



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

TÍTULO PROYECTO

Desarrollo web para la administración de las encuestas realizadas en los eventos de la
USTA Tunja

Presentado por:

Leidy Melissa Marín Fajardo
1.099.217.312
2251490

Fabián Danilo Rodríguez Ochoa
1.051.476.909
2251163

Director

Luz Elena Gutiérrez López

Codirector

Luis Fernando Castellanos Guarín

Tunja
Noviembre de 2022

CONTENIDO

1. FICHA TÉCNICA DEL PROYECTO.....	8
2. PLANTEAMIENTO DEL PROBLEMA	9
3. JUSTIFICACIÓN.....	10
4. OBJETIVOS.....	11
4.1. OBJETIVO GENERAL	11
4.2. OBJETIVOS ESPECÍFICOS.....	11
5. MARCO TEÓRICO	12
5.1. ESTADO DEL ARTE	13
5.2. ARQUITECTURA ORIENTADA A MICROSERVICIOS	15
5.3. MODELO “4+1”	15
5.3.1. VISTA LÓGICA	16
5.3.2. VISTA DE PROCESOS	16
5.3.3. VISTA FÍSICA.....	16
5.3.4. VISTA DE IMPLEMENTACIÓN.....	16
5.4. CASOS DE PRUEBA	17
5.4.1. PRUEBAS DE COMPONENTES	18
5.4.2. PRUEBAS DE INTEGRACIÓN	18
5.4.3. PRUEBAS DEL SISTEMA	18
5.4.4. PRUEBAS DE ACEPTACIÓN	18
5.5. ESTÁNDAR IEEE 830-1998.....	18
5.5.1. REQUERIMIENTOS FUNCIONALES.....	19
5.5.2. REQUERIMIENTOS NO FUNCIONALES.....	19
6. METODOLOGÍA	20
6.1. FASES DE LA METODOLOGÍA.....	20
6.1.1. LEVANTAMIENTO DE REQUERIMIENTOS	20
6.1.1.1 Elicitación de requerimientos	21
6.1.1.2 Especificación.....	21

6.1.1.3	Validación.....	21
6.1.2.	DISEÑO	21
6.1.3.	CONSTRUCCIÓN DEL SISTEMA	22
6.1.3.1	Backend.....	22
6.1.3.2	Frontend.....	22
6.1.4.	VALIDACIONES	22

7. CRONOGRAMA DE ACTIVIDADES.....¡ERROR! MARCADOR NO DEFINIDO.

8. DESARROLLO DEL PROYECTO 24

8.1. FASE 1. LEVANTAMIENTO DE REQUERIMIENTOS 24

8.1.1. ESPECIFICACIÓN DE REQUERIMIENTOS 24

8.1.2. VALIDACIÓN DE REQUERIMIENTOS 24

8.2. FASE 2. DISEÑO DEL SISTEMA 25

8.2.1. DISEÑO DE DIAGRAMAS 25

8.2.1.1 Diagramas de casos de uso 25

8.2.1.2 Diagrama de clases..... 26

8.2.1.3 Diagrama de actividades..... 27

8.2.2. DISEÑO DE LA BASE DE DATOS 31

8.2.3. DISEÑO DE LA INTERFAZ 32

8.3. FASE 3. CONSTRUCCIÓN DEL SISTEMA 34

8.3.1. SELECCIÓN DE LAS TECNOLOGÍAS 34

8.3.2. OTRAS HERRAMIENTAS..... 35

8.3.3. CONSTRUCCIÓN DE LA BASE DE DATOS..... 37

8.3.3.1 Sentencias DDL para la construcción de la base de datos 38

8.3.3.2 Sentencias DML para la inserción de información a la base de datos..... 38

8.3.4. CONSTRUCCIÓN DEL BACKEND..... 40

8.3.4.1 Librerías..... 40

8.3.4.2 Herramientas 41

8.3.4.3 Estructura..... 41

8.3.4.4 Comandos 44

8.3.5. CONSTRUCCIÓN DEL FRONTEND..... 45

8.3.5.1 Estructura.....	45
8.3.5.2 Comando.....	50
8.3.5.3 Servicios y API.....	50
8.4. FASE 4. PRUEBAS DEL PROTOTIPO DESARROLLADO	52
8.4.1. PRUEBAS DE FUNCIONALIDAD.....	52
8.4.2. PRUEBA DE ACEPTACIÓN	60
<u>9. CONCLUSIONES Y ANÁLISIS DE RESULTADOS</u>	<u>63</u>
<u>10. REFERENCIAS.....</u>	<u>64</u>
<u>11. ANEXOS.....</u>	<u>67</u>
11.1. ANEXO 1. DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS	67
11.2. ANEXO 2. MANUAL DE USUARIO.....	89
11.3. ANEXO 3. DISEÑO DE MOCKUPS	113
11.4. ANEXO 4: SENTENCIAS DDL PARA LA CONSTRUCCIÓN DE LA BASE DE DATOS	123
11.5. ANEXO 5: ACCESO A REPOSITORIO	130

ÍNDICE DE TABLAS

Tabla 1 Ficha técnica.....	8
Tabla 2 Objetivos específicos	11
Tabla 3 Cronograma de actividades	¡Error! Marcador no definido.
Tabla 4 Fase 1. Levantamiento de requerimientos	24
Tabla 5 Fase 2 Diseño del prototipo	33
Tabla 6 Fase 2.1 Selección de las tecnologías.....	35
Tabla 7 Fase 2.2 Versión tecnologías utilizadas	35
Tabla 8 Fase 3 Desarrollo del prototipo	52
Tabla 9 Fase 4 Pruebas.....	62

ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1 Modelo "4+1"</i>	15
<i>Ilustración 2 Proceso General Del Testing</i>	17
<i>Ilustración 3 Modelo evolutivo</i>	20
<i>Ilustración 4 Caso de uso: Administrador</i>	25
<i>Ilustración 5 Caso de uso: Docente, Invitado y Secretaria</i>	25
<i>Ilustración 6 Caso de uso: Estudiante</i>	26
<i>Ilustración 7 Diagrama de clases</i>	27
<i>Ilustración 8 Diagrama de actividades: Registrar, iniciar sesión y crear encuestas</i>	28
<i>Ilustración 9 Diagrama de actividades: Listar encuestas, ver resultados y responder encuestas</i>	28
<i>Ilustración 10 Diagrama de actividades: Filtrar búsquedas, editar perfil y agregar usuarios</i>	29
<i>Ilustración 11 Diagrama de actividades: Listar, modificar usuarios y listar roles</i>	30
<i>Ilustración 12 Diagrama de actividades: Editar roles</i>	31
<i>Ilustración 13 Modelo entidad relación de la base de datos</i>	32
<i>Ilustración 14 Historial de cambios de la rama Origin en GitKraken</i>	36
<i>Ilustración 15 Repositorio GitHub del proyecto</i>	36
<i>Ilustración 16 Base de datos PgAdmin</i>	37
<i>Ilustración 17 Sentencia DML Creación de roles</i>	38
<i>Ilustración 18 Sentencia DML Creación de dependencias</i>	39
<i>Ilustración 19 Sentencia DML Creación de tipo dependencias</i>	39
<i>Ilustración 20 Sentencia DML Creación de imágenes</i>	39
<i>Ilustración 21 Sentencia DML Creación de usuarios</i>	40
<i>Ilustración 22 Sentencia DML Creación de acceso</i>	40
<i>Ilustración 23 Sentencia DML Creación tipos de eventos</i>	40
<i>Ilustración 24 Sentencia DML Creación tipo preguntas</i>	40
<i>Ilustración 25 Herramienta Postman</i>	41
<i>Ilustración 26 Uso librería Express</i>	42
<i>Ilustración 27 Estructura backend</i>	43
<i>Ilustración 28 Configuración conexión BD</i>	43
<i>Ilustración 29 Estructura del frontend Modelos</i>	45
<i>Ilustración 30 Estructura del frontend Contenedor</i>	47

<i>Ilustración 31 Estructura del frontend Módulo Privado Encuestas.....</i>	<i>47</i>
<i>Ilustración 32 Estructura frontend Módulo Privado Estudiante</i>	<i>48</i>
<i>Ilustración 33 Estructura del frontend Módulo Privado Inicio.....</i>	<i>48</i>
<i>Ilustración 34 Estructura del frontend Módulo Privado Roles</i>	<i>48</i>
<i>Ilustración 35 Estructura del frontend Módulo Privado Usuarios</i>	<i>49</i>
<i>Ilustración 36 Estructura del frontend Modulo Público.....</i>	<i>49</i>
<i>Ilustración 37 Estructura frontend servicios.....</i>	<i>49</i>
<i>Ilustración 38 Estructura utilidades</i>	<i>50</i>
<i>Ilustración 39 Ejemplo Servicio Opciones</i>	<i>51</i>
<i>Ilustración 40 Ejemplo de observable</i>	<i>51</i>
<i>Ilustración 41 Validar token.....</i>	<i>52</i>
<i>Ilustración 42 Prueba Funcional Login</i>	<i>53</i>
<i>Ilustración 43 Prueba agregar nuevos usuarios</i>	<i>53</i>
<i>Ilustración 44 Prueba listar usuarios</i>	<i>54</i>
<i>Ilustración 45 Prueba modificar usuarios</i>	<i>54</i>
<i>Ilustración 46 Prueba modificar acceso usuario.....</i>	<i>54</i>
<i>Ilustración 47 Prueba listar roles.....</i>	<i>55</i>
<i>Ilustración 48 Prueba modificar estado rol</i>	<i>55</i>
<i>Ilustración 49 Prueba buscador listar encuesta.....</i>	<i>55</i>
<i>Ilustración 50 Prueba crear encuesta.....</i>	<i>56</i>
<i>Ilustración 51 Prueba listar encuestas</i>	<i>57</i>
<i>Ilustración 52 Prueba modificar información encuesta</i>	<i>57</i>
<i>Ilustración 53 Prueba crear pregunta encuesta.....</i>	<i>57</i>
<i>Ilustración 54 Prueba modificar preguntas encuesta.....</i>	<i>58</i>
<i>Ilustración 55 Prueba buscador listar encuesta.....</i>	<i>58</i>
<i>Ilustración 56 Prueba responder encuesta.....</i>	<i>59</i>
<i>Ilustración 57 Prueba resultados encuesta</i>	<i>59</i>
<i>Ilustración 58 Prueba buscador listar usuarios.....</i>	<i>59</i>
<i>Ilustración 59 Prueba registrar usuarios nuevos</i>	<i>60</i>
<i>Ilustración 60 Ingreso al aplicativo</i>	<i>113</i>
<i>Ilustración 61 Crear cuenta.....</i>	<i>114</i>
<i>Ilustración 62 Inicio.....</i>	<i>115</i>
<i>Ilustración 63 Listar usuarios</i>	<i>115</i>

<i>Ilustración 64 Crear usuario</i>	116
<i>Ilustración 65 Modificar usuario</i>	116
<i>Ilustración 66 Listar roles</i>	117
<i>Ilustración 67 Modificar rol</i>	117
<i>Ilustración 68 Editar perfil</i>	118
<i>Ilustración 69 Listar encuestas</i>	118
<i>Ilustración 70 Crear encuestas</i>	119
<i>Ilustración 71 Modificar encuesta</i>	120
<i>Ilustración 72 Formulario responder encuesta</i>	121
<i>Ilustración 73 Resultados de la encuesta</i>	122

1. FICHA TÉCNICA DEL PROYECTO

Tabla 1 Ficha técnica

Título	Desarrollo web para la administración de las encuestas realizadas en los eventos de la USTA Tunja
Autor (es)	Leidy Melissa Marín Fajardo y Fabián Danilo Rodríguez Ochoa
Directores del proyecto	Luz Elena Gutiérrez López Luis Fernando Castellanos Guarín
Palabras claves	Encuestas, análisis, estadísticas.
Descripción	<p>Este sistema de información orientado a la web buscó satisfacer las necesidades de la Universidad Santo Tomás Seccional Tunja, en torno a la complejidad que se presentaba al evaluar la información obtenida de las encuestas en los eventos realizados por parte de la Universidad.</p> <p>Dicho desarrollo permitió que la información recopilada acerca de estos eventos quedará almacenada y se pudiera manejar de manera sistematizada, de esta forma se logró un mejor control de los resultados que dejaban las encuestas de estos eventos, y así, se pudieron realizar análisis estadísticos, y se generaron sus respectivos reportes.</p>

Fuente: Autor

2. PLANTEAMIENTO DEL PROBLEMA

Actualmente los sistemas de información son de gran utilidad en pequeñas o grandes empresas, instituciones, u organizaciones, ya que dichos sistemas ayudan a administrar, recolectar, analizar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada entidad.

Al día de hoy, la Universidad Santo Tomás Seccional Tunja, presenta una complejidad al momento de evaluar la información obtenida de las encuestas que se realizan en los distintos eventos, ya que la información recopilada de estos eventos es manipulada o tratada de forma manual a través de archivos en formato Excel, lo cual genera una gran cantidad de documentación que ocasiona que este proceso sea más confuso.

Es por este motivo, que en muchas ocasiones los reportes de cada evento no están unificados en un solo sistema de información o base de datos, y al momento de comparar y consolidar la información, puede haber varias versiones de estos archivos, puesto que no se cuenta con permisos y roles definidos para que solo el personal autorizado pueda tener acceso a esta información.

Además, esto dificulta realizar búsquedas más específicas, debido a la escasa disponibilidad de información que pueden tener los usuarios en tiempo real, haciendo que se dedique demasiado tiempo del personal de la Universidad para preparar y analizar datos, lo que ocasiona un gasto elevado en tiempo y recursos económicos por parte de la Universidad.

3. JUSTIFICACIÓN

La implementación de sistemas de información, brinda la posibilidad de obtener grandes ventajas, puesto que permite que los usuarios puedan acceder a la información de manera rápida y oportuna, ayuda a establecer estrategias y planificaciones operativas, apoya la monitorización de las operaciones, y también favorece la reducción de costos.

Para ello, es necesario un sistema de información orientado a la web que ofrezca múltiples posibilidades, permitiendo acceder a los datos relevantes de manera frecuente y oportuna, donde la información se pueda almacenar y administrar de manera sistemática. Así los usuarios autorizados para la manipulación de la información podrán planificar operaciones y establecer objetivos.

Es por este motivo que el sistema de información orientado a la web, que se propone en este desarrollo, busca satisfacer las necesidades de la Universidad Santo Tomás Seccional Tunja, en torno a la complejidad que se presenta al evaluar la información obtenida de las encuestas en los eventos realizados por parte de la Universidad.

Dicho desarrollo permitiría que la información recopilada acerca de estos eventos quede almacenada y se pueda manejar de manera sistematizada, logrando así, que haya un mejor control de los resultados que dejan las encuestas de estos eventos, y de esta manera, se puedan realizar análisis estadísticos, generando sus respectivos reportes.

Además, con esta solución propuesta, se evitaría la excesiva documentación que actualmente se requiere para analizar y obtener los resultados de cada evento. Por otra parte, se definirían roles para la manipulación de la información, con el fin de que solo el personal autorizado pueda tener acceso y así evitar confusiones al momento de consolidar la información, teniendo en cuenta que además de todo esto, la información podrá ser consultada de manera oportuna y en tiempo real, para que los usuarios autorizados puedan hacer las búsquedas o consultas que necesiten. Y lo más importante, es que la información estará almacenada de manera segura, evitando que esté expuesta a cualquier manipulación por parte de personal no autorizado.

4. OBJETIVOS

4.1. Objetivo General

Construir una herramienta informática para la generación de instrumentos de evaluación de eventos realizados en la Universidad Santo Tomás Seccional Tunja, utilizando la arquitectura orientada a microservicios para su implementación.

4.2. Objetivos específicos

Tabla 2 Objetivos específicos

Nro.	Objetivo específico
1	Construir la especificación de requerimientos funcionales y no funcionales de la herramienta informática, utilizando el estándar IEEE 830-1998.
2	Diseñar la herramienta informática para establecer las vistas de usuario que permitan analizar el software a desarrollar, utilizando el modelo de referencia "4+1" propuesto por Kruchten.
3	Implementar la herramienta informática de generación de instrumentos de evaluación, definiendo dos componentes: backend y frontend, utilizando la arquitectura orientada a microservicios
4	Validar la herramienta informática diseñando casos de prueba, que permitan identificar las funcionalidades de acuerdo con los requerimientos del desarrollo.

Fuente: Autor

5. MARCO TEÓRICO

La evaluación de eventos es un proceso de observación detallada, medición absoluta y seguimiento puntualizado del desarrollo de un evento con el fin de valorar sus resultados con exactitud. (CARCAMO et al., 6)

Se trata de un proceso continuo que ocurre a través de todas las etapas del ciclo de vida de un evento. Permite la creación de estrategias que se utilizan para identificar y puntualizar las características básicas, y estadísticas clave de determinado evento. También es una herramienta o ayuda de retroalimentación que se debe proporcionar a los distintos grupos involucrados en el evento, jugando así un papel importante en el proceso de gestión, gracias a su característica de ser una poderosa herramienta de información que facilita el proceso de análisis. (CARCAMO et al., 6)

Las tres etapas básicas del proceso de evaluación de eventos son:

- Estudio de viabilidad o fase de evaluación previa al evento, que debe realizarse antes del evento para determinar, con un mínimo margen de error, si su inicio es viable o no. (CARCAMO et al., 6)
- Proceso de seguimiento y control, que ocurre durante la ejecución del evento para verificar que todo esté funcionando de acuerdo con el cronograma programado y, cuando sea necesario, se puedan tomar las acciones correctivas para modificar las fallas y omisiones del evento que se detecten. (CARCAMO et al., 6)
- La evaluación posterior al evento se enfoca en medir los resultados del evento en términos de impacto, imagen y satisfacción del cliente o usuario. De esta fase se toma información para indicar acciones de mejora continua para versiones posteriores del evento o para nuevos eventos futuros con características similares. (CARCAMO et al., 6)

Una plataforma puede facilitar el proceso de recolección de comentarios y opiniones de los participantes de un evento. Las encuestas de retroalimentación, de experiencia general y los diferentes tipos de preguntas alentarán a los participantes a compartir comentarios de una manera conveniente a través de sus dispositivos. (Wong, 2018)

Estas plataformas cumplen con factores claves para garantizar su buen funcionamiento, algunas de esos factores son:

1. Informes y análisis centralizados: Buscar informes, incluidos los datos de todas las opiniones de los participantes y hacer un análisis valioso determinará la calidad de la tecnología que atraerá a los participantes, además de aprender qué informes están disponibles y cómo pueden ser exportados. (Wong, 2018)
2. Integraciones: Si su equipo quiere crear una integración entre sus sistemas y plataformas de gestión de eventos, pídale al proveedor información acerca de los costos para acceder a una API. (Wong, 2018)
3. Opciones de seguridad y privacidad: Para mantener la seguridad de datos de los eventos, haga preguntas sobre la capacidad de la plataforma, su control de acceso, lista de registros, etc. Esto también puede ser importante para las políticas de privacidad y los Términos de uso, con el objetivo de que los participantes puedan entender cómo se utilizan sus datos antes de completar el registro o participar en una aplicación. (Wong, 2018)
4. Alojamiento de documentos: Pregunte acerca de la capacidad de organizar y compartir presentaciones, documentos, imágenes, mapas y más para mantener a los asistentes informados, educados y comprometidos durante su evento. Conozca los distintos tipos de archivos que se pueden descargar. (Wong, 2018)

5.1. Estado del arte

El software de encuestas en línea y las herramientas de cuestionarios aparecieron por primera vez a fines de la década de los 90. Estas son herramientas de encuestas bastante tradicionales que consisten en una larga lista de preguntas y respuestas anónimas. Sin embargo, desde el auge de las opiniones de los sitios web, las herramientas de encuestas en línea han evolucionado. Su continua popularidad también se debe al hecho de que las encuestas en línea son una manera fácil de contactar a una audiencia definida y obtener sus respuestas.

A continuación se presentan algunos softwares, los cuales fueron seleccionados porque ofrecen soluciones similares respecto a la administración de encuestas realizadas en eventos.

- Mopinion: Es una herramienta de análisis para la opinión de los clientes que hace uso de encuestas en línea, y ofrece oportunidades de análisis en profundidad para sus usuarios. Con la visualización de datos en tiempo real, en paneles de control y

gráficos personalizados, donde los usuarios pueden manejar grandes cantidades de datos de forma rápida y eficiente. (Gilliam Haije, 2017)

Sitio web: <https://mopinion.com/es/>

- Typeform: Es un software de encuestas en línea con una interfaz muy fácil de usar que permite a los usuarios construir sus propias encuestas. Como encuestado, el proceso para responder a una de estas encuestas es bastante fluido. Por otra parte, también incluye funciones como preguntas y respuestas ilimitadas, opciones para exportar datos, temas diseñados a petición, así como plantillas listas para utilizar y funciones básicas de generación de informes. (Gilliam Haije, 2017)

Sitio web: <https://www.typeform.com/>

- Survicate: Permite a los usuarios activar encuestas específicas en función de las visitas a diferentes partes de su sitio web. Los usuarios cuentan con la opción de enviar cuestionarios por correo electrónico. Cuando se trata de crear encuestas, Survicate ofrece una biblioteca de encuestas prefabricadas para que elijas. Esta herramienta de encuesta también incluye funcionalidad de panel, análisis NPS y opciones de exportación CSV y XLS. (Gilliam Haije, 2017)

Sitio web: <http://www.survicate.com>

- PollDaddy: Se trata de un software de encuestas en línea que permite a los usuarios investigar de dos maneras diferentes: investigan en su página web o invitar a la gente a responder por correo electrónico. Cuando los datos se han recopilado, se pueden establecer filtros para analizar y utilizarlos en los informes. Los resultados de estas encuestas se pueden ver en tiempo real, además de poderse exportar a archivos PDF, Excel, CSV, Google Docs y XML. (Gilliam Haije, 2017)

Sitio web: <http://www.polldaddy.com>

- Qualtrics: Es un software de encuestas en línea y una herramienta de cuestionario para organizar de manera mejorada las encuestas tradicionales. Proporciona más de 100 tipos de preguntas, algunos modelos de encuestas basados en trabajo, así como funciones lógicas, activistas de correo electrónico y bifurcación lógica. Los usuarios pueden seleccionar entre más de 30 tipos de gráficos diferentes para añadirlos a informes web en tiempo real y exportar resultados en CSV o SPSS (Gilliam Haije, 2017).

Sitio web: <http://www.qualtrics.com>

5.2. Arquitectura orientada a microservicios

Esta arquitectura se basa en el desarrollo de una aplicación que consta de muchos componentes pequeños, independientes y livianos con su propia funcionalidad, por lo cual se requiere una API REST HTTP para llamar a estos microservicios. (Abraham Hernández, 2019, p. 9)

El desarrollo actual se refiere al uso de procesos basados en el desarrollo ágil. En otras palabras, su aplicación debería ser más rápida. El uso de microservicios se ha vuelto muy popular en el campo del desarrollo de software y está cambiando la forma en que se perciben y diseñan las aplicaciones. (Abraham Hernández, 2019, p. 10)

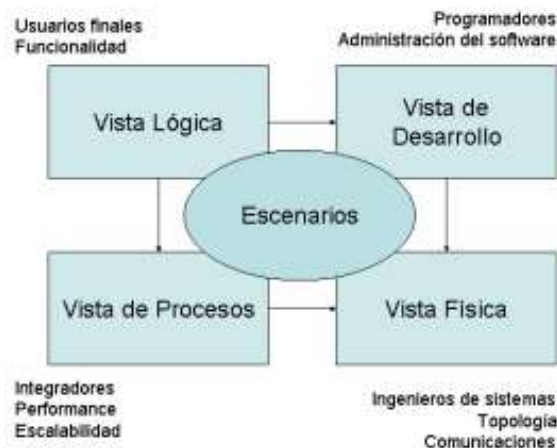
Este tipo de arquitectura requiere la división de la lógica en componentes o servicios más pequeños que contienen construcciones ligeras que permiten la integración con los procesos existentes. Su objetivo es crear microservicios que operen independientemente de otros servicios y se organicen de acuerdo con las funciones que realizan. (Abraham Hernández, 2019, p. 10)

Según Mazzara et al. (2018), los criterios principales que rigen el uso de microservicios son:

1. Un contexto delimitado que consiste en una combinación de las mismas funciones. (Abraham Hernández, 2019, p. 11)
2. El tamaño es una característica que caracteriza a los microservicios, por lo que el objetivo es hacerlos lo más pequeños posible, para poder cambiar en cualquier momento, y no afectar otras funcionalidades. (Abraham Hernández, 2019, p. 11)
3. Independencia, la cual es responsable del acoplamiento que existe entre ellos. (Abraham Hernández, 2019, p. 11)

5.3. Modelo “4+1”

Ilustración 1 Modelo "4+1"



Fuente: Kruchten, P. (1995).

Esta ilustración muestra uno de los modelos más reconocidos es el “4+1” de Philippe Kruchten. El cual identifica cuatro vistas diferentes: lógica, proceso, física e implementación.

5.3.1. Vista lógica

El propósito de esta vista es la descomposición orientada a objetos. En esta vista, se representa la implementación de casos de uso, subsistemas, paquetes y clases de los casos de uso más importantes en términos de arquitectura. (Moreno Sabido y Ucán Pech, s. f.)

5.3.2. Vista de procesos

El propósito de esta vista es la concurrencia y la coordinación de procesos, la integridad del sistema, la tolerancia a fallas y cómo las abstracciones clave de la vista lógica encajan en la arquitectura. (Moreno Sabido y Ucán Pech, s. f.)

5.3.3. Vista Física

Muestra la ubicación del software en el hardware. Los elementos definidos en la vista lógica se copian a los componentes del software (servicios, procesos, etc.) o componentes de hardware, los cuales definen con mayor precisión cómo se debe implementar la solución. (Moreno Sabido y Ucán Pech, s. f.)

5.3.4. Vista de implementación

Por último, la vista de implementación, la cual se descompone en subsistemas, describe la estructura del entorno de desarrollo; el software es asociado en pequeños subsistemas

que luego pueden ser desarrollados por uno o varios desarrolladores; Estos subsistemas tienen la forma de programas o bibliotecas organizados en capas y jerarquías. (Moreno Sabido y Ucán Pech, s. f.)

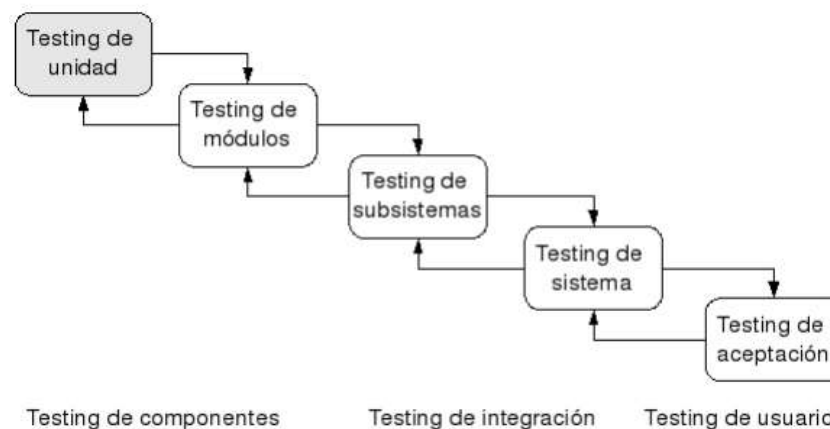
5.4. Casos de prueba

Para desarrollar un software de calidad y sin errores, la organización y los casos de prueba son muy importantes.

Según la norma ISO / IEC / IEEE 24765: 2010, al momento de diseñar o establecer los casos de prueba, se debe tener en cuenta lo siguiente:

- Verificación: Proceso de evaluación de un sistema o componente para determinar si un producto en una determinada etapa de desarrollo cumple con las condiciones establecidas al comienzo de esa fase. (Mera Paz, 2016)
- Validación: Proceso de evaluación de un sistema o componente, durante o al final del desarrollo, para establecer cuándo se están cumpliendo los requisitos específicos. (Mera Paz, 2016)

Ilustración 2 Proceso General Del Testing



Fuente: Sommerville, I. (1995)

Esta ilustración evidencia que el proceso general del Testing utiliza un conjunto finito de casos de prueba, comparando el comportamiento al momento de la ejecución contra el comportamiento esperado del desarrollo.

5.4.1. Pruebas de componentes

Su propósito es identificar fallas y verificar el comportamiento de módulos de software, programas, objetos, clases, etc. Que se pueden probar por separado; es decir, que se pueden ejecutar independientemente del resto del sistema, dependiendo del contexto. (Mera Paz, 2016)

5.4.2. Pruebas de integración

Son responsables de probar interfaces entre componentes o módulos; por ejemplo, autenticación de usuario del componente con el sistema operativo, el sistema de archivos integrado con el hardware, etc. (Mera Paz, 2016)

5.4.3. Pruebas del sistema

Se refieren a todo el sistema; se debe establecer un plan de prueba de manera clara y bien estructurada. (Mera Paz, 2016)

5.4.4. Pruebas de aceptación

Se enfocan en aceptar los criterios proporcionados en el contrato de desarrollo de software, que se acuerda entre el fabricante del software y el cliente. (Mera Paz, 2016)

5.5. Estándar IEEE 830-1998

El Comité de Estándares de Ingeniería de Software y Sistemas del IEEE (S2ESC) está constituido por el Consejo de Actividades de Estándares de la Sociedad Informática del IEEE para codificar las normas de las prácticas profesionales de ingeniería de software en estándares, incluyendo la estandarización de procesos, productos, recursos, notaciones, métodos, nomenclaturas, técnicas y soluciones para la ingeniería de software y sistemas dependientes de software. (IEEE Computer Society 2008)

El S2ESC hace referencia al uso de normas de ingeniería del software profesionales educadores y clientes. El S2ESC armoniza el desarrollo de normas nacionales e internacionales de ingeniería del software y promueve la disciplina y la profesionalización de la ingeniería del software. El S2ESC también promueve la coordinación con otras iniciativas del IEEE. (IEEE Computer Society 2008-1)

Sustituido por ISO/IEC/IEEE 29148:2011. Explica el contenido y las cualidades que se debe tener para una buena especificación de requisitos de software (SRS) y se presentan varios

ejemplos de esquemas de SRS. Esta práctica recomendada está destinada a especificar los requisitos del software que se va a desarrollar, pero también puede aplicarse para ayudar en la selección de productos de software internos y comerciales. También se proporcionan directrices para el cumplimiento de la norma IEEE/ EIA12207.1- 1997. (IEEE Computer Society 2008-2)

5.5.1. Requerimientos funcionales

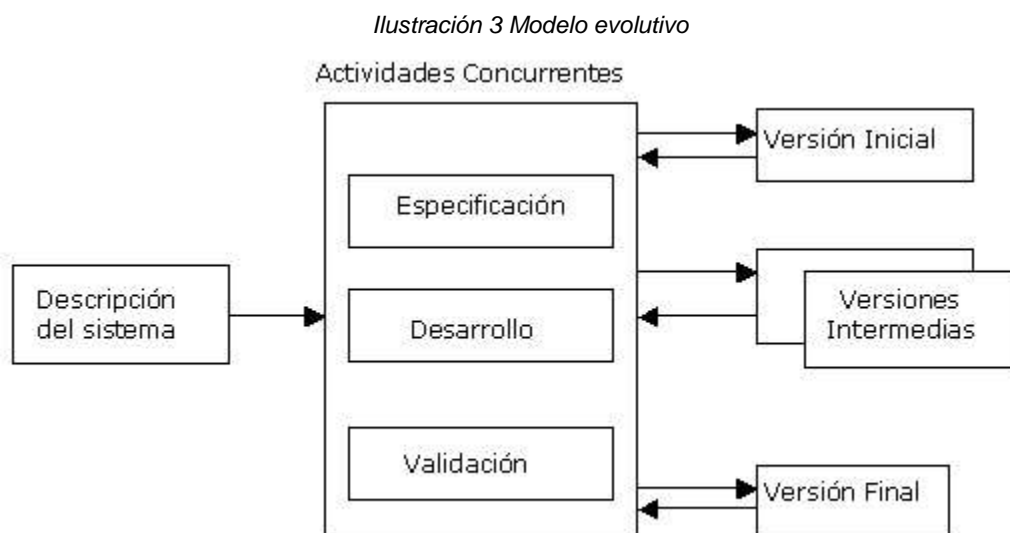
Los requerimientos funcionales son aquellos que definen las funciones que el sistema deberá realizar, mencionan las transformaciones que el sistema efectúa sobre las entradas para producir salidas. Es significativo que se describa el ¿Qué? y no el ¿Cómo? se debe hacer dichas transformaciones. Estos requerimientos a medida que avanza el proyecto de software se convierten en algoritmos, la lógica y gran parte del código del sistema (Arias Chaves, 2005).

5.5.2. Requerimientos no funcionales

Los requerimientos no funcionales hacen referencia a las características que de una u otra manera puedan limitar el sistema, como, por ejemplo, el rendimiento, interfaces de usuario, robustez del sistema, disponibilidad de equipo, mantenimiento, seguridad, esquemas, modelos, entre otros (Arias Chaves, 2005).

6. METODOLOGÍA

Para el desarrollo del proyecto se propone la metodología de desarrollo Prototipado. La cual se basa en la creación de una implementación parcial del sistema para estudiar los requerimientos del mismo. El prototipo se fabricará lo antes posible. Esto se entrega a los usuarios, clientes o sus representantes para que ellos puedan explorarlo y manipularlo. Luego, estas personas brindan comentarios sobre lo que les gusta y lo que no les gusta del prototipo propuesto, y en seguida, se toman en cuenta los aportes que los usuarios han proporcionado para desarrollar el sistema real (Repositorio Digital de la Facultad de Ingeniería UNAM, s. f.).



Fuente: Sommerville / Ingeniería del Software

En la ilustración anterior se puede ver como la metodología de Prototipado comienza con una serie de requisitos, posterior a esto se desarrollan una serie de prototipos, se exponen al usuario y se van mejorando paso a paso.

6.1. Fases de la metodología

Este desarrollo de software se realizará mediante las siguientes fases:

6.1.1. Levantamiento de requerimientos

Esta etapa se desarrollará a través de entrevistas con la facultad de Ingeniería de Sistemas de la Universidad Santo Tomás Seccional Tunja. Dichas entrevistas se van a realizar por medio de tareas previas, como: estudiar el dominio del problema, seleccionar a las

personas que se van a entrevistar, determinar el objetivo y el contenido de las reuniones, planificar la entrevista, llevarla a cabo y finalmente, realizar un análisis de la información recolectada en la entrevista.

Para el proceso del planteamiento de los requerimientos del software a desarrollar, y con el fin de entregar lo que la Ingeniería de Sistemas de la Universidad Santo Tomás Seccional Tunja desea, se establecen las siguientes etapas [Loucopoulos y Karakostas]: (INGENIERÍA DE REQUERIMIENTOS., s. f.)

6.1.1.1 Elicitación de requerimientos

El objetivo de la elicitación de requisitos es adquirir información relevante sobre el problema que se desea resolver, dicha información se utilizará para crear una especificación formal del software necesario para resolver el problema. Es por esto que es muy importante que haya una buena comunicación entre el cliente y los desarrolladores, para poder entender las necesidades del interesado. Para llevar a cabo la elicitación de requerimientos se plantea realizar unas entrevistas con el usuario (INGENIERÍA DE REQUERIMIENTOS., s. f.)

6.1.1.2 Especificación

La especificación de requerimientos puede definirse como un contrato entre el usuario y los desarrolladores del software, donde se describe el comportamiento funcional que se espera del software a desarrollar. Dicha especificación de requerimientos se desarrollará por medio de casos de uso, puesto que, estos casos de uso son una técnica que permite especificar el comportamiento del sistema, y la secuencia de interacciones entre el sistema y los actores (INGENIERÍA DE REQUERIMIENTOS., s. f.)

6.1.1.3 Validación

Por medio de la validación de requerimientos, se busca confirmar que el modelo de solicitud coincide con la intención del cliente y del usuario; este es un punto de vista más general, ya que se trata de asegurar que las opiniones y los conceptos que se presentaron en la descripción queden claros y explicados (INGENIERÍA DE REQUERIMIENTOS., s. f.)

6.1.2. Diseño

Para el desarrollo de esta fase se definen las siguientes actividades:

- Diseño de diagramas UML (Diagrama de clases y diagrama de actividades)
- Diseño de la base de datos
- Diseño de la interfaz

6.1.3. Construcción del sistema

Para la construcción del sistema informático, se tienen en cuenta dos componentes: Backend y Frontend, utilizando la arquitectura orientada a servicios para su implementación.

6.1.3.1 Backend

Se plantea desarrollar el backend con una arquitectura orientada a servicios, utilizando Node como entorno de ejecución y typescript como lenguaje de programación. Es un marco para desarrollar programas concurrentes de alto rendimiento que no se basa en el enfoque convencional de subprocesos múltiples, sino que utiliza E / S asincrónicas con un modelo de programación controlado por eventos. (Tilkov y Vinoski, 2010)

6.1.3.2 Frontend

El desarrollo del frontend se realizará en Angular bajo el lenguaje de typescript. Esta herramienta es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de una sola página eficientes y complejas. (Angular, 2021)

Angular ofrece:

- Marco basado en componentes para crear aplicaciones web escalables. (Angular, 2021)
- Una colección bien integrada de bibliotecas que cubren una amplia gama de funciones que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor, entre otros (Angular, 2021).
- Un conjunto de herramientas de desarrollo para ayudarlo a desarrollar, compilar, probar y actualizar su código (Angular, 2021).

6.1.4. Validaciones

Para validar la herramienta informática se diseñarán casos de prueba, que permitan identificar las funcionalidades de acuerdo con los requerimientos del desarrollo. Estas validaciones incluyen:

- Prueba de Componentes: Para identificar fallas y verificar el comportamiento de los módulos del sistema
- Prueba de Integración: Para probar interfaces entre componentes o módulos del Sistema.
- Prueba de Sistema: Para analizar defectos globales o para estudiar aspectos específicos de su comportamiento.
- Prueba de Aceptación: Para aceptar los criterios proporcionados en el contrato de desarrollo de software, que se acuerda entre el o los desarrolladores del software y el cliente-usuario.

7. DESARROLLO DEL PROYECTO

En esta sección se detallan cada una de las etapas realizadas para completar el proyecto.

7.1. Fase 1. Levantamiento de requerimientos

Esta fase se desarrolló a través de entrevistas con el Stakeholder encargado de la facultad de ingeniería de sistemas de la seccional Tunja. Esto con el fin de recolectar información para determinar los objetivos del proyecto a desarrollar. Esta etapa está compuesta de 2 actividades:

7.1.1. Especificación de requerimientos

El primer acercamiento se realizó por medio de entrevistas con la decana de la facultad de ingeniería de sistemas, esto con el fin de adquirir información relevante sobre el problema que se desea resolver, esta información se utilizó para crear una especificación formal del software necesario para dar solución al problema.

Una vez reunida y consolidada la información obtenida de estas entrevistas, se procedió a crear el documento de especificación de requerimientos con el formato IEEE 830, para especificar el comportamiento del sistema, y la secuencia de interacciones entre el sistema y los actores ([Ver anexo 1](#)).

7.1.2. Validación de requerimientos

Por medio de la validación de requerimientos, se buscó confirmar que el modelo de solicitud coincide con la intención del cliente y del usuario; este es un punto de vista más general, ya que se trata de asegurar que las opiniones y los conceptos que se presentaron en la descripción queden claros y explicados.

Tabla 3 Fase 1. Levantamiento de requerimientos

Tipo	Elementos
Entradas	<ul style="list-style-type: none">• Actas de reunión
Instrumentos y/o herramientas	<ul style="list-style-type: none">• Conexión a internet• Computador• Programas del computador<ul style="list-style-type: none">✓ Navegador Web✓ Word
Salidas	Documento de especificación de requerimientos
Tiempo	1 semana

Fuente: Autor

7.2. Fase 2. Diseño del sistema

El desarrollo de la fase de diseño estuvo compuesta por tres actividades: el diseño de diagramas, el diseño de la base de datos y el diseño de la interfaz

7.2.1. Diseño de diagramas

Para esta actividad se diseñaron 3 diagramas: Casos de uso, de clases y de actividades. Puesto que estos diagramas implementan una técnica que permite especificar el comportamiento del sistema, y la secuencia de interacciones entre el sistema y los actores.

7.2.1.1 Diagramas de casos de uso

Ilustración 4 Caso de uso: Administrador



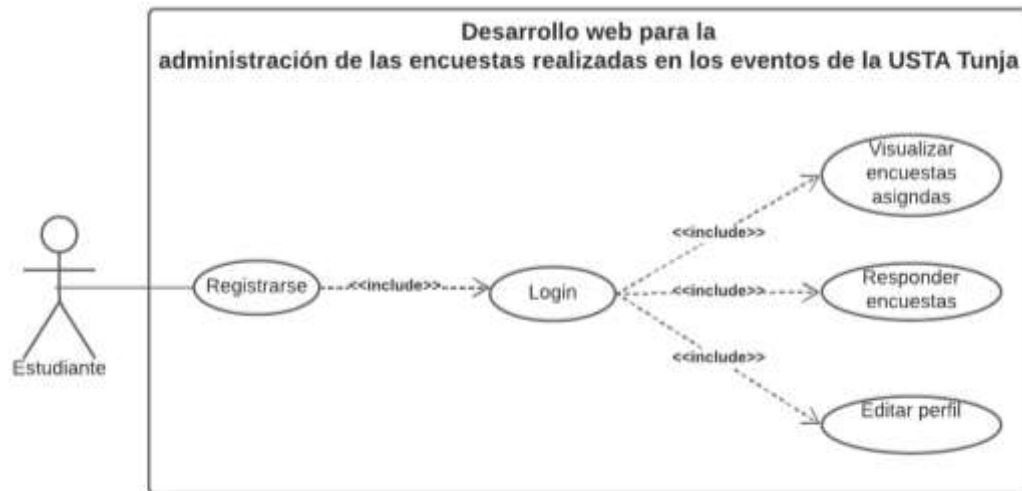
Fuente: Autor

Ilustración 5 Caso de uso: Docente, Invitado y Secretaria



Fuente: Autor

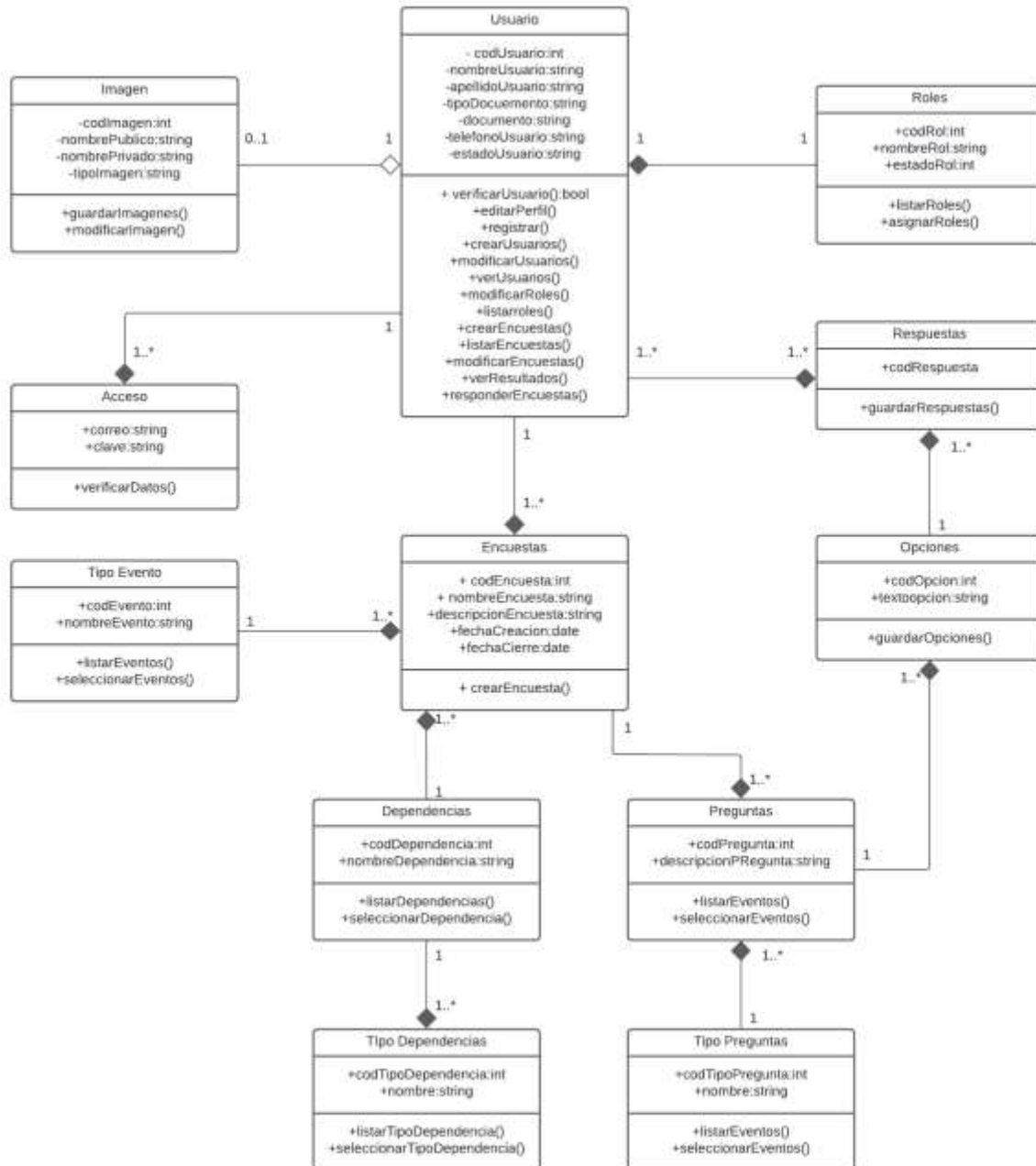
Ilustración 6 Caso de uso: Estudiante



Fuente: Autor

7.2.1.2 Diagrama de clases

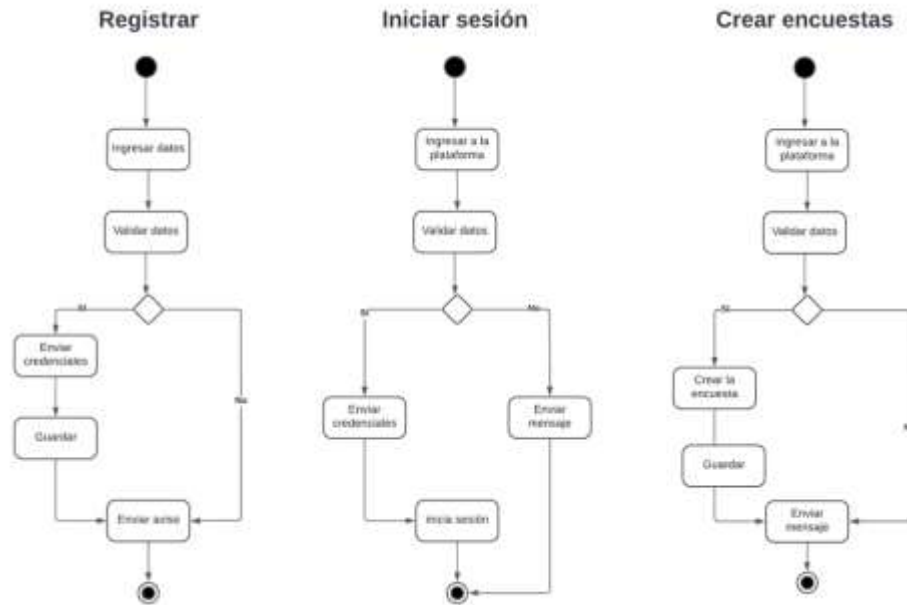
Ilustración 7 Diagrama de clases



Fuente: Autor

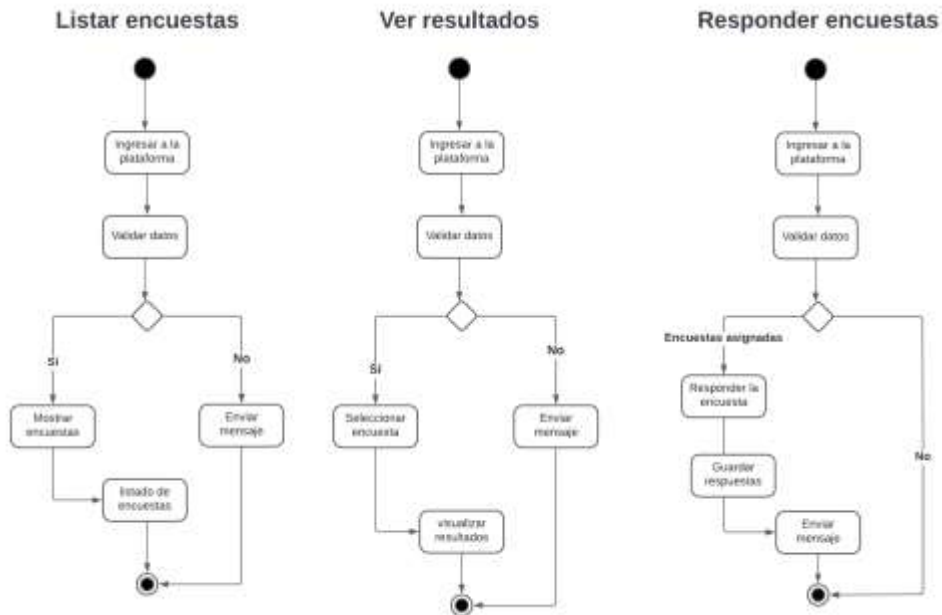
7.2.1.3 Diagrama de actividades

Ilustración 8 Diagrama de actividades: Registrar, iniciar sesión y crear encuestas



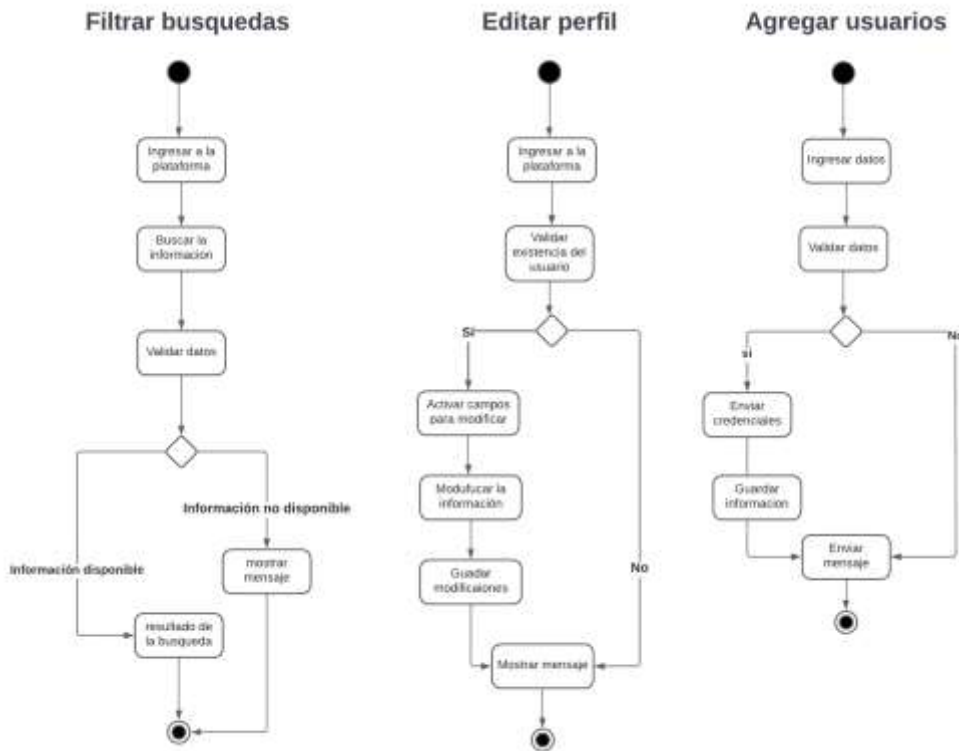
Fuente: Autor

Ilustración 9 Diagrama de actividades: Listar encuestas, ver resultados y responder encuestas



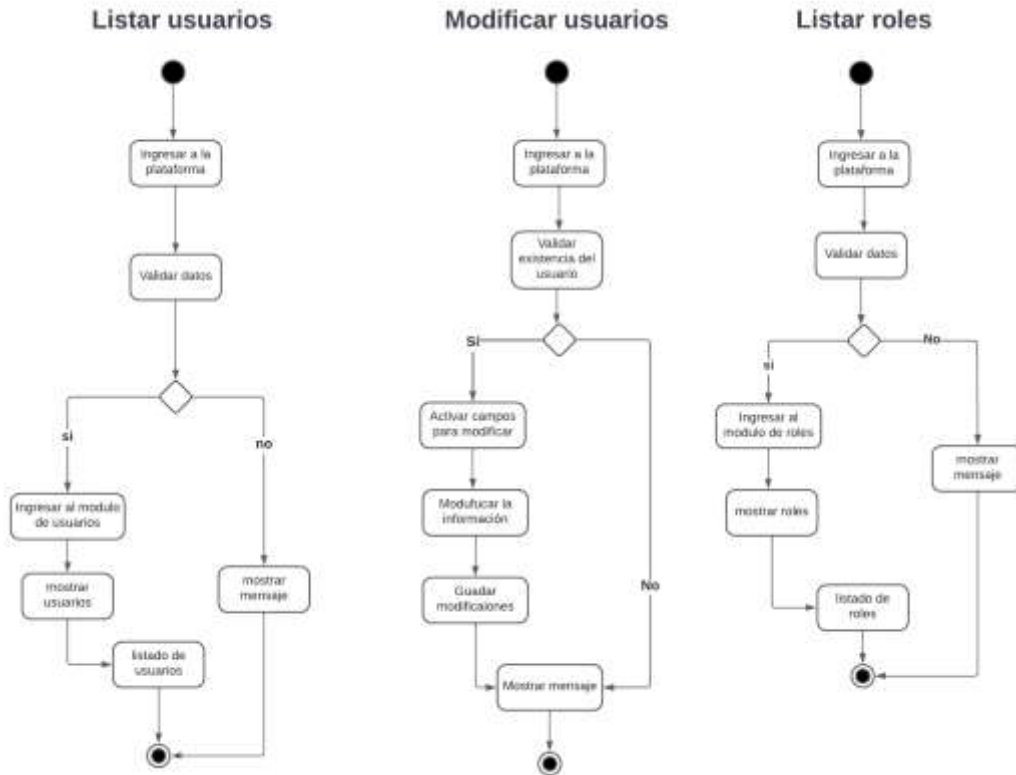
Fuente: Autor

Ilustración 10 Diagrama de actividades: Filtrar búsquedas, editar perfil y agregar usuarios



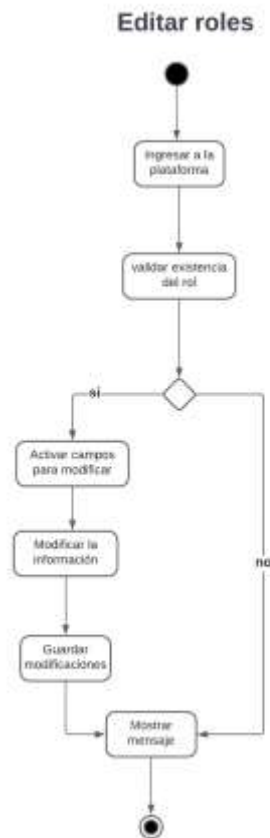
Fuente: Autor

Ilustración 11 Diagrama de actividades: Listar, modificar usuarios y listar roles



Fuente: Autor

Ilustración 12 Diagrama de actividades: Editar roles



Fuente: Autor

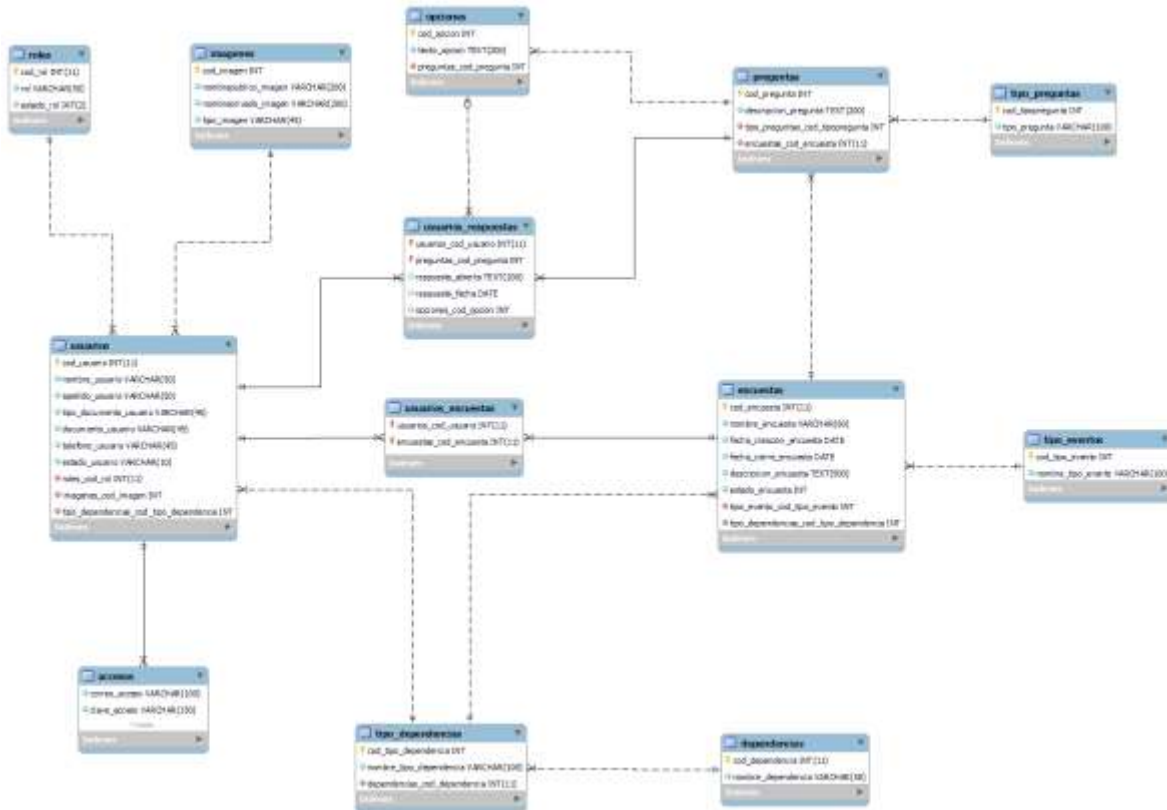
7.2.2. Diseño de la base de datos

A continuación se presenta el diseño entidad-relación de la base de datos, el cual está compuesto por las siguientes tablas:

- roles
- imágenes
- usuarios
- accesos
- usuarios_encuestas
- encuestas
- tipo_dependencias
- dependencias
- tipo_eventos
- preguntas

- tipo_preguntas
- opciones
- usuarios_preguntas

Ilustración 13 Modelo entidad relación de la base de datos



Fuente: Autor

7.2.3. Diseño de la interfaz

Para el diseño de la interfaz del sistema, se crearon unos Mockups del primer prototipo del aplicativo web, con la ayuda de la herramienta *Moqups · Mockups, Wireframes & Prototyping*. Para la creación de estos Mockups se tuvo en cuenta el diseño de un software que fue expuesto por la Stakeholder, esto con el fin de que se pudiera conservar algunos colores, diseño y estructura en el sistema a desarrollar.

Se desarrollaron 16 Mockups, los cuales son: Ingreso al aplicativo, registrar cuenta, Inicio del aplicativo, listar y modificar usuarios, listar y modificar roles, editar perfil, listar y modificar

encuestas, crear encuestas, ver resultados de las encuestas, formulario para responder encuestas, Mockups para roles distintos al administrador ([Ver anexo 3](#))

Tabla 4 Fase 2 Diseño del prototipo

Tipo	Elementos
Entradas	<ul style="list-style-type: none">• Documento de requerimientos
Instrumentos y/o herramientas	<ul style="list-style-type: none">• Conexión a internet• Computador• Programas del computador<ul style="list-style-type: none">✓ Navegador Web✓ Word• Moqups · Mockups, Wireframes & Prototyping
Salidas	Prototipos de los Mockups de la interfaz del sistema, diseño de los diagramas
Tiempo	3 semanas

Fuente: Autor

7.3. Fase 3. Construcción del sistema

Esta arquitectura se compone de dos partes: Backend y Frontend. El desarrollo de la fase de construcción estuvo compuesto de las siguientes actividades:

7.3.1. Selección de las tecnologías

Para elegir las tecnologías con las cuales se iba a desarrollar el proyecto se hizo una reunión junto con el director del proyecto de grado, allí se revisaron las tecnologías, ventajas y desventajas que tenían sobre el proyecto, y finalmente se optaron por las más adecuadas.

- Para el motor de base de datos se eligió PostgreSQL. Puesto que este motor de base de datos es relacional, gratuito, seguro y ampliamente usado en el mundo empresarial, además cuenta con buena documentación y soporte. Por otra parte, permite almacenar datos grandes de forma segura.
- Como lenguaje de programación se eligió Typescript, ya que es un lenguaje de programación basado en JavaScript fuertemente tipado que brinda mejores herramientas en cualquier escala. Además, typescript se convierte a JavaScript, el cual se ejecuta en cualquier lugar donde se ejecute JavaScript. Por otra parte, Typescript agrega sintaxis adicional a JavaScript para admitir una integración más estrecha con su editor y de esta manera detectar errores de forma temprana en su editor (JavaScript With Syntax For Types., s. f.)
- Para el entorno de ejecución se eligió Node js, ya que este entorno de ejecución es controlado por eventos diseñado para crear aplicaciones escalables, permitiendo establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no hay bloqueo de procesos (Simões, 2021)
- Como editor de código se utilizó Microsoft Visual Studio Code puesto que es ligero y además se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Tiene soporte integrado para JavaScript, Typescript y Node.js, y un rico sistema de complementos para otros lenguajes y tiempos de ejecución.
- Para el framework del front se optó Angular, puesto que este framework ofrece control sobre la escalabilidad. Satisface las demandas masivas de datos mediante la creación de modelos de datos en RxJS, Immutable.js o cualquier otro modelo de inserción. Además este Framework ofrece (Angular, 2021a)

- **Velocidad y rendimiento:** Angular transforma sus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript, ofreciendo todos los beneficios del código escrito a mano con la productividad de un marco. Además las aplicaciones angulares se cargan rápidamente con un nuevo enrutador de componentes que ofrece distribución automática de código, para que los usuarios carguen solo el código necesario para mostrar la vista que desean. (Angular, 2021a)
- **Productividad:** Crea rápidamente vistas de interfaz de usuario con una sintaxis de plantillas simple y precisa. Por otra parte obtiene finalización de código inteligente, errores instantáneos y otros comentarios en editores e IDE populares (Angular, 2021a)

Tabla 5 Fase 2.1 Selección de las tecnologías

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> • Documento de requerimientos • Listado de tecnologías a utilizar
Instrumentos y/o herramientas	<ul style="list-style-type: none"> • Conexión a internet • Computador
Salidas	Programas instalados en los equipos donde se va a desarrollar el proyecto
Tiempo	2 semanas

Fuente: Autor

Tabla 6 Fase 2.2 Versión tecnologías utilizadas

Herramienta	Versión
Angular	13.2.2
Visual Studio Code	1.72.1
Typescript	4.5.2
Node .js	16.15.0
Atom	1.60.0
PostgreSQL	14.3.1
PG Admin	6.8

Fuente: Autor

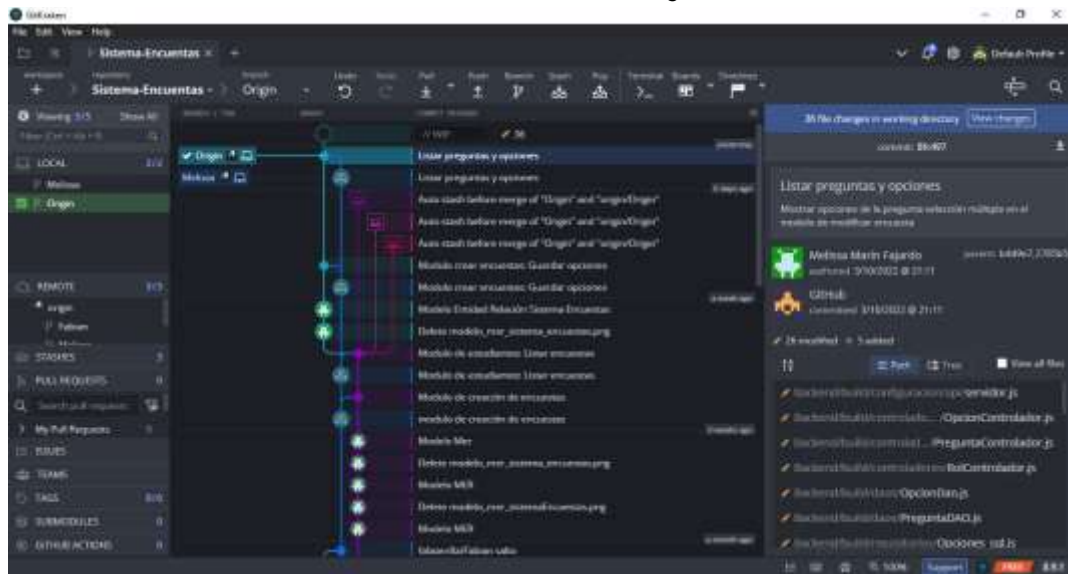
En la tabla anterior se muestran todas las herramientas usadas para el desarrollo del proyecto, con sus respectivas versiones.

7.3.2. Otras herramientas

Teniendo en cuenta que el desarrollo del proyecto estaba conformado por 2 integrantes, se usó la herramienta GitKraken y GitHub, para ir teniendo un control y conocimiento sobre los cambios que iba haciendo cada integrante del equipo.

- GitKraken: Es un software completo que ofrece una gama de herramientas para ayudar a los desarrolladores en la ejecución de sus proyectos. Además, los equipos de trabajo pueden colaborar más fácilmente, sin importar dónde se encuentren. Por otra parte es fácil de leer y permite visualizar la estructura de la rama y el historial de confirmación. Asimismo, ofrece la posibilidad de visualizar quién ha realizado cambios en el código y cuándo lo ha hecho. También tiene la función de autocompletar los comandos. (Qué es GitKraken | KeepCoding Tech School, s. f.)

Ilustración 14 Historial de cambios de la rama Origin en GitKraken

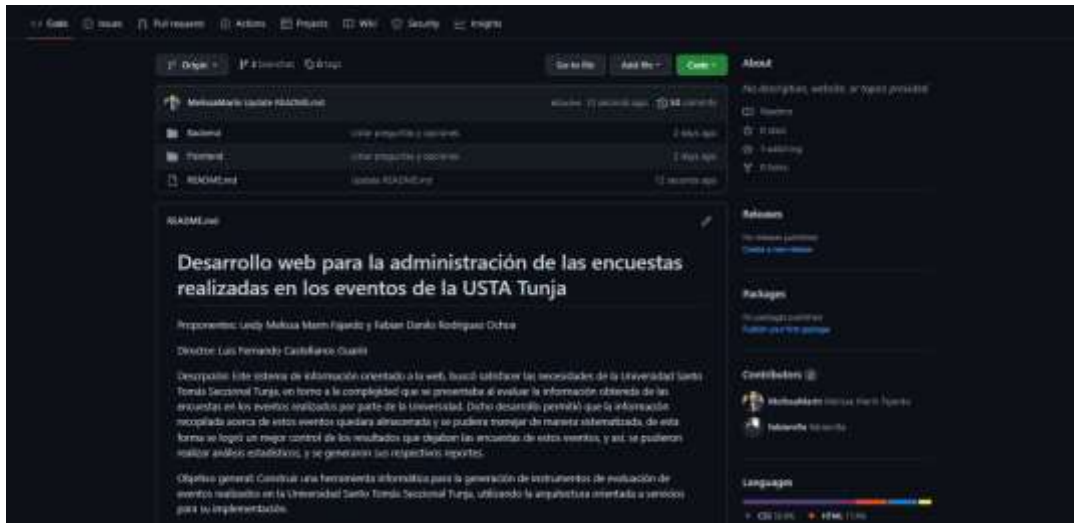


Fuente: Autor

En la imagen anterior se muestra el uso de la herramienta GitKraken por parte de los integrantes del equipo de desarrollo.

- GitHub: Es una plataforma de desarrollo inspirada en la forma de trabajar de cada desarrollador. Desde el código abierto hasta el comercial, puede alojar y revisar código, administrar proyectos y crear software junto con millones de otros desarrolladores.

Ilustración 15 Repositorio GitHub del proyecto

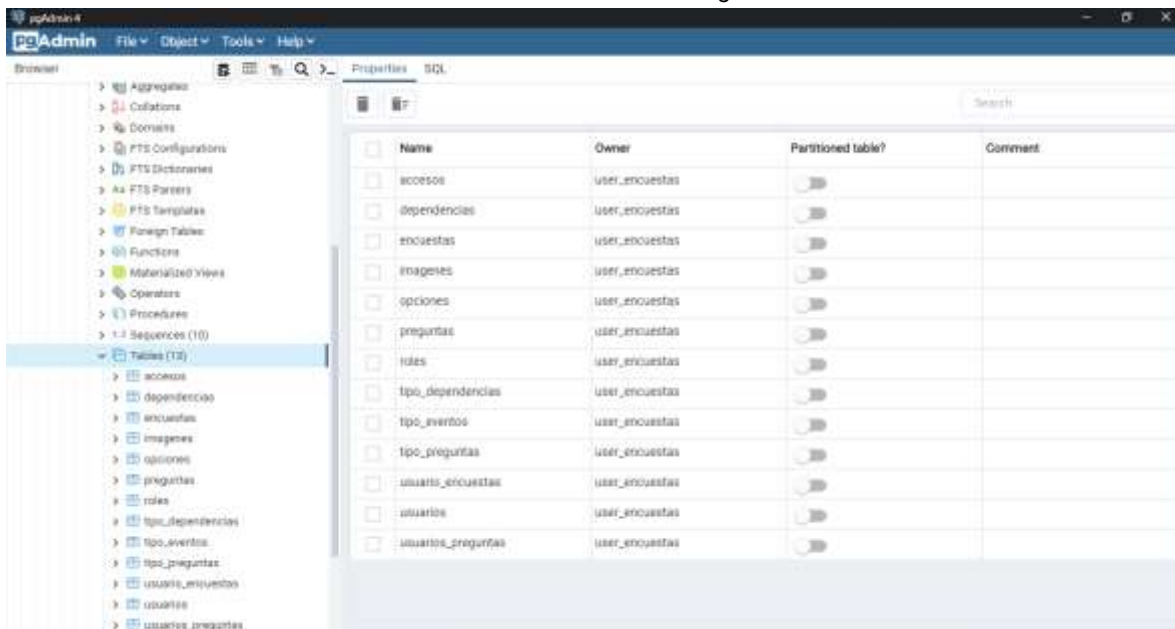


Fuente: Autor

En la imagen anterior se muestra el uso de la herramienta GitHub por parte de los integrantes del equipo de desarrollo.

7.3.3. Construcción de la base de datos

Ilustración 16 Base de datos PgAdmin



Para la construcción de la base de datos se eligió Postgres como ya se había mencionado en la sección de selección de tecnologías. PostgreSQL es un sistema de gestión de base de datos muy estable y robusto que permite la escalabilidad y el mantenimiento de la integridad de los datos gracias a sus políticas internas de seguridad. Además, PostgreSQL ofrece un nivel de consultas bastante amplio, lo cual era necesario para el sistema que se desarrolló, puesto que dicho sistema requiere consultas recursivas para encontrar información de manera eficaz de acuerdo con la lógica proyectada en el diseño de la base de datos.

7.3.3.1 Sentencias DDL para la construcción de la base de datos

La construcción de la base de datos se realizó con sentencias SQL de tipo DDL (Data Definition Language). Dichas sentencias permitieron definir la estructura y crear las tablas pertenecientes a la base de datos ([Ver anexo 4](#))

7.3.3.2 Sentencias DML para la inserción de información a la base de datos

Para que el funcionamiento de la base de datos se realice de manera exitosa se establecen de manera predeterminada unas condiciones en el DML (lenguaje de manipulación de datos). El orden de las condiciones es el siguiente:

1. Los roles ya están establecidos, incluyendo su nombre y estado, donde 1 significa activo y 2 inactivo, estos no se pueden modificar, ni eliminar, además no se crean nuevos roles, esto de acuerdo con los requerimientos establecidos para el proyecto. A continuación se muestra el DML para la correcta inserción de los roles.

Ilustración 17 Sentencia DML Creación de roles

```
INSERT INTO roles(nombre_rol, estado_rol) VALUES ('Administrador', 1);  
INSERT INTO roles(nombre_rol, estado_rol) VALUES ('Docente', 1);  
INSERT INTO roles(nombre_rol, estado_rol) VALUES ('Secretaria', 1);  
INSERT INTO roles(nombre_rol, estado_rol) VALUES ('Estudiante', 1);  
INSERT INTO roles(nombre_rol, estado_rol) VALUES ('Invitado', 1);
```

2. Como dependencias se estableció Facultades, a continuación se muestra la sentencia DML para la creación de facultades

Ilustración 18 Sentencia DML Creación de dependencias

```
INSERT INTO dependencias(nombre_dependencia) values ('Facultades');
```

3. Para poder crear los tipos de dependencia, primero se debe crear la dependencia. De acuerdo a los requerimientos del sistema se manejó solo Facultades como dependencia, y de allí todas las facultades que hay en la USTA Tunja como tipos de dependencia. A continuación se muestra la sentencia DML en que se debe crear los tipos de dependencia

Ilustración 19 Sentencia DML Creación de tipo dependencias

```
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Administración de empresas');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Arquitectura');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Contaduría pública');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Cultura física, recreación y deporte');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Derecho');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Diseño e Interacción');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Ingeniería ambiental');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Ingeniería civil');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Ingeniería de sistemas');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Ingeniería electrónica');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Ingeniería industrial');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Ingeniería mecánica');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Lic. En español y lenguas extranjeras');  
INSERT INTO tipo_dependencias(cod_dependencia,nombre_tipo_dependencia) values (1,'Negocios Internacionales');
```

4. El aplicativo asignará una imagen por defecto a cada nuevo usuario, la cual puede ser modificada después por el usuario. Un ejemplo de la sentencia para insertar de manera correcta la imagen es la siguiente:

Ilustración 20 Sentencia DML Creación de imágenes

```
INSERT INTO imagenes(cod_imagen, nombre_publico_imagen, nombre_privado_imagen, tipo_imagen)  
VALUES (1, 'publico.png', 'privado.png', 'png');
```

5. De manera predeterminada el sistema registra cada nuevo usuario con el rol de **Estudiante**, esto por motivos de seguridad y teniendo en cuenta que son muchos los estudiantes que se pueden registrar en el aplicativo, es más fácil para el administrador cambiar el rol de otros usuarios que asignar el rol a cada estudiante registrado. A continuación, se muestra un ejemplo de cómo crear usuarios de manera correcta.

Ilustración 21 Sentencia DML Creación de usuarios

```
INSERT INTO usuarios (cod_usuario, cod_rol, documento_usuario, tipo_documento_usuario, nombres_usuario, apellidos_usuario, telefono_usuario, estado_usuario, cod_imagen, cod_tipo_dependencia)
VALUES (1, 1, 109900000, 1, 'Melissa', 'Marín', 314400000, 1,1,8);
```

6. Para asignar los accesos al sistema, se debe tener el usuario primeramente creado, por eso la importancia de seguir el orden de las condiciones. Esta inserción debe realizarse como se muestra a continuación en el ejemplo.

Ilustración 22 Sentencia DML Creación de acceso

```
INSERT INTO accesos (cod_usuario, correo_acceso, clave_acceso)
VALUES (1, 'usuario.usta@usantoto.edu.co', 'laclaveencriptada');
```

7. Para que las encuestas se puedan crear de manera correcta se debe tener creado los tipos de eventos como lo muestra el siguiente ejemplo.

Ilustración 23 Sentencia DML Creación tipos de eventos

```
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Claustro');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Conferencia');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Congreso');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Foro');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Seminario');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Simposio');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Taller académico');
INSERT INTO tipo_eventos (nombre_tipo_evento) values ('Webinar');
```

8. Seguidamente se crean los tipos de pregunta como lo indica el siguiente ejemplo.

Ilustración 24 Sentencia DML Creación tipo preguntas

```
INSERT INTO tipo_preguntas (nombre_tipo_pregunta) values ('Fecha');
INSERT INTO tipo_preguntas (nombre_tipo_pregunta) values ('Pregunta abierta');
INSERT INTO tipo_preguntas (nombre_tipo_pregunta) values ('Opción múltiple');
```

7.3.4. Construcción del backend

7.3.4.1 Librerías

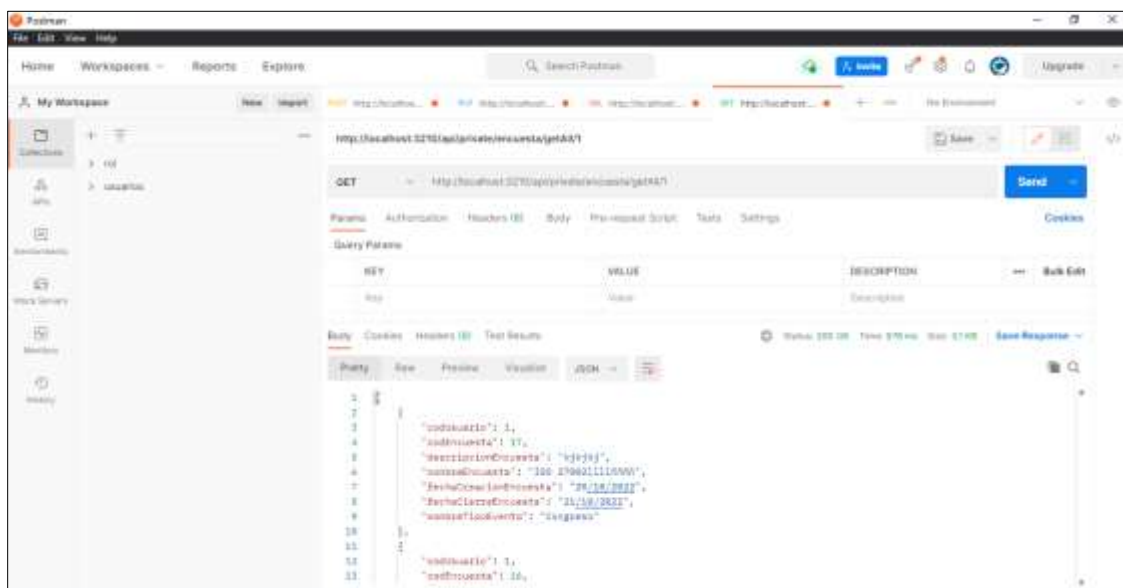
Las librerías ofrecen muchas funciones estándar para que los desarrolladores no tengan que preocuparse por demasiada funcionalidad. Por lo tanto, pueden usarse para crear

fácilmente páginas web con componentes de interfaz de usuario, utilidades de lenguaje, funciones matemáticas y más. Las librerías usadas para la construcción del backend fueron las siguientes:

- @types/cors: 2.8.12,
- @types/express: 4.17.13,
- @types/formidable: 2.0.4,
- @types/jsonwebtoken: 8.5.8,
- @types/morgan: 1.9.3

7.3.4.2 Herramientas

Ilustración 25 Herramienta Postman



Para el desarrollo del backend, se hizo uso de la herramienta conocida como **Postman**, que es una plataforma API, que como su nombre lo indica puede almacenar, realizar pruebas y mostrar resultados de todo lo relacionado con la API. Allí se validaron los métodos GET, POST, PUT Y DELETE según fuera necesario para la API.

7.3.4.3 Estructura

El framework de Node.js ofrece la posibilidad de estructurar el backend por módulos de acuerdo a la arquitectura que se planteó para el proyecto. Teniendo en cuenta que esto es una aplicación web que consumirá una API, se hace uso de la librería Express, la cual

permite hacer las distintas peticiones y proporciona las características necesarias para poder consumir la API.

Ilustración 26 Uso librería Express

```
import express from 'express';

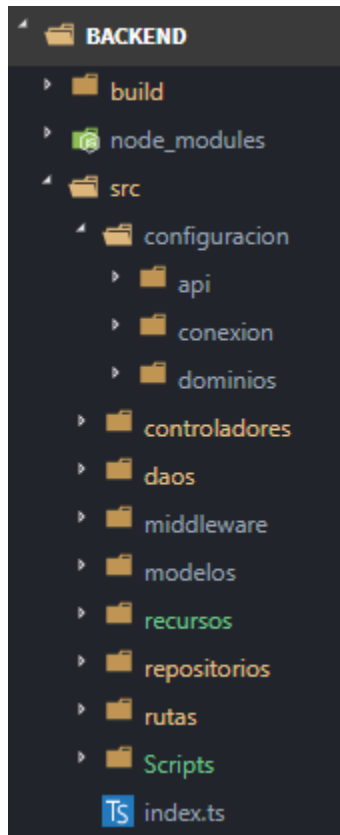
class Servidor {
  public app: express.Application;
  constructor() {
    this.app = express();
    this.cargarConfiguracion();
    this.cargarRutas();
  }
  public cargarConfiguracion(): void {
    this.app.set('PORT', 3210);
    this.app.use(cors());
    this.app.use(morgan('dev'));
    this.app.use(express.json({ limit: '50mb' }));
    this.app.use(express.urlencoded({ extended: true }));
  }

  public cargarRutas(): void {
    // Rutas públicas
    this.app.use('/api/public/access', rutaAcceso);

    // Rutas privadas
    this.app.use('/api/private/rol', miRutaRol);
    this.app.use('/api/private/usuario', miRutaUsuario);
    this.app.use('/api/private/encuesta', miRutaEncuesta);
    this.app.use('/api/private/estudiante', miRutaEstudiante);
    this.app.use('/api/private/pregunta', miRutaPregunta);
    this.app.use('/api/private/opciones', miRutaOpcion);
  }
}
```

La estructura backend del sistema está distribuido de la siguiente manera:

Ilustración 27 Estructura backend



- Carpeta node_modules: Esta carpeta se genera automáticamente cuando se crea por primera vez el proyecto, allí se almacenan todos los módulos que se ejecutan cuando se forma el proyecto.
- Carpeta configuración: Se divide en 3 subcarpetas: en la subcarpeta API se encuentran todas las rutas necesarias para poder consumir cada servicio. En la carpeta conexión se encuentran los archivos con las distintas configuraciones como el pool, para que la conexión se pueda realizar de manera exitosa, y finalmente en la subcarpeta dominios se define la lógica de conexión a la base de datos, es decir el usuario, la contraseña, el host, el puerto y el nombre de la base de datos.

Ilustración 28 Configuración conexión BD

```
export default {  
  user: 'user_encuestas',  
  password: 'xxxxxxx',  
  host: 'localhost',  
  port: 5432,  
  database: 'bd_sistemaencuestas'  
}
```

- Carpeta controladores: Los archivos allí situados actúan como intermediarios y permiten la gestión del flujo de información y las transformaciones para adaptar los datos a las necesidades de cada API.
- Carpeta DAOS: Estos archivos son los encargados de acceder a los datos a través de los métodos allí definidos, además se encarga de ordenar los resultados que arroja la ejecución de estos métodos.
- Carpeta middleware: Allí se encuentra el middleware de autenticación, el cual es el encargado de la verificación y autenticación al token que permite el acceso o ingreso al aplicativo.
- Carpeta modelos: Los archivos que allí se encuentran son utilizados para enviar el objeto completo de cada modelo, en el caso de la imagen se necesita el objeto completo ya que en la búsqueda solo trae los datos de la tabla imágenes y se requiere también enviar el base64 junto a la imagen
- Carpeta recursos: En esta carpeta se encuentran almacenadas las imágenes utilizadas por el usuario cuando modifica su perfil.
- Carpeta repositorios: Estos archivos son los encargados de acceder a las consultas a la base de datos y obtener la información solicitada por el controlador asignado.
- Carpeta rutas: Allí se encuentran los endpoints de la API, los cuales se encargan de gestionar las peticiones Get, Post y Put hacia cada controlador asignado.
- Carpeta Scripts: Allí se encuentra el DDL (lenguaje de definición de datos) para la creación de las tablas de la base de datos y el DML (lenguaje de manipulación de datos) con las sentencias para la inserción de los datos a la base de datos

7.3.4.4 Comandos

Para ejecutar el backend se requieren 2 comandos, los cuales deben ir en el siguiente orden:

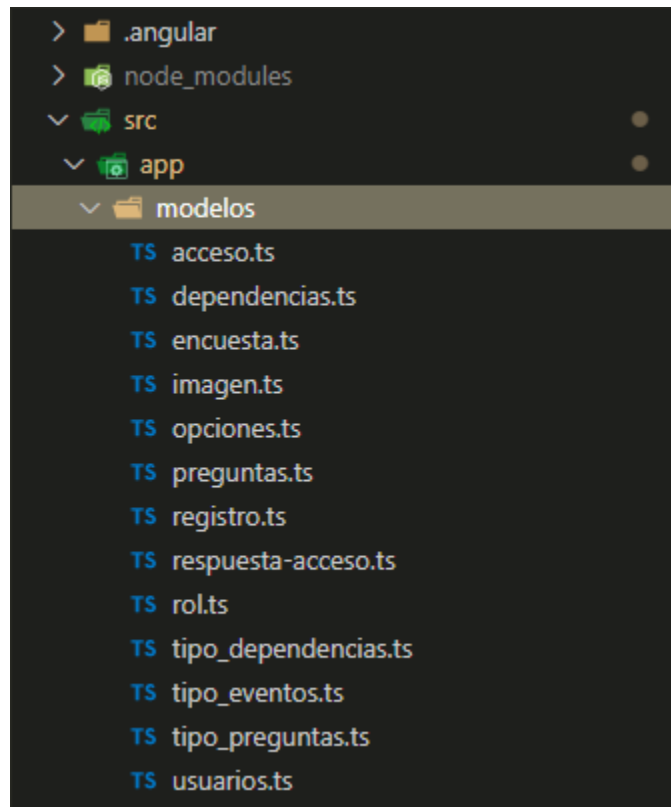
- 1) npm run build: Este comando se debe ingresar en la terminal, esto generará una carpeta automáticamente llamada build, dicho comando se encarga de compilar y construir el proyecto para su producción.
- 2) npm run dev: Esta instrucción también se debe poner en la terminal del proyecto y es la encargada de ejecutar el entorno de desarrollo.

7.3.5. Construcción del frontend

7.3.5.1 Estructura

El framework de angular ofrece la posibilidad de modularizar el proyecto de acuerdo a la arquitectura planteada. Para el desarrollo de este proyecto se definieron los siguientes módulos:

- Modelos: Aquí se definen las propiedades para cada modelo, una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.



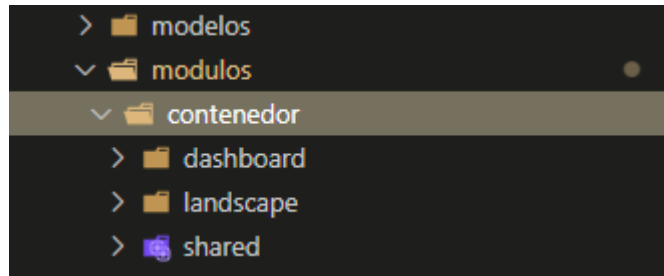
- Módulos: Dentro de esta carpeta hay situadas 3 subcarpetas:
 - Contenedor:

Aquí se encuentran 3 subcarpetas: dashboard, landscape y shared.

 - ✓ Dashboard incluye 4 unidades: la cabecera-dash se encarga de desplegar el menú superior de la aplicación. El contenedor-dash, despliega los templates de los componentes privados. La subcarpeta error-dash se encarga de desplegar el componente cuando hay un error en el direccionamiento del proyecto. Y finalmente el sidebar-dash el cual es el encargado de desplegar el menú de navegación.
 - ✓ Landscape: Contiene 2 módulos: contenedor-land y error-land. El contenedor-land se encarga de abrir los componentes del módulo público y el error-land se encarga de desplegar el componente cuando hay un error en el direccionamiento del módulo público.

- ✓ Shared: Allí se encuentran ubicados el componente scroll-to-top, el cual es un componente compartido, es decir, que puede ser utilizado por cualquier otro componente o módulo del proyecto

Ilustración 30 Estructura del frontend Contenedor

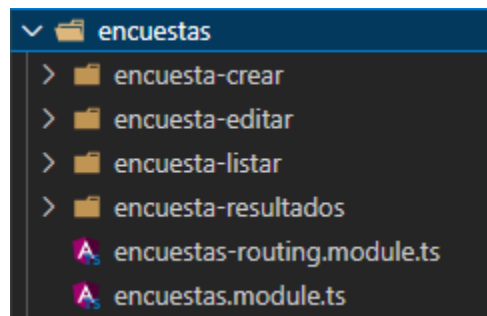


- Privado:

Este módulo privado incluye las unidades establecidas con la arquitectura y los requerimientos del proyecto, dichas unidades son las siguientes:

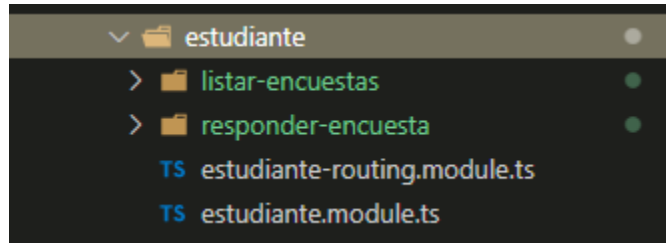
- ✓ Encuestas: Este submódulo incluye los componentes de encuesta-crear, encuesta-editar y encuesta-listar. Allí se da toda la lógica para el proceso de crear la encuesta.

Ilustración 31 Estructura del frontend Módulo Privado Encuestas



- ✓ Estudiante: Aquí se encuentran 2 componentes: listar-encuestas y responder-encuestas. Ya que de acuerdo a los requerimientos del proyecto, a diferencia de otros roles, el estudiante solo podrá listar y responder las encuestas que se le hayan asignado.

Ilustración 32 Estructura frontend Módulo Privado Estudiante



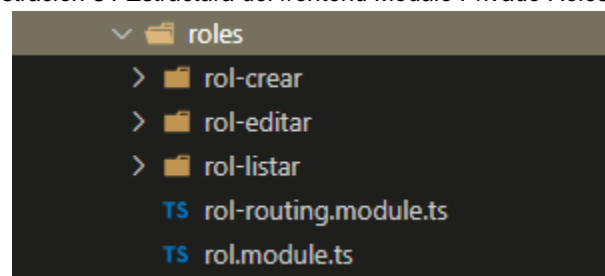
- ✓ Inicio: Este es el componente principal, es la primera vista que tiene el usuario cada vez que ingresa al aplicativo.

Ilustración 33 Estructura del frontend Módulo Privado Inicio



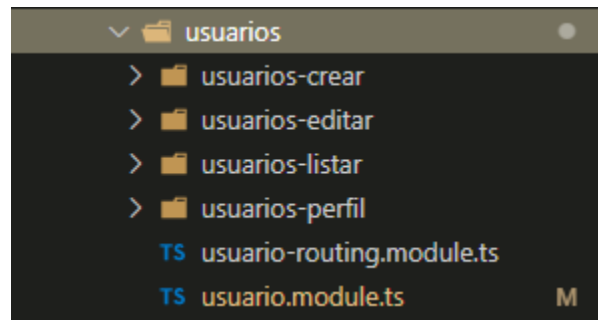
- ✓ Roles: En este submódulo se encuentran los componentes: rol-editar y rol-listar. Hay que recalcar que según los requerimientos establecidos para el proyecto, los roles solo podrán listarse y solo se podrá modificar el estado de cada rol. A este submódulo solo tiene acceso el administrador.

Ilustración 34 Estructura del frontend Módulo Privado Roles



- ✓ Usuarios: En este submódulo están los componentes: usuarios-crear, usuarios-listar, usuarios-modificar, usuarios-listar y usuarios-perfil. A este módulo solo tiene acceso el administrador, sin embargo, al componente de **usuario-perfil** sí tienen acceso todos los usuarios, ya que allí es donde pueden modificar los datos del perfil de cada uno.

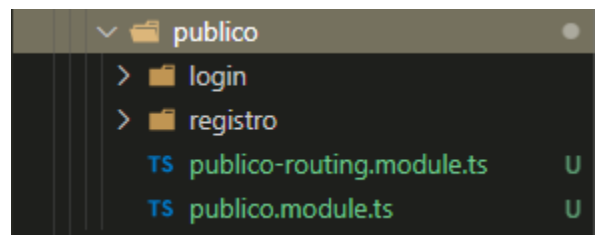
Ilustración 35 Estructura del frontend Módulo Privado Usuarios



- Público:

Este submódulo incluye 2 componentes: Login y registro. Los cuales son los encargados de realizar el respectivo ingreso y registro a la aplicación.

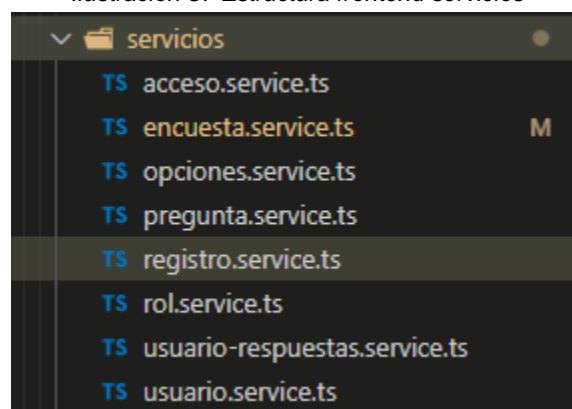
Ilustración 36 Estructura del frontend Modulo Público



- Servicios:

Aquí se encuentran todos los servicios necesarios para poder intercambiar información y consumir la API

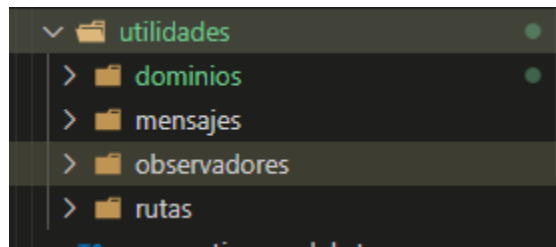
Ilustración 37 Estructura frontend servicios



- Utilidades: Dentro de esta carpeta se encuentran 4 subcarpetas:

✓ Dominios:

Ilustración 38 Estructura utilidades



7.3.5.2 Comando

Se usa el comando **ng serve** para construir la aplicación y servirla localmente, el servidor automáticamente reconstruye la aplicación y recarga la página cuando se hace un cambio en cualquiera de los archivos base.

7.3.5.3 Servicios y API

La idea principal de Angular es que su programación sea asíncrona, es decir, que el código construido permita a un programa solicitar que una función se realice al mismo tiempo que la función o las funciones principales, sin detenerse a esperar a que la primera se haya completado por lo que se crean los servicios los cuales son encargados de procesar la información, transformándola en caso de ser necesario y presentándola de la forma más sencilla a la vista.

Angular proporciona el módulo HttpClient para llamadas HTTP (@angular/common/http). Algunas de sus características: API RxJS Observable, devuelve objetos de determinado tipo para solicitudes o respuestas, manejo de errores, entre otros.

Para lograr la programación asíncrona se utilizó RxJS, que es una librería para la programación reactiva que facilita la construcción de código asíncrono basado en secuencias de observables y así poder realizar peticiones de forma asíncrona.

Los observables son los encargados de propagar la información y notificar sus cambios, para ello proporciona métodos a partir de los cuales el observador puede suscribirse o cancelar la suscripción de sus flujos de datos.

Para consumir el servicio creado, el observador debe suscribirse a dicho servicio. Por otra parte, para conectar el servicio con la API, se crea el método, el cual retorna un observable

de tipo Opciones como en el ejemplo, el cual nos permite obtener la respuesta del servidor, de acuerdo con la API que se haya establecido, la cual contiene la ruta con la petición al servidor.

Ilustración 39 Ejemplo Servicio Opciones

```
public listarOpciones(codPregunta: number): Observable<Opciones[]> {  
  return this.http.get<Opciones[]>(  
    this.apiUrlListarOpciones + '/' + codPregunta  
  );  
}
```

En la siguiente imagen se evidencia un ejemplo de cómo funciona un observable al realizar una petición al servicio.

Ilustración 40 Ejemplo de observable

```
public listarOpciones(codigoPregunta:number): void {  
  this.miSuscripcion = this.opcionService  
    .listarOpciones(codigoPregunta)  
    .pipe(  
      map((respuesta: Opciones[]) => {  
        this.objPregunta.arregloOpciones = respuesta;  
        return respuesta;  
      }  
    ),  
      catchError((err) => {  
        throw err;  
      })  
    )  
    .subscribe(observadorAny);  
}
```

Por otra parte, se cuenta con un guardián llamado vigilante-guard, el cual es el encargado de validar la seguridad del aplicativo, evitando saltos en la seguridad o que se pueda acceder al sistema sin el token generado respectivamente para cada usuario.

Ilustración 41 Validar token

```
import { Injectable } from '@angular/core';
import { CanActivate, Router } from '@angular/router';
import { AccesoService } from '../servicios/acceso.service';

...

@Injectable({
  providedIn: 'root',
})
export class VigilanteGuard implements CanActivate {
  constructor(private accesoService: AccesoService, private router: Router) {}
  canActivate(): boolean {
    if (this.accesoService.verificarUsuario()) {
      return true;
    }
    this.router.navigate(['/land/public/login']);
    return false;
  }
}
```

Tabla 7 Fase 3 Desarrollo del prototipo

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> Programas instalados para el desarrollo del prototipo
Instrumentos y/o herramientas	<ul style="list-style-type: none"> Computador Programas de las herramientas seleccionadas <ul style="list-style-type: none"> ✓ Atom ✓ Visual Studio Code ✓ PGAdmin
Salidas	Desarrollo del prototipo, cumpliendo con los requerimientos de software establecidos, con los métodos y funciones acordadas con el cliente desde un principio, respetando la estructura establecida para la ejecución de este proyecto (back y front).
Tiempo	4 meses

7.4. Fase 4. Pruebas del prototipo desarrollado

7.4.1. Pruebas de funcionalidad

Para el desarrollo de este tipo de prueba se utilizó la herramienta Selenium, la cual permite hacer pruebas automatizadas de código abierto, probando cada una de las funcionalidades del sistema y de esta manera poder verificar su correcta ejecución. A continuación se muestra el resultado de las 18 pruebas de funcionalidad que se le realizaron al sistema:

Ilustración 42 Prueba Funcional Login

Running 'Login'	16:37:10
1. open on /land/public/login OK	16:37:10
2. setWindowSize on 877x853 OK	16:37:12
3. click on id=usu OK	16:37:14
4. type on id=usu with value fabian.rodriquezo@usantoto.edu.co OK	16:37:16
5. click on id=cla OK	16:37:18
6. type on id=cla with value 12345 OK	16:37:20
7. click on css=.btn OK	16:37:22
'Login' completed successfully	16:37:22

Ilustración 43 Prueba agregar nuevos usuarios

Running 'Agregar nuevos usuarios'	17:49:41
1. open on /land/public/login OK	17:49:41
2. setWindowSize on 1001x857 OK	17:49:42
3. click on id=usu OK	17:49:42
4. type on id=usu with value fabian.rodriquezo@usantoto.edu.co OK	17:49:43
5. click on id=cla OK	17:49:43
6. type on id=cla with value 12345 OK	17:49:44
7. click on css=.btn OK	17:49:45
8. click on linkText=Usuarios OK	17:49:45
9. mouseOver on linkText=Usuarios OK	17:49:46
10. mouseOut on linkText=Usuarios OK	17:49:46
11. click on linkText=Agregar usuario OK	17:49:47
12. click on id=nom OK	17:49:48
13. type on id=nom with value zaa OK	17:49:48
14. click on id=ape OK	17:49:49
15. type on id=ape with value duke OK	17:49:49
16. click on id=doc OK	17:49:50
17. type on id=doc with value 1234567890 OK	17:49:51
18. click on name=idpic OK	17:49:51
19. select on name=idpic with value label=Tarjeta de identidad OK	17:49:52
20. Trying to find css=.ng-touched > .ng-star-inserted:nth-child(3)... OK	17:49:52
21. click on id=tel OK	17:50:02
22. type on id=tel with value 123456 OK	17:50:03
23. click on id=cor OK	17:50:04
24. type on id=cor with value zaa.duke@usantoto.edu.co OK	17:50:04
25. select on name=nomrol with value label=Secretaria OK	17:50:05
26. click on css=.col-md-6:nth-child(1) .ng-star-inserted:nth-child(2) OK	17:50:05
27. click on css=.row:nth-child(5) > .col-md-6:nth-child(2) .form-select OK	17:50:06
28. select on css=.row:nth-child(5) > .col-md-6:nth-child(2) .form-select with value label=Administración de empresas OK	17:50:07
29. click on css=.row:nth-child(5) > .col-md-6:nth-child(2) .ng-star-inserted:nth-child(1) OK	17:50:07
30. click on id=cont OK	17:50:08
31. type on id=cont with value 12345 OK	17:50:08
32. click on id=cla OK	17:50:09
33. type on id=cla with value 12345 OK	17:50:09
34. click on css=.btn-primary OK	17:50:10
35. mouseOver on css=.btn-primary OK	17:50:11
'Agregar nuevos usuarios' completed successfully	17:50:11

Ilustración 44 Prueba listar usuarios

Running "Listar Usuarios"	17:37:48
1. open on /land/public/login OK	17:37:49
2. setWindowSize on 1051x857 OK	17:37:49
3. click on id=usu OK	17:37:50
4. type on id=usu with value fabian.rodriguez@usantoto.edu.co OK	17:37:50
5. click on id=cla OK	17:37:51
6. type on id=cla with value 12345 OK	17:37:52
7. click on css= btn OK	17:37:52
8. click on linkText=Usuarios OK	17:37:53
9. mouseOver on linkText=Usuarios OK	17:37:53
10. mouseOut on linkText=Usuarios OK	17:37:54
11. mouseOver on linkText=Usuarios OK	17:37:54
12. mouseOut on linkText=Usuarios OK	17:37:55
"Listar Usuarios" completed successfully	17:37:56

Ilustración 45 Prueba modificar usuarios

Running "Modificar Usuarios"	18:00:49
1. open on /land/public/login OK	18:00:49
2. setWindowSize on 1383x880 OK	18:00:50
3. click on id=usu OK	18:00:50
4. type on id=usu with value fabian.rodriguez@usantoto.edu.co OK	18:00:51
5. click on id=cla OK	18:00:51
6. type on id=cla with value 12345 OK	18:00:52
7. click on css= btn OK	18:00:53
8. click on linkText=Usuarios OK	18:00:53
9. mouseOver on linkText=Usuarios OK	18:00:54
10. mouseOut on linkText=Usuarios OK	18:00:54
11. mouseOver on linkText=Usuarios OK	18:00:55
12. mouseOut on linkText=Usuarios OK	18:00:55
13. click on css= pagination-next = .ng-star-inserted OK	18:00:56
14. click on css= .far OK	18:00:57
15. click on id=mem OK	18:00:57
16. type on id=mem with value aara OK	18:00:58
17. click on css= .s-grid-item-child(7) = .btn-primary OK	18:00:58
"Modificar Usuarios" completed successfully	18:00:59

Ilustración 46 Prueba modificar acceso usuario

Running "Modificar acceso usuario"	18:58:40
1. open on /land/public/login OK	18:58:41
2. setWindowSize on 1178x860 OK	18:58:41
3. click on id=usu OK	18:58:42
4. type on id=usu with value fabian.rodriguez@usantoto.edu.co OK	18:58:42
5. click on id=cla OK	18:58:43
6. type on id=cla with value 12345 OK	18:58:44
7. click on css= btn OK	18:58:44
8. click on linkText=Usuarios OK	18:58:45
9. mouseOver on linkText=Usuarios OK	18:58:45
10. mouseOut on linkText=Usuarios OK	18:58:46
11. click on css= pagination-next = .ng-star-inserted OK	18:58:47
12. click on css= .far OK	18:58:47
13. click on id=profile-tab OK	18:58:48
14. click on id=cor OK	18:58:48
15. type on id=cor with value aara.dube@usantoto.edu.co OK	18:58:49
16. click on css= .s-grid-item-child(5) = .btn-primary OK	18:58:50
"Modificar acceso usuario" completed successfully	18:58:50

Ilustración 47 Prueba listar roles

Running 'Listar roles'	16:56:20
1. open on /land/public/login OK	16:56:21
2. setWindowSize on 864x860 OK	16:56:22
3. click on id=usu OK	16:56:23
4. type on id=usu with value fabian.rodriquezo@usantoto.edu.co OK	16:56:24
5. click on id=cla OK	16:56:25
6. type on id=cla with value 12345 OK	16:56:26
7. click on css=.btn OK	16:56:26
8. click on linkText=Roles OK	16:56:27
9. mouseOver on linkText=Roles OK	16:56:28
10. mouseOut on linkText=Roles OK	16:56:29
'Listar roles' completed successfully	16:56:30

Ilustración 48 Prueba modificar estado rol

Running 'Modificar estado rol'	17:02:15
1. open on /land/public/login OK	17:02:16
2. setWindowSize on 1051x800 OK	17:02:16
3. click on id=usu OK	17:02:17
4. type on id=usu with value fabian.rodriquezo@usantoto.edu.co OK	17:02:17
5. click on id=cla OK	17:02:18
6. type on id=cla with value 12345 OK	17:02:18
7. click on css=.btn OK	17:02:19
8. click on linkText=Roles OK	17:02:20
9. mouseOver on linkText=Roles OK	17:02:20
10. mouseOut on linkText=Roles OK	17:02:21
11. mouseOver on linkText=Roles OK	17:02:21
12. mouseOut on linkText=Roles OK	17:02:22
13. click on css=.ng-star-inserted:nth-child(3) .fa OK	17:02:22
14. click on name=estadoRol OK	17:02:23
15. select on name=estadoRol with value label=Inactivo OK	17:02:24
16. click on css=.form-control > .ng-star-inserted:nth-child(3) OK	17:02:24
17. click on css=.btn-primary OK	17:02:25
'Modificar estado rol' completed successfully	17:02:25

Ilustración 49 Prueba buscador listar encuesta

Running 'Buscador listar encuesta'	17:35:18
1. open on /land/public/login OK	17:35:19
2. setWindowSize on 1051x800 OK	17:35:19
3. click on id=usu OK	17:35:20
4. type on id=usu with value fabian.rodriquezo@usantoto.edu.co OK	17:35:20
5. click on id=cla OK	17:35:21
6. type on id=cla with value 12345 OK	17:35:21
7. click on css=.btn OK	17:35:22
8. click on linkText=Roles OK	17:35:22
9. mouseOver on linkText=Roles OK	17:35:23
10. mouseOut on linkText=Roles OK	17:35:24
11. click on id=buscar OK	17:35:24
12. type on id=buscar with value administrador OK	17:35:25
'Buscador listar encuesta' completed successfully	17:35:25

Ilustración 50 Prueba crear encuesta

Running "Crear Encuesta"	18:22:59
1. open on #andipubliclogin OK	18:22:59
2. setWindowSize on 1051x850 OK	18:23:00
3. click on id=usu OK	18:23:00
4. type on id=usu with value fabian.rodriguez@usantito.edu.co OK	18:23:01
5. click on id=cia OK	18:23:02
6. type on id=cia with value 12345 OK	18:23:02
7. click on css=btn OK	18:23:03
8. click on linkText=Encuestas OK	18:23:03
9. click on linkText=Crear encuesta OK	18:23:04
10. mouseOver on linkText=Crear encuesta OK	18:23:05
11. mouseOut on linkText=Crear encuesta OK	18:23:05
12. mouseOver on linkText=Crear encuesta OK	18:23:06
13. mouseOut on linkText=Crear encuesta OK	18:23:06
14. click on id=titulo OK	18:23:07
15. type on id=titulo with value Proyecto Sistema Encuestas OK	18:23:08
16. click on id=fecha_inicio OK	18:23:08
17. type on id=fecha_inicio with value 2022-11-24 OK	18:23:09
18. click on id=fecha_cierra OK	18:23:10
19. type on id=fecha_cierra with value 2022-11-29 OK	18:23:10
20. click on id=eventos OK	18:23:11
21. select on id=eventos with value label=Conferencia OK	18:23:11
22. click on css=#eventos > .ng-star-inserted:nth-child(3) OK	18:23:12
23. click on id=tipo_dependencia OK	18:23:13
24. select on id=tipo_dependencia with value label=Ingeniería de sistemas OK	18:23:13
25. click on css=#tipo_dependencia > .ng-star-inserted:nth-child(3) OK	18:23:14
26. click on id=descripcion OK	18:23:14
27. type on id=descripcion with value descripción OK	18:23:15
28. click on id=tipo_pregunta OK	18:23:16
29. select on id=tipo_pregunta with value label=Fecha OK	18:23:16
30. click on css=#tipo_pregunta > .ng-star-inserted:nth-child(2) OK	18:23:17
31. click on css=.fa OK	18:23:18
32. click on id=tipo_pregunta OK	18:23:18
33. select on id=tipo_pregunta with value label=Pregunta abierta OK	18:23:19
34. click on css=#tipo_pregunta > .ng-star-inserted:nth-child(3) OK	18:23:19
35. click on css=.fa OK	18:23:20
36. click on id=tipo_pregunta OK	18:23:20
37. select on id=tipo_pregunta with value label=Opción múltiple OK	18:23:21
38. click on css=#tipo_pregunta > .ng-star-inserted:nth-child(4) OK	18:23:22
39. click on css=.fa OK	18:23:22
40. click on id=floatingTextareaFecha OK	18:23:23
41. type on id=floatingTextareaFecha with value ¿Que día es hoy? OK	18:23:23
42. click on id=floatingTextareaAbierta OK	18:23:24
43. type on id=floatingTextareaAbierta with value ¿Como estas? OK	18:23:25
44. click on id=seleccion2 OK	18:23:25
45. type on id=seleccion2 with value Deporte favorito OK	18:23:26
46. click on css=otra-opcion OK	18:23:27
47. click on id=opcion02 OK	18:23:27
48. type on id=opcion02 with value Fútbol OK	18:23:28
49. click on css=otra-opcion OK	18:23:28
50. click on id=opcion12 OK	18:23:29
51. type on id=opcion12 with value Baloncesto OK	18:23:30
52. click on css=btn-primary OK	18:23:30
53. runScript on window.scrollTo(0,0) OK	18:23:31
"Crear Encuesta" completed successfully	18:23:31

Ilustración 51 Prueba listar encuestas

Running 'Listar Encuestas'	18:29:58
1. open on /randpublic/login OK	18:29:59
2. setWindowSize on 1051x860 OK	18:29:59
3. click on id=usu OK	18:30:00
4. type on id=usu with value fabian.rodriguez@usantofe.edu.co OK	18:30:00
5. click on id=cla OK	18:30:01
6. type on id=cla with value 12345 OK	18:30:02
7. click on cas=> btn OK	18:30:02
8. click on linkText=Encuestas OK	18:30:03
9. click on linkText=Listar encuestas OK	18:30:03
10. mouseOver on linkText=Listar encuestas OK	18:30:04
11. mouseOut on linkText=Listar encuestas OK	18:30:05
12. mouseOver on linkText=Listar encuestas OK	18:30:05
13. mouseOut on linkText=Listar encuestas OK	18:30:06
'Listar Encuestas' completed successfully	18:30:06

Ilustración 52 Prueba modificar información encuesta

Running 'Modificar información encuesta'	18:33:58
1. open on /randpublic/login OK	18:33:59
2. setWindowSize on 1051x860 OK	18:33:59
3. click on id=usu OK	18:34:00
4. type on id=usu with value fabian.rodriguez@usantofe.edu.co OK	18:34:00
5. click on id=cla OK	18:34:01
6. type on id=cla with value 12345 OK	18:34:01
7. click on cas=> btn OK	18:34:02
8. click on linkText=Encuestas OK	18:34:02
9. click on linkText=Listar encuestas OK	18:34:03
10. click on cas=> ng-star-inserted > p .far OK	18:34:04
11. click on id=descripcion OK	18:34:04
12. type on id=descripcion with value descripción encuesta OK	18:34:05
13. click on cas=> createWithCtrl(1) OK	18:34:06
'Modificar información encuesta' completed successfully	18:34:06

Ilustración 53 Prueba crear pregunta encuesta

Running 'Crear pregunta encuesta'	18:53:39
1. open on /randpublic/login OK	18:53:39
2. setWindowSize on 1051x860 OK	18:53:39
3. click on id=usu OK	18:53:39
4. type on id=usu with value fabian.rodriguez@usantofe.edu.co OK	18:53:40
5. click on id=cla OK	18:53:40
6. type on id=cla with value 12345 OK	18:53:40
7. click on cas=> btn OK	18:53:40
8. click on linkText=Encuestas OK	18:53:40
9. click on linkText=Listar encuestas OK	18:53:41
10. mouseOver on linkText=Listar encuestas OK	18:53:41
11. mouseOut on linkText=Listar encuestas OK	18:53:41
12. click on cas=> ng-star-inserted > p .far OK	18:53:41
13. click on id=contact-tab OK	18:53:41
14. click on name=tipo_pregunta OK	18:53:42
15. select on name=tipo_pregunta with value label=Pregunta abierta OK	18:53:42
16. click on cas=> form-control > ng-star-inserted:nth-child(3) OK	18:53:42
17. click on cas=> fa OK	18:53:42
18. click on id=pAbierta OK	18:53:42
19. type on id=pAbierta with value Estas hay? OK	18:53:43
20. click on cas=> s-grid > .cancelar-preguntas:nth-child(1) OK	18:53:43
'Crear pregunta encuesta' completed successfully	18:53:43

Ilustración 54 Prueba modificar preguntas encuesta

Running "Modificar preguntas encuesta"	18:51:27
1. open on /land/public/login OK	18:51:27
2. setWindowSize on 1051x860 OK	18:51:27
3. click on id=usu OK	18:51:27
4. type on id=usu with value fabian.rodriguez@usantola.edu.co OK	18:51:38
5. click on id=cla OK	18:51:38
6. type on id=cla with value 12345 OK	18:51:38
7. click on css= btn OK	18:51:38
8. click on linkText=Encuestas OK	18:51:38
9. click on linkText=Listar encuestas OK	18:51:39
10. mouseOut on linkText=Listar encuestas OK	18:51:39
11. click on css= ng-star-inserted > p, for OK	18:51:39
12. click on id=profile-tab OK	18:51:39
13. click on id=floatingTextareaAbierta OK	18:51:39
14. type on id=floatingTextareaAbierta with value Como estas? OK	18:51:40
15. click on css= ng-star-inserted:nth-child(3) card, fa-solid OK	18:51:40
"Modificar preguntas encuesta" completed successfully	18:51:40

Ilustración 55 Prueba buscador listar encuesta

Running "Buscador listar encuesta"	19:00:25
1. open on /land/public/login OK	19:00:25
2. setWindowSize on 1051x860 OK	19:00:26
3. click on id=usu OK	19:00:26
4. type on id=usu with value fabian.rodriguez@usantola.edu.co OK	19:00:26
5. click on id=cla OK	19:00:26
6. type on id=cla with value 12345 OK	19:00:27
7. click on css= btn OK	19:00:27
8. click on linkText=Encuestas OK	19:00:27
9. click on linkText=Listar encuestas OK	19:00:27
10. mouseOver on linkText=Listar encuestas OK	19:00:27
11. mouseOut on linkText=Listar encuestas OK	19:00:28
12. mouseOver on linkText=Listar encuestas OK	19:00:28
13. mouseOut on linkText=Listar encuestas OK	19:00:28
14. click on id=buscar OK	19:00:28
15. type on id=buscar with value proyecto sistema encuesta OK	19:00:28
16. sendKeys on id=buscar with value \${KEY_ENTER} OK	19:00:29
"Buscador listar encuesta" completed successfully	19:00:29

Ilustración 56 Prueba responder encuesta

Running 'Responder encuesta'	19:12:32
1. open on /and/public/login OK	19:12:33
2. setWindowSize on 1051x860 OK	19:12:33
3. click on id=usu OK	19:12:33
4. type on id=usu with value sara.duke@usantia.edu.co OK	19:12:33
5. click on id=cia OK	19:12:34
6. type on id=cia with value 12345 OK	19:12:34
7. click on css=> btn OK	19:12:34
8. click on linkText=Encuestas OK	19:12:34
9. click on linkText=Latar encuestas OK	19:12:35
10. mouseOver on linkText=Latar encuestas OK	19:12:35
11. mouseOut on linkText=Latar encuestas OK	19:12:35
12. click on css=> .ng-star-inserted:nth-child(2) > .tr OK	19:12:35
13. click on id=fecha OK	19:12:35
14. type on id=fecha with value 2022-11-23 OK	19:12:36
15. click on css=> .row:nth-child(3) #resOpcion OK	19:12:36
16. click on id=resAbierta OK	19:12:36
17. type on id=resAbierta with value bien OK	19:12:36
18. click on css=> .ng-star-inserted:nth-child(5) #resAbierta OK	19:12:37
19. type on css=> .ng-star-inserted:nth-child(5) #resAbierta with value jueves OK	19:12:37
20. type on css=> .ng-star-inserted:nth-child(5) #resAbierta with value miercoles OK	19:12:37
21. click on css=> .btn-primary OK	19:12:37
22. runScript on window.scrollTo(0,0) OK	19:12:37
'Responder encuesta' completed successfully	19:12:38

Ilustración 57 Prueba resultados encuesta

Running 'Resultado encuesta'	17:28:12
1. open on /and/public/login OK	17:28:12
2. setWindowSize on 1280x800 OK	17:28:12
3. click on id=usu OK	17:28:12
4. type on id=usu with value fabian.rodriguez@usantia.edu.co OK	17:28:13
5. click on id=cia OK	17:28:13
6. type on id=cia with value 12345 OK	17:28:13
7. click on css=> btn OK	17:28:13
8. click on linkText=Encuestas OK	17:28:13
9. click on linkText=Latar encuestas OK	17:28:13
10. click on linkText=Resultados OK	17:28:14
'Resultado Encuestas' completed successfully	17:28:14

Ilustración 58 Prueba buscador listar usuarios

Running 'Buscador listar usuarios'	17:44:34
1. open on /and/public/login OK	17:44:34
2. setWindowSize on 1051x857 OK	17:44:35
3. click on id=usu OK	17:44:35
4. type on id=usu with value fabian.rodriguez@usantia.edu.co OK	17:44:36
5. click on id=cia OK	17:44:38
6. type on id=cia with value 12345 OK	17:44:37
7. click on css=> btn OK	17:44:37
8. click on linkText=Usuarios OK	17:44:38
9. mouseOver on linkText=Usuarios OK	17:44:38
10. mouseOut on linkText=Usuarios OK	17:44:39
11. click on id=buscar OK	17:44:40
12. type on id=buscar with value fabian OK	17:44:40
13. sendKeys on id=buscar with value S[KEY_ENTER] OK	17:44:41
14. click on id=buscar OK	17:44:41
15. type on id=buscar OK	17:44:42
'Buscador listar usuarios' completed successfully	17:44:42

Ilustración 59 Prueba registrar usuarios nuevos

Running 'Registrar usuario nuevos'	16:40:24
1. open on /land/public/login OK	16:40:25
2. setWindowSize on 864x853 OK	16:40:27
3. click on linkText=Registre su cuenta aquí OK	16:40:29
4. click on id=nom OK	16:40:31
5. type on id=nom with value laura OK	16:40:33
6. click on id=ape OK	16:40:35
7. type on id=ape with value Gonzales OK	16:40:37
8. click on id=doc OK	16:40:39
9. type on id=doc with value 12345678 OK	16:40:41
10. click on id=cor OK	16:40:43
11. type on id=cor with value laura.gonzales@usantoto.edu.co OK	16:40:45
12. click on id=telefono OK	16:40:47
13. type on id=telefono with value 123456 OK	16:40:49
14. click on name=nomRol OK	16:40:51
15. select on name=nomRol with value label=Ingeniería de sistemas OK	16:40:53
16. click on css=option:nth-child(8) OK	16:40:55
17. click on id=contra OK	16:40:57
18. type on id=contra with value 12345 OK	16:40:59
19. click on id=cla OK	16:41:01
20. type on id=cla with value 12345 OK	16:41:03
21. click on css=.btn OK	16:41:05
'Registrar usuario nuevos' completed successfully	16:41:05

7.4.2. Prueba de aceptación

Esta prueba se realizó con el fin de aceptar los criterios proporcionados en el contrato de desarrollo de software, que se acordaron entre los desarrolladores y el cliente-usuario. Para ello se realizó una reunión con el Stakeholder que en este caso es la decana de la facultad de ingeniería de sistemas, allí se planteó una lista de chequeo con el propósito de validar que cada uno de los requerimientos principales del proyecto se cumplieron a cabalidad. A continuación se adjunta un acta de la reunión que se llevó a cabo y de la lista de chequeo que se tuvo en cuenta para realizar esta prueba de aceptación.

ACTA Nº 13					
Proyecto: DESARROLLO WEB PARA LA ADMINISTRACIÓN DE LAS ENCUESTAS REALIZADAS EN LOS EVENTOS DE LA USTA TUNJA					
LUGAR:	Reunión por medio de Meet	FECHA:	DD	MM	AA
			25	11	2022
HORA DE INICIO:	7:00 pm	HORA DE FINALIZACIÓN:	8:00 pm		
ASISTENTES:	Nombre		Rol		
	Luz Elena Gutiérrez López		Stakeholder		
	Leidy Melissa Marín Fajardo		Estudiante		
	Fabián Danilo Rodríguez Ochoa		Estudiante		
OBJETIVO:	Prueba de aceptación: Esta reunión se realiza con el fin de aceptar los criterios proporcionados en el contrato de desarrollo de software, que se acordaron entre los desarrolladores y el cliente-usuario.				
AGENDA					
1. Saludo 2. intervención de cada uno de los participantes 3. Revisión del proyecto 4. Recomendaciones 5. Firma del acta					
DESARROLLO DE LA REUNIÓN					
Esta reunión se realiza con el fin de aceptar los criterios proporcionados en el contrato de desarrollo de software, que se acordaron entre los desarrolladores del software y el cliente-usuario. A continuación se mencionan los requerimientos que fueron planteados inicialmente y realiza el respectivo check a cada uno de estos requerimientos					

REQUERIMIENTO	CUMPLE	NO CUMPLE
Autenticación de usuarios	X	
Creación de nuevos usuarios	X	
Modificación de datos de usuarios	X	
Asignación de rol al usuario	X	
Creación de encuestas	X	

Asignación de encuestas	X	
Guardar respuestas de las encuestas	X	
Responder encuestas en el aplicativo	X	
Visualización de los resultados de las encuestas	X	
Base de datos relacional	X	
Diseño del sistema	X	
Adaptabilidad del sistema	X	
Permisos a los usuarios (Administrador, Docente, Estudiante, Secretaria e Invitado)	X	
Escalabilidad del proyecto	X	

Se da por terminada la reunión y en constancia se firma el acta correspondiente.
 Firmas.

Luz Elena Gutiérrez López	
Leidy Melissa Marín Fajardo	
Fabián Danilo Rodríguez Ochoa	

Tabla 8 Fase 4 Pruebas

Tipo	Elementos
Entradas	<ul style="list-style-type: none"> • Computador • Prototipo final
Instrumentos y/o herramientas	<ul style="list-style-type: none"> • Conexión a internet • Computador • Herramientas o programas para hacer las pruebas finales <ul style="list-style-type: none"> ✓ Navegador Web ✓ Word ✓ Excel
Salidas	Finalización y aprobación del prototipo desarrollado.
Tiempo	1 semana

Fuente: Autor

8. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

- El desarrollo del proyecto web se llevó a cabo de manera exitosa gracias a la metodología de prototipos y herramientas de código abierto como Angular y Node.js. Fue por ello que se pudo mantener una mejoría constante en el desarrollo del aplicativo web, el cual se enfocó en una arquitectura de servicios que posteriormente iban a ser consumidos.
- Se definió un documento que especifica los requerimientos funcionales y no funcionales necesarios para la construcción del proyecto, el cual permitió puntualizar el comportamiento del sistema y la secuencia de interacciones entre la aplicación web y los actores, para así poder validar que el modelo de solicitud coincidía con la intención del cliente y del usuario.
- La implementación del modelo “4+1” propuesto por Kruchten, se identificó como un patrón bastante apropiado para explicar la arquitectura de un sistema de software, haciendo uso de los diagramas UML (Lenguaje de Modelado Unificado), los cuales permiten que los “Stakeholders” o personas interesadas en el proyecto pueden entender el aplicativo web.
- Las pruebas y resultados obtenidos al momento de evaluar el sistema, permitieron asegurar la funcionalidad y el correcto cumplimiento de cada uno de los requerimientos acordados inicialmente, demostrando así la calidad del proyecto desarrollado y descartando cualquier posible fallo en su desempeño.

9. REFERENCIAS

- [1] CARCAMO, O., CORTES, Y. H. y SALAMANCA, J. C. (2021, julio). PROGRAMACIÓN EVENTO COMERCIAL Y DISEÑO DE LA CAMPAÑA DEDIFUSIÓN. Coursehero.Com. <https://www.coursehero.com/file/116166576/evidencia-de-trabajo-de-eventodocx/>
- [2] Wong, M. (2018, noviembre 12). 21 factors for evaluating event management software. EventMobi. <https://www.eventmobi.com/blog/21-factors-for-evaluating-event-management-software/>
- [3] Gilliam Haije, E. (2017, julio 25). 21 mejores software de encuestas en línea y herramientas de cuestionarios. Mopinion. <https://mopinion.com/es/mejores-software-de-encuestas-en-linea-herramientas-de-cuestionarios/>
- [4] Sabido, M. R. M., & Pech, J. P. U. Proceso Unificado de Desarrollo para Aplicaciones con Arquitectura Orientada a Servicios. <https://www.iiis.org/CDs2008/CD2008CSC/CISCI2008/PapersPdf/C316QM.pdf>
- [5] Mera Paz, J. A. (2016). Análisis del proceso de pruebas de calidad de software. Ingeniería solidaria, 12(20), 163-176. <https://revistas.ucc.edu.co/index.php/in/article/view/1482>
- [6] IEEE Computer Society. (2008). IEEE Standard for Software and System Test Documentation. IEEE Standard 829, Section 3 “Definition”, pp. 8.
- [7] IEEE Computer Society. (2008-1). IEEE Standard for Software and System Test Documentation. IEEE Standard 829, Section 10 “Level test design”, pp. 50. 8.
- [8] IEEE Computer Society. (2008-2). IEEE Standard for Software and System Test Documentation. IEEE Standard 829.

[9] Chaves, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. InterSedes: Revista de las Sedes Regionales, 6(10), 1-13. <https://www.redalyc.org/pdf/666/66612870011.pdf>

[10] Repositorio Digital de la Facultad de Ingeniería UNAM. (s. f.). Metodologías y procesos de análisis de software. Apache Tomcat/8.5.68. <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A5%20Capítulo%202.pdf>

[11] INGENIERÍA DE REQUERIMIENTOS. (s. f.). SEDICI - Repositorio de la Universidad Nacional de La Plata. [http://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2_-_Ingeniería_de_requerimientos.pdf?sequence=4&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2/_Ingeniería_de_requerimientos.pdf?sequence=4&isAllowed=y)

[12] Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. IEEE Internet Computing, 14(6), 80-83. <https://ieeexplore.ieee.org/abstract/document/5617064/citations>

[13] Angular.io. 2021. Angular. [online] Available at: <<https://angular.io/docs>> [Accessed 5 December 2021]

[14] JavaScript with syntax for types. (s/f). Typescriptlang.org. Recuperado el 5 de octubre de 2022, de <https://www.typescriptlang.org/>

[15] Simões, C. (2021, julio 27). ¿Qué es Node.js, y para qué sirve? Blog ITDO - Agencia de desarrollo Web, APPs y Marketing en Barcelona. <https://www.itdo.com/blog/que-es-node-js-y-para-que-sirve/>

[16] Angular. (2021). Angular.io. Recuperado el 5 de octubre de 2022, de <https://angular.io/>

[17] Qué es GitKraken. (2022, mayo 11). KeepCoding Tech School. <https://keepcoding.io/blog/que-es-gitkraken/>

[18] Abraham Hernández, R. (2019). DESARROLLO ÁGIL DEL NUEVO SISTEMA INSTITUCIONAL BASADO EN UNA ARQUITECTURA ORIENTADA A MICROSERVICIOS [Maestría, Universidad de Montemorelos]. <https://dspace.um.edu.mx/bitstream/handle/20.500.11972/1045/Tesis%20Abraham%20Hernández%20Rivadeneira.pdf?sequence=1>



10. ANEXOS

10.1. Anexo 1. Documento de especificación de requerimientos

Especificación de requisitos de software

**Proyecto: DESARROLLO WEB PARA LA
ADMINISTRACIÓN DE LAS ENCUESTAS REALIZADAS
EN LOS EVENTOS DE LA USTA TUNJA**

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
11/11/2022		Leidy Melissa Marin Fajardo Fabián Danilo Rodríguez Ochoa	

Documento validado por las partes en fecha: 13/11/2022

Por el cliente	Por los desarrolladores
Decana de la Facultad de Ingeniería de Sistemas Luz Elena Gutiérrez López	Leidy Melissa Marin Fajardo Fabian Danilo Rodriguez Ochoa

Contenido

FICHA DEL DOCUMENTO	68
CONTENIDO	69
1 INTRODUCCIÓN	71
1.1 Propósito	71
1.2 Alcance	71
1.3 Personal involucrado	71
1.4 Definiciones, acrónimos y abreviaturas	71
1.5 Referencias	72
1.6 Resumen	72
2 DESCRIPCIÓN GENERAL	72
2.1 Perspectiva del producto	72
2.2 Funcionalidad del producto	72
2.2.1 Caso de uso administrador	72
2.2.2 Caso de uso Docente, Invitado y Secretaria	74
2.2.3 Caso de uso estudiante	74
2.3 Características de los usuarios	74
2.3.1 Administrador	74
2.3.2 Secretaria	75
2.3.3 Docente	75
2.3.4 Invitado	75
2.3.5 Estudiante	76
2.4 Restricciones	76
2.5 Suposiciones y dependencias	76
3 REQUISITOS ESPECÍFICOS	76
3.1 Requisitos comunes de los interfaces	79

3.1.1	Interfaces de usuario	79
3.1.2	Interfaces de hardware	80
3.1.3	Interfaces de software	80
3.1.4	Interfaces de comunicación	80
3.2	Requisitos funcionales	80
3.2.1	Requisito funcional 1	80
3.2.2	Requisito funcional 2	80
3.2.3	Requisito funcional 3	81
3.2.4	Requisito funcional 4	81
3.2.5	Requisito funcional 5	81
3.2.6	Requisito funcional 6	82
3.2.7	Requisito funcional 7	82
3.2.8	Requisito funcional 8	82
3.2.9	Requisito funcional 9	82
3.2.10	Requisito funcional 10	83
3.2.11	Requisito funcional 11	83
3.2.12	Requisito funcional 12	83
3.2.13	Requisito funcional 13	83
3.3	Requisitos no funcionales	83
3.3.1	Requisitos de rendimiento	83
3.3.2	Seguridad	84
3.3.3	Fiabilidad	84
3.3.4	Mantenibilidad	84
3.3.5	Portabilidad	84
4	APÉNDICES	84
4.1	Diagrama de clases	84
4.2	Diagrama de actividades	85

1 Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el proyecto **Desarrollo web para la administración de las encuestas realizadas en los eventos de la USTA Tunja**. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998

1.1 Propósito

Construir una herramienta informática, para la generación de instrumentos de evaluación de eventos realizados en la Universidad Santo Tomás Seccional Tunja. El desarrollo web va dirigido a la facultad Ingeniería de Sistemas de la Universidad Santo Tomás Seccional Tunja.

1.2 Alcance

Mediante este desarrollo web se busca que la información recopilada acerca de estos eventos quede almacenada y se pueda manejar de manera sistematizada, logrando así, que haya un mejor control de los resultados que dejan las encuestas de estos eventos, y de esta manera, se puedan realizar análisis sus respectivos resultados.

1.3 Personal involucrado

Nombre	Leidy Melissa Marin Fajardo
Rol	Desarrolladora
Categoría profesional	Estudiante de ingeniería de sistemas
Responsabilidades	Análisis de información, diseño y programación del proyecto
Información de contacto	leidy.marin@usantoto.edu.co

Nombre	Fabián Danilo Rodríguez Ochoa
Rol	Desarrollador
Categoría profesional	Estudiante de ingeniería de sistemas
Responsabilidades	Análisis de información, diseño y programación del proyecto
Información de contacto	fabian.rodriguez@usantoto.edu.co

Nombre	Luis Fernando Castellanos Guarín
Rol	Director del proyecto
Categoría profesional	Ingeniero de Sistemas
Responsabilidades	Orientar el desarrollo del proyecto
Información de contacto	luis.castellanosg@usantoto.edu.co

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
API	Application Programming Interfaces (interfaz de programación de aplicaciones)
CRUD	Create, Read, Update, Delete (Crear, Listar, Actualizar y Eliminar)
RF	Requisito funcional
NF	Requisito no funcional
MER	Modelo Entidad Relación

1.5 Referencias

Título del documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6 Resumen

Este documento está conformado por tres secciones. En el primer apartado se realiza una introducción al mismo y se proporciona una visión general de la especificación de requerimientos del sistema.

En el segundo apartado del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones a las que debe responder el proyecto, los datos asociados, factores, restricciones, supuestos y dependencias que afectan al desarrollo.

Por último, en el tercer apartado del documento se definen detalladamente los requisitos que debe satisfacer el sistema

2 Descripción general

2.1 Perspectiva del producto

El sistema de información orientado a la web, que se propone en este desarrollo, busca satisfacer las necesidades de la Universidad Santo Tomás Seccional Tunja, en torno a la complejidad que se presenta al evaluar la información obtenida de las encuestas en los eventos realizados por parte de la Universidad

2.2 Funcionalidad del producto

2.2.1 Caso de uso Administrador

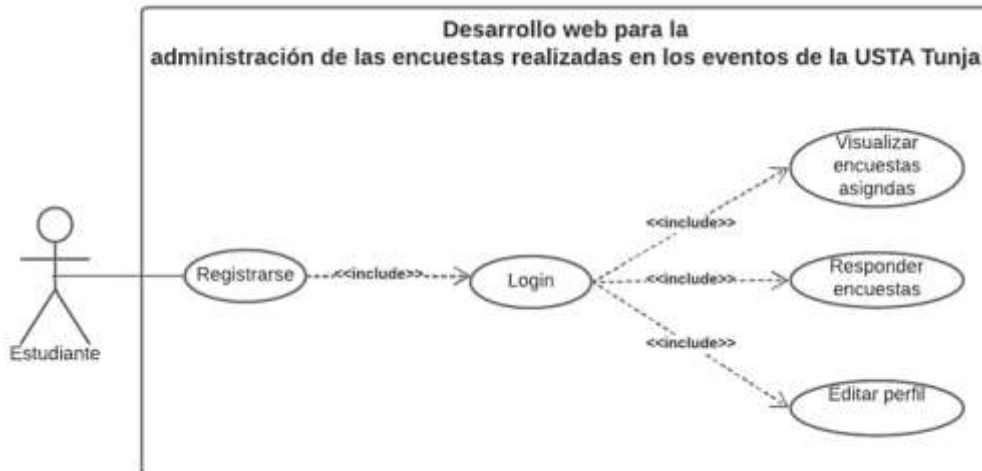
ESPECIFICACIÓN DE REQUERIMIENTOS DESARROLLO WEB PARA LA ADMINISTRACIÓN DE LAS ENCUESTAS REALIZADAS EN LOS EVENTOS DE LA USTA TUNJA



2.2.2 Caso de uso Docente, Invitado y Secretaria



2.2.3 Caso de uso Estudiante



2.3 Características de los usuarios

2.3.1 Administrador

Tipo de usuario	Administrador del sistema
Formación	Administrador
Actividades	<ul style="list-style-type: none"> • Acceder al sistema (Login) • Crear usuarios • Visualizar usuarios existente en el sistema • Modificar usuarios • Visualizar roles

	<ul style="list-style-type: none"> • Modificar estado del rol • Crear encuestas • Visualizar encuestas realizadas • Visualizar los resultados • Filtrar búsquedas • Editar el perfil
--	--

2.3.2 Secretaria

Tipo de usuario	Secretaria
Formación	Profesional en asistencia administrativa
Actividades	<ul style="list-style-type: none"> • Registrarse • Acceder al sistema (Login) • Editar el perfil • Crear encuestas • Visualizar encuestas realizadas • Visualizar los resultados • Filtrar búsquedas

2.3.3 Docente

Tipo de usuario	Docente
Formación	Educador de la universidad
Actividades	<ul style="list-style-type: none"> • Registrarse • Acceder al sistema (Login) • Editar el perfil • Crear encuestas • Visualizar encuestas realizadas • Visualizar los resultados • Filtrar búsquedas

2.3.4 Invitado

Tipo de usuario	Invitado
Formación	N/A
Actividades	<ul style="list-style-type: none"> • Registrarse • Acceder al sistema (Login) • Editar el perfil • Crear encuestas • Visualizar encuestas realizadas • Visualizar los resultados • Filtrar búsquedas

2.3.5 Estudiante

Tipo de usuario	Estudiante
Formación	Alumno de la universidad
Actividades	<ul style="list-style-type: none"> • Registrarse • Acceder al sistema (Login) • Editar el perfil • Filtrar búsquedas • Responder encuestas

2.4 Restricciones

- Interfaz para ser usada con internet.
- Lenguajes y tecnologías en uso: HTML, TYPESCRIPT.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

2.5 Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos indicados para garantizar una ejecución correcta de la misma

3 Requisitos específicos

Requerimientos Funcionales

Identificación del requerimiento	RF01
Nombre de requerimiento	Ingreso al aplicativo
Descripción del requerimiento	Para el ingreso al aplicativo, se debe ingresar con el correo y la respectiva contraseña.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF02
Nombre de requerimiento	Creación de usuarios
Descripción del requerimiento	El administrador podrá crear usuarios llenando los campos establecidos en el formulario.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF03
Nombre de requerimiento	Visualización de usuarios registrados en el sistema
Descripción del requerimiento	El administrador podrá ver los usuarios que están registrados en el aplicativo, mediante una lista.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF04
Nombre de requerimiento	Modificación de usuarios
Descripción del requerimiento	El administrador podrá modificar datos del usuario como: nombres, apellidos, número y tipo de documento, teléfono, correo, rol, estado e imagen del usuario.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF05
Nombre de requerimiento	Filtros de búsqueda
Descripción del requerimiento	Los usuarios contarán con filtros de búsquedas que permitan que sus búsquedas sean más rápidas y precisas.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF06
Nombre de requerimiento	Visualización de los roles existentes en el sistema
Descripción del requerimiento	El sistema le permitirá al administrador visualizar los roles que actualmente existen en el sistema.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF07
Nombre de requerimiento	Modificación de los roles
Descripción del requerimiento	El administrador podrá modificar el estado de los roles.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF08
Nombre de requerimiento	Cantidad de usuarios por cada rol
Descripción del requerimiento	El administrador podrá ver cuántos usuarios tiene asignados cada uno de los roles.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF09
---	------

Nombre de requerimiento	Creación de encuestas
Descripción del requerimiento	Para crear una encuesta es necesario llenar el formulario con sus respectivos campos.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF10
Nombre de requerimiento	Modificación de encuestas
Descripción del requerimiento	La plantilla de las encuestas se podrá modificar, cabe resaltar que esto no implica modificar resultados.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF11
Nombre de requerimiento	Visualización de encuestas realizadas
Descripción del requerimiento	Cada usuario podrá ver las encuestas que ha realizado.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF12
Nombre de requerimiento	Visualización de los resultados de cada encuesta
Descripción del requerimiento	Cada encuesta debe mostrar los resultados obtenidos al finalizar la fecha de cierre.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF13
Nombre de requerimiento	Guardar respuestas en el sistema
Descripción del requerimiento	El aplicativo debe permitir que se puedan responder encuestas y las respuestas deben quedar guardadas de una vez en la base de datos del sistema.
Prioridad del requerimiento	Alta

Requerimientos No Funcionales

Identificación del requerimiento	RNF 01
Nombre de requerimiento	Interfaz del sistema.
Descripción del requerimiento	El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF 02
---	--------

Nombre de requerimiento	Ayuda en el uso del sistema.
Descripción del requerimiento	El sistema deberá de tener un manual de usuario para facilitar la manipulación del sistema.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF 03
Nombre de requerimiento	Diseño de la interfaz
Descripción del requerimiento	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF 04
Nombre de requerimiento	Desempeño
Descripción del requerimiento	Garantizar el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF 05
Nombre de requerimiento	Nivel de usuario
Descripción del requerimiento	Garantizar al usuario el acceso de información de acuerdo al nivel que posee.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF 06
Nombre de requerimiento	Seguridad de la información
Descripción del requerimiento	El sistema garantizará a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
Prioridad del requerimiento	Alta

3.1 Requisitos comunes de los interfaces

3.1.1 Interfaces de usuario

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

3.1.2 Interfaces de hardware

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

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

3.1.3 Interfaces de software

- Sistema Operativo: Windows 10 o superior.
- Explorador: Mozilla o Chrome.

3.1.4 Interfaces de comunicación

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible y necesario.

3.2 Requisitos funcionales

3.2.1 Requisito funcional 1

Ingreso al aplicativo: Para el ingreso al aplicativo, se debe ingresar con el correo y la respectiva contraseña. Para esto se llenará un formulario con dos campos de texto, uno para el correo y otro para la contraseña, y un botón para ejecutar el ingreso.

- ✓ **Registrarse en el sistema:** Si no tiene una cuenta para poder ingresar al sistema, deberá registrarse, llenando todos los datos que pide el formulario para que el registro se haga de manera exitosa, posterior a ello podrá ingresar al aplicativo con el correo y la contraseña que utilizó en el registro.

3.2.2 Requisito funcional 2

Creación de usuarios: El administrador podrá crear usuarios llenando el formulario, donde irán los siguientes campos:

- Nombre: Allí deberá poner el nombre del usuario (obligatorio).
- Apellido: Allí deberá poner los apellidos del usuario (obligatorio).
- Correo: Allí deberá poner el correo del usuario (obligatorio).

- Contraseña: Allí deberá poner una contraseña, la cual después puede ser modificada por el usuario (obligatorio).
 - Tipo de documento: Indicar que tipo de documento tiene el usuario (obligatorio).
 - Documento: Indica el número de documento del usuario (obligatorio).
 - Teléfono: Número de contacto telefónico del usuario (obligatorio).
 - Estado: Por defecto el estado del usuario es activo (obligatorio).
 - Confirmar contraseña: Para mayor seguridad, se debe confirmar la contraseña (obligatorio).
 - Rol: Selecciona el rol que le desea asignar al usuario (obligatorio).
- ✓ **Asignación del estado para el usuario:** Por defecto la creación de cada usuario quedará con un estado de ACTIVO, el cual posteriormente puede ser modificado.
- ✓ **Asignación de rol para el usuario:** El administrador podrá asignar el rol de Docente, Secretaria, Invitado o Estudiante según corresponda el caso.

3.2.3 Requisito funcional 3

Visualización de usuarios registrados en el sistema: El administrador podrá ver los usuarios que están registrados en el aplicativo, mediante una lista.

3.2.4 Requisito funcional 4

Modificación de usuarios: El administrador podrá modificar datos del usuario como: nombres, apellidos, número y tipo de documento, teléfono, correo, rol, estado e imagen del usuario.

3.2.5 Requisito funcional 5

Filtros de búsqueda: Los usuarios contarán con filtros de búsquedas que permitan que sus búsquedas sean más rápidas y precisas.

- ✓ **Filtro de búsqueda para los usuarios existentes en el sistema:** El administrador podrá realizar una búsqueda de usuarios por medio del nombre, documento o teléfono.
- ✓ **Filtro de búsqueda para los roles existentes en el sistema:** El administrador podrá buscar un rol por su nombre.
- ✓ **Filtro de búsqueda para las encuestas realizadas:** Cada encuesta debe contar con un filtro de búsqueda, que permita hacer una consulta de manera más ágil, ya sea por el nombre del evento, el tipo de evento o la facultad a la cual fue asignada

3.2.6 Requisito funcional 6

Visualización de los roles existentes en el sistema: El administrador podrá visualizar los roles que actualmente existen en el sistema. Para este proyecto se manejarán 5 roles: Administrador, Docente, Invitado, Secretaria y Estudiante. Cabe destacar que no contará con la opción de agregar roles.

3.2.7 Requisito funcional 7

Modificación de los roles: El administrador podrá modificar el estado de los roles.

- ✓ **Modificación del estado del rol:** Por defecto cada rol se creó con un estado ACTIVO, el cual después puede ser modificado.

3.2.8 Requisito funcional 8

Cantidad de usuarios por cada rol: El administrador podrá ver cuántos usuarios tiene asignados cada uno de los roles.

3.2.9 Requisito funcional 9

Creación de encuestas: Para crear una encuesta es necesario llenar el formulario donde irán los siguientes campos de texto:

- Nombre: Evento al que va dirigida la encuesta. (Obligatorio)
- Fecha de creación: Indica el día en que se crea la encuesta. (Obligatorio)
- Fecha de cierre: Indica el día en que se cierra la encuesta. (Obligatorio)
- Dependencia: A que facultad va dirigida la encuesta. Para esto habrá una lista desplegable donde aparecerán las opciones permitidas (Obligatorio)
- Tipo de dependencia: Dependiendo la dependencia que elija, así mismo podrá seleccionar el respectivo tipo de dependencia (obligatorio)
- Descripción: Indica una breve descripción a que hace referencia la encuesta. (Obligatorio)

- Tipo de pregunta: Esta opción le permitirá al usuario seleccionar que pregunta desea utilizar, cada tipo de pregunta viene incluida con sus posibles opciones. Cabe resaltar que solo se manejan 4 tipos de preguntas: pregunta abierta, pregunta de fecha, pregunta de selección única y pregunta de escala (del 1 al 5). No se tendrá ninguna opción para agregar más preguntas. (Obligatorio)
- Pregunta: Allí debe diligenciar la pregunta a responder. (Obligatorio)
- ✓ **Asignación de encuestas:** Cada encuesta realizada debe ser asignada a una facultad. Para que posterior a ello, los estudiantes puedan responder la encuesta, para esto se contará con una lista desplegable que permite elegir la opción necesaria.

3.2.10 Requisito funcional 10

Modificación de encuestas: La plantilla de las encuestas se podrán modificar, cabe resaltar que esto no implica modificar resultados.

3.2.11 Requisito funcional 11

Visualización de encuestas realizadas: Cada usuario podrá ver las encuestas que ha realizado.

3.2.12 Requisito funcional 12

Visualización de los resultados de cada encuesta: Cada encuesta debe mostrar los resultados obtenidos al finalizar la fecha de cierre.

3.2.13 Requisito funcional 13

Guardar respuestas en el sistema: El aplicativo debe permitir que se puedan responder encuestas y las respuestas deben quedar guardadas de una vez en la base de datos del sistema.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

3.3.2 Seguridad

- Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
- Brindar seguridad al sistema con respecto a la información y datos que se manejan tales como documentos, archivos o contraseñas.
- Facilidades y controles para permitir el acceso a la información al personal autorizado, con la intención de consultar o subir información.

3.3.3 Fiabilidad

- El sistema debe tener una interfaz de uso intuitiva y sencilla.
- El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.

3.3.4 Mantenibilidad

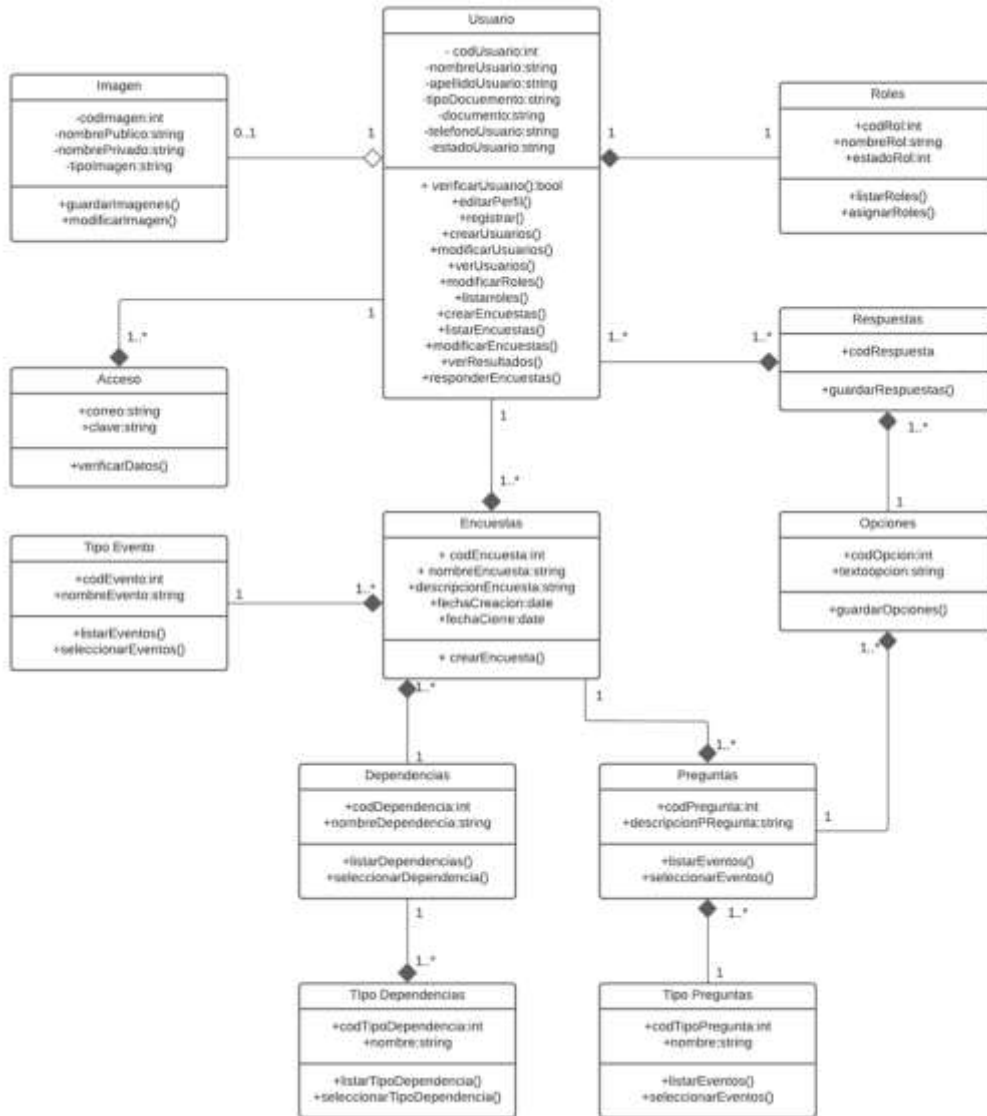
- El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento de manera fácil.
- La interfaz debe estar complementada con un manual de usuario que le facilite al usuario la manipulación del sistema.

3.3.5 Portabilidad

El sistema será creado bajo la plataforma de Windows.

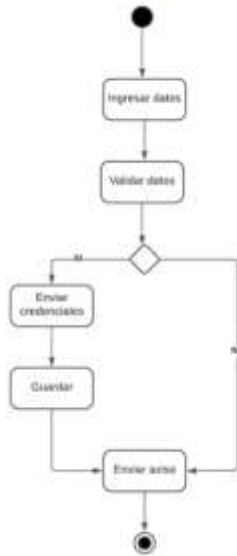
4 Apéndices

4.1 Diagrama de clases

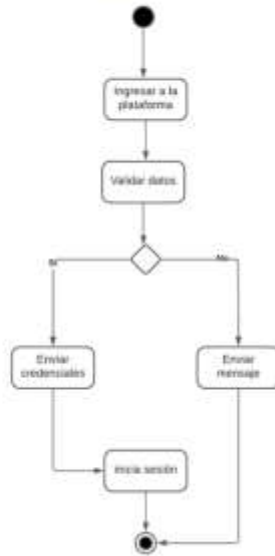


4.2 Diagrama de actividades

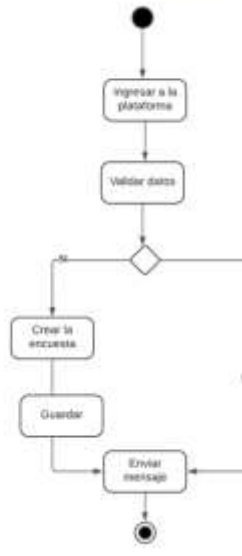
Registrar



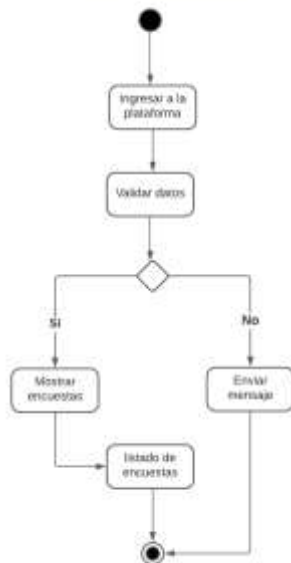
Iniciar sesión



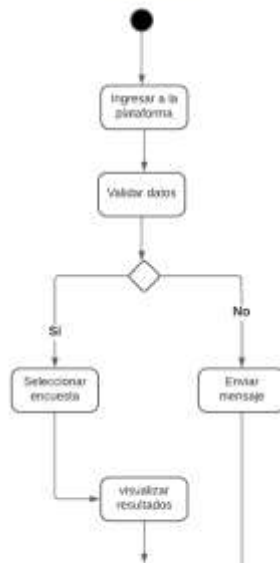
Crear encuestas



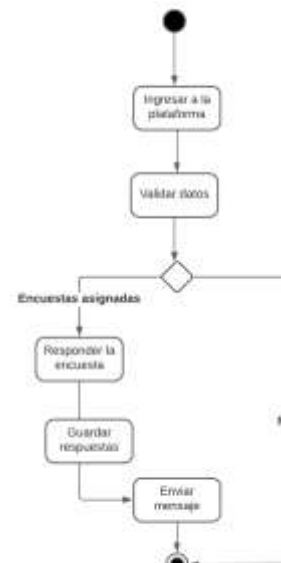
Listar encuestas

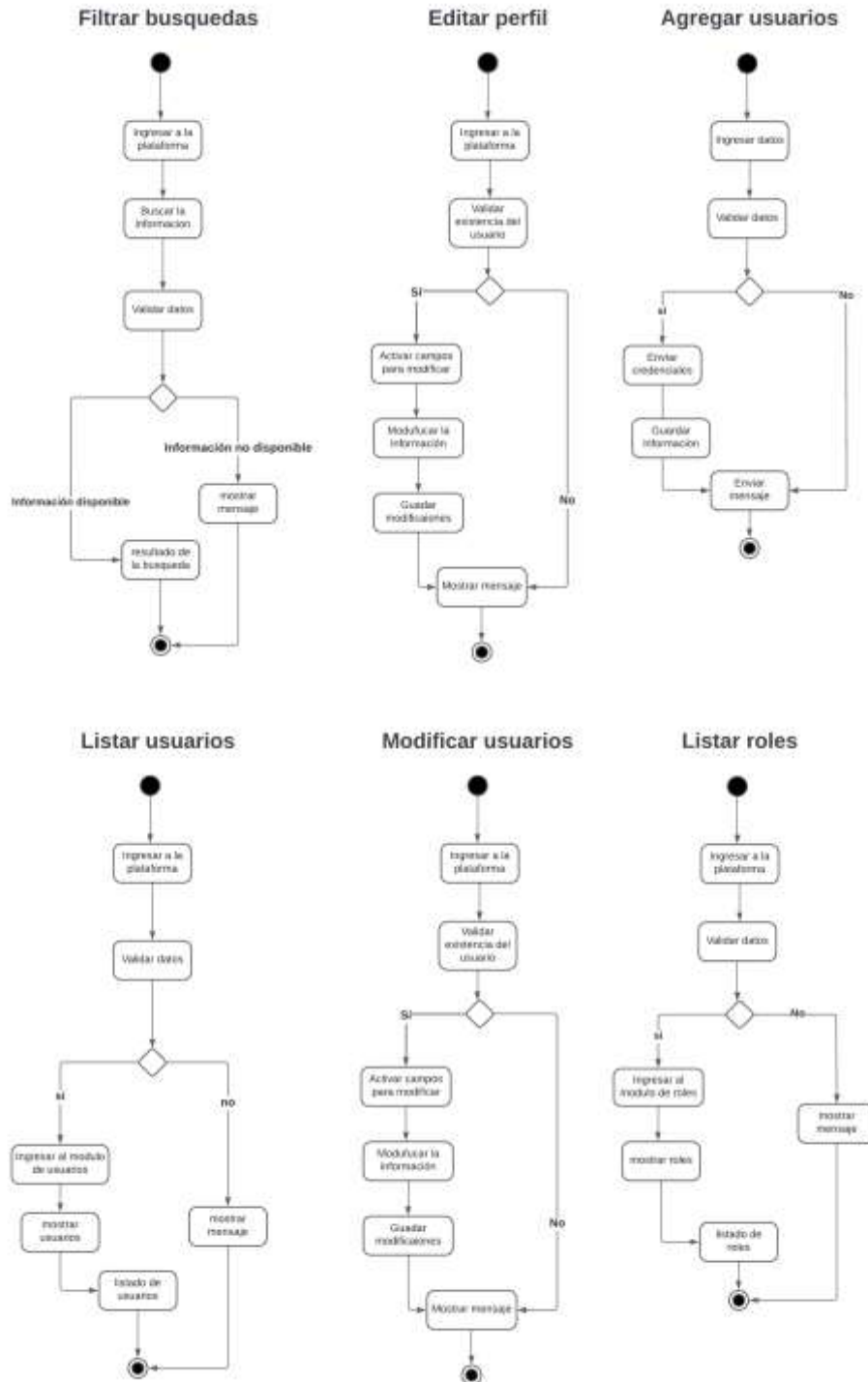


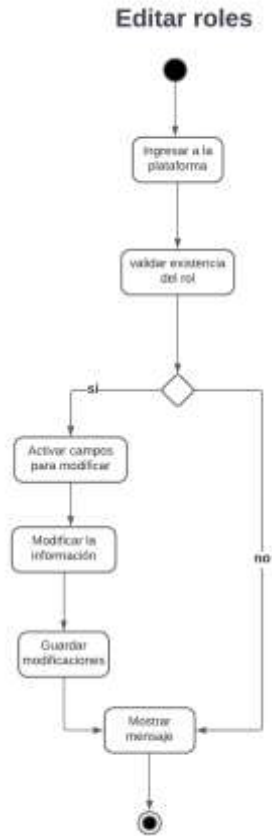
Ver resultados



Responder encuestas







10.2. Anexo 2. Manual de usuario



MANUAL DE USUARIO

Desarrollo web para la administración de las encuestas realizadas en los eventos de la
USTA Tunja

Presentado por:

Leidy Melissa Marín Fajardo
1.099.217.312
2251490

Fabián Danilo Rodríguez Ochoa
1.051.476.909
2251163

Director

Luz Elena Gutiérrez López

Codirector

Luis Fernando Castellanos Guarín

Tunja
Noviembre de 2022

Tabla de contenido

1. ADMINISTRADOR	91
1.1. Ingresar al aplicativo.....	91
1.2. Editar el perfil.....	91
1.3. Roles	92
1.3.1. Listar roles	92
1.3.2. Buscar roles.....	93
1.3.3. Editar roles	93
1.4. Usuarios	94
1.4.1. Listar usuarios	94
1.4.2. Buscar usuarios.....	95
1.4.3. Agregar usuarios	95
1.4.4. Editar datos de usuario.....	96
1.5. Encuestas.....	97
1.5.1. Creación de encuestas.....	97
1.5.2. Visualización de encuestas realizadas.....	98
1.5.3. Buscar encuestas	99
1.5.4. Visualización de resultados.....	99
1.5.5. Modificar encuestas.....	100
2. ESTUDIANTE	102
2.1. Registrarse en la plataforma	102
2.2. Ingresar al aplicativo.....	102
2.3. ISTAR ENCUESTAS	103
2.4. Buscar encuestas	103
2.5. Responder encuesta	103
2.6. Editar perfil	104
3. DOCENTE, SECRETARIA E INVITADO	106
3.1. Registrarse en la plataforma	106
3.2. Ingresar al aplicativo.....	106
3.3. Editar perfil:	107
3.4. Creación de encuestas.....	107
3.5. Visualización de encuestas realizadas.....	109
3.6. Buscar encuestas	110
3.7. MODIFICAR ENCUESTA.....	110
3.8. Visualización de resultados:.....	111

1. ADMINISTRADOR

1.1. INGRESAR AL APLICATIVO

Para ingresar al sistema por primera vez como administrador, habrá un usuario creado de manera predeterminada con sus respectivas credenciales, las cuales serán:

Usuario: C

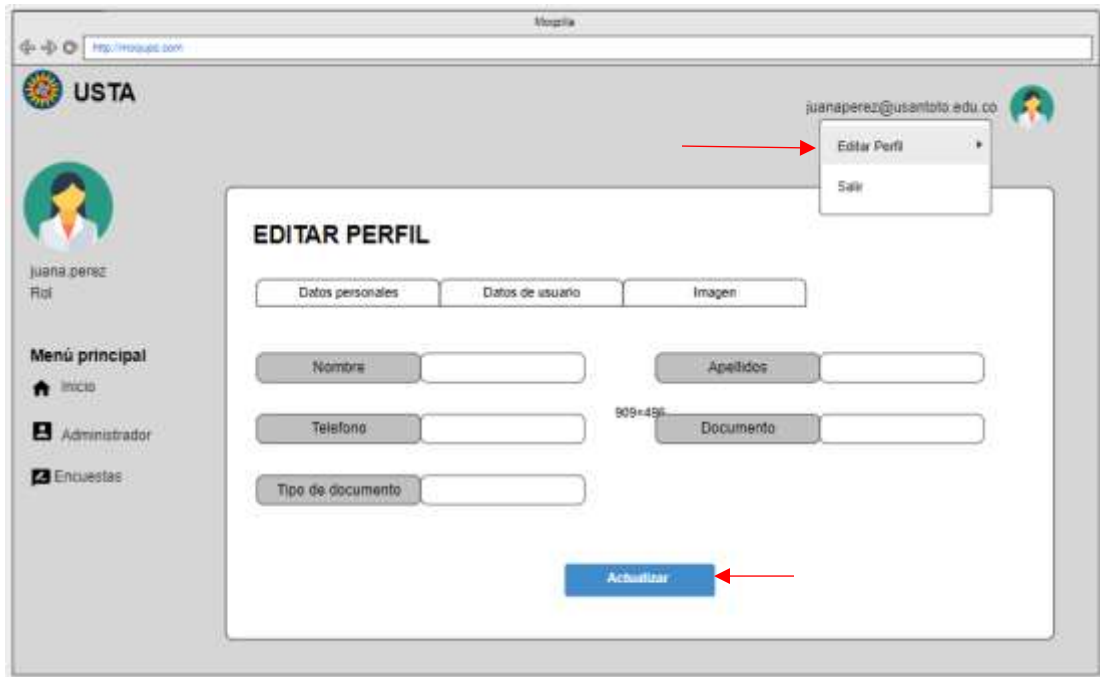
Contraseña: administrador123

El administrador debe ingresar a la plataforma con su respectivo usuario y contraseña para posteriormente hacer su correcta entrada al aplicativo. Finalmente selecciona la opción **Iniciar sesión**.



1.2. EDITAR EL PERFIL

En la parte superior derecha, el usuario despliega una lista donde aparecerá la opción de **Editar perfil**, allí puede cambiar datos que considere necesarios como: imagen, nombres o apellidos, correo, contraseña, documento y tipo de documento. Después de haber realizado los cambios que consideraba pertinentes, selecciona la opción **Actualizar**.

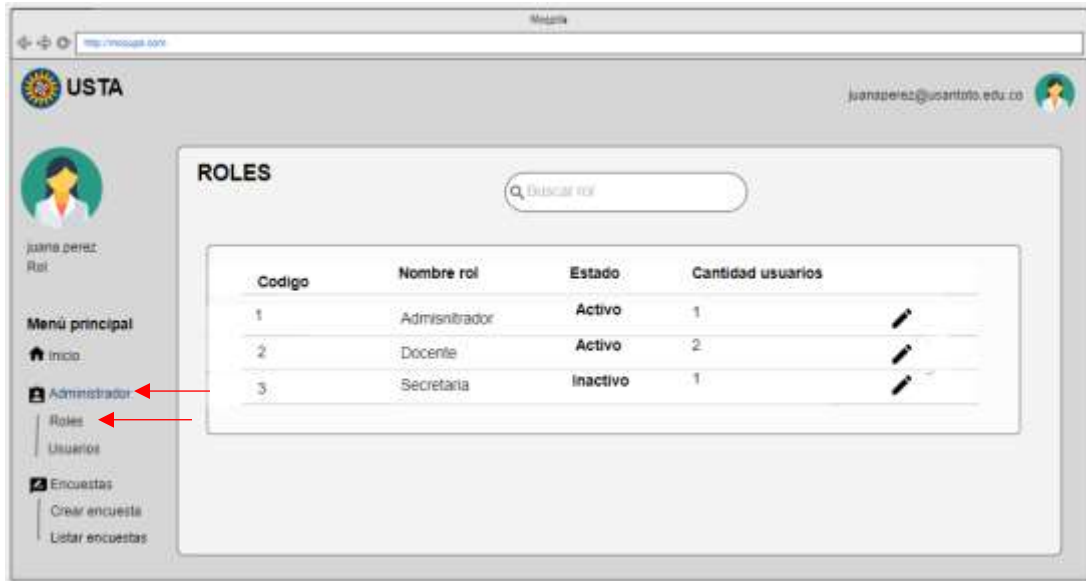


1.3. ROLES

1.3.1. Listar roles

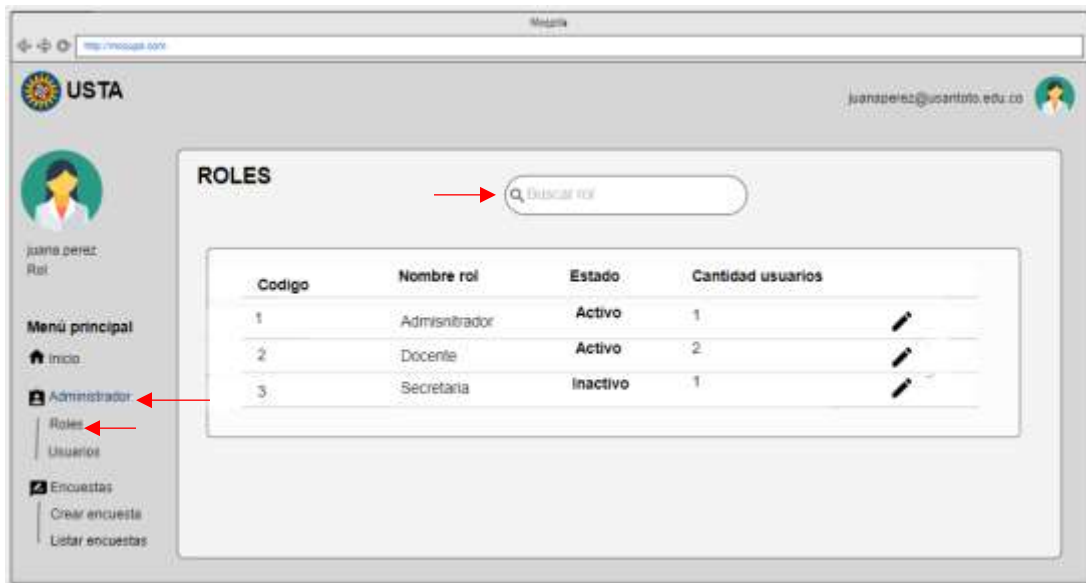
El administrador puede listar los roles que existen actualmente en la plataforma. Para ello debe seleccionar en el menú la opción **Administrador**, y de allí se despliegan otras opciones, y selecciona la opción **Roles**. En esta lista aparecen los siguientes datos:

- Código del rol
- Nombre del rol
- Estado del rol
- Cantidad de usuarios que tiene asignado ese rol
- Opciones de modificar el rol



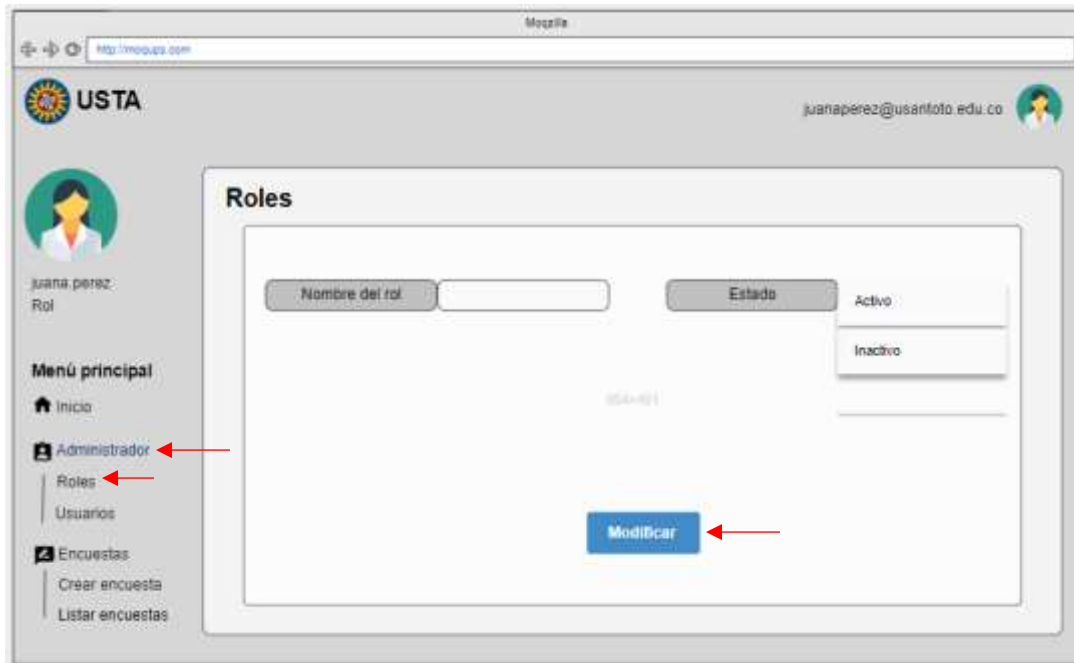
1.3.2. Buscar roles

El administrador puede buscar un rol en específico por medio del nombre, el cual se digita en la barra de búsqueda y a medida que va escribiendo se va ejecutando la búsqueda



1.3.3. Editar roles

El administrador puede editar estado del rol si lo considera necesario, para ello debe posicionarse sobre la opción editar del rol que desea modificar, y finalmente seleccionar la opción actualizar rol.



1.4. USUARIOS

1.4.1. Listar usuarios

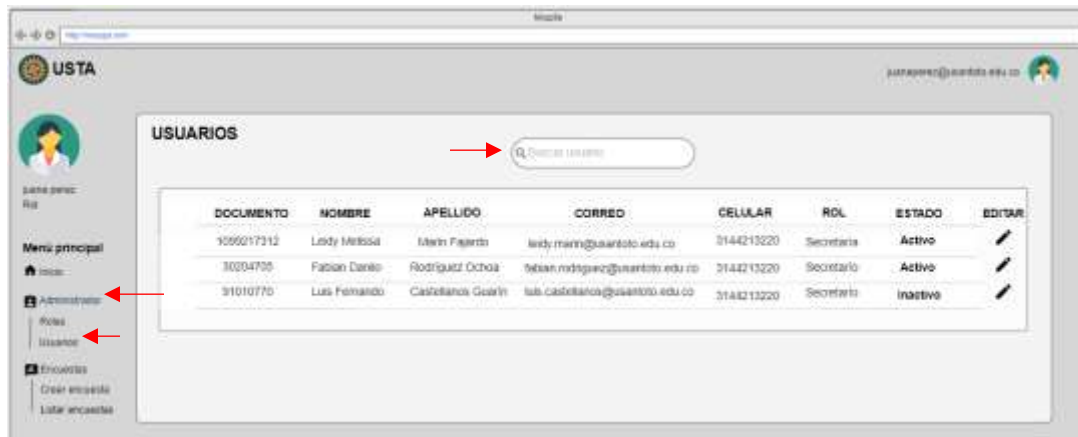
El administrador puede visualizar los usuarios registrados en la plataforma. En este listado aparecen los siguientes datos de usuario:

- Código
- Nombre
- Apellido
- Documento
- Correo
- Rol
- Celular
- Estado



1.4.2. Buscar usuarios

El administrador puede buscar un usuario en específico a través de la barra de búsqueda, este filtro lo puede hacer mediante el nombre, documento, apellido o correo.



1.4.3. Agregar usuarios

El administrador puede agregar cuantos usuarios sean necesarios, para ello deberá ingresar al módulo de listar usuarios, allí seleccionar la opción **Agregar usuario**, y posteriormente llenar los siguientes campos que aparecen en el formulario, finalmente selecciona la opción de crear. Todos los campos son obligatorios.

- Nombre
- Apellido
- Correo
- Contraseña
- Tipo de documento
- Documento
- Teléfono

- Estado
- Confirmar contraseña
- Rol

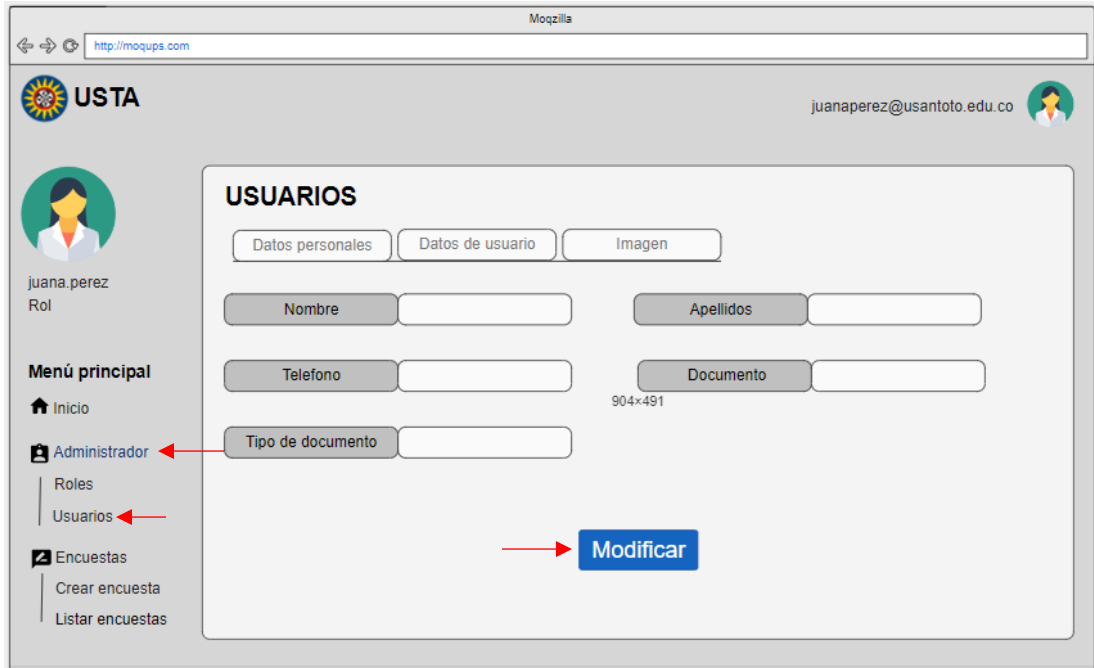


The screenshot shows a web browser window with the URL 'http://vtrouaga.com'. The page header includes the USTA logo and the user email 'juanaperez@usantofe.edu.co'. The main content area is titled 'USUARIOS' and contains a form with the following fields: 'Nombre', 'Apellidos', 'Telefono', 'Correo', 'Estado', 'Rol', 'Contraseña', 'Confirmar contraseña', and 'Tipo de documento'. A blue 'Crear' button is located at the bottom right of the form. The left sidebar shows a user profile for 'juana.perez' and a 'Menú principal' with options: 'Inicio', 'Administrador', 'Roles', 'Usuarios', 'Encuestas', 'Crear encuesta', and 'Listar encuestas'. Red arrows point to 'Administrador', 'Usuarios', and 'Crear'.

1.4.4. Editar datos de usuario

El administrador puede editar los siguientes datos de un usuario, para ello deberá buscar el listado de usuarios y seleccionar la opción de editar, y finalmente selecciona la opción de actualizar datos.

- Imagen
- Nombre
- Apellido
- Correo
- Contraseña
- Tipo de documento
- Documento
- Teléfono
- Estado
- Confirmar contraseña
- Rol



1.5. ENCUESTAS

1.5.1. Creación de encuestas

Para ello el usuario debe ingresar a la plataforma, en el menú seleccionar la opción de encuestas, allí se despliega una lista de opciones y elige la opción de **Crear encuesta**. La encuesta debe crearse con los siguientes campos, y finalmente seleccionar **Enviar**.

- Nombre: Evento al que va dirigida la encuesta. (obligatorio)
- Fecha de creación: Indica el día en que se crea la encuesta. (obligatorio)
- Fecha de cierre: Indica el día en que se cierra la encuesta. (obligatorio)
- Dependencia: A que facultad va dirigida la encuesta. (obligatorio)
- Tipo de evento: Indica el tipo de evento al que va dirigida la encuesta (seminario, web vinar, claustro, charla, simposio, foro, etc.) (obligatorio)
- Descripción: Indica una breve descripción a que hace referencia la encuesta. (obligatorio)
- Tipo de pregunta: Esta opción le permitirá al usuario seleccionar que pregunta desea utilizar, cada tipo de pregunta viene incluida con sus posibles respuestas. Cabe resaltar que solo hay 4 tipos de pregunta: abierta, fecha, selección única y de escala. (obligatorio)
- Pregunta: Allí debe diligenciar la pregunta a responder. Además cuenta con la opción de eliminar la pregunta. (obligatorio).
- Opciones: De acuerdo al tipo de pregunta deberá poner las opciones que ofrece la pregunta. Además cuenta con la opción de eliminar opciones. (obligatorio)

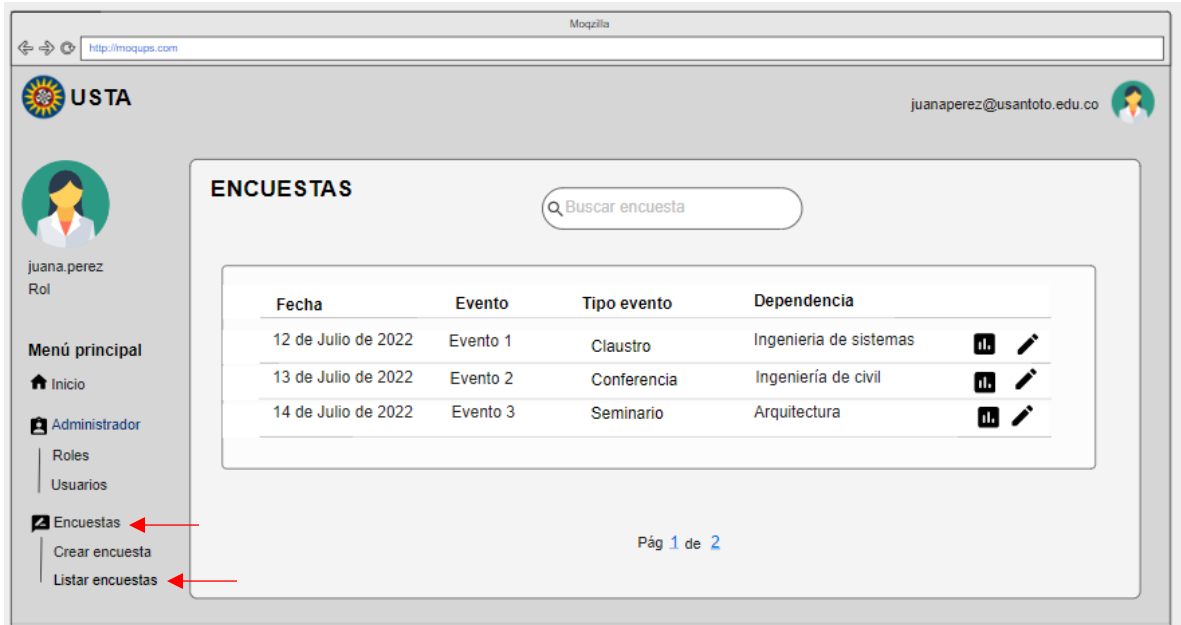
MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA

The screenshot shows the USTA system interface for creating a survey form. The page title is "FERIA DEL LIBRO". The interface includes a sidebar menu with options: Inicio, Administrador, Encuestas, Crear encuesta, and Listar encuestas. The main content area contains a form with the following fields: Fecha inicio (08/12/2022), Fecha cierre (08/12/2022), Tipo evento (Select), and Dependencias (Select). Below these fields is a section titled "Descripción del formulario" with three question types: "Pregunta selección única" (with radio buttons for "Ejemplo" and "Añadir opción"), "Pregunta respuesta abierta" (with text input fields), and "Pregunta de fecha" (with a date picker set to 03/12/2022). A "Crear" button is located at the bottom right of the form.

1.5.2. Visualización de encuestas realizadas

El administrador podrá ver las encuestas realizadas, donde aparecerán datos como la fecha de creación y cierre, el evento y la dependencia a la cual fue asignada la encuesta

MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA



Moqzila
http://moqups.com

USTA
juanaperez@usantoto.edu.co

juana.perez
Rol

ENCUESTAS

Fecha	Evento	Tipo evento	Dependencia		
12 de Julio de 2022	Evento 1	Claustro	Ingenieria de sistemas		
13 de Julio de 2022	Evento 2	Conferencia	Ingeniería de civil		
14 de Julio de 2022	Evento 3	Seminario	Arquitectura		

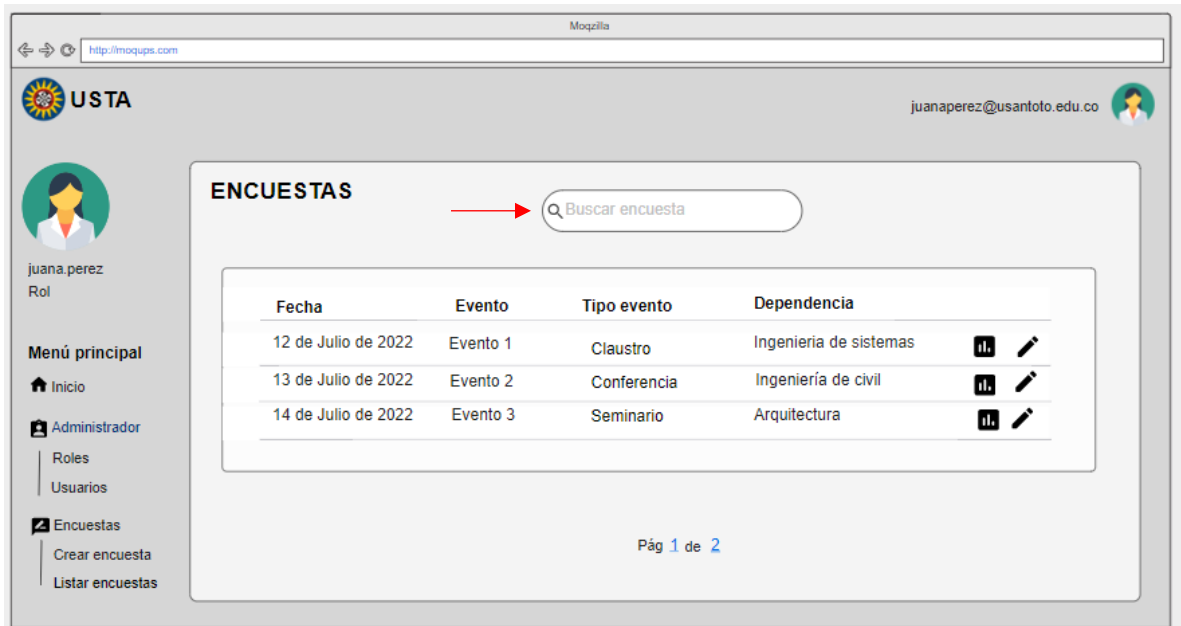
Pág 1 de 2

Menú principal

- Inicio
- Administrador
 - Roles
 - Usuarios
- Encuestas**
 - Crear encuesta
 - Listar encuestas

1.5.3. Buscar encuestas

Para buscar una encuesta es específico, lo puede hacer por medio de la barra de búsqueda, allí puede digitar el nombre de la encuesta o del evento, y de esta manera se irá ejecutando la búsqueda



Moqzila
http://moqups.com

USTA
juanaperez@usantoto.edu.co

juana.perez
Rol

ENCUESTAS

Fecha	Evento	Tipo evento	Dependencia		
12 de Julio de 2022	Evento 1	Claustro	Ingenieria de sistemas		
13 de Julio de 2022	Evento 2	Conferencia	Ingeniería de civil		
14 de Julio de 2022	Evento 3	Seminario	Arquitectura		

Pág 1 de 2

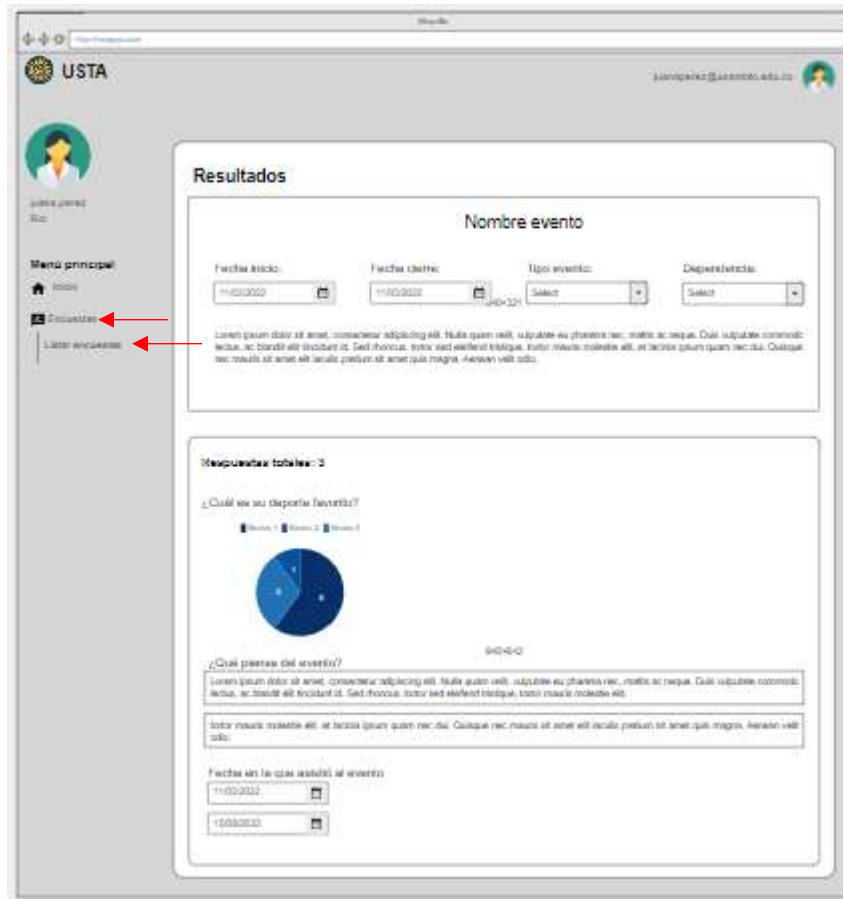
Menú principal

- Inicio
- Administrador
 - Roles
 - Usuarios
- Encuestas**
 - Crear encuesta
 - Listar encuestas

1.5.4. Visualización de resultados

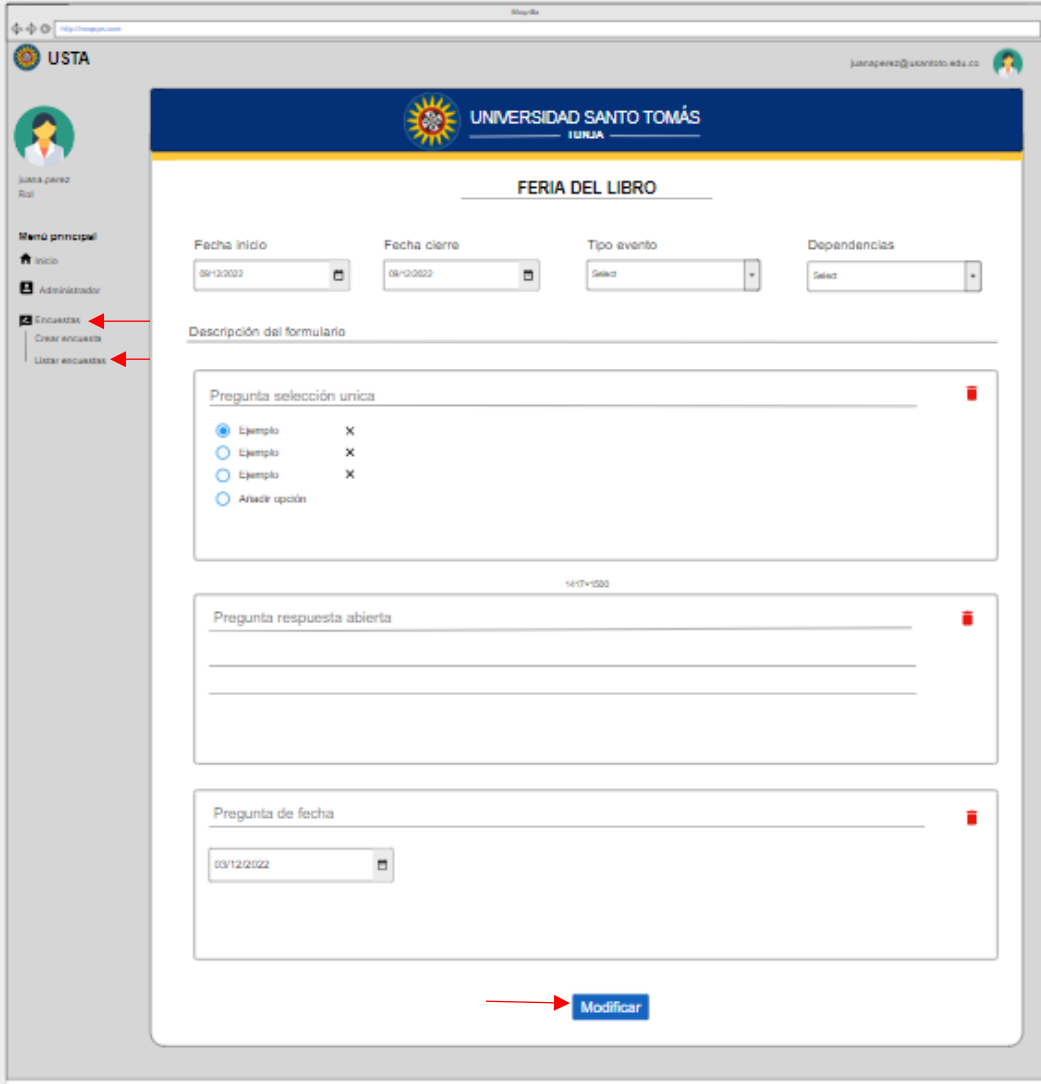
MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA

El administrador puede ver las estadísticas de los resultados de la encuesta de cada evento. Para ello se debe posicionar sobre la opción de listar encuestas, y posterior a esto seleccionar el resultado de la encuesta que desea ver



1.5.5. Modificar encuestas

Para ello el usuario debe ingresar a la plataforma, en el menú seleccionar la opción de encuestas, allí se despliega una lista de opciones y elige la opción de **Listar encuesta**. Posterior a esto debe seleccionar la encuesta que desea modificar. Allí aparecen los datos de la encuesta que puede modificar y finalmente seleccionar **Modificar**.



The screenshot shows the USTA system interface for creating a survey form. The page title is "FERIA DEL LIBRO". The interface includes a sidebar with a "Menú principal" containing "Inicio", "Administrador", "Encuestas", "Crear encuestas", and "Listar encuestas". The main content area has a header with the USTA logo and the event title. Below the header, there are four input fields: "Fecha inicio" (09/12/2022), "Fecha cierre" (09/12/2022), "Tipo evento" (Select), and "Dependencias" (Select). The "Descripción del formulario" section contains three question types: "Pregunta selección única" (with three radio button options: "Ejemplo", "Ejemplo", "Ejemplo", and an "Añadir opción" link), "Pregunta respuesta abierta" (with three text input lines), and "Pregunta de fecha" (with a date input field showing "09/12/2022"). A "Modificar" button is located at the bottom right of the form, with a red arrow pointing to it. The sidebar also has red arrows pointing to the "Encuestas" and "Crear encuestas" items.

2. ESTUDIANTE

2.1. REGISTRARSE EN LA PLATAFORMA

El estudiante puede registrarse en la plataforma, para esto debe ingresar los siguientes datos: nombre, apellidos, documento, correo, contraseña y la confirmación de la contraseña. Todos los campos son obligatorios para realizar el registro de manera exitosa. Posterior a esto seleccionara la opción **Registrar**.



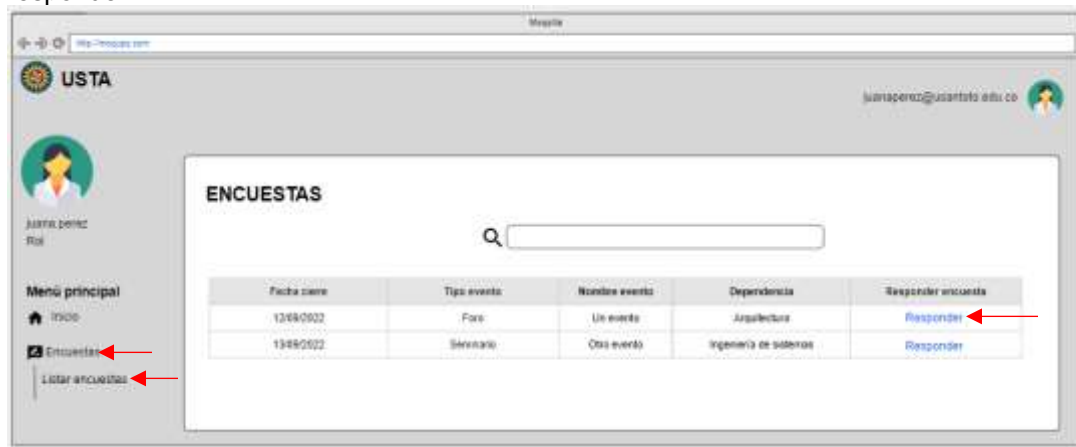
2.2. INGRESAR AL APLICATIVO

El estudiante puede ingresar al aplicativo con su respectivo correo y clave. Posterior a esto selecciona la opción de **Iniciar sesión**.



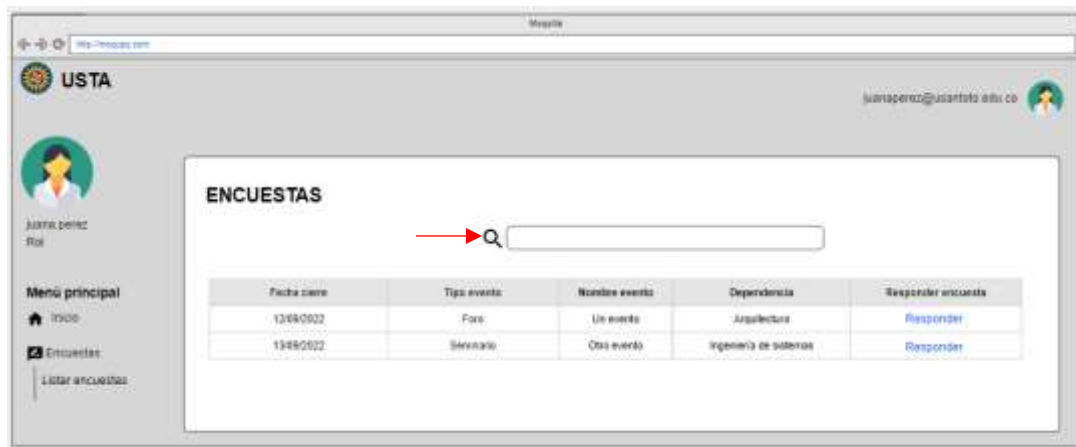
2.3. LISTAR ENCUESTAS

El estudiante puede ver las encuestas que tiene para responder, ingresando al aplicativo y selecciona el módulo de encuestas, listar y posterior a esto selecciona la encuesta que desea responder



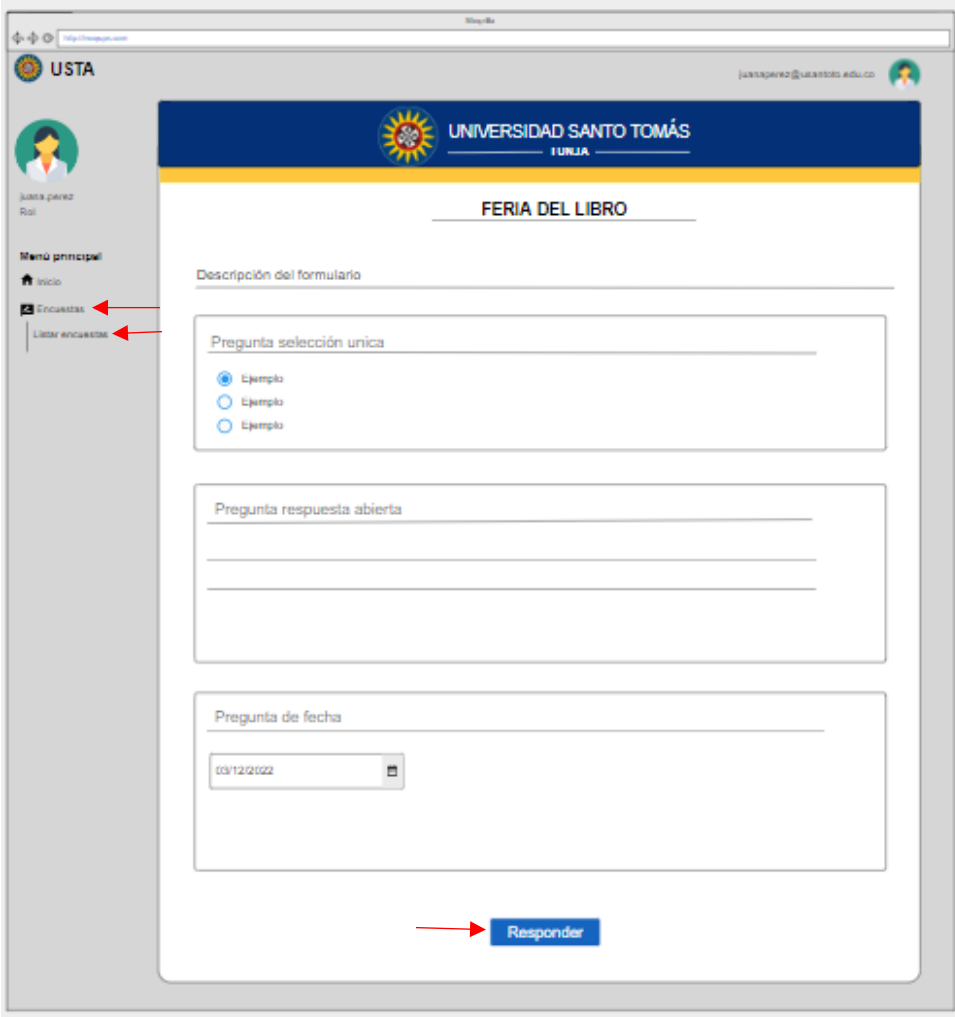
2.4. BUSCAR ENCUESTAS

El estudiante puede filtrar las encuestas seleccionando su facultad, de esta manera solo aparecerán las encuestas que pertenecen a su facultad. Para ellos en la barra de búsqueda deberá poner el nombre de su facultad.



2.5. RESPONDER ENCUESTA

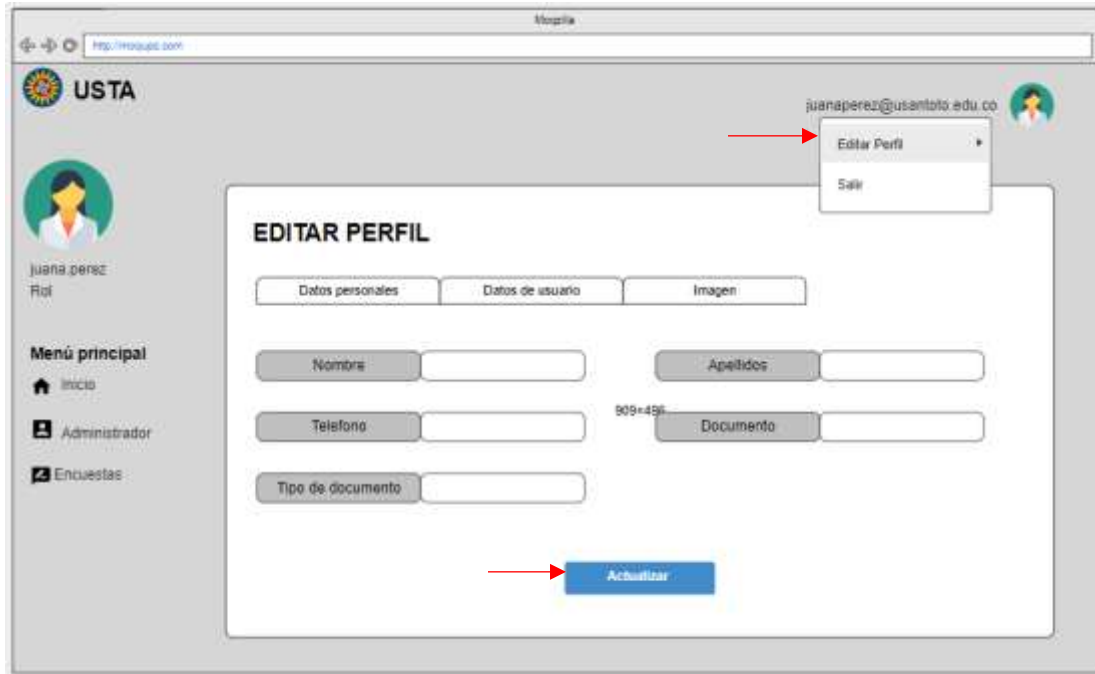
El estudiante deberá responder todas las preguntas para que la acción se ejecute de manera exitosa



The screenshot displays the USTA survey system interface. At the top, there is a header with the USTA logo and the text 'UNIVERSIDAD SANTO TOMÁS TUNJA'. Below the header, the survey title 'FERIA DEL LIBRO' is centered. The main content area is titled 'Descripción del formulario' and contains three question types: 'Pregunta selección única' with three radio button options labeled 'Ejemplo', 'Pregunta respuesta abierta' with two text input fields, and 'Pregunta de fecha' with a date picker showing '03/12/2022'. A blue 'Responder' button is located at the bottom right of the form. On the left side, there is a sidebar menu with the following items: 'Inicio', 'Encuestas', and 'Listar encuestas'. Red arrows point to the 'Encuestas' and 'Listar encuestas' items. At the top left of the sidebar, there is a user profile section for 'Jana Perez' with the role 'Rol'.

2.6. EDITAR PERFIL

En la parte superior derecha, el usuario despliega una lista donde aparecerá la opción de Editar perfil, allí puede cambiar datos que considere necesarios como: imagen, nombres o apellidos, correo, contraseña, documento y tipo de documento. Después de haber realizado los cambios que consideraba pertinentes, selecciona la opción Actualizar.



3. DOCENTE, AUXILIAR E INVITADO

3.1. REGISTRARSE EN LA PLATAFORMA

Puede registrarse en la plataforma, para esto debe ingresar los siguientes datos: nombre, apellidos, documento, correo, contraseña y la confirmación de la contraseña. Todos los datos son obligatorios para hacer el registro de manera exitosa. Posterior a esto seleccionara la opción Registrar.



The image shows a registration form titled "Crear cuenta". It contains six input fields arranged in two columns: "Nombre", "Apellidos", "Documento", "Correo", "Contraseña", and "Confirmar contraseña". Below the fields is a blue "Registrar" button with a red arrow pointing to it. At the bottom of the form, there is a link that says "¿OLVIDASTE TU CONTRASEÑA?".

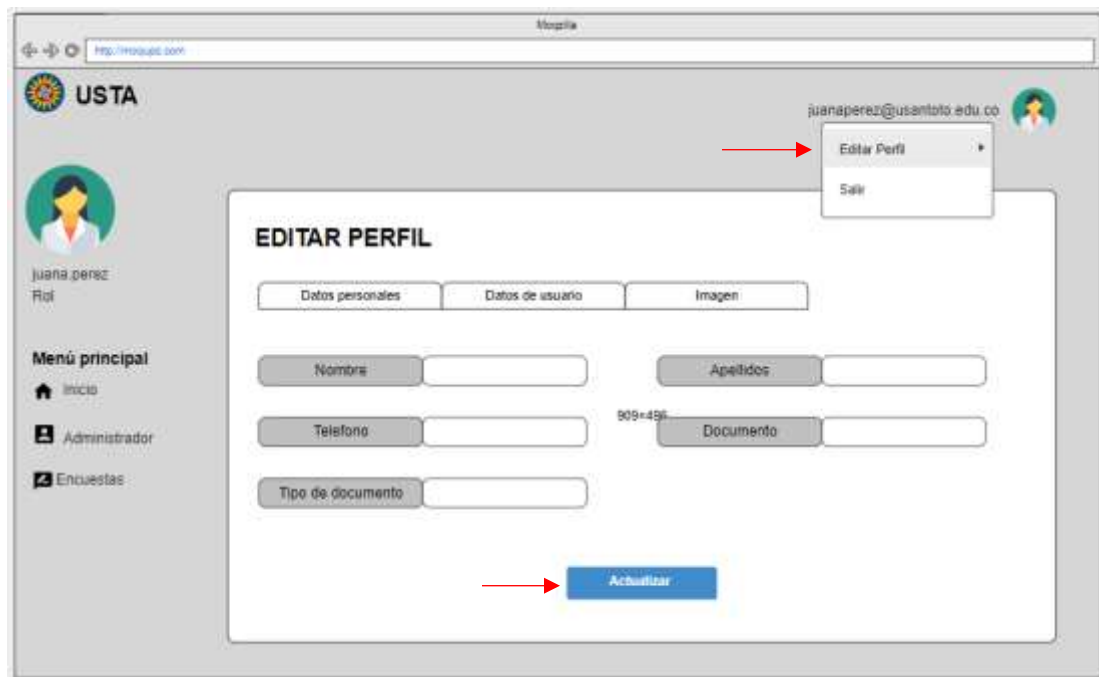
3.2. INGRESAR AL APLICATIVO

Debe ingresar al aplicativo con su respectivo correo y clave, posterior a esto debe comunicarse con el administrador para cambiar el rol de ingreso, ya que por defecto todos los usuarios que se registren ingresan con el rol de estudiante.



3.3. EDITAR PERFIL

En la parte superior derecha, el usuario despliega una lista donde aparecerá la opción de Editar perfil, allí puede cambiar datos que considere necesarios como: imagen, nombres o apellidos, correo, contraseña, documento y tipo de documento. Después de haber realizado los cambios que consideraba pertinentes, selecciona la opción Actualizar.

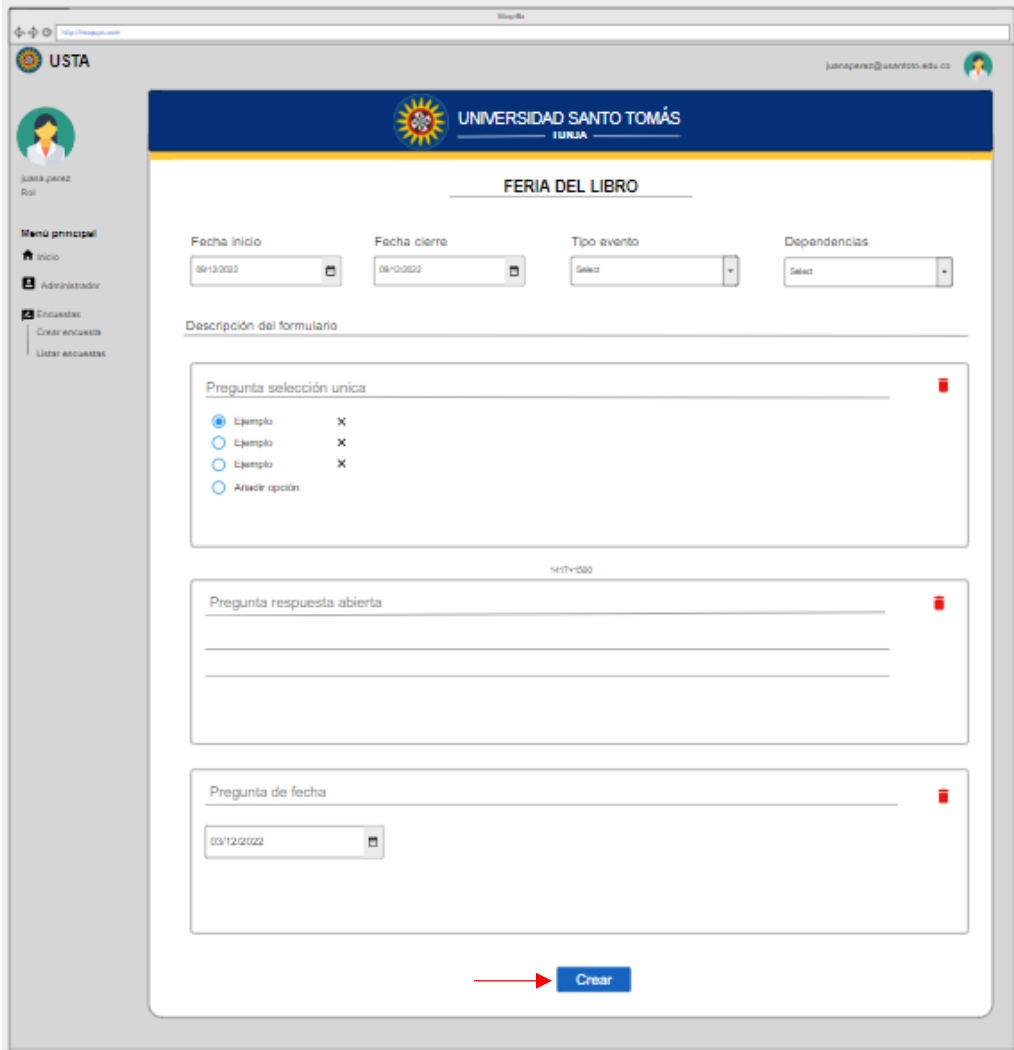


3.4. CREACIÓN DE ENCUESTAS

MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA

Para ello el usuario debe ingresar a la plataforma, en el menú seleccionar la opción de encuestas, allí se despliega una lista de opciones y elige la opción de **Crear encuesta**. La encuesta debe crearse con los siguientes campos, y finalmente seleccionar **Crear**.

- Nombre: Evento al que va dirigida la encuesta. (obligatorio)
- Fecha de creación: Indica el día en que se crea la encuesta. (obligatorio)
- Fecha de cierre: Indica el día en que se cierra la encuesta. (obligatorio)
- Dependencia: A que facultad va dirigida la encuesta. (obligatorio)
- Tipo de evento: Indica el tipo de evento al que va dirigida la encuesta (seminario, webinar, claustro, charla, simposio, foro, etc.) (obligatorio)
- Descripción: Indica una breve descripción a que hace referencia la encuesta. (obligatorio)
- Tipo de pregunta: Esta opción le permitirá al usuario seleccionar que pregunta desea utilizar, cada tipo de pregunta viene incluida con sus posibles respuestas. Cabe resaltar que solo hay 4 tipos de pregunta: abierta, fecha, selección única y de escala. (obligatorio)
- Pregunta: Allí debe diligenciar la pregunta a responder. Además cuenta con la opción de eliminar la pregunta. (obligatorio).
- Opciones: De acuerdo al tipo de pregunta deberá poner las opciones que ofrece la pregunta. Además cuenta con la opción de eliminar opciones. (obligatorio)

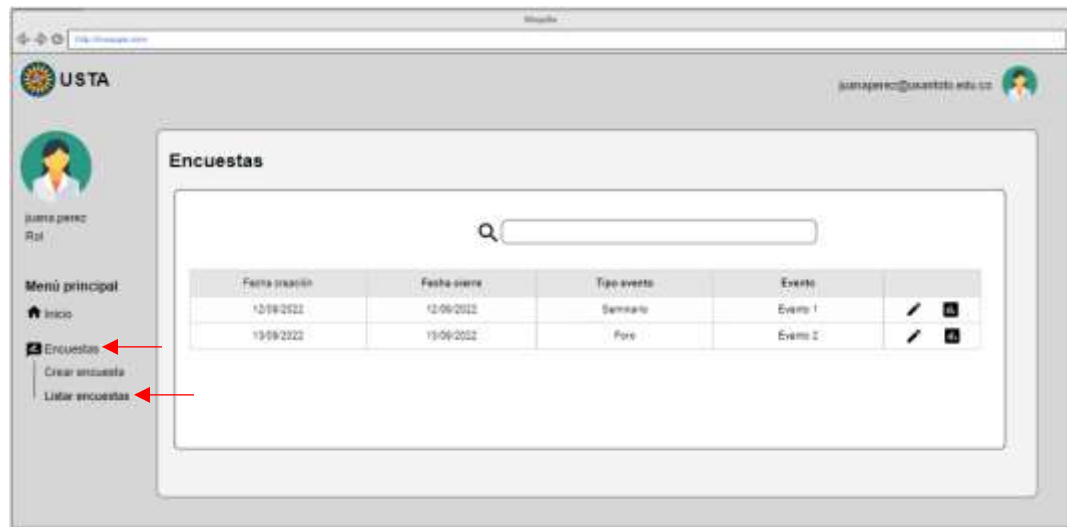


The screenshot shows the USTA system interface for creating a survey form. The page title is "FERIA DEL LIBRO". The form includes fields for "Fecha inicio" (09/12/2022), "Fecha cierre" (09/12/2022), "Tipo evento" (Select), and "Dependencias" (Select). Below these fields is a "Descripción del formulario" section with three question types: "Pregunta selección única" (with radio buttons for "Ejemplo" and "Añadir opción"), "Pregunta respuesta abierta" (with text input fields), and "Pregunta de fecha" (with a date picker set to 03/12/2022). A red arrow points to a blue "Crear" button at the bottom right.

3.5. VISUALIZACIÓN DE ENCUESTAS REALIZADAS

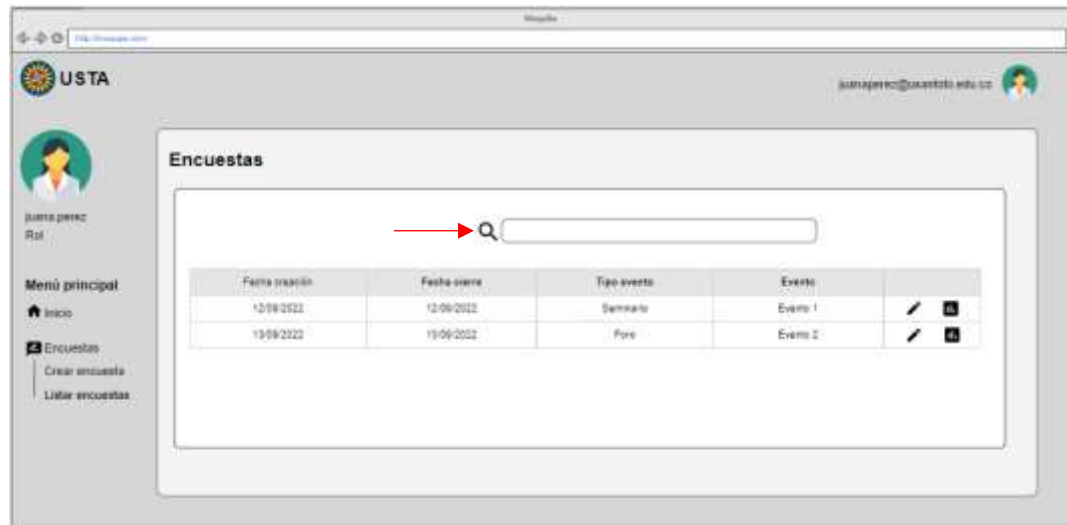
Para ver las encuestas realizadas, debe seleccionar el módulo de encuestas, de allí seleccionar la opción listar encuestas, donde aparecerán datos como la fecha de creación y cierre, el evento y la dependencia a la cual fue asignada la encuesta

MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA



3.6. BUSCAR ENCUESTAS

Para buscar una encuesta es específico, lo puede hacer por medio de la barra de búsqueda, allí puede digitar el nombre de la encuesta o del evento, y de esta manera se ira ejecutando la búsqueda



3.7. MODIFICAR ENCUESTA

Para ello el usuario debe ingresar a la plataforma, en el menú seleccionar la opción de encuestas, allí se despliega una lista de opciones y elige la opción de Listar encuesta. Posterior a esto debe seleccionar la encuesta que desea modificar. Allí aparecen los datos de la encuesta que puede modificar y finalmente seleccionar Modificar.

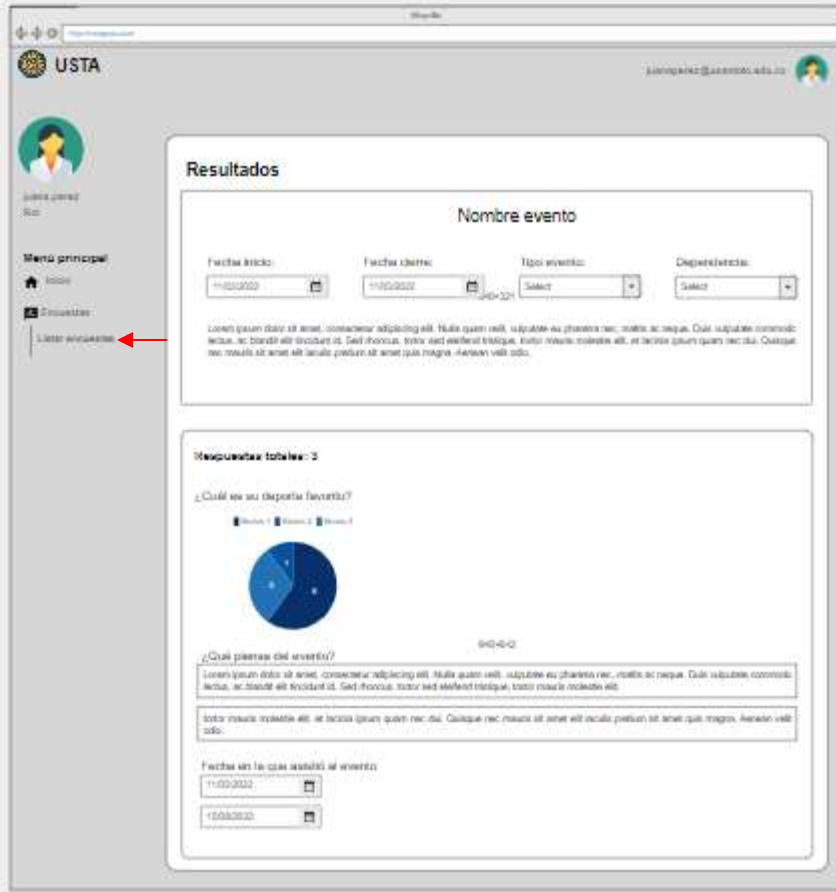
MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA

The screenshot shows the USTA system interface for creating a survey form. The page title is "FERIA DEL LIBRO". The interface includes a sidebar with a user profile for "Jana Perez" and a "Menú principal" with options: "Inicio", "Encuestas" (highlighted with a red arrow), "Crear encuesta", and "Listar encuestas" (highlighted with a red arrow). The main content area has a header for "UNIVERSIDAD SANTO TOMÁS TUNJA" and a form for "FERIA DEL LIBRO". The form includes fields for "Fecha inicio" (09/12/2022), "Fecha cierre" (09/12/2022), "Tipo evento" (Select), and "Dependencias" (Select). Below these is a "Descripción del formulario" section with three question types: "Pregunta selección única" (with radio buttons for "Ejemplo" and "Añadir opción"), "Pregunta respuesta abierta" (with text input fields), and "Pregunta de fecha" (with a date input field). A "Modificar" button is located at the bottom right of the form area, with a red arrow pointing to it.

3.8. VISUALIZACIÓN DE RESULTADOS

El usuario puede ver las estadísticas de los resultados de la encuesta de cada evento. Para ello se debe posicionar sobre la opción de listar encuestas, y posterior a esto seleccionar el resultado de la encuesta que desea ver

MANUAL DE USUARIO SISTEMA DE ENCUESTAS USTA TUNJA




The screenshot shows a web application interface for survey results. At the top left is the USTA logo and a user profile icon. A main navigation menu on the left includes 'Inicio' and 'Encuestas', with 'Lista encuestas' highlighted and pointed to by a red arrow. The main content area is titled 'Resultados' and contains a form for filtering results by 'Nombre evento', 'Fecha inicio', 'Fecha cierre', 'Tipo evento', and 'Deporte/evento'. Below the form is a section for 'Respuestas totales: 3' with a pie chart and a question: '¿Cuál es su deporte favorito?'. A second question, '¿Cuál piensa del evento?', is also visible with a text input field and a date range selector.

10.3. Anexo 3. Diseño de Mockups


Ilustración 60 Ingreso al aplicativo

Mozilla

http://moqups.com

 UNIVERSIDAD SANTO TOMÁS
TUNJA

Acceso al sistema



Usuario

Clave

Entrar

[Registre su cuenta aquí](#)

Ilustración 61 Crear cuenta

The image shows a web browser window with the URL <http://moqups.com>. The page header features the logo of Universidad Santo Tomás Tunja, which consists of a sunburst emblem and the text "UNIVERSIDAD SANTO TOMÁS TUNJA". Below the header is a form titled "Crear cuenta". The form contains the following fields:

- Nombre:
- Apellidos:
- Documento: (with a small "920x636" label above the field)
- Correo:
- Contraseña:
- Confirmar Contraseña:

At the bottom of the form is a blue "Registrar" button and a link that says "Ingrese con su cuenta aquí".

Ilustración 62 Inicio



Ilustración 63 Listar usuarios

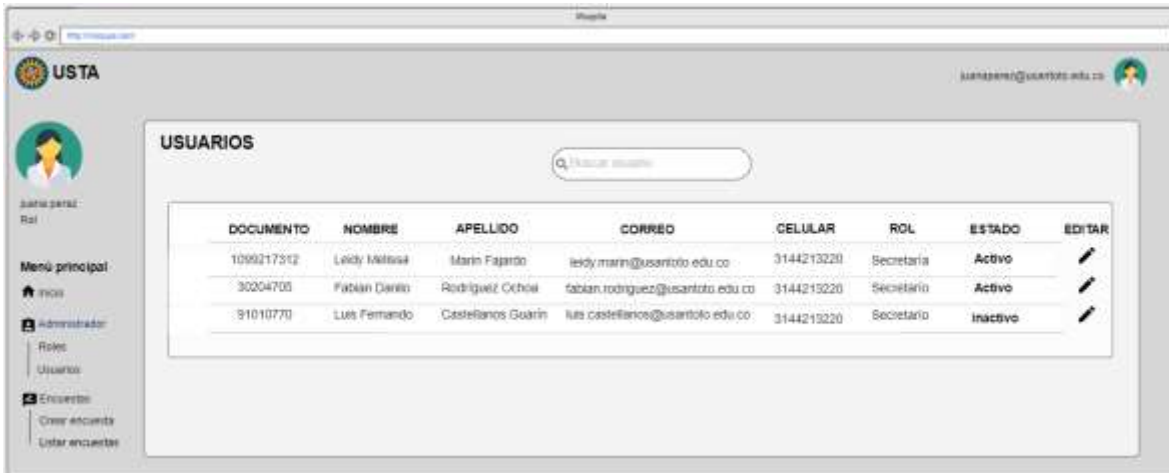


Ilustración 64 Crear usuario

The screenshot shows a web browser window with the URL <http://moqups.com>. The page header includes the USTA logo and the user's email juanaperez@usantoto.edu.co. On the left, a sidebar shows the user's profile (juana.perez, Rol) and a 'Menú principal' with options: Inicio, Administrador (Roles, Usuarios), Encuestas (Crear encuesta, Listar encuestas). The main content area is titled 'USUARIOS' and contains the following form fields:

- Nombre
- Apellidos
- Telefono
- Correo
- Estado
- Rol
- Contraseña
- Confirmar contraseña
- Tipo de documento

A blue 'Crear' button is located at the bottom right of the form.

Ilustración 65 Modificar usuario

The screenshot shows the same web browser window as above. The main content area is titled 'USUARIOS' and features three tabs: 'Datos personales', 'Datos de usuario', and 'Imagen'. The 'Datos de usuario' tab is active, showing the following form fields:

- Nombre
- Apellidos
- Telefono
- Documento
- Tipo de documento

A blue 'Modificar' button is located at the bottom right of the form.

Ilustración 66 Listar roles

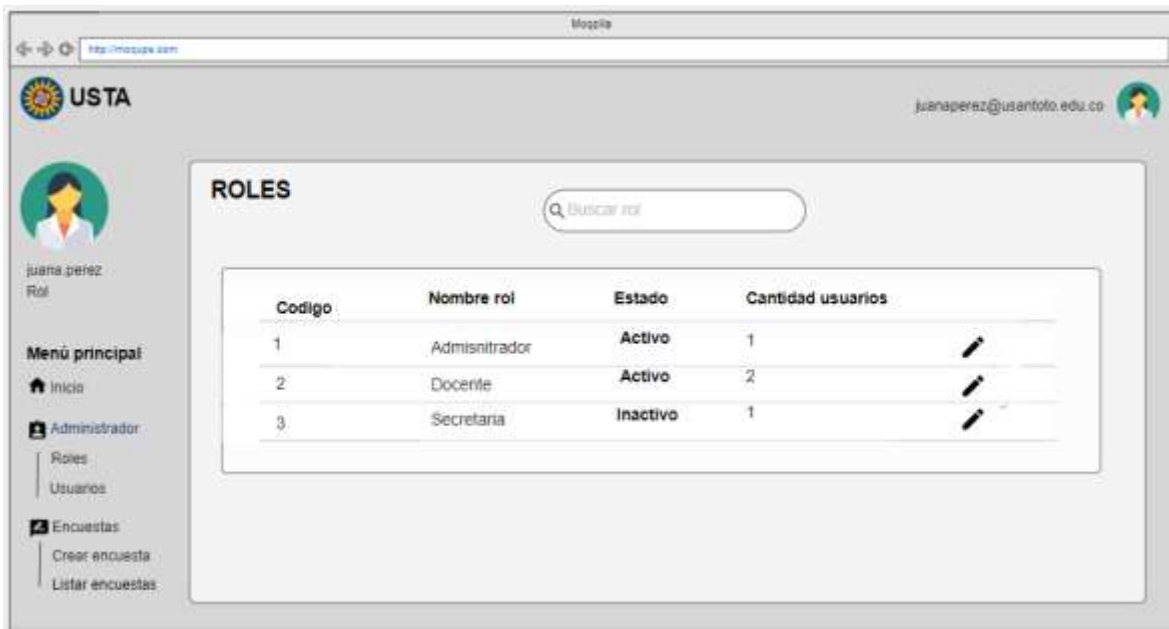


Ilustración 67 Modificar rol

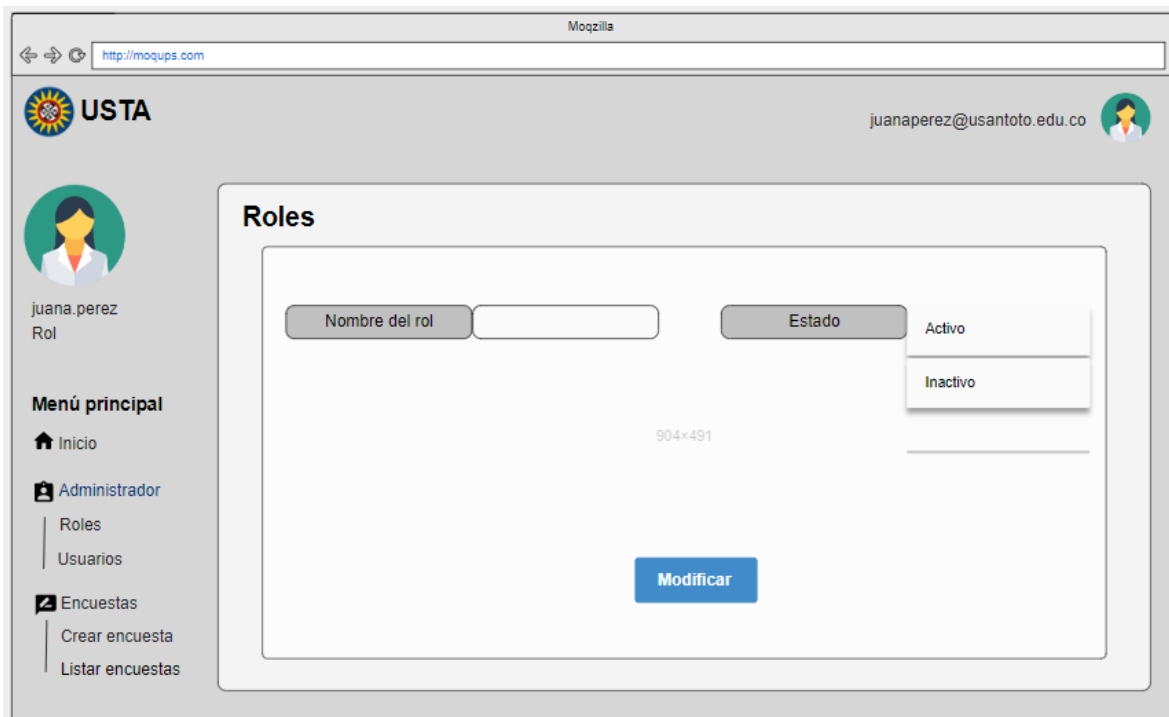


Ilustración 68 Editar perfil

Moqzila

http://moqups.com

USTA

juana.perez
Rol

Menú principal

- Inicio
- Administrador
- Encuestas

juanaperez@usantoto.edu.co

Editar Perfil

Salir

EDITAR PERFIL

Datos personales Datos de usuario Imagen

Nombre

Apellidos

Telefono

Documento

Tipo de documento

Actualizar

Ilustración 69 Listar encuestas

Moqzila

http://moqups.com

USTA

juana.perez
Rol

Menú principal

- Inicio
- Administrador
 - Roles
 - Usuarios
- Encuestas
 - Crear encuesta
 - Listar encuestas

juanaperez@usantoto.edu.co

ENCUESTAS

Buscar encuesta

Fecha	Evento	Tipo evento	Dependencia		
12 de Julio de 2022	Evento 1	Claustro	Ingenieria de sistemas	il	✎
13 de Julio de 2022	Evento 2	Conferencia	Ingeniería de civil	il	✎
14 de Julio de 2022	Evento 3	Seminario	Arquitectura	il	✎

Pág 1 de 2

Ilustración 70 Crear encuestas

The screenshot shows a web browser window displaying the USTA (Universidad Santo Tomás) interface. The page title is "FERIA DEL LIBRO". The interface includes a header with the USTA logo and name, and a user profile for Juan Pérez. A left sidebar contains a "Menú principal" with options: Inicio, Administrador, Encuestas, Crear encuesta, and Listar encuestas. The main content area features a form for creating a survey with the following fields:

- Fecha inicio: 09/12/2022
- Fecha cierre: 09/12/2022
- Tipo evento: Select
- Dependencias: Select

Below these fields is a section titled "Descripción del formulario" containing three question types:

- Pregunta selección única:** A list of radio button options: "Ejemplo" (selected), "Ejemplo", "Ejemplo", and "Añadir opción".
- Pregunta respuesta abierta:** Three horizontal text input lines.
- Pregunta de fecha:** A date input field showing "03/12/2022".

At the bottom of the form is a blue "Crear" button.

Ilustración 71 Modificar encuesta

The screenshot displays the 'Modificar encuesta' (Modify survey) interface within the USTA web application. The interface is titled 'FERIA DEL LIBRO' and includes the following elements:

- Header:** USTA logo and 'UNIVERSIDAD SANTO TOMÁS IUNJA'.
- Form Fields:**
 - Fecha inicio:** 08/12/2022
 - Fecha cierre:** 08/12/2022
 - Tipo evento:** Select
 - Dependencias:** Select
- Descripción del formulario:**
 - Pregunta selección única:** A section with a title and a red delete icon. It contains four radio button options: 'Ejemplo' (selected), 'Ejemplo', 'Ejemplo', and 'Añadir opción'. Each option has an 'X' icon to its right.
 - Pregunta respuesta abierta:** A section with a title and a red delete icon, containing three horizontal lines for text input.
 - Pregunta de fecha:** A section with a title and a red delete icon, containing a date input field with the value '03/12/2022'.
- Footer:** A blue 'Modificar' button.

The left sidebar shows the user 'juana.perez' with the role 'Rol' and a 'Menú principal' with options: 'Inicio', 'Administrador', 'Encuestas', 'Crear encuestas', and 'Listar encuestas'.

Ilustración 72 Formulario responder encuesta

The image shows a web browser window displaying a survey form. The browser's address bar shows 'http://respon.usta.edu.co'. The page header features the USTA logo and the text 'UNIVERSIDAD SANTO TOMÁS I USTA'. A sidebar on the left contains a user profile for 'Jana Perez Rol' and a 'Menú principal' with options for 'Inicio', 'Encuestas', and 'Listar encuestas'. The main content area is titled 'FERIA DEL LIBRO' and contains a 'Descripción del formulario' section. Below this are three question types: 1) 'Pregunta selección única' with three radio button options, all labeled 'Ejemplo'; 2) 'Pregunta respuesta abierta' with three horizontal text input lines; and 3) 'Pregunta de fecha' with a date picker showing '03/12/2022'. A blue 'Responder' button is located at the bottom center of the form.

Ilustración 73 Resultados de la encuesta

The screenshot displays a web application interface for survey results. The browser address bar shows 'http://mrosppa.com'. The page header includes the 'USTA' logo and the user email 'juanaperez@usantolo.edu.co'. The sidebar on the left contains a user profile for 'juana.perez' with the role 'Rol', a 'Menú principal' with 'Inicio', and 'Encuestas' with 'Listar encuestas'. The main content area is titled 'Resultados' and features a form for 'Nombre evento' with fields for 'Fecha inicio' (11/03/2022), 'Fecha cierre' (11/03/2022), 'Tipo evento' (Select), and 'Dependencia' (Select). Below the form is a text block with placeholder text. A section titled '¿Cuál es su deporte favorito?' includes a legend for 'Serie 1', 'Serie 2', and 'Serie 3' and a pie chart with segments labeled 1, 3, and 6. Below the chart is another text block with placeholder text. The final section, 'Fecha en la que asistió al evento', contains two date pickers with values '11/03/2022' and '15/08/2022'.

10.4. Anexo 4: Sentencias DDL para la construcción de la base de datos

```
CREATE TABLE usuarios(  
cod_usuario SERIAL NOT NULL,  
cod_rol int4 not NULL,  
documento_usuario varchar (50) not NULL,  
tipo_documento_usuario int2 not null default 1  
constraint ckc_tipo_documento_usu_usuarios check (tipo_documento_usuario in  
(1,2,3,4,5,6)),  
nombres_usuario varchar (60) not null,  
apellidos_usuario varchar (80) not null,  
telefono_usuario varchar (50) not null,  
estado_usuario int2 null default 1,  
cod_imagen int not null,  
cod_tipo_dependencia integer not null,  
CONSTRAINT ckc_estado_usuario_usuarios check (estado_usuario is null or  
(estado_usuario in (1,2))),  
constraint pk_usuarios primary key (cod_usuario)  
);  
  
comment on column usuarios.tipo_documento_usuario is '  
codigo: 1, acronimo: CC, tipo: Cedula de Ciudadania  
codigo: 2, acronimo: TI, tipo: Tarjeta de Identidad  
codigo: 3, acronimo: TE, tipo: Tarjeta Extranjeria  
codigo: 4, acronimo: PN, tipo: Pasaporte Nacional  
codigo: 5, acronimo: TR, tipo: Tarjeta Residente  
codigo: 6, acronimo: RG, tipo: Registro Civil';  
  
comment on column usuarios.estado_usuario is  
'1 activo  
2 inactivo';  
  
alter table usuarios owner to user_encuestas;  
create unique index indice_doc on usuarios (documento_usuario);
```

```
CREATE TABLE accesos(  
cod_usuario int4 NOT NULL,  
correo_acceso varchar (100) not null,  
clave_acceso varchar (150) not null,  
CONSTRAINT pk_accesos primary key (cod_usuario)  
);
```

```
alter table accesos owner to user_encuestas;
```

```
create unique index indice_correo on accesos (correo_acceso);
```

```
CREATE TABLE imagenes(  
cod_imagen SERIAL not null,  
nombre_publico_imagen varchar(200 ) not null,  
nombre_privado_imagen varchar (200) not null,  
tipo_imagen varchar (50) not null,  
constraint pk_imagenes primary key (cod_imagen)  
);
```

```
alter table imagenes owner to user_encuestas;
```

```
CREATE TABLE roles(  
cod_rol SERIAL NOT NULL,  
nombre_rol varchar(200) not null,  
estado_rol int2 NOT NULL  
constraint ckc_estado_rol_roles check (estado_rol in(1,2)),  
constraint pk_roles primary key (cod_rol)  
);
```

```
comment on column roles.estado_rol is  
'1 activo  
2 inactivo';
```

```
alter table roles owner to user_encuestas;
```

```
CREATE TABLE encuestas(  
cod_encuesta SERIAL NOT NULL,  
cod_tipo_dependencia int not null,  
cod_tipo_evento int not null,  
nombre_encuesta varchar(200) not null,  
fecha_creacion_encuesta DATE not null,  
fecha_cierre_encuesta DATE not null,  
descripcion_encuesta text not null,  
estado_encuesta int2 not null default 2,  
CONSTRAINT ckc_estado_encuesta_encuestas check (estado_encuesta is null or  
(estado_encuesta in (1,2))),  
constraint pk_encuestas primary key (cod_encuesta)  
);
```

```
comment on column encuestas.estado_encuesta is  
'1 Contestado  
2 Sin contestar';
```

```
alter table encuestas owner to user_encuestas;
```

```
CREATE TABLE dependencias(  
cod_dependencia SERIAL NOT NULL,  
nombre_dependencia varchar(200) not null,  
constraint pk_dependencias primary key (cod_dependencia)  
);
```

```
alter table dependencias owner to user_encuestas;
```

```
CREATE TABLE tipo_dependencias(  
cod_tipo_dependencia SERIAL NOT NULL,  
cod_dependencia int not null,
```

```
nombre_tipo_dependencia varchar(200) not null,  
constraint pk_tipo_dependencias primary key (cod_tipo_dependencia)  
);
```

```
alter table tipo_dependencias owner to user_encuestas;
```

```
CREATE TABLE tipo_eventos(  
cod_tipo_evento SERIAL NOT NULL,  
nombre_tipo_evento varchar(200) not null,  
constraint pk_tipo_eventos primary key (cod_tipo_evento)  
);
```

```
alter table tipo_eventos owner to user_encuestas;
```

```
CREATE TABLE preguntas(  
cod_pregunta SERIAL NOT NULL,  
cod_tipo_pregunta int not null,  
cod_encuesta int not null,  
descripcion_pregunta text not null,  
constraint pk_preguntas primary key (cod_pregunta)  
);
```

```
alter table preguntas owner to user_encuestas;
```

```
CREATE TABLE tipo_preguntas(  
cod_tipo_pregunta SERIAL NOT NULL,  
nombre_tipo_pregunta varchar(100) not null,  
constraint pk_tipo_preguntas primary key (cod_tipo_pregunta)  
);
```

```
alter table tipo_preguntas owner to user_encuestas;
```

```
CREATE TABLE opciones(  
cod_opcion SERIAL NOT NULL,  
cod_pregunta int,  
texto_opcion varchar(100),  
constraint pk_opciones primary key (cod_opcion)  
);
```

```
alter table opciones owner to user_encuestas;
```

```
CREATE TABLE usuarios_respuestas(  
cod_usuario int not null,  
cod_pregunta int not null,  
respuesta_abierta varchar(500),  
respuesta_fecha DATE,  
cod_opcion int,  
constraint pk_usuarios_respuestas primary key (cod_usuario,cod_pregunta)  
);
```

```
alter table usuarios_respuestas owner to user_encuestas;
```

```
CREATE TABLE usuario_encuestas(  
    cod_usuario int not null,  
    cod_encuesta int not null,  
    constraint pk_usuarios_encuestas primary key (cod_usuario,cod_encuesta)  
);
```

```
alter table usuario_encuestas owner to user_encuestas;
```

```
alter table accesos  
add constraint fk_accesos_ref_usuarios foreign key (cod_usuario)  
references usuarios (cod_usuario)  
on delete restrict on update cascade;
```

alter table usuarios

add constraint fk_usuarios_ref_rols foreign key (cod_rol)

references roles (cod_rol),

add constraint fk_usuarios_ref_imagenes foreign key (cod_imagen)

references imagenes (cod_imagen),

add constraint fk_usuarios_ref_tipo_dependencias foreign key (cod_tipo_dependencia)

references tipo_dependencias (cod_tipo_dependencia)

on delete restrict on update cascade;

alter table usuario_encuestas

add constraint fk_usuario_encuestas_ref_usuarios foreign key (cod_usuario)

references usuarios (cod_usuario),

add constraint fk_usuario_encuestas_ref_encuestas foreign key (cod_encuesta)

references encuestas (cod_encuesta)

on delete restrict on update cascade;

alter table encuestas

add constraint fk_encuestas_ref_tipo_dependencias foreign key (cod_tipo_dependencia)

references tipo_dependencias (cod_tipo_dependencia),

add constraint fk_encuestas_ref_tipo_eventos foreign key (cod_tipo_evento)

references tipo_eventos (cod_tipo_evento)

on delete restrict on update cascade;

alter table tipo_dependencias

add constraint fk_tipo_dependencias_ref_dependencias foreign key (cod_dependencia)

references dependencias (cod_dependencia)

on delete restrict on update cascade;

alter table preguntas

add constraint fk_preguntas_ref_tipo_pregunta foreign key (cod_tipo_pregunta)

references tipo_preguntas (cod_tipo_pregunta),

add constraint fk_preguntas_ref_encuestas foreign key (cod_encuesta)

references encuestas (cod_encuesta)

on delete restrict on update cascade;

alter table opciones

add constraint fk_opciones_ref_preguntas foreign key (cod_pregunta)
references preguntas (cod_pregunta)
on delete restrict on update cascade;

alter table usuarios_respuestas

add constraint fk_respuestas_ref_usuarios foreign key (cod_usuario)
references usuarios (cod_usuario),
add constraint fk_respuestas_ref_preguntas foreign key (cod_pregunta)
references preguntas (cod_pregunta),
add constraint fk_respuestas_ref_opciones foreign key (cod_opcion)
references opciones (cod_opcion)
on delete restrict on update cascade;

10.5. Anexo 5: Acceso a repositorio

Para que la facultad pueda tener acceso al repositorio donde se encuentra el código fuente tanto del backend y del frontend del proyecto “**Desarrollo web para la administración de las encuestas realizadas en los eventos de la USTA Tunja**”. Esta deberá suministrar una dirección de una cuenta GitHub y así poderla agregar como colaborador, teniendo en cuenta que el repositorio de este proyecto es privado, de esta manera podrá descargar o visualizar el repositorio con su contenido.

<https://github.com/fabianr8a/Sistema-Encuestas/tree/Origin/Backend>

<https://github.com/fabianr8a/Sistema-Encuestas/tree/Origin/Frontend>