

El presente informe es el resultado de la evaluación del desempeño del estudiante en el desarrollo de las competencias profesionales adquiridas durante su formación en la Carrera de TECNOLOGÍAS DE LA INFORMACIÓN MODALIDAD EN LÍNEA.

## DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

### CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN MODALIDAD EN LÍNEA

#### INFORME DE:



Pasantía



Ayudante de Cátedra



Práctica Pre Profesional No Remunerada



Ayudante de Investigación



Servicio a la comunidad

#### FUNDACIÓN CON CRISTO

NOMBRES Y APELLIDOS DEL ESTUDIANTE: BETTY LIZETH RODRIGUEZSALAS

NOMBRES Y APELLIDOS DEL TUTOR ACADÉMICO: JENNY ALEXANDRA RUIZ ROBALINO

#### CALIFICACIÓN DEL INFORME

**18**

  
**Ing. Jenny Alexandra Ruiz Robalino, PhD**

TUTORA ACADÉMICA

  
**Betty Lizeth Rodriguez Salas**

ESTUDIANTE

  
**Ing. José Alberto Sancho Arias**

TUTOR EMPRESARIAL



Sangolqui, 30/01/2025

## 1. PORTADA

## 2. INTRODUCCIÓN

El presente informe describe las actividades realizadas durante las prácticas preprofesionales no remuneradas en la Fundación Con Cristo, una institución sin fines de lucro dedicada a brindar atención terapéutica y médica a niños y niñas con discapacidades físicas y cognitivas. Estas prácticas se llevaron a cabo en el área de desarrollo informático, abordando específicamente la implementación de una nueva funcionalidad denominada **sección de auditorías** en su aplicación web.

La Fundación Con Cristo cuenta con más de 10 empleados y opera en el sector salud, enfocándose en la automatización de historias clínicas y la gestión de intervenciones terapéuticas. Actualmente, los procesos de registro y manejo de historias clínicas se realizan de forma manual, lo que limita la eficiencia y el seguimiento adecuado de las actividades del personal de salud. Se identificó como problemática principal la ausencia de una herramienta tecnológica que permita registrar y auditar las actividades realizadas dentro de la aplicación web. Por ello, se planteó el desarrollo de un módulo de auditoría que facilite la supervisión de acciones realizadas por los usuarios y promueva la mejora continua en los procesos de la fundación.

Las actividades realizadas estuvieron alineadas con el ciclo de vida del desarrollo de software, acorde al perfil profesional de Tecnologías de la Información. Estas incluyeron el levantamiento de información, revisión de procesos existentes, diseño de requisitos funcionales y no funcionales, diseño de interfaces de usuario, modelado de la base de datos, codificación de funcionalidades en el backend y frontend, realización de pruebas y entrega del prototipo del módulo de auditoría.

Las prácticas se desarrollaron desde el **18 de noviembre de 2024** hasta el **30 de enero de 2025**, cumpliendo con las fechas estipuladas en el convenio. Las tareas realizadas estuvieron justificadas por la necesidad de mejorar la eficiencia de los procesos de la Fundación Con Cristo, proporcionando una herramienta que permita un registro transparente de las acciones realizadas por los usuarios del sistema.

El principal objetivo de este trabajo fue reforzar el aprendizaje práctico del ciclo de vida del desarrollo de software y conceptualizar los conocimientos teóricos adquiridos durante la carrera. Como resultado, se alcanzó el desarrollo de una funcionalidad que registra y audita las actividades realizadas por los usuarios, tanto personal de salud como administradores, fortaleciendo la gestión y trazabilidad de los procesos en la fundación.

## 3. DESARROLLO

En el transcurso de la práctica pre profesional, se llevaron a cabo diversas actividades en el área de desarrollo informático de la organización “Fundación con Cristo”. Estas actividades fueron ejecutadas bajo la metodología tradicional en cascada, con un enfoque estructurado que incluyó fases claramente definidas de análisis, diseño, implementación, pruebas y entrega. El objetivo principal del proyecto fue diseñar e implementar un módulo de auditorías que permitiera a los usuarios visualizar, filtrar y exportar datos relevantes de forma eficiente.

Para el desarrollo de este módulo, se creó un sistema web basado en una arquitectura cliente-servidor, utilizando Node.js, React y PostgreSQL. Este sistema fue diseñado con el propósito de optimizar y modernizar los procesos internos de la fundación, facilitando la gestión de información y mejorando la eficiencia operativa. En este contexto, el módulo de auditorías se enfocó en el registro y seguimiento de eventos críticos dentro del sistema, permitiendo un mayor control sobre las actividades realizadas en la plataforma.

Las actividades realizadas incluyeron el análisis de requerimientos, donde se identificaron las necesidades específicas del módulo de auditorías; la estructuración y diseño de bases de datos, garantizando un almacenamiento seguro y eficiente de los registros de auditoría; el desarrollo de API REST, facilitando la interacción entre el cliente y el servidor para la recuperación y visualización de datos históricos; y la implementación de interfaces de usuario con React, brindando una experiencia intuitiva y accesible para los usuarios.

### 3.1. Actividades Realizadas

#### Fase de Requerimientos

La primera etapa de la práctica consistió en la inducción sobre el sistema existente, su documentación y los procesos internos de la institución. Durante esta fase, se llevó a cabo una revisión de la documentación técnica del sistema previo y se analizó la arquitectura del aplicativo web, con el propósito de comprender su funcionamiento y establecer una base para el desarrollo del módulo de auditorías.

Para el levantamiento de requerimientos, se utilizaron la observación directa y el análisis de casos de uso como técnicas principales. Se realizaron los procesos de la Ingeniería de Requisitos, incluyendo la identificación, recolección y análisis de información para la gestión de requisitos del sistema (SRS). La información recopilada fue documentada en el (Anexo 1; Rodríguez, B, 2024).

En esta etapa, se determinaron los aspectos clave del módulo de auditorías, tales como:

- **Datos a mostrar** en el historial de auditorías.
- **Filtros requeridos**, como usuario, fecha.
- **Formatos de exportación** necesarios para la generación de reportes.
- **Permisos de acceso** según los diferentes tipos de usuarios.

Finalmente, se llevó a cabo la especificación y validación de requisitos, asegurando que el módulo de auditorías cumpliera con las necesidades de la institución y se alineara con los estándares de seguridad y eficiencia del sistema.

#### Fase de Análisis y Diseño

Posteriormente, se definió un proceso de modelado de artefactos en la fase de análisis y diseño, proyectando la nueva arquitectura lógica del sistema. Esta arquitectura fue dividida en capas, asignando componentes específicos a cada una, con el objetivo de garantizar la escalabilidad y optimizar los procesos de transición en el desarrollo del

software. Se identificaron claramente los componentes del Back End y Front End, asegurando que el nuevo sistema cumpliera con los objetivos de la institución y facilitara futuras actualizaciones.

En la fase de diseño, se elaboraron diagramas de secuencia UML (Anexo 2) y diagramas de estados (Anexo 3) para modelar la arquitectura del sistema, complementados con un prototipo (Anexo 4) creado en Figma para validar la interfaz de usuario con el cliente. También se diseñó una base de datos (Anexo 5) específica para el módulo de auditorías, que incluyó una tabla con campos como usuario, acción, fecha y detalle.

Además, se realizó el flujo de navegación (Anexo 6) entre las secciones "Ver Auditorías" y "Exportar Auditorías", asegurando una interacción intuitiva dentro del sistema. Como parte del proceso, se efectuaron actualizaciones en el documento de especificación de requisitos (Anexo 7; Yanzapanta, B. S y Yépez, C. D, 2024) y en el documento de casos de uso de la aplicación web (Anexo 8; Yanzapanta, B. S y Yépez, C. D, 2024). Estas modificaciones se centraron en el módulo de Administración, donde se implementará la nueva sección de auditorías.

### Fase de Desarrollo Back-end

El desarrollo del módulo de auditoría en el backend se llevó a cabo utilizando Node.js y Express.js, asegurando una arquitectura escalable y eficiente. Se configuró el entorno en un servidor VPS con RockyLinux, integrando PostgreSQL como base de datos y utilizando Sequelize como ORM para gestionar las interacciones con la base de datos. En esta fase, se definieron los esquemas de datos y las tablas principales del sistema, estableciendo relaciones entre ellas para garantizar la integridad de la información.

Se implementó una API REST en Node.js con Express.js, proporcionando servicios CRUD específicos para la gestión de auditorías. Esta API permitió registrar automáticamente cada acción relevante dentro del sistema, asegurando que todas las operaciones fueran auditadas correctamente. Además, se incorporó un sistema de autenticación y gestión de roles, estableciendo permisos diferenciados que garantizaron el acceso controlado a los datos, restringiendo la capacidad de exportación únicamente a los administradores. (Anexo 9)

### Fase de Desarrollo Front-End

Para la construcción de la interfaz de usuario del módulo de auditoría, se utilizó React, estableciendo una comunicación eficiente con la API REST a través de métodos HTTP (GET, POST, PUT, DELETE). Se configuraron las librerías necesarias y se diseñaron pantallas específicas para la visualización del historial de auditorías, la aplicación de filtros de búsqueda por usuario y fecha, y la exportación de registros en formato PDF.

El diseño de la interfaz priorizó una experiencia de usuario intuitiva y accesible, asegurando que la navegación dentro del módulo fuera clara y funcional. Se implementaron componentes reutilizables que optimizaron la presentación de la información, facilitando la interacción del usuario con los datos auditados. Además, se realizaron pruebas de integración para verificar la correcta comunicación entre el frontend y el backend, asegurando el correcto funcionamiento del sistema. (Anexo 10)

## Fase de Pruebas

Para evaluar el correcto funcionamiento del módulo de auditoría, se llevó a cabo una fase de pruebas en las instalaciones de la **Fundación con Cristo**. En esta etapa, la directora encargada de la institución interactuó directamente con el sistema, probando sus funcionalidades en un entorno controlado. La validación se realizó de manera práctica a través de la interfaz del módulo de auditorías, asegurando que el sistema cumpliera con las expectativas y necesidades establecidas durante las fases previas del desarrollo.

Las pruebas incluyeron los siguientes aspectos clave:

- **Validación de la carga de datos:** Se comprobó que el historial de auditorías mostrara la información de manera precisa y en tiempo real.
- **Funcionalidad de los filtros:** Se verificó que los filtros aplicados (por usuario, fecha) permitieran segmentar la información de manera eficiente.
- **Exportación de datos:** Se realizaron pruebas de exportación en diferentes formatos, asegurando que los archivos generados conservaran la estructura y detalles requeridos.
- **Gestión de accesos y permisos:** Se validó que solo los usuarios con permisos adecuados es decir con rol de administrador pudieran acceder a ciertas funciones críticas, como la exportación de registros.

Este proceso de prueba se llevó a cabo de forma **voluntaria** por parte del personal de la institución, con una breve inducción previa sobre el uso del sistema. Finalmente, se recopilaron observaciones y comentarios respecto a la experiencia de uso, lo que permitió ajustar detalles y mejorar la usabilidad del módulo antes de su implementación final. (*Anexo 11*)

## Fase de Capacitación y Entrega

Para garantizar el correcto uso del módulo de auditorías, se llevó a cabo una capacitación dirigida a la directora de la institución, quien posee el rol de administradora, así como al personal de salud involucrado en el sistema. Durante la sesión, se explicó detalladamente el funcionamiento del módulo, destacando que cada acción realizada en el sistema, como la creación, modificación, eliminación o desactivación de eventos, sería automáticamente registrada y visualizada en la sección de auditorías.

Además, se instruyó a los usuarios sobre cómo navegar y utilizar las funciones clave del módulo, incluyendo el uso de la tabla de auditorías, que permite visualizar registros organizados por fecha, usuario, acción y detalles. Como complemento a la capacitación, se elaboró una guía de usuario detallada (*Anexo 12; Rodríguez, B., 2024*), proporcionando instrucciones paso a paso para facilitar la adopción del sistema.

Finalmente, el código funcional del módulo fue subido al repositorio FTP de la fundación, asegurando su disponibilidad y facilitando futuras actualizaciones o mantenimientos del sistema.

## Limitaciones y Éxitos

Entre las limitaciones encontradas durante el desarrollo, se destacaron la falta de documentación inicial sobre los procesos existentes y la necesidad de aprender nuevas herramientas como Docker y DBeaver. A pesar de estas dificultades, se logró:

- Diseñar e implementar un módulo funcional de auditorías que cumple con los requisitos del cliente.
- Integrar eficientemente el módulo de auditoria con las herramientas existentes de la organización.
- Mejorar la gestión de auditorías mediante funcionalidades avanzadas como filtros y exportación de datos.

### Recursos Utilizados

Para el desarrollo del proyecto, se emplearon diversas herramientas y tecnologías que facilitaron su implementación. Como entorno de desarrollo, se utilizó Visual Studio Code, complementado con Docker para la gestión de contenedores y Ubuntu como sistema operativo. La administración y visualización de la base de datos se realizaron mediante DBeaver. En cuanto a las tecnologías utilizadas, se desarrolló la aplicación con React para la interfaz de usuario y Node.js en el backend. Además, la documentación del proyecto incluyó diagramas UML para el diseño estructural, un prototipo en Figma para la representación visual de la interfaz y una guía de usuario que facilita la comprensión y uso del sistema.

### Cronograma de Actividades

El cronograma abarca desde la inducción y recopilación de información (18/11 - 22/11), incluyendo la capacitación en React, Docker y otras herramientas, hasta la definición de requisitos y diseño del sistema (25/11 - 2/12), donde se establecen los datos, filtros y permisos. Luego, se desarrolla la interfaz y la base de datos (3/12 - 12/12), seguido de la implementación del módulo de auditorías en diferentes niveles (13/12 - 10/01), integrando funciones clave como visualización, filtrado y exportación. Posteriormente, se conectan los servicios frontend y backend (13/01 - 24/01), culminando con pruebas, validación y entrega final (27/01 - 30/01), incluyendo la capacitación de usuarios y la subida del proyecto al FTP. Se recomienda establecer revisiones periódicas, distribuir mejor la capacitación y aprovechar los días sin tareas asignadas para pruebas y mejoras. (Anexo 13)

## 4. CONCLUSIONES

La implementación del módulo de auditoría en la aplicación web de la Fundación Con Cristo representó un avance significativo en la gestión de sus procesos internos, demostrando la importancia de analizar necesidades de información para optimizar la trazabilidad y supervisión de actividades. Este desarrollo permitió mejorar el registro y control de las acciones del personal de salud y administradores, asegurando mayor transparencia en el manejo de historias clínicas. Además, la incorporación de filtros y opciones de exportación de datos facilitó la toma de decisiones basada en información estructurada. El código fue documentado, ingresado al repositorio FTP y validado por la directora, garantizando su correcto funcionamiento.

Durante el desarrollo de las prácticas preprofesionales, se aplicaron conocimientos en el diseño y programación de software, logrando desarrollar aplicativos de mediana escala mediante el uso de tecnologías modernas como React, Node.js y PostgreSQL. La experiencia permitió reforzar competencias en metodologías ágiles, pruebas de software y administración de bases de datos. Además, se trabajó en la integración de

herramientas como Docker y DBeaver, fortaleciendo la capacidad de gestionar entornos de desarrollo eficientes y escalables. La aplicación de buenas prácticas en la estructuración del sistema contribuyó a una implementación funcional y alineada con los requerimientos de la fundación.

A lo largo del proyecto, se identificaron desafíos como la falta de documentación previa y la necesidad de adquirir nuevas habilidades en corto tiempo. Sin embargo, estos retos fueron abordados eficazmente, logrando desarrollar una aplicación web funcional que responde a las necesidades institucionales. Como mejora futura, se recomienda continuar optimizando el módulo de auditoría, explorando la automatización de reportes y su integración con otras funcionalidades del sistema para fortalecer la eficiencia operativa de la fundación.

## 5. RECOMENDACIONES

**Para la Fundación con Cristo:**

➤ **Capacitación Continua del Personal**

Para garantizar un uso eficiente del módulo de auditoría, se recomienda diseñar e implementar un programa de capacitación especializado dirigido al personal encargado de su gestión. Este programa debe incluir el uso avanzado de las herramientas de auditoría, la interpretación y análisis de registros generados, especialmente aquellos relacionados con historias clínicas y terapias, así como la identificación de anomalías o comportamientos sospechosos. Además, es fundamental realizar sesiones de actualización periódicas para mantener al equipo al día con las mejores prácticas y nuevas funcionalidades.

➤ **Monitoreo Proactivo y Ajustes Periódicos**

Se sugiere establecer un sistema de monitoreo continuo del rendimiento de la sección de auditoría mediante métricas clave que permitan evaluar su eficacia. Asimismo, es recomendable realizar revisiones trimestrales para identificar áreas de mejora y ajustar los procesos según las necesidades cambiantes de la fundación. Implementar un enfoque de mejora continua, basado en el análisis de datos y el feedback de los usuarios, contribuirá a optimizar el funcionamiento del módulo y su impacto en la gestión institucional.

➤ **Implementación de Rastreo de IP para Verificación de Información**

Para mejorar la seguridad y la trazabilidad de los datos, se recomienda integrar un sistema de rastreo de IP en la aplicación web, lo que permitirá verificar la autenticidad y procedencia de la información ingresada. Este sistema debe cumplir con las normativas de privacidad y protección de datos vigentes, asegurando que su implementación no comprometa la confidencialidad de la información. Adicionalmente, se debe capacitar al personal en la interpretación de los datos de rastreo para detectar posibles irregularidades y tomar medidas preventivas.

➤ **Optimización del Almacenamiento de Datos de Auditoría**

Con el fin de mejorar el rendimiento y la escalabilidad del sistema, se recomienda separar la base de datos de auditoría en un contenedor independiente. Además, es esencial implementar un sistema de gestión de bases de datos que permita un crecimiento controlado y eficiente, considerando el incremento exponencial de registros.

debido a las acciones del personal de salud. También se deben establecer políticas de retención y archivo de datos para evitar la saturación del sistema y garantizar un acceso rápido a información relevante.

#### ➤ Reforzar la Seguridad y Privacidad de los Datos

Dado que la aplicación web maneja información sensible, es crucial reforzar la seguridad y privacidad de los datos almacenados en el sistema. Se recomienda la implementación de protocolos de cifrado, autenticación multifactor y control de acceso basado en roles para minimizar riesgos. Asimismo, es importante realizar auditorías de seguridad periódicas para identificar vulnerabilidades y aplicar medidas correctivas que garanticen la integridad y confidencialidad de la información.

#### Para la universidad:

- Promover proyectos que integren prácticas de auditoría y seguridad informática en aplicaciones web, dado su impacto positivo en la calidad de los sistemas.
- Fomentar la colaboración entre estudiantes y empresas para desarrollar soluciones innovadoras en el ámbito de la auditoría y el control de sistemas.

## 6. ANEXOS

### Anexo 1.Documento Levantamiento de la información

El formato titulado FOR-SIS-02-Levantamiento de Información.V1.0 (2024) es un documento utilizado para recopilar datos relevantes durante la etapa inicial del diseño del sistema. Este incluye información clave sobre los requerimientos funcionales y no funcionales identificados en reuniones con la Fundación con Cristo.

Rodriguez B. (2024). FOR-SIS-02-Levantamiento de Información.V1.0 [PDF]. Recuperado de

[https://drive.google.com/file/d/1DFn2IuccS7og5fYzY07EYoR8StAW\\_Y-s/view?usp=sharing](https://drive.google.com/file/d/1DFn2IuccS7og5fYzY07EYoR8StAW_Y-s/view?usp=sharing)

### Anexo 2.Diagrama de secuencia UML

Diagramas de secuencia UML que detallan los intercambios entre los actores y los sistemas para la ejecución de las funcionalidades. Los diagramas describen la secuencia de eventos y las interacciones entre los actores (Fundación con Cristo, Sistema de Gestión de Salud, Personal de Salud) y los sistemas (Sistema de Información, Plataforma Web).

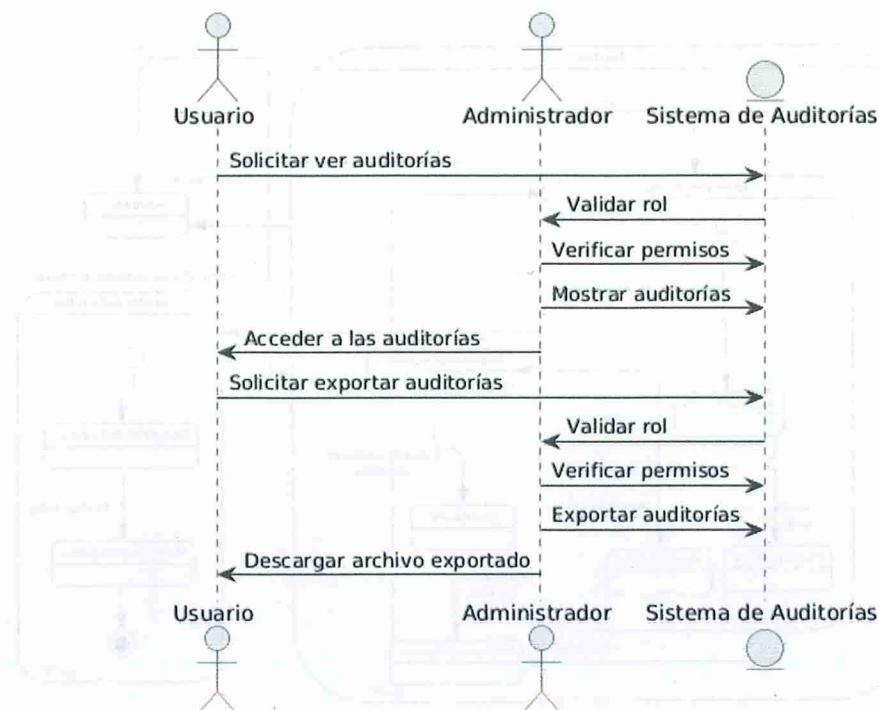


Figura 1. Diagrama Secuencia Auditorias describe la interacción entre un usuario, un administrador y el sistema de auditorías. El usuario puede solicitar ver o exportar auditorías, pero el sistema verifica primero si tiene los permisos adecuados, validando el rol con el administrador. Si el usuario es administrador, el sistema permite el acceso a las auditorías y facilita la exportación en un archivo descargable. Este flujo asegura que solo los administradores puedan realizar dichas acciones.

### Anexo 3. Diagrama de estado

Presenta los diagramas que describen el ciclo de vida de los objetos del sistema, detallando los estados y transiciones relacionadas con el módulo de auditorías.



Este diagrama muestra el ciclo de vida de los objetos del sistema, detallando los estados y transiciones relacionadas con el módulo de auditorías.

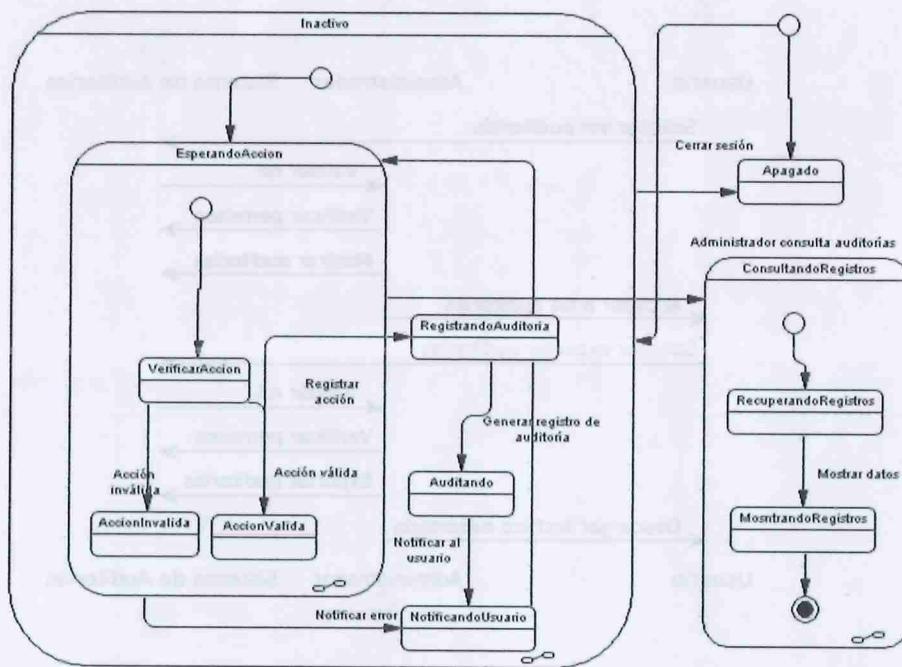


Figura 2. Diagrama de estado-Auditoría El diagrama de estado para auditorías describe el ciclo de vida de las acciones en el sistema. Comienza en el estado Inactivo, con el sistema esperando una acción del usuario en el estado EsperaAccion. Si la acción es válida, el sistema sigue el flujo de estados: RegistrandoAuditoria → Auditando → NotificandoUsuario. Si la acción es inválida, se salta directamente a NotificandoUsuario para informar del error. Cuando un administrador consulta registros, el sistema entra en el estado ConsultandoRegistros, donde recupera y muestra los datos. Finalmente, el sistema puede finalizar su proceso y pasar al estado Apagado cuando se cierra sesión.

#### Anexo 4. Prototipo de Interfaz de Auditoría

Prototipo desarrollado en Figma para validar la interfaz con el cliente. Este incluye las pantallas principales del módulo de auditorías, tales como "Ver Auditorías" y "Exportar Auditorías".

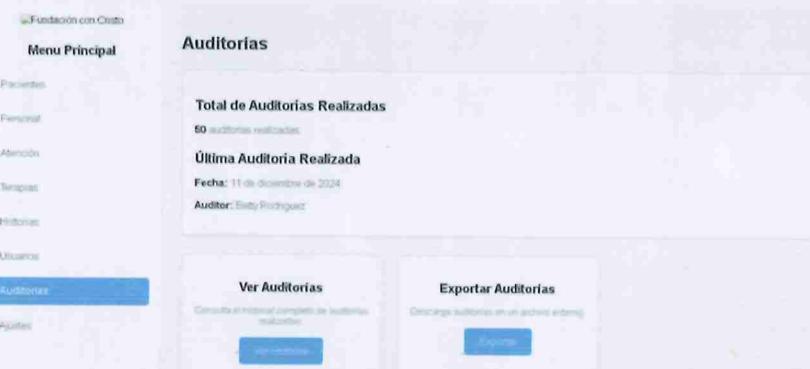


Figura 3. El prototipo de Auditorías muestra una sección que contabiliza todas las auditorías realizadas, junto con el último registro ingresado en la base de datos. En la parte inferior, se encuentran las secciones que permiten acceder a una nueva pantalla

con un listado de registros, donde es posible realizar búsquedas por usuario o fecha. Además, ofrece la opción de exportar los resultados en formato PDF.

Rodriguez B. (2024). *Mockup-fcc-Auditoria*[Figma]. Recuperado de <https://www.figma.com/design/V0oreASGzt4z7g2j1x6LKe/Mockup-fcc-Auditoria?node-id=0-1&m=dev&t=6f19N17UDxq9hPsp-1>

#### Anexo 5.Diseño de Base de Datos

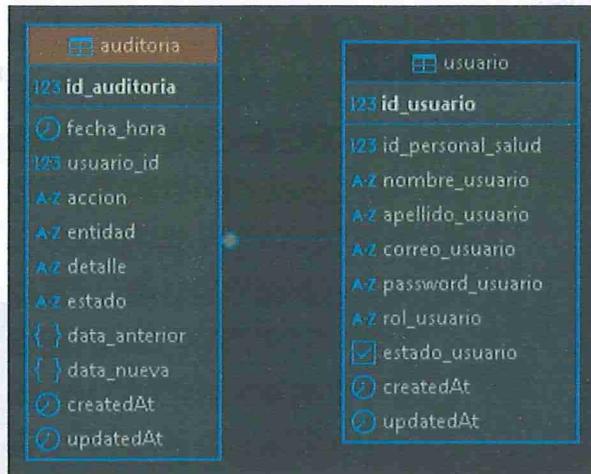


Figura 4.Diseño de BD Esquema de la base de datos diseñada para el módulo de auditorías. Contiene la estructura de la tabla principal con campos como: usuario, acción, fecha y detalle además de la relación con la tabla usuarios.

#### Anexo 6.Diagrama de flujo de navegación



Figura 5.Diagrama de flujo Este anexo presenta las modificaciones realizadas al flujo de navegación entre las secciones "Ver Auditorías" y "Exportar Auditorías". Estas actualizaciones incluyen ajustes en los diagramas de flujo y la implementación de nuevos puntos de interacción para mejorar la experiencia del usuario y garantizar un acceso más intuitivo a las funcionalidades del módulo de auditorías.

#### Anexo 7. Especificación de Requisitos V1.1

Documento actualizado que detalla los requisitos funcionales y no funcionales del módulo de auditorías.

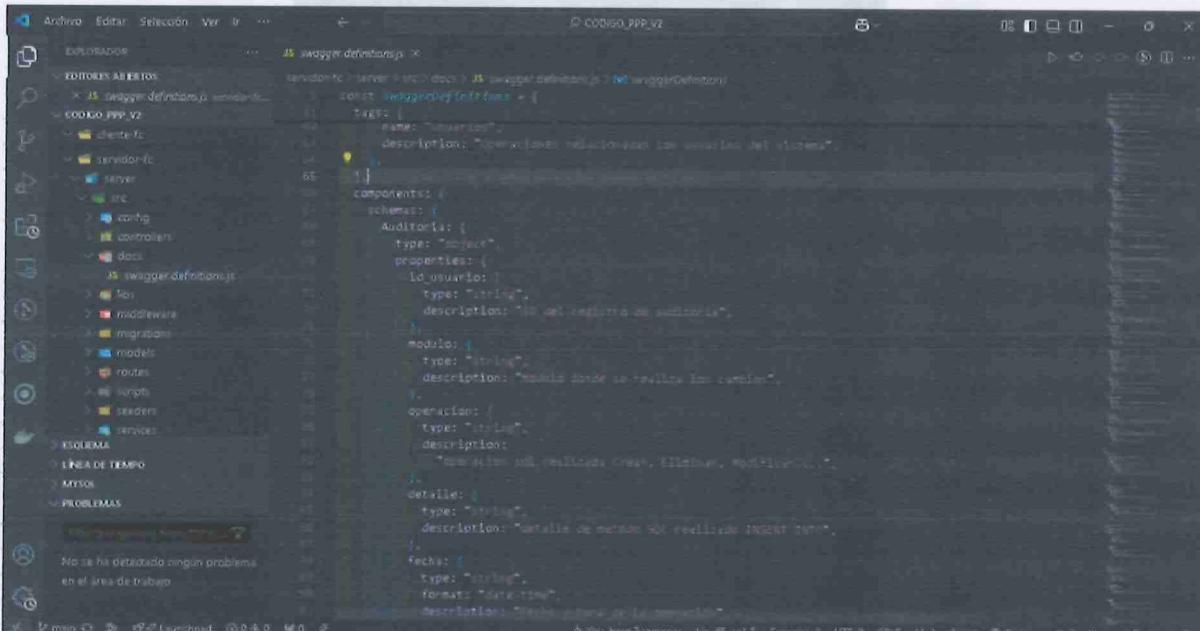
Yanzapanta, B. S y Yépez, C. D. (2024). *Especificación de requisitos versión 1.1* [PDF]. Recuperado de [https://drive.google.com/file/d/1WRcgG-jhtkF4r\\_FLZPSGkrAL5R0IFwCD/view?usp=sharing](https://drive.google.com/file/d/1WRcgG-jhtkF4r_FLZPSGkrAL5R0IFwCD/view?usp=sharing)

### Anexo 8. Casos de Uso Especificación Expandida V1.2

Documento que contiene una descripción detallada de los casos de uso, enfocándose en el módulo de Administración para la implementación de la sección de auditorías en las páginas 51-54 y 65.

Yanzapanta, B. S y Yépez, C. (2024). Casos de uso especificación expandida versión 1.2 [PDF]. Recuperado de <https://drive.google.com/file/d/1GilfB66AdAMXAoRTYoeJ4oU4HdojP3Tl/view?usp=sharing>

### Anexo 9. Desarrollo Back-end

A screenshot of a code editor window titled "CODIGO\_PPP\_V2". The file being edited is "swaggerDefinitions.js". The code defines a "tag" object with a "name" of "Auditoria" and a "description" of "Operaciones realizadas por usuarios del sistema". It also defines a "componentes" object with a "schemas" array containing an "Auditoria" schema. This schema includes properties for "id\_usuario" (type: "string", description: "ID del usuario de auditoría"), "modulo" (type: "string", description: "Módulo donde se realiza la acción"), "operacion" (type: "string", description: "Operación que se realizó: Create, Edit, Delete, Read"), "detalle" (type: "string", description: "Detalle de método que se realizó insert, update"), and "fecha" (type: "string", format: "date-time", description: "Fecha y hora de realización").

```
codigoppv2
swaggerDefinitions.js
tag: {
    name: "Auditoria",
    description: "Operaciones realizadas por usuarios del sistema"
}
componentes: {
    schemas: [
        Auditoria: {
            type: "object",
            properties: {
                id_usuario: {
                    type: "string",
                    description: "ID del usuario de auditoría"
                },
                modulo: {
                    type: "string",
                    description: "Módulo donde se realiza la acción"
                },
                operacion: {
                    type: "string",
                    description: "Operación que se realizó: Create, Edit, Delete, Read"
                },
                detalle: {
                    type: "string",
                    description: "Detalle de método que se realizó insert, update"
                },
                fecha: {
                    type: "string",
                    format: "date-time",
                    description: "Fecha y hora de realización"
                }
            }
        }
    ]
}
```

Figura 6.Captura de pantalla del código de implementación de las rutas en Swagger, donde se define el tag y la descripción de cada componente. Además, se incluyen los métodos HTTP GET, SET y PUT, asegurando una documentación clara y estructurada de la API.

### Anexo 10. Desarrollo Front-end

```

const Auditoria = () => {
  return (
    <Box sx={{ display: "flex" }}>
      <Grid container spacing={2} justifyContent="center">
        <Grid item xs={2} md={6} lg={5} sx={{ display: "flex" }}>
          <Card>
            <CardContent sx={{ flex: 1, display: "flex", flexDirection: "column", transition: "transform 0.3s", "&:hover": { transform: "translateY(-5px)", boxShadow: 4 } }}>
              <Box>
                <Typography>

```

Figura 7. Captura de pantalla del diseño gráfico de la pantalla principal del módulo de auditorías, que presenta tarjetas de selección para visualizar y exportar registros de auditoría, facilitando la navegación y gestión de la información.

### Anexo 11. Fase de pruebas



Figura 8. Evidencia fotográfica de la sesión de inducción sobre el módulo de auditorías, dirigida a la directora de la fundación y al personal de salud. Durante la sesión, se explicó el funcionamiento del sistema y se realizaron pruebas para validar su correcto desempeño.

Video demostrativo del funcionamiento del módulo de auditorías, con una duración de 10 a 20 minutos. Incluye la interacción de la directora de la fundación utilizando la nueva sección, así como comentarios y retroalimentación del personal de salud.

<https://youtu.be/Cgz-caOJiX8>

### Anexo 12. Documento Manual de uso

El documento titulado Manual de uso. V1.0 (2024) es un documento utilizado para recopilar datos relevantes durante la etapa inicial del diseño del sistema. Este incluye información clave sobre los requerimientos funcionales y no funcionales identificados en reuniones con la Fundación con Cristo.

Rodriguez B. (2024). Manual de usuario. V1.0 [PDF]. Recuperado de [https://github.com/saoricoder/4805\\_PPP\\_Documentacion/blob/93cd318415ec482b07f4542a5e2ef6615f9151b5/Entregables/Documentaci%C3%B3n/Manual%20de%20uso%20-%20fcc%20auditorias%20V1.0.pdf](https://github.com/saoricoder/4805_PPP_Documentacion/blob/93cd318415ec482b07f4542a5e2ef6615f9151b5/Entregables/Documentaci%C3%B3n/Manual%20de%20uso%20-%20fcc%20auditorias%20V1.0.pdf)

### **Anexo 13.Cronograma de actividades**

Se presenta un cronograma detallado de actividades y tareas realizadas, el cual ha sido elaborado utilizando Google Sheets. Este cronograma refleja las diferentes fases y tareas involucradas en el proyecto, proporcionando un seguimiento claro y ordenado de las actividades realizadas.

Rodríguez B. (2024). *Actividades prácticas* versión 1.0[Google Sheets]. Recuperado de [https://docs.google.com/spreadsheets/d/1tmuojXZ0p4O7cGevxCR31\\_Pf40zR91G18Unaet5IOg/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1tmuojXZ0p4O7cGevxCR31_Pf40zR91G18Unaet5IOg/edit?usp=sharing)