



UNIVERSIDAD DE CIENFUEGOS
INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA

Universidad de Cienfuegos

“Carlos Rafael Rodríguez”

Facultad de Ingeniería

Título:

Registro informático de Entrega de guardia administrativa de servicios no médicos en el Hospital Provincial de Cienfuegos

Autores:

Eddy Pereira Yera

Tutor:

Ing. Alberto Emilio Nuñez Blanco

Cienfuegos, Cuba

Curso: 2023

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad Cienfuegos los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

(Autor)

(Autor)

(Tutor)

(Cotutor)

Resumen:

El presente trabajo se desarrolló en el Hospital Provincial Dr. Gustavo Aldereguía Lima de la provincia Cienfuegos y tuvo como objetivo la elaboración de un sistema informático para agilizar el proceso de entrega de guardia administrativa de servicios. Mejorar la gestión del proceso de la entrega de guardia es de gran importancia para mantener la sostenibilidad y el correcto funcionamiento administrativo del hospital.

Para el desarrollo del sistema se utilizó la metodología de desarrollo de software AUP (Proceso Ágil Unificado) y el Lenguaje Unificado de Modelado (UML), así como el sistema gestor de bases de datos MySQL, PHP y JavaScript como lenguaje de programación.

Palabras Claves: entrega de guardia, proceso, servicios, administrativa.

Abstract:

The present work was carried out at the Dr. Gustavo Aldereguía Lima Provincial Hospital in the Cienfuegos province and had as objective the development of a computer system to streamline the process of delivery of administrative guard services. Improving the management of the on-call delivery process is of great importance to maintain the sustainability and proper administrative functioning of the hospital.

For the development of the system, the AUP software development methodology (Agile Unified Process) and the Unified Modeling Language (UML) were used, as well as the MySQL database management system, PHP and JavaScript as programming language.

Keywords: delivery of guard, process, services, administrative.

Índice

Introducción.....	6
Capítulo 1: Fundamentación teórica.....	9
Introducción.....	9
Descripción del Dominio del Problema.....	9
1.1.1 Conceptos asociados.....	10
Descripción del objeto de estudio	11
1.1.2 Flujo actual de los procesos y análisis crítico de ejecución de estos	11
Antecedentes nacionales.....	11
Antecedentes internacionales	13
Descripción de los sistemas existentes.....	14
1.1.3 Causas de la no utilización de los sistemas existentes	15
Tendencias, metodologías y/o tecnologías actuales	15
1.1.4 Metodología AUP	16
1.1.5 Lenguaje Unificado de Modelado(UML)	18
1.1.6 Arquitectura del sistema (Cliente-Servidor)	19
1.1.7 Patrón de arquitectura Modelo-Vista-Controlador (MVC)	20
1.1.8 Entornos de desarrollo y herramientas utilizadas	21
1.1.9 Lenguajes de programación utilizados.....	24
1.1.10 Otras tecnologías o librerías utilizadas	29
1.1.11 Frameworks utilizados	29
1.1.12 Sistema Gestor de Base de Datos(SGBD)	29
Conclusiones	33
Capítulo 2: Descripción de la solución propuesta.....	33
2.1 Análisis del proceso actual	33

2.2 Requisitos del software	34
2.2.1 Requisitos funcionales	34
2.2.2 Requisitos no funcionales	38
2.3 Historias de usuario	39
2.4 Descripción de la arquitectura.....	42
2.5 Diagrama de clases de diseño.....	44
2.6 Patrones de diseño utilizados	45
2.7 Modelo de base de datos	45
2.8 Estándar de codificación empleado	46
2.9 Conclusiones	47
Capítulo 3: Validación	48
3.1 Técnicas de validación de requisitos	48
3.2 Métricas aplicadas a los requisitos	49
3.3 Validación del diseño	49
Referencias Bibliográficas	66

Introducción

La Tecnologías de la Información y Comunicación (TIC) han permitido llevar la globalidad al mundo de la comunicación, facilitando la interconexión entre las personas e instituciones a nivel mundial, y eliminando barreras espaciales y temporales.[1]

Se denominan Tecnologías de la Información y las Comunicación al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las Tics incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual.[1]

Las tecnologías de la información de la comunicación son muy importantes en la actualidad ya que gracias a estas estamos en contacto y comunicación con muchas otras personas de nuestra comunidad, nuestra región del país o tal vez de todo el mundo. Nos comunicamos a través de mensajes por ejemplo por el teléfono celular, imágenes y sonidos por ejemplo la televisión que día a día nos muestra los sucesos más importantes que pasan en nuestro alrededor (las noticias). Otra de estas tecnologías es el internet que esta entrelazada por millones de redes en todo el mundo y pues de esa manera y por medio de algunos programas conocidos como correo electrónico podemos enviar distintos tipos de información a algunos conocidos. Bueno estos fueron algunos ejemplos y las importancias que tienen estas tecnologías.[1]

Las industrias manufactureras y de servicios que producen o comercializan bienes o servicios de tecnología de información y comunicaciones, son industrias TIC. En la región hay países con desempeño medio en TIC, como Colombia, México o Panamá, y otros con oportunidad de mejorar, como Bolivia y Paraguay.[2]

Los programas más innovadores y estratégicos en América Latina se derivan de una comprensión de la importancia de la banda ancha en el crecimiento económico y en la baja del nivel de pobreza. A partir de que ha habido estrategias para trabajar aspectos de infraestructura, de mejorar de marcos institucionales y regulatorios, de capital humano, el ecosistema digital incorpora TIC en el sector productivo.[2]

Estas tecnologías también toman gran importancia en la salud debido a que mediante las TIC es posible recopilar datos sanitarios voluminosos de información para mejorar las políticas de salud pública y determinar dónde asignar recursos, facilitar la investigación sanitaria, evaluar la eficacia de los productos farmacéuticos, ofrecer información a los pacientes interesados en determinados tratamientos, entre otras.

Tras el triunfo revolucionario del 1 de enero de 1959, Cuba se propuso un camino de desarrollo que pudiera satisfacer por igual las necesidades espirituales y materiales básicas de su población, sobre la base de una distribución más justa y equitativa de la riqueza. De esa forma, se logró satisfacer, con un acceso universal, las necesidades primarias de salud, educación, empleo, libertad y participación política, seguridad y asistencia social, desarrollo

cultural, deportes y educación física, a la vez que se emprendieron varias líneas de desarrollo científico-técnico que en algunas ramas la han situado en un lugar destacado a nivel mundial. Cuba, con un proyecto de desarrollo que tiene como pilares la justicia social, la participación popular, la equidad y la solidaridad, ha diseñado e iniciado la aplicación de estrategias que permiten convertir los conocimientos y las tecnologías de la información y las comunicaciones en instrumentos a disposición del avance y las profundas transformaciones revolucionarias.[3]

El Hospital Provincial de Cienfuegos Gustavo Alderreguía Lima tiene como misión promover la salud, enfrentar las enfermedades, sus riesgos y secuelas, y mejorar la calidad de vida de los cienfuegueros, con servicios y programas de calidad y calidez, concertados en red, en un ambiente solidario, seguro y confortable, a la vez que forma y desarrolla capital humano íntegro, gestiona un sistema de ciencia e innovación tecnológica y usa responsablemente los recursos para generar bienestar y satisfacción sostenible.[4]

Cada día durante ciertos horarios específicos se realiza un proceso llamado entrega de guardia, dicho proceso consiste en comprobar que todo se encuentre en orden, durante los cambios de turnos de los trabajadores, con información relacionada a recursos y situación perteneciente al hospital y en caso de tener que tomar medidas respecto a la información obtenida poder llevarlas a cabo, dicha información se registra en documentos escritos a mano para tener constancia de estos.

Con el objetivo de facilitar el proceso de entrega de guardias y asegurar que la información obtenida no se pierda a causa de posibles accidentes se desea crear un sistema capaz de recibir toda la información relacionada a dicho proceso, el cual una vez tenga todos los datos los almacene en una base de datos. Dicho sistema permitiría agilizar este proceso de entregas de guardias, ya que los trabajadores relacionados con el proceso tendrían una herramienta ágil y sencilla donde trabajar, además de que garantizaría que la información no se perdiese y esta podría ser consultada en cualquier momento tanto para llevar a cabo estudios estadísticos como para tomar decisiones.

Problema:

Lo expuesto anteriormente permite identificar el siguiente **problema a resolver**: ¿Cómo agilizar de manera eficiente el proceso de entrega de guardia de servicios del Hospital Provincial de Cienfuegos?

Objeto de estudio:

Proceso de entrega de guardia en el Hospital Provincial de Cienfuegos

Objetivo general:

Desarrollar un sistema informático para agilizar el proceso de entrega de guardia en el Hospital Provincial de Cienfuegos.

Objetivos específicos:

- Analizar el proceso actual de la entrega de guardia de servicios del Hospital Provincial de Cienfuegos.
- Diseñar un sistema informático que agilice el acceso a la información.
- Implementar el sistema informático propuesto.
- Realizar la estimación de esfuerzo necesario para la ejecución del proyecto.

Campo de acción: Evaluación del proceso de entrega de guardia de servicios del Hospital Provincial de Cienfuegos

Tareas a cumplir para el desarrollo de la investigación:

- Entrevista a trabajadores del Hospital Provincial de Cienfuegos que participan en el proceso de entrega de guardia.
- Estudio del procedimiento de entrega de guardia.
- Estudio de las herramientas, metodologías, lenguajes y tecnologías para la solución del problema a resolver seleccionando las más adecuadas en cada caso.
- Diseño de una arquitectura que responda a las características del trabajo.
- Estudio de los costos y beneficios que trae consigo la puesta en marcha de la solución propuesta.
- Documentación de la información generada del análisis, diseño e implementación del sistema.

Aporte práctico: Con la utilización de un sistema informático capaz de facilitar el proceso de entrega de guardia que el hospital no posee actualmente mejorará significativamente la gestión de información dentro del centro, debido a que permitirá agilizar procesos que hasta el momento no se realizan de manera óptima ni segura para los trabajadores.

Estructura capitular:

Capítulo 1. En el mismo se abordan los fundamentos teóricos donde se analizan el flujo actual del proceso de entrega de guardia que se mantiene en el Hospital Provincial de Cienfuegos, seguido de la revisión bibliográfica de antecedentes nacionales e internacionales, analizando los sistemas existentes y se dan a conocer la metodología y herramientas utilizadas en la solución del problema de investigación.

Capítulo 2. Se realiza una caracterización de la concepción general del sistema a desarrollar, donde se especifican sus requisitos funcionales y no funcionales, se conceptualizan los objetos del modelo del dominio y se representan. Se lleva a cabo la modelación de los casos de uso y sus descripciones detalladas junto a los requerimientos, así como sus análisis de robustez. Por último, se enmarca el diseño detallado del sistema a desarrollar y sus principios.

Capítulo 3. Teniendo en cuenta el capítulo 2 se realiza el estudio de factibilidad a través de la estimación por casos de uso, donde se realiza un seguimiento de los factores de cálculo de esfuerzo entre los que están, el factor de peso de los actores del sistema, el factor de peso sin ajustar de los casos de uso, los factores de complejidad técnica y los factores ambientales, permitiendo obtener los resultados de la estimación de esfuerzo y esclarecer los beneficios tangibles e intangibles y el análisis de costo del proyecto.

Capítulo 1: Fundamentación teórica

Introducción

En este capítulo se analizan los principales aspectos relacionados con la actualidad y necesidad del trabajo, el flujo actual de los procesos y análisis crítico de estos, los antecedentes nacionales e internacionales y la descripción de los sistemas existentes. Se realiza un análisis de la metodología de desarrollo de software que se utilizó para la construcción de la solución propuesta.

Descripción del Dominio del Problema

1.1.1 Conceptos asociados

Para lograr una correcta comprensión del problema resulta necesario conocer algunos conceptos relacionados con el objeto de estudio:

El formulario por el que está compuesto esta entrega de guardia en concreto se divide en **indicadores y valores**

Indicadores: Es el campo donde van indicados los diferentes servicios los cuales están agrupados en **oxígeno, volumen de agua, calderas, temperaturas, consumos eléctricos, combustible, ascensores, equipo de lavandería, situación del transporte y otros servicios.**

Oxígeno: El grupo de oxígeno (gas medicinal) está compuesto por cantidad en bala, presión en bala, presión en línea y reserva de oxígeno en unidades(u) de el mismo. Estos datos se registran en sus respectivas unidades de medida.

Volumen de agua: El grupo volumen de agua está dividido en cisterna y tanque la cual su cantidad está dada en litros(L).

Calderas: El grupo de calderas las cuales son utilizadas para generar vapor está dividido en cantidad funcionando, cantidad en reserva y cantidad rotas.

Consumos eléctricos: El grupo de consumos eléctricos en el hospital está dividido en consumo eléctrico y consumo del día pico los cuales se miden en mega watts(MW).

Combustible: El grupo combustible(diésel) está dividido en planta, gasolina y diésel incinerador.

Ascensores: Este grupo se divide en cantidad de ascensores funcionando, cantidad de ascensores rotos y cantidad de ascensoristas los cuales son los encargados de controlar los ascensores y asistir a los pacientes en su uso.

Equipo de lavandería: Este grupo se divide en cantidad de equipos de lavandería rotos y cantidad funcionado.

Situación de transporte: Aquí la situación se divide en guardia médica(si está funcionando o no) y guardia medica legal(si está funcionando o no).

Otros servicios: Entre otros servicios están la cantidad de auxiliares de limpieza, camilleros (encargados de las camillas), cantidad de camillas disponibles, cantidad de sillas de ruedas disponibles, desechos peligrosos, recogida de comunales donde se entra si llegó o no la recogida, clima central (funcionando o no) y la alimentación la cual es descriptiva.

Valores: Es el campo donde se determinan los valores de cada uno de los servicios el cual se divide en el **real 4pm** que es donde se llenan los datos que posee el servicio referido en el horario de 4pm y **real 7pm** que sería igual con la diferencia del horario a las 7pm. También están las **observaciones** donde se describen las mismas de los servicios.

Descripción del objeto de estudio

1.1.2 Flujo actual de los procesos y análisis crítico de ejecución de estos

El Hospital Provincial de Cienfuegos sigue un proceso de entrega de guardia administrativa a una hora determinada del día, en la cual se determina que todo está en orden. Este proceso no cuenta con un sistema informático que ayude a facilitar la tarea. Se realiza manual en hoja y con bolígrafo llenando una plantilla ya elaborada. Los registros resultantes de este proceso se almacenan en estantes. Todo este proceso es tedioso además de que consume recursos constantemente todos los días como tinta de impresora para elaborar la plantilla, hojas y bolígrafos para llenar la misma, además de que es posible que ocurran errores a la hora de llenar el registro y en otros casos se pueden perder dichos registros tanto antes como después de ser almacenados. Esta forma de realizar la entrega de guardia es actualmente una manera ineficiente para el hospital y sus trabajadores.

Antecedentes nacionales

Sistema de entrega de guardia para los laboratorios

Autores: Omar Mar Cornelio, Nilmar Sanchez Muguercia, Barbara Bron Fonseca, Pedro M. Puig Díaz

Localización: Serie Científica de la Universidad de las Ciencias Informáticas(UCI), 2018

Títulos paralelos: Guard delivery system for laboratorios

Resumen:

El control de los medios tecnológico es un elemento esencial para garantizar el soporte a los procesos fundamentales de una institución docente de las tecnologías, en la Universidad de las Ciencias Informática (UCI) se efectúa a través del Proceso de Entrega de Guardia para las áreas técnicas. Sin embargo es muy compleja la gestión del flujo de información obtenida de modelos previamente impresos. El presente trabajo describe la solución a la problemática planteada a partir de la informatización de proceso para lo cual se implementó una aplicación Web codificada mediante PHP, utilizando AJAX. CCS para controlar el aspecto de los documentos HTML, PostgreSQL como gestor de bases de datos y la utilización de una Infraestructura de Clave Pública para firmar documentos electrónicos.[5]

Desarrollo del módulo Técnicos de Laboratorio del sistema para la gestión de los procesos administrativos SIGPA del centro de Informatización de Entidades

Autor : Escobar Couto, Marcos Antonio

Tutor: Llarch Leyva, Silvia María López Reyes, Yoansy

Palabras clave: MODULOS; GESTION DE PROCESOS ADMINISTRATIVOS

Fecha de publicación: jun-2017

Editorial: Universidad de las Ciencias Informáticas. Facultad 3. Centro de Informatización de Entidades

Resumen:

El Centro de Informatización de Entidades, tiene la necesidad de controlar los procesos que se desarrollan en torno a los Técnicos en Ciencias Informáticas (TCI), los cuales tienen a su cargo la custodia de los laboratorios docentes y de producción. El presente trabajo tiene como objetivo la realización de un sistema web que permita informatizar los procesos sobre los cuales interactúan los TCI, tales como: proceso de entrega y recibo de laboratorios, asignación y seguimiento de sus tareas y evaluaciones de desempeño, que hoy se realizan de manera manual. La solución permitirá establecer las bases para lograr un mayor control y seguridad sobre los procesos mencionados anteriormente, logrando así reducir gastos y disminuir el tiempo en que ocurren las actividades realizadas. El desarrollo del sistema informático estuvo guiado por la metodología AUP-UCI. Para la implementación del sistema se utilizó como marco de trabajo Symfony 2.8, Bootstrap 3.1, como lenguaje del lado del servidor PHP 5.5 y del lado

del cliente HTML 5, CSS3 y JavaScript, como Entorno de Desarrollo Integrado Neatbeans 8.0, servidor de aplicaciones web Apache 2.4.9. La validación del sistema se realizó mediante la ejecución de pruebas que permitieron identificar defectos, que al ser corregidos; incrementan la calidad y aceptación del producto.[6]

Antecedentes internacionales

Evaluación de la gestión administrativa en centros médicos auspiciados por organismos internacionales, caso Club Rotario

Autor: Soto Paredes, Jennyffer Patricia

Director de Tesis: Bucheli Rosales, Carlos, dir.

Identificador de lugar: ECUADOR

Fecha de Publicación: 2011

Ciudad: Editorial: Universidad Andina Simón Bolívar, Sede Ecuador

Resumen:

La eficiente gestión administrativa es un elemento competitivo en el ámbito empresarial y con mucha más razón en instituciones que ofrecen un servicio básico para la vida del ser humano, como es la salud. En el transcurso de los siguientes capítulos se presentan los fundamentos teóricos de la gestión administrativa y se evalúa su aplicación en dos centros médicos auspiciados por organismos internacionales tomando como caso al Club Rotario. La evaluación se centró en medir el cumplimiento de las fases del proceso administrativo, encontrándose que existe una marcada diferencia entre los establecimientos evaluados, ya que a pesar de ser normados por la misma institución aplican las fases administrativas de manera diferente y en algunos casos de manera incompleta, lo que no les permite lograr su principal objetivo de eficiencia. Debido a los resultados hallados en el diagnóstico de los dos centros médicos, se propone una alternativa para mejorar su eficiencia administrativa. Partiendo de la aplicación de las fases del proceso administrativo, se definió un modelo de gestión, que ofrece a los directivos un camino a seguir para ser más eficientes y cumplir con su fin social fue diseñado con base a la fundamentación teórica, a las mejores experiencias observadas en los centros médicos evaluados y a las prácticas que son necesarias para alcanzar los objetivos deseados.[7]

Desarrollo e implementación de un sistema de gestión administrativa para el hotel Dorado del cantón Playas.

metadata.dc.contributor.advisor: Abad Sacoto, Karla Yadira

Autores: Vilela Álava, Carlos

Palabras claves: gestión; estrategia; sistematización; política

fecha: 2014

Publicado: La Libertad: Universidad Estatal Península de Santa Elena, 2014.

Resumen:

En el presente sistema contaremos con un módulo de reporte de habitaciones reservadas, reservaciones por fecha, facturación de reservaciones, y de servicios y otro módulo de consulta de habitaciones ocupadas, habitaciones libres y habitaciones reservadas. En el desarrollo del sistema se llegaron a utilizar las siguientes herramientas, Lenguaje de programación Php, diseño de pantallas y formularios, servidor local apache, para el diseño de lo que es la base de datos se utilizó Mysql. El Cantón Playas es un destino turístico con grandes atractivos y al cual visitan una importante cantidad de turistas locales, nacionales e internacionales, la demanda hotelera nacional e internacional es muy exigente y se requiere estar preparado para corresponder a dichos niveles de exigencia. Del presente proyecto se obtuvo como conclusión que la sistematización de la gestión administrativa y operativa del Hotel Dorado cumple con el objetivo general y los objetivos específicos planteados en el diseño de la tesis, es una herramienta eficaz en la gestión administrativa de la empresa y sobre todo se obtendrá mayor índices de satisfacción de parte de los huéspedes del hotel.[8]

Descripción de los sistemas existentes

Amion:

Amion es un sistema informático capaz de crear, publicar, y manejar horarios de proveedores para grupos individuales, hospitales completos o sistemas de hospitales enteros. Ayuda a simplificar comunicación entre personal, mediante mensáfono y celulares. Digamos que está viendo un paciente y necesita una consulta. Se puede iniciar sesión en amion.com para buscar el servicio que se necesite, y enviar un mensaje a el doctor en turno. Se puede usar Amion para cualquier tipo de grupo médico, de prácticas individuales a prácticas de programas de residencias. Proveedores pueden ver horarios en amion.com, sincronizar a sus sistemas de calendario personal, y usar la aplicación mobile de Amion, desarrollado en colaboración con Doximity.

Hospitales enteros pueden acceder un conjunto de funciones switchboard a nivel empresario con ningún cargo adicional. Acceso switchboard muestra información de contacto adicional. También puede rastrear mensajes de textos reciente (esta la pagina que enviamos mas temprano). Puede ver mensajes retrocediendo semanas y meses. Mantener la lista diaria de tareas precisa constantemente con "switchboard signouts' Si el planificador de horario de Anestesia no esta disponible, aquellos con acceso switchboard pueden mantener el horario al día. Cualquiera con acceso a tus horarios de Amion encontrara la persona correcta en turno, y obtendrán el cuido necesario para sus pacientes sin demora.

Los proveedores, podrán solicitar días o hasta turnos específicos que le gustaría tener libre, y al mismo tiempo, revisar las solicitudes que tus compañeros de trabajo ya han hecho. También podrá adjuntar notas a sus solicitudes. Una vez que todos hayan entrado sus solicitudes, planificadores de horarios se sientan con nuestra potente aplicación de creación de horarios, 'OnCall', para diseñar tareas que honren el tiempo libre solicitado por los individuos y satisfagan todas las reglas y patrones. La aplicación te demuestra quien esta disponible para trabajar cada día. Puedes hacer clic en nombres por su cuenta, usar plantillas, patrones, o usar el auto planificador para diseñar el borrador inicial. Ejecute recuentos rápidamente por servicio, día de la semana y en cualquier rango de fechas. Una vez el horario este publicado en amion.com, administradores podrán hacer cambios rápidos desde cualquier navegador web. El personal también podrá proponer o aceptar cambios de horarios con sus compañeros de trabajo en línea. Si el compañero de trabajo aceptara, su horario cambia automáticamente, y su planificador de horario es mantenido al tanto por correo electrónico. Sobre 250,000 proveedores médicos confían sus horarios a Amion. A \$399 por año (USD) para cualquier practica única, Amion es una gran oferta. Revise nuestro prueba gratis, y pregúntenos por asesoramiento. Solicite una cuenta piloto completa para una evaluación absoluta.[9]

ShiftWizard:

ShiftWizard™ de HealthStream™ fue diseñado para su equipo y creado para aumentar la productividad. Desde horarios simplificados hasta comunicación abierta, este software de programación de personal de atención médica y la aplicación móvil se crearon para la industria y miles de personas confían todos los días para elevar la atención al paciente y mejorar el personal de apoyo.

Involucre, capacite y retenga a su personal con la solución de programación de personal de atención médica de más rápido crecimiento. Desarrollada por profesionales de la industria y en la que confían miles de personas, esta solución se creó para respaldar mejor a su equipo y eliminar las frustraciones de programación.[10]

1.1.3 Causas de la no utilización de los sistemas existentes

Los sistemas descritos anteriormente están dirigidos a la gestión de varios procesos y servicios llevados a cabo dentro de un Hospital, sin embargo, el sistema informático que se desarrolla en esta investigación solo estará relacionado con la gestión del proceso de entrega de guardia administrativo. También estos sistemas se adaptan a los recursos de otros países y sus procesos de entrega de guardia no funcionan de la misma forma que la de nuestros hospitales.

Tendencias, metodologías y/o tecnologías actuales

1.1.4 Metodología AUP

El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Agil, Gestión de Cambios Agil, y Refactorización de Base de Datos para mejorar la productividad.

El proceso unificado (Unified Process o UP) es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (Rational Unified Process) de IBM.[11]

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.[11]

Al igual que en RUP, en AUP se establecen **cuatro fases** que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- **Incepción(Concepción):** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.[11]

Las disciplinas se llevan a cabo de manera sistemática, a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software de trabajo que responda a las necesidades de sus interlocutores. **Las disciplinas son:**

- **Modelo.** El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.
- **Aplicación.** El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.
- **Prueba.** El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.
- **Despliegue.** El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.
- **Gestión de configuración.** El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
- **Gestión de proyectos.** El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.
- **Entorno.** El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc) estén disponibles para el equipo según sea necesario.[11]

Los equipos de AUP suelen ofrecer versiones de desarrollo al final de cada iteración en preproducción área (s). Una versión de desarrollo de una aplicación es algo que podrían ser liberados en la producción si se ponen a través de su pre-producción de garantía de calidad (QA), las pruebas y los procesos de despliegue. La primera producción de liberación a menudo toma más tiempo para entregar versiones posteriores. La primera producción de liberación puede tomar doce meses para entregar la segunda versión de nueve meses, y luego otras liberaciones se entregan cada seis meses. Una de las primeras se centra en cuestiones de despliegue, no sólo permite evitar los problemas, sino que también permite tomar ventaja de sus experiencias durante el desarrollo. Por ejemplo, cuando despliegue un software en su área

deberá tomar notas de lo que funciona y lo que no, toma nota de que puede servir como la columna vertebral de su instalación de scripts.[11]

La AUP es ágil, porque está basada en los siguientes **principios**:

- **El personal sabe lo que está haciendo.** La gente no va a leer detallado el proceso de documentación, pero algunos quieren una orientación de alto nivel y / o formación de vez en cuando. La AUP producto proporciona enlaces a muchos de los detalles, si usted está interesado, pero no obliga a aquellos que no lo deseen.
- **Simplicidad.** Todo se describe concisamente utilizando un puñado de páginas, no miles de ellos.
- **Agilidad.** El ajuste a los valores y principios de la Alianza Ágil.
- **Centrarse en actividades de alto valor.** La atención se centra en las actividades que se ve que son esenciales para el de desarrollo, no todas las actividades que suceden forman parte del proyecto.
- **Herramienta de la independencia.** Usted puede usar cualquier conjunto de herramientas que usted desea con el ágil UP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.
- **Adaptación de este producto para satisfacer sus propias necesidades.** La AUP producto es de fácil acomodo común a través de cualquier herramienta de edición de HTML. No se necesita comprar una herramienta especial, o tomar un curso, para adaptar la AUP.[11]

1.1.5 Lenguaje Unificado de Modelado(UML)

Aup utiliza el Lenguaje Unificado de Modelado (UML) que fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos.[12]

1.1.6 Arquitectura del sistema (Cliente-Servidor)

Cliente-Servidor es uno de los estilos arquitectónicos distribuidos más conocidos, el cual está compuesto por dos componentes, el proveedor y el consumidor. El proveedor es un servidor que brinda una serie de servicios o recursos los cuales son consumido por el Cliente.

En una arquitectura Cliente-Servidor existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar, en este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado.[13]

Cliente-Servidor es considerada una arquitectura distribuida debido a que el servidor y el cliente se encuentran distribuidos en diferentes equipos (aunque podrían estar en la misma máquina) y se comunican únicamente por medio de la RED o Internet.

En esta arquitectura, el Cliente y el Servidor son desarrollados como dos aplicaciones diferentes, de tal forma que cada una puede ser desarrollada de forma independiente, dando como resultado dos aplicaciones separadas, las cuales pueden ser construidas en tecnologías diferentes, pero siempre respetando el mismo protocolo de comunicación para establecer comunicación.

La idea central de separar al cliente del servidor radica en la idea de centralizar la información y la separación de responsabilidades, por una parte, el servidor será la única entidad que tendrá acceso a los datos y los servirá solo a los clientes del cual el confía, y de esta forma, protegemos la información y la lógica detrás del procesamiento de los datos, además, el servidor puede atender simultáneamente a varios clientes, por lo que suele ser instalado en un equipo con muchos recursos. Por otro lado, el cliente suele ser instalado en computadoras con bajos recursos, pues desde allí no se procesa nada, simplemente actúa como un visor de los datos y delega las operaciones pesadas al servidor.

Quizás uno de los puntos más característicos de la arquitectura Cliente-Servidor es la centralización de los datos, pues el server recibe, procesa y almacena todos los datos provenientes de todos los clientes. Si bien los clientes por lo general solo se conectan a un solo servidor, existen variantes donde hay clientes que se conectan a múltiples servidores para funcionar, tal es el caso de los navegadores, los cuales, para consultar cada página establece una conexión a un servidor diferentes.[13]

Ventajas:

- Centralización: El servidor fungirá como única fuente de la verdad, lo que impide que los clientes conserven información desactualizada.
- Seguridad: El servidor por lo general está protegido por firewall o subredes que impiden que los atacantes puedan acceder a la base de datos o los recursos sin pasar por el servidor.

- Fácil de instalar (cliente): El cliente es por lo general una aplicación simple que no tiene dependencias, por lo que es muy fácil de instalar.
- Separación de responsabilidades: La arquitectura cliente-servidor permite implementar la lógica de negocio de forma separada del cliente.
- Portabilidad: Una de las ventajas de tener dos aplicaciones es que podemos desarrollar cada parte para correr en diferentes plataformas, por ejemplo, el servidor solo en Linux, mientras que el cliente podría ser multiplataforma.

Desventajas:

- Actualizaciones (clientes): Una de las complicaciones es gestionar las actualizaciones en los clientes, pues puede haber muchos terminales con el cliente instalado y tenemos que asegurar que todas sean actualizadas cuando salga una nueva versión.
- Concurrencia: Una cantidad no esperada de usuarios concurrentes puede ser un problema para el servidor, quien tendrá que atender todas las peticiones de forma simultánea, aunque se puede mitigar con una estrategia de escalamiento, siempre será un problema que tendremos que tener presente.
- Todo o nada: Si el servidor se cae, todos los clientes quedarán totalmente inoperables.
- Protocolos de bajo nivel: Los protocolos más utilizados para establecer comunicación entre el cliente y el servidor suelen ser de bajo nivel, como Sockets, HTTP, RPC, etc. Lo que puede implicar un reto para los desarrolladores.
- Depuración: Es difícil analizar un error, pues los clientes están distribuidos en diferentes máquinas, incluso, equipos a los cuales no tenemos acceso, lo que hace complicado recopilar la traza del error.

1.1.7 Patrón de arquitectura Modelo-Vista-Controlador (MVC)

Este **patrón de diseño**, es una excelente práctica para definir la arquitectura de sistemas robustos de forma clara y eficiente. Es muy útil cuando se necesita desarrollar sistemas escalables, que requieran mantenimiento y permite que nuestra estructura sea más comprensible para los demás programadores que deseen colaborar. **¿Pero qué significan estas siglas? En este post, explicare en que consiste.**

MVC es un **patrón de diseño** que se estructura mediante tres componentes: **modelo, vista y controlador**. Este patrón tiene como principio que cada uno de los componentes esté separado en diferentes objetos, esto significa que los componentes no se pueden combinar

dentro de una misma clase. Sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario.

Modelo: este componente se encarga de manipular, gestionar y actualizar los datos de una base de datos. No contiene ninguna lógica que describa como presentar los datos a un usuario.

Vista: este componente presenta los datos del modelo al usuario. La vista sabe cómo acceder a los datos del modelo, pero no sabe que significa esta información o que puede hacer el usuario para manipularla.

Controlador: este componente se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. Por medio del controlador se comunican el modelo y la vista: solicitando los datos necesarios, manipularlos para obtener los resultados y entregarlos a la vista para que pueda mostrarlos.

Los tres componentes de **MVC** están interconectados. La vista muestra el modelo para el usuario, después el controlador acepta la entrada del usuario y actualiza el modelo y debido a esta acción la vista vuelve a tener un cambio con los datos actualizados.

Este patrón de diseño es uno de los más utilizados en la actualidad. Existen muchos **frameworks** en diferentes lenguajes que utilizan MVC, como **Laravel para PHP**, **Django para Python**, **Ruby on Rails para Ruby**, **Express para NodeJS** entre otros.

En mi experiencia, es importante que para que apliques los principios y buenas prácticas del **MVC**, tengas antes un conocimiento sólido de la programación orientada a objetos.[14]

1.1.8 Entornos de desarrollo y herramientas utilizadas

Visual Studio Code: Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación.

Para tener una idea de la popularidad de Visual Studio Code y la aceptación que ha tenido en el mundo de desarrollo, podemos consultar datos. Según una encuesta realizada por Stack Overflow a más de 80,000 desarrolladores en mayo del 2021, Visual Studio Code es el entorno de desarrollo más usado y con mucha diferencia, un 71.06%. En la siguiente ilustración, puedes ver el top 10.[15]

VS Code tiene una gran variedad de características útiles para agilizar el trabajo, que lo hacen el editor preferido por muchos para trabajar los proyectos.[15]

Multiplataforma: Es una característica importante en cualquier aplicación y más si trata de desarrollo. Visual Studio Code está disponible para Windows, GNU/Linux y macOS.[15]

IntelliSense: Esta característica está relacionada con la edición de código, autocompletado y resaltado de sintaxis, lo que permite ser más ágil a la hora de escribir código. Como su nombre lo indica, proporciona sugerencias de código y terminaciones inteligentes en base a los tipos de variables, funciones, etc. Con la ayuda de extensiones se puede personalizar y conseguir un IntelliSense más completo para cualquier lenguaje.[15]

Depuración: Visual Studio Code incluye la función de depuración que ayuda a detectar errores en el código. De esta manera, nos evitamos tener que revisar línea por línea a puro ojo humano para encontrar errores. VS Code también es capaz de detectar pequeños errores de forma automática antes de ejecutar el código o la depuración como tal.[15]

Uso del control de versiones: Visual Studio Code tiene compatibilidad con Git, por lo que puedes revisar diferencias o lo que conocemos con git diff, organizar archivos, realizar commits desde el editor, y hacer push y pull desde cualquier servicio de gestión de código fuente (SMC). Los demás SMC están disponible por medio de extensiones.[15]

Extensiones: Hasta ahora, he mencionado varias veces el término *extensiones* porque es uno de los puntos fuertes. Visual Studio Code es un editor potente y en gran parte por las extensiones. Las extensiones nos permiten personalizar y agregar funcionalidad adicional de forma modular y aislada. Por ejemplo, para programar en diferentes lenguajes, agregar nuevos temas al editor, y conectar con otros servicios. Realmente las extensiones nos permiten tener una mejor experiencia, y lo más importante, no afectan en el rendimiento del editor, ya que se ejecutan en procesos independientes.[15]

Enterprise Architect: es una plataforma de modelado, diseño y administración, colaborativa, basada en UML(lenguaje unificado de modelado) 2.5 y estándares relacionados. Ágil, intuitiva y extensible, con poderosas características para dominios específicos totalmente integradas, a una fracción del costo de muchos competidores. Una solución para toda la empresa que permite visualizar , analizar, modelar, probar y mantener un amplio rango de sistemas, software, procesos y arquitecturas. Basado en equipos, con un historial probado y más de 740,000 usuarios en todo el mundo; Enterprise Architect es el conjunto de herramientas escalable, compatible con muchos estándares y es perfecta para su próximo proyecto.[16]

Características:

- Velocidad
- Escalabilidad
- Baja Complejidad
- Buena Documentación

- Alta popularidad
- Soporte para pruebas
- Desarrollo distribuido[17]

Ventajas y Desventajas:

Ventajas:

- Fácil de usar
- Utiliza los 13 diagramas UML
- Genera gráficos de gran calidad
- Excelente Documentación

Desventajas:

- Herramienta en idioma Inglés
- Difícil de encontrar los diagramas
- Es Pagado[17]

Zotero: es una aplicación para la administración de referencias bibliográficas. Es una aplicación (software libre) que funciona con un conector para los navegadores Chrome, Mozilla Firefox, Safari y Opera. Los conectores detectan automáticamente el contenido mientras navega por la web y le permiten guardarlo en Zotero con un solo clic.[18]

Zotero detecta cuando un libro, artículo u otro recurso se está visualizando y con un clic, encuentra y guarda la información de referencia completa a un archivo local. Si la fuente es un artículo en línea o una página Web, Zotero puede, opcionalmente, guardar una copia de la fuente. Los usuarios pueden añadir notas, etiquetas, y sus propios metadatos a través de la interfaz del navegador.[18]

Características de Zotero:

- Recupera referencias: Permite guardar nuestras referencias de forma automática con un solo clic.
- Descarta duplicados: Identifica de manera automática los duplicados y, además, te permite fusionarlos en un solo ítem.
- Apoya el control de autoridades: ayuda a uniformar los nombres de los autores de la forma en como ellos lo han pedido, o para evitar ambigüedades.

- Excluye las autocitas: solo es cuestión de buscar al autor dentro de la colección donde guardamos las referencias y nos arrojará las autocitas.
- Identifica los tipos de citas: Identificar los diferentes tipos de citas. Por ejemplo, señala los diferentes tipos de citas, es decir, las consideras como A y las consideras como B.
- Crea bibliografías: nos ayudan a generar listados de bibliografías en el estilo bibliográfico que más nos convenga.[19]

1.1.9 Lenguajes de programación utilizados

PHP: PHP es un **lenguaje de programación de uso general** que se utiliza, sobre todo, en el entorno del **desarrollo web**. Este lenguaje se utiliza generalmente para desarrollar el backend de una web, el lado del servidor. Aun así, tiene numerosas utilidades en frontend. Es por esto que es uno de los principales lenguajes de programación en el mundo de la programación web.[20]

PHP tiene una característica que lo diferencia de los demás lenguajes, y es que **está diseñado para incrustarse en HTML**. Esto significa que un documento HTML puede contener elementos de PHP integrados, siempre y cuando se sigan unas normas determinadas.

Una de las particularidades de PHP es que **el código donde se integre se ejecutará en el lado del servidor** y sólo después se enviará al cliente. En consecuencia, el cliente únicamente recibirá el resultado del script, no el código que lo conforma. Esto significa que, al ejecutar un archivo PHP, este llega al navegador en forma de HTML. Debido a esto, el código incrustado será invisible para los usuarios.

Con esto deberías hacerte una idea de qué es lo que diferencia PHP de los demás lenguajes, incluso de aquellos utilizados en desarrollo web.[20]

HTML: HTML es el lenguaje con el que se define el contenido de las páginas web.

Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc.[21]

Etiquetas en HTML: son fragmentos de texto rodeados por corchetes cuyo uso es escribir el código HTML. Se delimitan usando paréntesis angulares (“< >”) de la siguiente forma:

<etiqueta>. Las etiquetas se utilizan para describir algo que se quiere representar en una página web, su apariencia.[22]

Tipos de etiquetas:

- El HTML tiene definidas gran variedad de etiquetas para distintos usos en este artículo te vamos a explicar las más esenciales.

- Etiquetas de apertura. <etiqueta> → Son el inicio. Indican dónde comienza el elemento o donde comienza a tener efecto. Consisten en el nombre del elemento encerrado entre los corchetes angulares.
- Etiquetas de cierre. </etiqueta> → Son lo mismo que las etiquetas de apertura, pero estas indican donde finaliza el elemento. Además, se diferencian de las otras ya que antes de escribir el elemento entre los corchetes, debemos escribir una barra diagonal “/”.
- Etiquetas de título. <title> → Se usa para indicar que lo que se pondrá a continuación es el título de la página web.
- Etiquetas de cuerpo. <body> → Esta etiqueta indica la parte del cuerpo o desarrollo del texto.
- Etiqueta de sección inferior. <footer> → Indica la parte inferior del texto. Puede ser una conclusión o la parte con iconos de redes sociales.
- Etiqueta de sección superior. <head> → Esta etiqueta indica la parte superior del texto o encabezado.
- Etiqueta de título. <h1> → Título o encabezado de la página.
- Etiqueta de subtítulo. <h2> → Título de nivel 2 o subtítulo.
- Etiqueta de apartado. <h3> → Subapartado.
- Etiquetas de párrafo. <p> → Hace que cada elemento o texto aparezca en una nueva línea agrupado.
- Etiquetas de negrita. → Sirven para que el texto que encierran se encuentre resaltado.
- Etiquetas de cursiva. → Los elementos que encierra aparecen en cursiva.
- Etiquetas de imagen. → Se emplea para insertar una imagen en la página en el lugar donde se quiere incluir. *¡Cuidado! Esta etiqueta no necesita una de cierre.*
- Etiquetas de tachado. → Los elementos que encierra aparecen tachados.
- Etiquetas de enlace. <a> → Permite generar un vínculo o link.

Cada una de las etiquetas debe acabar con su equivalente de cierre. Debemos tener claro que es muy importante que por cada etiqueta que abramos, deberemos incluir la correspondiente etiqueta de cierre. Así conseguiremos un código HTML bien formado y que los navegadores puedan interpretar sin espacio a error ni ambigüedad.[22]

CSS: CSS son las siglas en inglés para «hojas de estilo en cascada» (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de las páginas.

Se les denomina hojas de estilo «en cascada» porque puedes tener varias hojas y una de ellas con las propiedades heredadas (o «en cascada») de otras.

Para muchas personas una simple plantilla de blog es suficiente. Aun así, cuando quieras personalizar la apariencia de un sitio necesitarás implementar CSS que, en conjunto con un buen CMS, te ayudará a potenciar el alcance de tu contenido.[23]

Ventajas:

El lenguaje CSS tiene la ventaja de ser mucho más simple, por lo que implica menos código y probabilidad de errores, así como una mayor velocidad de carga y facilidad de lectura. Además, con él tienes una mayor gama de posibilidades de edición para hacer que tu sitio luzca tal como quieres.

Al desarrollar una página web es necesario hacer énfasis en la importancia de los elementos visuales para comunicar un mensaje. La presentación de los datos de una empresa y la usabilidad de la plataforma son factores fundamentales para atraer visitantes a tu sitio y retener su atención en el contenido que presentas. Algunas ventajas adicionales en el uso de CSS son que:

Optimiza la edición

Los sitios web de algunas empresas contienen grandes cantidades de información que deben ser accesibles a los usuarios. Homologar los estilos de todos los accesos a esta información puede ser complicado si no se cuenta con una herramienta que facilite el proceso. CSS permite crear estilos que pueden aplicarse a todas las páginas de un sitio web. Esto ahorra tiempo y posibilita crear una imagen de la marca a través de tipografías, colores y recursos visuales.

Facilita la accesibilidad al usuario

La multiplicidad de visitantes de un sitio web es tan amplia como la diversidad de dispositivos utilizados para acceder a ellos. Al momento de diseñar una página es necesario tener presentes las posibilidades de interacción y diferencias en la presentación de contenidos en distintos dispositivos. Adecuar la plataforma para medios de acceso como [teléfonos](#), [tabletas](#), [computadoras de escritorio](#) o [laptops](#) puede resultar una tarea complicada. CSS tiene la ventaja de facilitar la accesibilidad al usuario, gracias a las hojas de estilo estandarizadas.

Promueve la creatividad

El uso de CSS para la construcción de páginas web tiene la ventaja de permitir explotar la creatividad de los diseñadores de manera rápida e intuitiva. Esto acelera el proceso de personalización de los sitios, que pueden ser contruidos con especificaciones claras o bien sujetarse a las particularidades de los distintos navegadores.

Al construir una imagen de marca es importante modificar, innovar y proponer soluciones. CSS simplifica esta tarea para los desarrolladores.

Prioriza la limpieza del código

Una estrategia común, pero poco eficiente, consiste en escribir las instrucciones de un programa en lenguaje HTML. Esto implica la redacción de una gran cantidad de líneas de código que se intercalan con el contenido de un sitio web.

Al utilizar CSS puedes separar todo el código relacionado con el estilo de un sitio web del contenido base de una página, que puede estar en lenguaje HTML. De este modo, mantienes una limpieza en ambos conjuntos de información y evitarás tener líneas de código de contenido interfiriendo entre sí.[23]

Características:

Es un lenguaje de programación diferente a HTML.

Como hemos revisado, HTML es un lenguaje que sirve para gestionar la información contenida en un sitio web; por otro lado, CSS tiene la función de estructurar el estilo de las páginas. Ambos lenguajes trabajan en conjunto para presentar la información al público final.

Permite el apilamiento de instrucciones para definir formatos específicos

Esto significa que se pueden crear bloques de instrucciones anidadas que permiten hacer modificaciones generales de modo sencillo, lo cual simplifica la tarea de diseño y la creación de estilos estandarizados. Así, se crean formatos específicos que pueden aplicarse a distintas páginas y que son fácilmente modificables.

Es utilizable en todos los navegadores y plataformas

Debido a que es un lenguaje ampliamente utilizado para dar formato a los sitios web, su uso es universal para un amplio número de dispositivos, formatos y plataformas como Edge, Safari, Chrome, etc. Por ello es fácil dar formato a las páginas dependiendo del navegador utilizado por cada usuario.

Optimiza el funcionamiento de las páginas web

Al separar el código de contenido del código de estilo es mucho más rápido el procesamiento de la información, lo cual se traduce en una experiencia más fluida para los usuarios y en una carga de trabajo menor para los procesadores. Es importante e imprescindible la sincronización del código en HTML con CSS para que la información se presente de modo correcto.

Tiene una sintaxis específica

Si bien la mayoría de los lenguajes comparten algunas funciones y signos, existen particularidades en el uso de CSS, por lo que es necesario conocer el lenguaje de programación, así como las características de apilamiento. Más adelante revisaremos algunas de sus utilidades específicas.

Permite personalizar totalmente la apariencia de las páginas

CSS da cabida a una gran libertad creativa, lo que significa que los diseñadores tienen un amplio espectro de posibilidades en sus herramientas. El uso de diferentes códigos de colores y de fuentes permite emplear paletas de muchísimas tonalidades y múltiples tipografías; asimismo, se pueden disponer los elementos visuales de un sitio de acuerdo con las necesidades del diseño.[23]

JavaScript: JavaScript es un lenguaje de programación creado para cumplir con las necesidades del paisaje envolvente del internet. Desde su concepción, ha crecido en popularidad y utilidad, y ahora está presente no solo en el desarrollo web. JavaScript es la insignia del desarrollo web interactivo y, como resultado, es universal en esta industria.

En el contexto actual, JavaScript se utiliza para todo, gracias a la introducción de Node.js. Esta tecnología crea software robusto para empresas en todo el mundo. Por si fuera poco, organizaciones como LinkedIn y Medium lo implementan al construir plataformas para que los usuarios tengan acceso a sus servicios.[24]

Lo que se puede hacer con JavaScript abarca diferentes tipos de software, como juegos, programas de computadora, aplicaciones web y hasta tecnologías de blockchain. JavaScript es posiblemente el lenguaje de programación más popular de la web. Por ejemplo, más de 125.000 empleos en LinkedIn buscan profesionales con habilidades en JavaScript.

Ya que hemos discutido el alcance de JavaScript en la industria laboral, hablemos más de su uso más común: el desarrollo web.

El uso más popular de JavaScript es para el desarrollo web, y es una de las herramientas más poderosas que un desarrollador puede tener en sus manos. Los desarrolladores usan JavaScript en esta área para añadir interactividad y funciones que mejoren la experiencia del usuario y hagan a internet mucho más disfrutable.

JavaScript se ha expandido más allá del desarrollo de interfaz, que es donde comenzó. Recientemente, JavaScript ha llegado al back-end, o dorsal de desarrollo web. Esto quiere decir que los desarrolladores tienen acceso de interfaz a métodos CRUD (Create, Read, Update, Destroy; en español: Crear, Leer, Actualizar, Destruir) y hasta puede utilizarse en el motor de un sitio web.

Además, de acuerdo con W3techs, más del 90 % de todos los sitios web funcionan con JavaScript. Esto lo convierte en el líder primordial en tecnología de desarrollo web.[24]

Usos específicos de JavaScript en desarrollo web:

- Interactividad de interfaz o front-end: el desarrollo web mejora solamente por el aumento de la interactividad y funciones que JavaScript ofrece.

- Aplicaciones Web: las aplicaciones web son similares a los sitios, pero en su lugar pueden empacarse en una caja más compacta, que mejora el control de la seguridad y otros aspectos.
- Juegos de navegador: los navegadores web actuales han cambiado mucho; los desarrolladores pueden crear juegos robustos que funcionan en ellos.
- Desarrollo web dorsal o back-end: el desarrollo web se ha transformado tanto, que JavaScript puede utilizarse para gestionar el back-end de sitios y aplicaciones.[24]

JavaScript está considerado como un lenguaje de programación del lado del cliente. Esto quiere decir que opera en el navegador del usuario y no funciona en un dispositivo externo. Un ejemplo de un lenguaje que no está del lado del cliente sería MySQL, un lenguaje del lado del servidor que gestiona peticiones de cualquier base de datos.

Por otro lado, JavaScript no requiere que nada se descargue a los dispositivos del usuario, ya que los navegadores modernos tienen el software requerido integrado a ellos. Esto hace que JavaScript sea más amigable con el usuario que otros lenguajes.[24]

1.1.10 Otras tecnologías o librerías utilizadas

1.1.11 Frameworks utilizados

Bootstrap: Bootstrap es un [framework](#) CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina [CSS](#) y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, **su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles.**

Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada.

1.1.12 Sistema Gestor de Base de Datos(SGBD)

MySQL: MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL). MySQL funciona prácticamente en todas las plataformas, incluyendo Linux, UNIX y Windows. Aunque puede utilizarse en una amplia gama de aplicaciones, MySQL se asocia más a menudo con las aplicaciones web y la publicación en línea.

MySQL es un componente importante de una pila empresarial de código abierto llamada LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de bases de datos relacionales y PHP como lenguaje de scripting orientado a objetos (a veces se utiliza Perl o Python en lugar de PHP).

Originalmente concebido por la empresa sueca MySQL AB, MySQL fue adquirido por Sun Microsystems en 2008 y luego por Oracle cuando compró Sun en 2010. Los desarrolladores pueden utilizar MySQL bajo la licencia pública general GNU (GPL), pero las empresas deben obtener una licencia comercial de Oracle.

Hoy en día, MySQL es el RDBMS que está detrás de muchos de los principales sitios web del mundo y de innumerables aplicaciones corporativas y de consumo basadas en la web, como Facebook, Twitter y YouTube.[27]

Entre las ramificaciones de MySQL, también conocidas como forks, se encuentran las siguientes:

- Drizzle, un sistema ligero de gestión de bases de datos de código abierto en desarrollo basado en MySQL 6.0;
- MariaDB, un popular sustituto de MySQL desarrollado por la comunidad que utiliza las API y los comandos de MySQL; y
- Percona Server con XtraDB, una versión mejorada de MySQL conocida por su escalabilidad horizontal.[27]

MySQL es uno de los sistemas más popularizados para almacenar y administrar datos. Con administrar nos referimos a las acciones CRUD:

- Create: crear
- Read: leer
- Update: actualizar
- Delete: borrar

Características de MySQL:

- Código abierto

- MySQL utiliza la Licencia Pública General de GNU, por lo que se puede descargar, utilizar y modificar a voluntad. Esto facilita su uso tanto académico como profesional.
- Uso multiplataforma
- Una de sus características principales y de mayor ventaja es que puede instalarse en entornos con sistemas operativos diversos como Windows, Mac y la mayoría de distribuciones Linux, así como en ambientes Unix.
- Escalabilidad
- Tiene soporte para 40-50 millones de registros, 150.000-200.000 tablas y 5000 millones de filas.
- Tipos de datos
- Soporta una amplia gama de tipos de datos, lo que permite tener una gran versatilidad en cuanto a las situaciones, industrias o casos de uso donde puede implementarse una base de datos MySQL. Puede emplearse para la industria financiera, al manejar datos con mucha precisión; por otro lado, también puede utilizarse en ámbitos de geolocalización por sus datos de tipo espacial. De igual forma puede competir, en ciertas situaciones, con las bases de datos no relacionales con su tipo de dato JSON.
- Conjuntos de caracteres
- Es compatible con un gran listado de conjuntos de caracteres e idiomas, lo que le permite adaptarse a cualquier parte del mundo. Sin duda alguna, es un aspecto que le ha ayudado a posicionarse en los sistemas de internet a lo largo y ancho del planeta.
- Clientes gráficos
- Si bien MySQL utiliza su propio lenguaje para administrar los datos almacenados, existen diversas herramientas o clientes gráficos que nos permiten interactuar con las bases de datos, ayudando a que dicha interacción sea más sencilla y, por lo tanto, más rápida. Algunas herramientas son:
- PHPMyAdmin
- MySQL Workbench
- Soporte para lenguajes de programación
- Las características y ventajas de MySQL son muchas, pero sin duda todas ellas son mejor explotadas cuando están integradas dentro de un sistema de información. Para ello existe un amplio abanico de API nativas, librerías, paquetes, etc. que permiten integrar una base de datos MySQL en un sistema desarrollado en cualquier lenguaje de programación.

- Documentación actualizada
- Al ser muy popular y utilizado, permite que exista una documentación oficial muy amplia, además de una comunidad enorme siempre dispuesta a ayudar, colaborar y aportar al conocimiento compartido.

PHPMyAdmin: phpMyAdmin es una herramienta gratuita, que permite de una manera muy completa acceder a todas las funciones de la base de datos MySQL, mediante una interfaz web muy intuitiva.

Esta aplicación consta de un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web y así cuando accedamos a esos archivos nos mostrara unas páginas donde estarán las base de datos a las que tenemos acceso en nuestro servidor de base de datos con sus tablas.[28]

Características:

- Esta aplicación nos permitirá realizar las operaciones básicas en base de datos MySQL, como son: crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves de campos, administrar privilegios y exportar datos en varios formatos. La función de exportar datos se emplea muchas veces para realizar backups de la base de datos y poder restaurar esta copia de seguridad en el futuro a través de phpMyAdmin mediante la opción “importar”.
- phpMyAdmin es el administrador de bases de datos por defecto en muchos paneles de control comerciales como son cPanel, Plesk o DirectAdmin.
- Los usuarios no deberían tener problemas a la hora de manejar esta herramienta, ya que es fácil de usar.
- Otra de las funciones más importantes que nos ofrece es que permite optimizar y reparar tablas, las cuales son dos tareas de mantenimiento fundamentales.
- Nos da la posibilidad de realizar búsquedas en las base de datos, además de poder escribir nuestras propias consultas SQL de manera directa y ejecutarlas.
- Esta herramienta también es de gran ayuda para desarrolladores de aplicaciones que empleen MySQL, ya que permite depurar consultas y hacer tests de forma rápida y sencilla.[28]

Especificaciones:

- Interface Web.
- Manejador de base de datos MySQL, MariaDB, Drizzle.

- Importación de datos desde CSV y SQL.
- Exporta datos a varios formatos: CSV, SQL, XML, PDF (vía la biblioteca TCPDF), ISO/IEC 26300 OpenDocument Text y Spreadsheet, Word, Excel, LaTeX y otros.
- Administración de múltiples servidores.
- Crea gráficos PDF del diseño de la base de datos.
- Crea consultas complejas usando Query-by-Example (QBE).
- Búsqueda global en una base de datos o un subconjunto de esta.
- Transforma datos almacenados a cualquier formato usando un conjunto de funciones predefinidas, tal como BLOB.

Conclusiones

En este capítulo se realizó un análisis de los procesos, destacando así la necesidad de llevar a cabo la presente investigación. Se analizaron los sistemas existentes relacionados con el campo de acción y se determinó que no cumplen con las prestaciones que se requieren para este trabajo. Luego de haber realizado un análisis de las tecnologías y tendencias de desarrollo web, se seleccionó la metodología AUP y lenguaje unificado de modelado(UML) como guía para la documentación de la aplicación propuesta. Se utilizará MySQL como SGBD ya que los datos del centro están implementados en dicho gestor, y como lenguaje de programación PHP y JavaScript, además de VUEjs como framework para la implementación de la aplicación.

Capítulo 2: Descripción de la solución propuesta

En el presente capítulo se describe la propuesta de solución y los artefactos generados por el escenario #4 en la metodología Agile Iniciad Procesos (AUP) en las etapas de análisis, diseño e implementación.

2.1 Análisis del proceso actual

El proceso que se lleva a cabo en el Hospital de Cienfuegos para la realización de la entrega de guardia administrativa con el fin de mantener una correcta administración no tiene una vía

sencilla para llevarse a cabo ya que se realiza de una forma manual y tediosa y con un gasto constante de recursos.

2.2 Requisitos del software

2.2.1 Requisitos funcionales

Un requisito funcional(RF) es una declaración de cómo debe comportarse un sistema informático. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como atributos que el usuario detecta.

- Administrador: usuario del que se requerirá su información de inicio de sesión, el cual podrá realizar cambios sobre la información que se provee en el sistema los datos relacionados a la entrega de guardia.
- Usuario: usuario que también requerirá un inicio de sesión, este se encargará de llenar el formulario propuesto para la entrega de guardia, así como modificar sus propias entradas de datos.

Tabla 1Requisitos Funcionales

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Iniciar sesión	El usuario introduce sus datos, el sistema lo comprueba y si son válidos, el usuario queda autenticado como administrador o usuario, permitiendo acceder a servicios que se permitan en dependencia del rol iniciado	Alta	Alta
RF2	Listar datos requeridos del formulario	El usuario con permisos administrativos accede a una lista de todos los datos requeridos del formulario y su estado actual(habilitado o	Alta	Baja

		deshabilitado) y se le permite modificarlos		
RF3	Agregar un nuevo dato al formulario	El usuario con permisos administrativos accede a un formulario, a través del cual llenando todos los campos requeridos, los cuales se comprueban y validan correctamente, inserta la información a tener en cuenta en la entrega de guardia	Alta	Media
RF4	Modificar un dato del formulario	El usuario con permisos administrativos accede a la modificación de la información de un dato requerido del formulario, a través de un formulario que muestra automáticamente la información ya insertada en el sistema y permitiendo que se pueda sobrescribir con nueva información, la cual se comprueba y valida correctamente además de poder habilitar o deshabilitar el dato a modificar.	Alta	Alta
RF5	Listar usuarios registrados	El usuario con permisos administrativos accede a una lista de todos los usuarios disponibles y se le permite	Alta	Baja

		cambiar la contraseña del usuario elegido.		
RF6	Listar muestras(horarios de las entregas de guardia)	El usuario con permisos administrativos accede a una lista de las muestras requeridas y se le permite modificarlas	Alta	Baja
RF7	Listar entregas de guardia realizadas	El usuario accede a una lista de todas las entregas de guardias realizadas, de manera que el sistema permita una vez los datos estén mostrados, elegir si desea ver los detalles de la entrega de guardia seleccionada asi como modificarlos y un filtro de búsqueda para encontrar una entrega de guardia en especifico a partir de su fecha y tipo de muestra.	Alta	Baja
RF8	Listar detalles de los datos de las entregas de guardia	El usuario accede a todos los datos detallados de la entrega de guardia seleccionada	Alta	Baja
RF9	Modificar estado de la muestra seleccionada	El usuario con permisos administrativos accede a la modificación de la muestra seleccionada, a través de un formulario que muestra los estados a modificar.	Alta	Alta

RF10	Registrar nuevo usuario	El usuario con permisos administrativos accede a un formulario donde se inserta en los campos disponibles el nombre del nuevo usuario su rol y su contraseña comprobando con otro campo si esta es correcta	Alta	Alta
RF11	Cambiar contraseña de usuario	El usuario con permisos administrativos accede al cambio de contraseña del usuario elegido, a través de un formulario que muestra automáticamente la información ya insertada en el sistema y permitiendo que se pueda sobrescribir con nueva información, la cual se comprueba y valida correctamente.	Alta	Alta
RF12	Modificar datos guardados de la entrega de guardia realizada	El usuario accede a la modificación de los datos guardados de la entrega de guardia seleccionada a través de un formulario que muestra automáticamente la información ya insertada en el sistema y permitiendo que se pueda sobrescribir con nueva información, la cual se comprueba y valida correctamente.	Alta	Alta

RF13	Llenar formulario de entrega de guardia	El usuario accede a la introducción de los datos requeridos de la entrega de guardia a llenar a través de un formulario que muestra automáticamente la información requerida permitiendo que se pueda introducir con información, la cual se comprueba y valida correctamente.	Alta	Media
RF14	Buscar entrega de guardia especificada	El usuario accede a un filtro de búsqueda donde a partir de la fecha y muestra seleccionada se muestre la entrega de guardia con dichos datos especificados tanto para modificarse o para ver sus detalles	Alta	Media
RF15	Agregar nueva muestra	El usuario con permisos administrativos accede a un formulario en el cual se debe introducir el dato de la muestra que se desee agregar	Alta	Media

2.2.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) se refieren a las características y propiedades que debe tener el producto relacionado con su desempeño, calidad y éxito.

Tabla 2 Requisitos no funcionales

No.	Descripción	Tipo de requisito
RNF1	La interfaz debe ser fácil, intuitiva, legible, amistosa y mantener el formato en vistas similares.	Requisito de interfaz o apariencia externa
RNF2	Se requiere un navegador web instalado en las computadoras clientes	Requisito de software
RNF3	El sistema requiere de una computadora que haga la función de servidor, la misma puede poseer sistema operativo Linux como Windows, esta debe cumplir con las siguientes características: • Memoria RAM con 1 GB o más • Capacidad de Disco Duro de 50 GB o más	Requisito de hardware
RNF4	La información estará protegida contra accesos no autorizados mediante mecanismos de autenticación	Requisito de seguridad
RNF5	Para garantizar la integridad de la información se utilizarán políticas de salvaguarda de la base de datos	Requisito de seguridad

2.3 Historias de usuario

Las historias de usuarios son una descripción breve, informal y en lenguaje sencillo de lo que un usuario desee hacer dentro de un sistema informático para obtener información que le resulte valiosa. En total se definieron 11 historias de usuarios, a continuación, se muestra una de ellas:

Tabla 3 Historia de usuario

Número: 12	Requisito: Modificar datos guardado de la entrega de guardia realizada
Programador: Eddy Pereira Yera	Iteración asignada: 1

Prioridad: Alta	Tiempo estimado: 14 horas
Riesgo en desarrollo:	Tiempo real:
Descripción: El componente permitirá modificar los datos guardados de una entrega de guardia ya realizada, mostrando un formulario con los datos actuales, permitiendo modificarlos e introducir otros nuevos. Se permitirán modificar los siguientes campos que pueden variar en dependencias de los cambios realizados por el administrador. • Muestra: desplegable donde estarán disponibles todas las opciones a modificar. • Guardia: campo de texto donde se introduce el nombre del encargado de esa entrega de guardia. • Chofer de guardia: campo de texto donde se introduce el nombre del chofer encargado en esa entrega de guardia. • Cantidad en balas de oxígeno: campo de texto donde se introduce la información ya nombrada. • Presión en balas de oxígeno: campo de texto donde se introduce la información ya nombrada. • Presión en línea: campo de texto donde se introduce la información ya nombrada. • Consumo de 23-7: campo de texto donde se introduce la información ya nombrada. • Consumo de 7-15: campo de texto donde se introduce la información ya nombrada. • Reserva de oxígeno: campo de texto donde se introduce la información ya nombrada. • Cantidad de nitroso: campo de texto donde se introduce la información ya nombrada. • Cantidad de CO2: campo de texto donde se introduce la información ya nombrada. • Aire comprimido: campo de texto donde se introduce la información ya nombrada. • Cisterna: campo de texto donde se introduce la información ya nombrada.	

- Tanque: campo de texto donde se introduce la información ya nombrada.
- Calderas funcionando: campo de texto donde se introduce la información ya nombrada.
- Calderas en reserva: campo de texto donde se introduce la información ya nombrada.
- Calderas rotas: campo de texto donde se introduce la información ya nombrada.
- Clima/Temperatura agua: campo de texto donde se introduce la información ya nombrada.
- Consumo eléctrico (MW): campo de texto donde se introduce la información ya nombrada.
- Consumo Pico Día (MW): campo de texto donde se introduce la información ya nombrada.
- Combustible Diésel Planta: campo de texto donde se introduce la información ya nombrada.
- Fuel Oil: campo de texto donde se introduce la información ya nombrada.
- Diésel incinerador: campo de texto donde se introduce la información ya nombrada.
- Ascensores Funcionando: campo de texto donde se introduce la información ya nombrada.
- Ascensores Rotos: campo de texto donde se introduce la información ya nombrada.
- Ascensoristas: campo de texto donde se introduce la información ya nombrada.
- Auxiliares de limpieza: campo de texto donde se introduce la información ya nombrada.
- Camilleros: campo de texto donde se introduce la información ya nombrada.
- Camillas: campo de texto donde se introduce la información ya nombrada.

- Sillas de ruedas: campo de texto donde se introduce la información ya nombrada.
- Desechos peligrosos: campo de texto donde se introduce la información ya nombrada en bien, regular o mal.
- Recogida de comunales: campo de texto donde se introduce la información ya nombrada en si o no.
- Situación del transporte (Guardia medica): campo de texto donde se introduce la información ya nombrada en funcionando o no.
- Situación del transporte (Guardia medica legal): campo de texto donde se introduce la información ya nombrada en funcionando o no.
- Equipo de lavandería roto: campo de texto donde se introduce la información ya nombrada.
- Equipo de lavandería funcionando: campo de texto donde se introduce la información ya nombrada.

Observaciones: Al modificar una entrega de guardia ya realizada, se necesitan llenar todos los campos.

2.4 Descripción de la arquitectura

La arquitectura del software constituye una especificación de las principales ideas del diseño proporcionando una descripción más detallada de cómo realizar dicho sistema.

Patrón de arquitectura Modelo Vista Controlado

La propuesta de solución está basada en el patrón Modelo-Vista-Controlador, el cual describe la forma de organizar el código de la aplicación separando los datos de la misma, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- Modelo: Componente encargado del acceso a datos, en este componente se encuentran las tablas de la base de datos.

- Vista: Transforma el modelo en una página web que permite al usuario interactuar con ella, en este componente están las clases servidoras, las clientes, los formularios, la portada y el Layout. Dicho Layout tiene relación de inclusión con todas las clases servidoras, las servidoras construyen las clientes y las clientes contienen los formularios. La portada es la que se relaciona directamente con el componente controlador.

Controlador: Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista, en este componente esta la clase controladora y la index.php, la controladora se encarga de manejar toda la información tanto del modelo como de la vista y la index.php es la que se relaciona con la portada y envía información a los componentes.

En la siguiente figura, se muestra cómo interactúan estos tres componentes en la arquitectura definida:

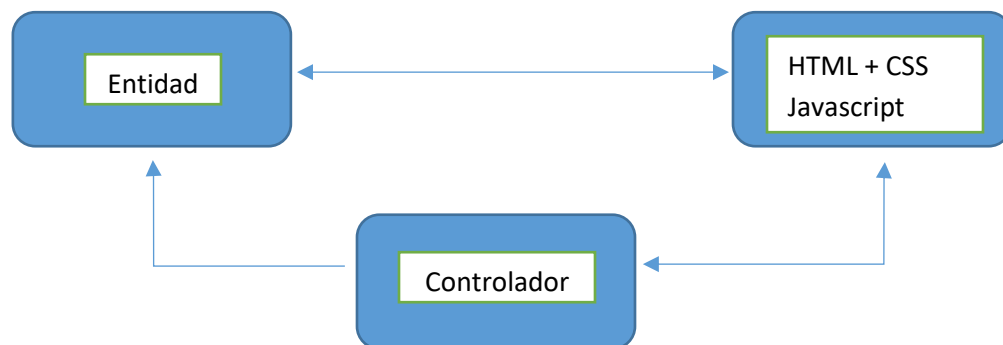


Ilustración 1 Modelo Vista Controlador

A continuación, se muestra la aplicación de la arquitectura a la propuesta de solución:

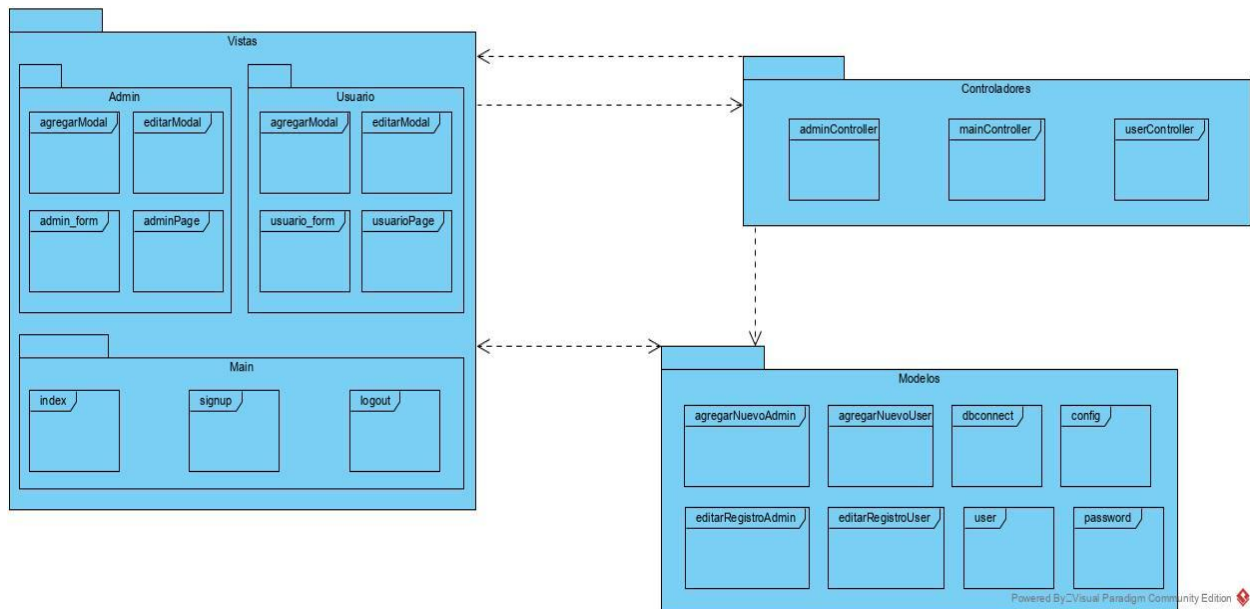


Ilustración 2 Arquitectura Modelo Vista Controlador

2.5 Diagrama de clases de diseño

Un **diagrama de clases** es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

A continuación, se muestra el diagrama de clases de la propuesta de solución:

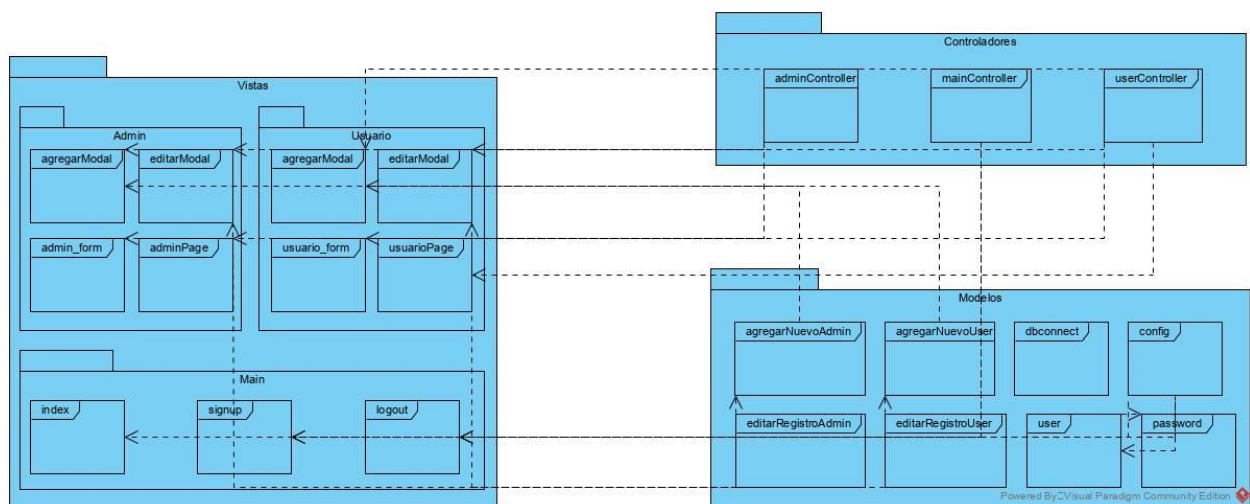


Ilustración 3 Diagrama de clases de diseño

2.6 Patrones de diseño utilizados

Los **patrones de diseño** son unas técnicas para resolver problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces.

Patrones GRASP: son *patrones generales de software para asignación de responsabilidades*, es el acrónimo de "GRASP (object-oriented design General Responsibility Assignment Software Patterns)". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

Experto: El GRASP de experto en información es el principio básico de asignación de responsabilidades. Nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Creador: El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

Controlador: El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control.

Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Alta cohesión y Bajo acoplamiento: Los conceptos de cohesión y acoplamiento no están íntimamente relacionados, sin embargo se recomienda tener un mayor grado de cohesión con un menor grado de acoplamiento. De esta forma se tiene menor dependencia y se especifican los propósitos de cada objeto en el sistema.

2.7 Modelo de base de datos

Un **modelo de base de datos** es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos.

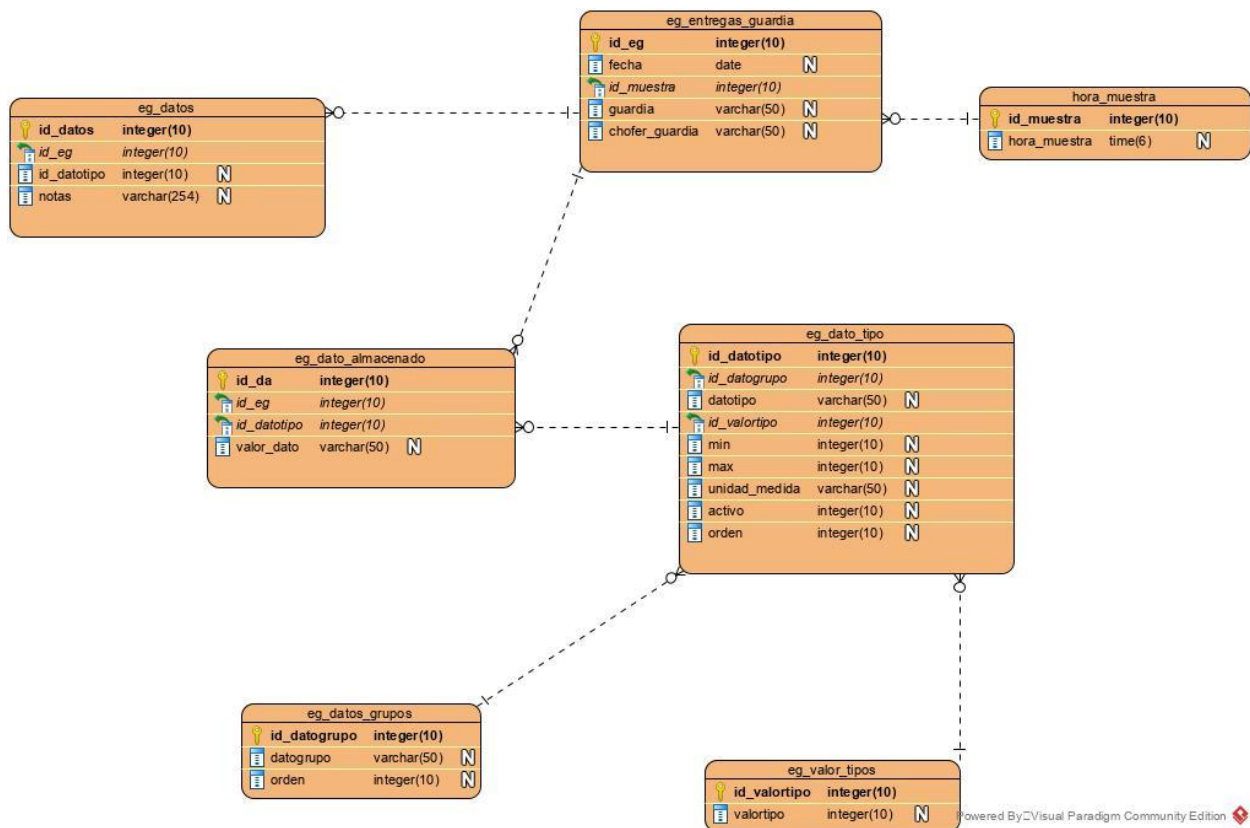


Ilustración 4 Modelo de base de datos

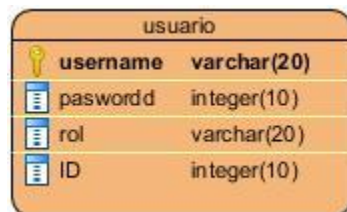


Ilustración 5 Modelo de base de datos (2da parte)

2.8 Estándar de codificación empleado

Con el objetivo de lograr una estandarización en la programación del sistema se decide aplicar el estándar de codificación CamelCase, el cual facilitará la lectura, comprensión y mantenimiento del código. A continuación se describe a grandes rasgos las convenciones de nomenclatura.

Convenciones de nomenclatura

General

- Se exceptúan el uso de las tildes y la letra ñ, la que será sustituida por nn.
- En todo momento se utilizarán nombres que sean claros, concretos y libres de ambigüedades.

- El nombre de todas las variables y métodos comenzarán con letra minúscula y si este está compuesto por varias palabras, todas las palabras internas que lo componen comienzan con mayúscula.

Indentación

- El contenido siempre se indentará con tb para las tablas y nm para los nomencladores, nunca utilizando espacios en blanco.

Clases

- El nombre de las clases comenzará con minúsculas si este está compuesto por varias palabras, todas las palabras internas que lo componen comienzan con mayúscula.
- Intentar mantener los nombres de las clases descriptivos y simples. Usar palabras completas, evitar acrónimos y abreviaturas, a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML.

Nombre de variables

- No se utilizarán nombres de variables que puedan ser ambiguos.
- Se procurará evitar dar nombres sin sentido a variables temporales.
- Las variables booleanas deben tener nombres que sugieran respuestas o contenidos de tipo Sí/No.
- Los nombres de las variables booleanas deben ser positivos

2.9 Conclusiones

El sistema diseñado garantiza y facilita el acceso a un formulario de entregas de guardia donde se encontrará la información referente a los campos requeridos y creados por el administrador. La arquitectura permite la ampliación de las funcionalidades del sistema. Las aplicaciones de los patrones de diseño garantizan lograr un esquema del mismo de fácil entendimiento, extensión y mantenimiento, además de permitir la reutilización de componentes en aplicaciones futuras. Por otra parte, el empleo del estándar de codificación facilita la lectura, comprensión y mantenimiento del código para los desarrolladores.

Capítulo 3: Validación

En el presente capítulo mediante la validación de los requisitos y el diseño propuesto se valora la viabilidad de la propuesta de solución, además de la realización de pruebas funcionales, unitarias y de aceptación al sistema informático de Entrega de guardia administrativa de servicios no médicos.

3.1 Técnicas de validación de requisitos

Es muy importante asegurar la validez de los requisitos previamente a comenzar un desarrollo de software. Para ello debe de hacerse una comprobación de la correspondencia entre la descripciones iniciales y si el modelo es capaz de responder al planteamiento inicial. Para llevar a cabo esto, se suele realizar comprobando que el modelo obtenido responde de la misma forma deseada que la que el cliente pide por un lado, y por otro a la inversa si otras respuestas del modelo convencen al cliente. En algunos casos será necesario construir prototipos con una funcionalidad similar muy reducida para que el cliente se haga una idea aproximada del resultado.(29)

La validación de los requisitos, obviamente tiene como objetivo comprobar que estos son correctos. Esta fase debe realizarse o de lo contrario se corre el riesgo de implementar una mala especificación, con el costo que eso conlleva. Los parámetros a validar en los requisitos son :

- Validez: No basta con preguntar a un usuario, todos los potenciales usuarios pueden tener puntos de vista distintos y necesitar otros requisitos.
- Consistencia: No debe haber contradicciones entre unos requisitos y otros.
- Completitud: Deben estar todos los requisitos. Esto es imposible en un desarrollo iterativo, pero, al menos, deben estar disponibles todos los requisitos de la iteración en curso.
- Realismo: Se pueden implementar con la tecnología actual.
- Verificabilidad: Tiene que existir alguna forma de comprobar que cada requisito se cumple.(29)

3.2 Métricas aplicadas a los requisitos

Con el objetivo de medir la calidad de la especificación de los requisitos se aplicó una de las métricas Calidad de la especificación (CE). Para obtener cuán entendibles y precisos son los requisitos, primeramente, se calcula el total de requisitos de la especificación como se muestra a continuación:

(Nr): El total de requisitos de especificación.

(Nf): Cantidad de requisitos funcionales.

(Nnf): Cantidad de requisitos no funcionales.

Como resultado de la sustitución de los valores en las variables, para el sistema se obtiene:

$$Nr = Nf + Nnf$$

$$Nr = 15 + 5$$

$$Nr = 20$$

Para determinar, finalmente, la Especificidad de los Requisitos (ER) o ausencia de ambigüedad en los mismos se realiza la siguiente operación:

$$ER = Nui / Nr$$

Donde es el número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas. Mientras más cerca de 1 esté el valor de ER, menor será la ambigüedad.

Para el caso de los requisitos obtenidos solo uno produjo contradicción en las interpretaciones. Sustituyendo los valores en las variables se obtiene:

$$ER = 19/20$$

$$ER = 0.95$$

Arrojando un resultado final satisfactorio, indicando que el grado de ambigüedad de los requisitos es sumamente bajo (5%) ya que el 95% es entendible. Los requisitos ambiguos fueron modificados y validados para garantizar su correcta comprensión.

3.3 Validación del diseño

Métricas de diseño: Permiten medir de forma cuantitativa la calidad de los atributos internos del software. Esto permite al ingeniero evaluar la calidad durante el desarrollo del sistema.

Son varios los puntos de vista relacionados con la calidad del software. Las métricas de diseño a nivel de componentes se concentran en las características internas de los componentes del software con medidas que pueden ayudar al desarrollador a juzgar la calidad de un diseño a nivel de componente. (30)

Atributos de calidad

Las métricas se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el desarrollo de software. Inclina sus objetivos a mejorar la comprensión de la calidad del producto, a estimar la efectividad del proceso y mejorar la calidad del trabajo.

Las métricas empleadas están diseñadas para evaluar los siguientes atributos de calidad:

- Responsabilidad . Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- Complejidad de implementación. Consiste en el grado de dificultad que tiene implementado un diseño de clases determinado.
- Reutilización. Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- Acoplamiento. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.
- Complejidad del mantenimiento. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
- Cantidad de pruebas. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado. (30)

Tamaño operacional de clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

- Responsabilidad: Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
- Complejidad de implementación: Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
- Reutilización: Un aumento del TOC implica una disminución del grado de reutilización de la clase. (30)

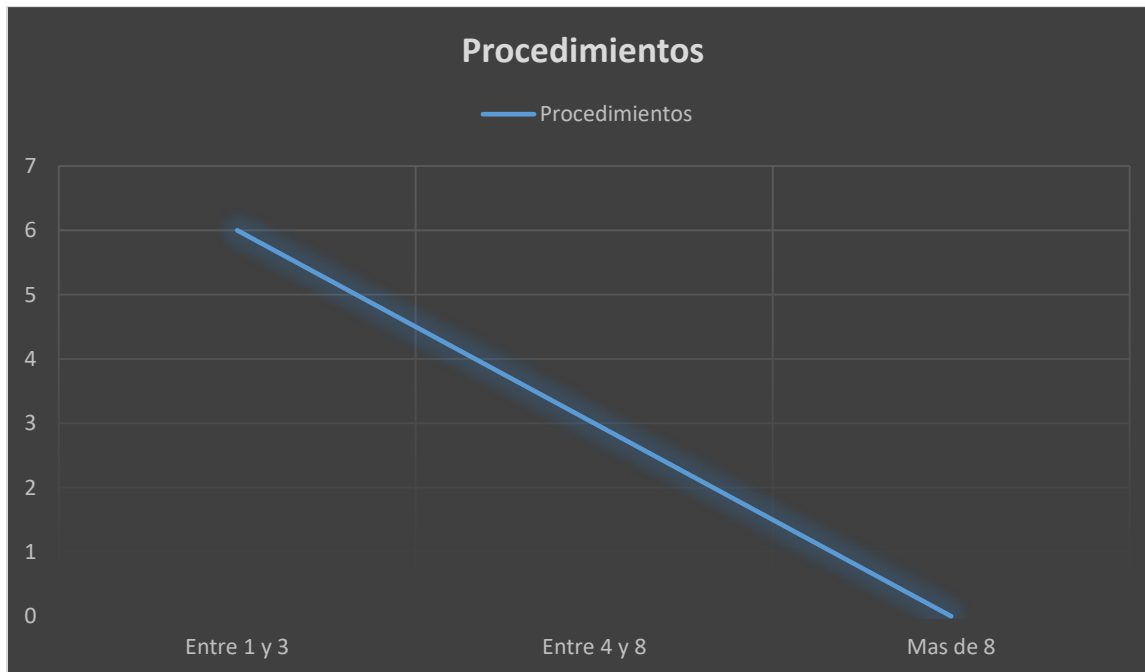


Ilustración 6 Tamaño Operacional de Clases

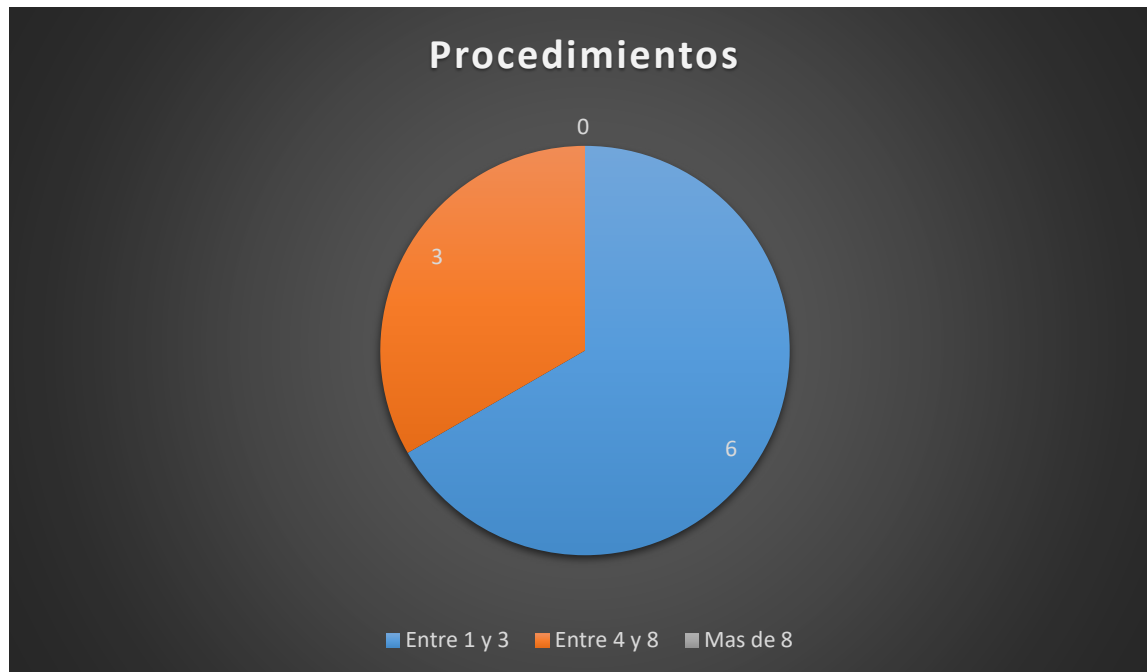


Ilustración 7 Tamaño Operacional de Clases (2da parte)

Relaciones entre clases (RC)

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

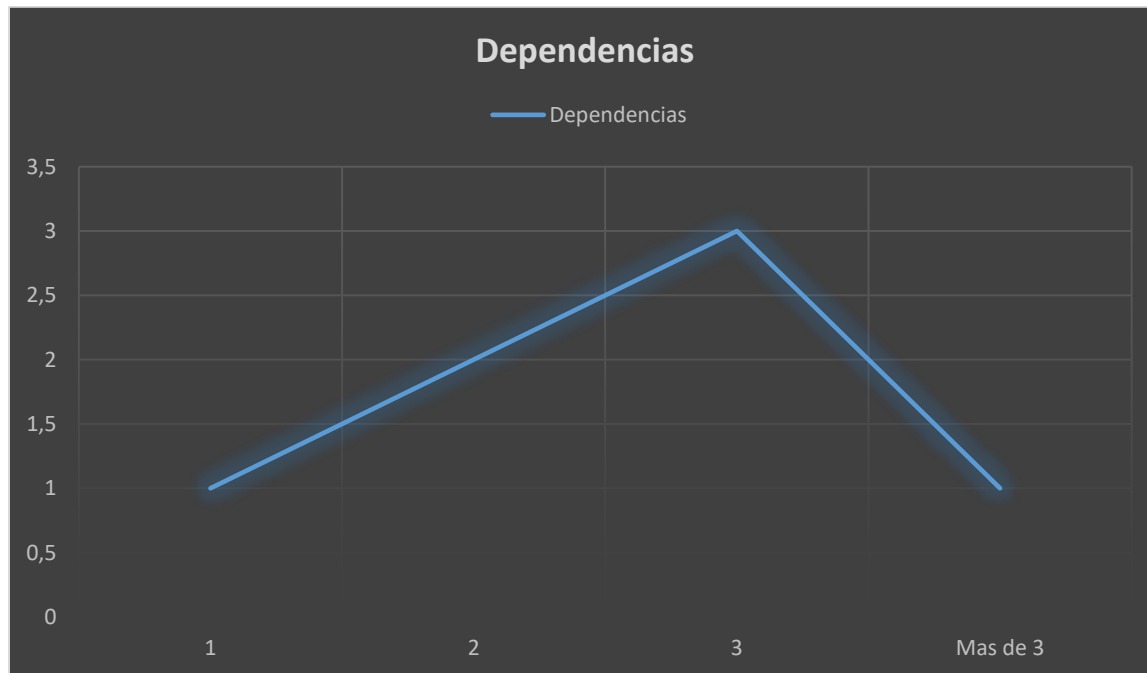
- Acoplamiento: Un aumento del RC implica un aumento del Acoplamiento de la clase.
- Complejidad de mantenimiento: Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- Reutilización: Un aumento del RC implica una disminución en el grado de reutilización de la clase.
- Cantidad de pruebas: Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase. (30)

Matriz de inferencia de indicadores de calidad

También llamada matriz de cubrimiento, es una representación estructurada de los atributos de calidad y métricas utilizadas para evaluar la calidad del diseño propuesto.

Dicha matriz permite conocer si los resultados obtenidos de las relaciones atributo/métrica es positivo o no, llevando estos resultados a una escalabilidad numérica donde, si los resultados son positivos se le asigna el valor de 1, si son negativos toma valor 0 y si no existe relación es considerada como nula y es representada con un guión simple (-). Luego se puede obtener un resultado general para cada atributo promediando todas sus relaciones no nulas.

La gráfica muestra la representación de los resultados obtenidos agrupados en los intervalos definidos. Después de un análisis se decidió que no es posible eliminar estas dependencias por la responsabilidad arquitectónica que deben mantener las clases. (30)



Haciendo un análisis de los resultados en la evaluación de la métrica RC, se puede concluir que el diseño del sistema está entre los niveles de calidad requeridos, pudiéndose observar que más de la mitad de las clases posee entre 1 y 3 dependencias. Igualmente, el resto posee índices aceptables en cuanto a Acoplamiento. Así mismo los atributos de calidad Complejidad de mantenimiento, Reutilización y Cantidad de pruebas se comportan satisfactoriamente. Confirmando la elevada reutilización, el bajo acoplamiento, la baja complejidad y la baja cantidad de pruebas que se necesitan realizar.

3.4 Pruebas de software

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo.

Pruebas como proceso

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

Objetivos de las pruebas de software.

La prueba de software es un elemento crítico para la garantía del correcto funcionamiento del software. Entre sus objetivos están:

- Detectar defectos en el software.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Para lograr los objetivos propuestos, un ingeniero de software deberá conocer los principios básicos que guían las pruebas del software. (31)

Principios de las pruebas de software.

Las pruebas se rigen por una serie de principios, una buena comprensión de estos facilitará el posterior uso de los métodos en un efectivo diseño de casos de prueba. A continuación se citan:

- La prueba puede ser usada para mostrar la presencia de errores, pero nunca su ausencia.
- La principal dificultad del proceso de prueba es decidir cuándo parar.
- Evitar casos de pruebas no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.
- Una parte necesaria de un caso de prueba es la definición del resultado esperado.
- Los casos de pruebas tienen que ser escritos no solo para condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.
- El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
- Estas leyes que definen básicamente la aplicación de las pruebas de software ayudan a refinar el producto de software a través de las etapas involucradas. (31)

Niveles de prueba

Un nivel de prueba es un grupo de actividades que se organizan y administran juntas dentro de la ejecución de un proceso de pruebas que tiene como objetivo verificar y validar los componentes de un producto. Estos niveles permiten seleccionar diferentes tipos y técnicas de pruebas a realizar en cada nivel.

Prueba de unidad: Tienen como objetivo probar las unidades de software que componen el producto. Típicamente las pruebas de unidad se aplican a componentes de producto tales como: código fuente, archivos binarios, archivos de datos, entre otros, y tienen como objetivo verificar que los flujos de control, los flujos funcionales y los flujos de datos del elemento software son cubiertos, y que ellos funcionan como se espera.

Prueba de integración: La prueba de integración es ejecutada para asegurar que los componentes software y hardware del producto operan adecuadamente cuando interactúan entre ellos para ejecutar un caso de uso (o transacción de negocio). Las pruebas de integración exponen la no completación o los errores en las especificaciones de la interfaz de cada paquete software siendo integrado con los demás.

Prueba de sistema: Tradicionalmente, las pruebas del sistema se realizan cuando el producto software está completado. El objetivo es evaluar si un producto software cumple con los requisitos que han sido especificados. Un ciclo de vida iterativo permite probar el sistema mucho más tempranamente tan pronto como los subconjuntos bien formados de requisitos funcionales se han construido.

Prueba de aceptación: La prueba de aceptación de usuario es la prueba final antes de desplegar el software en los ambientes de operación. El objetivo de la prueba de aceptación es verificar que el software está listo, que puede ser utilizado, que satisface los criterios de aceptación, y que cubre aquellas necesidades y expectativas de los clientes para los cuales el software construyó.

En diferentes libros se han propuesto varias estrategias de prueba de software, todas ellas proporcionan una plantilla, o sea brindan una guía al profesional y un conjunto de hitos al jefe de proyecto para la correcta realización de las pruebas. Estas deben incluir pruebas de bajo nivel que verifiquen que se ha implementado correctamente todos los pequeños segmentos de código y pruebas de alto nivel que validen que las funcionalidades del sistema tengan correspondencia con lo especificado por el cliente. Para los efectos de la presente investigación se propone la siguiente estrategia de pruebas:

- **Caja Blanca-Tipo unitaria**
- **Caja Negra-Tipo funcionalidad**

3.4.1 Pruebas de caja blanca

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.

- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se les aplican a los softwares, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Camino Básico:

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe.

Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. (31)

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

- A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
- Se calcula la complejidad ciclomática del grafo.
- Se determina un conjunto básico de caminos independientes.
- Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. (31)

Empleando la técnica del camino básico se muestra el proceso de pruebas realizado a la funcionalidad `validarEntregaGuardia()`. Primeramente, se comienza por analizar el código y enumerar las instrucciones:

```

public function validarEntregaGuardia($fecha, $id_muestra) {

    $sql = "SELECT EXISTS(SELECT fecha FROM eg_entregas_guardia

        WHERE fecha = :fecha

        AND id_muestra = :id_muestra) as existencia";//1

    $stmt = $this->db->prepare($sql);//2

    $stmt->bindParam(':fecha', $fecha, PDO::PARAM_STR);//3

    $stmt->bindParam(':id_muestra', $id_muestra, PDO::PARAM_STR);//4

    $stmt->execute();//5

    $result = $stmt->fetch(PDO::FETCH_ASSOC);//6


    if ($result['existencia']=='1') {//7

        return false;//8

    } else {

        return true; //9

    }

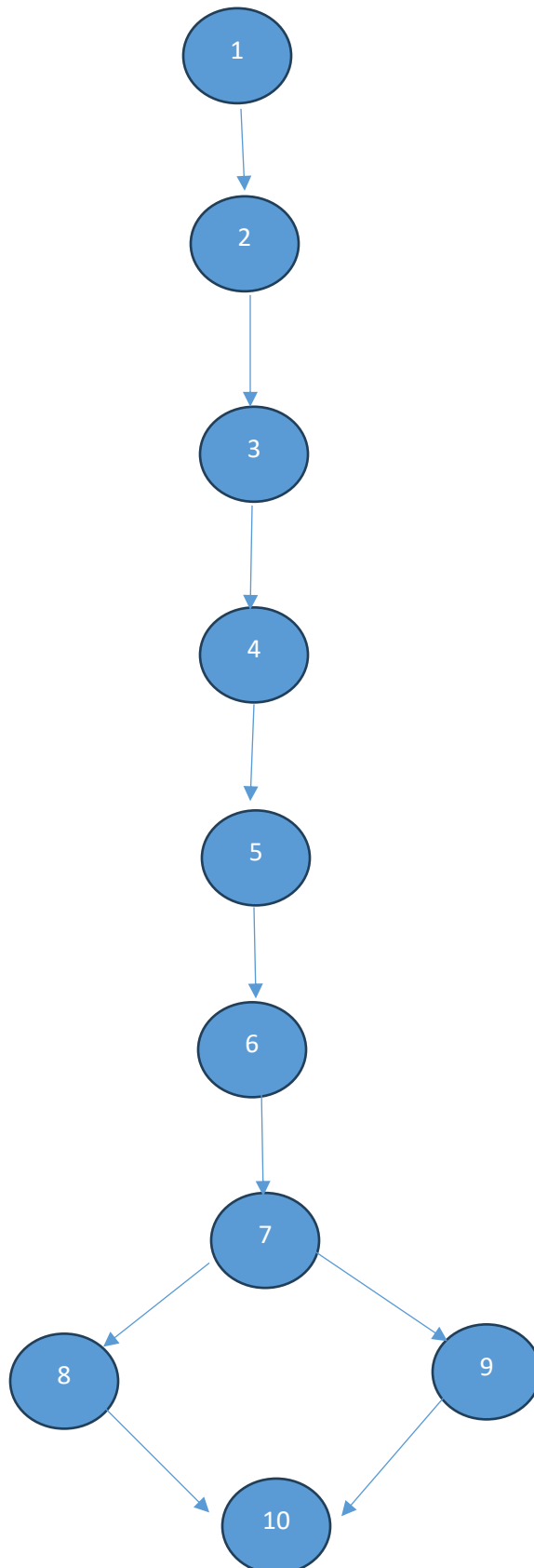
}

} //10

```

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir seguidamente el grafo de flujo correspondiente al código de la función como se muestra en la figura.

Ilustración 8 Diagrama de flujo de caja blanca



Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición. La complejidad ciclomática puede ser calculada de 3 formas:

1. $V(G) = a - n + 2$, siendo a el número de arcos o aristas del grafo y n el número de nodos.

2. $V(G) = r$, siendo r el número de regiones cerradas del grafo.

3. $V(G) = c + 1$, siendo c el número de nodos de condición.

Realizando los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

1. $V(G) = a - n + 2$

$$V(G) = 10 - 10 + 2$$

$$V(G) = 2$$

2. $V(G) = 2$

3. $V(G) = c + 1$

$$V(G) = 1 + 1$$

$$V(G) = 2$$

Por lo que el conjunto de caminos básico sería:

- Camino básico 1: 1-2-3-4-5-6-7-8-10
- Camino básico 2: 1-2-3-4-5-6-7-9-10

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación se muestra el resultado de las pruebas aplicadas a los caminos básicos 1 y 2.

Descripción: Se verificará que no se agregue una entrega de guardia con una fecha y muestra igual a alguna que ya esté almacenada en la base de datos.

--Condición de ejecución: se debe haber cargado el modelo agregarNuevoUser, para ello se debe haber seleccionado la opción Llenar formulario, y haber llenado los campos correspondientes al mismo, donde se validan los datos

--Entrada: \$sql (Entregas de guardias almacenadas en la base de datos)

--Resultados esperados: Que se inserten correctamente los datos del formulario y los muestre

--Resultado obtenido: satisfactorio

Descripción: Se verificará que no se agregue una entrega de guardia con una fecha y muestra igual a alguna que ya esté almacenada en la base de datos.

--Condición de ejecución: se debe haber cargado el modelo agregarNuevoUser, para ello se debe haber seleccionado la opción Llenar formulario, y haber llenado los campos correspondientes al mismo, donde se validan los datos

-- Entrada: \$sql (Entregas de guardias almacenadas en la base de datos)

--Resultados esperados: Que se inserte correctamente un nuevo empleado al directorio y muestre los datos específicos del mismo

--Resultado obtenido: satisfactorio

3.4.2 Pruebas de caja negra

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. (31)

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. Muchos autores consideran que estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien. (31)

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar. (31)

Con el objetivo de comprobar que las funcionalidades del sistema se realizan de forma correcta y responden a las necesidades del cliente se realizaron un total de 2 iteraciones de pruebas de caja negra, obteniéndose los siguientes resultados:



Ilustración 9 Cantidad de no conformidades obtenidas de la prueba de caja negra

En la primera iteración se detectaron 2 no conformidades (NC), 2 errores de validación. En la segunda iteración se obtuvieron los resultados satisfactorios al no detectarse no conformidades.

3.5 Conclusiones

La aplicación de las técnicas de validación de requisitos permitieron hacer correspondencia entre las solicitudes del cliente y los requisitos obtenidos. Además, el uso de la métrica: (calidad de la especificación) proporcionó una medida de la calidad de la especificación de estos, destacando las ambigüedades existentes para su corrección. La validación del diseño mediante las métricas TOC y RC permitieron obtener, de forma general, el grado de complejidad de implementación, mantenimiento, de responsabilidad, reutilización, acoplamiento y la cantidad de pruebas necesarias en las clases favoreciendo la creación de un diseño flexible, de fácil mantenimiento e implementación y que promoviera la reutilización. Además, las pruebas de Caja Blanca sirvieron para comprobar internamente las funciones del sistema, facilitando la detección de no conformidades para su corrección y las pruebas de Caja Negra permitieron comprobar que las funciones son viables a través de la interfaz del software, que la entrada se acepta de forma adecuada y se produce un resultado correcto, manteniendo así la integridad de la información externa.

Conclusiones generales

Con el desarrollo del presente trabajo se obtuvo una aplicación web capaz de proveer los servicios necesarios para el proceso de entrega de guardia administrativa del Hospital Provincial de Cienfuegos y lograr una rápida y fácil implementación del mismo para lo cual:

- El estudio de las entregas de guardia realizadas permitió conocer la carencia de sistemas que informaticen el proceso de entrega de guardia administrativa.
- La necesidad de organizar de forma correcta las tareas a desarrollar, trajo consigo realizar un estudio acerca de la metodología a utilizar de forma tal que guíe el proceso de desarrollo de la propuesta de solución.
- El estudio de las herramientas y tecnologías establecidas, permitió profundizar los conocimientos necesarios para el desarrollo del sistema para la entrega de guardia administrativa.
- La aplicación de técnicas para la captura de requisitos posibilitó identificar las funcionalidades a partir de las necesidades del cliente, además la especificación de requisitos brindó una mayor claridad de las funcionalidades que se debían incluir.
- Las aplicaciones de los patrones de diseño garantizan lograr un esquema del mismo de fácil entendimiento, extensión y mantenimiento, además de permitir la reutilización de componentes en aplicaciones futuras

- La generación de los diagramas correspondientes al diseño de la aplicación facilitó la comprensión de las relaciones entre las clases presentes en el modelo.
- La realización de pruebas y técnicas de validación permitieron asegurar que la aplicación desarrollada cumpla con los estándares y se ofrezca al cliente un producto de calidad

Como resultado, se tiene una aplicación web capaz de proporcionar resultados satisfactorios con respecto a este proceso, permitiendo la rápida y fácil entrada y salida de datos del mismo para mejorar el orden administrativo y ayudar en la toma de decisiones.

Referencias Bibliográficas

- [1] «La importancia de las tics en el mundo actual – estefany289». <https://estefany289.wordpress.com/la-importancia-de-las-tics-en-el-mundo-actual/> (accedido 10 de junio de 2023).
- [2] «TIC en América Latina». <https://www.caf.com/es/actualidad/noticias/2014/04/el-avance-de-las-tic-en-america-latina/> (accedido 10 de junio de 2023).
- [3] Y. M. Carbo, «Cuba y el impacto de las TIC en la informatización de la sociedad», *Monografias.com*, 14 de junio de 2016. <https://www.monografias.com/trabajos109/cuba-y-impacto-tic-informatizacion-sociedad/cuba-y-impacto-tic-informatizacion-sociedad> (accedido 10 de junio de 2023).
- [4] «Directorio de Instituciones de Salud de Cuba - Hospital General Universitario Dr. Gustavo Aldereguía Lima». <http://dirinstituciones.sld.cu/index.php?P=FullRecord&ID=4645> (accedido 10 de junio de 2023).
- [5] O. Mar Cornelio, N. Sanchez Muguercia, B. Bron Fonseca, y P. M. Puig Díaz, «Sistema de entrega de guardia para los laboratorios», *Ser. Científica Univ. Las Cienc. Informáticas*, vol. 11, n.º 10, pp. 1-13, 2018.
- [6] M. A. Escobar Couto, «Desarrollo del módulo Técnicos de Laboratorio del sistema para la gestión de los procesos administrativos SIGPA del centro de Informatización de Entidades», bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 3. Centro de Informatización de Entidades, 2017. Accedido: 21 de junio de 2023. [En línea]. <https://repositorio.uci.cu/jspui/handle/123456789/9327>
- [7] J. P. Soto Paredes, «Evaluación de la gestión administrativa en centros médicos auspiciados por organismos internacionales, caso Club Rotario», masterThesis, Universidad Andina Simón Bolívar, Sede Ecuador, 2011. Accedido: 21 de junio de 2023. [En línea]. <http://repositorio.uasb.edu.ec/handle/10644/2960>
- [8] C. Vilela Álava, «Desarrollo e implementación de un sistema de gestión administrativa para el hotel Dorado del cantón Playas.», bachelorThesis, La Libertad: Universidad Estatal Península de Santa Elena, 2014., 2014. Accedido: 21 de junio de 2023. [En línea]. <https://repositorio.upse.edu.ec/handle/46000/1580>
- [9] «Amion En Espagnol». <http://www.amion.com/cgi-bin/ocs?Page=Help:209> (accedido 20 de junio de 2023).
- [10] «About Us | ShiftWizard Healthcare and Nurse Scheduling Software», *ShiftWizard*. <https://www.shiftwizard.com/about-us/> (accedido 20 de junio de 2023).
- [11] «AUP Ingeniería de Software». https://ingenieriadesoftware.mex.tl/63758_aup.html (accedido 16 de junio de 2023).
- [12] «Qué es el lenguaje unificado de modelado (UML)», *Lucidchart*. <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml> (accedido 21 de junio de 2023).
- [13] «Arquitectura Cliente-Servidor». <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor> (accedido 16 de junio de 2023).
- [14] «Patrón de diseño MVC. ¿Qué es y cómo puedo utilizarlo? | Easy App CODE». <https://www.easyappcode.com/patron-de-diseno-mvc-que-es-y-como-puedo-utilizarlo> (accedido 16 de junio de 2023).
- [15] «Qué es Visual Studio Code y qué ventajas ofrece», *OpenWebinars.net*, 22 de julio de 2022. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/> (accedido 14 de junio de 2023).
- [16] «Características de Enterprise Architect herramienta de modelado UML». <http://www.sparxsystems.com.ar/products/ea/features.php> (accedido 14 de junio de 2023).

- [17] «Enterprise Architect», *prezi.com*. <https://prezi.com/yhx-2wlbez6b/enterprise-architect/> (accedido 14 de junio de 2023).
- [18] C. Vilches, «Biblioguias: Zotero y recursos CEPAL: Zotero». <https://biblioguias.cepal.org/c.php?g=159506&p=1044394> (accedido 14 de junio de 2023).
- [19] «Características de Zotero», *prezi.com*. <https://prezi.com/p/iqie-3ezox98/caracteristicas-de-zotero/> (accedido 14 de junio de 2023).
- [20] J. de A. Institute, «¿Qué es PHP y para qué sirve?», *Assembler Institute*, 15 de marzo de 2022. <https://assemblerinstitute.com/blog/que-es-php/> (accedido 16 de junio de 2023).
- [21] «Qué es HTML». <https://desarrolloweb.com/articulos/que-es-html.html> (accedido 14 de junio de 2023).
- [22] «> HTML: Qué es y para qué sirve - VADAVO», 21 de marzo de 2022. <https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/> (accedido 14 de junio de 2023).
- [23] «Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes». <https://blog.hubspot.es/website/que-es-css#que-es> (accedido 14 de junio de 2023).
- [24] M. Coppola, «Qué es JavaScript, para qué sirve y cómo funciona». <https://blog.hubspot.es/website/que-es-javascript> (accedido 14 de junio de 2023).
- [27] «¿Qué es MySQL? - Definición en Whatls.com», *ComputerWeekly.es*. <https://www.computerweekly.com/es/definicion/MySQL> (accedido 16 de junio de 2023).
- [28] J. M. V. Pineda, «Qué es y qué nos ofrece la herramienta phpMyAdmin», *CoriaWeb Hosting*, 28 de junio de 2016. <https://www.coriaweb.hosting/nos-ofrece-phpmyadmin/> (accedido 16 de junio de 2023).
- [29] Marco de la junta de Andalucía. <https://www.juntaandalucia.es/servicios/madeja/contenido/recurso/419>.
- [30] Departamento de lenguajes y sistemas informáticos. <https://www.isi.ugr.es>
- [31] ZXQ <https://zxq.net/what-happened-to-the-old-zxq-website/>