



Universidad de las Ciencias Informáticas

Facultad 1



TITULO

SISTEMA PARA EL CONTROL DE CUADROS DE LA FACULTAD 1

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

Autor

Jander Hernández Águila

Tutores

Ing. Estela Odelsa Martín Coronel
Ing. Maykel Ramírez Reyes

La Habana, Cuba
Septiembre de 2020

DECLARACIÓN DE AUTORÍA



DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) de Ciudad de La Habana, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste, firmo la presente a los _____ días del mes de _____ de 2020.

Jander Hernández Águila

Ing. Estela Odelsa Martín Coronel

Ing. Maykel Ramírez Reyes

OPINIÓN DEL USUARIO



OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El trabajo de diploma titulado, "SISTEMA DE GESTIÓN PARA EL CONTROL DE CUADROS DE LA FACULTAD 1" fue realizado en la Universidad de la Ciencias Informáticas. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un _____ %.

Los resultados de este trabajo de diploma le reportan a esta entidad los siguientes beneficios:

Y para que así conste, se firma la presente a los _____ días del mes de _____ de 2020.

Jander Hernández Águila

Firma

Ing. Estela Odelsa Martín Coronel

Firma

Ing. Maykel Ramírez Reyes

Firma

Pensamiento

Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte...



Resumen

La informatización de la sociedad representa hoy en día uno de los principales retos para nuestra Revolución dentro del desarrollo tecnológico sostenible. La gestión de la información constituye la base para fomentar el buen funcionamiento de una empresa y de esta forma obtener resultados con alto grado de calidad y eficiencia. El objetivo principal de esta investigación es desarrollar un sistema que contribuya a la gestión de la información de los cuadros en la Facultad 1 de la Universidad de las Ciencias Informáticas. Surge ante la necesidad de mejorar la gestión de los procesos de solicitud, presentación de la información y evaluación de los cuadros de la facultad, procesos que tienen lugar hoy día dentro de la institución. Para llevar a cabo estos procesos, actualmente no se cuenta con un sistema computacional que gestione la información que es necesario manejar, esto contribuye a que se vuelva muy engorroso elaborar informes a nivel central y provoca, además, que estos informes no se ajusten en su totalidad a la realidad objetiva, de ahí la meta principal de la presente investigación. Para la búsqueda y recopilación de información se emplearon los métodos de investigación: analítico-sintético, inductivo-deductivo, histórico-lógico y entrevista. Se llevó a cabo un estudio de varios sistemas de gestión de información, centrándose el mismo fundamentalmente en su estructura y funcionamiento para adquirir una mayor visión a la hora de realizar la propuesta de solución. A partir del desarrollo de este trabajo se obtuvo como resultado una aplicación web, la cual soluciona los aspectos establecidos en la problemática antes planteada y constituye a la vez un aporte tecnológico a la Facultad 1 de la Universidad de las Ciencias Informáticas.

Palabras clave: control, cuadro, gestión de información, sistema informático

Abstract

The informatization of society today represents one of the main challenges for our Revolution within sustainable technological development. Information management is the basis for promoting the proper functioning of a company and thus obtaining results with a high degree of quality and efficiency. The main objective of this research is to develop a system that contributes to the information management of the managers in the School 1 of the University of Computer Science. It arises from the need to improve the management of application processes, presentation of information and evaluation of the faculty, processes that take place today within the institution. In order to carry out these processes, there is currently no computer system to manage the information that needs to be handled. This contributes to the fact that it becomes very difficult to prepare reports at a central level and also causes these reports not to be completely in line with objective reality, hence the main purpose of this research. For the search and collection of information, the following research methods were used: analytical-synthetic, inductive-deductive, historical-logical and interview. A study of several information management systems was carried out, focusing mainly on their structure and operation in order to acquire a greater vision at the time of carrying out the proposal of solution. From the development of this work, a web application was obtained as a result, which solves the aspects established in the problem before mentioned and constitutes at the same time a technological contribution to the School 1 of the University of Computer Science.

Keywords: control, cadre, information management, computer system

TABLA DE CONTENIDOS

INTRODUCCIÓN	- 8 -
CAPÍTULO I: MARCO TEÓRICO REFERENCIAL DEL SISTEMA DE CONTROL DE LOS CUADROS DE LA FACULTAD 1	- 13 -
1.1 Introducción del capítulo	¡Error! Marcador no definido.
1.2 Conceptos asociados	- 16 -
1.3 Análisis de sistemas homólogos	- 18 -
1.4 La Gestión de cuadro	¡Error! Marcador no definido.
1.4.1 Política de Cuadros en el Ministerio de Educación Superior	- 13 -
1.5 Funcionamiento del Trabajo con los Cuadros del Estado y el Gobierno en la Facultad 1 de la UCI	- 14 -
1.6 Situación actual del proceso de control de cuadros en la Facultad 1	¡Error! Marcador no definido.
1.7 Herramientas, Lenguajes, metodologías y Entornos de desarrollo	- 22 -
1.7.1 Metodologías de Desarrollo de Software	- 22 -
1.7.2 Lenguaje de programación del lado del cliente	- 24 -
1.7.3 Lenguaje de programación del lado del servidor	- 26 -
1.7.4 Entorno de desarrollo	- 27 -
1.7.5 Administración y Sistema Gestor de Base de Datos (SGBD)	- 28 -
1.7.6 Lenguaje de Modelado	- 29 -
1.7.7 Herramienta CASE	- 30 -
1.7.8 Herramienta para el prototipado	- 30 -
1.8 Conclusiones del capítulo	- 31 -
CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA DE CONTROL DE CUADRO DE LA FACULTAD 1 .	- 32 -
2.1 Introducción del capítulo	¡Error! Marcador no definido.
2.2 Descripción de la propuesta de solución	- 32 -
2.3 Modelo conceptual	- 33 -
2.3.1 Modelo de dominio	- 33 -

2.3.2 Conceptos del dominio.....	- 34 -
2.4 Requisitos del sistema.....	- 36 -
2.4.1 Requisitos Funcionales (RF).....	- 36 -
2.4.2 Requisitos No Funcionales (RNF).....	- 41 -
2.5 Modelo de Casos de Uso del Sistema.....	- 43 -
2.5.1 Descripción de los actores del sistema.....	¡Error! Marcador no definido.
2.5.2 Diagrama de actores del sistema.....	¡Error! Marcador no definido.
2.5.3 Diagrama de casos de uso del sistema.....	- 43 -
2.5.4 Patrones de casos de uso del sistema.....	- 44 -
2.5.5 Descripción textual de los casos de uso del sistema.....	- 45 -
2.6 Arquitectura del sistema.....	- 49 -
2.7 Patrones de diseño.....	- 50 -
2.7.1 Patrones GOF.....	- 51 -
2.7.2 Patrones de asignación de responsabilidades GRASP.....	- 52 -
2.8 Modelado del diseño para la propuesta de solución.....	- 54 -
2.8.1 Diagrama de clases del diseño.....	- 54 -
2.9 Modelo de datos.....	- 55 -
2.9.1 Modelo de Entidad/Relación.....	- 55 -
2.10 Conclusiones parciales.....	- 57 -
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE CONTROL DE CUADROS DE LA FACULTAD 1.	- 59 -
CONCLUSIONES GENERALES.....	- 66 -
RECOMENDACIONES.....	- 67 -
REFERENCIAS BIBLIOGRÁFICAS.....	¡Error! Marcador no definido.
Anexos.....	¡Error! Marcador no definido.

INTRODUCCIÓN

Cada vez son más las instituciones que hoy en día llevan a cabo todo su proceso y control laboral de forma digital. Entre tanto personal cualificado en la sociedad cubana, se encuentran de manera relevante aquellos que ocupan cargos administrativos y tienen responsabilidades que responden a un proceso de formación de ideales. Esto hace que dichas personas se conviertan en “cuadros”, teniendo en cuenta sus responsabilidades dentro de la sociedad para con la institución en la que laboran.

Siempre que se trate el tema de control y política de cuadros, es necesario referirse a la definición de cuadro de Ernesto Guevara, cuando en "El cuadro, columna vertebral de la Revolución" relaciona que "... ¿qué es un cuadro? Debemos decir que, un cuadro es un individuo que ha alcanzado el suficiente desarrollo político como para poder interpretar las grandes directivas emanadas del poder central, hacerlas suyas y transmitir las como orientación a la masa. Es un individuo de disciplina ideológica y administrativa, que conoce y practica el centralismo democrático y sabe valorar las contradicciones existentes en el método para aprovechar al máximo sus múltiples facetas; que sabe practicar en la producción el principio de la discusión colectiva y decisión y responsabilidad únicas, cuya fidelidad está probada y cuyo valor físico y moral se ha desarrollado al compás de su desarrollo ideológico, de tal manera que está dispuesto siempre a afrontar cualquier debate y a responder hasta con su vida de la buena marcha de la Revolución..." "el cuadro es la columna vertebral de la Revolución..."(MARTÍN, 2016).

Los cuadros tienen la responsabilidad de desarrollar el liderazgo político de forma tal que con su palabra y acción estén a la vanguardia de sus colectivos, contribuyendo con su superación permanente a la reflexión, análisis y valoración de los principales acontecimientos político, ideológico, social y económico de Cuba y el mundo.

Los cuadros cubanos tienen una elevada responsabilidad ante la sociedad que abarca el orden económico, político y social. Poseen encargos variados como son la aplicación de la Planificación Estratégica y la Dirección por Objetivos basada en valores, la Estrategia Nacional de Preparación y Superación de Cuadros y Reservas y la aplicación del Sistema de Trabajo con los Cuadros (STC), entre otros, lo que demanda su influencia multiplicadora y acción positiva sobre las personas que componen la organización (BORRELL, 2006).

Según el Decreto Ley No. 196/1999, el artículo 27 define que: la evaluación de los cuadros es un proceso ininterrumpido que se realizará a partir de los resultados del trabajo y su desempeño individual,

constatando el progreso de la actividad que dirige o realiza de forma cotidiana y sistemática, teniendo en cuenta su actitud, ejemplaridad y autoridad personal (Ruz, 1999).

Los cuadros serán evaluados periódicamente sobre la base del cumplimiento de los requisitos y condiciones generales y específicas que se tuvieron en cuenta para su designación o elección.

Asimismo, el artículo 28 establece: el proceso de evaluación constituirá la vía para asegurar el estudio y la profundización, por cada cuadro, de los preceptos y principios establecidos en el Código de Ética, que deben ser reflejados en el análisis autocrítico de la autoevaluación presentada por él, y en la ratificación oficial de su compromiso, al momento de recibir las conclusiones de su evaluación.

El proceso de preparación y superación es un proceso sistemático y continuo de formación y desarrollo de los cuadros y sus reservas, que ocupa el lugar rector en el sistema de formación de cuadros, pues moviliza al sistema de trabajo con los cuadros y garantiza su cualidad resultante: la ubicación en cada cargo de dirección, en cada momento y lugar, de la persona con más capacidad, mayor preparación y mejores condiciones para desempeñarlo.

Las formas de trabajo, el rigor técnico y los resultados satisfactorios de las organizaciones tienen mucho que ver con la manera en que opera el sistema de trabajo con los cuadros, lo que requiere coherencia entre el trabajo de estos y las principales políticas estatales. Así mismo, en la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), se encuentran trabajadores, los cuales laboran dentro del marco docente-educativo y al mismo tiempo, ocupan cargos y poseen responsabilidades que tributan a su condición como cuadros.

El decanato de la facultad 1 de la UCI, lleva a cabo el control de los trabajadores con la condición de cuadros durante todo el año, para mantener con calidad la gestión de la información de estos y evitar de esta forma un proceso engorroso y desorganizado. Para ello se mantiene actualizada toda la información de forma manual, y con apoyo de la herramienta informática Excel. Esto provoca que se genere una cantidad de documentos que deben ser archivados, existiendo duplicidad y pérdida de información. La información relacionada con la evaluación trimestral y anual de los cuadros debe ser archivada, además de llevar un control de los planes de trabajo de los mismos, el plan de resultado individual y la información relacionada con los expedientes de las reservas; de todos estos documentos debe existir un histórico de 5 años y son almacenados en archivos en formato duro sin existir una base de datos digital.

Actualmente, la facultad no cuenta con un sistema que permita archivar y gestionar todo el cúmulo de información que se genera como resultado del trabajo con los cuadros para su proceso de control en la

Facultad 1.

Teniendo en cuenta la **problemática** anteriormente expuesta, se define como problema a científico: ¿Cómo contribuir a la gestión de la información en el proceso de control de los cuadros de la Facultad 1?

El referido problema se manifiesta y constata en el **objeto de estudio** de dicha investigación, el cual se enmarca en la gestión del control de Recursos Humanos. Dicha investigación establece como **objetivo general**: desarrollar un sistema informático que contribuya a la gestión de la información en el proceso de control de los cuadros de la Facultad 1.

Se precisa como **campo de acción** los sistemas de gestión de información para el control de los cuadros en instituciones.

El **objetivo general** de esta investigación es posible mediante el cumplimiento de los siguientes objetivos específicos:

- Definir el marco teórico de la investigación para la gestión de la información en el proceso de control de los cuadros de la Facultad 1.
- Elaborar un estudio del arte de los principales sistemas similares de gestión de control de recursos humanos existentes a nivel nacional e internacional.
- Diseñar un sistema que contribuya a la gestión de la información en el proceso de control de los cuadros de la Facultad 1.
- Implementar un sistema teniendo en cuenta las funcionalidades identificadas para satisfacer las necesidades del cliente en cuanto al proceso de control de los cuadros.
- Verificar la calidad del sistema para el control de los cuadros de la Facultad 1.

La presente investigación establece como **hipótesis** que, la implementación de un sistema de gestión para el control de los cuadros contribuirá a disminuir la duplicidad y pérdida de la información asociada a los mismos, a la vez que aumentará el control en el proceso para el trabajo con los cuadros en la Facultad 1 de la Universidad de la Ciencias Informáticas.

Para dar cumplimiento a los objetivos planteados han sido utilizados los métodos científicos de la investigación: teórico y empírico.

Métodos Empíricos

Entrevistas: para identificar el proceso de control de cuadros de la Facultad 1, su contenido de trabajo y responsabilidades de acuerdo a los elementos que se deben tener en cuenta para su preservación. (Ver Anexo 1)

Observación: para conocer la realidad mediante la sensopercepción directa de los cuadros de la Facultad 1 y el proceso de control de los mismos, para lo cual se debe poseer algunas cualidades que le dan un carácter distintivo.

Métodos teóricos

Analítico-Sintético: para procesar la información relacionada con los sistemas de gestión y control de personal, algunos conceptos asociados a la investigación, así como para las conclusiones de manera general.

Inductivo-Deductivo: es empleado para realizar un análisis de las necesidades de la Facultad 1, de disponer de un sistema de gestión para el control de los cuadros.

Histórico-Lógico: se utiliza para el estudio de los antecedentes de las investigaciones relacionadas con los sistemas de gestión y control de información de personal cualificado, formando parte del propósito de esta investigación.

El presente trabajo de diploma está estructurado de la siguiente forma: Introducción, Capítulo I, Capítulo II, Capítulo III, Conclusiones generales, Recomendaciones, Bibliografía y Anexos.

Capítulo I: Marco Teórico Referencial sobre el Sistema de Control de Cuadros de la Facultad 1: en este capítulo se analizan los principales conceptos asociados al objeto de estudio para el dominio de la propuesta de solución; se hace un estudio del arte sobre el uso de los sistemas gestión de recursos humanos enfocándose en los sistemas para el control de los cuadros.

Capítulo II: Análisis y Diseño del Sistema de Control de Cuadros de la Facultad 1: se realiza una descripción del sistema a desarrollar, se exponen los fundamentos teóricos del modelo de dominio, se realiza el levantamiento de requisitos, así como la modelación del esquema de base de datos. Se realiza además una descripción del funcionamiento del sistema para el control de la información relacionada con los cuadros de la Facultad 1 y se representa la estructura de este con el empleo de la metodología seleccionada.

Capítulo III: Implementación y Prueba del Sistema de Control de Cuadros de la Facultad 1: en este capítulo se procede a la implementación a partir del diseño realizado. Se describen las pruebas a realizar,

con el objetivo de probar el correcto funcionamiento del sistema informático para el control de los cuadros de la Facultad 1 de la UCI.

CAPÍTULO I: MARCO TEÓRICO REFERENCIAL DEL SISTEMA DE CONTROL DE LOS CUADROS DE LA FACULTAD 1.

En este capítulo se exponen los elementos teóricos y conceptuales asociados al control de cuadros. Se realiza un estudio de los principales conceptos asociados para ayudar a entender mejor el objetivo propuesto, así como un estudio de los sistemas desarrollados anteriormente con fines de gestionar información acerca de los recursos humanos de una institución a través del estudio del estado del arte para analizar sus características, funcionalidades y utilidades actuales.

Teniendo en cuenta que la solución de esta investigación es un sistema web para el control de cuadros de la Facultad 1, se exponen además las diferentes herramientas computacionales, metodologías, lenguajes de programación, entre otros, que se emplean en el proceso de diseño y desarrollo de la solución planteada.

1.1 La gestión de cuadro

El cuadro es la pieza maestra del motor ideológico que es el Partido Unido de la Revolución. Es lo que se llama un tornillo dinámico de este motor; tornillo en cuanto a pieza funcional que asegura su correcto funcionamiento; dinámico a que no es un simple transmisor hacia arriba o hacia abajo de lemas y demandas, sino un creador que ayudará al desarrollo de las masas y a la información de los dirigentes, sirviendo de punto de contacto con aquellas. "Son considerados cuadros, los trabajadores que por sus cualidades, capacidad de organización y dirección, voluntad y compromiso expreso de cumplir los principios establecidos en el Código de Ética de los Cuadros del Estado, son designados o electos para ocupar cargos de dirección en la función pública, así como en los sistemas de las organizaciones económicas y unidades presupuestadas del Estado y del Gobierno" (CABRERA, 2013).

1.2 Política de Cuadros en el Ministerio de Educación Superior

El desarrollo de una correcta política en la formación, selección, ubicación y promoción de los cuadros, así como en su preparación, superación y desarrollo, ha sido considerado en la Tesis y Resolución del VII Congreso del PCC sobre la Política de Cuadros como una de las tareas fundamentales en la actividad de todos los organismos y organizaciones del país.

Generalidades

- Los órganos de cuadro del Ministerio de Educación Superior en todas sus instancias se regirán para el desarrollo de su trabajo por las directrices trazadas por el Partido y el Gobierno. Serán considerados como cuadros del Ministerio de Educación Superior y sus dependencias aquellos compañeros que tengan cargos de dirección administrativa, docente y/o de investigación.

- La dirección de cuadro del MES prestará principal atención y control a los cuadros de dirección docente y administrativa que estén en la nomenclatura del MES, así como los cuadros científico-pedagógicos que trabajen directamente en el organismo central y a los que desarrollan su labor en los centros de educación superior y que tengan las categorías principales de profesor Titular y Profesor Auxiliar y/o los grados científicos de Doctor o Candidato a Doctor en Ciencias.

- Todo compañero que tenga una preparación adecuada, iniciativa, perspectiva de desarrollo y una buena formación política e ideológica, debe ser considerado como una cantera de cuadro para los distintos cargos existentes.

- La preparación de los cuadros, tanto para una mayor eficiencia en los cargos que desempeñan, como para las posibles promociones, debe constituir una base fundamental junto a la selección y ubicación, de la política del MES y sus dependencias. (PCC, 2016)

1.3 Funcionamiento del trabajo con los Cuadros del Estado y el Gobierno en la Facultad 1 de la UCI

Para el cumplimiento de lo establecido en la Política de Cuadros, la UCI y en específico la Facultad 1, se rige por lo orientado en los documentos rectores de esta actividad, a los que se hace mención anteriormente. La Facultad 1 tiene como objetivo trabajar por lograr una consciente y previsoramente aplicación de la Política de Cuadros, de acuerdo con la premisa de lograr avances en la selección, preparación, superación y control de la actuación de los cuadros y reservas de la Proyección Estratégica de la UCI, centrando la atención principal en el fortalecimiento y elevación del prestigio y la ejemplaridad de los cuadros; la formación de una efectiva reserva, que posibilite la necesaria renovación con la inyección de nuevos cuadros, entre los acorde con sus cualidades y capacidad, debe darse prioridad a los jóvenes, las mujeres y los negros.

Actualmente, la plantilla de cuadros de la Facultad 1 está compuesta por 13 cargos aprobados, de ellos cubiertos 12 con 1 vacante, para un 92.3 % de completamiento. Tienen reservas un total de 13 cuadros, clasificándose como listos 18 de ellos. Se encuentran cubiertos por mujeres un total de 7 cargos que representan un 58.3%. La composición étnica está dada por 4 blancos y 8 entre mestizos y negros como se representa en la Figura 2. La composición política de los cuadros representa el 100% dada por 12 militantes del PCC y 1 de la UJC; el promedio de edad es 35 años, el nivel educacional lo componen 0 Técnicos Medios, referentes a grados científicos, 5 son Doctores, 4 son Máster y 3 son Ingenieros o Licenciados.

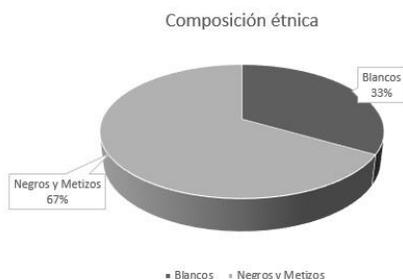


Figura 2: Composición de la plantilla de cuadros según la raza (Elaboración propia)

La plantilla de la reserva está integrada por 32 compañeros y compañeras, de ellos 6 son Cuadros. Compuesta genéricamente por 18 hombres y 14 mujeres, representando estas últimas un 43.75 %, y étnicamente por 12 blancos, 20 mestizos y negros. Militan en las filas del PCC 12 reservistas, en la UJC 20 que representan un 90.62 % del total y 3 no pertenecen a ninguna organización política como se puede ver en la Figura 3. El promedio de edad es de 36 años. Se considera como reserva lista para ser promovida un total de 18 que representan un 56.25 % del total.

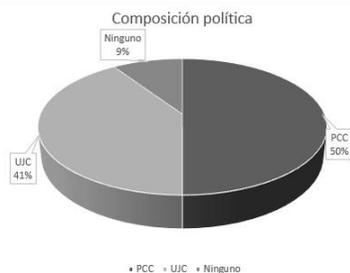


Figura 3: Esquema de la plantilla de Cuadros según la composición política (Elaboración propia)

1.4 Conceptos asociados

Para una mejor y mayor comprensión de la investigación teniendo en cuenta los elementos relacionados previamente, se tuvieron en cuenta algunos conceptos fundamentales que influyen en gran medida en el desarrollo y cumplimiento de los objetivos anteriormente planteados. Entre ellos se tienen:

Gestión

La gestión es el arte de saber lo que se quiere hacer y a continuación, hacerlo de la mejor manera y por el camino más eficiente (LÓPEZ, AIDA MANRIQUE, 2016).

Por otra parte, (ISO:9000, 2015) lo define como: actividades coordinadas para dirigir y controlar una organización.

La gestión puede ser tomada desde múltiples aristas, en este caso se hace necesario analizar la gestión de información.

Gestión de información

Es el proceso de organizar la información, evaluar, presentar, comparar los datos en un determinado contexto, controlar la calidad, veracidad, que sea oportuna, significativa, exacta y útil y que esté disponible en el momento que se le necesite. Ella se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y otras actividades en función de los objetivos estratégicos de una organización (LEDO, 2012).

Sistema de gestión

Un sistema de gestión es un conjunto de reglas y principios relacionados entre sí de forma ordenada, para contribuir a la gestión de procesos generales o específicos de una organización. Permite establecer una política, unos objetivos y alcanzar dichos objetivos (LÓPEZ, JORGE GONZÁLEZ, 2019)

Similarmente, (ISO:9001, 2015) establece que un sistema de gestión es el conjunto de elementos de una organización interrelacionados o que interactúan para establecer políticas, objetivos y procesos para lograr estos objetivos.

La administración tiene como funciones fundamentales el planeamiento, la organización, la dirección y control (FLORES, 2012).

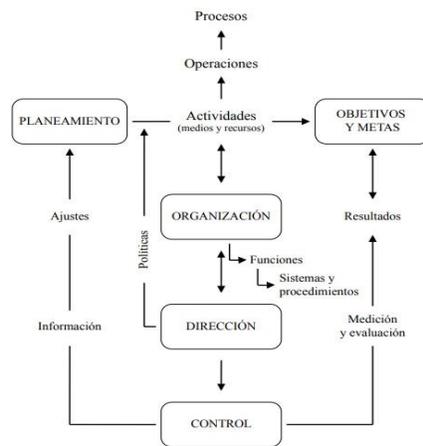


Figura 1: Funciones de la administración (FLORES, 2012)

Teniendo en cuenta lo anterior, es necesario analizar otros conceptos como los que se reflejan a continuación.

Proceso de Control

Al tratar sobre el proceso de control, (FLORES, 2012) expresa que «el proceso de control se aplica a toda la organización. Todos los aspectos del desempeño de una institución deben monitorearse y evaluarse en sus tres niveles jerárquicos principales: estratégico, administrativo (o funcional) y operativo».

1.5 Análisis de sistemas homólogos

A nivel internacional y nacional, existen sistemas computacionales con el objetivo de gestionar la información en determinadas insituciones y empresas. Hoy en día el desarrollo de las TIC ha traído como resultado un gran avance en el uso de técnicas y tecnologías, que permiten el análisis de la información con la calidad requerida. A partir del estudio realizado, se identificó que actualmente existen sitios web dedicados a la gestión de información y/o control en los recursos humanos de una empresa u organización, las cuales se describen a continuación:

Sistemas Internacionales

GestPeople

GestPeople es una herramienta de gestión de recursos humanos para todo tipo de empresas. Se puede definir la estructura organizativa de la empresa, definir los puestos profesionales de cada empleado, delimitar las competencias y establecer los objetivos tanto a corto como a largo plazo (TOLEDO, 2010).

Ofrece una amplia funcionalidad estructurada por módulos o secciones integrados entre ellos al 100%. Esta integración le permite acceder a la información desde diferentes sitios, como por ejemplo, navegar en el organigrama viendo datos generales del departamento y del puesto, información específica del perfil profesional o cualquier otro dato relacionado con la ficha del empleado.

Permite obtener informes sobre los empleados, los perfiles, las evaluaciones y objetivos, el mapa de ocupaciones, etc. o imprimir su organigrama (MATOS, 2011).

Tras el estudio de este sistema se puede constatar que el mismo persigue el objetivo común de gestionar la información que generan los RRHH. Este cuenta con solo algunas funcionalidades semejantes a las necesitadas: la gestión de los planes de trabajo, la gestión de las fichas con información de la vida laboral de cada empleado y la generación de documentos en diferentes formatos. Sin embargo, no cuenta con un método que permita la evaluación del empleado a partir de los documentos que definen dicha evaluación.

eMagister RRHH

Es un paquete de gestión de recursos humanos que pone el acento en la formación y los planes de desarrollo de los empleados. Organiza todos los informes en pestañas. Además de la lista de empleados, puedes generar gráficas de retribución. Permite gestionar la información de las personas de la organización y el personal histórico (datos personales y profesionales, currículum, puestos ocupados, salarios, competencias, etc.).

Comentado [A1]: referencia

A pesar de los beneficios de este sistema, actualmente el mismo no se encuentra disponible al alcance de los usuarios debido a que los administradores dejaron de dar soporte al *software*. Esto imposibilita el acceso a la información y estudio de funcionalidades y características en general.

ControlLaboral

Es un sistema basado en aplicación androide de acceso multidispositivo desde su móvil o tablet, para gestionar las horas laborales de los empleados y generar informes legales al instante. Los trabajadores de diferentes centros de trabajo pueden validar su jornada, vacaciones, bajas, horas de entradas y salidas. La aplicación genera informes legales con verificación jurídica de manera inmediata, perfectos para entregar a inspección de trabajo en caso de una auditoría por su parte. Se tiene acceso a las horas trabajadas de cada empleado, disponiendo de una bolsa de horas para el seguimiento de las horas extraordinarias. Genera los informes laborales necesarios para inspección de trabajo y los exporta a Excel, PDF y otros formatos (FERNÁNDEZ, 2018).

A pesar de las facilidades de este sistema, solo existe su desarrollo para aplicaciones móviles y otros dispositivos similares, lo cual a pesar de las ventajas que tiene, no es el entorno de desarrollo que se desea llevar para el sistema de control de los cuadros de la Facultad 1.

E-Valúame. Software online para evaluar el desempeño y el clima laboral.

Es un sistema orientado exclusivamente a la evaluación del desempeño de los empleados de cualquier institución (módulo para evaluar el desempeño), así como del clima laboral de los mismos atendiendo a las responsabilidades y cumplimiento de tareas (módulo para evaluar el clima laboral). Crea un cuestionario personalizado, lanza encuestas de evaluación del clima laboral a toda la organización con la posibilidad de conocer los resultados al instante.

Dentro de sus principales características se pueden describir dentro de las más importantes (E-VALÚAME, 2012):

- Biblioteca de competencias: permite introducir las competencias propias del empleado o las que E-Valúame proporciona.
- Notificaciones por email: hace uso del email para notificar los diferentes sucesos y estados por los que pasa una encuesta.
- Informes en PDF y Excel: permite acceder a informes en PDF y Excel a través de la web, así como obtener los mismos mediante impresión.

- Informes individuales y comparativos: genera un informe para cada evaluador y permite comparar los resultados de cada uno de ellos.

De forma concluyente, este sistema posee excelentes características y funcionalidades que son de gran utilidad para alcanzar parte de los objetivos del sistema que se pretende desarrollar. Tiene un enfoque específico por lo que no cumple con todos los requerimientos para alcanzar el logro de los objetivos propuestos.

HRider. Software de Evaluación del Desempeño y Feedback 360^o

Sistema en línea que evalúa competencias y objetivos por separado o juntos en la misma evaluación. Permite dar seguimiento al reabrir encuestas y enviar recordatorios, añadir y modificar encuestas sobre la marcha, modificar objetivos durante el proceso, establecer un flujo de trabajo de validación de objetivos, así como realizar un proceso de revisión para las reuniones de retroalimentación.

Incluye además un plan de acción donde los empleados y supervisores podrán colaborar definiendo objetivos y las acciones necesarias para lograrlos, registrar sus progresos y conocer fácilmente las acciones pendientes, completadas o atrasadas. Evalúa de forma fácil y profesional el clima laboral mediante el impulso de procesos de retroalimentación participativo, poniendo a las personas en el centro de la organización y logrando altos niveles de satisfacción laboral (HRIDER, 2020).

Atendiendo a los objetivos del sistema a implementar dentro de la presente investigación, se puede comprobar que este *software* no satisface todas las necesidades de gestión para el trabajo con los cuadros dentro de la Facultad 1. Posee una excelente propuesta de gestión evaluativa y control de empleados, las cuales se tienen en cuenta de manera positiva para el desarrollo de algunas funcionalidades. Esto ha posibilitado ampliar y profundizar las estrategias de diseño del sistema aunque no constituya en sí una solución inmediata para la problemática inicialmente expuesta.

Sistemas Nacionales

Módulo servidor para la gestión de la información en la Dirección de Cuadros de la Administración Provincial (AP) de Artemisa.

Este sistema tiene como objetivo mejorar los procesos de gestión de la información en la Dirección de Cuadros de la AP de Artemisa. Cuenta con las funcionalidades correspondientes a la gestión de los cuadros existentes en la provincia y sus municipios. Además, permite gestionar la superación, vacaciones,

evaluaciones, medidas disciplinarias, estimulaciones, chequeos médicos y movimientos de los cuadros en la provincia, y de esta forma garantizar la eficiencia y confiabilidad de esos procesos.

Teniendo en cuenta tales afirmaciones, se puede constatar que este sistema pudiera ajustarse en gran medida a la propuesta de solución de esta investigación de acuerdo a la situación que hoy tiene lugar en la Facultad 1 de la UCI, sin embargo, dicho sistema no satisface del todo las necesidades de información que presenta la facultad, pues el personal que labora en esa institución requiere que los reportes sean generados lo más flexible posible, o sea, que no exista una plantilla estricta de reportes sino que puedan ser generadas de acuerdo a las necesidades del momento. Por tales motivos no satisface todos los requerimientos, constituyendo de esta forma una desventaja para la investigación.

Sistema de Gestión de la Unión de Jóvenes Comunistas (UJC) de la Facultad 4

Esta herramienta posee como elemento a favor, que está desarrollada para la propia institución y alcanza algunos de los objetivos planteados en la investigación, como es el control de personas con cargos político-administrativos. Fue creada esencialmente para la gestión, control y recuperación de la información que se genera en los procesos llevados a cabo en la UJC de la Facultad 4. Este *software* no gestiona los reportes de actas, lo que trae consigo que con su uso, no se pueda realizar un resumen de los indicadores del cierre de funcionamiento en dicha facultad, tampoco hace referencia a ningún proceso que intervenga en el diagnóstico sociopolítico en la UCI ni autoevaluación automatizada de los militantes de la UJC (AMADOR, 2015).

Sistema Integral de Gestión de la UJC UCI_Módulo Militante

Sistema que fue diseñado para componer el sistema integral de Gestión de la UJC en la UCI y dar la posibilidad de obtener mejoras en la gestión de la información de los militantes y el universo juvenil (DÍAZ, 2009). Algunos procesos fundamentales que el sistema automatiza se listan a continuación:

- Crear estructuras organizativas.
- Realizar Crecimiento del Universo Juvenil.
- Dar Alta por Crecimiento.
- Crear Expediente del Militante.
- Dar Baja por Ajuste.

El código de este sistema no se encuentra disponible en la actualidad. Del mismo solo se puede encontrar la documentación y la misma no está completa. Su estudio, sin embargo, de este sistema, ha servido de apoyo a la presente investigación para conocer cómo se llevan a cabo internamente los procesos referentes a la UJC dentro de la organización.

El estudio de los sistemas homólogos, ha demostrado que ninguno de ellos satisface las necesidades que hoy presenta la Facultad 1 para el control de los cuadros existentes. Algunos porque utilizan *software* privativo, lo que imposibilita la obtención y modificación del código fuente para ser utilizado por la institución, y otros porque, a pesar de que se asemejan bastante, están desarrollados en lenguajes de programación diferentes o no están validados funcionalmente en otras instituciones. Sin embargo, a pesar de que ninguno de estos sistemas constituye una solución integral al problema en cuestión, su estudio permitió obtener una visión más amplia de la solución y proporcionó funcionalidades de interés para enriquecer la misma. Dichos elementos enfatizaron más la necesidad de desarrollar un sistema específico y exclusivo para el control de cuadros de la Facultad 1, de modo que responda a las necesidades planteadas.

Por lo expresado anteriormente, se ha decidido implementar un sistema de control de cuadros para la Facultad 1, de modo que satisfaga las necesidades de la administración y facilite el trabajo referente al control de la información relacionada con los cuadros dentro de la Facultad.

1.6 Herramientas, lenguajes, metodologías y entornos de desarrollo

1.6.1 Metodologías de desarrollo de *software*

Una metodología de desarrollo de *software* es un conjunto de reglas y pautas que se utilizan en el proceso de investigación, planificación, diseño, desarrollo, prueba, configuración y mantenimiento de un producto de *software*. La metodología también incluye valores centrales que son respaldados por el equipo y las herramientas del proyecto utilizado en la planificación, desarrollo y proceso de implementación (LIVIU, 2014).

Por otra parte (SIVARANJANI, 2017), establece como metodología de desarrollo de *software*: marco necesario con un conjunto de reglas para planificar todas las etapas del desarrollo de *software*.

Las metodologías de desarrollo son las encargadas de guiar a los desarrolladores a la hora de crear un *software*, brindando una serie de pasos definidos y estructurados desde el comienzo hasta el lanzamiento del producto.

Proceso Unificado Ágil (AUP)

Es una versión simplificada del *Rational Unified Process* (RUP). Describe un enfoque simple y fácil de entender para desarrollar *software* de aplicación empresarial utilizando técnicas y conceptos ágiles, pero que sigue siendo fiel al RUP. El enfoque aplica técnicas ágiles que incluyen desarrollo impulsado por pruebas (TDD por sus siglas en inglés), desarrollo dirigido por modelos ágiles (AMDD por sus siglas en inglés), gestión de cambios ágiles y refactorización de bases de datos para mejorar su productividad (AMBLER, 2017).

La figura 4 muestra el ciclo de vida de la AUP. Primero, se puede apreciar el cambio en las disciplinas. La disciplina "Modelo" abarca las disciplinas de Modelado, Requisitos y Análisis y Diseño de Negocios de RUP. Como puede ver, el modelo es una parte importante de la AUP, pero no domina el proceso. En segundo lugar, la disciplina de "Gestión de configuración y cambio" es ahora la disciplina de Gestión de configuración. En el desarrollo ágil, las actividades de gestión del cambio suelen formar parte de los esfuerzos de gestión de requisitos, que forma parte de la disciplina Modelo (AMBLER, 2017).

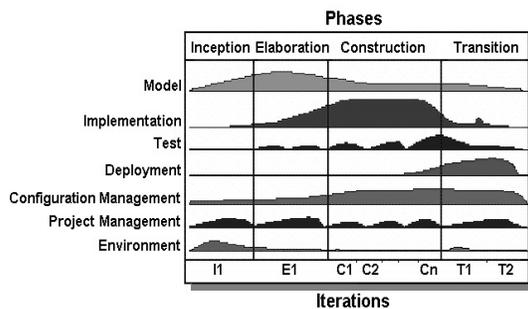


Figura 4: Ciclo de vida del Proceso Unificado Ágil (AUP) (W. Ambler & Lines, 2017)

Fundamentación de la metodología a utilizar

Como resultado, se determinó guiar el proceso de desarrollo de la aplicación web con la metodología AUP-UCI. La misma es una modificación realizada por la UCI del Proceso Unificado Ágil (AUP por sus siglas en inglés) que es a su vez, una versión simplificada del Proceso Unificado de Racional (RUP por sus siglas en inglés) para el proceso de desarrollo de *software*; para de esta forma lograr estándares de

calidad como los exige el Modelo de Madurez de Capacidad – Integración (*Capability Maturity Model Integration* o CMMI por sus siglas en inglés) (SÁNCHEZ, 2014).

La metodología de desarrollo AUP-UCI propone para el ciclo de vida de los proyectos las fases: Inicio, Ejecución y Cierre (Ver anexo 1). De igual manera propone siete disciplinas (Ver anexo 2 para mayor comprensión) pero a un nivel más elemental, las cuales se muestran a continuación (SÁNCHEZ, 2014):

1. Modelado de negocio
2. Requisitos, análisis y diseño
3. Implementación
4. Pruebas internas
5. Pruebas de liberación
6. Pruebas de aceptación
7. Se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían Gestión de la configuración (CM), Planeación de proyecto (PP) y Monitoreo y control de proyecto (PMC).

Esta metodología propone también once roles (Ver anexo 3) para el ciclo de vida de los proyectos en la UCI y además cuatro escenarios para el modelado del negocio y el sistema en los proyectos, en el caso de la presente investigación se empleó el escenario dos, que se describe a continuación (SÁNCHEZ, 2014):

- **Escenario 2:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

1.6.2 Lenguaje de programación del lado del cliente

Cuando se programa una página web, en la mayoría de los casos se utiliza los que se conocen como “lenguajes del lado del cliente”. Es decir, que el servidor no interviene para nada en el proceso de crear la página web solicitada por el usuario (ALEJO, 2016). La programación del lado del cliente permite validar algunos datos en la máquina de éste, antes de que sean enviados al servidor; de esta manera, los usuarios pueden tener un informe de error de forma inmediata (S.L, 2018).

HTML v5

Lenguaje de Marcado para Hipertextos (*HyperText Markup Language*) es el bloque de construcción más básico de una página web que se usa para crear y representarla visualmente. Determina el contenido de la misma, pero no su funcionalidad. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos (DOCS, 2019).

Para el desarrollo de esta investigación se ha empleado HTML5, siendo la última versión de HTML. El término representa dos conceptos diferentes (MDN, 2019):

- ❖ Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- ❖ Contiene un conjunto más amplio de tecnologías que permite a los sitios web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5 (MDN, 2019).

CSS v3

CSS (siglas en inglés de *Cascading Style Sheets*), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado (MDN, 2019). Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML (*eXtensible HyperText Markup Language*); el lenguaje puede ser aplicado a cualquier documento XML (*eXtensible Markup Language*), incluyendo XHTML, SVG (*Scalable Vector Graphics*), XUL (*XML User Interface Language*), RSS (*Really Simple Syndication*), etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas (CLARK, 2016).

Type Script v.3.4.3

TypeScript es un lenguaje de programación de alto nivel que implementa muchos de los mecanismos más habituales de la programación orientada a objetos, pudiendo extraer grandes beneficios que serán especialmente deseables en aplicaciones grandes, capaces de escalar correctamente durante todo su tiempo de mantenimiento.

La característica fundamental de TypeScript es que compila en JavaScript nativo, por lo que se puede usar en todo proyecto donde se esté usando JavaScript. El navegador, o cualquier otra plataforma donde

se ejecuta JavaScript, nunca llegará a enterarse que el código original estaba escrito en TypeScript, porque lo único que llegará a ejecutar es el JavaScript resultante de la compilación (GUERRA, 2016).

Bootstrap v. 4.4.1

Bootstrap es el *framework* que permite crear interfaces web con CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una *Tablet* sin que el usuario tenga que hacer nada; esto se denomina diseño adaptativo o *Responsive Design*. El framework trae varios elementos con estilos predefinidos fáciles de configurar: botones, menús desplegables, formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos. Bootstrap proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los componentes de HTML (OTTO, 2016-2020).

Angular v.8.0

Angular es un *framework* para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo-Vista-Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer la unión de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar) (PRECHT, 2016).

1.6.3 Lenguaje de programación del lado del servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente (CASTILLO, 2014).

Java v.8.0

Java 8 es la versión más reciente de Java que incluye nuevas características, mejoras y correcciones de errores para mejorar la eficacia en el desarrollo y la ejecución de programas Java (JAVA, 2019).

Entre las mejoras que ofrece esta versión se encuentran:

- Métodos de extensión virtual y expresión Lambda: una de las funciones destacables de Java SE 8 es la implantación de expresiones Lambda y funciones adyacentes a la plataforma y el lenguaje de programación Java.
- *Application Program Interface* (API) de fecha y hora: esta nueva API permitirá a los administradores gestionar datos de fecha y hora de forma mucho más natural y fácil de comprender.
- Seguridad mejorada: sustitución de la lista de métodos sensibles al emisor mantenida a mano existente por un mecanismo que identifica con mayor precisión dichos métodos y permite detectar a los emisores de forma fiable.

Spring Boot v. 2.2.4

Spring Boot facilita la creación de aplicaciones independientes y de grado de producción basada en Spring que puedan ser corridas. Se puede usar Spring Boot para crear aplicaciones Java que se pueden iniciar usando `java -jar`. También proporciona una herramienta de línea de comandos que ejecuta "Spring scripts".

Los objetivos principales son:

- Proporciona una experiencia de inicio radicalmente más rápida y ampliamente accesible para todo el desarrollo de Spring.
- Proporciona una gama de características no funcionales que son comunes a grandes clases de proyectos (como servidores integrados, seguridad, métricas, comprobaciones de estado y configuración externa).
- Absolutamente ninguna generación de código y ningún requisito para la configuración XML (PHILLIP WEBB, 2019).

1.6.4 Entorno de desarrollo

Los entornos de desarrollo o bien conocido como IDE (por sus siglas en inglés *Integrated Development Environment*), son las herramientas con las cuales los programadores crean aplicaciones. Es cierto que pueden programarse con un editor y un compilador (a veces, con un depurador), pero, en entornos profesionales, casi siempre se utiliza un IDE (PÉREZ, JUAN CARLOS MORENO, 2018).

IntelliJ IDEA V. 2018.2.4

Para el lenguaje de programación por el lado del servidor se emplea la herramienta IntelliJ IDEA. Es un entorno de desarrollo integrado (IDE) escrito en Java para desarrollar *software* de computadora. Está desarrollado por *JetBrains* (anteriormente conocido como IntelliJ), y está disponible como una edición comunitaria con licencia Apache 2, y en una edición comercial patentada (JETBRAINS.ORG, 2020).

Cada aspecto de IntelliJ IDEA ha sido diseñado para maximizar la productividad del desarrollador. En conjunto, la asistencia de codificación inteligente y el diseño ergonómico hacen que el desarrollo no solo sea productivo sino también agradable. La asistencia de codificación en IntelliJ IDEA no se trata solo del editor, también le ayuda a mantenerse productivo cuando se ocupa de otros aspectos, por ejemplo: rellenando un campo, buscando en una lista de elementos; acceder a una ventana de herramientas; o alternar una configuración, etc. (JETBRAINS, 2019).

Visual Studio Code v.1.41.1

Para el lenguaje de programación por el lado del cliente se emplea Visual Studio Code v.1.41.1. Es un editor de código fuente desarrollado por *Microsoft* para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. Asimismo, combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo. Visual Studio Code es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos podemos destacar los principales del desarrollo Web: HTML, CSS, y JavaScript, entre otros (LUCA, 2019).

1.6.5 Administración y Sistema Gestor de Base de Datos (SGBD)

Teniendo en cuenta los aspectos técnicos estudiados y después de realizar un estudio sobre los SGBD, el autor de esta investigación concuerda con (MELGAREJO, 2014), en que un Sistema Gestor de Base de Datos o *Date Base Management System (DBMS)*, es una colección de datos relacionados entre sí, estructurados y organizados y que se define como un conjunto de programas que acceden a dichos datos.

PostgreSQL 9.4

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*) y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo

en uno de los procesos no afectará el resto y el sistema continuará funcionando. Entre sus características se encuentran (POSTGRESQL, 2019):

- ❖ Copias de seguridad en caliente (*Online/hot backups*).
- ❖ Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- ❖ Licencia BSD

Para el desarrollo de la propuesta de solución, se ha empleado como SGBD, PostgreSQL en su versión 9.4.

PgAdmin v.4.13

Es la herramienta que permite la gestión y administración de base de datos SQL de la plataforma de código abierto PostgreSQL. La aplicación se puede utilizar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Está diseñada para responder a las necesidades de todos los usuarios, desde escribir consultas SQL sencillas hasta el desarrollo de bases de datos complejas. La conexión del servidor puede hacerse a través de TCP/IP (*Transfer Control Protocol / Internet Protocol*) o *sockets* de dominio Unix (en plataformas * nix), y puede ser encriptado SSL para la seguridad. Es un *software* libre publicado bajo la Licencia PostgreSQL (PGADMIN, 2019).

1.6.6 Lenguaje de Modelado

Un lenguaje de modelado es un lenguaje artificial para expresar modelos, los cuales se muestran en forma de diagramas por comodidad. Los lenguajes de modelado, al igual que los lenguajes naturales, poseen un conjunto de palabras que existen en el lenguaje (léxico); y un conjunto de reglas que nos dicen cómo se pueden combinar dichas palabras para componer "frases" que tengan sentido (sintaxis) (PÉREZ, CESAR GONZÁLEZ, 2011).

UML

El Lenguaje Unificado de Modelado o *Unified Model Language* (UML) es un lenguaje de modelado visual usado para especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Captura decisiones y conocimiento de los sistemas que se deben construir. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios (RUMBAUGH, 2000). Es comúnmente usado en el desarrollo de *software* y diseño orientado a objetos (WEB20.CO, 2015).

1.6.7 Herramienta CASE

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de *software* asistida por computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el coste de las mismas en términos de tiempo y dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo de *software*, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, entre otros (BELTRÁN, 2016).

Visual Paradigm for UML Enterprise Edition 8

Es una potente herramienta CASE de modelado visual que ayuda a construir y administrar los diagramas y elementos de modelo, además de asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de *software* (INGENIERÍA-DE-SOFTWARE, Curso 2013-2014).

Algunas de las ventajas que proporciona Visual Paradigm for UML son:

- Dibujo: facilita el modelado de UML, ya que proporciona herramientas específicas para ello.
- Corrección sintáctica: controla que el modelado con UML sea correcto.
- Coherencia entre diagramas: al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- ❖ Integración con otras aplicaciones: permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.

1.6.8 Herramienta para el prototipado

Axure RP v. 9.0

Es una herramienta muy completa, orientada a diseñar estructuras y prototipos básicos o avanzados de forma fácil, dirigido a aplicaciones web y de escritorio. Brinda la posibilidad de crear documentos maestros, los cuales son muy útiles pues son reutilizables en todas las interfaces que se diseñen, y los cambios que se apliquen a estos se ven reflejados en todas las páginas en las que se estén utilizando (JULIE, 2019).

1.7 Conclusiones del capítulo

El estudio de los conceptos asociados a la propuesta de solución, posibilitó una mejor comprensión de esta. El análisis de los sistemas informáticos existentes que gestionan recursos humanos y la información concerniente a ellos, permitió conocer que no existe en Cuba sistemas que cumplan con todas las necesidades requeridas para el control de cuadros de la Facultad 1, constituyendo esto la base para el desarrollo del sistema propuesto. El estudio realizado de las metodología de desarrollo de *software*, así como las herramientas y tecnologías facilitó la selección de un entorno de desarrollo adecuado para desarrollar la solución propuesta.

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA DE CONTROL DE CUADROS DE LA FACULTAD 1

Para el desarrollo de una herramienta informática es muy importante conocer cómo funcionan cada uno de los procesos que van a ser automatizados, así como mantener una comunicación importante con el cliente en aras de lograr el producto que se desea y el objetivo por el cual se decidió informatizar dichos procesos. Con vistas a lograr esto, se debe realizar una correcta modelación del negocio para comprender la estructura y el funcionamiento de la organización en la cual se va a hacer uso de la herramienta, además se debe realizar un estudio de los problemas actuales e identificar las posibles mejoras que puede traer consigo el producto desarrollado.

En este capítulo se evidencian las principales características de la propuesta de solución para el control de cuadros de la Facultad 1. Se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que el sistema web funcione correctamente. Además, se construyen los artefactos correspondientes al análisis y diseño acorde con la metodología seleccionada en el capítulo anterior.

2.1 Descripción de la propuesta de solución

El control de los cuadros de la Facultad 1, se corresponde con un proceso sustantivo que lleva a cabo la dirección de dicha área en su funcionamiento interno en cuanto al control y gestión de la información de los cuadros.

Se proponen 4 módulos: Administración, Cuadros, Reservas y Nomencladores. En la presente investigación se implementarán cada uno de ellos.

El módulo “Administración” se encarga de gestionar todos los datos personales relacionados con los cuadros y reservas de la Facultad 1, los cuales se pueden agrupar de la siguiente forma: datos personales, laborales, referentes a las organizaciones u organismos a los que pertenece, trayectoria laboral, estudios realizados, etc. También permitirá gestionar la evaluación correspondiente a los cuadros y reservas, así como los movimientos de los cuadros. Brindará la opción de llevar el control de asistencia de los cuadros a las distintas actividades a las que participan, entre otros aspectos.

El módulo “Cuadros” permite gestionar la información referente al expediente de los cuadros, el cual está conformado por una serie de documentos distribuidos en evaluaciones, planes de trabajo, entre otros, de modo que sirva para tener un mayor control del trabajo de estos. De igual forma, permitirá exportar dichos documentos en formato PDF, así como su impresión para la culminación del proceso.

El módulo “Reservas” ofrece la posibilidad de gestionar la información relacionada al expediente de las reservas, conformado por una serie de documentos que tributan al trabajo cotidiano, así como la superación y formación de la reserva como futuro cuadro, entre otros aspectos. De igual forma, permitirá exportar dichos documentos en formato PDF, así como su impresión para la culminación del proceso. Ello permitirá tener un mayor control sobre las reservas.

El módulo Nomencladores tiene la función de mantener el sistema lo más extensible posible, de manera tal que se pueda aumentar el contenido por defecto del sistema actual sin que haya necesidad de adquirir un sistema nuevo. Este módulo ofrece la posibilidad de gestionar todos los nomencladores.

2.2 Modelo conceptual

2.2.1 Modelo de dominio

Un modelo de dominio es una representación de conceptos del dominio de un problema determinado. Permite representar elementos del mundo real y no componentes del *software*. Provee conocimiento de la nomenclatura del dominio al comunicarle a las partes interesadas los términos importantes y cómo se relacionan entre sí, siendo un artefacto fundamental para realizar el análisis orientado a objetos (SYNERGIX., 2008).

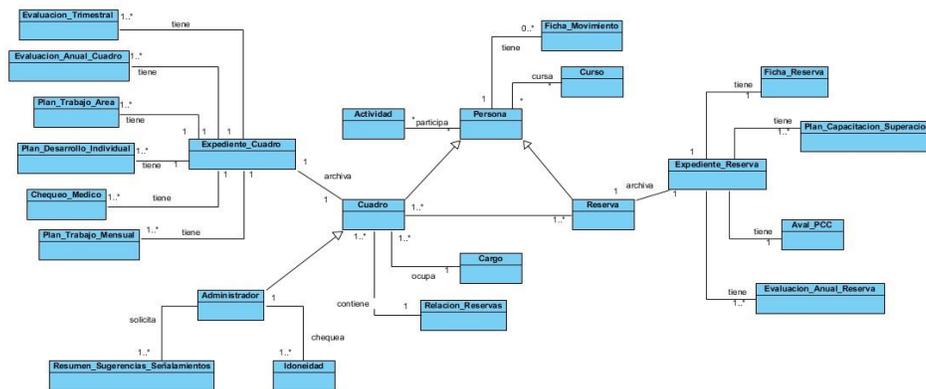


Figura No 5: Modelo de dominio del negocio

2.2.2 Conceptos del dominio

- **Persona:** conjunto o grupo general de personas que tendrá acceso al sistema. Este puede ser de tipo “cuadro” o “administrador”.
- **Cuadro:** condición que adquiere una persona cuando pasa a ocupar un cargo administrativo dentro de la Institución.
- **Cargo:** responsabilidad específica que ocupa una persona y que lo convierte en un cuadro.
- **Reserva:** persona que tiene su propia responsabilidad dentro del sistema de trabajo pero que puede ser o no un cuadro. Es una propuesta a ocupar un cargo igual o superior de otra persona que tenga la condición de cuadro.
- **Cargo actual:** es la responsabilidad que ocupa una reserva ya sea como condición de cuadro o no.
- **Relación de reservas:** documento que poseen los cuadros en el cual está plasmado de manera oficial todas las personas que son reservas de dicho cuadro.
- **Actividad:** proceso o evento al cual están requeridos los cuadros a participar dentro de su desempeño laboral.
- **Curso:** modalidad de aprendizaje en un tema específico, donde los cuadros y reservas deben participar dentro del proceso de superación.
- **Ficha de movimiento:** documento que recoge el historial de movimientos que ha tenido un cuadro determinado. Los movimientos pueden ser por liberación, promoción y democión.
- **Expediente_Cuadro:** conjunto de documentos que recogen la información personal del cuadro, así como los que definen el sustento laboral del mismo para el alcance de sus resultados a través de su control y evaluación.
- **Expediente_Reserva:** conjunto de documentos que recogen la información personal de las reserva, así como los que definen el sustento laboral para la formación, superación y cumplimiento de los objetivos para ocupar responsabilidades dentro de la Institución.
- **Idoneidad:** aptitud, buena disposición o capacidad de una persona para desempeñar determinados cargos o funciones dentro de una organización.
- **Resumen_sugerencias_señalamientos:** documento que reúne todos los elementos expuestos en cuanto a sugerencias y señalamientos dentro del proceso de evaluación de cada cuadro.

- **Plan_Trabajo_Mes:** documento que recoge todas las actividades, tareas y obligaciones que tiene un cuadro para su labor durante el mes.
- **Plan_Desarrollo_Individual:** documento que recoge toda la planificación de tareas y metas que debe cumplir cada cuadro de manera individual.
- **Plan_Trabajo_Área:** documento que recoge todas las actividades, tareas y obligaciones relacionadas con el cargo que ocupa cada cuadro.
- **Evaluación_Anual_Cuadro:** documento que recoge todas las actividades en las que participó el cuadro, todas las tareas que le fueron asignadas y cumplidas satisfactoriamente, todos los resultados alcanzados a partir del cumplimiento de los objetivos en cada etapa del año. Finalmente se emite una evaluación la cual está comprendida entre la categoría de Bien, Muy Bien y Excelente.
- **Evaluación_Desempeño_Trimestral:** documento que recoge todas las actividades en las que participó el cuadro, todas las tareas que le fueron asignadas y cumplidas satisfactoriamente, todos los resultados alcanzados a partir del cumplimiento de los objetivos de los tres meses anteriores a la fecha en la que se emite dicha evaluación.
- **Chequeo_Médico:** documento o plantilla que reúne todos los indicadores médicos que deben ser analizados como estudio médico dentro del proceso laboral de cada Cuadro.
- **Evaluación_Anual_Reserva:** documento que recoge todas las actividades en las que participó una reserva, todas las tareas que le fueron asignadas y cumplidas satisfactoriamente, todos los resultados alcanzados a partir del cumplimiento de los objetivos en cada etapa del año. Finalmente se emite una evaluación la cual está comprendida entre la categoría de Bien, Muy Bien y Excelente.
- **Aval_PCC:** documento emitido por el/la secretaria del PCC (Partido Comunista de Cuba) de la Institución a la que pertenece la reserva. Dicho documento contiene elementos que demuestran el desempeño de dicha reserva en la participación y cumplimiento de las tareas, actividades y responsabilidades dentro de la organización.
- **Plan_Capacitación_Superación:** documento que recoge toda la planificación para una reserva en cuanto a su participación en actividades, cursos y todo lo que tribute a su capacitación y superación personal, laboral y profesional.
- **Ficha_Reserva:** documento que recoge toda la información personal de una reserva.

2.3 Requisitos del sistema

En el proceso de ingeniería de *software*, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto que se desea lograr. Estos representan qué debe hacer el sistema, pero no cómo hacerlo y, además, incorporan las cualidades o propiedades que el mismo debe poseer (SOMMERVILLE, 2007).

2.3.1 Requisitos Funcionales (RF)

Los requisitos funcionales definen el comportamiento interno de un *software*, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de *software* (SOMMERVILLE, 2007).

Tabla 1: Requisitos Funcionales del Módulo "Administración". Elaboración propia

No.	Nombre	Descripción
Módulo "Administración"		
RF_1	Insertar cuadro	El sistema permitirá insertar los datos de un cuadro.
RF_2	Modificar cuadro	El sistema permitirá modificar los datos de un cuadro.
RF_3	Eliminar cuadro	El sistema permitirá eliminar los datos de un cuadro.
RF_4	Listar cuadro	El sistema permitirá listar los datos de un cuadro.
RF_5	Buscar cuadro	El sistema permitirá buscar los datos de un cuadro de acuerdo a los criterios de búsqueda que sean seleccionados.
RF_6	Insertar reserva	El sistema permitirá insertar los datos de una reserva.
RF_7	Modificar reserva	El sistema permitirá modificar los datos de una reserva.
RF_8	Eliminar reserva	El sistema permitirá eliminar los datos de una reserva.
RF_9	Listar reserva	El sistema permitirá listar los datos de una reserva.
RF_10	Buscar reserva	El sistema permitirá buscar los datos de una reserva de acuerdo a los criterios de búsqueda que sean seleccionados.
RF_11	Insertar ficha de cuadro	El sistema permitirá insertar los datos de una ficha de cuadro.
RF_12	Modificar ficha de cuadro	El sistema permitirá modificar los datos de una ficha de cuadro.
RF_13	Eliminar ficha de cuadro	El sistema permitirá eliminar los datos de una ficha de cuadro.
RF_14	Mostrar ficha de cuadro	El sistema permitirá mostrar los datos de una ficha de cuadro.
RF_15	Generar ficha de cuadro	El sistema permitirá generar un documento con los datos de una ficha de cuadro.

RF_16	Imprimir ficha de cuadro	El sistema permitirá imprimir un documento generado con los datos de una ficha de cuadro.
RF_17	Insertar evaluación	El sistema permitirá insertar los datos de una evaluación.
RF_18	Modificar evaluación	El sistema permitirá modificar los datos de una evaluación.
RF_19	Eliminar evaluación	El sistema permitirá eliminar los datos de una evaluación.
RF_20	Mostrar evaluación	El sistema permitirá mostrar los datos de una evaluación.
RF_21	Generar evaluación	El sistema permitirá generar un documentos con los datos de una evaluación.
RF_22	Imprimir evaluación	El sistema permitirá imprimir un documento generado con los datos de una evaluación.
RF_23	Insertar movimiento de cuadro	El sistema permitirá insertar los datos de un movimiento de cuadro.
RF_24	Modificar movimiento de cuadro	El sistema permitirá modificar los datos de un movimiento de cuadro.
RF_25	Eliminar movimiento de cuadro	El sistema permitirá eliminar los datos de un movimiento de cuadro.
RF_26	Listar movimiento de cuadro	El sistema permitirá listar los datos de un movimiento de cuadro.
RF_27	Insertar asistencia a actividad	El sistema permitirá insertar los datos de una asistencia a actividad.
RF_28	Modificar asistencia a actividad	El sistema permitirá modificar los datos de una asistencia a actividad.
RF_29	Eliminar asistencia a actividad	El sistema permitirá eliminar los datos de una asistencia a actividad.
RF_30	Mostrar asistencia a actividad	El sistema permitirá mostrar los datos de una asistencia a actividad.
RF_31	Obtener idoneidad de cuadro	El sistema permitirá obtener la idoneidad de un cuadro.
RF_32	Obtener resumen de sugerencias y recomendaciones	El sistema permitirá obtener un resumen de sugerencias y recomendaciones.
RF_33	Obtener porciento de estabilidad	El sistema permitirá obtener el porciento de estabilidad de los cuadros.
RF_34	Obtener porciento de raza	El sistema permitirá obtener el porciento de raza de los cuadros.
RF_35	Listar cursos pasados	El sistema permitirá listar los cursos pasados por los cuadros.

Tabla 2: Requisitos Funcionales del Módulo "Cuadros". Elaboración propia

No.	Nombre	Descripción
Módulo "Cuadros"		

RF_36	Insertar evaluación trimestral	El sistema permitirá insertar los datos de una evaluación trimestral.
RF_37	Modificar evaluación trimestral	El sistema permitirá modificar los datos de una evaluación trimestral.
RF_38	Eliminar evaluación trimestral	El sistema permitirá eliminar los datos de una evaluación trimestral.
RF_39	Mostrar evaluación trimestral	El sistema permitirá mostrar los datos de una evaluación trimestral.
RF_40	Buscar evaluación trimestral	El sistema permitirá buscar los datos de una evaluación trimestral.
RF_41	Generar evaluación trimestral	El sistema permitirá generar un documento con los datos de una evaluación trimestral.
RF_42	Imprimir evaluación trimestral	El sistema permitirá imprimir un documento generado con los datos de una evaluación trimestral.
RF_43	Insertar evaluación anual de cuadro	El sistema permitirá insertar los datos de una evaluación anual de cuadro.
RF_44	Modificar evaluación anual de cuadro	El sistema permitirá modificar los datos de una evaluación anual de cuadro.
RF_45	Eliminar evaluación anual de cuadro	El sistema permitirá eliminar los datos de una evaluación anual de cuadro.
RF_46	Mostrar evaluación anual de cuadro	El sistema permitirá mostrar los datos de una evaluación anual de cuadro.
RF_47	Buscar evaluación anual de cuadro	El sistema permitirá buscar los datos de una evaluación anual de cuadro.
RF_48	Generar evaluación anual de cuadro	El sistema permitirá generar un documento con los datos de una evaluación anual de cuadro.
RF_49	Imprimir evaluación anual de cuadro	El sistema permitirá imprimir un documento con los datos de una evaluación anual de cuadro.
RF_50	Anexar plan de trabajo del área	El sistema permitirá anexar un plan de trabajo del área.
RF_51	Eliminar plan de trabajo del área	El sistema permitirá eliminar un plan de trabajo del área.
RF_52	Mostrar plan de trabajo del área	El sistema permitirá mostrar un plan de trabajo del área.
RF_53	Generar plan de trabajo del área	El sistema permitirá generar un plan de trabajo del área.
RF_54	Imprimir plan de trabajo del área	El sistema permitirá imprimir un plan de trabajo del área.
RF_55	Anexar plan de trabajo mensual	El sistema permitirá anexar un plan de trabajo mensual.
RF_56	Eliminar plan de trabajo mensual	El sistema permitirá eliminar un plan de trabajo mensual.
RF_57	Mostrar plan de trabajo mensual	El sistema permitirá mostrar un plan de trabajo mensual.
RF_58	Generar plan de trabajo mensual	El sistema permitirá generar un plan de trabajo mensual.
RF_59	Imprimir plan de trabajo mensual	El sistema permitirá imprimir un plan de trabajo mensual.
RF_60	Anexar plan de desarrollo	El sistema permitirá anexar un plan de trabajo de desarrollo

	individual	individual.
RF_61	Eliminar plan de desarrollo individual	El sistema permitirá eliminar un plan de trabajo de desarrollo individual.
RF_62	Mostrar plan de desarrollo individual	El sistema permitirá mostrar un plan de trabajo de desarrollo individual.
RF_63	Generar plan de desarrollo individual	El sistema permitirá generar un plan de trabajo de desarrollo individual.
RF_64	Imprimir plan de desarrollo individual	El sistema permitirá imprimir un plan de trabajo de desarrollo individual.
RF_65	Insertar relación de reservas	El sistema permitirá insertar los datos de la relación de reservas.
RF_66	Modificar relación de reservas	El sistema permitirá modificar los datos de la relación de reservas.
RF_67	Eliminar relación de reservas	El sistema permitirá eliminar los datos de la relación de reservas.
RF_68	Mostrar relación de reservas	El sistema permitirá mostrar los datos de la relación de reservas.
RF_69	Obtener plantilla de chequeo médico	El sistema permitirá obtener una plantilla con los datos de la relación de reservas.

Tabla 3: Requisitos Funcionales del Módulo "Reservas". Elaboración propia

No.	Nombre	Descripción
Módulo "Reservas"		
RF_70	Insertar ficha de reserva	El sistema permitirá insertar los datos de una ficha de reserva.
RF_71	Modificar ficha de reserva	El sistema permitirá modificar los datos de una ficha de reserva.
RF_72	Eliminar ficha de reserva	El sistema permitirá eliminar los datos de una ficha de reserva.
RF_73	Mostrar ficha de reserva	El sistema permitirá mostrar los datos de una ficha de reserva.
RF_74	Generar ficha de reserva	El sistema permitirá generar el documento de una ficha de reserva.
RF_75	Imprimir ficha de reserva	El sistema permitirá imprimir el documento de una ficha de reserva.
RF_76	Anexar plan de capacitación y superación	El sistema permitirá anexar el plan de capacitación y superación.
RF_77	Eliminar plan de capacitación y superación	El sistema permitirá eliminar el plan de capacitación y superación.
RF_78	Mostrar plan de capacitación y superación	El sistema permitirá mostrar el plan de capacitación

		y superación.
RF_79	Generar plan de capacitación y superación	El sistema permitirá generar el documento con el plan de capacitación y superación.
RF_80	Imprimir plan de capacitación y superación	El sistema permitirá imprimir el documento con el plan de capacitación y superación.
RF_81	Anexar aval del PCC	El sistema permitirá anexar el aval del PCC.
RF_82	Eliminar aval del PCC	El sistema permitirá eliminar el aval del PCC.
RF_83	Mostrar aval del PCC	El sistema permitirá mostrar el aval del PCC.
RF_84	Insertar evaluación anual de reserva	El sistema permitirá insertar los datos de la evaluación anual de reserva.
RF_85	Modificar evaluación anual de reserva	El sistema permitirá modificar los datos de la evaluación anual de reserva.
RF_86	Eliminar evaluación anual de reserva	El sistema permitirá eliminar los datos de la evaluación anual de reserva.
RF_87	Mostrar evaluación anual de reserva	El sistema permitirá mostrar los datos de la evaluación anual de reserva.
RF_88	Buscar evaluación anual de reserva	El sistema permitirá buscar los datos de la evaluación anual de reserva.
RF_89	Generar evaluación anual de reserva	El sistema permitirá generar el documentos con los datos de la evaluación anual de reserva.
RF_90	Imprimir evaluación anual de reserva	El sistema permitirá imprimir el documento con los datos de la evaluación anual de reserva.

Tabla 4: Requisitos Funcionales del Módulo "Nomencladores". Elaboración propia

No.	Nombre	Descripción
Módulo "Nomencladores"		
RF_91	Insertar actividad	El sistema permitirá insertar los datos de una actividad.
RF_92	Modificar actividad	El sistema permitirá modificar los datos de una actividad.
RF_93	Eliminar actividad	El sistema permitirá eliminar los datos de una actividad.
RF_94	Listar actividad	El sistema permitirá listar los datos de una actividad.
RF_95	Buscar actividad	El sistema permitirá buscar los datos de una actividad.
RF_96	Insertar curso	El sistema permitirá insertar los datos de un curso.
RF_97	Modificar curso	El sistema permitirá modificar los datos de un curso.
RF_98	Eliminar curso	El sistema permitirá eliminar los datos de un curso.
RF_99	Listar curso	El sistema permitirá listar los datos de un curso.

RF_100	Buscar curso	El sistema permitirá buscar los datos de un curso.
--------	--------------	--

Tabla 5: Requisitos Funcionales "Autenticar usuario". Elaboración propia

No.	Nombre	Descripción
RF_101	Autenticar usuario	El sistema permitirá insertar los datos de un usuario para ser autenticado al sistema.

2.3.2 Requisitos No Funcionales (RNF)

De acuerdo con (SOMMERVILLE, 2007), los requisitos no funcionales son aquellos requisitos que no describen información a guardar, ni funciones a realizar, sino que son propiedades que hacen al producto atractivo, usable, rápido o confiable. Además, se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el *software*.

Requisitos de interfaz o apariencia

RNF_1: Las ventanas del sistema contendrán los datos bien claros y bien estructurados, permitirán la interpretación correcta de la información.

RNF_2: La entrada de datos de manera incorrecta será detectada claramente e informada al usuario.

RNF_3: Los íconos utilizados deberán corresponderse con la función que realizan.

RNF_4: Todos los textos y mensajes en pantalla aparecerán en idioma español.

RNF_5: Su funcionamiento será intuitivo y requerirá información mínima.

Usabilidad

RNF_6: Las funcionalidades y diseño del sistema deben ser intuitivas de forma tal que los usuarios puedan interactuar con el sistema sin necesidad de poseer conocimientos básicos de informática.

RNF_7: Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista el acceso a la misma.

Rendimiento

RNF_8: El tiempo de respuesta del sistema debe ser rápido, no mayor de los 5 segundos para cualquier operación realizada por el usuario.

Soporte

RNF_9: Se debe brindar a los usuarios la facilidad de instalación.

Portabilidad

RNF_10: El sistema de permitir ser ejecutado sobre los Sistemas Operativos Windows y Linux.

Eficiencia

RNF_12: El sistema soportará la conexión simultánea de todos los posibles usuarios habituales, un promedio de 25 y un máximo de 600.

RNF_13: El sistema deberá tener por cada transacción un tiempo de respuesta promedio de 1,5 segundos y un máximo de 3 segundos.

Seguridad

RNF_14: Se mantendrá seguridad y control a nivel de usuarios y contraseñas, garantizando el acceso de los mismos solo a los niveles establecidos de acuerdo a la función que realizan.

RNF_15: Ofrecer una advertencia antes de realizar acciones irreversibles (Ej: eliminar cualquier documento o información).

Confidencialidad

RNF_16: La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.

Integridad

RNF_17: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.

Disponibilidad

RNF_18: La disponibilidad del sistema debe estar bajo la modalidad de 24 horas al día y los 7 días de las semana.

Restricciones en el diseño y la implementación

RNF_19: Como lenguaje de implementación: Java, Type Script, Bootstrap

RNF_20: Como metodología de desarrollo: AUP-UCI

RNF_21: Como lenguaje de modelado: UML

RNF_22: Como herramienta CASE: Visual Paradigm

RNF_23: Como marco de trabajo: Spring Boot, Angular

Software

RNF_24: Para el funcionamiento del sistema se requiere disponer de un servidor que cuente con un sistema operativo Linux o Windows 2000 o superior, PostgreSQL 9.4 o versiones superiores. Los usuarios del sistema deberán contar con un navegador Internet Explorer 8.0 o Moxilla Firefox 33.0 o superior, para poder acceder a las opciones que brinda el sistema.

Hardware

Los requerimientos mínimos que necesita el sistema pueden ser los siguientes:

RNF_25: Microprocesador Intel Pentium IV a 3.0 GHz o superior.

RNF_26: 512 MB de RAM (*Random Access Memory*, por sus siglas en inglés) o superior.

RNF_27: 200 MB o más de disco duro.

2.4 Modelo de Casos de Uso del Sistema

El modelo de casos de uso permite que los desarrolladores del *software* y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas. Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones y además describe lo que hace el sistema para cada tipo de usuario (VARGAS, 2010).

2.4.1 Diagrama de casos de uso del sistema

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un caso de uso es una unidad simple de trabajo significativo. Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso o "extender" a otro caso de uso con su propio comportamiento (ARCHITECT, 2016).

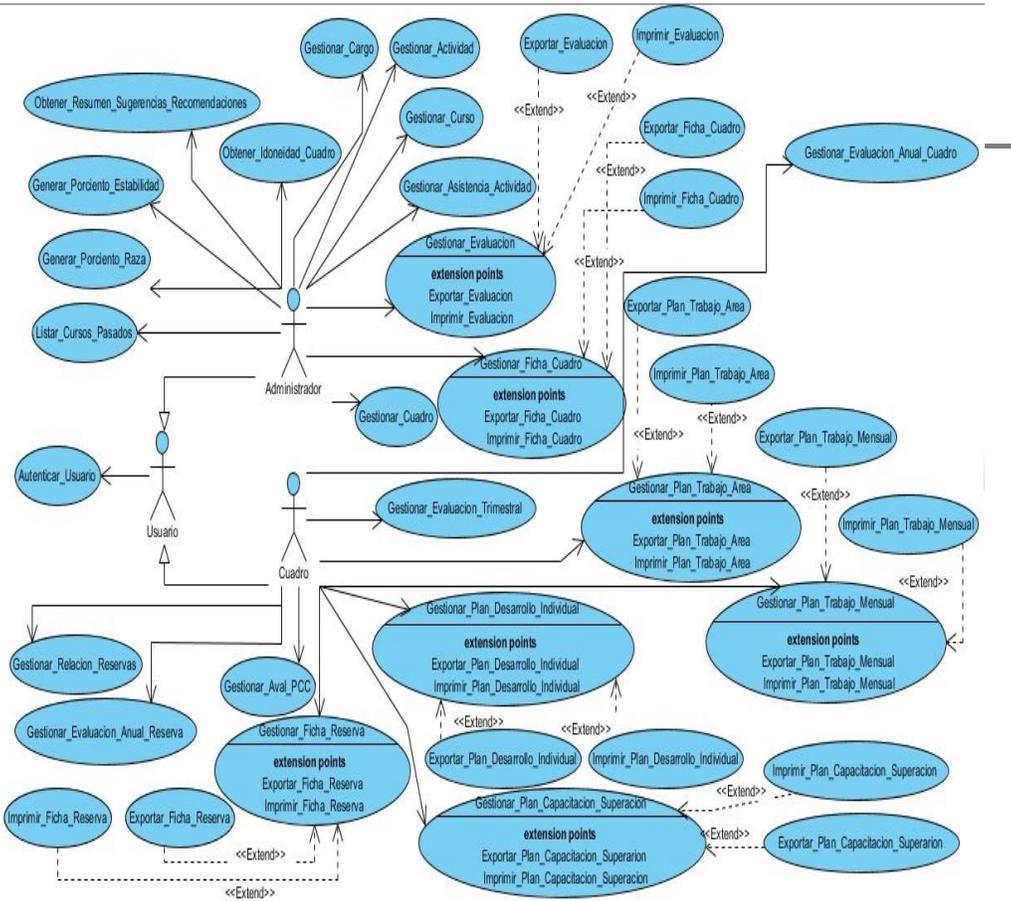


Figura 7: Diagrama de casos de uso del sistema

2.4.2 Patrones de casos de uso del sistema

Son patrones que capturan mejores prácticas para modelar Casos de Uso (CU), permiten representar los comportamientos que deben existir en el sistema. Estos patrones ayudan a un mejor entendimiento de la funcionalidad de un sistema y la interacción del usuario con el mismo.

CRUD: Es el acrónimo de Crear, Obtener, Modificar y Eliminar (del original en inglés: *Create, Read, Update and Delete*). Este patrón es usado en los casos donde se realizan tareas de gestión, ejemplo de ellos los casos gestionar cuadro, gestionar reservas, gestionar Evaluación trimestral y otros. En los casos

donde se realizan parte de estas tareas se denomina CRUD parcial. Este patrón se utilizó en casos como: gestionar Plan de trabajo del área, gestionar Plan de capacitación y superación, entre otros.

Múltiples Actores: Este patrón se refiere al ingreso de dos o más actores del sistema y todos tienen un rol común. Este patrón se utilizó en algunos casos de uso como: gestionar Plan de trabajo del área, gestionar Ficha de reserva y autenticar usuario.

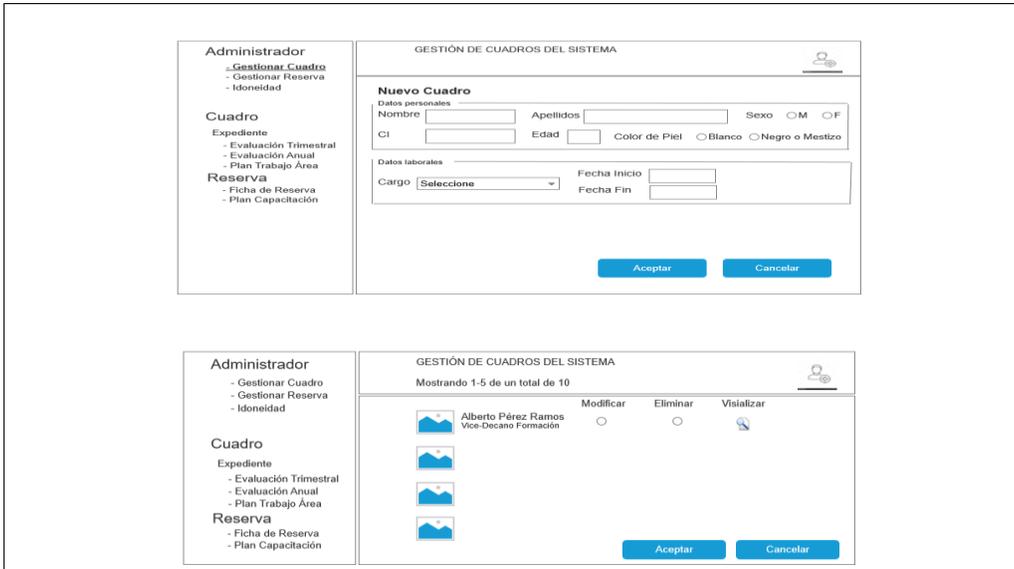
2.4.3 Descripción textual de los casos de uso del sistema

A continuación se muestra la descripción textual del caso de uso Gestionar cuadro. Las demás descripciones y para mayor información, pueden ser consultadas en el Anexo No 4.

Tabla 7: Descripción textual del caso de uso Gestionar cuadro (Elaboración propia)

Caso de uso	Gestionar cuadro
Actores	Usuario
Resumen	El caso de uso inicia cuando el actor accede a la ventana inicial que el sistema muestra según su rol y accede dentro del módulo Cuadros a la opción Gestionar cuadro. Muestra un listado con los cuadros ya registrados y la opción de crear un cuadro nuevo. Además por cada uno permite: modificar y eliminar cuadro. El actor crea, modifica o elimina un cuadro finalizando así el caso de uso.
Precondiciones	El usuario de estar autenticado con el rol de Administrador.
Postcondición	Se creó, se modificó o se eliminó un cuadro.
Referencias	RF_1, RF_2, RF_3
Flujo de eventos	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. El caso de uso comienza cuando el actor accede al sistema y selecciona la opción Gestionar cuadro en el menú del módulo Cuadros.	
	2. El sistema muestra los datos de cada uno de los cuadros almacenados en el sistema y permite: <ul style="list-style-type: none"> • Crear nuevo cuadro • Modificar cuadro. Ver sección 1:

	<p>“Modificar cuadro”</p> <ul style="list-style-type: none"> • Eliminar cuadro. Ver sección 2: “Eliminar cuadro”
3. Selecciona la opción “Crear cuadro”	
	<p>4. El sistema muestra un formulario brindando la posibilidad de insertar los siguientes datos.</p> <ul style="list-style-type: none"> • Nombre del cuadro • Apellidos del cuadro • CI • Edad • Sexo • Cargo • Color de piel • Fecha de Inicio • Fecha de Fin <p>y las opciones:</p> <ul style="list-style-type: none"> • Aceptar • Cancelar
5. Introduce los datos en los campos requeridos y selecciona la opción “Aceptar”.	
	6. Valida los datos introducidos.
	7. Registra un nuevo cuadro en el sistema.
	8. Muestra un mensaje: “El cuadro se ha creado exitosamente”.
	9. El caso de uso termina.
Flujos alternos	
*a. Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema regresa a la vista anterior
	4. a. 2 El caso de uso termina.
4.a Existen campos vacíos	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema muestra un mensaje: “Por favor, debe llenar los campos vacíos”.
	4. a.2 El sistema regresa al paso 4.
Prototipo de interfaz	



Sección 1: Modificar cuadro

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción "Modificar cuadro"	
	2. El sistema muestra una ventana auxiliar con los datos a modificar del cuadro, y las opciones: <ul style="list-style-type: none"> • Aceptar • Cancelar
3. Actualiza los campos requeridos en el formulario y selecciona la opción "Aceptar".	
	4. Valida los datos introducidos.
	5. Actualiza los datos del cuadro.
	6. El caso de uso termina.

Flujo alterno

Sección 1: "Modificar cuadro".

*.a Selecciona la opción "Cancelar".

Acciones del actor	Respuesta del sistema
	4. a.1 El sistema regresa a la vista anterior.
	4. a.2 El caso de uso termina.
4.a Existen campos vacíos	
Acciones del actor	Respuesta del sistema
	4. a.1 El sistema muestra un mensaje "Verifique que no existan campos vacíos".
	4. a.2 Muestra el indicador de campos que no pueden estar vacíos.

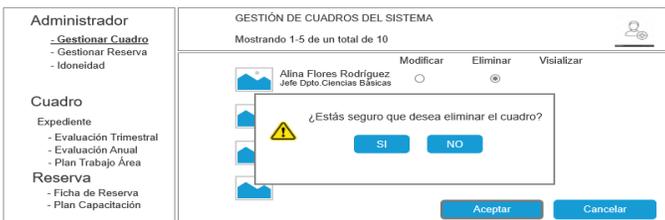
Prototipo de interfaz



Sección 2: Eliminar cuadro

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción "Eliminar cuadro y luego el boton "Aceptar".	
	2. El sistema muestra una ventana emergente con el mensaje: "¿Estás seguro que desea eliminar el cuadro?"

	y las opciones: <ul style="list-style-type: none"> • Si • No
3. Selecciona la opción "Si".	
	4. Elimina el cuadro del sistema
	5. El sistema muestra un mensaje: "El cuadro fue eliminado con éxito".
	6. Termina el caso de uso.
Flujo alterno	
Sección 2: "Eliminar cuadro".	
*.a Selecciona la opción "No".	
Acciones del actor	Respuesta del sistema
	*.a.1 Cierra el mensaje de confirmación.
	*.a.2 El sistema regresa a la vista anterior.
Prototipo de Interfaz	
 <p>The screenshot shows a web application interface. On the left is a navigation menu with categories: Administrador (Gestionar Cuadro, Gestionar Reserva, Idoneidad), Cuadro (Expediente, Evaluación Trimestral, Evaluación Anual, Plan Trabajo Área), and Reserva (Ficha de Reserva, Plan Capacitación). The main content area is titled 'GESTIÓN DE CUADROS DEL SISTEMA' and shows a list of records. A modal dialog box is open, displaying a warning icon and the text '¿Estás seguro que desea eliminar el cuadro?'. The dialog has 'SI' and 'NO' buttons. Below the dialog, there are 'Aceptar' and 'Cancelar' buttons. The record being targeted is for 'Alina Flores Rodríguez, Jefe Dpto. Ciencias Básicas'.</p>	

2.6 Arquitectura del sistema

La arquitectura de *software* es la estructura de un sistema o bien la forma en que va a estar organizado este; incluye los elementos que lo conforman, sus propiedades visibles desde el exterior y las relaciones que existen entre ellos (PRESSMAN, 2010).

Para el desarrollo de la propuesta de solución se hace uso del patrón arquitectónico Modelo-Vista-Controlador, teniendo en cuenta que las herramientas y tecnologías empleadas en el desarrollo del sistema están basadas en la estructura y enfoque principal de este patrón arquitectónico. Estos elementos se pueden ver en la figura 8.

Modelo: Maneja el comportamiento y los datos del dominio de la aplicación, responde a los requerimientos de información acerca de su estado (usualmente desde la vista) y responde a las instrucciones para cambiar de estado (usualmente desde el controlador). Elementos (objetos) que contienen los datos y definen la lógica para manipular dichos datos. A menudo los objetos tienen una naturaleza reutilizable, distribuida, persistente y portátil para una variedad de plataformas (RUBENFA, 2014).

Vista: Maneja el despliegue de la información. Hace referencia a los elementos que representan algo visible en la interfaz de usuario, por ejemplo, un panel o botones. Con el fin de mostrar datos de los objetos de modelo es posible que desee crear sus propios objetos personalizados (RUBENFA, 2014).

Controlador: Interpreta las acciones del usuario de teclado y ratón, informando al modelo y/o a la vista para cambiar apropiadamente sus estados. Actúa como un mediador entre los objetos del modelo y la vista. Un objeto Controlador comunica datos de ida y vuelta entre los objetos del modelo y de la vista. Un controlador también realiza todas las tareas específicas de la aplicación, tales como la entrada del usuario o la carga de procesamiento de datos de configuración (RUBENFA, 2014).

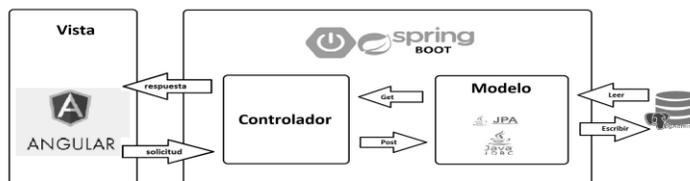


Figura 8: Patrón arquitectónico del sistema. Elaboración propia.

2.7 Patrones de diseño

Los patrones de diseño se centran en una solución a un problema concreto en el ámbito de la programación orientada a objetos, proponiendo una solución genérica de clases y relaciones para resolver dicho problema. Favorecen la reutilización de código y ayudan a construir *software* basado en la reutilización (MUNDO-INFORMÁTICO, 2013).

2.7.1 Patrones GOF

Los patrones GOF (*Gang of Four*) se dividen en tres categorías diferentes (GUERRERO, 2013):

- De creación: forma de crear instancias. Abstractar y ocultar cómo son creados e inicializados los objetos (Factoría (*Factory*), *Singleton*).
- Estructurales: cómo se combinan clases y objetos para formar nuevas estructuras y proporcionar nuevas funciones (Adaptador (*Adapter*)).
- De comportamiento: ayudan a definir la comunicación e interacción de los objetos. Reducir acoplamiento (*Observer*).

Para el desarrollo de la solución se utilizaron los patrones:

Decorador (*Decorator*): el patrón Decorador permite añadir responsabilidades a objetos concretos de forma dinámica. Los decoradores ofrecen una alternativa más flexible que la herencia para extender las funcionalidades. En Angular la carpeta “src” contiene al archivo index.html que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada una.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>ProyectoTesis</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

Figura 9. Fragmento de código del patrón Decorador.

Observador (*Observer*): este patrón define una dependencia “uno-a muchos” entre objetos, de modo que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan

actualizarse convenientemente. Este patrón es utilizado al realizar algún cambio en los datos de `ficha_movimiento`.

```
@Service
public class FichaMovimientoServicioImpl {

    @Autowired
    private FichaMovimientoRepositorio fichaMovimientoRepositorio;

    public void addFichaMovimiento(FichaMovimiento fichaMovimiento) {
        fichaMovimientoRepositorio.save(fichaMovimiento);
    }

    public FichaMovimiento getFichaMovimiento (Integer id){
        return fichaMovimientoRepositorio.findById(id).get();
    }

    public List< FichaMovimiento > listFichaMovimiento (){
        return (List< FichaMovimiento >) fichaMovimientoRepositorio.findAll();
    }

    public void editFichaMovimiento (Integer id, FichaMovimiento fichaMovimiento){
        FichaMovimiento newFichaMovimiento = fichaMovimientoRepositorio.findById(id).get();

        newFichaMovimiento.setFechaActual(fichaMovimiento.getFechaActual());
        newFichaMovimiento.setTipoMovimiento(fichaMovimiento.getTipoMovimiento());
        newFichaMovimiento.setCargoMovimiento(fichaMovimiento.getCargoMovimiento());

        fichaMovimientoRepositorio.save(newFichaMovimiento);
    }

    public void deleteFichaMovimiento (Integer id){
        fichaMovimientoRepositorio.deleteById(id);
    }
}
```

Figura 1. Fragmento de código del patrón Observador.

2.7.2 Patrones de asignación de responsabilidades GRASP

Según el patrón de arquitectura antes seleccionado y las tecnologías y herramientas elegidas para el desarrollo del *software* se decide emplear en el diseño del sistema los Patrones Generales de *Software* para Asignar Responsabilidades (GRASP: *General Responsibility Assignment Software Patterns*, según sus siglas en inglés) para apoyar la implementación del sistema, facilitar la reutilización de código y lograr que el mismo tenga alto grado de mantenibilidad.

Los patrones GRASP son esenciales dentro del desarrollo de software, pues su utilidad está dada a la hora de caracterizar objetos y clases, saber donde ubicar una responsabilidad (servicio), quien es el responsable de crear objetos, manejar eventos, etc. (GARZAS, 2014).

Para la propuesta de solución se usaron los siguientes:

Creador: la utilización de este patrón se evidencia en las clases gestoras, las cuales se encargan de crear los objetos de las clases que representan las entidades, evidenciando de esta manera que las clases gestoras son las creadoras de las entidades. En la clase CuadroServicioImpl.java se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de la clase Cuadro.java, lo que evidencia que la clase CuadroServicioImpl.java es “creador” de dichas entidades.

Controlador: asigna la responsabilidad del manejo de mensajes de los eventos de un sistema a una clase que represente el sistema global, organización, elemento activo del mundo real o manejador artificial de los eventos del sistema. El empleo de este patrón será muy útil para aquellas clases encargadas de registrar datos de entidades que manejará el sistema, y este se aplica en las clases: CuadroControlador.java, ReservaControlador.java, EvaluaciónControlador.java y demás controladoras.

Experto: se basa en asignar la responsabilidad a una clase que se encargará de ser experto de la información, la misma cuenta con la información necesaria para cumplir con la responsabilidad. Al aplicarlo al sistema, este patrón permitirá identificar las clases que poseen la información requerida, para que luego estas puedan ser manipuladas por otras. Se evidencia en las clases Entidad, las cuales son expertas en su propia información, tales como: Cuadro, Reserva, Idoniedad y demás entidades.

Alta cohesión: se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta, resolviendo el problema de mantener la complejidad en las clases dentro de los límites manejables, permitiendo definir en las clases las funcionalidades relacionadas para que estas no realicen un trabajo enorme. El controlador CuadroControlador.java asigna responsabilidades de controlar el flujo de eventos del sistema a la clase específica CuadroServicioImpl.java y este delega en otra clase las actividades con las que mantiene un modelo de alta cohesión.

Bajo acoplamiento: asigna una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Este patrón se

evidencia en toda la aplicación web, pues cada pieza de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas. Por ejemplo, se puede cambiar la URL de cierta parte de la aplicación sin tener que afectar la implementación subyacente o se puede modificar una clase de tipo `ServicioImpl.java` sin tener que modificar el código de otra clase.

2.8 Modelado del diseño para la propuesta de solución

El modelo de diseño crea un modelo de *software* enfocado en la representación de los datos, las funciones y el comportamiento requerido. Permite al ingeniero de *software* modelar el sistema o producto que se va a construir, posibilitando evaluar su calidad y efectuar mejoras antes de generar el código. Con este se obtiene una representación arquitectónica, de interfaz y despliegue del sistema (PRESSMAN, 2010).

2.8.1 Diagrama de clases del diseño

Los diagramas de clases del diseño son empleados para representar las relaciones que se establecen entre las clases. Estos incluyen la siguiente información:

- Clases, asociaciones y atributos
- Interfaces, con sus operaciones y constantes
- Métodos
- Navegabilidad
- Dependencias (PRESSMAN, 2010).

A continuación, se presenta el diagrama de clase del diseño correspondiente al RF Gestionar cuadro donde se observan los paquetes que agrupan las clases participantes y sus respectivas relaciones, el resto de los diagramas se encuentran disponibles en el Anexo 5 del documento.

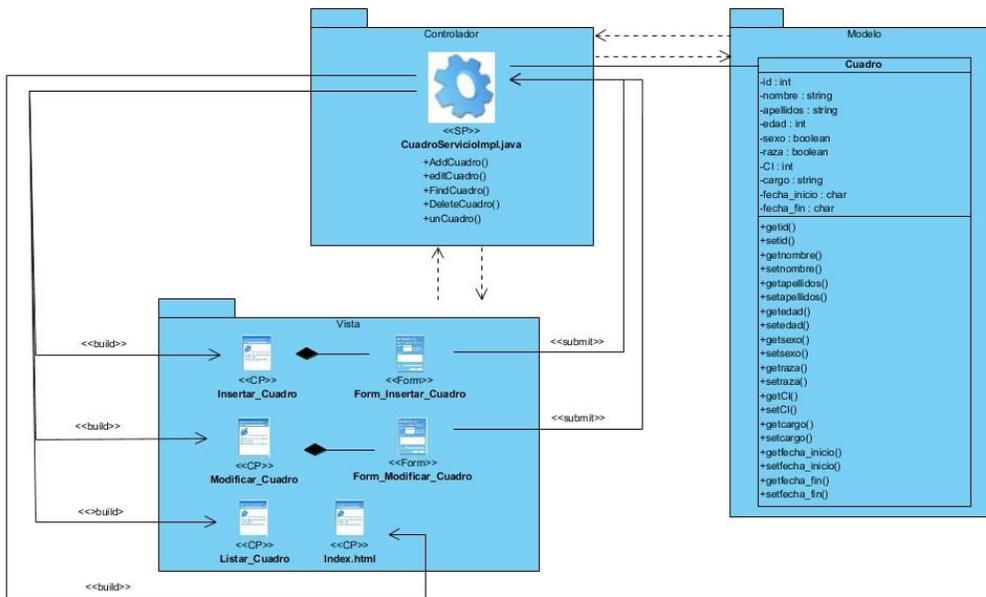


Figura 10: Diagrama de clases del diseño del caso de uso Gestionar Cuadro (Elaboración propia)

2.9 Modelo de datos

Es un medio formal para representar y manipular (en el buen sentido) los datos e información asociados a una situación real. Un modelo de datos representa la estructura estática del mundo real, es decir, respresenta los hechos existentes, no la forma en que estos cambian o evolucionan (funcionalidad)(GUTIERREZ, 2011).

2.9.1 Modelo de Entidad/Relación

El modelo Entidad/Interrelación (E/R), es un método de diseño de bases de datos, que representa los datos mediante una serie de entidades que disponen de atributos (PRESSMAN, 2010). El resultado de este proceso es una base de datos normalizada que facilita el acceso a los datos y evita su duplicado. A

continuación, se expone la estructura y modelo del diagrama de entidad relación que se ha diseñado para la propuesta de solución.

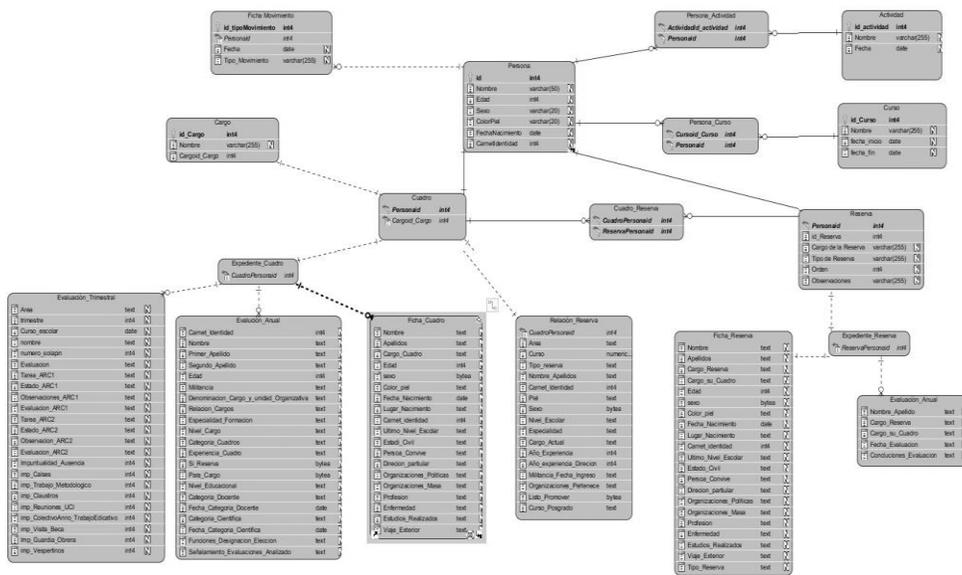


Figura 11: Modelo de datos del sistema propuesto (Elaboración propia)

2.10 Conclusiones del capítulo

Una vez culminadas las actividades de análisis y diseño de la propuesta de solución se obtuvo una visión más clara y detallada del sistema para el control de cuadros de la Facultad 1. Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de requisitos, así como la especificación de los mismos, constituyeron elementos claves en la construcción de la propuesta de solución. La utilización de los patrones de diseño permitió identificar aspectos importantes de la estructura del diseño del sistema web propuesto, lo que garantizó una mayor organización e hizo el código más

legible. Los artefactos generados constituyeron no solo la base del sistema, sino que también permitieron una mejor comprensión del mismo. Además, adoptar la arquitectura de *software* Modelo-Vista-Controlador atendiendo al uso de las herramientas y marco de trabajo de desarrollo utilizado, garantizó una propicia organización del sistema a implementar.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE CONTROL DE CUADROS DE LA FACULTAD 1.

En el presente capítulo se muestran los artefactos correspondientes a las etapas de implementación y prueba del sistema para el control de cuadros de la Facultad 1, así como los estándares de codificación que debe seguir el equipo de desarrollo para implementar el software. De acuerdo a la metodología AUO-UCI, se especifica, de los tipos de pruebas que esta plantea, el que será empleado para validar el sistema.

3.1 Diagrama de componentes

Un diagrama de componentes es un diagrama de tipo del Lenguaje Unificado de Modelo que representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre ellos, estos proveen una vista arquitectónica de alto nivel del sistema, permite además tomar decisiones respecto a las tareas de implementación y ayuda a los desarrolladores a visualizar el camino de la implementación (PRESSMAN, 2010).

De los enfoques propuestos por Pressman para la definición de componentes se decidió utilizar el enfoque Orientado a Objetos, donde se mapean las clases a componentes. Teniendo como referencias el diagrama de clases del capítulo anterior se obtiene un diagrama de componentes para el caso de uso Gestionar Cuadro, el resto de los diagramas de componentes siguen un comportamiento similar al presentado a continuación.

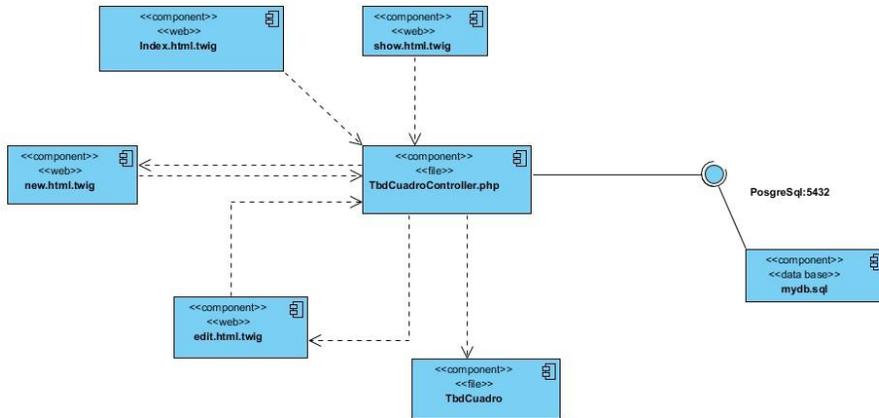


Figura 2. Diagrama de componentes. (Elaboración propia)

3.1.1 Descripción del diagrama de componentes

Plantilla: Paquete que agrupa a las plantillas encargadas de presentar la información al usuario.

Vista: Paquete que agrupa a todos los componentes que interactúan con el paquete de clases Modelo; estos componentes permiten trabajar con algunas utilidades sobre los formularios y la renderización de la información en las plantillas apropiadas.

Modelo: Paquete que agrupa las clases que representan el dominio de entidades de la base de datos y que permiten la interacción directa con el paquete Vista.

3.2 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura de un sistema en tiempo de ejecución; muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del

software se trazan en esos nodos. Las relaciones que existen entre nodos representan los protocolos de comunicación que se utilizan para acceder a cada uno.

A continuación, se muestra el diagrama de despliegue definido para la solución propuesta que cuenta con 3 nodos principales. El nodo PC Cliente que requiere de un navegador web, el nodo Servidor Web donde debe estar instalado el Sistema web para la gestión de la información del Departamento de Cuadros y el nodo SGBD en el cual debe estar instalado el Sistema Gestor de Bases de Datos PostgreSQL con la base de datos de los cuadros de la Facultad 1. El nodo PC Cliente representa las estaciones de trabajo de los usuarios que se conectan al sistema, las mismas realizan peticiones al Servidor Web mediante el protocolo HTTPS. El nodo Servidor Web, a su vez, estará conectado al nodo SGBD mediante el protocolo TCP/IP.

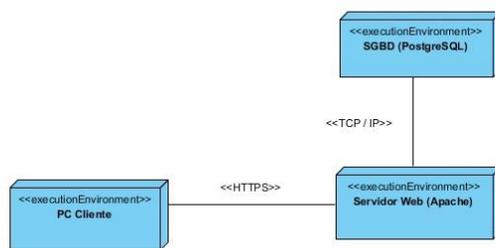


Figura 13: Diagrama de despliegue (Elaboración propia)

3.3 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación del código fuente de un software. Es preciso definir una serie de pautas que lleven por objetivo uniformar la estructura del código de manera tal que este sea lo más legible posible, como si un único programador hubiese escrito todo el código de una sola vez (MICROSOFT, 2015). Las pautas fundamentales definidas para la implementación son las siguientes:

- 1- El tamaño máximo de las líneas de código debe ser de cien a ciento veinte caracteres aproximadamente, de manera tal que se garantice la completa visibilidad de las líneas de código sin necesidad de realizar desplazamiento horizontal.

```
@Service
public class CargoServicioImpl implements CargoServicio {

    @Autowired
    CargoRepositorio cargo_repositorio;

    public List<Cargo> ListaCargo(){
        return (List<Cargo>) cargo_repositorio.findAll();
    }

    public Cargo ObtenerCargo(Integer id){
        return cargo_repositorio.findById(id).get();
    }
}
```

Figura 3. Fragmento de código de estándar de codificación 1.

- 2- Los nombres de las clases y las funciones adoptarán la notación *UpperCamelCase* y no se utilizará el guion bajo como delimitador entre palabras.

```
public void EditCargo( Integer id, Cargo newCargo){
    Cargo cargo = cargoRepositorio.findById(id).get();
    cargo.setNombre(newCargo.getNombre());
    cargoRepositorio.save(cargo);
}
```

Figura 4. Fragmento de código de estándar de codificación 2.

- 3- Las sentencias de importación no tienen líneas en blanco entre ellas.

```
import { Component, OnInit } from '@angular/core';
import { CuadroService } from '../cuadro.service';
import { Router } from '@angular/router';
import { CargoService } from '../../cargo/cargo.service';
import { Observable } from 'rxjs';
import { Cargo } from '../../cargo/modelo/cargo.model';
import { Cuadro } from '../modelo/cuadro.model';

@Component({
  selector: 'app-cuadro-crear',
  templateUrl: './cuadro-crear.component.html',
  styleUrls: ['./cuadro-crear.component.css']
})
```

Figura 5. Fragmento de código de estándar de codificación 3.

- 4- Los nombres de los atributos, variables y parámetros tendrán todas las letras en minúsculas y usarán el guion bajo como delimitador entre palabras.

```
public class Persona {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private Integer id;

    private String carne_didentidad;
    private String nombre;
    private String primer apellido;
    private String segundo apellido;
    private String numero solapin;
    private Integer edad;
    private Sexo sexo;
    private ColorPiel color piel;
    private Date fecha nacimiento;
    private String lugar nacimiento;
    private String direccion particular;
    private String estado civil;
    private String militancia;
    private String enfermedad;
```

Figura 6. Fragmento de código de estándar de codificación 4.

3.4 Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de calidad del software, y representan una revisión final de las especificaciones del diseño y de la codificación. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (PRESSMAN, 2010).

La metodología que guía el proceso de desarrollo de la presente investigación propone realizar pruebas internas, de liberación y de aceptación. Sin embargo, para la solución propuesta solo se formalizarán las pruebas internas, puesto que se encargan de verificar el resultado de la implementación al probar cada construcción. Las otras pruebas que propone la metodología requieren de una entidad certificadora y una aprobación por parte del cliente, elementos que no se contemplan dentro de los requerimientos del presente trabajo.

3.4.1 Estrategias de prueba

Una estrategia de prueba del software integra los métodos de diseño de casos de prueba en una serie bien planteada de pasos que va a converger en la eficaz construcción del software. Dicha estrategia debe ser lo suficientemente flexible como para promover un enfoque personalizado, y al mismo tiempo lo adecuadamente rígido como para originar una planeación razonable y un seguimiento administrativo del avance del producto (PRESSMAN, 2010).

Pressman plantea las siguientes estrategias de prueba:

- Pruebas de unidad
- Pruebas de validación
- Pruebas de sistema

A la solución que se propone se le realizará cada una de las pruebas antes definidas.

3.4.1.1 Pruebas de unidad

Las pruebas de unidad descomponen las funciones del programa en comportamientos comprobables discretos que se pueden probar como unidades individuales. Están destinadas a verificar las unidades más pequeñas del software. Se aplican a las funcionalidades para verificar que los flujos de control y datos están cubiertos y funcionan tal como se espera. El objetivo de estas pruebas es ejecutar un código fuente al llamar directamente a los métodos de una clase pasándole a estos los parámetros adecuados (MICROSOFT, 2015). La prueba de unidad siempre está orientada a caja blanca.

Según plantea (BAUTISTA, 2012), las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está en gran medida ligado al código fuente. Para realizar este tipo de prueba, el probador escoge diferentes valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y asegurarse de que se devuelven los valores de salida adecuados.

3.4.1.2 Pruebas de validación

Las pruebas de validación consisten en realizar acciones visibles para el usuario y en la salida que el sistema tiene ante cada una de estas acciones. La validación del software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad con los requisitos funcionales. Las pruebas de caja negra se realizan sobre la interfaz del software, sin tener en cuenta el funcionamiento interno del sistema, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas (PRESSMAN, 2010).

CONCLUSIONES GENERALES

Una vez finalizada la investigación y el desarrollo del Sistema para el control de cuadros de la Facultad 1 se concluye que:

- Con el estudio de los principales conceptos, así como la investigación mediante la consulta de bibliografía nacional e internacional relacionada con el trabajo con cuadros, permitió sentar las bases para el desarrollo e implementación del Sistema para el control de cuadros de la Facultad 1
- La modelación del negocio permitió comprender el funcionamiento del sistema a desarrollar, y sirvió de base para la captura y descripción de requisitos, los cuales fueron correctamente validados.
- A través del diseño del sistema partiendo de los requisitos capturados, permitió demostrar que las clases presentan una elevada reutilización, baja complejidad de implementación y responsabilidad, además de una elevada reutilización, bajo acoplamiento, complejidad y necesitan poca cantidad de pruebas.
- Se implementaron las funcionalidades del Sistema para el control de cuadros de la Facultad 1 partiendo del modelo propuesto, dándole solución al problema planteado.
- Finalmente se evaluó la funcionalidad del sistema a través de pruebas de software efectuadas para el nivel de unidad, las cuales arrojaron que el software implementado cumple con las condiciones deseadas.

RECOMENDACIONES

Una vez concluido este trabajo de diploma el autor recomienda:

- Terminar la aplicación y aplicarle la estrategia de pruebas para un correcto funcionamiento.
- Agregar funcionalidades al sistema que permitan la gestión de los demás subprocesos que lo integran, dígase Disciplina Laboral y Estimulación moral y material.
- Ampliar su utilización a todas las áreas de la Universidad de las Ciencias Informáticas.

REFERENCIAS BIBLIOGRÁFICAS

1. ALEJO, E. N. (2016). *Programación del lado del cliente* [Consultado el: 22 Febrero 2020 Disponible en: <https://prezi.com/fupw8qyatpuq/programacion-del-lado-del-cliente/>].
2. AMADOR, C. M. (2015). *Sistema Integral para la Gestión del los Procesos Sustantivos de la Unión de Jóvenes Comunistas en la UCI*. Universidad de las Ciencias Informáticas, 2015.
3. AMBLER, S. W. (2017). *An Executive's Guide to Disciplined Agile: Winning the Race to Business Agility*. 2017. ISBN 978 153 985 2964.
4. ARCHITECT, E. (2016). *Modelo de Caso de Uso de 22 Febrero 2020*. Disponible en: http://www.sparxsystems.com.ar/resources/tutorial/use_case_model.html.
5. BAUTISTA, Y. M. G. J. C. C. (2012). *Subsistema de aprovisionamiento de usuarios para el Sistema de Administración de Identidades*. CISED. Universidad de las Ciencias Informáticas, 2012.
6. BELL, D. (2004). *UML basics: The component diagram*. 2004,
7. BELTRÁN, P. (2016). *¿Qué es una herramienta CASE?* 2016,
8. BORRELL, M. D. L. Á. L. (2006). *Modelo de Gestión para el Trabajo con Cuadros en Cuba*. 2006
9. CABRERA, D. T. (2013). *Perfeccionamiento del Sistema de Gestión de Cuadro en la Universidad Central "Marta Abrey" de Las Villas*. Grado, Universidad Central Marta Abreu, 2013.
10. CASTILLO, E. J. M. (2014). *Lenguajes de Programación del lado del Servidor* MichelleTorres, Disponible en: <https://blog.michelletorres.mx/lenguajes-de-programacion-del-lado-del-servidor/>.
11. CLARK, S. (2016). *Web-based Mobile Apps of the future using HTML5, CSS and JavaScript* HTMLGOODIES, Disponible en: <https://www.htmlgoodies.com/beyond/article.php/3893911/Web-based-Mobile-Apps-of-the-Future-Using-HTML-5-CSS-and-JavaScript.html>.
12. DÍAZ, M. F. (2009). *Sistema Integral de Gestión de la UJC UCI: módulo militante*. Universidad de las Ciencias Informáticas, 2009.
13. DOCS, W. P. (2019). *Web Platform*.org, Disponible en: <https://webplatform.github.io/docs/#HTML>.
14. E-VALÚAME (2012). *Software Online para Evaluar el Desempeño y el Clima Laboral*. E-Valúame, de 2020]. Disponible en: <http://www.evaluame.es/index.html>.
15. FERNÁNDEZ, A. (2018). 2018, Disponible en: <https://www.controllaboral.es/>.
16. FLORES, J. C. D. (2012). *Administración. ¿Control de Gestión o Gestión de Control?* 2012, vol. 7, 69-80 p. ISBN 1992-1896.
17. GARZAS, J. (2014). *Los Patrones GRASP* Disponible en: <https://www.javiergarzas.com/2014/08/los-patrones-grasp.html>.
18. GUERRA, E. F. (2016). *Introducción a TypeScript* desarrolloweb.com, de 2020].
19. GUTIERREZ, D. (2011). *Modelo de Datos*. Venezuela: 2011,
20. HRIDER (2020). *Software de Evaluación del Desempeño y Feedback* 360 HRider, Disponible en: <https://www.hrider.net>.
21. INGENIERÍA-DE-SOFTWARE (Curso 2013-2014). *Guión Visual Paradigm for UML*. España: Curso 2013-2014.
22. ISO:9000 (2015). 2015,
23. ISO:9001 (2015). 2015,
24. JAVA (2019). *Java 8 Java*, Disponible en: <https://www.java.com/es/download/faq/java8.xml>.
25. JETBRAINS (2019). *IntelliJ IDEA. Capable and Ergonomic IDE for JVM* de 2020]. Disponible en: <https://www.jetbrains.com/idea/>.

26. JETBRAINS.ORG (2020). *FAQ. Intellij IDEA Community Edition FAQ* JetBrains.org, [Consultado el: 2020 Disponible en: <https://www.jetbrains.org/display/LJOS/FAQ>].
27. JULIE (2019). *Powerful Prototyping and Developer Handoff* Axure, Disponible en: <https://www.axure.com/>.
28. LEDO, M. J. V. (2012). *Gestión de la Información y el Conocimiento*. Educ Med Super ed. La Habana: 2012, vol. 26, ISBN 0864-2141.
29. LIVIU, M. (2014). Comparative Study on Software Development Methodologies. *Database Systems Journal*, 2014, n° p. 41.
30. LÓPEZ, A. M. (2016). *Gestión y diseño: Convergencia disciplinar*. 2016, 40 p.
31. LÓPEZ, J. G. (2019). *THINK & SELL* Disponible en: <https://thinkandsell.com/servicios/consultoria/software-y-sistemas/sistemas-d-gestion-normalizados/>.
32. LUCA, D. D. (2019). *Desarrollo Web Capacitación y Consultoría. Visual Studio Code: características principales de 2020*. Disponible en: <https://damiandeluca.com.ar/visual-studio-code-caracteristicas-principales>.
33. MARTÍN, R. G. (2016). Cuba: Scielo, 2016, vol. 20, ISBN 1029-3019.
34. MATOS, D. G. (2011). *Gestión de Recursos Humanos* Última actualización: 10 de Julio. Disponible en: http://gesrehu.blogspot.com/2011/07/integrantes_10.html.
35. MDN (2019). Disponible en: <https://developer.mozilla.org/es/docs/HTML/HTML5>.
36. MELGAREJO, A. I. (2014). *Sistemas Gestores de Base de Datos*. 2014,
37. MICROSOFT (2015). Convenciones de código.. Estándares de codificación. 2015, n°
38. MUNDO-INFORMÁTICO (2013). *GoF – Patrones de diseño (II): ¿Qué son?* [Consultado el: 20 Febrero 2020 Disponible en: <https://infow.wordpress.com/2013/01/30/gof-patrones-de-disenoii-que-son/>].
39. OTTO, M. (2016-2020). *Bootstrap 4, el manual oficial*. Traducido por: Eguiluz, J. Uniwebsidad, 2016-2020.
40. PÉREZ, C. G. (2011). *ConML. Conceptual Modelling Language Technical Specification. Versión 1.1* Disponible en: <https://digital.csic.es/handle/10261/36481>.
41. PÉREZ, J. C. M. (2018). *Entornos de Desarrollo*. EDITORIAL SÍNTESIS ed. España: 2018. ISBN 878-84-9171-161-2.
42. PGADMIN (2019).
43. PHILLIP WEBB, E. A. (2019). *Spring Boot Reference Documentation*. 2019, Disponible en: <https://docs.spring.io/spring-boot/docs/2.2.4.RELEASE/reference/pdf/spring-boot-reference.pdf>.
44. POSTGRESQL (2019). *PostgreSQL: The World's Most Advanced Open Source Relational Database* PostgreSQL, Disponible en: <https://www.postgresql.org>.
45. PRECHT, P. (2016). *Angular 2 is out - Get started here*. Thoughtram, 2016, Disponible en: <https://blog.thoughtram.io/angular/2016/09/15/angular-2-final-is-out.html>.
46. PRESSMAN, R. S. (2010). *Software Engineering: A practitioner's Approach*. 7 ed. Estados Unidos: 2010.
47. RUBENFA (2014). *Patrones de diseño: qué son y por qué debes usarlos*. Genbeta:dev, 2014, Disponible en: <http://www.genbetadev.com/metodologias-deprogramacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>.
48. RUMBAUGH, J. (2000). *El Lenguaje Unificado del Modelado. Manual de referencia*. Madrid-España: 2000. ISBN 84-7829-037-0.

REFERENCIAS BIBLIOGRÁFICAS



49. S.L, A.-C. (2018). Axarnet-Comunications S.L, [Consultado el: 22 Febrero 2020 Disponible en: <https://www.axarnet.es/blog/lenguajes-del-lado-del-servidor/>].
50. SÁNCHEZ, T. R. (2014). *Metodología de Desarrollo para la Actividad*. Editado por: Uci. La Habana: 2014,
51. SIVARANJANI, J. (2017). A Study on Software Development. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2017, vol. 5, nº 4, p. 7728.
52. SOMMERVILLE, I. (2007). *Ingeniería de Software* Traducido por: María Isabel Alfonso Galipienso, A. B. M., Francisco Mora Lizán. 7ma ed. Reino Unido: 2007. ISBN 84-7829-074-5.
53. SYNERGIX., T. Y. (2008). *Modelo de Dominio* [Consultado el: 20 de Febrero de 2020]. Disponible en: <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
54. TOLEDO, Á. (2010). *GestPeople. Completo sistema de gestión de recursos humanos*. UptoDown, Última actualización: 19 Marzo. Disponible en: <https://gestpeople.uptodown.com/windows>.
55. VARGAS (2010). *Modelo de Casos de Uso del Sistema* Expo Tecnomá, [Consultado el: 22 Febrero 2020 Disponible en: <http://facilte.blogspot.com/2010/02/modelo-de-caso-deuso-del-sistema.html>].
56. WEB20.CO (2015). *Lenguaje Unificado de Modelo UML* Disponible en: <https://www.we20.co/desarrollo/lenguaje-unificado-de-modelado-uml/>.

ANEXOS

Anexos

Anexo 1. Fases de la Metodología AUP-UCI

Fases	Descripción
Inicio	El objetivo de esta fase es obtener una comprensión común cliente -equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
Elaboración	El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
Construcción	Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
Transición	El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Anexo 2. Disciplinas de la Metodología AUP-UCI

Disciplina	Descripción
Modelo	El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio. Agrupa los flujos de

	trabajos de Modelado de negocio, Requisitos y Análisis y Diseño.
Implementación	El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.
Prueba	El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.
Despliegue	El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.
Gestión de configuración	El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
Gestión de proyectos	El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc.), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.
Entorno	El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (<i>hardware</i> , <i>software</i> , etc) estén disponibles para el equipo según sea necesario.

Anexo 3: Roles de la Metodología AUP-UCI

Roles	Descripción
Administrador de proyecto	Maneja a los miembros construye relaciones con los <i>stakeholders</i> , coordina interacciones con los <i>stakeholders</i> , planea, maneja y asigna los recursos.
Ingeniero de procesos	Desarrolla, adapta y apoya sus materiales del proceso del <i>software</i> .
Desarrollador	Escribe, testea y construye <i>software</i> .
Administrador de Base de Datos	Diseña, prueba, desarrolla, y apoya los esquemas de la BD.
Modelador ágil	Crea y desarrolla modelos, bosquejos o los archivos de la herramienta CASE, de una manera evolutiva y de colaboración.
Administrador de la configuración	Un encargado de la configuración es responsable de proporcionar la infraestructura total y el ambiente del CM al equipo de desarrollo.
Stakeholder	
Administrador de pruebas	Responsables del éxito de la prueba, incluyendo el planeamiento, la gerencia, y la defensa para la prueba y las actividades de la calidad.
Probador	Encargado de ejecutar las pruebas.