

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniera e Ingeniero de Sistemas**

**TEMA:
PLATAFORMA DE GESTIÓN DE CAMAS E INDICADORES DE CALIDAD
PARA EL HOSPITAL GINECO OBSTÉTRICO PEDIÁTRICO DE NUEVA
AURORA LUZ ELENA ARISMENDI**

**AUTORES:
Evelyn Michelle Cadena Panchi
Juan Carlos Perugachi Velasco**

**TUTOR:
Daniel Giovanni Díaz Ortiz**

Quito, septiembre de 2020

CESIÓN DE DERECHOS DE AUTOR

Nosotros, JUAN CARLOS PERUGACHI VELASCO, con documento de identificación N° 1721731337, y EVELYN MICHELLE CADENA PANCHI, con documento de identificación N° 1725422800, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “PLATAFORMA DE GESTIÓN DE CAMAS E INDICADORES DE CALIDAD PARA EL HOSPITAL GINECO OBSTÉTRICO PEDIÁTRICO DE NUEVA AURORA LUZ ELENA ARISMENDI”, mismo que ha sido desarrollado para optar por el título de INGENIERO E INGENIERA DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, subscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



JUAN CARLOS PERUGACHI

VELASCO

CI: 1721731337



EVELYN MICHELLE CADENA

PANCHI

CI: 1725422800

Quito, septiembre del 2020

DECLARATORIA DE COAUTORIA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: PLATAFORMA DE GESTIÓN DE CAMAS E INDICADORES DE CALIDAD PARA EL HOSPITAL GINECO OBSTÉTRICO PEDIÁTRICO DE NUEVA AURORA LUZ ELENA ARISMENDI, realizado por EVELYN MICHELLE CADENA PANCHI Y JUAN CARLOS PERUGACHI VELASCO, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, septiembre del 2020



.....

Daniel Giovanni Díaz Ortiz

Ci: 1716975501

DEDICATORIA

Dedico este trabajo a mis padres Luis y Nancy, por toda su dedicación, amor y apoyo en cada momento de mi vida, me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mí el ejemplo de esfuerzo y perseverancia.

A mis hermanos por ser ese consejo, ese apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias.

A toda mi familia y amigos, con sus oraciones y sus palabras de aliento hicieron de mí una mejor persona y de una u otra manera forman parte en todos mis sueños y metas.

Juan Carlos Perugachi Velasco

DEDICATORIA

Este trabajo va dedicado a mis queridos padres Alfredo Cadena y Piedad Panchi que siempre han dado todo su esfuerzo, sacrificio, dedicación y cariño para hoy culminar esta etapa de mi vida, siempre me impulsaron y fueron mi inspiración para llegar hasta el final y poder ser digna del orgullo que sienten por mí. A mis hermanos por el apoyo y el ánimo que me han ofrecido todo este tiempo.

Y como no dedicarlo a mi pequeña familia que Dios me ha regalado, gracias, amor por tu apoyo, por las palabras de aliento en los momentos difíciles. A mi pequeña que forja en mí el deseo de superarme todos los días para transmitirle todas las enseñanzas y cariño que yo he recibido.

Evelyn Michelle Cadena Panchi

AGRADECIMIENTO

Al finalizar este trabajo quiero utilizar este espacio para agradecer a Dios por todas sus bendiciones, a mis padres que han sabido darme su ejemplo de dedicación y esfuerzo, finalmente a mis hermanos por su apoyo incondicional en todo momento.

De igual manera mis agradecimientos a la Universidad Politécnica Salesiana, a la facultad de Ingeniería en Sistemas y en especial a todos y cada uno de los docentes que han formado parte de mi formación académica, quienes con la enseñanza de sus valiosos conocimientos hicieron que cada día pueda crecer humana y profesionalmente.

Finalmente quiero expresar mis más sinceros agradecimientos al Ing. Daniel Diaz, principal colaborador durante todo este proceso, quien con su dirección, conocimiento, enseñanza y apoyo permitió el desarrollo de este trabajo.

Juan Carlos Perugachi Velasco

AGRADECIMIENTO

Dios ha sido la base fundamental de mi vida y merece todo el agradecimiento por brindarme vida y salud para culminar una de mis metas. A mis padres por el sacrificio y su apoyo incondicional en cada momento de mi vida.

Agradezco a la Universidad Politécnica Salesiana que contribuyo en mi formación profesional, a todos los profesores de la Carrera que compartieron sus conocimientos y sabiduría.

Un agradecimiento especial a mi tutor el Ingeniero Daniel Díaz que bajo su tutela ha encaminado este proyecto y ha sido posible la culminación. A mi compañero de proyecto Juan Perugachi quien con su amistad y apoyo se pudo realizar un trabajo en equipo para culminar con éxito nuestro proyecto.

Evelyn Michelle Cadena Panchi

ÍNDICE

INTRODUCCION _____	1
Antecedentes _____	1
Sistemas relacionados _____	2
Justificación _____	3
Objetivos _____	5
Objetivo general: _____	5
Objetivos específicos: _____	5
Marco metodológico _____	6
Cuadro comparativo de las metodologías SCRUM, XP y RUP _____	6
CAPITULO I MARCO TEÓRICO _____	8
MARCO INSTITUCIONAL _____	8
Servicios: _____	8
MARCO TEÓRICO _____	10
Ingeniería del Software _____	10
Modelo Vista Controlador _____	10
Arquitectura Cliente Servidor _____	11
Servidor Apache _____	11
Herramientas de Desarrollo _____	12
Librerías y Tecnologías Adicionales: _____	15
METODOLOGÍA XP _____	17
FASES DE LA METODOLOGÍA _____	17
CAPITULO II ANÁLISIS Y DISEÑO _____	19
REQUISITOS FUNCIONALES _____	19
REQUISITOS NO FUNCIONALES _____	22
Usuarios y Perfiles _____	24
Historias de Usuario _____	24
Diagramas de casos de Uso (CU) _____	32
CU: Perfil Administrador _____	32
CU: Perfil Admisión Emergencia _____	32
CU: Perfil Atención Usuario _____	33
CU: Perfil Enfermería _____	33
CU: Creación de nueva cuenta _____	33

CU: Ingreso al sistema _____	34
CU: Gestión de perfil _____	34
CU: Gestión de permisos _____	34
CU: Recuperación de contraseña _____	35
CU: Registro paciente _____	35
CU: Editar paciente _____	35
CU: Gestión asignar cama _____	36
CU: Gestión de cambio/egreso _____	36
CU: Generar Reporte _____	36
CU: Visualizar datos _____	37
DIAGRAMAS DE SECUENCIA (DS) _____	37
DS: Crear una nueva cuenta _____	37
DS: Ingreso al sistema _____	38
DS: Gestión de perfil _____	38
DS: Gestión de permisos _____	39
DS: Recuperación de contraseña _____	39
DS: Registro paciente _____	40
DS: Editar paciente _____	40
DS: Gestión asignar cama _____	41
DS: Gestión de cambio/egreso _____	41
DS: Visualizar datos pacientes _____	42
DS: Generar reportes _____	42
Diagrama de navegación _____	43
Diagrama de navegación: perfil administrador _____	43
Diagrama de navegación: perfil atención usuario _____	43
Diagrama de navegación: perfil admisión emergencia _____	44
Diagrama de navegación: perfil enfermería _____	44
Diseño conceptual de base de datos _____	45
CAPÍTULO III CONSTRUCCIÓN Y PRUEBAS _____	46
Diagrama de despliegue: _____	46
Código importante: _____	47
Pruebas: _____	49
Pruebas de Caja Negra _____	49

Pruebas de Carga	58
Pruebas de Estrés	60
Análisis de resultados de las pruebas	60
Pruebas de caja negra	60
Pruebas de carga	61
Pruebas de estrés	62
CONCLUSIONES	67
RECOMENDACIONES	68
REFERENCIAS	70

ÍNDICE DE TABLAS

Tabla 1. Cuadro comparativo de Metodologías	6
Tabla 2. Distribución de camas	8
Tabla 3. Requisitos Funcionales	19
Tabla 4. Requisitos No Funcionales	22
Tabla 5. Perfiles de usuario y descripción	24
Tabla 6. Proceso para Creación de nueva cuenta	24
Tabla 7. Proceso para ingresar al sistema	25
Tabla 8. Proceso para la gestión de perfiles.	25
Tabla 9. Proceso para asignar permisos al usuario	26
Tabla 10. Proceso para recuperar la contraseña	27
Tabla 11. Proceso para registrar un paciente	27
Tabla 12. Proceso para la gestión de entidades	28
Tabla 13. Proceso para asignar una cama	28
Tabla 14. Proceso para generar reportes	29
Tabla 15. Proceso para cambio o egreso de paciente	30
Tabla 16. Proceso para visualizar datos	31
Tabla 17. Descripción para generar la Historia Clínica Única	48
Tabla 18. Significado de cada atributo de las clases de equivalencia	50
Tabla 19. Clase de Equivalencia para la creación de nueva cuenta	50
Tabla 20. Caso de prueba para creación de cuenta.	51
Tabla 21. Clase equivalente para: Ingreso al sistema	52
Tabla 22. Caso de prueba para ingreso al sistema	53
Tabla 23. Clase equivalencia para: Gestión asignar cama	54
Tabla 24. Caso de Prueba para Gestión asignar cama	54
Tabla 25. Clase equivalente para: Gestión de cambio/egreso	55
Tabla 26. Caso de prueba gestión cambio/egreso	55
Tabla 27. Clase equivalente para: Generar Reporte	57
Tabla 28. Caso de prueba para Generar reportes	57
Tabla 29. Características del hardware donde se realizaron las pruebas de carga y estrés.	58
Tabla 30. Peticiones Http de las que se tomó las muestras.	58
Tabla 31. Parámetros del temporizador	59
Tabla 32. Resultado de las pruebas de carga	59
Tabla 33. Resultados de las pruebas de estrés	60
Tabla 34. Resultados de las pruebas de caja negra	61

ÍNDICE DE FIGURAS

Figura 1. Flujograma de proceso de asignación de cama	9
Figura 2. Esquema de funcionamiento de PHP	12
Figura 3. CU: perfil administrador	32
Figura 4. CU: Perfil Admisión emergencia	32
Figura 5. CU: perfil atención usuario	33
Figura 6. CU: Perfil Enfermería	33
Figura 7. CU: creación de nueva cuenta.....	33
Figura 8. CU: Ingreso al sistema	34
Figura 9. CU: Gestión de Perfil	34
Figura 10. CU: Gestión de Permisos	34
Figura 11. CU: Recuperación de contraseña	35
Figura 12. CU: Registro de Paciente	35
Figura 13. CU: Editar paciente	35
Figura 14. CU: Gestión asignar cama.....	36
Figura 15. CU: Gestión de cambio/egreso	36
Figura 16. CU: Generar Reporte	36
Figura 17. CU: Visualizar datos	37
Figura 18. DS: crear cuenta nueva.....	37
Figura 19. DS: Ingreso al sistema.....	38
Figura 20. DS: Gestión de Perfil	38
Figura 21. DS: Gestión de Permisos.....	39
Figura 22. DS: Recuperación de Contraseña.....	39
Figura 23. DS: Registro Paciente	40
Figura 24. DS: Editar paciente	40
Figura 25. DS: Gestión asignar de cama	41
Figura 26. DS: Gestión de cambio/egreso	41
Figura 27. DS: Visualizar datos pacientes.....	42
Figura 28. DS: Generar reportes.....	42
Figura 29. Diagrama de navegación: perfil administrador	43
Figura 30. Diagrama de navegación: perfil atención usuario	43
Figura 31. Diagrama de navegación: perfil admisión emergencia	44
Figura 32. Diagrama de navegación: perfil enfermería.....	44
Figura 33. Diseño conceptual de base de datos	45
Figura 34. Diagrama de despliegue	46
Figura 35. Algoritmo para asignar una Historia Clínica	48
Figura 36. Temporizador Aleatorio Uniforme Fuente: Apache JMeter	59
Figura 37. % de Error	62
Figura 38. Rendimiento	63
Figura 39. Tiempo de respuesta máximo.....	63
Figura 40. Rendimiento vs %error	64
Figura 41. Comportamiento de los recursos de hardware a los 100 usuarios	65
Figura 42. Comportamiento de los recursos de hardware a los 400 usuarios	65
Figura 43. Colapso del sistema.....	66

Resumen

Actualmente en las instituciones de salud pública del Ecuador es evidente la falta de un sistema de información que automatice y mejore el proceso de asignación y gestión de camas, por lo que el presente proyecto surge para cubrir con dicha necesidad.

El diseño y desarrollo de la Plataforma de gestión de camas e indicadores de calidad para el Hospital Gineco Obstétrico Pediátrico de Nueva Aurora Luz Elena Arismendi permite la automatización y estandarización de los procesos del departamento de Hospitalización y Estadística, optimizando recursos y brindando un servicio eficiente a pacientes y familiares.

La utilización de la metodología XP (Programación Extrema), fue fundamental para el desarrollo del presente proyecto, ya que se adapta completamente a las necesidades planteadas y permite la interacción constante y directa con el cliente. La planificación, codificación en pareja y pruebas constantes ayudaron a detectar posibles problemas a tiempo permitiendo iteraciones ágiles durante toda la etapa del proyecto.

Este sistema ayuda a emplear el flujo correcto del proceso para la asignación de camas, así como las funciones que el personal debe cumplir para brindar un servicio de calidad, además el sistema genera reportes y gráficas de acuerdo con la necesidad del usuario para el respectivo análisis y toma de decisiones.

Palabras claves: Gestión de camas, Programación Extrema, Zend Framework.

Abstract

Actually, Ecuador's public health institutions clearly lack an information system that would automate and improve the process of bed allocation and management, and this project was created to address this need.

The design and development of the Platform for bed management and quality indicators for the Pediatric Obstetrics and Gynecology Hospital of Nueva Aurora Luz Elena Arismendi allows the automation and standardization of the processes of the Hospitalization and Statistics Department, optimizing resources and providing an efficient service to patients and families.

The use of the XP (Extreme Programming) methodology was fundamental for the development of this project since it is completely adapted to the proposed needs and allows constant and direct interaction with the client. Planning, pair coding, and constant testing helped detect potential problems early, allowing for agile iterations throughout the project stage.

This system helps to employ the correct process flow for bed allocation, as well as the functions that the personnel must fulfill to provide a quality service. In addition, the system generates reports and graphs according to the user's needs for the respective analysis and decision making.

INTRODUCCION

Antecedentes

EL HOSPITAL GINECO OBSTETRICO PEDIATRICO DE NUEVA AURORA LUZ ELENA ARISMENDI, se encuentra en funcionamiento desde el año 2014 , es catalogado como un establecimiento de salud de tercer nivel de atención y tiene como objetivo ofrecer servicios de salud con calidad y calidez a sus pacientes (Ecuador, 2014).

El área de hospitalización cuenta con 172 camas y realiza procesos como el ingreso de pacientes, trabaja con el plan “hospital del día”, registran los procedimientos médicos asignados al paciente, ejecutan traslados o interconsultas, dan de alta a los pacientes y permiten la consolidación de datos. Sin embargo, existe una incorrecta organización para la asignación de camas de los servicios mencionados, no cumplen con un flujo correcto para este proceso y el personal no realiza sus funciones asignadas.

Cuando un paciente necesita ser hospitalizado debería ser el departamento de estadística el encargado de asignar una cama verificando en el sistema la disponibilidad de camas que poseen y posteriormente comunicar al personal de enfermería para que ubiquen al paciente en la cama asignada, pero este proceso no se cumple por la carencia de dicho sistema, actualmente este proceso se realiza de forma contraria, la asignación de cama es realizado por el personal de enfermería, ya que ellos poseen información de las todas las camas y posteriormente se informa al departamento de estadística para generar el registro del paciente.

Con el paso del tiempo la cantidad de pacientes ha ido incrementando por lo tanto la organización que se tiene para la asignación de camas dejó de ser óptima al no poseer un proceso automatizado para el servicio del área de hospitalización, ya que la información de dicho proceso se maneja manualmente. La asignación de manera manual no se encuentra exento del error humano por lo que puede existir una confusión entre las camas disponibles y las que se

encuentran ocupadas. El mal manejo de estas actividades provoca que los pacientes que deban ser internados tengan que esperar la confirmación de un espacio disponible y adecuado para su estancia en el hospital, lo que suscita molestias tanto en los pacientes como en los familiares.

Sistemas relacionados

Se ha podido observar que actualmente tanto en Ecuador como en otros países existen sistemas parecidos al que se propone en este documento, se referenciará a tres de estos sistemas y se lo comparará con el sistema propuesto para observar sus diferencias y poder justificar el motivo del desarrollo de la plataforma propuesta. En la universidad del Salvador estudiantes de la carrera de sistemas como proyecto de titulación desarrollaron un SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE SERVICIOS MÉDICOS EN LA UNIDAD DE HEMATO-ONCOLOGÍA DEL HOSPITAL NACIONAL ROSALES (AGUILAR RIVERA, MENDOZA MENCOS, & TOBIAS RIVAS, 2013). Este sistema tenía como finalidad realizar los siguientes procesos: “programar citas de quimioterapia, registrar procedimientos aplicados, manejar el movimiento de pacientes hospitalizados en la unidad, registrar las interconsultas brindadas por los médicos” (AGUILAR RIVERA, MENDOZA MENCOS, & TOBIAS RIVAS, 2013). Estudiantes de la universidad autónoma de México realizaron un SISTEMA DE INFORMACIÓN HOSPITALARIO, su función estaba dirigida a la automatización de los procesos de admisión de los pacientes y el de hospitalización. (SÁNCHEZ, NAVA, & PEREYRA, 2013). Del mismo modo en Ecuador en la provincia de Cotopaxi estudiantes realizaron un “SOFTWARE DE MANEJO DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS MÉDICAS PARA LA CLÍNICA DE LA FUERZA AÉREA ECUATORIANA DEL ALA No. 11 DE QUITO” (MOLINA CALVOPIÑA & COLLAGUAZO LOACHAMIN, 2008). El propósito de este trabajo era “automatizar las historias clínicas de sus pacientes y permitir el ingreso desde cualquier lugar para tomar una

cita médica en la especialidad que requiera el usuario” (MOLINA CALVOPÍÑA & COLLAGUAZO LOACHAMIN, 2008).

Actualmente existen varios sistemas para el área de salud, algunos abarcan todo el funcionamiento que posee un hospital como citas médicas, diagnósticos, exámenes de laboratorio, ingreso y egreso de pacientes; otros sistemas se enfocan en alguna área específica dando prioridad a ciertos procesos. Cada uno de estos sistemas fue diseñado y construido según las necesidades de sus clientes teniendo todos como objetivo principal la automatización de procesos y generando reportes para una mejor toma de decisiones.

El sistema propuesto está dirigido solamente al área de hospitalización ya que actualmente el hospital cuenta con sistemas informáticos para el agendamiento de citas médicas y para el área de laboratorio con los cuales ha trabajado este tiempo. Se ha analizado los requerimientos del usuario y como función principal se necesita la automatización del proceso de asignación de camas, con esto se busca tener un flujo correcto de las funciones de su personal y con los reportes que generará el sistema se podrá tener indicadores los cuales ayudarán a mejorar el servicio hospitalario. Este proyecto deberá adaptarse a los sistemas que ya posee el hospital, por ello deberá existir una comunicación y consolidación de la información.

Como un plus el sistema propuesto consta de una interfaz informativa con el fin de mostrar a los familiares sobre la ubicación de los pacientes tanto en los servicios, salas, habitaciones y camas que se encuentre, esta información se reflejará en los monitores de las salas de espera.

Justificación

La problemática del HOSPITAL GINECO OBSTETRICO PEDIATRICO DE NUEVA AURORA LUZ ELENA ARISMENDI, tiene origen en la carencia de una plataforma para la gestión de camas a los pacientes que necesiten hospitalización, la asignación de responsabilidades al personal no se encuentra bien definida ya que dicho proceso no se

encuentra optimizado, por lo tanto, se produce un mal manejo de la información, y una inadecuada automatización de los datos. La información obtenida es la principal fuente para obtener indicadores de salud, la toma de decisiones y brindar información acertada y oportuna de los pacientes a sus familiares.

El funcionamiento de un hospital o de cualquier empresa depende de la buena administración de todos sus recursos, siendo la información el recurso más valioso que posee cada institución pública o privada y todos los demás recursos giran y dependen de ella. La demanda de una información precisa y oportuna está creciendo cada vez por ello en la actualidad se emplea el uso de nuevas tecnologías de la información con nuevas herramientas para almacenamiento y procesamiento de datos.

El desarrollo de este proyecto de titulación permitirá incorporar nuevos procedimientos para el monitoreo y asignación de camas a los pacientes que necesiten hospitalización de una manera ágil, recopilando y almacenando la información, con la posibilidad de generar reportes estadísticos, eliminando tareas innecesarias, mejorando la calidad del servicio de hospitalización, llevando un flujo adecuado de los procesos que se manejan en el área de hospitalización y haciendo cumplir las funciones asignadas de forma correcta al personal.

Los beneficios de contar con un sistema informático para la gestión de camas a través de una mejor estructuración de la información podrá proporcionar mayor fluidez en la asignación de camas, mejor organización para acceder rápidamente a los diagnósticos del paciente, menos frecuencia de errores a causa de información ilegible, reducción del uso de papel y la redundancia de la información, un considerable ahorro de tiempo en cada proceso, favorecer en la programación de los recursos y en la toma de decisiones, información actualizada y oportuna para construir los indicadores de calidad.

Se busca a través de este proyecto que se realice una automatización en los procesos, y establecer los flujos correctos del proceso para una mejor organización, de manera que todas las capacidades del sistema sean aprovechadas adecuadamente.

Objetivos

Objetivo general:

Analizar, diseñar y desarrollar una plataforma de gestión de camas e indicadores de calidad para el Hospital Gineco Obstétrico Pediátrico de Nueva Aurora Luz Elena Arismendi.

Objetivos específicos:

Analizar el proceso habitual para la gestión de camas mediante la aplicación de entrevistas al personal involucrado con el propósito de obtener las entradas y variables necesarias para el desarrollo del sistema.

Diseñar el modelo conceptual, lógico y físico de la base de datos que almacenará la información obtenida mediante el análisis de entradas y variables del proceso de asignación de camas.

Mantener la seguridad de la información mediante el uso de sesiones, perfiles y roles de usuario de tal manera que sus actividades estén definidas y delimitadas.

Implementar un módulo de gestión del sistema para cada una de las entradas y variables que lo conforman.

Proveer de un módulo de ingreso, alta, asignación y traslado de cama para pacientes de acuerdo con las necesidades del usuario.

Generar reportes estadísticos con información relevante del proceso de gestión de camas de acuerdo con las necesidades del usuario.

Construir una interfaz informativa para la sala de espera del hospital que permita mostrar la ubicación del paciente de entre los diferentes servicios a sus familiares.

Marco metodológico

Cuadro comparativo de las metodologías SCRUM, XP y RUP

Para el desarrollo de un proyecto de software se debe realizar un análisis para conocer cuál es la metodología adecuada que se debe aplicar a dicho proyecto, el grupo de trabajo debe tener en consideración el proceso para la construcción del software y los requerimientos del cliente.

Tabla 1. Cuadro comparativo de Metodologías

CARACTERÍSTICAS	RUP	SCRUM	XP
Permite el desarrollo de la aplicación en parejas			X
Es iterativo		X	X
Analiza los resultados con el cliente		X	X
Capacidad de respuesta a los cambios		X	X
Refactorización del código		X	X
Tiene un cronograma definido	X	X	X
El funcionamiento del software es más primordial que su documentación		X	X
Se enfoca en proyectos a corto plazo		X	X
Simplicidad de código			X
Énfasis en la gestión del recurso humano		X	
Permite el uso de tecnologías	X	X	X
Diseño simple del código			X
Se enfoca en calidad del software	X	X	X
Mitiga los riesgos	X	X	X
Resultados rápidos		X	X
El costo de los cambios es bajo		X	X

Elaborado por: Cadena & Perugachi

Teniendo en cuenta el anterior cuadro comparativo, la metodología que mejor se acopla al proyecto propuesto es la metodología XP(Extreme Programming), dicha metodología se adapta a las necesidades que tiene este proyecto, ya que se encuentra orientada a proyectos de corto plazo, compuesto por pequeños grupos permitiendo el desarrollo de la aplicación en parejas, e interactúa con el cliente haciéndolo parte del equipo. Una de las características principales de

XP es la simplicidad del código, es mejor realizar algo sencillo y añadir cambios sí se lo requiere, a realizar algo muy complejo que quizás nunca se utilizará, además con una adecuada planificación, codificación y pruebas se podrá detectar posibles problemas o desviaciones, esto evitará pérdida de tiempo y ayudará a corregir errores inmediatamente, los costos de los cambios que se puedan presentar no son muy elevados por lo que el presupuesto no sufrirá grandes cambios. La metodología XP permite ir mejorando el código constantemente, ayuda a simplificar las funciones y a eliminar código duplicado, todos estos cambios que se realizan no afectarán el comportamiento del sistema. (Abrahamsson, Salo, & Ronkainen, 2002)

Adicionalmente se emplearán buenas prácticas de las otras metodologías por ejemplo el desarrollo de forma iterativa e incremental llamada Sprint que es una buena práctica de la metodología SCRUM, donde cada Sprint dura entre 2 y 4 semanas y se obtiene una versión de software con nuevas características listas para ser usadas.

CAPITULO I MARCO TEÓRICO

MARCO INSTITUCIONAL

En la Ciudad de Quito el 17 de diciembre de 2014 por acuerdo ministerial se crea la institución de salud denominada “HOSPITAL GINECO OBSTETRICO PEDIATRICO DE NUEVA AURORA LUZ ELENA ARISMENDI” localizado en el sector sur de la capital del Ecuador, por la calidad y capacidad de los servicios que ofrece, se lo considera un hospital de tipo gineco obstétrico de tercer nivel de atención. Se estableció con un total de 172 camas para el servicio de hospitalización (Ecuador, 2014).

Servicios:

El hospital entre sus principales servicios que ofrece está: hospitalización, consulta externa, emergencia, imagenología, laboratorio, farmacia, etc. El sistema propuesto está orientado específicamente al servicio de Hospitalización en la cual se hará énfasis de sus servicios y la capacidad que dispone para la atención de sus pacientes a continuación se detalla la distribución de camas.

Tabla 2. Distribución de camas

SERVICIO	DOTACIÓN NORMAL DE CAMAS	CAMAS DISPONIBLES
GINECOLOGIA	22	22
ALTO RIESGO OBSTETRICO	86	71
OBSTETRICIA ADOLESCENTES	21	21
UCI MATERNA	6	6
UCI PEDIATRICA	3	3
EMERGENCIA	15	15
CENTRO OBSTÉTRICO	24	24
CENTRO QUIRÚRGICO	10	10
TOTAL DE CAMAS	187	172

Nota: Distribución de camas obtenida de los archivos del Hospital

Los departamentos que se verán involucrados en este servicio son:

- Área de emergencia: El paciente puede ingresar a hospitalización por emergencia, el doctor de turno emitirá la orden.
- Área de consulta externa: Si en el transcurso de la consulta el doctor ve la necesidad de hospitalizar a un paciente puede generar la orden para hacerlo.
- Estadística: Con la orden emitida por emergencia o consulta externa el usuario registrado en el sistema le asignará una cama a la especialidad requerida.
- Área de hospitalización: Con la orden de estadística el personal de enfermería que se encuentre de turno será el encargado de ubicar al paciente en su cama asignada.

Proceso de asignación de camas:

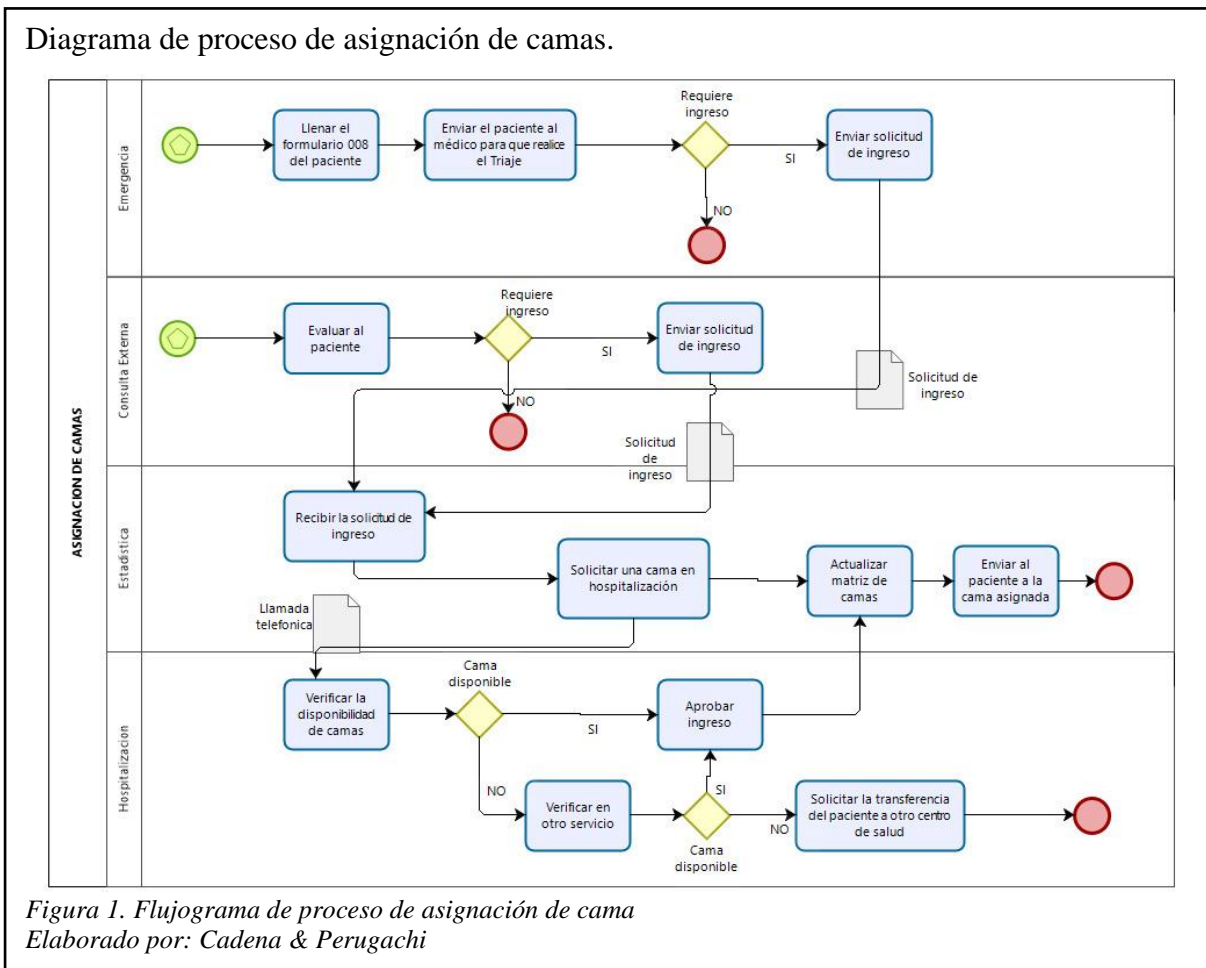


Figura 1. Flujograma de proceso de asignación de cama
Elaborado por: Cadena & Perugachi

MARCO TEÓRICO

Ingeniería del Software

La ingeniería del software es el conjunto de métodos y técnicas para el desarrollo y administración de los programas informáticos, donde se debe poner en práctica los conocimientos de diseño, construcción y documentación para que el software sea confiable.

Los principios de la ingeniería para el desarrollo de estos sistemas están compuestos por aspectos técnicos y no técnicos, con el fin de tener como resultado final un software de calidad, con una funcionalidad eficiente, estableciendo hitos y gestionando de manera adecuada los cambios. (Pressman, 2010)

Modelo Vista Controlador

Se dio a conocer por primera vez en 1979 para el desarrollo de aplicaciones web, se basa en la reutilización del código y en que cada concepto trabaja de manera independiente al separar por capas la lógica del negocio de la funcionalidad (González, 2012). El MVC divide en tres partes a una aplicación web para analizar cada elemento por separado, asignando a cada capa una responsabilidad y obtener un producto de calidad. Las capas son:

- Vista: La página HTML, es la interfaz del usuario.
- Controlador: Responde a las acciones del usuario siendo un intermediario entre el modelo y la vista.
- Modelo: Es la lógica del negocio donde se gestiona todos los accesos a la información que posee la aplicación web almacenada en una base de datos.

El MVC separa la lógica del negocio de la interfaz del usuario para que sean más sencillo las modificaciones y no altere a todo el sistema, además promueve la reutilización de código y reduce el tiempo de mantenimiento. (González, 2012)

Arquitectura Cliente Servidor

La arquitectura cliente servidor fue desarrollada con la finalidad de cubrir las necesidades de cualquier organización para realizar sus funciones de manera ágil y eficiente, la cual procesa la información de la empresa y permite obtener dicha información en tiempo real para tener una oportuna toma de decisiones, esta arquitectura también ayuda a reducir costos de operación y mantenimiento sin la necesidad de afectar la calidad de los servicios.

En esta arquitectura actúan dos participantes el cliente y el servidor. El cliente realiza peticiones de información y el servidor responde a la petición realizadas por el cliente. (R. Orfali, 2002)

La arquitectura que se propone para la solución de este proyecto de titulación está basada en arquitectura cliente servidor en donde el sistema propuesto estará montado en un servidor apache que el hospital posee, y se pueda acceder desde cualquier computador que se encuentre en la misma red de la institución.

Servidor Apache

Apache HTTP Server es un servidor web de aplicaciones gratuito de código abierto que lanzó su primera versión en 1995, desarrollada al principio para servidores de Linux/Unix, pero con el pasar del tiempo y su aceptación se crearon versiones compatibles con sistemas operativos Windows y MacOS.

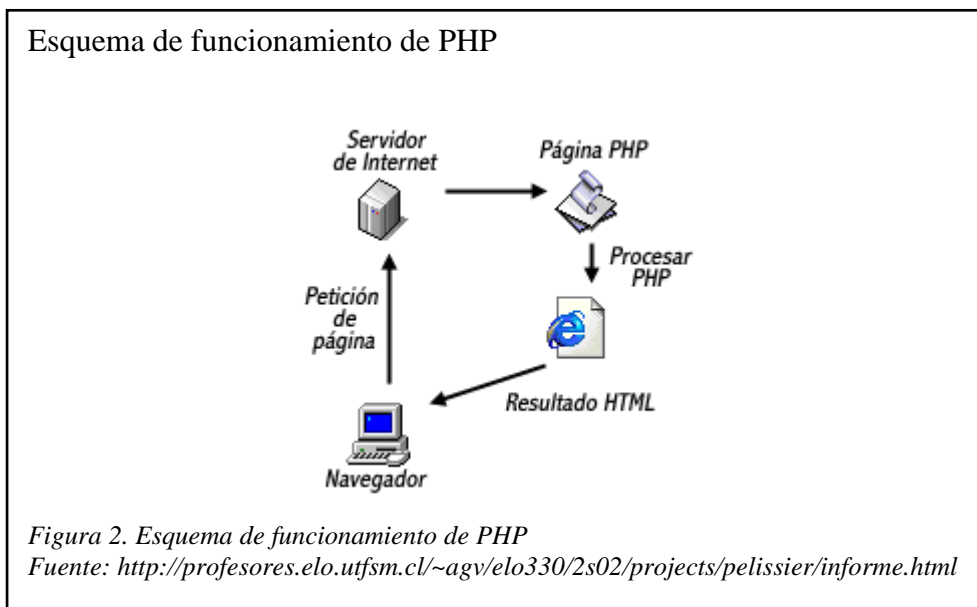
Su finalidad es construir una conexión fluida y segura entre el servidor y los navegadores que los usuarios utilizan para enviar archivos entre ellos mediante la arquitectura cliente-servidor, dicha conexión se establece a través del protocolo http. Cuando un usuario necesita ingresar a un sitio web, el navegador envía una petición al servidor y Apache retorna una respuesta con la información solicitada (B., 2019).

Herramientas de Desarrollo

Lenguaje de programación PHP

Es un lenguaje de programación orientado a objetos de código abierto que posee un servidor gratuito, diseñado para la creación de aplicaciones web dinámicas, puede ser introducido en código HTML y es compatible con diferentes tipos de bases de datos.

Se debe diferenciar entre una página estática la cual tiene contenido permanente y la página dinámica tiene contenido que está en continuo cambio, el usuario realiza una petición la información se procesa y se prepara para dar una respuesta. (Bustos, 2019)



El lenguaje PHP se caracteriza por:

1. Velocidad: PHP no consume demasiados recursos del sistema y se integra sin ningún problema con otras aplicaciones (Bustos, 2019).
2. Estabilidad: PHP posee un sistema sólido y estable, cuenta con un propio sistema de gestión de recursos y tiene herramientas para la administración de variables (Bustos, 2019).

3. Seguridad: PHP no permite a los clientes ver su código, el servidor es el encargado de ejecutar el código y da como resultado una página HTML (Bustos, 2019).
4. Simplicidad: Un lenguaje fácil de aprender, los desarrolladores que tengan experiencia en lenguaje C o C++ se podrán acoplar fácilmente a PHP ya que poseen una sintaxis similar. (Bustos, 2019)

Bases de Datos

Es un conjunto de datos que se encuentran relacionados entre sí y que por medio de esta se podrán realizar consultas, actualizaciones o modificaciones, inserciones o eliminaciones de datos (Mariscal, 2015).

Modelo Entidad Relación: Es un método que permite diseñar la estructura lógica de la base de datos de una organización mediante un diagrama compuesto de diferentes elementos y simbologías. Está compuesta de objetos que representan cosas del mundo real llamados entidades los cuales tienen sus atributos, propiedades y relaciones con otros objetos.

Conceptos básicos de este modelo:

- *Entidad:* Es un objeto o una cosa de la vida real que es único, el cual puede ser identificado, poseer atributos y almacenar información. Ejemplo cliente, usuario.
- *Atributo:* Representan las características de una entidad. Ejemplo nombre, dirección.
- *Relación:* Es la encargada de asociar una o más entidades compartiendo atributos de forma indispensable.

Normalización: Es un proceso que permite detectar y prevenir posibles problemas que se pueden presentar al construir el diseño de la base de datos, estos inconvenientes

se los resolverá mediante la aplicación de reglas y normas a las relaciones realizadas en el modelo Entidad-Relación.

Tiene como objetivo elaborar un buen diseño lógico para evitar la repetición de datos, tener una adecuada actualización en la base de datos y protección de estos aplicando las siguientes formas normales:

- *Primera forma normal (1FN)*: Una relación se encuentra en la primera forma normal al no poseer grupos reiterados, es decir que sus atributos no tengan más de un valor.
- *Segunda forma normal (2FN)*: Una relación cumple con la segunda forma normal estando en la 1FN y todos sus atributos deben ser dependientes de la llave primaria compuesta.
- *Tercera forma normal (3FN)*: Una relación cumple con la tercera forma normal estando en la 2FN y que no existan dependencias funcionales entre atributos que no son claves primarias.

PostgreSQL: Es un gestor de base de datos de código abierto orientado a objetos, se puede instalar y utilizar las veces que sean necesarias. Es de fácil acceso, la instalación y configuración dependen del hardware disponible y además está disponible para la mayoría de los sistemas operativos sin afectar su rendimiento (Silberschatz, Korth, & Sudarshan, 2002).

Zend Framework

El término “framework” se lo debe considerar como marco de desarrollo que estandariza y brinda una estructura de desarrollo definida, su objetivo principal es brindar orden, facilidad y agilidad en la creación de aplicaciones.

Zend framework es un framework basado en el lenguaje de programación PHP destinado al desarrollo de aplicaciones web, utiliza el 100% de código orientado a objetos y está destinado al desarrollo de pequeños y grandes proyectos.

Cuenta con soporte de internacionalización para captar el interés de más personas ya que ofrece construir sitios web que posean varios idiomas, además convertir formatos de fechas y monedas para tener un enfoque global. (Zend, s.f.)

Librerías y Tecnologías Adicionales:

AJAX: Es una tecnología asíncrona para el desarrollo web para crear aplicaciones interactivas, es válida para diversas plataformas y se la utiliza en diferentes sistemas operativos y navegadores ya que se basa en estándares abiertos. Dicha aplicación se ejecuta en el cliente por tanto no saturan el servidor, solamente se recargará los datos necesarios sin necesidad de actualizar todo el documento, esto evitará el consumo excesivo de recursos, aumentará la velocidad y mejorará la interacción del usuario con la aplicación (Mesa, Sanz, & Granada, 2014).

GitHub: Es una plataforma de código abierto destinada al alojamiento de repositorios y proyectos, brinda herramientas para el trabajo en equipo, utiliza el sistema de control de versiones para administrar los proyectos y evitar la pérdida de código o confusiones, si se actualiza el código de un proyecto GitHub guardará las versiones anteriores y el usuario podrá restaurarlos cuando desee para comparar los cambios realizados. (Fm, 2019)

NPM: Es el gestor de paquetes por defecto para Node.js, ayuda a gestionar los módulos, distribuir paquetes y añadir dependencias fácilmente. Se puede agregar nuevos paquetes a cualquier proyecto de manera local los cuales estarán ubicados en el directorio *node_modules* o a su vez se los puede instalar de manera global y que estos puedan ser utilizados por cualquier aplicación (Ali, 2013).

Sass: Es un preprocesador CSS, permite generar de manera automática, hojas de estilo, además permite que el CSS sea más fácil de mantener y más reutilizable. CSS por sí solo no ofrece algunas opciones, pero con la combinación de Sass como por ejemplo variables, funciones matemáticas, bucles entre otros. (Desarrollo Web, 2018)

Bootstrap 4: Es un framework que permite desarrollar webs adaptándose a la mayoría de los navegadores, su característica principal es ofrecer clases CSS responsive que permiten el acoplamiento a cualquier resolución y dispositivo. Es una atractiva herramienta para la creación de interfaces amigables con el usuario mediante el uso de las librerías CSS se puede añadir botones, menús, cuadros y una variedad de elementos. (Team, 2020)

JQUERY: Es una librería que tiene como objetivo facilitar el uso de JavaScript, permite simplificar la interacción con los documentos HTML, manipular el DOM(textos, imágenes, enlaces, etc.). JQuery agrupa varias líneas de código en una solo función y lo reutiliza para lograr una sola tarea y usarla. JQuery facilita el uso de Ajax, ayudando a crear interfaces de usuario dinámicas donde la información se actualice evitando la recarga total del documento. La creación de animaciones, realización de efectos y el manejo de eventos son más sencillos con JQuery ya que solamente se debe insertar las variables. (Cabello, 2017)

DataTables: Es una extensión de JQuery que permite agregar a las tablas diversas funcionalidades avanzadas (agregar, eliminar, modificar), ordenar alfabética o numéricamente columnas de datos de la tabla, paginar resultados, filtrar cada columna por algún criterio y manipular los datos de manera eficiente y en tiempo real. (Cabello, 2017)

Sweet Alert 2: Es una librería de JavaScript para desplegar mensajes de alerta y cuadros de diálogos de confirmación con un diseño agradable para el usuario.

ChartJS: Es una librería de JavaScript para generar diferentes tipos de gráficos y trazar datos en distintas escalas, soporta animaciones que pueden ser aplicadas cuando se cambian los datos o se actualizan los colores.

METODOLOGÍA XP

Para la consecución del objetivo del proyecto se empleará la metodología de desarrollo de software ágil “Extreme Programming (XP)” como parte del proceso de investigación y desarrollo del proyecto adoptando su filosofía, procesos, técnicas y herramientas.

Programación Extrema (XP)

La programación Extrema surgió de un grupo de desarrolladores que propusieron la metodología ágil, la cual es utilizada para gestionar proyectos con eficacia, flexibilidad y control, su objetivo principal es la comunicación continua entre el cliente y el equipo de trabajo para tener una planificación adecuada cumpliendo los lineamientos o técnicas de desarrollo, buscando siempre la satisfacción del cliente.

FASES DE LA METODOLOGÍA

Al proceso de la metodología de programación extrema lo componen las siguientes fases o actividades: la planificación, el diseño, la codificación y las pruebas. A continuación, se describen cada una de ellas:

Planificación:

Esta actividad tiene el propósito de recopilar la información o requerimientos necesarios por parte del cliente, dicho proceso se lo realiza con el uso de “historias de usuario”, que representan las salidas esperadas, y las características funcionales del sistema que se pretende desarrollar. Las historias de usuario son escritas por el cliente, a las cuales les asigna un valor o importancia y una prioridad para que los miembros del equipo XP las analicen y les asignen un costo,

medido en tiempo de desarrollo generalmente este tiempo de desarrollo esta medido en semanas (Pressman, 2010).

Si el tiempo de desarrollo de una historia de usuario supera las tres semanas esta se puede descomponer en historias más cortas y asignarles una nueva prioridad.

Diseño:

La Programación Extrema fomenta la utilización de tarjetas CRC (clase-responsabilidad-colaborador) en estas tarjetas se incluye y organiza las clases o módulos más relevantes para los incrementos del software. Conforme el desarrollo del software avanza la etapa del diseño se encuentra en un constante cambio y las tarjetas CRC son la evidencia y constancia de esta etapa (Pressman, 2010).

Codificación:

Programación Extrema recomienda “la programación por parejas”, es decir que, para la consecución de una historia de usuario, deban trabajar dos personas simultáneamente en su desarrollo. Esta metodología ayuda a solucionar problemas en tiempo real ya que cada una de las personas puede tomar un rol diferente durante el desarrollo, una puede encargarse de los detalles del código, mientras la otra se encarga de seguir los estándares de codificación (Pressman, 2010).

Pruebas:

Las pruebas deben realizarse de manera constante y automática, incluso el propio cliente puede realizar o proponer nuevas pruebas e ir validando las mini versiones del software. El cliente es el encargado de especificar las pruebas de aceptación o también llamadas pruebas funcionales, las cuales se centran en las características y funcionalidades que se esperan del software (Pressman, 2010).

CAPITULO II ANÁLISIS Y DISEÑO

REQUISITOS FUNCIONALES

Tabla 3. Requisitos Funcionales

SEGURIDAD	
CÓDIGO	DESCRIPCION
RF1	El sistema controlará el acceso solamente a usuarios autorizados. Para ingresar al sistema el usuario deberá proporcionar su correo electrónico y su contraseña.
RF2	Cada usuario pertenecerá algún perfil el cual tendrá habilitado determinados permisos que lo habiliten de realizar acciones en el sistema.
RF3	El sistema permitirá la recuperación de contraseña. Se enviará una clave temporal al email del usuario que ha solicitado la recuperación de su contraseña.
PROCESO	
CÓDIGO	DESCRIPCIÓN
RF4	El sistema debe permitir la búsqueda de pacientes por número de archivo o por número de cedula.
RF5	El sistema creará una notificación cuando se registre alguna de las siguientes transacciones: registro de un paciente, asignación de cama, cambio de cama, cambio de servicio, egreso o alta del paciente, referencia, contra referencia y defunción del paciente.
RF6	El sistema permitirá la asignación de cama a un paciente que ingrese por consulta externa. Mediante un buscador por número de archivo o número de cedula, realizará la conexión a una base de datos alterna obteniendo la información del paciente al que se le asignará una cama.
RF7	El sistema deberá restringir que en la asignación de cama se ingrese al menos un diagnóstico.
RF8	En el formulario de asignación de cama el sistema deberá mostrar una matriz de camas identificando su servicio, habitación y disponibilidad de manera intuitiva al usuario.

RF9	El sistema deberá restringir la asignación de cama a un paciente que ya tiene una cama asignada. Se le solicitará hacer un cambio de cama.
RF10	CRUD para las diferentes: áreas, pisos, servicios, habitaciones, camas, usuarios, perfiles, permisos, y demás elementos que forman parte del sistema.
RF11	El sistema permitirá el registro de cambio o traslado de un paciente entre los diferentes servicios.
RF12	El sistema permitirá el registro de alta o egreso de un paciente.
RF13	El sistema proveerá una interfaz con información de los pacientes internos y su ubicación actual dispuesta a mostrarse en las pantallas de sala de espera.
RF14	El sistema debe registrar el cumplimiento de procedimientos según la patología y el servicio de hospitalización. En donde se tomará en cuenta indicadores como: ingresos, egresos, traslados, motivos de altas, tiempos de atención, diagnósticos de ingreso, diagnósticos de alta, tasa de morbilidad y mortalidad de pacientes.
RF15	Se deberá emitir reportes diarios, mensuales, anuales de acuerdo con los indicadores de calidad.
RF16	En el formulario de cambio o egreso de paciente la búsqueda de pacientes deberá ser por nombres y apellidos, filtrando solamente los pacientes con una cama asignada.
RF17	En el formulario de cambio o egreso de paciente el sistema permitirá modificar los diagnósticos del paciente.
RF18	En el campo de búsqueda de diagnóstico se deberá filtrar los resultados ya sea por el código del diagnóstico o su descripción.
RF19	El sistema generará una interfaz con el historial de cada paciente y así poder evidenciar las transacciones realizadas durante su estadía en el hospital.
RF20	El sistema permitirá la modificación de los datos de un paciente únicamente si su origen es emergencia. La información de pacientes de consulta externa no debe ser modificable.
RF21	En el formulario de registro de pacientes, para los pacientes extranjeros o que no posean un documento de identidad, el sistema debe

	generar automáticamente un código único conformado por 17 caracteres alfanuméricos, el cual deberá ser utilizado como su número único de identificación clínica.
RF22	En el listado de pacientes internos el sistema debe permitir el filtro por servicio. Así mismo en el formulario de cambio o egreso del paciente, antes de su búsqueda el sistema debe permitir filtrar los pacientes de acuerdo con el servicio ingresado.
RF23	El sistema deberá permitir que los reportes puedan ser exportados en formatos de PDF.
INTERFAZ GRAFICA	
CODIGO	DESCRIPCION
RF24	El campo primer y segundo nombre, apellido paterno y materno del formulario registro de paciente, aceptará únicamente caracteres alfabéticos.
RF25	El campo número de cedula, y número de teléfono del formulario registro de paciente, aceptara caracteres numéricos únicamente con una longitud de 10 caracteres.
RF26	Los campos cantón y parroquia dependerán de la provincia seleccionada previamente.
RF27	El campo barrio y dirección tanto del paciente como del contacto de emergencia aceptara caracteres alfanuméricos y caracteres especiales.
RF28	El campo edad se deberá calcular automáticamente al ingresar la fecha en el campo fecha de nacimiento.
RF29	En el formulario de registro y de asignación de cama en la búsqueda de paciente se permitirá solo datos numéricos.

Elaborado por: Cadena & Perugachi

REQUISITOS NO FUNCIONALES

Tabla 4. Requisitos No Funcionales

REQUISITOS DE ORGANIZACION	
RESTRICCION TECNICA	
RNF1	El sistema será implementado en el lenguaje de programación PHP utilizando como marco de desarrollo “Zend Framework” versión 1.12
RNF2	El sistema deberá utilizar como gestor de base de datos PostgreSQL.
RNF3	La interfaz de usuario ha de ser compatible con versiones de navegador web Firefox 30 o superior y Chrome 20 o superior.
REQUISITOS DE PRODUCTO	
RENDIMIENTO	
RNF4	El 95% de las transacciones no debe tardar más de 3 segundos.
RNF5	El sistema debe soportar y funcionar eficientemente hasta con 100 usuarios simultáneos. Esto se medirá por medio de la herramienta Apache JMeter.
SEGURIDAD	
RNF6	Solamente el usuario de perfil ADMINISTRADOR podrá denegar o conceder permisos a los diferentes perfiles.
RNF7	Un usuario puede crear una nueva cuenta, pero esta no estará habilitada hasta que un usuario de perfil ADMINISTRADOR la habilite.
RNF8	Las contraseñas de usuario serán gestionadas mediante el algoritmo de encriptación MD5.
RNF9	El sistema almacenara pistas de auditoria y trazabilidad sobre las transacciones principales del sistema en la base de datos.
FIABILIDAD	
RNF10	El sistema deberá recuperarse ante alguna caída total o parcial en máximo 5 minutos el 95% de las ocasiones.
DISPONIBILIDAD	
RNF11	El sistema debe estar disponible un 99,99% de las veces que el usuario intente accederlo.

USABILIDAD	
RNF12	La información solicitada deberá presentarse al usuario en un máximo de 5 clics el 90% de las veces.
RNF13	El sistema debe proporcionar los manuales técnicos y de usuario estructurados de manera simple y adecuada.
RNF14	El sistema mostrará al usuario notificaciones de alerta, de error e información para cada una de las acciones realizadas.
RNF15	El sistema web debe poseer un diseño responsive que garantice su adecuada visualización y manejo desde diferentes dispositivos como computadores personales, tabletas y teléfonos inteligentes.
PORTABILIDAD	
RNF16	El 80% del código es dependiente del servidor, el 20% restante corresponde a código JavaScript que se ejecuta del lado del navegador web.
RNF17	El sistema deberá ser compatible para sistema operativo Windows, Linux y MacOS, dependiendo de un servidor de aplicaciones PHP, de la librería Zend Framework y un navegador web.
MANTENIBILIDAD	
RNF18	Las tareas de mantenimiento del sistema deberá realizarlas un miembro de la unidad de TICS que cuente con un usuario en el perfil ADMINISTRADOR. El mantenimiento debe ser preventivo y correctivo tanto como para las diferentes entradas del sistema como para la base de datos.

Elaborado por: Cadena & Perugachi

Usuarios y Perfiles

Tabla 5. Perfiles de usuario y descripción

PERFIL	DESCRIPCIÓN
Administrador	Este usuario tiene acceso a toda la gestión del sistema, posee todos los privilegios para administrarlo.
Enfermería	Posee un privilegio básico, estos usuarios únicamente podrán visualizar los datos e inicializar el temporizador de tiempo para la desinfección de la cama.
Admisión Emergencia	Este usuario tendrá el privilegio de crear en el Módulo de registro de admisión de paciente, además podrá visualizar los datos de los otros módulos.
Atención Usuario	Este usuario tiene acceso y privilegios para manejar el módulo de Gestión de camas.

Elaborado por: Cadena & Perugachi

Historias de Usuario

Tabla 6. Proceso para Creación de nueva cuenta

Código Requerimiento	RF1, RF2
Nombre:	Creación de nueva cuenta
Usuarios:	Todos los usuarios
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa en la opción “crear cuenta” 2. El sistema carga en pantalla un formulario para el registro. 3. El usuario ingresa los datos correspondientes y los envía a grabar. 4. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado y debe esperar la activación de la cuenta por parte del administrador. 5. El administrador activa la cuenta creada.

	<p>6. El sistema envía un correo al usuario informando la activación.</p> <p>7. El usuario puede ingresar al sistema.</p>
Flujo Alternativo	<p>4.a El sistema detecta que existen campos obligatorios que no fueron ingresado e informa el suceso de que campos son los faltantes.</p> <p>4.b El sistema no pudo guardar los datos e informa el problema al usuario.</p> <p>5.a El administrador detecta que el usuario no es válido y no activa la cuenta notificando al usuario el problema.</p>

Elaborado por: Cadena & Perugachi

Tabla 7. Proceso para ingresar al sistema

Código Requerimiento RF1, RF2	
Nombre:	Ingreso al sistema
Usuarios:	Todos los usuarios
Flujo Normal	<p>1. El usuario ingresa a la plataforma.</p> <p>2. El sistema carga en pantalla un formulario para el ingreso.</p> <p>3. El usuario ingresa su usuario y contraseña y los envía.</p> <p>4. El sistema verifica los datos e inicializa satisfactoriamente.</p> <p>5. El usuario puede acceder a los módulos permitidos según su perfil.</p>
Flujo Alternativo	<p>4.a El sistema detecta que existen campos obligatorios que no fueron ingresados e informa el suceso de que campos son los faltantes.</p> <p>4.c El sistema detecta que el nombre y/o contraseña ingresada es incorrecta y notifica al usuario.</p>

Elaborado por: Cadena & Perugachi

Tabla 8. Proceso para la gestión de perfiles.

Código Requerimiento RF2	
Nombre:	Gestión de Perfiles
Usuarios:	Administrador

Flujo Normal	<ol style="list-style-type: none"> 1. El usuario administrador accede a la opción de registro de perfiles. 2. El sistema carga en pantalla un formulario para el registro. 3. El administrador ingresa los datos correspondientes y los envía a grabar. 4. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado
Flujo Alternativo	<p>4.a El sistema detecta que existen campos obligatorios que no fue ingresado e informa el suceso de que campos son los faltantes</p> <p>4.b El sistema no pudo guardar los datos e informa el problema al usuario</p>

Elaborado por: Cadena & Perugachi

Tabla 9. Proceso para asignar permisos al usuario

Código Requerimiento RF2	
Nombre:	Gestión de Permisos
Usuarios:	Administrador
Flujo Normal	<ol style="list-style-type: none"> 1. Mediante login el administrador ingresa al sistema y accede al módulo usuarios y perfiles. 2. El sistema carga en pantalla todos los perfiles registrados. 3. El administrador escoge un perfil y escoge la opción editar permisos. 4. El sistema despliega el formulario para escoger los permisos a proporcionar. 5. El administrador asigna a dicho usuario los permisos que le corresponden según su perfil y envía a guardar. 6. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado
Flujo Alternativo	<p>6.a El sistema no pudo guardar los datos e informa el problema al usuario</p>

Elaborado por: Cadena & Perugachi

Tabla 10. Proceso para recuperar la contraseña

Código Requerimiento RF3	
Nombre:	Recuperación de contraseña
Usuarios:	Todos los usuarios
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario accede a la opción de “recuperar contraseña” en el módulo “Iniciar Sesión” 2. El sistema carga en pantalla con un formulario. 3. El usuario ingresa el correo personal y los envía a grabar. 4. El sistema guarda satisfactoriamente y enviará una clave provisional al correo ingresado. 5. El usuario tendrá que ingresar a su correo personal y revisar su bandeja de entrada para obtener el código temporal. 6. El usuario deberá ingresar al sistema el código temporal para acceder e ingresar su nueva contraseña. 7. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado
Flujo Alternativo	<p>4.a y 7.a El sistema detecta que existen campos obligatorios que no fue ingresado e informa el suceso de que campos son los faltantes.</p> <p>4.b y 7.b El sistema no pudo guardar los datos e informa el problema al usuario.</p> <p>4.c El sistema no reconoce los datos ingresados y notifica al usuario el problema.</p>

Elaborado por: Cadena & Perugachi

Tabla 11. Proceso para registrar un paciente

Código Requerimiento RF20, RF21, RF24, RF25, RF26, RF27, RF28	
Nombre:	Gestión de pacientes
Usuarios:	Administrador, Admisión Emergencia
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario accede al módulo “registrar paciente” 2. El sistema carga en pantalla un formulario para el registro 3. El usuario ingresa los datos correspondientes y los envía a grabar.

	4. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado.
Flujo Alternativo	4.a El sistema detecta que existen campos obligatorios que no fueron ingresados e informa el suceso de que campos son los faltantes. 4.b El sistema no pudo guardar los datos e informa el problema al usuario.

Elaborado por: Cadena & Perugachi

Tabla 12. Proceso para la gestión de entidades

Código Requerimiento RF10	
Nombre:	Gestión de entidades del sistema
Usuarios:	Administrador
Flujo Normal	1. El administrador puede crear, leer, modificar y borrar datos del sistema.
Flujo Alternativo	1.a El sistema detecta que existen campos obligatorios que no fueron ingresados e informa el suceso de que campos son los faltantes. 1.b El sistema no pudo guardar los datos e informa el problema al usuario.

Elaborado por: Cadena & Perugachi

Tabla 13. Proceso para asignar una cama

Código Requerimiento RF4, RF6, RF7, RF8, RF9, RF11, RF13, RF18, RF29	
Nombre:	Asignación de cama
Usuarios:	Atención Usuario
Flujo Normal	1. El usuario accede a la opción de “búsqueda de paciente” en el módulo “Asignar cama” 2. El usuario ingresa el número de cedula o historia clínica del paciente que desee. 3. El sistema va filtrando hasta encontrar al paciente. 4. El usuario escoge la especialidad correspondiente. 5. El sistema despliega una matriz mostrando la disponibilidad de camas en esa especialidad.

	<ol style="list-style-type: none"> 6. El usuario asigna una cama a dicho paciente, su diagnóstico de ingreso y envía a guardar. 7. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado. 8. El sistema reflejará la información del paciente en los monitores de sala de espera 9. El personal de enfermería revisa el sistema e instala al paciente en su cama.
Flujo Alternativo	<ol style="list-style-type: none"> 3.a El sistema no encontró al paciente e informa al usuario. 5. a El sistema refleja que no existe disponibilidad de camas y notifica el problema. 7.a El sistema detecta que existen campos obligatorios que no fueron ingresados e informa el suceso de que campos son los faltantes. 7.b El sistema detecta que al paciente ya se le asignó una cama e informa al usuario dándole la opción de cambio de cama o servicio. 8.a Los monitores no reflejan la información y se informa al administrador.

Elaborado por: Cadena & Perugachi

Tabla 14. Proceso para generar reportes

Código Requerimiento	RF5, RF14, RF15, RF19, RF23
Nombre:	Gestión de Reportes
Usuarios:	Administrador, Atención Usuario
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario ingresa al sistema al módulo de reportes. 2. El sistema carga los datos en la pantalla. 3. El usuario selecciona los datos que necesita para el reporte. 4. El usuario escoge las fechas en las que quiere que se genere el reporte. 5. El sistema generará los reportes en formato PDF.
Flujo Alternativo	<ol style="list-style-type: none"> 3.a El sistema no carga datos porque no hay información e informa al usuario.

	<p>4.a El sistema detecta que en las fechas seleccionadas no hay datos e informa el problema al usuario</p> <p>5.a El sistema no genera el reporte y lo notifica al usuario.</p>
--	--

Elaborado por: Cadena & Perugachi

Tabla 15. Proceso para cambio o egreso de paciente

Código Requerimiento RF12, RF16, RF17, RF22	
Nombre:	Cambio o egreso de paciente
Usuarios:	Atención Usuario.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario debe elegir el servicio donde se encuentra el paciente. 2. El usuario debe buscar al paciente por los nombres y apellidos. 3. El sistema filtra solamente los pacientes con una cama asignada. 4. El usuario debe escoger una causa de cambio o egreso y llenar los campos que se requiere y envía a guardar. <ol style="list-style-type: none"> 4.1 Si es un cambio de cama, el usuario elige la nueva cama y envía a guardar. 4.2 Si es una transferencia a otro servicio, el usuario debe escoger primero el servicio, escoger la cama y enviar a guardar 4.3 Si es un egreso, el usuario simplemente envía a guardar. 5. El sistema guarda satisfactoriamente los datos e informa al usuario del proceso realizado 6. El sistema se actualiza y la nueva información es desplegada en la pantalla de la sala de espera.
Flujo Alternativo	<p>3.a El sistema no encuentra al paciente y notifica dicho problema al usuario.</p> <p>4.1 y 4.2 El usuario verifica que no hay una cama en dicho servicio y busca en otro servicio</p> <p>5.a El sistema detecta que existen campos obligatorios que no fueron ingresados e informa el suceso de que campos son los faltantes</p>

	<p>5.b El sistema no pudo guardar los datos e informa el problema al usuario.</p> <p>6.a La información no se actualiza y se informa al administrador.</p>
--	--

Elaborado por: Cadena & Perugachi

Tabla 16. Proceso para visualizar datos

Código Requerimiento RF22	
Nombre:	Visualización de pacientes internos
Usuarios:	Atención Usuario
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema desplegará una pantalla con todos los pacientes registrados en el sistema 2. El usuario podrá visualizar toda la información de cada paciente.
Flujo Alternativo	1.a El sistema no posee información

Elaborado por: Cadena & Perugachi

Diagramas de casos de Uso (CU)

CU: Perfil Administrador

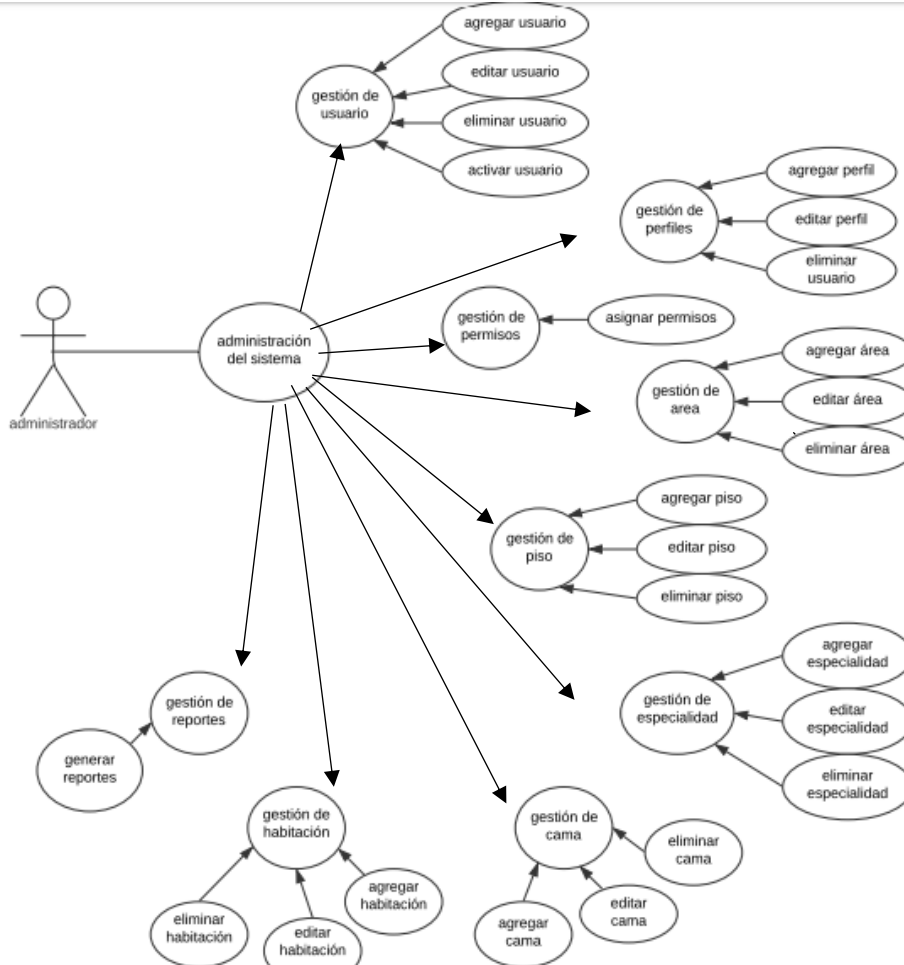


Figura 3. CU: Perfil administrador
Elaborado por: Cadena & Perugachi

CU: Perfil Admisión Emergencia

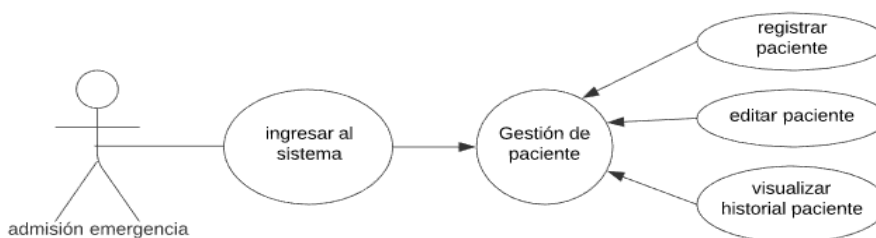


Figura 4. CU: Perfil Admisión emergencia
Elaborado por: Cadena & Perugachi

CU: Perfil Atención Usuario

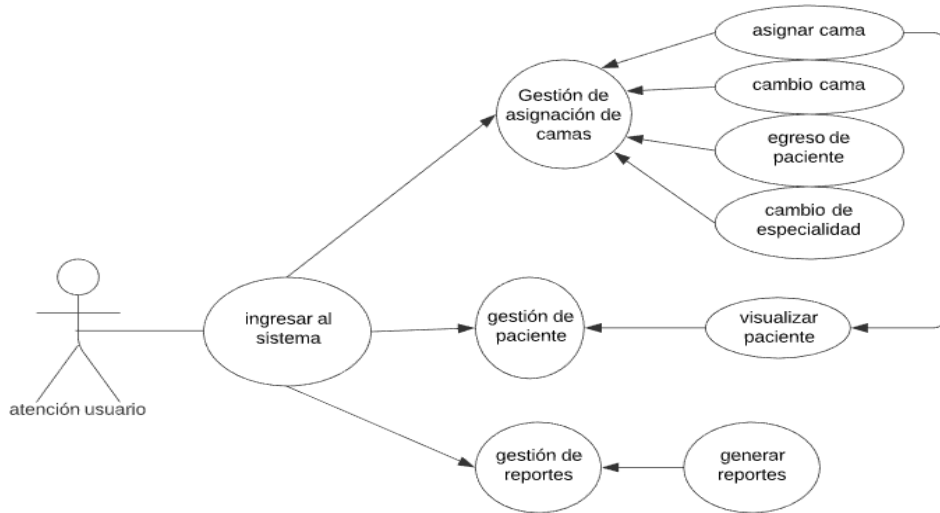


Figura 5. CU: Perfil atención usuario
Elaborado por: Cadena & Perugachi

CU: Perfil Enfermería

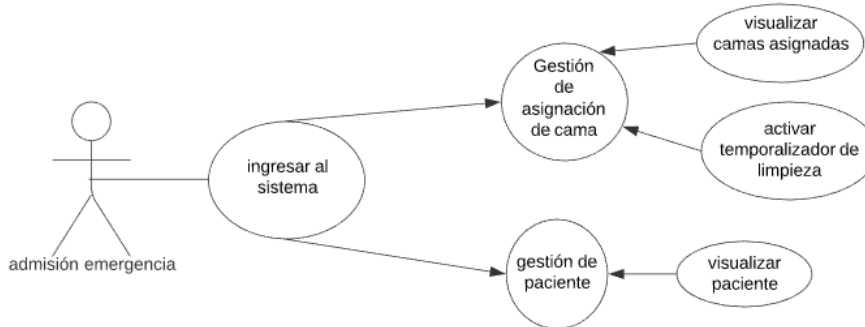


Figura 6. CU: Perfil Enfermería
Elaborado por: Cadena & Perugachi

CU: Creación de nueva cuenta

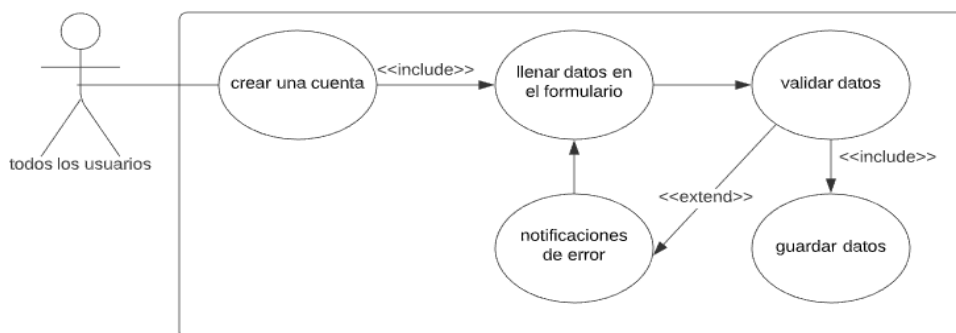


Figura 7. CU: Creación de nueva cuenta
Elaborado por: Cadena & Perugachi

CU: Ingreso al sistema

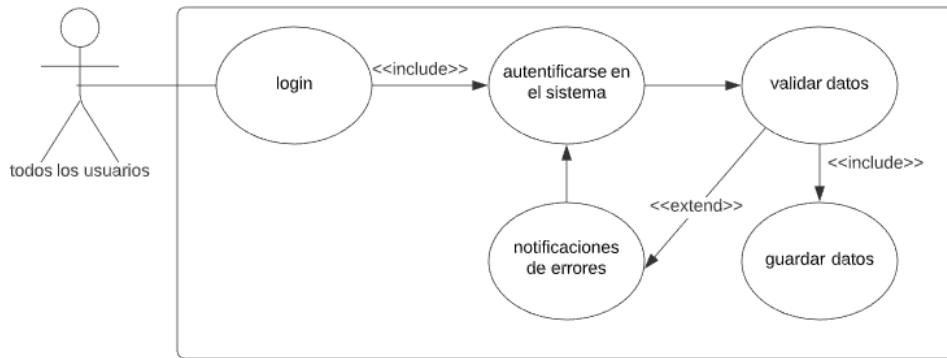


Figura 8. CU: Ingreso al sistema
Elaborado por: Cadena & Perugachi

CU: Gestión de perfil

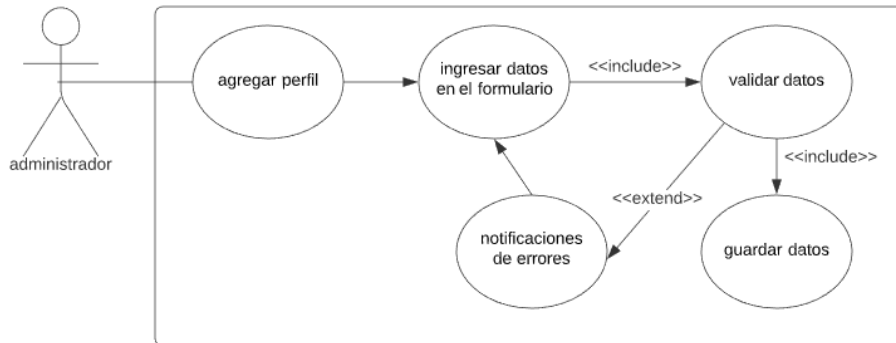


Figura 9. CU: Gestión de Perfil
Elaborado por: Cadena & Perugachi

CU: Gestión de permisos

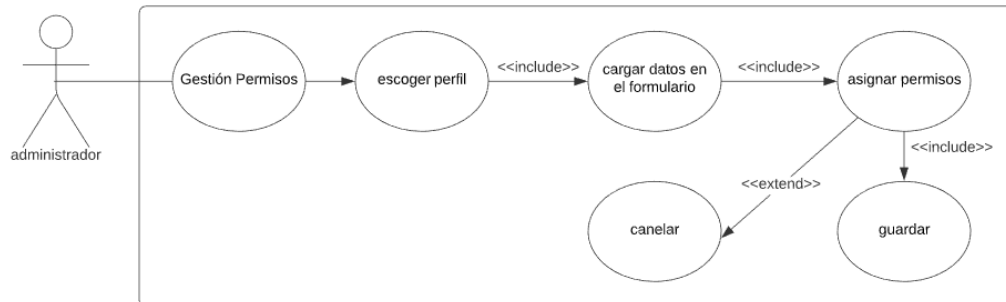


Figura 10. CU: Gestión de Permisos
Elaborado por: Cadena & Perugachi

CU: Recuperación de contraseña

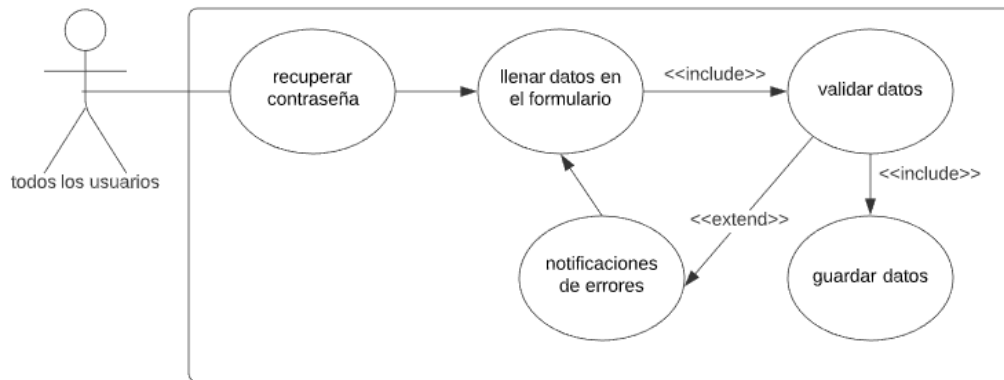


Figura 11. CU: Recuperación de contraseña
Elaborado por: Cadena & Perugachi

CU: Registro paciente

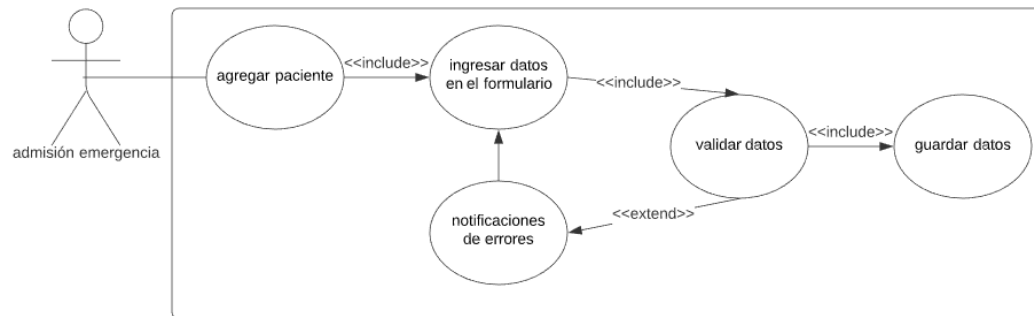


Figura 12. CU: Registro de Paciente
Elaborado por: Cadena & Perugachi

CU: Editar paciente

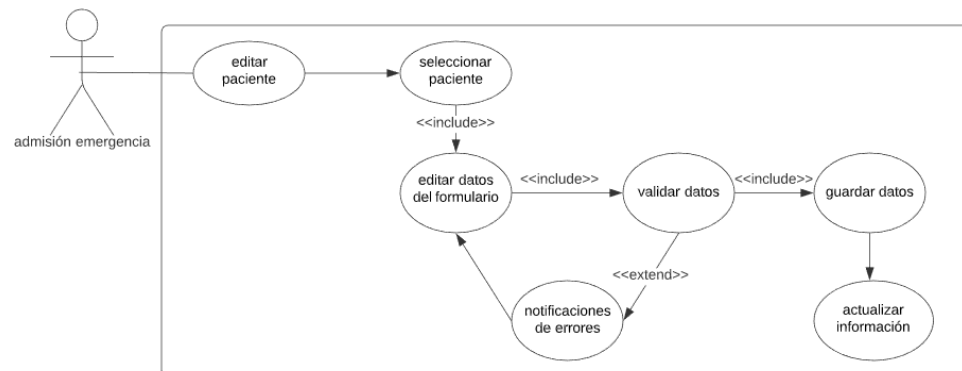


Figura 13. CU: Editar paciente
Elaborado por: Cadena & Perugachi

CU: Gestión asignar cama

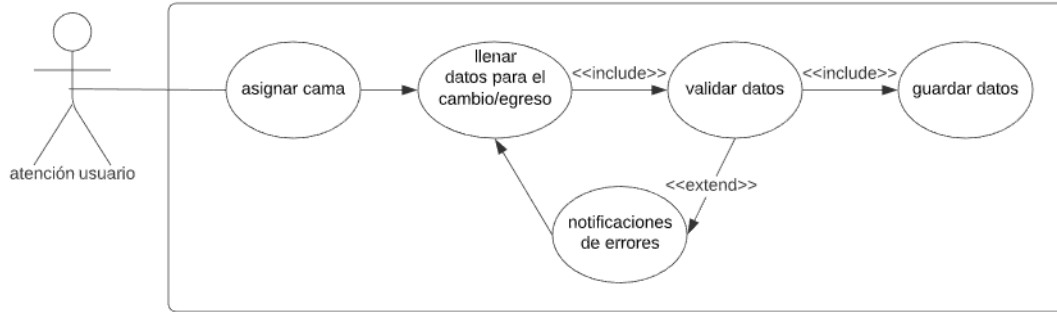


Figura 14. CU: Gestión asignar cama
Elaborado por: Cadena & Perugachi

CU: Gestión de cambio/egreso

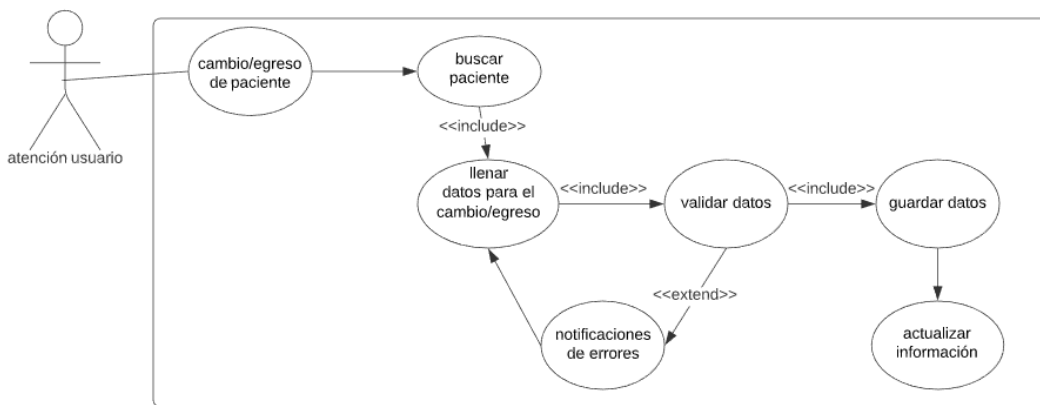


Figura 15. CU: Gestión de cambio/egreso
Elaborado por: Cadena & Perugachi

CU: Generar Reporte

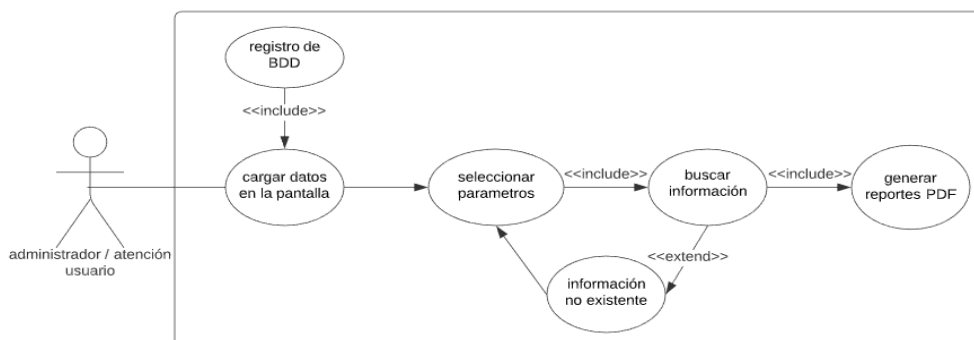


Figura 16. CU: Generar Reporte
Elaborado por: Cadena & Perugachi

CU: Visualizar datos

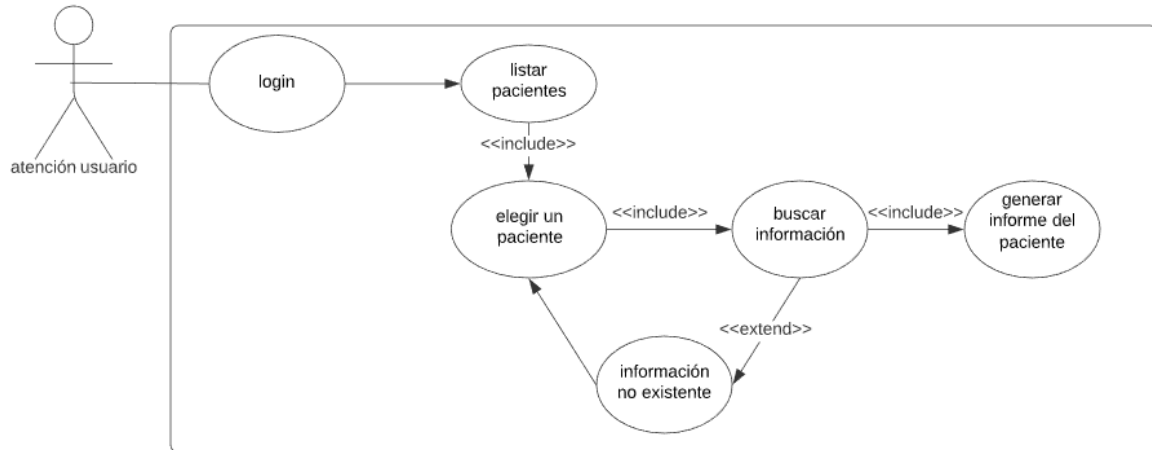


Figura 17. CU: Visualizar datos
Elaborado por: Cadena & Perugachi

DIAGRAMAS DE SECUENCIA (DS)

DS: Crear una nueva cuenta

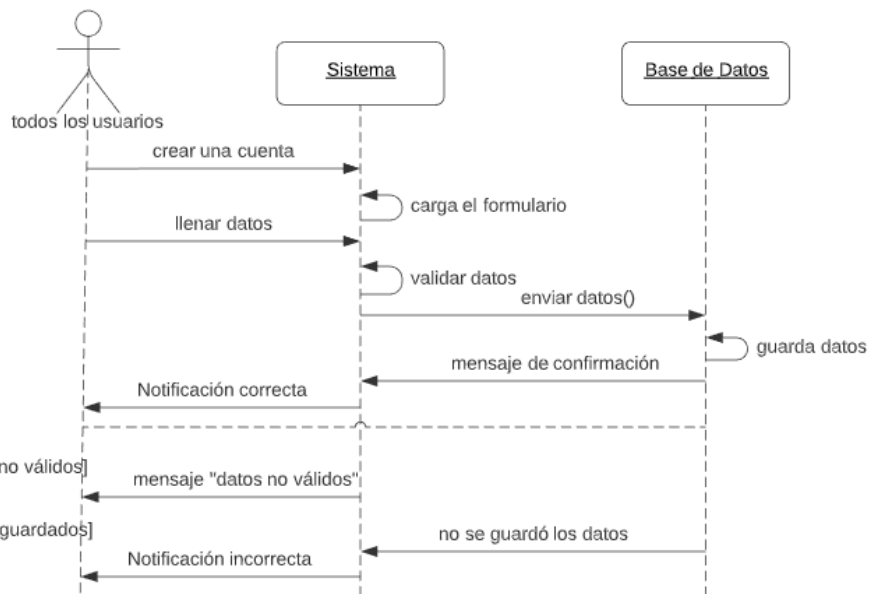
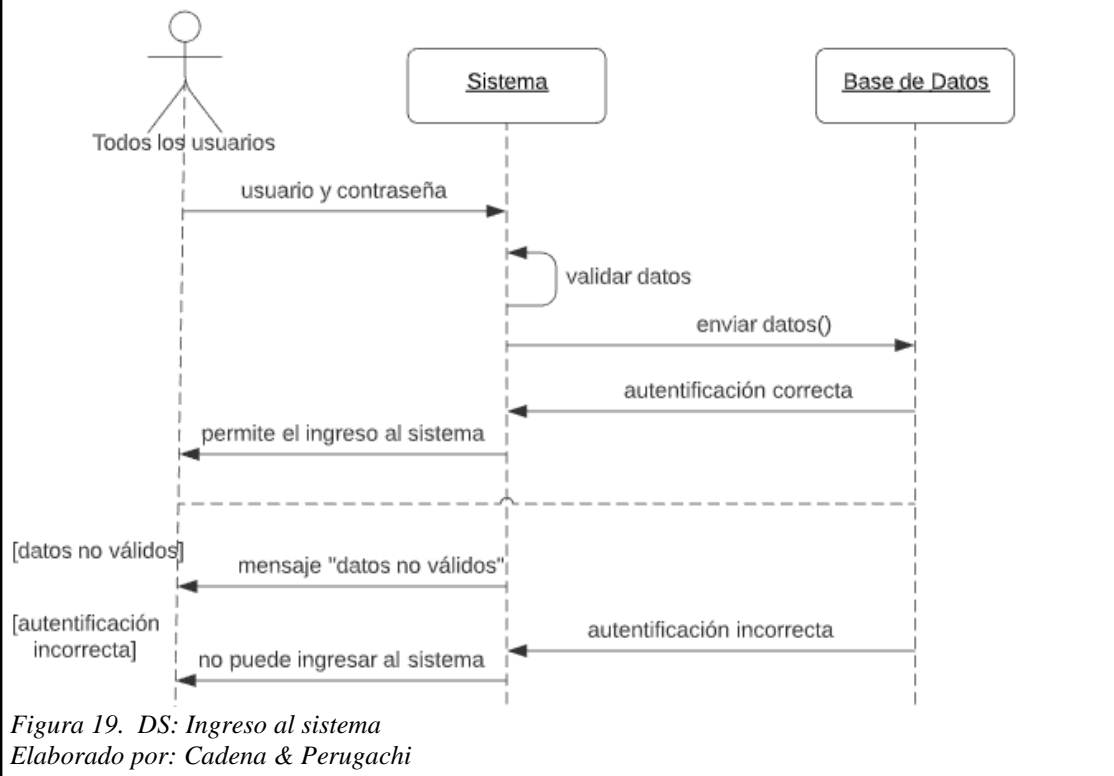
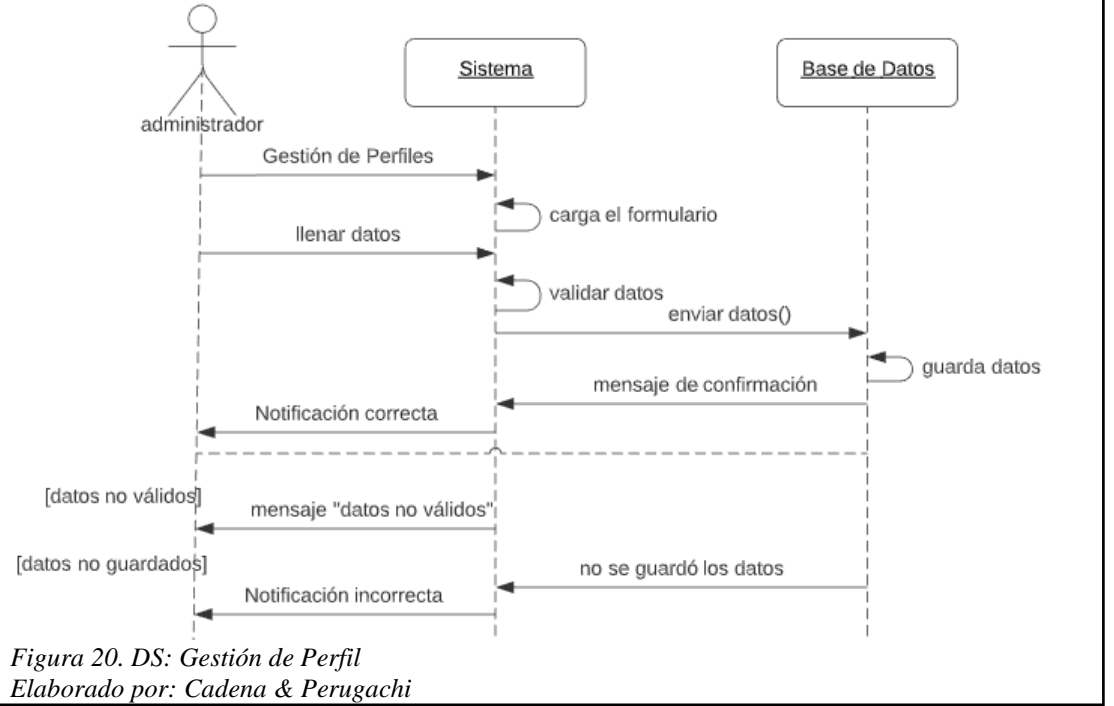


Figura 18. DS: Crear cuenta nueva
Elaborado por: Cadena & Perugachi

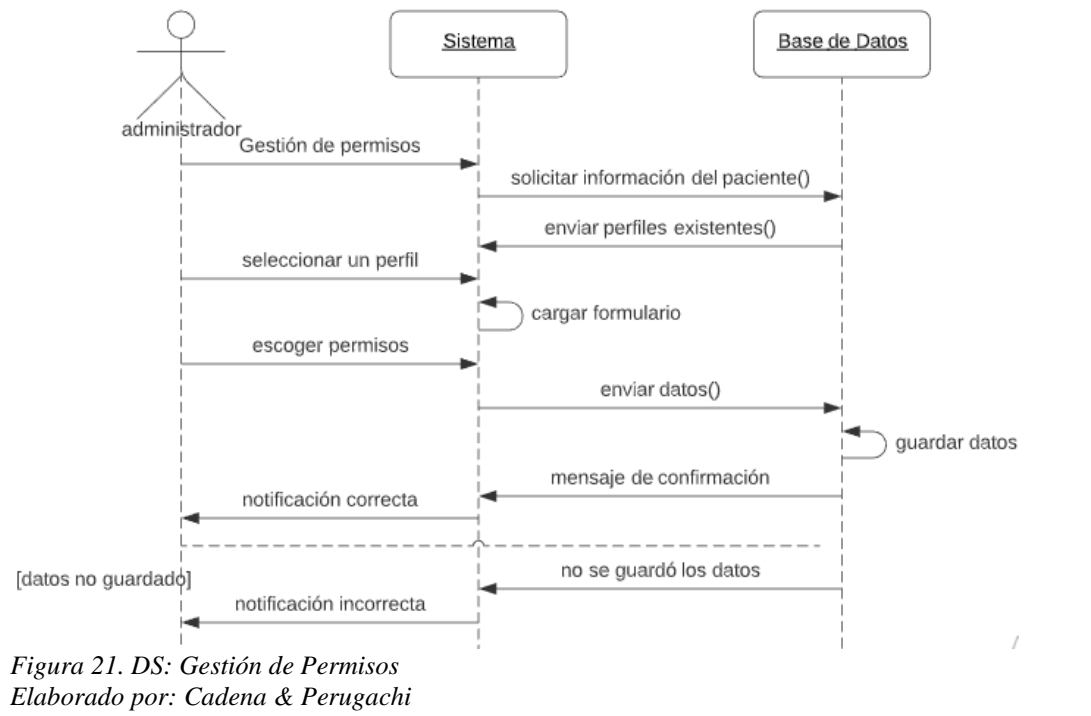
DS: Ingreso al sistema



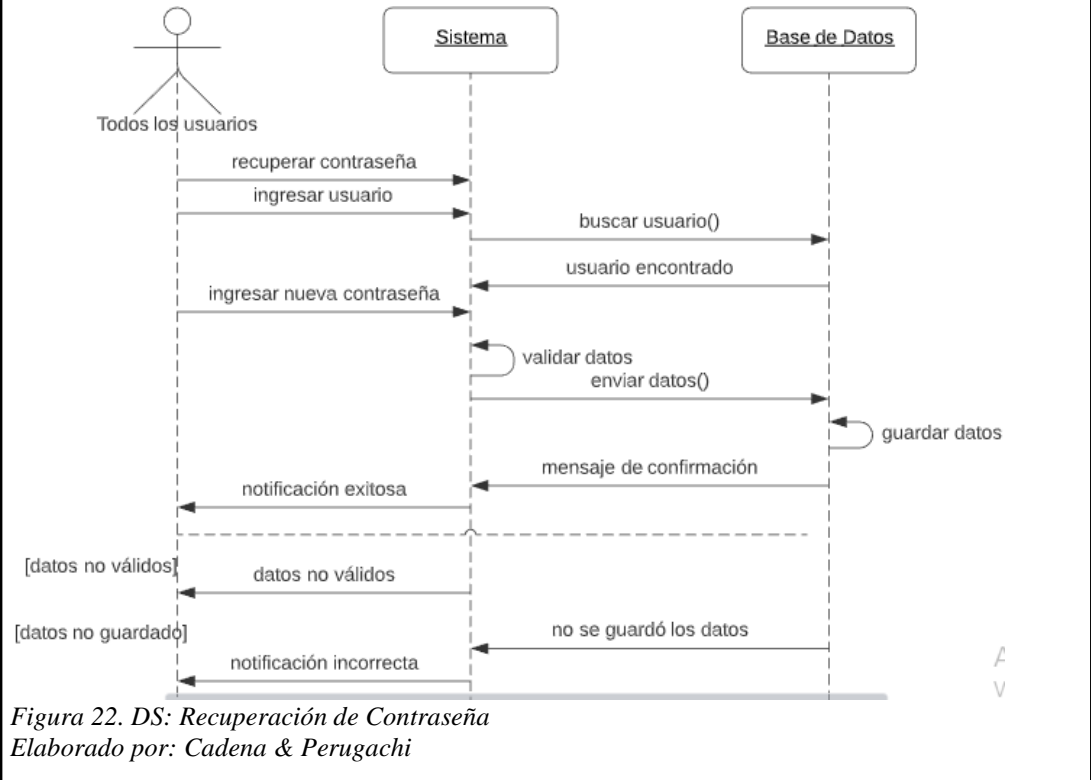
DS: Gestión de perfil



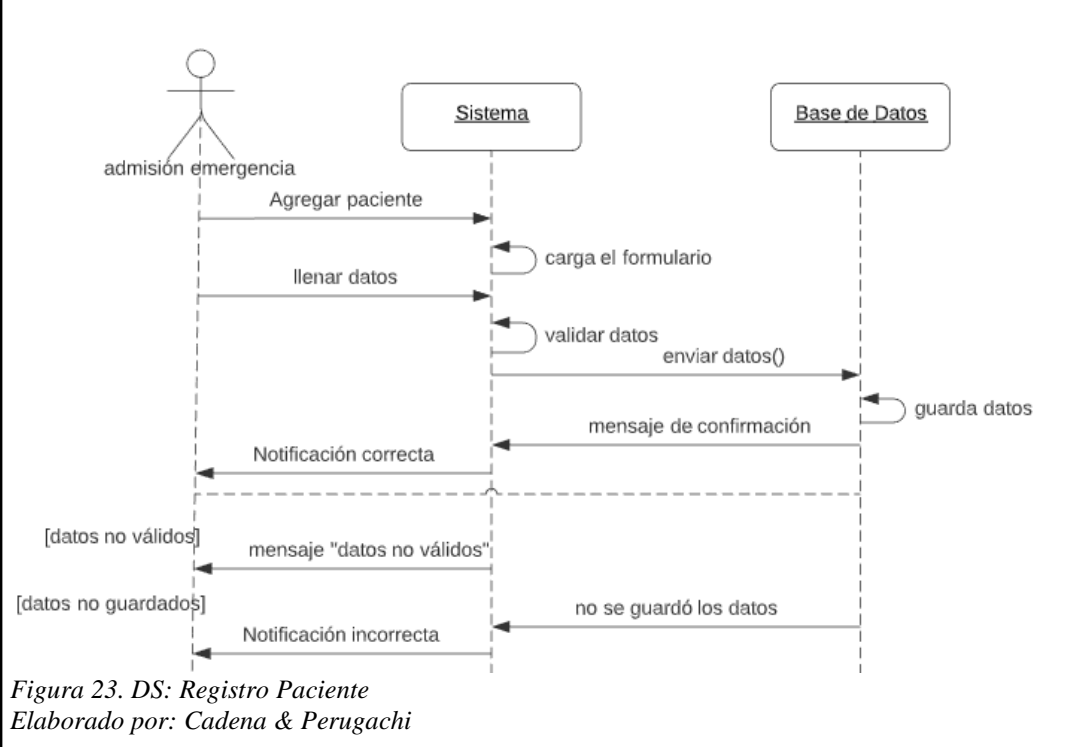
DS: Gestión de permisos



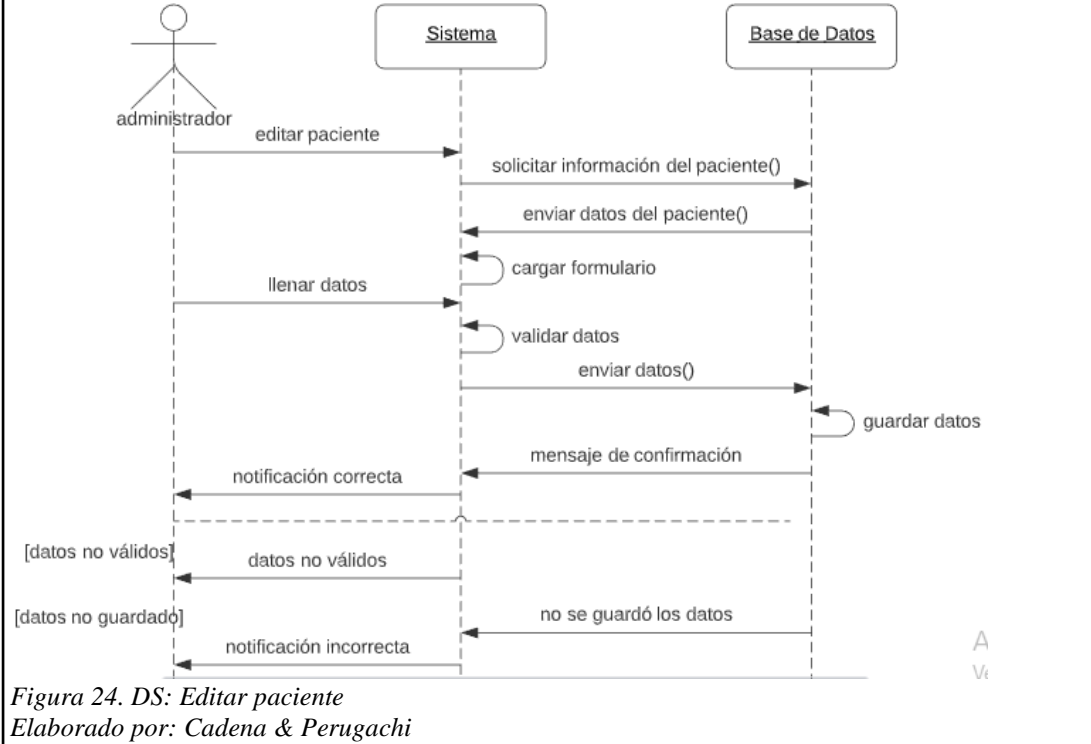
DS: Recuperación de contraseña



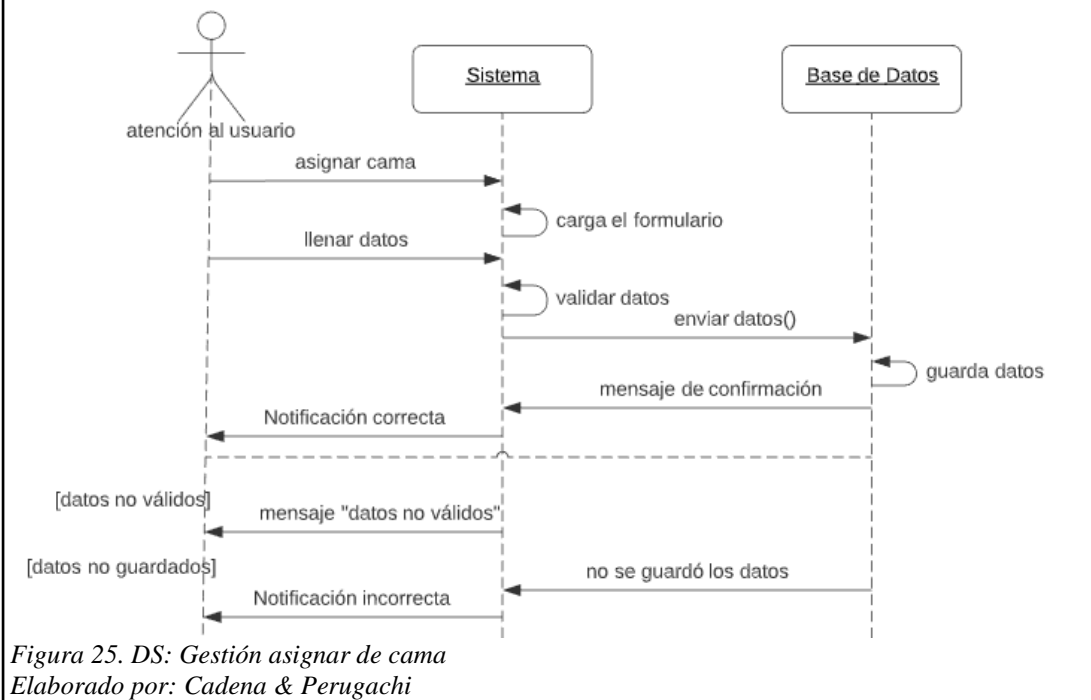
DS: Registro paciente



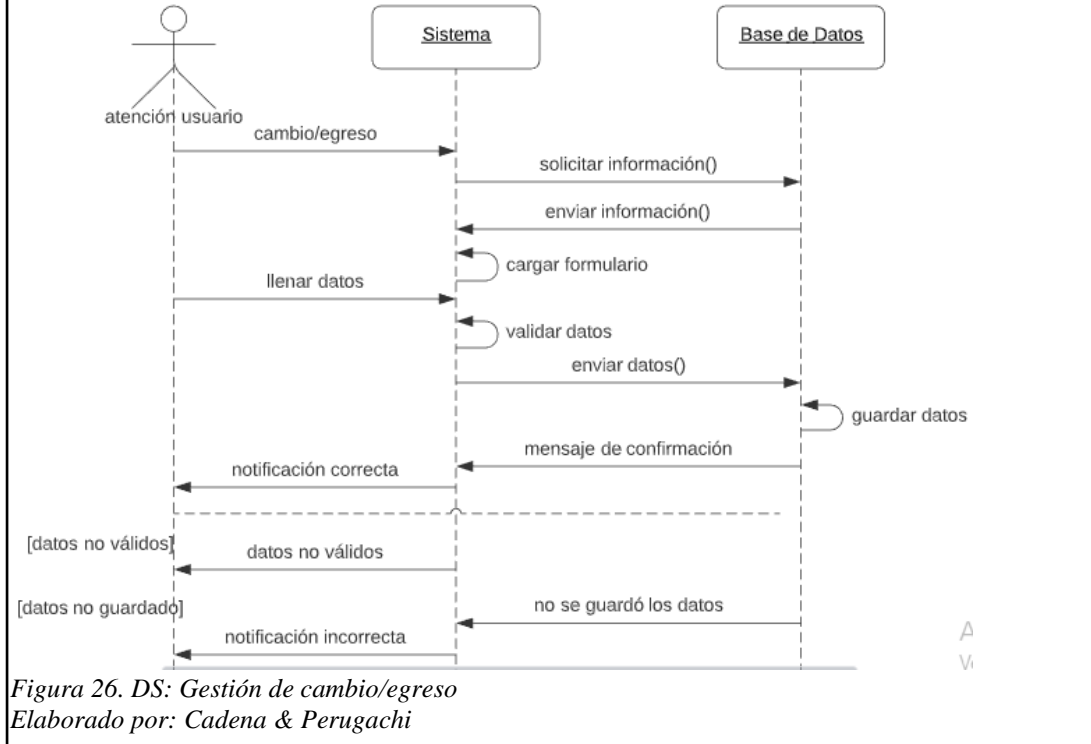
DS: Editar paciente



DS: Gestión asignar cama



DS: Gestión de cambio/egreso



DS: Visualizar datos pacientes

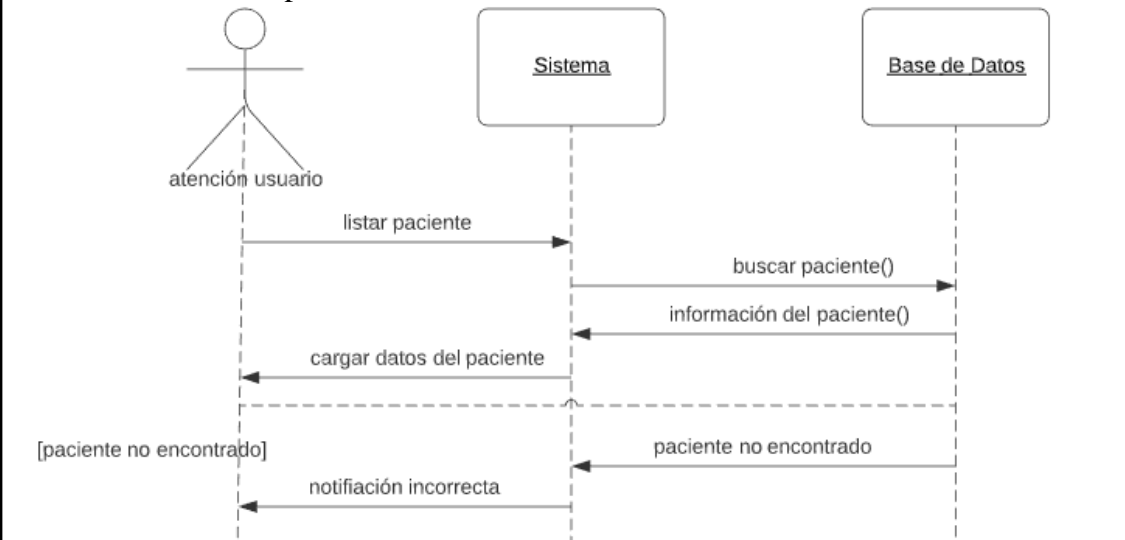


Figura 27. DS: Visualizar datos pacientes
Elaborado por: Cadena & Perugachi

DS: Generar reportes

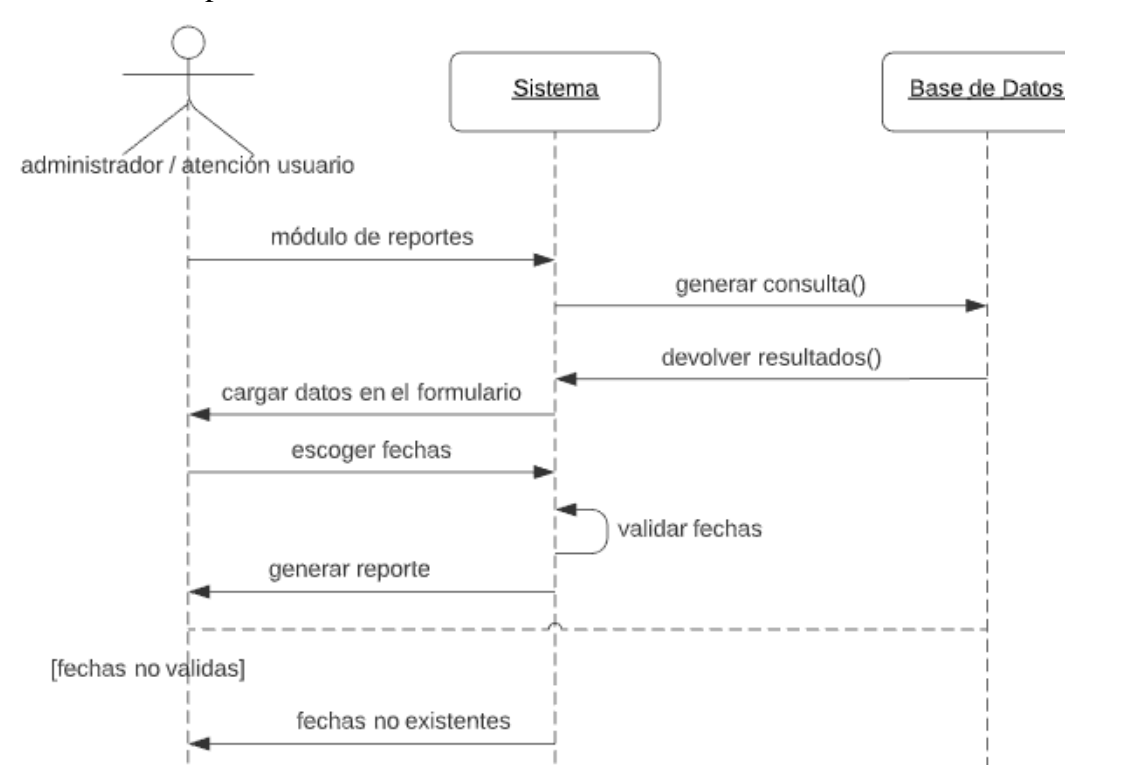


Figura 28. DS: Generar reportes
Elaborado por: Cadena & Perugachi

Diagrama de navegación

Diagrama de navegación: Perfil Administrador

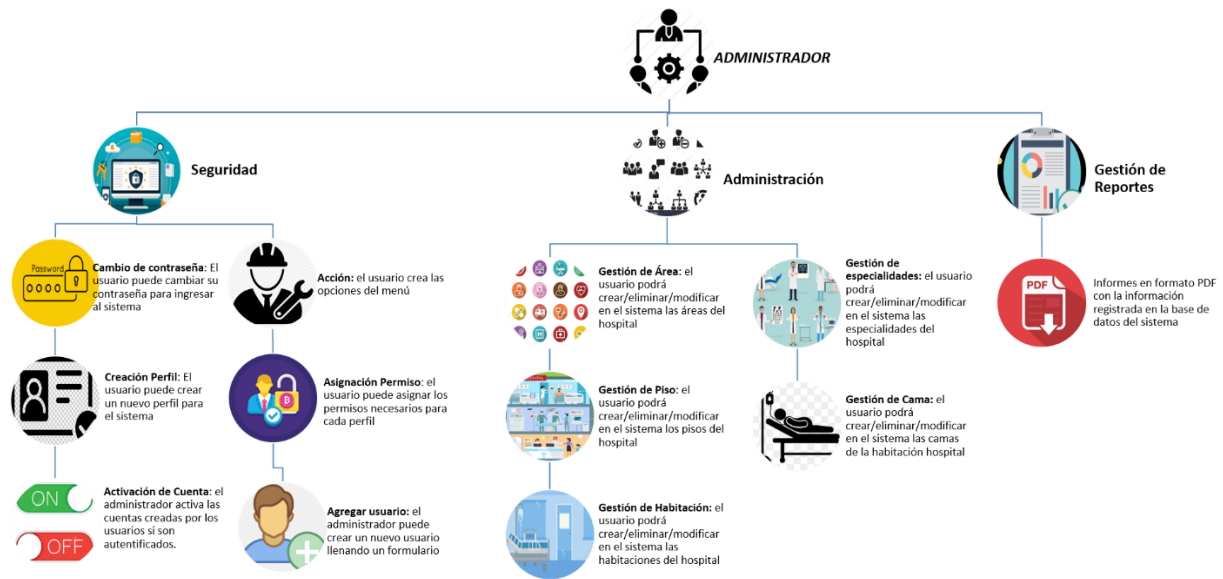


Figura 29. Diagrama de navegación: Perfil Administrador
Elaborado por: Cadena & Perugachi

Diagrama de navegación: Perfil atención usuario

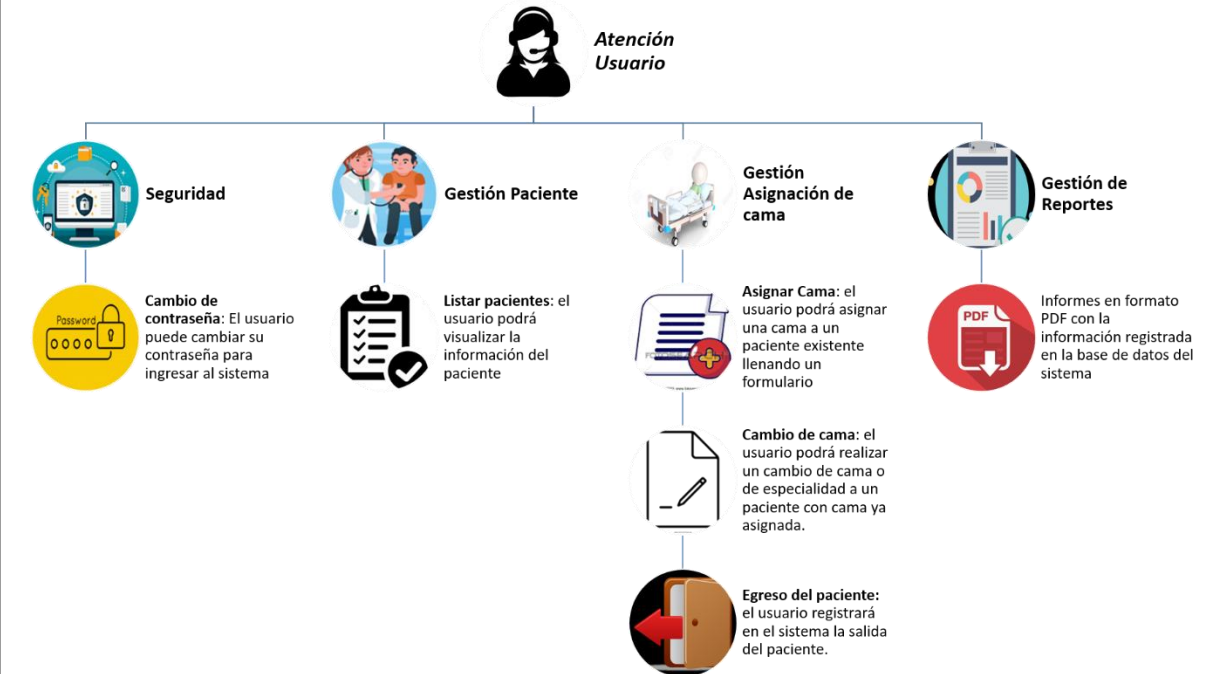


Figura 30. Diagrama de navegación: Perfil atención usuario
Elaborado por: Cadena & Perugachi

Diagrama de navegación: Perfil admisión emergencia



Figura 31. Diagrama de navegación: Perfil admisión emergencia
Elaborado por: Cadena & Perugachi

Diagrama de navegación: Perfil enfermería

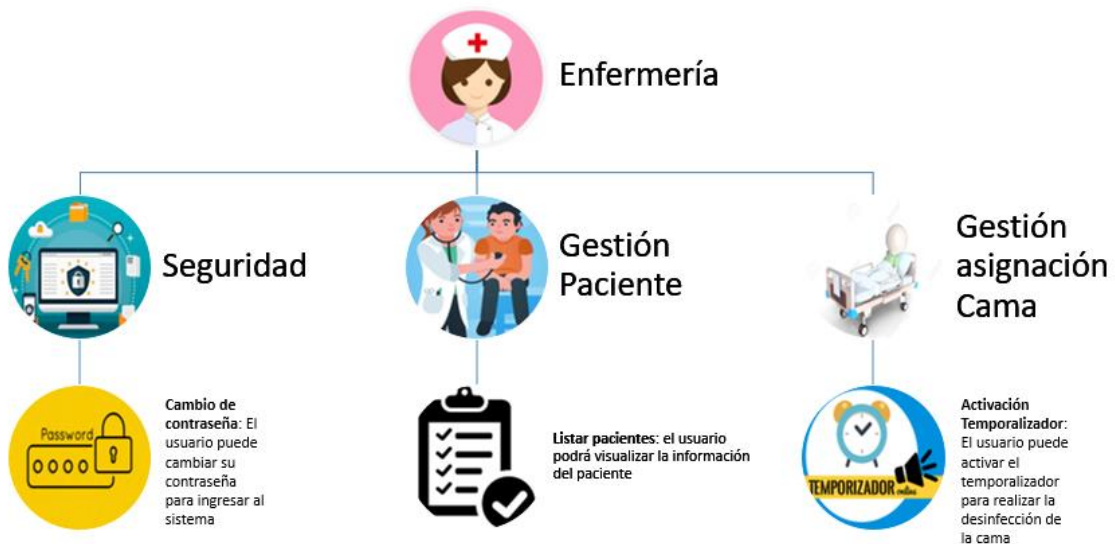


Figura 32. Diagrama de navegación: Perfil enfermería
Elaborado por: Cadena & Perugachi

Diseño conceptual de base de datos

Diseño conceptual de base de datos

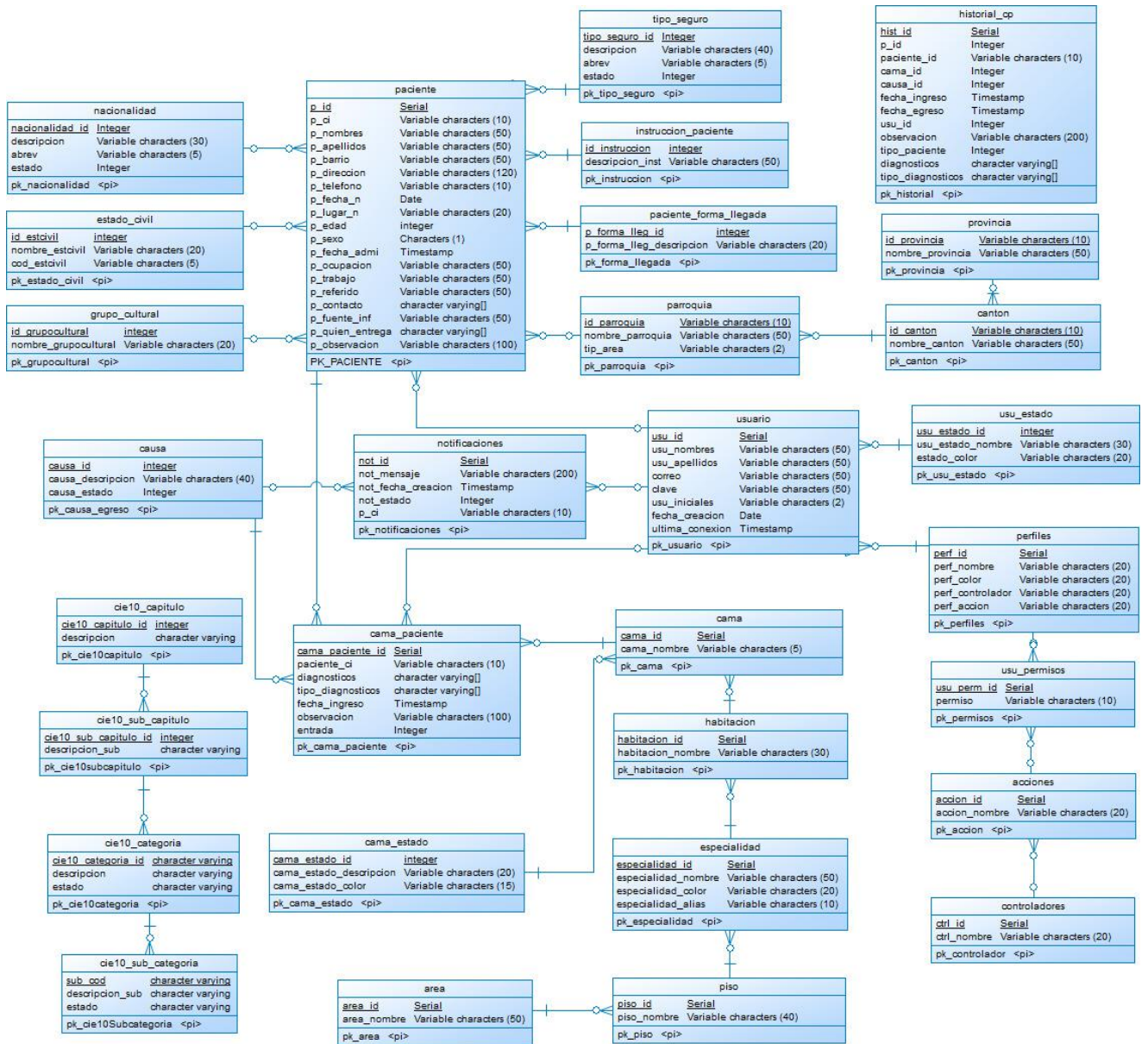
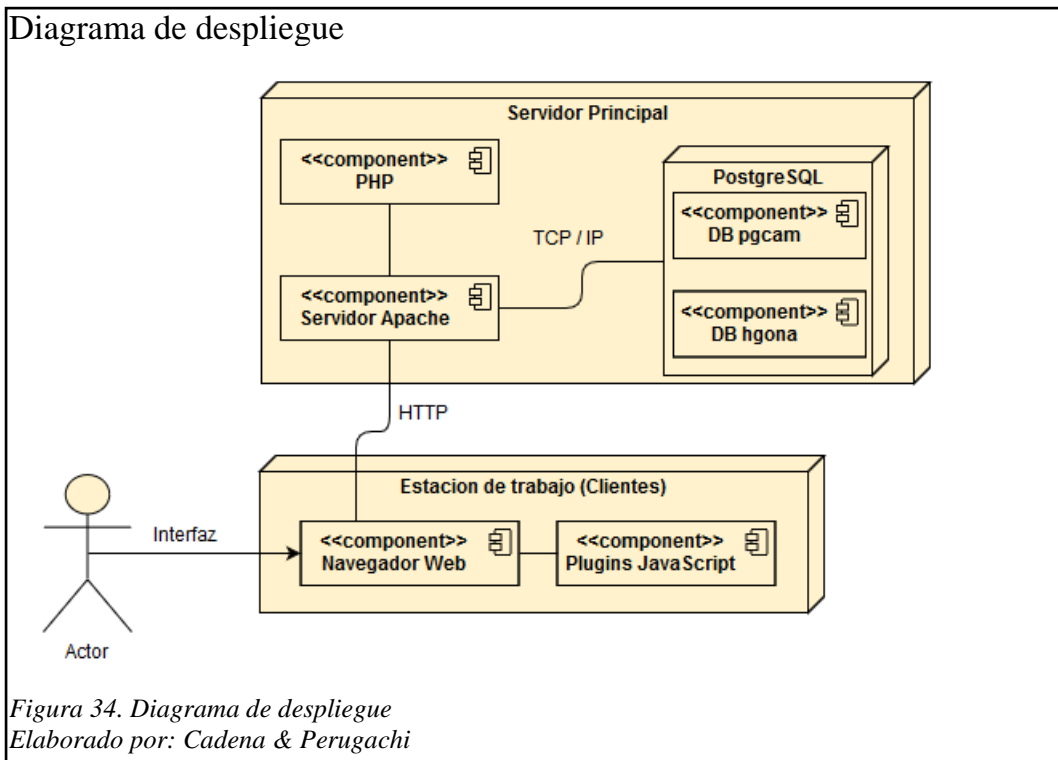


Figura 33. Diseño conceptual de base de datos
Elaborado por: Cadena & Perugachi

CAPÍTULO III CONSTRUCCIÓN Y PRUEBAS



El diseño del diagrama de despliegue es utilizado para conocer la topología del sistema, muestra las relaciones físicas de los nodos que lo componen y como cada uno de estos nodos se conecta mediante enlaces de comunicación. Un nodo representa un recurso computacional de hardware o software. En resumen, el diagrama de despliegue es un grupo de componentes o nodos unidos por conexiones de comunicación.

A continuación, se describen cada uno de los nodos que lo componen, sus características y como estos se comunican.

Nodo - Servidor principal:

El trabajo del servidor principal es esperar a que lleguen las solicitudes o peticiones de los clientes, procesarlas y si es necesario enviar una respuesta. El cual está compuesto por los siguientes componentes y subnodos:

- Servidor Apache: Servidor Web HTTP
- PHP: Preprocesador de hipertexto, lenguaje de programación orientado a objetos.
- PostgreSQL: Servidor de bases de datos PostgreSQL, considerado como un subnodo al contener y administrar varias bases de datos.

Nodo - Estación de trabajo (clientes):

La estación de trabajo o cliente es quien envía peticiones o solicitudes hacia el servidor, es quien permite la interacción entre el usuario final y la aplicación por medio de una interfaz. El cual está compuesto por los siguientes componentes:

- Navegador web: Software que permite el acceso a la web.
- Plugin JavaScript: Se ejecutan del lado del cliente específicamente en el navegador y no dependen directamente ni consumen recursos del servidor.

Subnodo - PostgreSQL:

Servidor de base de datos PostgreSQL encargado de administrar varias bases de datos, se ha considerado las siguientes bases de datos fundamentales para el sistema:

- DB pcam: Base de datos principal del sistema encargada de la lógica del negocio y en la que se ejecutan la mayor cantidad de las transacciones del sistema.
- DB hgona: Base de datos externa destinada al manejo de información de otros sistemas de la institución de salud, el sistema solicita información específica de ciertas tablas.

Comunicación:

El diagrama representa un modelo de comunicación cliente servidor, en el cual como primera instancia el cliente desde su estación de trabajo y mediante un navegador web solicita información mediante peticiones HTTP por el puerto 80 habilitado por el servidor de aplicaciones, este servidor interpreta dicha petición, se comunica con el servidor de base de datos mediante el protocolo de comunicación TCP/IP, obtiene los datos requeridos los procesa y retorna la información al cliente mediante una interfaz de tal forma que pueda ser entendible por el usuario final.

Código importante:

Según el acuerdo No. 00004934 emitido por LA MINISTRA DE SALUD PUBLICA DEL ECUADOR, en el Artículo 1 inciso 1.1 y 1.2. (Pública, Acuerdo Ministerial, 2014)

Para ciudadanos ecuatorianos que no posean un documento de identificación o para ciudadanos extranjeros, se debe generar un Código de Historia Único de acuerdo con la estructura de la tabla a continuación.

Tabla 17. Descripción para generar la Historia Clínica Única

# DE CARACTERES	DETALLE	OBSERVACIÓN
1 y 2	Dos primeras letras del primer nombre	
3	Primera letra del segundo nombre	En el caso de no poseer se ingresa el número 0.
4 y 5	Dos primeras letras del primer apellido	
6	Primera letra del segundo apellido	En el caso de no poseer se ingresa el número 0.
7 y 8	Código de provincia	En el caso de ciudadanos extranjeros se ingresa el numero 99
9 al 12	Año nacimiento	Formato: YYYY
13 y 14	Mes de nacimiento	Formato: MM
15 y 16	Día de nacimiento	Formato: DD
17	Control	Década de nacimiento

Elaborado por: Cadena & Perugachi

Para cubrir con dicha necesidad se desarrolló el siguiente algoritmo para la generación del Código Único de Historia Clínica.

Algoritmo generación de Código único

```

public function generadorCodigoUnico($primer_nombre, $segundo_nombre, $apellido_paterno,
    $apellido_materno, $codigoProv, $fecha_n)
{
    $nuhc = '';
    /**
     * ? Comprobacion de campo vacio en apellido materno y/o segundo nombre
     * ! si el campo es vacio reemplaza con '0'
     */
    $apellido_materno = ($apellido_materno=='') ? '0' : $apellido_materno; //operador ternario de comprobacion
    $segundo_nombre = ($segundo_nombre=='') ? '0' : $segundo_nombre;
    /**
     * ? Conversion a Array del campo fecha
     */
    $arrayFecha = explode('-', $fecha_n);
    /**
     * ? Adicion de strings en orden especifico
     */
    $nuhc = substr($primer_nombre, 0, 2) //Primeras dos letras del Primer nombre
        .substr($segundo_nombre, 0, 1) //Primera letra del segundo nombre
        .substr($apellido_paterno, 0, 2) //Primeras dos letras del Apellido Paterno
        .substr($apellido_materno, 0, 1) //Primera letra del Apellido Materno
        .$codigoProv //Codigo de provincia
        .$arrayFecha[0] // Año de nacimiento formato: YYYY
        .$arrayFecha[1] // Mes de nacimiento formato: MM
        .$arrayFecha[2] // Dia de nacimiento formato: DD
        .$arrayFecha[0][2]; // Decada del año formato: Y
    return (strlen($nuhc)==17) ? $nuhc : 'error'; // Comprobador de longitud de codigo igual a 17 caracteres
}
    
```

Figura 35. Algoritmo para asignar una Historia Clínica
 Elaborado por: Cadena & Perugachi

Explicación del código:

1. Realiza una comprobación de los campos: *\$apellido_materno* y *\$segundo_nombre*, en la cual verifica si el campo es vacío lo cambia por cero.
2. Separa el campo *\$fecha_n* en un dato de tipo Array que contendrá el año, el mes y el día por separado y de acuerdo con el siguiente formato: año=YYYY, mes=MM, día=DD.
3. En la variable *\$nuhc* se concatena la siguiente información:
4. Mediante el método *substr()* realiza las siguientes operaciones:
 - 4.1 Separa las primeras dos letras del (*\$primer_nombre*).
 - 4.2 Separa la primera letra del (*\$segundo_nombre*).
 - 4.3 Separa las primeras dos letras del (*\$primer_apellido*).
 - 4.4 Separa la primera letra del (*\$segundo_apellido*).
5. Agrega el código de provincia (*\$codidoProv*).
6. Obtiene el valor de cada elemento del array creado en el paso 2 y los concatena a la variable *\$nuhc* en el siguiente orden: año (YYYY), mes (MM) y día (DD).
7. Obtiene la década mediante el tercer carácter del elemento perteneciente al año del campo *\$arrayFecha* creado en el paso 2.
8. Verifica que la cadena *\$nuhc* que se va a enviar contenga una longitud igual a 17 caracteres.

Pruebas:

Pruebas de Caja Negra

Las pruebas de caja negra permiten verificar entradas y salidas en base a las condiciones preexistentes en los requerimientos funcionales del software, evidenciando que las salidas mostradas sean iguales a las salidas esperadas. Para ello se implementó la técnica de particionamiento equivalente.

Particionamiento equivalente: Permite examinar los valores válidos e inválidos de las entradas existentes y así poder someterlos a los casos de prueba respectivos.

Clases de equivalencia: Representan un grupo de estados admitidos y no admitidos de acuerdo con las condiciones de entrada del software, se recomienda asignar un identificador a

cada clase de equivalencia válida y no válida, en la *Tabla 18* se explica el significado de cada atributo que lo compone.

Tabla 18. Significado de cada atributo de las clases de equivalencia

CLASES DE EQUIVALENCIA	
ATRIBUTO	SIGNIFICADO
Condición de entrada	Cualquier tipo de entrada para el sistema
Tipo	Representa el tipo de entrada que será procesada por el sistema, en las cuales pueden haber de tipo: valor, conjunto, rango, lógicas.
Clase Equivalencia Valida	Representan todo tipo de entrada válida para el sistema.
Clase Equivalencia No Valida	Representa todo tipo de entrada inválida para el sistema.

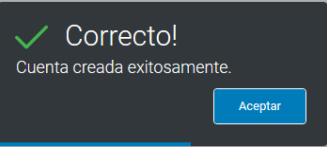
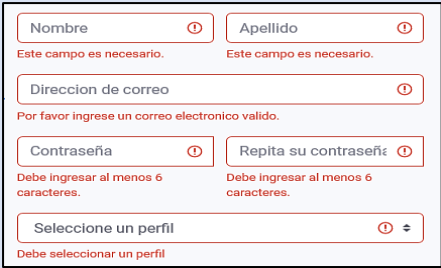


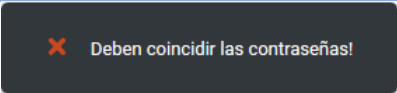
Elaborado por: Cadena & Perugachi

Tabla 19. Clase de Equivalencia para la creación de nueva cuenta

CLASES DE EQUIVALENCIA			
Condición de Entrada	Tipo	Clase Equivalencia Valida	Clase Equivalencia No Valida
Nombre, Apellido	Valor	CE1: Cualquier cadena de máximo 50 caracteres alfabéticos cada uno.	CE2: En blanco. CE3: Un valor no alfabético.
Dirección de correo	Valor	CE4: Cualquier cadena de máximo 50 caracteres alfanuméricos y caracteres especiales.	CE5: En blanco. CE6: Letras acentuadas. CE7: Valor empieza o termina con "punto" CE8: Solo un caracter @ es permitido.
Contraseña	Valor	CE9: Cualquier cadena de máximo 50 caracteres alfanuméricos y caracteres especiales.	CE10: En blanco
Verificador contraseña	Valor	CE11: Mismo valor que la entrada CE9	CE12: En blanco. CE13: Diferente valor que la entrada CE9
Perfil	Conjunto	CE14: Administrador. CE15: Admisión Emergencia. CE16: Atención Usuario. CE17: Enfermería	CE18: En blanco.

Elaborado por: Cadena & Perugachi

Tabla 20. Caso de prueba para creación de cuenta.

CASOS DE PRUEBA							
N° Caso	Clase de Equivalencia	Nombre, Apellido	Dirección de correo	Contraseña	Verificador contraseña	Perfil	Resultado
1	CE1, CE4, CE9, CE11, CE14	"Jose", "Perez"	josep@email.com	123456jp	123456jp	Administrador	
2	CE2, CE5, CE10, CE12, CE18	"" , ""	""	""	""	""	
3	CE3, CE4, CE9, CE11, CE14	"Jos4e", "P8erez"	josep@email.com	123456jp	123456jp	Administrador	
4	CE6, CE7, CE8, CE4, CE9, CE11, CE14	"Jose", "Perez"	.jos@ép@email.com	123456jp	123456jp	Administrador	
5	CE13, CE4, CE9, CE11, CE14	"Jose", "Perez"	josep@email.com	123456jp	xxx456jp	Administrador	

Elaborado por: Cadena & Perugachi

Tabla 21. Clase equivalente para: Ingreso al sistema

CLASES DE EQUIVALENCIA			
Condición de Entrada	Tipo	Clase Equivalencia Valida	Clase Equivalencia No Valida
Dirección de correo	Valor	CE1: Cualquier cadena de máximo 50 caracteres alfanuméricos y caracteres especiales.	CE2: En blanco. CE3: Letras acentuadas. CE4: Valor empieza o termina con "punto" CE5: Solo un caracter @ es permitido.
Contraseña	Valor	CE6: Cualquier cadena de máximo 50 caracteres alfanuméricos y caracteres especiales.	CE7: En blanco
Existe	Lógica	CE8: true	CE9: false
Habilitado	Lógica	CE10: true	CE11: false

Elaborado por: Cadena & Perugachi

Tabla 22. Caso de prueba para ingreso al sistema

CASOS DE PRUEBA								
N° Caso	Clase de Equivalencia	de	Dirección de correo	de	Contraseña	Existe	Habilitado	Resultado
1	CE1, CE6, CE8, CE10		josep@gmail.com		123456jp	true	true	
2	CE2, CE7, CE8, CE10		""		""	true	true	
3	CE3, CE4, CE5, CE6, CE8, CE10		.jos@ép@gmail.com		123456jp	true	true	
4	CE1, CE6, CE9, CE10		josep@gmail.com		123456jp	false	true	
5	CE1, CE6, CE8, CE11		josep@gmail.com		123456jp	true	false	








Elaborado por: Cadena & Perugachi

Tabla 23. Clase equivalencia para: Gestión asignar cama

CLASES DE EQUIVALENCIA			
Condición de Entrada	Tipo	Clase Equivalencia Valida	Clase Equivalencia No Valida
Paciente	Valor	CE1: Valor numérico correspondiente al número de archivo del paciente.	CE2: En blanco. CE3: Valor no numérico.
Existe	Lógica	CE4: true	CE5: false
Cama	Valor	CE6: Valor numérico correspondiente al identificador de la cama.	CE7: En blanco CE8: Cama ocupada. CE9: Cama en desinfección.
Diagnósticos	Valores	CE10: Al menos un valor alfanumérico correspondiente al código del diagnóstico.	CE11: Ningún valor.

Elaborado por: Cadena & Perugachi

Tabla 24. Caso de Prueba para Gestión asignar cama

CASOS DE PRUEBA						
N° Caso	Clase de Equivalencia	Paciente	Existe	Cama	Diagnósticos	Resultado
1	CE1, CE4, CE6, CE10	25	true	151	{Z35.5}	 Cama asignada correctamente
2	CE2, CE4, CE6, CE10		true	151	{Z35.5}	 Debe ingresar un paciente
3	CE1, CE5, CE6, CE10	25	false	151	{Z35.5}	 No se encontraron resultados [500]
4	CE1, CE4, CE7, CE10	25	true		{Z35.5}	 Debe seleccionar una cama
5	CE1, CE4, CE6, CE11	26	true	151		 Debe ingresar al menos un diagnostico
6	CE3, CE4, CE6, CE10	asd*-	true	151	{Z35.5}	 Solo datos numericos.
7	CE1, CE4, CE8, CE10	25	true	152	{Z35.5}	 La cama se encuentra ocupada!

8	CE1, CE4, CE9, CE10	25	true	153	{Z35.5}	 La cama se encuentra en desinfección!
---	------------------------	----	------	-----	---------	---



Elaborado por: Cadena & Perugachi

Tabla 25. Clase equivalente para: Gestión de cambio/egreso

CLASES DE EQUIVALENCIA			
Condición de Entrada	Tipo	Clase Equivalencia Valida	Clase Equivalencia No Valida
Paciente	Valor	CE1: Valor numérico correspondiente al número de archivo del paciente.	CE2: En blanco. CE3: Valor no numérico.
Causa de cambio o egreso	Conjunto	CE4: Cambio de cama CE5: Traslado de servicio CE6: Referencia CE7: Contrareferencia CE8: Egreso / Alta CE9: Defunción	CE10: Ningún valor
Cama	Valor	CE11: Valor numérico correspondiente al identificador de la cama.	CE12: En blanco CE13: Cama ocupada. CE14: Cama en desinfección.
Diagnósticos	Valores	CE15: Al menos un valor alfanumérico correspondiente al código del diagnóstico.	CE16: Ningún valor.

Elaborado por: Cadena & Perugachi

Tabla 26. Caso de prueba gestión cambio/egreso

CASOS DE PRUEBA						
N° Caso	Clase de Equivalencia	Paciente	Causa de cambio o egreso	Cama	Diagnóstico	Resultado
1	CE1, CE4, CE11, CE15	25	Cambio de cama	151	{Z35.5}	 Cama asignada correctamente
2	CE2,CE4, CE11,CE16	25	Cambio de cama	151	{Z35.5}	 Debe ingresar un paciente

3	CE1,CE4, CE11,CE15	25	Cambio de cama	151	{Z35.5}	! Debe seleccionar una cama
4	CE1,CE4, CE11,CE15	26	Cambio de cama	151		! Debe ingresar al menos un diagnostico
5	CE1,CE4, CE11,CE15	25	Cambio de cama	152	{Z35.5}	✗ La cama se encuentra ocupada!
6	CE1,CE4, CE11,CE15	25	Cambio de cama	153	{Z35.5}	! La cama se encuentra en desinfeccion!
7	CE1,CE5, CE11,CE15	25	Trasferenc ia de servicio	151	{Z35.5}	✓ Cama asignada correctamente
8	CE2,CE5, CE11,CE16	25	Trasferenc ia de servicio	151	{Z35.5}	! Debe ingresar un paciente
9	CE1,CE5, CE11,CE15	25	Trasferenc ia de servicio	151	{Z35.5}	! Debe seleccionar una cama
10	CE1,CE5, CE11,CE15	26	Trasferenc ia de servicio	151		! Debe ingresar al menos un diagnostico
11	CE1,CE5, CE11,CE15	25	Trasferenc ia de servicio	152	{Z35.5}	✗ La cama se encuentra ocupada!
12	CE1,CE5, CE11,CE15	25	Trasferenc ia de servicio	153	{Z35.5}	! La cama se encuentra en desinfeccion!
13	CE1,(CE6 o CE7 o CE8 o CE9),CE11, CE15	25	Referencia Contrarefe rencia Egreso / Alta Defunción		{Z35.5}	✓ Informacion actualizada correctamente

14	CE1,CE10, CE11,CE15	25		152	{Z35.5}	
----	------------------------	----	--	-----	---------	--

Elaborado por: Cadena & Perugachi

Tabla 27. Clase equivalente para: Generar Reporte

CLASES DE EQUIVALENCIA			
Condición de Entrada	Tipo	Clase Equivalencia Valida	Clase Equivalencia No Valida
Indicador	Conjunto	CE1: Reporte de ocupación de camas. CE2: Ocupación por servicio. CE3: Ocupación por paciente. CE4: Historial de paciente. CE5: Tasas de mortalidad y morbilidad. CE6: Egresos / Altas. CE7: Ocupación por diagnóstico.	CE8: Ningún valor
Fecha inicial	Valor	CE9: Valor de tipo fecha en el siguiente formato: dd/mm/yyyy o dd-mm-yyyy	CE10: Ningún valor
Fecha final	Valor	CE11: Valor de tipo fecha en el siguiente formato: dd/mm/yyyy o dd-mm-yyyy. CE12: Valor mayor o igual a la entrada CE9	CE13: Ningún valor CE14: Fecha final menor a la inicial

Elaborado por: Cadena & Perugachi

Tabla 28. Caso de prueba para Generar reportes

CASOS DE PRUEBA					
N° Caso	Clase de Equivalencia	Indicador	Fecha Inicial	Fecha Final	Resultado
1	CE1,CE9, CE11,CE12	Reporte de ocupación de camas	15/05/2020	15/06/2020	

2	CE8, CE10, CE13				✘ Debe seleccionar un servicio, sala y una fecha!
3	CE1,CE10, CE13	Reporte de ocupación de camas			✘ Debe seleccionar una fecha!
4	CE1,CE9, CE11,CE14	Reporte de ocupación de camas	15/05/202 0	15/03/202 0	✘ La fecha final debe ser mayor o igual a la inicial!

Elaborado por: Cadena & Perugachi

Pruebas de Carga

Las pruebas de carga se realizan para conocer el rendimiento del software con una carga variable de usuarios bajo condiciones esperadas. Se consideró las transacciones o peticiones más relevantes o que puedan someterse a mayor demanda de recursos por los usuarios.

Tabla 29. Características del hardware donde se realizaron las pruebas de carga y estrés.

Características del Hardware	
Procesador	Intel® Core™ i5-2430M
Núcleos	4
Velocidad CPU	2.40 GHz
Memoria RAM	6 GB
SO	Ubuntu 16
Disco	HDD 500 GB

Elaborado por: Cadena & Perugachi

Tabla 30. Peticiones Http de las que se tomó las muestras.

Petición	Tipo	Protocolo	Puerto
Login	POST	http	80
Nueva Cuenta	POST	http	80
Admisión Paciente	GET	http	80
Listar Pacientes	GET	http	80
Asignar Cama	GET	http	80
Cambio Cama	GET	http	80

Elaborado por: Cadena & Perugachi

Con la utilización de la herramienta Apache JMeter se sometió a las peticiones http del software a pruebas de carga incrementales de 100, 200, 300 y 400 usuarios concurrentes. Para

obtener una simulación más cercana a la realizada, se configuro un temporizador aleatorio uniforme capaz de agregar pausas entre transacciones ya que, realmente el comportamiento del software está determinado por el usuario que lo emplea y técnicamente no todos los usuarios se comportan o interactúan con los sistemas de la misma manera.

Temporizador Aleatorio Uniforme:

El temporizador aleatorio uniforme retrasa cada solicitud de usuario durante un período de tiempo aleatorio.

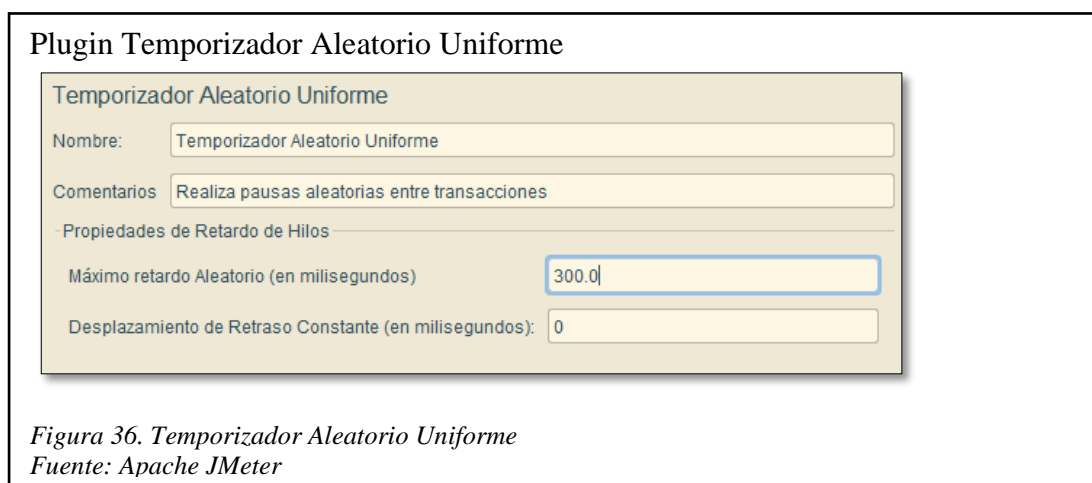


Figura 36. Temporizador Aleatorio Uniforme
Fuente: Apache JMeter

Tabla 31. Parámetros del temporalizador

ATRIBUTO	DESCRIPCION
Nombre	Nombre descriptivo para este temporizador
Máximo retardo Aleatorio	Número aleatorio máximo de milisegundos para retrasar.
Desplazamiento de Retraso Constante	Valor adicional en milisegundos.

Elaborado por: Cadena & Perugachi

Una vez realizadas las pruebas se obtuvo los siguientes resultados.

Tabla 32. Resultado de las pruebas de carga

Usuarios	# Muestras	Media	Min	Max	Desv. Estandar	Error %	Rendimiento	Recibido KB/sec	Enviado KB/sec	Media. Bytes
100	600	3358	157	21314	4883,22	0,00%	18,72601	186,96	5,28	10223,8
200	1200	7360	149	55304	10180,63	0,00%	17,47997	174,52	4,92	10223,8
400	2400	7944	167	48851	10116,06	44,00%	30,04657	226,64	4,68	7723,9

Elaborado por: Cadena & Perugachi

Pruebas de Estrés

Las pruebas de estrés se realizan para conocer la solidez del software, se somete al software a cargas incrementales hasta que este se rompa o deje de responder de manera eficiente. Para ello al igual que a las pruebas de carga, se utilizó las mismas peticiones y los mismos recursos de hardware.

Una vez realizadas las pruebas de estrés se obtuvo los siguientes resultados.

Tabla 33. Resultados de las pruebas de estrés

Usuarios	Muestras	Media	Min	Max	Desv. Est.	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
100	600	3358	157	21314	4883,22	0,00%	18,72601	186,96	5,28	10223,8
200	1200	7360	149	55304	10180,63	0,00%	17,47997	174,52	4,92	10223,8
400	2400	7944	167	48851	10116,06	44,00%	30,04657	226,64	4,68	7723,9
800	4800	7256	214	54315	11881,5	66,48%	43,9919	241,75	4,08	5627,2
1600	9600	5047	204	54339	8295,77	82,89%	93,41611	385,97	4,43	4230,9
3200	19200	3920	223	81475	6152,33	91,55%	177,931	600,84	4,17	3457,9
6400	38400	4942	163	96989	5899,11	96,27%	332,03057	991,86	3,41	3058,9
12800	76800	4935	1	110896	7353,81	96,79%	380,12839	1110,58	3,44	2991,7

Elaborado por: Cadena & Perugachi

Análisis de resultados de las pruebas

Pruebas de caja negra

Una vez realizadas las pruebas de caja negra mediante la técnica de particionamiento equivalente se comprueba que, en todos los casos de prueba realizados las salidas mostradas por el software son iguales a las salidas esperadas de cada uno de los requerimientos funcionales, en la *tabla 34* se muestra a detalle las historias de usuario y los requerimientos funcionales cubiertos por las pruebas de caja negra.

Tabla 34. Resultados de las pruebas de caja negra

Historia de Usuario	Requisitos Funcionales	Casos de prueba realizados	Salidas esperadas vs Salidas mostradas
Creación de nueva cuenta	RF1, RF2	5	5 de 5
Ingreso al sistema	RF1, RF2	5	5 de 5
Asignación cama	RF4, RF6, RF7, RF8, RF9, RF11, RF13, RF18, RF29	8	8 de 8
Cambio o egreso de paciente	RF12, RF16, RF17, RF22	14	14 de 14
Gestión de reportes	RF5, RF14, RF15, RF19, RF23	4	4 de 4

Elaborado por: Cadena & Perugachi

Pruebas de carga

Según los resultados alcanzados en la *Tabla 32*, se evidencia que el software sometido a una carga de hasta 200 usuarios concurrentes responde eficientemente con un error promedio del 0,00% y un rendimiento promedio de 17,48 solicitudes procesadas por segundo, con lo que se cumple el Requerimiento No Funcional RNF4 en el que dice que el 95% de las transacciones deben realizarse en menos de 3 segundos.

Explicación:

Muestras o transacciones: 1200

Tiempo de ejecución: 1 segundo

1200 → 1 "*Transacciones en 1 segundo*"

$T \rightarrow 2,99$ "*Transacciones en menos de 3 segundos*"

$T = 2,99(1200)$

$T = 3588$

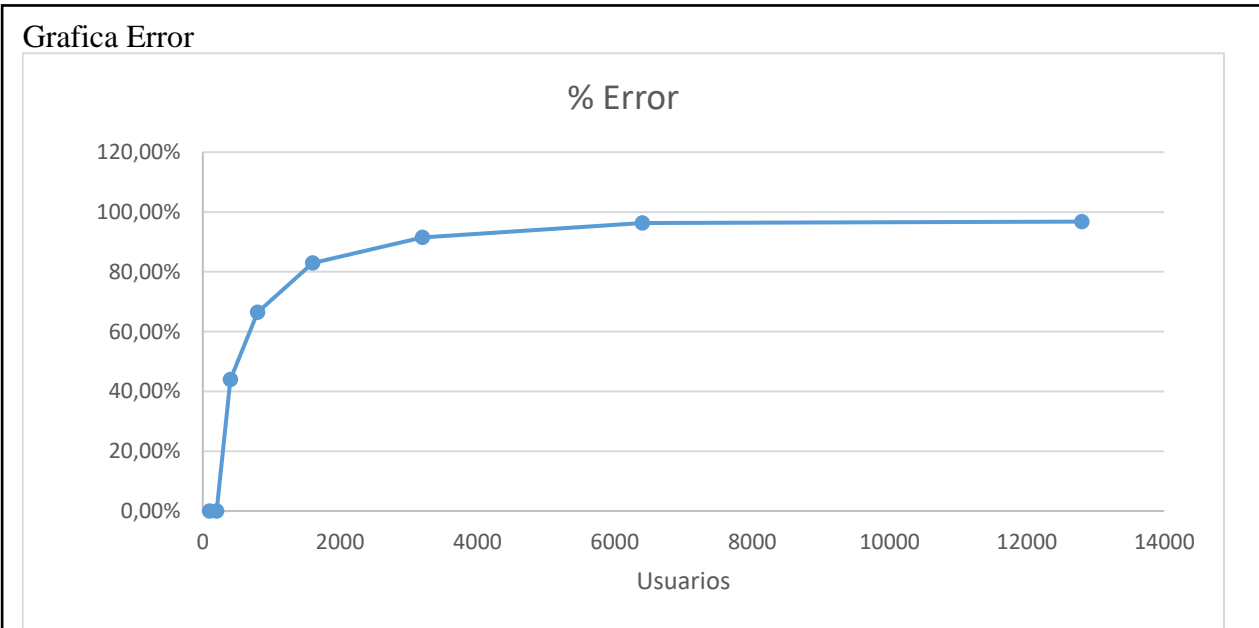
El sistema podrá procesar 3588 transacciones en menos de 3 segundos a un 100% de efectividad, con esto supera el 95% solicitado que representa un total de 3408,6 transacciones.

Al someter al software a una carga de 300 usuarios se observa un error promedio del 17,89% con un rendimiento de 21,96 solicitudes procesadas por segundo.

De acuerdo con los resultados el sistema es capaz de operar adecuadamente hasta con 200 usuarios con sesiones concurrentes en periodos de un segundo con lo que se cumple y supera el Requisito No Funcional (RNF5) que solicita la operación adecuada del software hasta con 100 usuarios concurrentes.

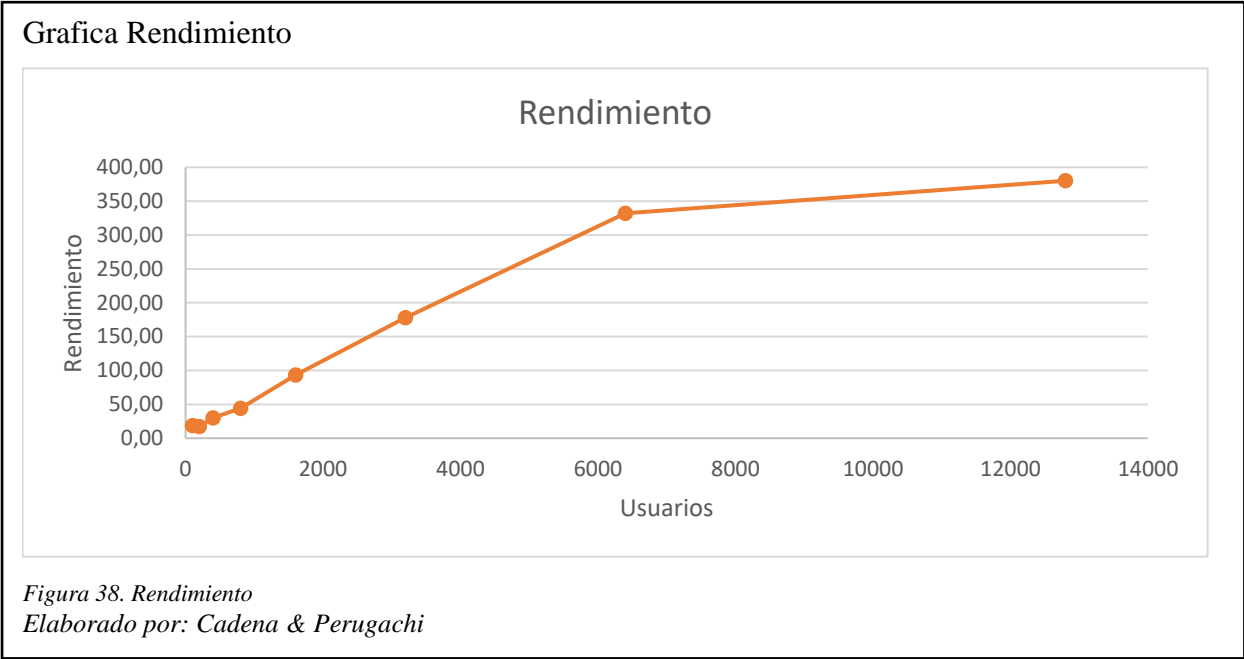
Pruebas de estrés

Según los resultados conseguidos en la *Tabla 33*. se observa que el software con una carga de 400 usuarios concurrentes en periodos de un segundo muestra un error promedio de 44% y un rendimiento de 30,04 solicitudes procesadas por segundo. A partir de los 400 usuarios se observa un incremento exponencial en el porcentaje de error como se muestra en la siguiente Figura registrando un error máximo del 96,79% con una carga de 12800 usuarios.

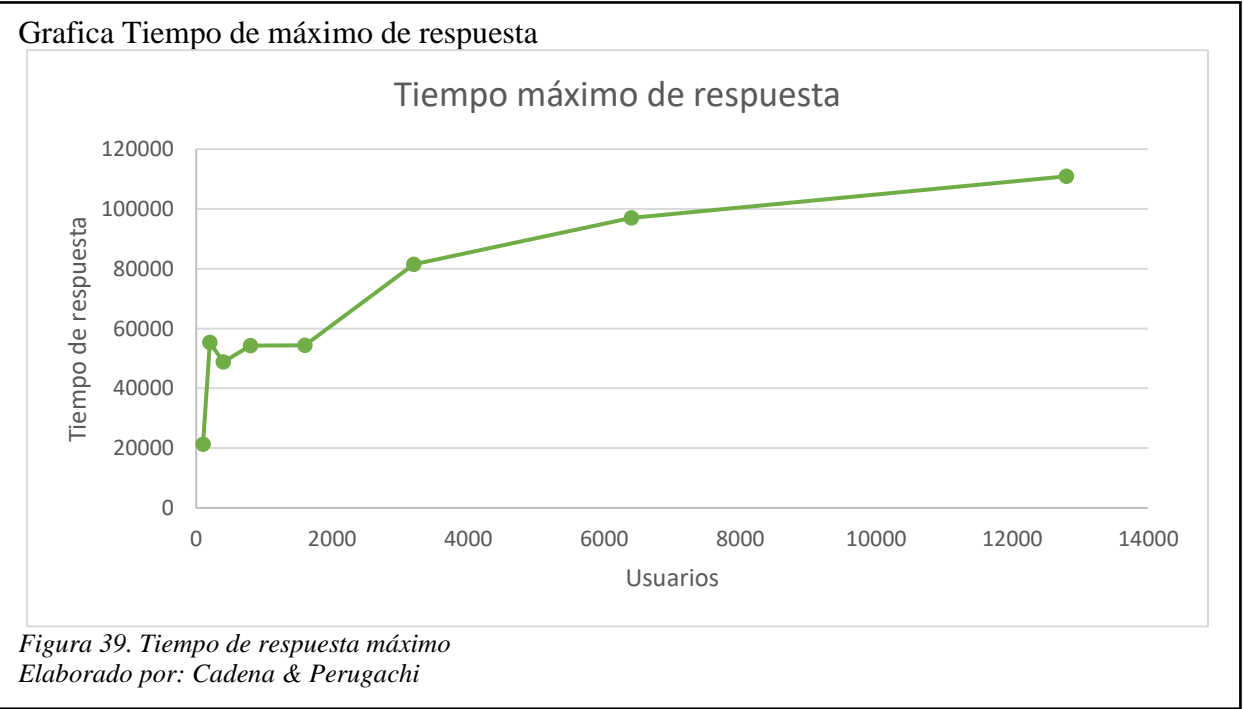


*Figura 37. Porcentaje de Error
Elaborado por: Cadena & Perugachi*

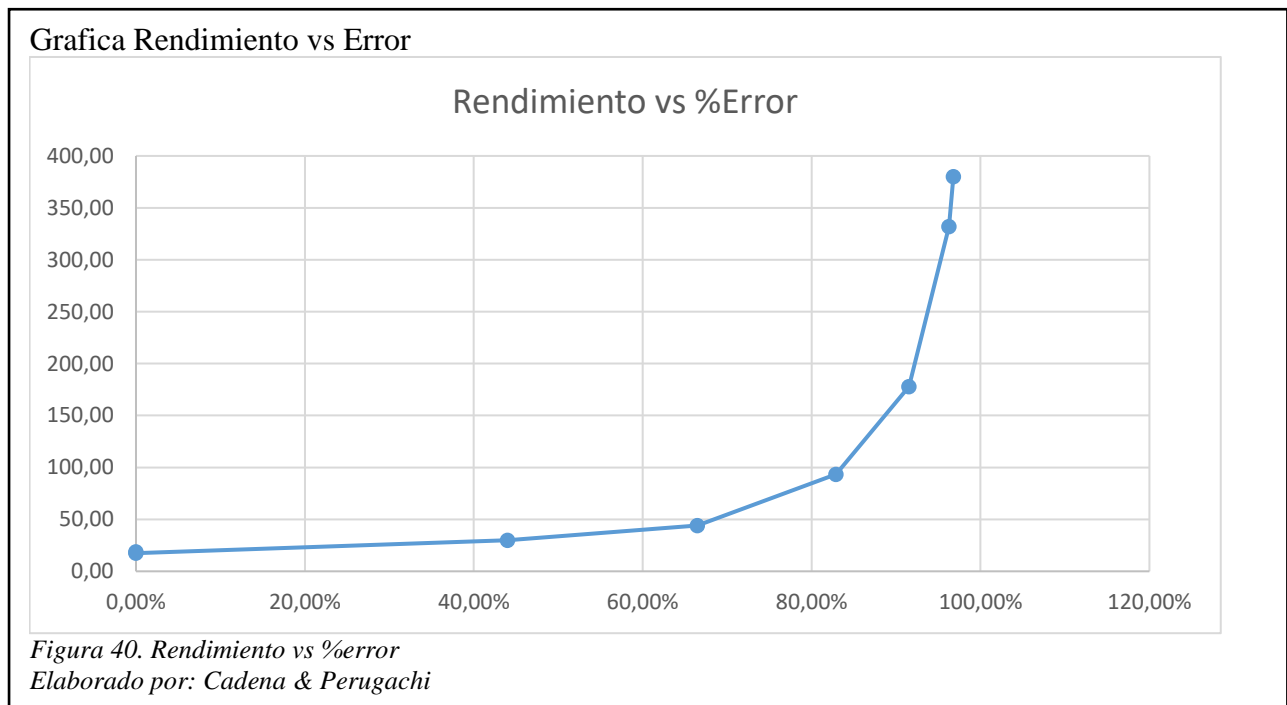
Respecto al rendimiento, conforme la carga de usuarios es duplicada el rendimiento del software se incrementa llegando a un pico máximo de 380,13 solicitudes procesadas por segundo. Ya que el rendimiento viene dado por el número de solicitudes procesadas en un determinado periodo de tiempo, a mayores usuarios mayores serán las solicitudes por procesar.



El tiempo máximo de respuesta es el tiempo máximo que tarda el software en procesar una solicitud, el valor más alto que se registro fue de 110896 milisegundos en la prueba de 12800 usuarios simultáneos.



Según la *Figura 37* de rendimiento, erróneamente se diría que a mayores usuarios simultáneos el software tiene un mejor rendimiento, información lejana a la realidad, ya que si lo compara con el error generado a la carga masiva de usuarios es diferente su apreciación, a continuación, un gráfico representativo del rendimiento del software con respecto al error



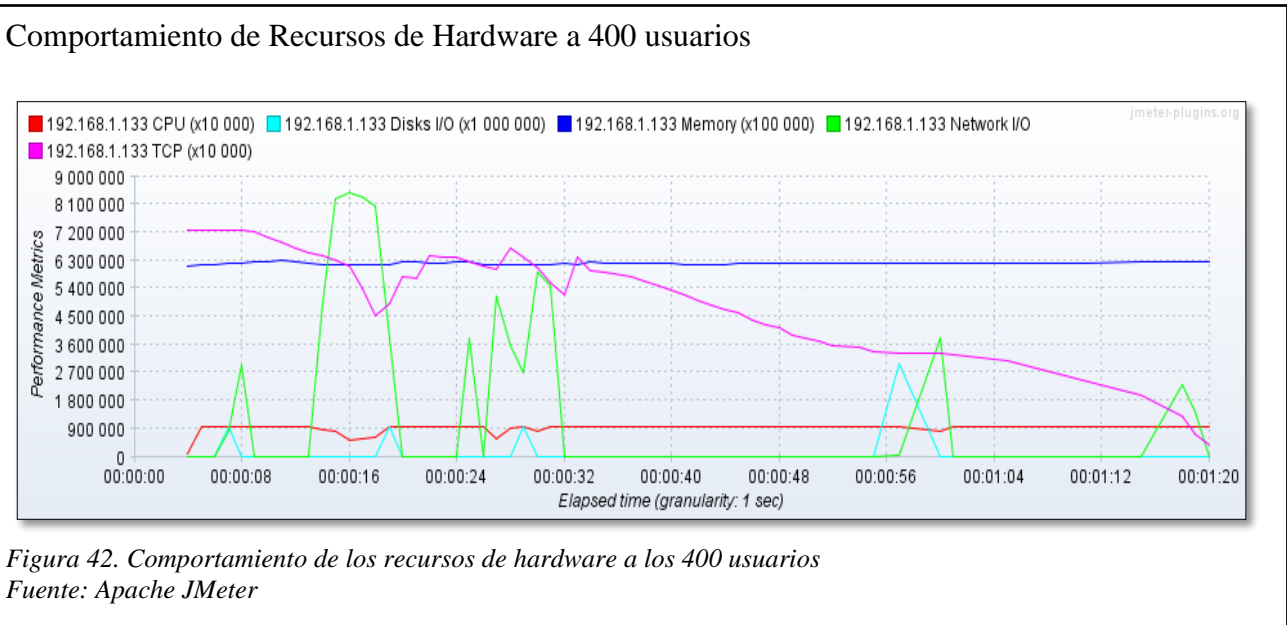
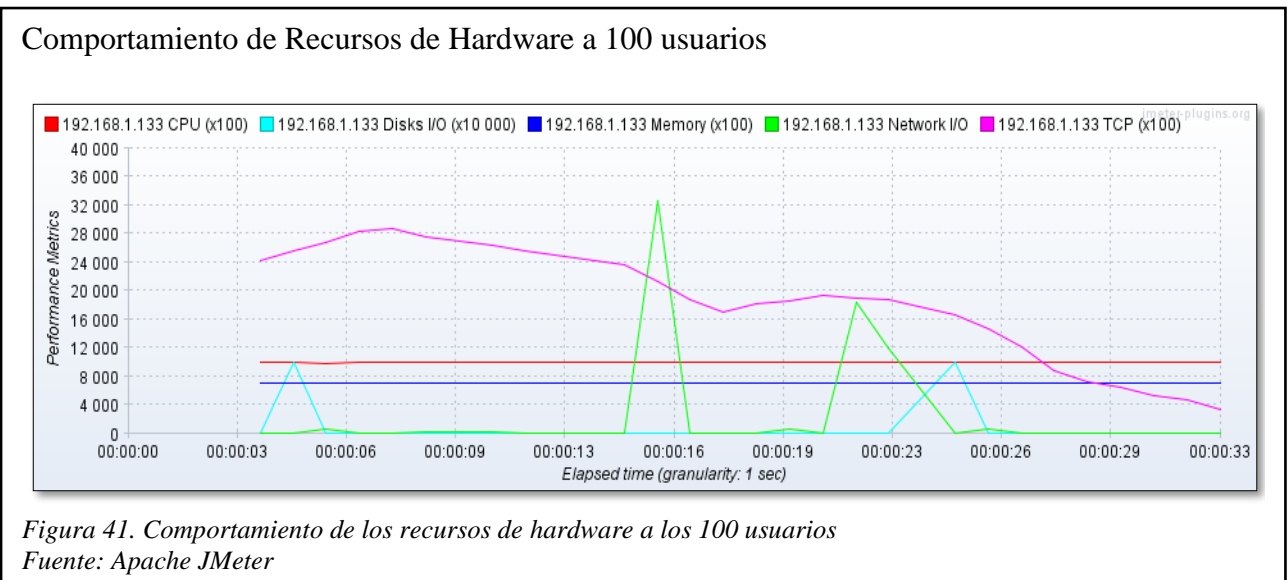
Como la Figura muestra a mayor rendimiento es decir si las solicitudes por procesar incrementan, el porcentaje de error aumenta significativamente.

Las pruebas y resultados analizados ayudan a evidenciar el comportamiento del software del lado del cliente, es decir para tener una apreciación completa se debe monitorear también el lado del servidor, el hardware que soportará todas las peticiones.

Para dicho monitoreo Apache JMeter y el plugin **PerfMon Metrics Collector** proporciona un reporte gráfico del comportamiento y demanda de recursos de hardware en tiempo real al usuario.

PerfMon Metrics Collector, permite medir el consumo de CPU, el consumo de Memoria, entradas y salidas de red, lecturas y escrituras de disco, comunicaciones TCP, entre otras.

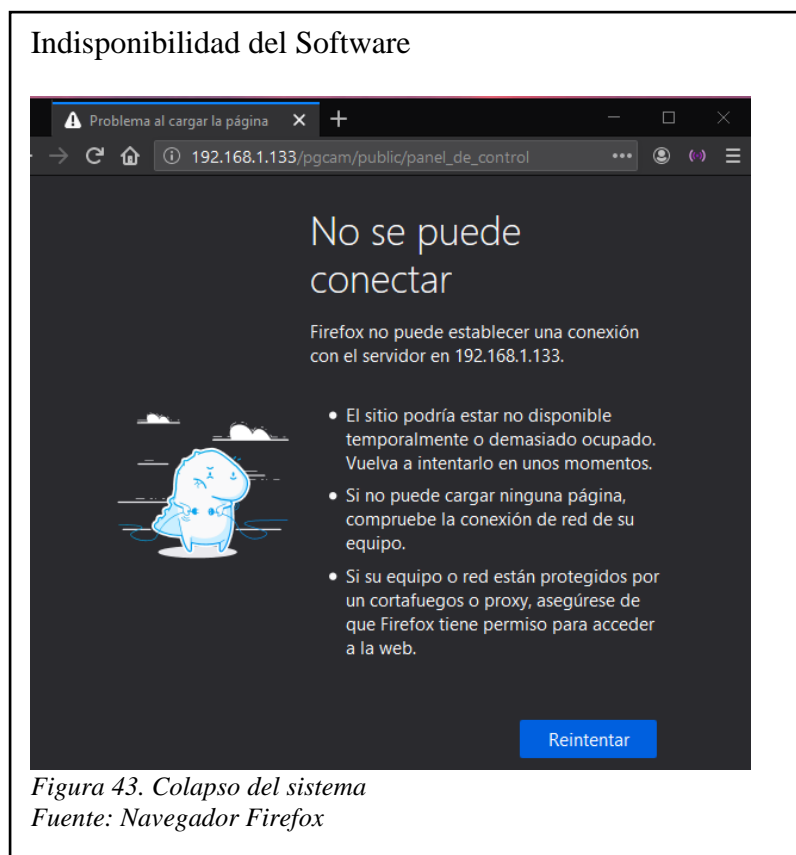
Aplicando la carga masiva de usuarios y con el plugin **PerfMon Metrics Collector** fue posible obtener los siguientes resultados:



Lo más evidente es el incremento en consumo de memoria que va de 7000 a 6300000 aproximadamente, registrando un aumento del 99,88%. Logrando con esto que el software colapse. Se puede apreciar que este aumento es relativamente alto, pero se debe tomar en cuenta

que son 400 usuarios simultáneos en 1 segundo, en situaciones reales este comportamiento es casi imposible de presentarse ya que para un inicio al sistema accederán no más de 50 usuarios no concurrentes. Para dar un apoyo o solución a este cuello de botella que pueda presentarse en un futuro se implementó el uso de AJAX reduciendo la carga de peticiones en el servidor al ejecutar varios procesos del lado del cliente.

Indisponibilidad del software sometido a un estrés de 400 usuarios simultáneos en periodos de 1 segundo:



CONCLUSIONES

- La aplicación y validación de estrategias y metodologías de desarrollo ágil de software permitió la finalización del proyecto de manera exitosa en términos de cumplimiento de metas, diseño y presupuesto (en cuanto a alcances y disponibilidad).
- El uso de tecnologías de software libre para el desarrollo del proyecto, a más de cumplir con la política gubernamental (decreto 1014) sobre el uso de software libre en instituciones públicas del Ecuador, redujo los costos de desarrollo significativamente.
- La emergencia sanitaria originada por el nuevo coronavirus (SARS-CoV-2) sin duda alguna ofuscó el desenvolvimiento normal para el desarrollo del proyecto, por lo que el uso de tecnologías de comunicación en conjunto con la participación y colaboración constante de los involucrados del proyecto fueron clave para llevar a cabo el desarrollo del proyecto durante toda su etapa.
- La utilización de la arquitectura Modelo Vista Controlador (MVC) permitió una fácil gestión del software, al tener todo organizado por capas facilita la modificación, incorporación o eliminación de módulos al sistema, reduciendo el uso de tiempo y recursos.
- La incorporación de un módulo para la gestión de entidades del sistema permite una administración fácil e integral de cada una de las entradas y variables que lo conforman, al organizar, estandarizar y automatizar la información de entre los diferentes servicios.
- Gracias a la implementación de un módulo que muestre en las pantallas de sala de espera la ubicación actual de los pacientes internos, los familiares podrán visualizar dicha información directamente, mejorando el proceso actual de visitas el reducir tiempo en consultas en salas de información.

RECOMENDACIONES

- La correcta aplicación de una metodología de desarrollo en conjunto con la utilización de buenas prácticas en la gestión de proyectos de software aumenta las posibilidades de éxito de este, es por ello se recomienda mantener una comunicación directa y constante con los involucrados del proyecto sobre todo en la etapa de análisis y codificación con el fin de minimizar los riesgos, facilitar el uso de recursos y mejorar los resultados.
- Es recomendable que las tareas de mantenimiento del sistema deberá realizarlas un miembro de la unidad de TICS que cuente con un usuario de perfil ADMINISTRADOR, dicho mantenimiento debe ser preventivo y correctivo tanto como para las diferentes entradas del sistema como para la base de datos.
- Se recomienda que para una nueva versión o actualización del sistema se capacite a los médicos encargados de realizar el Triage a los pacientes, para que sean ellos quienes ingresen el diagnostico de cada paciente y puedan cerrar el formulario 008.
- Se recomienda que la base de datos se encuentre alojada en un servidor dedicado, y así poder asegurar un mejor rendimiento a lo largo del tiempo.
- Para el desarrollo e incorporación de nuevos módulos al sistema se recomienda conocer a fondo acerca de Zend Framework y así mantener homogeneidad y una estructura ordenada en el manejo del código fuente.
- Para futuros proyectos o implementación de nuevos módulos se recomienda considerar los datos de entrada y salida que se requieran y así evitar la redundancia de información garantizando una correcta comunicación entre módulos y/o sistemas.
- Se recomienda que para la incorporación de un nuevo módulo al sistema se deba realizarlo en una nueva rama del repositorio público del proyecto alojado en GitHub, garantizando así la integridad del proyecto manteniendo un control de cambios y versiones.

- De acuerdo con políticas y normativas que exigen se tenga una firma del médico tratante en los formularios físicos, se recomienda la digitalización de formularios incentivando el uso de la firma electrónica con esto se obtendrá una notable reducción en el uso de papel, información centralizada y agilidad en los procesos.
- Se recomienda la estandarización de datos de entre los diferentes sistemas utilizados en el hospital con el fin de garantizar la intercomunicación de estos o su vez, la creación de web services que faciliten el envío y recepción de datos entre sistemas, evitando la redundancia de datos y trabajo.

REFERENCIAS

Bibliografía

- Abrahamsson, P., Salo, O., & Ronkainen, J. &. (2002). *Agile software development methods: Review and analysis*. Espoo: VTT Publications 478.
- Ali, J. (2013). *Instant Node Package Manager*. Van Haren Publishing.
- Cabello, J. C. (2017). *Diseño de páginas Web con XHTML, JavaScript y CSS*. España: RA-MA.
- Mariscal, A. B. (2015). *Diseño de bases de datos relacionales*. España: Elearning S.L.
- Mesa, J. M., Sanz, M. L., & Granada, D. (2014). *Desarrollo Web en Entorno Cliente*. España: RA-MA. Obtenido de <https://si.ua.es/es/documentacion/mootools/ajax.html>
- Pressman, R. S. (2010). *Ingeniería de software un enfoque practico*. México: McGraw-Hill.
- Pública, M. d. (2014). *Acuerdo Ministerial*. Quito.
- R. Orfali, D. H. (2002). *Cliente/Servidor y objetos: Guía de Supervivencia*. Mexico: Edición McGraw -Hill Interamericana.
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de bases de datos*. Madrid: McGraw Hill.

Sitios Web

- B., G. (01 de 11 de 2019). *Hostinger Tutoriales*. Obtenido de ¿Qué es Apache? Descripción completa del servidor web Apache: <https://www.hostinger.es/tutoriales/que-es-apache/>
- Bustos, G. (2019). *El lenguaje de programación PHP*. Obtenido de <https://redgrafica.com/el-lenguaje-de-programacion-php/#:~:text=El%20lenguaje%20PHP%20siempre%20va,el%20uso%20de%20servidores%20linux.&text=PHP%20se%20caracteriza%20por%20ser%20un%20lenguaje%20gratis%20y%20multiplataforma>.
- Desarrollo Web*. (08 de 11 de 2018). Obtenido de Qué es Sass, como usar Sass: <https://desarrolloweb.com/articulos/que-es-sass-usar-sass.html>
- Ecuador, D. (22 de 12 de 2014). *Derechoecuador.com*. Obtenido de Registro Oficial: <https://www.derechoecuador.com/registro-oficial/2015/02/registro-oficial-no-402---lunes-22-de-diciembre-de-2014-suplemento>
- Fm, Y. (30 de 10 de 2019). *Xataka*. Obtenido de Qué es Github y qué es lo que le ofrece a los desarrolladores: <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- González, Y. (12 de 06 de 2012). *Telemática Magazine*. Obtenido de Patrón Modelo-Vista-Controlador: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>

Pública, M. d. (s.f.). *Hospital Gineco Obstetrico de Nueva Aurora "LUZ ELENA ARISMENDI"*.
Obtenido de Misión y Visión:
http://hgona.gob.ec/index.php?option=com_content&view=article&id=64&Itemid=287

Team, B. (2020). *Bootstrap* . Obtenido de <https://getbootstrap.com/>

Zend, R. W. (s.f.). *Zend Framework*. Obtenido de About - Zend Framework:
<https://framework.zend.com/about>

Tesis / trabajos de titulación

AGUILAR RIVERA, R. A., MENDOZA MENCOS, D. J., & TOBIAS RIVAS, M. J. (Agosto de 2013). *SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE SERVICIOS MÉDICOS EN LA UNIDAD DE HEMATO-ONCOLOGÍA DEL HOSPITAL NACIONAL ROSALES* . Obtenido de <http://ri.ues.edu.sv/id/eprint/4508/1/Sistema%20inform%C3%A1tico%20para%20la%20administraci%C3%B3n%20de%20servicios%20m%C3%A9dicos%20en%20la%20unidad%20de%20hematolog%C3%ADa%20del%20Hospital%20Nacional%20Rosales.pdf>

MOLINA CALVOPÍÑA, R. G., & COLLAGUAZO LOACHAMIN, J. C. (2008). *SOFTWARE DE MANEJO DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS MÉDICAS PARA LA CLÍNICA DE LA FUERZA AÉREA ECUATORIANA DEL ALA No. 11 DE QUITO*.
Obtenido de <http://repositorio.utc.edu.ec/bitstream/27000/459/1/T-UTC-1028.pdf>

SÁNCHEZ, D. V., NAVA, M. E., & PEREYRA, C. U. (2013). *Universidad Autónoma de México*. Obtenido de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4164/Tesis.pdf?sequence=1>