



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A

TITULO DEL PROYECTO

PLATAFORMA INFORMÁTICA PARA LA RECOLECCIÓN DE INFORMACIÓN DEL SG-SST

PROPONENTE

Cristhian Stiven Camargo Reyes

C.C 1049657040

2251168

DIRECTOR

Carlos Andres Guerrero Alarcon

Tunja

22 de noviembre del 2022

TABLA DE CONTENIDIO

1	FICHA TÉCNICA DEL PROYECTO	7
2	PLANTEAMIENTO DEL PROBLEMA	8
3	JUSTIFICACIÓN	10
4	OBJETIVOS	11
4.1	Objetivo general	11
4.2	Objetivos específicos	11
5	MARCO TEÓRICO	12
5.1	Lenguaje de programación:	12
5.1.1	JavaScript	13
5.1.2	TypeScript	14
5.2	Bases de datos	14
5.3	Frontend	15
5.3.1	Html	15
5.3.2	CSS	16
5.3.3	Angular	17
5.3.4	Angular CLI	17
5.4	Backend	18
5.4.1	Express	19
5.4.2	Sequelize	19
5.4.3	Json Web Token (JWT)	20
6	METODOLOGÍA	21

6.1	Fase 1: Análisis (Requerimientos)	22
6.2	Fase 2: Diseño	22
6.3	Fase 3: Codificación	23
6.4	Fase 4: Pruebas	23
7	DESARROLLO	24
7.1	Análisis	24
7.2	Diseño	29
7.2.1	Diagrama de clases	31
7.2.2	Diagrama de casos de uso	31
7.2.3	Diagrama de actividades	33
7.2.4	Diagrama de componentes	36
7.2.5	Diagrama de despliegue	36
7.2.6	Base de datos	37
7.2.7	Diseño Visual	39
7.3	Codificación	42
7.3.1	Backend:	42
7.3.2	Frontend:	48
7.4	Pruebas	59
8	CONCLUSIONES	60
9	REFERENCIAS	61
10	ANEXOS	63

INDICE DE TABLAS

Tabla 1.	Objetivos específicos	11
Tabla 2.	Historias de usuario - Administración	25
Tabla 3.	Historias de usuario – Participación.....	27

INDICE DE FIGURAS

Figura 1.	Diagrama de Ishikawa _____	9
Figura 2.	Modelo de desarrollo iterativo e incremental_____	21
Figura 3.	Diagrama de clases_____	31
Figura 4.	Diagrama de casos de uso _____	32
Figura 5.	Diagrama de actividades – Validación del rol _____	34

Figura 6.	Diagrama de actividades - Repuesta de encuesta	35
Figura 7.	Diagrama de componetes	36
Figura 8.	Diagrama de despliegue	37
Figura 9.	Diagrama de base de datos	38
Figura 10.	Wireframe - Acceso	40
Figura 11.	Wireframe - Dashboard Admnistrador	41
Figura 12.	Wireframe - Dashboard participantes	41
Figura 13.	Arquetipo backend	43
Figura 14.	Carpeta - controllers	44
Figura 15.	Carpeta - Database	44
Figura 16.	configuración de conexión	45
Figura 17.	carpeta dist	45
Figura 18.	Carpeta helpers	46
Figura 19.	Carpeta middlewares	46
Figura 20.	Carpeta models	47
Figura 21.	Carpeta routes	48
Figura 22.	Acceso frontend	49
Figura 23.	Home administrador	50

Figura 24.	Creación de usuarios	51
Figura 25.	Administrar usuarios	51
Figura 26.	Administrar usuarios	52
Figura 27.	Administrar preguntas	53
Figura 28.	Listar comentarios	54
Figura 29.	Analítica	55
Figura 30.	Home participante	56
Figura 31.	Encuesta	57
Figura 32.	Encuesta respondida	58
Figura 33.	Ver resultados	58
Figura 34.	Ver resultados – inactivo	59
Figura 35.	Colección de pruebas en postman	60

1 FICHA TÉCNICA DEL PROYECTO

Título	Desempeño del SG-SST
Nombre Estudiante	Cristhian Stiven Camargo Reyes
Documento estudiante	C.C 1.049.657.040
Correo electrónico	cristhian.camargo@usantoto.edu.co
Director	Carlos Andres Guerrero Alarcon
Lugar de ejecución del proyecto	Universidad Santo Tomás de Tunja
Duración	6 meses
Costo	35.140.000 de pesos
Palabras claves	Diseño de sistemas, Seguridad en el trabajo, Datos estadísticos



Firma del autor

Cristhian Stiven Camargo Reyes

Firma del director

Carlos Andres Guerrero Alarcon

2 PLANTEAMIENTO DEL PROBLEMA

El Sistema de Gestión de Seguridad y Salud en el Trabajo comprende la planificación y organización de mejoras sobre la prevención de riesgos laborales que afecten a la seguridad y salud de los trabajadores, así mismo, los empleadores implementan el SG-SST haciendo partícipe a los empleados para garantizar la mejora del ambiente laboral y la optimización de todas las medidas de seguridad y salud.

El SG-SST está basado en el ciclo **PHVA**, consiste en elaborar un proceso donde se tengan en cuenta unas etapas de planificación, hacer, verificar y actuar, donde la **planificación** debe basarse en buscar las mejoras que se pueden implementar en la seguridad y salud para luego ser implementadas en la etapa del **hacer**, donde será evaluado en la etapa de **verificación** para revisar que la implementación conseguirá los resultados esperados y terminar con la fase de **actuar** donde se realizan las acciones necesarias para la mejora de la seguridad y salud de los empleados.

El sistema de gestión aplica a todos los empleadores públicos y privados, los trabajadores dependientes e independientes, los trabajadores cooperados, los trabajadores en misión, los contratantes de personal bajo modalidad de contrato civil, comercial o administrativo, las organizaciones de economía solidaria y del sector cooperativo, las empresas de servicios temporales, las agremiaciones y asociaciones que afilian trabajadores independientes al Sistema de Seguridad Social Integral; las administradoras de riesgos laborales; la Policía Nacional en lo que corresponde a su personal no uniformado y al personal civil de las Fuerzas Militares. [1]

Por lo anterior, se conoce que la Universidad Santo Tomás aplica el Sistema de Gestión de Seguridad y Salud en el Trabajo para la mejora del comportamiento y de las condiciones del ambiente laboral, para ello, se requiere de colaboradores para tener un buen control de los riesgos en el trabajo. La participación de este rol es efectiva para mantener un orden en los lugares de trabajo, obtener información sobre el estado de salud de los empleados y verificar el cumplimiento de los estándares mínimos del SG-SST.

Teniendo en cuenta la implementación del SG-SST en la Universidad Santo Tomás, se planteó el desarrollo de una plataforma para la recolección de información sobre la seguridad y salud de los empleados, facilitando la mejora y la optimización de este sistema dentro de la institución, ya que el proceso actual puede llegar a requerir de gran cantidad de tiempo para la recolección de estos datos y darles un debido proceso.

A continuación se presenta el diagrama de Ishikawa donde se representan los problemas del proyecto (Figura 1).

Figura 1. Diagrama de Ishikawa



3 JUSTIFICACIÓN

Basándose en la capacidad de mejora y evolución de la Universidad Santo Tomás, se desea implementar una plataforma que permita la comunicación con el fin de relacionar los trabajadores de la universidad con el desempeño del Sistema de Gestión de Seguridad y Salud en el Trabajo.

Planteando la elaboración de las encuestas que serán contestadas por los mismos trabajadores, se comprenderá de forma más sencilla y segura el desempeño de los estándares mínimos requeridos, acompañado de una serie de gráficas que facilitarán la comprensión de los datos, orientados al nivel de desempeño de esta normativa dentro de la Universidad Santo Tomás.

Esta plataforma ayudará a brindar un espacio específico para determinar el desempeño del SG-SST a partir de los involucrados en la misma, lo que ayudará a dar resultados acertados con un mínimo de margen de error y permitiendo generar un espacio de opinión controlado por un sistema de acceso único para determinados empleadores garantizando la veracidad de la información. Contará con un módulo para la administración de estas encuestas y así mantener un control sobre la plataforma.

4 OBJETIVOS

4.1 Objetivo general

Desarrollar una plataforma web para recolectar información sobre el cumplimiento de estándares del SG-SST en la Universidad Santo Tomás, utilizando encuestas basadas en la normativa establecida por el Ministerio de Trabajo.

4.2 Objetivos específicos

Tabla 1. Objetivos específicos

No.	Objetivo específico
1	Elaborar el levantamiento de requerimientos para identificar las necesidades de la plataforma Web, con la elaboración de documentación y diagramas.
2	Diseñar prototipos y arquitectura de la plataforma utilizando el modelo de vistas de arquitectura 4+1, para establecer las pautas de diseño para el desarrollo.
3	Desarrollar una plataforma web para el seguimiento del Sistema de Gestión de Seguridad y Salud en el Trabajo, diseñando encuestas teniendo en cuenta los estándares mínimos requeridos estipulados.
4	Demostrar el funcionamiento óptimo de la plataforma para garantizar la exactitud de los datos y una buena experiencia de usuario utilizando las pruebas de usabilidad, unitarias y de integración.

Fuente: Autor

5 MARCO TEÓRICO

En el desarrollo web se aplican gran variedad de conceptos que son aplicados para ejecutar cada fase del proyecto. A continuación se presentaran los conceptos utilizados en el desarrollo de la plataforma para medir el desempeño del SG-SST.

5.1 Lenguaje de programación:

Un lenguaje de programación es una herramienta que permite desarrollar software o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora. Lo anterior se logra mediante la creación e implementación de algoritmos de precisión que se utilizan como una forma de comunicación humana con la computadora.[10]

Los lenguajes de programación son la base para el desarrollo de plataformas web, pues estos nos ayudan a establecer las funcionalidades que son establecidas por la planeación. Estos lenguajes ayudan a establecer el orden, modo y lógica con la que se ejecuta una rutina dentro de la plataforma, como por ejemplo, con el lenguaje de programación typescript establecimos los lineamientos que debe tener el programa para establecer si un usuario puede o no responder la encuesta. Así mismo, estas peticiones son desarrolladas de acuerdo a unos requerimientos establecidos, y de este modo se lleva una serie de planeaciones y requisitos a lógica programada con el lenguaje seleccionado.

5.1.1 JavaScript

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web.

Anteriormente, las páginas web eran estáticas, similares a las páginas de un libro. Una página estática mostraba principalmente información en un diseño fijo y no todo aquello que esperamos de un sitio web moderno. JavaScript surgió como una tecnología del lado del navegador para hacer que las aplicaciones web fueran más dinámicas. Por medio de JavaScript, los navegadores eran capaces de responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.[7]

Varias de las librerías usadas para frontend o backend están basadas en javascript, esto permite efectuar reuso de librerías para el ahorro de tiempos de desarrollo. Así mismo javascript se usa de forma indirecta, pues estas librerías permiten usar este código sin necesidad de entrar en detalles o profundizar con la programación. De esta misma forma se usan librerías para transpilar el código typescript a javascript, el cual es el interprete que se usa para el montaje de un servicio.

5.1.2 TypeScript

TypeScript es un lenguaje de programación que perfecciona JavaScript al agregar tipos en la sintaxis. TypeScript agrega sintaxis adicional a JavaScript de modo que las herramientas de edición de código puedan detectar errores de manera temprana. Al mismo tiempo, el código de TypeScript se convierte a JavaScript y brinda los mismos beneficios que JavaScript. También se ejecuta en aplicaciones, así como con marcos y bibliotecas de JavaScript.[7]

Este lenguaje a evolucionado y se ha convertido en un lenguaje muy potente al poder usarse tanto del lado del cliente como del servidor, ofreciendo librerías para el desarrollo que más convenga dependiendo de la capa que se esté desarrollando.

5.2 Bases de datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.[11]

5.3 Frontend

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.[2]

De esta forma, se crea el frontend teniendo en cuenta quienes serán los usuarios finales que darán uso a la plataforma, además de procurar mantener una buena experiencia de usuario haciendo que esta capa sea lo más intuitiva y fácil de usar para los clientes. Este objetivo se planea desde la fase inicial del proyecto, ya que se elaboran diferentes prototipos pensando en la usabilidad y el formato visual que tendrá la plataforma para poder continuar con el desarrollo de los componentes que serán necesarios.

5.3.1 *Html*

El lenguaje de marcado de hipertexto (HTML) es un lenguaje informático que forma parte de la mayoría de las páginas web y aplicaciones en línea. Un hipertexto es un texto que se utiliza para enlazar con otros textos, mientras que un lenguaje de marcado es una serie de marcas que indican a los servidores web la estructura y el estilo de un documento.[3]

Con HTML se elabora el maquetado y la estructura base que va a tener la plataforma web, utilizando etiquetas con propiedades específicas que van a ayudar a identificar los componentes y su orden. El navegador se encarga de interpretar esta estructura desarrollada para mostrarlos de la forma indicada según su finalidad. Estas instrucciones

vienen acompañadas por otros lenguajes que ayudarán a la visualización de estas etiquetas aplicando animaciones, colores, tamaños y funcionalidades con otras funcionalidades como CSS.

5.3.2 CSS

CSS son las siglas en inglés de Cascading Style Sheets, que significa «hojas de estilo en cascada». Es un lenguaje que se usa para estilizar elementos escritos en un lenguaje de marcado como HTML.[4]

Aunque HTML es un lenguaje muy poderoso para el maquetado de elementos y componentes, no es suficiente para que el sitio web cuente con una buena interfaz de usuario. Por esta razón, se usa CSS para agregar estilos a toda la plataforma teniendo en cuenta las etiquetas usadas en la estructura del sitio web, de esta forma se puede modificar los colores, formas, transiciones, tamaños, posiciones de componentes, entre otras más funcionalidades.

Actualmente existen otros lenguajes que ayudan a estilizar componentes en HTML, sin embargo, CSS sigue siendo de los más populares debido a su confiabilidad y su sencillez al momento de aplicarlo en un proyecto.

5.3.3 Angular

Angular es una plataforma de desarrollo, construida sobre TypeScript. Es un framework basado en componentes para crear aplicaciones web escalables. Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más. Un conjunto de herramientas para desarrolladores que permiten desarrollar, compilar, probar y actualizar el código fuente de la aplicación.[5]

Este framework nos permite elaborar componentes de forma más rápida y sencilla debido a que tiene diferentes funcionalidades que en conjunto con HTML y el lenguaje de programación TypeScript, hacen que el desarrollo y la comunicación de componentes sea mucho más efectiva. Angular cuenta con mucha documentación y librerías que permiten elaborar componentes a partir de unas estructuras definidas, de esta forma se ahorra tiempo en el desarrollo de funcionalidades en la página web y garantiza una experiencia más agradable para el usuario.

5.3.4 Angular CLI

Angular CLI es un Command Line Interface (interfaz en línea de comandos, en español), desarrollada por los equipos de Angular. Este CLI permite crear proyectos en los que el CLI podrá añadir archivos y más exactamente, entidades Angular. Será posible añadir módulos, componentes, servicios o directivas en una línea de comandos.[6]

La interfaz en línea de comandos facilita la configuración de un proyecto angular, ofrece los comandos suficientes para la creación de la estructura del proyecto facilitando la utilización, enrutamiento y desarrollo de todas las utilidades del sitio web.

5.4 Backend

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios. Además, contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma en la que el navegador hace solicitudes.[2]

Para el desarrollo de backend se hace uso de diferentes librerías, sin embargo, se debe tener en cuenta las tecnologías que se trabajaran desde un principio, pues no todas las librerías soportan el uso de determinadas tecnologías. Se debe tener en cuenta la planeación inicial para iniciar un desarrollo efectivo y productivo que permita completar la capa de backend sin vulnerabilidades.

El backend será esta fase que vuelve funcional el frontend, y de este modo, por medio de servicios y controladores se pueden efectuar cambios directos sobre la base de datos. Estos datos se actualizarán y cambiarán dinámicamente por medio de lógica ejecutada de forma ordenada y sincronizada.

Cada petición se le puede denominar operaciones, las cuales pertenecen a unas denominaciones generales conocidas como “métodos de peticiones HTTP”. Estos métodos

permitirán generar determinadas acciones sobre datos dinámicos en una base de datos, los cuales serán programados en el backend para ser llamados o solicitados desde el frontend.

5.4.1 Express

Es un framework de backend Node.js minimalista, rápido y similar a Sinatra, que proporciona características y herramientas robustas para desarrollar aplicaciones de backend escalables. Te ofrece el sistema de enrutamiento y características simplificadas para ampliar el framework con componentes y partes más potentes en función de los casos de uso de tu aplicación.[8]

Express nos ayuda en gran medida para la programación lógica del backend ya que nos permite desarrollar las peticiones con los métodos HTTP para el manejo de los datos, además de poder manejar las rutas de dichas peticiones para poder ser localizadas desde el frontend.

5.4.2 Sequelize

Sequelize es una herramienta ORM de Node.js basada en promesas para Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server, Amazon Redshift y Snowflake's Data Cloud. Cuenta con un sólido soporte de transacciones, relaciones, carga ansiosa y diferida, replicación de lectura y más.[9]

Esta herramienta nos permite mapear la estructura de las bases de datos relacionales, de esta forma se puede vincular la distribución con código lógico basado en entidades u objetos con identidades específicas. Esto permite establecer una sincronización más adecuada entre la estructura de la base de datos y el backend, facilitando el tratamiento de datos y la equidad de los mismos.

5.4.3 Json Web Token (JWT)

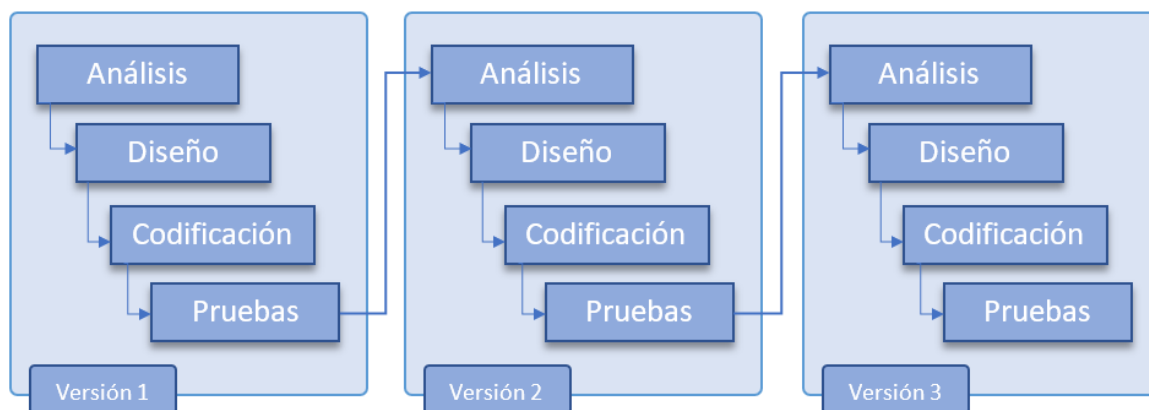
JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto JSON. Esta información se puede verificar y confiar porque está firmada digitalmente. Los JWT se pueden firmar usando un secreto (con el algoritmo HMAC) o un par de claves pública/privada usando RSA o ECDSA.[12]

Esta librería se usa para la autenticación de usuarios o para el envío de información que requiere de permisos o accesos restringidos. La forma de transmitir la información se hace a través “tokens”, los cuales comprenden una serie de caracteres generados automáticamente y de carácter único bajo un tiempo de caducidad donde la información podrá ser recibida únicamente si el token es válido.

6 METODOLOGÍA

Se realiza un análisis respecto a las necesidades del proyecto, evaluando las fases que se requieren para el desarrollo teniendo en cuenta los requerimientos del sistema. Por ello, se plantea un modelo de desarrollo iterativo e incremental para el presente proyecto ya que permite ejecutar las fases del proyecto una vez concluidas, dando paso a revisiones antes de iniciar la siguiente fase. Además, es un modelo que permite elaborar una buena documentación durante el desarrollo, estimando el tiempo y costes desde la primera fase. Toma la mejor parte del modelo cascada, pero complementando las iteraciones para un mejor flujo de trabajo y determinando cada una de las fases estrictamente.

Figura 2. Modelo de desarrollo iterativo e incremental



6.1 Fase 1: Análisis (Requerimientos)

Partiendo de una fase investigativa, se hará toda la obtención de datos y conocimientos, además de la planificación de las siguientes fases. Se planifica la investigación y las fuentes de la información para verificar la veracidad de los datos que se registran en la plataforma.

Se determina la gran importancia de la investigación y la obtención del conocimiento sobre estos estándares emitidos por el ministerio de trabajo ya que puede ser información sensible. Por ello, la primera fase se concentrará en la consulta e investigación de la resolución 0312 del 13 de febrero de 2019, donde se encuentra la información completa sobre los estándares que se deben aplicar en la Universidad Santo Tomás.

Se iniciará con el levantamiento de requerimientos del negocio para poder comprender cuales son las necesidades de la plataforma WEB, de esta forma, se tendrá un panorama más amplio para el momento del desarrollo.

Luego de profundizar sobre el tema principal de la plataforma y los requerimientos, se establecerán las tecnologías que se usarán para el desarrollo del proyecto, teniendo enfoque en el desarrollo web.

6.2 Fase 2: Diseño

Con la información recolectada en la fase 1, se realizarán wireframes y mockups para determinar con seguridad la aprobación de una propuesta gráfica aplicando los estándares de la Universidad Santo Tomás.

Esta propuesta gráfica deberá contar con todas las utilidades de la plataforma de manera no funcional trabajando los espacios, tamaños, colores y todos los aspectos visuales que inspirarán comodidad a los usuarios.

Se elaborará un plan de diseño donde se especificarán los entornos de trabajo que se usarán, componentes, librerías entre otros aspectos relacionados con la arquitectura de software. Desde esta fase, se planificará el proceso de desarrollo.

6.3 Fase 3: Codificación

Teniendo en cuenta las tecnologías seleccionadas, se comenzará con el desarrollo de la aplicación web y la transformación de los prototipos no funcionales a unos prototipos codificados y funcionales, buscando el desarrollo del front-end para las primeras versiones. En versiones posteriores se trabajará el back-end una vez aprobadas las propuestas y diseños del front-end.

6.4 Fase 4: Pruebas

Se procede a testear la funcionalidad de la plataforma, buscando posibles errores antes de finalizar la iteración buscando crear una versión sin fallos para no contar con errores antiguos en futuras iteraciones y no disminuir el ritmo y aumentar el tiempo de desarrollo.

Se tendrán en cuenta inicialmente las pruebas funcionales y unitarias para verificar el correcto funcionamiento de la plataforma.

7 DESARROLLO

7.1 Análisis

Teniendo en cuenta los requerimientos planteados en la tapa inicial del proyecto “Plataforma informática para la recolección de información del SG-SST”, se determinan una serie de diagramas, mapeos de datos, módulos y/o modelos con los que puede contar la plataforma web. Estos requerimientos funcionales y no funcionales nos indicaron cuales son las interacciones y funcionalidades necesarias para cumplir con el objetivo del software, por ello, se procede a evaluar los escenarios donde el usuario ejecuta diferentes acciones en la plataforma.

Como punto de partida se establece las personas involucradas en el uso del software, ya que esto indica el punto de partida para poder identificar cuáles son las interacciones que tiene un usuario determinado dentro de la plataforma. Por esta razón, se establecen dos interacciones a nivel general, denominadas:

- **Administración:** usuarios con capacidad de edición, modificación e interacción de datos dentro de la plataforma, además de visualización de datos detallados para la recolección de datos específicos necesarios para la evaluación del desempeño del SG-SST.
- **Participación:** usuarios sin ningún tipo de permisos de modificación de datos. Tendrá la capacidad de contestar la encuesta y de visualización de resultados generales sin gran detalle. Este usuario contribuirá por medio de sus respuestas a el calculo de nivel de satisfacción del SG-SST, por lo tanto, es el usuario base y primordial para el funcionamiento de esta plataforma.

A partir de estas definiciones, se puede comenzar con el análisis de las historias de usuario teniendo en cuenta los tipos de interacción mencionados anteriormente.

Tabla 2. Historias de usuario - Administración

Administración	
HU1	Yo como usuario administrador, deseo agregar usuarios nuevos a la plataforma para que en calidad de participantes puedan diligenciar la encuesta.
HU2	Yo como usuario administrador, deseo agregar usuarios nuevos a la plataforma para que en calidad de administrador para que pueda interactuar con la información registrada en la plataforma.
HU3	Yo como usuario administrador, deseo editar usuarios para soportar actualizaciones de datos o cambios de rol que sean necesarios.
HU4	Yo como usuario administrador, deseo eliminar usuarios para no tener en cuenta el cálculo de respuestas de usuarios inactivos.
HU5	Yo como usuario administrador, deseo agregar preguntas a la encuesta cuando sea necesario para dar continuidad a la evolución del SG-SST en la Universidad Santo Tomás seccional Bogotá.
HU6	Yo como usuario administrador, deseo editar preguntas de la encuesta para tener la oportunidad de corrección o rectificación de preguntas.
HU7	Yo como usuario administrador, deseo eliminar preguntas para mantener preguntas de interés y evaluar diferentes puntos de vista de los participantes.

HU8	Yo como usuario administrador, deseo visualizar los comentarios anónimos de los participantes para tener opiniones y sugerencias más claras.
HU9	Yo como usuario administrador, deseo eliminar comentarios para poder llevar visualizar de forma correcta los nuevos comentarios de los participantes.
HU10	Yo como usuario administrador, deseo visualizar gráficos de los resultados para tener la información más clara y legible.
HU11	Yo como usuario administrador, deseo responder encuesta para contribuir con el sistema de evaluación y proyectar mi nivel de satisfacción en el sistema.
HU12	Yo como usuario administrador, deseo visualizar mis resultados personales para tener en cuenta mis respuestas seleccionadas y mi nivel de satisfacción.
HU13	Yo como usuario administrador, deseo acceder por medio de correo y contraseña para poder hacer uso de la plataforma.
HU14	Yo como usuario administrador, deseo poder cerrar sesión para conservar mi privacidad y posibles accesos indeseados.

Fuente: Autor

Tabla 3. Historias de usuario – Participación

Participación	
HU1	Yo como usuario participante, deseo responder encuesta para contribuir con el sistema de evaluación y proyectar mi nivel de satisfacción en el sistema.
HU2	Yo como usuario participante, deseo visualizar mis resultados personales para tener en cuenta mis respuestas seleccionadas y mi nivel de satisfacción.
HU3	Yo como usuario participante, deseo acceder por medio de correo y contraseña para poder hacer uso de la plataforma.
HU4	Yo como usuario participante, deseo poder cerrar sesión para conservar mi privacidad y posibles accesos indeseados.

Fuente: Autor

Con estas historias de usuario podemos definir los posibles módulos que serán necesarios en la plataforma, donde la mayoría de estos serán pertenecientes al usuario administrador debido a su amplia capacidad de uso que tiene en la plataforma. A continuación, se definirán los escenarios pautados en la presente fase de análisis junto con el tipo de usuario con acceso a ellos.

- **Modulo 1: Administración de usuarios (Administrador).**
 - Crear usuario.
 - Borrar usuario.
 - Editar usuario.
 - Obtener nivel de satisfacción por usuario.
- **Modulo 2: Administración de preguntas (Administrador).**

- Crear pregunta.
- Borrar pregunta.
- Editar pregunta.
- Obtener nivel de satisfacción por pregunta.
- **Modulo 3: Comentarios (Administrador).**
 - Obtener comentarios.
 - Borrar comentario.
- **Modulo 4: Analítica (Administrador).**
 - Visualización de gráficos.
 - Obtener porcentajes de satisfacción por pregunta.
 - Obtener número de respuestas por opción.
- **Modulo 5: Encuesta (Administrador, Participante).**
 - Envío de respuestas.
- **Modulo 6: Ver resultados (Administrador, Participante).**
 - Obtener nivel de satisfacción genera.
 - Obtener nivel de satisfacción personal.
 - Obtener listado de preguntas con respuesta personal.
- **Modulo 7: Login (Administrador, Participante)**

Una vez analizados los requerimientos y definidos los que contiene el software, se puede pasar a la siguiente fase que es la de diseño. La fase de análisis se da por concluida gracias a la claridad de los requerimientos y el nivel de detalle de los mismos.

7.2 Diseño

Una vez completado el análisis del proyecto, se comienza con la elaboración de diagramas, graficos, warframes, y escenarios que surgen a partir de los módulos generados en la fase de análisis.

A partir de los usuarios que interactúan en la plataforma, se puede evaluar las clases requeridas para comenzar con un mapeo de diagrama de clases y/o esquema de base de datos, identificando así:

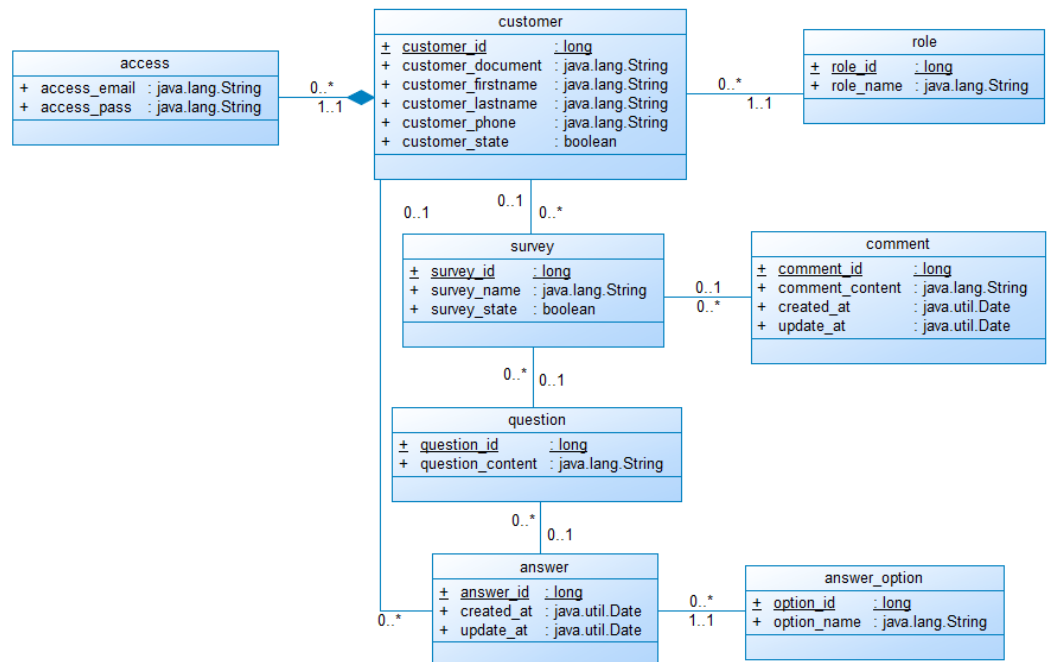
- **Customer:** usuario que interactúa con la plataforma teniendo en cuenta las asignaciones determinadas por la Universidad Santo Tomás. Este usuario podrá tener interacción con la plataforma de dos formas diferentes. De esta forma se determina los atributos:
 - Customer_id: identificador único de usuario dentro de la plataforma.
 - Customer_document: documento de identificación personal del usuario.
 - Customer_firstname: primer o primeros nombres del usuario.
 - Customer_lastname: apellidos del usuario.
 - Customer_phone: número telefónico de contacto del usuario.
 - Customer_state: estado actual del usuario, siendo **pendiente** si el usuario no a respondido la encuesta, o **completada** si el usuario ya respondió la encuesta.
- **Role:** role que el usuario desempeña dentro de la Universidad Santo Tomás de forma general.
 - Role_id: identificador único del rol dentro de la plataforma.
 - Role_name: nombre del rol establecido para la plataforma.

- **Access:** datos de acceso del usuario con el que podrá ingresar a la plataforma teniendo en cuenta las pautas institucionales.
 - Access_email: correo electrónico del usuario.
 - Access_pass: contraseña privada y encriptada del usuario.
- **Survey:** encuesta especifica donde se guardarán las preguntas.
 - Survey_name: nombre de la encuesta.
 - Survey_state: indica si la encuesta esta activa para ser respondida.
- **Comment:** comentarios de los usuarios guardados de manera anónima.
 - Comment_id: identificador único para los comentarios dentro de la plataforma.
 - Comment_content: comentario escrito por el usuario.
 - Created_at: fecha de creación del comentario.
 - Update_at: fecha de actualización del comentario.
- **Question:** preguntas pertenecientes a la encuesta para visualización de los usuarios.
 - Question_id: identificador único de pregunta dentro de la plataforma.
 - Question_content: contenido de la pregunta a publicar.
- **Answer:** respuesta especifica con guardado de datos de usuario.
 - Answer_id: identificador de la respuesta guardada por usuario.
 - Created_at: fecha de creación de la respuesta.
 - Update_at: fecha de actualización de la respuesta.
- **Answer_option:** posibles opciones a seleccionar en la encuesta.
 - Option_id: identificador único de la opción.
 - Option_name: nombre de la opción que será publicada en la encuesta.

7.2.1 Diagrama de clases

El diagrama de clases evidencia el flujo de los datos entre objetos. Cada clase con sus atributos determinados representan los datos que serán necesarios para el correcto flujo de información. Se establecen estas clases ya que son las suficientes para las respuestas de las encuestas, y son los datos necesarios que requiere el software para un análisis.

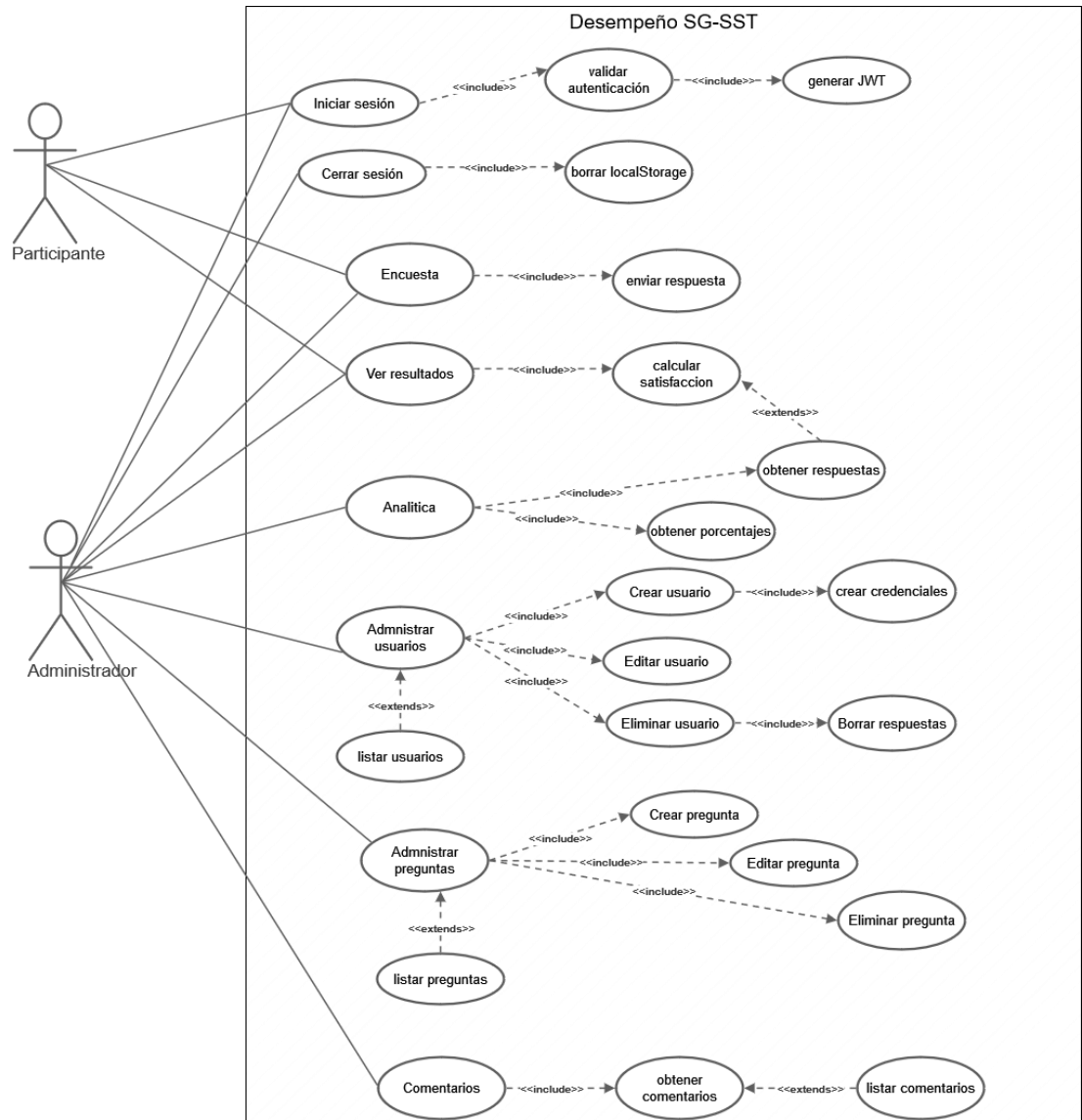
Figura 3. Diagrama de clases



7.2.2 Diagrama de casos de uso

Los casos de uso representan a dos participantes, los cuales se especificaron en la fase de análisis según correspondía la capacidad de uso del software, es decir, existían dos participantes esencialmente con la diferencia de permisos de acceso a la plataforma. Dependiendo de estos accesos, el usuario podrá pertenecer a determinado caso de uso como se evidencia en la **Figura 4**.

Figura 4. Diagrama de casos de uso



Se puede evidenciar que el administrador presenta más casos de uso, pues tiene acceso a más funcionalidades dentro de la plataforma. El participante es el usuario que responderá las encuestas con más frecuencia, en este aspecto, solamente podrá pertenecer a los casos de uso que se adecúan al uso de las encuestas y visualización de resultados generales y personales. De este modo, el único usuario que puede acceder a todas las funcionalidades será el administrador.

7.2.3 Diagrama de actividades

Se clasifican dos diagramas de actividades principalmente, para el proceso de la validación del rol que se hace desde la autenticación del usuario por medio del registro de credenciales en el formulario Login, y el diagrama del proceso para responder una encuesta. Estos dos procesos son los base para el uso del software, pues la validación del rol permite ejecutar los primeros pasos del programa con el que se validará el tipo de usuario y la pantalla que será mostrada, y la respuesta de la encuesta nos permite generar el calculo del nivel de satisfacción.

Figura 5. Diagrama de actividades – Validación del rol

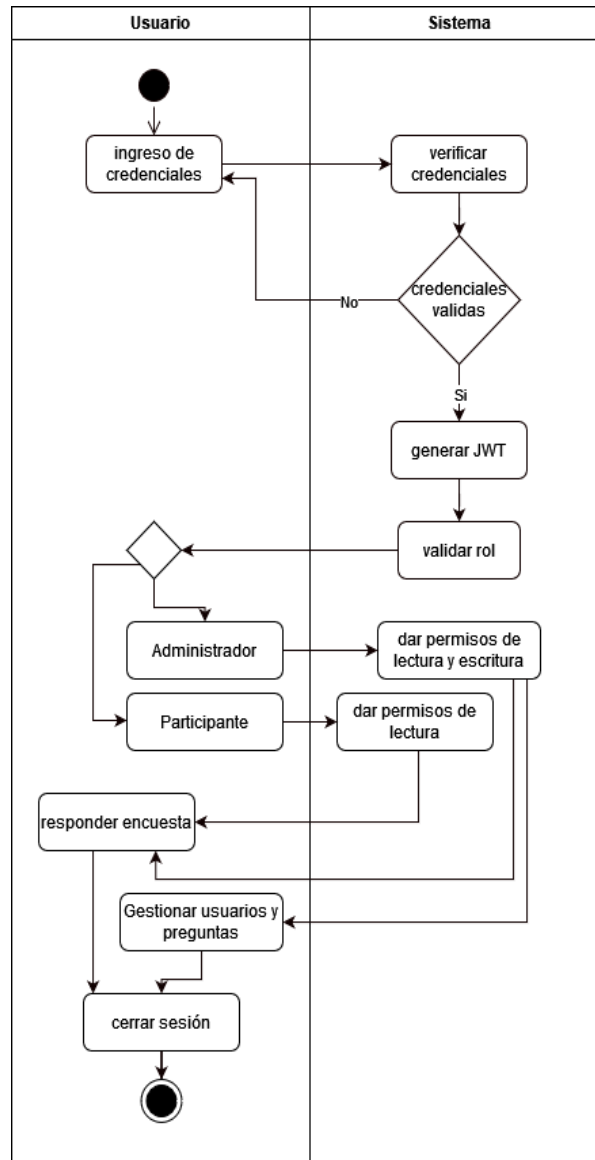
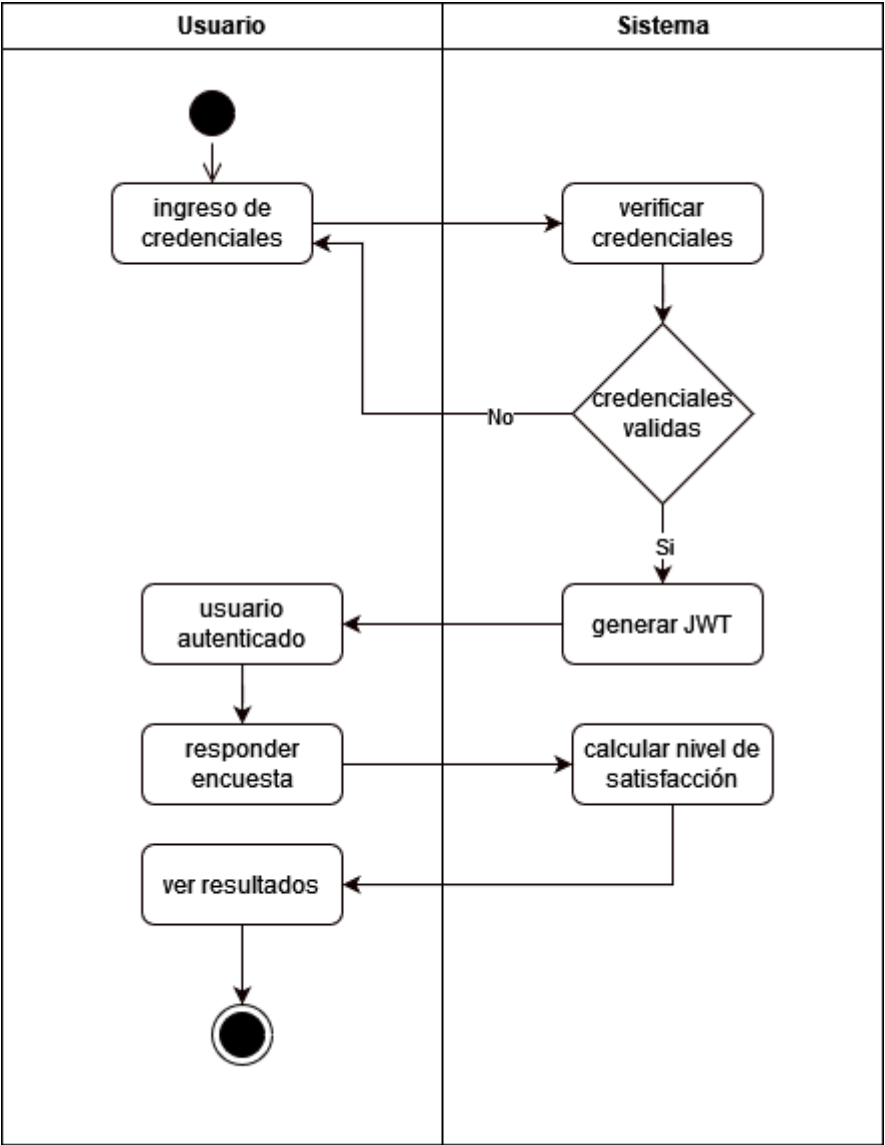


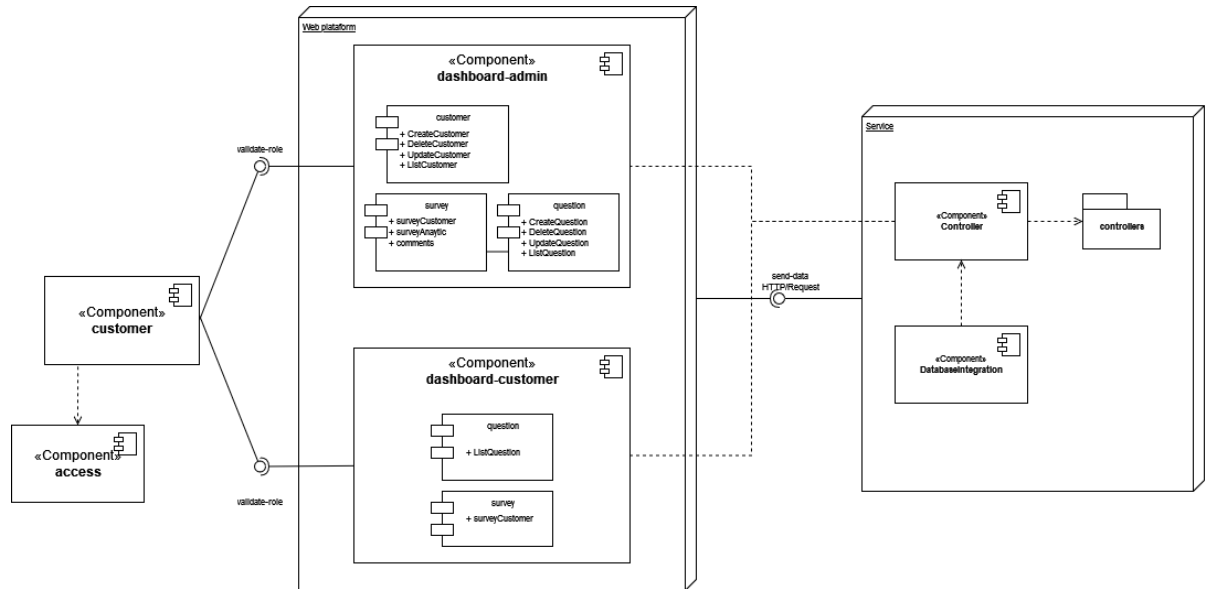
Figura 6. Diagrama de actividades - Repuesta de encuesta



7.2.4 Diagrama de componentes

En este diagrama se puede apreciar todos los componentes, módulos y escenarios principales que interactúan en el sistema, esta vista se genera a partir de las funcionalidades existentes y validaciones ejecutadas desde el frontend hasta el servidor.

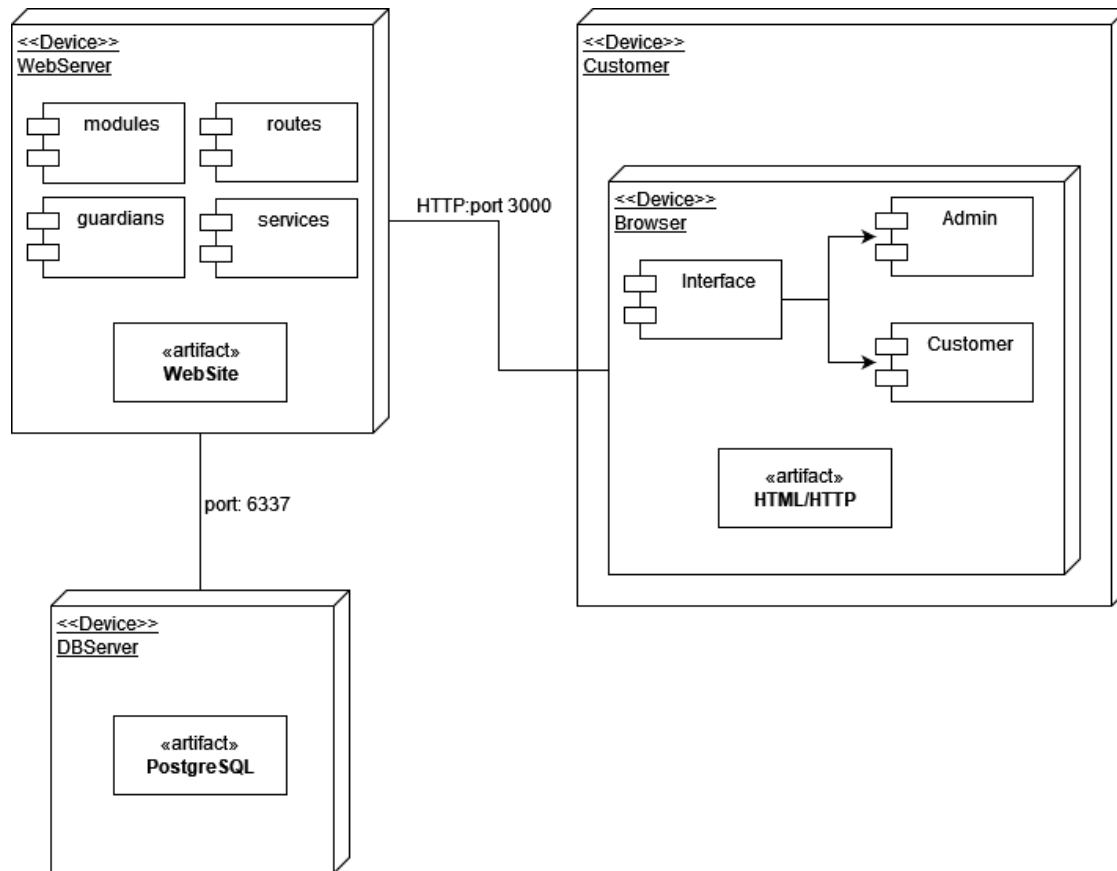
Figura 7. Diagrama de componetes



7.2.5 Diagrama de despliegue

Acompañado del diagrama de componentes, podemos dar una vista por nodos identificando los componentes que se conectan a través de la red, de esta forma se generan otros escenarios teniendo en cuenta los protocolos y los puertos por el que interactúa el software.

Figura 8. Diagrama de despliegue

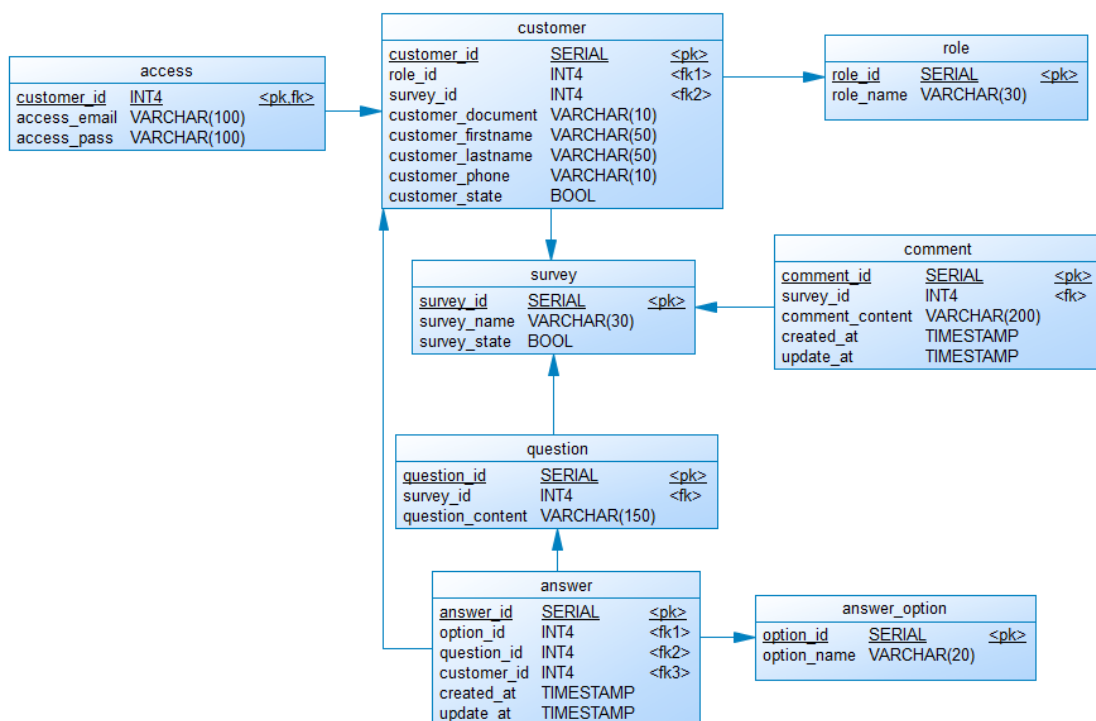


7.2.6 Base de datos

Teniendo en cuenta la fase de análisis, se determina 7 tablas para la base de datos, cada una corresponde a los elementos principales que almacenarán datos de valor. Estas tablas son las necesarias para el almacenamiento y tratamiento de datos. Para el diseño de la base de datos se utilizó la herramienta Power Designer, la cual nos ayuda a mapear de forma eficiente las tablas y relaciones que se establecen a partir del diseño de los diagramas y el análisis.

Cada tabla pertenece a una función específica que ayudara al almacenamiento de datos específicos según su contexto. En la fase de diseño tenemos en cuenta los tamaños, tipos de datos y relaciones que tendrá cada tabla, ya que se hizo el estudio que tendrá el esquema al principio del diseño, como apartado final tendremos el diagrama que representará todo el detalle de la base de datos para encuestas de satisfacción del SG-SST.

Figura 9. Diagrama de base de datos



Se llega a esta versión final de esquema ya que en versiones anteriores se insertaban tablas para el tratamiento de imágenes, sin embargo, al tratarse de una plataforma de encuestas y enlazada directamente al correo institucional, se determina que sería un tratamiento de datos adicional que no tendría ninguna funcionalidad dentro de la plataforma, por ello, se establece que las tablas presentadas en la **ilustración 7** son suficientes para un tratamiento de información correcto y anónimo.

Adicionalmente, con el objetivo de crear una plataforma escalable se establece una tabla “survey”, esto para poder crear encuestas diferentes en dado caso que se requiera elaborar diferentes encuestas.

Se validaron los campos con requerimientos específicos tales como datos únicos, campos que no pueden ser nulos o llaves primarias. Estos datos deben ser tratados por parte del backend y frontend para garantizar dichas propiedades.

7.2.7 Diseño Visual

En la fase del análisis se determinaron los módulos que podrán ser usados por cada rol, de esta forma conocemos cual será la carga visual que tendrá el usuario al ingresar a la plataforma, y de este modo, distribuir los elementos de forma ordenada e intuitiva para los clientes. Se hizo una selección de componentes que podrían ser compartidos, esto con el fin de disminuir el tiempo de desarrollo y efectuar reúso de código que ayude a no redundar entre componentes, a partir de esto se pudo determinar 3 pantallas principales:

- **Acceso:** la pantalla de acceso presentara un formulario con dos campos, uno para el ingreso del correo electrónico y otro para el ingreso de la contraseña asociada al correo. Este apartado no presenta registro ya que esta plataforma deberá ir vinculada a la base de datos donde almacena los correos institucionales de los trabajadores de la universidad santo tomas. Por ello, se opta por un diseño centralizado y minimalista, evitando la sobrecarga de información y las acciones puntuales en la plataforma.

Figura 10. Wireframe - Acceso

Titulo de plataforma

Unicamente correos institucionales

Email

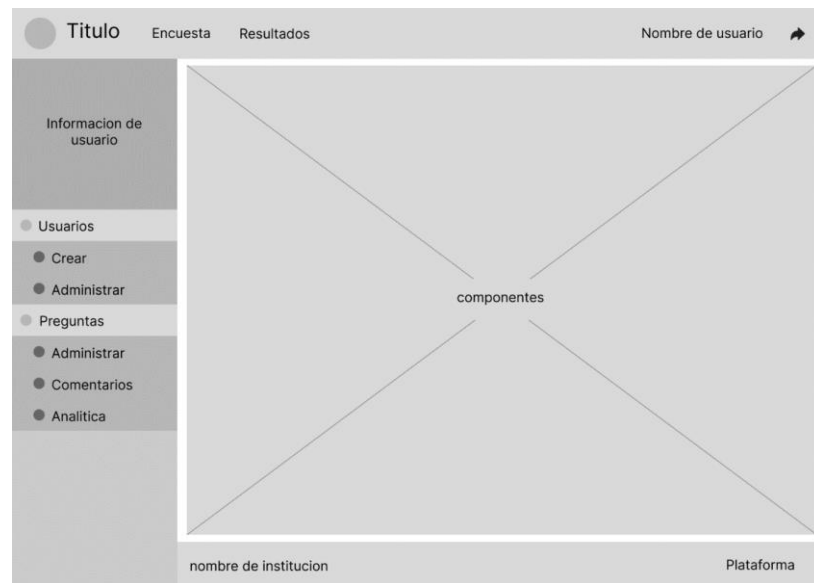
Contraseña

Iniciar sesion

- **Dashboard de administrador:** para llegar a esta pantalla se debe ser usuario administrador, el cual tendrá más funcionalidades indicadas en el menú lateral ubicado al lado izquierdo de la pantalla. Este menú mostrará las opciones de gestión que tendrá el administrador, donde en la parte central se hará el cambio de componentes según corresponda el enrutamiento de las opciones.

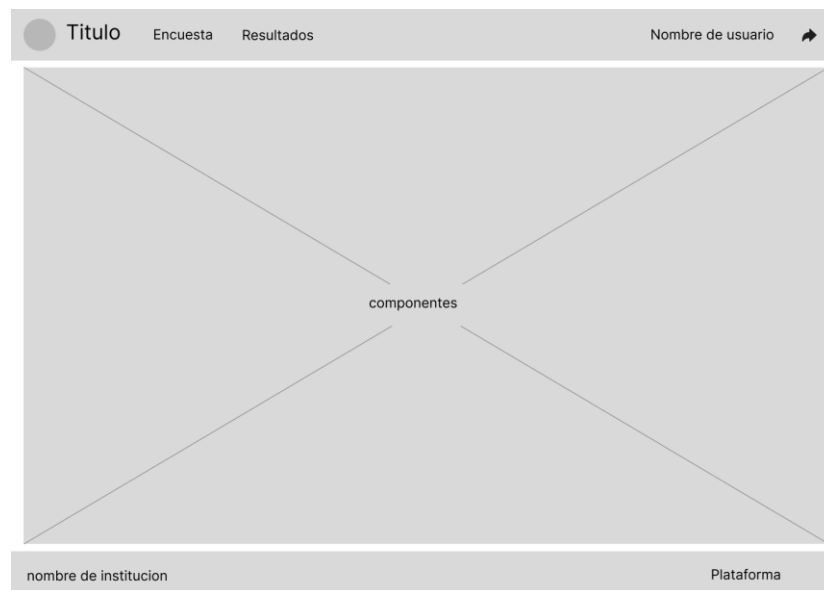
Adicionalmente, se muestra la información personal del administrador en la parte superior del menú evitando invadir mucho espacio de menú lateral y mostrando datos básicos del acceso.

Figura 11. Wireframe - Dashboard Administrador



- **Dashboard para participantes:** esta poseerá componentes compartidos con el dashboard de administrador, únicamente no mostrará el menú lateral con las funciones de usuarios administrador. La diferencia será únicamente las dimensiones de los componentes, la barra de navegación superior y el footer respectivamente.

Figura 12. Wireframe - Dashboard participantes



Como se puede evidenciar, estas son las 3 pantallas que en la fase de diseño se especifica que podrían cumplir con los objetivos, ya que identifica el acceso dependiendo del rol que posea el usuario y además, permite visualizar el contenido de los componentes de forma clara y sin perjudicar espacio para su usabilidad.

7.3 Codificación

A partir de las 2 primeras fases anteriormente contextualizadas, se empieza con el desarrollo del frontend y backend de la plataforma. El análisis y diseño dan guía para la codificación de módulos, componentes, aspectos visuales y creación del servidor, con estos datos y conociendo la necesidad de los requisitos, se establecen las tecnologías que se adecúan más al desarrollo de la plataforma de encuestas.

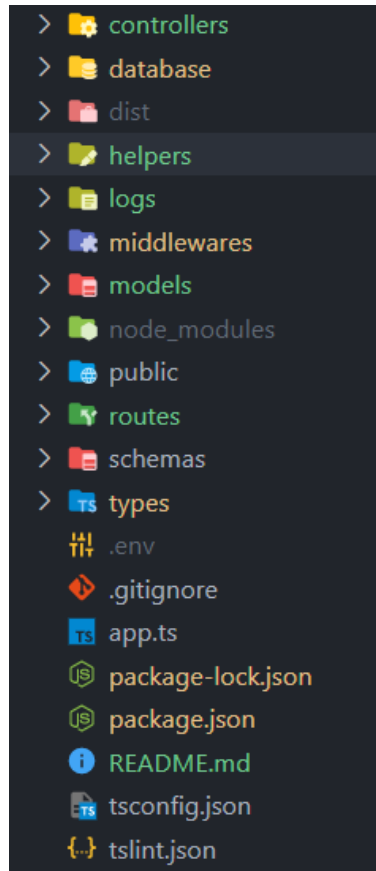
Conociendo las dos partes principales de desarrollo, se obtiene:

7.3.1 Backend:

El desarrollo del backend se planifico utilizando como base nodejs con el lenguaje de programación typescript que luego será transpilado para correr en con la sintaxis de javascript en el servidor y usando el framework Express teniendo en cuenta la construcción de API REST y la base de datos creada con PostgreSQL.

Se determina la estructura que maneja el proyecto teniendo en cuenta las funciones que va a cumplir cada complemento, de este modo se define el arquetipo y las necesidades que requiere el servicio [Figura 11].

Figura 13. Arquetipo backend

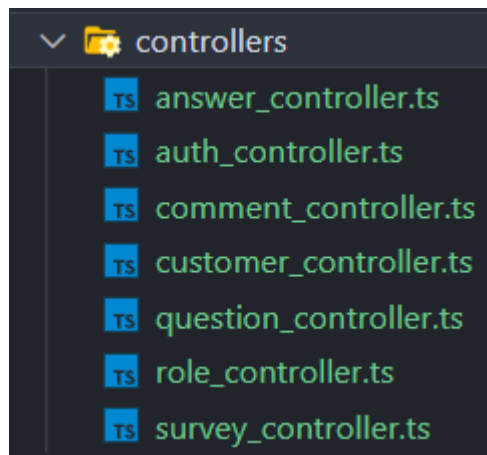


Para el desarrollo de los modelos y la interoperabilidad de los datos, se utiliza Sequelize ORM para este objetivo, ya que ofrece una forma sencilla de mapear datos, generar consultas, generar conexiones, etc.

A continuación, se dará una vista general a las carpetas para contextualizar de su función:

- **Controllers:** los controladores nos van a permitir generar los métodos que permitirán el tratamiento de los datos según corresponda el módulo. Estos métodos son las peticiones que van a llevar las respuestas al frontend cuando el usuario ejecute una acción. Según nuestros módulos vistos en la fase de análisis y diseño, se determinan los siguientes controladores:

Figura 14. Carpeta - controllers



Estos controladores nos van a permitir obtener y/o modificar datos de la base de datos, cada archivo .ts contiene crud completo para cada tabla correspondientemente, sin embargo, algunos módulos requirieron de controladores adicionales para un mejor tratamiento de datos.

- **Database:** en este fichero se puede encontrar el archivo de configuración para la conexión con la base de datos [Ilustración 14]. Para este fin también se hace uso de Sequelize y las credenciales especificadas al momento de crear la base de datos PostgreSQL.

Figura 15. Carpeta - Database

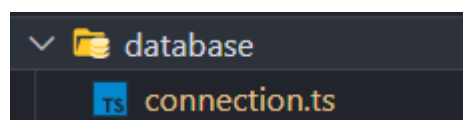


Figura 16. configuración de conexión

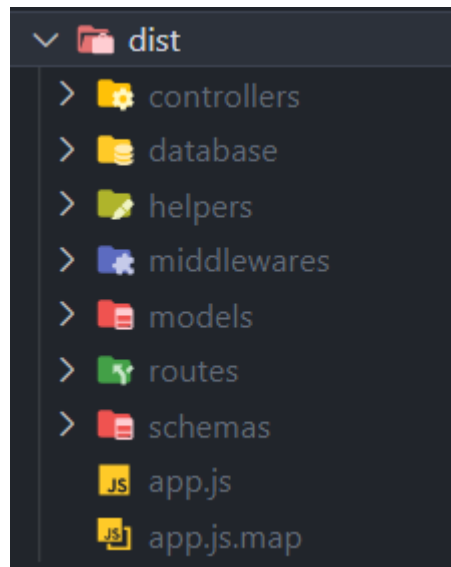
```
import { Sequelize } from 'sequelize';

const dataBase = new Sequelize('bd_sgsst', 'postgres', '123456', {
  host: 'localhost',
  dialect: 'postgres',
  define: {
    freezeTableName: true }
});

export default dataBase;
```

- **Dist:** en esta carpeta se transpila todo el código de typescript a javascript, de este modo al momento de ser desplegado en producción se puede llevar esta carpeta que será la que hará funcional el backend.

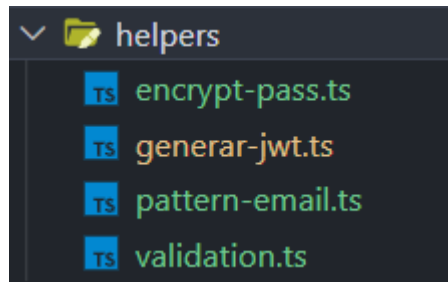
Figura 17. carpeta dist



Como se puede evidenciar, esta carpeta contiene los mismos ficheros creados externamente en el proyecto, la diferencia es el lenguaje sobre el que esta desplegando.

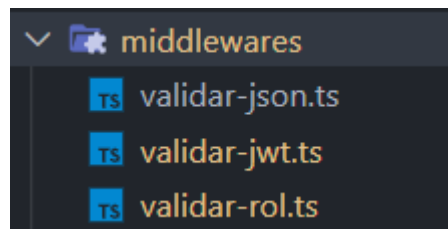
- **Helpers:** los helpers contribuyen a la ejecución de ciertas operaciones disponibles en todo el proyecto. En este caso, se utilizaron helpers para validaciones, la generación del JWT, algoritmo para encriptar contraseña y la validación del patrón del correo electrónico.

Figura 18. Carpeta helpers



- **Middlewares:** los middlewares nos van a permitir conectar diferentes aplicaciones para generar comunicación entre componentes. Estos nos sirven para generar validaciones desde el servicio para generar una mejor seguridad en el tratamiento de datos. En este caso se generan 3 middlewares que se usan para autenticación con JWT y validación de rol de usuario para que el frontend pueda ejercer acciones respecto a estas respuestas.

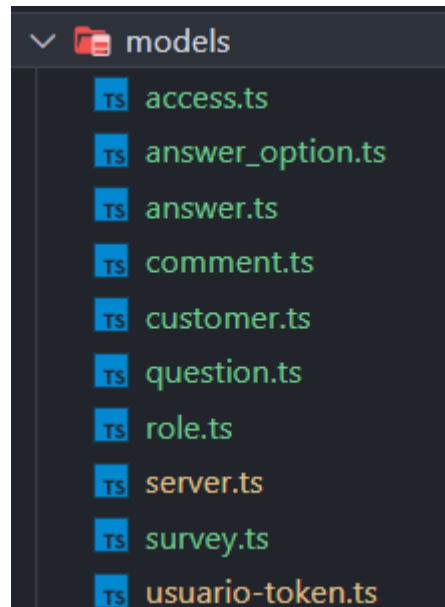
Figura 19. Carpeta middlewares



- **Models:** los modelos se elaboraron con sequelize, nos permite generar el mapeo de los datos entre el backend y la base de datos. Este mapeo tiene en cuenta todas las tablas, atributos,

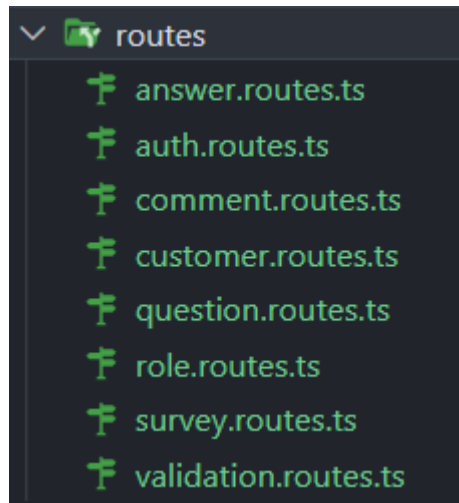
tipos de datos y relaciones que pueda tener la base de datos. De esta forma al momento de realizar peticiones los datos se mapean de forma más precisa al momento de generar respuestas.

Figura 20. Carpeta models



- **Routes:** las rutas son los paths que nos van a ayudar a llamar determinada petición desde el frontend, para ello se establece un archivo de rutas por controlador. Aquí mismo se especifica el tipo de petición, seguridad que requiere y método específico que se usara del controlador. Estas rutas se van a generar a partir del archivo **server.ts** en la carpeta “models” presentada en la **figura 18**.

Figura 21. Carpeta routes



A partir de todas estas funcionalidades se determina que el método de seguridad será API Key con JWT. Los json web token son unas credenciales únicas que se crearan aleatoriamente a partir de una petición de acceso. De esta forma el back recibirá un token que se genera durante la autenticación, si este token es válido el usuario podrá acceder sin problemas, de lo contrario, arrojará un error de autorización y no podrá acceder a determinada capacidad.

7.3.2 Frontend:

Esta es la capa visual con la que interactúan los usuarios, por ello, se debe generar unas vistas agradables e intuitivas para que el uso de esta plataforma sea lo más acertada posible. A continuación, veremos las pantallas diseñadas para dar cumplimiento a los requerimientos planteados en anteriores etapas, empezando por la pantalla de acceso para el ingreso a la plataforma:

Figura 22. Acceso frontend

The image shows a login interface for a system titled "DESEMPEÑO SG-SST". The interface is centered on a light blue background. It features a white rectangular box containing the login form. At the top of the box, the text "Ingrese unicamente con correo institucional" is displayed. Below this, there are two input fields: "Email" with an envelope icon and "Contraseña" with a lock icon. A dark grey button labeled "Iniciar Sesión" is positioned below the password field. The entire form is enclosed in a thin red border.

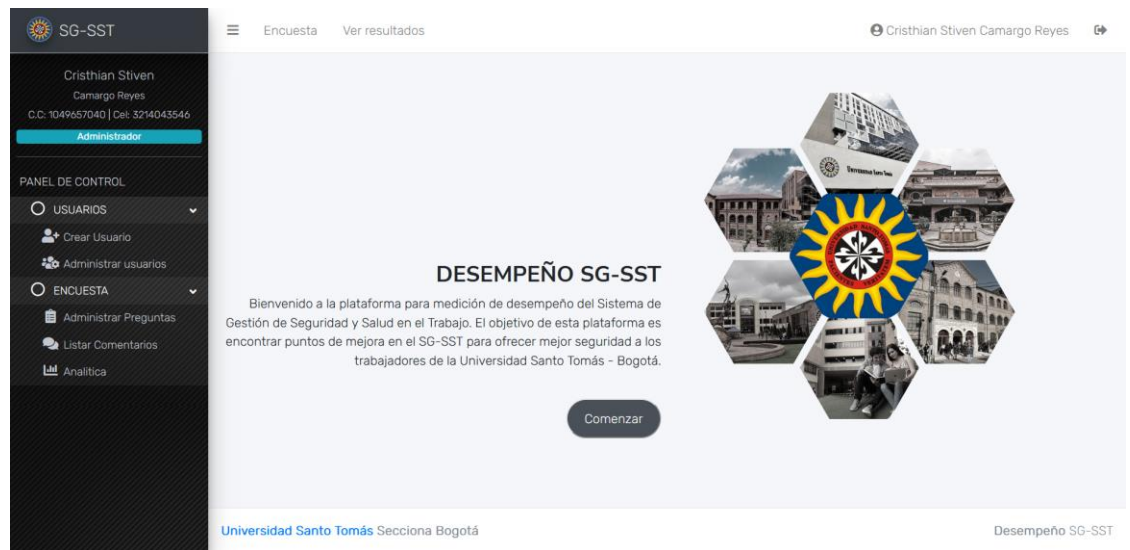
A partir del acceso se generan las validaciones del patrón del correo electrónico, campos requeridos y validez de la contraseña. Desde este momento se comienza con las validaciones de JWT para generar el token que permitirá el acceso a la plataforma.

Una vez el usuario intente acceder a la plataforma se ejecuta el proceso de validación del rol, generándose dos casos:

7.3.2.1 Módulo administrador

Este módulo será el exclusivo para los administradores de las encuestas, donde podrán visualizar los datos y ejecutar acciones de edición, borrado y creación de usuarios y preguntas.

Figura 23. Home administrador



Se presenta un home que será un componente compartido entre los dos roles, a partir de aquí el usuario podrá ingresar a los módulos de administración del menú lateral izquierdo. Estos modulos son:

- **Crear Usuario:** este apartado nos permite crear un usuario nuevo con sus datos personales y de plataforma. El formulario presenta validaciones relacionadas con campos únicos, verificación de contraseña, verificación de patrón de correo electrónico y campos nulos.

Figura 24. Creación de usuarios

SG-SST

Cristhian Stiven Camargo Reyes
C.C: 1049657040 | Cel: 3214043546
Administrador

PANEL DE CONTROL

- USUARIOS
 - Crear Usuario
 - Administrar usuarios
- ENCUESTA
 - Administrar Preguntas
 - Listar Comentarios
 - Analítica

Encuesta Ver resultados

Cristhian Stiven Camargo Reyes

Creación de usuarios

Datos personales

Nombres:

Apellidos:

Documento:

Telefono:

Datos institucionales

Email:

Encuesta: ☒

Contraseña:

Rol:

Confirmar contraseña:

- **Administrar usuarios:** este módulo me permite visualizar información sobre el nivel de satisfacción individual del usuario, estado de encuesta y la edición o borrado de usuario. Además, me hace un conteo en los widgets superiores sobre el numero de usuarios registrados y numero de usuarios con encuestas completadas o pendientes por responder.

Figura 25. Administrar usuarios

SG-SST

Cristhian Stiven Camargo Reyes
C.C: 1049657040 | Cel: 3214043546
Administrador

PANEL DE CONTROL

- USUARIOS
 - Crear Usuario
 - Administrar usuarios
- ENCUESTA
 - Administrar Preguntas
 - Listar Comentarios
 - Analítica

Encuesta Ver resultados

Cristhian Stiven Camargo Reyes

5
Numero de usuarios

3
Completadas

2
Pendientes

Filtrar:

ID	Nombre Completo	Nivel de satisfacción	Estado
1	Cristhian Stiven Camargo Reyes Administrador	Muy Satisfecho	Completada
2	Jose Alejandro Garcia Preciado Auxiliar	Algo Satisfecho	Completada
3	Juan Garcia Preciado Auxiliar	Sin respuestas	Pendiente
6	Daniel Sebastian Perez Hernandez Administrador	Sin respuestas	Pendiente
8	prueba correo Auxiliar	Insatisfecho	Completada

En el apartado de edición de usuarios no encontraremos todos los datos, pues no sería correcto que un usuario administrador tuviera el poder de generar contraseñas a su voluntad, de esta forma, solamente se especifica los campos que identifican al usuario.

Figura 26. Administrar usuarios

SG-SST

Cristhian Stiven
Camargo Reyes
C.C. 1049657040 | Cel: 3214043546
Administrador

PANEL DE CONTROL

- USUARIOS
 - Crear Usuario
 - Administrar usuarios
- ENCUESTA
 - Administrar Preguntas
 - Listar Comentarios
 - Analítica

Encuesta Ver resultados

Cristhian Stiven Camargo Reyes

5 Actualizar usuario

Datos personales

Nombres: Cristhian Stiven Apellidos: Camargo Reyes

Documento: C.C. 1049657040 Telefono: +57 3214043546

Datos institucionales

Email: cristhian@usantoto.edu.co Encuesta: Desempeño SG-SST Rol: Administrador

Cancelar Guardar Cambios

ID	Nombre	Estado	Acciones
1		Completada	Editar Borrar
2		Completada	Editar Borrar
3		Pendiente	Editar Borrar
6		Pendiente	Editar Borrar
8	prueba correo Auxiliar	Insatisfecho	Completada Editar Borrar

- **Administrar preguntas:** en esta pantalla podemos ver todas las preguntas que pertenecen a la encuesta, justo con sus acciones de crear, editar y borrar preguntas. Adicionalmente se puede observar el nivel de satisfacción que se presenta por pregunta y un conteo de respuestas por nivel de satisfacción en los widgets de la parte superior.

Figura 27. Administrar preguntas

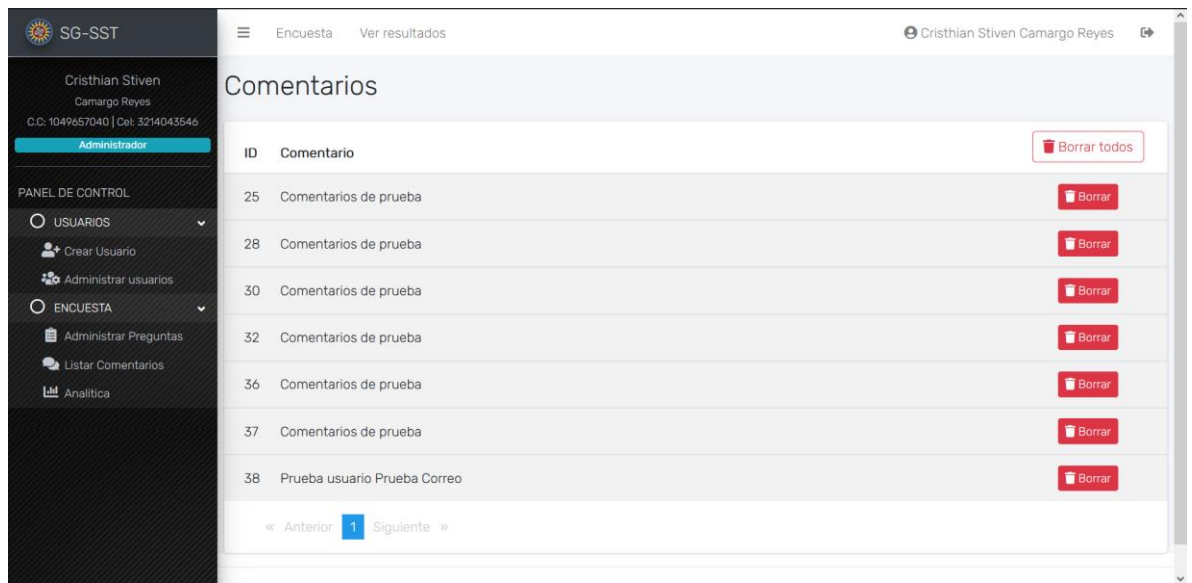
The screenshot displays the 'Administrar preguntas' (Manage questions) interface of the SG-SST system. The interface is divided into several sections:

- Header:** Includes the SG-SST logo, contact information (C.D. 1049657040 | Cel. 3214043546), and the role 'Administrador'. It also shows the user 'Cristhian Stiven Camargo Reyes'.
- Navigation Sidebar:** Contains a 'PANEL DE CONTROL' with options for 'USUARIOS' (Crear Usuario, Administrar usuarios) and 'ENCUESTA' (Administrar Preguntas, Listar Comentarios, Analitica).
- Survey Status Legend:** A row of five colored boxes representing satisfaction levels: 'Muy Satisfecho' (green), 'Satisfecho' (green), 'Algo Satisfecho' (yellow), 'Insatisfecho' (orange), and 'Muy Insatisfecho' (red).
- Form Area:** A text input field with an 'Agregar' (Add) button.
- Table:** A table listing survey questions with columns for 'ID', 'Pregunta', and 'Nivel de satisfacción'. It includes 'Editar' (Edit) and 'Borrar' (Delete) buttons for each question.
- Pagination:** A bottom section with 'Previous', '1', and 'Next' links.

ID	Pregunta	Nivel de satisfacción	Editar	Borrar
1	¿Todos reciben capacitación obligatoria en salud y seguridad?	Algo Satisfecho	Editar	Borrar
2	¿La Universidad ha liderado la participación de los trabajadores en la identificación eficaz de peligros y riesgos en el lugar de trabajo?	Algo Satisfecho	Editar	Borrar
10	¿Ha difundido la Universidad la política del sistema de gestión de la seguridad y salud en el trabajo a todos los niveles de la organización?	Satisfecho	Editar	Borrar

- **Listar comentarios:** en esta pantalla se pueden ver todos los comentarios generados por los usuarios que responden las encuestas, adicionalmente se genera un botón para eliminar todos los comentarios y de igual forma, un botón para eliminar preguntas individualmente. Se añade paginación para evitar gran listado de usuarios y que el scroll sea muy tedioso de utilizar.

Figura 28. Listar comentarios



The screenshot shows a web application interface for managing comments. On the left is a dark sidebar with the logo 'SG-SST' and user information for 'Cristhian Stiven Camargo Reyes' (C.C: 1049657040 | Cel: 3214043546) with an 'Administrador' role. Below this is a 'PANEL DE CONTROL' with two sections: 'USUARIOS' (containing 'Crear Usuario' and 'Administrar usuarios') and 'ENCUESTA' (containing 'Administrar Preguntas', 'Listar Comentarios', and 'Analítica'). The top header has a menu icon, 'Encuesta', 'Ver resultados', and the user's name. The main content area is titled 'Comentarios' and contains a table with columns 'ID' and 'Comentario'. There are 8 rows of data, all labeled 'Comentarios de prueba' except for the last one, 'Prueba usuario Prueba Correo'. Each row has a red 'Borrar' button. A 'Borrar todos' button is in the top right of the table area. At the bottom of the table is a pagination bar showing '« Anterior 1 Siguiente »'.

ID	Comentario	
25	Comentarios de prueba	Borrar
28	Comentarios de prueba	Borrar
30	Comentarios de prueba	Borrar
32	Comentarios de prueba	Borrar
36	Comentarios de prueba	Borrar
37	Comentarios de prueba	Borrar
38	Prueba usuario Prueba Correo	Borrar

- **Analítica:** los cálculos que se realizan en cuanto el usuario envía una encuesta se ven reflejados en este apartado, muestra gráficamente porcentajes y numero de datos por nivel de satisfacción. Se establece estos gráficos porque se consideran que son fáciles de leer y no contiene gran cantidad de carga visual para el usuario. Adicionalmente se añade un botón para reiniciar las estadísticas, es decir, se eliminaran todas las respuestas existentes y cambiara el estado de todos los usuarios para que puedan volver a responder la encuesta.

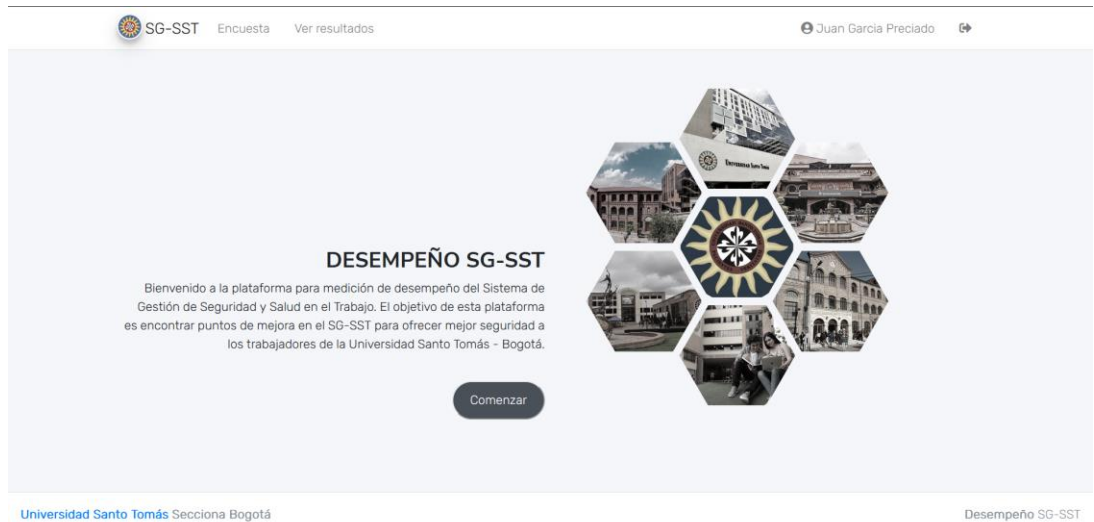
Figura 29. Analítica



7.3.2.2 Módulo participante

Este módulo es para las personas que únicamente tendrán permisos para resolver encuestas y aportar por medio de sus resultados para la medición del nivel de satisfacción. Se establece un apartado home para dar la bienvenida al usuario donde se da una breve descripción del objetivo de la plataforma.

Figura 30. Home participante



Desde este momento podemos ver el cambio visual que tiene la plataforma, pues ya no posee el menú lateral que cuenta el administrador. A partir de aquí el usuario puede acceder a dos módulos únicamente:

- **Encuesta:** se puede acceder por medio del botón del home ó el botón de encuesta en la barra de navegación. Este apartado contiene las preguntas con sus respectivas opciones. Se opta por dejar emoticones para generar un ambiente más descriptivo e intuitivo al momento de responder las preguntas. Adicionalmente existe un campo para comentarios opcional y anónimos, es decir, el usuario podrá generar un comentario a voluntad y su identidad estará protegida. También encontraremos un botón para que el usuario pueda guiarse gráficamente sobre las opciones.

Figura 31. Encuesta

#	Pregunta					
1	¿Todos reciben capacitación obligatoria en salud y seguridad?	☐	☐	☐	☐	☐
2	¿La Universidad ha liderado la participación de los trabajadores en la identificación eficaz de peligros y riesgos en el lugar de trabajo?	☐	☐	☐	☐	☐
10	¿Ha difundido la Universidad la política del sistema de gestión de la seguridad y salud en el trabajo a todos los niveles de la organización?	☐	☐	☐	☐	☐

Comentarios (Opcional)

Enviar

Quando el usuario envíe las respuestas, la plataforma lo redireccionará al home con el objetivo de dar por finalizada la encuesta, de tal forma que si el usuario vuelve a acceder a la encuesta aparecerá un mensaje que le indicará que no puede responder la encuesta nuevamente. Su estado habrá sido alterado internamente hasta que el administrador lo desee cambiar.

Figura 32. Encuesta respondida



- **Ver resultados:** el usuario podrá ver el nivel de satisfacción general y personal sin gran detalle, con el fin de que esta analítica sea privada y exclusiva para el administrador. Adicionalmente encontrara el listado de las preguntas con su pregunta seleccionada.

Figura 33. Ver resultados



Sin embargo, si el usuario aún no ha respondido una encuesta, la pantalla mostrará un mensaje indicándole que aun no tiene resultados debido a su estado actual, primero debe resolver la encuesta para poder habilitar la pantalla de “ver resultados”.

Figura 34. Ver resultados – inactivo

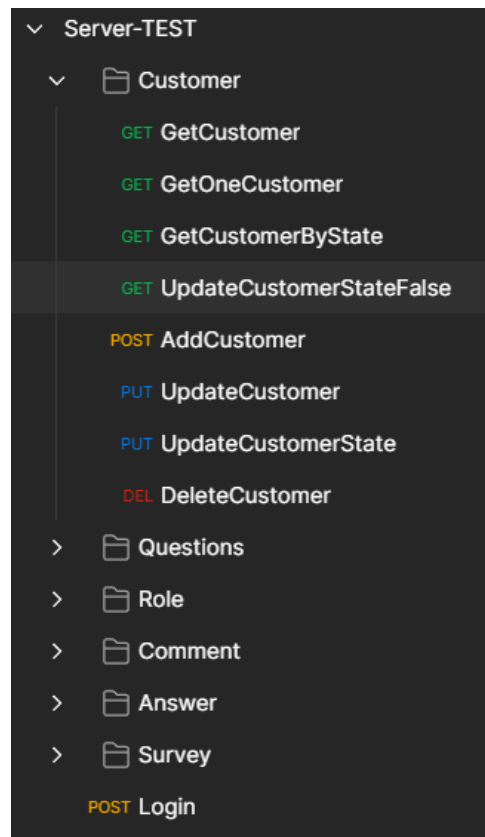


7.4 Pruebas

Para comprobar que un software es funcional se deben ejecutar una serie de pruebas, para ello hay diversas formas de hacerlo, sin embargo, en este proyecto se opta por hacer pruebas unitarias y modulares comprobando escenarios por parte del backend y ejecutando pruebas manuales en el frontend.

Por parte del backend se generan pruebas locales de postman, generando una colección para que se pueda probar desde otro equipo si así se desea. A partir de esta colección se puede generar pruebas para cada una de las operaciones por módulo, ya se encuentran mapeadas y listas para ser probadas manualmente y localmente.

Figura 35. Colección de pruebas en postman



Respecto al frontend, se hicieron pruebas modulares verificando el funcionamiento individual de cada componente. Se comprobó la seguridad de las rutas y la veracidad de los resultados ejecutando los cálculos con diferentes métodos para ver la varianza y margen de error de los resultados.

8 CONCLUSIONES

- El desarrollo de la plataforma de encuestas para medir el desempeño del Sistema de Gestión de Seguridad y Salud en el Trabajo fue exitoso. Se ejecuto el desarrollo teniendo en cuenta las 4 fases planteadas en la propuesta del proyecto, dando como resultado el desarrollo

finalizado de la plataforma de desempeño SG-SST, con las tecnologías especificadas y los estándares cumplidos.

- Las tecnologías trabajadas en este proyecto son bastante compatibles entre sí, permitiendo dar solución oportuna a los inconvenientes o percances ocurridos durante el desarrollo, debido a que todas las librerías, entornos, programas y herramientas cuentan con soporte y documentación actualizada para evitar retrasos en el desarrollo. Cada tecnología cuenta con una integración sencilla debido al gestor de paquetes de node (NPM).
- Se tomo en cuenta el proceso de levantamiento de requerimientos para comenzar con el análisis y diseño de la plataforma con el objetivo de cumplir el modelo de desarrollo iterativo e incremental. En la fase de desarrollo se determina que este modelo de desarrollo en compañía de un buen levantamiento de requisitos puede generar entregas de valor continuas, pues el modelo permite realizar análisis del avance para poder cumplir a cabalidad todas las necesidades que requiere el software.

9 REFERENCIAS

[1]"Sistema de Gestión de Seguridad y Salud en el Trabajo - Ministerio del trabajo", *Mintrabajo.gov.co*, 2021. [Online]. Available: <https://www.mintrabajo.gov.co/relaciones-laborales/riesgos-laborales/sistema-de-gestion-de-seguridad-y-salud-en-el-trabajo>.

[2]"Qué es Frontend y Backend: diferencias y características - Platzi", *platzi.com*, 2017. [Online]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/>

[3]"¿Qué es HTML? Explicación de los fundamentos del Lenguaje de marcado de hipertexto", *hostinger.co*, 2022. [Online]. Available: <https://www.hostinger.co/tutoriales/que-es-html>

[4]"¿Qué es CSS?", *hostinger.co*, 2022. [Online]. Available: <https://www.hostinger.co/tutoriales/que-es-css>

[5]"¿Qué es Angular y para qué sirve?", *hiberus.com*, 2021. [Online]. Available: <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>

[6]"¿Qué es Angular CLI?", *ediciones-eni.com*, 2022. [Online]. Available: <https://www.ediciones-eni.com/open/mediabook.aspx?idR=59febeb081a24f089644d9223202988f>

[7]"¿Qué es JavaScript?", *aws.amazon.com*, 2022. [Online]. Available: <https://aws.amazon.com/es/what-is/javascript/>

[8]"¿Qué es Express.js? Todo lo que Debes Saber", *kinsta.com*, 2022. [Online]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-express/>

[9]"Sequelize v6", *sequelize.org*, 2022. [Online]. Available: <https://sequelize.org/docs/v6/>

[10]"Lenguajes de Programación", *programas.cuaed.unam.mx*, (s.f). [Online]. Available: https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html#contenido

[11]"¿Qué es una base de datos?", *oracle.com*, (s.f). [Online]. Available:
<https://www.oracle.com/co/database/what-is-database/>

[12]"Introduction to JSON Web Tokens", *jwt.io*, (s.f). [Online]. Available:
<https://jwt.io/introduction>

10 ANEXOS

- Anexo 1. Especificación de requisitos

Enlace: [Google Drive – Especificación de requisitos](#)

- Anexo 2. Colección de postman

Enlace: [Google Drive – Colección de postman](#)

- Anexo 3. Propuesta inicial de proyecto de grado

Enlace: [Google Drive – Propuesta Desempeño SGSST](#)

- Anexo 4: Repositorio del frontend

Enlace: <https://github.com/CristhianCamargo/frontendSGSST.git>

- Anexo 5: Repositorio del backend

Enlace: <https://github.com/CristhianCamargo/backendSGSST.git>

- Anexo 6: DDL_SGSST inicial

Enlace: [Google Drive – DDL_SGSST.sql](#)

- Anexo 7: DML_SGSST inicial

Enlace: [Google Drive – DML_SGSST.sql](#)

- Anexo 8. Backup de base de datos con datos de prueba

Enlace: [Google Drive – BU_SGSSTv1](#)

- Anexo 9. Plataforma de encuestas SG-SST desplegada

Enlace: <https://ecuestas-sgst.web.app>

-