

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE SISTEMA WEB PARA PROCESOS
ELECTORALES DE LA ESFOT BASADO EN BLOCKCHAIN**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS**

Kevin Alexander Curay Lasluisa

Kevin.curay@epn.edu.ec

DIRECTOR: MSc. Juan Pablo Zaldumbide

juan.zaldumbide@epn.edu.ec

CODIRECTORA: MSc. Mónica Vinueza Rhor

monica.vinueza@epn.edu.ec

Quito, Abril 2022

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Kevin Alexander Curay Lasluisa como requerimiento parcial a la obtención del título de TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS, bajo nuestra supervisión:

Juan Pablo Zaldumbide

Proaño

DIRECTOR DEL PROYECTO

Mónica Vinueza Rhor

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo Kevin Alexander Curay Lasluisa con CI: 0503908857 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.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación - COESC, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

DEDICATORIA

Este proyecto está dedicado para las personas que creyeron en mí, como mi padre Cesar y mi madre Elsa a quienes les debo la vida y por quienes seguiré progresando, a mi tía Aida y mi tío Mario que me permitieron quedarme en su casa mientras estudiaba y mi tío Hernán que me dio la herramienta que me ayudaría a definir mi carrera como programador.

También dedico este trabajo a mis hermanos Eddy y Cristian, y a una persona muy importante como lo es mi novia Alejandra que me apoyo por mucho tiempo y no permitió que me dé por vencido, me animo por mucho tiempo para que alcanzara esta meta.

Kevin Alexander Curay Lasluisa

AGRADECIMIENTO

Un agradecimiento infinito al Ing. Juan Zaldumbide que me guio y ayudo con este proyecto, además de los trámites pertinentes.

Un agradecimiento a la empresa ZimaIT que me permitió culminar mis estudios mientras laboraba con ellos.

Nuevamente un agradecimiento para todos los que creyeron que lo conseguiría.

Kevin Alexander Curay Lasluisa

ÍNDICE DE CONTENIDO

1	INTRODUCCIÓN.....	1
1.1	Objetivo general.....	2
1.2	Objetivos específicos	2
1.3	Alcance	2
2	METODOLOGÍA	4
2.1	Metodología de Desarrollo	4
	Roles	4
	Artefactos.....	5
2.2	Diseño de interfaces	6
	Herramienta utilizada para el diseño.....	6
	Sistema Web	6
2.3	Diseño de la arquitectura	7
	Arquitectura de Datos.....	7
	Patrón arquitectónico	8
2.4	Herramientas de desarrollo.....	9
3	RESULTADOS.....	11
3.1	Sprint 0. Definición estructura de proyecto backend.....	11
3.2	Sprint 1. Definición estructura de proyecto frontend	12
3.3	Sprint 2. Manejo de sesión activa	15
3.4	Sprint 3. Seguridad	18
3.5	Sprint 4. Proceso Electoral, Listas participantes y Candidatos	23
3.6	Sprint 5. Estudiante.....	28
3.7	Sprint 6. Resultados Votación.....	29
4	Conclusiones y Recomendaciones	30
4.1	Conclusiones	30
4.2	Recomendaciones	31
5	REFERENCIAS BIBLIOGRÁFICAS.....	32
6	ANEXOS.....	34
	ANEXO I. Certificado de Originalidad.....	35
	ANEXO II. Manual Técnico.....	36
	ANEXO III. Manual de usuario	48
	ANEXO IV. Manual de instalación.....	49

ÍNDICE DE FIGURAS

Fig. 1: Interfaz de inicio de sesión	7
Fig. 2: Diagrama entidad relación	8
Fig. 3: Arquitectura del sistema web	9
Fig. 4: Tablas de la base de datos EEP	11
Fig. 5: Estructura de la aplicación web	12
Fig. 6: Estructura proyecto EEP	13
Fig. 7: Estructura de modulo y componentes reutilizables.....	14
Fig. 8: Archivo data-central	15
Fig. 9: Método que autoriza el acceso a rutas del sistema.	15
Fig. 10: Formulario inicio de sesión.	16
Fig. 11: Formulario del email	17
Fig. 12: Formulario cambio de contraseña.....	17
Fig. 13: Menú dinámico.....	18
Fig. 14: Mensaje de alerta para eliminar un registro.....	19
Fig. 15: Formulario de transacciones.....	19
Fig. 16: Tabla de transacciones ingresadas.	20
Fig. 17: Formulario para editar transacción.	20
Fig. 18: Formulario de registro de roles.	21
Fig. 19: Tabla de roles ingresados.....	21
Fig. 20: Formulario para editar registro de rol.....	22
Fig. 21: Formulario registrar menú de opciones.	22
Fig. 22: Formulario para modificar registro de menú.	23
Fig. 23: Formulario para registrar proceso electoral.	24
Fig. 24: Tabla que muestra los registros de procesos electorales.	24
Fig. 25: Formulario para modificar registro de proceso electoral.....	25
Fig. 26: Formulario para registrar lista participante.....	25
Fig. 27: Listas participantes para ese proceso de elección.	26
Fig. 28: Formulario para editar lista participante.....	26
Fig. 29: Formulario para registrar candidato de una lista participante.	27
Fig. 30: Tabla de registros de candidatos.....	27
Fig. 31: Formulario para modificar registro de candidato.....	27
Fig. 32: Información el proceso y listas participantes.	28
Fig. 33: Formulario para subir plantilla de estudiantes.	29
Fig. 34: Conteo de votos.....	29

ÍNDICE DE TABLAS

TABLA I: Historias de usuario. Login de acceso a usuarios con distintos roles	5
TABLA II: Herramientas para el desarrollo de la API Rest	9
TABLA III: Herramientas para el desarrollo del Sistema Web	10
TABLA IV: Tabla de requerimientos.	36
TABLA V: Historia de usuario Nro.1: Login de acceso a usuarios con distintos roles.	37
TABLA VI: Historia de usuario Nro.2: Reestablecer contraseña	37
TABLA VII: Historia de usuario Nro.3: Cerrar sesión del sistema web.	38
TABLA VIII: Historia de usuario Nro.4: Selecciona candidato y vota.....	38
TABLA IX: Historia de usuario Nro.5: Registra, edita y elimina proceso electoral	39
TABLA X: Historia de usuario Nro.6: Registra, edita y elimina lista participante .	39
TABLA XI: Historia de usuario Nro.7: Registra, edita y elimina candidatos de una lista.	40
TABLA XII: Historia de usuario Nro.8: Importar archivo de estudiantes.	40
TABLA XIII: Historia de usuario Nro.9: CRUD de rol.	41
TABLA XIV: Historia de usuario Nro.10: CRUD de transacciones.	41
TABLA XV: Historia de usuario Nro.11: Políticas de seguridad (menú).	42
TABLA XVI: Historia de usuario Nro.12: Registro individual de usuario.	42
TABLA XVII: Product Backlog: Ordenada por prioridad.	43
TABLA XVIII: Sprint Backlog	44

RESUMEN

En la actualidad observamos que la tecnología es nuestra mejor aliada para afrontar crisis, como la reciente crisis sanitaria del COVID-19 que desde su aparición puso en pausa varias actividades y procesos que se realizaban de forma presencial, como es el caso de las votaciones para elegir a representantes de la Escuela de formación de tecnólogos.

Con el avance actual de la tecnología se puede crear un sistema web con la capacidad de procesar y soportar todo el proceso de votaciones de la ESFOT. Para logra esto es necesario usar blockchain, que brinda la integridad y transparencia requerida en un sistema de votación electrónico, además al separar el sistema en la aplicación web y una API REST permite el control total de los recursos, la escalabilidad del sistema y un rápido despliegue.

A lo largo del proyecto se usó la metodología SCRUM, que permite un desarrollo rápido con múltiples interacciones en los cuales mejorar la calidad del software entregado, ajustándose a los requerimientos del cliente.

PALABRAS CLAVE: Blockchain, Api Rest, Postgresql, Angular, SCRUM, MVC, Docker.

ABSTRACT

Currently we observe that technology is our best ally to face crises, such as the recent health crisis of COVID-19, which since its appearance put on pause several activities and processes that were carried out in person, as is the case of voting for elect representatives of the Technologist Training School.

With the current advances in technology, a web system can be created with the capacity to process and support the entire ESFOT voting process. To achieve this, it is necessary to use blockchain, which provides the integrity and transparency required in an electronic voting system, in addition to separating the system in the web application and a REST API allows total control of resources, system scalability and rapid deployment.

Throughout the project, the SCRUM methodology was used, which allows rapid development with multiple interactions in which to improve the quality of the delivered software, adjusting to the client's requirements.

KEYWORDS: Blockchain, Api Rest, Postgresql, Angular, SCRUM, MVC, Docker.

1 INTRODUCCIÓN

La ausencia de votantes en los distintos procesos electorales que efectúa la Escuela de Formación de Tecnólogos es preocupante, ya que en el proceso electoral para elegir a los representantes ante el Consejo de la ESFOT realizado en el periodo 2019-A existió una ausencia del 30% de estudiantes [1], mientras que en las elecciones del periodo 2019-B para representantes de la Asociación de Estudiantes de la ESFOT existió una ausencia del 39% de votantes [2]. Estas ausencias pueden tener varios motivos como la falta de comunicación sobre la existencia de un nuevo proceso electoral, el tiempo de los estudiantes ya que algunos se encuentran laborando o realizando sus pasantías, también olvido de credencial necesaria para poder ejercer su derecho a votar.

El sistema tradicional de votación cuenta con sus facetas básicas que son: garantizar un voto por votante, mantener el anonimato de los votantes, la precisión del voto, la seguridad del sistema y la prevención del fraude [3]. Por años se ha generado duda en la viabilidad de usar estos sistemas electrónicos, ya que no proporcionaban toda la seguridad y fiabilidad del proceso, sin embargo, en esta última década se ha utilizado blockchain (cadena de bloques) para crear sistemas de votación electrónica seguros, gracias a que blockchain se despliega en una red de datos distribuidos que comparten y ejecutan las transacciones en cada una de las máquinas, esto ayuda a que la información no pueda ser alterada o borrada [4].

Otro impedimento que se ha presentado es la actual crisis sanitaria del Covid-19 la cual afecto al país y al mundo entero, por lo tanto, no es posible convocar a nuevas elecciones para ocupar los cargos que dejaran los actuales representantes en el próximo periodo académico, ya que aún existe la incertidumbre de cuando se podrá retornar de forma presencial a las clases y las actividades institucionales, por lo cual desarrollar un sistema de votación online es una clara alternativa a usar.

El proyecto del sistema de votación online para los procesos electorales de la esfot permitirá:

- Ingresar a los candidatos y los cargos a los cuales postulan a través de un rol administrador.
- Ingreso al sistema con una cuenta personal de cada estudiante.
- Ejercer el derecho a votar por cada candidato.
- Resultados del proceso electoral de forma inmediata al termino de las votaciones.

La tecnología es nuestra mejor aliada a la hora de implementar soluciones que permita que la humanidad continúe con sus actividades remotamente. A través de internet se puede realizar varias actividades como trasferencias bancarias, servicios delivery, entretenimiento en streaming, telemedicina, teletrabajo y muchos más. En este contexto, el desarrollo de un sistema web de votación electrónico permitirá a la comunica de la ESFOT continuar con sus procesos electorales.

1.1 Objetivo general

Desarrollar un sistema web para validar la seguridad en los procesos electorales de la ESFOT

1.2 Objetivos específicos

1. Determinar los requerimientos del sistema web.
2. Diseñar la arquitectura y prototipos de la interfaz de usuario del sistema.
3. Desarrollar el sistema web.
4. Desplegar la aplicación con Docker.

1.3 Alcance

Un sistema de votación electrónico basado en Blockchain es la mejor opción para poder convocar a nuevos procesos electorales, ya que puede solucionar varios inconvenientes como reducir el ausentismo por parte de los estudiantes de la ESFOT al poder ejercer su derecho a voto desde un computador, celular, tablet u otro dispositivo con conectividad a internet y acceso a un navegador. Otro punto que considerar es la presente crisis sanitaria que aun agobia al país, por las fuertes restricciones ejecutadas desde la declaración de pandemia en marzo de 2020, impidiendo así que la ESFOT continúe con sus actividades electorales llevadas a cabo semestre.

Blockchain proporciona seguridad e integridad de la información, al ser un registro inmutable en una amplia red distribuida de máquinas que ejecutan procesos y comparten información entre sí, gracias a su concepto de conceso puede verificar la validación de cada bloque sea el mismo en todas las maquinas que componen la red, si el bloque en cuestión no es el mismo que los otros, ese bloque se corrige de acuerdo con lo que la mayoría tiene en sus registros [5].

La Escuela de Formación de Tecnólogos se caracteriza por innovar, modernizar sus instalación y sistemas, así que siguiendo esa misma línea se puede implementar un sistema de votación electrónico para solucionar sus actuales necesidades.

El sistema es multi roles con sus respectivos permisos y acceso a los recursos, sin embargo, se definió los siguientes roles principales:

- Super Administrador.
- Administrador:
 - Profesores de nombramiento.
 - Profesores de contrato.
 - Personal Administrativo.
- Estudiantes.

2 METODOLOGÍA

La metodología ágil que se usará es Scrum con un marco de trabajo Kanban y se iniciará con la toma de requerimientos al cliente, para poder realizar las respectivas historias de usuarios. Esta información ayudará a formar el Producto Backlog con el cual se podrá redactar cada sprint detallando su lista de actividades (Beas, 2011).

Kanban se usará de la siguiente manera: A través de un tablero de tareas se visualizará el flujo de trabajo, el tablero está dividido en tres secciones To Do, Doing y Done, traducido al español sería Pendiente, En proceso y Finalizado. Las tareas son las actividades del Sprint que serán colocadas en el tablero (Mesh, 2020).

2.1 Metodología de Desarrollo

Roles

Scrum cuenta con tres tipos de participantes en el proceso del proyecto, cada uno cuenta con sus actividades y funciones específicas que ayudan a que esta metodología sea una de las más usadas en el campo laboral. A continuación, se describe a los participantes:

Product Owner

Es un representante de la Escuela de Formación de Tecnólogos, es quien se encargará de comunicar al equipo, los requerimientos necesarios para iniciar con el proyecto, los requerimientos esenciales son los funcionales y no funcionales.

Scrum Master

El Ing. Juan Pablo Zaldumbide ocupa este rol. Se encarga de gestionar y asegurar que el proceso se cumpla de acuerdo con lo planificado. Es quien constantemente asigna las actividades con un tiempo estimado de tal manera que se cumpla el proyecto en los plazos establecidos para el cliente.

Development Team

El equipo de desarrollo consta de un solo miembro que es el autor de este proyecto. Cuenta con los suficientes conocimientos técnicos para la construcción de la plataforma web dirigida al personal y estudiantes de la ESFOT, además del desarrollo de una API Rest propia que ayude en el procesamiento de la información.

Artefactos

Los artefactos aumentan con el uso de una metodología combinada entre *Scrum* y *Kanban*, se agregan beneficios al momento del desarrollo como observar el progreso de las actividades, el estado en el que se encuentran. Consiguiendo eficiencia y desarrollo de funcionalidades en un lapso de tiempo corto en los que el programador puede enfocarse.

Recopilación de Requerimientos

Es una fase en la cual se recolecta información del cliente y sus necesidades para luego transformarla en lógica del negocio que el sistema contendrá. Con la información recopilada se detalló una tabla que contiene los requerimientos esenciales y delimitados que el sistema tendrá, en el ANEXO II puede observar con más detalle.

Historias de Usuario

La etapa inicial de la toma de requerimientos sirve como base en la construcción de las historias de usuarios, las cuales muestran a nivel de usuario las funcionalidades y controles que se desarrollaran en el sistema. En la **TABLA I** se puede apreciar un ejemplo de una historia de usuario usada en el proyecto. Las historias de usuario faltantes se las puede visualizar en el ANEXO II Manual técnico sección historias de usuario.

TABLA I: Historias de usuario. Login de acceso a usuarios con distintos roles

Historia de usuario	
Número: 1	Usuario: Usuario de cualquier Rol
Nombre Historia: Login de acceso a usuarios con distintos roles.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: Inicio de sesión desde la aplicación web, válida para cualquier usuario, en la validación de la API devolver el rol que el usuario tiene con su menú de opciones y permisos de lectura, edición y eliminación de registros en cada una de las transacciones.	
Observaciones: <ul style="list-style-type: none">• En el formulario de inicio de sesión agregar las validaciones.• Validaciones en la API Rest.	

Product Backlog

En un proyecto que utiliza *SCRUM* es fundamental la creación del *Product Backlog* que se considera un listado en el cual se ordena por prioridad los requerimientos para la implementación [6]. Además, contiene una breve descripción de las funcionalidades y la estimación en tiempo. En el ANEXO II Manual Técnico sección Product Backlog se encuentra el detallado más información.

Sprint Backlog

El Scrum team es el encargado de identificar una lista de tareas para realizar el sprint backlog, estas tareas deben ser totalmente completadas en cada sprint. Se representa en un tablero en el cual se agregan tareas, de esta forma el trabajo se vuelve visible mientras dura el sprint [7]. En el ANEXO II Manual Técnico sección Sprint Backlog se encuentra detallado más información.

2.2 Diseño de interfaces

Herramienta utilizada para el diseño

Aceptados los requerimientos del cliente, se procede con la descripción de la herramienta para realizar los prototipos del sistema web.

Justinmind Prototyper

Es una herramienta para realizar prototipos de sitios web, aplicaciones de escritorio y móviles, adaptable a sistemas operativos como Windows, Mac, iOS y Android. Empresas gigantes como HP, Google o el Grupo Alibaba la avalan por sus múltiples beneficios, al permitir una reducción de costos en el desarrollo de la aplicación con su función de simulación de procesos en el cual se puede detectar errores en fase temprana [8].

Sistema Web

En el presente proyecto justinmind permitió la elaboración de interfaces de forma eficiente como se muestra en la **Fig. 1**

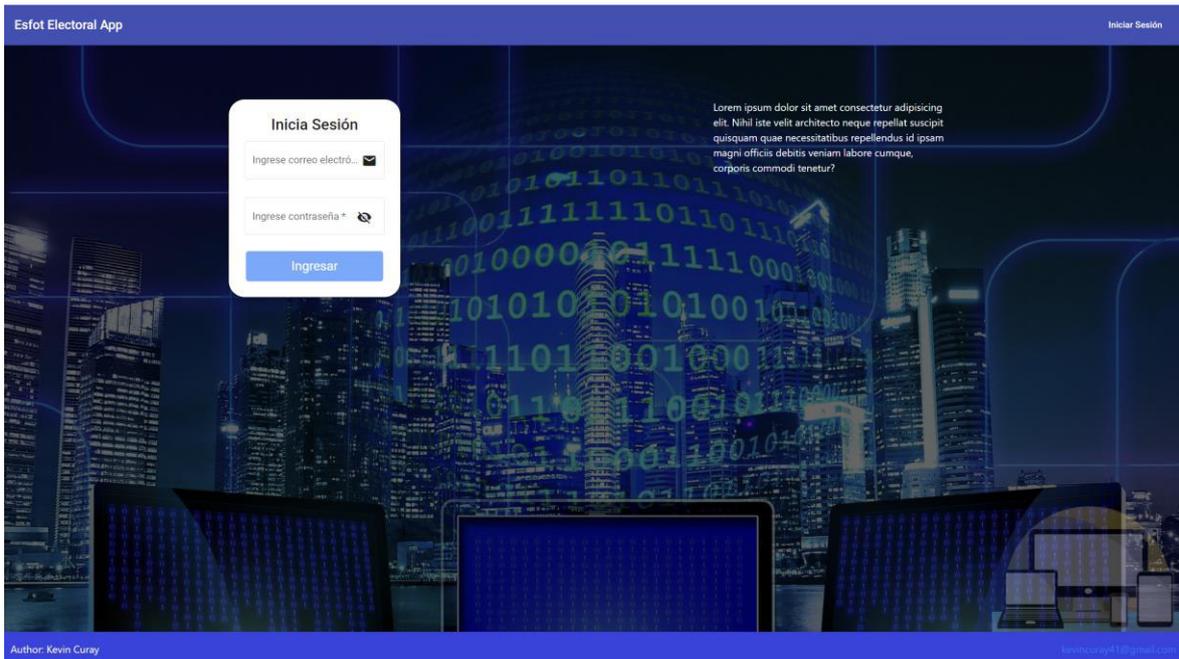


Fig. 1: Interfaz de inicio de sesión

2.3 Diseño de la arquitectura

Arquitectura de Datos

El diseño de la base de datos cambio constantemente, de acuerdo con las necesidades de parametrización y lógica de negocio.

Se define a través de un sistema de transacciones en el cual se agregan los nuevos módulos construidos en el front-end, además de la definición de n cantidad de roles que se pueden asociar a las transacciones por una tabla de menú de opciones, logrando de esta forma controlar los accesos, el esquema entidad relación se ilustra en la **Fig. 2**.

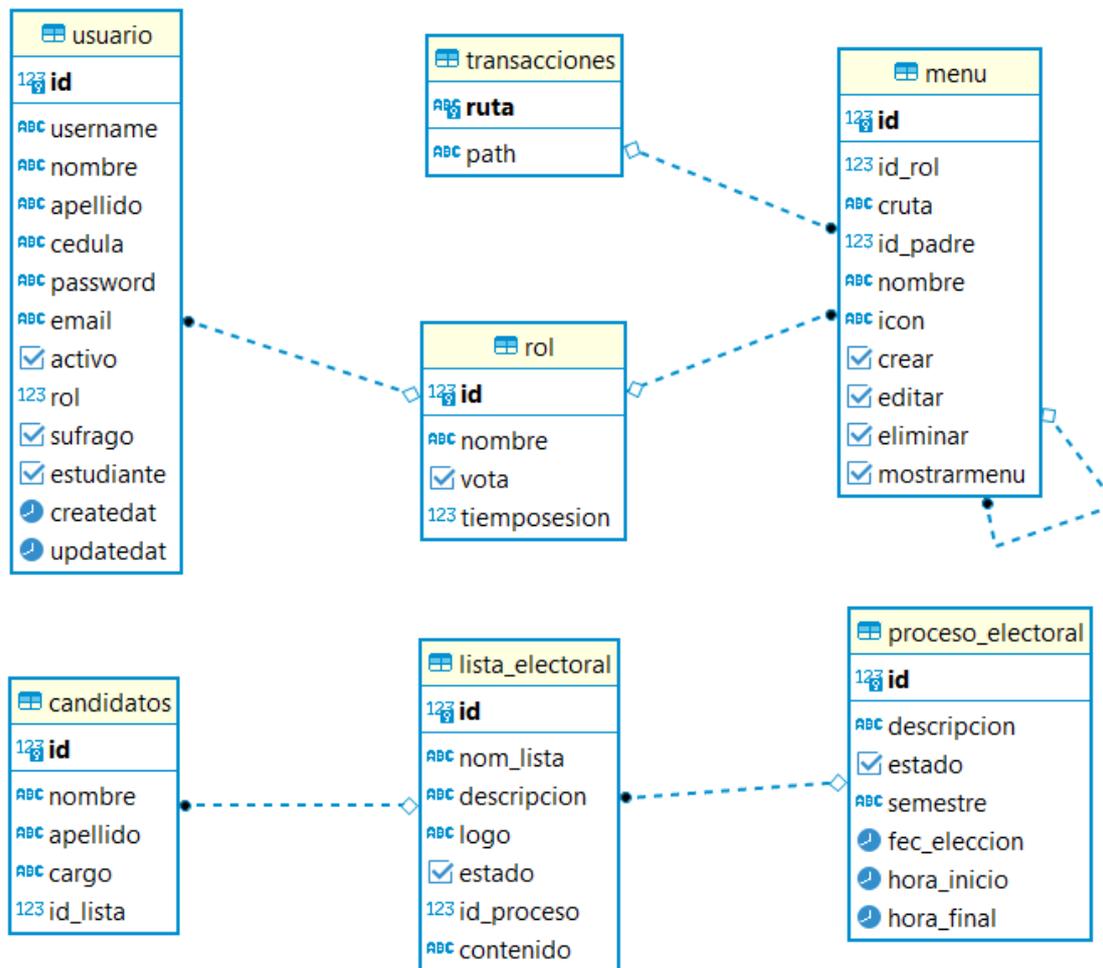


Fig. 2: Diagrama entidad relación

Patrón arquitectónico

La arquitectura a usar es MVC (Modelo Vista Controlador) que es un estilo de arquitectura de software que consiste en dividir en tres componentes distintos, empezando por los datos del sistema, la lógica de control y las interfaces de usuarios [9].

- **Modelo:** representa el manejo de la información en datos.
- **Vista:** interacción grafica del usuario con el sistema.
- **Controlador:** intermediario y encargado de procesar las peticiones del cliente devolviendo una respuesta de los datos procesados de nuevo al cliente.

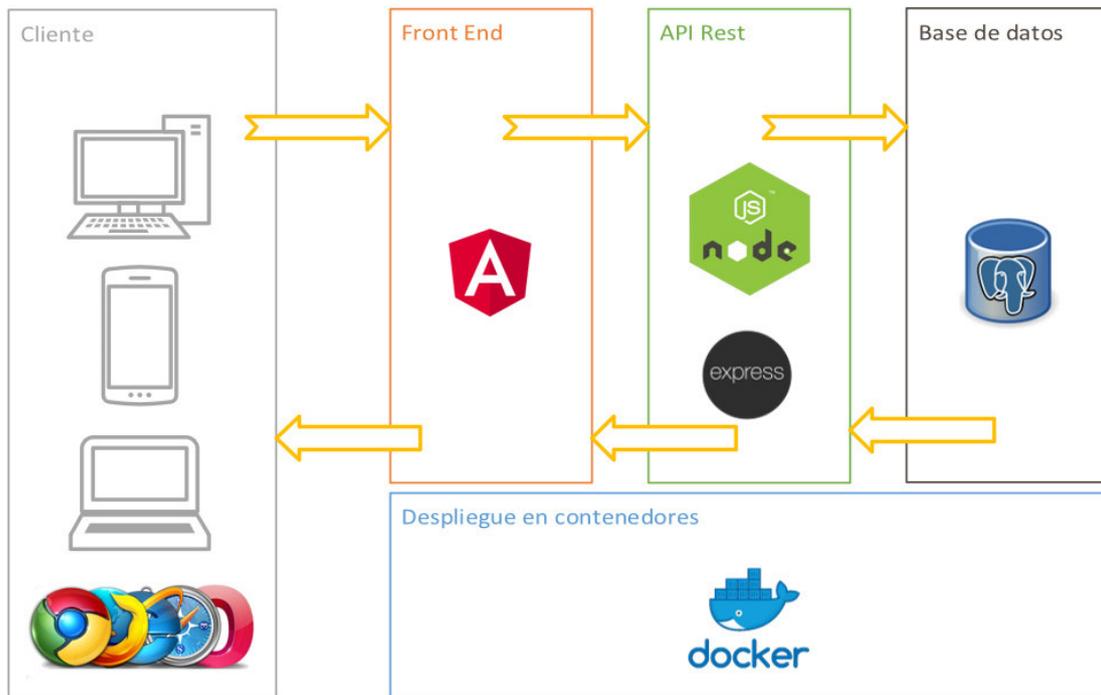


Fig. 3: Arquitectura del sistema web

2.4 Herramientas de desarrollo

Las herramientas que se usaron en el desarrollo fueron seleccionadas de acuerdo a los conocimientos adquiridos por el equipo de desarrollo y las ventajas de que cada herramienta brinda.

TABLA II: Herramientas para el desarrollo de la API Rest

Herramienta	Justificación
Nodejs: Basado en JavaScript y orientado a la capa del servidor como un entorno en tiempo de ejecución multiplataforma. También permite escalabilidad, gestionando múltiples conexiones simultáneamente [10].	Nodejs ayudará en el desarrollo del backend, que contendrá toda la lógica de negocio como son el manejo de autenticación, el acceso a recursos a través de rutas, el control y la conexión a la instancia de la base de datos.
Typescript: TypeScript es un lenguaje de programación desarrollado por Microsoft, se lo considera como un	Typescript mejora la escritura y lectura de código, al contar con tipado de datos

superconjunto de JavaScript, que añade Objetos basados en clases y tipado estático. [11].	y el manejo de clases que puedan extender de otras.
---	---

TABLA III: Herramientas para el desarrollo del Sistema Web

Herramienta	Justificación
Angular: Es un framework opensource desarrollado por Google, facilita el desarrollo de aplicaciones web SPA. Al usar el concepto de componentes se puede crear comportamiento específico de un componente y reutilizarlo en varios módulos definidos en el sistema [12].	La reutilización de componentes ayuda al mantenimiento y acelera el tiempo de desarrollo, al definir componentes que realicen la misma lógica y al pasarle parámetros el componente realice el comportamiento definido. El manejo de seguridad de las rutas es de forma dinámica que es parametrizada desde la base de datos, al igual que el manejo de roles con sus respectivos permisos.
Angular Material: Es una librería de estilos similar a Material Design [13], existe con el propósito de integrarse a un proyecto de Angular con el objetivo de ahorrar tiempo de desarrollo en el diseño y estilo de la aplicación web.	Esta librería permite utilizar componentes prefabricados como formularios, tablas, barras de progresos, botones entre otras opciones que facilitan el proceso de diseño de la aplicación.

3 RESULTADOS

Los resultados del proyecto se presentan a continuación.

3.1 Sprint 0. Definición estructura de proyecto backend

El Sprint 0 enlista las actividades necesarias para la definición, configuración y creación de la API Rest del sistema web. Las actividades realizadas son

- Crear la Base de datos.
- Estructura proyecto de la API REST.

Creación de base de datos.

La base de datos esta implementada en PostgreSQL con un modelo entidad relación. En la **Fig. 4** se observa una conexión a la base de datos EEP que contiene las tablas que almacenan la información del sistema.



Fig. 4: Tablas de la base de datos EEP

Estructura proyecto de la API REST

La API REST está construida con nodejs y express, juntas proporcionan un potente servidor web al realizar las configuraciones necesarias. La **Fig. 5** se observa la distribución del directorio principal del proyecto, A continuación, se detalla el contenido de las carpetas.

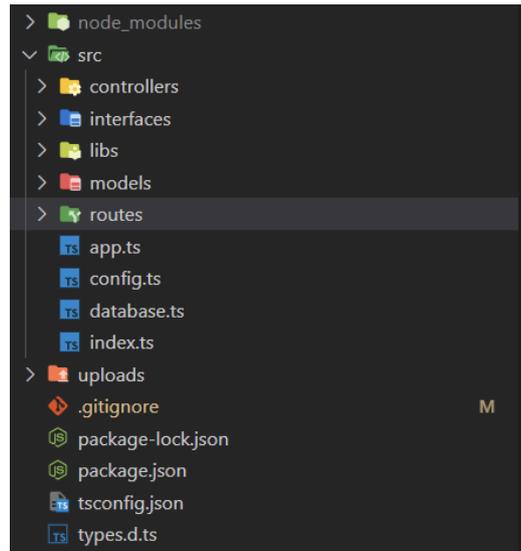


Fig. 5: Estructura de la aplicación web

- **node_modules:** contiene las librerías necesarias para la ejecución del proyecto.
- **uploads:** donde se almacenan las imágenes de las listas.
- **controllers:** contienen los archivos con código que controle las llamadas a la base de datos de forma asíncrona.
- **interfaces:** interfaces que ayudan al tipado de datos.
- **libs:** archivos con métodos y funciones necesarias para el sistema.
- **models:** contiene archivos de modelos que pueden mapear tablas de una BDD relacional.
- **routes:** contiene archivos con la definición de distintas rutas que la aplicación web usara.
- **index:** contiene el código inicial del servidor web.

3.2 Sprint 1. Definición estructura de proyecto frontend

El Sprint 1 tiene actividades para la con definición, configuración y creación de la aplicación web con el framework de Angular. Además de la creación de módulos, servicios y componentes que serán reutilizados a lo largo de proyecto. Las actividades realizadas son:

- Estructura proyecto de Angular.
- Crear de módulos y componentes para reutilización.
- Crear servicio centrar que gestione información del usuario logueado.
- Manejar acceso a rutas y permisos del sistema de forma dinámica.

Estructura proyecto de Angular.

La aplicación web esta realizada con el framework de angular el cual puede hacer aplicaciones de una sola página (SPA). La estructura de sus carpetas se puede apreciar en la **Fig. 6**, a continuación, se describe el contenido de esta.

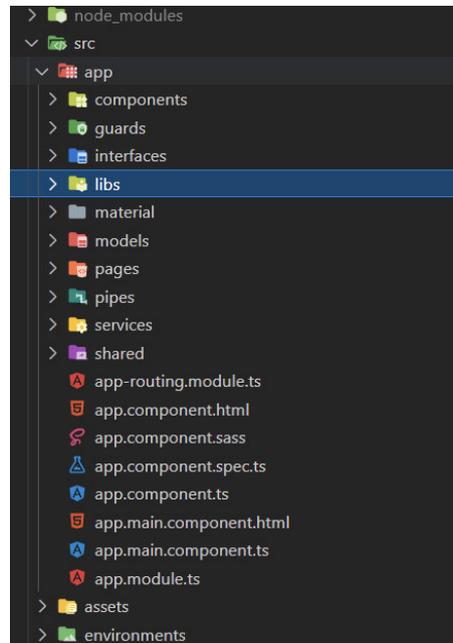


Fig. 6: Estructura proyecto EEP

- **node_modules:** contiene las librerías necesarias para la ejecución del proyecto.
- **src:** contiene los directorios y ficheros de toda la aplicación.
- **assets:** contiene imágenes de la aplicación.
- **environments:** contiene ficheros con la información a la conexión de la API.
- **app:** contiene los directorios y archivos fuente necesarios para el desarrollo de la aplicación.
 - **componentes:** esta carpeta contiene los componentes que se reutilizan a lo largo del proyecto como son los botones de acciones para diálogos, tablas y formularios.
 - **guards:** contiene un archivo que se encarga de validar las rutas a la que el usuario con ese rol tiene acceso.
 - **interfaces:** contiene archivos con interfaces para manejar el tipado de datos de forma extrita, además ayuda el momento de la codificación en el mapeo de los datos.

- **libs:** contiene los archivos que maneja la información del usuario logueado y el interceptor de respuestas del backend para mostrar mensajes de error o éxito en todo el proyecto.
- **material:** contiene un archivo de modulo en el que importa los módulos de angular material necesarias en todo el proyecto.
- **pages:** se encuentran las carpetas de las páginas de todo el proyecto en módulos independientes.
- **pipes:** contiene los filtros en tablas del sistema.
- **services:** se encuentran los archivos de servicios que se encarga de realizar las peticiones al backend.
- **shared:** contiene componentes que forman la parte visual de un usuario logeado.

Crear módulos y componentes para reutilización.

Se crean módulos de la carpeta components el cual se podrá importar y reutilizar en las páginas, también el módulo de material angular para el uso de diseños preestablecidos en formularios, tablas, etc. En la **Fig. 7** se observa un ejemplo.

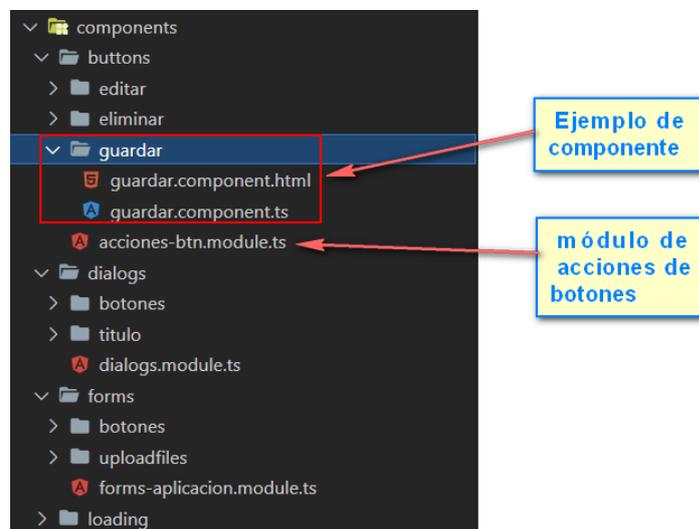


Fig. 7: Estructura de modulo y componentes reutilizables.

Crear servicio centrar que gestione información del usuario logueado.

En la **Fig. 8** se observa el archivo del servicio data-central que contiene código que maneja la información del usuario logueado encriptado con AES, los permisos, el menú asignado a su rol, validaciones, etc.

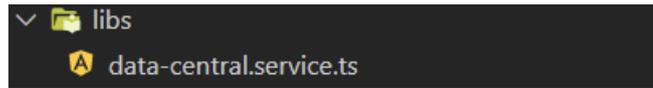


Fig. 8: Archivo data-central

Manejar acceso a rutas del sistema de forma dinámica

En la **Fig. 9** se observa el contenido del archivo `auth.guard.ts` contiene una función que permite el acceso a rutas según el menú del usuario que es consultado cada vez que recargan la página. De esta forma la única forma de acceder a una ruta es si tiene asignado desde el backend. El método funciona solo si el usuario está logeado, caso contrario lo envía a la ruta raíz del proyecto.

```
export class AuthGuard implements CanActivate {
  constructor(
    private loginService: LoginService,
    private router: Router,
    private dcentral: DataCentralService
  ) {}

  public get menu(): Menu[] {
    return this.dcentral.menu
  }

  canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): boolean {
    if (this.loginService.loggedIn()) {
      const url = this.menu.filter(o => {
        let [r] = o.hijos.filter(c => { return c.cruta == route.routeConfig.path })
        .map(c => {
          this.dcentral.setPermisos(c);
          return c;
        })
        return r !== undefined
      })

      if (url.length > 0) {
        return true;
      }
    }

    this.router.navigate(['']);
    return false;
  }
}
```

Fig. 9: Método que autoriza el acceso a rutas del sistema.

3.3 Sprint 2. Manejo de sesión activa

El Sprint 2 contiene actividades que permitirá el manejo de las sesiones de los usuarios, de sus permisos en las acciones y la recuperación de contraseña.

- Iniciar sesión.
- Recuperar contraseña.
- Manejar Roles con su respectivo menú.

Iniciar sesión

El inicio de sesión se ejecuta en el frontend y se valida en el backend. El mismo inicio de sesión sirva para cualquier tipo de rol. Solo en caso de ser estudiante tiene un tiempo de sesión de 5 minutos, caso contrario cualquier tipo de rol tiene acceso de 24 horas, luego de eso debe volver a logearse. En la **Fig. 10** se observa el formulario de inicio de sesión.

The image shows a login form titled "Inicia Sesión" centered on a dark blue background with a faint grid pattern. The form is a white rounded rectangle containing two input fields: "Ingrese correo electrónico *" with an envelope icon and "Ingrese contraseña *" with an eye icon. Below these is a blue "Ingresar" button. At the bottom, there is a link "Olvidaste tu contraseña? Recuperar".

Fig. 10: Formulario inicio de sesión.

Recuperar contraseña

El flujo de recuperación de contraseña empieza enviado el mail del usuario el cual el replicara un correo electrónico con el link de acceso para el cambio de contraseña del usuario correcto. En la **Fig. 11** se ve el formulario de envió del nombre del correo y en la **Fig. 12** se observa el formulario para el cambio de contraseña.



Fig. 13: Menú dinámico.

3.4 Sprint 3. Seguridad

El Sprint 3 contiene actividades de editar, visualizar y crear un registro para las tablas de transacción, rol y menú con la respectiva asignación de rol y transacciones. También se creó un método central con su respectivo componente para eliminar cualquier registro tan solo utilizando parámetros.

Cabe mencionar que se reutilizaron componentes de botones en los formularios con el objetivo de poder usar la parametrización de los permisos, de esta manera si un rol no tiene permiso para editar los registros de una determinada transacción no podrá hacerlo.

- Estándar de método para eliminar registro parametrizado.
- Crear transacción.
- Visualizar transacción.
- Editar transacción.
- Crear rol.
- Visualizar rol.
- Editar rol.
- Asignar transacción y rol a menú.
- Editar menú.

Estándar de método para eliminar registro parametrizado

Se creó un componente para la eliminación de registros de forma paramétrica, el cual envía el nombre de la tabla y el pk del registro, esta información se envía al backend el cual procesa la petición. Para llamar al componente se debe importar el módulo y colocar la etiqueta que invoca al componente. En la **Fig. 14** se despliega el mensaje de alerta cuando se presiona el botón o el icono de eliminación.



Fig. 14: Mensaje de alerta para eliminar un registro.

Crear transacción.

La definición de la transacción es importante si se escala la aplicación con nuevos módulos que se puede ir registrando desde esta sección. La parte importante en esta sección es ingresar la url y el path como el ejemplo de la **Fig. 15**

Un formulario con el título "Registra Transaccion". Tiene dos campos de texto: "Ruta url *" con el valor "listas/:id" y "Path modulo *" con el valor "src/app/pages/admin/listas/". Cada campo tiene un icono de ayuda (círculo con signo de interrogación). Al final hay dos botones: "Guardar" (azul) y "Cancelar" (rojo).

Fig. 15: Formulario de transacciones.

Visualizar transacción

La **Fig. 16** muestra una tabla con los registros de las transacciones previamente instaladas y necesarias para que el sistema funcione, de preferencia no modificar ni eliminar estos registros ya que el sistema dejaría de funcionar como es debido.

Transaccion		
Ruta	Path	Acciones
Buscar _____		
candidatos/:id	src/app/pages/admin/candidatos/candidatos.module	 
estudiantes	src/app/pages/admin/estudiantes/estudiantes.module	 
home-estudiante	src/app/pages/estudiante/principal-estudiante/principal-estudiante.module	 
listas/:id	src/app/pages/admin/listas/listas.module	 
opcionesmenu	src/app/pages/admin/seguridad/opcionesmenu/opcionesmenu.module	 

Items per page: 5 1 - 5 of 9 < > >> <<

Fig. 16: Tabla de transacciones ingresadas.

Editar transacción.

Se puede editar la transacción, pero no es recomendable modificar las transacciones que fueron definidas anteriormente. En la **Fig. 17** se observa el formulario que permite editar un registro.

Editar Transacción

Ruta url *

Path modulo *

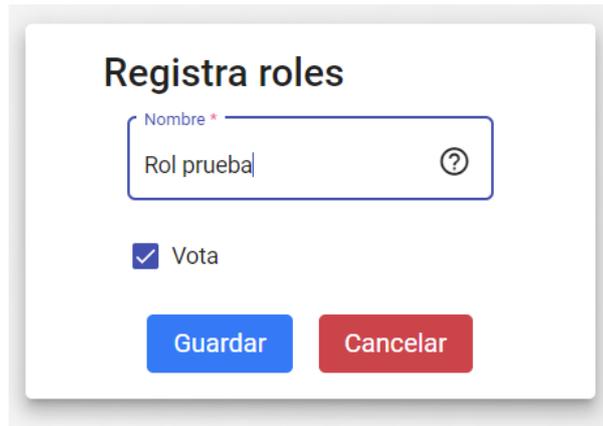
Guardar

Cancelar

Fig. 17: Formulario para editar transacción.

Crear rol

Se puede crear n cantidad de roles para el sistema, adicional se define si el rol puede votar o no. En la **Fig. 18** se observa el formulario de registro.



Registra roles

Nombre *
Rol prueba

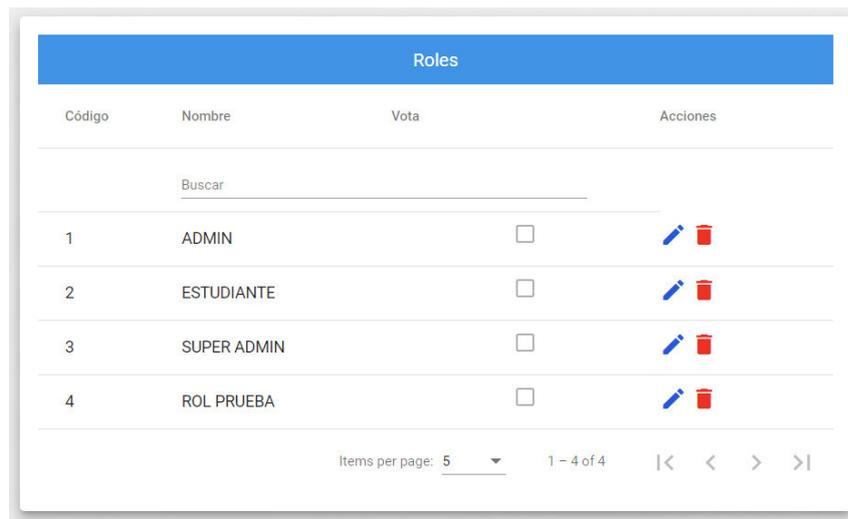
Vota

Guardar Cancelar

Fig. 18: Formulario de registro de roles.

Visualizar rol

La **Fig. 19** muestra la tabla con los registros de los roles que tiene el sistema, en la columna de acciones puede dar clic en el icono de editar o eliminar un registro.



Roles			
Código	Nombre	Vota	Acciones
Buscar			
1	ADMIN	<input type="checkbox"/>	 
2	ESTUDIANTE	<input type="checkbox"/>	 
3	SUPER ADMIN	<input type="checkbox"/>	 
4	ROL PRUEBA	<input type="checkbox"/>	 

Items per page: 5 1 - 4 of 4

Fig. 19: Tabla de roles ingresados.

Editar rol

Permite editar el nombre del rol o si puede votar o no. Editar un rol solo en caso de ser estrictamente necesario. La **Fig. 20** muestra el formulario para la edición del rol.

Fig. 20: Formulario para editar registro de rol. El formulario tiene un título "Editar Rol" en un encabezado azul. Debajo hay un campo de texto "Nombre *" con el valor "ROL PRUEBA" y un ícono de ayuda. A continuación, hay un campo de selección "Vota" con un ícono de verificación. En la parte inferior, hay dos botones: "Guardar" (azul) y "Cancelar" (rojo).

Fig. 20: Formulario para editar registro de rol.

Asignar transacción y rol a menú

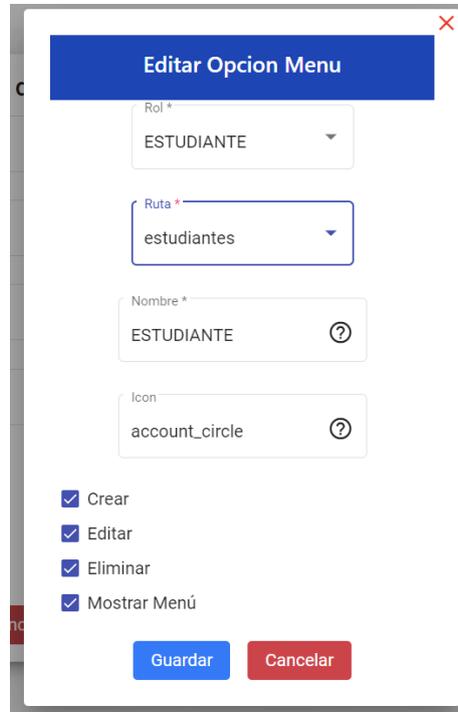
En este formulario se puede crear el menú, partiendo por seleccionar un rol, la transacción, escribir un nombre que aparecerá en el menú y es opcional colocar el nombre del icono a mostrar en el menú. Luego definir los permisos de crear, editar y eliminar que tendrá esa opción, finalmente el ítem de mostrar menú se usa para decirle al sistema si esa opción se mostrara o no en el apartado del menú dinámico, caso contrario solo se usará como un acceso a una transacción de forma oculta. La **Fig. 21** muestra el formulario para crear la opción de un menú.

Fig. 21: Formulario registrar menú de opciones. El formulario tiene un título "Registra opción de menú". Hay cuatro campos de selección: "Rol *" con el valor "ESTUDIANTE", "Ruta *" con el valor "estudiantes", "Nombre *" con el valor "ESTUDIANTE" y un ícono de ayuda, e "Icon" con un ícono de ayuda. Debajo, hay cuatro campos de selección con íconos de verificación: "Crear", "Editar", "Eliminar" y "Mostrar Menú". En la parte inferior, hay dos botones: "Guardar" (azul) y "Cancelar" (rojo).

Fig. 21: Formulario registrar menú de opciones.

Editar menú

En el formulario de la **Fig. 22** puede editar la opción del menú, reasignar permisos y mas opciones que el formulario permite.



El formulario 'Editar Opcion Menu' contiene los siguientes campos y elementos:

- Rol ***: Selector de lista desplegable con 'ESTUDIANTE' seleccionado.
- Ruta ***: Selector de lista desplegable con 'estudiantes' seleccionado.
- Nombre ***: Campo de texto con 'ESTUDIANTE' y un ícono de ayuda (?).
- Icon**: Campo de texto con 'account_circle' y un ícono de ayuda (?).
- Permisos**: Lista de casillas de verificación con los siguientes ítems:
 - Crear
 - Editar
 - Eliminar
 - Mostrar Menú
- Botones**: 'Guardar' (azul) y 'Cancelar' (rojo).

Fig. 22: Formulario para modificar registro de menú.

3.5 Sprint 4. Proceso Electoral, Listas participantes y Candidatos

El Sprint 4 se centra en la lógica del negocio, el cual es poder crear, editar, visualizar y eliminar procesos electorales con las respectivas listas y sus candidatos.

- Crear proceso electoral.
- Visualizar proceso electoral.
- Editar proceso electoral.
- Crear lista participante.
- Visualizar lista participante.
- Editar lista participante.
- Crear candidato.
- Visualizar candidato.
- Editar candidato.

Crear proceso electoral

El proceso electoral se define la descripción del proceso, el semestre que aplica, la fecha de la elección y la hora de inicio y finalización del proceso. Llenar todos los campos del formulario de la **Fig. 23**.



Registra nuevo proceso

Ingrese Descripción * 

Ingrese Semestre * 

Fecha de la elección * 

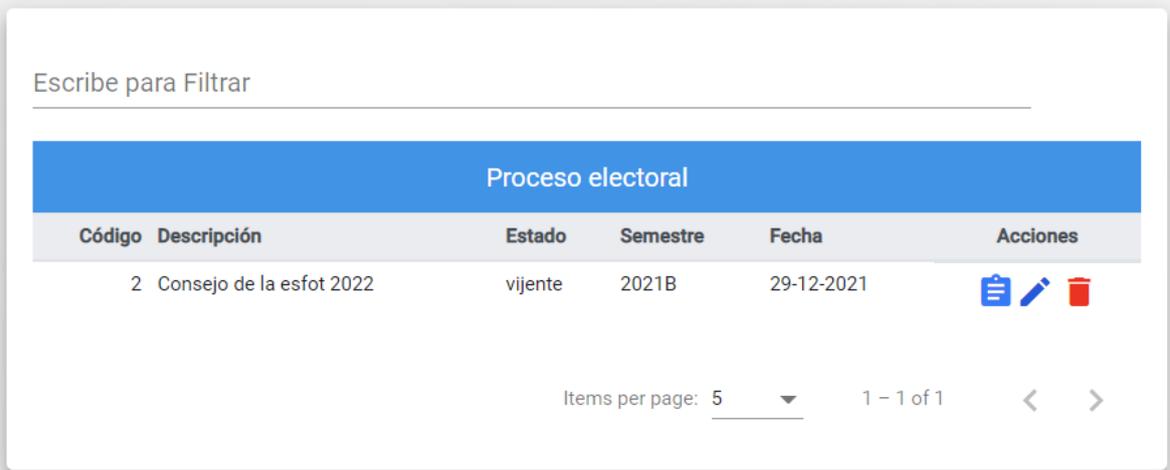
Hora de Inicio *
--:-- 

Hora de Fin *
--:-- 

Fig. 23: Formulario para registrar proceso electoral.

Visualizar proceso electoral

La **Fig. 24** muestra una tabla con los registros de procesos electorales vigentes o anteriores, en los cuales puede visualizar la información del proceso al dar clic en el icono de acciones.



Escribe para Filtrar

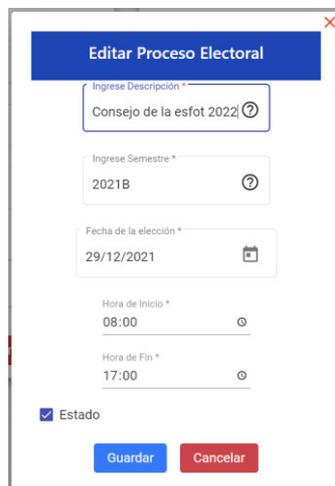
Proceso electoral					
Código	Descripción	Estado	Semestre	Fecha	Acciones
2	Consejo de la esfot 2022	vigente	2021B	29-12-2021	

Items per page: 5  1 - 1 of 1  

Fig. 24: Tabla que muestra los registros de procesos electorales.

Editar proceso electoral

Puede editar el proceso electoral solo si el proceso está vigente. En la **Fig. 25** muestra los campos del formulario para modificar el proceso electoral.



El formulario 'Editar Proceso Electoral' contiene los siguientes campos:

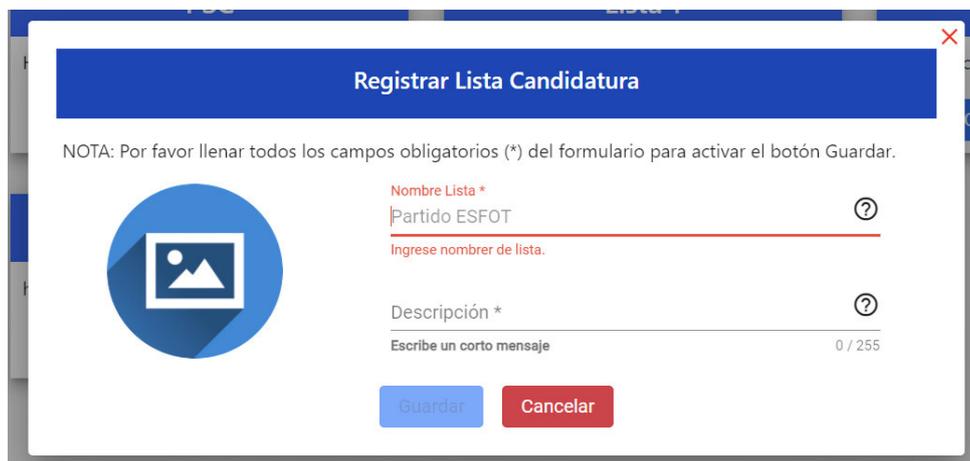
- Ingrese Descripción *: Consejo de la esfof 2022
- Ingrese Semestre *: 2021B
- Fecha de la elección *: 29/12/2021
- Hora de Inicio *: 08:00
- Hora de Fin *: 17:00
- Estado:

Botones: Guardar (azul), Cancelar (rojo).

Fig. 25: Formulario para modificar registro de proceso electoral.

Crear lista participante

La **Fig. 26** muestra el formulario para el registro de la lista participante en el proceso electoral. Tiene la opción de subir una foto que representa a la lista, es opcional subir la foto. Debe ingresar el nombre de la lista y una descripción.



El formulario 'Registrar Lista Candidatura' incluye:

- Nota: Por favor llenar todos los campos obligatorios (*) del formulario para activar el botón Guardar.
- Imagen: Icono de carga de foto.
- Nombre Lista *: Partido ESFOT
- Descripción *: Escribe un corto mensaje (0 / 255)

Botones: Guardar (azul), Cancelar (rojo).

Fig. 26: Formulario para registrar lista participante.

Visualizar lista participante.

La **Fig. 27** muestra información del proceso y sus respectivas listas participantes, se recomienda crear en cada proceso las opciones de voto nulo y voto en blanco. Además, en cada lista puede agregar sus respectivos candidatos.



Fig. 27: Listas participantes para ese proceso de elección.

Editar lista participante

El formulario de la **Fig. 28** puede modificar el registro de la lista participante.

Fig. 28: Formulario para editar lista participante.

Crear candidato

El formulario de la **Fig. 29** cuenta con los campos de nombres y apellidos del candidato, también debe ingresar el cargo al que postula.

Registra candidato

Ingrese nombres * 

Ingrese apellidos * 

Ingrese cargo postula * 

Fig. 29: Formulario para registrar candidato de una lista participante.

Visualizar candidato

La **Fig. 30** muestra la tabla de los registros de candidatos de una determinada lista. Puede editar o eliminar el registro dando clic en los iconos de acciones.

Código	Candidato	Cargo	Acciones
4	Kevin Curay	vicepresidente financiero	 

Items per page: 5 1 - 1 of 1 < >

Fig. 30: Tabla de registros de candidatos.

Editar candidato

El formulario de la **Fig. 31** muestra los campos que puede modificar del registro.



Editar candidato

Ingrese nombres * 

Ingrese apellidos * 

Ingrese cargo postula * 

Fig. 31: Formulario para modificar registro de candidato.

3.6 Sprint 5. Estudiante

El Sprint 5 se centra en el votante y la transacción en la cual se puede ejercer el voto, al ser transacciones paramétricas, esta transacción se puede asignar a cualquier rol. También con la importación de archivos se puede enviar una plantilla con los estudiantes recién matriculados para que realizar el proceso de forma rápida.

- Elegir y votar por un candidato.
- Importar plantilla de estudiantes.

Elegir y votar por un candidato.

El rol por defecto que puede sufragar es el rol de ESTUDIANTE, sin embargo, puede asignar la misma transacción a otro rol y darle permiso para ejercer el voto. La **Fig. 32** muestra las listas participantes en el proceso, dar clic en el botón de VOTAR para sufragar por una lista.



Fig. 32: Información el proceso y listas participantes.

Importar plantilla de estudiantes.

Para un rápido ingreso de estudiantes al sistema, tiene la opción de subir una plantilla con los datos de los estudiantes matriculados. Seleccionar el archivo de la plantilla y dar clic en enviar. La **Fig. 33** muestra un archivo cargado para enviar.



Fig. 33: Formulario para subir plantilla de estudiantes.

3.7 Sprint 6. Resultados Votación

El Sprint 6 contiene la actividad para mostrar los resultados de las votaciones.

- Mostrar el conteo de votantes y los resultados del proceso.

Mostrar el conteo de votantes y los resultados del proceso.

La transacción solicita toda la información de los votos y los calcula de forma dinámica en los resultados como se observa en la **Fig. 34**.

```

Lista 1: 1 votos
Voto nulo: 1 votos
lista 2: 1 votos
[
  {
    "index": 0,
    "hashAnterior": "",
    "timestamp": 1648032163835,
    "data": "Bloque Genesis",
    "hash": "f7c33b7ed5dca61eafb469c4319f17e4ec3b1d4e4590ad68501c99ae53c50661"
  },
  {
    "index": 1,
    "hashAnterior": "f7c33b7ed5dca61eafb469c4319f17e4ec3b1d4e4590ad68501c99ae53c50661",
    "timestamp": 1648037984321,
    "data": {
      "id": 2,
      "nom_lista": "Lista 1",
      "descripcion": "vamos a mejorar",
      "logo": null,
      "estado": true,
      "id_proceso": 2
    },
    "hash": "56a16345bcf75f3ed7bdeef41b6948d878e9837c02768d4c1dce51baad0ebcc9"
  },
]

```

Fig. 34: Conteo de votos.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En la siguiente sección se enlista las conclusiones que se obtuvo luego del desarrollo del proyecto.

- Un sistema de votación online transforma la forma de realizar este proceso de manera tradicional, ya que por medio de un link puede un usuario ejercer su derecho a votar, sin la necesidad de ir de forma presencial a la institución, lo cual ayuda a incrementar el número de participantes en el proceso electoral.
- Los sistemas de votación requieren de varias características como es la integridad de los votos, cada voto sea completamente anónimo, la transparencia del proceso y un rápido acceso desde cualquier navegador.
- Angular es un framework muy útil al momento de crear aplicaciones web, permite la creación de módulos independientes los cuales pueden ser cargados usando el concepto de Lazy Loading, de esta forma se optimizan los recursos de la aplicación, además al parametrizar y reutilizar componentes como botones para formularios, botones de acciones en tablas, etc. en todo el proyecto, se puede escalar aún más la aplicación en poco tiempo.
- La API REST está conectada directamente con la base de datos, se utilizaron dos formas de conexión, una a través de ORM (Mapeo Objeto Relacional) que se implementó para las tablas de Usuario, Rol, Transacción y Menú, la otra forma de conexión es directa para manejar a través de sentencias.
- La seguridad en el sistema esta manejada a través de tokens de sesión usando JWT, estos tokens tienen un tiempo valido en el cual se almacena información como los permisos del Menú, su identificador Único y el rol al que pertenece. Esta información es constantemente validada en la aplicación web para el manejo de los recursos, conseguimos seguridad, al restringir el acceso no autorizado a transacciones y menos a la manipulación de los registros.

4.2 Recomendaciones

Las siguientes recomendaciones ayudaran a un mejor entendimiento del proyecto.

- En el proyecto se plasmó todos los conocimientos técnicos adquiridos en clases, sin embargo, se recomienda adquirir conocimientos avanzados para el uso del Framework de Angular, así como el uso de la tecnología de blockchain, la construcción de una API REST desde cero y conexión a bases de datos usando ORM.
- Se recomienda usar el estándar de los dialogs, botones y más componentes reutilizables, ya que de esta manera agiliza el desarrollo de nuevas pantallas, es importante no olvidar parametrizar el nuevo módulo con una transacción, el rol y el menú para poder visualizarlo en pantalla.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] E. Junta, «Esfot,» 9 Julio 2019. [En línea]. Available: <https://esfot.epn.edu.ec/index.php/esfot/339-res-elec-jul-2019>.
- [2] «Acta de votaciones,» de *Elecciones de los representantes de la asociacion de estudiantes de la escuela de formacion de tecnologos A.E.ESFOT*, Quito, 2019.
- [3] G. L. a. N. Espinoza, «Electronic Voting,» Stanford University, 2007. [En línea]. Available: https://cs.stanford.edu/people/eroberts/courses/cs181/projects/2006-07/electronic-voting/index_files/page0002.html. [Último acceso: 30 Junio 2021].
- [4] B. University, «<https://scet.berkeley.edu/>,» 16 Octubre 2015. [En línea]. Available: <http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf>. [Último acceso: 30 Junio 2021].
- [5] J. Tavira, «Santander Global Tech,» Santander , 12 Julio 2019. [En línea]. Available: <https://santanderglobaltech.com/concepto-consenso-blockchain-relevancia-consecuencias/>. [Último acceso: 8 Julio 2021].
- [6] I. Cañete, «bbva.com,» 11 Marzo 2019. [En línea]. Available: <https://www.bbva.com/es/que-es-un-product-backlog-y-cual-es-su-funcion/>. [Último acceso: 27 Marzo 2022].
- [7] Integra IT, «Integra IT,» [En línea]. Available: <https://integrait.com.mx/blog/sprint-y-sprint-backlog/>. [Último acceso: 27 Marzo 2022].
- [8] D. CALDENTEY, «Unir la universidad en internet,» 9 Febrero 2022. [En línea]. Available: <https://www.unir.net/ingenieria/revista/justinmind-la-herramienta-que-se-estudia-en-stanford-y-berkeley-llega-a-unir/>.
- [9] Universidad de Alicante, «UA,» Universidad de Alicante, [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 9 Febrero 2022].
- [10] C. Simões, «www.itdo.com,» ITDO, 21 Julio 2021. [En línea]. Available: <https://www.itdo.com/blog/que-es-node-js-y-para-que-sirve/>. [Último acceso: 1 Marzo 2022].
- [11] C. Simões, «www.itdo.com,» ITDO, 1 Marzo 2022. [En línea]. Available: <https://www.itdo.com/blog/que-es-typescript-y-por-que-utilizarlo/>. [Último acceso: 6 Julio 2021].
- [12] Quality devs, «Quality Devs,» 16 Septiembre 2019. [En línea]. Available: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>. [Último acceso: 3 Marzo 2022].

- [13] S. Veliz, «Medium,» 29 Marzo 2019. [En línea]. Available: <https://medium.com/@sandy.e.veliz/angular-material-design-instalaci%C3%B3n-angular-material-790caca5677b>. [Último acceso: 4 Marzo 2022].
- [14] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [15] Santander Universidades, «Santander becas,» Santander, 21 Diciembre 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 26 Marzo 2022].
- [16] Nicole maldeadora, «Platzi,» 2018. [En línea]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/>. [Último acceso: 28 Marzo 2022].
- [17] John Díaz, «EDteam,» 2020. [En línea]. Available: <https://ed.team/blog/que-es-backend-y-frontend-guia-completa>. [Último acceso: 27 Marzo 2022].
- [18] grupoconsultorefe, «gcefe,» [En línea]. Available: <https://grupoconsultorefe.com/servicio/tecnologias-de-la-informacion/sistemas-web>. [Último acceso: 27 Marzo 2022].

6 ANEXOS

A continuación, se presentan los siguientes anexos que complementan y transparentan la información del proyecto integrador

ANEXO I. Turnitin porcentaje máximo 12%.

ANEXO II. Manual técnico

ANEXO III. Manual de usuario

ANEXO IV. Manual de instalación

ANEXO I. Certificado de Originalidad

El este apartado se observa el certificado de originalidad emitido por el director del proyecto.

ANEXO II. Manual Técnico.

1. RECOPIACIÓN DE REQUERIMIENTOS

La **TABLA IV** enumera y lista los requerimientos que la aplicación web necesita, la información fue tomada en base a las reuniones con los clientes.

TABLA IV: Tabla de requerimientos.

# Requerimiento	Detalle
1	Inicio de sesión de la aplicación al sistema web consta del ingreso de credenciales como es el correo electrónico de la EPN(@epn.edu.ec) y la contraseña.
2	El sistema debe manejar n cantidad de roles y asignar permisos para cada uno de ellos.
3	Se define tres roles por defecto que son: SuperAdmin – Admin – Estudiante. El estudiante puede votar por defecto y los usuarios con otro tipo de rol solo podrá votar si se asigna la opción de votar en las elecciones.
4	El sistema debe manejar el concepto de transacciones que se asignaran a un rol por medio de un menú de opciones.
5	Un usuario con permisos suficientes para crear, editar o eliminar el proceso electoral, el partido y la lista de candidatos.
6	Un usuario con permisos suficientes para crear, editar o eliminar las políticas de seguridad como son la definición de menús, los roles y las transacciones del sistema.
7	Un usuario que pueda dar mantenimiento a los usuarios con rol de estudiantes, además de importar estudiantes de nuevo ingreso al sistema.
8	Verificar los resultados de las votaciones.

2. HISTORIAS DE USUARIO

A continuación, se presenta un total de XIII historias de usuario generadas del análisis y levantamiento de requerimientos para el desarrollo del sistema web.

TABLA V: Historia de usuario Nro.1: Login de acceso a usuarios con distintos roles.

Historia de usuario	
Identificador: LOG1	Usuario: Usuario de cualquier Rol
Nombre Historia: Login de acceso a usuarios con distintos roles.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: Inicio de sesión desde la aplicación web, válida para cualquier usuario, en la validación de la API devolver el rol que el usuario tiene con su menú de opciones y permisos de creación, edición y eliminación de registros en cada una de las transacciones.	
Observaciones: <ul style="list-style-type: none"> • Validaciones en el formulario de inicio de sesión. • Tiempo de sesión parametrizado por rol. • Validaciones en la API Rest. 	

TABLA VI: Historia de usuario Nro.2: Reestablecer contraseña

Historia de usuario	
Identificador: LOG2	Usuario: Usuario de cualquier Rol
Nombre Historia: Reestablecer contraseña.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	

Descripción: El usuario podrá reestablecer su contraseña por medio de un correo el cual le entrega un link en donde podrá ingresar la nueva contraseña.
Observaciones: <ul style="list-style-type: none"> • Validaciones en el formulario de confirmación de contraseña iguales • Validaciones en la API Rest.

TABLA VII: Historia de usuario Nro.3: Cerrar sesión del sistema web.

Historia de usuario	
Identificador: LOG3	Usuario: Usuario de cualquier Rol
Nombre Historia: Cerrar sesión del sistema web.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: Cierre de sesión.	
Observaciones: <ul style="list-style-type: none"> • Elimina información del localStorage y el sessionStorage del navegador • Cierre de sesión por caducidad de token. 	

TABLA VIII: Historia de usuario Nro.4: Selecciona candidato y vota.

Historia de usuario	
Identificador: VOT1	Usuario: Votante
Nombre Historia: Selecciona candidato y vota.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	

Descripción: El usuario podrá elegir al candidato o a su vez la opción de voto en blanco o nulo.
Observaciones: <ul style="list-style-type: none"> • Validación del voto sea registrado en el sistema y que sea anónimo el voto. • Una vez realizado el voto no puede volver a votar.

TABLA IX: Historia de usuario Nro.5: Registra, edita y elimina proceso electoral

Historia de usuario	
Identificador: ADM1	Usuario: Rol Administradores
Nombre Historia: Registra, edita y elimina proceso electoral.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: El usuario podrá realizar un CRUD de los procesos electorales.	
Observaciones: <ul style="list-style-type: none"> • Asignación de la fecha que llevara a cabo el proceso electoral. • Validar formulario de ingreso y edición de registro. 	

TABLA X: Historia de usuario Nro.6: Registra, edita y elimina lista participante

Historia de usuario	
Identificador: ADM2	Usuario: Rol Administradores
Nombre Historia: Registra, edita y elimina lista participante.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	

Descripción: El usuario podrá realizar un CRUD de las listas participantes para un determinado proceso electoral.
Observaciones: <ul style="list-style-type: none"> • La lista participante debe estar ligada a un proceso electoral. • Validar formulario de ingreso y edición de registro.

TABLA XI: Historia de usuario Nro.7: Registra, edita y elimina candidatos de una lista.

Historia de usuario	
Identificador: ADM3	Usuario: Rol Administradores
Nombre Historia: Registra, edita y elimina candidatos de una lista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: El usuario podrá realizar un CRUD de los candidatos de una determinada lista participante.	
Observaciones: <ul style="list-style-type: none"> • Los candidatos pertenecen a una sola lista participante. • Validar formulario de ingreso y edición de registro. 	

TABLA XII: Historia de usuario Nro.8: Importar archivo de estudiantes.

Historia de usuario	
Identificador: ADM4	Usuario: Rol Administradores
Nombre Historia: Importar archivo de estudiantes.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 2	Iteración asignada: 1
Programador responsable: Kevin Curay	

Descripción: El usuario podrá importar un archivo Excel con la información relevante para crear nuevos usuarios con rol estudiante.
Observaciones: <ul style="list-style-type: none"> Validación en la API Rest, controla si el usuario existe solo actualizarlo caso contrario ingresarlo como nuevo.

TABLA XIII: Historia de usuario Nro.9: CRUD de rol.

Historia de usuario	
Identificador: ADM5	Usuario: Rol Administradores
Nombre Historia: CRUD de rol.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Días estimados: 1	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: El usuario podrá realizar un CRUD de los roles que se crearan para el sistema.	
Observaciones: <ul style="list-style-type: none"> Validación en el formulario de ingreso o edición. 	

TABLA XIV: Historia de usuario Nro.10: CRUD de transacciones.

Historia de usuario	
Identificador: ADM6	Usuario: Rol Administradores
Nombre Historia: CRUD de transacciones.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Días estimados: 1	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: El usuario podrá realizar un CRUD de las transacciones que se definan de acuerdo con el progreso del proyecto.	

Observaciones: <ul style="list-style-type: none"> Validación en el formulario de ingreso o edición.

TABLA XV: Historia de usuario Nro.11: Políticas de seguridad (menú).

Historia de usuario	
Identificador: ADM7	Usuario: Rol Administradores
Nombre Historia: Políticas de seguridad (menú).	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 1	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: El usuario podrá realizar un crud de los permisos para asignar un menú de una determinada transacción a un rol.	
Observaciones: <ul style="list-style-type: none"> Validación en el formulario de registro. 	

TABLA XVI: Historia de usuario Nro.12: Registro individual de usuario.

Historia de usuario	
Identificador: ADM8	Usuario: Rol Administradores
Nombre Historia: Registro de usuario individual.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Días estimados: 1	Iteración asignada: 1
Programador responsable: Kevin Curay	
Descripción: El usuario administrador podrá registrar un nuevo usuario asignando un rol.	
Observaciones: <ul style="list-style-type: none"> Validación en el formulario de registro. 	

3. PRODUCT BACKLOG

La **TABLA XVII** detalla el Product Backlog que sirve como guía a nivel macro el desarrollo del proyecto, se lo realiza en base a la información de las Historias de Usuario.

TABLA XVII: Product Backlog: Ordenada por prioridad.

Identificador	Usuario	Historia	Días Estimados	Prioridad
ADM5	Rol Administradores	CRUD de roles	1	10
ADM6	Rol Administradores	CRUD de transacciones	1	10
ADM7	Rol Administradores	Políticas de seguridad (menu)	1	10
ADM8	Rol Administradores	Registrar usuario	1	10
LOG1	Usuario de cualquier Rol	Inicio de sesión	2	9
LOG2	Usuario de cualquier Rol	Reestablecer contraseña	2	9
LOG3	Usuario de cualquier Rol	Cerrar sesión	2	9
ADM1	Rol Administradores	Registra, edita y elimina proceso electoral.	2	8
ADM2	Rol Administradores	Registra, edita y elimina lista participante.	2	8
ADM3	Rol Administradores	Registra, edita y elimina candidatos de una lista.	2	8
ADM4	Rol Administradores	Importar archivo de estudiantes.	2	8
VOT1	Votante	Selecciona candidato y vota.	2	7

4. SPRINT BACKLOG

A continuación, se muestra las actividades a realizar dentro de cada Sprint. Para la elaboración del Sprint Backlog se toma las características definidas en el Product Backlog.

TABLA XVIII: Sprint Backlog

ID	Nombre	Tareas	Duración
0	Definición estructura de proyecto backend	<ul style="list-style-type: none"> • Crear BDD. • Estructura proyecto de la API REST. 	2
1	Definición estructura de proyecto frontend	<ul style="list-style-type: none"> • Estructura proyecto con angular. • Crear módulos y componentes para reutilización. • Crear servicio centrar que gestione información del usuario logueado. • Manejar acceso a rutas del sistema de forma dinámica. 	30
2	Manejo de sesión activa	<ul style="list-style-type: none"> • Iniciar sesión. • Recuperar contraseña. • Manejar Roles con su respectivo menú. 	15
3	Seguridad	<ul style="list-style-type: none"> • Estándar de método para eliminar registro parametrizado. • Crear transacción. • Visualizar transacción. • Editar transacción. • Crear rol. • Visualizar rol. • Editar rol. • Asignar transacción y rol a menú. • Editar menú. 	9
4	Proceso Electoral, Listas participantes y Candidatos	<ul style="list-style-type: none"> • Crear proceso electoral. • Visualizar proceso electoral. • Editar proceso electoral. • Crear lista participante. • Visualizar lista participante. • Editar lista participante. • Crear candidato. • Visualizar candidato. • Editar candidato. 	20
5	Estudiante	<ul style="list-style-type: none"> • Elegir y votar por un candidato. 	10

		<ul style="list-style-type: none"> • Importar plantilla de estudiantes. 	
6	Resultados Votación	<ul style="list-style-type: none"> • Mostrar el conteo de votantes y los resultados del proceso. 	8
-	Pruebas en el sistema web.	<ul style="list-style-type: none"> • Prueba de usabilidad. • Prueba de carga. 	10
-	Despliegue del sistema web	<ul style="list-style-type: none"> • Despliegue sistema web. 	20

5. PROTOTIPO DE INTERFACES

En este apartado se presenta los prototipos de interfaz gráfica para el sistema web.

6. DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos SQL implementada en PostgreSQL, mediante DDL definir los datos y sus relaciones, además de DML para manipular los datos.

La tabla de **usuarios** consta de los siguientes atributos:

- ID: Atributo de tipo Number, pk secuencial.
- USERNAME: Atributo de tipo String, almacena el sobrenombre.
- NOMBRE: Atributo de tipo String, almacena el nombre.
- APELLIDO: Atributo de tipo String, almacena el apellido.
- CEDULA: Atributo de tipo String, almacena la identificación.
- PASSWORD: Atributo de tipo String, almacena la contraseña.
- EMAIL: Atributo de tipo String, almacena el email.
- ACTIVO: Atributo de tipo Boolean, almacena el estado del usuario.
- ROL: Atributo de tipo Number, fk del rol asignado.
- SUFRAGO: Atributo de tipo Boolean, almacena si ya sufragó o no.
- ESTUDIANTE: Atributo de tipo Boolean, referencia si el usuario es estudiante o no.
- CREATEDAT: Atributo de tipo Timestamp, almacena fecha que se crea el registro.

- UPDATEDAT: Atributo de tipo Timestamp, almacena fecha que se actualiza el registro.

La tabla de **rol** consta de los siguientes atributos:

- ID: Atributo de tipo Number, pk secuencial.
- NOMBRE: Atributo de tipo String, almacena el nombre del rol.
- VOTA: Atributo de tipo Boolean, almacena si el rol puede votar.

La tabla de **transacciones** consta de los siguientes atributos:

- RUTA: Atributo de tipo String, almacena la url de la página de angular.
- PATH: Atributo de tipo String, almacena el path o dirección del módulo de angular que se ejecuta.

La tabla de **menú** consta de los siguientes atributos:

- ID: Atributo de tipo Number, pk secuencial.
- ID_ROL: Atributo de tipo Number, fk Rol.
- CRUTA: Atributo de tipo String, fk Transacción.
- ID_PADRE: Atributo de tipo Number, fk ID de menú.
- NOMBRE: Atributo de tipo String, nombre del menú.
- ICON: Atributo de tipo String, nombre del icon.
- CREAR: Atributo de tipo Boolean, si tiene permiso para crear.
- EDITAR: Atributo de tipo Boolean, si tiene permiso para editar.
- ELIMINAR: Atributo de tipo Boolean, si tiene permiso para eliminar.
- MOSTRARMENU: Atributo de tipo Boolean, para usar en el menú.

La tabla de **proceso_electoral** consta de los siguientes atributos:

- ID: Atributo de tipo Number, pk secuencial.
- DESCRIPCION: Atributo de tipo String, descripción de proceso electoral.
- ESTADO: Atributo de tipo Boolean, estado el proceso electoral.
- SEMESTRE: Atributo de tipo String, semestre del proceso.
- FEC_ELECCION: Atributo de tipo Date, fecha de la elección.
- HORA_INICIO: Atributo de tipo Time, hora de inicio.
- HORA_FINAL: Atributo de tipo Time, hora de fin.

La tabla de **lista_electoral** consta de los siguientes atributos:

- ID: Atributo de tipo Number, pk secuencial.
- NOM_LISTA: Atributo de tipo String, nombre de la lista electoral.
- DESCRIPCION: Atributo de tipo String, descripción de la lista electoral.
- LOGO: Atributo de tipo String, nombre del archivo del logo.
- ESTADO: Atributo de tipo Boolean, estado de la lista.
- ID_PROCESO: Atributo de tipo Number, fk del proceso electoral.
- CONTENIDO: Atributo de tipo Text, almacena una imagen codificada a un string en base64.

La tabla de **candidatos** consta de los siguientes atributos:

- ID: Atributo de tipo Number, pk secuencial.
- NOMBRE: Atributo de tipo String, almacena el nombre del candidato.
- APELLIDO: Atributo de tipo String, almacena el apellido del candidato.
- CARGO: Atributo de tipo String, cargo que desempeñaría el candidato.
- ID_LISTA: Atributo de tipo Number, fk de la lista electoral.

ANEXO III. Manual de usuario

En esta sección se agrega un link del video del manual de usuario del sistema web para la votación electrónica.

<https://www.youtube.com/watch?v=pi6xiHleRk0>

ANEXO IV. Manual de instalación

El código fuente del sistema se encuentra alojado en github junto con la configuración para desplegar la aplicación con Docker.

Credenciales de acceso:

Por despliegues continuos y gratuitos el link para ingresar al sistema web se encuentra en el repositorio de github.

Repositorio del código fuente:

En el repositorio de GitHub se encuentra alojado el código fuente del sistema web que contempla el frontEnd con angular, la API REST con nodejs y los scripts de la base de datos. En el siguiente link puede acceder al repositorio.

<https://github.com/kaxer1/eep-esfot>