



Facultad de Ingeniería y Ciencias Técnicas de la Educación

Tesis presentada en opción al Título de Ingeniero Informático

Sistema Informático para la Gestión del Proceso Docente Educativo de Estancia Médica en el servicio terapia intermedia del Hospital General Docente de Guantánamo.

(SIGEM)

Autora: Arletis Matos Rodríguez

Tutora: MsC. Mary Luz Matos Utria

Guantánamo, Julio de 2020

“Año 62 de la Revolución”

AGRADECIMIENTOS

Quiero agradecer a mi esposo, a mi madre y familia por ayudarme y apoyarme en emprender este camino de superación, por darme tanto amor, comprensión y consejos, por estar siempre a mi lado.

A mis compañeros de estudios por siempre estar ahí apoyándome, porque siempre hemos compartidos los momentos buenos y malos, a las personas que me han ayudado de una forma u otra a que este sueño se haga realidad.

A los profesores de la carrera por transmitir sus conocimientos y valores en el transcurso de estos años.

A mí tutora MsC: Mary Luz Matos Utria por creer en mí, por su tiempo y dedicación, por compartir sus conocimientos y ayudarme a que este trabajo se haya realizado, gracias por su esfuerzo a que todo saliera adelante.

Mi más sincero agradecimiento a todos los que han hecho posible este trabajo.

DEDICATORIA

Dedico este trabajo a las personas más importantes de mi vida, a mi esposo, a mi madre y familia por todo el amor brindado, a mis amigos y compañeros de estudios por todo su apoyo incondicional. A mí tutora por guiarme, a todas aquellas personas que confiaron en mí.

RESUMEN

En la presente investigación se exponen los antecedentes, fundamentos teóricos y el estado actual del proceso docente educativo de estancia médica en el Hospital General Docente (HGD) Dr. Agostinho Neto de Guantánamo, específicamente en el servicio de terapia intermedia.

Para lograr una mejor calidad en los resultados de la investigación del proceso a informatizar se aplicaron diferentes métodos teóricos y empíricos, los cuales permitieron determinar las insuficiencias existentes en la gestión del proceso docente educativo de estancia médica en el servicio de terapia intermedia del Hospital General Docente (HGD) Dr. Agostinho Neto. Para dar solución al problema se desarrolla una aplicación web, utilizando como metodología de desarrollo de software el Proceso Unificado de Rational (RUP) y como tecnologías el framework Laravel, el gestor de base de datos MySQL y el servidor web Apache, los lenguajes HTML y las hojas de estilos CSS. Se pudo además demostrar la factibilidad de desarrollar la aplicación web.

Palabras clave: Estancia; Atención médica; Proceso docente educativo, aplicación web.

ABSTRACT

In investigation this letter expose the antecedents, theoretical foundations and the states current of the educational educational process of medical stay in the service of intermediate therapy of the general educational hospital (HGD) Dr. AgostinhoNeto.

To achieve a better quality in the results of the investigation of the process to computerize it applied different theoretical and empiric methods, those which permitted decide the existent insufficiencies in the step of the educational educational process of medical stay in the service of intermediate therapy of the general educational hospital (HGD) Dr. AgostinhoNeto, for the who develops a web application, by using as methodology of development of software the unified process of Rational (RUP) and as technologies the Laravel framework, the manager of MySQL database and the Apache web server, the HTML and CSS languages. It could moreover demonstrate the feasibility to develop the web application.

Keywords: Stay; Medical attention; Process educational educational, web application.

INDICE

INTRODUCCIÓN.....	1
Capítulo I: Caracterización del proceso docente educativo de estancia médica del HGD Dr. Agostinho Neto.....	7
1.1 El proceso docente educativo de estancia médica en el HGD Dr. Agostinho Neto y su gestión en el servicio de Terapia intermedia.....	7
1.2 Descripción de las tendencias y tecnologías actuales utilizadas en el desarrollo del sistema propuesto.	10
Metodologías de desarrollo del software	10
Los frameworks en el desarrollo de sistemas Web.....	16
Sistemas de Gestión de Bases de Datos (SGBD)	18
1.3 Modelado del negocio	19
Actores del negocio	19
Trabajadores del Negocio	20
Diagramas de casos de uso del negocio.....	21
Descripción del caso de uso del negocio Gestionar estudiantes	21
Diagrama de actividades	23
Descripción literal de caso de uso del negocio controlar evaluaciones	¡Error!
Marcador no definido.	
Reglas del negocio	24
Conclusiones del capítulo.....	24
Capítulo 2. Diseño del sistema informático para la gestión de las estancias médicas en la sala de terapia intermedia del Hospital General Docente Dr. Agostinho Neto.	25
2.1 Especificación de los requisitos de software.	25
Requisitos funcionales:	25
Requerimientos no funcionales.....	26
Descripción de actores del sistema	29
Descripción de bajo nivel de los Casos de Uso del sistema	30
2.2 Análisis del sistema.	36
2.3 Diseño del sistema.	38
Patrón de diseño.....	38
Descripción de la arquitectura del software.....	41
Modelo de datos	43

2.4 Implementación.....	44
Diagrama de despliegue	45
2.5 Diseño de casos de prueba	45
2.6 Estudio de factibilidad.....	49
Estimación basada en el método de puntos en casos de uso.....	49
Paso 4. Estimación del tiempo de desarrollo del proyecto (TDES):.....	54
Análisis de factibilidad	55
Conclusiones del capítulo.....	60
CONCLUSIONES GENERALES	61
RECOMENDACIONES	62
BIBLIOGRAFÍA.....	63
ANEXOS	

INTRODUCCIÓN

Cuba es un país de referencia internacional por sus indicadores sanitarios y la preparación integral del personal que trabaja en el sistema nacional de salud. Uno de los pilares de la calidad formativa consiste en los planes de estudio. A partir del triunfo de la revolución en 1959 se produce un cambio radical en la sociedad cubana, se proclama la salud como un derecho del pueblo y deber del Estado, imprimiéndole a la atención médica un carácter gratuito y una orientación preventivo-curativa, con énfasis en los aspectos profilácticos (Valdés, 2010).

En octubre de 1959 comenzó el primer curso de Medicina que se formó totalmente con la Revolución. El Comandante en Jefe los graduó el 14 de noviembre de 1965 en el Pico Turquino. Al inicio del período revolucionario socialista tenía una sola facultad de Medicina, cuenta hoy ante el asombro del mundo con 4 institutos superiores de Ciencias Médicas, 21 universidades de ciencias médicas y 20 filiales que abarcan todas las provincias del país, el Instituto de Ciencias Básicas y Preclínicas "Victoria de Girón", la Escuela Nacional de Salud Pública de posgrado, la Escuela Latinoamericana de Medicina y la Escuela de Medicina Caribeña unida a la Facultad de Medicina No.2 de Santiago de Cuba y un plan de estudios enfocado a la atención médica primaria, que ha hecho posible la dispensarización de acciones de salud a casi la totalidad de la población del país con el modelo de atención médica primaria del médico y la enfermera de la familia¹.

A partir de la Reforma Universitaria en 1962, la enseñanza de la Medicina sufrió cambios paulatinos en el tiempo de acuerdo a las necesidades de la población y a las premisas de la educación superior en general y a la educación médica superior en particular, donde se establecieron las estancias clínicas para todos los estudiantes.

Las estancias se corresponden con las actividades docentes de una sola asignatura o de una asignatura principal que coordina o integra contenidos de

¹Tomado del Plande estudio Medicina en Cuba 1959 a 2010, del autor V. Valdés (2010). Publicado por la Editorial Ciencias Médicas. Educación Médica Superior.

otras asignaturas básicas, básico-específicas o especiales durante la carrera. Mediante la educación en el trabajo, en las estancias los educandos adquieren habilidades sensoriales o de semiotecnia, de raciocinio clínico y de los procedimientos atencionales y terapéuticos de tipo manual, así como en las modificaciones de la esfera afectiva en una actitud consecuente con el desarrollo de la salud pública en la sociedad socialista.

Existen numerosas actividades enmarcadas en la educación en el trabajo, clasificadas sobre la base de los objetivos que se deben alcanzar, aunque entre las principales relacionadas con la formación del egresado de ciencias médicas figuran: el pase de visita, la reunión de alta, la atención ambulatoria, la guardia médica, la entrega de guardia, la discusión diagnóstica con distintas modalidades y la atención médico quirúrgica.

Las estancias clínicas se establecieron para todos los estudiantes. Se amplió la duración de la carrera a seis años para incorporar al final de la carrera el Internado Rotatorio, a fin de que todos los estudiantes logran la práctica médica que les permitiera al egreso ser efectivos en su actuación como médicos. Esta decisión dio origen, sin lugar a duda, a una de las transformaciones más fructíferas y plausibles en la educación médica cubana².

El Hospital General Docente (HGD) Dr. Agostinho Neto de Guantánamo es un hospital clínico y quirúrgico donde se llevan a cabo varios procesos, dentro de ellos el proceso docente educativo, teniendo un amplio claustro de profesores que brindan atención docente de pregrado y postgrado. La formación de los profesionales de la salud se basa en el principio martiano y marxista de la vinculación estudio-trabajo, a lo que se denomina educación en el trabajo, como modalidad fundamental de organización del proceso docente educativo en el ciclo clínico de las diferentes carreras de la educación médica superior.

En el servicio de terapia intermedia se desarrolla de forma diaria una entrega de guardia donde se analizan los resultados alcanzados por los estudiantes, se le da

²[https://scielo.sld.cu/Formación Médica](https://scielo.sld.cu/Formación_Médica)

nota y se puntualizan las actividades docentes a realizar. Este proceso es realizado por el médico de guardia entrante y saliente. En entrevistas realizadas a este personal médico para conocer cómo se lleva a cabo actualmente dicho proceso se detectaron las siguientes insuficiencias:

- Se lleva manualmente la tarjeta de control y evaluación de estancia médica lo que induce a la duplicidad de los datos debido al cúmulo de información que se procesa, lo cual genera complejidad en el trabajo que se realiza, dificultando un posterior análisis estadístico.
- La dispersión de los datos y el gran volumen de información en soporte plano, provoca demora en la actualización y búsqueda de la información.

Lo antes expuesto trae consigo la siguiente **problemática a resolver**: Insuficiencias en la gestión del proceso docente educativo de estancias médicas en el HGD Dr. Agostinho Neto de Guantánamo. El problema antes planteado enmarca como **objeto de estudio** el proceso docente educativo de estancias médicas en el HGD Dr. Agostinho Neto y como **campo de acción** la gestión del proceso docente educativo de estancia médica en el servicio de terapia intermedia del HGD Dr. Agostinho Neto de Guantánamo.

Esta investigación tiene como **objetivo** mejorar la gestión del proceso docente educativo de estancias médicas del HGD Dr. Agostinho Neto de Guantánamo a través del desarrollo de una aplicación web.

Como **idea a defender** se plantea que con el desarrollo de una aplicación web se contribuirá a disminuir las insuficiencias en la gestión del proceso docente educativo de estancias médicas del HGD Dr. Agostinho Neto de Guantánamo.

Para dar cumplimiento al objetivo trazado se determinaron las siguientes tareas:

1. Caracterizar el proceso docente educativo de estancias médicas en el HGD Dr. Agostinho Neto de Guantánamo.
2. Seleccionar las herramientas y tecnologías adecuadas para el desarrollo de la aplicación propuesta.

3. Modelar el negocio para comprender la estructura y dinámica del proceso gestión del proceso docente educativo de estancia médica en el HGD Dr. Agostinho Neto
4. Realizar el estudio de factibilidad del sistema informático implementado.
5. Realizar el análisis y diseño del sistema informático para la gestión del proceso docente educativo de estancia médica en el servicio de terapia intermedia en el HGD Guantánamo.
6. Implementar un sistema informático para la gestión del proceso docente educativo de estancia médica en el servicio de terapia intermedia en el HGD Guantánamo.
7. Realizar pruebas de caja negra con el objetivo de detectar errores antes de poner en práctica la aplicación informática.

Para profundizar en el conocimiento de las regularidades y cualidades esenciales del proceso docente educativo de estancia en el servicio de terapia intermedia del HGD Dr. Agostinho Neto, se aplicaron los siguientes métodos teóricos:

Análisis y síntesis: fue utilizado para lograr una adecuada comprensión sobre las tendencias y peculiaridades del proceso docente educativo de estancia en el servicio de terapia intermedia del HGD Dr. Agostinho Neto, la elaboración de los fundamentos teóricos y la formulación de una propuesta de solución a las insuficiencias detectadas; dicho método también permitió procesar e interpretar la documentación referente a las etapas de análisis y diseño del sistema.

Modelación: se empleó durante la etapa de elaboración del sistema propuesto con el uso del Lenguaje Unificado de Modelado (UML) y la herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) Visual Paradigm.

Inducción y deducción: estos métodos permitieron identificar las diferentes insuficiencias encontradas a partir de la investigación realizada, haciendo posible

establecer problemas más generales y permitiendo llegar a conclusiones, las cuales fueron concretadas en el sistema a informatizar.

Histórico y lógico: se empleó con el fin de estudiar la trayectoria de la problemática existente, conocer su evolución y desarrollo según los diferentes procesos de perfeccionamiento de los planes de estudio de la carrera de medicina y los aspectos teóricos fundamentales relacionados con este.

Para la obtención y elaboración de los datos y el conocimiento de los hechos fundamentales que caracterizan el proceso docente educativo de estancia en el servicio de terapia intermedia del HGD Dr. Agostinho Neto, se aplicó el **método empírico observación**, logrando observar el funcionamiento de este proceso, detectándose las insuficiencias que posee y diagnosticando su estado actual.

Se utilizaron las siguientes técnicas de recopilación de información:

Entrevistas: se aplicaron con el objetivo de obtener información, opiniones y criterios de los profesores que de alguna manera intervienen en el proceso estudiado.

Revisión de documentos: se realizó para estudiar los reglamentos, instrucciones, resoluciones legales y documentos utilizados en el proceso docente educativo de estancia del HGD Dr. Agostinho Neto, con el objetivo de comprender su funcionamiento, detectar las deficiencias existentes y recopilar los posibles requerimientos funcionales del sistema. Se analizaron los planes de estudio de medicina y el modelo de la Tarjeta de control y evaluación de estancia, así como los programas de las asignaturas relacionadas con la estancia de medicina interna.

El **aporte práctico** de la presente investigación radica en el desarrollo de una aplicación web que contribuye a disminuir las insuficiencias en la gestión del proceso docente educativo de estancia en el servicio de terapia intermedia en el HGD Dr. Agostinho Neto, permitiendo además la búsqueda de información con mayor rapidez y confiabilidad.

El **valor social** de la investigación está dado en la repercusión que tiene la aplicación informática propuesta al perfeccionamiento del proceso docente educativo de estancias médicas en el servicio de terapia intermedia en el HGD Dr. Agostinho Neto, lo que contribuye a que este proceso se desarrolle de forma más eficiente.

Capítulo I: Caracterización del proceso docente educativo de estancia médica en el HGD Dr. Agostinho Neto.

Este capítulo ofrece una caracterización del objeto de estudio, se justifican las tendencias y tecnologías utilizadas en la construcción de la propuesta; además se realiza el modelado de negocio, donde se determinan los actores, trabajadores, caso de usos del negocio y las reglas a cumplir para el proceso docente educativo de estancia médica en el HGD Dr. Agostinho Neto.

Capítulo II: Sistema Informático para la Gestión del Proceso Docente Educativo de Estancia Médica en el servicio terapia intermedia del Hospital General Docente de Guantánamo.

Este capítulo presenta la solución propuesta a partir del desarrollo de la metodología empleada para la implementación del sistema informático. Aquí se exponen los requerimientos funcionales y no funcionales, se definen los actores y casos de uso del sistema, se presentan los flujos de análisis y diseño junto a sus diagramas, se muestran el diagrama de despliegue y el modelo de datos; además del diseño de casos de prueba y un estudio de factibilidad donde se realiza una estimación de tiempo y esfuerzo basado en casos de uso.

Capítulo I: Caracterización del proceso docente educativo de estancia médica del HGD Dr. Agostinho Neto.

En el presente capítulo se describe el proceso docente educativo de estanciamédica en el HGD Dr. Agostinho Neto como objeto de estudio de la investigación y específicamente la gestión de dicho proceso en la sala de Terapia Intermedia. Además, se brinda una panorámica de las tendencias y tecnologías actuales y un análisis crítico de las herramientas utilizadas para el desarrollo del sistema propuesto. Se definen los actores, casos de usos y trabajadores del negocio, así como los diagramas de casos de uso del negocio.

1.1 El proceso docente educativo de estancia médica en el HGD Dr. Agostinho Neto y su gestión en el servicio de Terapia intermedia.

El HGD Dr. Agostinho Neto tiene como misión garantizar la atención médica gratuita y de alta calidad, enfocado fundamentalmente en el paciente grave, la atención materno-infantil y el programa quirúrgico, posee un amplio claustro de profesores que brindan la atención de docencia de pregrado y postgrado. Para ello cuenta con varios departamentos y salas, los cuales desarrollan disímiles funciones entre las que se encuentra el servicio de Terapia Intermedia. Dentro de los procesos que se desarrollan en esta sala se encuentra la gestión de la estancia médica, donde se controlan los resultados académicos de los internos durante su estancia en dicha sala.

Como parte del perfeccionamiento del sistema educativo, los planes de estudio han sido transformados de acuerdo a las demandas del sistema nacional de salud y los compromisos de misiones internacionalistas. La formación integral de los estudiantes, la integración docente–asistencial–investigativa, la educación en el trabajo, el aumento de las horas lectivas en la atención primaria de salud y en disciplinas de salud pública, la formación científica, y la vinculación básico-clínica y clínico-epidemiológica-social ha prevalecido.

Mediante la educación en el trabajo, en las estancias los educandos adquieren habilidades censo-perceptuales o de semiotecnia, de raciocinio clínico y de los

procedimientos atencionales y terapéuticos de tipo manual, así como en las modificaciones de la esfera afectiva en una actitud consecuente con el desarrollo de la salud pública. Las estancias se corresponden con las actividades docentes de una sola asignatura o de una asignatura principal que coordina o integra contenidos de otras asignaturas básicas, básico-específicas o especiales durante la carrera. Estas se desarrollan a partir del tercer año de la carrera.

Existen numerosas actividades enmarcadas en la educación en el trabajo. Estas se clasifican sobre la base de los objetivos que se deben alcanzar. Entre las principales, relacionadas con la formación del egresado de ciencias médicas figuran: el pase de visita, la reunión de alta, la atención ambulatoria, la guardia médica, la entrega de guardia, la discusión diagnóstica con distintas modalidades y la atención médico quirúrgica.

Los estudiantes de medicina, fundamentalmente de tercer año en adelante desarrollan una serie de rotaciones por las diferentes especialidades entre las que se encuentran medicina interna, otorrino, ortopedia, propedéutica. Este proceso también es desarrollado de forma concentrada por los estudiantes de sexto año durante el internado. De igual forma los que cursan una especialidad determinada, también desarrollan estancias. No obstante, la variedad de estudiantes, los objetivos a evaluar a estos no difieren, sino el nivel de profundidad en que se evalúan los mismos. Las habilidades a evaluar varían en dependencia de la especialidad en la cual desarrollen la estancia.

La evaluación de los internos durante su estancia forma parte del proceso docente educativo que se desarrolla en el hospital. Específicamente en el servicio de Terapia Intermedia todos los días se realiza una entrega de guardia, donde se analizan los resultados alcanzados por los estudiantes, se le da nota y se puntualizan las actividades docentes a realizar. Este proceso es realizado por el médico de guardia que sale y el entrante. Estas evaluaciones se realizan de forma diaria teniendo en cuenta la participación de los estudiantes en la docencia a través de preguntas orales y escritas. Los seminarios se evalúan con una frecuencia semanal. Las habilidades que van adquiriendo y los procedimientos manuales que

realizan en la práctica tienen un ciclo de evaluación mayormente diario. Parte de estas evaluaciones se recogen en un modelo de tarjeta de control y evaluación de estancia junto con otros datos como: datos de identificación, evaluación de la estancia, objetivos educativos, calificaciones, evaluación de clases y evaluación de educación en el trabajo. Dentro de este parámetro se mide discusión diagnóstica, guardias, pase de visitas, reuniones de alta, entrevistas médicas entre otros. En el caso de propedéutica, por ejemplo, se mide el trabajo en el llenado de las historias clínicas, la descripción de sus habilidades, dentro de las que se encuentran Inyección endovenosa, Inyección intramuscular, Inyección subcutánea, Inicio de venoclisis, Cateterismo nasogástrico y Enema evacuante.

El estudio del estado del arte realizado durante la investigación permitió conocer que en el país se han desarrollado varios sistemas informáticos con el objetivo de evaluar a los estudiantes de medicina, entre ellos figuran:

- Sistema de tareas docentes (Provincia Cienfuegos): utilizado para adquirir habilidades en las tomas de decisiones clínicas mediante cuestionarios.
- Sistema para la gestión de datos del docente (Provincia La Habana): utilizado para la preparación posgraduada de los médicos y sus evaluaciones.
- Software educativo para la autoevaluación de Morfo fisiología (Universidad de Ciencias Médicas de La Habana): para evaluar los objetivos y contenido del programa de esa asignatura en específico.

Aún con las funcionalidades que estos sistemas informáticos aportan, no cumplen con las exigencias actuales al no implementar funcionalidades tales como: datos de identificación, evaluación de la estancia, objetivos educativos-calificaciones, evaluación de clases, evaluación de educación en el trabajo, propedéutica y la descripción de habilidades.

Teniendo en cuenta las insuficiencias de los sistemas informáticos antes mencionados para resolver la problemática planteada, en esta investigación la propuesta realizada por la autora incluye las siguientes mejoras:

- Informatizar el registro de los estudiantes y la asignación de los mismos a una estancia determinada según el año o la especialidad que cursan.
- Informatizar el control de la evaluación de la estancia médica en el servicio de terapia intermedia de los internos en el HGD Dr. Agostinho Neto para una mejor organización y rapidez en la búsqueda de información.
- Mostrar las evaluaciones de la estancia.
- Realizar gráficos por los diferentes tipos de evaluaciones.
- Realizar gráficos por los diferentes tipos de habilidades.
- Una mejor visibilidad y centralización de la información.
- Se podrán generar reportes y comparaciones con respecto a otros años.

1.2 Descripción de las tendencias y tecnologías actuales utilizadas en el desarrollo del sistema propuesto.

A continuación, se describen las tendencias y tecnologías de la información empleadas en la realización de este trabajo. Estas tecnologías se eligieron teniendo en cuenta los contextos de hardware del lugar donde se establecerá el sistema, analizando las ventajas y desventajas que las caracterizan y las necesidades que debe satisfacer a los usuarios.

Metodologías de desarrollo del software

Todo desarrollo de software es riesgoso y difícil de controlar, esto significa que debe ser un proceso organizado, de lo contrario el resultado obtenido será clientes insatisfechos y desarrolladores aún más insatisfechos. No obstante, con independencia que existen metodologías de desarrollo ágiles como Scrum y XP se escoge para el desarrollo de esta investigación **Rational Unified Process (RUP)**. RUP es uno de los procesos de desarrollo de software más completo de los existentes actualmente. Unifica al equipo de desarrollo de software y optimiza la productividad de cada uno de los miembros, aprovechando que está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes. La metodología RUP se caracteriza por ser dirigido por casos de

uso, que definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio (Jacobson & Rumbaugh, 2006).

Es centrado en la arquitectura, característica que brinda una visión completa del sistema, donde se describen los procesos del negocio que son más importantes para comprenderlo, desarrollarlo y producirlo de una forma eficaz. Es iterativo e incremental permitiendo desarrollar en cada fase las iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo.

Esta metodología de desarrollo utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software.

Lenguaje Unificado de Modelado (UML, Unified Modeling Language): es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados (Craig, 2015).

UML nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que, al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad.

Para el diseño y modelado se utiliza como herramienta **(CASE)** Computer Aided Software Engineering el Visual Paradigm, la cual permite obtener los diferentes diagramas que se requieren durante los diferentes flujos de desarrollo de la aplicación web. Dentro las ventajas que esta ofrece están:

- Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requerimientos y de especificación de componentes.
- Tiene apoyo adicional en cuanto a generación de artefactos automáticamente.
- Generación de documentación en formatos HTML y PDF.

- Disponibilidad en múltiples plataformas.
- Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visión y Rational Rose.
- Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para las plataformas como .Net, Java y PHP, así como obtener los diagramas a partir del código.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

Visual Paradigm for UML Enterprise Edition (VP-UML EE): es la edición top de la línea de productos, lo que representa todo lo más moderno y agrega valor en términos de modelado de datos orientado a objetos, hace posible la documentación del proyecto, mapeo relacional de objetos para Java, .NET y PHP, reduciendo costos y aumentando su productividad.

Tipo de sistema a desarrollar: Aplicación Web.

En términos de la ingeniería de software se denomina aplicación Web a aquellas herramientas que los usuarios pueden utilizar, accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador. Estas son populares debido a lo práctico del navegador Web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs y tiendas en línea que son ejemplos bien conocidos de aplicaciones Web.

Las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar, como HTML (Hyper Text MarkupLanguage) o XHTML (Extensible HypertextMarkupLanguage), soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins, para añadir elementos dinámicos a la interfaz de usuario. Generalmente

cada página Web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. (Murugarren, 2004).

Los desarrolladores Web generalmente utilizan lenguajes interpretados (scripts) en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez, lo que suele resultar molesto a los usuarios.

Ventajas de las aplicaciones Web.

- Multiplataforma: se pueden usar desde cualquier sistema operativo porque solo es necesario tener un navegador.
- Portable: es independiente del ordenador donde se utilice porque se accede a través de una página Web, sólo es necesario disponer de acceso a la red institucional.
- Ahorra tiempo: se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: basta tener un navegador actualizado para poder utilizarlas. No ocupan espacio en el disco duro.
- Actualizaciones inmediatas: como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consume recursos en la PC porque se realizan desde otro ordenador.

Para el desarrollo del sistema se empleará la arquitectura **cliente-servidor** y el patrón arquitectónico Modelo Vista Controlador.

Modelo Vista Controlador (MVC)

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario (Clements, Kazman & Klein, 2000). Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

De manera genérica, los componentes de MVC se podrían definir como sigue:

- El Modelo: es dónde se procesa y obtienen los datos, la conexión con la base es decir es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tantas consultas como actualizaciones e inserciones.
- El Controlador: controla los datos, dicho de forma rápida obtiene datos de un modelo, los procesa, y se los pasa a la vista. Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su vista asociada, por tanto, se podría decir que el controlador hace de intermediario entre la vista y el modelo.
- La Vista: Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (interfaz de usuario).

Lenguaje del lado del servidor.

La programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta. Los lenguajes de lado del servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, JSP, PERL y PHP.

Lenguaje del lado del servidor a emplear: PHP

PHP es un acrónimo recursivo en inglés de Hypertext Pre-processor. Es un lenguaje de programación de propósito general de código del lado del servidor

originalmente diseñado para el desarrollo web de contenido. Permite la conexión a diferentes tipos de servidores de bases de datos, tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite o MongoDB. Posibilita aplicar técnicas de programación orientada a objetos. Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Lenguaje de programación del lado del cliente:HTML

Como lenguajes de programación del lado del cliente se utilizarán el Lenguaje de Marcas de Hipertexto (HTML) y Hojas de Estilo en Cascada (CSS). A continuación, se muestran algunas de las ventajas de su utilización³.

Ventajas de HTML:

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Admiten todos los exploradores.

CSS

Con una Hoja de Estilo podemos modificar la presentación de cada elemento sin modificar el código HTML, ahorrando esfuerzo y tiempo de edición. Así, el mantenimiento del sitio web se hace más sencillo. El lenguaje CSS ofrece una

³<http://lenguaje-html-cufinho.blogspot.com/p/ventajas-sencillo-que-permite-describir.html>

amplia gama de herramientas de composición más potentes que HTML. Con CSS se evita tener que recurrir a “trucos” para conseguir algunos efectos⁴.

Las Hojas de Estilo pueden usarse con otros lenguajes de programación (por ejemplo, JavaScript) para conseguir efectos dinámicos en las páginas. Se pueden especificar Hojas de Estilo para distintos navegadores. Los usuarios con alguna discapacidad pueden definir su propia Hoja de Estilo y la regla importante obligará al navegador a suplantar la Hoja de Estilo del autor de la página.

JavaScript: Es un lenguaje de programación interpretado, es decir, que no requiere compilación. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico (Bradenbaugh, 2000). Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador Web permitiendo mejoras en la interfaz del usuario y páginas Web dinámicas.

Los frameworks en el desarrollo de sistemas Web.

Un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un marco de trabajo son: acelerar el proceso de desarrollo, reutilizar códigos ya existentes y promover buenas prácticas de desarrollo como el uso de patrones.

Características de los frameworks:

- Abstracción de URLs y sesiones: no es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
- Acceso a datos: incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BDD (Bases de Datos Distribuidas), XML (Extensible Markup Lenguaje), entre otras.
- Controladores: la mayoría de los frameworks implementan una serie de controladores para gestionar eventos, como una introducción de datos

⁴<https://www.prosolutions.es/blog/ventajas-y-desventajas-de-css/>

mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.

- Autenticación y control de acceso: incluyen mecanismos para la identificación de usuarios mediante un nombre y contraseña, y permiten restringir el acceso a determinadas páginas para algunos usuarios.

Frameworks del lado del cliente a utilizar: Bootstrap y JQuery

Bootstrap: es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript opcionales (Cochran, 2012).

JQuery: es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Frameworks del lado del servidor a utilizar: Laravel

Laravel es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad en el mundo de Internet. Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web. Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas (Alvarez & Ruiz, 2018).

El frameworkLaravel trabaja con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos. Por supuesto, dispone también de una arquitectura de clases también muy adecuada, que promueve la separación del código por responsabilidades. Su estilo arquitectónico es MVC(Cesar, 2015).

Laravel dispone de diversos paquetes entre los que se destaca **InfyOmLaravelGenerator**.Infyom es un generador de código, que permite crear CRUDs medianteArtisan, APIs y Test Cases, seleccionando que campos necesitas tanto en la base de datos como para la web y sus validaciones, así mismo tiene la posibilidad de generar todo esto en un panel apoyado en el famoso templateAdminLTE.

Sistemas de Gestión de Bases de Datos (SGBD)

Los Sistemas de Gestión de Bases de Datos (Database Management System, SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Sistema de Gestión de Base de Datos a utilizar: MYSQL

Actualmente existen varios gestores de bases de datos y MySQL está entre los más usados. Entre los diferentes gestores de bases de datos encontramos SQLite, Microsoft SQL Server, Oracle, Microsoft Access, entre otros. A la hora de elegir un gestor u otro, hay que tener en cuenta las ventajas que aporta cada uno a la hora de crear, gestionar y administrar bases de datos, así como de qué forma almacena y busca la información.

MySQL es un sistema de base de datos relacionales muy popular y de los más utilizados para los sistemas de gestión de contenidos WordPress, Joomla o

Drupal. Además, tiene un conjunto de ventajas que lo hacen una excelente elección⁵:

- Es una base de datos muy rápida. Su rendimiento es estupendo sin añadirle ninguna funcionalidad avanzada.
- Es una base de datos gratuita. Al ser de código abierto, no tiene coste, con el ahorro que eso conlleva.
- Es compatible con múltiples sistemas operativos.
- Es muy fácil de usar. Podemos empezar a usar la base de datos MySQL sabiendo unos pocos comandos.
- Utiliza varias capas de seguridad. Contraseñas encriptados, derechos de acceso y privilegios para los usuarios.
- Pocos requerimientos y eficiencia de memoria. Tiene una baja fuga de memoria y necesita pocos recursos de CPU o RAM.

1.3 Modelado del negocio

Para llegar a un entendimiento más completo de la estructura y la dinámica de la gestión del proceso docente educativo de estancia médica en el servicio de terapia intermedia del HGD Dr. Agosthino Neto, se procede a realizar el modelado de este negocio, haciendo uso de diferentes artefactos modelados con el Visual Paradigm y bajo los preceptos de RUP, los cuales son:

- Actores del Negocio
- Trabajadores del Negocio
- Casos de Uso del Negocio (CUN)
- Diagrama de Clases del modelo de objetos

Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos. Lo que se modela como actor es el rol que se

⁵<https://fp.uoc.fje.edu/blog/por-que-elegir-el-gestor-de-base-de-datos-mysql/>

juega cuando se interactúa con el negocio para beneficiarse de sus resultados. El actor que participa en el subproceso de control de estancia es el siguiente:

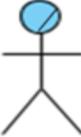
Actor del negocio	Descripción
 Estudiante	Demuestra las habilidades adquiridas durante los estudios en el servicio de estancia para obtener una evaluación.

Tabla 1.1: Actores del negocio.

Trabajadores del Negocio

Un trabajador del negocio representa un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. En este negocio se determinó el siguiente trabajador:

Trabajador del negocio	Descripción
 Profesor	Evalúa las actividades desarrolladas por los estudiantes. Lleva el control de la asistencia y el cumplimiento de los objetivos educativos propuestos durante la estancia.
 Secretaria Docente	Interactúa con los profesores, proporcionando los listados de la matrícula de los estudiantes y la ubicación correspondiente según el grado escolar o la especialidad que cursa.

Tabla 1.2: Trabajador del negocio.

Diagramas de casos de uso del negocio

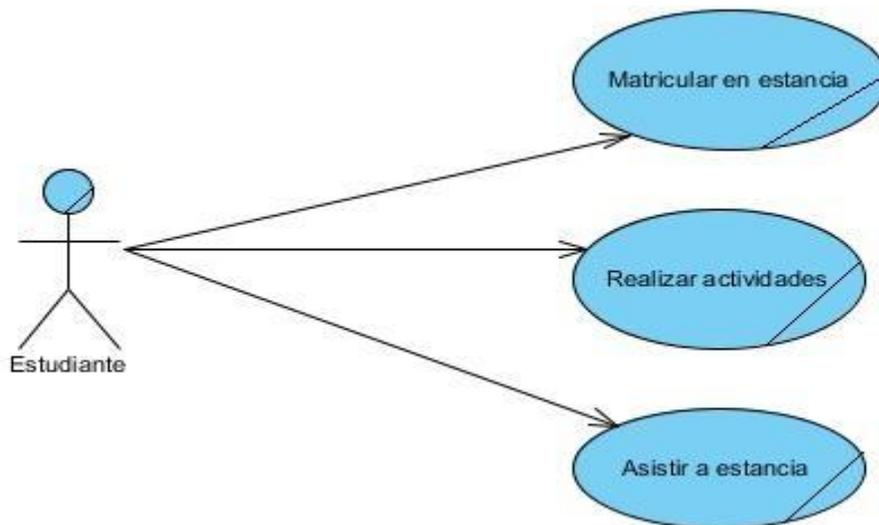


Figura 1. Diagrama de Casos de Uso del negocio actor estudiante
Descripción del caso de uso del negocio Matricular estancia

Caso de Uso del negocio “Matricular estancia”	
Actores del negocio	Estudiante
Propósito	El estudiante se matricula en una estancia determinada
Resumen	El Caso de Uso inicia cuando el estudiante acude a secretaria docente para matricular en una estancia.
Caso de Uso asociado	Asociar estudiante a estancia (incluido)
Flujo de Trabajo	
Acción del actor	Respuesta del proceso del negocio
1. El estudiante presenta en secretaria su carnet de identidad.	2. La secretaria docente registrados los datos generales de los estudiantes. Que incluyen. Nombre y Apellidos, CI, Año que cursa. 3. Los estudiantes son organizados por grupos de estudio y asignados a una estancia según el año que cursan.
Prioridad	Media
Mejoras propuestas:	

- Informatizar el registro de los estudiantes.
- Informatizar la asignación de los mismos a una estancia determinada según el año o la especialidad que cursan.

Caso de Uso del negocio “Asistir a estancia”	
Actores del negocio	Estudiante
Propósito	El estudiante se presenta al servicio donde realizara la estancia.
Resumen	El Caso de Uso inicia cuando el estudiante acude a la servicio de estancia médica para desarrollar las actividades planificadas.
Caso de Uso asociado	Asociar estudiante a estancia (incluido)
Flujo de Trabajo	
Acción del actor	Respuesta del proceso del negocio
1. El estudiante acude a la sala donde realizara la estancia según el año que cursa.	2. El profesor registrara su asistencia a la sala.
Prioridad	Media
Mejoras propuestas:	
<ul style="list-style-type: none"> • Informatizar el registro de asistencia. 	

Caso de Uso del negocio “Realizar actividades”	
Actores del negocio	Estudiante
Propósito	El estudiante realiza las actividades orientadas por el profesor según la estancia en la que se encuentre
Resumen	El Caso de Uso inicia cuando el estudiante acude a la servicio de estancia médica para desarrollar las actividades planificadas.
Caso de Uso asociado	Asociar estudiante a estancia (incluido)

Flujo de Trabajo	
Acción del actor	Respuesta del proceso del negocio
1. El estudiante acude a la sala donde realizara la estancia según el año que cursa.	2. El profesor asigna las actividades a desarrollar, las cuales pueden ser seminarios o clases prácticas. 3. El estudiante realiza las actividades y es evaluado según la calidad de su ejecución.
Prioridad	Media
Mejoras propuestas:	
<ul style="list-style-type: none"> Informatizar el registro de las actividades asignadas al estudiante y las evaluaciones obtenidas por el estudiante. 	

Diagrama de actividades

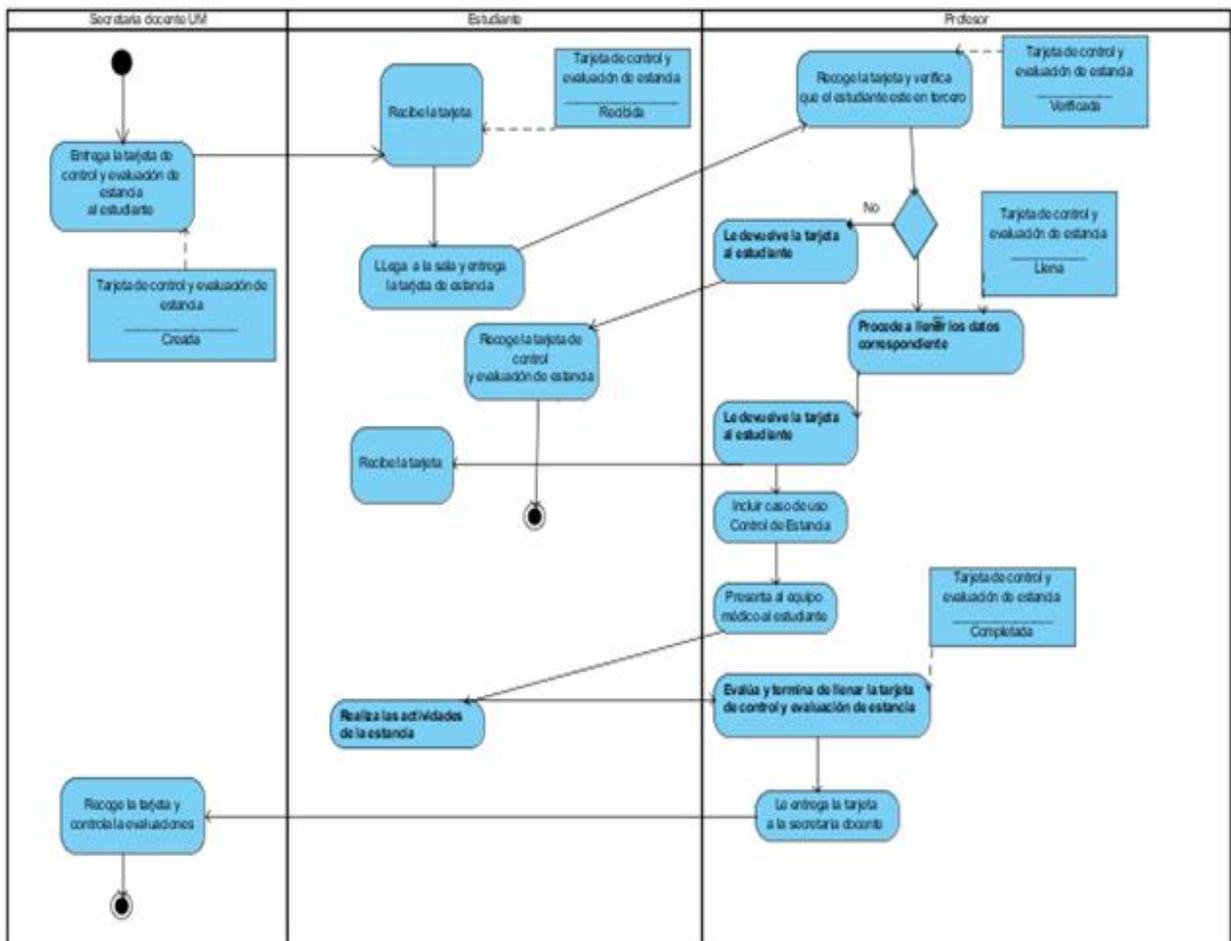


Figura 3. Diagrama de Actividades

Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, puesto que regulan algún aspecto del negocio. El proceso de especificación implica que hay que identificarlas dentro del negocio, evaluar si son relevantes dentro del campo de acción que se está modelando e implementarlas en la propuesta de solución.

Las reglas del negocio identificadas son:

1. Sólo pueden realizar la estancia los estudiantes de 3er año a 6to año.
2. Las evaluaciones se asignan en el rango de 2-5 puntos.
3. El aprobado es de 3 puntos como mínimo.
4. Los estudiantes tienen que aprobar como mínimo un seminario y dos preguntas escritas de las clases para realizar la práctica del semestre.
5. Para aprobar propedéutica los estudiantes tienen que hacer 10 historias clínicas.

Conclusiones del capítulo

1. Se caracterizó el proceso docente educativo de estancia médica que se desarrolla en el HGD Dr. Agostinho Neto, haciendo énfasis en la gestión de dicho proceso en el servicio de terapia intermedia y llegando a la conclusión de que el mismo presenta insuficiencias.
2. Fueron analizadas las herramientas y tecnologías actuales, considerando las tendencias y las políticas de migración hacia el software, seleccionando las más idóneas para la implementación de la aplicación informática propuesta.
3. La aplicación del flujo de trabajo modelado del negocio, permitió identificar los actores, trabajadores, casos de uso, así como modelar los diagramas correspondientes para comprender el proceso docente educativo de estancia médica que se desarrolla en el HGD Dr. Agostinho Neto en el servicio de terapia intermedia.

Capítulo 2. Diseño del sistema informático para la gestión de las estancias médicas en la sala de terapia intermedia del Hospital General Docente Dr. Agostinho Neto.

En este capítulo se describen los requerimientos funcionales y no funcionales, se muestran los actores del sistema, se describen los casos de uso del sistema, se mencionan los casos de uso más significativos, se efectúa el diagrama de caso de uso del sistema estructurado, se desarrollan los flujos de análisis, diseño, donde se muestran los diagramas de colaboración, modelo de entidad relación y el diagrama de despliegue. Además de la realización del estudio de factibilidad.

2.1 Especificación de los requisitos de software.

Requisitos funcionales:

Un requerimiento funcional es un requisito que especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas; requisito que especifica el comportamiento de entrada/salida de un sistema.

Durante el desarrollo de la investigación se establecieron los siguientes requisitos funcionales:

RF 1 Gestionar Estudiante.

RF 1.1 Insertar Estudiante.

RF 1.2 Modificar Estudiante.

RF 1.3 Mostrar listado de estudiantes.

RF 2 Gestionar Notas.

RF 2.1 Insertar Notas.

RF 2.2 Modificar Notas.

RF 3 Calcular promedio de notas por estudiantes.

RF 4 Mostrar listado con las notas finales.

RF 5 Mostrar listado de estudiantes por habilidades.

RF 6 Mostrar listado de estudiantes que falten por demostrar habilidades.

RF 7 Mostrar listado de estudiantes que hayan completado las habilidades.

RF 8 Categoría de evaluación.

RF 8.1 Agregar usuario.

RF 8.2 Modificar un usuario.

RF 8.3 Eliminar un usuario.

RF 9 Administrar usuarios del sistema.

RF 9.1 Agregar usuario.

RF 9.2 Modificar un usuario.

RF 9.3 Eliminar un usuario.

RF 10 Autenticar Usuario.

RF 11 Gestionar Tipo de evaluación

RF 11.1 Agregar usuario.

RF 11.2 Modificar un usuario.

RF 11.3 Eliminar un usuario.

Requerimientos no funcionales.

Los requerimientos no funcionales son requisitos que especifican propiedades o cualidades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad y confiabilidad. Requisito que especifica restricciones físicas sobre un requisito funcional.

El sistema propuesto posee los siguientes requerimientos no funcionales:

RNF 1 - Apariencia o interfaz externa.

El sistema debe estar acorde con los colores de la entidad (blanco, verde claro, azul). Debe poseer una interfaz sencilla, agradable y legible. El contenido se debe mostrar de forma comprensible y fácil de leer.

RNF 2 - Usabilidad.

Para utilizar el sistema es necesario poseer conocimientos elementales de computación y de ambiente Web en sentido general. El sistema debe garantizar

un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios.

RNF 3 - Rendimiento.

El sistema deberá responder de forma rápida ante las solicitudes de los usuarios y en el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo cliente servidor y la velocidad en las consultas a la base de datos.

RNF 4 - Soporte.

Una vez finalizado el desarrollo de la aplicación Web, será probada en los servidores donde radicará de forma permanente para poder detectar y eliminar cualquier anomalía previa a la explotación por parte de los usuarios. Se deberá realizar el mantenimiento de la Base de Datos (BD) cada un período de 3 meses como máximo.

a) Software.

Para el servidor se requiere:

- Sistema Operativo Linux, Windows Server 2008 o superior.
- Servidor de Bases de Datos MySQL 10.1.37
- Servidor Web Apache 2.4.37
- Intérprete de script PHP 7.2.x

Para el cliente se requiere:

1. Sistema Operativo: Windows XP, Windows 7, Windows 10.
2. Navegador Web: Mozilla Firefox versión 50.0 o superior

b) Hardware (mínimo):

Para el servidor se requiere:

1. Microprocesador Corei3 de 4 Generación 2.5 GHz.
2. Memoria RAM de 4GB.
3. Disco duro de 80 GB.

Para el cliente se requiere al menos:

1. Microprocesador DualCore2.5GHz.

2. Memoria RAM de 1 GB.

RNF 5 - Portabilidad.

La aplicación estará diseñada para ser ejecutada sobre diversas plataformas (Windows, GNU/Linux) y a su vez soportada por los navegadores más populares usados: Mozilla Firefox, Internet Explorer, Opera y Google Chrome.

RNF 6 - Seguridad.

El sistema propuesto debe estar diseñado en correspondencia a las políticas de seguridad informática establecidas en el Ministerio de Salud Pública y en las normativas sobre seguridad informática aprobadas en el país. Entre las áreas que abarca dicha seguridad aparece el Control de acceso, Autenticación y Auditoría (AAA).

- **Control de acceso:** está dirigido a prevenir el acceso no autorizado, daño e interferencia a la información. En el sistema propuesto cada usuario accederá a la información en correspondencia con el rol que desempeña.
- **Autenticación:** establece que cada usuario deberá estar autenticado en el sistema para poder interactuar con la información disponible. Los usuarios deberán seguir buenas prácticas en la selección y uso de contraseñas.
- **Auditoría:** el sistema propuesto debe ser monitoreado para detectar desviaciones de las políticas de control de accesos.

La información será procesada en la capa del servidor y en funciones dentro de la base de datos.

- a) Confidencialidad:** La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- b) Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estado no consistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.
- c) Disponibilidad:** La disponibilidad del sistema debe estar bajo la modalidad de 24 horas al día y los 7 días de la semana.

RNF 7 - Legales.

La aplicación informática debe cumplir con las normas establecidas en el Reglamento para la Utilización de las TIC en las entidades conectadas a la red de salud pública del país.

RNF 8 - Confiabilidad.

El sistema debe recuperarse en el menor tiempo posible en caso de producirse una falla. Además, se debe pronosticar el mayor número de errores en el sistema y cómo solucionarlos.

RNF 9 - Ayuda.

El sistema contará con notificaciones mediante mensajes antes de la realización de cualquier acción crítica que involucre algún cambio o actualización de los datos.

Descripción de actores del sistema

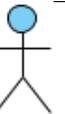
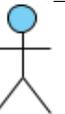
Actor del sistema	Descripción
 Profesor	Gestiona las notas de los estudiantes, gestiona reportes y visualiza gráficamente la información relacionada con la realización de las evaluaciones de los estudiantes.
 Secretaria docente	Gestiona datos del estudiante y obtiene información de los reportes de evaluación.
 Administrador del sistema	Gestiona los datos del usuario.

Tabla 2.1 Actores del sistema.

Descripción de bajo nivel de los Casos de Uso del sistema

Nombre del caso de uso del sistema	Autenticar
Actores	Profesor, Administrador del sistema, Secretaria docente
Propósito	Autenticar un usuario en el sistema.
El caso de uso inicia cuando el actor desea autenticarse en el sistema, para esto ingresa su usuario y contraseña, da clic izquierdo en el botón aceptar. Si es correcto abre su sesión, de lo contrario sale un cartel de error. El caso de uso finaliza cuando el actor se ha autenticado en el sistema.	
Referencia	RF 10

Tabla 2.2 Descripción de bajo nivel del caso de uso del sistema Autenticar.

Nombre del caso de uso del sistema	Gestionar Estudiantes.
Actores	Secretaria docente
Propósito	Registrar o modificarlos datos personales de un estudiante.
El caso de uso inicia cuando el actor Secretaria docente accede a Gestionar Estudiante en la interfaz principal, el sistema le permitirá registrar o modificar un estudiante. El caso de uso termina cuando el actor realiza una de las operaciones y esta queda almacenada.	
Referencia	RF 1

Tabla 2.3 Descripción de bajo nivel del caso de uso del sistema Gestionar Estudiantes.

Nombre del caso de	Gestionar notas
--------------------	-----------------

uso del sistema	
Actores	Profesor
Propósito	Llevar un control de las notas.
El caso de uso inicia cuando el actor profesor accede a Controlar evaluaciones, el sistema le permitirá insertar, modificar las notas del estudiante y emitir reportes. El caso de uso termina cuando el actor realiza una de las operaciones y esta queda registrada.	
Referencia	RF 2

Tabla 2.4 Descripción de bajo nivel del caso de uso del sistema Gestionar notas.

Nombre del caso de uso del sistema	Generar reportes
Actores	Secretaría docente UM
Propósito	Generar los reportes de las notas.
El caso de uso inicia cuando el actor secretaria docente desea generar algún reporte con respecto a las evaluaciones. El actor selecciona en la interfaz principal la opción Generar reportes, es mostrado un listado con los diferentes tipos de reportes, donde puede seleccionar uno de estos y gestionar la información solicitada. El caso de uso finaliza cuando se muestran los datos de dicho reporte.	
Referencia	RF 4, RF 5, RF 6, RF 7

Tabla 2.5 Descripción de bajo nivel del caso de uso del sistema Generar reportes.

Nombre del caso de uso del sistema	Administrar usuarios del sistema
Actores	Administrador del sistema
Propósito	Administrar los usuarios en el sistema y

	gestionarcodificadores.
El caso de uso inicia cuando el actor administrador del sistema desea gestionar la información referente a los usuarios. El actor selecciona en la interfaz principal la opción Administrar usuarios del sistema, donde puede agregar, modificar o eliminar un usuario, darles permisos para la gestión dentro de la aplicación web.El caso de uso finaliza cuando se ejecuta una de estas acciones.	
Referencia	RF 9

Tabla 2.6 Descripción de bajo nivel del caso de uso del sistema Administrar usuarios del sistema.

Diagrama de caso de uso estructurado.

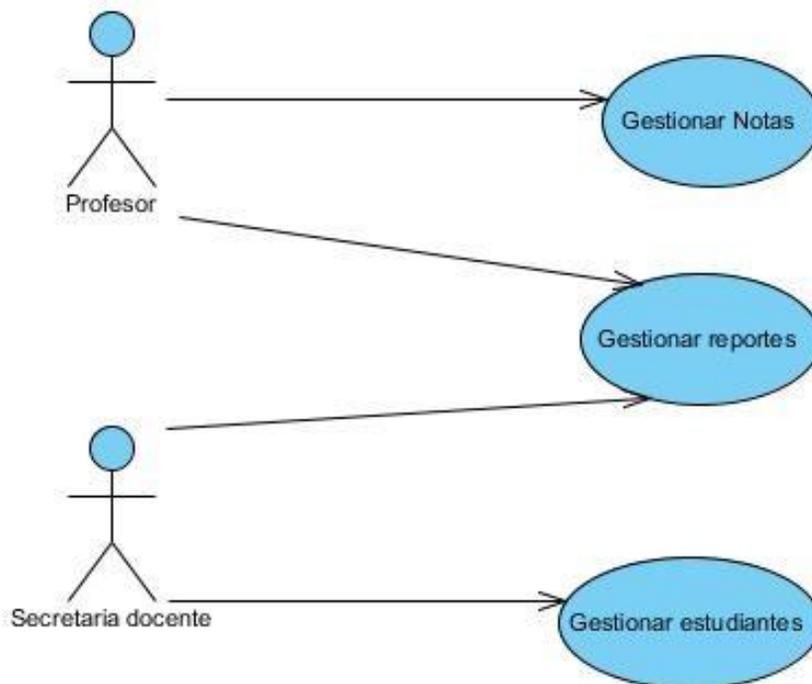


Figura 3.1 Diagrama de casos de uso del sistema estructurado

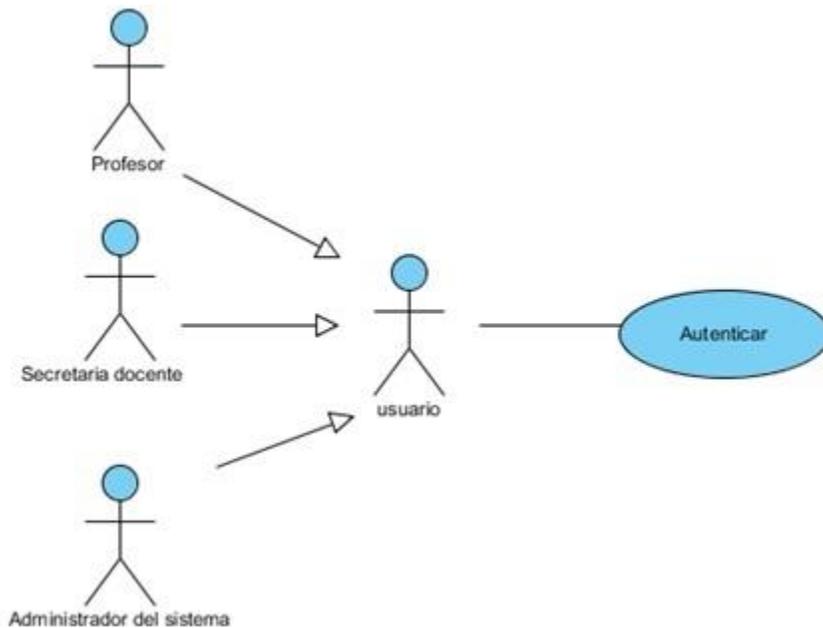


Figura 3.2 Diagrama de casos de uso para el actor usuario.

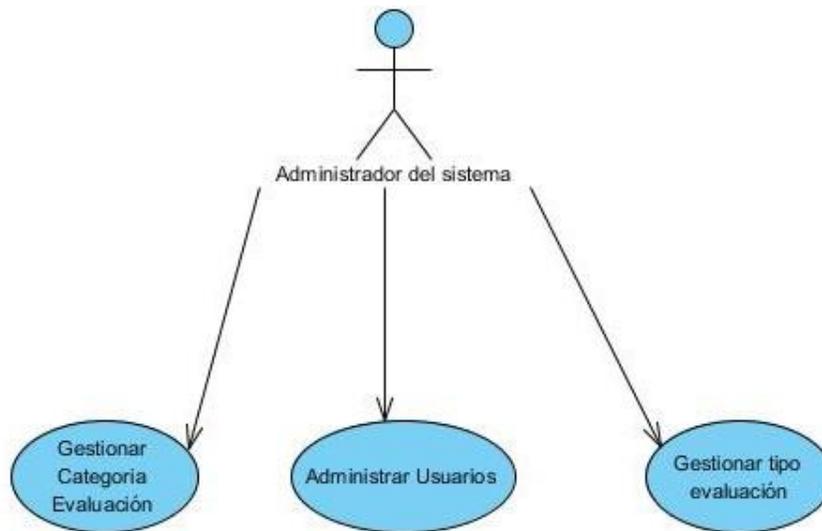


Figura 3.3 Diagrama de casos de uso para el administrador del sistema.

Vista de arquitectura de los casos de uso.

1. Controlar evaluaciones.

Descripción de alto nivel de los casos de uso del sistema.

Caso de uso del sistema	
1	Controlar evaluaciones.
Propósito	Controlar las evaluaciones de los estudiantes.
Actor: Profesor	
Resumen: El caso de uso inicia cuando el actor profesor accede a Controlar evaluaciones en la interfaz principal, busca y selecciona al estudiante y va a la opción evaluaciones, el sistema le permitirá insertar y modificar las notas del estudiante. El caso de uso termina cuando el actor realiza una de las operaciones e inserta la nota y esta queda registrada en el sistema	
CU asociados	-
Acción del actor	Respuesta del sistema
1.El actor da clic en la opción estudiantes en la interfaz principal.	2.El sistema despliega un listado con los nombres de los estudiantes.
3.El actor selecciona al estudiante a evaluar.	4.El sistema muestra los tipos de evaluaciones(Notas de clases, Notas de práctica, Notas de seminarios, Notas de educación en el trabajo, Notas de propedeútica, Notas finales.)
5.El actor selecciona el tipo de	

<p>9 Selecciona la opción Modificar nota.</p> <p>11.Modifica las notas y da clic en el botón aceptar.</p>	<p>10 Muestra un formulario con los datos a modificar .</p> <p>12. El sistema valida los datos ingresados.</p> <p>12.1 Si los datos no están correctos el sistema muestra un mensaje informándolo y no realiza la operación.</p> <p>12.2 Si los datos están correctos el sistema los registra y muestra un mensaje indicando que fue satisfactorio</p>
<p>Referencias</p>	<p>RF 1, RF 2, RF 3, RF 4</p>

Tabla 2.7. Descripción de alto nivel del caso de uso del sistema Controlar evaluaciones

2.2 Análisis del sistema.

El objetivo del flujo Análisis del Sistema es obtener una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero incluyendo su arquitectura. El análisis consiste en obtener una visión del sistema, de modo que sólo se interesa por los requisitos funcionales.

Modelo de Análisis.

El propósito principal del modelo de análisis es examinar los requerimientos definidos en la captura de requisitos, mediante su refinamiento y estructuración. Además, proporciona una estructura centrada en el mantenimiento. En aspectos tales como flexibilidad ante cambios y la reutilización puede considerarse una primera aproximación al modelo de diseño.

Una aplicación cliente/servidor de tres capas, se centra en separar la interfaz gráfica del resto del código. La vista se encarga de presentar la interfaz al usuario; en sistemas Web, esto es típicamente HTML, aunque pueden existir otro tipo de vistas. En la capa intermedia están las clases de control ya que en ella se agrupan los servicios que pueden ser compartidos por múltiples aplicaciones. La capa modelo incorpora la capa de dominio y persistencia, es la encargada de guardar los datos en un medio persistente.

Los diagramas de colaboración no son más que la representación de la interacción de los objetos a partir del envío de sus mensajes y diagramas de clases, dichas clases se pueden clasificar en:

- **Interfaz:** modela la interacción entre el sistema y sus actores.
- **Controladora:** realiza la coordinación, secuenciado de transacciones y, en definitiva, el control sobre otros objetos del sistema.
- **Entidad:** modela información que persiste en el tiempo o tiene una larga vida.

Diagrama de colaboración del caso de uso del sistema: Controlar evaluaciones.

Sesión # 1

Insertar notas.

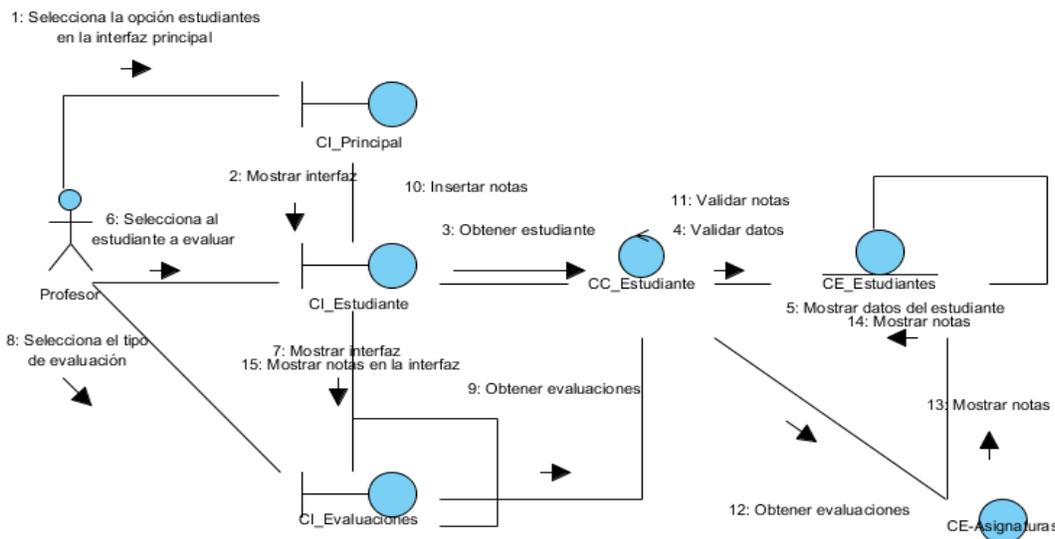


Figura 3.3 Diagrama de colaboración de clases. Sección insertar notas.

Diagrama de colaboración del caso de uso del sistema: Controlar evaluaciones.
Sesión # 2 Modificar notas.

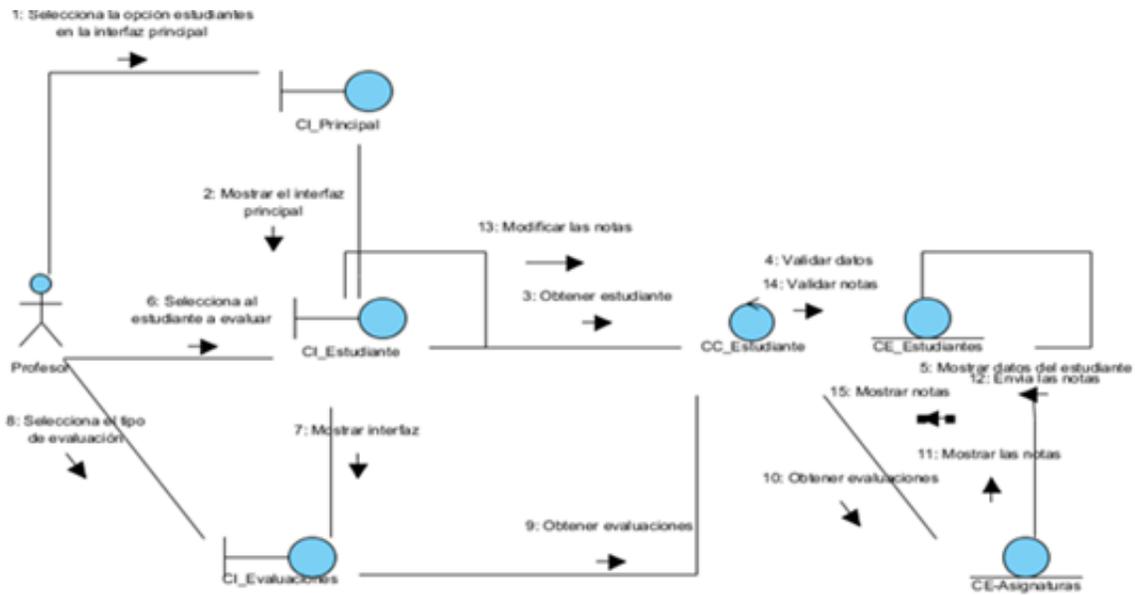


Figura 3.4 Diagrama de colaboración de clases. Sección modificar notas.

2.3 Diseño del sistema.

El modelo de diseño es un modelo de objetos en el cual se describen la realización física de los casos de usos, centrándose tanto en los requisitos funcionales como en los no funcionales junto con otras restricciones de implementación, teniendo impacto en el sistema a considerar.

Patrón de diseño

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí, los cuales deben ser adaptados para resolver problemas de diseño general en un particular contexto. Ellos identifican clases, instancias, roles, colaboraciones y la distribución de responsabilidades (Gamma, Johnson&Vlissides, 2010).

Se clasifican en patrones de creación, estructural y de comportamiento. Entre los patrones de diseño básicos podemos encontrar los patrones Bajo Acoplamiento, Creador, Experto, Alta Cohesión y Controlador. Estos, se identifican como

patrones GRASP, acrónimo en inglés de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Su nombre se eligió para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos. Ellos tienen como objetivo describir los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Durante la realización de la aplicación, se utilizaron mucho estos patrones. El patrón Experto fue empleado al asignar responsabilidades de realizar determinada labor de las clases creadas que contenían datos involucrados (atributos).

El patrón Creador fue implementado como guía en la asignación de responsabilidades relacionadas con la creación de los objetos de las clases creadas. Un ejemplo es cuando se desea agregar una nueva evaluación a un estudiante, o una nueva nota.

Se utilizó el patrón de Bajo Acoplamiento moderadamente para que las clases no dependieran de muchas otras, de forma que los cambios que surjan no ocasionen cambios locales y las clases puedan ser reutilizables. En este caso, se tienen por ejemplo tres clases, Objetivo Educativo, Profesor y Estudiante. El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión, sin embargo, es un factor que se consideró cuando se realizó el diseño ya que en un patrón Bajo Acoplamiento debe haber pocas dependencias entre las clases. Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia.

Este diseño también garantiza la aplicación del patrón de Alta Cohesión ya que da soporte a una alta cohesión y un bajo acoplamiento. Esto trae como beneficio la claridad y facilidad con que se entiende el diseño, simplificando el mantenimiento y las mejoras de funcionalidad. El patrón Controlador se empleó para asignar la responsabilidad de controlar el flujo de eventos del sistema en clases específicas.

Los patrones arquitectónicos son estándares de diseño de software que ofrecen soluciones a problemas de arquitectura de software; dan una descripción de los

elementos, y el tipo de relación que tienen, junto a un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural, esencial para un sistema de software que consta de subsistemas, responsabilidades e interrelaciones.

Uno de los patrones arquitectónicos más utilizados actualmente es el Modelo Vista Controlador (MVC), este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

El patrón arquitectónico de llamada y retorno MVC se ve frecuentemente en aplicaciones web, donde el modelo se encarga de la representación específica de toda la información con la cual el sistema va a trabajar, en este la lógica de datos puede llegar a asegurar la integridad y permitirá derivar nuevos datos; la vista o interfaz, presenta el modelo con el que el usuario interactuará, y el controlador responde a eventos o acciones que el usuario invoca, implicando cambios en el modelo y en la vista.

Para la implementación del sistema informático propuesto, se optó por el patrón arquitectónico MVC, debido a que éste es usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas, donde se requiere una mejor separación de los conceptos, para que el desarrollo esté mejor estructurado. El MVC, facilita la programación en diferentes capas de forma paralela e independiente, ya que al existir la separación de vistas, controladores y modelos es más sencillo realizar labores de mejoras como: agregar nuevas vistas, dar mantenimiento y realizar modificaciones sin necesidad de provocar que todo el sistema se paralice.

Un ejemplo sencillo de cómo se pone de manifiesto el uso del patrón arquitectónico Modelo Vista Controlador en el sistema propuesto es cuando el actor Profesor desea observar el listado de los estudiantes que faltan por demostrar habilidades. A través de la clase estudiante selecciona la opción mostrar listado de los estudiantes que faltan por demostrar habilidades, la vista le envía la información solicitada a la clase habilidades el cual le envía la petición del

listado de habilidades, este procesa la solicitud de la información y gestiona la misma devolviéndole a la clase habilidades el listado de los estudiantes que faltan por demostrar habilidades.

Descripción de la arquitectura del software

La arquitectura del software es quizás el artefacto más importante que puede emplearse para desarrollar un sistema informático, no tiene que ver solamente con la estructura y el comportamiento del sistema, sino también con el uso, la funcionalidad, el rendimiento, la capacidad de adaptación y comprensión, la reutilización, las restricciones económicas y de tecnología, los compromisos entre alternativas y los aspectos estéticos.

La arquitectura de software surge como un conjunto de patrones y abstracciones coherentes, que proporcionan un marco de referencia necesario para guiar la construcción de un sistema informático; establece la estructura del sistema, la cual comprende elementos del software, las propiedades externamente visibles de esos elementos y las relaciones entre ellos. Entre las arquitecturas de software más universales se encuentran:

Cliente – Servidor

Según IBM (International Business Machines) es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo, a través de la organización, en múltiples plataformas. El modelo soporta un ambiente distribuido en el cual los requerimientos de servicio, hechos por estaciones de trabajo o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores.

La arquitectura cliente - servidor puede ser clasificada en dos capas, tres capas o n capas; la separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

- **Arquitectura de tres niveles:** especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones; una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

A continuación, se muestran algunas ventajas de la arquitectura cliente-servidor:

- **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Centralización del control:** los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- **Existen tecnologías:** suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz y la facilidad de empleo.
- **Fácil mantenimiento:** Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible realizar actualizaciones o reemplazos de tecnologías donde el cliente no se ve afectado por ese cambio (o se afectarán mínimamente).

Para el desarrollo del sistema propuesto se empleó la arquitectura de tres capas; donde todos los elementos de la aplicación se organizan en componentes independientes, alcanzando de una forma u otra mayor efectividad en el momento de administrarlos, teniendo en cuenta fundamentos de la programación en múltiples capas, por la facilidad de administrar de forma eficiente los componentes que la integran, y la rapidez en las funcionalidades cliente/servidor. Estas capas son las siguientes:

- **Capa de presentación:** es la que ve el usuario, le comunica la información y captura los datos que se han introducido, dando un mínimo de proceso

(realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

- **Capa de negocio:** en esta capa se encuentran los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Aquí se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.
- **Capa de datos:** aquí es donde residen los datos y es la encargada de acceder a ellos. Está formada por una base de datos que realiza todo el almacenamiento de datos y recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Modelo de datos

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido y recuperar la información. Un modelo de datos es aquel que describe de una forma abstracta cómo se representan los datos, sea en una empresa, en un sistema de información o en un sistema de gestión de base de datos. Básicamente consiste en una descripción de algo conocido como contenedor de datos (algo en donde se guarda la información).

El modelado de datos es el proceso de ordenar los datos y sus relaciones con el fin de desarrollar el modelo lógico de la base de datos. Tiene como objetivo conseguir estructuras de datos flexibles, estables y normalizados y separar el proceso de los datos.

En la siguiente figura se representa el modelo de datos de este sistema propuesto.

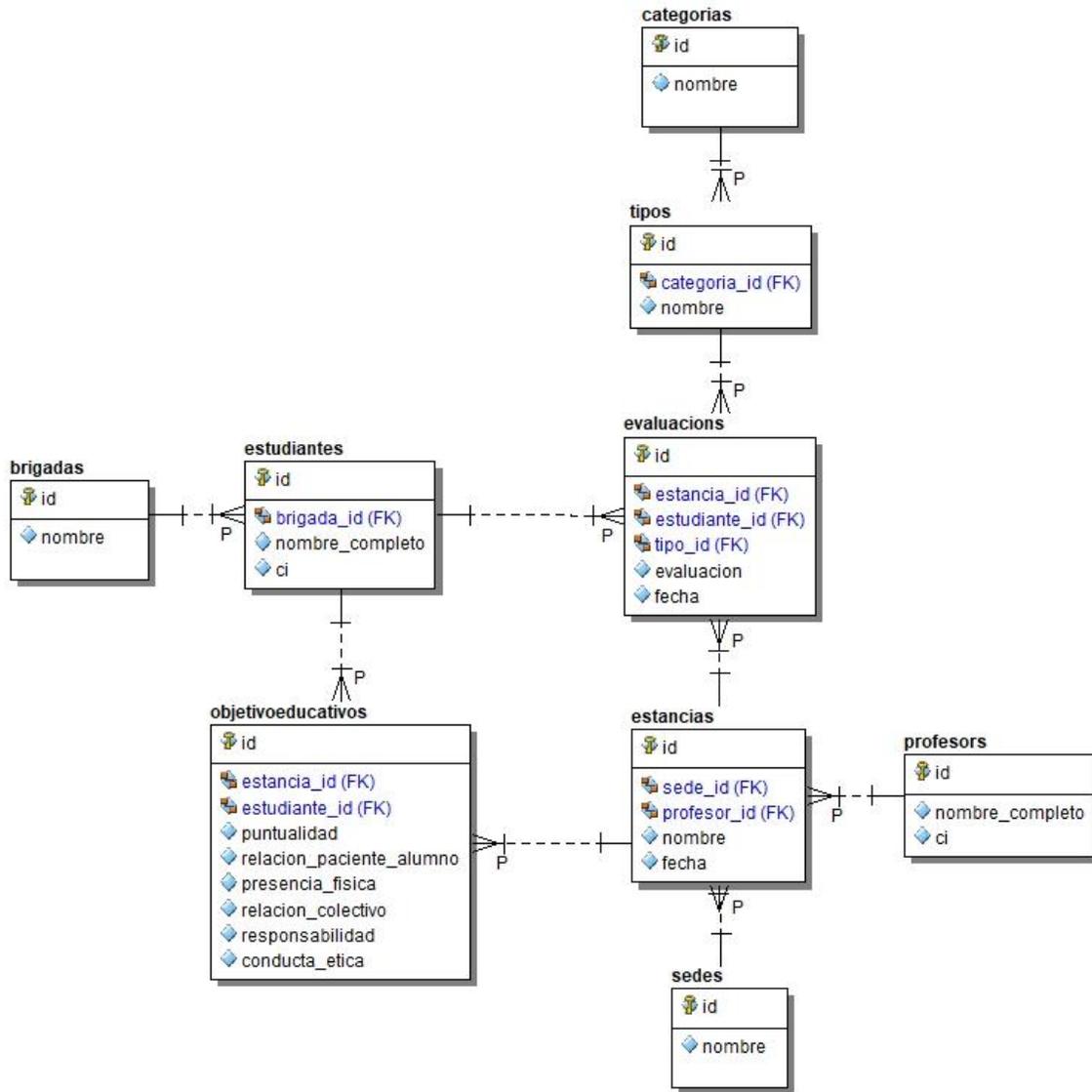


Figura 3.5 Modelo de datos

2.4 Implementación

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

Diagrama de despliegue

El diagrama de despliegue de este sistema está compuesto por tres nodos: la estación de trabajo del cliente, un servidor web y un servidor de base de datos. El propósito de la PC del cliente es interactuar con la aplicación mediante un navegador web, el cual se conectará mediante protocolo HTTPS al nodo servidor web. El servidor web contendrá la aplicación web para presentar los servicios del sistema y a su vez accederá al servidor de base de datos, mediante el protocolo TCP/IP; siendo este tercer nodo el encargado de soportar y administrar toda la infraestructura de la base de datos de la aplicación.

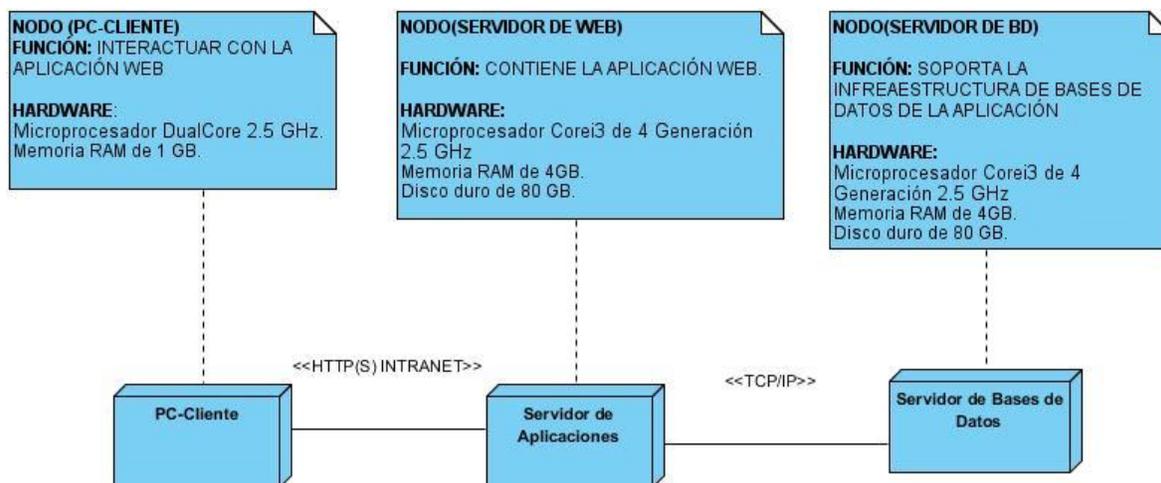


Figura 3.6Diagrama de despliegue

2.5 Diseño de casos de prueba

Un caso de prueba define una forma de probar el sistema. Es una forma de organizar las pruebas del sistema relacionadas con la funcionalidad que tenga, es por eso que los casos de prueba están directamente relacionados con los casos de uso que prueban.

Prueba de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software; los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se

produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Caso de uso del sistema	Autenticar Usuario
Caso de prueba	Comprobar que no se acceda al sistema con un usuario y/o contraseña incorrecta.
Descripción	Se intentará acceder al sistema con un usuario no válido o una contraseña incorrecta.
Descripción	Usuario: admin, Contraseña: *****
Estado Inicial.	
Condiciones de Ejecución	El usuario admin no debe estar registrado en el sistema

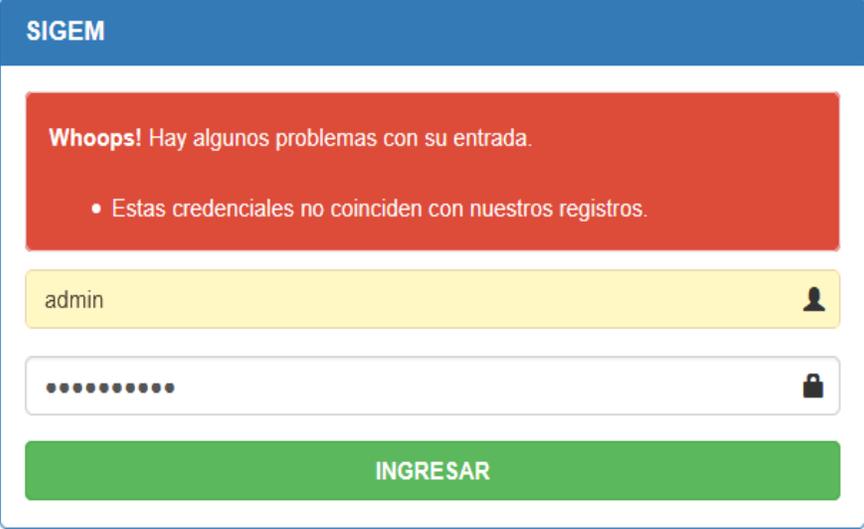
Resultados esperados	
-----------------------------	--

Tabla 2.8 Diseño de prueba de caja negra del caso de prueba: Comprobar que no se acceda al sistema con un usuario y/o una contraseña incorrecta o inexistente.

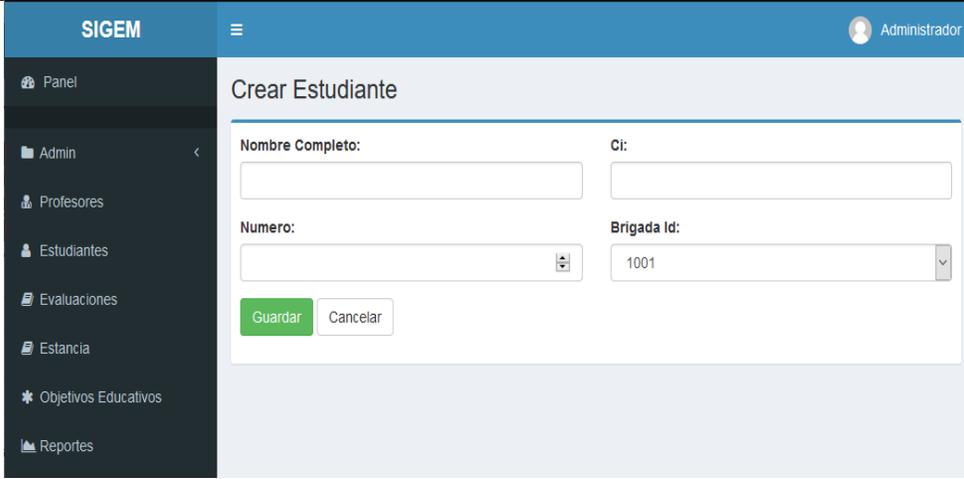
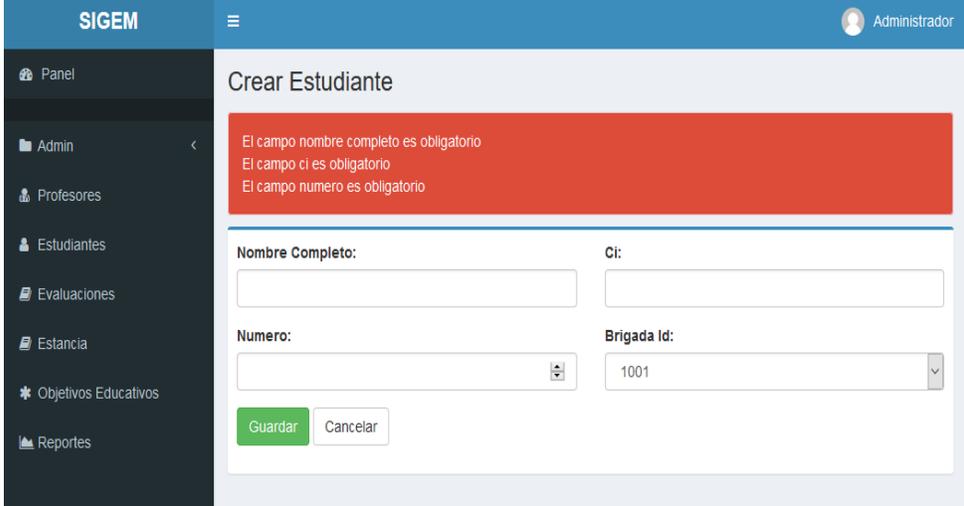
Caso de uso del sistema	Insertar Estudiante
Caso de prueba	Comprobar que no se introduzca un estudiante sin completar todos los datos referentes a él.
Imagen	
Descripción	Se intentará guardar los datos de un estudiante con los campos nombre, ci y numero vacíos. El sistema debe mostrar una alerta indicándolos como campos obligatorios.
Condiciones de Ejecución	Los campos nombre, ci, brigada y número no pueden estar vacíos.
Resultados esperados	

Tabla 2.9 Diseño de prueba de caja negra del caso de prueba: Comprobar que no se introduzca un estudiante sin llenar todos los datos.

2.6 Estudio de factibilidad

Después de determinar los casos de uso que guiarán el desarrollo del software, se puede pronosticar una estimación del tiempo de duración del proyecto mediante el análisis de Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Estimación basada en el método de puntos en casos de uso.

Paso 1. Calcular los Puntos de Casos de Uso sin ajustar (PCU).

El cálculo de Puntos de Casos de Uso sin ajustar se calcula mediante la siguiente ecuación:

$$\text{PCU} = \text{FPA} + \text{FPCU}$$

Donde:

- PCU: Puntos de Casos de Uso sin ajustar.
- FPA: Factor de Peso de los Actores sin ajustar.
- FPCU: Factor de Peso de los Casos de Uso sin ajustar.

El valor del Factor de Peso de los Actores (FPA) sin ajustar se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos.

Actores del sistema	Complejidad	Peso
Administrador del sistema	Complejo	3
Profesor	Complejo	3
Secretaria docente	Complejo	3
Total		9

Tabla 2.10 Cálculo del Factor de Peso de los Actores.

El valor del Factor de Peso de los Casos de Uso (FPCU) sin ajustar se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción es una secuencia de actividades atómicas, es decir, se efectúa la secuencia de actividades completa.

Casos de Uso	Complejidad	Transacciones	Peso
Gestionar notas	Medio	4	10
Generar reporte	Simple	3	5
Gestionar estudiantes	Simple	2	5
Administrar usuarios	Medio	4	10
Autenticar	Simple	1	5
Visualizar gráfica	Simple	2	5
Total			40

Tabla 2.11: Cálculo del Factor de Peso de los Casos de Uso.

Los Puntos de Casos de Uso sin ajustar son:

$$PCU = FPA + FPCU$$

$$= 9 + 40$$

$$= 49$$

Paso 2. Cálculo de los Puntos de Casos de Uso Ajustados (PCUA).

Con el resultado obtenido del cálculo de Puntos de Casos de Uso sin ajustar se procede a ajustar este valor mediante la ecuación $PCUA = PCU * FCT * FA$ donde:

- PCUA: Puntos de Casos de Uso ajustados
- PCU: Puntos de Casos de Uso sin ajustar
- FCT: Factor de complejidad técnica
- FA: Factor de ambiente

El Factor de Complejidad Técnica (FCT) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Criterio	Peso	Valor
T1	Sistema distribuido	El sistema es centralizado.	2	0
T2	Objetivos de performance o tiempo de respuesta	El sistema responde rápidamente a las solicitudes de los usuarios.	1	5
T3	Eficiencia del usuario final	Los usuarios finales deben poseer conocimiento en el trabajo con aplicaciones web.	1	3
T4	Procesamiento interno complejo	El procesamiento interno del sistema tiene complejidad media.	1	3
T5	El código debe ser reutilizable	El código del sistema es completamente reutilizable.	1	4
T6	Facilidad de instalación	El sistema es fácil de instalar.	0.5	5
T7	Facilidad de uso	El sistema es fácil de usar.	0.5	5
T8	Portabilidad	El sistema es multiplataforma.	2	5
T9	Facilidad de cambio	El sistema puede ser cambiado con facilidad.	1	3
T10	Concurrencia	Pueden existir varios usuarios trabajando en el sistema al mismo tiempo.	1	5
T11	Incluye objetivos especiales de seguridad	El sistema establece buenos mecanismos de seguridad.	1	5
T12	Provee acceso directo a terceras partes	No provee.	1	0

T13	Se requieren facilidades especiales de entrenamiento a usuarios	Se proporciona ayuda y capacitación a los usuarios sobre cómo funciona el sistema.	1	4
-----	---	--	---	---

Tabla 2.12: Cálculo de Factor de Complejidad Técnica.

$$\begin{aligned}
 FCT &= 0,6 + 0,01 * \sum (\text{Peso } i \times \text{Valor } i) \\
 &= 0,6 + 0,01 * (2*0+1*5+1*3+1*3+1*4+0,5*5+0,5*5+2*5+1*3+1*5+1*5+1*0+1*4) \\
 &= 0,6 + 0,01 * 47 \\
 &= 1,07
 \end{aligned}$$

El Factor de Ambiente (FA) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor
E1	Familiaridad con el modelo de proyecto utilizado.	1,5	4
E2	Experiencia en la aplicación.	0,5	5
E3	Experiencia en orientación a objetos.	1	5
E4	Capacidad del analista líder.	0,5	5
E5	Motivación.	1	5
E6	Estabilidad de los requerimientos.	2	2
E7	Trabajadores a tiempo parciales.	-1	0
E8	Dificultad del lenguaje de programación.	-1	4

Tabla 2.13: Cálculo de Factor Ambiente.

Luego de haber obtenido estos valores, se procede a calcular el valor de FA mediante la siguiente fórmula:

$$\begin{aligned}
 FA &= 1,4 - 0,03 * \sum (\text{Peso } i \times \text{Valor } i) \\
 &= 1,4 - 0,03 * (1,5*4+0,5*5+1*5+0,5*5+1*5+2*2-1*0-1*4) \\
 &= 1,4 - 0,03 * 21 \\
 &= 1,4 - 0,585
 \end{aligned}$$

=0,77

Una vez obtenidos los valores de PCU, FCT y FA ya es posible calcular el valor de PCUA:

$$\begin{aligned} \text{PCUA} &= \text{PCU} * \text{FCT} * \text{FA} \\ &= 49 * 1,07 * 0,77 \\ &= 40,3711 \end{aligned}$$

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso (E).

El esfuerzo en horas-hombre viene dado por:

$$\mathbf{E = PCUA * FC}$$

Donde:

- E: Esfuerzo estimado en horas hombres.
- PCUA: Punto de casos de usos ajustados.
- FC: Factor de conversión. Se utiliza el factor de conversión 20 horas-hombre/PCU, es decir, un PCU toma 20 horas-hombre.
- $E = \text{PCUA} * \text{FC}$
 $= 40,3711 * 20$
 $= 807,422$

Paso 4. Cálculo del esfuerzo de todo el proyecto E (Total).

Este esfuerzo es el que se requiere para la implementación. Si se tiene en cuenta que este representa un 40 % del esfuerzo total para desarrollar el software, entonces tenemos que el esfuerzo total es el siguiente:

$$\begin{aligned} \text{E (Total)} &= E / 0,4 \\ &= 807,422 / 0,4 \\ &= 2018,555 \end{aligned}$$

Actividad	Porcentaje (%)	Valor
Análisis	10	201,8555
Diseño	20	403,711
Implementación	40	807,422
Pruebas	15	302,78325
Sobrecarga (otras actividades)	15	302,78325
Total		2018,555

Tabla 2.14 Esfuerzo en horas – hombre.

Paso 4. Estimación del tiempo de desarrollo del proyecto (TDES):

E (Total): Esfuerzo total.

- TDES: Tiempo de Desarrollo.
- CH: Cantidad de Hombres que desarrollan el proyecto

Por lo tanto, para un desarrollador, la estimación del tiempo de desarrollo del proyecto es:

$$\begin{aligned} \text{TDES} &= E \text{ (total en HH)} / \text{CH (total)} \\ &= 2018,555\text{Horas/Hombre} / 1 \text{ Hombre} \\ &= 2018,555\text{Horas/Hombre} \end{aligned}$$

Paso 5. Estimación del costo de desarrollo del proyecto C (total).

Una vez estimado el tiempo de desarrollo del proyecto y conociendo la cantidad de desarrolladores y la retribución que recibe cada uno de estos se puede llevar a cabo una estimación del costo total del proyecto, referidos a los recursos humanos.

- CHH: Costo por Hombre Horas.
- K: Coeficiente que tiene en cuenta los costos indirectos
- THP: Tarifa Horaria Promedio
- SP: El salario promedio del desarrollador de este sistema es:

$$\begin{aligned} \text{THP} &= \text{SP} / 160 \\ &= 100 / 160 \\ &= 0.625 \end{aligned}$$

$$\text{CHH} = \text{K} * \text{THP}$$

$$= 2 * 0,625$$

$$= 1,25$$

$$C \text{ (Total)} = E \text{ (total en HH)} * CHH$$

$$= 2018,555 * 1,25$$

$$= 2523,19375$$

Análisis de factibilidad

El estudio de factibilidad es un instrumento que sirve para orientar la toma de decisiones en la evaluación de un proyecto, en el cual se realiza una exploración del costo-beneficio del proyecto y se exponen los beneficios tangibles e intangibles y un análisis de los riesgos del proyecto. Se formula con base en información que tiene la menor incertidumbre posible para medir las posibilidades de éxito o fracaso de un proyecto de inversión.

Beneficios Tangibles

- Ahorro en papel.
- Ahorro de lápices.
- Ahorro de bolígrafos.
- Ahorro de gomas.
- Ahorro de horas- hombre.
- Ahorro en tóner de impresora.
- Ahorro en cinta de impresora.

Beneficios Intangibles

- Mejora de la calidad de la información, será más confiable.
- Agilidad en la transferencia de la información.
- Mejoras en cuanto al control y seguridad de la información.
- Asegura la continua participación y colaboración de todo el personal en el proceso.
- Disponibilidad de mecanismos para una mejor gestión y optimización de procesos.
- Mejora las condiciones de trabajo del personal.

Costos Tangibles

- Gastos de papel.
- Gasto de lápices.
- Gasto de bolígrafos.
- Gasto de gomas.
- Gastos de tonel de impresora.
- Gastos en cinta de impresora.
- Depreciación de la computadora y la impresora.

Costos Intangibles

- Posible resistencia al cambio.
- Cambios en la forma de trabajo.

En la siguiente tabla se cuantifica el ahorro de dinero que se obtendrá una vez implementado el sistema en período de un año:

Beneficio	Gasto Anual Antes (A)	Gasto Anual Después (B)	Precio Unitario (C)	Ahorro (A-B)*C
Ahorro en papel.	7	3	\$80,00	\$320,00
Ahorro de lápices.	16	4	\$50,00	\$600,00
Ahorro de bolígrafos.	10	3	\$50,00	\$350,00
Ahorro de gomas.	20	6	\$20,00	\$280,00
Ahorro de horas-hombre.	100	30	\$60,00	\$4200,00
Ahorro en tóner de impresora.	8	2	\$100,00	\$600,00
Ahorro en cinta de impresora.	15	4	\$150,00	\$1650,00
Total				\$8000,00

Tabla 2.15 Cuantificación en dinero de todos los beneficios tangibles.

Punto de equilibrio

El punto de equilibrio es una técnica analítica para estudiar las relaciones existentes entre los costos fijos, variables y las ventas o cualquier otro aspecto que haga variar los costos; la cual se vale de una variable discreta que permite

ajustar los costos bajo condiciones específicas; las que permiten a su vez determinar en qué punto, o bajo qué condiciones el proyecto propuesto cuesta menos que la forma en que se desarrollan las actividades actualmente. Para tener una visión general de los gastos actuales y futuros en caso del establecimiento del sistema se describen a continuación un antes y un después del mismo.

Proyección de beneficios y costos del proyecto.

Recursos	Costo actual	Costo con el sistema propuesto
Ahorro en papel.	\$774,40	\$387,20
Ahorro de lápices.	\$50,00	\$20,00
Ahorro de bolígrafos.	\$80,00	\$30,00
Ahorro de gomas.	\$100,00	\$44,00
Ahorro de horas- hombre.	\$2792,60	\$496.34
Ahorro en tóner de impresora.	\$1475,00	\$590,00
Ahorro en cinta de impresora.	\$306,00	\$208,00
Total	\$5578,00	\$1775,54
Costo del sistema	-	\$2523.19375

Tabla 2.16 Relación de costos en los que se incurren antes y después del sistema.

Seguidamente se define la variable discreta Cantidad de Servicio de Terapia Intermedia, al ser aplicada al estudio expone los costos en la forma actual y con proyecto, aclarando que son cifras que representan gastos en el transcurso de un año.

Cantidad de Servicios	Costo Actual	Costo con el sistema propuesto
1	\$5578,00	\$4298,73375
2	\$11156,00	\$8597,4675
3	\$16374,00	\$12896,2012
4	\$22312,00	\$17194,935
5	\$27890,00	\$21493,6688

Tabla 2.17 Relación de costos con respecto a la cantidad de servicios.

Una vez analizados los costos y los datos de la tabla anterior, es posible determinar el punto de equilibrio, el cual muestra que si se establece el proyecto en al menos un Servicio de Terapia Intermedia, los costos del proyecto serán menores que los costos en la forma actual, como se muestra a continuación:

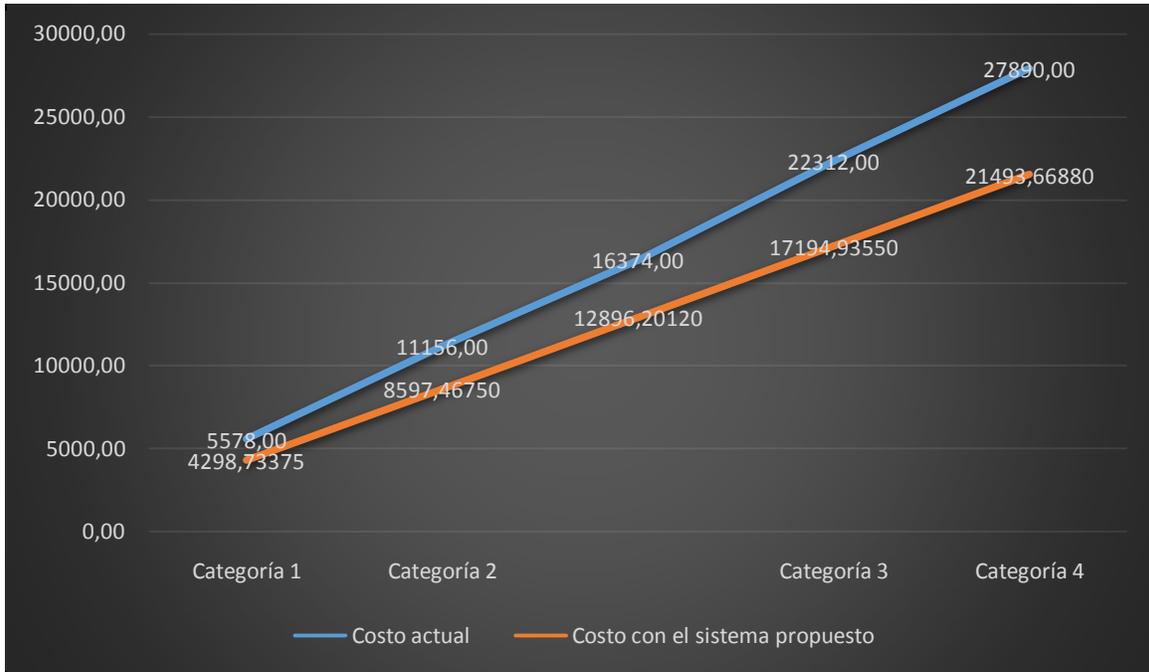


Figura 2.6: Punto de Equilibrio.

Retorno de la Inversión

Es de gran importancia el análisis de la inversión y su recuperación, a partir de la disminución de los costos y el aumento de los beneficios tangibles en el transcurrir del tiempo. Se toma una tasa de interés bancario al 8%.

Año	Costos	Costos acumulados	Beneficios	Beneficios acumulados	Flujo efectivo	Flujo acumulado
0	\$7523,19735	\$7523,19375	0	0	\$-7523,19375	\$-7523,19375
1	\$1775,54	\$9298,73375	\$8000,00	\$8000,00	\$6224,46	\$-1298,7338
2	\$1775,54	\$11074,2738	\$8000,00	\$16000,00	\$6224,46	\$4925,7262
3	\$1775,54	\$12849,8138	\$8000,00	\$24000,00	\$6224,46	\$11150,1862
4	\$1775,54	\$14625,3538	\$8000,00	\$32000,00	\$6224,46	\$17374,6462

5	\$1775,54	\$16400,8938	\$8000,00	\$40000,00	\$6224,46	\$23599,1062
---	-----------	--------------	-----------	------------	-----------	--------------

Tabla 2.18 Retorno de la inversión.

VAN (8%)= 23599,1062

TIR (8%)= 91%

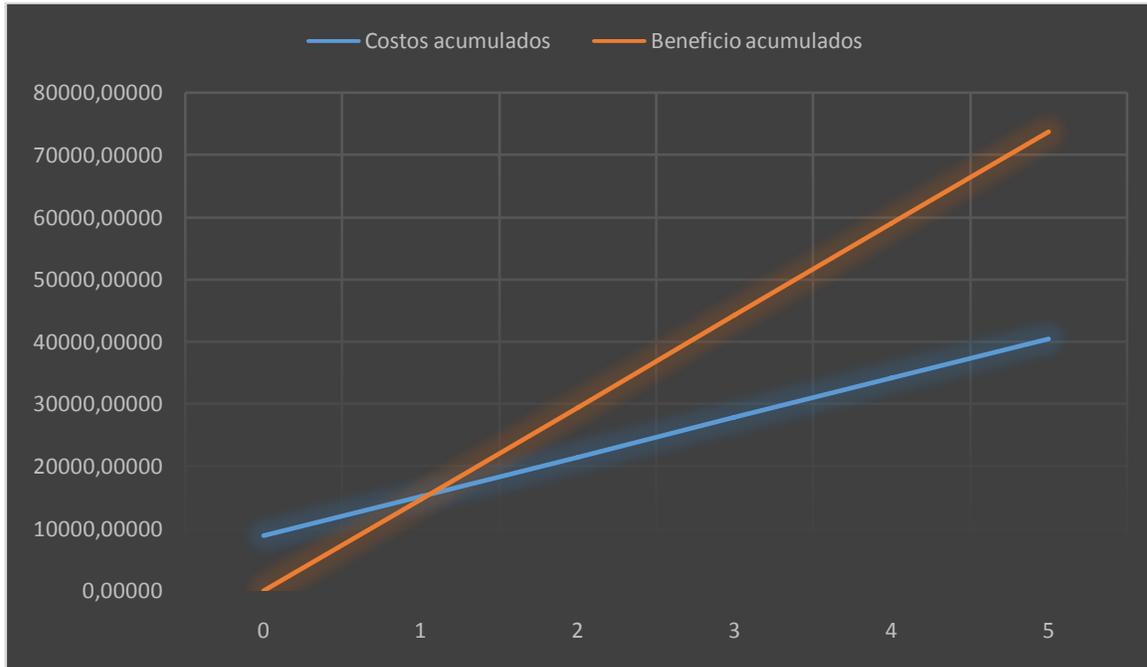


Figura2.7: Retorno de la inversión

Calculamos el Punto de Retorno de la Inversión (PRI) en años: este es el tiempo en que se igualan los costos y los beneficios acumulados; y es entonces, cuando se recupera la inversión y se comienza a obtener ganancias.

$$\begin{aligned}
 \text{PRI} &= N - 1 + \text{ABS} (\text{FAN} - 1 / \text{FN}) \\
 &= 2 - 1 + \text{ABS} (-1298,7338 / 6224,46) \\
 &= 1 + 0,20865 \\
 &= 1,20865
 \end{aligned}$$

= 1 Año.

El VAN (Valor Actual Neto) es mayor que cero y el TIR (Tasa Interna de Retorno) es mayor que el 8% por tanto, estamos en condición de decir que el sistema es factible.

Conclusiones del capítulo

- Se realizó la captura de los requisitos tanto funcionales como no funcionales, se identificaron los actores y casos de uso del sistema.
- Se realizaron los flujos de trabajo de análisis y diseño quedando como resultado de estos los diagramas de colaboración y despliegue.
- Se realizó el proceso de modelado del esquema de base de datos. Una vez realizado el análisis de factibilidad económica, quedó demostrado la viabilidad del desarrollo del sistema informático.

CONCLUSIONES GENERALES

El desarrollo de esta investigación permitió arribar a las siguientes conclusiones:

1. La caracterización del proceso docente educativo de estancia médica en el HGD Dr. Agostinho Neto y su gestión específicamente en el servicio de terapia intermedia reflejó que el mismo presenta insuficiencias. Dada su complejidad e importancia se propone mejorarlo a través de su informatización.
2. Se desarrolló el Sistema Informático para la Gestión del Proceso Docente Educativo de Estancia Médica en el servicio terapia intermedia del Hospital General Docente de Guantánamo que contribuirá a disminuir las insuficiencias en la gestión de dicho proceso, permitiendo además la búsqueda de información con mayor rapidez y confiabilidad.

RECOMENDACIONES

Luego de realizar la investigación y teniendo en cuenta los resultados obtenidos, se recomienda:

1. Generalizar el uso del sistema referido en todos los servicios del HGD Dr. Agostinho Neto.
2. Continuar incorporando nuevas funcionalidades al sistema según la demanda del cliente y los usuarios.

BIBLIOGRAFÍA

1. Alvarez, M. A. (2019). Modelo-Vista-Controlador en CodeIgniter. Recuperado de: <http://www.desarrolloweb.com/articulos/modelo-vista-controlador-codeigniter.html>. [Consultado el: 6 de febrero de 2018].
2. Autores, C. d. (2005), Manual de HTML.
3. Autores, C. d. (2006), Manual de MySQL.
4. Autores, C. d, (2017), CodeIgniter User Guide. British Columbia Institute of Technology.
5. Avellà, M.S., Javascript y jQuery. 16.
6. Booch, G., The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, P. Kroll and P. Kruchten, Editors. 2013.
7. B. G. Rumbaugh Jim, Jacobson Ivar. Guía de la Notación del UML. Versión 1.0, 2017.
8. Castañeda Garay, Miguel. (2018). Base de Datos Avanzadas.
9. Cesar (2015) Laravel, el mejor framework en PHP. Recuperado de: <https://platzi.com/blog/laravel-framework-php/> (Fecha de consulta: 4/04/2020).
10. Cochran, David (2012). Twitter Bootstrap Web Development (1st edición). Packt Publishing. p. 100. ISBN 978-1849518826.
11. Craig, L. (2015) UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson Alhambra. España.
12. C. J. Date, An introduction to database systems, 6ta Edition ed: Addison-Wesley Publishing Company, 1995.
13. Date, C. J. (1999). An introduction to database systems. (6ta ed.). Addison-Wesley Publishing Company. 16. Date, C. J. (2003).
14. Domínguez Dorado, Manuel. (2012). CodeIgniter Autocomplete. Recuperado de: <http://www.weband-development.com/codeigniter-and-eclipse-autocomplete/>
15. Erich Gamma, R. H., Ralph Johnson, John Vlissides. "Design Patterns: Elements of Reusable."

16. Gauchat, J. D.,(2012), "El gran libro de HTML5, CSS3 y JavaScript", Primera edición. Gran Via de les Corts Catalanes Gran Via de les Corts Catalanes.
17. [https://scielo.sld.cu/Formación Médica](https://scielo.sld.cu/Formación_Médica).
18. <http://lenguaje-html-cufinho.blogspot.com/p/ventajas-sencillo-que-permite-describir.html>
19. <https://fp.uoc.fje.edu/blog/por-que-elegir-el-gestor-de-base-de-datos-mysql/>
20. JacopsonIvar, G. B., James Rumbaugh (2006). El Proceso Unificado de desarrollo de Software. La Habana, Editorial Félix Varela.
21. Jacobson, Ivar; Booch, Grady; Rumbaugh, James, *El Proceso Unificado de Desarrollo de Software*. La Habana. Cuba. Editorial Félix Varela. 2004. p. 432.
22. J. Date, An Introduction to Database Systems (8th ed.): Addison-Wesley Longman, 1999.
23. L. D., Champeaux, Dennis. (2000). Object-Oriented System Development.
24. Larman, C., UML y patrones. Introducción al análisis y diseño orientado a objeto. Editorial Félix Varela ed. 2004, La Habana. Cuba.
25. Matos García, R. M. (2004). Sistema de Base de Datos. La Habana: Félix Varela.
26. Manual de CodeIgniter. Recuperado de:
<http://www.desarrolloweb.com/manuales/manual-codeigniter.html>.
Autores, C. d. (2005).
27. Manual de HTML. 5. Autores, C. d. (2008). Manual de JavaScript. Recuperado de: www.librosweb.es/javascript. [Consultado el: 18 de febrero de 2020].
28. Murugarren J.G. (2004). El lenguaje HTML.
29. Pressman, D.R.S., Ingeniería de Software Un enfoque práctico. Vol. 7. 2010. 805.
30. Pressman, Roger S. (2010). Software Engineering, a practitioner's approach. New York:McGraw-Hill.

31. Peralta, M. (2011). Estimación del esfuerzo basado en casos de uso, Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS). Buenos Aires, Argentina.
32. Paradigm (2007). "Why Visual Paradigm for UML."
33. P. Clements, R. Kazman, and M. Klein (2000), Evaluating Software Architectures: Methods and Case Studies: MA: Addison-Wesley.
34. Pressman, R (1997). Ingeniería del Software. Un enfoque práctico.
35. Rational, Rational Unified Process: Best Practices for Software Development Teams, 1999.
36. R. Pressman, Ingeniería del Software. Un enfoque práctico: McGraw-Hill/Interamericana de España, 2002.
37. Sanz, M.L., Validación y verificación: pruebas de software. Vol. Ingeniería del Software I. 79.
38. Sampieri, D.R.H., Metodología de la Investigación. abril 2006 ed. Vol. 4. 2006.
39. Sabordi Loidi Jose R.(2017) Reglamento de Organización Docente de la Educación Superior.
40. Sevilla, J. (2006). Arquitectura Cliente/Servidor.
41. Sommersville, I. (2010) Ingeniería del Software. Un enfoque práctico. 9ª edición. EditorialAddison-Wesley.
42. S. Pressman., R., Ingeniería del Software, Un enfoque práctico 2. Quinta Edición. ed. Vol. 2. 2007., La Habana. Cuba.: Editorial Félix Varela.
43. S. H. Álvarez, A.(2000), Metodología para el desarrollo de aplicaciones con Tecnología Orientada a Objetos utilizando la notación UML. La Habana,.
44. Tedeschi, N., ¿Qué es un Patrón de Diseño? 2018. 5.
45. Todo Programación (Vol. 3). Madrid: Editorial Iberprensa. 19. EVA, V.-R. D. F. U. C. I. (2012).
46. Vela Valdés (2010) Plan de estudio Medicina en Cuba 1959 a 2010. Editorial Ciencias Médicas. Educación Médica Superior. Cuba.

47. Villanueva, J. G. (2015). HTML para diseñadores. Obtenido de video2brain:
<http://www.intercambiosos.org>. Recuperado: 27 de marzo de 2020.
48. W. J. Gilmore, An Introduction to Database Normalization, 2000.
49. W. A. (2015), Introducción a los Sistemas de Base de Datos.
50. Zayas, Á. d. (2002). Metodología de la Investigación Científica.
51. Zakas, N. C. (2005). Professional JavaScript for Web Developers. (1ra Edición). Indianápolis, Indiana.

GLOSARIO DE TÉRMINOS

Aplicación web: son softwares que realizan intercambio de información a través de internet, también conocido como software web, este flujo informativo permite mediante una dirección URL (Localizador de Recursos Uniforme), que se acceda o actualice ciertas informaciones depositadas en un sitio o página web, permitiendo mantener actualizada la información si es contenida en una base de datos.

Estancia: se corresponden con las actividades docentes de una sola asignatura o de una asignatura principal que coordina o integra contenidos de otras asignaturas básicas, básico-específicas o especiales durante la carrera.

Proceso docente educativo: Es la integración holística y sistémica de la enseñanza y el aprendizaje de todos sus componentes junto con las cualidades, niveles de asimilación, de profundidad y estructural, en sus tres dimensiones: educativa, isntructiva y desarrolladora.

Atención primaria de salud: Es la asistencia sanitaria esencial accesible a todos los individuos y familias de la comunidad a través de medios aceptables para ellos, con su plena participación y aun costo asequible para la comunidad y el país.

Guía de entrevista realizada al jefe de servicio de Terapia Intermedia.

Objetivo: Recopilar información sobre el proceso docente educativo de estancia médica en el HGD Dr. Agosthino Neto y su gestión en el servicio de Terapia Intermedia.

Aspecto a evaluar

- Un proceso

Presentación

Compañero(a), es de utilidad conocer sobre cómo se lleva a cabo el proceso docente educativo de estancia médica en el hospital y específicamente en el servicio de Terapia Intermedia. Le pedimos que, dentro de sus posibilidades, nos aporte alguna información al respecto.

Cuestionario:

1. ¿Qué es una estancia?
2. ¿Dónde se realiza esta estancia?
3. ¿Cómo funciona la estancia?
4. ¿Quién está al frente de la estancia?
5. ¿Cómo se gestiona el proceso de evaluación de la estancia?
6. ¿Cómo distribuyen a los estudiantes?
7. ¿Cómo es el funcionamiento de las guardias?

Sistema informático para la gestión del proceso docente educativo de estancia en el servicio terapia intermedia del Hospital General Docente Guantánamo

Imágenes del sistema



Figura #. Portada del Sistema.

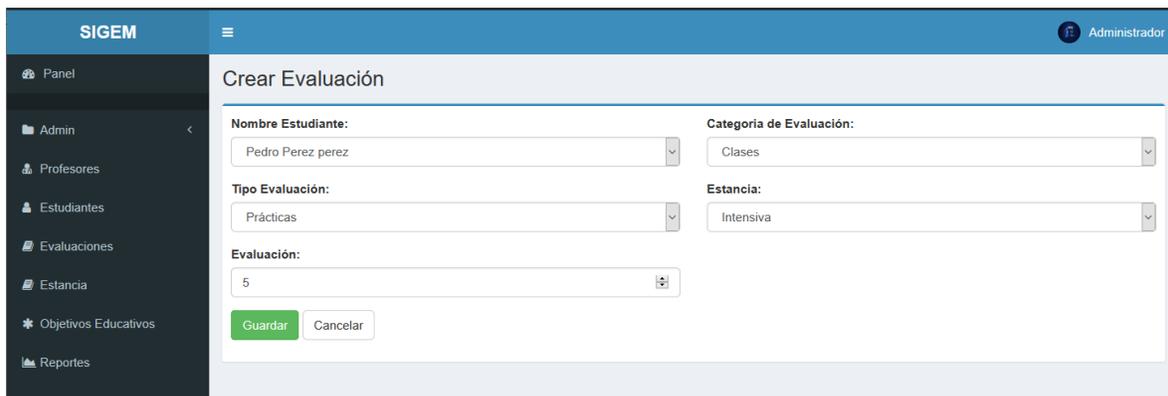


Figura #. Interfaz Insertar Evaluación.

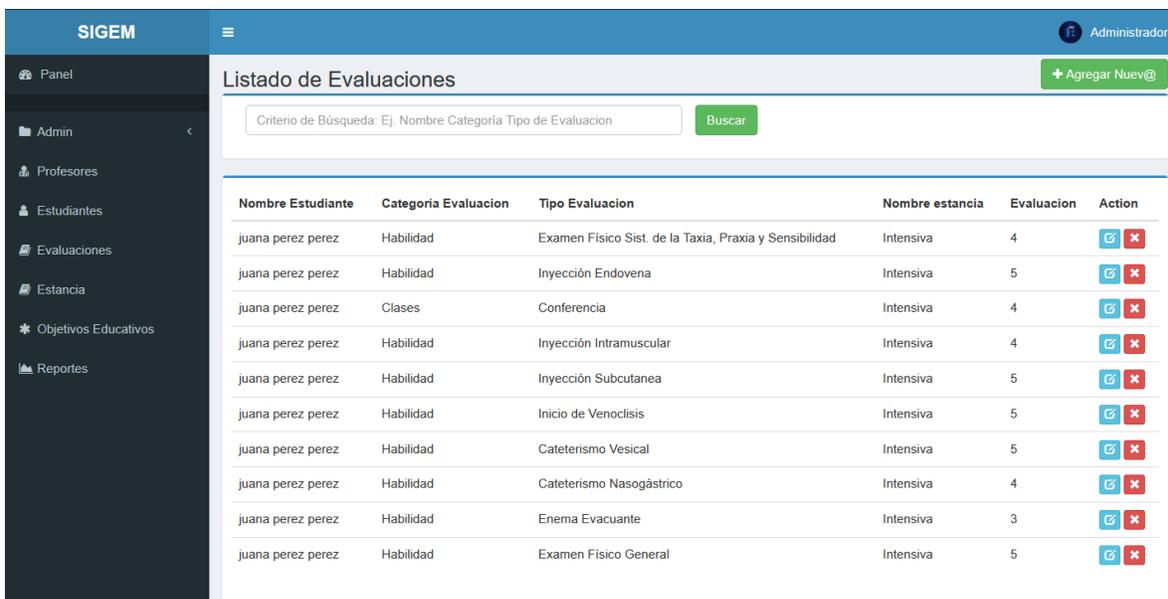


Figura #. Interfaz Listado de Evaluaciones.

Sistema informático para la gestión del proceso docente educativo de estancia en el servicio terapia intermedia del Hospital General Docente Guantánamo

SIGEM Administrador

Crear Estudiante

Nombre Completo: Ernesto Diaz Gonzalez

CI: 85478547574

Número: 3

Brigada: 1001

Guardar **Cancelar**

Figura #. Interfaz Insertar Estudiante.

SIGEM Administrador

Estudiantes

+ Agregar Nuev@

Criterio de Búsqueda: Ej. Nombre Apellidos. CI. **Buscar**

Nombre	CI	Número	Brigada	Action
Juana Perez Perez	89524152545	3	1001	
Pedro Perez Perez	86083127889	2	1001	

Figura #. Interfaz Listado de Estudiantes.