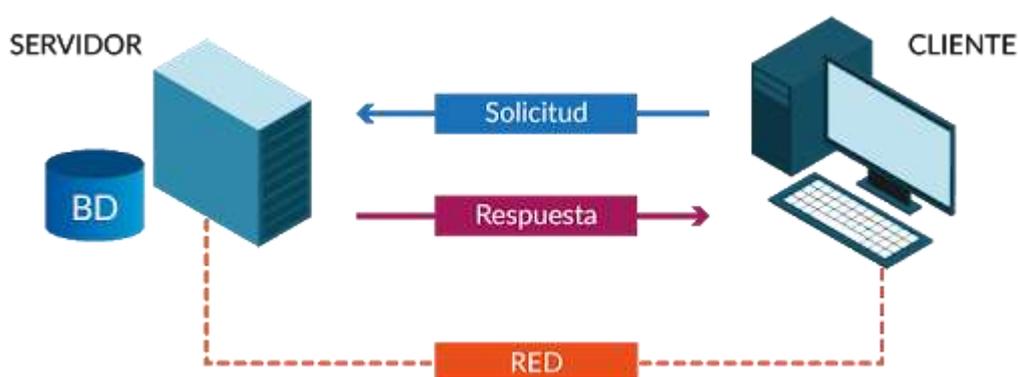


Figura 3. Arquitectura Cliente-Servidor

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[37].

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[37].

2.6.1. Patrones arquitectónicos

En este trabajo habrá presencia de dos patrones arquitectónicos diferentes debido a que la información en las diferentes capas es tratada por diferentes framework estos patrones arquitectónicos son modelo-vista-plantilla perteneciente al framework Django y modelo-vista-vistamodelo perteneciente al framework angular a continuación se explicara el funcionamiento de estos patrones en cada framework:

Modelo Vista Plantilla

Debido a que el controlador es manejado por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV debido a los componentes que utiliza.

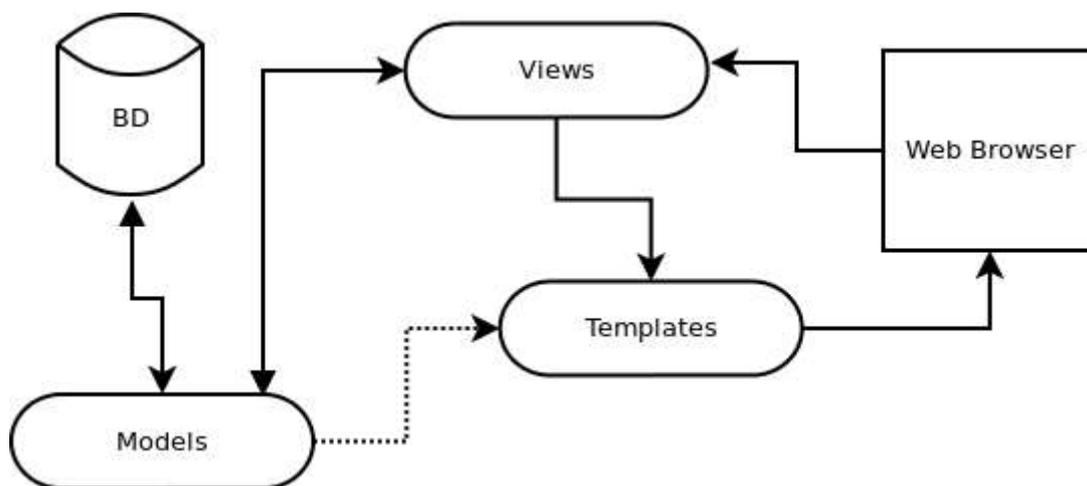


Figura 4. Patrón Arquitectónico Modelo-Vista-Plantilla

En el patrón de diseño Modelo Vista Plantilla,

- M significa Modelo, 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.
- T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.
- V significa Vista, la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre los modelos y las plantillas.

Modelo Vista VistaModelo

El patrón *Model-View-ViewModel* (MVVM) nos ayuda a separar la lógica de negocios de la interfaz de usuario, facilitando las pruebas, mantenimiento y la escalabilidad de los proyectos. Angular es un framework MVVM por lo que está dividido en tres componentes principales:

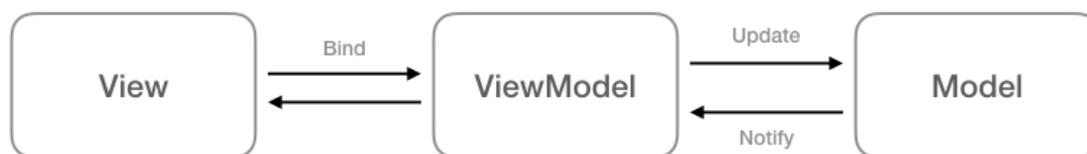


Figura 5. Patrón Arquitectónico Modelo-Vista-ViewModelo

Modelo: El modelo, dentro de MVVM es el encargado de representar el modelo del negocio, proveyendo de esta manera la base necesaria para la manipulación de los datos de la aplicación, además parte del modelo se lo puede usar como clases POCO (*Plain Old CLR Objects*) para poder usarlas con *Entity Framework Code First* o algún otro ORM. Cabe resaltar que, en el modelo, no debería de existir ninguna lógica de negocio o código que afecte a como se visualizan sus datos en pantalla.

Vista: La vista es la parte encargada de la parte visual de nuestra aplicación, no teniéndose que ocupar en ningún momento en el manejo de datos. En MVVM la vista tiene un rol activo, esto significa que en algún momento la vista recibirá o manejará algún evento (Clic en un botón, alguna tecla presionada, etc.) y tendrá que comunicarse con el modelo, para poder cumplir el requerimiento.

ViewModelo: El *ViewModel* (modelo de vista en español) es el encargado de ser la capa intermedia entre el modelo y la vista, procesando todas las peticiones que tenga la vista hacia el modelo, además

de tener que ocuparse de manejar las reglas del negocio, la comunicación con aplicaciones externas o consumir datos desde alguna fuente (Bases de Datos, Web Services, Sensores, etc.).

2.6.2. Patrones de diseño

Los patrones de diseño son soluciones a problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación. Aunque la aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas, entre otras. Incluso algunas funcionalidades y estructuras similares dentro del propio sistema. Establecen un lenguaje común para programar. Identifica clases, instancias, roles, colaboraciones y distribución de responsabilidades. Esto permite un lenguaje de programación a alto nivel. Es la forma práctica de describir aspectos de la organización de un programa[38].

Dentro de los más conocidos se encuentran los Patrones de Principios Generales para Asignar Responsabilidades (GRASP) y los patrones de la *Gang of Four* (GoF).

GRASP: Principios Generales para Asignar Responsabilidades

GRASP es un acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades). El nombre se eligió para sugerir la importancia de aprehender (*grasping* en inglés) estos principios para diseñar con éxito el software orientado a objetos. Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones[39].

A continuación, se presentan los patrones GRASP utilizados.

- **Experto:** El GRASP de experto en información es el principio básico de asignación de responsabilidades. Determina cuál es la clase que debe asumir una responsabilidad a partir de la información que posee. O sea, la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Este patrón es evidenciado en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases `StructureCreateComponent.ts` que se encarga de la creación de una estructura. La figura 6 muestra el uso del patrón Experto.

```

15 export class StructureCreateComponent implements OnInit {
16
17     createForm: FormGroup;
18     provinces: any[] = [];
19     municipalities: any[] = [];
20     parentStructures: any[] = [];
21     typeStructures: any[] = [];
22     buildings: any[] = [];
23
24     formMessages = FormMessages;
25

```

Figura 6. Patrón Experto

- **Creador:** Ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Se identifica al asignar a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes:
 1. B agrega objetos de A.
 2. B contiene objetos de A.
 3. B registra instancias de objetos de A.
 4. B tiene los datos de inicialización que se pasarán a un objeto de A cuando sea creado.
 5. B es un creador de los objetos A.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización. La figura 7 muestra un ejemplo del patrón creador:

```

) {
    this.createForm = this.fb.group({
        name: [null, [Validators.required, Validators.minLength(3)]],
        province: [null, [Validators.required]],
        municipality: [null, [Validators.required]],
        parent: [null, [Validators.required]],
        type: [null, [Validators.required]],
        address: [null, [Validators.required]],
        popular_council: [null, [Validators.required]],
        location: [null, [Validators.required]],
        building: [null, [Validators.required]],
    });
}

```

Figura 7. Patrón Creador

- **Alta cohesión:** Indica que la información que almacena una clase debe ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Este patrón es evidenciado en el modelo de datos perteneciente a la clase EvaluationRecords. La figura 8 muestra un ejemplo del patrón alta cohesión.

```

{
  "owner_id": 0,
  "is_disable": true,
  "name": "string",
  "group_id": 0,
  "school_id": 0,
  "study_program_id": 0
}

```

Figura 8. Patrón Alta cohesión

GOF: Patrones de la “pandilla de los cuatro”

Los patrones que se presentan proceden de Design Patterns, un libro básico y muy popular que presenta 23 patrones que son útiles durante el diseño de objetos. Puesto que el libro fue escrito por cuatro autores, estos patrones se conocen como los patrones de la "pandilla de los cuatro" o patrones "GoF". Describen las formas comunes en que objetos de tipos diferentes pueden organizarse para trabajar entre sí[39].

Los patrones GoF utilizados en el desarrollo de la aplicación son:

- **Mediador (Mediator):** Este patrón se evidencia en las librerías, las cuales son mediadoras entre las clases controladoras y las modelos o acceso a datos.

Se evidencia en la clase models.py donde se hace uso de la librería models para declarar el tipo de dato de los diferentes atributos. La figura 9 evidencia el uso de este patrón:

```

5 class SummaryDailyAssistance(BaseAkModel):
6     name = models.CharField(max_length=100, null=True)
7     daily_assistance = models.IntegerField()
8     date = models.DateField()
9     school_id = models.IntegerField()
10
11
12 class TeachingGroup(BaseAkModel):
13     name = models.CharField(max_length=100, null=True)
14     group_id = models.IntegerField()
15     school_id = models.IntegerField()
16     study_program_id = models.IntegerField()

```

Figura 9. Patrón Mediador

- **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.

En la figura 10 se muestra la clase “school.component.ts” la cual tiene en su constructor una instancia NgxPermissions “NgxPermissionsService”. De esta manera no hay necesidad de instanciar a NgxPermissions en cualquier otra parte de la clase. Su intención consiste en

garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

```
constructor(  
  private schoolService: StructureManageService,  
  private rolesService: NgxRolesService,  
  private permissionService: NgxPermissionsService,  
  private userPermissions: PermissionsService,  
  private message: NzMessageService,  
) {  
  super();  
}
```

Figura 10. Patrón Instancia única

2.7. Diagramas de clases de diseño

Los diagramas de clases de diseño describen gráficamente las especificaciones de las clases de software y de las interfaces que contiene la propuesta de solución y las relaciones entre ellas. Las relaciones se basan en las peticiones que se hacen entre ellas y el flujo de datos que intercambian. En la figura 11 se muestra un ejemplo de diagrama de clases de diseño Gestionar evaluaciones:

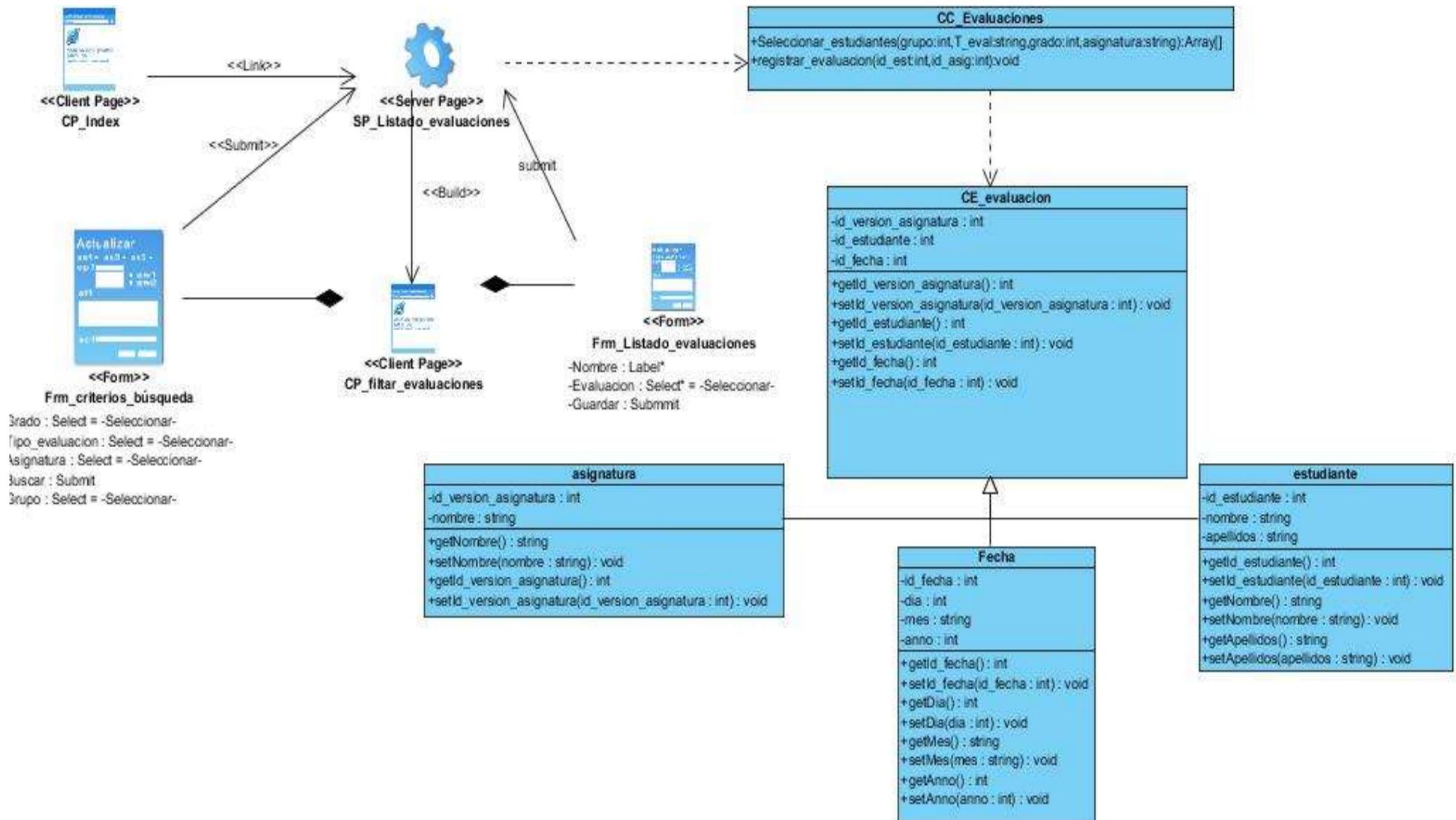


Figura 11. Diagrama de clases de diseño: Gestión de Evaluaciones

2.8. Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño[40].

En la figura 12 se muestra el diagrama de despliegue definido para la propuesta de solución:

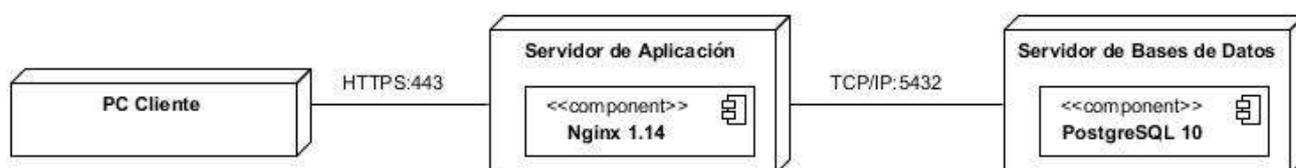


Figura 12. Diagrama de despliegue

PC Cliente: ordenador desde donde los usuarios acceden a la aplicación.

Servidor de aplicación: ordenador donde se encuentra el servidor web Nginx. Es de donde se gestiona todo el contenido de la aplicación. Las máquinas de los clientes acceden a él, a través del navegador web.

Servidor de Bases de Datos: servidor donde se ubica la base de datos que almacena toda la información generada por el sistema y que va a ser utilizada por los servicios que brinda el módulo.

HTTPS: Protocolo transferencia de hipertexto seguro, por sus siglas en inglés, *Hypertext Transfer Secure Protocol* (HTTPS) es un protocolo de aplicación basado en el protocolo HTTP que permite la transferencia de segura de datos de hipertexto siguiendo el esquema petición-respuesta entre un cliente y un servidor.

TCP/IP: es una familia de protocolos de internet que permite la conexión entre el servidor web y el servidor de bases de datos.

2.9. Conclusiones del capítulo

Después de realizado el análisis y diseño de la propuesta de solución y haber generado los diferentes artefactos que dispone la metodología AUP-UCI, se arriba a las siguientes conclusiones:

- El análisis de requisitos con el cliente permitió recoger 22 requisitos funcionales y 11 requisitos no funcionales.
- La obtención de requisitos facilitó una correcta modelación del problema.

- Se definen 5 patrones de diseño que se ajustan a los patrones arquitectónicos a utilizar que son Modelo Vista Controlador y Modelo Vista Vista-Modelo.

CAPÍTULO 3: Implementación y pruebas del módulo “Control Docente” para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.

En el presente capítulo se definirán los estándares de codificación con los que debe cumplir el equipo de desarrollo durante la implementación, así como los métodos y técnicas para la realización de pruebas, además de definir la estrategia de validación de las mismas. El proceso de pruebas está dirigido a componentes del software, con el objetivo de medir el grado en que se cumplen los requisitos exigidos por el cliente y si posee la calidad requerida.

3.1. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, siendo los componentes piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos componentes de código fuente, binarios o ejecutables. Normalmente contienen componentes, interfaces y relaciones entre ellos, y como todos los diagramas también puede contener paquetes utilizados para agrupar elementos del modelo[41]. En la figura 13 se muestra el diagrama de componente del sistema:

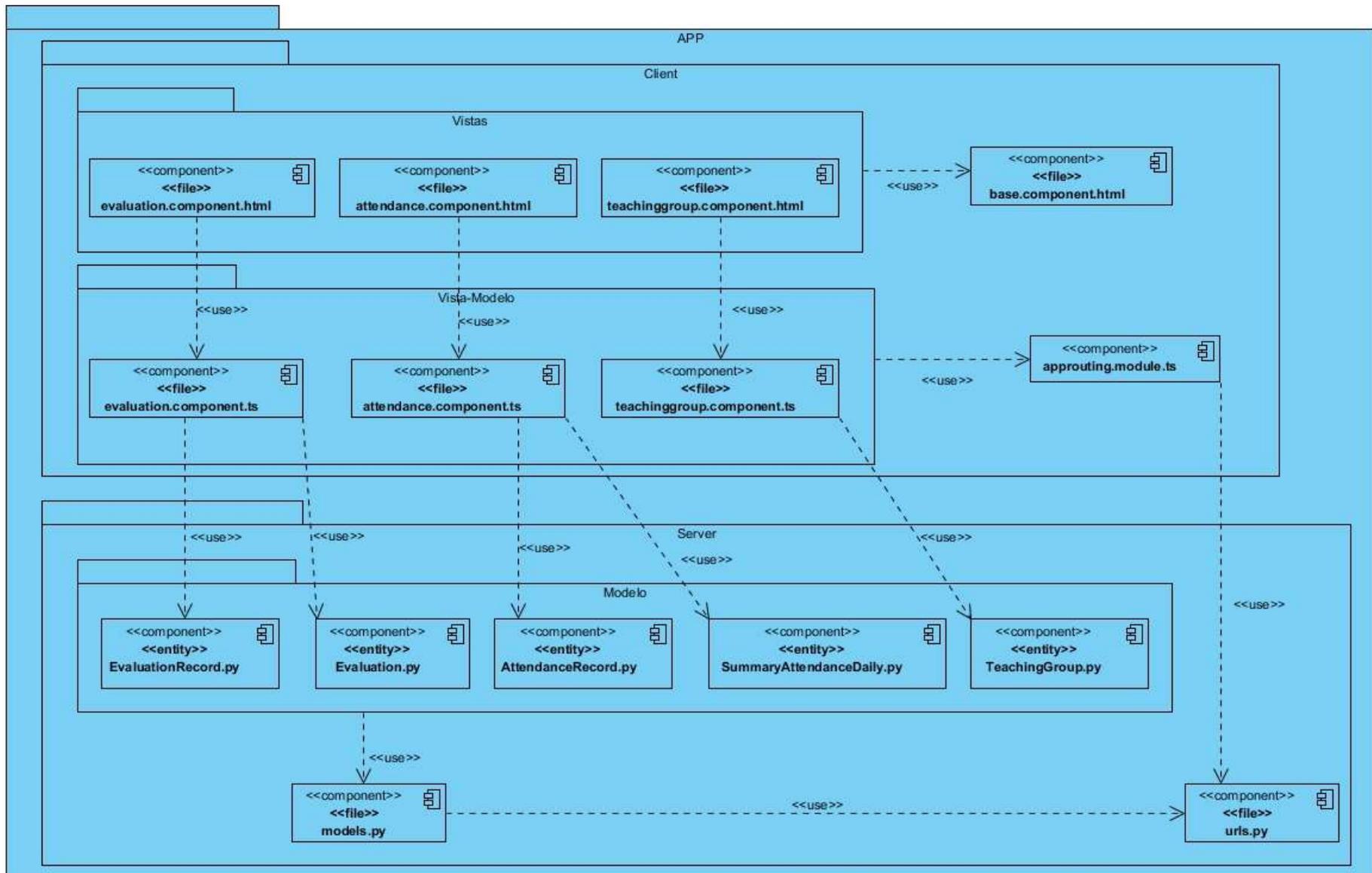


Figura 13. Diagrama de componentes

3.2. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada[33]. En el desarrollo del proyecto se definieron los siguientes estándares por los especialistas del Centro de Tecnologías para la Formación (FORTES):

- Se debe utilizar como idioma el inglés.
- 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.
 - Antes de un comentario de bloque o de un comentario de una línea.
 - Entre las distintas secciones lógicas de un método para facilitar la lectura.
- Se debe dar un espacio en blanco en la siguiente situación:
 - Entre una palabra clave del lenguaje y un paréntesis.
- Respecto a las normas de inicialización, declaración y colocación de variables, constantes, clases y métodos:
 - Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.
 - Los nombres de las variables declaradas deben aparecer totalmente en minúscula separando las palabras con un guion bajo ("_").
 - Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúscula.
 - Los nombres de las clases deben ser simples y descriptivos. Usar palabras completas, evitar acrónimos y abreviaturas.
 - Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula y la primera letra de las siguientes palabras que lo forman en mayúscula.

- Evitar las líneas de más de ochenta caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

3.3. Estrategia de Pruebas

Las pruebas de software son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo. Una estrategia para las pruebas de software debe incluir pruebas a bajo nivel, que son necesarias para verificar que un pequeño segmento de código fuente se implementó correctamente, así como pruebas de alto nivel, que validan las principales funcionalidades del sistema a partir de los requerimientos del cliente.

La estrategia de pruebas pudiera verse como un proceso en espiral, al igual que el proceso de desarrollo de software, comenzando desde la ingeniería del sistema, el levantamiento de requisitos, el diseño de los requerimientos y la implementación de los mismos. Al mismo nivel de cada una de estas etapas de desarrollo de software se realizan las pruebas del sistema, de validación, de integración y de unidad. En la figura 14 se muestra la representación descrita anteriormente:

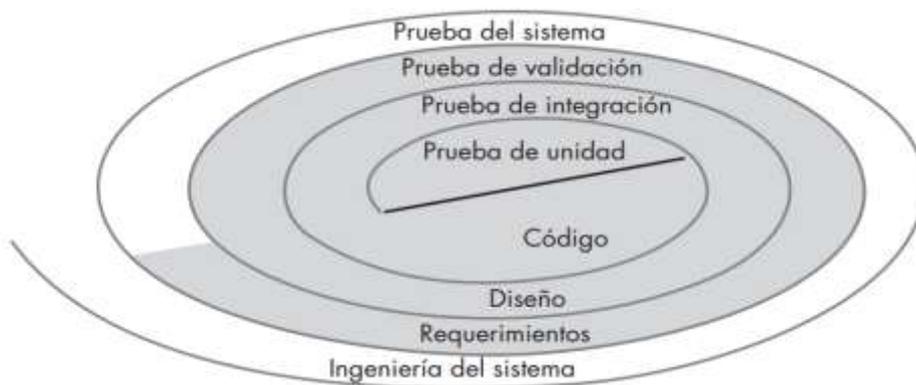


Figura 14. Estrategia de pruebas.

Para lograr verificar y validar la propuesta de solución se realizaron una serie de pruebas pertenecientes a cada uno de los niveles especificados anteriormente. Las pruebas aplicadas son las siguientes:

3.3.1. Pruebas Unitarias

Las pruebas unitarias son a bajo nivel (cercanas al código fuente de la aplicación). Este tipo de testing consiste en probar de forma individual las funciones y/o métodos (de las clases, componentes y/o módulos que son usados por un software). Cuando se planean las pruebas unitarias, se aísla la

funcionalidad hasta un punto en que no se pueda desglosar más, y entonces escribir pruebas a partir de ello. Justamente, el nombre de este tipo de prueba hace referencia a una "unidad de código", que es independiente del resto[42].

Estas pruebas verifican que el nombre de la función o método sea adecuado, que los nombres y tipos de los parámetros sean correctos, y así mismo el tipo y valor de lo que se devuelve como resultado. Para llevar a cabo esta prueba se utilizarán las pruebas de caja blanca, utilizando la técnica del camino básico.

Pruebas de caja blanca

Estas pruebas intentan garantizar que se ejecuten al menos una vez todos los caminos independientes que presente el módulo y que todas las estructuras de datos internas sean usadas. Se realizaron pruebas a las principales funcionalidades del módulo utilizando la técnica del camino básico, para lo cual es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática.

El cálculo de la complejidad ciclomática se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como: $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G también se define como: $V(G) = P + 1$, donde P es el número de nodos predicado (son los nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G .

A crear el grafo de flujo basándose en la unidad de códigos seleccionada, se calcula la complejidad ciclomática ($V(G)$) y se determina que la funcionalidad presenta una complejidad ciclomática de n , lo que deriva que existen a lo sumo n caminos lógicos por donde ejecutarse dicha funcionalidad. Al tener los caminos ya definidos se procede a realizar los casos de prueba para la funcionalidad en cuestión. Siempre se debe elaborar al menos un caso de prueba por cada camino básico.

3.3.2. Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software y, al mismo tiempo, realizar pruebas para descubrir errores asociados con la interconexión. El objetivo es tomar componentes probados por unidades y construir la estructura del programa que se diseñó. A menudo hay una tendencia a intentar la integración no incremental; es decir, para construir el programa usando un enfoque de "Big Bang". Todos los componentes se combinan de antemano. Todo el programa se prueba como un todo. Y el caos suele resultar. Se encuentra un conjunto de

errores. La corrección es difícil porque el aislamiento de las causas se complica por la vasta extensión de todo el programa. Una vez que se corrigen estos errores, aparecen otros nuevos y el proceso continúa en un bucle aparentemente interminable.

La integración incremental es la antítesis del enfoque del “Big Bang”. El programa se construye y se prueba en pequeños incrementos, donde los errores son más fáciles de aislar y corregir. Es más probable que las interfaces se prueben completamente y se puede aplicar un enfoque de prueba sistemático[33].

Se emplea el método de integración incremental por las facilidades que brinda al desarrollador de realizar pruebas al programa menos complejas y obteniendo mejores resultados. A continuación, se presentan dos estrategias del método seleccionado.

Top-down (descendente): La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del software. Los módulos se integran moviéndose hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal). Los módulos subordinados al módulo de control principal se incorporan a la estructura ya sea en profundidad o primero en amplitud[33]. El proceso de integración se realiza en una serie de cinco pasos:

1. El módulo de control principal se utiliza como controlador de prueba y los apéndices se sustituyen por todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero en profundidad o primero en amplitud), los apéndices subordinados se reemplazan uno a la vez con componentes reales.
3. Las pruebas se realizan a medida que se integra cada componente.
4. Al completar cada conjunto de pruebas, se reemplaza otro apéndice con el componente real.
5. Se pueden realizar pruebas de regresión para garantizar que no se hayan introducido nuevos errores.

El proceso continúa desde el paso 2 hasta que se construye toda la estructura del programa.

Bottom-up (ascendente): Las pruebas de integración ascendente, como su nombre lo indica, comienzan la construcción y las pruebas con módulos atómicos (es decir, componentes en los niveles más bajos en la estructura del programa). Debido a que los componentes se integran de abajo hacia arriba, la funcionalidad proporcionada por los componentes subordinados a un nivel dado siempre está disponible y se elimina la necesidad de apéndices[33]. Una estrategia de integración de abajo hacia arriba se puede implementar con los siguientes pasos:

1. Los componentes de bajo nivel se combinan en grupos (a veces llamados construcciones) que realizan una sub-función de software específica.
2. Un controlador (un programa de control para pruebas) está escrito para coordinar el ingreso y salida de prueba.
3. Se prueba el conjunto o construcción.
4. Los controladores se eliminan y los grupos se combinan moviéndose hacia arriba en la estructura del programa.

En este caso se empleará un acercamiento práctico (a veces llamado Sandwich testing) que utiliza Topdown para los niveles superiores de la estructura del programa y Bottom-up para los niveles inferiores. Combinando lo mejor de ambas técnicas[33]. La prueba se encontrará dividida en tres fases:

- **Interacción de usuario:** En esta fase se revisan los datos de entrada y salida, el formato de presentación de los datos, el procesamiento de errores y la presentación de los mismos.
- **Procesamiento de datos:** En esta fase se revisan la obtención de datos, manipulación del formato de los datos, determinación de condiciones de procesamiento, acciones requeridas como consecuencia de las condiciones encontradas.
- **Comunicación:** En esta fase se revisan el traspaso de datos entre funciones y clases, la llamada a funciones, las rutas de comunicación entre clases y funciones.

En cada prueba se deberán aplicar los siguientes criterios:

- **Integridad de interfaz:** Son comprobadas las interfaces internas y externas de cada subsistema o funcionalidad que es integrada al módulo.
- **Validación funcional:** Se aplican pruebas diseñadas para descubrir errores funcionales.
- **Contenido de información:** Se aplican pruebas diseñadas para descubrir errores asociados con estructuras de datos locales o globales.
- **Desempeño:** Se aplican pruebas durante el diseño del software para verificar el desempeño del programa se encuentre entre los límites establecidos.

3.3.3. Pruebas de aceptación

Cuando se construye software a medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Las realiza el usuario final en lugar del responsable del desarrollo del sistema, una prueba de aceptación puede ir desde un informal

“paso de prueba” hasta la ejecución sistemática de una serie de pruebas bien planificadas. De hecho, la prueba de aceptación puede tener lugar a lo largo de semanas o meses, descubriendo así errores acumulados que pueden ir degradando el sistema[33].

La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado prueba alfa y beta para descubrir errores que parezca que solo el usuario final puede descubrir. La prueba alfa se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y los problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado[33].

A la propuesta de solución le serán aplicadas pruebas de aceptación de tipo alfa, pues el cliente, en este caso el proyecto, junto al desarrollador realizarán pruebas guiadas por casos de prueba mediante el método de caja negra, utilizando la técnica de partición equivalente.

Pruebas de caja negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación [33].

De las técnicas de prueba de caja negra se utiliza la técnica de partición equivalente la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

3.3.4. Pruebas de Rendimiento

Con la realización de las pruebas de rendimiento se analiza el comportamiento de los servidores y las deficiencias que generan el bajo rendimiento. Este tipo de pruebas permiten identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones de negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras, determinar el cumplimiento de los objetivos de rendimiento[43].

En esta prueba se ejecutan las pruebas de carga y estrés que analizamos a continuación:

- **Pruebas de estrés:** Permite identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o procesos. Lo que posibilita asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio[43].
- **Pruebas de carga:** Permite identificar la capacidad de recuperación del sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Con la ejecución de estas pruebas se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella[43].

3.3.5. Pruebas de Seguridad

Las pruebas de seguridad garantizan que los usuarios estén restringidos a funciones específicas o que su acceso esté limitado. Sólo aquellos usuarios autorizados por el sistema son capaces de ejecutar las funcionalidades disponibles. El objetivo fundamental de este tipo de pruebas es comprobar los niveles de seguridad lógica del sistema. Para la realización de esta prueba se utilizará la herramienta Acunetix Web Vulnerability Scanner 8.0.

La aplicación de las pruebas de seguridad permite detectar las vulnerabilidades a las que se ve expuesto el sistema, así como tomar medidas para la disminución de amenazas de ataques, para la realización de esta prueba se tendrán en cuenta las siguientes categorías de vulnerabilidades:

- **Transición insegura de HTTP a HTTPS:** Esta vulnerabilidad permite a un atacante reemplazar el destino del formulario.
- **Denegación de servicios:** Esto provoca que los usuarios no accedan a la información deseada.
- **Integridad de la información:** Esta forma de ataque conlleva a que la información almacenada se pueda ver afectada, ya sea modificada o eliminada.

3.4. Estrategia de Validación

El principal objetivo del flujo de pruebas es evaluar la calidad del producto a través de la búsqueda de errores, la validación del cumplimiento de los requisitos y la validación del desempeño. Si bien es importante comprobar que se desarrolla un producto correctamente, también es esencial corroborar que se desarrolló el producto correcto, para ello es aconsejable aplicar técnicas que permitan medir la satisfacción del cliente, para ello se empleará la técnica Iadov.

Esta técnica se compone de cinco preguntas: tres cerradas y dos abiertas, las cuales se reformulan en la investigación para valorar el grado de satisfacción de los clientes sobre un tema en específico.

Las dos preguntas abiertas son de carácter complementarias, dichas preguntas abiertas permiten profundizar en la naturaleza de las causas que originan los diferentes niveles de satisfacción[44]. Ellas pueden ser clasificadas en 2 tipos:

- Causas relativas
- Causas externas

Una vez establecidas las preguntas se conforma el cuadro lógico de ladov presente en la figura número 15 y el número resultante de la interrelación de las tres preguntas, indica la posición de los sujetos en la escala de satisfacción. La escala de satisfacción está dada por los criterios:

- Máxima satisfacción.
- Más satisfecho que insatisfecho.
- No definida.
- Más insatisfecho que satisfecho.
- Máxima insatisfacción.
- Contradictoria.

Pregunta Cerrada 3	Pregunta Cerrada 1								
	No			No Sé			Si		
	Pregunta Cerrada 2								
	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 mas 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	6
No se que decir	2	3	6	3	3	3	6	3	4

Figura 15. Cuadro lógico de ladov

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:

Tabla 3. Niveles de satisfacción

Índice de satisfacción	Escala
Máxima satisfacción	1
Más satisfecho que insatisfecho	0.5
No definido y contradictorio	0

Más insatisfecho que satisfecho	-0.5
Máxima insatisfacción	-1

La satisfacción grupal (ISG) se calcula por la siguiente fórmula:

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

- A representa el número de sujetos con índice individual 1
- B representa el número de sujetos con índice individual 2
- C representa el número de sujetos con índice individual 3 ó 6
- D representa el número de sujetos con índice individual 4
- E representa el número de sujetos con índice individual 5
- N representa el número total de sujetos del grupo

El índice de satisfacción grupal arroja valores entre + 1 y - 1. Los valores que se encuentren 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.

3.5. Conclusiones del capítulo

En el presente capítulo se abordaron elementos de la implementación del módulo Control Docente para el sistema de gestión académica del Ministerio de Educación de la República de Cuba, así como las pruebas a realizar. Debido a esto se arriban a las siguientes conclusiones:

- Las identificaciones de estándares guiarán la etapa de desarrollo del software.
- La estrategia de pruebas definida permitirá identificar y corregir oportunamente errores que puedan aparecer en el módulo.
- La estrategia de validación de las pruebas definida permitirá garantizar que los resultados arrojados por las pruebas sean correctos.

Conclusiones Generales

Una vez realizadas las tareas trazadas en la presente investigación se concluye lo siguiente:

- El estudio de diferentes sistemas para la gestión del Control Docente permitió obtener una idea temprana de los principales conceptos asociados a la investigación.
- El análisis de las tecnologías, herramientas y lenguajes permitió definir la arquitectura de la propuesta de solución.
- El levantamiento de requisitos permitió definir el comportamiento del sistema, obteniéndose 22 requisitos funcionales y 11 requisitos no funcionales.
- Se definió la estrategia de pruebas con el objetivo de garantizar el correcto funcionamiento del sistema, así como la estrategia de validación de dichas pruebas.

Recomendaciones

Con el fin de mejorar el resultado la investigación se realizan las siguientes recomendaciones:

- La implementación de las distintas funcionalidades pertenecientes al módulo Control Docente.
- Aplicar la estrategia de prueba definida, así como la estrategia de validación.
- Ampliar el alcance del sistema para las enseñanzas universitarias.

Bibliografía

- [1] “Cuba Tasa de alfabetización - Población.” [Online]. Available: https://www.indexmundi.com/es/cuba/tasa_de_alfabetizacion.html. [Accessed: 03-Dec-2019].
- [2] “Fidel Castro reformador social.”
- [3] “Misión | Ministerio de Educación de la República de Cuba,” 19-Feb-2019. [Online]. Available: <https://www.mined.gob.cu/mision/>. [Accessed: 03-Dec-2019].
- [4] Ministerio Educacion, “Anuario Estadístico de curso 2018-2019.” 2018.
- [5] “Definición de control.” [Online]. Available: <https://definicion.de/control/>. [Accessed: 09-Dec-2019].
- [6] “Docente: Concepto, Requisitos y Factores.” [Online]. Available: <https://concepto.de/docente/>. [Accessed: 09-Dec-2019].
- [7] “Concepto de gestión.” [Online]. Available: <https://definicion.de/gestion/>. [Accessed: 09-Dec-2019].
- [8] “Definicion de sistema,” 2013. [Online]. Available: <https://definicion.de/Sistema/>.
- [9] M. Enguita, “La Organización Escolar: Agregado, Estructura Y Sistema,” vol. 320, no. 255–67, 1999.
- [10] Carr, Wilfred, and Stephen Kemmis, “Becoming Critical: Education,” Knowledge and Action Research, Falmer, London, 1986.
- [11] Sergio Luján-Mora, “Definicion de evaluacion,” 2015. [Online]. Available: <https://definicion.de/evaluacion/>.
- [12] Vedung, Evert, “Evaluación de Políticas Públicas Y Programas.” Insero, 1997.
- [13] Sergio Luján Mora, “Definición ABC. Tu Diccionario Hecho Fácil,” 2007. [Online]. Available: <https://www.definicionabc.com/social/deberes.php>.
- [14] G. I. Martínez Barreno, “Desarrollo e implantación de un sistema automatizado de control de matrículas, pensiones y notas para la Unidad Educativa González Suárez,” 2006.
- [15] A. Extremeño, M. J. Amante, and A. F. da Costa, “La Universidad del siglo XXI: nueva docencia, nuevo bibliotecario,” vol. 36, p. 5, Jun. 2013.
- [16] R. A. Saavedra and G. M. G. Mora, “Sistema de gestión de la Nueva Universidad (SIGENU v4.0.0) como herramienta para la automatización de los procesos docentes y estadísticos de la Universidad de Granma,” vol. 2, no. 4, pp. 130–142, Dec-2018.
- [17] “GESTACAD. SISTEMA PARA LA GESTIÓN ACADÉMICA Resumen,” *studylib.es*. [Online]. Available: <https://studylib.es/doc/332766/gestacad.-sistema-para-la-gestion-academica-resumen>. [Accessed: 04-Dec-2019].
- [18] Y. P. Vera, S. M. P. Pérez, and J. E. S. Díaz, “Sistema de Gestión Académica de Pregrado de la Universidad de las Ciencias Informáticas Academic Undergraduate Management System of the University of Informatics Sciences.”
- [19] A. C. Vitulloch, I. C. Torres, and A. L. Davila, “Modelo de gestión económica de la empresa estatal socialista de alta tecnología a la luz de los Lineamientos del VI Congreso del Partido Comunista de Cuba (PCC),” *Econ. Desarro.*, vol. 148, no. 2, pp. 94–123, 2012.
- [20] Digitala, Bizkaia Empresa, and Parque Tecnológico de Zamudio. n.d., “Django & Google App Engine.”
- [21] T. Ludeña and J. Luis, “Sistema informático web de gestión de incidencias usando el FRAMEWORK angulares y NODEEJS para la empresa REDTEAM SOFTWARE LLC,” 2016.
- [22] “¿Qué es HTML5?” [Online]. Available: <https://www.arkaitzgarro.com/html5/capitulo-1.html>. [Accessed: 04-Dec-2019].
- [23] JetBrains, “PyCharm: The Python IDE for Professional Developers by JetBrains.” 2020. [Online]. Available: <https://www.jetbrains.com/pycharm/>. [Accessed: 09-Mar-2020].
- [24] “Introducción a CSS,” 04-Dec-2019. [Online]. Available: <https://uniwebsidad.com/libros/css/capitulo-1?from=librosweb>.
- [25] I. Jacobson, G. Booch, and J. Rumbaugh, “El lenguaje unificado de modelado.” Man. Ref., 2000.
- [26] “Typescript: Easy content management.” [Online]. Available: <http://typescript.org/>. [Accessed: 04-Dec-2019].

- [27] "Json." [Online]. Available: <https://www.json.org/json-es.html>. [Accessed: 04-Dec-2019].
- [28] "PostgreSQL: The world's most advanced open source database." [Online]. Available: <https://www.pgadmin.org/>. [Accessed: 04-Dec-2019].
- [29] Nedelcu, Clement, "Nginx HTTP Server." 2015.
- [30] pgAdmin Development Team, "pgAdmin PostgreSQL Tools," 2007. [Online]. Available: <http://www.pgadmin.org/>.
- [31] "Free UML Modeling Software - Visual Paradigm Community Edition." [Online]. Available: <https://www.visual-paradigm.com/editions/community/>. [Accessed: 04-Dec-2019].
- [32] Escuela de Python, "PyCharm: uno de los mejores IDE para Python," 2018. [Online]. Available: <https://www.escuelapython.com/pycharm-uno-de-los-mejores-ide-para-python/>.
- [33] B. R. M. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. New York: McGraw-Hill Education, 2015.
- [34] RODRIGUEZ SANCHEZ, TAMARA., "Metodología de Desarrollo Para La Actividad Productiva En La UCI." 2015.
- [35] I. Sommerville, *Ingeniería del software*. Pearson educación, 2005.
- [36] E. Marini, "El modelo cliente/servidor," *Recuperado el*, vol. 5, 2012.
- [37] S. Luján-Mora, *Programación en Internet: clientes web*. Editorial Club Universitario, 2001.
- [38] Erich, GAMMA, R Helm, R Johnson, and J Vlissides, "Patrones de Diseño," 2003.
- [39] C. Larman, *UML y Patrones*. Pearson Educación ^ eMadrid Madrid, 2003.
- [40] R. OJEDA ZARATE, "SISTEMA DE CONTROL PARA SALAS DE TRABAJO EN EQUIPO VÍA INTERNET," Instituto Tecnológico de Colima, 2018.
- [41] Valdes , J., "Diagrama de Componentes." 2014.
- [42] Guillermo Armada González, "Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas," Universidad de las Ciencias Informaticas, 2019.
- [43] V&v Quality S.A, "Pruebas de rendimiento," 2015. [Online]. Available: <http://vyvquality.com/pruebas-rendimiento/>.
- [44] Dr. Alejandro López Rodríguez, "Niveles de satisfacción por las clases de educación física." [Online]. Available: <http://www.efdeportes.com/>.

ANEXOS

Anexo 1 Diagramas de Proceso de Negocio

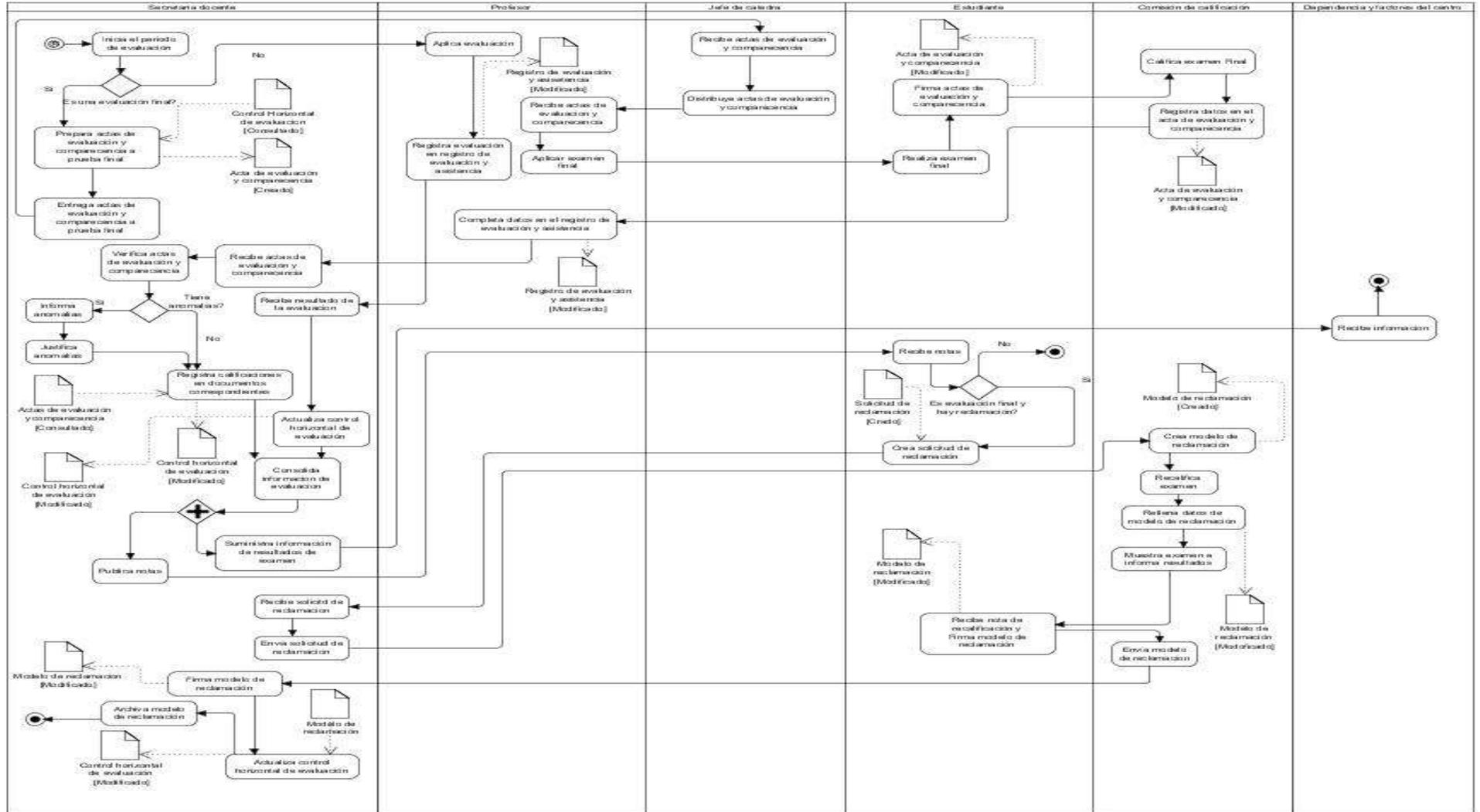


Figura 16. Diagramas de Proceso de Negocio: Control de evaluación

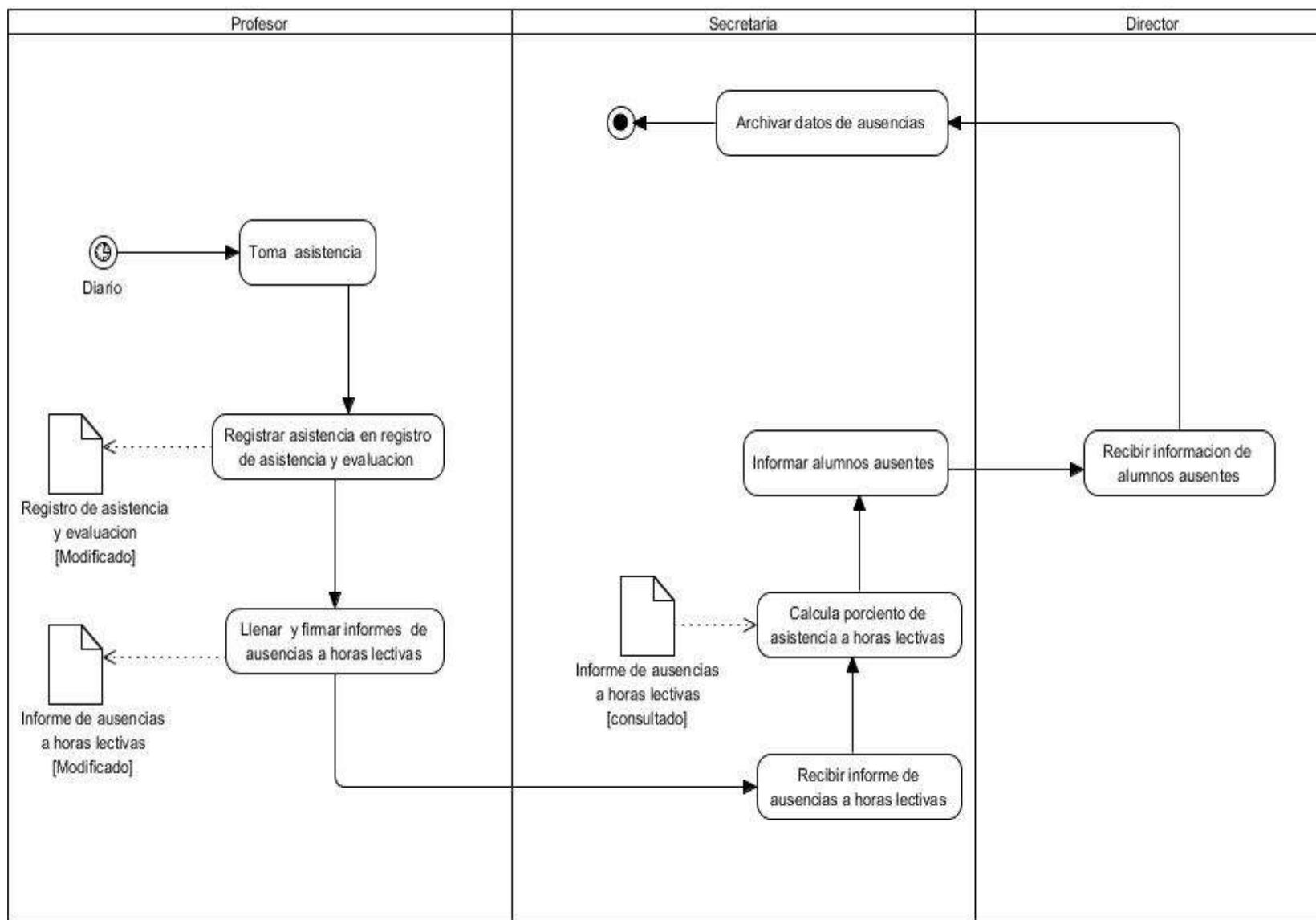


Figura 17. Diagramas de Proceso de Negocio: Control de Asistencia

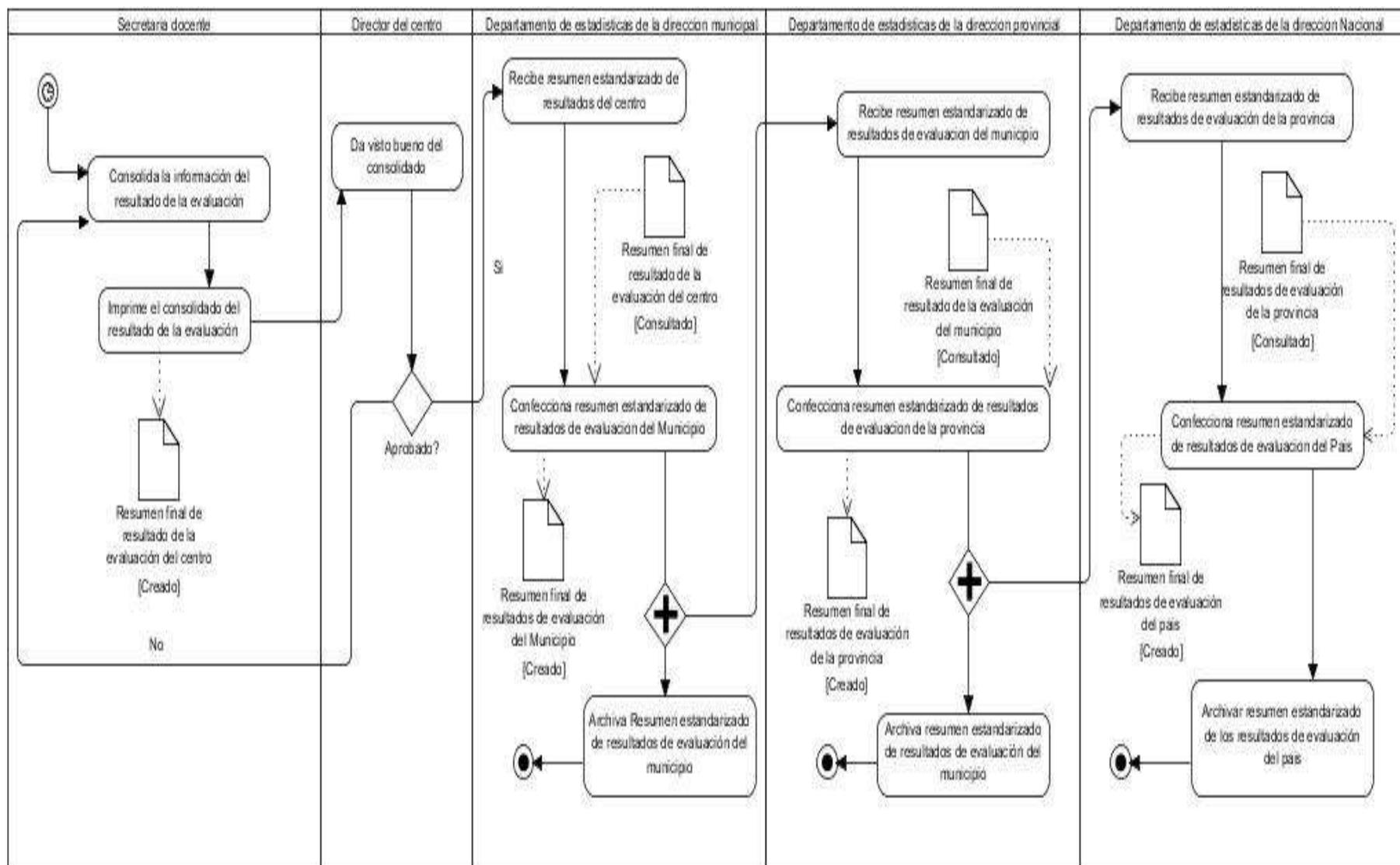


Figura 18. Diagramas de Proceso de Negocio: Consolidación de la Información

Anexo 2 Descripción de los Requisitos por Proceso

Tabla 4. Descripción de los requisitos por proceso: Registrar evaluación

Precondiciones		<ul style="list-style-type: none"> • El profesor debe estar registrado en el sistema. • El profesor solo registrar las evaluaciones de las asignaturas que imparte y solo a los estudiantes a los cuales le imparte dichas asignaturas. • El estudiante debe estar matriculado en un centro educacional perteneciente al Ministerio de Educación.
Flujo de eventos		
Flujo básico <<Registrar evaluación>>		
1	El requisito empieza en la página principal cuando el usuario selecciona el modulo control docente y dentro de este la opción registrar evaluación.	
2	El sistema mostrará los siguientes campos grado , asignatura y tipo de evaluación.	
3	El usuario seleccionara los valores de los campos de una lista de valores almacenados en base de datos y en caso de que los campos sean dejados en blanco ver flujo alterno 3.a.	
4	El sistema mostrará un filtro de estudiantes según los criterios de búsqueda especificados por el usuario en el formato nombre y apellidos y una lista desplegable con las posibles evaluaciones que pueden ser asignadas al estudiante.	
5	El usuario asignara las evaluaciones obtenidas a cada estudiante , todos los campos deben ser rellenados en caso contrario ver flujo alterno 5.a.	
6	El usuario pulsa el botón guardar y asigna las evaluaciones a cada estudiante	
7	El sistema regresa al paso 3.	
Pos-condiciones		
1.	Quedan registradas las evaluaciones de los estudiantes seleccionados.	

Flujos alternativos			
Flujo alternativo 3.a. Campos de filtro de búsqueda vacíos.			
1	El sistema mostrara un mensaje de advertencia “No pueden existir campos en blanco”.		
2	El usuario selecciona la opción aceptar y se marcan en rojo los campos que fueron dejados en blanco.		
3	El sistema regresa al paso 2 del flujo básico.		
Pos-condiciones			
•	N/A		
Flujo alternativo 5.a Campos de asignación de evaluaciones vacíos.			
1	El sistema mostrara un mensaje de advertencia “No se pueden asignar evaluaciones en blanco”.		
2	El usuario selecciona la opción aceptar y se marcan en rojo los campos que fueron dejados en blanco.		
•	El sistema regresa al paso 4 del flujo básico.		
Pos-condiciones			
1	N/A		
Validaciones			
#	Campo	Posibles valores	Valores no validos
1	Grado	Primero, octavo, decimo. onceno, tercero	-Seleccione-
2	Tipo de evaluación	Parcial, Final, Sistemática	-Seleccione-
3	Asignatura	Español, Matemática, Física, Química,	-Seleccione-

		Historia, Mecánica, Programación	
4	Evaluación	60, 80, MB,E,M,10	-Seleccione-
Conceptos	Evaluación	Visibles en la interfaz: Evaluación: String	
Requisitos especiales	RnF1, RnF3, RnF4, RnF6, RnF11		
Asuntos pendientes	N/A		
Prototipo elemental de interfaz gráfica de usuario			

Año académico

4

Grupo:*

2

Asignatura:*

Espanol

Tipo de evaluacion:*

Final

Asignar

Nombre y apellidos

Evaluación

Juan Pablo Leon Bencomo

-Seleccione-

Yudenis Regla Lara Leandro

100

Jose Ramon Martinez Hernandez

60

Resultados obtenidos 3 de 18

Pagina 1 de 6

Anterior

Siguiente

Guardar

