

UNIVERSIDAD NACIONAL DE CHIMBORAZO



FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Proyecto de Investigación previo a la obtención del título de Ingeniero en Sistemas y
Computación

TRABAJO DE TITULACIÓN

Proyecto de Investigación

**Desarrollo de un sistema de gestión de quirófanos para el Hospital Universitario
Andino aplicando GraphQL.**

Autor: Byron Fernando Paguay Duchi

Tutor: Mcs. Jorge Delgado

Riobamba - Ecuador

Año 2021

Veredicto de la investigación

Los miembros del Tribunal de Graduación del proyecto de investigación de título:

“DESARROLLO DE UN SISTEMA DE GESTIÓN DE QUIRÓFANOS PARA EL HOSPITAL UNIVERSITARIO ANDINO APLICANDO GRAPHQL”, presentado por el estudiante: Sr. Byron Fernando Paguay Duchi y dirigido por el MsC. Jorge Delgado.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Por Constancia de lo expuesto firman:



MsC. Jorge Delgado

.....

Tutora del Proyecto



MsC. Ana Congacha

.....

Miembro del Tribunal



MsC. Milton Lopez

.....

Miembro del Tribunal

Auditoria de la investigación

“La responsabilidad del contenido de este proyecto de investigación corresponde exclusivamente a: Byron Fernando Paguay Duchi con la dirección del Mcs. Jorge Delgado y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”

A handwritten signature in blue ink, appearing to read 'Jorge Delgado', with a horizontal line drawn through it.

Mcs. Jorge Delgado

060275938-3

Tutor de Proyecto de Investigación

A handwritten signature in blue ink, appearing to read 'Byron Fernando Paguay Duchi', with a horizontal line drawn through it.

Byron Fernando Paguay Duchi

060478967-7

Autor del Proyecto de Investigación

Dedicatoria

Este trabajo de investigación se lo quiero dedicar a mis hermanos y hermanas, A mi mamá Maria que siempre ha estado junto a mi apoyándome en los malos y buenos momentos que como persona he tenido, a mi papá Ángel que siempre me ha inculcado buenos valores en la vida y me ha apoyado en cada meta que me he propuesto, permitiéndome convertirme en una persona de bien y con buenos valores.

Byron Fernando Paguay Duchi

Agradecimiento

Agradezco a mi padres, María y Ángel, porque en los momentos más difíciles de mi vida me han apoyado incondicionalmente sean momentos buenos o malos durante mi transcurso como hijo, por inculcarme buenos valores en la vida que con el tiempo será recompensados.

Byron Fernando Paguay Duchi

Índice General

Veredicto de la investigación	II
Auditoria de la investigación.....	III
Dedicatoria.....	IV
Agradecimiento	V
Índice de tablas	IX
Índice de figuras	X
Resumen	XI
Abstract.....	XII
1. INTRODUCCIÓN	1
2. CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA	3
2.1 Problema y Justificación	3
3. OBJETIVOS:	5
3.1. Objetivos Generales:.....	5
3.2. Objetivos Específicos:	5
4. CAPÍTULO II. MARCO TEÓRICO	6
4.1 Sistemas web en el sector de la salud	6
4.2 Servicios web.....	6
4.3 Calidad de software	7
4.4 Norma SQuaRE ISO/IEC 25000	7
4.4.1. Norma ISO/IEC 25010.....	7
4.4.2. La eficiencia como parámetro de calidad.....	8
4.5 Base de Datos No SQL	8
4.5.1 MongoDB	8
4.6 GraphQL	9
4.6.1 Operaciones	9
4.6.2 Schema	10

4.6.3	Querys.....	10
4.6.4	Mutations	11
4.6.5	Subscriptions	11
4.6.6	Tipos de objetos.....	11
4.6.7	Argumentos	12
4.6.8	Tipos de enumeración.....	12
4.6.9	Tipos de entrada.....	12
4.6.10	Variables.....	12
4.6.11	Resolvers	12
4.7	GraphQL – Apollo	13
4.8	ReactJS.....	13
4.9	Apache JMeter	13
4.10	Metodología Scrum.....	13
5.	CAPITULO III METODOLOGÍA.....	15
5.1	Metodología.....	15
5.2	Identificación de variables	15
5.3	Tipo de Investigación.....	16
5.4	Enfoque de la investigación	16
5.5	Según la fuente de investigación.....	16
5.6	Unidad de análisis	16
5.7	Población de estudio y tamaño de la muestra	16
5.8	Técnicas de recolección de datos.....	18
5.9	Técnicas de análisis e interpretación de la información	18
5.10	Operacionalización de variables	19
5.11	Especificación de las características, subcaracterísticas y métricas de la norma ISO/IEC 25010.....	20
5.12	Análisis de niveles de puntuación para las métricas.....	22
5.13	Ponderación de las métricas.....	22
6.	CAPÍTULO IV: RESULTADOS Y DISCUSIÓN	23

6.1	Resultados	23
6.1.1	Análisis del lenguaje GraphQL	23
6.1.2	Desarrollo del sistema web de gestión de quirófanos para el Hospital Universitario Andino	24
6.1.3	Evaluación del rendimiento en el sistema web de gestión de quirófanos	28
6.2	Discusión	38
7.	Conclusiones	40
8.	Recomendaciones	41
9.	Referencias	42
10.	Anexos	44

Índice de tablas

Tabla 1: Rendimiento	8
Tabla 2. Definición de roles del proyecto.....	14
Tabla 3: Métricas y fórmulas.....	18
Tabla 4: Operacionalización de variables.....	19
Tabla 5: Métricas para la eficiencia de desempeño.....	21
Tabla 6: Rango de puntuación para las métricas	22
Tabla 7: Ponderación de la métrica	23
Tabla 8: Ventajas y desventajas de GraphQL	23
Tabla 9: 30 usuarios virtuales concurrentes (Tiempo de respuesta).....	29
Tabla 10: 60 usuarios concurrentes virtuales (tiempo de respuesta).....	30
Tabla 11: 30 usuarios virtuales concurrentes (tiempo de espera).....	31
Tabla 12:60 usuarios virtuales concurrentes (tiempo de espera).....	32
Tabla 13: 30 usuarios concurrentes(rendimiento)	33
Tabla 14:60 usuarios virtuales concurrentes(rendimiento)	33
Tabla 15: Resumen del resultado escenario1	33
Tabla 16: Resumen del resultado escenario 2	34
Tabla 17: Resumen de resultados escenario 1 y escenario 2.....	34

Índice de figuras

Figura 1. GraphQL server.....	9
Figura 2. Definición de esquema GraphQL	10
Figura 3. Query en GraphQL.....	10
Figura 4. Mutations en GraphQL	11
Figura 5. Resolvers en GraphQL.....	13
Figura 6: Metodología scrum	15
Figura 7: Fórmula de la ecuación estadística para proporciones poblacionales.....	17
Figura 8: Diagrama de procesos.	24
Figura 9: Módulo de gestión de horarios(Cuadrante).....	25
Figura 10: Módulo de agendamiento de quirófanos	25
Figura 11: Módulo de gestión de registros (Partos)	26
Figura 12: Módulo de gestión de registros (Cirugías).....	26
Figura 13: Módulo estadístico	27
Figura 14: Query obtener pacientes.....	28
Figura 15: Tiempo de respuesta.....	35
Figura 16: Tiempo de espera	36
Figura 17: Rendimiento	37

Resumen

En la presente investigación se realizó el análisis del rendimiento del sistema web de gestión de quirófanos del Hospital Universitario Andino desarrollado bajo la tecnología GraphQL, como tal se desarrolló 4 módulos: módulo de gestión de horarios, módulo de agendamiento de quirófano, módulo de registro de cirugías y partos, y el módulo de panel de gráficas para la toma de decisiones.

Mediante la aplicación de la norma SQuaRE ISO/IEC 25000 y su división la norma ISO/IEC 25010 haciendo uso de la subcaracterística comportamiento temporal: tiempo de espera, tiempo de respuesta, y rendimiento, se evaluó el comportamiento del sistema web.

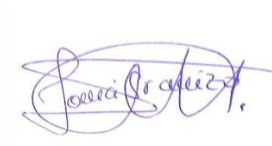
A través de un benchmark (apache JMeter) se obtuvo información sobre el comportamiento del servidor, tiempo de espera, tiempo de respuesta, y rendimiento, con el análisis de los resultados se llegó a concluir que el uso de GraphQL mejora notablemente el rendimiento del servidor, obteniendo un nivel de puntuación muy satisfactorio, con una calidad en uso del 86.1% y un déficit de calidad de 13.9%, de manera que esta ventaja se la obtiene gracias a la especificación de datos que se requiere en la consultas y las mismas son devueltas a través de un único endpoint disminuyendo la carga al servidor, a pesar del déficit de calidad del 13.9% el sistema cumple con los estándares establecidos por la norma SQuaRE ISO/IEC 25000.

Palabras Clave: GraphQL, Sistema de gestión de quirófanos, rendimiento.

Abstract

In the present research, the performance analysis of the operating room management web system of the Andino University Hospital developed under GraphQL technology was carried out, as such, four modules were developed: schedule management module, operating room scheduling module, registration module of surgeries, and deliveries, and the graphics panel module for decision making. By applying the SQuaRE ISO / IEC 25000 standard and its division, the ISO / IEC 25010 standard using the temporal behavior sub-characteristic: waiting for time, response time, and performance, the web's behavior system. Through a benchmark (apache JMeter), information was obtained on the behavior of the server, waiting time, response time, and performance; with the analysis of the results, it was concluded that the use of GraphQL significantly improves the performance of the server, obtaining a very satisfactory level of the score, with quality in the use of 86.1% and a quality deficit of 13.9%, so that this advantage is obtained thanks to the data specification that is required in the consultations. They are returned through a single endpoint reducing the load on the server; despite the quality deficit of 13.9%, the system complies with the SQuaRE ISO / IEC 25000 standard standards.

Keywords: GraphQL, operating room management system, performance.



Reviewed by:
Mgs: Sonia Granizo Lara.
ENGLISH PROFESSOR.
c.c. **0602088890**

1. INTRODUCCIÓN

La evolución de los sistemas web, desde que se formó la World Wide Web en 1990 han estado acompañados de arquitecturas, que apoyan para que la accesibilidad a la información y comunicación sea más sencilla, las APIs fueron creadas como un modelo arquitectónico orientado a servicios para el consumo de información.

Debido a la existencia de diferentes tecnologías como SOAP, REST, que permiten métodos de comunicación, y altos niveles de interoperabilidad, para recuperar o crear o modificar datos han presentado falta de flexibilidad, si bien en algunos componentes pueden necesitar solo ciertas partes o campos de datos de todo el sistema, o incluso una combinación de campos que pertenecen a diferentes objetos de datos, lo que hace que se requiera una serie de peticiones a diferentes criterios de valoración para reunir todos los datos que necesitan. (Andrea Vazquez, 2017)

Mejorar esta falta de flexibilidad, capacidad de mantenimiento y problemas de rendimiento derivado de dichas tecnologías SOAP y REST, son lo que el lenguaje GraphQL está tratando. El objetivo principal de GraphQL es unificar las solicitudes de datos de un sistema en un solo punto final único, lo que reduce el número de solicitudes necesarias para recuperar los datos, sin embargo también permite que los componentes y los usuarios puedan especificar en sus propias peticiones de los campos particulares que están pidiendo, recuperando únicamente lo que necesitan para sus propósitos. (Andrea Vazquez, 2017)

El Hospital Universitario Andino, es un centro de atención primaria y de hospitalización, uno de los servicios son las cirugías que se realizan en áreas de quirófanos, en la actualidad para la reservación de estas cirugías se considera los siguientes criterios: La gestión de horarios de enfermeras y médicos, la agenda de quirófanos, la solicitud de kits quirúrgicos, y el registro de cirugías y partos, esta forma de gestionar la información limita la atención a los pacientes, y la toma de decisiones.

Los usuarios de los quirófanos en el Hospital Universitario Andino: Médicos, enfermeras, requieren una gestión de la información de quirófanos a través del uso de un sistema web que permita: planificar, reservar, consultar el uso de los quirófanos, por consiguiente surge la propuesta del presente proyecto para desarrollar un sistema web con tecnologías actuales que permitan manejar la información de una forma adecuada, y que facilite las tareas diarias para poder brindar un servicio de calidad, y a la toma de decisiones con un nivel estratégico

El sistema web se desarrollará bajo la tecnología GraphQL, lo cual permite optimizar la gestión de la información y se integraran los siguientes módulos: módulo de gestión de horarios, módulo de agendamiento de quirófanos, módulo de registro de cirugías y partos, y módulo de panel de gráficas.

Además, se evaluará el rendimiento del sistema utilizando la norma SQuaRE-ISO/IEC 25000.

2. CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA

2.1 Problema y Justificación

El Hospital Universitario Andino ofrece los servicios de atención primaria y hospitalización, entre los cuales se destaca los siguientes: tratamiento de enfermedades físicas, cirugías quirúrgicas, partos etc.

El proceso de quirófanos utiliza en la actualidad formatos manuales que demanda de los profesionales responsables que diariamente revisen la información, lo cual dificulta la gestión de la información y administración de los quirófanos, algunas de las actividades que se deben desarrollar son las siguientes:

- La gestión de horarios del personal, tanto de los médicos, enfermeras, auxiliares.
- El agendamiento de quirófanos.
- El registro de cirugías y partos.
- La solicitud de kits quirúrgicos y.
- La generación de gráficas para la toma de decisiones.

El utilizar procesos manuales puede ocasionar problemas como: demora al realizar una intervención quirúrgica, la falta de kits quirúrgicos, pérdida de tiempo al gestionar el horario del personal hospitalario, y la falta de información para la toma de decisiones.

En la actualidad al tomar una decisión acerca de cuáles kits quirúrgicos son los más demandados y cuáles no, sufre de contratiempos porque es necesario recolectar la información almacenada en repositorios físicos, para posteriormente realizar un análisis y entregar los resultados.

El desarrollo del sistema web de gestión de quirófanos, optimizará la gestión de información, por medio de la administración de horarios de trabajo tanto de los médicos, enfermeras y auxiliares, el agendamiento del quirófano, la solicitud de kits quirúrgicos; permitiendo que las cirugías se desarrollen conforme la planificación del sistema. El análisis de esta información, así como los informes de las intervenciones quirúrgicas permitirá dar un soporte en cuanto a la toma de decisiones del área quirúrgica.

Para permitir que la gestión de la información del área quirúrgica se desarrolle de una mejor manera, el sistema web se desarrollará, bajo el lenguaje de consultas GraphQL, permitiendo mejorar el backend, así como también que este sea flexible y escalable en futuras versiones.

Al desarrollar el sistema web de gestión de quirófanos bajo esta tecnología permitirá que la información sea oportuna, rápida y en tiempo real, además apoya a la toma de decisiones gracias a los reportes que genera.

3. OBJETIVOS:

3.1.Objetivo General:

- Desarrollar el sistema de gestión de quirófanos para el Hospital Universitario Andino utilizando GraphQL.

3.2.Objetivos Específicos:

- Analizar el lenguaje de consultas GraphQL.
- Desarrollar el sistema de gestión de quirófanos para el Hospital Universitario Andino.
- Evaluar el rendimiento del sistema de gestión de quirófanos utilizando la norma SQuaRE-ISO/IEC 25000.

4. CAPÍTULO II. MARCO TEÓRICO

4.1 Sistemas web en el sector de la salud

Los sistemas web que se involucran en el área de salud, están relacionados con el acceso y la gestión de información, son diversos y de gran interés. Las herramientas propuestas para conseguir que la información en internet este organizada de una forma más comprensible y asequible no son únicas y seguramente se desarrollaran paralelamente, y se crearan perfiles e intereses “a la carta” para el usuario, incluso pueden aparecer nuevas soluciones que ahora son impensables. (Miguel Angel Mayera, 2009)

Sin embargo, todas ellas tienen en común que persiguen un mismo objetivo: hacer posible un acceso sencillo y rápido a información relevante en la red, economizar el tiempo de búsqueda y evitar el abundante ruido informativo, que constituye, en la actualidad, uno de los problemas más preocupantes del internet y que en temas de salud puede tener una especial relevancia, tanto por el interés creciente que despierta este tipo de información por su extraordinaria abundancia como por las potenciales repercusiones, tanto positivas como negativas del acceso a información de calidad muy variable y a los múltiples servicios de salud actuales y futuros que nos puede ofrecer este medio. (Miguel Angel Mayera, 2009)

4.2 Servicios web

La World Wide Web Consortium le define a un servicio web como “un software diseñado para soportar la interacción, interoperabilidad entre maquinas sobre una red”. También se define como un conjunto de tecnologías estándares de software para el intercambio de datos entre aplicaciones tales como SOAP, WDSL y UDDI. El éxito de la interoperabilidad se consigue gracias a la adopción de protocolos y estándares abiertos definidos por la Organización for the Advancement of Structured Information Standards y el World Wide Web Consortium. (Machuca)

Los servicios web son tecnologías emergentes que permiten la intercomunicación dinámica de procesos a nivel de aplicaciones utilizando la web como un canal de comunicación. Un estudio realizado en el 2002 menciona que utilizando HTTP como protocolo de comunicación se facilita el uso de la infraestructura existente y minimiza la inseguridad en la red. (JRG Pulido, 2009)

4.3 Calidad de software

Es una compleja combinación de factores, que varían entre diferentes aplicaciones. Autores como Pressman, McCall y estándares como la norma ISO/IEC 25010 permiten categorizar los factores que afectan a la calidad de un software. (Estayno, Dapozo, & Cuenca Pletch Liliana, 2009)

Pressman se refiere a la calidad como “la concordancia con los requisitos funcionales y de rendimientos explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se espera de todo software desarrollado profesionalmente”. (Estayno, Dapozo, & Cuenca Pletch Liliana, 2009)

Por otro lado la IEEE sostiene que la calidad de software se refiere al grado con el que un sistema, modulo o proceso cumple con la especificación de requisitos y las expectativas del cliente o usuario final. (Estayno, Dapozo, & Cuenca Pletch Liliana, 2009)

4.4 Norma SQuaRE ISO/IEC 25000

Esta norma proporciona el uso de las nuevas series de estándares internacionales conocidas con el nombre de “requisitos y evaluación de calidad de productos software” (SQuaRE). (ISO 25000, s.f.)

La norma establece métricas, su evaluación, así como también los criterios para la especificación de requisitos de calidad de productos software, e incorpora un modelo de calidad que unifica los conceptos de calidad por parte del usuario con los parámetros en el proceso de desarrollo. (ISO 25000, s.f.)

4.4.1. Norma ISO/IEC 25010

La calidad de software a tratado en varios trabajos en los que se ha aplicado tanto el estándar ISO/IEC9126 o su evolución la ISO/IEC25000. La ISO/IEC 25010 que es el modelo de calidad en la que se aplica esta investigación, es un conjunto de estándares de calidad para evaluar un producto de software por medio de la evaluación de varias características: adecuación funcional, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad y eficiencia de desempeño (ISO, 2020). Siendo esta última la característica central de la investigación tomando el comportamiento temporal en la ejecución de aplicaciones. (Fausto A. Salazar Fierro1, 2019)

4.4.2. La eficiencia como parámetro de calidad

La norma ISO/IEC 25010 define a la eficiencia como el comportamiento del sistema en cuanto a su funcionalidad, capacidad, y la utilización de recursos. Esta característica está relacionada con el rendimiento de un dispositivo en función de su comportamiento temporal para ejecutar una tarea determinada. (Fausto A. Salazar Fierro, 2019) (Paola Roa Molina, 2015)

Rendimiento	
Comportamiento en el tiempo	Grado en que los tiempos de respuesta, procesamiento y las tasas de rendimiento de un producto o sistema al realizar sus funciones, cumplen con los requisitos.
Utilización de recursos	Grado en el que las cantidades y tipos de recursos utilizados por un producto o sistema al realizar sus funciones cumple con los requisitos.
Capacidad	Grado en que cumplen los requisitos respecto a los límites máximos de los parámetros de un producto o sistema. Ejemplo: Cantidad de usuarios concurrentes.

Tabla 1: Rendimiento

Fuente: Elaboración propia

4.5 Base de Datos No SQL

En 1998 Carlo Strozzi por primera vez hizo uso del término No SQL para referirse a una base de datos open-source relacional, que prescindía del lenguaje SQL. Claramente esta no es la definición que se utiliza actualmente. (Migani, 2018)

Para Eric Evans en 2009 el término No SQL lo hace referencia a bases de datos no relacionales, distribuidas linealmente escalables, de código abierto y que no garantizaban las tradicionales propiedades ACID, sin embargo en la actualidad ya ofrecen esta función. (Migani, 2018)

4.5.1 MongoDB

Es un sistema de base de datos NoSQL orientado a documentos de código abierto. En lugar de almacenar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON con un esquema dinámico, permitiendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

4.6 GraphQL

Es un lenguaje de consultas API con un tiempo de ejecución del lado del servidor para ejecutar consultas mediante el uso de un sistema de tipos que defina para sus datos. GraphQL no está vinculado a ninguna base de datos o motor de almacenamiento específico, sino que está respaldado por su código y datos existentes, además GraphQL no soporta almacenamiento en cache por lo que esta funcionalidad debe ser controlado por el cliente. (GraphQL , n.d.)

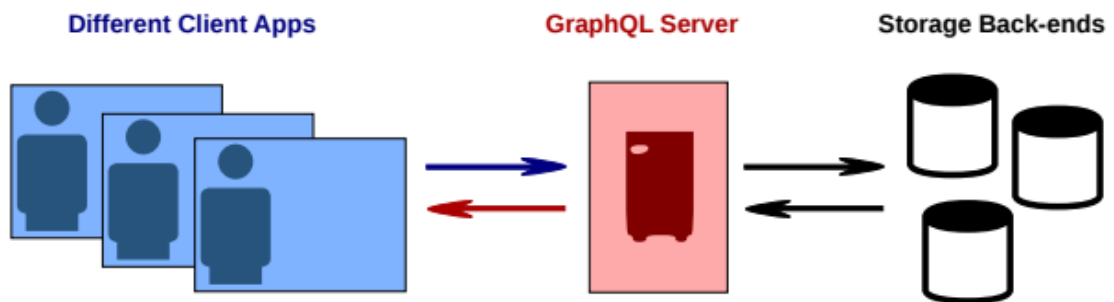


Figura 1. GraphQL server

Fuente: (Falco Nogatz)

4.6.1 Operaciones

GraphQL realiza cuatro operaciones fundamentales conocida como CRUD (crear, consultar, actualizar y eliminar) para manejar la información almacenada. GraphQL detalla tres tipos de operaciones independientes de una solicitud HTTP , estas operaciones son: consultas, mutaciones, y suscripciones. (Andrea Vazquez-Ingelmo, 2017)

- Consultas (queries): Obtención de datos.
- Mutaciones (mutations): Creación, actualización y eliminación de datos.
- Suscripciones (subscriptions): Solicitudes en respuesta a eventos fuente.

4.6.2 Schema

También llamado esquema permite describir los tipos de datos y sus relaciones que están disponibles en servidor GraphQL, en este punto también se describen las queries(consultas), mutations(mutaciones). (Erik Wittern, 2018)

```
type Procedimiento{
  id:ID
  nombres:String
  descripcion:String
  valorTotal:Float
  subprocedimiento:[SubprocedimientoArray]
}

type SubprocedimientoArray{
  id:String
  nombres:String
  valorTotal:Float
}

type Query{
  #Personal Hospitalario
  obtenerPersonalHospitalario:PersonalHospitalario
  obtenerTrabajadores:[PersonalHospitalario]
  totalTrabajadores:String
  obtenerPersonal(id:ID!):PersonalHospitalario
}
```

Figura 2. Definición de esquema GraphQL

Fuente: Elaboración propia

4.6.3 Querys

Son el elemento principal y más característico de este lenguaje de consulta de datos. Se trata de cadenas del tipo string que se usan para la consulta de datos y en GraphQL los tipos de query que se pueden lanzar al servidor deben estar predefinidos el schema del servidor. (Nogal, 2019)

```
type Query{
  #Personal Hospitalario
  obtenerPersonalHospitalario:PersonalHospitalario
  obtenerTrabajadores:[PersonalHospitalario]
  totalTrabajadores:String
  obtenerPersonal(id:ID!):PersonalHospitalario
  #Paciente
  obtenerPacientes:[Paciente]
  totalPacientes:String
  obtenerPaciente(id:ID!):Paciente
  #turnos
  obtenerTurnos:[Turno]
  obtenerTurno(id:ID!):Turno
  #horario
  obtenerHorarios:[Horario]
  obtenerHorariosByPersonal:[HorarioPersonal]
  #areaMedica
  obtenerAreasMedicas:[AreaMedica]
  obtenerAreaMedica(id:ID!):AreaMedica
  #subprocedimiento
  obtenerSubprocedimiento(id:ID!):Subprocedimiento
  obtenerSubprocedimientos:[Subprocedimiento]
  #procedimiento
  obtenerProcedimiento(id:ID!):Procedimiento
  obtenerProcedimientos:[Procedimiento]
}
```

Figura 3. Query en GraphQL

4.6.4 Mutations

Estas operaciones proporcionan una manera de modificar los datos, ya sea creando datos nuevos o actualizando los existentes, también devuelven un objeto creado o actualizado. Esto es útil para comprobar que el objeto se ha actualizado o creado correctamente. (Nogal, 2019)

```
type Mutation{
  #personal
  crearPersonalHospitalario(input:PersonalHospitalarioInput):PersonalHospitalario
  autenticarPersonal(usuario:String!,password:String!):Token
  eliminarPersonal(id:ID!):String
  actualizarPersonal(input:PersonalHospitalarioInput):PersonalHospitalario
  #paciente
  crearPaciente(input:PacienteInput):Paciente
  eliminarPaciente(id:ID!):String
  actualizarPaciente(input:PacienteInput):Paciente
  #turnos
  crearTurno(input:TurnoInput):Turno
  eliminarTurno(id:ID!):String
  actualizarTurno(input:TurnoInput):Turno
  #horarios
  crearHorario(input:HorarioInput):Horario
  actualizarHorario(input:HorarioInput):Horario
  eliminarHorario(id:ID!):String
  #areaMedica
  crearAreaMedica(input:AreaMedicaInput):AreaMedica
  actualizarAreaMedica(input:AreaMedicaInput):AreaMedica
  eliminarAreaMedica(id:ID!):String
  #subprocedimiento
  crearSubprocedimiento(input:SubprocedimientoInput):Subprocedimiento
  actualizarSubprocedimiento(input:SubprocedimientoInput):Subprocedimiento
  eliminarSubprocedimiento(id:ID!):String
}
```

Figura 4. Mtations en GraphQL

4.6.5 Subscriptions

Permiten obtener datos a través de una conexión activa con el servidor GraphQL a través de WebSocket. Esto permite que el servidor envíe actualizaciones en tiempo real al resultado de la suscripción a lo largo del tiempo. (Apollo, s.f.)

4.6.6 Tipos de objetos

Es el componente más básico del esquema GraphQL que determina el tipo de objeto que se va a recuperar del servicio. También determinan que campos ofrece el servicio GraphQL, cada tipo de objeto tiene un nombre y campos. Las mutaciones(mutation) y consultas(Query) son tipos de objetos especiales de GraphQL que actúan como punto de entrada de cada consulta. El esquema GraphQL admite los tipos escalares (int, string, float, boolean, ID). (Ghebremicael, 2017)

4.6.7 Argumentos

En GraphQL cada campo u objeto puede tener su propio conjunto de argumentos, estos argumentos permiten filtrar ciertos datos de una consulta. (Ghebremicael, 2017)

4.6.8 Tipos de enumeración

También llamados **enums**, los tipos de enumeración son un tipo especial de escalar que está restringido a un conjunto particular de valores permitidos. (GraphQL , s.f.)

4.6.9 Tipos de entrada

En el caso de mutaciones, donde se desee pasar un objeto completo para crearlo, los tipos de entrada tienen exactamente el mismo aspecto que los tipos de objetos normales, pero con la palabra clave input en lugar de type. (GraphQL , s.f.)

4.6.10 Variables

Las definiciones de variables en las consultas funcionan al igual que las definiciones de argumentos; Para declarar una variable se escribe el nombre de la variable precedidas por el signo de dólar (\$).

Todas la variables declaradas deben ser de tipo scalars, enums, input o tipo de objetos estas variables pueden ser opcionales u obligatorias en el caso de ser obligatoria se declara el signo de admiración (!) a lado de la variable. (GraphQL , s.f.)

4.6.11 Resolvers

En el momento en el cual se lanza una consulta a un servidor GraphQL, comienza la ejecución de esta. Para ello, se puede ver una consulta como un conjunto de tipos definidos en el schema, y el servidor se encarga de resolver cada tipo mediante unas funciones que se respaldan a todos estos tipos. Estas funciones son conocidas como resolvers, y se encuentran implementados por el servidor. Cuando un campo es ejecutado y resuelto, el correspondiente resolver es llamado para producir el siguiente valor. (Nogal, 2019)

En el caso de que un campo posea un valor escalar, como un entero o un string, entonces la ejecución de ese campo termina inmediatamente ya que no es necesario llamar a ningún resolvers. (Nogal, 2019)


```

export const resolvers = {
  Query: {
    //Querys para personal Hospitalario

    obtenerPersonalHospitalario: (root, args, { usuarioActual }) => {
      if (!usuarioActual) {
        return null;
      }
      console.log(usuarioActual);
      //obtener el usuario actual desde el request jwt index Verificado
      const usuario = PersonalHospitalario.findOne({ usuario: usuarioActual.usuario, estado: true });
      return usuario;
    },
  },
};

```

Figura 5. Resolvers en GraphQL

Fuente: Elaboración propia

4.7 GraphQL – Apollo

Apollo Client es una biblioteca completa de administración de estado para aplicaciones JavaScript. Simplemente se escribe una consulta GraphQL y Apollo Client se encargará de solicitar y almacenar en caché sus datos, así como de actualizar su UI(Interfaz de usuario). (Apollo, n.d.)

La obtención de datos con Apollo Client lo guía para estructurar su código de una manera predecible y declarativa consistente con las mejores prácticas modernas de ReactJS. Con Apollo se puede crear funciones de alta calidad más rápido sin la molestia de repetir consultas de datos. (Apollo, n.d.)

4.8 ReactJS

ReactJS es una biblioteca JavaScript para crear interfaces de usuarios, permite crear vistas simples para cada estado de la aplicación de forma sencilla. Esta librería está basado en componentes creando componentes encapsulados que manejan su propio estado, y convirtiéndoles en interfaces de usuario complejas. (React, 2020)

4.9 Apache JMeter

Es un software de código abierto desarrollado bajo el lenguaje Java que permite evaluar el comportamiento funcional de prueba y medir el rendimiento tanto en recursos estáticos como dinámicos, se puede utilizar para simular cargas de estrés en un servidor, grupo de servidores, red u objeto para probar su fuerza o analizar el rendimiento general bajo diferentes tipos de carga. (FOUNDATION, 2020)

4.10 Metodología Scrum

Es un proceso de desarrollo de software iterativo y creciente utilizado comúnmente en entornos basados en el desarrollo ágil de software. El trabajo es estructurado en ciclos de trabajo llamados Sprints, iteraciones de trabajo con una duración típica de dos a cuatro semanas. Durante cada sprint, los equipos eligen de una lista de requerimientos de cliente

priorizados, llamados historias de usuarios, para que las características que sean desarrolladas primero sean las de mayor valor para el cliente. Al final de cada sprint, se entrega un producto potencialmente lanzable/distribuable/comerciable. (Mariño, 2014)

Scrum se caracteriza por ser un modelo que define un conjunto de prácticas y roles como punto de inicio para definir el proceso de desarrollo que se llevara a cabo durante un proyecto. (Mariño, 2014)

Los roles principales en Scrum son el Scrum Master, el Product Owner, y el Equipo Scrum. (Mariño, 2014)

Las características más marcadas que se logran notar en Scrum son:

- Gestión regular de las expectativas del cliente.
- Resultados anticipados.
- Flexibilidad y adaptación.
- Retorno de inversión.
- Mitigación de riesgos.
- Productividad y calidad.
- Alineamiento entre cliente y equipo.
- Un equipo motivado

Definición de roles del proyecto

Definición de roles del proyecto	
SCRUM MASTER	Ing. Milton López.
PRODUCT OWNER	Dr. Wilson Nina.
EQUIPO	Byron Fernando Paguay.

Tabla 2. Definición de roles del proyecto

Fuente: Elaboración Propia

SCRUM MASTER: Se encargará de administrar el proceso del proyecto, su planificación, coordinación con el equipo y realizar un seguimiento e informes del progreso del proyecto, en términos de calidad, costo y plazos de entrega.

PRODUCT OWNER: Se encargará de crear la lista de funcionalidades del sistema, planificar el inicio de cada sprint y la revisión del producto al término de cada sprint para determinar si se cumplió con todas las funcionalidades.

EQUIPO: Las principales funciones son:

- Comprometerse al inicio de cada sprint desarrollar todas las funcionalidades en el tiempo determinado.
- Son responsables de entregar un producto a cada término del Sprint.

5. CAPITULO III METODOLOGÍA

5.1 Metodología

Para el desarrollo de la investigación se consideró desarrollar un sistema web de gestión de quirófanos para el Hospital Universitario Andino utilizando una API GraphQL, para el desarrollo del sistema web se aplicó la metodología scrum ya que se puede agregar cambios en cualquier momento permitiendo evaluar el retraso y actualizando las prioridades según se requiera. Para el desarrollo de los módulos se utilizó las siguientes herramientas: ReactJS, GraphQL, Apollo, NodeJS, MongoDB.

La investigación tiene un enfoque cuantitativo ya que busca medir únicamente magnitudes, mediante el uso de las métricas (tiempo de respuesta, tiempo de espera, rendimiento). Para la obtención de datos se utilizó la herramienta apache JMeter que permite simular peticiones de carga al servidor a través de un navegador Firefox, obteniendo resultados para el análisis del rendimiento del sistema web.

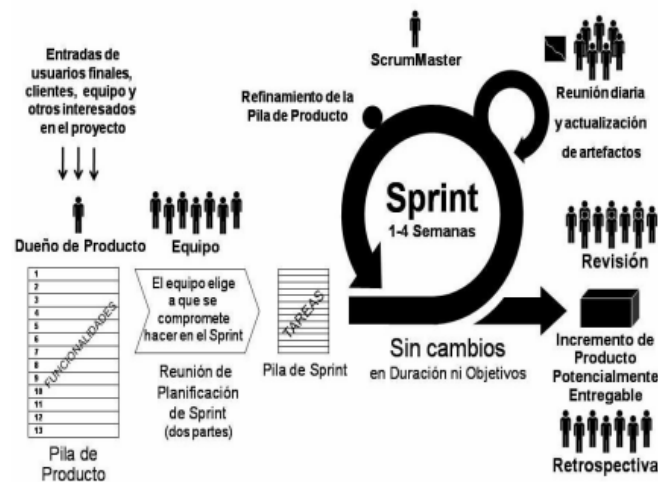


Figura 6: Metodología scrum

Fuente: (Pete Deemer)

5.2 Identificación de variables

Variable Independiente:

Sistema web de gestión de quirófanos para el Hospital Universitario Andino.

Variabes dependientes:

Eficiencia de desempeño del sistema web de gestión de quirófanos.

5.3 Tipo de Investigación

El tipo de estudio será explicativo, ya que requiere evaluar la eficiencia del sistema web, desarrollado bajo la tecnología GraphQL mediante un conjunto de variables planteadas para determinar el nivel de rendimiento, explicando las razones por la que esto ocurre.

5.4 Enfoque de la investigación

- **Cuantitativo:** El enfoque de la investigación será cuantitativo, porque la misma busca únicamente medir magnitudes que una vez obtenidas, permitan responder a la premisa planteada, que era conocer el rendimiento del sistema web desarrollado bajo la tecnología GraphQL.

5.5 Según la fuente de investigación

- **Investigación Documental:** Se realizará una técnica de investigación tipo documental porque es necesario la recopilación de antecedentes, mediante algunos documentos de diferentes autores, para ello la información será recolectada de diferentes fuentes como son: fuentes bibliográficas, fonográficas, artículos científicos, revistas y otros medios académicos y tecnológicos.

5.6 Unidad de análisis

Se realizará pruebas de carga y estrés del sistema web de gestión de quirófanos desarrollado a través de esta investigación, mediante la herramienta apache JMeter con dos escenarios de 30 y 60 peticiones al servidor para determinar si cumplen con los estándares de calidad de la norma SQuaRE ISO/IEC 25000 en base a la característica de eficiencia de desempeño analizando el comportamiento temporal.

5.7 Población de estudio y tamaño de la muestra

En la investigación se propone como objetivo final evaluar el rendimiento del sistema web, donde se manipula variables cuantitativas que permiten evaluar los indicadores de la eficiencia de desempeño haciendo uso de la subcaracterística comportamiento temporal definidos en la norma ISO/IEC 25010. Con esta premisa se simulará conexiones de usuarios virtuales concurrentes a través de un software de esfuerzo apache JMeter. Considerando como población a médicos, enfermeras y trabajadores que participan en los procesos de gestión de quirófanos del Hospital Universitario Andino; La población de

estudio se establece en 32 trabajadores que participan en los procesos de planificación y gestión de acuerdo a la información proporcionada por los directivos del hospital.

Para la simulación se establece el tamaño de la muestra determinada a través de la ecuación estadística para proporciones poblacionales, utilizando un margen de error del 5% con un nivel de confianza del 95%, este tamaño será el número de peticiones que simularan los usuarios virtuales concurrentes en un segundo realizadas hacia el sistema web.

$$n = \frac{z^2(p \cdot q)}{e^2 + \frac{z^2(p \cdot q)}{N}}$$

Figura 7: Fórmula de la ecuación estadística para proporciones poblacionales

Fuente: (Moreno, 2018)

Donde:

n= Tamaño de la muestra.

z= Nivel de confianza deseado (95% => 1.96).

p= Proporción de la población con la característica deseada (éxito=0.5).

q= Proporción de la población con la característica deseada (fracaso=0.5).

e= Nivel de error dispuesto a cometer (5% => 0.05).

N= Tamaño de la población (32).

Resultado:

$$n = \frac{1.96^2(0.5 * 0.5)}{0.05^2 + \frac{(1.96^2(0.5 * 0.5))}{32}}$$
$$n = 30$$

De acuerdo al resultado obtenido de la muestra el número de peticiones que se realizó al sistema web de gestión de quirófanos fue respectivamente 30 usuarios virtuales concurrentes (prueba de carga) y 60 usuarios virtuales concurrentes (prueba de estrés) divididos en dos escenarios:

Escenario 1: 30 usuarios virtuales concurrentes.

Escenario 2: 60 usuarios virtuales concurrentes.

5.8 Técnicas de recolección de datos

Técnica de observación: A través de la herramienta apache JMeter se obtuvieron datos para medir el desempeño del sistema web de gestión de quirófanos, mediante pruebas de carga y pruebas de estrés llamadas también pruebas de esfuerzo que tiene por objetivo superar los límites operativos del sistema, a partir de estos resultados se ejecutó la evaluación en base a las métricas de eficiencia de desempeño que define la norma ISO/IEC 25010.

5.9 Técnicas de análisis e interpretación de la información

Para evaluar la investigación se considera en base a métricas de la eficiencia de desempeño definidas por la norma ISO/IEC 25010, para lo cual se utilizó la herramienta apache JMeter.

Nombre de la métrica		
Tiempo de respuesta	Tiempo de espera	Rendimiento
Fórmula		
Ecuación: $X=B-A$ Donde A es el tiempo en que se envió la petición y B el tiempo en que se recibió la primera respuesta.	Ecuación: $X=B-A$ Donde la variable A es el tiempo en que se inició un trabajo y B el tiempo en el cual se completó un trabajo.	Ecuación: $X=A/T$ Donde A es el número de tareas cumplidas y T el intervalo de tiempo donde T es mayor que cero.

Tabla 3: Métricas y fórmulas

Fuente: Elaboración propia

5.10 Operacionalización de variables

Variables	Tipo	Objetivos	Indicadores
Sistema web de gestión de quirófanos para el Hospital Universitario Andino.	Independiente	Desarrollar el sistema web de gestión de quirófanos para el Hospital Universitario Andino, para la administración de quirófanos	Módulos Sistema web: <ul style="list-style-type: none"> • Módulo de gestión de horarios. • Módulo de agendamiento de quirófanos. • Módulo de registro de cirugías y partos. • Módulo de panel de gráficas.
Eficiencia de desempeño del sistema web de gestión de quirófanos.	Dependiente	Evaluar el rendimiento del sistema web de gestión de quirófanos utilizando la norma SQuaRE-ISO/IEC 25000.	Comportamiento temporal: <ul style="list-style-type: none"> • Tiempo de respuesta. • Tiempo de espera. • Rendimiento.

Tabla 4: Operacionalización de variables

Fuente: Elaboración propia

5.11 Especificación de las características, subcaracterísticas y métricas de la norma ISO/IEC 25010

De acuerdo con la norma ISO/IEC 25023 define medidas de la calidad que permiten evaluar cuantitativamente la calidad de productos de software en términos de características y subcaracterísticas descritas en la norma ISO/IEC 25010 (Ordoñez, 2018). Donde detallan: nombre, propósito, método de aplicación, fase del ciclo de calidad del producto, formula/ variables y umbrales de la métrica. Dentro de las características de la norma ISO/IEC 25010, la característica seleccionada fue la eficiencia de desempeño con la subcaracterística de comportamiento temporal haciendo uso de las siguientes métricas: tiempo de respuesta, tiempo de espera o procesamiento, y rendimiento.

De acuerdo al formato de especificación de las métricas detalladas en la norma ISO/IEC 25022 se presenta la tabla 5, donde se presenta el uso de las métricas utilizadas en la investigación.

Además, los valores deseados se obtuvieron de acuerdo al autor Jakob Nielsen, en el libro “Usability Engineering” donde menciona tres limites importantes en el tiempo de respuesta:

- 0.1 segundo: Provee una sensación de control directo, en donde el usuario percibe que está manipulando el sistema tal como lo haría con un objeto físico. (Nielsen, 1993)
- 1 segundo: El usuario percibe el retraso, pero aun siente control dentro del sistema o sitio. (Nielsen, 1993)
- 10 segundos: El usuario pierde su atención, no existe sensación de control, en este caso los mensajes ayudan a minimizar el abandono. (Nielsen, 1993)

En la investigación, la variable tiempo está relacionado con las siguientes métricas:

- Tiempo de respuesta y tiempo de espera: El valor deseado se espera que sea 1s donde el resultado más cercano a 0 es el mejor y en el peor de los casos el resultado sea mayor a 10s.
- Rendimiento: El valor deseado se espera que sea 10/1s donde el resultado más lejano a 10/1s será el mejor y en el peor de los casos el resultado sea 0/1s.

Subcaracterística	Nombre de la métrica	Fase de ciclo de vida de calidad del producto	Propósito	Método de aplicación	Formula	Valor deseado	Tipo de medida	Recursos Utilizados
Comportamiento temporal	Tiempo de respuesta	Interna/externa	¿Cuál es el tiempo estimado para completar una tarea?	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta.	$X=B-A$ A=tiempo de envío de petición B=tiempo en recibir la primera respuesta.	1 Peor caso: ≥ 10	X=tiempo-tiempo A=tiempo B=tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester.
	Tiempo de espera	Interna/externa	¿Cuál es el tiempo desde que se envía una instrucción, para que inicie un trabajo, hasta que lo completa?	Tomar el tiempo cuando se inicia un trabajo y tiempo en completar el trabajo.	$X=B-A$ A=tiempo cuando se inicia un trabajo. B=tiempo en completa un trabajo.	1 Peor caso: ≥ 10	X=tiempo-tiempo A=tiempo B=tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester.
	Rendimiento	Interna/externa	¿Cuántas tareas pueden ser procesadas por unidad de tiempo?	Contar el número de tareas completadas en un intervalo de tiempo.	$X=A/T$ A=número de tareas completadas T=Intervalo de tiempo Donde $T>0$	$\geq 10/t$ Peor caso: $=0/t$	X=Contable/tiempo A=contable B=tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester.

Tabla 5: Métricas para la eficiencia de desempeño

Fuente: ISO/IEC 25022

5.12 Análisis de niveles de puntuación para las métricas

Según la norma ISO/IEC 25040 define el proceso para llevar a cabo la evaluación del producto de software en los cuales se definen criterios de decisión para las métricas seleccionadas. Aquellos criterios son umbrales tipo numérico que permiten establecer la calidad del producto. (ISO 25000, s.f.) Por lo cual se establecieron rangos de medición, como se indica en la tabla 6, adaptados al criterio de las necesidades de solución del sistema. Estos valores se encuentran en el intervalo de 0 a 10, con cuatro niveles de puntuación y tres grados de satisfacción. (Reina, 2019)

Valor de medición	Niveles de puntuación	Grado de satisfacción
8-10	Cumple los requisitos.	Muy Satisfactorio.
5-7.99	Aceptable.	Satisfactorio.
1.91-5	Mínimamente aceptable.	Insatisfactorio.
0-1.9	Inaceptable.	

Tabla 6: Rango de puntuación para las métricas

Fuente: Elaboración Propia

De acuerdo con los valores cuantitativos se estableció una relación cualitativa como se observa en la tabla 6, obteniendo que los valores que no cumplan con la especificación de requisitos de software se establecerá en el rango inaceptable, y el rango objetivo los valores que cumplan las expectativas del cliente proporcionando una estabilidad funcional (Reina, 2019).

5.13 Ponderación de las métricas

En la tabla 7 se presenta la ponderación determinada, constituyendo de primer orden a la subcaracterística comportamiento temporal, definiendo al tiempo de respuesta con un 33 % siguiendo esta línea se estableció al tiempo de espera con un 33% y finalmente al rendimiento con un 34% permitiendo evaluar que tan eficiente es el sistema basado en el tiempo al concluir una tarea.

Característica	Subcaracterística	Métrica	Nivel de importancia	Ponderación
Eficiencia de desempeño	Comportamiento temporal	Tiempo de respuesta.	Fundamental	33%
		Tiempo de espera.	Fundamental	33%
		Rendimiento.	Fundamental	34%

Tabla 7: Ponderación de la métrica

Fuente: Elaboración Propia

6. CAPÍTULO IV: RESULTADOS Y DISCUSIÓN

6.1 Resultados

6.1.1 Análisis del lenguaje GraphQL

En la presente investigación se planteó como primer objetivo analizar el lenguaje de consultas GraphQL, a través de varias investigaciones documentales se obtuvo el siguiente cuadro comparativo:

API	Ventajas	Desventajas
GraphQL	<ul style="list-style-type: none"> • La definición de datos por parte del cliente permite obtener los datos requeridos, evitando tanto el underfetching y overfetching optimizando el rendimiento de la aplicación. • Es fuertemente tipado lo cual permite validar las consultas. • GraphQL al utilizar un solo endpoint para sus consultas, disminuye la carga útil de la red. 	<ul style="list-style-type: none"> • GraphQL presenta una curva de aprendizaje más larga para desarrolladores poco experimentados en servicios web. • Implementar un servidor GraphQL para proyectos que no van a tener gran volumen de datos, puede demandar más tiempo del necesario.

Tabla 8: Ventajas y desventajas de GraphQL

Fuente: Elaboración propia

GraphQL al ser un lenguaje de consultas API tiene la facilidad de unificar todas las consultas en un solo endpoint, por lo cual fue la tecnología seleccionada para desarrollar el sistema web de gestión de quirófanos, permitiendo obtener los datos solicitados sin que se genere una sobrecarga al servidor.

6.1.2 Desarrollo del sistema web de gestión de quirófanos para el Hospital Universitario Andino

Diagrama de proceso: Describe el conjunto de actividades que se deben realizar para cumplir con la gestión de quirófanos.

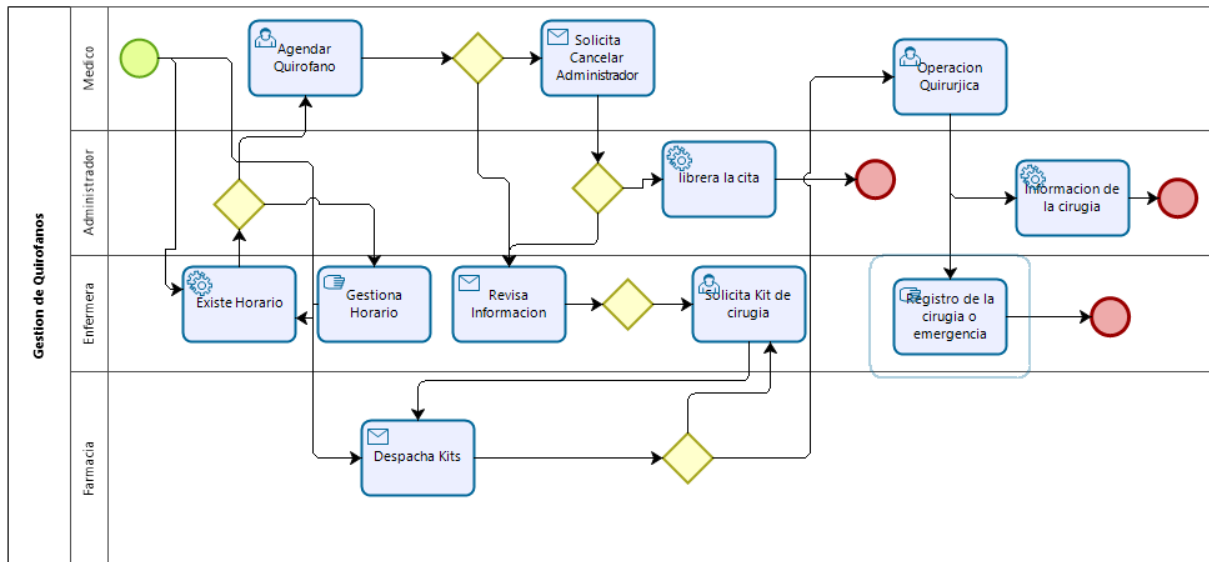


Figura 8: Diagrama de procesos.

Fuente: Elaboración Propia

Para desarrollar el sistema web de gestión de quirófanos se utilizó varias tecnologías como son:

- GraphQL: Permite realizar las consultas de datos.
- ReactJS: Utilizado para diseñar interfaces de usuarios complejas.
- Apollo Server: Permite crear un servidor GraphQL.
- Apollo Client: Permite administrar datos tanto locales como remotos con GraphQL.
- Visual Studio Code: Entorno de desarrollo para crear el sistema web.
- MongoDB: Permite crear base de datos no relacionales.
- Robot3T: Utilizado para administrar una base de datos no relacional.

Donde se obtuvieron como resultado 4 módulos que se describen a continuación:

Módulo de gestión de horarios: El presente módulo se encargará de gestionar las siguientes tareas.

- Gestionar los horarios de médicos, enfermeras, auxiliares y nutricionistas.
- Registrar la información de turnos rotativos.

- Informar a los usuarios su horario de trabajo.
- Eliminar horarios.

Figura 9: Módulo de gestión de horarios(Cuadrante)

Fuente: Elaboración propia

Módulo de agendamiento de quirófanos: El presente módulo se encargará de gestionar las siguientes tareas.

- Verificar los horarios de enfermeras y auxiliares.
- Registrar la agenda de quirófano.
- Filtrar una agenda por área médica o por fecha.
- Consultar agendas.
- Solicitar kits quirúrgicos.
- Importar datos a cirugías.

Figura 10: Módulo de agendamiento de quirófanos

Fuente: Elaboración propia

Módulo de registro de cirugías y partos: El presente módulo se encargará de gestionar las siguientes tareas.

- Registro de cirugías y partos.
- Actualización de cirugías y partos.
- Eliminación de cirugías y partos.

Fecha	Paciente	Edad	Procedencia	E.G.FUM	AGO	Obstetra	Pediatra	Acción
2020-09-24	Monica Erazo	24	Riobamba	fum	G	Tania Urquiza	Tania Urquiza	
2020-09-22	Monica Erazo	24	Riobamba	fum	G	Alberto Dario Diaz Parra	Jose Paul Parra Mayorga	
2020-06-29	María Angelica Barba Maggi	24	Loja	fum	G	Jose Paul Parra Mayorga	Alberto Dario Diaz Parra	
2020-08-28	Bianca Cecilia Badillo Conde	25	Loja	f	G	Janneth Mercedes Guallia	Cristian Roberto Sigcho Romero	
2020-08-28	Aida Mercedes Balladares Saltos	24	Loja	fum	P	Alberto Dario Diaz Parra	Jenny Alexandra Freire Rivera	
2020-06-28	Aida Mercedes Balladares Saltos	24	Loja	fum	P	Enrique Jesus Guambo Yerovi	Jose Paul Parra Mayorga	
2020-06-29	Bianca Cecilia Badillo Conde	25	Loja	fum	A	Enrique Jesus Guambo Yerovi	Enrique Jesus Guambo Yerovi	
2020-08-27	Bianca Cecilia Badillo Conde	25	Loja	fum	P	Jose Paul Parra Mayorga	Alberto Dario Diaz Parra	
2020-08-27	María Angelica Barba Maggi	24	Loja	fum	P	Alberto Dario Diaz Parra	Jose Paul Parra Mayorga	
2020-08-27	María Gabriela Romero Rodríguez	30	Ecuador	fum	P	Janneth Mercedes Guallia	Enrique Jesus Guambo Yerovi	

Figura 11: Módulo de gestión de registros (Partos)

Fuente: Elaboración propia

Fecha	Paciente	Operación Realizada	Medico	Observación	Acción
2020-09-23	Byron Fernando Paguay Duchí	laparoscopia quirurg. ver mas	Janneth Mercedes Guallia	ver mas	
2020-06-29	Sebastian Alejandro Pastor Romero	CODIGOS PARA SITUACI. ver mas	Bianca Narcisca Fuentes Lopez	ver mas	
2020-08-29	Cesar Eduardo Ponce Guerra	PERSONAS CON RIESGOS. ver mas	María Elizabeth Diaz Vallejo	ver mas	
2020-09-22	Cristian Eduardo Falconi Moreno	laparoscopia quirurg. ver mas	Janneth Mercedes Guallia	observacion. ver mas	
2020-09-20	María Gabriela Romero Rodríguez	EXPOSICION ACCIDENTA. ver mas	Janneth Mercedes Guallia	ver mas	
2020-09-03	Cesar Eduardo Ponce Guerra	EXAMEN PARA RECLUTAM. ver mas	Jenny Alexandra Freire Rivera	ver mas	
2020-09-03	Carlos Ivan Peñafiel Ortega	EXAMEN PARA RECLUTAM. ver mas	Bianca Narcisca Fuentes Lopez	ver mas	
2020-09-03	Carlos Ivan Peñafiel Ortega	EXAMEN PARA RECLUTAM. ver mas	Jenny Alexandra Freire Rivera	ver mas	
2020-09-02	Cesar Eduardo Ponce Guerra	EXAMEN PARA RECLUTAM. ver mas	Jenny Alexandra Freire Rivera	ver mas	
2020-06-29	Cesar Eduardo Ponce Guerra	FIBROMIALGIA. ver mas	Jenny Alexandra Freire Rivera	ver mas	

Figura 12: Módulo de gestión de registros (Cirugías)

Fuente: Elaboración propia

Módulo de panel de gráficas: El presente módulo se encargará de brindar información oportuna para la toma de decisiones gestionando varias consultas.

- Porcentaje de recién nacidos vivos vs muertos.
- Porcentaje de mujeres vs hombres recién nacidos vivos.
- Porcentaje de mujeres vs hombres recién nacidos muertos.

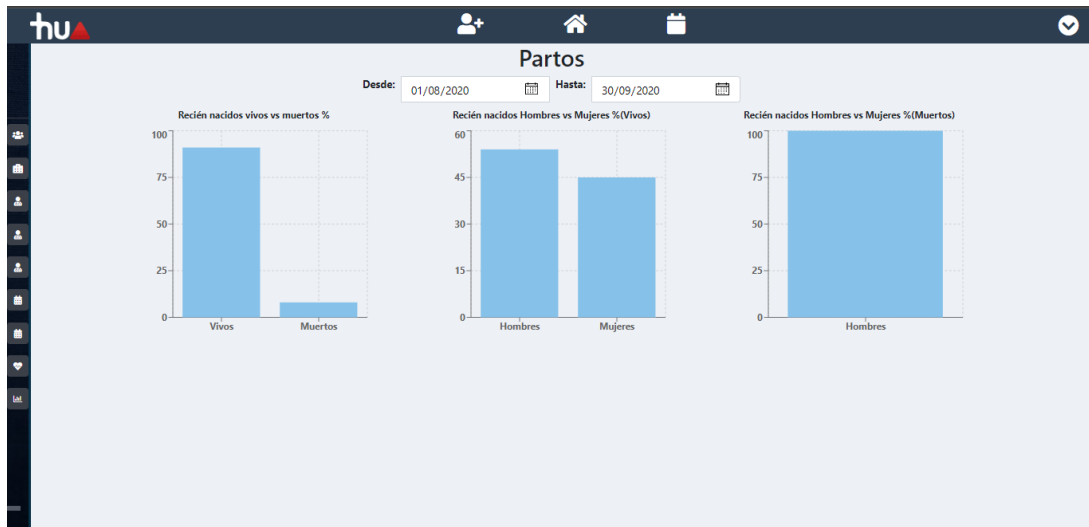


Figura 13: Módulo estadístico

Fuente: Elaboración propia

6.1.3 Evaluación del rendimiento en el sistema web de gestión de quirófanos

Para evaluar el rendimiento del sistema web de gestión de quirófanos se utilizó la norma SQuaRE-ISO/IEC 25000 que describe una división para gestión de la calidad, dentro de las normas que forman este apartado se encuentra la norma ISO/IEC 25010 que describe el modelo de calidad para el producto software, en la investigación se evaluó específicamente la eficiencia de desempeño, haciendo uso de la subcaracterística comportamiento temporal y de las métricas: tiempo de respuesta, tiempo de espera y rendimiento.

Para obtener los resultados de cada métrica se utilizó la herramienta apache JMeter, el cual permite simular diferentes tipos de carga al servidor, para posteriormente analizar el rendimiento general del sistema web bajo diferentes tipos de prueba, para ello se necesitó configurar el servidor proxy del navegador Firefox, configurar el número de usuarios concurrentes y finalmente analizar los resultados.

Para la obtención de los resultados de las métricas se consideró dos escenarios:

- El escenario 1 que cuenta con una simulación de carga de 30 usuarios.
- El escenario 2 que cuenta con una simulación de carga de 60 usuarios.

La consulta utilizada para esta simulación fue la siguiente:

```
export const OBTENER_PACIENTES = gql`
  query obtenerPacientes($limite: Int, $offset: Int, $filtro: String) {
    obtenerPacientes(limite: $limite, offset: $offset, filtro: $filtro) {
      id
      nombres
      apellidos
      identificacion
      estadoCivil
      sexo
      peso
      talla
      fechaNacimiento
      procedencia
      ocupacion
      emails {
        email
      }
      direcciones {
        direccion
      }
      telefonos {
        telefono
      }
      tipoSangre
      tipoIdentificacion
      nombreContacto
      telefonoContacto
      numeroHistoriaClinica
    }
    totalPacientes
  }
`;
```

Figura 14: Query obtener pacientes

Fuente: Elaboración propia

Tiempo de respuesta: La fórmula para medir el tiempo de respuesta de la consulta utilizada (Figura 14) para la obtención de resultados es la siguiente:

Formula:

$$X=B-A$$

Donde la variable **A** es el tiempo en el que se envió la petición y **B** el tiempo en que se recibió la primera respuesta.

En la tabla 9 se muestra los resultados de las mediciones obtenidos con respecto al tiempo de respuesta, en el cual se hizo uso del escenario 1.

#	Tiempo (ms)
1	1955
2	1923
3	1940
4	1899
5	1961
6	1932
7	1932
8	2032
9	1966
10	1867
11	2022
12	2259
13	2146
14	1883
15	1868
16	1876
17	2077
18	2147
19	2017
20	2074
21	2153
22	2152
23	2086
24	2166
25	2189
26	2154
27	2171
28	2077
29	2143
30	2223

	Tiempo (ms)
Promedio	2043
Error	0.00%
Desv.Estandar	117,17

Tabla 9: 30 usuarios virtuales concurrentes (Tiempo de respuesta)

Fuente: Elaboración propia

En la tabla 10 se muestra los resultados de las mediciones obtenidos con respecto al tiempo de respuesta, en el cual se hizo uso del escenario 2.

#	Tiempo (ms)	#	Tiempo (ms)
1	1953	31	2581
2	1982	32	3089
3	2001	33	2615
4	2004	34	2600
5	1997	35	2551
6	2046	36	2726
7	1778	37	2661
8	1894	38	2794
9	2056	39	2740
10	1841	40	2814
11	1992	41	3118
12	2031	42	2831
13	1766	43	2809
14	2134	44	2862
15	1866	45	2893
16	2011	46	3224
17	1808	47	2651
18	1772	48	2634
19	2002	49	2696
20	1912	50	2701
21	1909	51	3187
22	2160	52	3174
23	1962	53	3041
24	2055	54	3194
25	2281	55	3167
26	2609	56	3137
27	2805	57	3152
28	2627	58	3202
29	2806	59	3189
30	2767	60	3129

	Tiempo (ms)
Promedio	2499
Error	0.00%
Desv.Estandar	487,80

Tabla 10: 60 usuarios concurrentes virtuales (tiempo de respuesta)

Fuente: Elaboración propia

Tiempo de espera: La fórmula para calcular el tiempo de espera de la consulta utilizada (Figura 14) para la obtención de resultado es la siguiente:

Formula:

$$X=B-A$$

Donde la variable **A** corresponde al tiempo en que se inicia un trabajo y la variable **B** al tiempo en que se completa un trabajo.

En la tabla 11 se muestra los resultados de las mediciones obtenidos con respecto al tiempo de espera, en el cual se hizo uso del escenario 1.

#	Tiempo (ms)
1	1351
2	1319
3	1287
4	1157
5	1219
6	1189
7	991
8	1176
9	1208
10	1108
11	1241
12	1435
13	1305
14	1043
15	1127
16	1096
17	1260
18	1076
19	1150
20	1686
21	1752
22	1779
23	1757
24	1725
25	1812
26	1778
27	1903
28	1808
29	1873
30	1989

	Tiempo (ms)
Promedio	1420
Error	0.00%
Desv.Estandar	117,17

Tabla 11: 30 usuarios virtuales concurrentes (tiempo de espera)

Fuente: Elaboración propia

En la tabla 12 se muestra los resultados de las mediciones obtenidos con respecto al tiempo de espera, en el cual se hizo uso del escenario 2.

#	Tiempo (ms)	#	Tiempo (ms)
1	1429	31	1334
2	1664	32	1827
3	1680	33	1370
4	1653	34	1352
5	1660	35	1307
6	1701	36	1390
7	1422	37	1326
8	1534	38	1469
9	1687	39	1408
10	1471	40	1452
11	1616	41	1773
12	1653	42	1477
13	1429	43	1426
14	1699	44	1506
15	1503	45	1541
16	1626	46	1871
17	1400	47	1328
18	1383	48	1311
19	1591	49	1370
20	1497	50	1359
21	1489	51	1346
22	1689	52	1401
23	1522	53	1270
24	1536	54	1371
25	1824	55	1365
26	1553	56	1307
27	1789	57	1321
28	1611	58	1370
29	1725	59	1376
30	1500	60	1315

	Tiempo (ms)
Promedio	1503
Error	0.00%
Desv.Estandar	487,80

Tabla 12:60 usuarios virtuales concurrentes (tiempo de espera)

Fuente: Elaboración propia

Rendimiento o throughput: Para calcular el rendimiento de la consulta utilizada (Figura 14) para la obtención de resultado se utilizó la siguiente formula:

Fórmula:

A=Numero de tareas completadas.

T=Intervalo de tiempo donde: $T > 0$.

Ecuación:

$$X = A/T$$

En la tabla 13 se muestra los resultados de las mediciones obtenidos con respecto al rendimiento, en el cual se hizo uso del escenario 1.

#Muestras	Rendimiento	Error
30	9,5/sec	0.00%

Tabla 13: 30 usuarios concurrentes(rendimiento)

Fuente: Elaboración propia

En la tabla 14 se muestra los resultados de las mediciones obtenidos con respecto al rendimiento, en el cual se hizo uso del escenario 2.

#Muestras	Rendimiento	Error
60	15,0/sec	0.00%

Tabla 14:60 usuarios virtuales concurrentes(rendimiento)

Fuente: Elaboración Propia

6.1.3.1 Análisis del rendimiento del sistema web de gestión de quirófanos

Para el valor ponderado se multiplicó el valor obtenido por el porcentaje de ponderación que se muestra en la tabla 7 obteniendo los siguientes resultados que se muestran en las siguientes tablas respectivamente del escenario 1 y del escenario 2.

Evaluación de métricas 30 usuarios virtuales concurrentes:

Subcaracterística	Métrica	Peor caso	Valor deseado	Resultado Obtenido	Valor Métrica/10	Ponderación /10	Grado de satisfacción
Comportamiento temporal	Tiempo de respuesta	$\geq 10s$	1s	2.4s	7.6	2.5/3.3	Satisfactorio
	Tiempo de espera	$\geq 10s$	1s	1.4s	8.6	2.83/3.3	Muy Satisfactorio
	Rendimiento	0/s	$\geq 10/s$	9.5/s	9.5	3.23/3.4	Muy Satisfactorio
Total, valor ponderado						8.56/10	Muy Satisfactorio

Tabla 15: Resumen del resultado escenario1

Fuente: Elaboración Propia

Evaluación de métricas 60 usuarios virtuales concurrentes:

Subcaracterística	Métrica	Peor caso	Valor deseado	Resultado Obtenido	Valor Métrica/10	Ponderación /10	Grado de satisfacción
Comportamiento temporal	Tiempo de respuesta	>=10s	1s	2.5s	7.5	2.47/3.3	Satisfactorio
	Tiempo de espera	>=10s	1s	1.5s	8.5	2.80/3.3	Muy Satisfactorio
	Rendimiento	0/s	>=10/s	15/s	10	3.4/3.4	Muy Satisfactorio
Total, valor ponderado						8.67/10	Muy Satisfactorio

Tabla 16: Resumen del resultado escenario 2

Fuente: Elaboración Propia

En la tabla 17 se presenta un resumen de los resultados: En lo cual se procedió a sumar los valores resultantes tanto del escenario 1 como del escenario 2 y donde se evaluó cada métrica según la ponderación a la que corresponde obteniendo los resultados finales para cada escenario.

Subcaracterística	Valor ponderado	Calidad en uso	Faltante de calidad en uso	Deficiencia de calidad en uso del software
Comportamiento temporal(Escenario 1)	8.56	85.6	14.4	14.4
Comportamiento temporal(Escenario 2)	8.67	86.7	13.3	13.3
Comportamiento temporal promedio (Escenario 1 y Escenario 2)	8.61	86.1	13.9	13.9

Tabla 17: Resumen de resultados escenario 1 y escenario 2

Fuente: Elaboración Propia

6.1.3.1.1 Tiempo de respuesta

En la figura 15 se muestra una relación de los resultados obtenidos mostrando los siguientes tiempos de respuesta:

- Escenario 1: obtuvo un tiempo de respuesta de 2.4s (Tabla 9).
- Escenario 2: obtuvo un tiempo de respuesta de 2.5s (Tabla 10).

De acuerdo a los siguientes resultados, los tiempos de respuesta en promedio considerando diversas condiciones de carga como el escenario 1 y el escenario 2 es de 2.4s, tomando como referencia el modelo de evaluación de la norma ISO/IEC 25040 este resultado se encuentra en un nivel satisfactorio para atender las peticiones del usuario para cual el sistema fue diseñado. Sin embargo, al realizar pruebas adicionales en términos de usuarios el tiempo de respuesta que se obtuvo con una carga de 230 usuarios virtuales concurrentes el sistema se vuelve inaceptable, teniendo un valor promedio mayor a 8s, que de acuerdo al autor Jakob Nielsen este tiempo puede ser la causa de abandono del sistema.

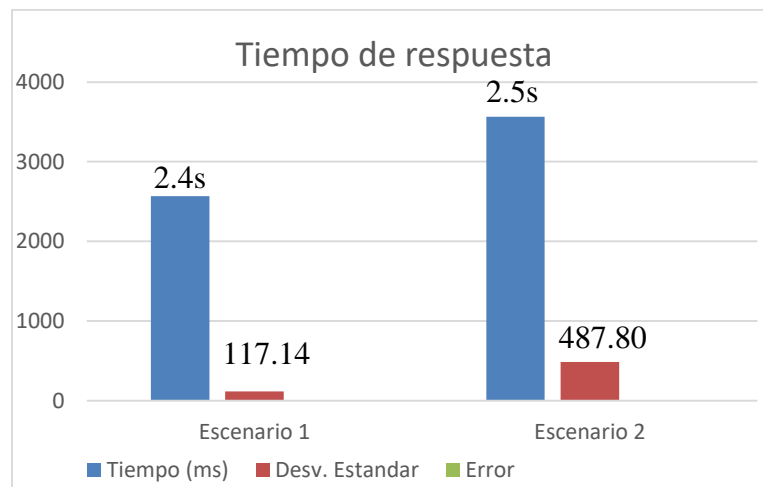


Figura 15: Tiempo de respuesta

Fuente: Elaboración propia

6.1.3.1.2 Tiempo de espera

En la figura 16 se muestra una relación de los resultados obtenidos mostrando los siguientes tiempos de espera:

- Escenario 1: obtuvo un tiempo de espera de 1.4s (Tabla 11).
- Escenario 2: obtuvo un tiempo de espera de 1.5s (Tabla 12).

Tomando como referencia la norma ISO/IEC 25040 la evaluación del sistema en promedio tiene una velocidad de procesamiento de 1.4s bajo el escenario 1 y el escenario 2 este resultado se encuentra en el rango objetivo especificado en la tabla 6. Estableciéndose como “cumple con los requisitos” es decir muy satisfactorio cumpliendo con los estándares de calidad que define la norma ISO/IEC 25040 donde nos permiten decidir la calidad del producto.

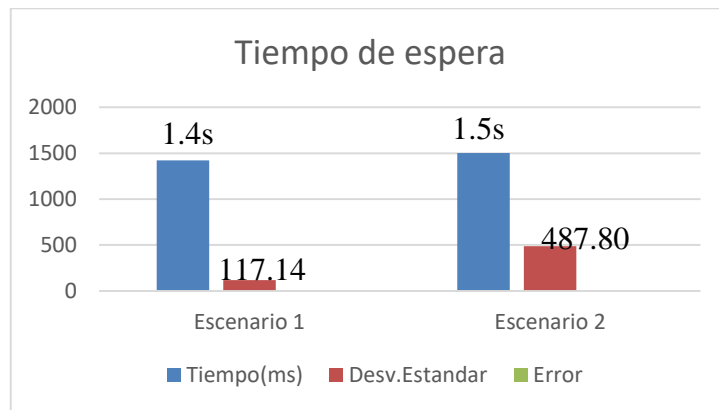


Figura 16: Tiempo de espera

Fuente: Elaboración propia

6.1.3.1.3 Rendimiento o throughput

En la figura 17 se muestra una relación de los resultados obtenidos mostrando el siguiente rendimiento:

- Escenario 1: obtuvo un rendimiento de 9.5/s (Tabla 13).
- Escenario 2: obtuvo un rendimiento de 15/s (Tabla 14).

Se puede definir que el rendimiento del sistema bajo el escenario 1 y el escenario 2 tomando como referencia los resultados obtenidos en la tabla 17, el sistema de gestión de quirófanos cuenta con una calidad en uso del 86.1% y aunque el sistema presente un déficit de calidad del 13.9 % el software cumple con los estándares establecido por la norma SQuaRE ISO/IEC 25000.

Aunque el sistema bajo el escenario 1 y el escenario 2 cumplen con los requisitos y estableciéndole como muy satisfactorio bajo el modelo de evaluación de la norma ISO/IEC 25040, pruebas adicionales revelan que a pesar de que el sistema no presento ningún porcentaje de error, la degradación del rendimiento se vuelve inaceptable en términos de usuarios, mayores a 230 usuarios virtuales concurrentes, donde la confiabilidad o precisión del sistema se puede ver afectada en este punto conforme crece la carga provoca que el usuario abandone el sistema.

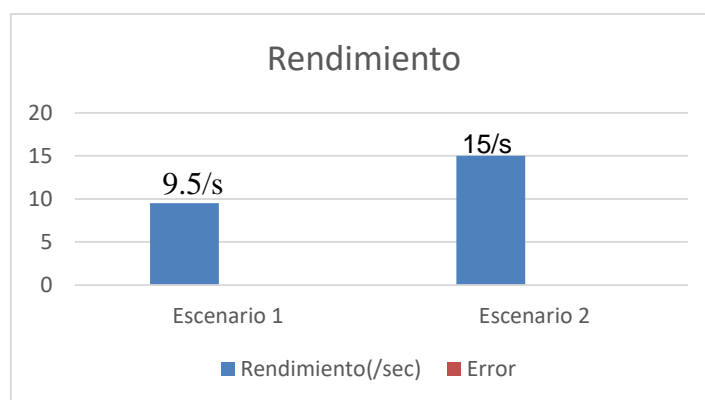


Figura 17: Rendimiento

Fuente: Elaboración propia

6.2 Discusión

En la presente investigación se utilizó el lenguaje GraphQL como medio para el consumo de información almacenada en una base de datos MongoDB, entre los resultados obtenidos, se revela que el lenguaje GraphQL proporcione varias mejoras respecto a:

- Escalabilidad: Ya que si es necesario añadir nuevos campos adicionales en el tiempo se puede hacer fácilmente sin perjudicar a los clientes existentes.
- Flexibilidad: Donde se permite crear consultas flexibles de acuerdo a los datos que se necesitan, el backend pone a su disposición sus recursos de datos y el cliente decide cuales consumirlos, además todos los cambios que se apliquen sobre la API se pueden ejecutar sin versionado.

Mediante la norma SQuaRE-ISO/IEC 25000 y su división para la gestión de calidad del producto software ISO/IEC 25010 a través de la cual se evaluó la eficiencia de desempeño, haciendo uso de la subcaracterística (“Comportamiento temporal”) y sus métricas: tiempo de espera, tiempo de respuesta, y rendimiento.

La herramienta apache JMeter fue de gran importancia para la obtención de resultados con respecto a los indicadores: tiempo de respuesta, tiempo de espera y rendimiento, para lo cual se simuló peticiones al servidor en dos escenarios.

- Escenario 1: 30 usuarios virtuales concurrentes.
- Escenario 2: 60 usuarios virtuales concurrentes.

El tiempo de espera, tiempo de respuesta y rendimiento se encuentra en un grado de satisfacción “muy satisfactorio” de acuerdo a la tabla 6, donde se evidencia que el servidor GraphQL se mantiene estable hasta los 60 usuarios virtuales concurrentes, no obstante, al llegar a los 230 usuarios virtuales concurrentes, el servidor presenta un tiempo de respuesta mayor a 8 segundos presentando un déficit de calidad en uso mayor a 13.9%.

Sin embargo, aunque el sistema es capaz de responder a los requisitos establecidos, pruebas adicionales evidenciaron que a partir de 230 usuarios virtuales concurrentes el sistema puede presentar un déficit de rendimiento, obteniendo un tiempo de espera mayor a 10 segundos donde este tiempo según el autor Jakob Nielsen promueve el abando de control al usuario, el usuario no siente sensación de control en el sistema web a pesar de que el sistema no presenta ninguna desconexión con el servidor.

Finalmente, Andrea Vázquez (Andrea Vazquez, 2017) menciona que otro tipo de API como lo es API REST, tiene un mejor tiempo de respuesta con relación a GraphQL, pero el rendimiento de la red, al utilizar API REST es mucho más alta ocasionando que el

consumo de ancho de banda sea mayor, debido a que API REST a diferencia de GraphQL necesita un endpoint por cada consulta, al utilizar una API GraphQL minimiza en gran cantidad de consumo en ancho de banda. La diferencia se debe a que al utilizar un solo endpoint para consumir los datos y especificar cuáles requerimos da como resultado esta ventaja.

7. Conclusiones

- El utilizar GraphQL como medio para el consumo de información permite crear peticiones dinámicas, en las que se pueden especificar los campos que se requieren, esta ventaja permite acabar con el underfetching (la consulta no obtiene todos los datos requeridos) y overfetching (la consulta devuelve más datos de los requeridos) mejorando notablemente el rendimiento del sistema o aplicación.
- GraphQL al no tener almacenamiento en cache, este debe ser controlado por el cliente, Apollo Client permite que los resultados de las consultas GraphQL se almacenen en una cache normalizada. Esto permite que el cliente responda a consultas futuras, de los mismos datos sin enviar solicitudes de red innecesarias.
- Al desarrollar el sistema web de gestión de quirófanos permite que el personal hospitalario agilice las tareas diarias que conllevan al agendamiento de un quirófano y con ello obtener información oportuna y en tiempo real para la toma de decisiones.
- Mediante la aplicación de una metodología benchmark se pudo obtener los resultados con respecto al tiempo de respuesta y tiempo de espera, GraphQL al llegar a 230 usuarios virtuales concurrentes presento tiempos de respuesta y espera mayor a 8 segundos, donde el usuario pierde la sensación de control en el sistema en este caso los mensajes de error ayudaran a minimizar el abandono del sistema.
- La aplicación de la norma SQuaRE-ISO/IEC 25000 en la evaluación del rendimiento del sistema web de gestión de quirófanos proporcionó los indicadores correctos como el tiempo de respuesta, tiempo de espera, y el rendimiento para determinar si la implementación del lenguaje de consultas GraphQL mejora el rendimiento al momento de consumir información, a partir de los resultados obtenidos, el sistema web de gestión de quirófanos presento una calidad en uso del 86.1% y un déficit de calidad del 13.9%.

8. Recomendaciones

- Para tener una ventaja sobre el área de underfetching y overfetching es recomendable realizar un levantamiento de requerimientos esto garantizará cuáles datos son los que se van a utilizar, así como también sus consultas adicionales que se harán al servidor GraphQL, en caso de no tener estos requerimientos claros se puede ocasionar una sobrecarga innecesaria al servidor.
- Apollo Client al proporcionar una ventaja sobre el almacenamiento de datos en una cache normalizada permite disminuir peticiones innecesarias al servidor, para mejorar esta ventaja se recomienda que tanto al crear, actualizar, eliminar un objeto este se actualice en la cache de manera automática actualizando la UI (interfaz de usuario) del cliente en tiempo real sin hacer consultas adicionales al servidor.
- Para desarrollar un sistema web o cualquier producto de software es necesario detectar necesidades reales a solucionar y realizar un análisis de las áreas funcionales de la empresa para determinar cuáles son los procesos que pueden optimizarse.
- Al desarrollar un sistema web o cualquier otro software es importante tener buenas prácticas, como la evaluación de calidad de producto de software y una adecuada interfaz de usuario, la norma SQuaRE-ISO/IEC 25000 contiene una serie de normas que permiten evaluar la calidad de software, de manera que garantice al cliente un sistema confiable.

9. Referencias

- Andrea Vazquez, J. C. (2017). Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL. *RIGHTSLINK*, 8.
- Andrea Vazquez-Ingelmo, J. C.-B.-P. (2017). Improving the OEEU's data-driven technological ecosystem's. *Association for Computing Machinery*, 8.
- Apollo. (s.f.). *Apollo*. Obtenido de <https://www.apollographql.com/docs/react/>
- Erik Wittem, A. C. (2018). Generating GraphQL-Wrappers for REST(-like) APIs. *The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI)*, 18.
- Estayno, M., Dapozo, G., & Cuenca Pletch Liliana, G. C. (2009). MODELOS Y MÉTRICAS PARA EVALUAR CALIDAD DE SOFTWARE. *In XI Workshop de Investigadores en Ciencias de la Computación*.
- Falco Nogatz, D. S. (s.f.). Implementing GraphQL as a Query Language for Deductive Databases in SWI-Prolog Using DCGs, Quasi Quotations, and Dicts.
- Fausto A. Salazar Fierro, C. A. (2019). Análisis de la eficiencia de desempeño en aplicaciones de Relidad Aumentada utilizando la normativa ISO/IEC/25010. *RISTI*.
- Faustox A. Salazar Fierro1, C. A. (2019). Análisis de la eficiencia de desempeño en aplicaciones de Realidad Aumentada utilizando la normativa ISO/IEC/25010. *risti*, 11.
- FOUNDATION, A. S. (29 de OCTUBRE de 2020). *APACHE JMETER*. Obtenido de <https://jmeter.apache.org/>
- Ghebremicael, E. S. (08 de 05 de 2017). Transformation of REST API to GraphQL for OpenTOSCA. *Transformation of REST API to GraphQL for OpenTOSCA*.
- GraphQL*. (s.f.). Obtenido de GraphQL Foundation: <https://graphql.org/>
- INTECO*. (20 de 03 de 2020). Obtenido de INTECO: <https://www.inteco.org/shop/product/ingenieria-de-sistemas-y-de-software-requisitos-y-evaluacion-de-la-calidad-de-sistemas-y-del-software-square-medicion-de-la-calidad-de-sistemas-y-productos-de-software-5949>
- ISO 25000*. (s.f.). Obtenido de <https://iso25000.com/index.php/normas-iso-25000>
- ISO, N. (2020). *Iso 25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

- JRG Pulido, J. C.-C.-G. (2009). TÓPICOS SELECTOS DE TECNOLOGIAS DE INFORMACION CON APLICACIONES PRACTICAS. *Universidad de Colima*.
- Machuca, C. A. (s.f.). Estado del Arte: Servicios Web.
- Mariño, S. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. *Scientia Et Technica*, 7.
- Migani, S. V. (2018). NoSQL: modelos de datos y sistemas de gestión de bases de datos. *In XX Workshop de Investigadores en Ciencias de la Computación* .
- Miguel Angel Mayera, A. L. (2009). Concepto y aplicaciones de la Web 3.0: una introducción.
- Moreno, M. R. (2018). An Investigación sobre recién egresados con empleode la Facultad de Ciencias Políticas y Sociales. *The International Journal of Social Sciences and Humanities Invention*, 4.
- Nielsen, J. (1993). *Usability Engineering*.
- Nogal, A. S. (2019). *Generacion de servidores GraphQL a partir de mappings RML utlizando Java y MongoDB*. Madrid.
- Ordoñez, P. A. (2018). *Evaluacion de la calidad sobre una aplicación movil*.
- Paola Roa Molina, C. M. (2015). Norma ISO/IEC 2500. *Udistrital*.
- Pete Deemer, G. B. (s.f.). INFORMACIÓN BÁSICA DE SCRUM . *Scrum training Institute*, 20.
- React. (2020). *React*. Obtenido de <https://es.reactjs.org/>
- Reina, E. (2019). Evaluación de la calidad en uso de un sistema web/móvil de control de asistencia a clases de docentes y estudiantes aplicando la norma ISO/IEC 25000 SQuaRe. *risti*, 12.

10. Anexos

Especificación de requisitos de software

Especificación de requisitos de software

Proyecto: Desarrollo de un sistema de gestión de quirófanos para el Hospital Universitario Andino aplicando GraphQL.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
31/07/2020		Byron Fernando Paguay Duchi	

Documento validado por las partes en fecha:

Por la comunidad	Por la universidad
Hospital Universitario Andino	Universidad Nacional de Chimborazo

1 Antecedentes

En la actualidad el **HOSPITAL UNIVERSITARIO ANDINO** no cuenta con un Sistema Informático que le permita cubrir los requerimientos Institucionales en el área de quirófanos a través de canales tecnológicos que necesitan ser automatizados.

2 Objetivo

Desarrollo de un sistema de gestión de quirófanos para el Hospital Universitario Andino aplicando GraphQL.

3 Descripción del esquema propuesto

Requerimientos generales para la gestión de quirófanos del HOSPITAL ANDINO DE CHIMBORAZO:

- Desarrollo bajo plataforma Visual Code (React, Apollo, GraphQL), MongoDB, otros.

4 Perspectiva del producto

El software interactuará con los siguientes servicios:

- Gestión de horarios de quirófanos.
- Gestión de horarios laborales de médicos y enfermeras
- Gestión de kits quirúrgicos.
- Informes de consultas de las operaciones más demandadas entre otros.

5 Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema web para la gestión de quirófanos del Hospital Universitario Andino basada la directriz del estándar IEEE.

5.1 Propósito

El presente documento tiene como propósito definir las especificaciones de requisitos funcionales y no funcionales para el desarrollo de un sistema web que permitirá gestionar el área quirúrgica del Hospital Universitario Andino. Éste será utilizado por el personal hospitalario.

5.2 Alcance

Esta especificación de requisitos está dirigida al usuario del sistema, para continuar con el desarrollo de aplicaciones informáticas, la cual tiene por objetivo gestionar los distintos procesos que se llevan a cabo para realizar una operación y/o cirugía (Gestión de horarios rotativos del personal hospitalario, agendamiento, reportes).

5.3 Personal involucrado

Nombre	Byron Fernando Paguay Duchi
Rol	Analista, diseñador y programador
Categoría Profesional	Estudiante de la carrera de Ingeniería en Sistemas y Computación.
Responsabilidad	Análisis de información, diseño y programación del SGQ.
Información de contacto	byronyjhova@hotmail.es

5.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos.
SGQ	Sistema de gestión de Quirófanos.
ERS	Especificación de Requisitos Software.
RF	Requerimiento Funcional.
RNF	Requerimiento No Funcional.

5.5 Referencias

Título del Documento	Referencia
Standard IEEE 830 – 1998	IEEE

5.6 Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema web de gestión de quirófanos, con el fin de conocer las principales funciones que éste debe cumplir.

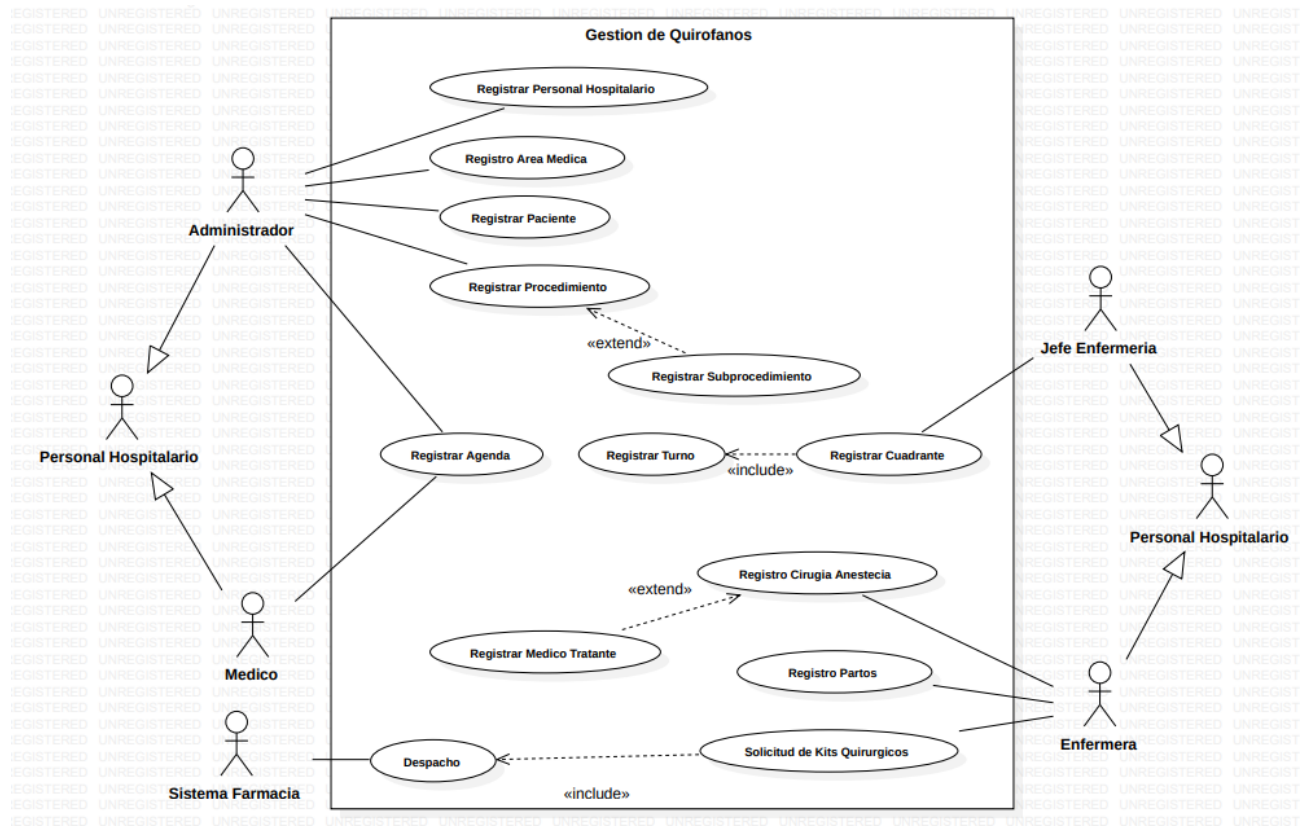
Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

6 Descripción general

6.1 Perspectiva del producto

El sistema web de gestión de quirófanos será un producto de software diseñado para trabajar en entornos WEB, su utilización permite que la gestión de información se lo realice de una forma más rápida y eficaz para mejorar los procesos que intervienen al realizar una operación y/o cirugía.

6.2 Funcionalidad del producto



6.3 Características de los usuarios

Tipo de usuario	Administrador.
Formación	NA.
Actividades	Control y manejo del sistema en general.

Tipo de usuario	Jefe de Enfermería.
Formación	Enfermera.
Actividades	Gestiona el horario rotativo del personal hospitalario.

Tipo de usuario	Enfermera.
Formación	Enfermera.
Actividades	Gestiona el registro de partos y cirugías. Gestiona la solicitud de kits quirúrgicos.

Tipo de usuario	Medico
Formación	Medico
Actividades	Agenda una operación y/o cirugía.

6.4 Restricciones

- Interfaz para ser usada con internet.
- Lenguajes y tecnologías en uso: ReactJS, MongoDB, NodeJS, GraphQL, JavaScript.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema deberá contar con un UI (interfaz de usuario) e implementación sencilla.

6.5 Suposiciones y dependencias

- Se asume que los requisitos descritos en el presente documento fueron revisados y aprobados por parte del cliente.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar el correcto funcionamiento del sistema.

7 Requisitos específicos

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autenticación de Usuario.
Características:	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.
Requerimiento NO funcional:	<ul style="list-style-type: none">• RNF01• RNF02• RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Registro, Actualización, Eliminación de los Datos del Personal Hospitalario.
Características:	El Administrador podrá registrar, actualizar, eliminar los datos del personal hospitalario, así como también dar el respectivo acceso al sistema.
Descripción del requerimiento:	El sistema permitirá al Administrador registrar, actualizar, eliminar los datos del personal hospitalario.
Requerimiento NO funcional:	<ul style="list-style-type: none">• RNF01• RNF02• RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Registro, Actualización, Eliminación de los Datos del Paciente.
Características:	El Administrador podrá registrar, actualizar, eliminar los datos del paciente.
Descripción del requerimiento:	El sistema permitirá al Administrador registrar, actualizar, eliminar los datos del paciente.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Registro, Actualización, Eliminación de los Datos del área médica.
Características:	El Administrador podrá registrar, actualizar, eliminar los datos del área médica.
Descripción del requerimiento:	El sistema permitirá al Administrador registrar, actualizar, eliminar los datos del área médica.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Registro, Actualización, Eliminación de los Datos del procedimiento.
Características:	El Administrador podrá registrar, actualizar, eliminar los datos del procedimiento.
Descripción del requerimiento:	El sistema permitirá al Administrador registrar, actualizar, eliminar los datos del procedimiento.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Registro, Actualización, Eliminación de los Datos del subprocedimiento.
Características:	El Administrador podrá registrar, actualizar, eliminar los datos del subprocedimiento.
Descripción del requerimiento:	El sistema permitirá al Administrador registrar, actualizar, eliminar los datos del subprocedimiento.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Registro, Eliminación de la agenda.
Características:	<ul style="list-style-type: none"> • El Administrador podrá registrar, eliminar la agenda. • El medico podrá registrar la agenda.
Descripción del requerimiento:	<ul style="list-style-type: none"> • El sistema permitirá al Administrador registrar, eliminar la agenda. • El sistema permitirá al Medico registrar la agenda.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Registro, Actualización, Eliminación del turno de trabajo.
Características:	El jefe de enfermería podrá registrar, actualizar, eliminar los datos del turno de trabajo.
Descripción del requerimiento:	El sistema permitirá al jefe de enfermería registrar, actualizar, eliminar el turno de trabajo.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Registro, Actualización, Eliminación del cuadrante laboral.
Características:	El jefe de enfermería podrá registrar, actualizar, eliminar el cuadrante laboral.
Descripción del requerimiento:	El sistema permitirá al jefe de enfermería registrar, actualizar, eliminar el cuadrante laboral.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Registro, Actualización, Eliminación de los Datos del médico tratante.
Características:	La Enfermera podrá registrar, actualizar, eliminar los datos del médico tratante.
Descripción del requerimiento:	El sistema permitirá a la enfermera registrar, actualizar, eliminar los datos del médico tratante.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Registro, Actualización, Eliminación del registro de cirugía anestesia.
Características:	La Enfermera podrá registrar, actualizar, eliminar el registro de cirugía anestesia.
Descripción del requerimiento:	El sistema permitirá a la enfermera registrar, actualizar, eliminar el registro de cirugía anestesia.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Registro, Actualización, Eliminación del registro de partos.
Características:	La Enfermera podrá registrar, actualizar, eliminar el registro de partos.
Descripción del requerimiento:	El sistema permitirá a la enfermera registrar, actualizar, eliminar el registro de partos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Registro de historial de acciones.
Características:	El sistema deberá registrar un historial de acciones.
Descripción del requerimiento:	El sistema registrara un historial de acciones.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Consultar Información uso de quirófano.
Características:	El sistema ofrecerá al usuario información general acerca del uso actual del quirófano.
Descripción del requerimiento:	Muestra a los usuarios información relevante acerca del uso actual del quirófano.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF15
Nombre del Requerimiento:	Gestión de horarios.
Características:	Permite gestionar información referente a los horarios del personal hospitalario.
Descripción del requerimiento:	Permite al jefe de enfermería crear un horario laboral y administrar el horario de trabajo de acuerdo a sus turnos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF16
Nombre del Requerimiento:	Gestión de la agenda.
Características:	Permite gestionar información referente a la agenda de un quirófano.
Descripción del requerimiento:	<p>El sistema permite gestionar el agendamiento de quirófanos de acuerdo al nivel de accesibilidad en cuanto:</p> <ul style="list-style-type: none"> • El medico podrá registrar un agendamiento del quirófano. • El administrador podrá registrar y también cancelar un agendamiento.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF17
Nombre del Requerimiento:	Gestionar Reportes.
Características:	El sistema permitirá generar reportes.
Descripción del requerimiento:	Permite al administrador y enfermera revisar reportes en cuanto al uso del quirófano, cirugías más demandadas, tipo de partos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento: Alta	

Requerimientos No Funcionales.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Ayuda en el uso del sistema.
Características:	La interfaz del usuario deberá de presentar un sistema de ayuda para que los mismos usuarios del sistema se les faciliten el trabajo en cuanto al manejo del sistema.
Descripción del requerimiento:	La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Seguridad en información
Características:	El sistema garantizara a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
Descripción del requerimiento:	Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
Prioridad del requerimiento: Alta	

7.1 Requisitos comunes de las interfaces

7.1.1 Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada desde un navegador de internet.

7.1.2 Interfaces de hardware

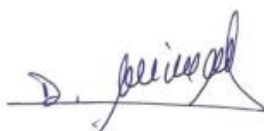
Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

- Adaptadores de red.
- Procesador de 1.66GHz o superior.
- Memoria mínima de 256Mb.
- Mouse.
- Teclado.

7.1.3 Interfaces de software

- Explorador: Chrome o Microsoft Edge.

• **REVISADO Y APROBADO POR:**



Dr. Wilson Nina Mayancela MA.

Gerente General del Hospital Universitario Andino



Est. Byron Paguay

Desarrollador



Ing. Milton Lopez

Analista de Tics HUA