

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA WEB PROTOTIPO PARA GENERAR Y GESTIONAR DOCUMENTOS NECESARIOS EN EL PROCESO DE CREACIÓN DE CSIRT ACADÉMICOS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

ERICK PAÚL RÍOS FIALLOS

DIRECTOR: MSc. FRANKLIN LEONEL SÁNCHEZ CATOTA

Quito, mayo 2022

AVAL

Certifico que el presente trabajo fue desarrollado por Erick Paúl Ríos Fiallos, bajo mi supervisión.

MSc. FRANKLIN LEONEL SÁNCHEZ CATOTA
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Erick Paúl Ríos Fiallos, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

ERICK PAUL RÍOS FIALLOS

DEDICATORIA

A mi familia.

AGRADECIMIENTO

A mis padres.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XVI
RESUMEN	XVII
ABSTRACT	XVIII
1 INTRODUCCIÓN	1
1.1 OBJETIVOS	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	4
1.3.1 CSIRT	5
1.3.2 ENTIDADES QUE PROMUEVEN LA CREACIÓN DE CSIRT	6
1.3.2.1 FIRST.....	6
1.3.2.2 CARNEGIE MELLON SEI	6
1.3.2.3 ENISA	6
1.3.2.4 TF-CSIRT.....	6
1.3.2.5 OPEN-CSIRT	6
1.3.3 STACK MEAN	7
1.3.3.1 MONGODB.....	7
1.3.3.2 EXPRESS	7
1.3.3.3 ANGULAR.....	7
1.3.3.4 NODE.....	8
1.3.3.5 GRAPHQL.....	8
1.3.4 BASES DE DATOS NO RELACIONALES.....	7
1.3.5 SISTEMAS DISTRIBUIDOS	8
1.3.6 SERVICIOS WEB	8
1.3.7 BOOTSTRAP.....	9
1.3.8 JAVASCRIPT.....	9
1.3.9 TYPESCRIPT	9
1.3.10 SCRUM	9

1.3.11	HERRAMIENTAS TECNOLÓGICAS.....	10
2	METODOLOGÍA.....	12
2.1	DISEÑO.....	12
2.1.1	FIRST.....	12
2.1.2	CARNEGIE MELLON SEI.....	13
2.1.3	ENISA.....	14
2.1.4	TF-CSIRT.....	14
2.1.5	OPEN-CSIRT.....	15
2.1.6	REUNIÓN CON CSIRT-EPN.....	16
2.1.7	DOCUMENTOS DEL SISTEMA WEB.....	16
2.1.7.1	PILAR ORGANIZACIÓN.....	16
2.1.7.2	PILAR TALENTO HUMANO.....	17
2.1.7.3	PILAR POLÍTICAS.....	17
2.1.7.4	PILAR INFRAESTRUCTURA.....	19
2.1.8	REQUERIMIENTOS.....	20
2.1.8.1	REQUERIMIENTOS FUNCIONALES.....	20
2.1.8.2	REQUERIMIENTOS NO FUNCIONALES.....	21
2.1.9	EQUIPO DE TRABAJO.....	21
2.1.10	HISTORIAS DE USUARIO.....	21
2.1.11	BACKLOG PRIORIZADO DEL PRODUCTO.....	23
2.1.12	SPRINT BACKLOG.....	24
2.1.13	DIAGRAMA DE CASOS DE USOS.....	25
2.1.14	DIAGRAMA DE CLASES.....	29
2.1.15	MODELADO BASE DE DATOS.....	29
2.1.16	DISEÑO WEB.....	31
2.1.16.1	WIREFRAMES.....	31
2.1.16.2	MOCKUPS.....	38
2.2	IMPLEMENTACIÓN.....	46
2.2.1	SPRINT 0: CREACIÓN DEL AMBIENTE DE DESARROLLO.....	46
2.2.1.1	BASE DE DATOS EN MONGODB.....	46
2.2.1.2	REPOSITORIOS DE GITHUB.....	47
2.2.1.3	RECURSOS NECESARIOS.....	48
2.2.1.4	REPOSITORIO FRONTEND LOCAL.....	49
2.2.1.5	REPOSITORIO BACKEND LOCAL.....	50

2.2.2	SPRINT 1: AUTENTICACIÓN	55
2.2.2.1	CODIFICACIÓN BACKEND.....	55
2.2.2.2	CODIFICACIÓN FRONTEND	61
2.2.3	SPRINT 2: GESTIÓN GESTOR	63
2.2.3.1	CODIFICACIÓN BACKEND.....	64
2.2.3.2	CODIFICACIÓN FRONTEND	66
2.2.4	SPRINT 3: GESTIÓN RESPONSABLE.....	67
2.2.4.1	CODIFICACIÓN BACKEND.....	67
2.2.4.2	CODIFICACIÓN FRONTEND	69
2.2.5	SPRINT 4: GESTIÓN COLABORADOR.....	69
2.2.5.1	CODIFICACIÓN BACKEND.....	70
2.2.5.2	CODIFICACIÓN FRONTEND	72
2.2.6	SPRINT 5: GESTIÓN SUPERVISOR	73
2.2.6.1	CODIFICACIÓN BACKEND.....	73
2.2.6.2	CODIFICACIÓN FRONTEND	75
2.2.7	SPRINT 6: GENERACIÓN DE DOCUMENTOS FINALES	76
2.2.7.1	CODIFICACIÓN FRONTEND	76
2.2.8	SPRINT 7: GESTIÓN PANEL HOME	78
2.2.8.1	CODIFICACIÓN FRONTEND	78
3	RESULTADOS Y DISCUSIÓN	81
3.1	PRUEBAS SPRINT 1: AUTENTICACIÓN.....	81
3.1.1	INICIO DE SESIÓN	81
3.1.2	REGISTRO DE UN USUARIO.....	82
3.1.3	CAMBIO DE CONTRASEÑA.....	84
3.2	PRUEBAS SPRINT 2: GESTIÓN GESTOR.....	87
3.2.1	PANEL PRINCIPAL	88
3.2.2	PANEL UNIVERSIDADES.....	88
3.2.2.1	CREAR UNIVERSIDADES	89
3.2.2.2	EDITAR UNIVERSIDADES.....	91
3.2.2.3	VER INFORMACIÓN DE UNA UNIVERSIDAD	92
3.2.2.4	BLOQUEAR UNIVERSIDADES	92
3.2.2.5	ELIMINAR UNIVERSIDADES	93
3.2.3	PANEL USUARIOS	93
3.2.3.1	CREAR USUARIOS.....	94

3.2.3.2	EDITAR USUARIOS	96
3.2.3.3	VER INFORMACIÓN DE USUARIOS.....	98
3.2.3.4	BLOQUEAR/DESBLOQUEAR USUARIOS	98
3.2.3.5	ELIMINAR USUARIOS.....	99
3.3	PRUEBAS SPRINT 3: GESTIÓN RESPONSABLE	100
3.3.1	PANEL PRINCIPAL.....	100
3.3.2	PANEL USUARIOS	101
3.3.2.1	CREAR USUARIOS.....	101
3.3.2.2	EDITAR USUARIOS	103
3.3.2.3	VER INFORMACIÓN DE USUARIOS.....	105
3.3.2.4	BLOQUEAR/DESBLOQUEAR USUARIOS	105
3.3.2.5	ELIMINAR USUARIOS.....	106
3.3.3	PANEL DOCUMENTOS	107
3.3.3.1	ASIGNAR DOCUMENTOS	107
3.3.3.2	VER INFORMACIÓN DE DOCUMENTOS.....	109
3.4	PRUEBAS SPRINT 4: GESTIÓN COLABORADORES	109
3.4.1	EDITAR CONTENIDO DE UN DOCUMENTO.....	110
3.4.2	PANEL CAMBIOS DEL DOCUMENTO	112
3.4.3	CAMBIAR ESTADO DOCUMENTO	112
3.5	PRUEBAS SPRINT 5: GESTIÓN SUPERVISORES.....	113
3.5.1	EDITAR CONTENIDO DE UN DOCUMENTO.....	114
3.5.2	PANEL CAMBIOS	115
3.5.2.1	CREAR CAMBIOS	115
3.5.2.2	LISTA DE CAMBIOS.....	116
3.5.2.3	EDITAR CAMBIOS	116
3.5.2.4	APROBAR CAMBIOS	117
3.5.2.5	ELIMINAR CAMBIOS.....	117
3.6	PRUEBAS SPRINT 6: GENERACIÓN DE DOCUMENTOS FINALES .	118
3.7	PRUEBAS SPRINT 7: GESTIÓN PANEL HOME	121
3.7.1	PANEL HOME	121
3.7.2	VISUALIZACIÓN CALENDARIO	122
3.8	FORMULACIÓN DE LAS PREGUNTAS DE LAS ENCUESTAS	122
3.9	RESULTADOS DE LAS ENCUESTAS	125
3.9.1	RESULTADOS DE FUNCIONALIDAD POR ROL DE USUARIO...	125

3.9.1.1	RESULTADOS USUARIO CON ROL GESTOR	125
3.9.1.2	RESULTADOS USUARIO CON ROL RESPONSABLE.....	126
3.9.1.3	RESULTADOS USUARIO CON ROL SUPERVISOR.....	126
3.9.1.4	RESULTADOS USUARIO CON ROL COLABORADOR	127
3.9.2	RESULTADOS DE LA INTERFAZ GRÁFICA.....	127
3.9.3	RESULTADOS FUNCIONAMIENTO GENERAL DEL SISTEMA ...	129
3.9.4	RESULTADOS INCONVENIENTES.....	131
3.10	RECTIFICACIONES EN EL SISTEMA WEB	132
4	CONCLUSIONES Y RECOMENDACIONES.....	133
4.1	CONCLUSIONES.....	133
4.2	RECOMENDACIONES	133
5	REFERENCIAS BIBLIOGRÁFICAS	135
6	ANEXOS.....	137

ÍNDICE DE FIGURAS

Figura 1.1. Diagrama del prototipo	4
Figura 2.1. Historia de usuario HU002	22
Figura 2.2. Diagrama de casos de uso Módulo Autenticación.....	26
Figura 2.3. Diagrama de casos de uso Módulo Gestor	26
Figura 2.4. Diagrama de casos de uso Módulo Responsable	27
Figura 2.5. Diagrama de casos de uso Panel Home	27
Figura 2.6. Diagrama de casos de uso Módulo Colaborador.....	28
Figura 2.7. Diagrama de casos de uso Módulo Supervisor	28
Figura 2.8. Diagrama de clases del sistema web	29
Figura 2.9. Diagrama de base de datos MongoDB.....	30
Figura 2.10. Wireframes del módulo de Autenticación. (a) Inicio de sesión, (b) Registro de usuario, (c) Recuperación de contraseña, (d) Cambio de contraseña	32
Figura 2.11. Wireframes del Módulo Gestor. (a) Vista principal, (b) Universidades, (c) Usuarios	34
Figura 2.12. Wireframes del Módulo Responsable. (a) Vista principal, (b) Documentos, (c) Asignación de documento, (d) Usuarios	35
Figura 2.13. Wireframe del Módulo Colaborador.....	36
Figura 2.14. Wireframe del Panel Usuario Supervisor.....	36
Figura 2.15. Wireframe del Módulo Estado de Tareas	37
Figura 2.16. Wireframe del Módulo Agenda de Actividades.....	37
Figura 2.17. Wireframe del Módulo Formularios Documentos.....	38
Figura 2.18. Mockups del Módulo Autenticación. (a) Inicio de sesión, (b) Registro de usuario, (c) Recuperación de contraseña, (d) Cambio de contraseña	40
Figura 2.19. Mockups del Módulo Gestor. (a) Vista principal, (b) Universidades, (c) Usuarios	41
Figura 2.20. Mockups del Módulo Gestor. (a) Vista principal, (b) Universidades, (c) Usuarios	43
Figura 2.21. Mockup del Módulo Colaborador.....	44
Figura 2.22. Mockup del Módulo Supervisor	44
Figura 2.23. Mockup del Módulo Estado de Tareas	45
Figura 2.24. Mockup del Módulo Agenda de Actividades.....	45
Figura 2.25. Creación base de datos en MongoDB Compass.....	46
Figura 2.26. Base de datos del proyecto creada	47
Figura 2.27. Añadir una universidad nueva en la base de datos	47

Figura 2.28. Repositorios en Github. (a) Repositorio frontend, (b) Repositorio backend.....	48
Figura 2.29. Espacio de trabajo de Angular	49
Figura 2.30. Añadir librería de Bootstrap.....	50
Figura 2.31. Complementos de JS para Bootstrap.....	50
Figura 2.32. Fichero package.json	51
Figura 2.33. Versiones mínimas de Node y npm.....	51
Figura 2.34. Definición de scripts	52
Figura 2.35. Variables de entorno	52
Figura 2.36. Clase que inicializa la base de datos.....	54
Figura 2.37. Función para montar el servidor.....	54
Figura 2.38. Estructura esquema GraphQL.....	55
Figura 2.39. Funciones de operaciones CRUD. (a) Función findOneElement, (b) Función insertOneElement, (c) Función updateOneElement, (d) Función deleteOneElement.....	56
Figura 2.40. Función para asignar Id.....	57
Figura 2.41. Funciones para JWT. (a) Función para firmar un token, (b) Función para verificar un token.....	58
Figura 2.42. Queries login y me.....	58
Figura 2.43. Resolvers login y me	59
Figura 2.44. Mutation register.....	59
Figura 2.45. Resolver register	59
Figura 2.46. Mutations resetPassword y changePassword	60
Figura 2.47. Resolvers resetPassword y changePassword.....	61
Figura 2.48. Configuración Módulo GraphQL	62
Figura 2.49. Servicio login	62
Figura 2.50. Servicio register.....	63
Figura 2.51. Servicios para cambiar contraseña	63
Figura 2.52. Queries de universidades.....	64
Figura 2.53. Mutations de universidades.....	64
Figura 2.54. Resolvers de queries de universidades.....	65
Figura 2.55. Resolvers de mutations de universidades	65
Figura 2.56. Código archivo dashboard-info.component.html	66
Figura 2.57. Código servicios para universidades	66
Figura 2.58. Código queries de documentos.....	67

Figura 2.59. Código mutation de documentos	68
Figura 2.60. Resolvers de queries para documentos	68
Figura 2.61. Resolver de mutation para documentos	69
Figura 2.62. Código de dashboard.componente.html	69
Figura 2.63. Código queries de los cambios.....	70
Figura 2.64. Código mutations de los cambios.....	71
Figura 2.65. Código resolvers-queries de los cambios.....	71
Figura 2.66. Código resolvers-mutations de los cambios	71
Figura 2.67. Archivos del componente text-editor	72
Figura 2.68. Código de text-editor.component.html.....	72
Figura 2.69. Código queries de items.....	74
Figura 2.70. Código mutations de items	74
Figura 2.71. Resolvers-queries de ítems	75
Figura 2.72. Resolvers-mutations de ítems	75
Figura 2.73. Código archivo comments.component.html.....	76
Figura 2.74. Código archivo document-one-final.component.ts.....	77
Figura 2.75. Función PrintElem().....	77
Figura 2.76. Estructura módulo Home.....	78
Figura 2.77. Código archivo home.component.html	79
Figura 2.78. Configuración librería fullcalendar	79
Figura 3.1. Campos no válidos en pantalla de inicio	81
Figura 3.2. Ingreso de correo y contraseña en formulario	82
Figura 3.3. Mensajes de acceso al sistema.....	82
Figura 3.4. Mensajes de validación de campos.....	83
Figura 3.5. Ingreso de datos requeridos para el registro	83
Figura 3.6. Verificación del usuario en la base de datos	84
Figura 3.7. Pantalla para restablecer contraseña.....	84
Figura 3.8. Validación de correo electrónico.....	84
Figura 3.9. Mensaje de envío de restablecimiento de contraseña.....	85
Figura 3.10. Email recibido en el servicio de correo electrónico.....	85
Figura 3.11. Formulario para cambiar la contraseña	86
Figura 3.12. Campos inválidos para cambio de contraseña.....	86
Figura 3.14. Mensaje de cambio de contraseña exitoso	87
Figura 3.15. Panel principal del usuario gestor.....	88

Figura 3.16. Barra lateral oculta en panel principal	88
Figura 3.17. Panel gestor-crud-universidades	89
Figura 3.18. Cuadro de diálogo para registro de universidad	89
Figura 3.19. Mensajes de error cuadro de diálogo	90
Figura 3.20. Campos válidos para el registro de la universidad	90
Figura 3.21. Universidad registrada en la base de datos	90
Figura 3.22. Cuadro de diálogo para modificar la universidad.....	91
Figura 3.23. Cuadro de diálogo con la universidad modificada	91
Figura 3.24. Universidad actualizada en la base de datos	91
Figura 3.25. Información de la universidad seleccionada	92
Figura 3.26. Cuadro de diálogo para desactivar una universidad.....	92
Figura 3.27. Universidad bloqueada en la base de datos.....	92
Figura 3.28. Cuadro de diálogo para eliminar una universidad	93
Figura 3.29. Universidad eliminada de la base de datos	93
Figura 3.30. Panel gestor-crud-usuarios	94
Figura 3.31. Formulario para registro de un usuario.....	94
Figura 3.32. Mensajes de error formulario crear usuario	95
Figura 3.33. Campos válidos para el registro de la universidad	95
Figura 3.34. Usuario registrado en la base de datos	96
Figura 3.35. Formulario para activar el usuario	96
Figura 3.36. Formulario para modificar un usuario	96
Figura 3.37. Formulario con el usuario modificado.....	97
Figura 3.38. Usuario actualizado en la base de datos.....	97
Figura 3.39. Alerta al tratar de editar un usuario	97
Figura 3.40. Información del usuario seleccionado	98
Figura 3.41. Cuadro de diálogo para desactivar un usuario	98
Figura 3.42. Usuario bloqueado en la base de datos	99
Figura 3.43. Cuadro de diálogo para eliminar un usuario	99
Figura 3.44. Usuario eliminado de la base de datos.....	99
Figura 3.45. Panel principal del usuario responsable	100
Figura 3.46. Barra lateral oculta en panel principal	100
Figura 3.47. Panel responsable-crud-usuarios	101
Figura 3.48. Formulario para registro de un usuario.....	102
Figura 3.49. Mensajes de error formulario crear usuario	102

Figura 3.50. Formulario crear usuario campos correctos	103
Figura 3.51. Enlace de activación enviado al nuevo usuario	103
Figura 3.52. Usuario registrado en la base de datos	103
Figura 3.53. Formulario para modificar un usuario	104
Figura 3.54. Formulario con el usuario modificado	104
Figura 3.55. Usuario actualizado en la base de datos	105
Figura 3.56. Información del usuario seleccionado	105
Figura 3.57. Cuadro de diálogo para bloquear un usuario	105
Figura 3.58. Usuario bloqueado en la base de datos	106
Figura 3.59. Cuadro de diálogo para eliminar un usuario	106
Figura 3.60. Usuario eliminado de la base de datos.....	106
Figura 3.61. Panel responsable-crud-universidades	107
Figura 3.62. Formulario asignación documentos	107
Figura 3.63. Formulario asignación documentos lleno	108
Figura 3.64. Alerta para asignar el documento	108
Figura 3.65. Documento actualizado en la base de datos	108
Figura 3.66. Información del documento seleccionado	109
Figura 3.67. Formulario edición-documento-colaborador	109
Figura 3.68. Edición ítem de un documento	110
Figura 3.69. Alerta crear versión	110
Figura 3.70. Nueva versión creada de un ítem	110
Figura 3.71. Nueva versión en la base de datos	111
Figura 3.72. Establecer la versión actual del ítem	111
Figura 3.73. Versiones del primer ítem del documento en BD	111
Figura 3.74. Cambios pendientes.....	112
Figura 3.75. Cambios realizados	112
Figura 3.76. Alerta enviar documento a Módulo Supervisor	113
Figura 3.77. Documento actualizado en la base de datos	113
Figura 3.78. Formulario edición-documento-supervisor	114
Figura 3.79. Edición del ítem seleccionado	114
Figura 3.80. Versión del ítem actualizado en la base de datos	115
Figura 3.81. Panel cambios-documento-supervisor	115
Figura 3.82. Formulario para crear nuevo cambio	115
Figura 3.83. Lista de cambios.....	116

Figura 3.84. Edición de un cambio	116
Figura 3.85. Cambio actualizado en la base de datos.....	117
Figura 3.86. Aprobar un cambio	117
Figura 3.87. Cambio finalizado en la base de datos.....	117
Figura 3.88. Eliminar un cambio	118
Figura 3.89. Cambio eliminado de la base de datos.....	118
Figura 3.90. Panel generación de documentos finales.....	119
Figura 3.91. Ventana emergente para descargar documento	119
Figura 3.92. Ubicación del documento PDF a descargar.	120
Figura 3.93. Contenido PDF descargado	120
Figura 3.94. Panel Home.....	121
Figura 3.95. Alertas al acceder a un documento particular	121
Figura 3.96. Calendario con las fechas de entrega de los documentos	122
Figura 3.97. Información del documento en el calendario	122
Figura 3.98. Resultados preguntas para usuario con rol Gestor	125
Figura 3.99. Resultados preguntas para usuario con rol Responsable	126
Figura 3.100. Resultados preguntas para usuario con rol Supervisor	126
Figura 3.101. Resultados preguntas para usuario con rol Colaborador	127
Figura 3.102. Resultados de las preguntas sobre la interfaz gráfica	128
Figura 3.103. Resultados calificación de la interfaz gráfica.....	128
Figura 3.104. Resultados funcionamiento del sistema roles responsable- colaborador-supervisor.....	129
Figura 3.105. Resultados funcionamiento del sistema para todos los roles	130
Figura 3.106. Resultados funcionamiento del sistema para todos los roles	130
Figura 3.107. Resultados calificación del funcionamiento general del sistema web	131
Figura 3.108. Resultados inconvenientes presentados	131

ÍNDICE DE TABLAS

Tabla 1.1. Fases de SCRUM	10
Tabla 2.1. Pilares fundamentales y estándares recomendados por FIRST	13
Tabla 2.2. Pilares fundamentales y estándares recomendados por CMU SEI	13
Tabla 2.3. Pilares fundamentales y estándares recomendados por ENISA.....	14
Tabla 2.4. Pilares fundamentales y estándares recomendados por TF-CSIRT ...	15
Tabla 2.5. Pilares fundamentales y estándares recomendados por OPEN-CSIRT	15
Tabla 2.6. Documentos del pilar Organización	16
Tabla 2.7. Documentos del pilar Talento Humano	17
Tabla 2.8. Ítems de cada una de las políticas establecidas	18
Tabla 2.9. Ítems de cada uno de los procedimientos establecidos	19
Tabla 2.10. Requisitos establecidos por el pilar Infraestructura.....	20
Tabla 2.11. Roles SCRUM.....	21
Tabla 2.12. Backlog priorizado del producto	23
Tabla 2.13. Sprint Backlog	24
Tabla 2.14. Recursos necesarios en el proyecto	49
Tabla 2.15. Reglas TSLINT.....	53
Tabla 3.1. Establecimiento de usuarios encuestados por roles	81
Tabla 3.2. Preguntas formuladas para las encuestas	123
Tabla 3.3. Errores y correcciones del sistema web.....	132

RESUMEN

Este trabajo de titulación desarrolló un sistema web prototipo que permite generar y gestionar documentos que son imprescindibles en el proceso de creación de CSIRT académicos.

Para el desarrollo del sistema web se utilizó el stack MEAN+G (MongoDB, Express, Angular, Node.JS, GraphQL) y se basó en la metodología de desarrollo ágil SCRUM.

De esta manera, el primer capítulo abordó temas concernientes a las diferentes tecnologías y herramientas usadas en el prototipo. Así como también, nociones básicas sobre el marco de trabajo SCRUM.

En el segundo capítulo, se realizó el diseño e implementación del sistema web. Por un lado, en el diseño se llevó a cabo una comparación de diferentes estándares que promueven el establecimiento de CSIRT, también se desarrollaron las historias de usuario, se realizaron los diagramas de clases y casos de uso, se bosquejaron los wireframes y se elaboraron los mockups relacionados con el prototipo web. Por otra parte, en la etapa de implementación, se generó la base de datos en MongoDB y se codificaron todos los módulos propuestos del sistema.

En el tercer capítulo, se realizaron pruebas de funcionamiento para validar el cumplimiento de requerimientos del sistema con un usuario gestor y una universidad, misma que tiene un usuario responsable, 2 usuarios colaboradores y 2 usuarios supervisores. Además, se muestran los resultados de las encuestas realizadas a las personas que probaron el sistema web y se corrigieron errores detectados en los diferentes módulos.

Finalmente, en el cuarto capítulo se establecieron las conclusiones y recomendaciones conseguidas en el transcurso de la realización de este trabajo de titulación.

PALABRAS CLAVE: CSIRT, MONGODB, EXPRESS, ANGULAR, NODE.JS, GRAHPQL

ABSTRACT

This thesis developed a prototype web system that allows the generation and management of documents that are essential in the process of creating academic CSIRT.

For the development of this web system, the MEAN + G stack (MongoDB, Express, Angular, Node.JS, GrahpQL) was used and it was based on the SCRUM agile development methodology.

In the first chapter, the issues concerning the different technologies and tools used in the prototype and basic topics about the SCRUM framework were explained.

In the second chapter, the design and implementation of the web system were carried out. On the one hand, in the design, a comparison of different standards that promote the establishment of CSIRT were made. In addition, user stories, class diagrams, and use cases diagrams were made. Then, wireframes were drawn, and mockups related to the web prototype were sketched. On the other hand, in the implementation phase, the database was generated in MongoDB and all the proposed modules of the system were coded.

In the third chapter, performance tests were carried out to validate compliance with system requirements with a manager user and a university, which has a user with responsible role, 2 users with collaborator role and 2 users with supervisor role. In addition, the results of the surveys carried out by the people who tested the web system are shown and errors detected in the different modules were fixed.

Finally, in the fourth chapter, the conclusions and recommendations obtained while carrying out this degree work were established.

KEYWORDS: CSIRT, MONGODB, EXPRESS, ANGULAR, NODE.JS, GRAHPQL

1 INTRODUCCIÓN

El avance acelerado de la tecnología ha generado grandes beneficios económicos a las organizaciones ya que se facilita la digitalización de la información, pero a la vez ha ocasionado enormes problemas de seguridad, protección de datos y privacidad debido a los millones de ataques informáticos que se producen cada año a nivel mundial. [1]

Microsoft Security Intelligence¹ reportó que el 61.75% de los 5 713 930 de dispositivos infectados con malware² en el mes de septiembre de 2020 provinieron del sector educativo [2] y según [3], en el 2020 las violaciones de datos a dicho sector costaron alrededor de 3.9 millones de dólares. Además, se estableció que las empresas que carecían de mecanismos de gestión de incidentes de seguridad experimentaron un mayor costo de reparación. Este antecedente ha promovido que las instituciones de educación superior realicen inversiones en temas de ciberseguridad y en la creación de sus propios Equipos de Respuesta a Incidentes de Seguridad Informática (CSIRT) [4].

Los CSIRT son importantes, ya que ayudan a las organizaciones a reducir y prevenir incidentes ocasionados por ciberataques mediante la compartición de recursos con otros CSIRT y un control centralizado de los problemas informáticos de seguridad. De esta manera, se protege información sensible y en el caso de que un incidente afecte a una institución, el daño sea mínimo y se pueda recuperar en el menor tiempo posible [5].

Sin embargo, a pesar de su importancia en el sector educativo, en el Ecuador solo existen 3 CSIRT académicos, y a nivel de universidades, la Escuela Politécnica Nacional (EPN) es la única que cuenta con un CSIRT certificado por el Forum of Incident Response and Security Teams (FIRST). El problema de la no implementación de CSIRT académicos surge debido a que el establecimiento de un CSIRT requiere documentación específica para alcanzar su certificación, y actualmente, la mayor parte de universidades en el país no cuentan con guías claras o herramientas que les permitan generar y llevar un seguimiento de dicha documentación para el proceso de creación de un CSIRT.

El presente trabajo de titulación propone desarrollar un sistema web prototipo que permita generar y gestionar los documentos necesarios para la creación de CSIRT académicos. En el caso de no desarrollar este sistema web prototipo, las universidades interesadas en establecer su propio CSIRT y tener un seguimiento de la documentación tendrán que

¹ Microsoft Security Intelligence: Es una red de global de expertos en seguridad de Microsoft que realizan investigaciones relacionadas con la seguridad y el manejo de amenazas cibernéticas.

² Malware: Es un programa o software malicioso que afecta el correcto funcionamiento de un sistema informático.

realizar sus procesos de forma manual, lo cual, dificultaría alcanzar dicho objetivo en el menor tiempo posible.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es:

- Desarrollar un sistema web prototipo para la generación y gestión de documentos necesarios en el proceso de creación de CSIRT académicos

Los objetivos específicos del Proyecto Técnico son:

- Analizar los fundamentos teóricos necesarios para el desarrollo del proyecto.
- Diseñar los módulos que conforman el prototipo.
- Codificar los módulos del sistema web diseñados.
- Analizar los resultados de las pruebas de funcionamiento realizadas.

1.2 ALCANCE

En el presente proyecto se propone el desarrollo de un sistema web prototipo que permita la generación y gestión de documentos necesarios para la creación de un CSIRT académico.

El sistema web prototipo consta de los siguientes módulos:

- **Módulo de Autenticación:** Este módulo permite autenticar a los diferentes usuarios del sistema mediante JWT³ y se puede enviar notificaciones de correo electrónico en el caso de cambio de contraseña.
- **Módulo Gestor:** Este módulo facilita la creación, actualización y eliminación de universidades en el sistema de gestión, con su respectivo responsable.
- **Módulo Responsables:** Este módulo permite la gestión de los equipos de trabajo para la generación de documentación, mediante la asignación de documentos a los diferentes usuarios y definición de fecha de entrega. Además, permitirá crear, actualizar y eliminar usuarios con roles de colaboradores y supervisores pertenecientes a una misma universidad.

³ JWT (JSON Web Token): Es un estándar abierto basado en JSON para la generación de tokens de acceso.

- **Módulo Colaboradores:** Este módulo facilita a los colaboradores editar los documentos que les han sido asignados, además de reportar cuando alguna de las secciones haya sido concluida.
- **Módulo Supervisores:** Este módulo permite a los supervisores verificar el avance de los documentos. Además de visualizar cuando una sección haya sido reportada como concluida, revisarla, y devolverla a los colaboradores en el caso de que hagan falta correcciones, incluyendo comentarios.
- **Módulo Formularios Documentos:** Este módulo permite visualizar de manera gráfica el contenido de cada documento que se generará en el sistema, además de contar con información de los usuarios que están a cargo de dichos documentos.
- **Módulo de Generación de Documentos Finales:** En este módulo se puede imprimir un archivo PDF⁴ por cada documento finalizado y aprobado por los usuarios supervisores.
- **Módulo de Estado de Tareas:** Este módulo permite la visualización centralizada del estado de avance de cada uno de los documentos.
- **Módulo Agenda Actividades:** En este módulo se muestra un calendario con las fechas de entrega de las actividades creadas por el usuario responsable de la organización.

El sistema web cuenta con los siguientes roles de usuarios:

- **Gestor:** Usuario encargado de agregar, modificar o eliminar universidades, en el sistema de gestión, así como su respectivo responsable.
- **Responsable:** Usuario encargado de asignar documentos a colaboradores y supervisores, y definir fechas de entrega. Además, podrá crear, modificar y eliminar usuarios para su correspondiente equipo de trabajo.
- **Supervisores:** Usuarios encargados de la revisión de cada documento asignado por el responsable de la organización. Además, podrán devolver un documento en el caso de que se requieran correcciones y generar un documento final en formato PDF cuando éste se encuentre listo.

⁴ PDF (Formato de Documento Portátil): Es un formato de archivos desarrollado por Adobe para intercambiar documentos digitales.

- **Colaboradores:** Usuarios encargados de la edición de cada documento asignado por el responsable de la organización. Además, podrán realizar las respectivas correcciones establecidas por los supervisores.

El sistema prototipo no consta de una interfaz de administración, sino que, en caso de requerir un CRUD general del sistema se lo realizará directamente en la base de datos.

El código del proyecto se lo alojó en la plataforma de desarrollo colaborativo GitHub y se utilizó el patrón de arquitectura de software Modelo-Vista-Controlador (MVC) y se basó en la metodología de desarrollo ágil SCRUM. Además, consta de un servidor de aplicaciones web Node.JS implementado con el framework Express y una conexión hacia una base de datos implementada con MongoDB, que almacena la información relacionada al sistema. En el lado del cliente se tiene la interfaz de usuario elaborada con Angular 9, y las peticiones al servidor se las realizan mediante GraphQL, como se muestra en la Figura 1.1.

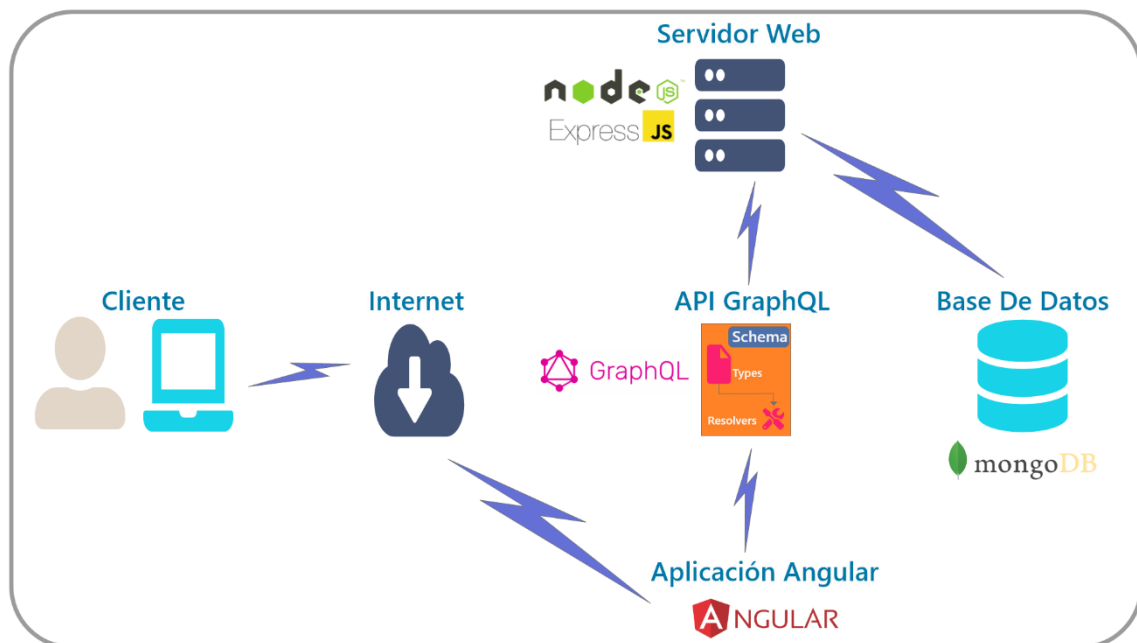


Figura 1.1. Diagrama del prototipo

1.3 MARCO TEÓRICO

En esta sección se realizó una revisión sobre las principales organizaciones que promueven la creación y cooperación entre CSIRT (FIRST, ENISA, CMU SEI, TF-CSIRT, OPEN-CSIRT) y el stack MEAN+G (MongoDB, Express, Angular, Node.JS, GraphQL) con el que se desarrolló el sistema web prototipo. Después, se revisaron los conceptos básicos acerca de las bases de datos no relacionales, sistemas distribuidos, servicios

web, Bootstrap, Typescript, Javascript y la metodología ágil SCRUM. Por último, se especificaron las herramientas tecnológicas utilizadas en el sistema prototipo.

1.3.1 CSIRT

Un Equipo de Respuesta a Incidentes de Seguridad Informática es una organización que se encarga de recibir, revisar y responder a los informes y actividades relacionados con incidentes de seguridad, con el fin de determinar acciones para minimizar los daños ocasionados por dichos incidentes y permitir que el CSIRT se recupere en el menor tiempo posible [6].

Los CSIRT se pueden clasificar dependiendo del sector al que prestan sus servicios. Según la Agencia Europea de Seguridad de las Redes y de la Información (ENISA)[5] se tienen los siguientes sectores:

- CSIRT del sector académico: Se relaciona con instituciones académicas y educativas.
- CSIRT comercial: Se relaciona con empresas que ofrecen servicios comerciales, tales como un proveedor de servicios de internet (ISP).
- CSIRT del sector de la protección de la información crítica y de la información y las infraestructuras críticas (CIP/CIIP): Se relaciona con la infraestructura de TI que es crítica para un gobierno y sus ciudadanos.
- CSIRT del sector público: Se relaciona con las instituciones públicas de un país.
- CSIRT interno: Se relaciona solamente con la empresa a la que pertenece.
- CSIRT del sector militar: Se relaciona con organizaciones militares con propósitos de defensa nacional.
- CSIRT nacional: Se relaciona con un país y es considerado como un punto central de contacto.
- CSIRT del sector de la pequeña y mediana empresa (PYME): Se relaciona con empresas que ofrecen servicios a un grupo de interés.
- CSIRT de soporte: Se relaciona con productos específicos de un fabricante.

1.3.2 ENTIDADES QUE PROMUEVEN LA CREACIÓN DE CSIRT

1.3.2.1 FIRST

El Foro de equipos de seguridad y respuesta a incidentes (FIRST) es una organización fundada en 1990 debido a los constantes incidentes a sistemas informáticos generados en las redes interconectadas por Internet. Sus integrantes son CSIRT de varios sectores que cooperan entre sí de manera voluntaria con la finalidad de abordar problemas de seguridad informática, así como la mejora de las habilidades de sus miembros para tratar y prevenir dichos incidentes de seguridad informática [7].

1.3.2.2 CARNEGIE MELLON SEI

El Instituto de ingeniería de Software (SEI) de la Universidad Carnegie Mellon (CMU), es un centro de investigación y desarrollo que se encarga del estudio de vulnerabilidades de seguridad en productos de software. Además, ofrecen cursos y capacitaciones para ayudar a organizaciones interesadas con la creación de su propio CSIRT y mejorar prácticas de ciberseguridad [8].

1.3.2.3 ENISA

La Agencia de la Unión Europea para la Ciberseguridad (ENISA) es la encargada de contribuir con la seguridad de las redes y la información de la Unión Europea (UE) mediante políticas, esquemas de certificación de ciberseguridad, cooperación e intercambio de conocimientos con sus estados miembros [9].

1.3.2.4 TF-CSIRT

El Grupo de Trabajo sobre Equipos de Respuesta a Incidentes de Seguridad Informática (TF-CSIRT) facilita la colaboración y coordinación entre los CSIRT de Europa y países vecinos mediante el impulso de uso de estándares y procedimientos para el manejo de incidentes de seguridad. Además, TF-CSIRT apoya el desarrollo de nuevos CSIRT, ofrece cursos de capacitación al personal de TI y mantiene un programa de certificación de Equipos de Respuesta a Incidentes de Seguridad Informática dependiendo de su nivel de madurez [10].

1.3.2.5 OPEN-CSIRT

OPEN-CSIRT Es una fundación independiente sin fines de lucro creada en 2016, cuya misión se basa en contribuir a la seguridad de internet alrededor del mundo mediante la oferta de servicios como capacitaciones, investigaciones, estandarización y mejores prácticas de seguridad informática [11].

1.3.3 BASES DE DATOS NO RELACIONALES

Las bases de datos no relacionales o NO-SQL (Not only SQL⁵) son aquellas que no utilizan esquemas basados en filas y columnas como en las bases de datos tradicionales, sino que los datos pueden ser almacenados como pares clave-valor simples, documentos de tipo JSON o en gráficos. Este tipo de bases de datos garantizan la gestión de grandes volúmenes de datos en ambientes distribuidos, y proveen fácil acceso a través de una API. Para la implementación del sistema prototipo, se optó por una base de datos NO-SQL debido a que se puede guardar y recuperar información usando interfaces simples sin uniones como en las bases de datos relacionales, evitando así el requerimiento de sistemas complejos de mapeo relacional de objetos para manipular datos. Por otro lado, cuando se agregan más procesadores se obtiene un aumento constante en el rendimiento [12].

1.3.4 STACK MEAN

El conjunto de tecnologías MEAN se fundamentan en Javascript como lenguaje de programación y son usadas para la creación de aplicaciones, mientras que GraphQL es usado para ejecutar consultas al servidor.

1.3.4.1 MONGODB

MongoDB es una base de datos no relacional de código abierto y orientada a documentos, la cual ofrece un modelo de consultas avanzado. MongoDB almacena los datos en esquemas de estructuras dinámicas de tipo JSON, en lugar de tablas como en las bases de datos tradicionales [13].

1.3.4.2 EXPRESS

Express es una infraestructura web y de código abierto para Node, que es usado para el diseño y creación de aplicaciones web de una manera rápida y sencilla debido a que es compatible con JavaScript. De esta manera, se puede administrar el enrutamiento, las sesiones, las solicitudes HTTP⁶ y el manejo de errores de una aplicación web [14].

1.3.4.3 ANGULAR

Angular es un framework de código abierto y multiplataforma desarrollado por Google que se usa en el lado del frontend⁷ para realizar aplicaciones de una sola página. Este

⁵ SQL (Structured Query Language): Es un lenguaje de consultas que permite la administración y recuperación de información de las bases de datos relacionales.

⁶ HTTP (Hypertext Transfer Protocol): Es un protocolo de comunicación que facilita la transmisión de información por medio de archivos en la web.

⁷ Frontend: Es la parte del lado del cliente e interactúa con los usuarios de un sistema o aplicación web.

framework utiliza HTML⁸ y TypeScript, y está compuesto por módulos, componentes y servicios. Los módulos son librerías que contienen componentes, proveedores de servicios y archivos de código. Los componentes son elementos que contienen la lógica y los datos de la aplicación. Además, se relaciona con la vista por medio de una plantilla HTML y los servicios permiten establecer comunicación con el backend⁹ [15].

1.3.4.4 NODE

Node es un entorno para ejecutar JavaScript desde lado del servidor, el cual, soporta múltiples plataformas informáticas y es de código abierto. Este entorno fue construido con el motor JavaScript V8¹⁰ y se caracteriza por utilizar un modelo asíncrono dirigido por eventos y mantener miles de conexiones abiertas y en espera con un solo servidor [16].

1.3.4.5 GRAPHQL

GraphQL es un lenguaje de consultas de código abierto creado por Facebook en el 2012 y surge como una alternativa a REST. Así, GraphQL expone un único endpoint y responde exactamente los datos que el cliente solicitó a una API¹¹, en vez de varios endpoints que retornan estructuras de datos fijas [17]. Las operaciones que se pueden realizar con GraphQL se las definen a continuación:

- Queries: Son operaciones que permiten consultar datos.
- Mutations: Son operaciones que sirven para modificar datos.
- Subscriptions: Son operaciones que permiten obtener notificaciones sobre cambios realizados en los datos en tiempo real.

1.3.5 SISTEMAS DISTRIBUIDOS

Un sistema distribuido se puede considerar como un conjunto de computadoras autónomas, las cuales, colaboran entre sí y dan a entender a un usuario que se está interactuando con un único sistema. Entre las principales ventajas se destacan la escalabilidad y facilidad de integración de diferentes aplicaciones ejecutadas en diversas computadoras [18].

1.3.6 SERVICIOS WEB

Un servicio web es una tecnología, la cual utiliza una agrupación de protocolos que permiten intercambiar datos entre aplicaciones de tipo cliente-servidor,

⁸ HTML (HyperText Markup Language): Es un lenguaje usado para la creación de páginas web.

⁹ Backend: Es la parte del lado del servidor que procesa y envía la información solicitada por el frontend.

¹⁰ Motor JavaScript V8: Es un motor de código abierto para JavaScript desarrollado por Google

¹¹ API (Application Programming Interfaces): Es un conjunto de protocolos y definiciones que permiten la comunicación entre varios sistemas.

independientemente del lenguaje o plataforma en la cual se desarrolla. Se puede crear un servicio web mediante el protocolo SOAP¹², la arquitectura REST¹³ o mediante GraphQL [19].

1.3.7 BOOTSTRAP

Bootstrap es uno de los framework CSS¹⁴ más populares para frontend que permite realizar el diseño de sitios web responsivos de una manera sencilla por medio de una biblioteca de componentes ya prediseñados [20].

1.3.8 JAVASCRIPT

JavaScript (JS) es un lenguaje de programación interpretado, basado en objetos, no tipado y de alto nivel, el cual fue diseñado para que las páginas web tengan un comportamiento más interactivo con los usuarios [21].

1.3.9 TYPESCRIPT

Typescript (TS) es un lenguaje de programación fuertemente tipado y de código abierto desarrollado por Microsoft. Typescript se considera un superconjunto de Javascript ya que permite expandir sus capacidades y facilitar la detección de errores en el código del programa sin necesidad de ejecutarlo [22].

1.3.10 SCRUM

SCRUM es un framework utilizado en el desarrollo de software para ofrecer entregas continuas de valor al cliente durante la realización del proyecto. Esto se logra debido a que se crean ciclos de trabajo cortos y bien estructurados llamados Sprints y se establece una responsabilidad colectiva y de progreso continuo entre todos los involucrados del proyecto [23].

Los roles centrales y no centrales de SCRUM, así como ciertos conceptos básicos, se definen a continuación:

- **Product Owner:** Persona responsable de obtener todos los requisitos del proyecto por parte del cliente y de lograr el máximo valor del proyecto al mínimo costo.
- **Scrum Master:** Persona encargada de guiar y enseñar los procesos de SCRUM al equipo involucrado con el fin de completar con éxito el proyecto.

¹² SOAP (Simple Object Access Protocol): Es un protocolo basado en XML desarrollado para el intercambio de información.

¹³ REST (Representational State Transfer): Es una arquitectura que usa HTTP entre varios sistemas para la obtención de datos en formatos como XML o JSON.

¹⁴ CSS (Cascading Style Sheets): Es un lenguaje que se usa para dar estilos a un documento de tipo HTML.

- **Equipo Scrum:** Personas encargadas de comprender los requerimientos dados por el Product Owner y de crear los entregables del proyecto.
- **Stakeholder(s):** Clientes, usuarios y patrocinadores que se comunican con el equipo SCRUM.
- **Vendedores:** Personas o instituciones externas que ofertan servicios o productos fuera de las competencias de la organización del proyecto.
- **Backlog priorizado del producto:** Lista de requisitos del proyecto en forma de historias de usuario ordenados en orden de importancia.
- **Historias de usuario:** Estructuras simples que indican los requerimientos y funcionalidades que desea el cliente.
- **Sprint Backlog:** Lista de tareas que deben ser completadas en un Sprint.

Las fases de SCRUM con sus respectivas actividades se resumen en la Tabla 1.1.

Tabla 1.1. Fases de SCRUM

No.	Fase	Actividades
1	Inicio	<ul style="list-style-type: none"> • Creación de la visión del proyecto. • Formación del equipo de trabajo. • Realización del Backlog priorizado del producto. • Planificación de los lanzamientos.
2	Planificación y estimación	<ul style="list-style-type: none"> • Creación y estimación de las historias de usuario y de las tareas. • Creación del Sprint backlog.
3	Implementación	<ul style="list-style-type: none"> • Crear los entregables del producto. • Cambiar y actualizar el Backlog Priorizado del producto según sea necesario.
4	Revisión y retrospectiva	<ul style="list-style-type: none"> • Demostración y validación de los Sprints. • Realizar la retrospectiva del Sprint.
5	Lanzamiento	<ul style="list-style-type: none"> • Envío de los entregables del producto. • Realizar la retrospectiva del proyecto.

1.3.11 HERRAMIENTAS TECNOLÓGICAS

A continuación, se describen las principales herramientas tecnológicas utilizadas para la creación del sistema web prototipo.

- **Visual Studio Code:** Editor de código fuente multiplataforma, gratuito y potente que soporta múltiples lenguajes de programación [24].

- **Github:** Plataforma que permite alojar código fuente de programas, además, facilita el control de versiones y colaboración con otros usuarios desde cualquier parte del mundo [25].
- **Adobe XD:** Plataforma de diseño basado en vectores que permite la creación de prototipos de páginas web y aplicaciones móviles. Esta plataforma se usará en este trabajo de titulación para crear los wireframes y mockups del sistema web [26].
- **Lucidchart:** Software que permite crear diagramas de cualquier tipo de manera online [27].
- **MongoDB Compass:** GUI¹⁵ gratuita para realizar consultas de los datos de MongoDB. Además, se permite agregar y analizar la información almacenada gracias a su interfaz centralizada de fácil navegación para el usuario [28].

¹⁵ GUI (Graphic User Interface): Es una interfaz gráfica, la cual permite la interacción entre usuario y máquina de una manera sencilla.

2 METODOLOGÍA

En esta sección se especifica el diseño e implementación del sistema web. En la etapa de diseño se determinaron los documentos necesarios en el proceso de creación de un CSIRT con su respectivo contenido, mediante una comparación de diferentes estándares que promueven el establecimiento de CSIRT (FIRST, CMU-SEI, OPEN-CSIRT, TF-CSIRT, ENISA). A continuación, se definieron reuniones con CSIRT-EPN y GT-Ciberseguridad¹⁶ para determinar requerimientos funcionales y no funcionales del sistema web y se generaron historias de usuario en base a las funcionalidades encontradas. Más adelante, se creó el backlog priorizado del producto basado en la metodología ágil SCRUM. Después, se definió la cantidad y duración de los Sprints necesarios para el proyecto. También se establecieron todos los usuarios que podrán interactuar con el sistema mediante la asignación de roles, para acceder a ciertos recursos de la aplicación web. Posteriormente, se realizó un diagrama de clases y casos de usos para definir la funcionalidad y estructura del sistema. Luego se modeló la base de datos con sus respectivos esquemas. Por otro lado, se crearon wireframes para la interfaz del usuario y una vez aprobados por el GT-Ciberseguridad se procedió al diseño de mockups que sirvieron para los módulos del prototipo.

En la etapa de implementación se generó la base de datos en MongoDB de acuerdo con los esquemas establecidos en la etapa de diseño. Además, se crearon los repositorios del proyecto en la plataforma GitHub y de manera local. Después, se codificaron cada uno de los módulos del prototipo con sus respectivas interfaces en base al diseño realizado.

2.1 DISEÑO

A continuación, se resumen los diferentes estándares de las organizaciones que promueven la creación y cooperación entre CSIRT.

2.1.1 FIRST

La Tabla 2.1 muestra los pilares y estándares recomendados por FIRST.

¹⁶ GT-Ciberseguridad: Grupo de Trabajo en Analítica de Datos e Inteligencia Artificial aplicado a la Ciberseguridad que pertenece a la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia (CEDIA) y cuyo propósito es ayudar a las Instituciones de Educación Superior (IES) a neutralizar ataques de seguridad informáticos.

Tabla 2.1. Pilares fundamentales y estándares recomendados por FIRST

PARÁMETROS		DESCRIPCIÓN
PILARES	Constitución	Permite determinar las necesidades y las interacciones con los miembros de la organización.
	Políticas	Se definen políticas que establecen guías para cumplir con los objetivos que permitan el funcionamiento exitoso del CSIRT.
	Infraestructura	Se define el lugar de trabajo y las instalaciones de la organización.
	Manejo de Incidentes	Se establecen métodos, procesos y tecnologías utilizadas para la gestión de incidentes de seguridad.
ESTÁNDARES Y PROTOCOLOS	Traffic Light Protocol (TLP)	Protocolo de comunicación que se basa en 4 colores que establece un tipo específico de difusión de la información: Rojo (restringida), ámbar (limitada), verde (limitada dentro de la comunidad) y blanco (sin restricción).
	RFC 2350	Se establece la información, políticas y procedimientos relacionados con el CSIRT.
	Security Incident Management Maturity Model (SIM3)	Modelo para establecer maduración de un CSIRT. Consta de 44 parámetros divididos en 4 cuadrantes: Organización, talento humano, herramientas y procedimientos.
	Common Vulnerability Scoring System (CVSS)	Sistema que indica las características y vulnerabilidades de software y se basa en 3 métricas: Base, temporal y entorno.
	Information Exchange Policy (IEP)	Políticas que permiten establecer el intercambio de información entre diferentes CSIRT.
	Framework de servicios	Ayuda a las organizaciones a construir, mantener y aumentar las capacidades de sus CSIRT mediante 5 ámbitos de servicios: Gestión de eventos de seguridad de la información, gestión de incidentes de seguridad de la información, gestión de vulnerabilidades, consciencia coyuntural y transferencia de conocimientos.
	Intercambio DNS Pasivo	Define un formato de salida común de servidores DNS pasivos para los clientes.

2.1.2 CARNEGIE MELLON SEI

La Tabla 2.2 muestra los pilares y estándares recomendados por CMU SEI.

Tabla 2.2. Pilares fundamentales y estándares recomendados por CMU SEI

PARÁMETROS		DESCRIPCIÓN
PILARES	Framework	Se establecen las pautas con las que operará un nuevo CSIRT.
	Políticas	Principios implementados por el CSIRT.
	Equipo de trabajo	Procedimientos relacionados con el personal del CSIRT y la infraestructura.
	Manejo de Incidentes	Componentes fundamentales y procedimientos para el correcto manejo de alarmas, reportes e incidentes.

MANUALES	Manuales para establecimiento de CSIRTS SEI	- Evaluación de la capacidad de gestión de incidentes (2018).
		- Definición de procesos de gestión de incidentes para CSIRT: Un trabajo en progreso (2004).
		- Manual para equipos de respuesta a incidentes de seguridad informática (CSIRT) (2003).

2.1.3 ENISA

La Tabla 2.3 muestra los pilares y estándares recomendados por ENISA.

Tabla 2.3. Pilares fundamentales y estándares recomendados por ENISA

PARÁMETROS		DESCRIPCIÓN
PILARES	Constitución	Se describen la naturaleza y los objetivos del CSIRT.
	Políticas	Directrices que deben ser seguidas por el personal en la realización de operaciones relacionadas con el CSIRT.
	Infraestructura	Servicios básicos que el CSIRT debe tener, incluyendo la infraestructura de TI, arquitectura y diagramas de red.
	Recursos Humanos	Abarca conocimientos, experiencia y capacitación del personal del CSIRT.
ESTÁNDARES Y MANUALES	ISO 2700x	114 controles en 14 grupos y 35 objetivos de control.
	Security Incident Management Maturity Model (SIM3)	Modelo para establecer maduración de un CSIRT. Consta de 44 parámetros divididos en 4 cuadrantes: Organización, talento humano, herramientas y procedimientos.
	Manuales ENISA	Un enfoque paso a paso sobre cómo configurar un CSIRT (2006).
		Capacidades de un CSIRT: Directrices para CSIRT nacionales y gubernamentales. (2016).
		Actividades de desarrollo de capacidades relacionadas con CSIRT.
	Guía para establecer un CSIRT y SOC (2020).	

2.1.4 TF-CSIRT

La Tabla 2.4 muestra los pilares y estándares recomendados por TF-CSIRT.

Tabla 2.4. Pilares fundamentales y estándares recomendados por TF-CSIRT

PARÁMETROS		DESCRIPCIÓN
PILARES	Fundación	Son las bases principales para iniciar el establecimiento del CSIRT.
	Organización	Se define el mandato y otras estructuras organizativas internas dentro de la organización matriz y coordinación con otros CSIRT. (11 parámetros).
	Herramientas	Se establece todo lo necesario para realizar las tareas definidas (10 parámetros).
	Procesos	Se relaciona con el manejo o interacción de amenazas e incidentes con los medios (17 parámetros).
	Talento Humano	Se define el personal, estructura, experiencia, código de conducta y opciones de capacitación del equipo de trabajo del CSIRT. (7 parámetros).
ESTÁNDARES Y MANUALES	Traffic Light Protocol (TLP)	Protocolo de comunicación que se basa en 4 colores que establece un tipo específico de difusión de la información: Rojo (restringida), ámbar (limitada), verde (limitada dentro de la comunidad) y blanco (sin restricción).
	RFC 2350	Consta de información, políticas y procedimientos relacionados con el CSIRT.
	SIM3	Modelo para establecer maduración de un CSIRT. Consta de 44 parámetros divididos en 4 cuadrantes: Organización, talento humano, herramientas y procedimientos.
	RFC 2119	Palabras claves para establecer qué se debe, qué no se debe, qué se debería, qué no se debería y qué se puede realizar.
	CSIRT Code of Practice (CCoP)	Código de prácticas para los equipos de seguridad cibernética.

2.1.5 OPEN-CSIRT

La Tabla 2.5 indica los pilares y estándares recomendados por OPEN-CSIRT.

Tabla 2.5. Pilares fundamentales y estándares recomendados por OPEN-CSIRT

PARÁMETROS		DESCRIPCIÓN
PILARES	Organización	Conjunto de talento humano, recursos, herramientas e infraestructura que trabajan juntos de manera planificada. (Mandato, configuración y servicios).
	Herramientas	Programas, aplicaciones, servicios, instrumentos y equipos para alcanzar los objetivos y ofrecer los servicios definidos en el área de la organización.
	Proceso	Conjuntos de acciones secuenciadas lógicamente que son realizadas por el área humana o área de herramientas para lograr un resultado específico en el área de organización. Los procesos son documentados, medibles y repetibles.
	Talento Humano	Personal para proporcionar servicios y cumplir el mandato. Capacitaciones y experiencia.
	Traffic Light Protocol (TLP)	Protocolo de comunicación que se basa en 4 colores que establece un tipo específico de difusión de la información:

ESTÁNDARES Y MANUALES		Rojo (restringida), ámbar (limitada), verde (limitada dentro de la comunidad) y blanco (sin restricción).
	RFC 2350	Consta de información, políticas y procedimientos relacionados con el CSIRT.
	Capacitaciones	TRANSITS-I/II (ENISA ó FIRST), CERT/CC, SANS.
	RFC 2142	Enumera y describe las direcciones de correo de Internet que se utilizará al contactar al personal en una organización.
	CSIRT Code of Practice (CCoP)	Código de prácticas para los equipos de seguridad cibernética.

2.1.6 REUNIÓN CON CSIRT-EPN

Con el apoyo de la MSc. Liliana Córdova, encargada de CSIRT-EPN, se llevó a cabo una reunión en la que se identificaron 18 documentos básicos necesarios en el proceso de creación de un CSIRT y 11 requisitos físicos que no necesitan documentación. Además, se estableció que, de los 18 documentos, el acta de constitución, el código de ética y el programa de entrenamiento son propios de cada CSIRT, por lo tanto, dichos documentos no se los tomará en cuenta para la creación del sistema prototipo. En el Anexo A se encuentran los documentos establecidos en la reunión con CSIRT-EPN.

2.1.7 DOCUMENTOS DEL SISTEMA WEB

Para el proyecto se establecieron 15 documentos divididos en 4 pilares fundamentales de un CSIRT: Organización, Talento Humano, Políticas e Infraestructura, los cuales, se los indican a continuación:

2.1.7.1 PILAR ORGANIZACIÓN

En este pilar se establece el mandato y la estructura interna del CSIRT, además de la coordinación con otros CSIRT, en total se obtuvieron 3 documentos: El primero constituye el documento de creación de CSIRT. El segundo establece el acta de constitución propia de cada CSIRT y el tercero corresponde al plan estratégico. En la Tabla 2.6 se indica la estructura de los documentos mencionados anteriormente.

Tabla 2.6. Documentos del pilar Organización

DOCUMENTOS	CONTENIDO
DOCUMENTO DE PROYECTO DE	Nombre del proyecto.
	Descripción del proyecto.
	Propósito del proyecto.
	Definiciones, siglas y abreviaciones.
	Antecedentes: Marco legal de la constitución del CISRT.
	Objetivos del proyecto.
	Alcance.

CREACIÓN	Fases del proyecto.
	Supuestos, restricciones y riesgos del proyecto.
	Infraestructura.
	Recursos del proyecto.
	Presupuesto.
	Capacitaciones del personal.
	Nivel de clasificación del documento.
ACTA DE CONSTITUCIÓN DEL CSIRT	Documento de constitución propio de cada CSIRT. No se tiene un esquema definido.
PLAN ESTRATÉGICO	Misión.
	Visión.
	Comunidad objetivo.
	Objetivos.
	Relación con la comunidad objetivo: Completa, compartida, ninguna.
	Ubicación en la estructura organizacional.
	Roles y responsabilidades.
	Relaciones externas con otros CSIRT.
	Servicios.
	Políticas.
	Estructura del CSIRT.
	Infraestructura.
	Cronograma de actividades.
Nivel de clasificación del documento.	

2.1.7.2 PILAR TALENTO HUMANO

El pilar de Talento Humano involucra aspectos relacionados con el personal del CSIRT. Es decir, la experiencia del equipo, las capacitaciones al grupo de trabajo y el código de conducta que maneja la organización. En la Tabla 2.7 se indican los documentos requeridos.

Tabla 2.7. Documentos del pilar Talento Humano

DOCUMENTOS	DESCRIPCIÓN
CÓDIGO DE ÉTICA	Documento público y a la vista en físico. Propio de cada CSIRT por lo que no se tiene un esquema definido.
PROGRAMA DE ENTRENAMIENTO	Certificados de capacitaciones realizadas por el personal del CSIRT. Los cursos Transists I y Transists II ofertados por la organización GÉANT son obligatorios para el responsable del proyecto y para el responsable del triage.

2.1.7.3 PILAR POLÍTICAS

En este pilar se establecen las normas y directrices que garantizan una efectiva gestión de los incidentes informáticos. En total se definieron 11 políticas y 2 procedimientos ligados a las directrices y normativas de cada institución en la que se encuentren los usuarios.

- Política de clasificación de la información.

- Política de protección de la información.
- Política de eliminación de la información.
- Política de retención de la información.
- Política de difusión de la información.
- Política de acceso a la información.
- Política de uso aceptable de la información.
- Política de definición de incidentes y eventos de seguridad informática.
- Política de gestión de incidentes de seguridad informáticos.
- Política de cooperación con otros grupos.
- Política de acceso al CSIRT.
- Procedimiento para la gestión de incidentes de seguridad informática.
- Procedimiento de acceso a activos críticos del CSIRT.

Cada una de las políticas deben contar con los ítems que se indican en la Tabla 2.8. Cabe mencionar que para el desarrollo del sistema no se toman en cuenta los ítems de historial de revisiones e historial de aprobaciones ya que esto lo realiza el personal que pertenece a una organización certificadora de CSIRT.

Tabla 2.8. Ítems de cada una de las políticas establecidas

ITEM	CONTENIDO
OBJETIVO	Procesos para alcanzar las metas propuestas por el equipo de trabajo del CSIRT.
ALCANCE	Se establece el trabajo que se debe realizar para alcanzar los objetivos propuestos por el CSIRT.
POLÍTICA	Se establecen disposiciones generales, clasificación y aplicación de la información.
REFERENCIAS	Bibliografía relacionada con la creación de la política.
CUMPLIMIENTO	Acciones que se deben realizar en caso de infringir la política.
DEFINICIONES	Términos usados en la elaboración de la política.
NIVEL DE CLASIFICACIÓN DEL DOCUMENTO	Se usa el protocolo TLP (Traffic Light Protocol) para el intercambio de la información.
TABLA DE CONTROL DE CAMBIOS	Consta de la siguiente información mínima: Fecha del cambio, ítem, aspecto cambiado, persona que solicitó el cambio y la persona que implementa el cambio.
HISTORIAL DE REVISIONES	Tabla con las revisiones realizadas a la política por organizaciones externas al CSIRT. Información mínima requerida: Fecha de la revisión, persona que realizó la

	revisión y comentarios.
HISTORIAL DE APROBACIONES	Tabla con el historial de aprobaciones realizada a la política por organizaciones externas al CSIRT. Información mínima requerida: Fecha de la aprobación, persona que se realizó la aprobación y comentarios.

Cada uno de los procedimientos deben contar con los ítems que se indican en la Tabla 2.9. Cabe mencionar que para el desarrollo del sistema no se toman en cuenta los ítems de historial de revisiones e historial de aprobaciones ya que esto lo realiza el personal que pertenece a una organización certificadora de CSIRT.

Tabla 2.9. Ítems de cada uno de los procedimientos establecidos

ITEM	CONTENIDO
OBJETIVO	Procesos para alcanzar las metas propuestas por el equipo de trabajo del CSIRT.
ALCANCE	Se establece el trabajo que se debe realizar para alcanzar los objetivos propuestos por el CSIRT.
DEFINICIONES	Términos usados en el procedimiento.
ÁMBITO DE APLICACIÓN	Delimitación de la validez del procedimiento.
PROCEDIMIENTO	Consta de las siguientes fases: Preparación, detección, contención y actividades post incidentes.
INDICADORES	Se establece mediante la relación: Incidentes gestionados/Incidentes reportados.
REFERENCIAS	Bibliografía usada para la realización del procedimiento.
ANEXOS	Listado de activos críticos, roles y perfiles del personal del CSIRT.
NIVEL DE CLASIFICACIÓN DEL DOCUMENTO	Se utiliza el protocolo TLP (Traffic Light Protocol).
TABLA DE CONTROL DE CAMBIOS	Consta de la siguiente información mínima: Fecha del cambio, ítem, aspecto cambiado, persona que solicitó el cambio y la persona que implementa el cambio.
HISTORIAL DE REVISIONES	Tabla con las revisiones realizadas al procedimiento por organizaciones externas al CSIRT. Información mínima requerida: Fecha de la revisión, persona que realizó la revisión y comentarios.
HISTORIAL DE APROBACIONES	Tabla con el historial de aprobaciones realizada al procedimiento por organizaciones externas al CSIRT. Información mínima requerida: Fecha de la aprobación, persona que se realizó la aprobación y comentarios.

2.1.7.4 PILAR INFRAESTRUCTURA

La infraestructura incluye el equipamiento físico que constituye el CSIRT. Así, en la parte de infraestructura se establecieron 11 requisitos principales, sin embargo, dichos requisitos no necesitan documentación. La Tabla 2.10 indica los requisitos básicos para la parte de Infraestructura.

Tabla 2.10. Requisitos establecidos por el pilar Infraestructura

REQUISITOS	CARACTERÍSTICAS
PORTAL WEB	Es independiente de la universidad a la que pertenece el CSIRT.
TELÉFONO DE COMUNICACIÓN	Debe estar cifrado.
SISTEMA CORREO ELECTRÓNICO RESISTENTE	Propio del CSIRT, comunicación cifrada PGP o GPG. No en Nube.
ACCESO Y DOMINIO A INTERNET	Independiente de la universidad.
AUTENTICACIÓN PROPIA	Se puede hacer uso de Active Directory.
SISTEMA DE GESTIÓN DE INCIDENTES	Uso de sistemas OTRS (Open-source Ticket Request System) para realizar el seguimiento de un incidente.
SERVIDOR DNS	Propio del CSIRT.
SERVIDOR DHCP	Propio del CSIRT.
ALMACENAMIENTO FÍSICO	Se debe tener un espacio de almacenamiento físico mínimo de 7 a 10 años.
DESTRUCTOR	Aparatos para deshacerse de documentos y CD.
CCTV	Tener al menos acceso a las cámaras de seguridad.

2.1.8 REQUERIMIENTOS

Una vez identificados los documentos, se procedió a establecer los requerimientos funcionales y no funcionales del sistema web mediante reuniones establecidas con el GT-Ciberseguridad, los cuales, se los muestran a continuación:

2.1.8.1 REQUERIMIENTOS FUNCIONALES

- Autenticación de usuarios se la realizará por medio de un correo electrónico y contraseña.
- Cerrar la sesión de usuario desde cualquier página del sistema.
- Restablecimiento de la contraseña de un usuario vía correo electrónico.
- Crear, editar, eliminar y activar información relacionada con los usuarios
- Crear, editar, eliminar y activar información relacionada con las universidades
- Usuario con rol Responsable podrá asignar documentos a usuarios de diferentes roles pertenecientes a una misma universidad.
- Crear y editar versiones de cada ítem de un documento específico
- Crear, editar y eliminar comentarios de un documento específico.
- Generar documentos en formato PDF.

- Visualizar información de las fechas de entrega de un documento mediante un calendario.
- Visualizar el estado de los documentos mediante un cuadro de estado de tareas.
- No se contará con una interfaz de administrador general del sistema y en el caso de requerir operaciones CRUD se las realizará directamente en la base de datos.

2.1.8.2 REQUERIMIENTOS NO FUNCIONALES

- La aplicación web consumirá servicios web mediante una API GraphQL.
- La implementación de la base de datos se la realizará en MongoDB.
- Las credenciales de los usuarios no se almacenarán en la base de datos en texto plano, sino que se cifrarán usando el algoritmo bcrypt.

2.1.9 EQUIPO DE TRABAJO

Para el proyecto se realizó la formación del equipo de trabajo con sus respectivos roles y se los indican en la Tabla 2.11.

Tabla 2.11. Roles SCRUM

Nº	NOMBRE	ROL SCRUM	ORGANIZACIÓN
1	Msc. Franklin Sánchez	Scrum Master	EPN
2	GT-Ciberseguridad	Product Owner	CEDIA
3	Erick Ríos	Development Team	EPN

2.1.10 HISTORIAS DE USUARIO

Las historias de usuario (HU) sirven para describir el funcionamiento de un sistema desde el punto de vista del usuario [29]. Se establecieron 15 historias de usuarios para el sistema web y cada HU tiene los siguientes campos:

- **Número:** Este campo indica el número de la historia de usuario con el formato HU0X.
- **Usuario:** Este atributo especifica los usuarios del sistema web involucrados en la HU.
- **Nombre Historia:** Se establece el nombre de la historia de usuario en letras mayúsculas.

- **Prioridad en negocio:** Campo para establecer la prioridad de la historia de usuario en el sistema web. Los valores que puede tener son alto, medio y bajo.
- **Riesgo en desarrollo:** Campo para establecer el riesgo que conlleva el no cumplir con la HU en el sistema web. Los valores que se puede tener son alto, medio y bajo.
- **Puntos estimados:** Este campo establece mediante un valor del 1 al 5 la dificultad y tiempo necesario para la realización de la HU.
- **Iteraciones asignadas:** Este campo establece la cantidad de Sprints necesarios para finalizar la HU.
- **Programador responsable:** Nombre del programador encargado de codificar la funcionalidad de la HU.
- **Descripción:** En esta parte se escribe una breve definición de lo que quiere realizar un usuario para cumplir con la HU.
- **Validación:** Criterios para comprobar el correcto funcionamiento de la HU.

En la Figura 2.1 se muestra la historia de usuario HU002 en la que se especifica el inicio de sesión de un usuario con cualquier rol al sistema web. Las demás historias de usuarios se encuentran en el Anexo B.


 HISTORIA DE USUARIO	
Número: 002	Usuario: Gestor, Responsable, Supervisor, Colaborador
Nombre Historia: INICIO DE SESIÓN	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 5	Iteración asignada: 1
Programador Responsable: Erick Ríos	
Descripción: Como usuario gestor, responsable, supervisor o colaborador quiero ingresar un correo electrónico y una contraseña para poder acceder al sistema web.	
Validación: - El usuario debe ingresar una contraseña mayor a 4 caracteres. - El usuario debe ingresar un correo electrónico válido.	

Figura 2.1. Historia de usuario HU002

2.1.11 BACKLOG PRIORIZADO DEL PRODUCTO

La Tabla 2.12 muestra el backlog priorizado del producto que se desarrolló en base a las historias de usuarios definidas.

Tabla 2.12. Backlog priorizado del producto

N° HU	NOMBRE	CARACTERÍSTICA/FUNCIÓNALIDAD	ROL
HU001	CREACIÓN DEL AMBIENTE DE DESARROLLO	Codificación de la base de datos y creación de los repositorios de trabajo para el frontend y el backend.	Sistema web.
HU002	INICIO DE SESIÓN	Usuarios con diferentes roles pueden ingresar al sistema web.	Gestor, responsable, colaborador y supervisor.
HU003	RECUPERACIÓN DE CONTRASEÑA	Usuarios con diferentes roles pueden recuperar la contraseña en caso de olvidarse.	Gestor, responsable, colaborador y supervisor.
HU004	REGISTRO USUARIO	Usuarios pueden registrarse en el sistema.	Colaborador y supervisor.
HU005	GESTIÓN UNIVERSIDADES	Leer, Editar, asignar, eliminar y bloquear universidades	Gestor.
HU006	GESTIÓN USUARIOS	Leer, Editar, asignar, eliminar y bloquear usuarios	Gestor, responsable.
HU007	GESTIÓN DOCUMENTOS	Visualizar el contenido de un documento específico y asignarlo a los usuarios de la organización	Responsable.
HU008	GESTIÓN PANEL ADMINISTRADOR	Visualización de la cantidad de usuarios, universidades y documentos mediante gráficos estadísticos	Gestor, responsable.
HU009	ESTABLECIMIENTO PANEL HOME	Visualización de 3 columnas para el estado de cada documento y una columna para establecer las fechas de entrega del documento.	Responsable, colaborador, supervisor.
HU010	GESTIÓN MÓDULO COLABORADORES	Visualización de los ítems del documento, edición, creación de versiones y verificación de	Colaborador.

		comentarios	
HU011	GESTIÓN MÓDULO SUPERVISORES	Visualización y edición de los ítems del documento. Crear, editar, eliminar y actualizar comentarios.	Supervisor.
HU012	GESTIÓN MÓDULO FORMULARIO DOCUMENTOS	Visualización del contenido del documento con información de los usuarios a cargo.	Responsable, colaborador y supervisor.
HU013	GESTIÓN MÓDULO DOCUMENTOS FINALES	Generación de documentos en formato PDF.	Responsable, colaborador y supervisor.
HU014	TABLERO ESTADO DE TAREAS	Visualización centralizada del estado de avance de cada uno de los documentos.	Responsable, colaborador y supervisor.
HU015	VISUALIZACIÓN CALENDARIO	Visualización de un calendario con las fechas de entrega de los documentos.	Responsable, colaborador y supervisor.

2.1.12 SPRINT BACKLOG

Las tareas que deberán ser finalizadas en cada uno de los Sprints se detallan en la Tabla 2.13.

Tabla 2.13. Sprint Backlog

N°	NOMBRE SPRINT	HISTORIAS DE USUARIO	TIEMPO
0	Inicio del Proyecto	HU001: CREACIÓN DEL AMBIENTE DE DESARROLLO	1 semana
1	Autenticación	HU002: INICIO DE SESIÓN	4 semanas
		HU003: RECUPERACIÓN DE CONTRASEÑA	
		HU004: REGISTRO USUARIO	
2	Gestión Gestor	HU005: GESTIÓN UNIVERSIDADES	2 semanas
		HU006: GESTIÓN USUARIOS	
		HU008: GESTIÓN PANEL ADMINISTRADOR	
3	Gestión Responsables	HU006: GESTIÓN USUARIOS	2 semanas
		HU007: GESTIÓN DOCUMENTOS	
		HU008: GESTIÓN PANEL ADMINISTRADOR	
4	Gestión Colaboradores	HU010: GESTIÓN MÓDULO COLABORADORES	3 semanas

5	Gestión Supervisores	HU011: GESTIÓN MÓDULO SUPERVISORES	3 semanas
6	Generación de Documentos Finales	HU012: GESTIÓN MÓDULO FORMULARIO DOCUMENTOS	2 semanas
		HU013: GESTIÓN MÓDULO DOCUMENTOS FINALES	2 semanas
7	Gestión Panel Home	HU09: ESTABLECIMIENTO PANEL HOME	3 semanas
		HU014: TABLERO ESTADO DE TAREAS	
		HU015: VISUALIZACIÓN CALENDARIO	

2.1.13 DIAGRAMA DE CASOS DE USOS

Los diagramas de casos de usos son útiles en el desarrollo de software para establecer relaciones entre los distintos tipos de usuarios y un sistema [30]. Así, en el sistema web prototipo se definen los siguientes tipos de usuarios:

- **Gestor:** Usuario encargado de agregar, modificar o eliminar universidades, en el sistema de gestión, así como su respectivo responsable.
- **Responsable:** Usuario encargado de asignar documentos a colaboradores y supervisores, y definir fechas de entrega. Además, puede crear, modificar y eliminar usuarios para su correspondiente equipo de trabajo.
- **Supervisores:** Usuarios encargados de la revisión de cada documento asignado por el responsable de la organización. Además, pueden devolver un documento en el caso de que se requieran correcciones y generar un documento final en formato PDF cuando éste se encuentre listo.
- **Colaboradores:** Usuarios encargados de la edición de cada documento asignado por el responsable de la organización. Además, pueden realizar las respectivas correcciones establecidas por los supervisores.

De acuerdo con los requerimientos establecidos del sistema web, se realizaron los diagramas de casos de usos. Para el módulo de autenticación, en la Figura 2.2 se puede visualizar que los usuarios de tipo Gestor, Responsable, Colaborador y Supervisor pueden iniciar sesión y registrarse en el sistema. Mientras tanto, solo el usuario Gestor no puede registrarse en el sistema ya que dicho usuario es creado por defecto en la base de datos.

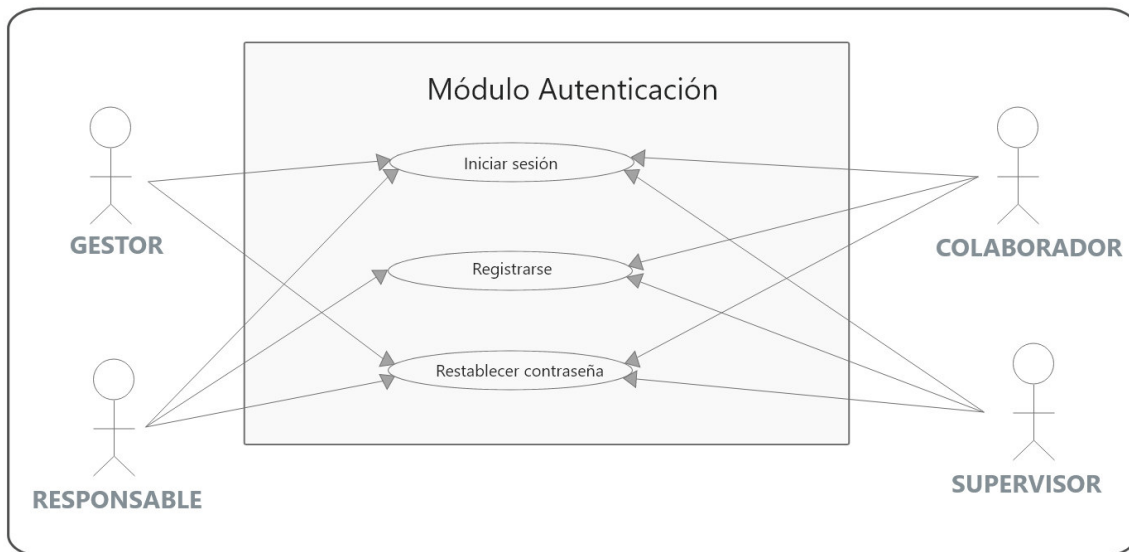


Figura 2.2. Diagrama de casos de uso Módulo Autenticación

En el Módulo Gestor, solamente el usuario con rol Gestor puede realizar funciones como visualizar el panel principal, hacer operaciones CRUD de universidades y usuarios y cerrar la sesión de usuario. En la Figura 2.3 se muestra el diagrama de casos de uso del Módulo Gestor.

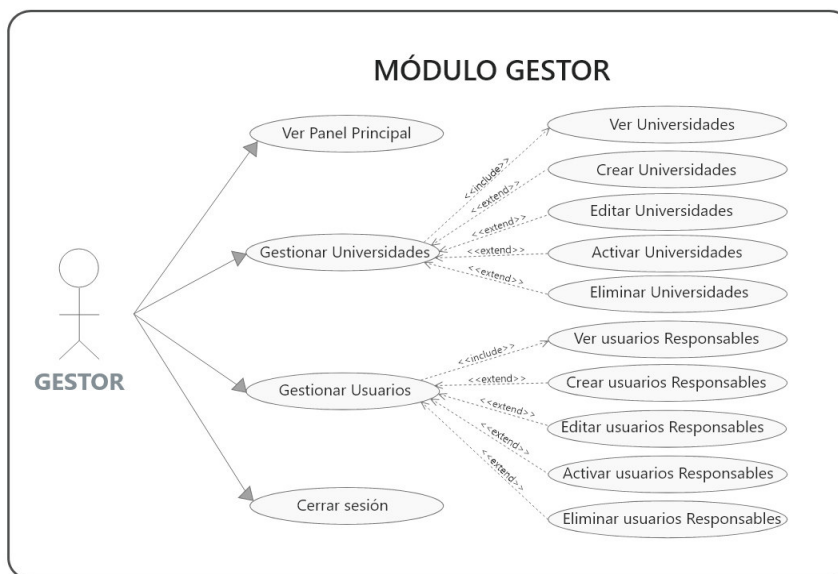


Figura 2.3. Diagrama de casos de uso Módulo Gestor

En el Módulo Gestor, solamente el usuario con rol Responsable puede realizar funciones como visualizar el panel principal, hacer operaciones CRUD de usuarios con rol Colaborador y Supervisor. Además, podrá visualizar los documentos pertenecientes a su

universidad, asignarlos a distintos usuarios y cerrar la sesión. En la Figura 2.4 se muestra el diagrama de casos de uso del Módulo Responsable.

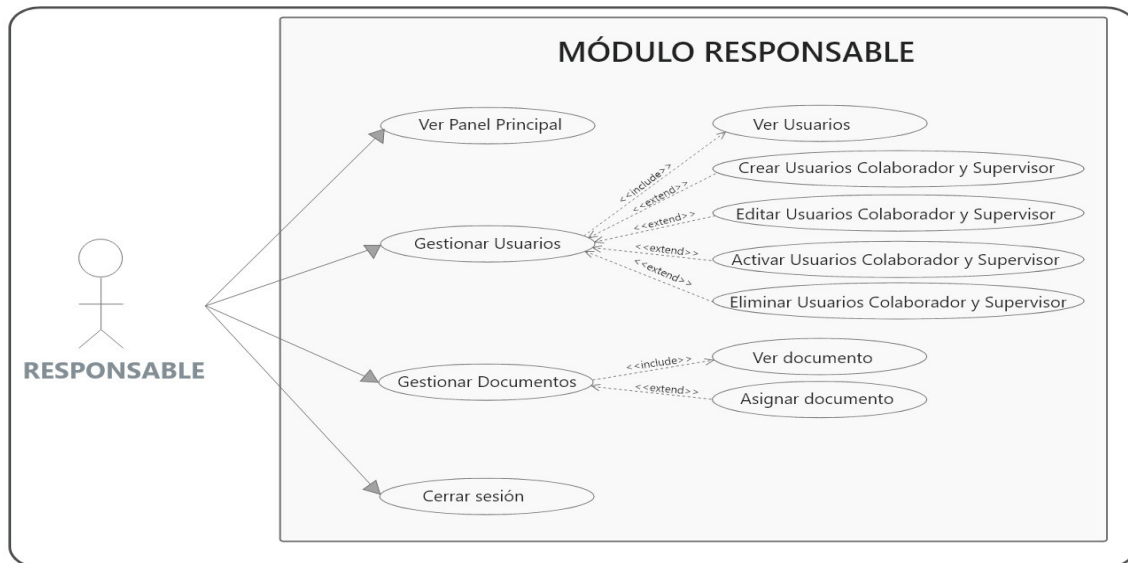


Figura 2.4. Diagrama de casos de uso Módulo Responsable

El Módulo de Home es un panel en el cual los usuarios con rol Responsable, Supervisor y Colaborador pueden visualizar el estado de los documentos y las diferentes tareas creadas. Además, se podrá mirar el calendario con la información de los diferentes documentos asignados, divisar un documento terminado y descargarlo en formato PDF. En la Figura 2.5 se muestra el diagrama de casos de uso del panel Home.

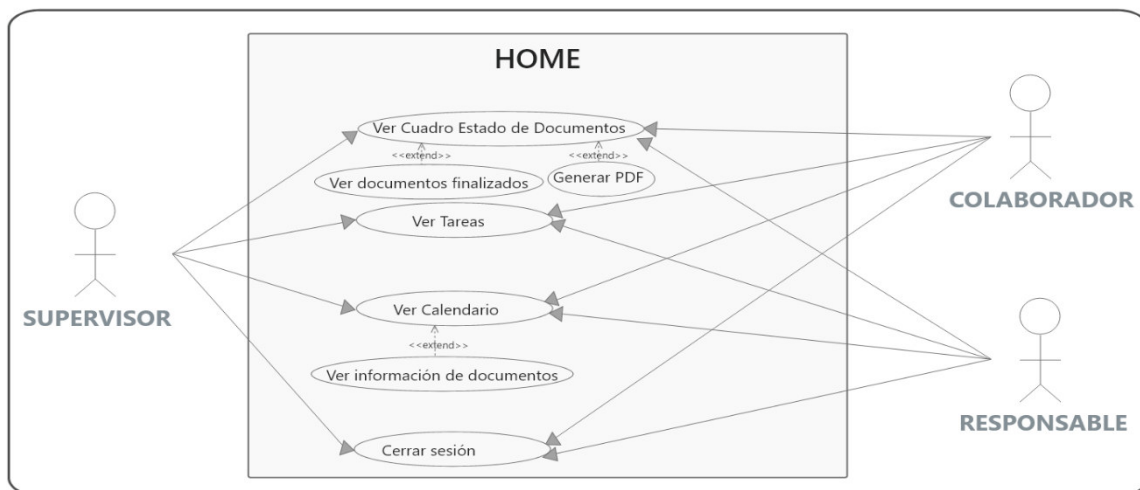


Figura 2.5. Diagrama de casos de uso Panel Home

El Módulo Colaborador permite a los usuarios con rol Colaborador editar el contenido de cada documento asignado, crear versiones de cada ítem de un documento y visualizar los cambios indicados por el usuario con rol Supervisor, así como también dar por

finalizado un cambio recomendado en un documento específico. Cabe mencionar que se cuenta con un usuario Colaborador Principal, el cual será el encargado de cambiar el estado del documento para enviarlo a revisión. En la Figura 2.6 se muestra el diagrama de casos de uso del Módulo Colaborador.

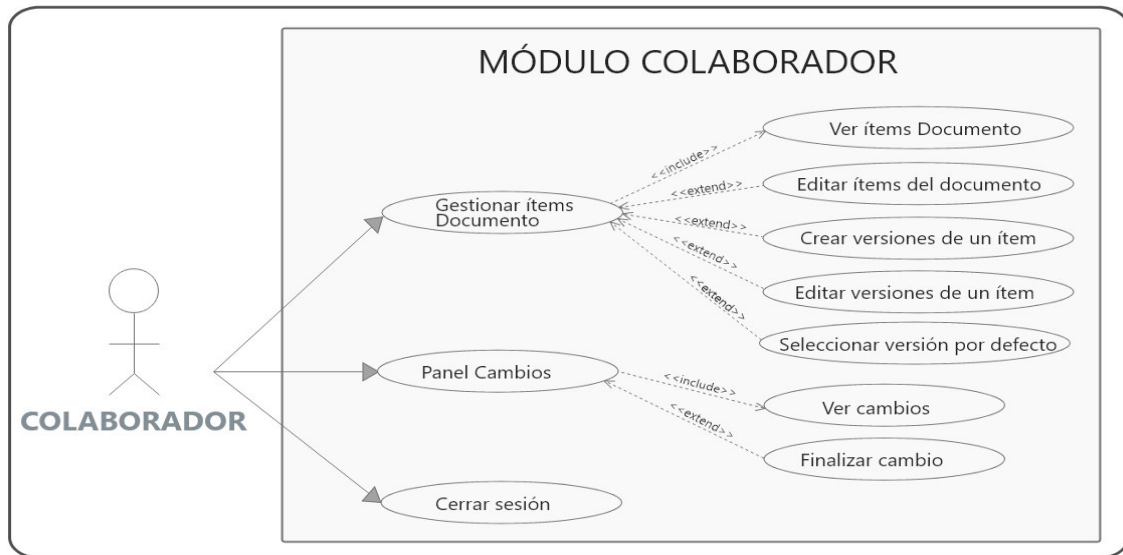


Figura 2.6. Diagrama de casos de uso Módulo Colaborador

El módulo del usuario Supervisor permite realizar la edición de los ítems de un documento, pero no se permite la creación de nuevas versiones de ítems. Además, el usuario Supervisor podrá realizar operaciones CRUD con los cambios indicados a los usuarios Colaboradores. En la Figura 2.7 se muestra el diagrama de casos de uso del Módulo Supervisor.

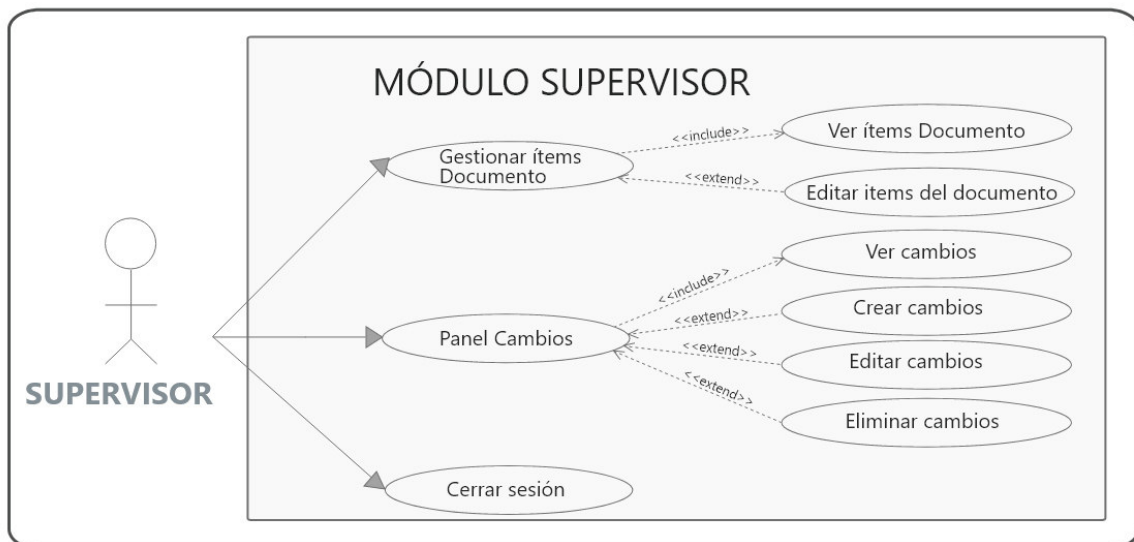


Figura 2.7. Diagrama de casos de uso Módulo Supervisor

2.1.14 DIAGRAMA DE CLASES

Los diagramas de clases son convenientes ya que permiten representar de manera gráfica las diferentes entidades que existen en un sistema y establecer la manera en cómo interactúan entre sí [30] . En la Figura 2.8 se muestra el diagrama de clases del sistema web, en el que se pueden visualizar las 6 entidades: Universidad, usuario, cambios, documento, ítem y versión con sus respectivas relaciones, contenido y funciones.

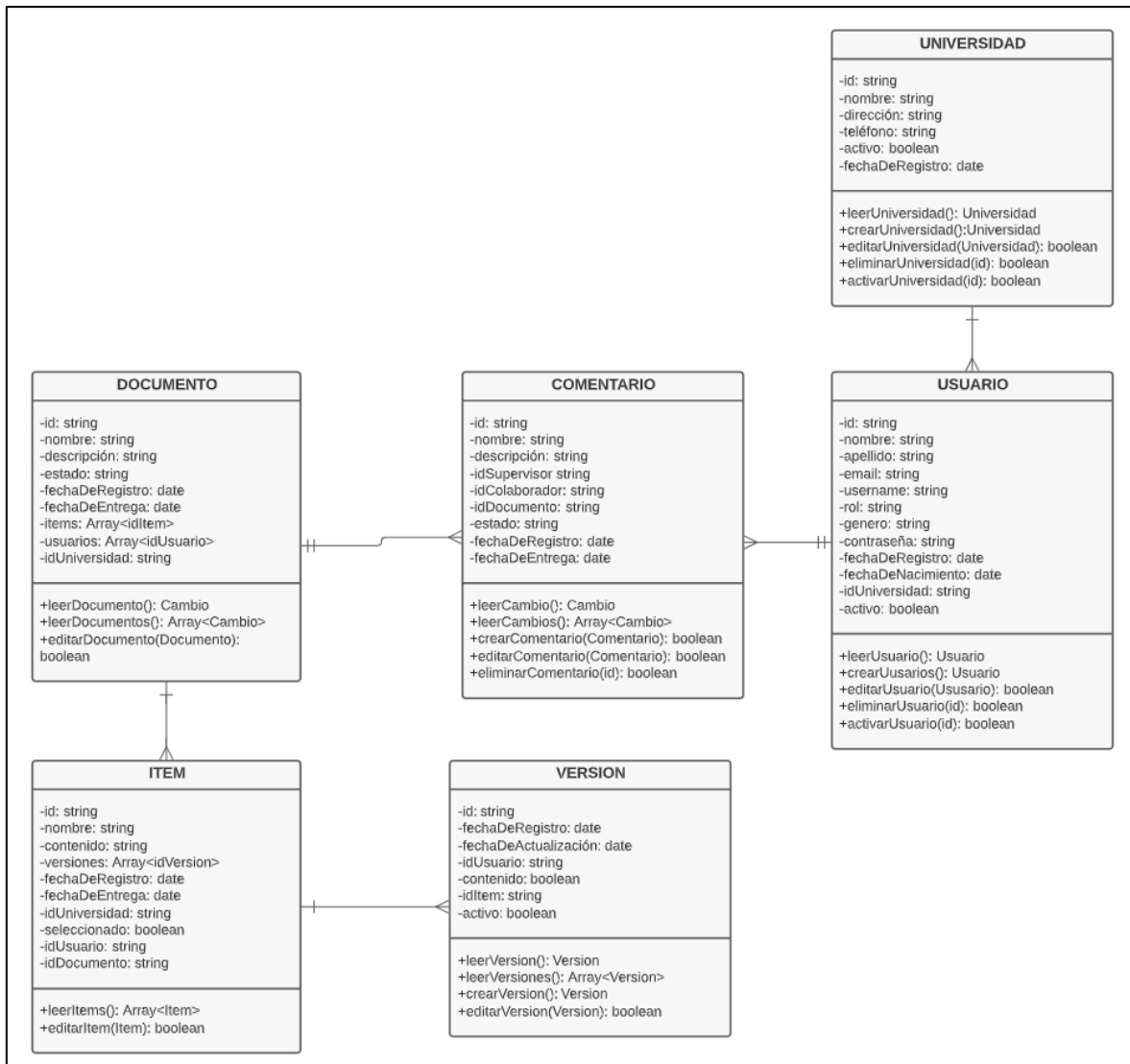


Figura 2.8. Diagrama de clases del sistema web

2.1.15 MODELADO BASE DE DATOS

Una de las particularidades de las bases de datos NO-SQL es la ausencia de esquemas debido a que los datos se almacenan en documentos de tipo JSON. Sin embargo, para fines prácticos es conveniente establecer un diagrama con asociaciones de referencia en

el que se muestre una colección de entidades, las versiones de las entidades y sus respectivas interacciones [31]. En la Figura 2.9 se visualizan las siguientes colecciones:

- users: Información relacionada con los usuarios del sistema web.
- universities: Datos de las universidades a las que pertenecen los usuarios.
- changes: Cambios realizados a los documentos del sistema web.
- documents: Se guarda la información general de los documentos.
- items: Contenido de los documentos.
- Versions: Se almacenan las versiones de los ítems que pertenecen a los documentos.

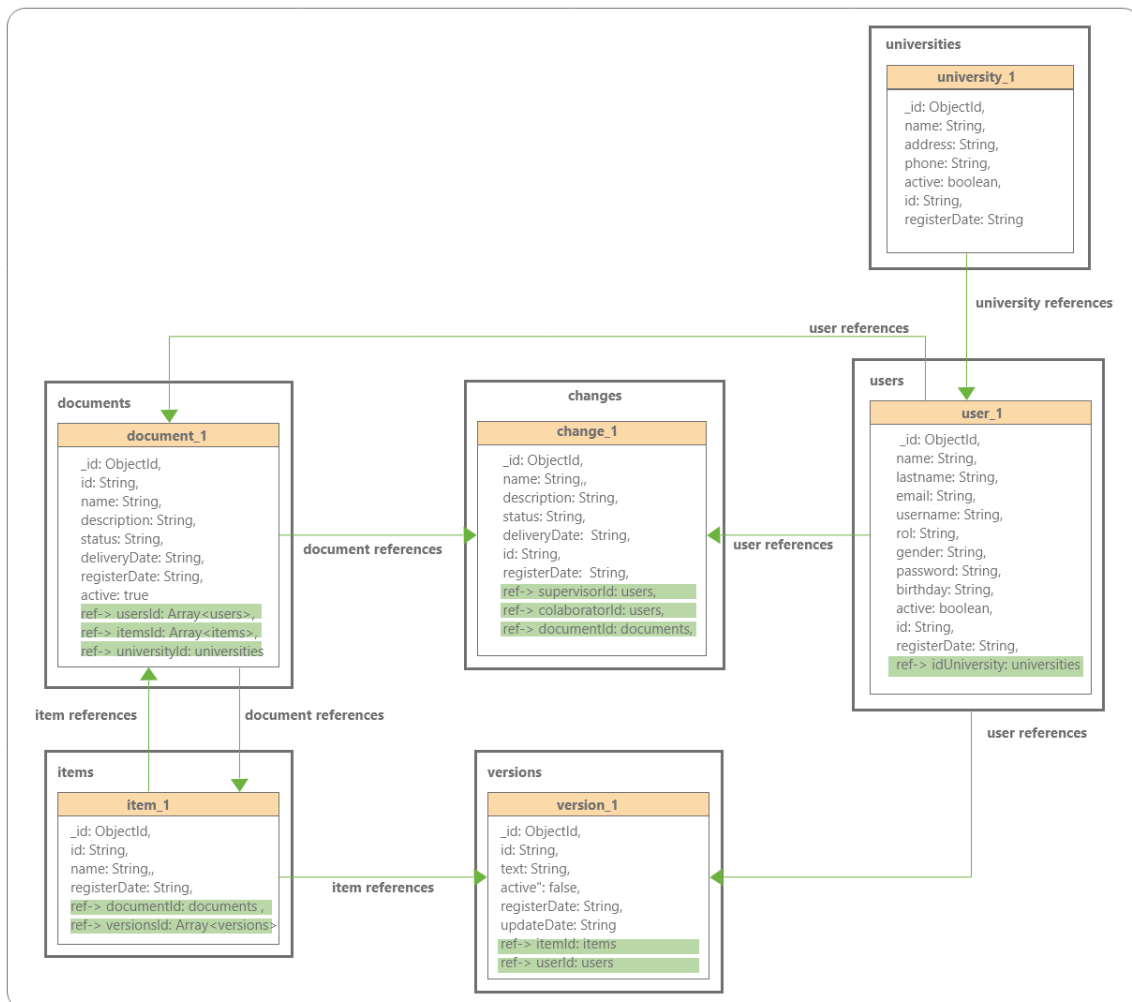


Figura 2.9. Diagrama de base de datos MongoDB

2.1.16 DISEÑO WEB

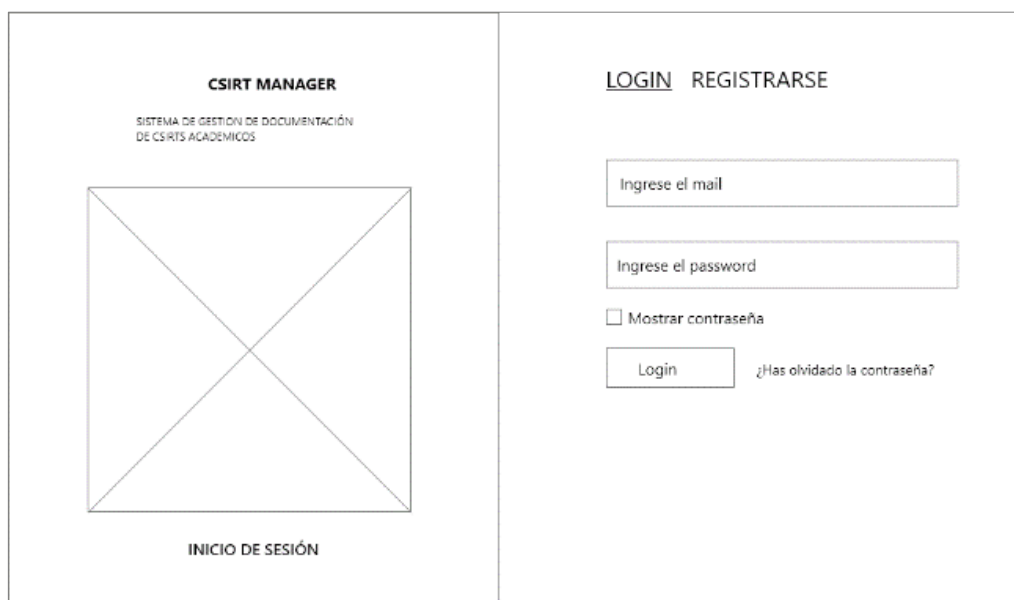
El diseño es una de las etapas principales previo al desarrollo de un sitio o aplicación web y la creación de wireframes y mockups son vitales para ayudar al equipo de trabajo a establecer la apariencia, la estructura y la funcionalidad del sistema.

2.1.16.1 WIREFRAMES

Los Wireframes son ilustraciones consideradas de baja fidelidad debido a que representan el contenido y la estructura de un sitio web sin la necesidad de agregar colores, gráficos y tipografía [32]. Esto los hace fáciles de crear y permiten al equipo de trabajo tener una idea básica de cómo se visualizará la interfaz del usuario. Mediante la herramienta de diseño gráfico Adobe XD y a través de las reuniones establecidas con el GT-ciberseguridad se realizaron y posteriormente se aprobaron los wireframes para el sistema web.

- **Módulo de Autenticación**

En la Figura 2.10 se muestran los wireframes correspondientes al módulo de autenticación. Así, en este módulo se establecieron formularios para el inicio de sesión de un usuario, registro de un usuario nuevo al sistema, envío de un correo de confirmación en caso de pérdida de contraseña y un formulario para cambio de la contraseña por una nueva.



(a)

<p>INICIO REGISTRARSE</p> <p>Ingrese el nombre</p> <p>Ingrese el apellido</p> <p>Ingrese el username</p> <p><input type="radio"/> Hombre <input type="radio"/> Mujer</p> <p>Seleccione una universidad v</p> <p>Escoja fecha de nacimiento </p> <p>Ingrese el mail</p> <p>Ingrese el password</p> <p>Confirme el password</p> <p style="text-align: center;"><input type="button" value="Registrar"/></p>	<p>CSIRT MANAGER</p> <p>SISTEMA DE GESTION DE DOCUMENTACIÓN DE CSIRTS ACADEMICOS</p> <div style="border: 1px solid black; width: 100px; height: 100px; margin: 20px auto; position: relative;"> X </div> <p>Registro Usuario</p>
--	---

(b)

<p>CSIRT MANAGER</p> <p>SISTEMA DE GESTION DE DOCUMENTACIÓN DE CSIRTS ACADEMICOS</p> <div style="border: 1px solid black; width: 100px; height: 100px; margin: 20px auto; position: relative;"> X </div> <p> Regresar al Inicio</p>	<p>Añade tu correo electrónico</p> <p>Te enviaremos un correo electrónico para resetear tu contraseña y cambiarla</p> <hr/> <p>Ingrese el mail</p> <p>Enviar el email de resetear contraseña</p>
--	---

(c)

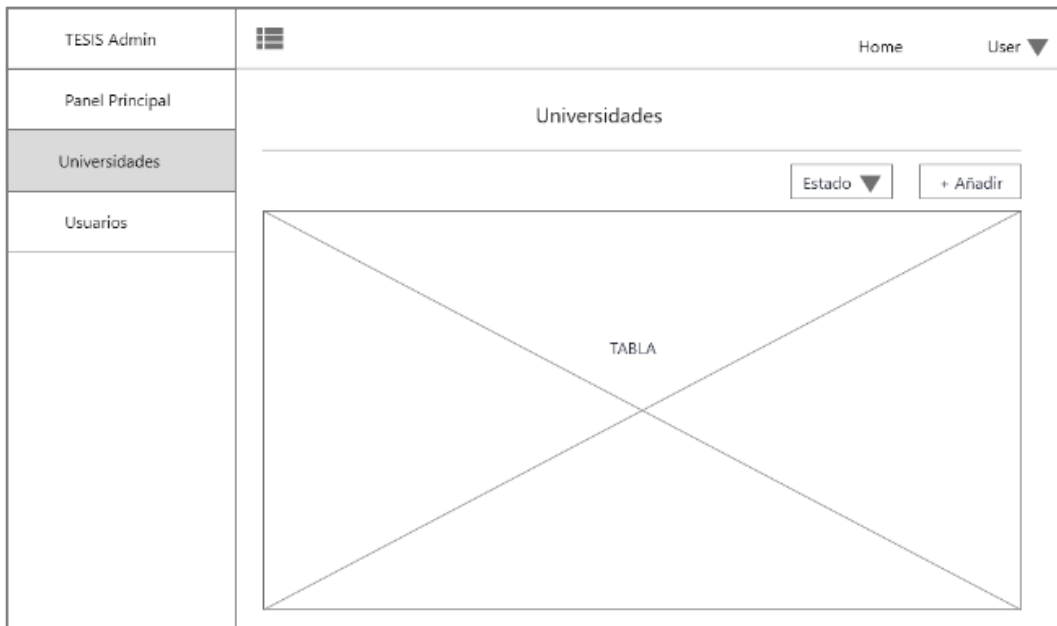
<p>Cambiar Contraseña</p> <p>Para realizar el cambio, por favor ingresa la nueva contraseña y verificalo antes del cambio</p> <hr/> <p>Ingrese el password</p> <p>Confirme el password</p> <p>Cambiar contraseña</p>	<p>CSIRT MANAGER</p> <p>SISTEMA DE GESTION DE DOCUMENTACIÓN DE CSIRTS ACADEMICOS</p> <div style="border: 1px solid black; width: 100px; height: 100px; margin: 20px auto; position: relative;"> X </div> <p>Cambio Contraseña</p>
---	--

(d)

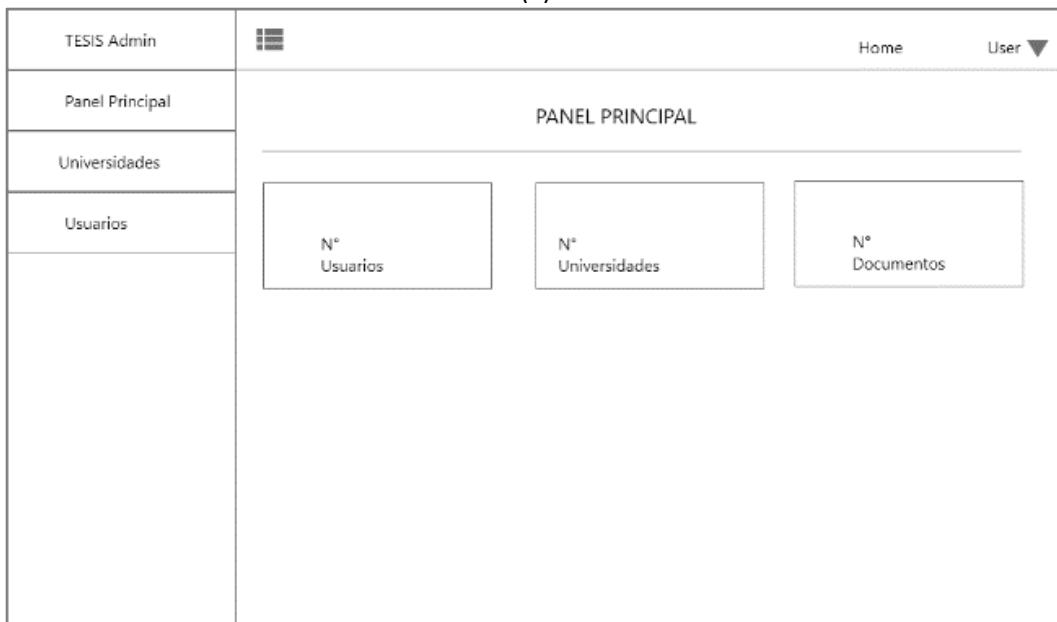
Figura 2.10. Wireframes del módulo de Autenticación. (a) Inicio de sesión, (b) Registro de usuario, (c) Recuperación de contraseña, (d) Cambio de contraseña

- **Módulo Gestor**

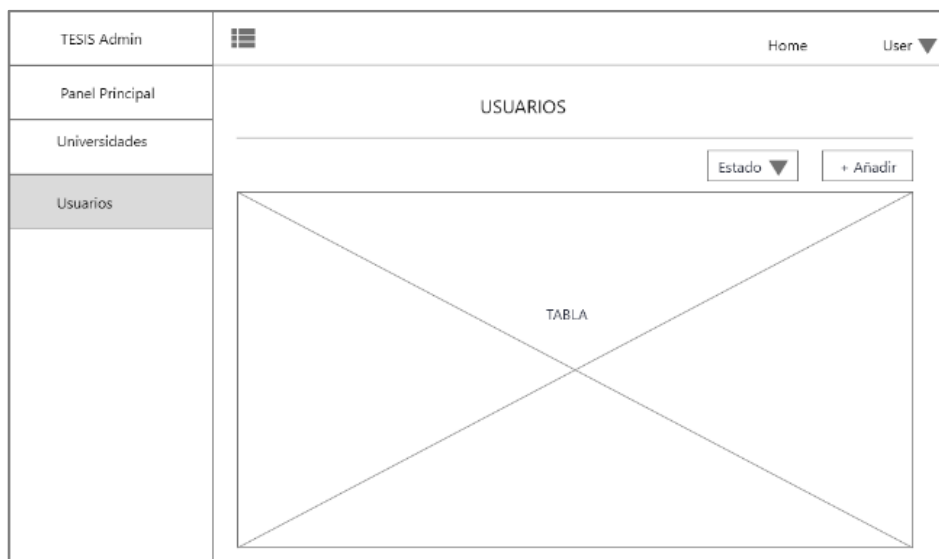
En la Figura 2.11 se muestran los wireframes correspondientes al módulo gestor. Este módulo será accedido únicamente por un usuario con rol Gestor y se podrá visualizar un panel principal con la cantidad de universidades, usuarios y documentos que tiene el sistema. Además, se podrá realizar operaciones CRUD de universidades y usuarios.



(a)



(b)

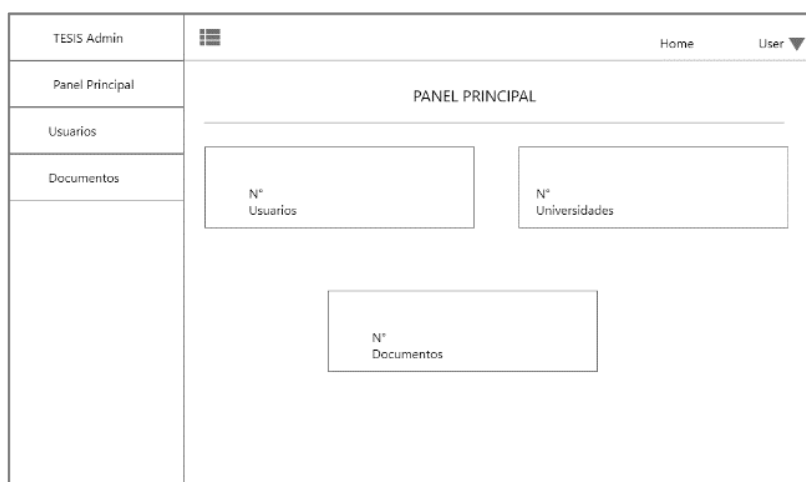


(c)

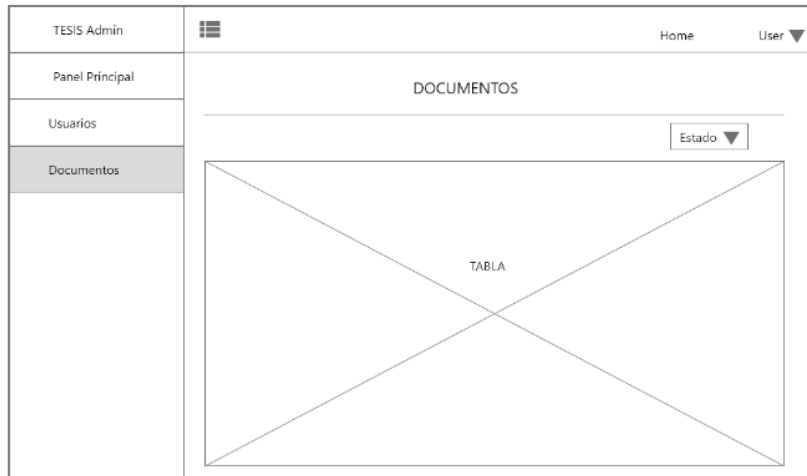
Figura 2.11. Wireframes del Módulo Gestor. (a) Vista principal, (b) Universidades, (c) Usuarios

- **Módulo Responsable**

En la Figura 2.12 se muestran los wireframes correspondientes al Módulo Responsable. Este módulo será accedido únicamente por un usuario con rol Responsable y se podrá visualizar al igual que en el módulo Gestor un Formulario Principal con la cantidad de universidades, usuarios y, además verificar el porcentaje de documentos que están sin asignar, asignados, revisados y terminados. Por otro lado, se tiene un formulario que permite realizar operaciones CRUD de usuarios con rol Responsable, Colaborador y Supervisor. Finalmente se tiene el formulario de Asignación de Documentos, en el cual, se podrá designar un documento a los diferentes usuarios que pertenecen a una misma universidad para que puedan trabajar sobre él.



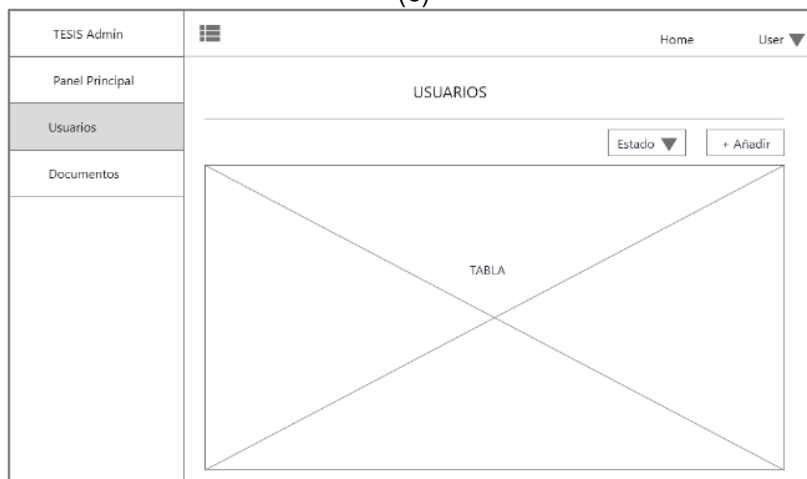
(a)



(b)



(c)



(d)

Figura 2.12. Wireframes del Módulo Responsable. (a) Vista principal, (b) Documentos, (c) Asignación de documento, (d) Usuarios

- **Módulo Colaborador**

En la Figura 2.13 se muestra el wireframe correspondiente al Módulo Colaborador. Este wireframe se divide en 2 partes: En la parte izquierda se podrá visualizar el contenido de un documento específico, además de editarlo y crear versiones de cada ítem. Por otro lado, en la parte derecha, se podrá visualizar los comentarios establecidos por los usuarios con rol Supervisor para realizar los cambios solicitados al documento.

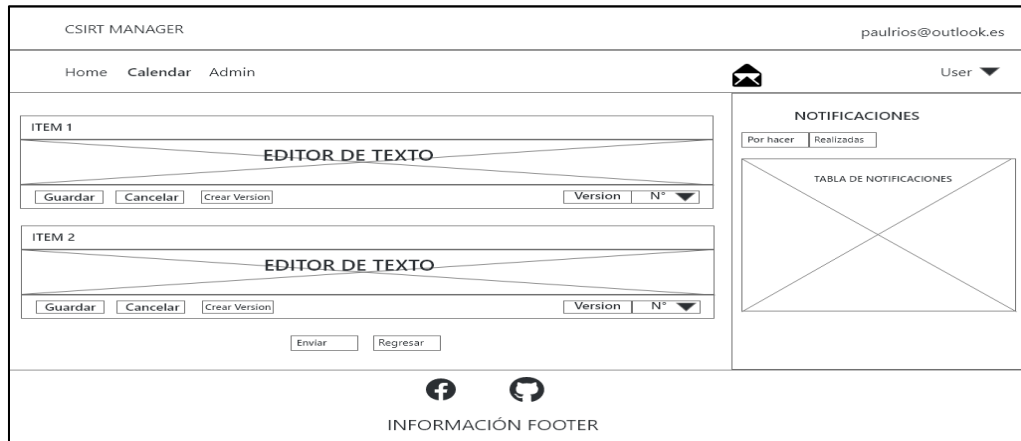


Figura 2.13. Wireframe del Módulo Colaborador

- **Módulo Supervisor**

En la Figura 2.14 se muestra el wireframe correspondiente al Módulo Supervisor. Este wireframe se divide en 2 partes: En la parte izquierda se podrá visualizar el contenido de un documento específico, sin embargo, a diferencia del wireframe del módulo Colaborador, solamente se podrán editar los ítems de un documento específico y no crear versiones de cada ítem. Por otro lado, en la parte derecha, se podrá realizar operaciones CRUD de los comentarios de un documento específico.

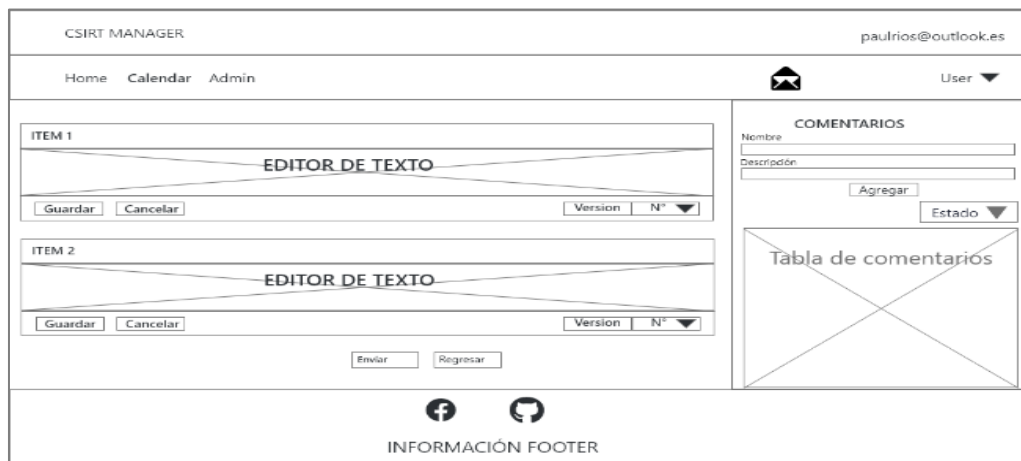


Figura 2.14. Wireframe del Panel Usuario Supervisor

- **Módulo Estado de Tareas**

En la Figura 2.15 se muestra el wireframe correspondiente al Módulo Estado de Tareas. Este wireframe se divide en 2 partes: En la parte lateral izquierda se podrá ver cada documento con su respectivo avance, mientras que en la parte lateral derecha se visualizarán 3 columnas que representan el estado de los documentos.

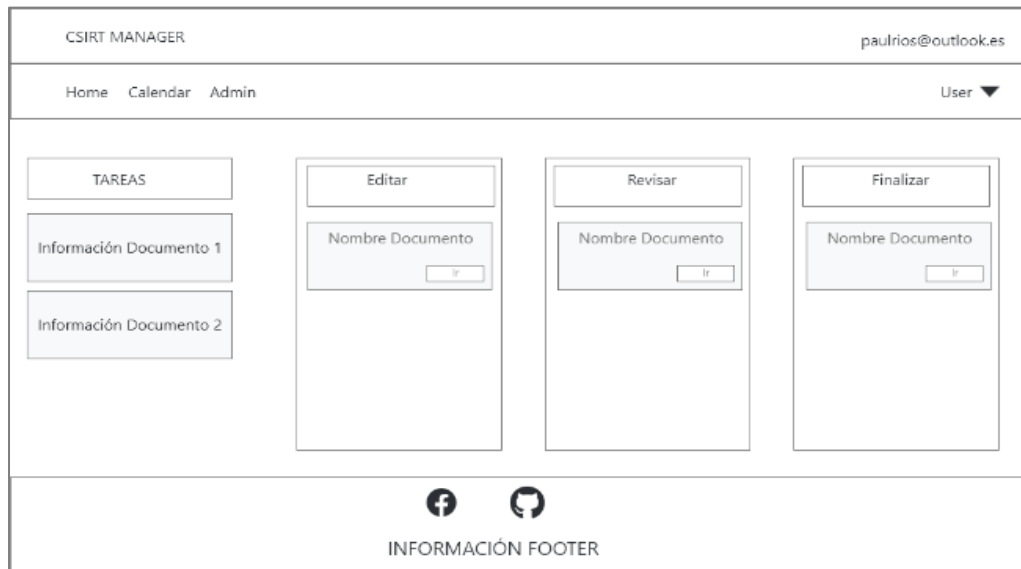


Figura 2.15. Wireframe del Módulo Estado de Tareas

- **Módulo Agenda Actividades**

En la Figura 2.16 se muestra el wireframe correspondiente al Módulo Agenda de Actividades. Este wireframe exhibirá un calendario con todas las fechas de entrega de los documentos asignados.

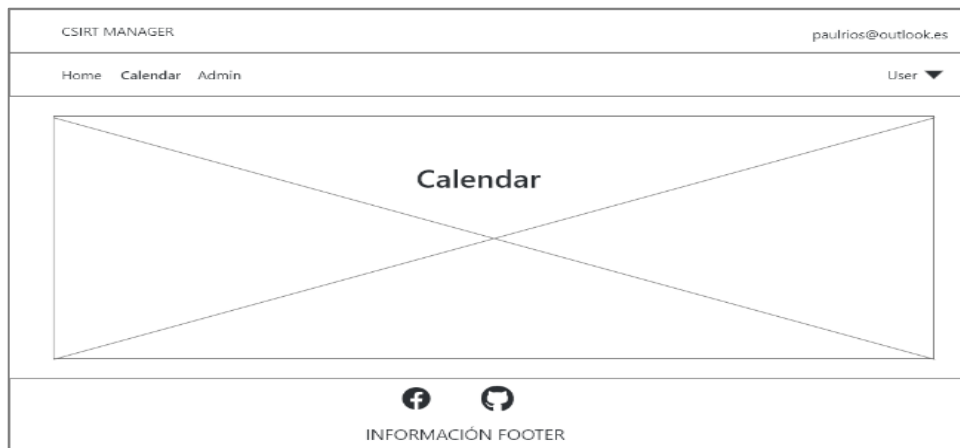


Figura 2.16. Wireframe del Módulo Agenda de Actividades

- **Módulo Formularios Documentos**

En la Figura 2.17 se muestra el wireframe correspondiente al Módulo Formularios Documentos. Este wireframe se divide en 2 partes: En la parte izquierda, se visualizará cada ítem que conforma un documento finalizado, mientras que en la parte derecha se mostrarán los integrantes asignados a un documento específico y un botón que permitirá imprimir el documento en formato PDF.

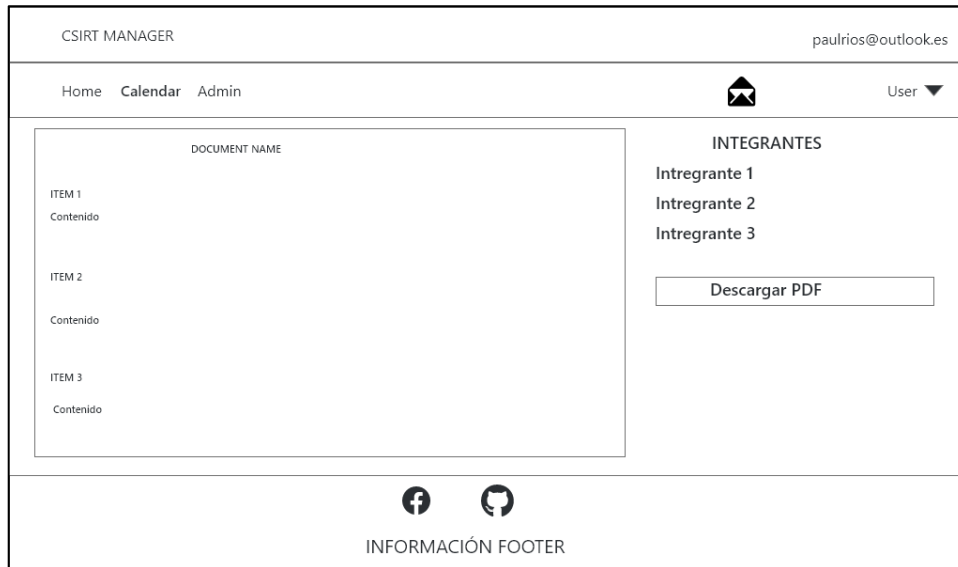


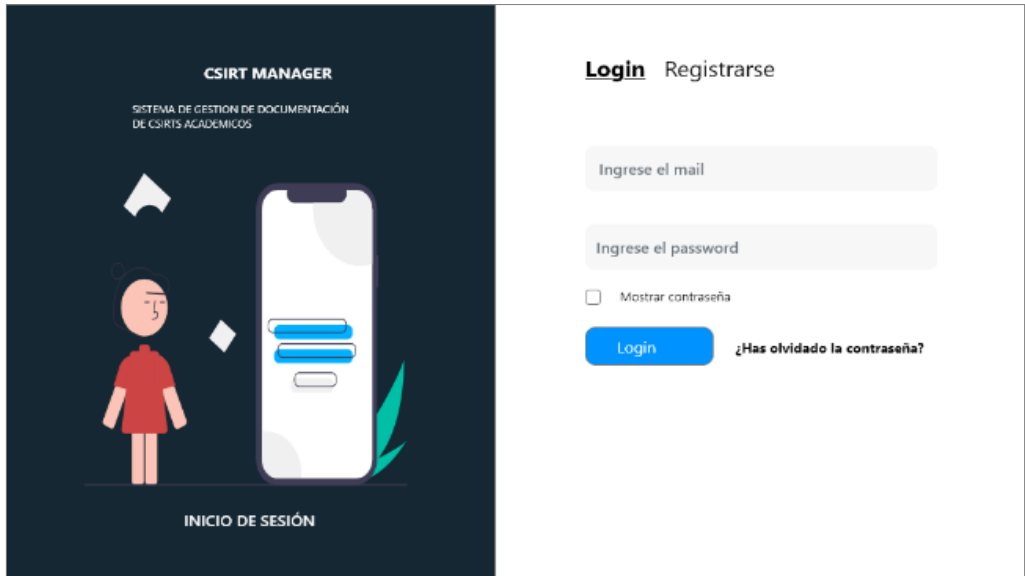
Figura 2.17. Wireframe del Módulo Formularios Documentos

2.1.16.2 MOCKUPS

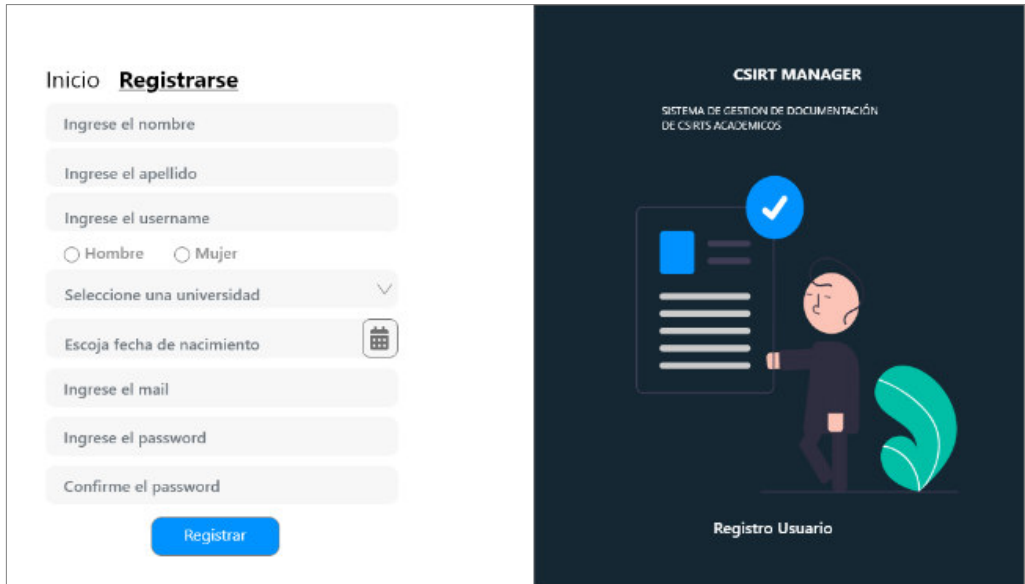
Los mockups son ilustraciones de mediana a alta fidelidad que representan la apariencia y funcionalidad de un producto. Éstos complementan a los wireframes ya que incluyen detalles visuales como colores, tipografías e imágenes, dando una idea de cómo se visualizará y funcionará un producto final de manera estática [33]. Mediante la herramienta de diseño gráfico Adobe XD y a través de las reuniones establecidas con el GT-ciberseguridad se realizaron y posteriormente se aprobaron los mockups para el sistema web.

- **Modulo Autenticación**

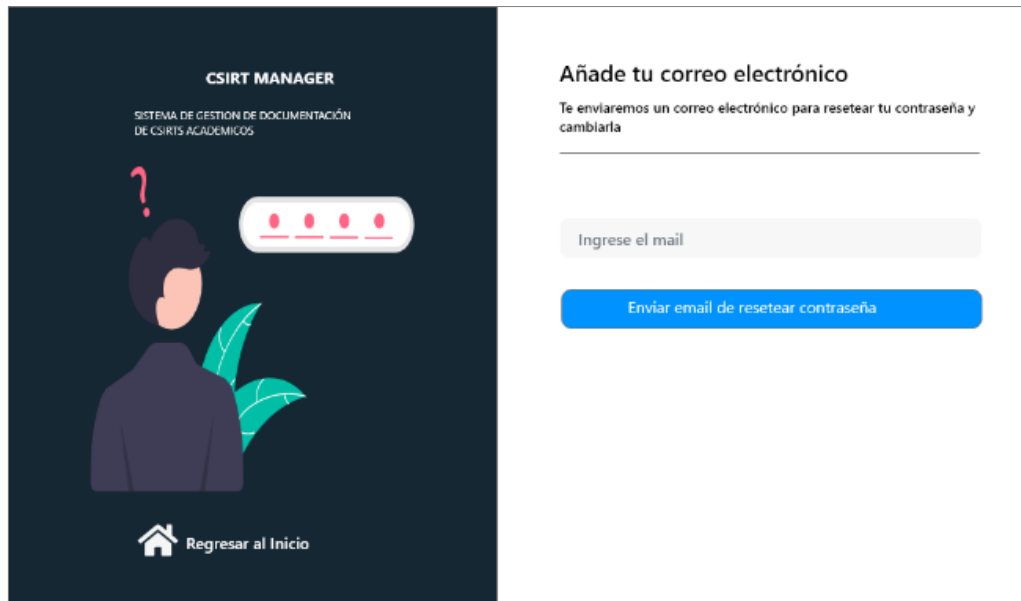
En la Figura 2.18 se presentan los mockups del módulo de autenticación.



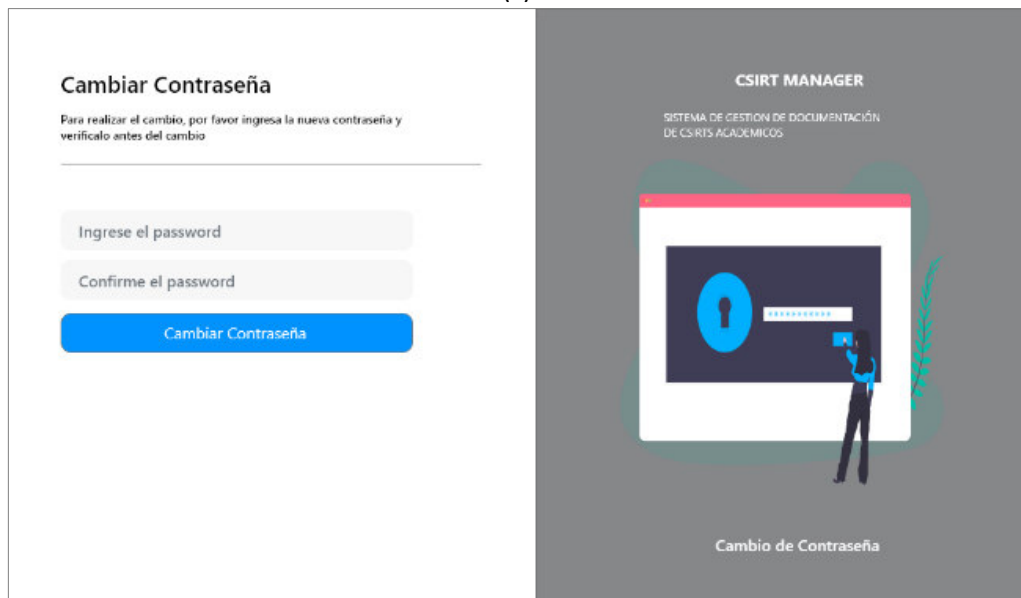
(a)



(b)



(c)

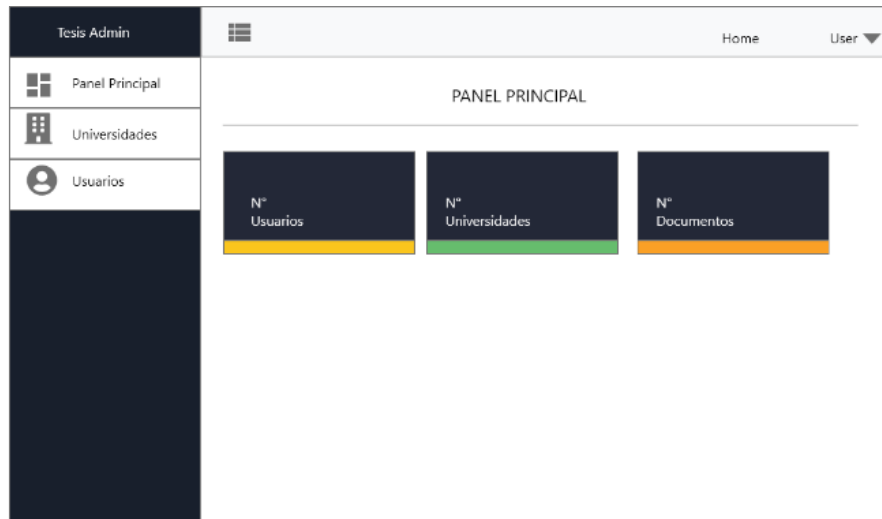


(d)

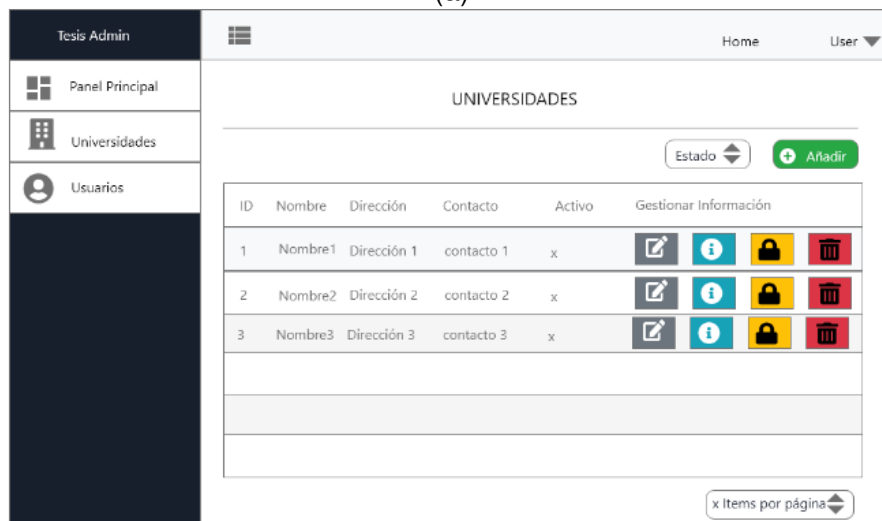
Figura 2.18. Mockups del Módulo Autenticación. (a) Inicio de sesión, (b) Registro de usuario, (c) Recuperación de contraseña, (d) Cambio de contraseña

- **Módulo Gestor**

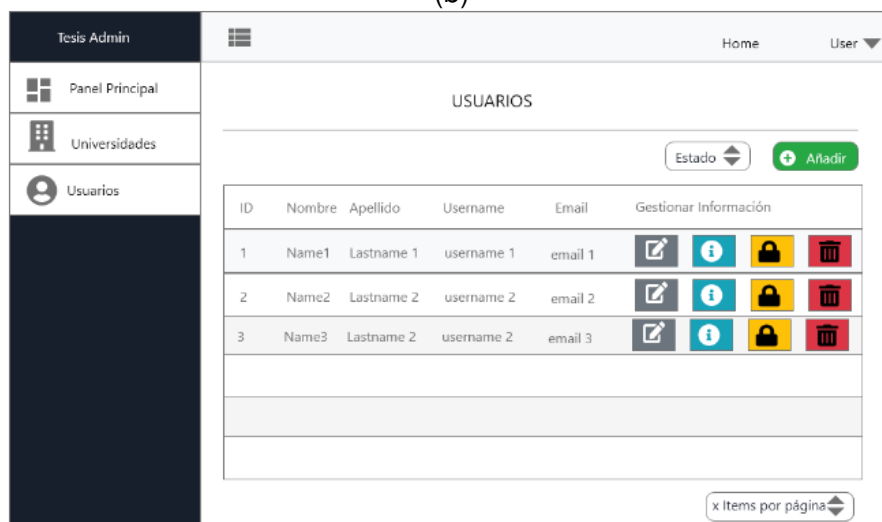
En la Figura 2.19 se presentan los mockups del módulo para el usuario con rol Gestor.



(a)



(b)

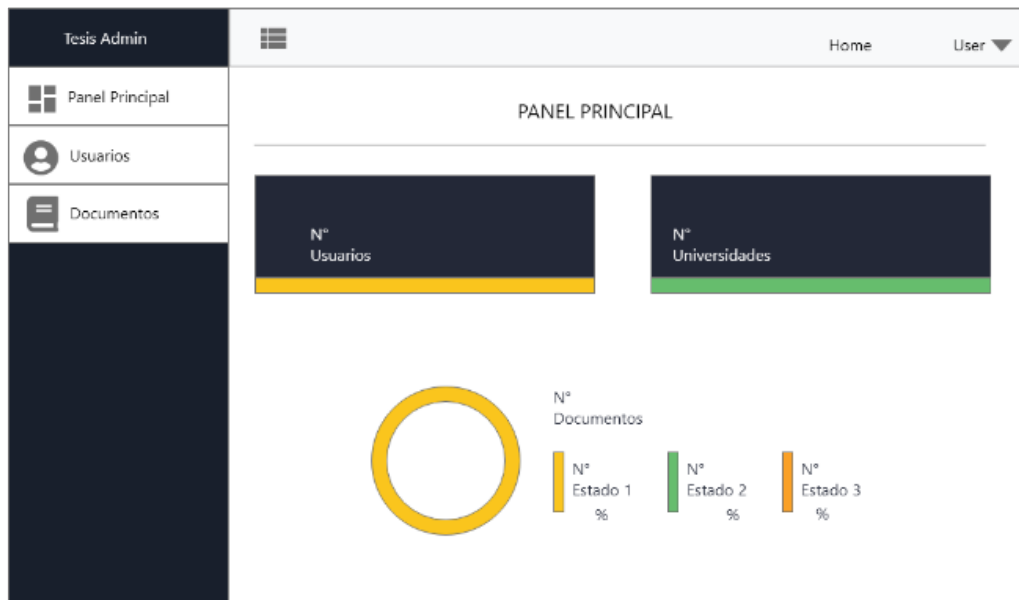


(c)

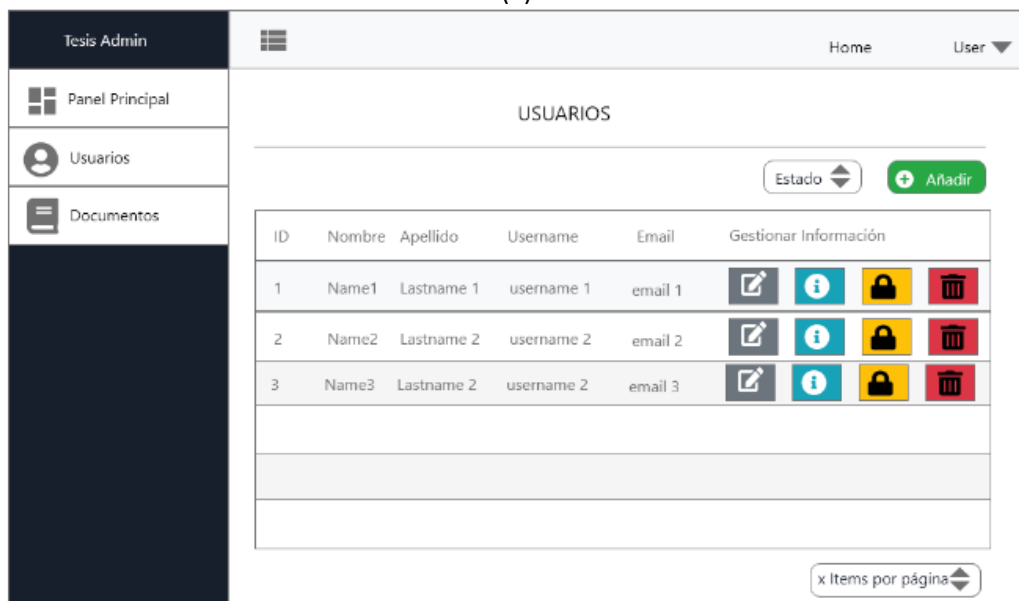
Figura 2.19. Mockups del Módulo Gestor. (a) Vista principal, (b) Universidades, (c) Usuarios

- **Módulo Responsable**

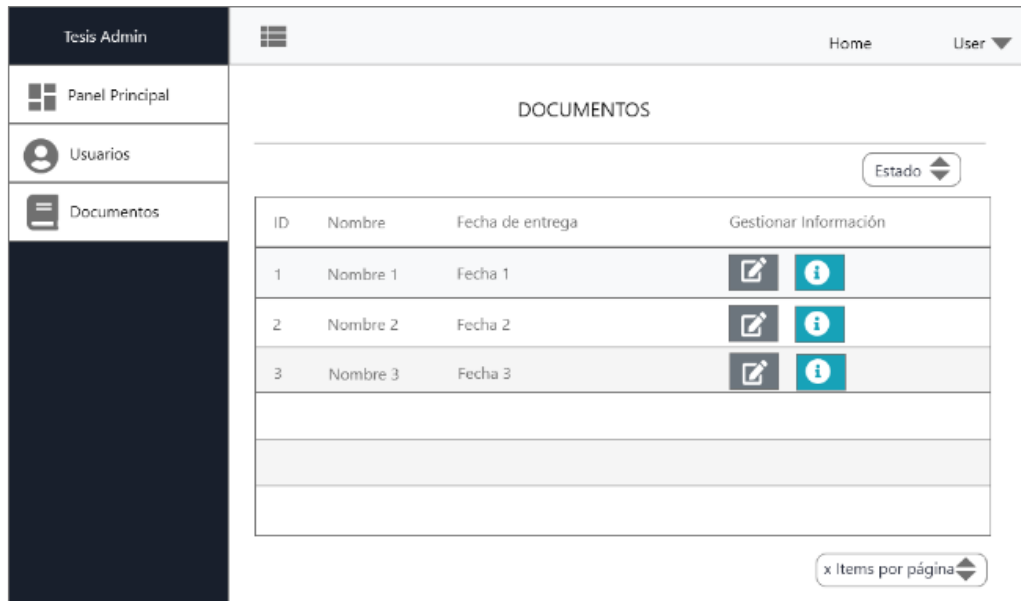
En la Figura 2.20 se presentan los mockups del módulo para el usuario con rol Responsable.



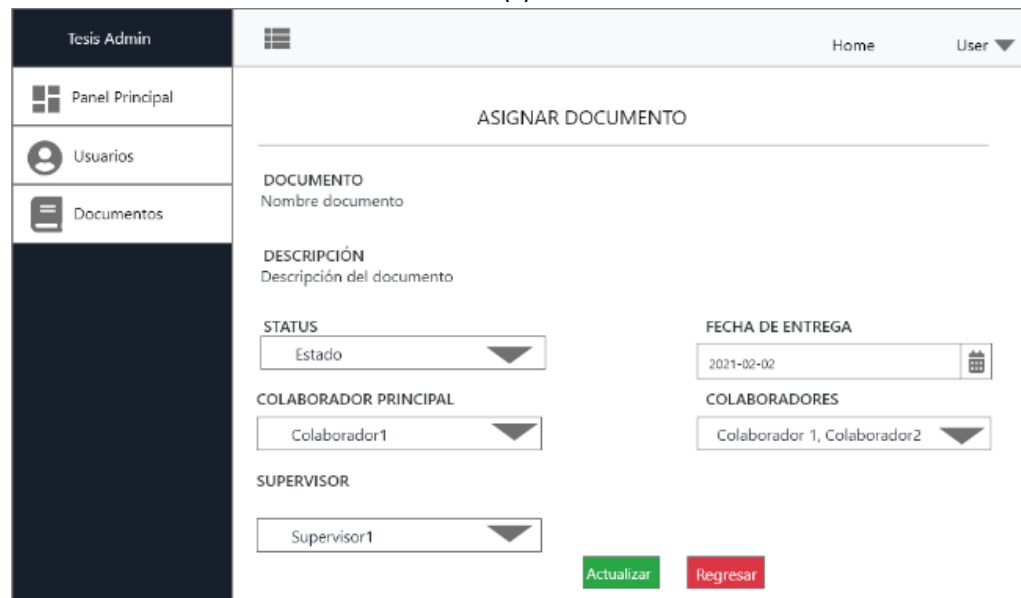
(a)



(b)



(c)



(d)

Figura 2.20. Mockups del Módulo Gestor. (a) Vista principal, (b) Universidades, (c) Usuarios

- **Módulo Colaborador**

En la Figura 2.21 se presenta el mockup del módulo para el usuario con rol Colaborador.

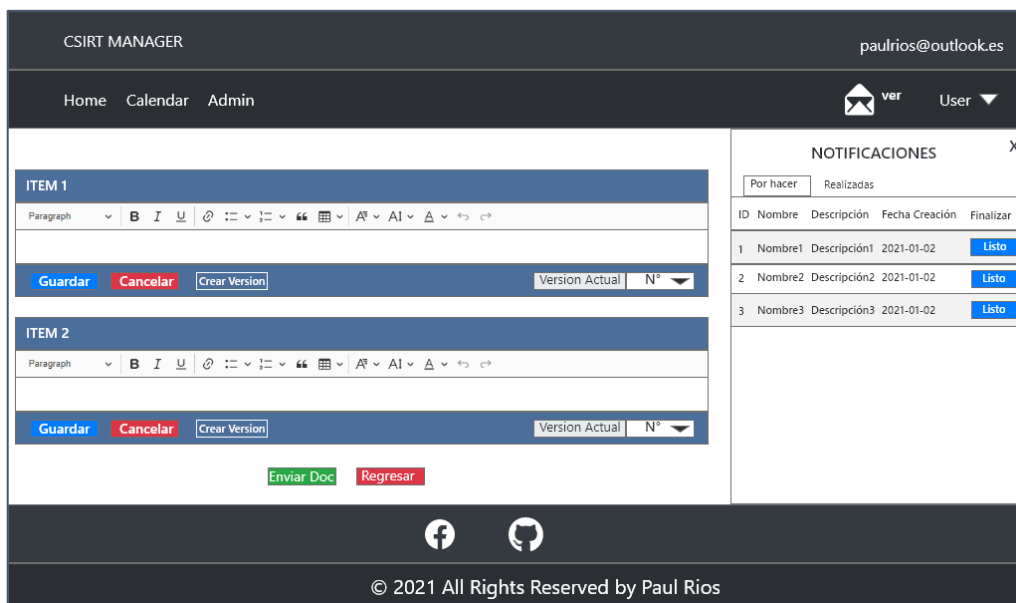


Figura 2.21. Mockup del Módulo Colaborador

- **Módulo Supervisor**

En la Figura 2.22 se presenta el mockup del módulo para el usuario con rol Supervisor.

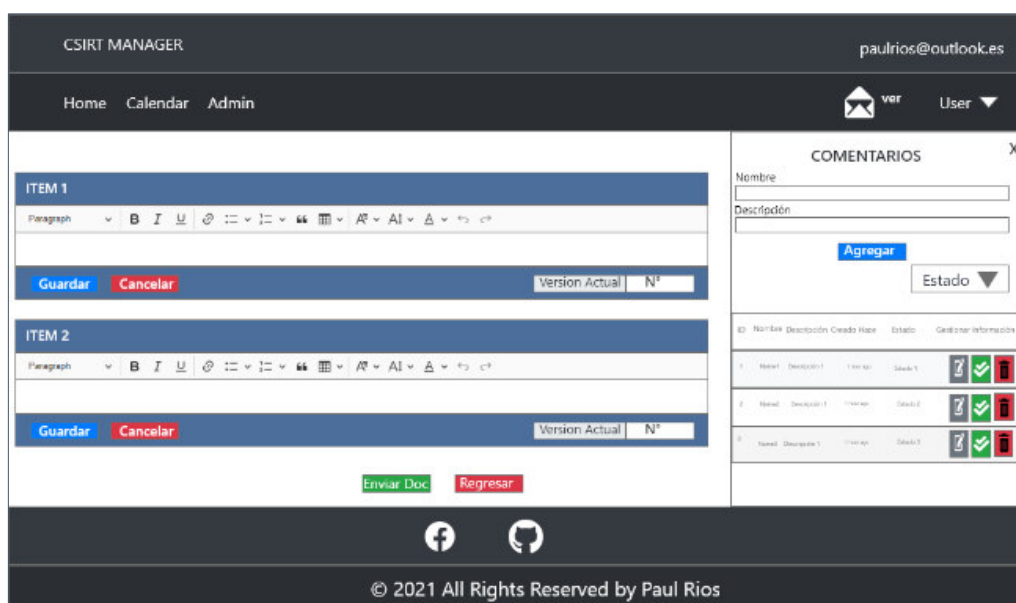


Figura 2.22. Mockup del Módulo Supervisor

- **Módulo de Estado de Tareas**

En la Figura 2.23 se presenta el mockup del módulo de Estado de Tareas.

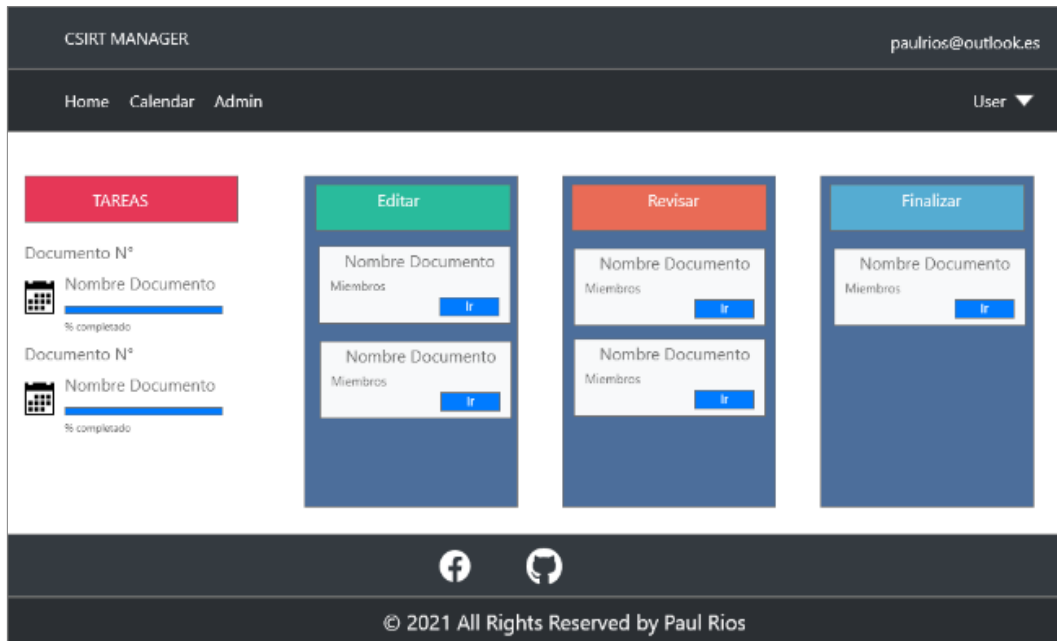


Figura 2.23. Mockup del Módulo Estado de Tareas

- **Módulo Agenda de Actividades**

En la Figura 2.24 se presenta el mockup del módulo de Agenda de actividades.



Figura 2.24. Mockup del Módulo Agenda de Actividades

2.2 IMPLEMENTACIÓN

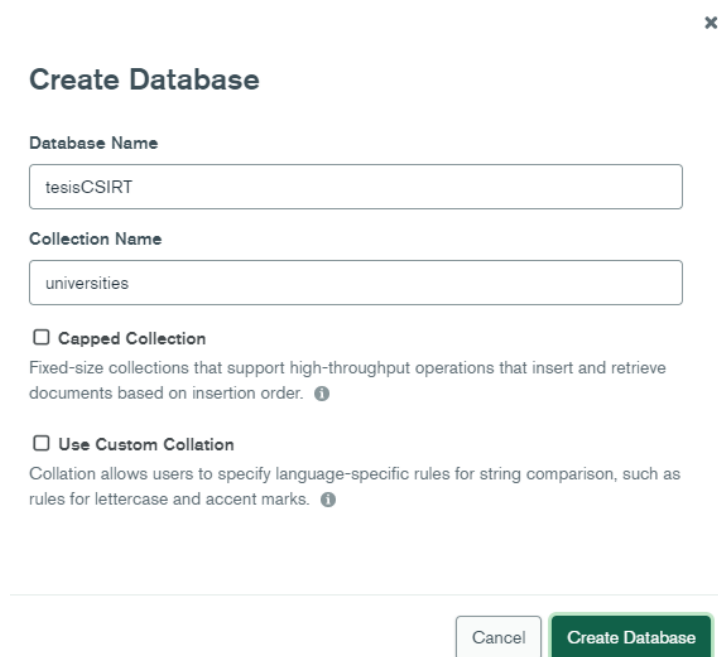
2.2.1 SPRINT 0: CREACIÓN DEL AMBIENTE DE DESARROLLO

En el Sprint 0 se crea la base de datos en MongoDB, además se establecen los repositorios para el frontend y backend con todas las dependencias necesarias para trabajar en el proyecto.

2.2.1.1 BASE DE DATOS EN MONGODB

Para la creación de la base de datos se utilizó MongoDB Compass y el proceso se lo detalla a continuación:

- Primero, se debe dar clic en el botón Create Database ubicado en la parte superior de MongoDB Compass.
- Una vez abierto el cuadro de diálogo, se debe especificar el nombre de la base de datos (TesisCSIRT) y el nombre de la primera colección (universities) como se indica en la Figura 2.25.
- Después, se debe dar clic en Create Database
- Luego, se debe dar clic en Create Colection y se debe ir agregando cada una de las colecciones de la base de datos.



Create Database

Database Name
tesisCSIRT

Collection Name
universities

Capped Collection
Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

Use Custom Collation
Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

Cancel Create Database

Figura 2.25. Creación base de datos en MongoDB Compass

En la Figura 2.26 se muestra la base de datos tesisCSIRT con las 6 colecciones creadas, las cuales servirán para almacenar datos y realizar consultas de la información proporcionada en el sistema web.

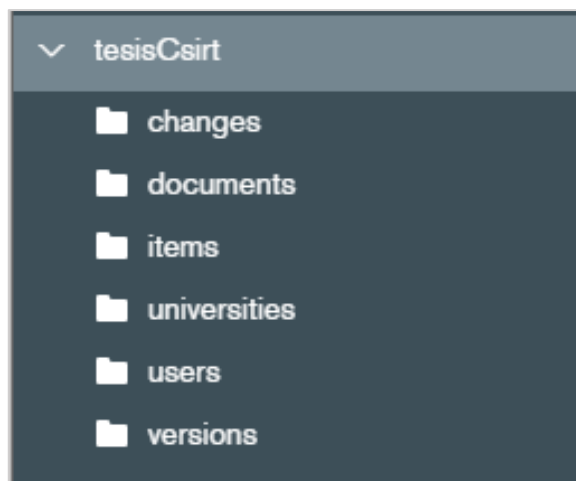


Figura 2.26. Base de datos del proyecto creada

Más adelante, se agregaron datos a las diferentes colecciones. En la Figura 2.27 se muestra un ejemplo de una universidad agregada cuyos campos son name, address, phone, active, id, registerDate. Además, MongoDB establece automáticamente el campo `_id` que permite identificar de forma única a un documento de una colección específica.



Figura 2.27. Añadir una universidad nueva en la base de datos

El archivo JSON con todos los datos agregados en las diferentes colecciones de la base de datos se encuentra en el Anexo C.


2.2.1.2 REPOSITORIOS DE GITHUB

Para el desarrollo del sistema web se crearon 2 repositorios privados en la plataforma Github para la parte del frontend y del backend. En dichos repositorios se almacenó un respaldo del código del proyecto conforme se fueron completando los Sprints. En la Figura 2.28 se muestra el proceso de creación de los repositorios llamados frontend-tesis y backend-tesis.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * Repository name *

 epriosf /

Great repository names Your new repository will be created as frontend-tesis- about [literate-fortnight?](#)

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.


Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾


This will set  main as the default branch. Change the default name in your [settings](#).

(a)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * Repository name *

 epriosf /

Great repository names Your new repository will be created as backend-tesis- about [literate-fortnight?](#)

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.


Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  main as the default branch. Change the default name in your [settings](#).

(b)

Figura 2.28. Repositorios en Github. (a) Repositorio frontend, (b) Repositorio backend

2.2.1.3 RECURSOS NECESARIOS

Antes de iniciar con la creación del proyecto, es necesario la instalación de recursos mínimos que se muestran en la Tabla 2.14.

Tabla 2.14. Recursos necesarios en el proyecto

N°	Recurso	Versión
1	Visual Studio Code	1.62.3
2	Git	2.28.0
3	Node	12.18.0
4	Angular CLI	9.1.8
5	TSLint	6.1.0
6	MongoDB	4.2.7
7	MongoDB Compass	1.29.4

2.2.1.4 REPOSITORIO FRONTEND LOCAL

Una vez instalados los recursos necesarios para el proyecto, procedemos a crear una aplicación en Angular en un repositorio local. Así, nos ubicamos en la interfaz de línea de comandos (CLI) y ejecutamos:

```
> npx new tesis-frontend -routing -style=scss
```

Esto crea un nuevo espacio de trabajo y una aplicación de inicio de Angular con todas las dependencias necesarias como se muestra en la Figura 2.29.

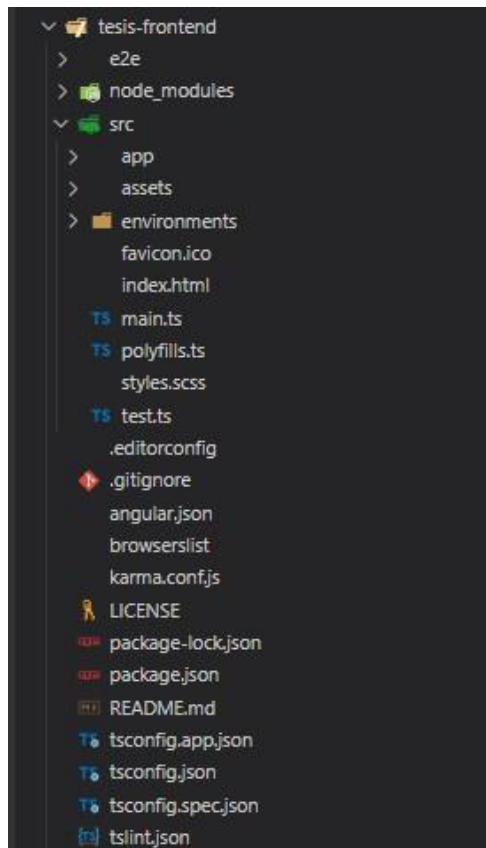


Figura 2.29. Espacio de trabajo de Angular

Luego, añadimos Bootstrap a nuestro proyecto. Para esto, en el archivo styles.scss escribimos el código que se muestra en la Figura 2.30.

```
@import "https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css";
```

Figura 2.30. Añadir librería de Bootstrap

Después se agrega en el archivo index.html los complementos de JS que utilizan algunos de los componentes de Bootstrap para poder funcionar correctamente. Esto se visualiza en la Figura 2.31.

```
<script
  src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
  integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+0GpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
  crossorigin="anonymous"
></script>
<script
  src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
  integrity="sha384-9/reFTGAW83EW2RDu2S0VKAiZap3H66lZ81PoY1FhbGU+6BZp6G7niu735Sk7lN"
  crossorigin="anonymous"
></script>
<script
  src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
  integrity="sha384-B4gt1jrGC7Jh4AgTPSdUt0Bvf08shuf57BaghqFfPLYxofvL8/KUEfYiJOMMV+rV"
  crossorigin="anonymous"
></script>
```

Figura 2.31. Complementos de JS para Bootstrap

Finalmente, para iniciar el servidor de Angular de manera local en el puerto 4200 ejecutamos en el CLI el comando:

```
> npm start
```

2.2.1.5 REPOSITORIO BACKEND LOCAL

Una vez creado el espacio de trabajo para el frontend, procedemos a crear y configurar el repositorio para el backend de manera local. Así, nos ubicamos en la interfaz de línea de comandos (CLI) y ejecutamos el comando:

```
> git clone git@github.com:epriosf/backend-tesis.git
```

Esto clona el repositorio para el backend creado en Github y lo añade a nuestro repositorio local. Después, procedemos a crear el fichero package.json con el comando:

```
> npm init
```

Así, especificamos el nombre del paquete, la versión, la descripción, el punto de entrada para inicializar la aplicación, palabras claves, el autor y el tipo de licencia como se muestra en la Figura 2.32.

```

C:\Users\HP\Desktop\Materias\Inicio Tesis\Plan De Tesis\backend-tesis>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (backend-tesis)
version: (1.0.0) 0.0.1
description: Proyecto backend API GraphQL Sistema CSIRT- MEAN+G
entry point: (index.js) build/server.js
test command:
git repository: (https://github.com/epriosf/backend-tesis.git)
keywords: grapghql, meang, api-graphql
author: Paul Rios <paulrios@outlook.es>
license: (ISC) MIT

```

Figura 2.32. Fichero package.json

- **Versiones mínimas Node y npm**

Ahora, especificamos la versiones mínimas para node y npm con las que se ejecutará el proyecto en el archivo package.json como se muestra en la Figura 2.33. La mínima versión de Node se establece en la 12.0.0 y la de npm en 6.0.0 ya que las dependencias que se instalarán más adelante en el backend son compatibles con estas versiones mínimas y con versiones más actuales. En el caso de que se utilicen versiones menores a estas, no se podrá ejecutar el proyecto de manera correcta.

```

"engines": {
  "node": ">=12.0.0",
  "npm": ">=6.0.0"
},

```

Figura 2.33. Versiones mínimas de Node y npm

- **Configuración archivo tsconfig.json**

A continuación, instalamos Typescript y creamos el fichero de configuración tsconfig.json con los comandos:

```
> npm install typescript
```

```
> npx tsc --init --rootDir src --outDir build --lib dom,es6 --
module commonjs --target es6 --removeComments
```

- **Dependencias del proyecto**

Seguidamente, instalamos las dependencias de producción y desarrollo en nuestro proyecto con los comandos:

```
> npm install express graphql ncp chalk graphql-import-node
compression cors graphql-tools graphql-playground-middeware-
express apollo-server-express
```

```
> npm install -D @types/cors @types/compression @types/express
@types/node @types/graphql nodemon ts- node
```

- **Definición de scripts**

Más tarde, se definen los scripts en el package.json: “start”, en la línea 7, para inicializar el proyecto con los ficheros compilados; “build”, en la línea 8, para compilar el proyecto ; y “dev”, en la línea 9, para ejecutar el servidor en ambiente de desarrollo como se muestra en la Figura 2.34.

```
6  "scripts": {
7    "start": "node build/server.js",
8    "build": "tsc -p .",
9    "dev": "nodemon \"src/server.ts\" --exec \"ts-node\" \"src/server.ts\" -e ts,graphql,json"
10 },
```

Figura 2.34. Definición de scripts

- **Ejecución inicial del servidor**

Finalmente, creamos una carpeta src y dentro de esta carpeta establecemos un archivo denominado server.ts. Para correr el servidor en desarrollo ejecutamos el comando:

```
> npm run dev
```

- **Variables de entorno**

Se procede a crear un archivo .dotenv que contendrá las variables de entorno a usar. Para esto, es necesario instalar el paquete dotenv mediante el comando:

```
> npm install dotenv
```

Además, creamos la carpeta config, la cual contiene los ficheros para establecer las constantes y configurar las variables de entorno que se usarán en la parte del backend como se muestra en la Figura 2.35.

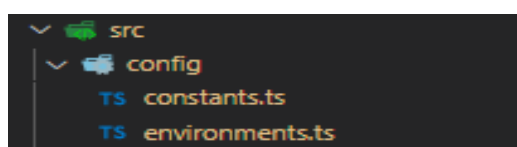


Figura 2.35. Variables de entorno

- **Configuración TSLINT**

TSLINT es una herramienta que permite detectar errores en código Typescript y corregirlos a través de reglas específicas. Para poder hacer uso de este formateador, instalamos el plugin de Typescript para TSLINT en las dependencias de desarrollo mediante el comando:

```
> npm install -D typescript-eslint-plugin
```

Para la escritura del código del backend se establecieron 5 reglas de TSLINT, las cuales se definen en la Tabla 2.15.

Tabla 2.15. Reglas TSLINT

N°	REGLA	SIGNIFICADO
1	"quotemark": [true, "single"]	Uso de comillas simples en el código.
2	"semicolon": true	Uso obligatorio de punto y coma al final de cada declaración.
3	"no-any": true	No se permite el uso de any para especificar algún tipo de dato.
4	"curly": true	Uso obligatorio de corchetes en estructuras condicionales como if-else, for, do y while.
5	"no-empty-interface": true	No se permite tener interfaces vacías sin ninguna línea de código.

- **Conexión a la base de datos MongoDB**

La conexión del backend a la base de datos necesita la instalación del controlador de Node para MongoDB. Este controlador contiene todas las dependencias de producción y desarrollo necesarias para poder usar ciertas definiciones en nuestro proyecto de Typescript. Los comandos a usar son:

```
> npm install mongodb@^3  
  
> npm install -D @types/mongodb@^3
```

Para establecer la conexión a la base de datos, se creó una carpeta lib que contiene un archivo database.ts. Aquí se tiene una clase llamada Database que contiene una función para inicializar la base de datos como se muestra en la Figura 2.36.


```

5  async init() {
6    const MONGO_DB =
7      process.env.DATABASE || 'mongodb://localhost:27017/tesisCsirt';
8    // Configurar la conexión constante del cliente
9    const client = await MongoClient.connect(MONGO_DB, {
10     useNewUrlParser: true,
11     useUnifiedTopology: true,
12   });
13
14   const db = client.db();
15   // Verificar si está conectado a la DB
16   if (client.isConnected()) {
17     console.log('=====DATABASE=====');
18     console.log(`STATUS: ${chalk.greenBright('ONLINE')}`);
19     console.log(`DATABASE: ${chalk.greenBright(db.databaseName)}`);
20     console.log('=====');
21   }
22   return db;
23 }

```

Figura 2.36. Clase que inicializa la base de datos

- **Creación del servidor Node con Express**

Una vez realizada la configuración de la base de datos, procedemos a montar el servidor de Express, inicializar la configuración de la base de datos y a configurar Apollo Server Express con los ajustes básicos como la activación de las peticiones de origen cruzado para poder realizar peticiones entre el frontend y el backend definido en la línea 26. Además, en la línea 28 se realiza la activación de compression para tener el proyecto lo más comprimido posible. En la Figura 2.37 se muestra la función para montar el servidor.

```

22  async function init() {
23    // Inicialización de Express
24    const app = express();
25    // Activar las peticiones de origen cruzado
26    app.use('*', cors());
27    // Activar compression para tener la app lo más comprimida posible
28    app.use(compression());
29
30    // Inicializar la clase de database
31    const database = new Database();
32
33    // Inicializar la configuración de la base de datos
34    const db = await database.init();
35
36    // Creación de un objeto Context
37    const context = async ({ req, connection }: IContext) => {
38      //Verificamos que el token sea para conexión(peticion request) o para conexión(WebSockets)
39      const token = req ? req.headers.authorization : connection.authorization;
40      return { db, token };
41    };
42
43    // Configurar el servidor de Apollo Server y añadir el contexto
44    const server = new ApolloServer({
45      schema,
46      introspection: true, /*para poder ver el schema en el playground*/
47      context,
48    });
49    //Middleware de la aplicación dentro del servidor
50    server.applyMiddleware({ app });
51    // Responder solo a las solicitudes http con la URL especificada
52    app.get(
53      '/',
54      expressPlayground({
55        endpoint: '/graphql',
56      })
57    );
58
59    const httpServer = createServer(app);
60    const PORT = process.env.PORT || 2002;
61    // Escuchar el servidor
62    httpServer.listen(
63      {
64        port: PORT,
65      },
66      () => {
67        console.log('=====SERVER API GRAPHQL');
68        console.log(`STATUS: ${chalk.greenBright('ONLINE')}`);
69        console.log(`MESSAGE: ${chalk.greenBright('API MEANG - Tesis CSIRT START')}`);
70        console.log(`http://localhost:${PORT}`);
71      }
72    );
73 }

```

Figura 2.37. Función para montar el servidor

2.2.2 SPRINT 1: AUTENTICACIÓN

El Sprint 1 (Autenticación) establece las funcionalidades para el inicio de sesión de un usuario en el sistema mediante JWT, registrar un usuario, hacer la petición de cambio de contraseña y realizar el cambio de esta.

2.2.2.1 CODIFICACIÓN BACKEND

El diseño de la API GraphQL se encuentra modularizado, es decir, se separaron en varias carpetas los diferentes archivos para tener un código más limpio como se muestra en la **Figura 2.38**. Además, se instalaron los paquetes necesarios que facilitan la modularización del esquema GraphQL mediante el comando:

```
> npm install @graphql-tools/load-files @graphql-tools/merge  
graphql-merge-resolvers
```

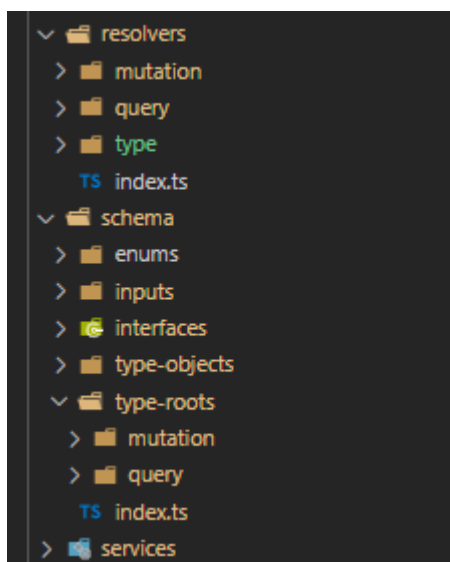


Figura 2.38. Estructura esquema GraphQL

- **Operaciones Base de Datos**

Dentro la carpeta lib, procedemos a crear un archivo db-operations.ts que contiene las operaciones CRUD que se ejecutarán en la base de datos como se muestran en la Figura 2.39.

La función findOneElement() sirve para encontrar un documento en la base de datos. Esta función recibe 3 parámetros: El nombre de la base de datos, la colección y un filtro que define diferentes criterios de búsqueda que se los especificarán más adelante. Esta función hace uso del método findOne() de MongoDB que devuelve el primer documento

que cumpla con los criterios de búsqueda especificados o retorna null en el caso de que ningún documento cumpla con dichos criterios.

La función `insertOneElement()` sirve para insertar un documento en la base de datos. Esta función recibe los mismos parámetros que `findOneElement()` y hace uso del método `insertOne()` de MongoDB que inserta un documento en la colección y devuelve `true` y el `id` del documento insertado si la operación fue exitosa o retorna `false` en el caso de que la operación no haya tenido éxito.

La función `updateOneElement()` sirve para actualizar un documento en la base de datos. Esta función recibe los mismos parámetros que `findOneElement()` y adiciona un objeto que contendrá los valores que se quieren actualizar en el documento. Esta función hace uso del método `updateOne()` de MongoDB que actualiza un elemento y devuelve un documento que contiene: Número de documentos que coincidieron con la búsqueda, cantidad de documentos modificados, un `id` en el caso de que un documento nuevo se haya insertado y un boolean en `true` si la operación de actualización es exitosa o `false` si hubo problemas.

La función `deleteOneElement()` recibe los mismos parámetros que `findOneElement()` y hace uso del método `deleteOne()` de MongoDB que elimina un documento de la colección y retorna un documento que contiene: La cantidad de elementos eliminados y un boolean en `true` si la operación fue exitosa o en `false` en el caso de que la operación no haya tenido éxito.

```
42 export const findOneElement = async (
43   database: Db,
44   collection: string,
45   filter: object
46 ) => {
47   return database.collection(collection).findOne(filter);
48 };
```

(a)

```
50 export const insertOneElement = async (
51   database: Db,
52   collection: string,
53   document: object
54 ) => {
55   return await database.collection(collection).insertOne(document);
56 };
```

(b)

```
65 export const updateOneElement = async (
66   database: Db,
67   collection: string,
68   filter: object,
69   updateObject: object
70 ) => {
71   return await database
72     .collection(collection)
73     .updateOne(filter, { $set: updateObject });
74 };
```

(c)

```
105 export const deleteOneElement = async (
106   database: Db,
107   collection: string,
108   filter: object = {}
109 ) => {
110   return await database.collection(collection).deleteOne(filter);
111 };
```

(d)

Figura 2.39. Funciones de operaciones CRUD. (a) Función `findOneElement`, (b) Función `insertOneElement`, (c) Función `updateOneElement`, (d) Función `deleteOneElement`

Cabe mencionar que además de las funciones especificadas, se estableció una función adicional para asignar correctamente el id de un documento de acuerdo con el último elemento que se ingresó en la colección. En el caso de que no haya documentos en la colección, se establece el id en 1 (línea 21). Por el contrario, si existen documentos almacenados, se obtiene el id del último documento ingresado de acuerdo con la fecha de registro y se aumenta el valor del id en 1 (línea 23). La Figura 2.40 muestra el código de esta función.

```
 9  export const asignDocumentId = async (
10    database: Db,
11    collection: string,
12    sort: object = { registerDate: -1 }
13  ) => {
14    const lastElement = await database
15      .collection(collection)
16      .find()
17      .limit(1)
18      .sort(sort)
19      .toArray();
20    if (lastElement.length === 0) {
21      return 1;
22    }
23    return String(+lastElement[0].id + 1);
24  };
```

Figura 2.40. Función para asignar Id

- **Login de un usuario**

Para realizar el login de un usuario específico se realizó la instalación de las dependencias de producción y desarrollo de bcrypt y JWT mediante los comandos:

```
> npm install jsonwebtoken bcrypt
```

```
> npm install @types/jsonwebtoken @types/bcrypt -D
```

Estas librerías permiten establecer las funciones para cifrar la contraseña de un usuario antes de almacenarla en la base de datos, firmar el token con JWT y poder verificarlo en el caso de que se requiera autenticar un usuario específico. Para firmar el token se usa la función sign() y se envía la información de un usuario, la clave secreta y el tiempo de expiración del token establecido en 24h. Para verificar el token, se utiliza la función verify(), la cual recibe el token y la clave secreta como se muestran en la Figura 2.41.

```

8  sign(data: IJwt, expiresIn:number=EXPIRETIME.H24) {
9    return jwt.sign(
10     { user: data.user },
11     this.secretKey,
12     { expiresIn }
13   );
14 }

```

(a)

```

16 verify(token: string) {
17   try {
18     return jwt.verify(token, this.secretKey);
19   } catch (error) {
20     return MESSAGES.TOKEN_VERIFICATION_FAILED;
21   }
22 }

```

(b)

Figura 2.41. Funciones para JWT. (a) Función para firmar un token, (b) Función para verificar un token

Queries

Las definiciones del tipo de raíz query-login y query-me se las definen en la Figura 2.42. El query-login admite como parámetros el email de un usuario (línea 6) y su contraseña (línea 8), dando como resultado el type ResultLogin (línea 9) que contiene el estado de la operación, un mensaje de feedback, el token del usuario creado y la información de dicho usuario.

```

1  "Definiciones de los tipos de raíz"
2  type Query {
3    "Para iniciar sesion de usuario"
4    login(
5      "Correo electrónico del usuario con el que se podrá loguear al sistema"
6      email: String!
7      "Contraseña para autenticación"
8      password: String!
9    ): ResultLogin
10   "Autenticación mediante el Token"
11   me: ResultUser
12 }
13

```

Figura 2.42. Queries login y me

Por otro lado, el query-me no recibe parámetros de entrada y devuelve el type ResultUser (línea 11) que contiene el estado de la operación, un mensaje de feedback y la información del usuario.

Resolvers

En la Figura 2.43 se muestran los resolvers para query-login y query-me especificados anteriormente. El resolver-login hace uso del método login() del servicio userService (línea 27), el cual, busca a un usuario en específico en la base de datos por medio del correo electrónico ingresado y en caso de encontrarlo, verifica que dicho usuario se encuentre activo. Además, se compara la contraseña ingresada con la que se encuentra cifrada en la base de datos mediante la librería bcrypt y en caso de ser iguales crea un

token para el usuario con una duración de 24h. Caso contrario, se envían mensajes de feedback indicando que el usuario no es válido, se encuentra pendiente su activación o email y contraseña no son correctos.

```
25 //Función para loguearse al sistema
26 async login(_, { email, password },context) {
27   return new UsersService(_, {user: {email, password}},context).login();
28 },
29
30 //Función de autenticación al sistema
31 me(_, __, {token}){
32   return new UsersService(_, __, {token}).auth();
33 },
```

Figura 2.43. Resolvers login y me

Por otro lado, el resolver-me hace uso del método auth() del servicio usersService (línea 32), el cual, verifica la caducidad del token y retorna el usuario en caso de que la operación sea exitosa o devuelve un usuario nulo en el caso de existir problemas.

- **Registro de un Usuario**

Para realizar el registro de un nuevo usuario en la base de datos, se estableció una operación mutation con su respectivo resolver y se los indican a continuación:

Mutation

La definición del tipo de raíz mutation-Register que se lo define en la Figura 2.44, admite como parámetros de entrada un usuario y da como resultado un tipo ResultUser (línea 3).

```
2 "Para registrar un usuario"
3   register(user: UserInput!): ResultUser
```

Figura 2.44. Mutation register

Resolver

En la Figura 2.45 se muestra el resolver para el mutation-register que permite a un usuario registrarse al sistema.

```
5 //Función para registrar usuario nuevo
6 register(_, { user }, context) {
7   return new UsersService(_, { user }, context).register();
8 },
```

Figura 2.45. Resolver register

Este resolver hace uso del método `register()` del servicio `userService` (línea 7). Este método verifica que el usuario no sea nulo e identifica que la contraseña no sea indefinida, vacía o nula. Una vez realizada dicha comprobación, se revisa que el email del usuario a registrar no se encuentre en la base de datos y en caso de no existir, se establece el ID del nuevo usuario y la fecha de registro. También se cifra la contraseña por medio de la librería `bcrypt` y se añade el usuario a la base de datos. En el caso de que la operación no sea exitosa se envían mensajes de feedback indicando que no se definió un usuario, la contraseña no es correcta o no se puede registrar el usuario con el email indicado.

- **Cambio de contraseña**

Para realizar el cambio de contraseña de un usuario específico, se realizó la instalación de las dependencias de producción y desarrollo de la librería `nodemailer` mediante los comandos:

```
> npm install nodemailer
> npm install @types/nodemailer -D
```

Esta librería permite enviar correos electrónicos de manera automática de una manera sencilla. A continuación, se definen las operaciones de mutation creadas con sus respectivos resolvers:

Mutations

La definición del tipo de raíz `mutation-email` define 2 operaciones que se las muestran en la Figura 2.46. La operación `resetPassword()` sirve para enviar el correo de petición para el cambio de contraseña, la cual, necesita como parámetro un email y retorna como resultado el type `ResultMail` (línea 3) que contiene el estado de la operación, un mensaje de feedback y el email con la información del emisor, receptor, asunto y contenido. Por otro lado, la operación `changePassword()` recibe el ID del usuario y la nueva contraseña retornando el type `ResultUser` (línea 5).

```
1 type Mutation {
2   "Resetear la contraseña"
3   resetPassword(email: String!): ResultMail
4   "Cambiar la contraseña"
5   changePassword(id: ID!, password: String!): ResultUser
6 }
```

Figura 2.46. Mutations `resetPassword` y `changePassword`

Resolvers

En la Figura 2.47 se muestran los resolvers para el mutation-resetPassword y mutation-changePassword indicados previamente. El resolver-resetPassword hace uso del método setEmail() del servicio passwordService (línea 16), el cual, verifica que el email ingresado exista en la base de datos, luego obtiene la información del usuario, después crea un usuario temporal que tiene un id y un email, más adelante crea un token con un tiempo de expiración de 15 minutos y agrega el contenido del correo en formato HTML. Finalmente, se llama al método send() del servicio MailService y envía el correo de restablecimiento de la contraseña y en caso de que las operaciones no sean exitosas devuelve un feedback de mensajes personalizados de error al usuario.

```
14 // Acción para resetear al password
15 async resetPassword(_, { email }, { db }) {
16   return new PasswordService(_, {user: {email}}, {db}).sendEmail();
17 },
18
19 //Acción para cambiar la contraseña
20
21 async changePassword(_, { id, password }, { db, token }) {
22   //1.- Verificar el token
23   const verify = verifyToken(token, id);
24   if (verify?.status === false) {
25     return {
26       status: false,
27       message: verify.message,
28     };
29   }
30   //2.- Llamada al servicio
31   return new PasswordService(_, {user:{id, password}}, {db}).change();
32 },
```

Figura 2.47. Resolvers resetPassword y changePassword

El resolver-changePassword hace uso del método change() del servicio passwordService (línea 31), el cual, verifica que el id del usuario y password sean correctos. Después, realiza el cifrado de la contraseña y actualiza dicho valor en la base de datos. En caso de que la operación de actualización no sea existosa devuelve un feedback de mensajes personalizados de error al usuario.

2.2.2.2 CODIFICACIÓN FRONTEND

Para el diseño de la interfaz gráfica del módulo de autenticación se deben instalar los siguientes paquetes necesarios:

```
> npm install --save apollo-angular@1.9.0 apollo-angular-link-
http@1.10.0 apollo-link@1.2.14 apollo-client@2.6.8 apollo-cache-
inmemory@1.6.5 graphql-tag@2.10.3 graphql@15.3.0
```


- **Configuración Módulo GraphQL**

Para establecer la comunicación entre la aplicación de Angular y la API GraphQL y, además poder capturar errores de red o errores al realizar consultas a la base de datos, se debe configurar el constructor de la clase GraphQLModule como se lo indica en la Figura 2.48.

```
13 constructor(apollo: Apollo, httpLink: HttpLink) {
14   /*Para capturar los errores de consultas y de red*/
15   const errorLink = onError(({ graphQLErrors, networkError }) => {
16     if (graphQLErrors) {
17       console.log('GraphQL Errors:', graphQLErrors);
18     }
19     if (networkError) {
20       console.log('Networks Errors:', networkError);
21     }
22   });
23   /*Constante uri que servirá para la creación del HTTPLINK
24   para linkeo entre captura de errores y la llamada a GRAPHQL*/
25   const uri = 'http://localhost:2002/graphql';
26   const link = ApolloLink.from([errorLink, httpLink.create({ uri })]);
27   apollo.create({
28     link,
29     cache: new InMemoryCache(),
30   });
31 }
```

Figura 2.48. Configuración Módulo GraphQL

- **Login de un usuario**

En la Figura 2.49 se muestra parte del código del servicio login() creado para consumir de la API de GraphQL, establecida en el backend, la operación de inicio de sesión.

```
55 login(email: string, password: string) {
56   return this.get(LOGIN_QUERY, { email, password, include: false}).pipe(
57     map((result: any) => {
58       return result.login;
59     })
60   );
61 }
```

Figura 2.49. Servicio login

- **Registro de un Usuario**

En la Figura 2.50 se muestra parte del código del servicio register() realizado para consumir de la API de GraphQL, la operación para registrar un nuevo usuario, la cual, fue creada en el backend.

```

93 register(user: IRegisterForm) {
94     return this.set(REGISTER_USER,
95         {
96             user,
97             include: false
98         })
99     .pipe(
100     map((result: any) => {
101         return result.register;
102     })
103     );
104 }

```

Figura 2.50. Servicio register

- **Cambio de contraseña**

En la Figura 3.51 se muestra parte del código de los servicios reset() y change(). Estos servicios son creados para consumir de la API de GraphQL las operaciones para enviar un correo (línea 19) y poder cambiar la contraseña (línea 30).

```

18 reset(email: string) {
19     return this.set(RESET_PASSWORD, {
20         email,
21     }).pipe(
22     map((result: any) => {
23         return result.resetPassword;
24     })
25     );
26 }
27
28 change(token: string, password: string) {
29     const user = JSON.parse(atob(token.split('.')[1])).user;
30     return this.set(
31         CHANGE_PASSWORD,
32         {
33             id: user.id,
34             password,
35         },
36         {
37             headers: new HttpHeaders({
38                 Authorization: token,
39             }),
40         }
41     ).pipe(
42     map((result: any) => {
43         return result.changePassword;
44     })
45     );
46 }

```

Figura 2.51. Servicios para cambiar contraseña

2.2.3 SPRINT 2: GESTIÓN GESTOR

El Sprint 2 (Gestión Gestor) establece la codificación para el panel del usuario con rol Gestor. En este Sprint se realizan las funcionalidades para las operaciones CRUD de

usuarios y universidades. Además, se visualiza la cantidad de universidades, usuarios y documentos que hay en el sistema.

2.2.3.1 CODIFICACIÓN BACKEND

Para la gestión del usuario Gestor, se establecieron los queries, mutations y resolvers para realizar operaciones CRUD de universidades y de usuarios con rol Responsable.

- **Queries**

En la Figura 2.52, se visualizan los queries `universities()` y `university()` necesarios para obtener una lista de universidades aplicando un sistema de paginación y para conseguir información de una universidad específica por medio de su ID.

```
1 type Query {
2   "Lista de Universidades registradas en la base de datos"
3   universities(
4     "Pagina seleccionada para acceder a ella"
5     page: Int = 1
6     "Universidades que se obtendrán por pagina"
7     itemsPage: Int = 20
8     "Especificar el estado de la universidad que queremos mostrar"
9     active: ActiveFilterEnum= ACTIVE
10  ): ResultUniversities
11  "item de la universidad seleccionada"
12  university(id: ID!): ResultUniversity
13 }
```

Figura 2.52. Queries de universidades

- **Mutations**

Los mutations para registrar (línea 3), actualizar (línea 5), eliminar (línea 7) y bloquear (línea 8) una universidad específica se los muestran en la Figura 2.53.

```
1 type Mutation {
2   "Añadir una Universidad"
3   registerUniversity(university: UniversityInput!): ResultUniversity
4   "Actualizar una Universidad"
5   updateUniversity(university: UniversityInput!): ResultUniversity
6   "Para Eliminar una universidad"
7   deleteUniversity(id: ID!): ResultUniversity
8   "Bloquear/Desbloquear Universidad. Por defecto unblock"
9   blockUniversity(id: ID!, unblock: Boolean= false,
10  "Se especifica si la procedencia de la accion llega desde el gestor"
11  admin: Boolean= false): ResultUniversity
12 }
```

Figura 2.53. Mutations de universidades

Todos los mutations especificados dan como resultado un type ResultUniversity que devuelve un estado de la operación, un mensaje de feedback y la información de la universidad a la cual, se le realizó alguna operación CRUD.

- **Resolvers**

Parte del código para los resolvers de los queries se muestran en la Figura 2.54. El resolver query-universities hace uso del método items del servicio UniversitiesService (línea 9) mientras que el resolver query-university utiliza el método details del mismo servicio (línea 13).

```
4 Query:{
5     /*Función para obtener todas las universidades de la base de datos*/
6     async universities(_, {page, itemsPage, active}, context) {
7         return new UniversitiesService(_, {
8             pagination: {page, itemsPage}
9         }, context).items(active);
10    },
11    /*Función para obtener una universidad especifica mediante id*/
12    async university(_, {id}, {db}){
13        return new UniversitiesService(_, {id}, {db}).details();
14    }
15 }
```

Figura 2.54. Resolvers de queries de universidades

Por otro lado, los resolvers para los mutations se especifican en la Figura 2.55. Todos los resolvers hacen uso de los métodos establecidos en el servicio UniversitiesService para realizar las operaciones CRUD en la base de datos.

```
4 Mutation: {
5     //Funcion para registrar una universidad nueva
6     registerUniversity(_, { university }, context) {
7         return new UniversitiesService(_, { university }, context).registerUniversity();
8     },
9     //Funcion para actualizar una universidad nueva
10    updateUniversity(_, { university }, context) {
11        return new UniversitiesService(_, { university }, context).modify();
12    },
13    //Funcion para eliminar una universidad
14    deleteUniversity(_, { id }, context) {
15        return new UniversitiesService(_, { id }, context).delete();
16    },
17    //Funcion para bloquear una universidad
18    blockUniversity(_, {id, unblock, admin}, context){
19        //ANADIMOS la llamada al servicio
20        return new UniversitiesService(_, {id}, context).unblock(unblock, admin);
21    }
22 }
```

Figura 2.55. Resolvers de mutations de universidades

Cabe mencionar que las demás operaciones para la gestión de usuarios se las pueden verificar en el Anexo D.

2.2.3.2 CODIFICACIÓN FRONTEND

En el diseño de la interfaz gráfica del módulo de gestión del usuario gestor, es necesario la instalación del siguiente paquete de npm:

```
> npm install @swimlane/ngx-charts@14.0.0 -save
```

Esta librería permite visualizar datos por medio de esquemas personalizados como gráficos de barras, circulares y tablas numéricas. El panel del gestor consta de una tabla numérica en la que se especifica la cantidad de usuarios, universidades y documentos que hay en el sistema web. En la Figura 2.56, se muestra parte del código html del componente dashboard-info que se lo creó para poder reutilizarlo en el módulo para el usuario Responsable.

```
1 <div class="container-fluid">
2   <div class="d-flex justify-content-center">
3     <ngx-charts-number-card
4       scheme="picnic"
5       [results]="items"
6       [cardColor]="cardColor"
7     >
8   </ngx-charts-number-card>
9 </div>
10 </div>
```

Figura 2.56. Código archivo dashboard-info.component.html

- **CRUD Universidades**

Parte del código para establecer las operaciones CRUD de las universidades que se consumirán de la API GraphQL se especifican en la Figura 2.57.

```
23 /*Metodo para registrar una universidad*/
24 register(university: IUniversityRegisterForm) {
25   return this.universitiesService.register(university);
26 }
27 /*Metodo para actualizar una universidad*/
28 update(university: IUniversityRegisterForm) {
29   return this.set(UPDATE_UNIVERSITY, {
30     university,
31     include: false,
32   }).pipe(
33     map((result: any) => {
34       return result.updateUniversity;
35     })
36   );
37 }
38 /*Metodo para bloquear/desbloquear una universidad*/
39 unblock(id: string, unblock: boolean = false, admin: boolean = false) {
40   return this.set(BLOCK_UNIVERSITY, {
41     id,
42     unblock,
43     admin,
44   }).pipe(
45     map((result: any) => {
46       return result.blockUniversity;
47     })
48   );
49 }
```

Figura 2.57. Código servicios para universidades

Las demás operaciones para la gestión de usuarios se las pueden verificar en el Anexo E.

2.2.4 SPRINT 3: GESTIÓN RESPONSABLE

El Sprint 3 (Gestión Responsable) establece la codificación para el panel del usuario con rol Responsable. Este Sprint tiene similitudes con el Sprint 2 ya que en la parte del backend se utilizan las mismas operaciones CRUD para usuarios y se establecen operaciones adicionales para operaciones de lectura y actualización de los documentos del sistema web. Además, en la parte del frontend, se reutilizan componentes establecidos en el Sprint 2 para la elaboración de la interfaz gráfica.

2.2.4.1 CODIFICACIÓN BACKEND

Para la gestión del usuario Responsable, se establecieron los queries, mutations y resolvers para realizar operaciones CRUD de universidades y de usuarios pertenecientes a una misma universidad. Además, se especificarán operaciones para listar y actualizar documentos particulares. A continuación, se muestran como ejemplo las operaciones para los documentos, mientras que las demás operaciones concernientes a este Sprint se las puede verificar en el Anexo D.

- **Queries**

En la Figura 2.58, se visualizan los queries `documents()`, `documentsByStatus()` y `document()`.

```
1 type Query {
2   "Lista de documentos Registrados en la base de datos de acuerdo al estado
3   y la universidad a la que corresponde"
4   documents(
5     "Página seleccionada para acceder a ella"
6     page: Int = 1
7     "Documentos que se obtendrán por pagina"
8     itemsPage: Int = 20
9     "Especificar el estado de los documentos que queremos mostrar"
10    status: StatusFilterEnum = ALL
11    "Especificamos el ID de la universidad de la cual queremos los documentos"
12    university: ID!
13  ): ResultDocuments
14  "Lista de documentos registrados en la base de datos de acuerdo al estado
15  diferente de unassigned y la universidad a la que corresponde"
16  documentsByStatus(
17    "Pagina seleccionada para acceder a ella"
18    page: Int = 1
19    "Documentos que se obtendrán por pagina"
20    itemsPage: Int = 20
21    "Especificamos el ID de la universidad de los cuales queremos los documentos"
22    university: ID!
23  ): ResultDocuments
24  "documento seleccionado"
25  document(id: ID!, user: ID): ResultDocument
26 }
```

Figura 2.58. Código queries de documentos

El primer query (línea 4) obtiene una lista de documentos de acuerdo con el estado que se encuentren (no asignados, asignados, revisados y finalizados) y a la universidad a la

que pertenecen. El segundo query (línea 16) obtiene una lista de documentos que ya hayan sido asignados, revisados o finalizados y que pertenezcan a una universidad específica. Finalmente, el tercer query (línea 25) obtiene la información de un documento particular.

- **Mutations**

El mutation para actualizar un documento específico denominado `updateDocument()` se lo indica en la Figura 2.59. Esta operación recibe como parámetro el documento a actualizar y devuelve como resultado el documento con los cambios realizados (línea 3).

```
1 type Mutation {
2   "Actualizar un Documento"
3   updateDocument(document: DocumentInput!): ResultDocument
4 }
```

Figura 2.59. Código mutation de documentos

- **Resolvers**

Los resolvers para los queries `documents()`, `documentsByStatus()` y `document()` se los indica en la Figura 2.60. Todos los resolvers, hacen uso de diferentes métodos especificados en el servicio `DocumentsService`.

```
5 Query: {
6   // root, variables, context Documentos con filtro por universidad y estatus
7   async documents(_, { page, itemsPage, status, university }, context) {
8     return new DocumentsService(
9       →
10      {
11        pagination: { page, itemsPage },
12      },
13      context
14    ).documents(status, university);
15  },
16  // root, variables, context Documentos con filtro por universidad y estatus diferente de unassigned
17  async documentsByStatus(_, { page, itemsPage, university }, context) {
18    return new DocumentsService(
19      →
20      {
21        pagination: { page, itemsPage },
22      },
23      context
24    ).documentsByStatus(university);
25  },
26  /*Funcion para obtener un documento especifico mediante id*/
27  async document(_, { id }, { db }) {
28    return new DocumentsService(_, { id }, { db }).details();
29  },
30 },
```

Figura 2.60. Resolvers de queries para documentos

Parte del resolver para el mutation `updateDocument()` se lo muestra en la Figura 2.61 y hace uso del método `modify` del servicio `DocumentsService` (línea 7). Este método verifica que el documento a actualizar no tenga valores nulos y mediante una búsqueda por id, utiliza la operación `updateOneElement` de la base de datos para la actualización de un documento específico y en caso de haber problemas, devuelve un feedback de mensajes con los errores obtenidos.

```
4 Mutation: {
5   //Funcion para actualizar un item nuevo
6   updateDocument(_, { document }, context) {
7     return new DocumentsService(_, { document }, context).modify();
8   },
9 }
```

Figura 2.61. Resolver de mutation para documentos

2.2.4.2 CODIFICACIÓN FRONTEND

El panel del usuario Responsable utiliza la misma librería `ngx-charts` especificada en el frontend del Sprint 2 para visualizar mediante una tabla numérica la cantidad de usuarios, universidades y a través de un gráfico circular el estado de los documentos. En la Figura 2.62 se muestra parte del código html del componente `dashboard`.

```
1 <div class="container-fluid" *ngIf="!loading">
2   <app-dashboard-info [items]="items" [items2]="items2">
3     </app-dashboard-info>
4 </div>
```

Figura 2.62. Código de `dashboard.componente.html`

`Dashboard` usa un componente reutilizable al que se le envía como primer parámetro (`items`) un array de objetos en el que se especifica la cantidad de usuarios y universidades. Además, como segundo parámetro (`items2`) se envía un array de objetos con los estados de los documentos que pertenecen a una misma universidad.

2.2.5 SPRINT 4: GESTIÓN COLABORADOR

El Sprint 4 (Gestión Colaborador) establece la codificación para el módulo del usuario con rol Colaborador. En la parte del backend se tienen operaciones creadas para los ítems y cambios que pertenecen a un documento. Además, se generaron operaciones para las versiones que tienen cada uno de los ítems de un documento. Por otro lado, en el frontend se tiene la interfaz gráfica que permite visualizar y editar el contenido de un

documento, aparte se implementó una barra lateral para visibilizar los comentarios realizados por el usuario Supervisor.

2.2.5.1 CODIFICACIÓN BACKEND

A continuación, se indicarán los queries, mutations y resolvers de los cambios realizados en un documento. Las demás operaciones se las puede verificar en el Anexo D.

- **Queries**

Para los cambios de un documento se especificaron 2 queries, los cuales se los indica en la Figura 2.63. El query `changesByStatus()` permite obtener una lista de cambios de un documento y acepta como parámetros el id y estado del documento, así como también las propiedades para el sistema de paginación y obtiene como resultado un type `ResultChanges` (línea 13) que contiene el estado de la operación y paginación, además, un mensaje de feedback con los cambios del documento. Por otro lado, el query `change()` permite obtener un type `ResultChange` (línea 17) que contiene el estado de la operación y paginación, además, un mensaje de feedback con el documento seleccionado.

```
1  type Query {
2    "Lista de cambios registrados en la base de datos
3    de acuerdo al estado al que corresponde"
4    changesByStatus(
5      "Pagina seleccionada para acceder a ella"
6      page: Int = 1
7      "Documentos que se obtendrán por pagina"
8      itemsPage: Int = 20
9      "status"
10     status: StatusFilterEnum = ALL
11     "especificamos el ID del documento"
12     documentId: ID!
13   ): ResultChanges
14   "cambio seleccionado"
15   change(
16     id: ID!
17   ): ResultChange
18 }
```

Figura 2.63. Código queries de los cambios

- **Mutations**

En los mutations que se muestran en la Figura 2.64, se especifican 3 operaciones para registrar (línea 3), actualizar (línea 5) y eliminar (línea 7) un cambio de un documento. Estas operaciones tienen una estructura similar a la establecida para la colección universidades en el Sprint 2.

```

1 type Mutation {
2   "Añadir un Cambio"
3   registerChange(change: ChangeInput!): ResultChange
4   "Actualizar un Cambio"
5   updateChange(change: ChangeInput!): ResultChange
6   "Para eliminar un Cambio"
7   deleteChange(id: ID!): ResultChange
8 }

```

Figura 2.64. Código mutations de los cambios

- **Resolvers**

Parte del código para los resolvers de los queries se muestran en la Figura 2.65. El resolver query-changesByStatus hace uso del método changesByStatus del servicio ChangesService (línea 10) mientras que el resolver query-change utiliza el método details del mismo servicio (línea 14).

```

4 Query: {
5   // Función para obtener cambios por status
6   async changesByStatus(_, {page, itemsPage, status, documentId}, context) {
7     //
8     return new ChangesService(_, {
9       pagination: {page, itemsPage}
10    }, context).changesByStatus(status, documentId);
11  },
12  // Funcion para obtener un cambio especifico mediante id*/
13  async change(_, {id}, {db}) {
14    return new ChangesService(_, {id}, {db}).details();
15  }
16 }

```

Figura 2.65. Código resolvers-queries de los cambios

Por otro lado, los resolvers para los mutations se especifican en la Figura 2.66. Todos los resolvers hacen uso de los métodos establecidos en el servicio ChangesService para realizar las operaciones para creación (línea 7), actualización (línea 11) y eliminación (línea 15) en la base de datos.

```

4 Mutation: {
5   //Funcion para registrar un cambio nuevo
6   registerChange(_, { change }, context) {
7     return new ChangesService(_, { change }, context).registerChange();
8   },
9   //Funcion para actualizar un cambio
10  updateChange(_, { change }, context) {
11    return new ChangesService(_, { change }, context).modify();
12  },
13  //Funcion para eliminar un cambio
14  deleteChange(_, {id}, context) {
15    return new ChangesService(_, {id}, context).delete();
16  },
17 }

```

Figura 2.66. Código resolvers-mutations de los cambios

2.2.5.2 CODIFICACIÓN FRONTEND

En el diseño de la interfaz gráfica del módulo de gestión del usuario con rol Colaborador, es necesario la instalación del editor de texto CKEditor mediante el siguiente paquete npm:

```
> npm install --save @ckeditor/ckeditor5-angular
```

Una vez instalada la librería, se procede a crear un componente reutilizable dentro de la carpeta shared denominado text-editor. Este componente contiene un módulo, un archivo .ts, un archivo .html y un archivo .scss como se indica en la Figura 2.67.

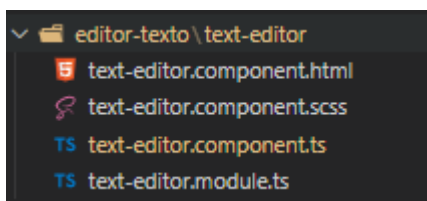


Figura 2.67. Archivos del componente text-editor

En el módulo del componente se procede a importar el módulo de CKEditor para poder usarlo en la aplicación de angular mientras que en el archivo html se debe configurar como se muestra en la Figura 2.68.

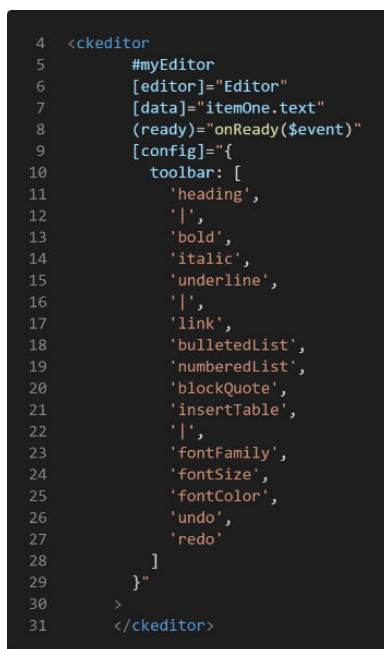


Figura 2.68. Código de text-editor.component.html

Esta configuración permite tener una barra de herramientas en el editor de texto y establecer un valor dentro del editor en formato WYSIWYG, así como también obtener dicho valor para almacenarlo en la base de datos en formato HTML.

2.2.6 SPRINT 5: GESTIÓN SUPERVISOR

El Sprint 5 (Gestión Supervisor) establece la gestión del módulo del usuario con rol Supervisor y es similar al panel del usuario con rol Colaborador, por lo tanto, en la parte del backend se utilizarán operaciones de queries, mutations y resolvers para documentos, ítems, cambios y versiones. Mientras que, en el frontend se agrega una barra lateral que permite realizar operaciones CRUD de los cambios que pertenecen a un documento.

2.2.6.1 CODIFICACIÓN BACKEND

A manera de ejemplo se especificarán los queries, mutations y resolvers de los ítems de un documento. Las demás operaciones se encuentran en el Anexo D.

- **Queries**

Para los ítems de un documento se especificaron 3 queries, los cuales se los indican en la Figura 2.69. El query ítems (línea 3) permite obtener una lista de los ítems registrados en la base de datos. Luego, el query itemsDocument (línea 12) adquiere una lista de ítems que pertenecen a un documento particular. Finalmente, el query itemsArrayIds (línea 23) permite obtener una lista de ítems al pasar como parámetro una lista de ids.

```

1  type Query {
2    "Lista de items de los documentos registrados en la BD"
3    items(
4      "Pagina seleccionada para acceder a ella"
5      page: Int = 1
6      "Items que se obtendrán por pagina"
7      itemsPage: Int = 20
8      "Especificar el estado del item que queremos mostrar"
9      active: ActiveFilterEnum = ACTIVE
10   ): ResultItem
11   "Lista de items de un documento especifico registrado en la BD"
12   itemsDocument(
13     "Pagina seleccionada para acceder a ella"
14     page: Int = 1
15     "Items que se obtendrán por pagina"
16     itemsPage: Int = 20
17     "Especificar el estado del item que queremos mostrar"
18     active: ActiveFilterEnum = ACTIVE
19     "Especificar el ID del Documento Seleccionado"
20     document: ID!
21   ): ResultItems
22   "Obtener un array de items dado un array de IDS"
23   itemsArray(ids: [String]): ResultItems
24 }
25

```

Figura 2.69. Código queries de ítems

- **Mutations**

En la Figura 2.70 se establece un mutation para añadir un nuevo ítem (línea 3) y un mutation para actualizar un ítem específico (línea 5). Las 2 operaciones reciben como parámetro de entrada un ítem y devuelve un type ResultItem que contiene el estado de la operación, además, un mensaje de feedback con el ítem creado o actualizado.

```

1  type Mutation {
2    "Añadir un Item"
3    registerItem(item: ItemInput!): ResultItem
4    "Actualizar un Item"
5    updateItem(item: ItemInput!): ResultItem
6  }

```

Figura 2.70. Código mutations de items

- **Resolvers**

En la Figura 2.71 se muestran los resolvers de los queries para los ítems, los cuales hacen uso de las operaciones establecidas en el servicio ItemsService.

```

5 Query:{
6   // Función para obtener todos los items
7   items(_, {page, itemsPage, active}, context){
8     return new ItemsService(_, {
9       pagination: {page, itemsPage}
10    }, context).items(active);
11  },
12  // Función para obtener Items mediante el ID del documento
13  itemsDocument(_, {page, itemsPage, active, documentId}, context){
14    return new ItemsService(_, {
15      pagination: {page, itemsPage}
16    }, context).items(active, documentId);
17  },
18  // Función para obtener los items de la BD pasando un array de Ids
19  async itemsArray(_, {ids}, context) {
20    return new ItemsService(_, {
21    }, context).itemsArray(ids);
22  },
23  },

```

Figura 2.71. Resolvers-queries de ítems

En la Figura 2.72 se muestran los resolvers de los mutations para los ítems, los cuales hacen uso de las operaciones establecidas en el servicio ItemsService.

```

4 Mutation: {
5   //Función para registrar un item nuevo
6   registerItem(_, { item }, context) {
7     return new ItemsService(_, { item }, context).registerItem();
8   },
9   //Función para actualizar un item
10  updateItem(_, { item }, context) {
11    return new ItemsService(_, { item }, context).modify();
12  },
13  },

```

Figura 2.72. Resolvers-mutations de ítems

2.2.6.2 CODIFICACIÓN FRONTEND

Parte del código para la implementación de la barra lateral para los cambios de un documento se lo visualiza en la Figura 2.73. En la barra lateral, se establece una tabla con 5 columnas (línea 55) que contienen información de los campos de cada documento de la colección changes y una columna con 3 botones para editar, aprobar o eliminar un cambio específico (líneas 77-92).

```

55 <table class="table table-striped table-hover" *ngIf="resultAllChanges.length>=1; else elseB">
56   <thead>
57     <tr>
58       <th scope="col" >ID</th>
59       <th scope="col">Nombre</th>
60       <th scope="col">Descripción</th>
61       <th scope="col">Creado hace</th>
62       <th scope="col">Estado</th>
63       <th scope="col">Gestionar Información</th>
64     </tr>
65   </thead>
66
67   <tbody>
68     <tr *ngFor="let data of resultAllChanges">
69       <td *ngFor="let val of tableColumns">
70         <div *ngIf="val.property==='registerDate'; else elseBlock">
71           {{data[val.property]|dateAgo}}
72         </div>
73         <ng-template #elseBlock>
74           {{ data[val.property]}}
75         </ng-template>
76       </td>
77       <td>
78         <button type="button" class="btn btn-secondary mb-2 mr-2" data-bs-target="#exampleModal"
79           (click)="manageAction('edit', data)">
80           <i class="far fa-edit"></i></button>
81         >
82         <button type="button" class="btn btn-success mb-2"
83           (click)="manageAction('review', data)">
84         >
85         <i class="fas fa-check-double"></i>
86       </button>
87       <button type="button" class="btn btn-danger"
88         (click)="manageAction('delete', data)">
89       >
90       <i class="fas fa-trash-alt"></i>
91     </button>
92   </td>
93 </tr>
94 </tbody>
95 </table>

```

Figura 2.73. Código archivo comments.component.html

2.2.7 SPRINT 6: GENERACIÓN DE DOCUMENTOS FINALES

El Sprint 6 (Generación de Documentos Finales) establece las funcionalidades de los módulos Formulario Documentos y Documentos Finales. En la parte del backend se utilizan las mismas operaciones de queries y resolvers de documentos, versiones y cambios, por lo que no se añadirá código adicional. Mientras tanto en la parte del frontend se crea un panel para visualizar información de un documento específico y un botón para imprimirlo en formato PDF.

2.2.7.1 CODIFICACIÓN FRONTEND

- **Módulo Formulario Documentos**

En la Figura 2.74 se muestra parte del código del componente document-one-final en el que se inicializa el componente mediante el método ngOnInit (línea 88).

```

88  ngOnInit(): void {
89    loadData('Cargando Datos', 'Por favor espere');
90    this.getParamsRouter();
91    this.documentsService
92      .getDocumentFinalById(
93        this.params.id
94      )
95      .subscribe((result) => {
96        console.log('Documento 1', result);
97        this.resultDocument = result;
98        this.name = `${this.resultDocument.name}`;
99        console.log('ITEms', result.items);
100       this.getVersionsActive(result.itemsId, result.id);
101       this.loading = false;
102       closeAlert();
103     });
104   }

```

Figura 2.74. Código archivo document-one-final.component.ts

En el ngOnInit() se muestra un mensaje de carga de datos (línea 89) mientras se inicializa el componente. Por otro lado, se obtienen los parámetros del id del documento y el id de la universidad enviados por la ruta (línea 90) para realizar la suscripción a los servicios del documento, versiones y de los cambios. De esta manera se obtienen todos los ítems con sus versiones activas que pertenecen a un documento específico y los cambios realizados para poder mostrarlos visualmente.

- **Módulo Documentos Finales**

Para poder generar un archivo en formato PDF se creó una función que hace uso del método Windows.print() y se la muestra en la Figura 2.75.

```

79  PrintElem() {
80    const mywindow = window.open('', 'PRINT', 'fullscreen="yes"');
81    mywindow.document.write(
82      '<html><title>' +
83        this.name +
84        '</title> <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"'
85    );
86    mywindow.document.write(document.getElementById('faq').innerHTML);
87    mywindow.document.close();
88    mywindow.focus();
89    mywindow.print();
90    mywindow.close();
91    return true;
92  }

```

Figura 2.75. Función PrintElem()

En dicha función se especifica el contenido que se quiere visualizar en el PDF, además, en las líneas 82-84, se establece un título para el documento y se agregan los estilos de bootstrap. Windows.print (línea 89) genera una ventana emergente que permite al usuario configurar los márgenes del documento y añadir encabezado y pie de página.

2.2.8 SPRINT 7: GESTIÓN PANEL HOME

El Sprint 7 (Gestión Panel Home) establece las funcionalidades de los módulos Tablero Estado de Tareas y Visualización Calendario. En la parte del backend se utilizan las mismas operaciones de queries y resolvers de documentos, por lo que no se añadirá código adicional. Mientras tanto en la parte del frontend se crean 3 columnas que representan el estado de los documentos y un panel para visualizar la fecha de entrega de cada documento mediante un calendario.

2.2.8.1 CODIFICACIÓN FRONTEND

Para el establecimiento del Panel Home se tiene la estructura que se muestra en la Figura 2.76. La carpeta @public contiene las carpetas core y pages:

- La carpeta core contiene componentes para el header, navbar y footer del sistema web. Además, contiene 2 servicios creados para abrir y cerrar la barra lateral de los cambios de un documento específico.
- La carpeta pages contiene el panel home como ruta principal por defecto y se compone de rutas hijas para acceder a los módulos de visualización de calendario, colaboradores, supervisores, y generación de documentos finales.

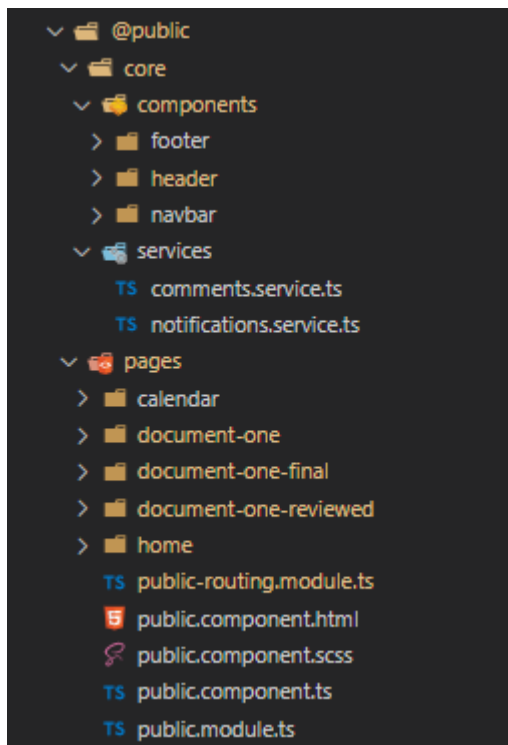


Figura 2.76. Estructura módulo Home

- **Tablero Estado de Tareas**

Para visualizar los documentos mediante su estado se establecieron 4 columnas para: Fechas de entrega de los documentos, documentos en edición, revisión y finalizados. En la Figura 2.77 se muestra parte del código de la columna de los documentos con el estado asignados. Esta columna hace uso de un componente secundario llamado documents-card (líneas 14-21), al que se le pasa como parámetros un id de la universidad, una consulta para obtener una lista de documentos y un filtro. El componente secundario muestra en la pantalla los documentos y al hacer clic sobre el documento seleccionado devuelve el id al componente principal para poder navegar al módulo del Colaborador.

```
12 <div class="item begin">
13   <div class="card-header principalH"><h4 class="text-center">Editar</h4></div>
14   <app-documents-card
15     [universityId]= "university"
16     [query]="query"
17     [itemsPage]="itemsPage"
18     [include]="include"
19     [filterValues] = "filterAssignedValues"
20     (idDocument)="goTo($event)"
21   ></app-documents-card>
22 </div>
```

Figura 2.77. Código archivo home.component.html

- **Visualización Calendario**

Para poder tener un calendario en el sistema web, se realizó la instalación del paquete fullcalendar mediante el siguiente comando:

```
> npm install --save @fullcalendar/angular @fullcalendar/daygrid
```

Esta librería proporciona un calendario que se integra fácilmente a una aplicación de Angular. En la Figura 2.78 se muestra parte de la configuración para poder visualizar el calendario.

```
19 calendarOptions: CalendarOptions = {
20   headerToolbar: {
21     left: 'prev,today,next',
22     center: 'title',
23     right: 'prevYear,nextYear'
24   },
25   eventClick: this.handleEventClick.bind(this),
26   locale: esLocale,
27   initialView: 'dayGridMonth',
28   events: this.documentsDates,
29 };
```

Figura 2.78. Configuración librería fullcalendar

Esta configuración permite generar una barra de herramientas y así poder desplazarse por el calendario. Además, se establece el idioma del calendario en español (línea 26) y se configura la visualización del calendario por mes (línea 27). Finalmente, se agregan eventos al calendario por medio de un array que contiene información de los documentos que han sido asignados por el usuario Responsable (línea 28).

3 RESULTADOS Y DISCUSIÓN

En esta etapa se realizan pruebas de funcionamiento para validar el cumplimiento de requerimientos funcionales y no funcionales del sistema con un usuario gestor y una universidad, misma que tiene: un responsable, dos colaboradores y dos supervisores. Luego se formularon las preguntas de las encuestas efectuadas a 16 usuarios que probaron el sistema web y se asignaron como se indica en la Tabla 3.1. Más adelante, se tabularon los resultados de las encuestas y se interpretaron de manera que se verificó el diseño de la interfaz gráfica y el funcionamiento del sistema web. Finalmente, se corrigieron los errores presentados en el sistema prototipo.

Tabla 3.1. Establecimiento de usuarios encuestados por roles

CANTIDAD DE USUARIOS	ROL
4	Gestor
4	Responsable
4	Supervisores
4	Colaboradores

3.1 PRUEBAS SPRINT 1: AUTENTICACIÓN

En este Sprint se presentan las pruebas de funcionamiento con un usuario con rol Responsable para verificar el correcto funcionamiento del módulo de Autenticación.

3.1.1 INICIO DE SESIÓN

En la Figura 3.1 se muestran las alertas que se presentan en la pantalla de inicio de sesión al tener campos no válidos en el correo y contraseña. En el caso de que el usuario no esté activo, el email no exista en la base de datos o la contraseña sea incorrecta se muestran mensajes al usuario y no se le permite el acceso al sistema.

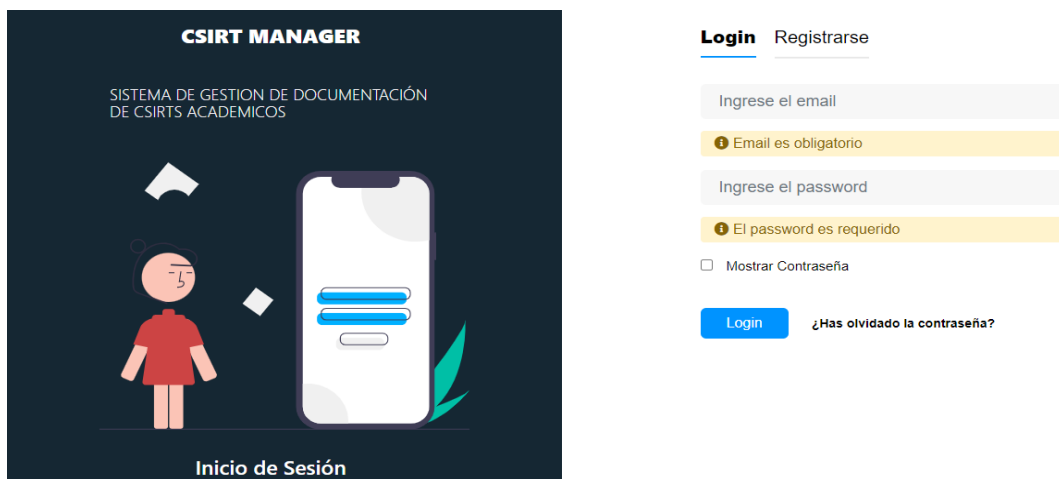


Figura 3.1. Campos no válidos en pantalla de inicio

En la Figura 3.2 se muestra un correo y contraseña válidos que se marcan con un visto verde, además, al dar click en el botón Login aparece un mensaje en la parte superior central indicando que el usuario inicia la sesión de manera correcta, como se presenta en la Figura 3.3.

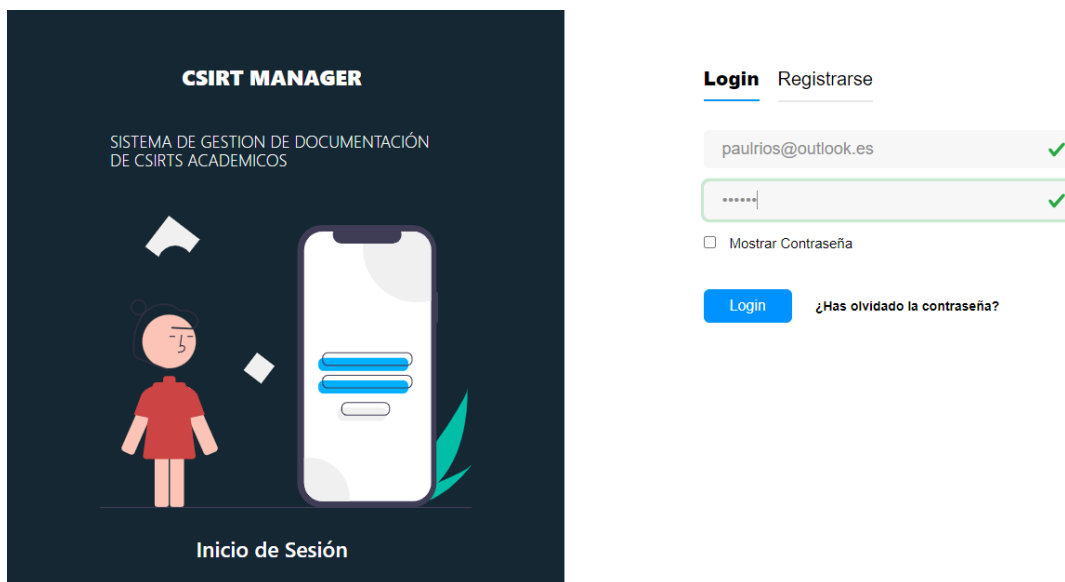


Figura 3.2. Ingreso de correo y contraseña en formulario

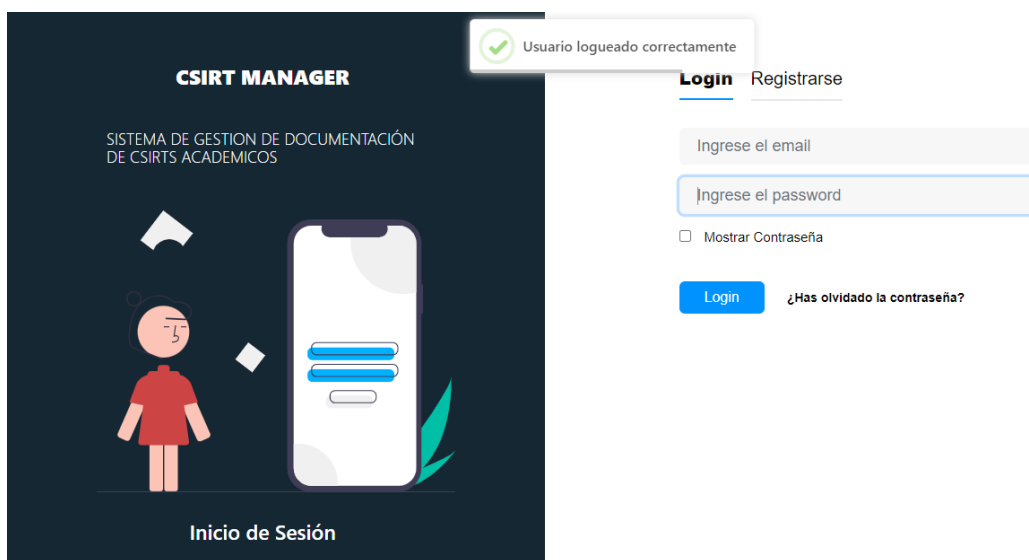


Figura 3.3. Mensajes de acceso al sistema

3.1.2 REGISTRO DE UN USUARIO

En la Figura 3.4 se muestra la pantalla de registro de un usuario con la validación de los campos vacíos. El usuario que se quiere registrar en el sistema web debe ingresar todos los datos requeridos como nombre, apellido, sexo, universidad, fecha de nacimiento, correo electrónico y contraseña como se indica en la Figura 3.5.

Inicio **Registrarse**

Ingrese el nombre El nombre es requerido

Ingrese el apellido El apellido es requerido

Ingrese el userName El username es requerido

Hombre Mujer

Seleccione una universidad Debe seleccionar una universidad

Escoja fecha de nacimiento Debe seleccionar una fecha de nacimiento válida

Ingrese el email Email es obligatorio

Ingrese el password Contraseña debe ser mayor a 6 caracteres

Confirme el password

Registrar



Figura 3.4. Mensajes de validación de campos

Inicio **Registrarse**

Franklin

Sánchez

fsanchez123

Hombre Mujer

EPN

1980-12-19

franklin.sanchez@epn.edu.ec

.....

.....

Registrar

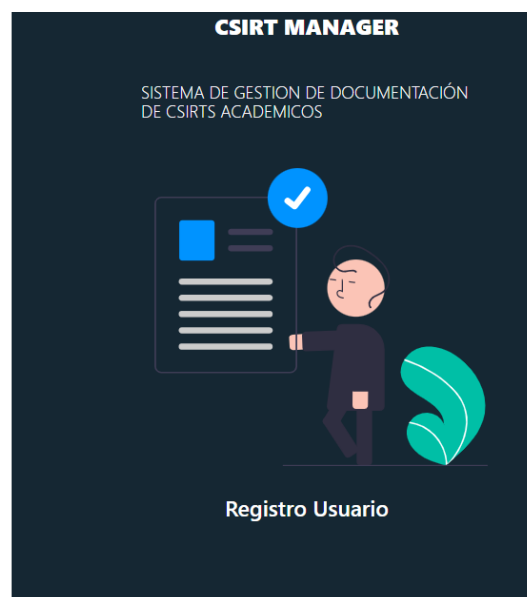


Figura 3.5. Ingreso de datos requeridos para el registro

Para comprobar que el usuario se registró en el sistema web, se procede a verificar en la base de datos a través de MongoDB Compass. Como se indica en la Figura 3.6, el usuario se encuentra registrado exitosamente y se evidencia que se cumple con el requerimiento no funcional de tener la contraseña cifrada con bcrypt. Además, se añade automáticamente el campo registerDate para saber la fecha en que el usuario realizó su registro. Por otro lado, se tiene el campo active en false ya que es necesario que el usuario con rol Gestor realice la activación del usuario con rol Responsable para poder acceder al sistema web.

```
_id: ObjectId("61b26bd81e4bc00163af74f")
name: "Franklin"
lastname: "Sánchez"
email: "franklin.sanchez@epn.edu.ec"
username: "fsanchez123"
rol: "RESPONSABLE"
gender: "MALE"
university: "2"
password: "$2b$10$JVw0DYAk34IGY10ETFQ.WU.RqHGV6jXv4m7zos2IwKRpsYqw6i8PC"
birthday: "1980-12-19"
active: false
id: "6"
registerDate: "2021-12-09T20:49:28.177Z"
```

Figura 3.6. Verificación del usuario en la base de datos

3.1.3 CAMBIO DE CONTRASEÑA

En la Figura 3.7 se presenta la pantalla para realizar la petición del cambio de contraseña de un usuario específico. Por defecto el botón para enviar la petición se encuentra desactivado hasta que el usuario ingrese un correo electrónico válido como se indica en la Figura 3.8.

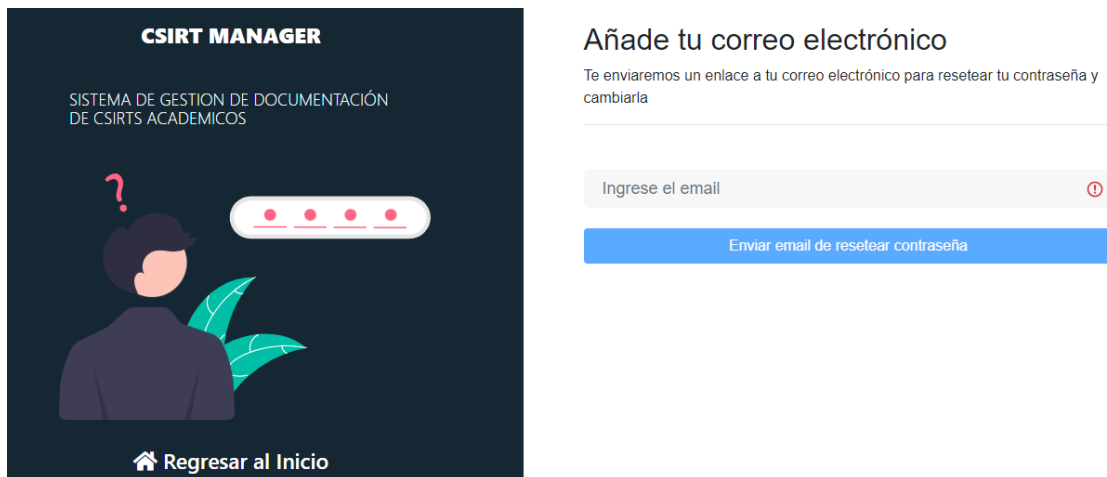


Figura 3.7. Pantalla para restablecer contraseña

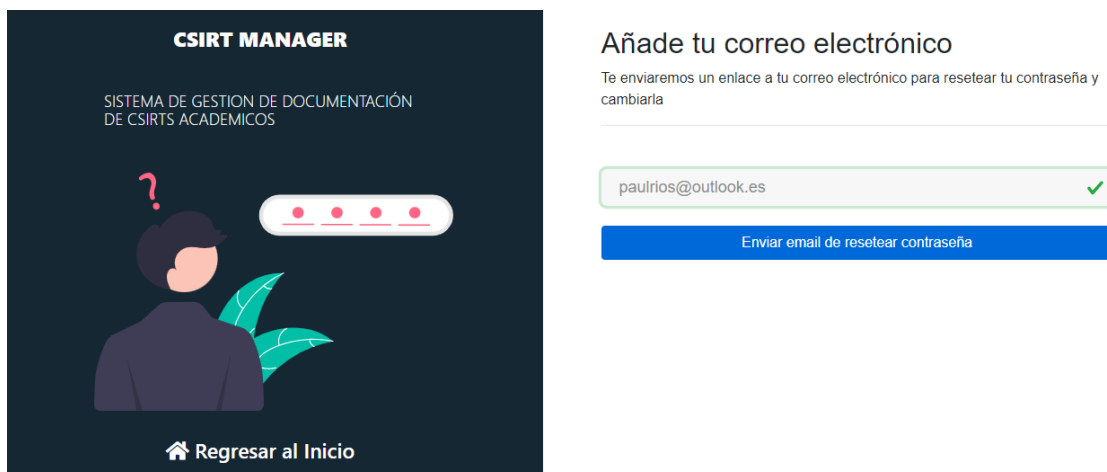


Figura 3.8. Validación de correo electrónico

Luego, al dar clic en el botón se muestra el mensaje de que la petición para el cambio de la contraseña fue enviada de manera correcta al correo electrónico proporcionado como se presenta en la Figura 3.9. Además, se cambia automáticamente del formulario de petición para el cambio de contraseña a la pantalla de inicio de sesión del sistema web.

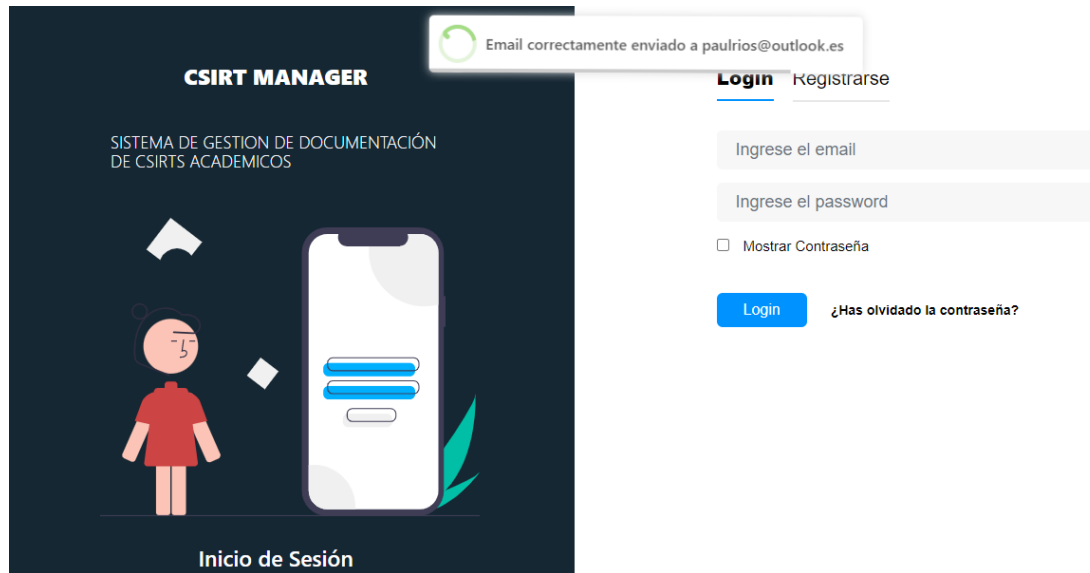


Figura 3.9. Mensaje de envío de restablecimiento de contraseña

Para comprobar que la petición para cambiar la contraseña se envió de manera correcta, se procede a verificar en la bandeja de entrada del sistema de correo electrónico como se muestra en la Figura 3.10. Este email recibido contiene un mensaje indicando que para realizar el restablecimiento de la contraseña se debe dar clic en el enlace.

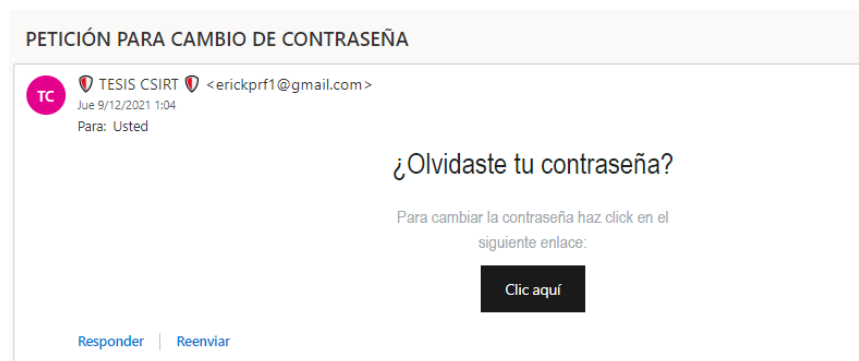


Figura 3.10. Email recibido en el servicio de correo electrónico

Al dar clic en el enlace proporcionado, se muestra el formulario para realizar el cambio de la contraseña que se lo presenta en la Figura 3.11. Este formulario tiene 2 campos para ingresar y confirmar la nueva contraseña. Además, se tiene un botón que se encuentra deshabilitado mientras no se llenen los campos requeridos de manera correcta.

Cambiar contraseña

Para realizar el cambio, por favor ingresa la nueva contraseña y verificalo antes del cambio

Cambiar contraseña



Figura 3.11. Formulario para cambiar la contraseña

En la Figura 3.12, se muestran los mensajes de error en caso de no ingresar una contraseña mayor a 6 caracteres o que las contraseñas no coincidan. Una vez ingresados los campos y que sean válidos, se procede a activar el botón para cambiar la contraseña como se muestra en la Figura 3.13.

Cambiar contraseña

Para realizar el cambio, por favor ingresa la nueva contraseña y verificalo antes del cambio

❗ Contraseña debe ser mayor a 6 caracteres

❗ Las contraseñas deben ser iguales

Cambiar contraseña



Figura 3.12. Campos inválidos para cambio de contraseña

Cambiar contraseña

Para realizar el cambio, por favor ingresa la nueva contraseña y verificalo antes del cambio

.....

.....

Cambiar contraseña



Figura 3.13. Campos válidos para cambiar la contraseña

Al dar clic en el botón, se realiza el cambio de la contraseña en la base de datos, luego se muestra un mensaje al usuario de que la operación fue exitosa y se cambia automáticamente a la pantalla de inicio de sesión como se muestra en la Figura 3.14.



Figura 3.14. Mensaje de cambio de contraseña exitoso

3.2 PRUEBAS SPRINT 2: GESTIÓN GESTOR

En este Sprint se muestran las pruebas de funcionamiento con un usuario con rol Gestor para realizar la verificación del módulo Gestión Gestor.

3.2.1 PANEL PRINCIPAL

Una vez que el usuario con rol Gestor accede al sistema se muestra el panel principal como se indica en la Figura 3.15. Este panel muestra al Gestor la cantidad de usuarios, universidades y documentos que se encuentran en el sistema web. Además, se tiene una barra lateral en la parte izquierda para acceder a la gestión de universidades y usuarios. Por otro lado, se visualiza en la parte superior derecha información de la sesión del usuario y una opción para salir del sistema web. Finalmente, se tiene un botón que permite ocultar la barra lateral para poder visualizar de mejor manera el panel principal como se muestra en la Figura 3.16.

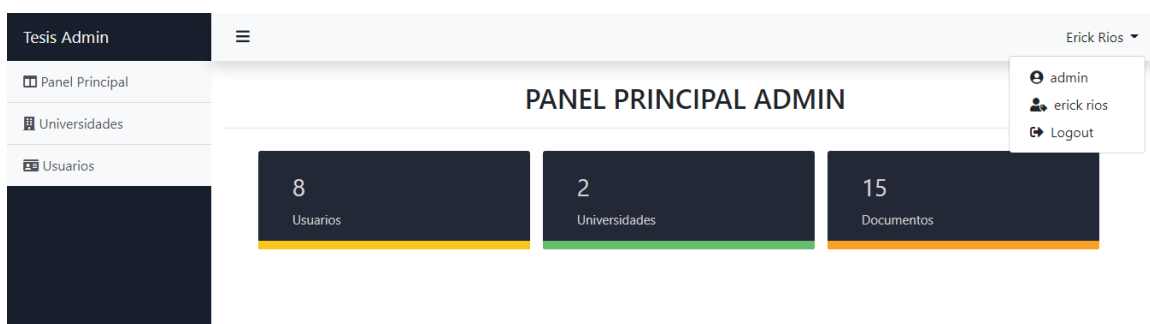


Figura 3.15. Panel principal del usuario gestor

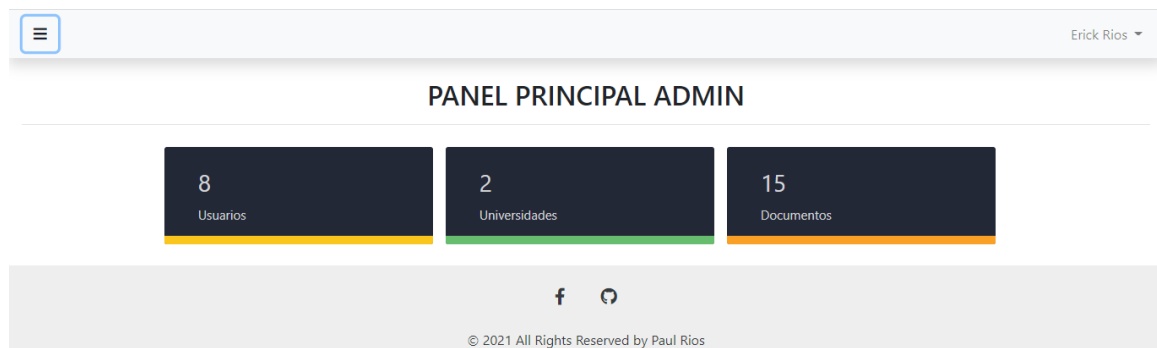


Figura 3.16. Barra lateral oculta en panel principal

3.2.2 PANEL UNIVERSIDADES

Al dar clic la opción Universidades de la barra lateral, el usuario accede a un panel para poder realizar operaciones CRUD sobre las universidades que hay en el sistema web como se muestra en la Figura 3.17. En este panel se puede visualizar información de las universidades que existen mediante una tabla. Además, se puede seleccionar un filtro para visualizar las universidades activas, inactivas o todas. Por otro lado, la tabla cuenta con sistema de paginación y se puede seleccionar que se visualicen 5, 10, 15 o 20 universidades por página.

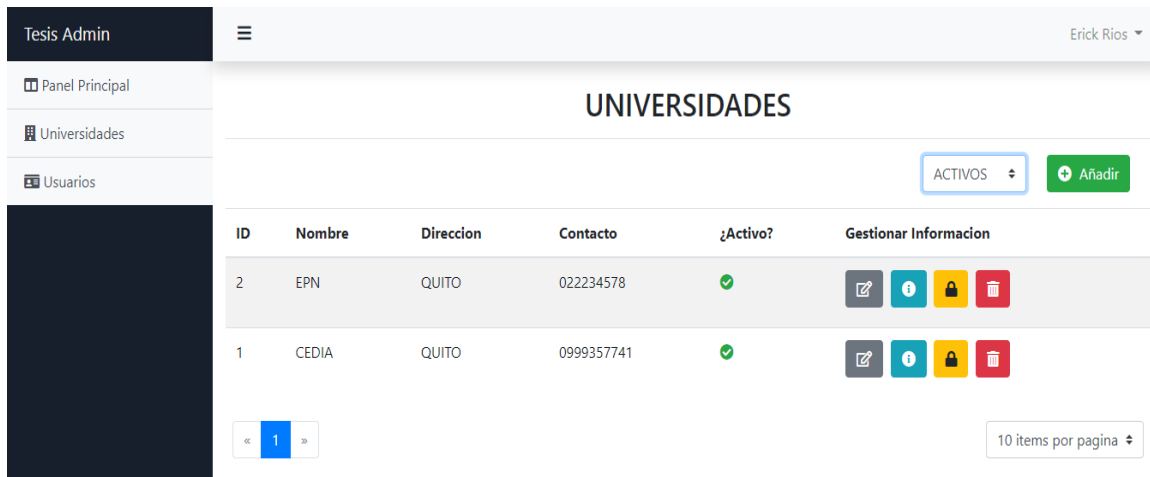


Figura 3.17. Panel gestor-crud-universidades

3.2.2.1 CREAR UNIVERSIDADES

Para crear una universidad nueva, se procede a dar clic en el botón Añadir y se muestra un cuadro de diálogo con los campos que se deben llenar para poder registrar una universidad como se muestra en la Figura 3.18.



Figura 3.18. Cuadro de diálogo para registro de universidad

En caso de no ingresar todos los campos requeridos se muestran mensajes de error y no se permite realizar el registro hasta que todo se encuentre validado como se indica en la Figura 3.19.

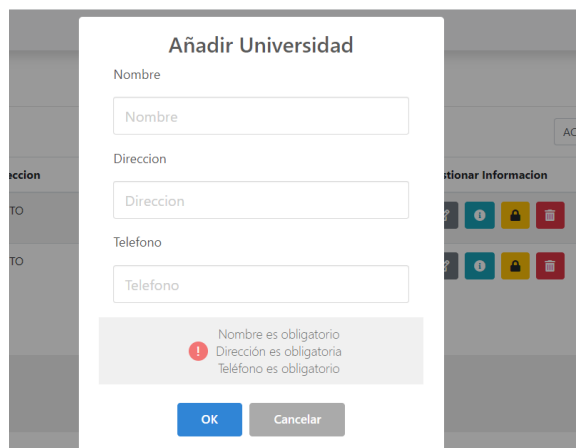


Figura 3.19. Mensajes de error cuadro de diálogo

Una vez ingresados los datos de manera correcta como se indica en la Figura 3.20, se procede a dar clic en el botón ok, se cierra el cuadro de diálogo y se muestra un mensaje al usuario indicando si el registro de la nueva universidad fue exitoso o hubo problemas.



Figura 3.20. Campos válidos para el registro de la universidad

Para comprobar que la universidad se registró en el sistema web, se procede a verificar en la base de datos a través de MongoDB Compass. En la Figura 3.21 se muestra la universidad registrada. Además, se añade automáticamente el campo registerDate para saber la fecha en que se realizó el registro de la universidad. Por otro lado, se tiene el campo active en false por defecto y en caso de querer activar la universidad se lo debe realizar desde el panel del gestor.

```

_id: ObjectId("61b297ed32bd1100160e6a57")
name: "ESPE"
address: "Av. General Rumiñahui S/N y Ambato"
phone: "023989400"
active: false
id: "3"
registerDate: "2021-12-09T23:57:33.754Z"

```

Figura 3.21. Universidad registrada en la base de datos

3.2.2.2 EDITAR UNIVERSIDADES

Para editar una universidad, se procede a dar clic en el botón de editar y se muestra un cuadro de diálogo con los campos de la universidad seleccionada para realizar los cambios como se muestra en la Figura 3.22. Una vez realizados los cambios como se presenta en la Figura 3.23, se procede a dar clic en el botón ok para guardar los cambios.



The image shows a dialog box titled "Modificar Universidad". It has three input fields: "Nombre" with the value "EPN", "Direccion" with the value "QUITO", and "Telefono" with the value "022234578". At the bottom, there are two buttons: "OK" (blue) and "Cancelar" (grey). The dialog is overlaid on a background that shows a list of universities with "EPN" and "CEDIA" visible, and a "Gestionar Inf" sidebar with edit and info icons.

Figura 3.22. Cuadro de diálogo para modificar la universidad



The image shows the same "Modificar Universidad" dialog box, but the "Direccion" field now contains the updated address: "Av. Ladrón de Guevara 253, Quito 170525". The "Nombre" and "Telefono" fields remain the same. The "OK" and "Cancelar" buttons are still present at the bottom.

Figura 3.23. Cuadro de diálogo con la universidad modificada

Para comprobar que la universidad se actualizó en el sistema web, se procede a verificar a través de MongoDB Compass. En la Figura 3.24 se indica la universidad editada en la base de datos.

```
_id: ObjectId("6165048709ffb431346da7cb")
name: "EPN"
address: "Av. Ladrón de Guevara 253, Quito 170525"
phone: "022234578"
active: true
id: "2"
registerDate: "2021-10-12"
```

Figura 3.24. Universidad actualizada en la base de datos

3.2.2.3 VER INFORMACIÓN DE UNA UNIVERSIDAD

Para ver los datos de una universidad específica, se da clic en el botón de información y se abre un cuadro de dialogo en el que se muestran los datos de la universidad seleccionada como en la Figura 3.25.



Figura 3.25. Información de la universidad seleccionada

3.2.2.4 BLOQUEAR UNIVERSIDADES

Para activar o desactivar una universidad específica, se da clic en el botón de bloqueo y se muestra un cuadro de dialogo como se indica en la Figura 3.26 con un mensaje para el usuario en el que se especifican 2 botones para confirmar o cancelar la acción.



Figura 3.26. Cuadro de diálogo para desactivar una universidad

En la **Figura 3.27** se muestra la comprobación de la universidad bloqueada en la que se evidencia el campo active en false mediante MongoDB Compass.

```
_id: ObjectId("6165048709ffb431346da7cb")
name: "EPN"
address: "Av. Ladrón de Guevara 253, Quito 170525"
phone: "022234578"
active: false
id: "2"
registerDate: "2021-10-12"
```

Figura 3.27. Universidad bloqueada en la base de datos

3.2.2.5 ELIMINAR UNIVERSIDADES

Para eliminar una universidad específica, se da clic en el botón de eliminar y se muestra un cuadro de dialogo con un mensaje para el usuario en el que se especifican 2 botones para confirmar o cancelar la acción de eliminación como se indica en la Figura 3.28.

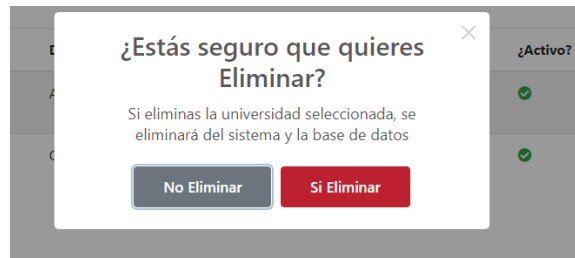


Figura 3.28. Cuadro de diálogo para eliminar una universidad

En la **Figura 3.29** se comprueba que la universidad se eliminó al realizar una consulta con el nombre de la universidad eliminada y no dar resultados en la búsqueda.

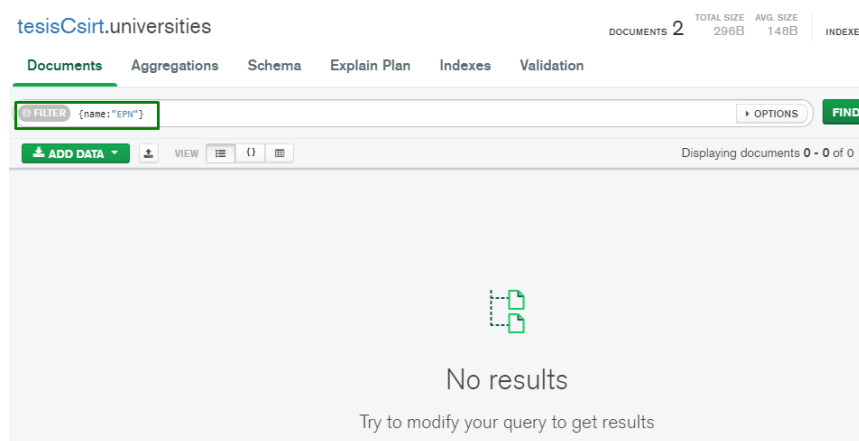








Figura 3.29. Universidad eliminada de la base de datos

3.2.3 PANEL USUARIOS

Al dar clic la opción Usuarios de la barra lateral, el usuario accede a un panel para poder realizar operaciones CRUD sobre los usuarios que hay en el sistema web como se muestra en la Figura 3.30. En este panel se puede visualizar información de los usuarios que existen mediante una tabla. Además, se puede seleccionar un filtro para visualizar los usuarios activos, inactivos o todos. Por otro lado, la tabla cuenta con sistema de paginación y se puede seleccionar que se visualicen 5, 10, 15 o 20 usuarios por página.

USUARIOS									
ID	Nombre	Apellido	Username	Email	Universidad	Género	Rol	¿Activo?	Gestionar Información
4	Karen	Salazar	krios	karen.salazar@epn.edu.ec	2	FEMALE	COLABORADORP	✓	   
5	Erick	Rios	erios	erickprf1@gmail.com	2	MALE	COLABORADOR	✓	   
6	Fernando	Ortega	fortega321	fernando.ortega@epn.edu.ec	2	MALE	SUPERVISOR	✓	   
1	Erick	Rios	erickprf	erick.rios@epn.edu.ec	1	MALE	ADMIN	✓	   
2	Paul	Fiallos	paulrf1	paulrios@outlook.es	2	MALE	RESPONSABLE	✓	   

ACTIVOS + Añadir

« 1 2 » 5 items por pagina

Figura 3.30. Panel gestor-crud-usuarios

3.2.3.1 CREAR USUARIOS

Para crear un usuario nuevo, se procede a dar clic en el botón Añadir y se muestra un formulario con los campos que se deben llenar para poder registrar un usuario como se muestra en la Figura 3.31. Cabe mencionar que el usuario Gestor solamente puede realizar registros de usuarios con rol Responsable y una vez realizado el registro se envía un correo electrónico al nuevo usuario con un enlace para que realice su activación y establezca una contraseña y su fecha de nacimiento.

Tesis Admin ☰ Erick Rios

- Panel Principal
- Universidades
- Usuarios

INFO USUARIO

Nombre Apellido

Username Género Hombre Mujer

Universidad Email

Figura 3.31. Formulario para registro de un usuario

En caso de no ingresar todos los campos requeridos se muestran mensajes de error y no se permite realizar el registro hasta que todo se encuentre validado como se indica en la Figura 3.32.

The screenshot shows a web form titled 'INFO USUARIO' within a 'Tesis Admin' interface. The form has a sidebar on the left with navigation options: 'Panel Principal', 'Universidades', and 'Usuarios'. The main content area contains several input fields with associated error messages:

- Nombre:** 'Ingrese el nombre' with error 'El nombre es requerido'.
- Apellido:** 'Ingrese el apellido' with error 'El apellido es requerido'.
- Username:** 'Ingrese el username' with error 'El username es requerido'.
- Género:** Radio buttons for 'Hombre' (selected) and 'Mujer'.
- Universidad:** A dropdown menu with 'Seleccione una universidad' and error 'Debe seleccionar una universidad'.
- Email:** 'Ingrese el email' with error 'Email es obligatorio'.

At the bottom of the form are two buttons: a blue 'Guardar' button and a red 'Cancelar' button.

Figura 3.32. Mensajes de error formulario crear usuario

Una vez ingresados los datos de manera correcta como se indica en la Figura 3.33, se procede a dar clic en el botón guardar y se muestra una alerta para verificar que se quiere registrar el nuevo usuario. Al realizar la comprobación, se regresa al panel de Usuarios y se muestra un mensaje indicando si el registro del nuevo usuario fue exitoso o hubo problemas. Además, se envía el enlace de activación al correo electrónico del usuario nuevo.

This screenshot shows the same 'INFO USUARIO' form, but now with valid data entered in the fields. A modal dialog box is centered over the form, asking for confirmation: '¿Estás seguro que quieres registrar un nuevo usuario?'. The dialog has a close button (X) in the top right corner and two buttons: a grey 'Sí registrar' button and a red 'No registrar' button. The background form is dimmed, showing the following data:

- Nombre:** David
- Apellido:** Cruz
- Username:** dcruz432
- Género:** Hombre (selected)
- Universidad:** EPN
- Email:** [partially visible]

Figura 3.33. Campos válidos para el registro de la universidad

Para comprobar que el usuario se registró en el sistema web, se procede a verificar en la base de datos a través de MongoDB Compass. En la Figura 3.34 se muestra el usuario registrado. Además, se tiene el campo active en false por defecto ya que el usuario debe realizar la activación mediante el enlace que se envía automáticamente al correo proporcionado. Los campos password y birthday se cambiarán al momento de acceder al

formulario de activación y la contraseña se encuentra cifrada con bcrypt como se muestra en la Figura 3.35.

```
_id: ObjectId("61b2ee4296a31e00161d7f0d")
name: "David"
lastname: "Cruz"
email: "david.cruz@epn.edu.ec"
username: "dcruz432"
rol: "RESPONSABLE"
gender: "MALE"
university: "4"
password: "$2b$10$GrqGTI/vPfzM2aN4nk08yeriMyw3QwtV0oq1cepLgq6afabxjC6ru"
birthday: "2003-12-10"
active: false
id: "7"
registerDate: "2021-12-10T06:05:54.729Z"
```

Figura 3.34. Usuario registrado en la base de datos

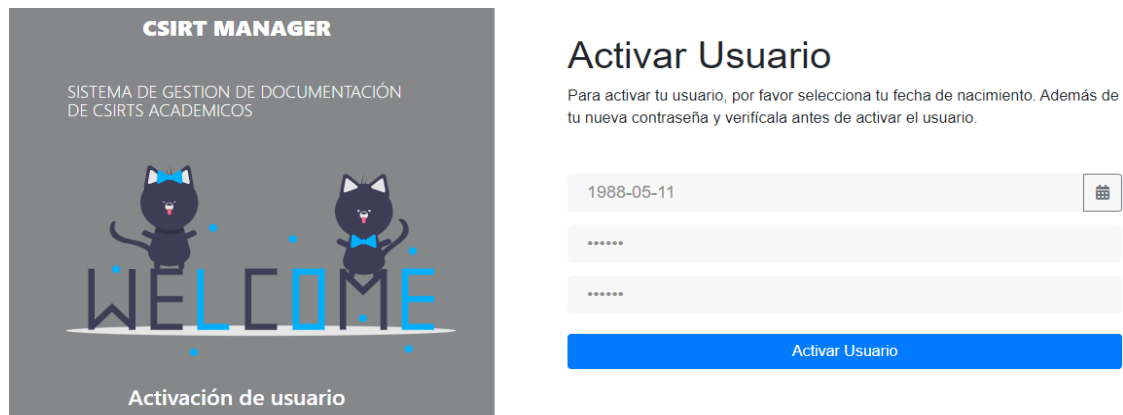


Figura 3.35. Formulario para activar el usuario

3.2.3.2 EDITAR USUARIOS

Para editar un usuario, se procede a dar clic en el botón de editar y se muestra el mismo formulario de registro, pero con la información de los campos del usuario seleccionado. De esta manera, se pueden realizar los cambios como se muestra en la Figura 3.36. Una vez realizados los cambios como se presenta en la Figura 3.37, se procede a dar clic en el botón guardar para actualizar los cambios en la base de datos.

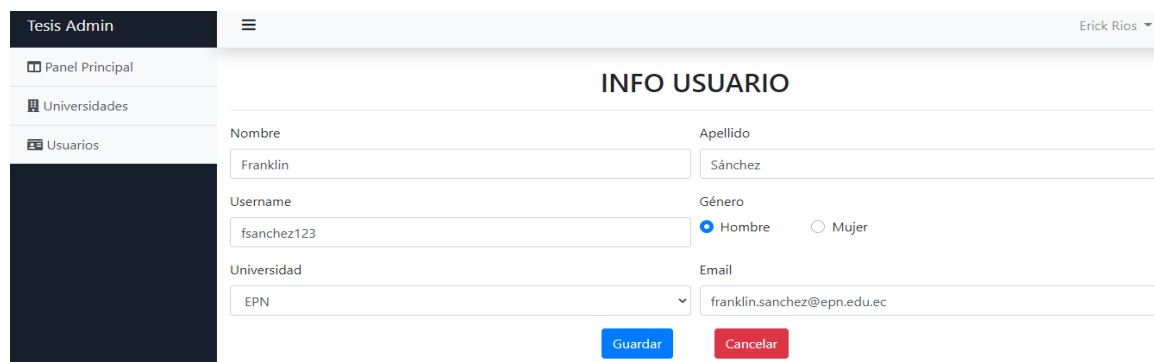


Figura 3.36. Formulario para modificar un usuario

INFO USUARIO

Nombre <input type="text" value="Franklin"/>	Apellido <input type="text" value="Sánchez"/>
Username <input type="text" value="frankS7285"/>	Género <input checked="" type="radio"/> Hombre <input type="radio"/> Mujer
Universidad <input type="text" value="EPN"/>	Email <input type="text" value="franklin.sanchez@epn.edu.ec"/>

Figura 3.37. Formulario con el usuario modificado

Para comprobar que el usuario se actualizó en el sistema web, se procede a verificar a través de MongoDB Compass y en la Figura 3.38 se presenta el usuario modificado en la base de datos.

```

_id: ObjectId("61b26bd81e4bc00163af74f")
name: "Franklin"
lastname: "Sánchez"
email: "franklin.sanchez@epn.edu.ec"
username: "frankS7285"
rol: "RESPONSABLE"
gender: "MALE"
university: "2"
password: "$2b$10$JVw0DYak34IGY10ETFQ.Wu.RqHGV6jXv4m7zos2IwKRpsYqw618PC"
birthday: "1980-12-19"
active: true
id: "8"
registerDate: "2021-12-09T20:49:28.177Z"

```

Figura 3.38. Usuario actualizado en la base de datos

Es importante mencionar que un usuario con rol Gestor solamente puede realizar operaciones de CRUD a usuarios con rol Responsable y Gestor. En el caso de que se quiera editar, bloquear o eliminar algún usuario con otro rol, se muestra una alerta en el que se especifica que la operación no se puede realizar como se indica en la Figura 3.39.

☰

! Solo es posible editar usuarios con rol RESPONSABLE y ADMIN

Erick Rios ▾

USUARIOS

ACTIVOS ▾
+ Añadir

ID	Nombre	Apellido	Username	Email	Universidad	Género	Rol	¿Activo?	Gestionar Información
4	Karen	Salazar	krios	karen.salazar@epn.edu.ec	2	FEMALE	COLABORADORP	✔	<div style="display: flex; gap: 5px;"> <div style="border: 1px solid #ccc; padding: 2px; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center;">✎</div> <div style="width: 20px; height: 20px; display: flex; align-items: center; justify-content: center;">ℹ</div> <div style="width: 20px; height: 20px; display: flex; align-items: center; justify-content: center;">🔒</div> <div style="width: 20px; height: 20px; display: flex; align-items: center; justify-content: center;">🗑</div> </div>

Figura 3.39. Alerta al tratar de editar un usuario

3.2.3.3 VER INFORMACIÓN DE USUARIOS

Para ver los datos de un usuario en especial, se da clic en el botón de información y se abre un cuadro de diálogo en el que se muestran los campos del usuario seleccionado como en la Figura 3.40.



Figura 3.40. Información del usuario seleccionado

3.2.3.4 BLOQUEAR/DESBLOQUEAR USUARIOS

Para activar o desactivar un usuario específico, se da clic en el botón de bloqueo y se muestra un cuadro de dialogo con un mensaje para el usuario en el que se especifican 2 botones para confirmar o cancelar la acción como se indica en la Figura 3.41.



Figura 3.41. Cuadro de diálogo para desactivar un usuario

En la Figura 3.42 se muestra la comprobación del usuario bloqueado en la que se evidencia el campo active en false mediante MongoDB Compass.

```
_id: ObjectId("61b26bd81e4bc00163af74f")
name: "Franklin"
lastname: "Sánchez"
email: "franklin.sanchez@epn.edu.ec"
username: "frank57285"
rol: "RESPONSABLE"
gender: "MALE"
university: "2"
password: "$2b$10$JVw0DYAk34IGY10ETFQ.WU.RqHGV6jXv4m7zos2IwKRsYqw6i8PC"
birthday: "1980-12-19"
active: false
id: "8"
registerDate: "2021-12-09T20:49:28.177Z"
```

Figura 3.42. Usuario bloqueado en la base de datos

3.2.3.5 ELIMINAR USUARIOS

Para eliminar un usuario específico, se da clic en el botón de eliminar y se muestra un cuadro de diálogo con un mensaje para el usuario en el que se especifican 2 botones para confirmar o cancelar la acción de eliminación, lo cual se lo indica en la Figura 3.43.

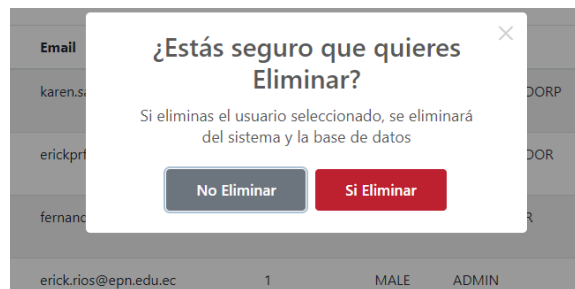


Figura 3.43. Cuadro de diálogo para eliminar un usuario

En la Figura 3.44 se comprueba que el usuario se eliminó al realizar una consulta con el correo del usuario eliminado y no obtener resultados en la búsqueda.

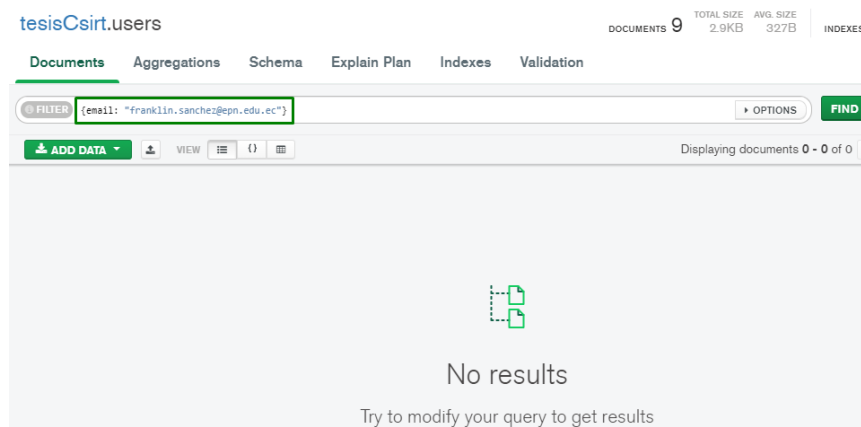


Figura 3.44. Usuario eliminado de la base de datos

3.3 PRUEBAS SPRINT 3: GESTIÓN RESPONSABLE

En este Sprint se indican las pruebas de funcionamiento con un usuario con rol Responsable para realizar la verificación del módulo Gestión Responsable.

3.3.1 PANEL PRINCIPAL

Una vez que el usuario con rol Responsable accede al sistema se muestra el panel principal como se indica en la Figura 3.45. Este panel muestra al Responsable, la cantidad de usuarios, universidades y documentos por estado que pertenecen a una universidad. Además, se tiene una barra lateral en la parte izquierda para acceder a la gestión de usuarios y documentos. Por otro lado, se visualiza en la parte superior derecha un enlace para acceder al panel Home, información de la sesión del usuario y una opción para salir del sistema web. Finalmente, se tiene un botón que permite ocultar la barra lateral para poder visualizar de mejor manera el panel principal como se muestra en la Figura 3.46.

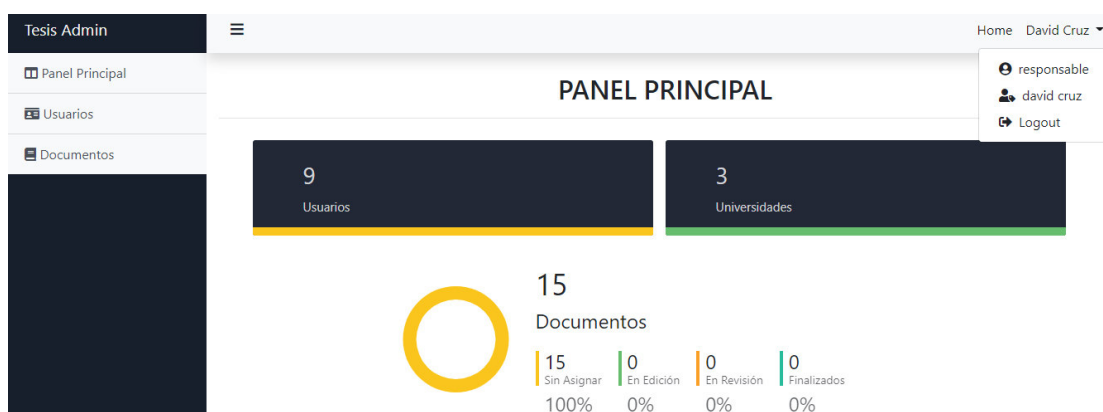


Figura 3.45. Panel principal del usuario responsable

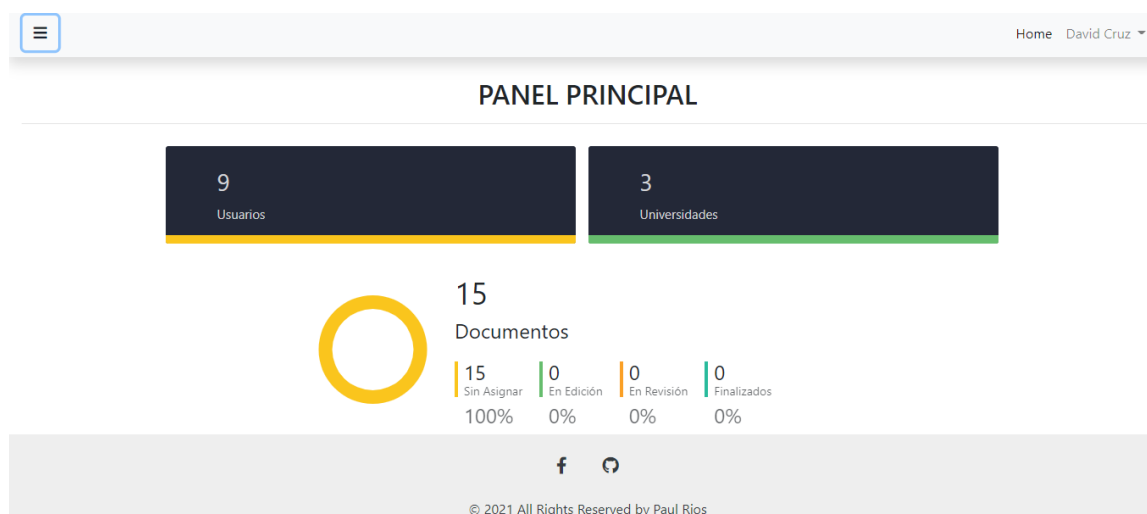
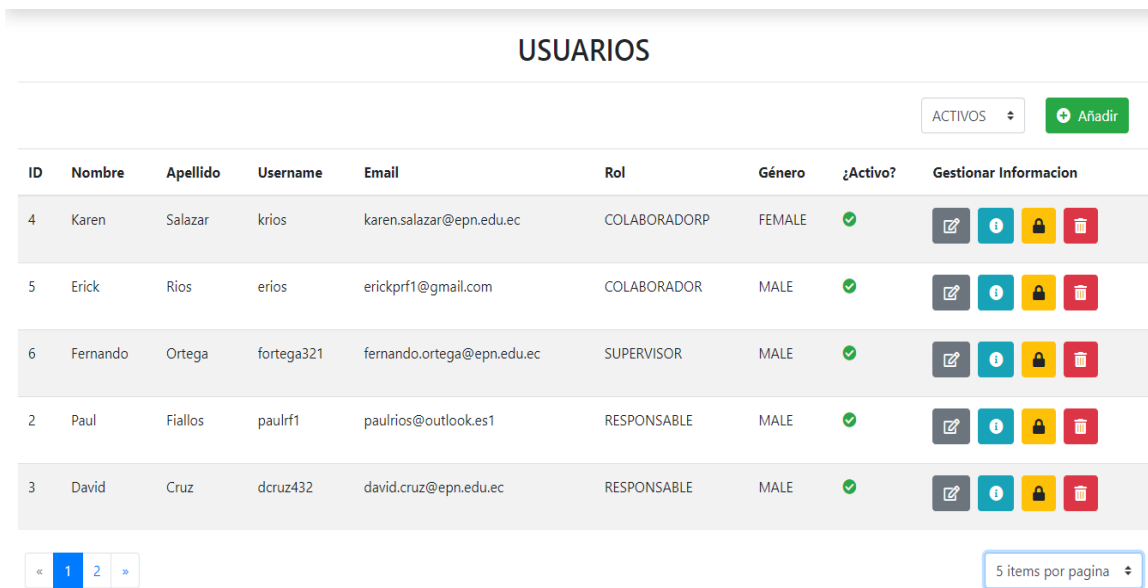


Figura 3.46. Barra lateral oculta en panel principal

3.3.2 PANEL USUARIOS

Al dar clic la opción Usuarios de la barra lateral, el usuario accede a un panel para poder realizar operaciones CRUD sobre los usuarios que hay en el sistema web como se muestra en la Figura 3.47. En este panel se puede visualizar información de los usuarios que pertenecen a una misma universidad mediante una tabla. Además, se puede seleccionar un filtro para visualizar los usuarios activos, inactivos o todos. Por otro lado, la tabla cuenta con sistema de paginación y se puede seleccionar que se visualicen 5, 10, 15 o 20 usuarios por página.























ID	Nombre	Apellido	Username	Email	Rol	Género	¿Activo?	Gestionar Información
4	Karen	Salazar	krios	karen.salazar@epn.edu.ec	COLABORADORP	FEMALE	✓	   
5	Erick	Rios	erios	erickprf1@gmail.com	COLABORADOR	MALE	✓	   
6	Fernando	Ortega	fortega321	fernando.ortega@epn.edu.ec	SUPERVISOR	MALE	✓	   
2	Paul	Fiallos	paulrf1	paulrios@outlook.es1	RESPONSABLE	MALE	✓	   
3	David	Cruz	dcruz432	david.cruz@epn.edu.ec	RESPONSABLE	MALE	✓	   

Figura 3.47. Panel responsable-crud-usuarios

3.3.2.1 CREAR USUARIOS

Para crear un usuario nuevo, se procede a dar clic en el botón Añadir y se muestra un formulario con los campos que se deben llenar para poder registrar un usuario con rol Responsable, Supervisor y Colaborador como se muestra en la Figura 3.48. Una vez realizado el registro se envía un correo electrónico al nuevo usuario con un enlace para que realice su activación y establezca una contraseña y su fecha de nacimiento.

Añadir Usuario

nombre

apellido

username

email

género

rol

Figura 3.48. Formulario para registro de un usuario

En caso de no ingresar todos los campos requeridos se muestran mensajes de error y no se permite realizar el registro hasta que todo se encuentre validado como se indica en la Figura 3.49.

Añadir Usuario

nombre

apellido

username

email

género

rol

Nombre es obligatorio
Apellido es obligatorio
Username es obligatorio
Email es obligatorio
Email no valido

Figura 3.49. Mensajes de error formulario crear usuario

Una vez ingresados los datos de manera correcta como se indica en la Figura 3.50, se procede a dar clic en el botón ok, se cierra el formulario y se regresa al panel de Usuarios, en donde se muestra un mensaje indicando si el registro del nuevo usuario fue exitoso o hubo problemas. Además, se envía el enlace de activación al correo electrónico del usuario nuevo como se muestra en la Figura 3.51.

Añadir Usuario

nombre

apellido

username

email

género

rol

Figura 3.50. Formulario crear usuario campos correctos

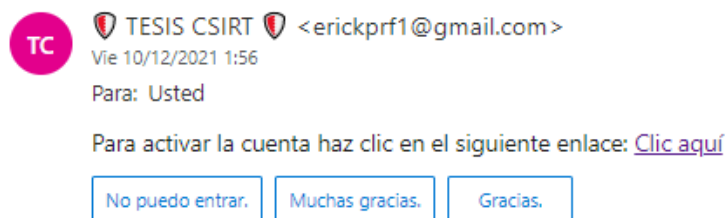


Figura 3.51. Enlace de activación enviado al nuevo usuario

Para comprobar que el usuario se registró en el sistema web, se procede a verificar en la base de datos a través de MongoDB Compass. En la Figura 3.52 se muestra el usuario registrado. Además, se tiene el campo active en false por defecto ya que el usuario debe realizar la activación mediante el enlace que se envió automáticamente al correo proporcionado. Los campos password y birthday se cambiarán al momento de acceder al formulario de activación.

```

_id: ObjectId("61b26bd81e4bcb00163af74f")
name: "Liliana"
lastname: "Córdova"
email: "liliana.cordova@epn.edu.ec"
username: "lc7285ER"
rol: "COLABORADOR"
gender: "FEMALE"
university: "2"
password: "$2b$10$JVw0DYAk34IGY10ETFQ.WU.RqHGV6jXv4m7zos2IWKRpsYqw6i8PC"
birthday: "2003-12-10"
active: false
id: "s"
registerDate: "2021-12-09T20:49:28.177Z"

```

Figura 3.52. Usuario registrado en la base de datos

3.3.2.2 EDITAR USUARIOS

Para editar un usuario, se procede a dar clic en el botón de editar y se muestra el mismo formulario de registro, pero con la información de los campos del usuario seleccionado.

De esta manera, se pueden realizar los cambios como se muestra en la Figura 3.53. Una vez realizados los cambios como se presenta en la Figura 3.54, se procede a dar clic en el botón guardar para actualizar los cambios en la base de datos.

Modificar Usuario

nombre

apellido

username

email

género

rol

Figura 3.53. Formulario para modificar un usuario

Modificar Usuario

nombre

apellido

username

email

género

rol

Figura 3.54. Formulario con el usuario modificado

Para comprobar que el usuario se actualizó en el sistema web, se procede a verificar a través de MongoDB Compass y en la Figura 3.55 se presenta el usuario modificado en la base de datos.

```
_id: ObjectId("617996d93488b2b9d911ce4c")
name: "Fernando"
lastname: "Ortega"
email: "fernando.ortega@epn.edu.ec"
username: "fortega1985"
rol: "SUPERVISOR"
gender: "MALE"
university: "2"
password: "$2b$10$o430DE.6.Oky1oQEaasYl04MfIFi8JUiaJXJFrd2vYbf91jqcOGPy"
birthday: "2003-10-16"
active: true
id: "6"
registerDate: "2021-10-27T05:11:04.911Z"
```

Figura 3.55. Usuario actualizado en la base de datos

3.3.2.3 VER INFORMACIÓN DE USUARIOS

Para ver los datos de un usuario en especial, se da clic en el botón de información y se abre un cuadro de diálogo en el que se muestran los campos del usuario seleccionado como en la Figura 3.56. Además, se tienen 2 botones para ir a la opción de editar el usuario o bloquearlo.



Figura 3.56. Información del usuario seleccionado

3.3.2.4 BLOQUEAR/DESBLOQUEAR USUARIOS

Para activar o desactivar un usuario específico, se da clic en el botón de bloqueo y se muestra un cuadro de dialogo con un mensaje para el usuario en el que se especifican 2 botones para confirmar o cancelar la acción como se indica en la Figura 3.57.



Figura 3.57. Cuadro de diálogo para bloquear un usuario

En la Figura 3.58 se muestra la comprobación del usuario desbloqueado en la que se evidencia el campo active en false mediante MongoDB Compass.

```
_id: ObjectId("617996d93488b2b9d911ce4c")
name: "Fernando"
lastname: "Ortega"
email: "fernando.ortega@epn.edu.ec"
username: "fortega1985"
rol: "SUPERVISOR"
gender: "MALE"
university: "2"
password: "$2b$10$o430DE.6.Oky1oQEaAsY104MfIFiBJuiaJXJFrd2vYbf91jqcOGPy"
birthday: "2003-10-16"
active: false
id: "6"
registerDate: "2021-10-27T05:11:04.911Z"
```

Figura 3.58. Usuario bloqueado en la base de datos

3.3.2.5 ELIMINAR USUARIOS

Para eliminar un usuario específico, se da clic en el botón de eliminar y se muestra un cuadro de dialogo con un mensaje para el usuario en el que se especifican 2 botones para confirmar o cancelar la acción de eliminación, lo cual se lo indica en la Figura 3.59.

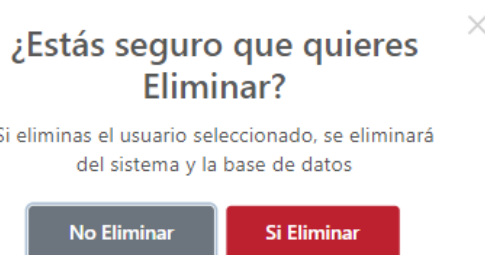


Figura 3.59. Cuadro de diálogo para eliminar un usuario

En la Figura 3.60 se comprueba que el usuario se eliminó al realizar una consulta con el correo del usuario eliminado y no obtener resultados en la búsqueda.

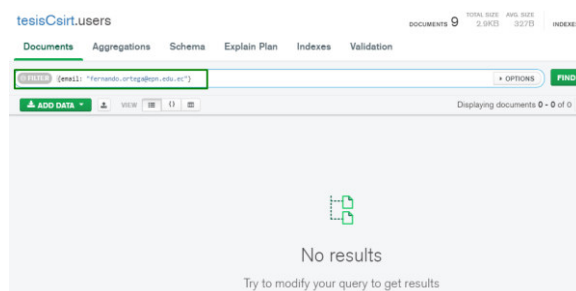


Figura 3.60. Usuario eliminado de la base de datos

3.3.3 PANEL DOCUMENTOS

Al dar clic la opción Documentos de la barra lateral, el usuario accede a un panel para poder asignar y ver información de los documentos que pertenecen a la universidad a la cual pertenece el usuario como se indica en la Figura 3.61. En este panel se puede visualizar información de los documentos mediante una tabla. Además, se puede seleccionar un filtro para visualizar los documentos sin asignar, asignados, revisados y finalizados o todos. Por otro lado, la tabla cuenta con sistema de paginación y se puede seleccionar que se visualicen 5, 10, 15 o 20 documentos por página.

ID	Nombre	Fecha de Entrega	Gestionar Información
1	POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN		
2	POLÍTICA DE PROTECCIÓN DE LA INFORMACIÓN		
3	POLÍTICA DE ELIMINACIÓN DE LA INFORMACIÓN		
4	POLÍTICA DE RETENCIÓN DE LA INFORMACIÓN		
5	POLÍTICA DE DIFUSIÓN DE LA INFORMACIÓN		

Figura 3.61. Panel responsable-crud-universidades

3.3.3.1 ASIGNAR DOCUMENTOS

Para asignar un documento a los usuarios de la universidad y así puedan realizar cambios, se procede dar clic en el botón de edición y se muestra el formulario como se indica en la Figura 3.62. Este formulario contiene selectores para escoger a los usuarios que trabajarán en el documento y su fecha de entrega.

ASIGNAR DOCUMENTO

DOCUMENTO
POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN

DESCRIPCION
Este documento es necesario para establecer las políticas de clasificación de la información

STATUS
UNASSIGNED x

FECHA DE ENTREGA
Escoja una fecha

COLABORADOR PRINCIPAL 1
escoja un ítem

COLABORADORES
escoja un ítem

SUPERVISOR
escoja un ítem

Actualizar Regresar

Figura 3.62. Formulario asignación documentos

Una vez realizadas las asignaciones como se muestra en la Figura 3.63, se procede a dar clic en el botón actualizar y se mostrará una alerta para continuar o no con la asignación como se presenta en la Figura 3.64.

ASIGNAR DOCUMENTO

DOCUMENTO
POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN

DESCRIPCION
Este documento es necesario para establecer las políticas de clasificación de la información

STATUS ASSIGNED x **FECHA DE ENTREGA** 2022-01-31

COLABORADOR PRINCIPAL 1 Karen x **COLABORADORES** Erick x Liliana x

SUPERVISOR Fernando x

Actualizar Regresar

Figura 3.63. Formulario asignación documentos lleno

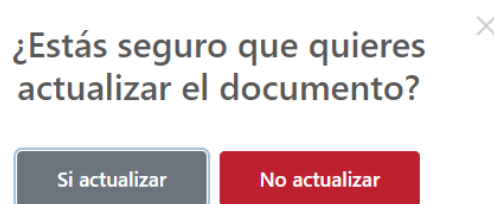


Figura 3.64. Alerta para asignar el documento

Una vez realizada la asignación se puede verificar que el documento se actualiza en la base de datos como se indica en la Figura 3.65.

```
_id: ObjectId("5ffd54ecbbc12502d8c90632")
id: "1"
name: "POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN"
description: "Este documento es necesario para establecer las políticas de clasifica..."
status: "ASSIGNED"
deliveryDate: "2022-01-31"
registerDate: "2021-11-22"
usersId: Array
  0: "2"
  1: "4"
  2: "5"
  3: "8"
  4: "6"
itemsId: Array
  0: "1"
  1: "2"
  2: "3"
  3: "4"
  4: "5"
  5: "6"
  6: "7"
universityId: "2"
```

Figura 3.65. Documento actualizado en la base de datos

3.3.3.2 VER INFORMACIÓN DE DOCUMENTOS

Para ver los datos de un documento particular, se da clic en el botón de información y se abre un cuadro de diálogo en el que se muestran ciertos campos del documento seleccionado como en la Figura 3.66. Además, se tiene 1 botón para ir a la opción de asignar documento.



Figura 3.66. Información del documento seleccionado

3.4 PRUEBAS SPRINT 4: GESTIÓN COLABORADORES

En el Sprint 4 se manifiestan las pruebas con un usuario con rol Colaborador para verificar el correcto funcionamiento del módulo Gestión Colaboradores. Como se indica en la Figura 3.67, se tiene un formulario con los ítems del documento y un panel de cambios al cual se accede al dar clic en el ícono ver.

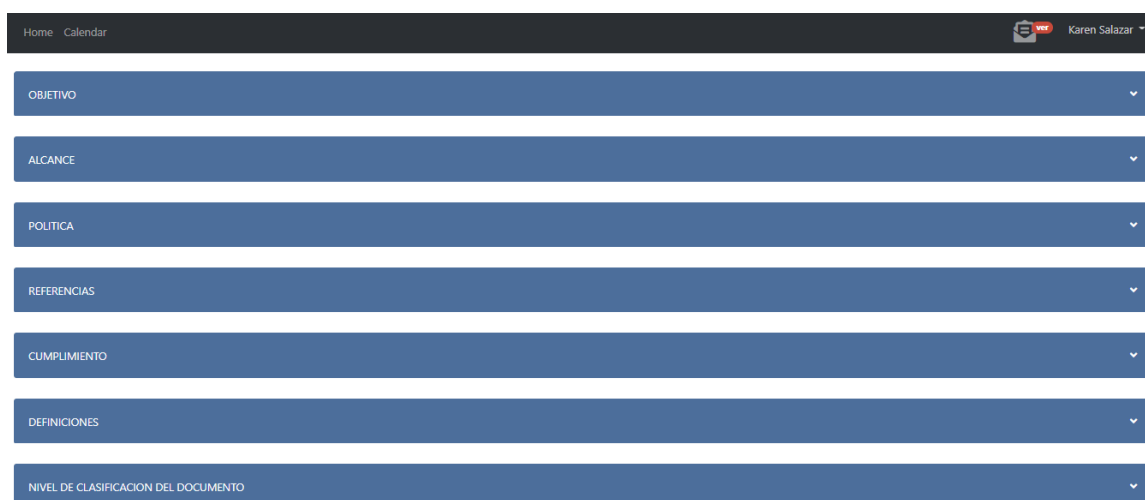


Figura 3.67. Formulario edición-documento-colaborador

3.4.1 EDITAR CONTENIDO DE UN DOCUMENTO

Para editar un ítem específico, se debe dar clic sobre el ítem e inmediatamente se abre un editor de texto como se muestra en la Figura 3.68. Este editor contiene en la parte superior opciones para estilo, color, tamaño, alineación y tipo de letra. Además, se pueden incluir enlaces, tablas, citas y operaciones para rehacer o deshacer un cambio realizado. Por otro lado, en la parte inferior se tienen botones para guardar o cancelar un cambio. Además, de un botón para crear una nueva versión del ítem. Finalmente se tiene un selector para mostrar en el editor el contenido de la versión actual del ítem.

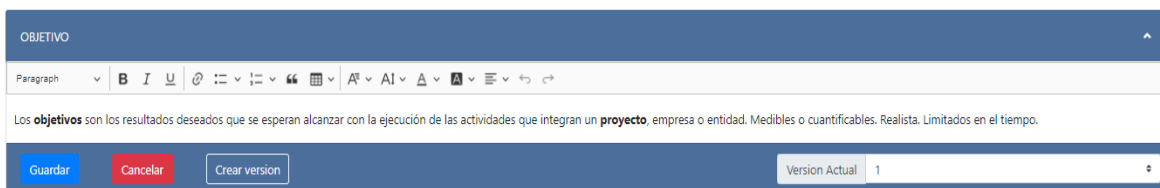


Figura 3.68. Edición ítem de un documento

Por defecto cada ítem tiene creada una versión, sin embargo, se pueden ir creando varias versiones según las necesidades del usuario. Al dar clic en Crear versión se muestra una alerta con 2 opciones como se indica en la Figura 3.69. Luego, al dar clic en agregar, la nueva versión se crea y se la puede ver en el editor de texto al seleccionarla en las opciones del selector como en la Figura 3.70.

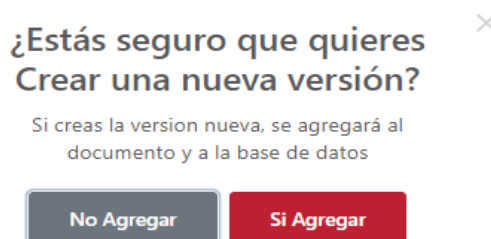


Figura 3.69. Alerta crear versión

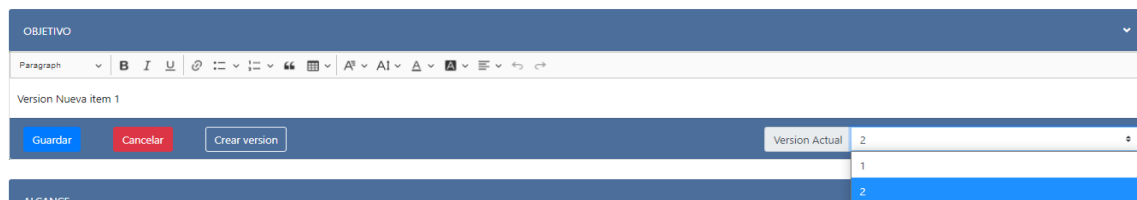


Figura 3.70. Nueva versión creada de un ítem

Para verificar que la versión se creó de manera correcta en la base de datos, se procede a realizar una consulta en MongoDB Compass con el resultado indicado en **la Figura 3.71.**

```
_id: ObjectId("61b6ed5570f67e0016c9f7b9")
itemId: "1"
text: "Version Nueva item 1"
userId: "4"
active: false
updateDate: "2021-12-13"
id: "124"
registerDate: "2021-12-13T06:51:01.521Z"
```

Figura 3.71. Nueva versión en la base de datos

Esta nueva versión creada se encuentra por defecto inactiva y para establecer dicha versión como la versión actual del ítem es necesario dar clic en el botón guardar del editor de texto como se indica en la Figura 3.72.

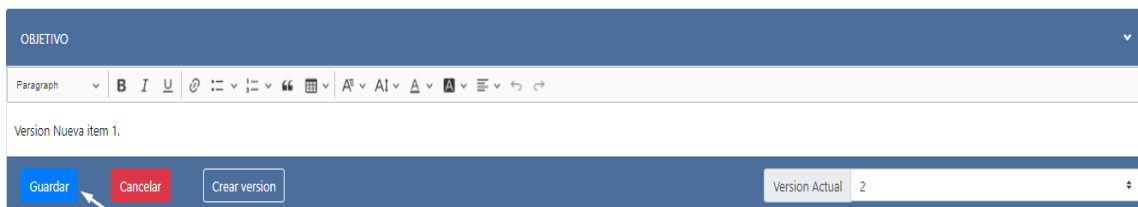


Figura 3.72. Establecer la versión actual del ítem

En la Figura 3.73 se indican las 2 versiones que pertenecen al primer ítem del documento en la base de datos y se puede verificar que la primera versión se encuentra inactiva, mientras que la segunda se encuentra activa como la versión actual del ítem.

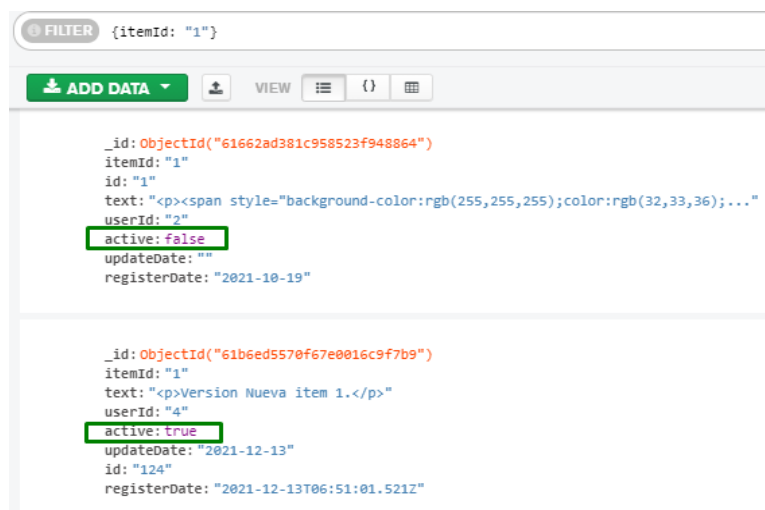


Figura 3.73. Versiones del primer ítem del documento en BD

3.4.2 PANEL CAMBIOS DEL DOCUMENTO

Para visualizar los cambios pendientes especificados por el usuario con Rol Supervisor como se indica en la Figura 3.74, es necesario dar clic en el botón Ver de la parte superior del sistema web.

ID	Nombre	Descripción	Fecha de creación	Finalizar
1	Cambiar Marco Teórico	Realizar el cambio del marco teórico	2021-09-23T02:21:40.632Z	<input checked="" type="checkbox"/> Listo
2	cambiar Definiciones	Se debe mejorar las definiciones	2021-09-23T02:22:28.370Z	<input checked="" type="checkbox"/> Listo

Figura 3.74. Cambios pendientes

Una vez finalizados los cambios indicados, se procede a dar clic en el botón Listo para que los cambios se establezcan en el estado Realizado como se especifica en la Figura 3.75.

ID	Nombre	Descripción	Fecha de creación	Estado
1	Cambiar Marco Teórico	Realizar el cambio del marco teórico	2021-09-23T02:21:40.632Z	Realizado
2	cambiar Definiciones	Se debe mejorar las definiciones	2021-09-23T02:22:28.370Z	Realizado

Figura 3.75. Cambios realizados

3.4.3 CAMBIAR ESTADO DOCUMENTO

Luego de haber finalizado con la edición de todos los ítems del documento, se procede a dar clic en el botón Enviar, el cual mostrará una alerta al usuario para continuar o cancelar la operación como se muestra en la Figura 3.76. Así el documento cambia de estado de asignado a revisado para que el usuario con rol Supervisor lo pueda verificar.

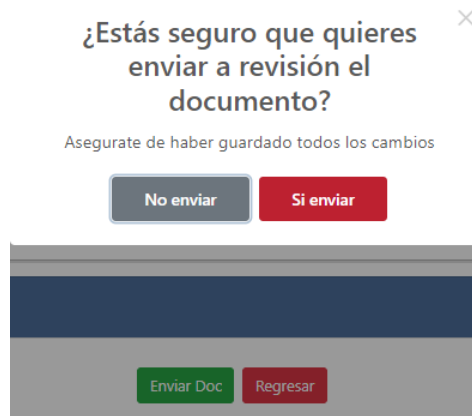


Figura 3.76. Alerta enviar documento a Módulo Supervisor

En la Figura 3.77, se puede verificar que el documento cambió de estado en la base de datos de asignado a revisado.

```

    _id: ObjectId("5ffd54ecbbc12502d8c90632")
    id: "1"
    name: "POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN"
    description: "Este documento es necesario para establecer las políticas de clasifica..."
    status: "REVIEWED"
    deliveryDate: "2022-01-31"
    registerDate: "2021-11-22"
    usersId: Array
      0: "2"
      1: "4"
      2: "5"
      3: "8"
      4: "6"
    itemsId: Array
      0: "1"
      1: "2"
      2: "3"
      3: "4"
      4: "5"
      5: "6"
      6: "7"
    universityId: "2"
  
```

Figura 3.77. Documento actualizado en la base de datos

3.5 PRUEBAS SPRINT 5: GESTIÓN SUPERVISORES

En el Sprint 5 se hicieron pruebas con un usuario con rol Supervisor para la comprobación del correcto funcionamiento del módulo Gestión Supervisores. En la Figura 3.78, se indica un formulario con los ítems del documento y un panel para la gestión de los cambios que se muestra al dar clic en el ícono ver.

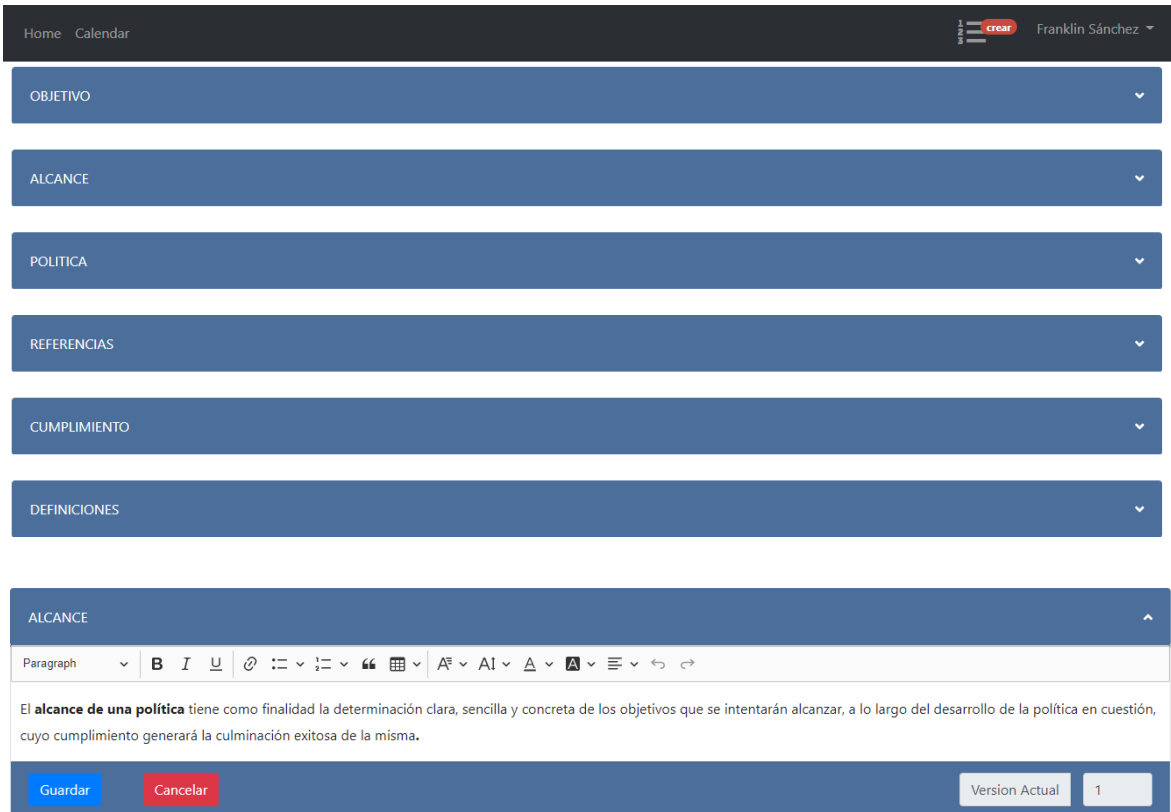


Figura 3.78. Formulario edición-documento-supervisor

3.5.1 EDITAR CONTENIDO DE UN DOCUMENTO

Para editar un ítem específico, se debe dar clic sobre el ítem e inmediatamente se abre un editor de texto similar al del módulo del usuario Responsable con la diferencia de que solo se puede editar el contenido de la versión actual del ítem seleccionado como se muestra en la Figura 3.79.

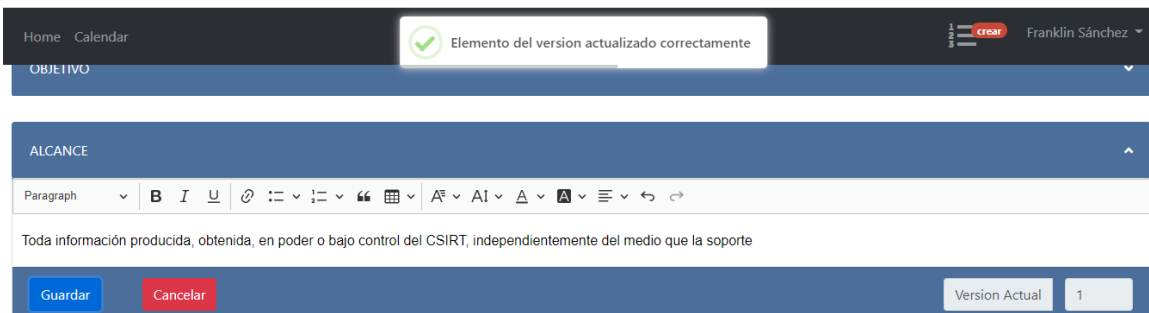


Figura 3.79. Edición del ítem seleccionado

Para verificar que el ítem se actualizó de manera correcta se procede a realizar una consulta en MongoDB como se muestra en la Figura 3.80, en la que se visualiza el ítem actualizado.

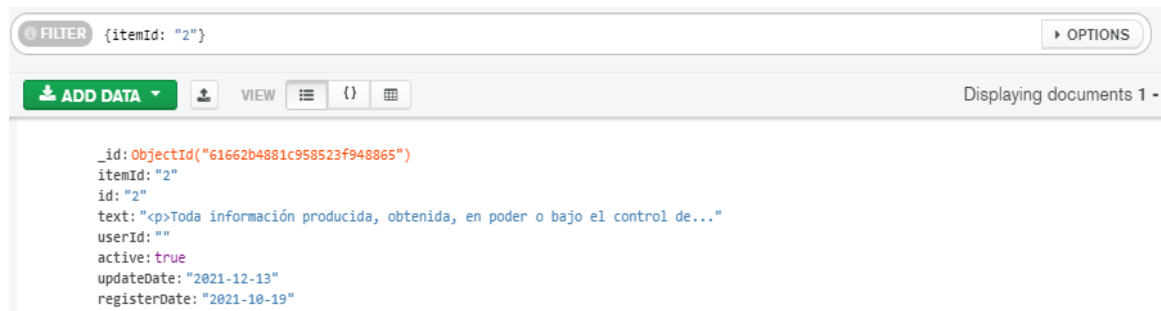


Figura 3.80. Versión del ítem actualizado en la base de datos

3.5.2 PANEL CAMBIOS

Para visualizar el panel de cambios y realizar operaciones CRUD como se muestra en la Figura 3.81, se debe dar clic en el botón Ver de la parte superior del sistema web. Este panel consta de un formulario con 2 campos para agregar o editar un cambio y una tabla con la lista de cambios de un documento particular.

COMENTARIOS

Nombre
Ingrese el nombre

Descripción
Ingrese la descripción

Agregar

ASIGNADOS

No tiene cambios creados aún. Agregue uno

Figura 3.81. Panel cambios-documento-supervisor

3.5.2.1 CREAR CAMBIOS

Para crear un cambio nuevo se deben llenar los 2 campos del Formulario que son obligatorios y dar clic en el botón Agregar como se indica en la Figura 3.82.

COMENTARIOS

Nombre
Cambiar Marco Teórico

Descripción
Mejorar los conceptos teóricos para comprender de mejor manera el desarrollo de la Política de Clasificación de la Información.

Agregar

Figura 3.82. Formulario para crear nuevo cambio

3.5.2.2 LISTA DE CAMBIOS

En la Figura 3.83 se muestra la lista de cambios que pertenecen a un documento específico. Esta lista se la puede filtrar por los estados asignados, revisados, finalizados o todos. Además, cada cambio tiene asignados 3 botones para editar, aprobar o eliminar el cambio de la base de datos.

The screenshot shows a form titled 'COMENTARIOS' with a close button (X). It contains two text input fields: 'Nombre' (with placeholder 'Ingrese el nombre') and 'Descripción' (with placeholder 'Ingrese la descripción'). Below these is a blue 'Agregar' button and a dropdown menu labeled 'ASIGNADOS'. At the bottom is a table with the following data:




ID	Nombre	Descripción	Creado hace	Estado	Gestionar Información
1	Cambiar Marco Teórico	Mejorar los conceptos teóricos para comprender de mejor manera el desarrollo de la Política de Clasificación de la Información.	2 days ago	ASSIGNED	  

Figura 3.83. Lista de cambios

3.5.2.3 EDITAR CAMBIOS

Al dar clic en el botón editar se muestra el contenido del cambio, se procede a editar los campos y dar clic en el botón agregar como se indica en la Figura 3.84.

The screenshot shows the 'COMENTARIOS' form in edit mode. The 'Nombre' field contains 'Cambiar Marco Teórico' and the 'Descripción' field contains 'Mejorar los conceptos teóricos para comprender de mejor manera el desarrollo de la política.'. At the bottom are two buttons: a blue 'Agregar' button and a red 'Cancelar' button.

Figura 3.84. Edición de un cambio

En la Figura 3.85 se verifica el contenido del cambio actualizado en la base de datos.

```

    "_id": {
      "$oid": "614be4b45e1abc31a0c52f22"
    },
    "name": "Cambiar Marco Teórico",
    "description": "Mejorar los conceptos teóricos para comprender de mejor manera el desarrollo de la Política de Clasificación de la Información.",
    "supervisorId": "4",
    "colaboratorId": "3",
    "documentId": "1",
    "status": "ASSIGNED",
    "deliveryDate": "2021-11-11",
    "id": "1",
    "registerDate": "2021-12-11T02:21:40.632Z"
  }
}

```

Figura 3.85. Cambio actualizado en la base de datos

3.5.2.4 APROBAR CAMBIOS

Para aprobar cambios realizados por los usuarios Colaboradores como se indica en la Figura 3.86, se procede a dar clic en el botón de Aprobar y se muestra una alerta para continuar o no con el proceso de aprobación.



Figura 3.86. Aprobar un cambio

En la Figura 3.87 se verifica el cambio con el estado actualizado de revisado a finalizado en la base de datos.

```

    "_id": {
      "$oid": "614be4b45e1abc31a0c52f22"
    },
    "name": "Cambiar Marco Teórico",
    "description": "Mejorar los conceptos teóricos para comprender de mejor manera el desarrollo de la política.",
    "supervisorId": "4",
    "colaboratorId": "3",
    "documentId": "1",
    "status": "FINISHED",
    "deliveryDate": "2021-11-11",
    "id": "1",
    "registerDate": "2021-12-11T02:21:40.632Z"
  }
}

```

Figura 3.87. Cambio finalizado en la base de datos

3.5.2.5 ELIMINAR CAMBIOS

Para eliminar un cambio, se procede a dar clic en el botón de eliminar y se muestra una alerta para continuar o no con el proceso de eliminación como se indica en la Figura 3.88.

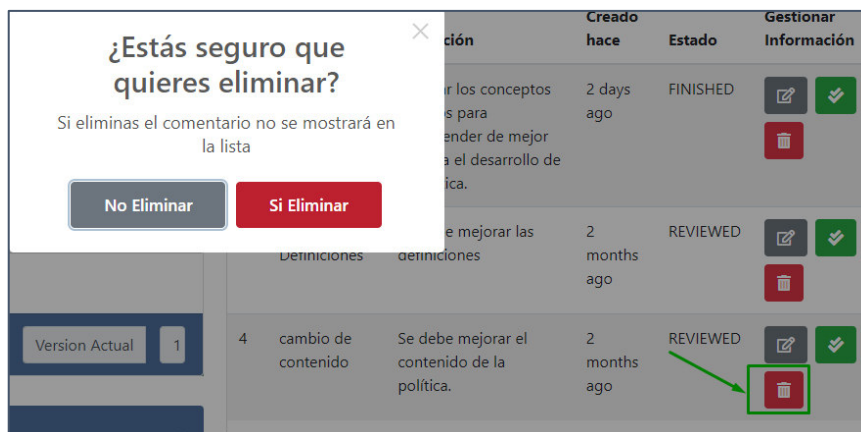


Figura 3.88. Eliminar un cambio

En la Figura 3.89 se verifica el cambio eliminado al realizar una consulta por Id y no encontrarlo en la base de datos.

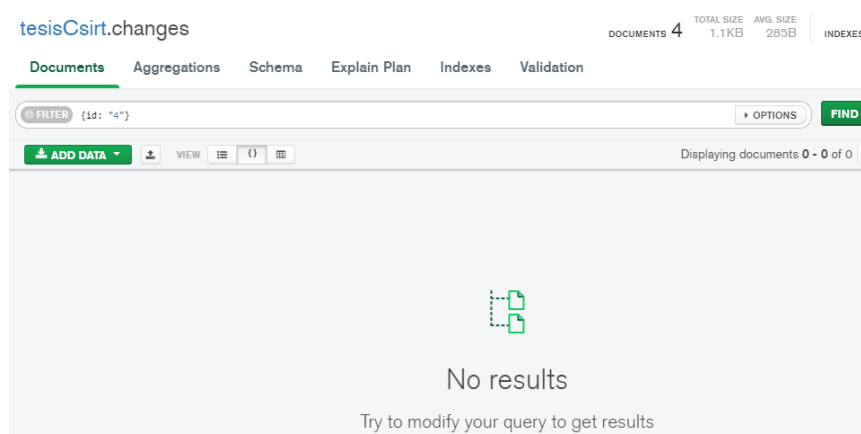


Figura 3.89. Cambio eliminado de la base de datos

3.6 PRUEBAS SPRINT 6: GENERACIÓN DE DOCUMENTOS FINALES

En este Sprint, se procedió a verificar el correcto funcionamiento del módulo Generación de Documentos Finales con un usuario con rol Responsable. Este módulo como se indica en la Figura 3.90 es accesible por cualquier usuario que pertenezca a la misma universidad y permite generar y descargar un documento en formato PDF por medio de un botón.

Home Calendar Admin David Cruz

POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN

OBJETIVO
El alcance de una política tiene como finalidad la determinación clara, sencilla y concreta de los objetivos que se intentarán alcanzar, a lo largo del desarrollo de la política en cuestión, cuyo cumplimiento generará la culminación exitosa de la misma.

ALCANCE
Descripción de la funcionalidad de la política.

POLITICA
Conjunto de elementos suficientemente detallados que permiten la identificación de la fuente de la cual se extrae la información. Al redactar una referencia bibliográfica, se deben considerar y anotar todos los elementos bibliográficos que permitan identificar en forma clara y precisa lo consultado, de acuerdo al autor citado.

REFERENCIAS
Detallar las acciones realizadas para poder cumplir con la realización de la política.

CUMPLIMIENTO
Conceptos que permitan la comprensión clara e inequívoca de la política por parte de los lectores.

DEFINICIÓN	SIGNIFICADO
DEFINICIONES	
Color	¿Cuándo utilizar?
ROJO	Cuando la información está limitada a personas concretas, debido a que su difusión a terceras personas podría tener un impacto en la seguridad

INTEGRANTES

- Paul Fiallos paulfios@outlook.es RESPONSABLE
- Karen Salazar karen.salazar@epn.edu.ec COLABORADOR P
- Erick Ríos erickprf1@gmail.com COLABORADOR
- Fernando Ortega fernando.ortega@epn.edu.ec1 SUPERVISOR
- Liliana Córdova liliana.cordova@epn.edu.ec COLABORADOR

FECHA DE ENTREGA
📅 2022-01-31

Download PDF

Figura 3.90. Panel generación de documentos finales

Al dar clic en el botón Download PDF se abre una ventana emergente en la que se puede modificar los márgenes del documento, además de agregar encabezado y pie de página como se indica en la Figura 3.91.

POLÍTICA DE CLASIFICACIÓN DE LA INFORMACIÓN - Google Chr... 1 página

about:blank

Imprimir

Destino: Guardar como PDF

Páginas: Todo

Páginas por hoja: 1

Márgenes: Predeterminados

Opciones:

- Encabezado y pie de página
- Gráficos de fondo

Guardar Cancelar

Figura 3.91. Ventana emergente para descargar documento

Una vez dado clic en Guardar, se abre una ventana en la que se puede cambiar el nombre del documento y la ubicación donde se almacenará dicho documento como se indica en la Figura 3.92.

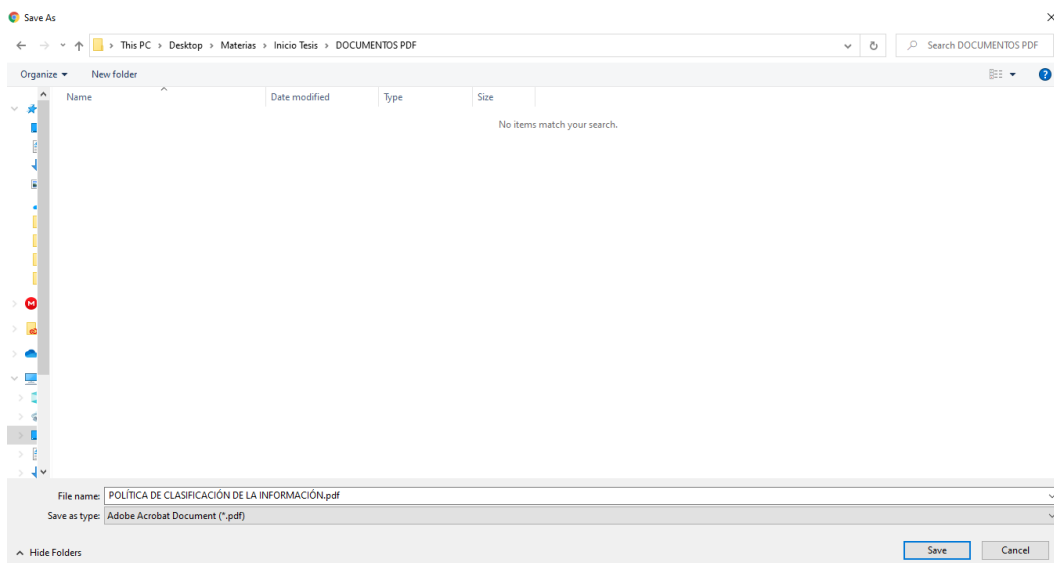


Figura 3.92. Ubicación del documento PDF a descargar.

Finalmente se descarga el documento en formato PDF y al abrirlo se puede visualizar el contenido generado como se indica en la Figura 3.93.



Figura 3.93. Contenido PDF descargado

3.7 PRUEBAS SPRINT 7: GESTIÓN PANEL HOME

Para la comprobación del módulo Gestión Home, se presentan las pruebas realizadas con uno de los usuarios con rol Responsable.

3.7.1 PANEL HOME

El Panel Home es visible para cualquier usuario que pertenezca a la misma universidad como se visualiza en la Figura 3.94. Además, en la parte izquierda se visualizan los documentos que están pendientes por realizar y en la parte derecha el cuadro de estados de tareas.



Figura 3.94. Panel Home

Los usuarios con rol Colaborador son los únicos que pueden dar clic en el botón Ir de la columna Editar y acceder al módulo para editar el documento mientras que los usuarios con rol Supervisor son los únicos usuarios que pueden dar clic en el botón Ir de la columna Revisar y acceder a editar el documento. Por otro lado, cualquier usuario de la misma universidad puede acceder a generar el documento PDF al dar clic en el botón Ir de la columna Finalizar. En el caso de que se quiera acceder a una columna que no se tenga los debidos permisos, se muestran alertas al usuario como en la Figura 3.95.

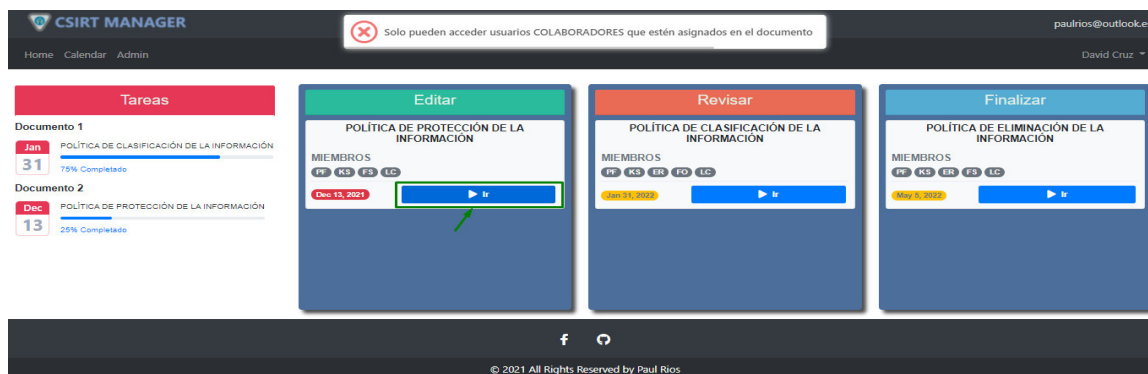


Figura 3.95. Alertas al acceder a un documento particular

3.7.2 VISUALIZACIÓN CALENDARIO

Al dar clic en el enlace Calendar de la barra superior del sistema web, se muestra un calendario con todas las fechas de entrega de los documentos como se indica en la Figura 3.96.

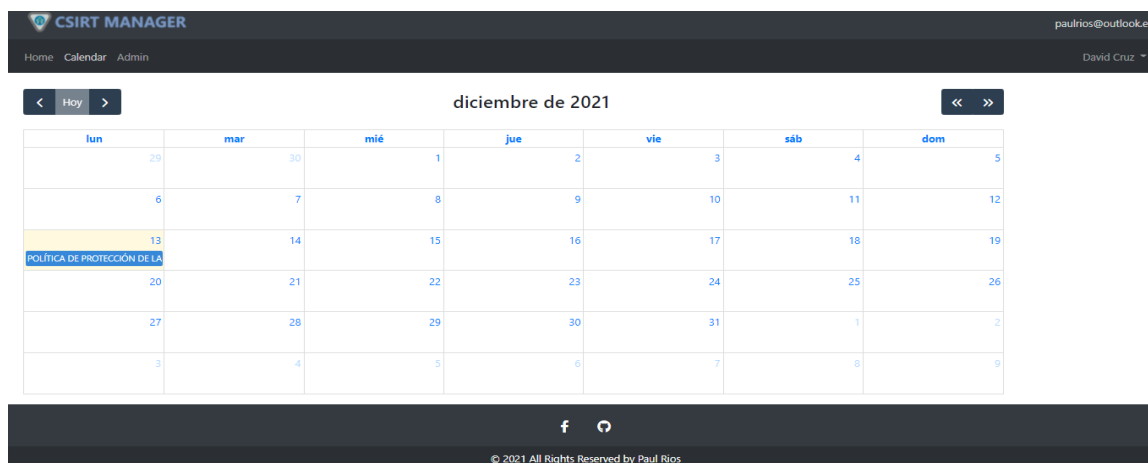


Figura 3.96. Calendario con las fechas de entrega de los documentos

En el caso de que se requiera visualizar la información de un documento con mayor detalle, se debe dar clic sobre el nombre del documento y se presenta mediante un cuadro de diálogo dicha información como se indica en la Figura 3.97.



Figura 3.97. Información del documento en el calendario

3.8 FORMULACIÓN DE LAS PREGUNTAS DE LAS ENCUESTAS

Para el análisis de los resultados, se formularon 30 preguntas que se presentan en la Tabla 3.2 y se dividen en 4 secciones. La primera sección permite validar la funcionalidad de los diferentes módulos del sistema de acuerdo con las operaciones que un usuario realiza en base al rol individual que desempeñan. En la segunda sección se verifica el

diseño visual de la interfaz gráfica orientándose en parámetros como colores, tipografía, contenido y lenguaje. Por otro lado, en la tercera sección se comprueba el funcionamiento general del sistema al establecer criterios como facilidad de uso, rapidez y operaciones realizadas por los usuarios. Finalmente, la cuarta sección permite identificar los inconvenientes que los usuarios presentaron durante la navegabilidad en el sistema.

Tabla 3.2. Preguntas formuladas para las encuestas

CÓDIGO	PREGUNTA	CRITERIO DE EVALUACIÓN	TIPO DE PREGUNTA
1P	Como usuario con rol Responsable, Colaborador o Supervisor, ¿usted pudo registrarse en el sistema web de manera correcta?	Sí - No	Cerrada
2P	¿Usted pudo restablecer su contraseña en caso de pérdida u olvido de la misma de manera correcta?	Sí - No	Cerrada
3P	¿Usted pudo realizar el inicio de sesión en el sistema web de manera correcta?	Sí - No	Cerrada
4P	¿Usted pudo cerrar su sesión de usuario manera correcta?	Sí - No	Cerrada
5P	Como usuario con rol Gestor, ¿usted pudo visualizar correctamente el panel Gestor?	Sí - No	Cerrada
6P	Como usuario con rol Gestor, ¿usted pudo crear, editar, ver, bloquear, desbloquear y eliminar universidades?	Sí - No	Cerrada
7P	Como usuario con rol Gestor, ¿usted pudo crear, editar, ver, bloquear, desbloquear y eliminar usuarios con rol Responsable?	Sí - No	Cerrada
8P	Como usuario con rol Responsable, ¿usted pudo visualizar correctamente el panel Responsable?	Sí - No	Cerrada
9P	Como usuario con rol Responsable, ¿usted pudo crear, editar, ver, bloquear, desbloquear y eliminar usuarios que pertenecen a su misma universidad?	Sí - No	Cerrada
10P	Como usuario con rol Responsable, ¿usted pudo ver y asignar documentos a los diferentes usuarios que pertenecen a su misma universidad?	Sí - No	Cerrada
11P	Como usuario con rol Supervisor, ¿usted pudo acceder a los documentos que tenía asignados para realizar la comprobación?	Sí - No	Cerrada
12P	Como usuario con rol Supervisor, ¿usted pudo editar el contenido de los documentos a los cuales fue asignado?	Sí - No	Cerrada
13P	Como usuario con rol Supervisor, ¿usted pudo ver, crear, editar y eliminar cambios en los documentos a los cuales fue asignado?	Sí - No	Cerrada
14P	Como usuario con rol Supervisor, ¿usted pudo enviar los documentos que supervisó a la columna FINALIZAR?	Sí - No	Cerrada

15P	Como usuario con rol Colaborador, ¿usted pudo acceder a los documentos que tenía asignados para editarlos?	Sí - No	Cerrada
16P	Como usuario con rol Colaborador, ¿usted pudo crear y editar versiones de ítems de un documento?	Sí - No	Cerrada
17P	Como usuario con rol Colaborador, ¿usted pudo visualizar y finalizar los cambios asignados por el usuario Supervisor?	Sí - No	Cerrada
18P	Como usuario con rol Colaborador, ¿usted pudo enviar los documentos realizados a la columna REVISAR?	Sí - No	Cerrada
19P	Como usuario con rol Colaborador, Supervisor o Responsable, ¿usted pudo visualizar el panel Home de manera correcta?	Sí - No	Cerrada
20P	Como usuario con rol Colaborador, Supervisor o Responsable, ¿usted pudo visualizar los documentos finalizados con la información de los usuarios asignados?	Sí - No	Cerrada
21P	Como usuario con rol Colaborador, Supervisor o Responsable, ¿usted pudo generar y descargar documentos en formato PDF?	Sí - No	Cerrada
22P	Como usuario con rol Colaborador, Supervisor o Responsable, ¿usted pudo visualizar las fechas de entrega de los documentos en el calendario?	Sí - No	Cerrada
23P	¿Usted cree que el lenguaje que se utilizó en el sistema web fue claro y entendible?	Opción múltiple <ul style="list-style-type: none"> • Totalmente en desacuerdo • En desacuerdo • De acuerdo • Totalmente de acuerdo 	Cerrada
24P	¿Usted considera que los tipos y tamaños de letra usados en el sistema web, así como los colores fueron los adecuados?	Opción múltiple <ul style="list-style-type: none"> • Totalmente en desacuerdo • En desacuerdo • De acuerdo • Totalmente de acuerdo 	Cerrada
25P	¿Usted considera que la forma en cómo está estructurado el contenido el sistema web fue el correcto?	Opción múltiple <ul style="list-style-type: none"> • Totalmente en desacuerdo • En desacuerdo • De acuerdo • Totalmente de acuerdo 	Cerrada
26P	En la escala del 1 al 5, ¿cómo calificaría la interfaz gráfica del sistema web?	Escala de satisfacción <ol style="list-style-type: none"> 1. Muy malo 2. Malo 3. Regular 4. Bueno 5. Muy bueno 	Cerrada
27P	¿Usted considera que el sistema web fue fácil de usar?	Opción múltiple <ol style="list-style-type: none"> 1. Totalmente en desacuerdo 	Cerrada

		2. En desacuerdo 3. De acuerdo 4. Totalmente de acuerdo	
28P	¿Usted considera que el sistema web fue generalmente rápido al momento de usarlo?	Opción múltiple 1. Totalmente en desacuerdo 2. En desacuerdo 3. De acuerdo 4. Totalmente de acuerdo	Cerrada
29P	¿Usted presentó algún inconveniente al momento de usar el sistema web? En el caso de que su respuesta sea afirmativa, por favor especifique.	Problemas de interacción usuario-sistema web	Abierta
30P	En la escala del 1 al 5, ¿cómo calificaría el funcionamiento del sistema web en general?	Escala de satisfacción 1. Muy malo 2. Malo 3. Regular 4. Bueno 5. Muy bueno	Cerrada

3.9 RESULTADOS DE LAS ENCUESTAS

Mediante el uso del software para gestionar formularios, Google Forms, se enviaron las encuestas a los diferentes usuarios que realizaron las pruebas en el sistema web (las encuestas se encuentran en el anexo F) y los resultados de las preguntas formuladas se presentan a continuación.

3.9.1 RESULTADOS DE FUNCIONALIDAD POR ROL DE USUARIO

3.9.1.1 RESULTADOS USUARIO CON ROL GESTOR

La Figura 3.98 muestra los resultados de las preguntas (5P, 6P y 7P) destinadas a los usuarios con rol Gestor. El 100% de los usuarios indicó que logró realizar satisfactoriamente la visualización del panel Gestor, las operaciones CRUD de universidades y las operaciones CRUD de los usuarios permitidos.

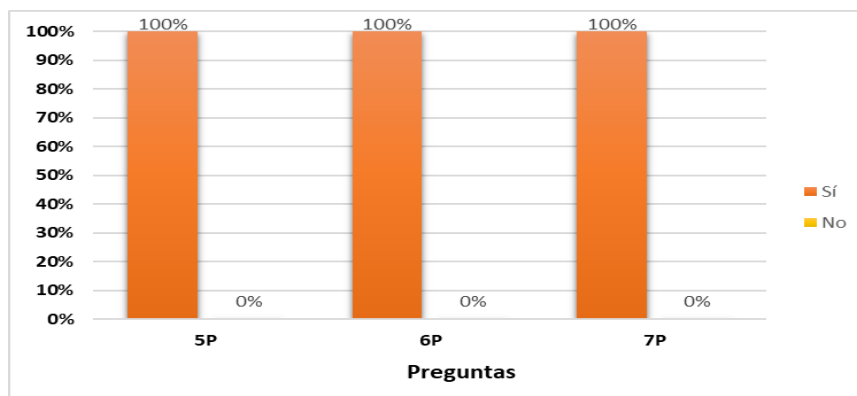


Figura 3.98. Resultados preguntas para usuario con rol Gestor

3.9.1.2 RESULTADOS USUARIO CON ROL RESPONSABLE

La Figura 3.99 muestra los resultados de las preguntas (8P, 9P y 10P) destinadas a los usuarios con rol Responsable. El 100% de los usuarios indicó que realizó satisfactoriamente las siguientes actividades: Visualización del panel Responsable, operaciones CRUD de los usuarios permitidos y asignación de documentos a usuarios de una misma universidad.

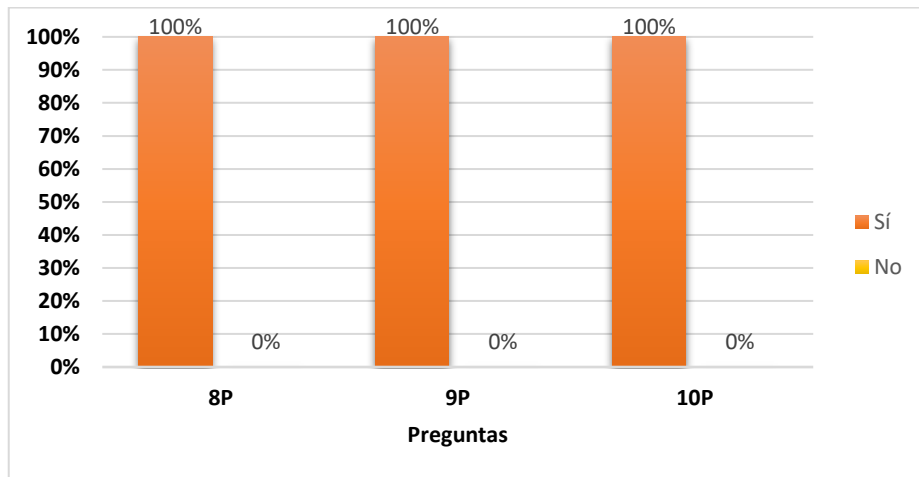


Figura 3.99. Resultados preguntas para usuario con rol Responsable

3.9.1.3 RESULTADOS USUARIO CON ROL SUPERVISOR

La Figura 3.100 muestra los resultados de las preguntas (11P, 12P, 13 y 14P) destinadas a los usuarios con rol Supervisor. El 100% de los usuarios indicó que llevó a cabo satisfactoriamente las siguientes actividades: Operaciones CRUD de los cambios que pertenecen a un documento específico, acceso y edición del contenido de los documentos asignados y actualización del estado de los documentos.

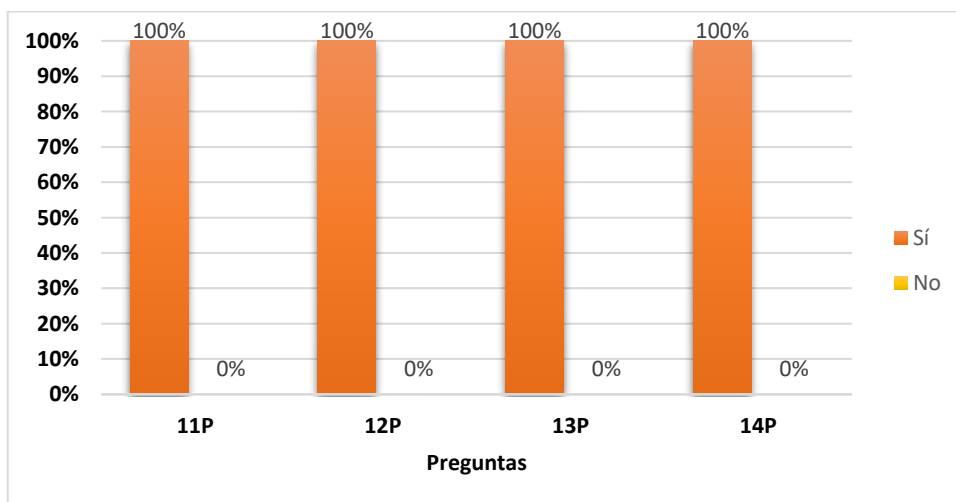


Figura 3.100. Resultados preguntas para usuario con rol Supervisor

3.9.1.4 RESULTADOS USUARIO CON ROL COLABORADOR

La Figura 3.101 muestra los resultados de las preguntas (15P, 16P, 17P y 18P) destinadas a los usuarios con rol Colaborador. El 75% de los usuarios con rol Colaborador logró crear y editar versiones de los ítems de un documento específico ya que, al momento de establecer una versión actual de un documento sin haber realizado ningún cambio, el sistema no lo permitía (el error y la corrección que se realizó se lo puede ver a más detalle en la sección Rectificaciones del Sistema Web). Por otro lado, el 100% de los usuarios pudo acceder a los documentos que tenía asignados y actualizar su estado, además de visualizar y finalizar los cambios asignados por los usuarios con rol Supervisor pertenecientes a un documento particular.

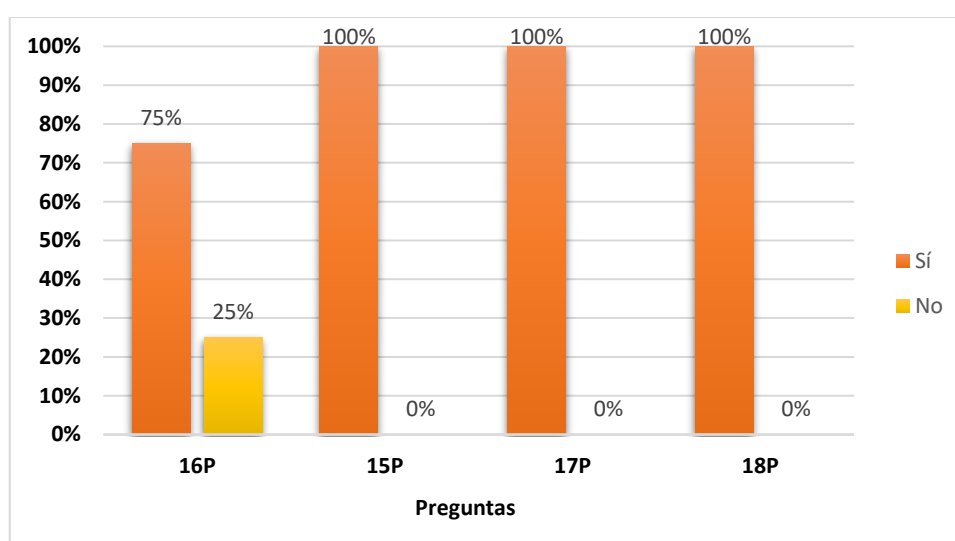


Figura 3.101. Resultados preguntas para usuario con rol Colaborador

3.9.2 RESULTADOS DE LA INTERFAZ GRÁFICA

La Figura 3.102 muestra los resultados de las preguntas (23P, 24P y 25P) destinadas a todos los usuarios con diferentes roles. El 43.75% de todos los usuarios estuvieron de acuerdo que el lenguaje utilizado en el sistema web fue claro y entendible, el 31.25% mencionó que estuvieron totalmente de acuerdo y solamente el 25% estuvo en desacuerdo debido a que se evidenció que ciertas alertas no se visualizaban de manera correcta, sin embargo, los problemas presentados se los corrigieron y se los puede ver a más detalle en la sección Rectificaciones del Sistema Web. Por otro lado, el 62.50% de los usuarios estuvieron de acuerdo en que los colores, tamaños y tipo de letra utilizados en el sistema web fueron los adecuados, mientras que el 37.5% estuvo totalmente de acuerdo. Finalmente, el 56.25% de los usuarios estuvo de acuerdo en que la forma como se estructuró el contenido del sistema web fue el correcto mientras que el 43.75% estuvo totalmente de acuerdo.

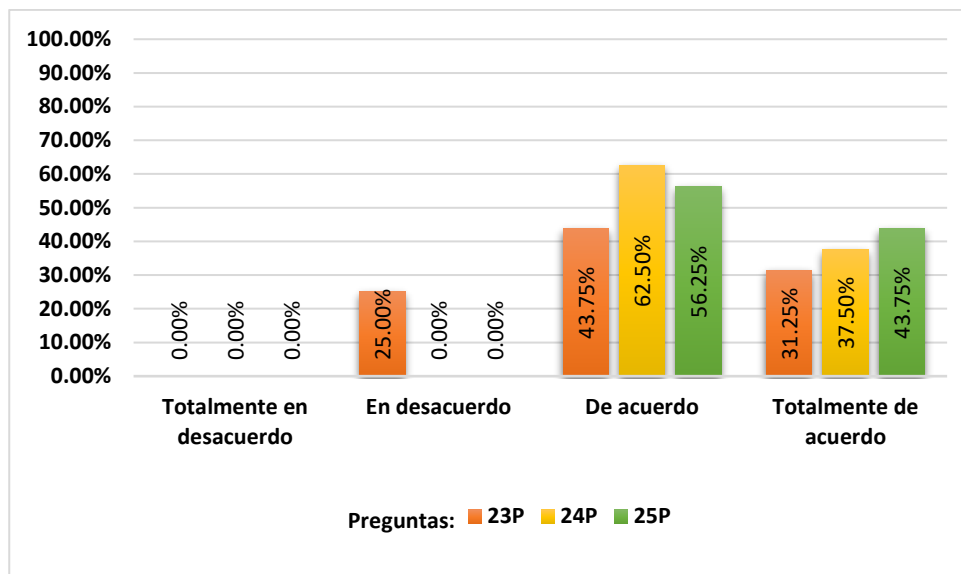


Figura 3.102. Resultados de las preguntas sobre la interfaz gráfica

La Figura 3.103 muestra los resultados de la pregunta 26P destinada a todos los usuarios con diferentes roles. El 62.50% de los usuarios encuestados señaló que el diseño de la interfaz gráfica fue bueno, el 25% indicó que fue muy bueno, el 12.5% indicó que fue regular y ningún usuario mencionó que el diseño de la interfaz gráfica del sistema web fue muy malo o malo. Cabe mencionar que la calificación de regular dada únicamente por 3 de los 16 usuarios encuestados, se produjo debido a que dichos usuarios se percataron de errores en la visualización de cierta información que se logró corregir de manera oportuna.

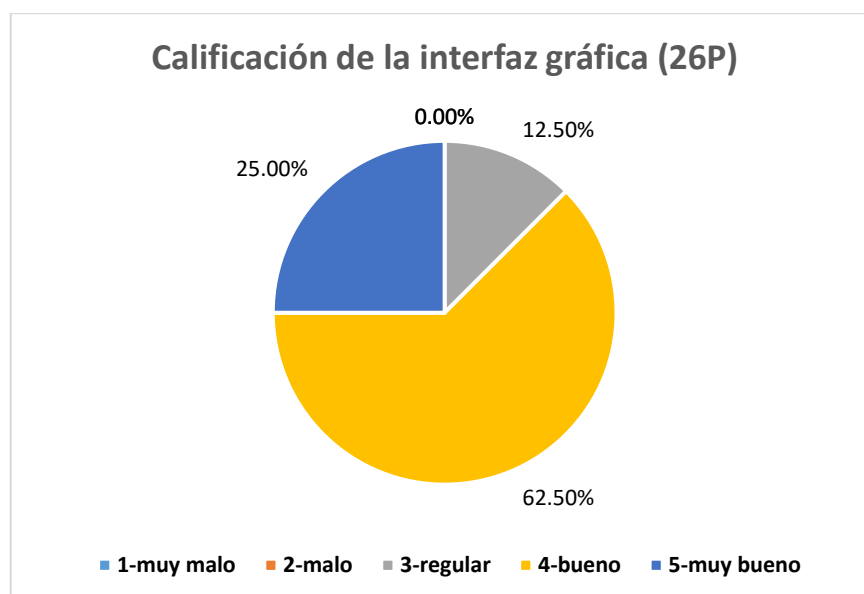


Figura 3.103. Resultados calificación de la interfaz gráfica

3.9.3 RESULTADOS FUNCIONAMIENTO GENERAL DEL SISTEMA

En la Figura 3.104 se muestran los resultados de las preguntas (1P, 19P, 20P, 21P Y 22P) destinadas para los usuarios con rol Responsable, Supervisor y Colaborador. En la pregunta 22P se tiene que el 92% de los usuarios pudo visualizar las fechas de entrega de los documentos en el calendario de manera correcta y solamente el 8% de los 12 usuarios (1 persona) no pudo ver los documentos de manera correcta debido a que dicho usuario no dio clic sobre la fecha de entrega del documento. Por otro lado, el 100% de los usuarios indicaron que pudieron registrarse de manera correcta en el sistema web, así como también visualizar el panel Home y ver los documentos finalizados con la información de los usuarios asignados. Además, el 100% de los usuarios indicó que pudo generar y descargar los documentos terminados en formato PDF sin ningún inconveniente.

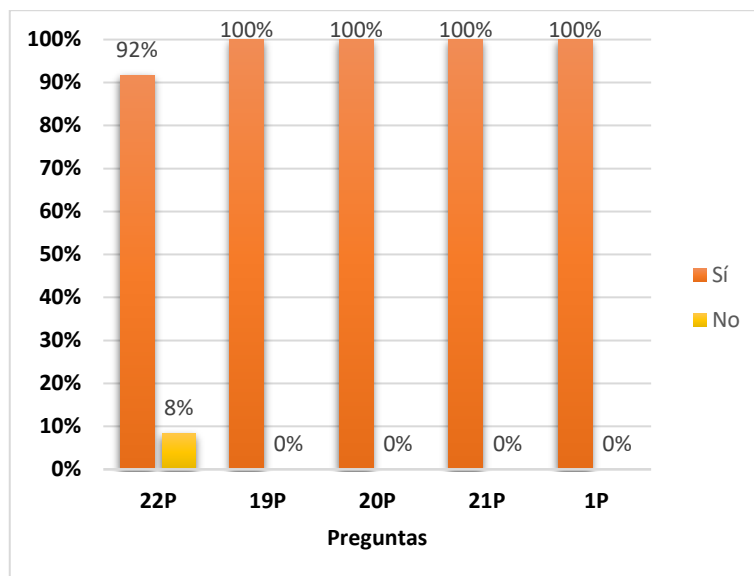


Figura 3.104. Resultados funcionamiento del sistema roles responsable-colaborador-supervisor

En la Figura 3.105 se muestran los resultados de las preguntas (2P, 3P y 4P) destinadas a los 16 usuarios con diferentes roles. El 100% de los usuarios mencionó que pudo iniciar y cerrar la sesión de manera correcta. Por otro lado, el 93.75% de los usuarios indicó que realizó el cambio de su contraseña de manera exitosa y solamente el 6.25% (1 usuario) no pudo realizar dicho cambio. Es importante mencionar que la persona que no pudo cambiar la contraseña explicó que intentó de hacer el cambio aproximadamente 25 minutos después de que recibió el correo con el enlace del formulario de cambio de contraseña, sin embargo, en el sistema web se designó un tiempo máximo de 15 minutos

y una vez transcurrido ese tiempo, el usuario recibe una alerta en pantalla indicándole que el tiempo finalizó y tiene que realizar una nueva petición de cambio de contraseña.

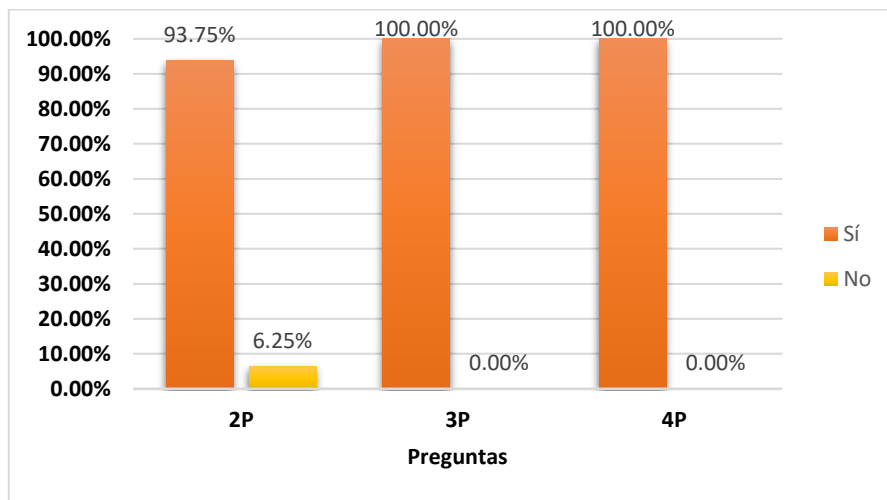


Figura 3.105. Resultados funcionamiento del sistema para todos los roles

En la Figura 3.106 se muestra los resultados de las preguntas (27P y 28P) destinada a todos los usuarios con diferentes roles. El 56.25% de los usuarios estuvo de acuerdo que el sistema fue fácil de usar mientras que el 43.75% estuvo totalmente de acuerdo. Por otro lado, el 50% de los usuarios estuvo de acuerdo que el sistema fue generalmente rápido al momento de usarlo, el 43.75% estuvo totalmente de acuerdo y solamente el 6.25% (1 persona) estuvo en desacuerdo.

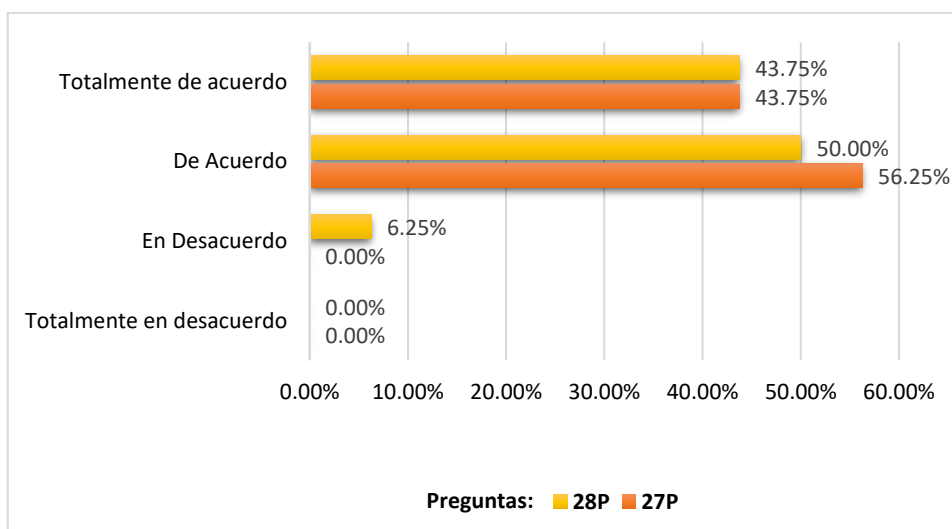


Figura 3.106. Resultados funcionamiento del sistema para todos los roles

En la Figura 3.107 se muestra los resultados de la pregunta 30P destinada a todos los usuarios con diferentes roles. El 62.50% de los usuarios encuestados señaló que el

funcionamiento general del sistema web fue bueno, el 37.5% indicó que fue muy bueno y ningún usuario mencionó que el funcionamiento del sistema web fue muy malo o malo.

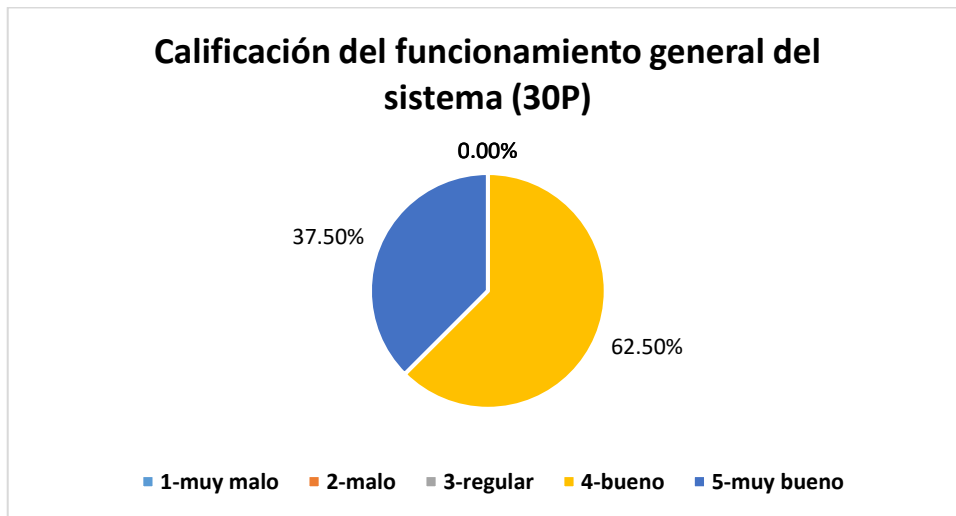


Figura 3.107. Resultados calificación del funcionamiento general del sistema web

3.9.4 RESULTADOS INCONVENIENTES

La Figura 3.108 muestra los resultados obtenidos en la pregunta 29P destinada para todos los usuarios con diferente rol. El 50% de los usuarios con rol Colaborador indicó que tuvo al menos algún inconveniente durante la navegación en el sistema web y el 25% de los usuarios con rol Gestor, Responsable y Supervisor mencionó que tuvieron al menos algún problema.

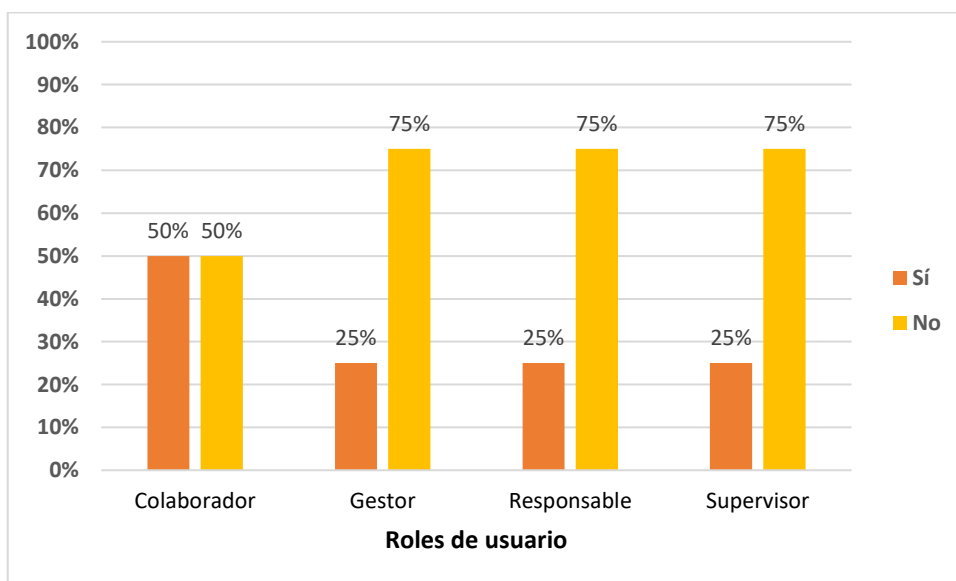


Figura 3.108. Resultados inconvenientes presentados

Es necesario indicar que, de los errores que presentaron los usuarios, el total de inconvenientes fueron leves y se los lograron solventar en el sistema web prototipo. Los detalles de los errores con sus debidas correcciones se los presentan a continuación.

3.10 RECTIFICACIONES EN EL SISTEMA WEB

En la Tabla 3.3, se indican los errores que se tuvieron en el sistema web prototipo por parte de los usuarios con diferentes roles y las correcciones realizadas.

Tabla 3.3. Errores y correcciones del sistema web

SPRINT	ERRORES	CORRECCIONES
SPRINT 1	Al momento de registrarse el sistema requiere 6 caracteres, sin embargo, al momento de restablecer la contraseña solo se pide 4 caracteres.	Se cambió el requerimiento de una contraseña mínima por una longitud de 6 caracteres. Además, se modificó para que la contraseña tenga al menos una letra mayúscula, una letra minúscula, un número y un carácter.
SPRINT 2	Ciertas alertas que se visualizan al realizar una operación, no se muestran de manera correcta ya que los nombres de las colecciones de la base de datos están en inglés.	Se cambió el mensaje de la alerta mostrada para que se visualice de mejor manera.
SPRINT 3	Ciertas alertas que se visualizan al realizar una operación, no se muestran de manera correcta ya que los nombres de las colecciones de la base de datos están en inglés.	Se cambió el mensaje de la alerta mostrada para que se visualice de mejor manera.
SPRINT 4	No se puede establecer una versión actual en el caso de que no se hayan realizado cambios en algún ítem.	Se verificó el inconveniente para que el usuario al dar clic en guardar establezca una versión sin la necesidad de haber realizado algún cambio.
	Los cambios realizados desaparecen al momento que un usuario Supervisor aprueba el cambio.	Se agregó en el filtro de la consulta a MongoDB para que se muestren los cambios realizados y finalizados en el panel.
	No es posible modificar el estado de un cambio realizado a asignado en caso de error.	Se agregó un botón para regresar al estado anterior del cambio.
SPRINT 5	No es posible modificar el estado de un cambio finalizado a realizado en caso de error.	Se agregó un botón para regresar al estado anterior del cambio.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El presente trabajo de titulación permitió establecer las bases para generar la documentación básica en el proceso de creación de un CSIRT académico. De esta manera, se creó un sistema web prototipo modularizado para facilitar que el proyecto sea escalable y fácil de controlar en el tiempo en el caso de que se requiera la implementación de nuevos módulos.
- Se establece que el sistema web prototipo es escalable debido a la separación del frontend y el backend en 2 repositorios diferentes, lo cual ayuda a dividir recursos de manera eficaz. Además, el sistema es modular ya que se tiene el código mejor distribuido y compartido, el mismo que se puede reusar en otros componentes, evitando así, tener cientos de líneas de programación en un solo archivo.
- El STACK MEAN+G facilitó el desarrollo del sistema web de manera rápida ya que solamente se utilizó Typescript para realizar la codificación de módulos tanto en el backend como en el frontend sin la necesidad de tener conocimientos de otros lenguajes de programación.
- El uso de la metodología ágil SCRUM permitió realizar el sistema web mediante varios Sprints, los cuales, se lograron finalizar en el plazo establecido. Además, las pruebas de funcionamiento y encuestas realizadas a los diferentes usuarios sirvieron para verificar inconvenientes que no se lograron discernir por parte del equipo de trabajo y se consiguió corregir los errores permitiendo mejorar la experiencia del usuario en el sistema web. Finalmente, se comprobó que los requerimientos funcionales y no funcionales, así como el diseño de la interfaz gráfica y el funcionamiento del sistema prototipo cumplieron con las expectativas de los usuarios.

4.2 RECOMENDACIONES

- Se recomienda el uso de plataformas para alojar código de un proyecto como GITHUB ya que se facilita tener un control de versiones al ir realizando cambios significativos en el proyecto
- Debido a las actualizaciones de software que realiza Angular cada 6 meses, es recomendable migrar de la versión 9 usada en el sistema web a la versión más reciente. Además, actualmente Apollo Server Express se encuentra disponible en

la versión 3, por lo que es recomendable migrar de la versión 2 con la que se trabajó el prototipo a la versión más estable.

- El uso de TSLINT para la detección de errores en el código ayudó a tener buenas prácticas de programación ya que se establecieron ciertas reglas al momento de codificar los módulos del sistema. Sin embargo, actualmente, la librería está obsoleta y es recomendable actualizar el proyecto a ESLINT, el cual también es un linter que detecta errores en código.
- Es recomendable crear un módulo para realizar operaciones CRUD de nuevos documentos con su respectivo contenido, ya que actualmente los documentos se los crean directamente en la base de datos. Además, se puede crear una interfaz Administrador para realizar operaciones CRUD de todos los módulos del sistema.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] Fundación Telefónica, “CIBERSEGURIDAD, LA PROTECCIÓN DE LA INFORMACIÓN EN UN MUNDO DIGITAL”. Ariel, septiembre de 2016. Consultado: el 5 de noviembre de 2021. [En línea]. Disponible en: <https://www.fundaciontelefonica.com/cultura-digital/publicaciones/>
- [2] Microsoft, “Cyberthreats, viruses, and malware - Microsoft Security Intelligence”. <https://www.microsoft.com/en-us/wdsi/threats> (consultado el 5 de noviembre de 2021).
- [3] IMB Security, “Cost of a Data Breach Report 2020”, p. 82.
- [4] J. Merchan-Lima, F. Astudillo-Salinas, L. Tello-Oquendo, F. Sanchez, G. Lopez-Fonseca, y D. Quiroz, “Information security management frameworks and strategies in higher education institutions: a systematic review”, *Ann. Telecommun.*, vol. 76, núm. 3, pp. 255–270, abr. 2021, doi: 10.1007/s12243-020-00783-2.
- [5] H. Bronk, M. Thorbruegge and M. Hakkaja, “A STEP-BY-STEP APPROACH ON HOW TO SET UP A CSIRT”, *ENISA*, el 22 de diciembre de 2006. <https://www.enisa.europa.eu/publications/csirt-setting-up-guide> (consultado el 5 de noviembre de 2021).
- [6] G. Killcrece, K.-P. Kossakowski, R. Ruefle, y M. Zajicek, “State of the Practice of Computer Security Incident Response Teams (CSIRTs)”, Defense Technical Information Center, Fort Belvoir, VA, oct. 2003. doi: 10.21236/ADA421664.
- [7] “About FIRST”, *FIRST — Forum of Incident Response and Security Teams*. <https://www.first.org/about> (consultado el 4 de noviembre de 2021).
- [8] “Software Engineering Institute”. <https://www.sei.cmu.edu/index.cfm> (consultado el 4 de noviembre de 2021).
- [9] “About ENISA - The European Union Agency for Cybersecurity”, *ENISA*. <https://www.enisa.europa.eu/about-enisa> (consultado el 4 de noviembre de 2021).
- [10] “Trusted Introducer : Home”. <https://www.trusted-introducer.org/> (consultado el 4 de noviembre de 2021).
- [11] “Open CSIRT Foundation”. <https://opencsirt.org/> (consultado el 4 de noviembre de 2021).
- [12] M. K. Andreas Meier, “SQL & NoSQL Databases”, en *SQL & NoSQL Databases*, 1a ed., Wiesbaden, Germany: Springer Vieweg, Wiesbaden, 2019, pp. 201–210.
- [13] “What is MongoDB? — MongoDB Manual”, *MongoDB: The application data platform / MongoDB*. <https://docs.mongodb.com/manual/> (consultado el 4 de noviembre de 2021).
- [14] “Express - Infraestructura de aplicaciones web Node.js”. <https://expressjs.com/es/> (consultado el 5 de noviembre de 2021).
- [15] “Angular - Introduction to the Angular Docs”. <https://angular.io/docs> (consultado el 7 de noviembre de 2021).
- [16] Node.js, “About”, *Node.js*. <https://nodejs.org/en/about/> (consultado el 6 de noviembre de 2021).
- [17] “GraphQL | A query language for your API”. <https://graphql.org/> (consultado el 22 de diciembre de 2020).
- [18] M. V. S. Andrew S. Tanenbaum, “SISTEMAS DISTRIBUIDOS SISTEMAS DISTRIBUIDOS: Principios y Paradigmas”, 2a ed., México: Prentice Hall, 2008, pp. 1–3.
- [19] J. Sayago Heredia, E. Flores, y A. Recalde, “Análisis Comparativo entre los Estándares Orientados a Servicios Web SOAP, REST y GRAPHQL”, dic. 2019, doi: 10.5281/zenodo.3592004.
- [20] M. O. contributors Jacob Thornton, and Bootstrap, “Bootstrap”. <https://getbootstrap.com/> (consultado el 7 de noviembre de 2021).
- [21] “JavaScript | MDN”. <https://developer.mozilla.org/es/docs/Web/JavaScript> (consultado el 7 de noviembre de 2021).

- [22] “JavaScript With Syntax For Types.” <https://www.typescriptlang.org/> (consultado el 7 de noviembre de 2021).
- [23] Tridibesh Satpathy, *Una guía para el Cuerpo de Conocimiento de Scrum*, 3a ed. Avondale, Arizona, USA: SCRUMstudy, 2017.
- [24] “Documentation for Visual Studio Code”. <https://code.visualstudio.com/docs> (consultado el 8 de noviembre de 2021).
- [25] “Inicio Rápido”, *GitHub Docs*. https://docs.github.com/_next/data/YqY7mV3MP6eH_SBXw64pg/es/free-pro-team%40latest/get-started/quickstart.json?versionId=free-pro-team%40latest&productId=get-started&restPage=quickstart (consultado el 8 de noviembre de 2021).
- [26] “¿Qué es Adobe XD y para qué se emplea?” <https://www.adobe.com/la/products/xd/learn/get-started/what-is-adobe-xd-used-for.html> (consultado el 8 de noviembre de 2021).
- [27] “Lucidchart overview | Lucidchart”. <https://www.lucidchart.com/pages/product> (consultado el 8 de noviembre de 2021).
- [28] “¿Qué es MongoDB Compass? - Brújula MongoDB”, *Compass. The GUI for MongoDB*, el 29 de noviembre de 2021. <https://docs.mongodb.com/compass/current/> (consultado el 15 de diciembre de 2021).
- [29] A. Menzinsky, “Historias de Usuario”, *Septiembre 2020*, pp. 10–23.
- [30] M. Fowler y K. Scott, *UML gota a gota*. Mexico: Pearson Educación, 1999.
- [31] S. F. Morales, J. G. Molina, y D. S. Ruiz, “Inferencia del esquema en bases de datos NoSQL a través de un enfoque MDE”, p. 14.
- [32] C. Salgado, “Sketchs, mockups, wireframes y prototipos”, *Mosaic*, núm. 130, sep. 2015, doi: 10.7238/m.n131.1524.
- [33] Jerry Cao, Matt Ellis, y Narek Khachatryan, “The Guide to Mockups”, en *THE GUIDE TO MOCKUPS: Mockup types, methods and best practices*, pp. 6–10.

6 ANEXOS

ANEXO A. Documento de la reunión con CSIRT-EPN

ANEXO B. Historias de Usuario

ANEXO C. Base de Datos

ANEXO D. Repositorio backend del proyecto

ANEXO E. Repositorio frontend del proyecto

ANEXO F. Encuestas realizadas a los usuarios

ORDEN DE EMPASTADO