



ENCUESTADOR LABORAL BASADO EN GOHISALO

PROPONENTE

Juan Camilo Rodríguez Salamanca

CC: 1115918612

CE: 2251138

DIRECTOR

Carlos Andrés Guerrero

FACULTAD DE INGENIERÍA DE SISTEMAS

2023

CONTENIDO

<u>1. FICHA TÉCNICA DEL PROYECTO.....</u>	<u>4</u>
<u>2. PLANTEAMIENTO DEL PROBLEMA</u>	<u>5</u>
<u>3. JUSTIFICACIÓN.....</u>	<u>7</u>
<u>4. OBJETIVOS.....</u>	<u>8</u>
4.1. OBJETIVO GENERAL	8
4.2. OBJETIVOS ESPECÍFICOS.....	8
<u>5. MARCO TEÓRICO</u>	<u>9</u>
5.1. CVT-GOHISALO.....	9
5.2. TYPESCRIPT	9
5.3. NODE.JS.....	9
5.4. NPM	10
5.5. ANGULAR	10
5.6. EXPRESS.....	10
5.7. JSON WEB TOKEN (JWT)	10
5.8. VISUAL STUDIO CODE	11
5.9. DATAGRIP	11
5.10. POSTGRESQL	11
5.11. POSTMAN	11
5.12. GIT-GITHUB.....	12
<u>6. METODOLOGÍA</u>	<u>13</u>
6.1. RECOPIACIÓN Y ANÁLISIS DE REQUISITOS.....	13
6.2. PLANIFICACIÓN Y VIABILIDAD.....	13
6.3. DISEÑO	13
6.4. DESARROLLO	13
6.5. IMPLEMENTACIÓN	13

7. <u>DESARROLLO DEL PROYECTO</u>	14
7.1. FASE DE RECOPIACIÓN Y ANÁLISIS DE REQUISITOS	14
7.2. FASE DE PLANIFICACIÓN Y VIABILIDAD	14
■ VISTA LÓGICA:.....	14
■ VISTA DE PROCESOS:.....	15
■ VISTA DE DESPLIEGUE:.....	17
■ VISTA DE FÍSICA:.....	19
■ "+1" VISTA DE ESCENARIOS:.....	20
7.3. FASE DE DISEÑO.....	21
■ DISEÑO DE LA BASE DE DATOS.....	21
■ DISEÑO PRELIMINAR.....	23
7.4. FASE DE DESARROLLO	24
DESARROLLO DEL BACKEND	24
■ DESARROLLO DEL FRONTEND.....	29
7.5. FASE DE IMPLEMENTACIÓN.....	40
■ BASE DE DATOS	40
■ BACKEND	42
■ FRONTEND	44
8. <u>LISTAS DE CHEQUEO</u>	47
9. <u>CONCLUSIONES Y ANÁLISIS DE RESULTADOS</u>	48
10. <u>REFERENCIAS</u>	49
11. <u>ANEXOS</u>	51

1. FICHA TÉCNICA DEL PROYECTO

Título	Encuestador Laboral Basado en Gohisalo
Nombre Estudiante	Juan Camilo Rodríguez Salamanca
Documento estudiante	1115918612
Correo electrónico	juan.rodriguez@usantoto.edu.co
Director	Carlos Andrés Guerrero
Lugar de ejecución del proyecto	Tunja-Boyacá
Duración	14 meses
Costo	\$5.580.000
Palabras claves	Encuestas, usuarios, procesos, registro, seguimiento



Firma del autor
Nombre autor

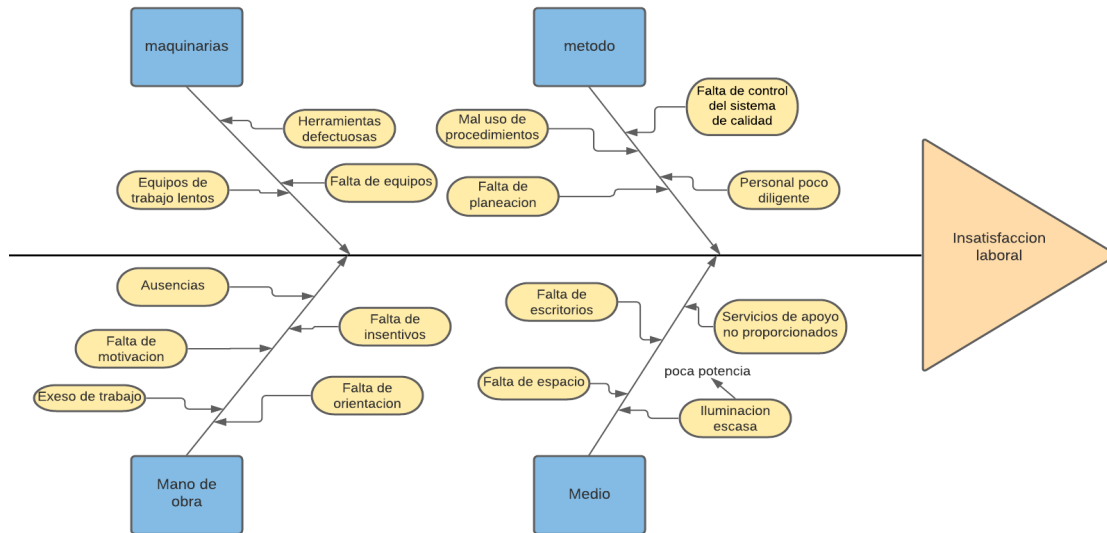
Firma del director
Nombre director del proyecto

2. PLANTEAMIENTO DEL PROBLEMA

La calidad de vida en el trabajo es un factor fundamental en la actualidad para cualquier empresa, puesto que en la mayoría de los casos la calidad de vida y condición que tengan los trabajadores dentro de una institución se refleja directamente en el resultado de los procesos. La economía mundial ha cambiado mucho debido a la globalización, donde se han dado reformas al sector laboral, provocando un impacto a la calidad de vida en el trabajo de las personas, esto debido a que las formas de empleo actuales han cambiado considerablemente; además crece la inestabilidad en los puestos de trabajo y una gran cantidad de trabajadores queda por fuera del sistema de protección previsional y de salud. Lo mencionado anteriormente, genera en los empleados sensación de inestabilidad, inseguridad y vulnerabilidad, por lo que las empresas se ven en la obligación de preocuparse por los trabajadores con el fin de mejorar su bienestar y promover una mejora en el ambiente laboral.

A continuación, se ha elaborado un diagrama Ishikawa [18] para comprender mejor cómo esta problemática afecta a las instituciones.

Figura 1. Diagrama de Ishikawa



Según la Figura 1, este es un problema que puede estar presente en instituciones de educación superior, como lo pueden ser la Universidad Santo Tomás Seccional Tunja, por lo tanto, es importante tomar medidas para mitigar estas problemáticas.

En este proyecto se propuso un desarrollo tecnológico que permitiera evaluar de manera sencilla pero eficaz la calidad de vida de los docentes y administrativos de la institución utilizando el cuestionario Gohisalo que es una herramienta creada por un grupo de investigadores que nos permite medir la calidad de vida en el trabajo a partir de encuestas tipo Likert a los trabajadores.

3. JUSTIFICACIÓN

Debido a la importancia que tiene la calidad de vida laboral para los docentes y trabajadores de la Universidad Santo Tomás Seccional Tunja, en este caso la Facultad de Ingeniería de Sistemas, es importante llevar un control del recurso humano mediante encuestas laborales, teniendo en cuenta factores como el bienestar logrado a través del trabajo, desarrollo personal, administración del tiempo libre, entre otros. Dicho esto, las mejoras en el ambiente laboral y sus métodos de ejecución son indispensables para que la institución cuente con mejores condiciones laborales.

Como se conoce estas encuestas laborales se hacen de manera manual, por lo que realizar una plataforma web, permitirá que el acceso a las encuestas sea rápido y eficiente para los empleados, puesto que no tienen que estar presencialmente, sino que se pueden realizar de manera remota, evitando así procesos engorrosos para el personal.

Cabe resaltar que este desarrollo tecnológico tendrá un enfoque educativo donde se utilizarán las herramientas y conocimientos adquiridos a lo largo de la carrera, utilizando también el cuestionario CVT-GOHISALO que se implementará dentro de la aplicación.

4. OBJETIVOS

4.1. Objetivo General

Construir una herramienta web que permita recopilar información sobre el ambiente laboral de la Facultad de Ingeniería de Sistemas de la Universidad Santo Tomás, mediante la aplicación de encuestas basadas en el cuestionario CVT-GOHISALO, utilizando herramientas de código abierto y métodos ágiles para su desarrollo.

4.2. Objetivos específicos

Tabla 1. Objetivos específicos

Nro.	Objetivos específicos
1	Definir la especificación de requerimientos con base en el cuestionario CVT-GOHISALO, para definir los requisitos funcionales y no funcionales de la herramienta web.
2	Diseñar la aplicación web utilizando el modelo 4+1 propuesto por Kruchten, para especificar las vistas: lógica, procesos, física y despliegue.
3	Implementar una herramienta web que permita realizar encuestas en el marco del cuestionario CTV-GOHISALO, incluyendo preguntas de tipo Likert en sus 7 dimensiones. Utilizando el framework angular para el desarrollo de las vistas y node express para el desarrollo del backend y PostgreSQL para el motor de base de datos.
4	Verificar la funcionalidad de herramienta web, mediante la aplicación de listas de chequeo definidas en la especificación de requisitos, para comprobar la implementación de todas las funcionalidades definidas.

5. MARCO TEÓRICO

5.1. CVT-Gohisalo

El cuestionario GOHISALO es un instrumento para medir la calidad de vida en el trabajo, fue creado principalmente por Raquel González Baltazar, doctora de ciencias de la salud y el trabajo médico de profesión. Profesora investigadora de la universidad de Guadalajara Jalisco, México, entre otros [4] . Este cuestionario está constituido por un compilado de siete dimensiones que permiten abarcar de una manera más efectiva la evaluación de la calidad de vida en el trabajo. A continuación, se mencionan las dimensiones:

- Soporte institucional para el trabajo
- Seguridad en el trabajo
- Integración al puesto de trabajo
- Satisfacción por el trabajo
- Bienestar logrado a través del trabajo
- Desarrollo personal del trabajador
- Administración del tiempo libre

5.2. TypeScript

TypeScript es un lenguaje tipado de programación basado en JavaScript creado por Microsoft. TypeScript agrega sintaxis adicional a JavaScript para admitir una integración más estrecha con su editor. Detecta errores temprano en su editor. Cabe resaltar que el código TypeScript se convierte a JavaScript, que se ejecuta en cualquier lugar donde se ejecute JavaScript: en un navegador, en Node.js o Deno y en sus aplicaciones. Además, TypeScript entiende JavaScript y utiliza la inferencia de tipos para brindarle excelentes herramientas sin código adicional.[6].

5.3. Node.js

Es un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. Las redes basadas en hilos son relativamente ineficientes y muy difíciles de usar. Además, los usuarios de Node.js están libres de preocuparse por el bloqueo del proceso, ya que no existe. Casi ninguna función

en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea lo cual permite gestionar varias peticiones a la vez. Por ello, es muy propicio desarrollar sistemas escalables en Node.js. [7]

5.4. NPM

Es el administrador de paquetes de nodejs. Fue creado en 2009 como un proyecto de código abierto para ayudar a los desarrolladores de JavaScript a compartir fácilmente módulos de código empaquetados. El registro de npm es una colección pública de paquetes de código fuente abierto para Node.js, aplicaciones web front-end, aplicaciones móviles, robots, enrutadores y muchas otras necesidades de la comunidad de JavaScript.[5]

5.5. Angular

Angular (a menudo denominado Angular 2+ o Angular 2) es un marco de aplicación web TypeScript de código abierto mantenido por Google para crear y mantener aplicaciones web de una sola página. Su objetivo es mejorar las aplicaciones basadas en navegador con la funcionalidad Model View Controller (MVC) para facilitar el desarrollo y las pruebas. La biblioteca lee HTML que contiene atributos de etiqueta personalizados adicionales, luego sigue las directivas de atributos personalizados y vincula los fragmentos de entrada o salida de la página a modelos representados por variables estándar de JavaScript. Angular se basa en clases de tipo "componente" cuyas propiedades se utilizan para enlazar datos. En estas clases tenemos propiedades (variables) y métodos (funciones a llamar). Angular es una evolución de AngularJS, aunque incompatible con su predecesor.[8]

5.6. Express

Una infraestructura web rápida, minimalista y flexible para las aplicaciones Node.js. En general, se prefiere "Express" a "Express.js", aunque esta también se acepta.[15]

5.7. 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. Aunque los JWT se pueden cifrar para proporcionar también

confidencialidad entre las partes, nos centraremos en los tokens firmados. Los tokens firmados pueden verificar la integridad de los reclamos contenidos en ellos, mientras que los tokens encriptados ocultan esos reclamos de otras partes. Cuando los tokens se firman utilizando pares de claves pública/privada, la firma también certifica que solo la parte que posee la clave privada es la que la firmó.[9]

5.8. Visual Studio Code

Visual Studio Code es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar las herramientas que un desarrollador necesita para una rápida creación y depuración de código.[10]

5.9. DataGrip

DataGrip es un entorno de gestión de bases de datos para desarrolladores. Está diseñado para consultar, crear y administrar bases de datos. Las bases de datos pueden funcionar localmente, en un servidor o en la nube. También permite generar diagramas UML que muestran gráficamente la estructura de la base de datos y las relaciones entre sus tablas.[11]

5.10. PostgreSQL

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES de la Universidad de California en Berkeley y cuentan con más de 35 años de desarrollo activo en la plataforma central.[12]

5.11. Postman

Postman es una plataforma API para construir y usar API. Postman puede almacenar y administrar especificaciones de API, documentación, recetas de flujo

de trabajo, casos de prueba y resultados, métricas y todo lo demás relacionado con las API.[13]

5.12. Git-Github

Git es un sistema de control de versiones distribuido, de código abierto. Está pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. GitHub está basada en el sistema de control de versiones distribuida de Git, por lo que se puede contar con sus funciones y herramientas, aunque GitHub ofrece varias opciones adicionales y su interfaz es mucho más fácil de manejar.[14]

6. METODOLOGÍA

Para el desarrollo web de esta aplicación se tuvo en cuenta una metodología “agile” que es una de las más utilizadas para desarrollos web por su agilidad y eficacia en la ejecución de los procesos esta metodología nos permite aplicar un plan de calidad óptimo de servicio y nos permite garantizar su efectiva implementación y mejora continua durante el proceso de desarrollo, durante el cual se realizarán las siguientes fases: [17].

6.1. Recopilación y análisis de requisitos

En esta fase se dará un enfoque frente a los alcances finales del proyecto, como lo pueden ser metas funcionales y no funcionales que se pueden presentar durante su desarrollo.

6.2. Planificación y viabilidad

En este apartado se incluye todo lo que debe diseñarse y desarrollarse durante el desarrollo del proyecto, por lo cual en esta fase utilizaremos la arquitectura de modelos de vista 4+1 propuesto por P. Kruchten.

6.3. Diseño

Se tendrá en cuenta el entorno donde se lleva a cabo el proyecto para definir colores e interacción con el usuario. En esta etapa también se llevará a cabo el diseño de la base de datos para definir un modelo de entidad relación.

6.4. Desarrollo

Una vez aprobados el diseño y la estructura, procederemos a la maquetación y programación del sitio e introduciremos todo el contenido. Para su codificación se hará uso de los frameworks, Angular para el desarrollo del frontend y nodeJS + Express para el desarrollo del backend.

6.5. Implementación

Desplegar el proyecto en un servidor de aplicaciones en la nube para ser consumido por el usuario final.

7. DESARROLLO DEL PROYECTO

7.1. Fase de Recopilación y análisis de requisitos

El desarrollo de la fase de análisis de requisitos estuvo compuesto por una serie de actividades de validación y aprobación de requerimientos con el objetivo de comprender los dominios de negocio que debía cumplir el proyecto. Estas actividades se pueden evidenciar en el Anexo 1.

Tabla 2 Fase 1. Recopilación y Análisis de Requisitos

Tipo	Elementos
Entradas	<ul style="list-style-type: none">● Narrativa del Proyecto● Descripción del Proyecto
Instrumentos y/o herramientas	<ul style="list-style-type: none">● Computador● Conexión a internet● Navegador Web● Google Docs
Salidas	Anexo 1. Documento de Requisitos del Sistema
Tiempo	4 semanas

Fuente: Autor

7.2. Fase de Planificación y viabilidad

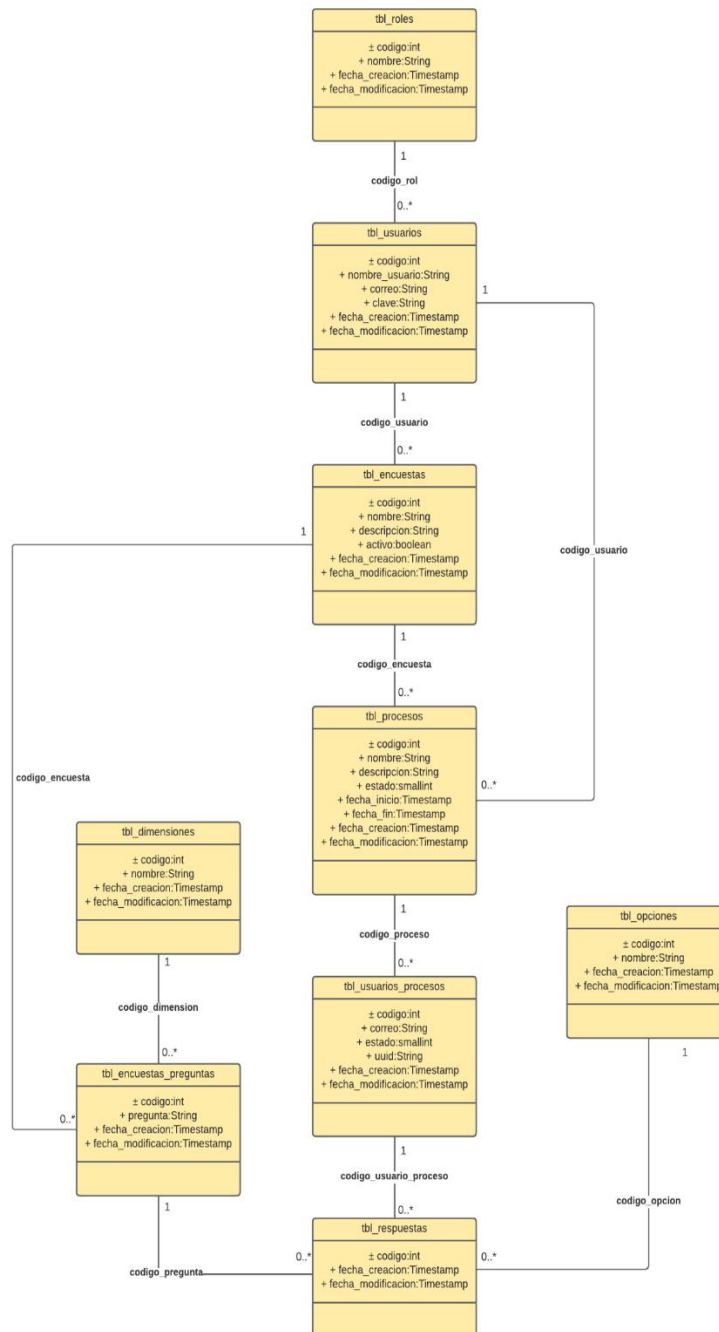
Para hacer una buena planificación y viabilidad fue necesario la implementación de una arquitectura para poder tener una vista más clara de los diferentes aspectos del proyecto, por lo que se ha seleccionado la arquitectura de modelos de vista 4+1 propuesto por P. Kruchten [16], el cual consiste en que el software se debe documentar por medio de cuatro vistas bien diferenciadas, además estas cuatro vistas se deben relacionar entre sí con una vista adicional que se denomina vista+1. Las cuatro vistas que definió Kruchten son: vista lógica, vista de procesos, vista de despliegue, vista física, y la vista +1 que tiene la función de relacionar las cuatro mencionadas, esta se denomina también como vista de escenarios.

A continuación, mostraremos el diseño de las vistas para el proyecto.

■ Vista Lógica:

Representa la funcionalidad que el sistema proporciona a los usuarios. Es decir, se ha de presentar lo que el sistema debe hacer y las funciones y servicios que ofrece.

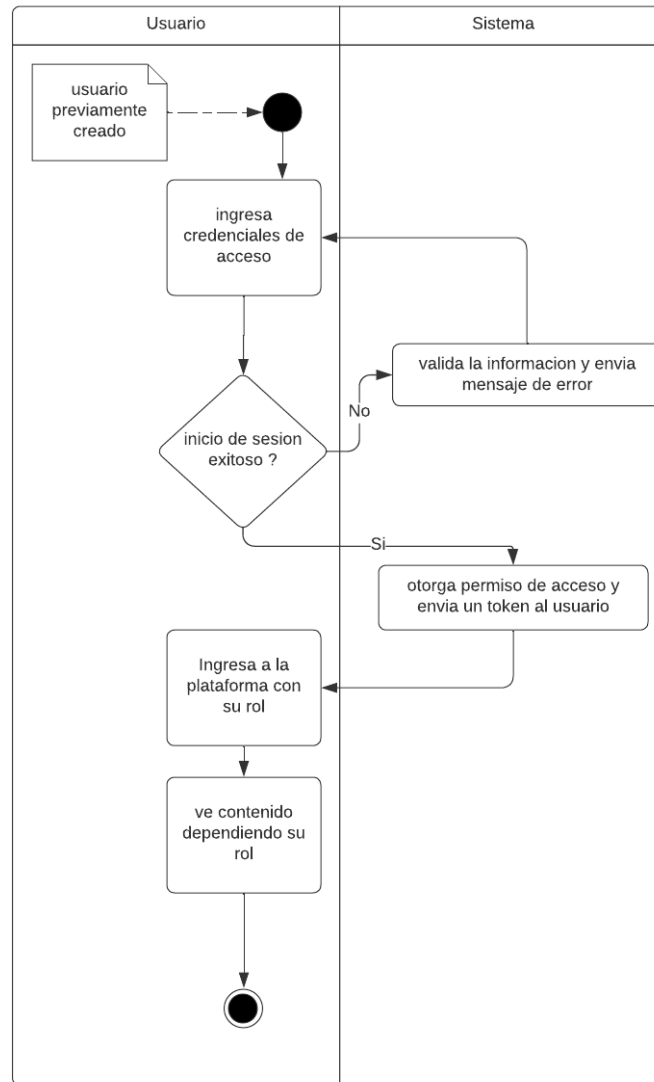
Figura 2. Diagrama de clases



■ **Vista de procesos:**

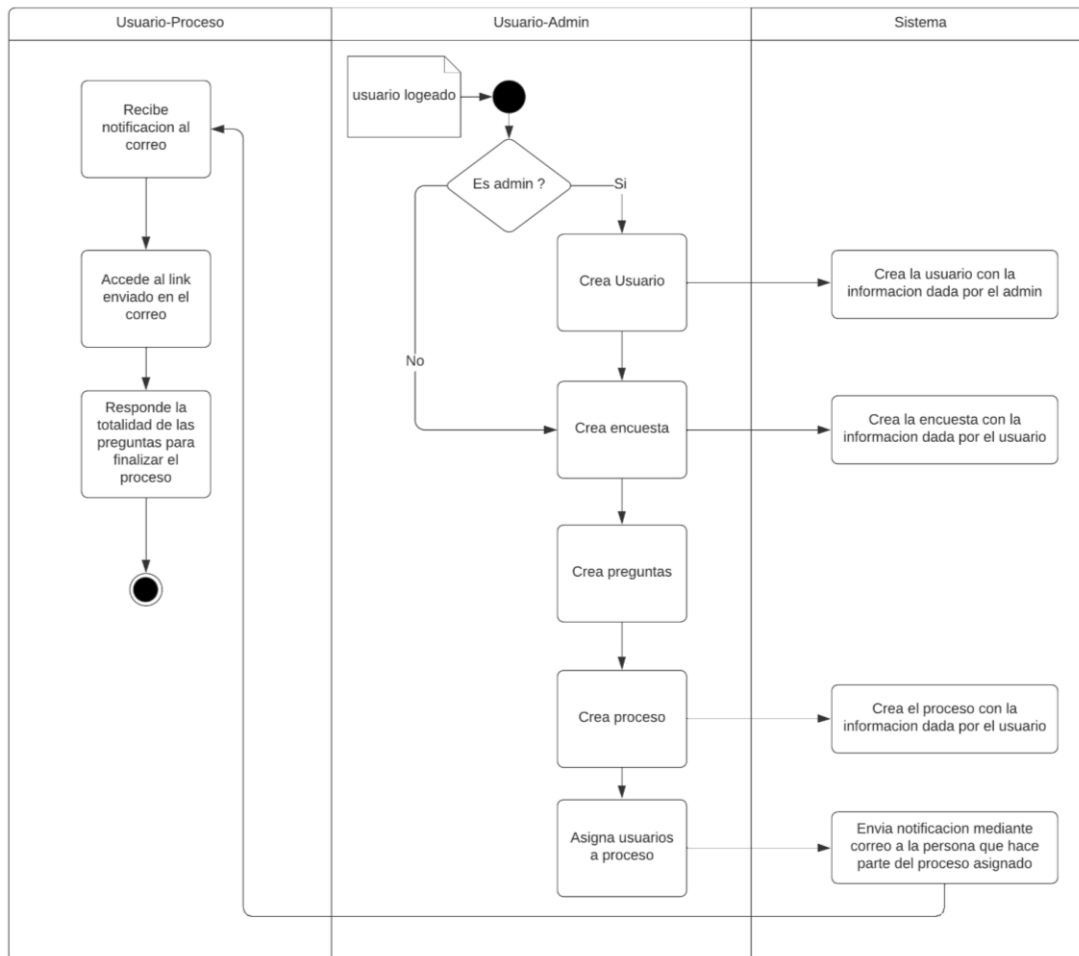
En esta vista se muestran los procesos que existen en el sistema y la forma en la que se comunican entre sí.

Figura 3. Diagrama de Actividad Inicio de Sesión



En la Figura 3 podemos ver cómo opera la funcionalidad de inicio de sesión y los pasos a seguir por el usuario.

Figura 4. Diagrama de Actividad Creación y Comportamiento de Procesos



En la Figura 4 podemos ver cómo opera la aplicación teniendo en cuenta los roles involucrados en el proceso.

■ **Vista de despliegue:**

Muestra el sistema desde la vista de un programador, en otras palabras, se va a mostrar cómo está dividido el sistema software en componentes y las dependencias que hay entre estos.

Figura 5. Diagrama de Componentes Login y Admin

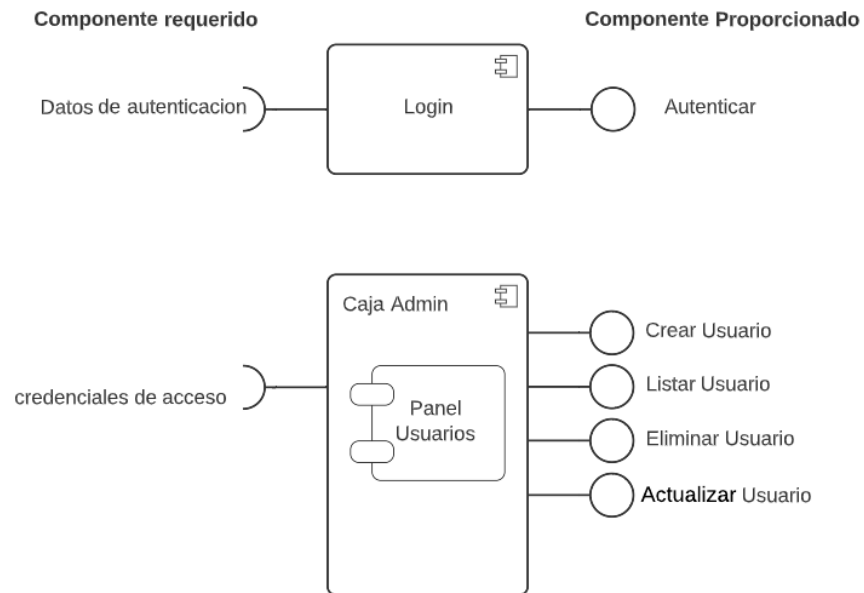


Figura 6. Diagrama de Componentes Encuestas y Procesos

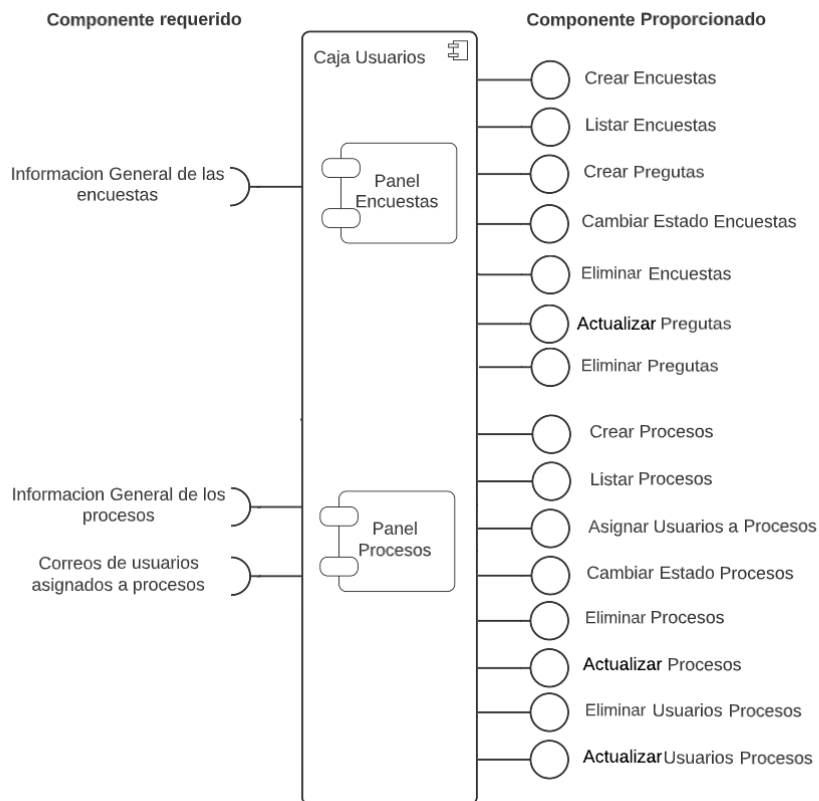
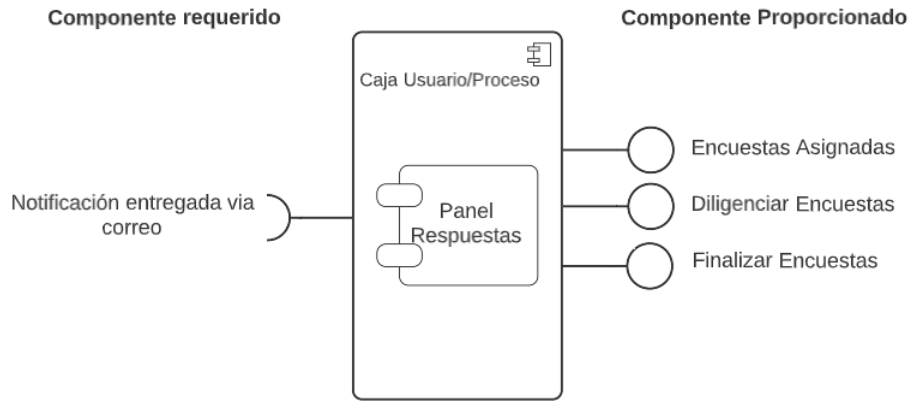


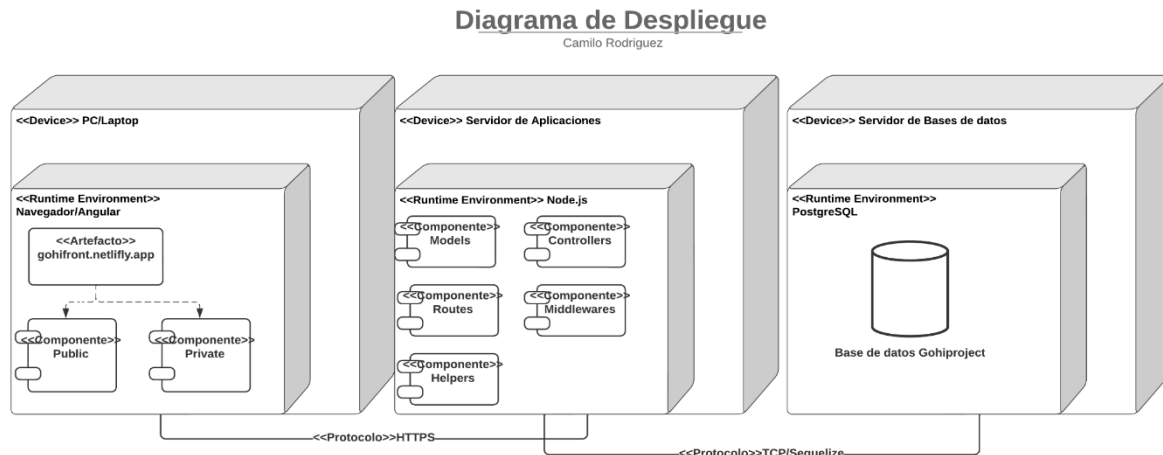
Figura 7. Diagrama de Componentes Respuestas



■ **Vista de física:**

Se muestran todos los componentes físicos del sistema, así como las conexiones entre estos componentes que conforman la solución.

Figura 8. Diagrama de Despliegue

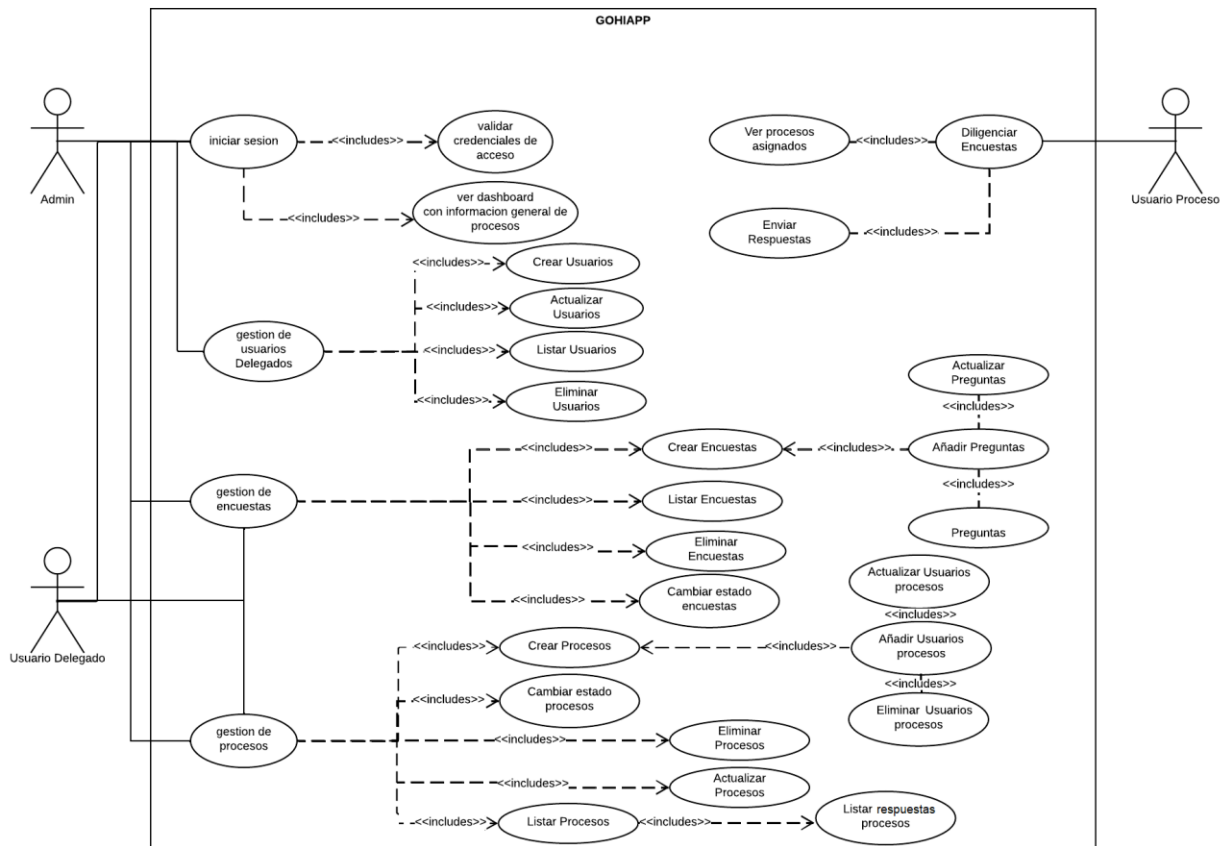


En la Figura 8 podemos ver el funcionamiento de la aplicación con sus nodos y la comunicación entre estos.

■ “+1” Vista de escenarios:

Esta vista se representa con el diagrama de casos de uso y tiene la función de unir y relacionar las 4 vistas.

Figura 9. Diagrama de Casos de Uso



Una vez hemos definido todas las vistas tenemos más claridad con la arquitectura del proyecto. En este punto ya podemos continuar con el desarrollo de la aplicación, el paso a seguir sería el diseño de la base de datos y el comienzo de la codificación.

Tabla 3 Fase 2. Planificación y viabilidad.

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> Anexo 1. Documento de Requisitos del Sistema
Instrumentos y/o herramientas	<ul style="list-style-type: none"> Computador Conexión a internet Navegador Web Google Docs. The 4+1 View Model of architecture
Salidas	<ul style="list-style-type: none"> Diagrama de Clases Diagrama de Actividades Diagrama de Componentes Diagrama de Despliegue

Tipo	Elementos
	● Diagrama de Casos de Uso
Tiempo	4 semanas

Fuente: Autor

7.3. Fase de Diseño

La fase 3 del desarrollo del proyecto consta de dos pasos: el diseño de la base de datos y un diseño preliminar.

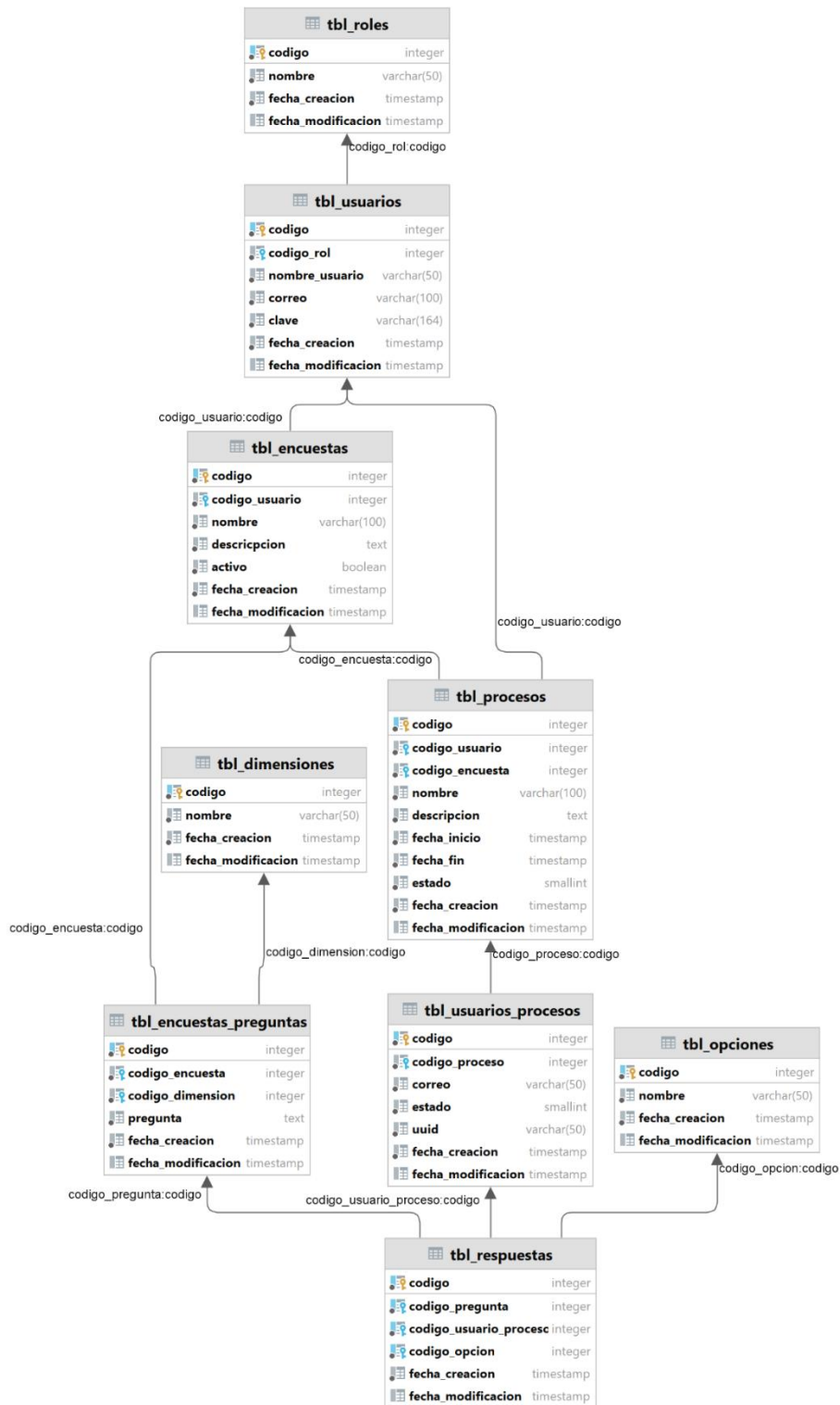
■ Diseño de la base de datos

Teniendo en cuenta el diagrama de clases previamente creado [Figura 2] se propone un diseño de diagrama entidad-relación de la base de datos, el cual está compuesto de las siguientes tablas:

- tbl_rolas: Los roles definidos dentro de la aplicación.
- tbl_usuarios: Información de usuarios.
- tbl_dimensiones: Información de las dimensiones a usar.
- tbl_encuestas: Información de las encuestas.
- tbl_encuestas_preguntas: Información de preguntas su dimensión y encuesta.
- tbl_opciones: Opciones estáticas para encuestas tipo likert.
- tbl_procesos: Información de procesos para el manejo de las encuestas.
- tbl_usuarios_procesos: Información de usuarios asignados a procesos.
- tbl_respuestas: Respuestas de los usuarios.

Cabe resaltar que el diseño de la base de datos se hizo con base a la ejecución del cuestionario Gohisalo y sus respectivas dimensiones, el cual debe implementarse de acuerdo a los objetivos de este proyecto.

Figura 9. Modelo Relacional



■ Diseño preliminar

De acuerdo con las especificaciones de entrega de proyecto se plantea un UX/UI sencillo y eficiente con características responsivas para el uso adecuado de la aplicación en diversos dispositivos, con el fin de la accesibilidad de cualquier usuario.

Para el apartado del UX se plantea una experiencia fácil y sencilla de navegación para cada una de las funcionalidades del sistema que tenga disponibles de acuerdo a su rol.

Teniendo en cuenta un diseño simple para esta aplicación, se planteó el uso del framework de interfaces llamado Bootstrap, con un uso de colores claros para el fondo, y colores oscuros en este caso el color azul para diferentes elementos como botones para ser resaltados, mejorando la accesibilidad.

Tabla 4. Fase 3. Explicación Paleta de Colores

Color	Función
Gris	Resalta sobre el color blanco del fondo para una mejor armonía de colores.
Blanco	Evoca pureza, inocencia y positividad. Ayuda a limpiar el diseño.
Azul	Evoca calma, confianza, honestidad.

Los colores se usaron para:

- Gris: Resaltar secciones como tablas, formularios o menús.
- Blanco: Fondo que permite el realce de otros componentes.
- Azul: Botones, acciones, títulos principales.

Tabla 5 Fase 3. Diseño.

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Anexo 1. Documento de Requisitos del Sistema ● Diagramas de Arquitectura 4+1
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Computador ● Conexión a internet ● Navegador Web ● Google Docs.
Salidas	<ul style="list-style-type: none"> ● Diseño entidad-relación de la base de datos ● Diseño preliminar del sistema
Tiempo	4 semanas

Fuente: Autor

7.4. Fase de Desarrollo

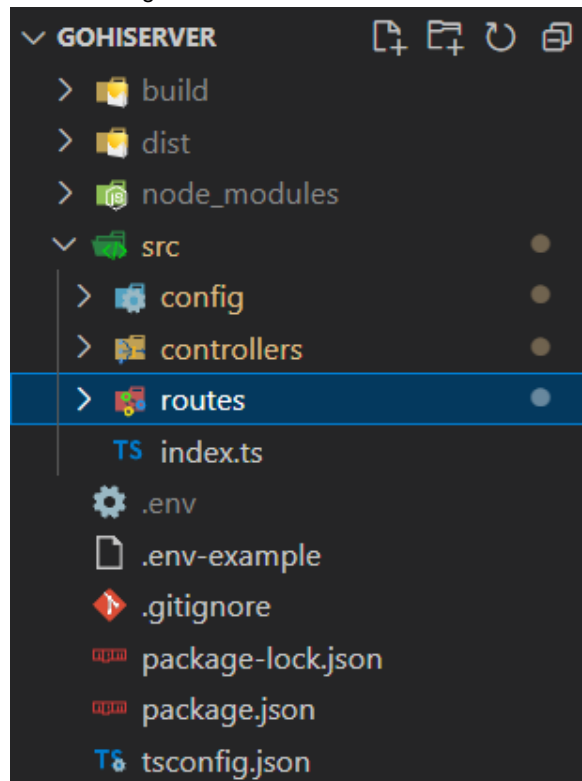
Para el desarrollo o la codificación de este proyecto se tienen en cuenta dos fases principales que son el Frontend y el Backend, a continuación, se muestran evidencias de cómo se llevaron a cabo dichas fases.

Desarrollo del Backend

La implementación del backend se construyó sobre el framework express de nodeJS, haciendo uso del lenguaje de programación TypeScript. Lo cual nos permitió modelar el servicio sobre diferentes rutas y controladores, permitiendo un manejo simple y escalable del servicio. La arquitectura API/REST de servicio que brinda las respuestas a las respectivas peticiones son:

- Configuración: Gestiona la conexión con la base de datos (dialecto, usuario, contraseña) y también el manejo de consultas (errores y resultados).
- Controladores: Permiten crear y almacenar consultas de todo tipo a la base de datos.
- Rutas: Son las direcciones a los que el cliente se va dirigir para obtener la información según lo necesite, todo en se entrega en formato json.

Figura 10. Estructura del Backend



● *Conexión con la base de datos*

Para la conexión exitosa con la base de datos fue necesario crear un archivo de environment que permitió hacer la configuración. Este archivo tiene como información clave: el nombre de la base de datos, usuario de la base de datos, contraseña, host y dialecto que en este caso es PostgreSQL.

Figura 11. archivo.env

```
1 PORT=8099
2 DIALECT=postgres
3 HOST=localhost
4 USER=Camilo_R
5 PASSWORD=juan1233
6 DATABASE=db_encuestas
7 FRONTLINK=http://localhost:4200
```

Luego de definir el entorno en el vamos a trabajar creamos un archivo de configuración para consumir esta información.

Figura 12. configdb.ts

```
import * as dotenv from 'dotenv';
dotenv.config();

export default {
  database: {
    dialect: process.env.DIALECT,
    host: process.env.HOST,
    user: process.env.USER,
    password: process.env.PASSWORD,
    database: process.env.DATABASE,
    ssl: Boolean(process.env.SSL)
  }
}
```

Para finalizar hacemos la conexión con la base de datos utilizando la librería pg de PostgreSQL haciendo un pool con la información previamente definida.

Figura 13. conn.ts

```
import variables from './configdb';
import {Pool} from 'pg';

const pool= new Pool (variables.database);

pool.connect(function(err, conn){
  if(err){
    console.log('el codigo del error es:',err.stack);
  }else{
    if(conn){
      conn.release();
    }
  }
  console.log('conexiones establecidas con:', variables.database);
});

export default pool;
```

Antes de comenzar con los controladores fue necesario crear un archivo para el manejo de los errores y respuestas de las peticiones a la base de datos. Esto con

el fin de dar un orden al flujo de las peticiones y errores de acuerdo a su tipo, ejemplo: insert, select, delete.

Figura 14. db_manager.ts

```
import { request, response, Response } from 'express';
import pool from './conn';
import jwt from 'jsonwebtoken';

class dbManager {
  protected static async ejecutarConsulta(sql: string, parametros: any, res: Response, tipo: string): Promise<any> {

    pool.query(sql, parametros, function (error, resultado) {
      if (error) {
        if (error.stack == 'ER_BAD_DB_ERROR') {
          console.log('No existe la database');
          res.status(400).json({ 'respuesta': 'base de datos no existe' });
        }
        if (error.stack == 'ER_ACCESS_DENIED_ERROR') {
          console.log('Nombre de usuario o contraseña es incorrecta');
          res.status(400).json({ 'respuesta': 'Nombre de usuario o contraseña es incorrecta' });
        }
        if (error.stack == 'ENOTFOUND') {
          console.log('Error servidor');
          res.status(400).json({ 'respuesta': 'Error servidor' });
        }
        if (error.stack == 'ER_PARSE_ERROR') {
          console.log('Esta consumiendo el servicio de manera incorrecta');
          res.status(400).json({ 'respuesta': 'Esta consumiendo el servicio de manera incorrecta' });
        }
        console.log('Se ha encontrado un error: ', error);
        res.status(400).json({ 'respuesta': error });
      } else {
        switch (tipo.toLowerCase()) {
          case 'select':
            res.status(200).json(resultado.rows);
            break;
          case 'insert':
            res.status(200).json({
              'mensaje': 'Registro creado',
              'id': resultado.rows
            });
            break;
        }
      }
    });
  }
}
```

● *Controladores*

Los controladores son los que almacenan las peticiones a la base de datos con el fin de enviar estas respuestas al usuario. A continuación, se muestra uno de los controladores:

Figura 15. encuestasController.ts

```
import { Request, Response } from 'express'
import dbManager from '../config/db_manager';

class EncuestasController extends dbManager {

  public obtenerEncuestas(req: Request, res: Response): Promise<any> {
    const sql: string = "SELECT * FROM tbl_encuestas ORDER BY codigo desc ";
    return EncuestasController.ejecutarConsulta(sql, [], res, 'select');
  }

  public obtenerEncuestasUuid(req: Request, res: Response): Promise<any> {
    if (String(req.params.uuid)) {
      const codigo = String(req.params.uuid);
      const { codigo_dimension } = req.body;
      const sql = `SELECT usu.codigo as "codigo_usuario_proceso", encu.nombre as "encuesta",
INNER join tbl_dimensiones dim ON dim.codigo = preg.codigo_dimension
INNER join tbl_procesos proc ON proc.codigo_encuesta = encu.codigo
INNER join tbl_usuarios_procesos usu ON usu.codigo_proceso = proc.codigo
WHERE usu.uuid= $1 and dim.codigo= $2 `;
      const parametros = [codigo,codigo_dimension];
      return EncuestasController.ejecutarConsulta(sql, parametros, res, 'select');
    } else {
      return Promise.resolve(res.status(400).json({
        'mensaje': 'Codigo invalido '
      }));
    }
  }
}
```

● *Rutas*

Las rutas o también conocidos endpoints permiten la comunicación entre el backend y el frontend. Estas se hacen mediante peticiones HTTP como GET, POST, PUT/PATCH, y DELETE.

Figura 16. encuestasRoutes.ts

```
class EncuestasRoutes {
  public router: Router;

  constructor() {
    this.router = Router();
    this.config();
  }

  public config(): void {
    this.router.get('/', encuestasController.obtenerEncuestas);
    this.router.post('/info/:codigo_encuesta', encuestasController.obtenerEncuestasInfo);
    this.router.post('/uuid/:uuid', encuestasController.obtenerEncuestasUuid);
    this.router.get('/opciones', encuestasController.obtenerOpciones);
    this.router.get('/grafica', encuestasController.obtenerDatosGrafica);
    this.router.post('/diligenciar', encuestasController.diligenciarEncuesta);
    this.router.post('/crear', encuestasController.crearEncuestas);
    this.router.delete('/:codigo_encuesta', encuestasController.borrarEncuesta);
    this.router.put('/actualizar/:codigo_encuesta', encuestasController.actualizarEstadoEncuestas);
  }
}
```

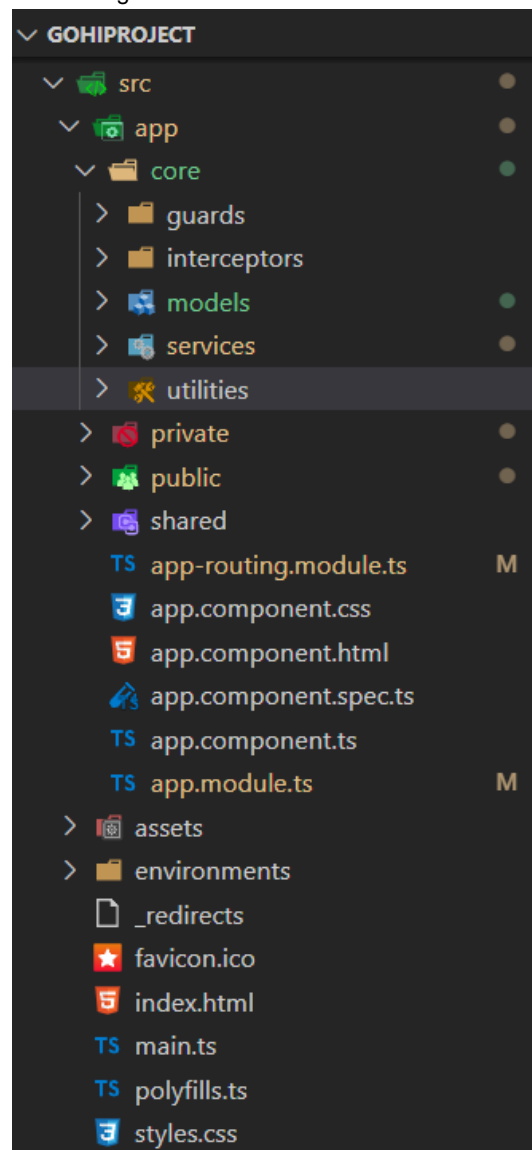
■ Desarrollo del Frontend

De acuerdo a las fases anteriores, la arquitectura del cliente web fue diseñada usando el framework para manejo de vistas angular haciendo un correcto uso de su estructura, haciendo uso de los siguientes elementos dentro del código y app:

- app.component: Contiene el componente principal donde se va a situar toda la single page app y el routing.
- Componentes: En este directorio se van a situar todos y cada uno de los componentes que se van a mostrar al usuario.
- Servicios: En este directorio se va a ubicar todos los servicios que van a hacer peticiones y van a gestionar la información en los componentes.
- Modelos: En este directorio se ubicarán los modelos de base de datos que se necesiten para el desarrollo.
- App.module.ts: Este archivo siendo uno de los más importantes ya que nos permitirá administrar los módulos de librerías externas que vayamos agregando a la aplicación.

- App.routing.ts: Este archivo nos permitirá manejar todo lo relacionado con las rutas y sus distintas utilidades para con la aplicación.
- Styles.css: Estilos CSS globales personalizados que se utilizan en componentes y páginas.
- Assets: Este directorio en particular nos permite almacenar las utilidades estáticas como imagen etc., para el uso en la aplicación.

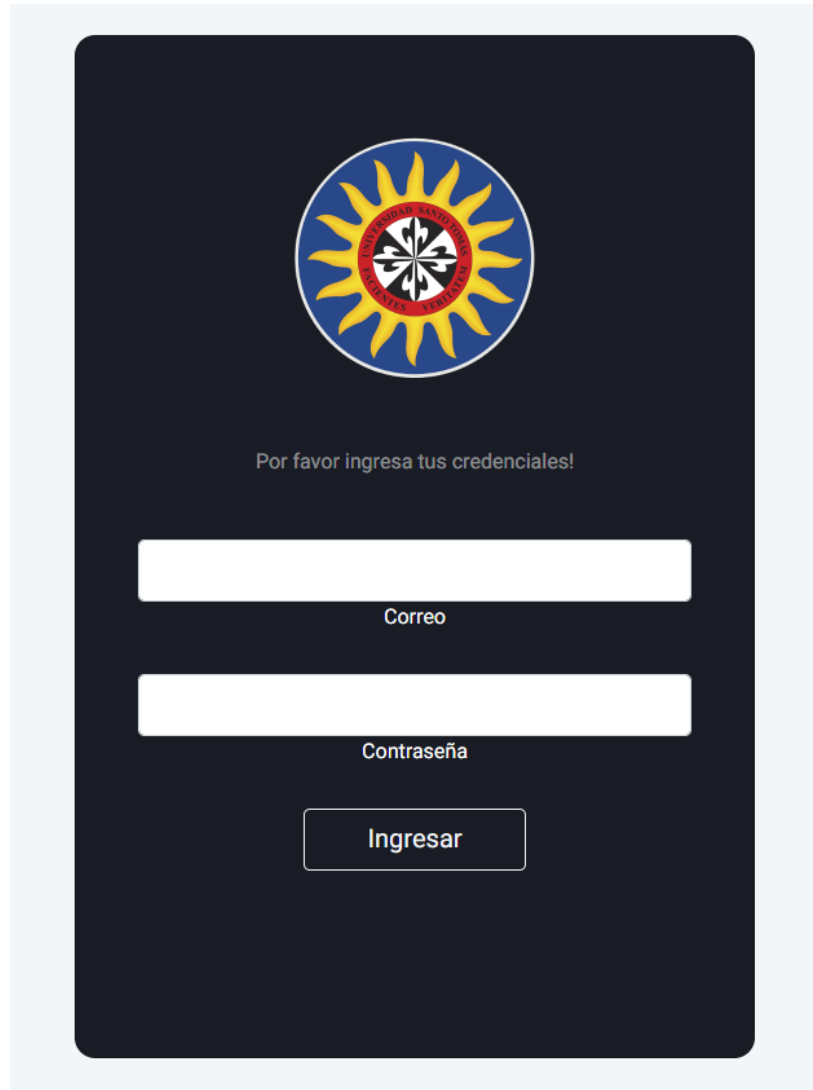
Figura 17. Estructura del Frontend



● *Desarrollo de interfaces*

Como primera vista del proyecto nos hace logear para ver los demás componentes de la aplicación los cuales unos serán compartidos y otros solo tendrán acceso ciertos usuarios de acuerdo a su rol.

Figura 18. Login



The image shows a login interface on a dark blue background. At the top center is the logo of the Universidad San Carlos de Guatemala, which features a sunburst design with a cross in the center and the university's name in Spanish. Below the logo, the text "Por favor ingresa tus credenciales!" is displayed. There are two white input fields: the first is labeled "Correo" and the second is labeled "Contraseña". At the bottom, there is a white button labeled "Ingresar".

1. Interfaces Admin

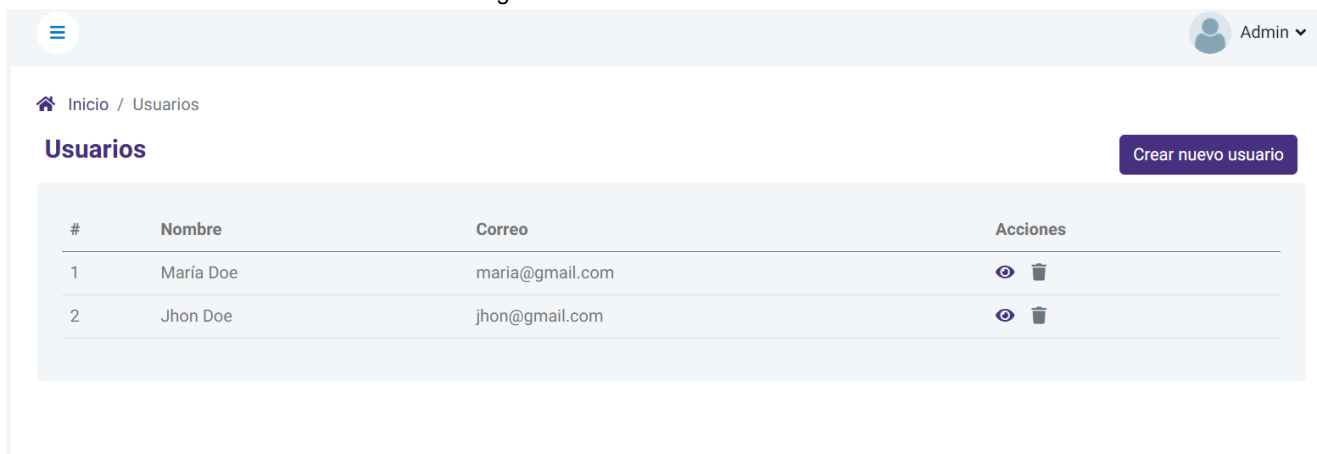
En este caso el rol de administrador puede acceder a todos los componentes que accede el usuario delegado, con la excepción de la gestión de usuarios lo cual solo puede hacer el.

- **Gestión de usuarios**





En este caso tenemos dos vistas, listar usuarios y administrarlos (en este caso solo puede eliminarlos), más la vista de crearlos.

- **Listar Usuarios**

Figura 19. Vista Listar Usuarios

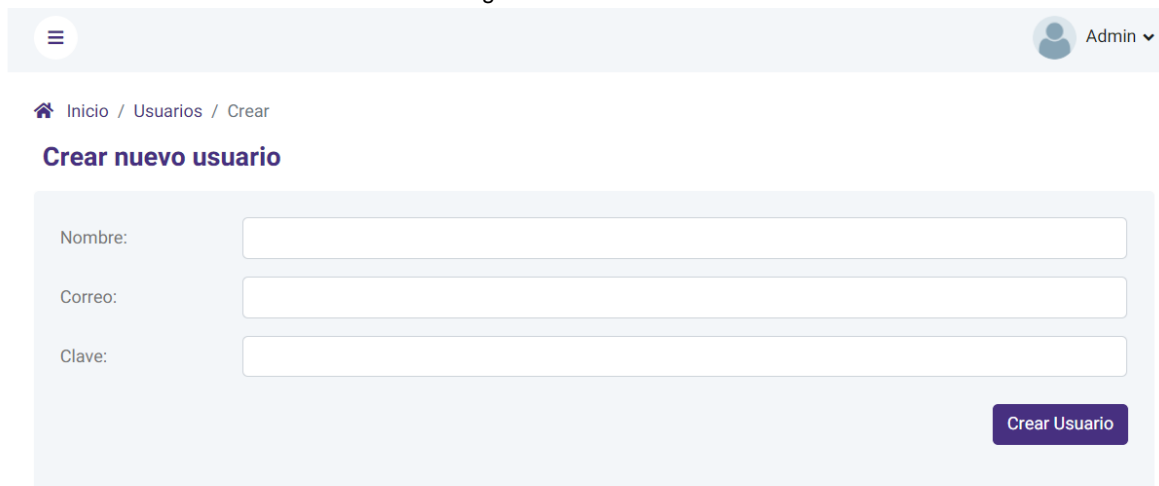


The screenshot shows the 'Listar Usuarios' view. At the top right, there is a user profile icon labeled 'Admin'. Below the header, there is a breadcrumb trail 'Inicio / Usuarios' and a 'Crear nuevo usuario' button. The main content is a table with the following data:

#	Nombre	Correo	Acciones
1	María Doe	maria@gmail.com	 
2	Jhon Doe	jhon@gmail.com	 

- **Crear Usuarios**

Figura 20. Vista Crear Usuarios



The screenshot shows the 'Crear Usuarios' view. At the top right, there is a user profile icon labeled 'Admin'. Below the header, there is a breadcrumb trail 'Inicio / Usuarios / Crear' and a 'Crear nuevo usuario' button. The main content is a form with the following fields:

Nombre:

Correo:

Clave:

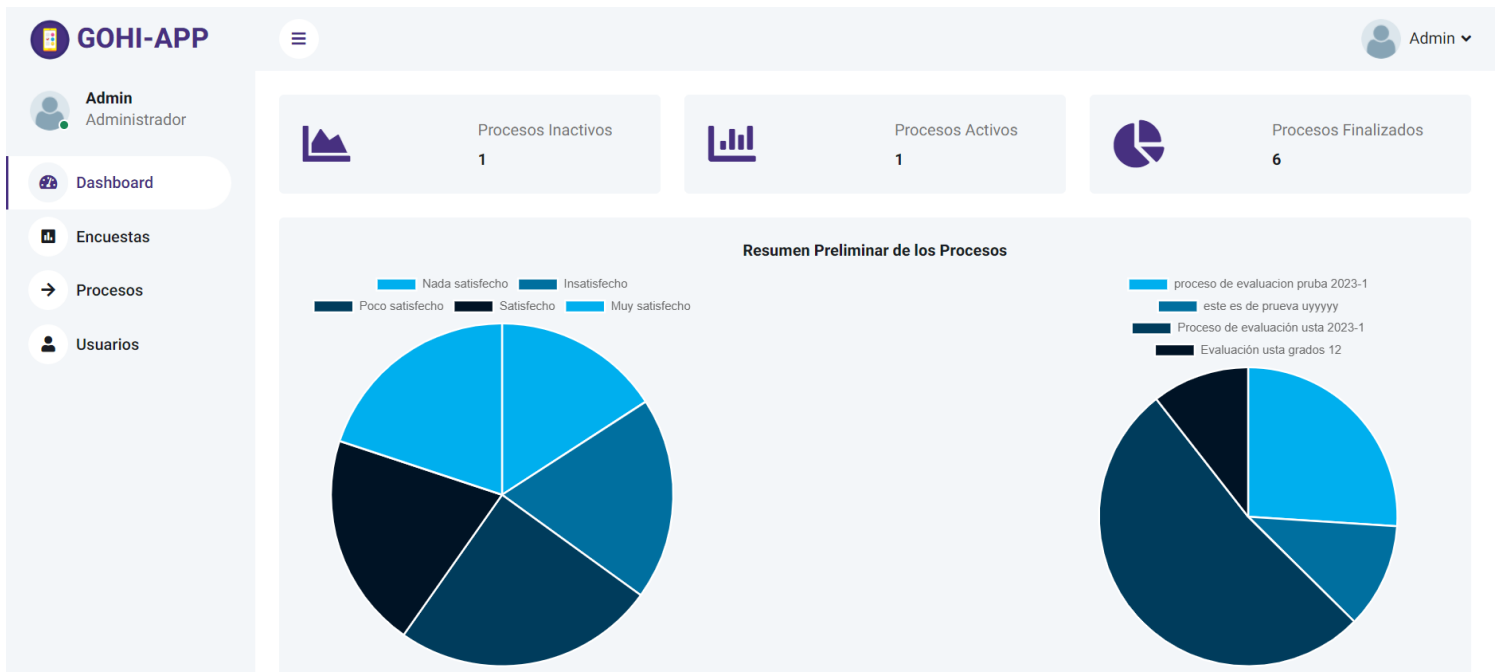
At the bottom right, there is a 'Crear Usuario' button.

2. Interfaces Usuario delegado + Admin

- Home

Esta es una interfaz compartida ya que nos permite ver la información general de las encuestas hechas tanto por el administrador como los usuarios delegados.

Figura 21. Vista Home



- Gestión de encuestas

En esta sección tenemos dos vistas una para listar y administrar (cambiar estado) y otra para crear encuesta y agregar preguntas a la encuesta.

- **Listar Encuestas**

Figura 22. Vista Listar Encuestas

Versión	Nombre	Estado	Acciones
1	Cuestionario Gohisalo	Activa	
2	Evaluación de 360 grados	Inactiva	
3	Encuesta de Gohi	Inactiva	

- **Crear encuestas**

Figura 23. Vista Crear Encuestas

Información general

Nombre encuesta:

Descripción:

Una vez creamos la encuesta se desplegará un acordeón hacia abajo con las dimensiones del cuestionario Gohisalo para llenarlas.

Figura 23. Vista Llenar Encuestas

Información general

Nombre encuesta:

Descripción:

Dimensiones

1 - Soporte institucional para el trabajo ^

1. ✓ 🗑️

2 - Seguridad en el trabajo ∨

3 - Integración al puesto de trabajo ∨

- **Editar Preguntas encuesta**

Figura 24. Editar Encuestas Preguntas

☰ Admin ∨

Inicio / Encuestas / Editar

Dimensiones

1 - Soporte institucional para el trabajo ^

1 nueva pregunta ✓ 🗑️

2 nueva pregunta 🗑️

3 mi nueba prueba nueva 🗑️

2 - Seguridad en el trabajo ∨

3 - Integración al puesto de trabajo ∨

4 - Satisfacción por el trabajo ∨

- **Gestión de procesos**

En esta sección tenemos dos vistas una para listar y administrar (cambiar estado) y otra para crear proceso y agregar usuarios al proceso.

- **Listar Procesos**

Figura 25. Vista Listar Procesos

Código	Nombre	Estado	Fecha inicio	Fecha fin	Acciones
1	proceso de evaluacion prueba 2023-1	Inactivo	2023-05-08 19:12:00	2023-06-08 19:12:00	
2	Proceso de evaluación usta 2023-1	Activo	2023-05-06 20:36:00	2023-06-05 20:36:00	
3	este es de prueba uyyyyy	Finalizado	2023-04-17 00:12:00	2023-04-21 00:12:00	
4	Evaluación usta grados 12	Finalizado	2023-03-01 22:24:00	2023-03-02 22:24:00	
5	Proceso de evaluación TI	Finalizado	2022-11-11 00:00:00	2022-12-12 00:00:00	
6	Proceso de evaluación recursos humanos	Finalizado	2022-10-10 00:00:00	2022-11-11 00:00:00	
7	Proceso de evaluación 2021-2	Finalizado	2022-02-02 00:00:00	2022-03-03 00:00:00	
8	Proceso de evaluación 2021-1	Finalizado	2022-07-07 00:00:00	2022-08-08 00:00:00	

- **Crear Procesos**

Figura 26. Vista Crear Procesos

Crear nuevo proceso de evaluación

Información general

Nombre proceso:

Descripción:

Encuesta:

Fecha inicio:

Fecha fin:

Crear Proceso

Una vez creamos el proceso se desplegará un formulario hacia abajo que nos permitirá agregar los correos de los usuarios al proceso.

Figura 26. Vista asignar usuarios Procesos

Crear nuevo proceso de evaluación

Información general

Nombre proceso:

Descripción:

Encuesta:

Fecha inicio:

Fecha fin:

Crear correos

Mis Correos

1 jhon4@gmail.com

2 jhon41@gmail.com

- **Editar Procesos y Usuarios Procesos**

Figura 27. Vista Editar Procesos y Usuarios de proceso

Inicio / Procesos / Editar
Admin

Información general

Nombre proceso:

Descripción:

Encuesta:

Fecha inicio:

Fecha fin:

Actualizar Proceso

Asignar Usuarios

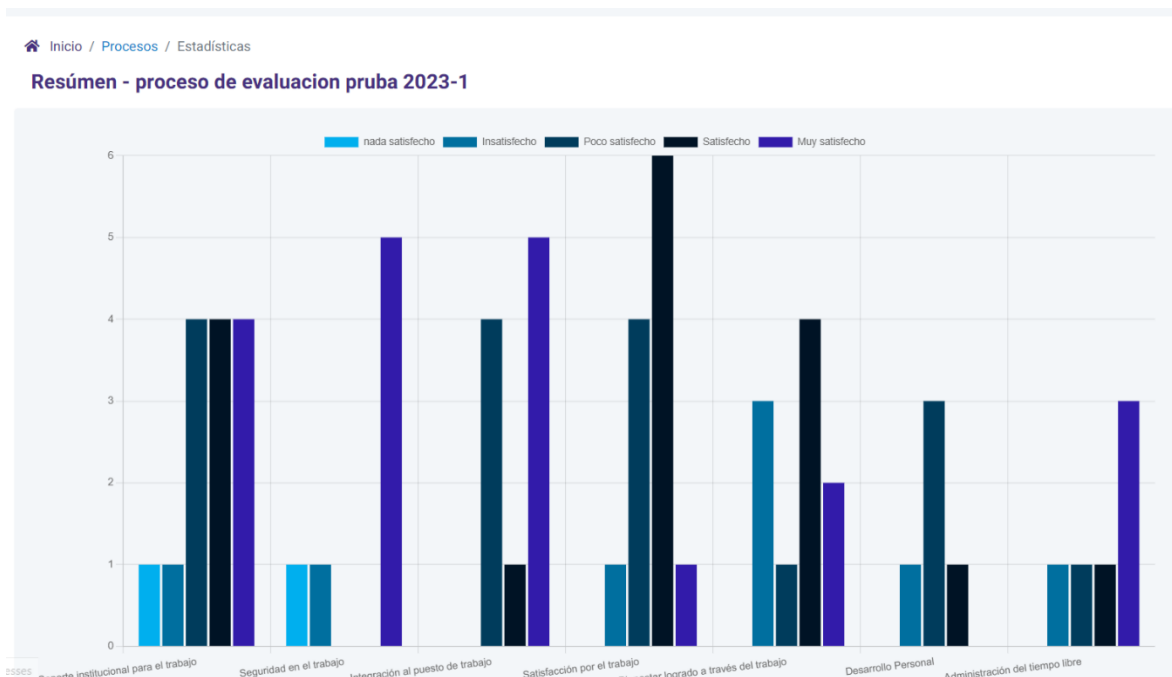
Mis Usuarios

1	juankmilo9902@gmail.com	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	binaribeast77@gmail.com	<input type="checkbox"/>	<input type="checkbox"/>

Asignar Usuarios

- **Ver respuestas a los procesos**

Figura 28. Vista grafica Respuestas



3. Interfaces Usuario Proceso

Este es el usuario final de las encuestas, el cual se encarga de contestar el contenido. Para esto una vez es asignado a un proceso dentro del aplicativo se le hace una notificación vía correo para que ingrese al sistema y responda las encuestas que tiene asignadas.

Figura 29. Vista Diligenciar Encuesta

Información general

Creado por: Juan Rodriguez **Fecha inicio:** 2022-01-02 **Fecha fin:** 2022-01-10

Nombre encuesta: Encuesta Gohi V2

Descipción de la encuesta:
 lorem pis

Nombre proceso: Proceso de evaluación año 2022

Descipción del proceso:
 lorem pis

Preguntas

1 - Soporte institucional para el trabajo ^

1 . Grado de satisfacción que tengo con respecto al trato que recibo de mis superiores

- Nada satisfecho
- Insatisfecho
- Poco satisfecho
- Satisfecho
- Muy satisfecho

2 . Considero que tengo libertad para expresar mis opiniones en cuanto al trabajo sin temor a represalias de mis jefes

- Nada satisfecho
- Insatisfecho

Tabla 6 Fase 4. Desarrollo de la aplicación.

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Anexo 1. Documento de Requisitos del Sistema ● Diagramas de Arquitectura 4+1 ● Modelo Entidad Relación de la Base de Datos
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Computador ● Conexión a internet ● Navegador Web ● Visual Studio Code ● PgAdmin

Tipo	Elementos
	<ul style="list-style-type: none">• Datagrip• Postman
Salidas	<ul style="list-style-type: none">• Construcción del Backend• Construcción del Frontend• Integración del Backend al Frontend
Tiempo	16 semanas

Fuente: Autor

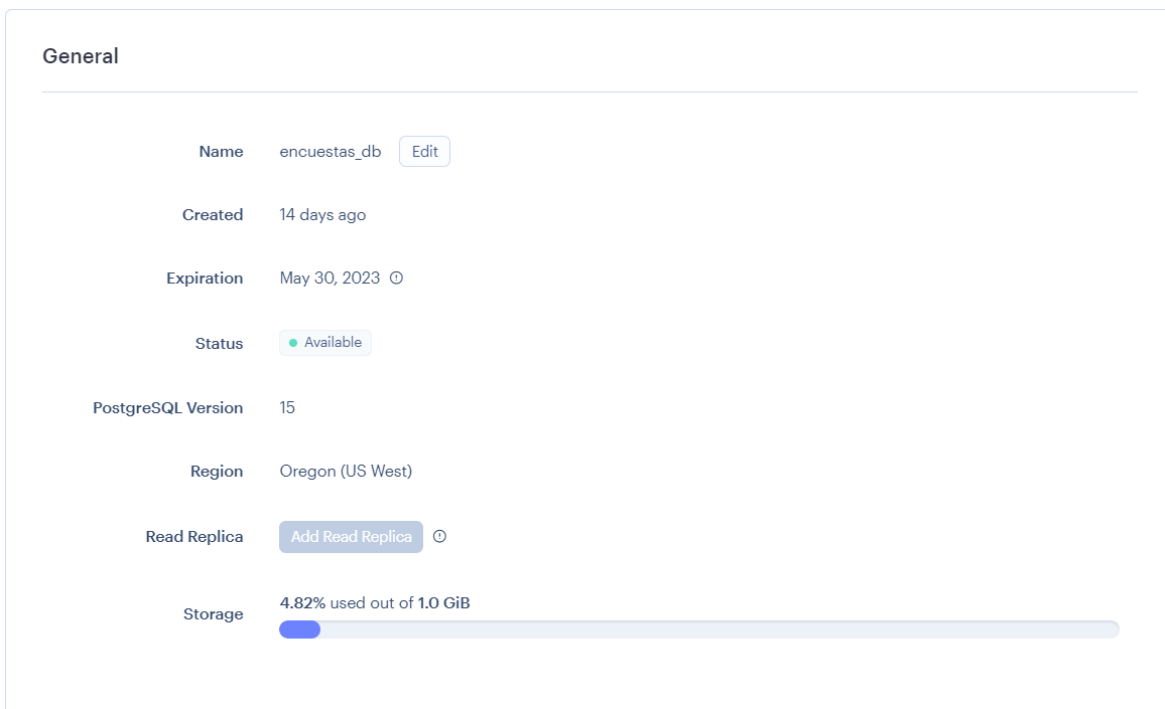
7.5. Fase de Implementación

Para la fase de implementación se dividen 3 fases que se conforman en hosteo de la base de datos, del backend y el frontend.

■ Base de Datos

Para la implementación de base de datos en la nube se utilizaron dos herramientas. Para la creación de la base de datos se hizo uso de la herramienta render que es una plataforma para alojar aplicaciones y sitios web en la nube.

Figura 30. Base de datos en la nube



Una vez creada la base de datos en la plataforma render ya podremos extraer las variables de configuración para hacer la posterior conexión con la misma.

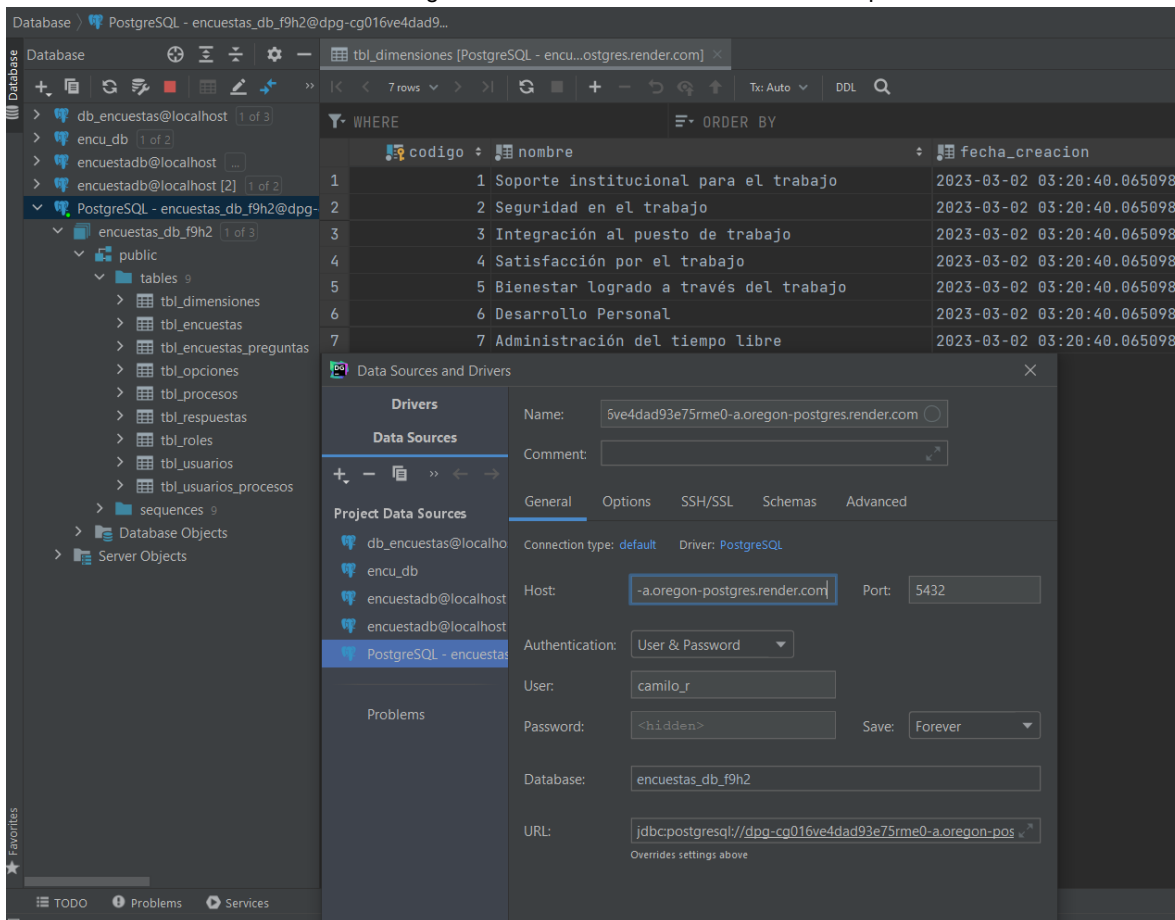
Figura 31. Variables de conexión base de datos

The screenshot shows a configuration window for database connections. The title is "Connections". Below the title, there are several fields for configuration:

- Hostname: dpg-cg016ve4dad93e75rme0-a
- Port: 5432
- Database: encuestas_db_f9h2
- Username: camilo_r
- Password: Ta0mlioIjEq2Gw3A6NjdVycGPjp7ygda
- Internal Database URL: postgres://camilo_r:Ta0mlioIjEq2Gw3A6NjdVycGPjp7ygda@dpg-cg016ve4dad93e75rme0-
- External Database URL: postgres://camilo_r:Ta0mlioIjEq2Gw3A6NjdVycGPjp7ygda@dpg-cg016ve4dad93e75rme0-
- PSQL Command: PGPASSWORD=Ta0mlioIjEq2Gw3A6NjdVycGPjp7ygda psql -h dpg-cg016ve4dad93e75rme0-a

Luego de tener las variables de conexión vamos a hacer uso de la herramienta DataGrip donde haremos la conexión remota con la base de datos, además vamos a hacer uso del Anexo 2 para la inserción de los scripts que contienen las tablas y los inserts.

Figura 32. Conexión base de datos DataGrip

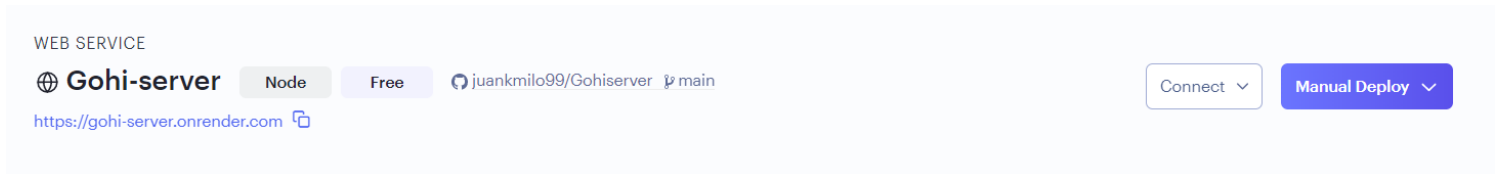


Una vez implementados los scripts ya podemos comenzar con la conexión del backend y su posterior implementación.

■ Backend

Para el backend también vamos a hacer uso de la herramienta render para alojar en la nube nuestro backend. Para su despliegue es necesario tener el código de nuestra aplicación en Github para poder consumir su contenido de forma remota. Dentro de render luego de crear la aplicación vamos a conectar nuestro repositorio con render.

Figura 33. Backend conectado con Github



Luego de crear nuestra aplicación o servidor veremos que ya nos ha generado un dominio para generar nuestras peticiones, el paso a seguir es configurar las variables de entorno para que nuestro servidor apunte hacia la base de datos que hemos creado anteriormente.

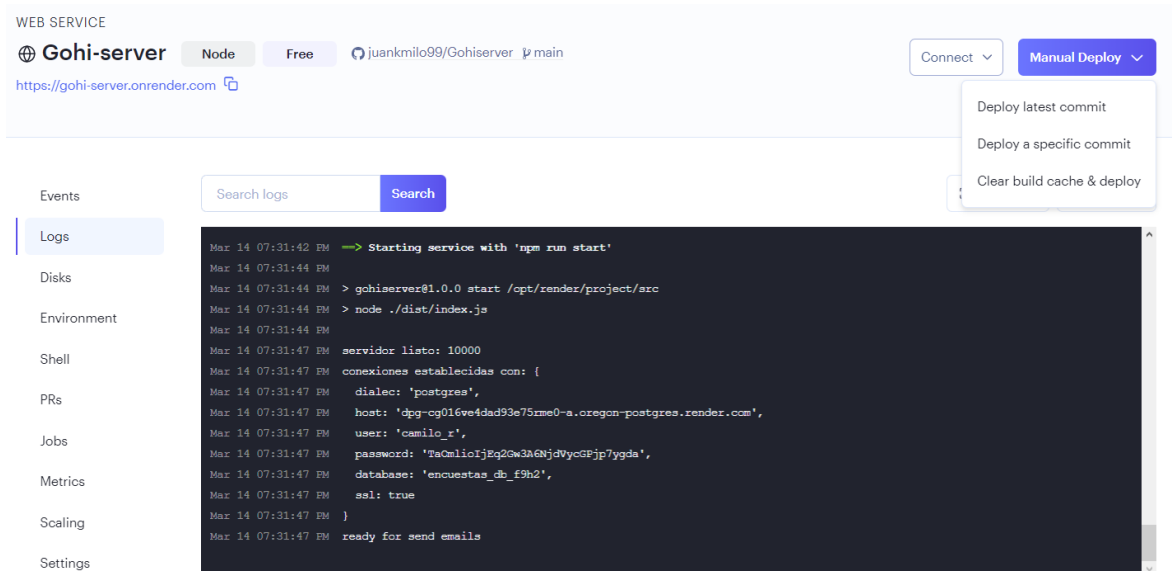
Figura 34. Variables de entorno backend

The screenshot shows the 'Environment Variables' configuration page. It has a title 'Environment Variables' and a subtitle 'Use environment variables to store API keys and other configuration values and secrets. You can access them in your code like regular environment variables, for example with `os.getenv()` in Python or `process.env` in Node.' Below this, there is a table with two columns: 'Key' and 'Value'. The table has seven rows, each with a key and a value field. The first row has the key 'DATABASE' and the value 'encuestas_db_f9h2'. The other rows have keys 'DIALECT', 'FRONTLINK', 'HOST', 'PASSWORD', 'SSL', and 'USER', and their values are represented by dots. To the right of each value field is a trash icon. At the bottom left, there is a link 'Create Environment Group'. At the bottom right, there are two buttons: 'Add Environment Variable' and 'Save Changes'.

Key	Value
DATABASE	encuestas_db_f9h2
DIALECT
FRONTLINK
HOST
PASSWORD
SSL
USER

El paso final de la implementación es sencillo ya que lo único que nos queda por hacer es un deploy al último commit al repositorio del backend para poder verificar si las conexiones están hechas.

Figura 35. Deploy del backend



Como podemos ver el deploy ha sido exitoso y ya podemos usar nuestro servidor para ser consumido por el frontend.

■ Frontend

En cuanto al despliegue del frontend se hizo uso de la herramienta netlify que es una plataforma que nos permite hacer hosting de nuestra aplicación sin costo y de manera sencilla. Para empezar, vamos a generar un build del frontend dentro de angular con el comando ng build, esto nos genera una carpeta con los componentes de nuestra aplicación para posteriormente subirla a la plataforma.

Figura 36. Build del frontend

```
PS D:\proyectos\Angular_Projects\gohiProject> ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✖ Generating index html...2 rules skipped due to selector errors:
  legend+* -> Cannot read property 'type' of undefined
  .table>>>* -> Did not expect successive traversals.
✓ Index html generation complete.
```

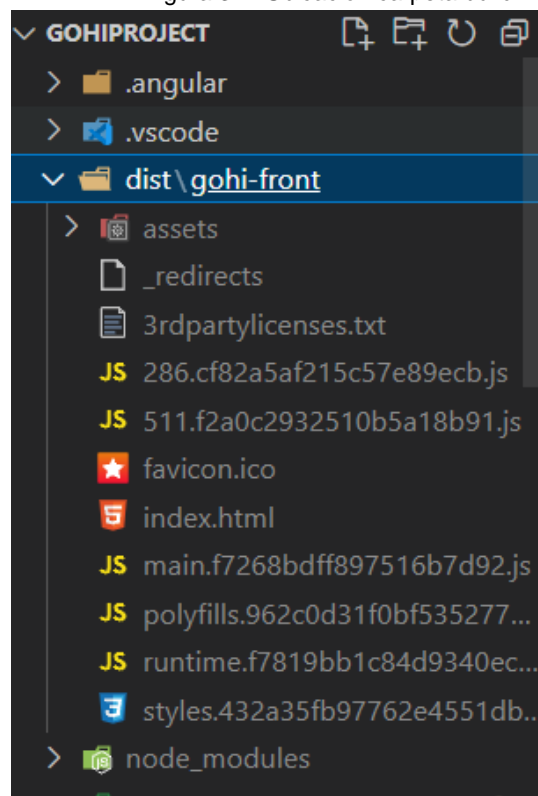
Initial Chunk Files	Names	Size
main.f7268bdf897516b7d92.js	main	725.46 kB
polyfills.962c0d31f0bf53527791.js	polyfills	33.11 kB
styles.432a35fb97762e4551db.css	styles	5.31 kB
runtime.f7819bb1c84d9340ec3d.js	runtime	2.74 kB
	Initial Total	766.63 kB

Lazy Chunk Files	Names	Size
511.f2a0c2932510b5a18b91.js	-	8.91 kB
286.cf82a5af215c57e89ecb.js	-	4.16 kB

```
Build at: 2023-03-08T17:26:04.317Z - Hash: cae827689e3cc18132ee - Time: 68611ms
```

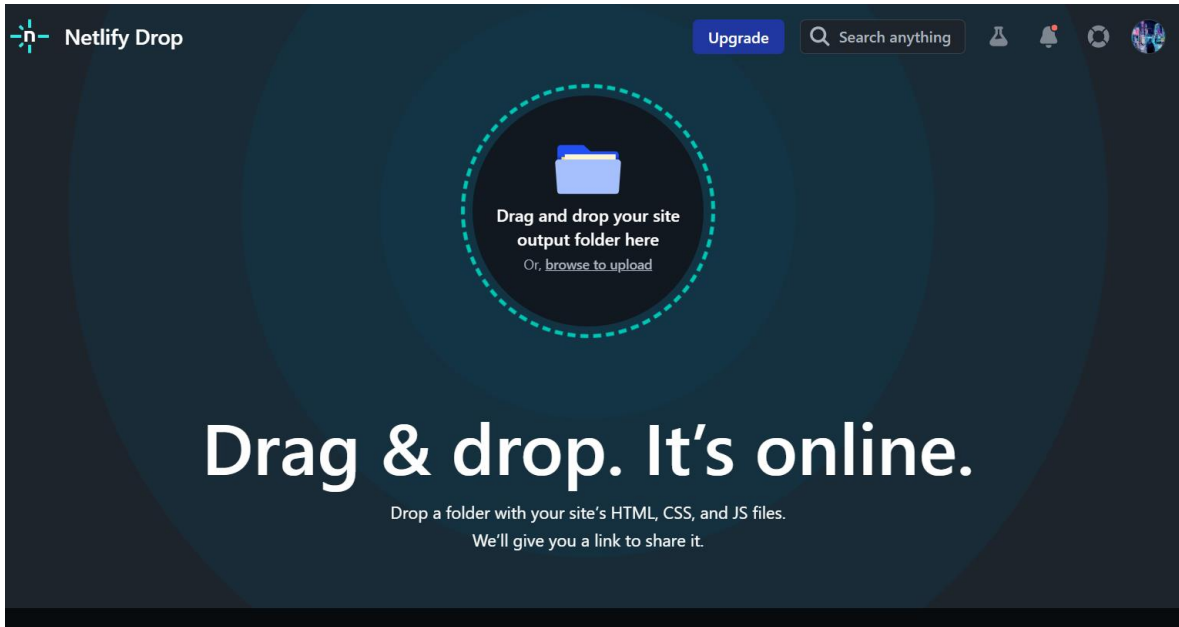
Una vez hecho esto tenemos creada la carpeta del proyecto que vamos a utilizar dentro de netlify.

Figura 37. Ubicación carpeta build



Luego de esto procedemos a hacer el despliegue del sitio en la plataforma netlify, para esto nos pide que subamos la carpeta del build para generar el dominio.

Figura 38. Deploy del frontend



Una vez finalizado el despliegue de la aplicación vemos que ya nos ha generado la URL de alojamiento del frontend la cual corresponde a <https://gohifront.netlify.app>

Figura 39. Dominio del frontend

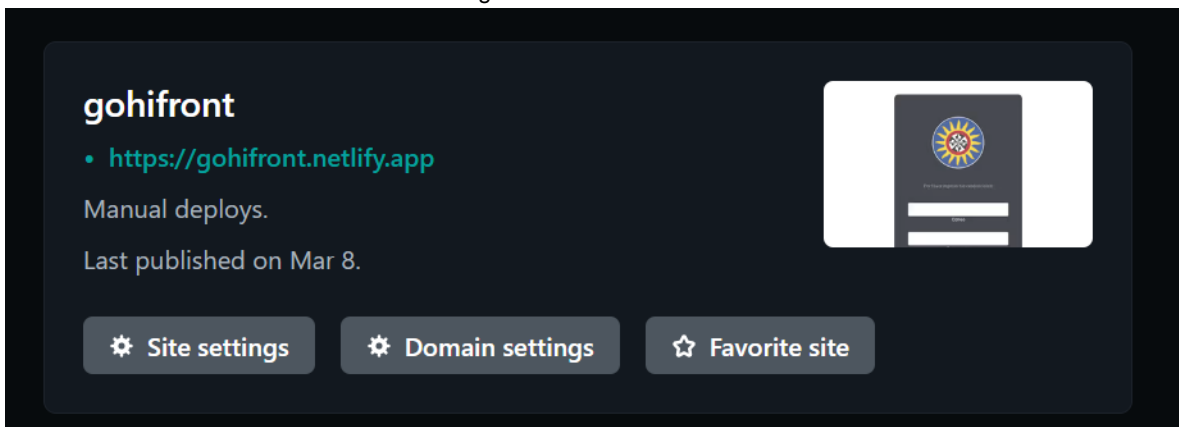


Tabla 7 Fase 5. Implementación

Tipo	Elementos
Entradas	<ul style="list-style-type: none">● Backend y Frontend compilados
Instrumentos y/o herramientas	<ul style="list-style-type: none">● Computador● Conexión a internet● Navegador Web

Tipo	Elementos
	<ul style="list-style-type: none"> ● Visual Studio Code ● Render ● Netlify
Salidas	<ul style="list-style-type: none"> ● https://gohifront.netlify.app
Tiempo	2 semanas

Fuente: Autor

8. LISTAS DE CHEQUEO

En el proyecto, se realizaron listas de chequeo para verificar la funcionalidad de cada uno de los requerimientos. Estas listas se utilizaron para asegurarse de que todas las características y funcionalidades definidas en los requerimientos se implementaron correctamente y cumplían con las expectativas del cliente.

Para crear las listas de chequeo, se revisaron cuidadosamente cada uno de los requerimientos y se definieron criterios precisos para verificar si se cumplían las condiciones y funcionalidades requeridas.

Gracias a estas listas de chequeo, se pudo detectar cualquier error o fallo en el software y corregirlos antes de que afectaran al usuario final.

Tabla 8 . Listas de chequeo

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> ● Requerimientos del sistema y aplicación desplegada
Instrumentos y/o herramientas	<ul style="list-style-type: none"> ● Computador ● Conexión a internet ● Navegador Web ● Google Docs
Salidas	<ul style="list-style-type: none"> ● Anexo 3. Documento de Listas de Chequeo del Sistema
Tiempo	2 semanas

Fuente: Autor

9. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

- El desarrollo de la herramienta con la selección de tecnologías adecuadas, como Angular, Node Express y PostgreSQL, permitió que la herramienta web fuera más rápida, escalable y segura. Además, la inclusión de preguntas de tipo Likert en sus 7 dimensiones permitió que los usuarios respondieran de manera más precisa y fácil a las encuestas.
- Se evidenció que el modelo 4+1 propuesto por Kruchten es una buena opción para especificar las vistas lógica, procesos, física y despliegue de la aplicación web. Este modelo proporciona una visión clara y detallada de la estructura de la aplicación web.
- La definición clara de los requisitos funcionales y no funcionales, así como el uso del modelo 4+1, permitió que el proceso de desarrollo se ejecutara de manera más eficiente y efectiva.
- En general, la implementación exitosa de la herramienta web utilizando el cuestionario CVT-GOHISALO y las tecnologías adecuadas puede permitir que las organizaciones obtengan información valiosa de sus clientes y mejoren la calidad de sus servicios y productos ofrecidos. Además, la herramienta puede ser utilizada en diferentes contextos y áreas de investigación.

10. REFERENCIAS

- [1] M. A. Vilchis Rodríguez, «La Web 2.0 y la nube», Universidad Autónoma del Estado de Hidalgo, 28/11/18.
- [2] P. Cadena, «Introducción al uso de la Web 2.0 en el estado colombiano.», 2010.
- [3] HTML: Lenguaje de etiquetas de hipertexto | MDN. (2022, 30 noviembre). <https://developer.mozilla.org/es/docs/Web/HTML>
- [4] González, R., Hidalgo, G., Salazar, J. G. y Preciado, M. de L. Elaboración y validación del instrumento para medir calidad de vida en el trabajo “CVT-GOHISALO”. *Ciencia & Trabajo* (2010), 332-340.
- [5] Npm Inc. (2009, 09 02). *npm Docs. npm Registry*. Retrieved September 1, 2022, from <https://www.npmjs.com/>
- [6] Microsoft. (2012, 10 01). *TypeScript Docs. TypeScript: JavaScript With Syntax For Types*. Retrieved September 1, 2022, from <https://www.typescriptlang.org/>
- [7] OpenJS. (2009, 04 27). *About. Node.js*. Retrieved September 3, 2022, from <https://nodejs.org/es/about/>
- [8] S. o. (Angular), «<https://angular.io/>,» [En línea]. Disponible en: <https://angular.io/>.
- [9] (2021). *JSON Web Tokens. JSON Web Tokens - jwt.io*. Retrieved September 5, 2022, from <https://jwt.io/>
- [10] Microsoft. (2015). *Visual Studio Code. Visual Studio Code - Code Editing. Redefined*. Retrieved September 6, 2022, from <https://code.visualstudio.com/>

[11] *DataGrip: el IDE multiplataforma para bases de datos y SQL, de. (2021, 8 junio). JetBrains. <https://www.jetbrains.com/es-es/datagrip/>*

[12] *The PostgreSQL Global Development Group. (1996). PostgreSQL. PostgreSQL: The world's most advanced open source database. Retrieved September 6, 2022, from <https://www.postgresql.org/>*

[13] *Postman, Inc. (2022). Postman Website. Postman API Platform | Sign Up for Free. Retrieved September 6, 2022, from <https://www.postman.com/>*

[14] *Vega, F. (2019). ¿Qué es Git y para qué sirve? Platzi. Retrieved September 6, 2022, from <https://platzi.com/clases/1557-git-github/20215-que-es-git/>*

[15] *OpenJS. (2010, 05 22). Glosario de Express. Express.js. Retrieved September 3, 2022, from <https://expressjs.com/es/resources/glossary.html>*

[16] *Kruchten, P. (1995). The 4+1 View Model of architecture (Vol. 12). IEEE Software. 10.1109/52.469759*

[17] *"What is AGILE? - What is SCRUM? - Agile FAQ's | Cprime", Cprime, 2021. [Online]. Available: [<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>.]*

[18] *Ishikawa, K. (2013). Diagrama de Ishikawa. Obtenido de http://www.academia.edu/download/45800691/Diagrama_de_Ishikawa.pdf.*

11. ANEXOS

- Anexo 1. Gestión de requerimientos.

Gestión de requerimientos

Autor: Juan Camilo Rodríguez

Narrativa: Debido a la importancia que tiene la calidad de vida laboral para los docentes y trabajadores de la Universidad Santo Tomás Seccional Tunja, en este caso la Facultad de Ingeniería de Sistemas, es importante llevar un control del recurso humano mediante encuestas laborales.

Como se conoce estas encuestas laborales se hacen de manera manual, por lo que se propone realizar una plataforma web, permitirá que el acceso a las encuestas sea rápido y eficiente para los empleados de forma remota.

A continuación, se da una breve descripción de los lineamientos del proyecto adjuntando los requerimientos funcionales y no funcionales del mismo.

Descripción: Recopilar información de los trabajadores mediante encuestas tipo Likert utilizando una plataforma web que permita el seguimiento y administración de los procesos de evaluación del cuestionario Gohisalo.

1. Que se permita ver el estado del cuestionario y sus respectivas preguntas.
2. Que sea posible conocer el estatus del proceso de evaluación(Activo, inactivo, finalizado).
3. Que permita modificar el estado de los procesos notificando al usuario.
4. Que sea posible agregar o quitar usuarios de los procesos.
5. Que permita realizar el seguimiento de los procesos y sus respuestas, por medio de graficas.

REQUERIMIENTOS FUNCIONALES

Identificación de Requerimientos	RF 01	Fecha	26/08/2022
Nombre de requerimiento	Ingreso de Usuario		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero acceder al sistema para poder administrar las encuestas.		
Criterio de aceptación	<ul style="list-style-type: none">• El usuario debe digitar su correo electrónico y contraseña.• El usuario presiona el botón ingresar.		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 02	Fecha	26/08/2022
Nombre de requerimiento	Crear versiones de la encuesta gohisalo.		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero crear versiones de la encuesta gohisalo desde la vista crear encuestas.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de encuestas una vez ingrese al sistema. • El usuario selecciona el ítem “Crear nueva versión” 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 03	Fecha	26/08/2022
Nombre de requerimiento	Listar versiones de la encuesta gohisalo.		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero listar encuestas desde la vista de encuestas.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de encuestas una vez ingrese al sistema. • El usuario selecciona el ítem “Crear nueva versión” 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		

Usuarios	Admin, Usuario Delegado
-----------------	-------------------------

Identificación de Requerimientos	RF 04	Fecha	26/08/2022
Nombre de requerimiento	Modificar/Actualizar Estado de la Encuesta.		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder cambiar estado a las encuestas (activa/inactiva)		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de encuestas una vez ingrese al sistema. El usuario selecciona el icono “play” si esta inactiva o “stop” si esta activa que se encuentra en la tabla de las encuestas. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 05	Fecha	26/08/2022
Nombre de requerimiento	Eliminar Versiones de la Encuesta		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder eliminar las versiones de la encuesta.		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de encuestas una vez ingrese al sistema. El usuario selecciona el icono “papelera” que se encuentra en la tabla de las encuestas. 		
Prioridad del	alta		

requerimiento	
Responsable de requerimientos	Juan Camilo Rodríguez
Usuarios	Admin, Usuario Delegado

Identificación de Requerimientos	RF 06	Fecha	26/08/2022
Nombre de requerimiento	Crear/Modificar Preguntas de la Encuesta		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder agregar preguntas a encuestas o modificarlas desde la vista crear encuestas.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de encuestas una vez ingrese al sistema. • El usuario selecciona el icono “Ver” que se encuentra a lado de la encuesta. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 07	Fecha	26/08/2022
Nombre de requerimiento	Eliminar Preguntas de la Encuesta		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder eliminar preguntas de las encuestas.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de encuestas una vez ingrese al sistema. 		

	<ul style="list-style-type: none"> El usuario selecciona el icono “Ver” y seleccionar la pregunta que quiere eliminar
Prioridad del requerimiento	alta
Responsable de requerimientos	Juan Camilo Rodríguez
Usuarios	Admin, Usuario Delegado

Identificación de Requerimientos	RF 08	Fecha	26/08/2022
Nombre de requerimiento	Listar Procesos de Evaluación.		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero listar procesos desde la vista de procesos.		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de procesos una vez ingrese al sistema. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 09	Fecha	26/08/2022
Nombre de requerimiento	Crear/Modificar Proceso de Evaluación		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder crear y modificar procesos desde la vista de procesos.		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de procesos una vez ingrese al sistema. 		

	<ul style="list-style-type: none"> El usuario selecciona el ítem “Crear nuevo proceso” 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		
Identificación de Requerimientos	RF 10	Fecha	26/08/2022
Nombre de requerimiento	Modificar/Actualizar Estado del Proceso		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder cambiar estado de los procesos (activo/finalizado)		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de procesos una vez ingrese al sistema. El usuario selecciona el icono “Play” que se encuentra a lado del proceso para activarlo y el icono “Stop” para finalizarlo. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 11	Fecha	26/08/2022
Nombre de requerimiento	Asignar Usuarios a Proceso de Evaluación.		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero poder asignar usuarios a procesos desde la vista de procesos.		

Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de procesos una vez ingrese al sistema. El usuario selecciona el ítem “ver” El usuario agrega el usuario-proceso 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		
Identificación de Requerimientos	RF 12	Fecha	26/08/2022
Nombre de requerimiento	Notificar al Usuario el Estado del Proceso		
Descripción del requerimiento	El usuario del proceso debe recibir una notificación vía correo del proceso al que hace parte.		
Criterio de aceptación	<ul style="list-style-type: none"> Una vez el usuario es asignado al proceso debe recibir una notificación con el fin de llenar las encuestas. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 13	Fecha	26/08/2022
Nombre de requerimiento	Listar Usuarios por Proceso		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT ver los usuarios que tiene cada proceso.		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de procesos una vez ingrese al sistema. 		

	<ul style="list-style-type: none"> El usuario selecciona el icono “Ver” y podrá ver todos los usuarios del proceso.
Prioridad del requerimiento	alta
Responsable de requerimientos	Juan Camilo Rodríguez
Usuarios	Admin, Usuario Delegado

Identificación de Requerimientos	RF 14	Fecha	26/08/2022
Nombre de requerimiento	Actualizar Usuarios de Proceso de Evaluación		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT actualizar los usuarios que tiene cada proceso.		
Criterio de aceptación	<ul style="list-style-type: none"> El usuario selecciona la pestaña de procesos una vez ingrese al sistema. El usuario selecciona el icono “Ver” y podrá ver todos los usuarios del proceso y poder modificar el usuario que quiera. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 15	Fecha	26/08/2022
Nombre de requerimiento	Eliminar Usuarios de Proceso de Evaluación		

Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT eliminar los usuarios que tiene cada proceso.
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de procesos una vez ingrese al sistema. • El usuario selecciona el icono “Ver” y podrá ver todos los usuarios del proceso y poder eliminar el usuario que quiera.
Prioridad del requerimiento	alta
Responsable de requerimientos	Juan Camilo Rodríguez
Usuarios	Admin, Usuario Delegado

Identificación de Requerimientos	RF 16	Fecha	26/08/2022
Nombre de requerimiento	Eliminar Proceso de Evaluación		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT eliminar procesos de evaluación.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de procesos una vez ingrese al sistema. • El usuario selecciona el icono “papelera” y podrá eliminar procesos. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 17	Fecha	26/08/2022
Nombre de requerimiento	Visualizar Resultados del Proceso en Graficas		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT ver los resultados de los procesos en graficas.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de procesos una vez ingrese al sistema. • El usuario selecciona el icono “grafica” y podrá ver los resultados de los procesos. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	RF 18	Fecha	26/08/2022
Nombre de requerimiento	Crear/Modificar Usuario Delegado.		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero crear usuarios delegados.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de usuarios una vez ingrese al sistema. • El usuario selecciona el botón “crear nuevo usuario” y podrá crear usuarios. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin		

Identificación de Requerimientos	RF 19	Fecha	26/08/2022
Nombre de requerimiento	Eliminar Usuario Delegado		
Descripción del requerimiento	Yo como usuario del sistema GOHIPROJECT quiero eliminar usuarios delegados.		
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario selecciona la pestaña de usuarios una vez ingrese al sistema. • El usuario selecciona el icono “papelera” y podrá borrar usuarios. 		
Prioridad del requerimiento	alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin		

REQUERIMIENTOS NO FUNCIONALES

Identificación de Requerimientos	NRF 01	Fecha	26/08/2022
Nombre de requerimiento	Disponibilidad		
Descripción del requerimiento	El sitio deberá funcionar en cualquier hora y momento por el medio que el usuario desee usarlo.		
Criterio de aceptación	Los usuarios podrán ver el contenido de la página y los empleados podrán modificarla.		
Prioridad del requerimiento	Media		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado, Usuario-Proceso		

Identificación de Requerimientos	NRF 02	Fecha	26/08/2022
Nombre de requerimiento	Usabilidad		
Descripción del requerimiento	El sitio debe ser intuitivo, sencilla, agradable y segura, donde los usuarios la puedan usar con facilidad sin problemas.		
Criterio de aceptación	El usuario representa comodidad al utilizar el sitio		
Prioridad del requerimiento	Media		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado, Usuario-Proceso		

Identificación de Requerimientos	NRF 03	Fecha	26/08/2022
Nombre de requerimiento	Seguridad		
Descripción del requerimiento	Mantener todo tipo de información confidencial, personal e intelectual segura por medio de contraseñas.		
Criterio de aceptación	El equipo de desarrollo será el encargado de ser las únicas personas que pueden modificar los accesos de seguridad del sistema.		
Prioridad del requerimiento	Alta		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado		

Identificación de Requerimientos	NRF 04	Fecha	26/08/2022
Nombre de requerimiento	Eficiencia		
Descripción del requerimiento	El sitio web debe ser capaz de funcionar u operar en cualquier momento y con una cantidad mínima de usuarios que ingrese a este mismo.		
Criterio de aceptación	La información que se modifica del sistema debe ser reflejada de forma inmediata para todos los usuarios que accedan al sistema.		
Prioridad del requerimiento	Media		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado, Usuario-Proceso		

Identificación de Requerimientos	NRF 04	Fecha	26/08/2022
Nombre de requerimiento	Interoperabilidad		
Descripción del requerimiento	El sitio deberá funcionar en distintos tipos de navegadores web y estos tendrán el mismo funcionamiento adecuado para su uso.		
Criterio de aceptación	Los usuarios pueden acceder desde sus navegadores favoritos sin ninguna restricción.		
Prioridad del requerimiento	Baja		
Responsable de requerimientos	Juan Camilo Rodríguez		
Usuarios	Admin, Usuario Delegado, Usuario-Proceso		

- Anexo 2. Scripts de la base de datos.

Enlace: [DDL de la base de datos.](#)

- Anexo 3. Listas de chequeo.

Listas De Chequeo De los Requisitos Del Sistema

Autor: Juan Camilo Rodríguez

Narrativa: De acuerdo al cumplimiento del proyecto, y sus objetivos definidos anteriormente, se llevó a cabo la aplicación de listas de chequeo para verificar que se halla implementado todas las funcionalidades que se especificaron en la recolección de requerimientos.

A continuación, se da una breve descripción de los lineamientos de la ejecución de las listas de chequeo, adjuntando los requerimientos funcionales y no funcionales.

Descripción: Al finalizar la actividad de implementación de requisitos de la aplicación encuestador laboral basado en Gohisalo, seleccione:

- **Cumple:** En caso de evidenciar que el requisito fue correctamente implementado en el sistema de información.
- **No cumple:** En caso de evidenciar que el requisito del sistema de información no fue implementado satisfactoriamente.

REQUERIMIENTOS FUNCIONALES

No.	Requisito del Sistema	Tipo de Requisito	Verificación de Cumplimiento	
			Cumple	No cumple
01	Ingreso de usuario.	Acceso	X	
02	Crear Versiones de la Encuesta Gohisalo.	Gestión de Encuestas	X	
03	Listar Versiones de la Encuesta Gohisalo.	Gestión de Encuestas	X	
04	Modificar/Actualizar Estado de la Encuesta.	Gestión de Encuestas	X	
05	Eliminar Versiones de la Encuesta	Gestión de Encuestas	X	
06	Crear/Modificar Preguntas de la Encuesta	Gestión de Encuestas	X	
07	Eliminar Preguntas de la Encuesta	Gestión de Encuestas	X	

No.	Requisito del Sistema	Tipo de Requisito	Verificación de Cumplimiento	
			Cumple	No cumple
08	Listar Procesos de Evaluación.	Gestión de Procesos	X	
09	Crear/Modificar Proceso de Evaluación	Gestión de Procesos	X	
10	Modificar/Actualizar Estado del Proceso	Gestión de Procesos	X	
11	Asignar Usuarios a Proceso de Evaluación.	Gestión de Procesos	X	
12	Notificar al Usuario el Estado del Proceso	Gestión de Procesos	X	
13	Listar Usuarios por Proceso	Gestión de Procesos	X	
14	Actualizar Usuarios de Proceso de Evaluación	Gestión de Procesos	X	
15	Eliminar Usuarios de Proceso de Evaluación	Gestión de Procesos	X	
16	Eliminar Proceso de Evaluación	Gestión de Procesos	X	
17	Visualizar Resultados del Proceso en Graficas	Gestión de Procesos	X	
18	Crear/Modificar Usuario Delegado.	Gestión de Usuarios	X	
19	Eliminar Usuario Delegado	Gestión de Usuarios	X	

REQUERIMIENTOS NO FUNCIONALES

No.	Requisito del Sistema	Tipo de Requisito	Verificación de Cumplimiento	
			Cumple	No cumple
01	El sitio deberá funcionar en cualquier hora y momento por el medio que el usuario desee usarlo.	Disponibilidad	X	
02	El sitio debe ser intuitivo, sencilla, agradable y segura, donde los usuarios la puedan usar con facilidad sin problemas.	Usabilidad	X	
03	Mantener todo tipo de información confidencial, personal e intelectual segura por medio de contraseñas.	Seguridad	X	
04	El sitio web debe ser capaz de funcionar u operar en cualquier momento y con una cantidad mínima de usuarios que ingrese a este mismo.	Eficiencia	X	
05	El sitio deberá funcionar en distintos tipos de navegadores web y estos tendrán el mismo funcionamiento adecuado para su uso.	Interoperabilidad	X	