



MANUAL TÉCNICO

Manual técnico del sistema de votación del proyecto universitario

27 DE OCTUBRE DE 2023

@estuardodev

estuardo.dev

[Proyecto de Algoritmos UMG \(github.com\)](https://github.com)

Índice

INTRODUCCIÓN	1
OBJETIVOS TÉCNICOS	1
ALCANCE DEL PROYECTO	1
<i>Configuración Inicial del Sistema</i>	<i>1</i>
<i>Registro de Usuarios</i>	<i>1</i>
<i>Registro de Votantes</i>	<i>1</i>
<i>Administración de Elecciones.....</i>	<i>1</i>
<i>Acceso a Votantes al Sistema.....</i>	<i>1</i>
DIAGRAMA DE CASOS DE USO	2
DIAGRAMA DE CLASES	3
ALGORITMO DE LA SOLUCIÓN	5
DIAGRAMA DE FLUJO	¡ERROR! MARCADOR NO DEFINIDO.
TECNOLOGÍAS UTILIZADAS.....	17
INSTRUCCIONES DE CONFIGURACIÓN DEL ENTORNO DE TRABAJO	17
REQUISITOS TÉCNICOS	17
CLONACIÓN DEL PROYECTO.....	17
PROCESO DE COMPILACIÓN DEL PROYECTO	18
SOLUCIÓN DE PROBLEMAS COMUNES	18
INFORMACIÓN DE CONTACTO.....	19
GLOSARIO	20
REFERENCIAS	21

Introducción

Objetivos Técnicos

El objetivo del presente proyecto es un sistema informático para votaciones que permita agilizar y optimizar el proceso de votaciones de diferentes contextos, ya sea para ámbitos políticos, empresariales u otros tipos de votaciones democráticas. El sistema está diseñado para reemplazar o complementar el sistema tradicional de votación en papel y lápiz, mediante la implementación de la tecnología. El enfoque principal del proyecto es brindar una solución que permita la administración eficiente y transparente de elecciones, registros de votantes, autenticación segura y generación de reportes detallados de los resultados electorales.

Alcance del Proyecto

El sistema de votación desarrollado contiene las siguientes funcionalidades:

Configuración Inicial del Sistema

- Establecer una contraseña única e inmutable para el administrador del sistema.

Registro de Usuarios

- Registrar, editar y deshabilitar usuarios.
- Capturar información de los usuarios.
- Roles: administrador, registrador de votantes, votante y auditor.
- Permitir la generación de nuevas contraseñas y restablecimiento de las contraseñas utilizando el correo electrónico.

Registro de Votantes

- Permitir a los registradores de votantes agregar nuevos votantes.
- Capturar información detallada de los votantes.
- Generar contraseñas aleatorias para los nuevos votantes.
- Dar de baja votantes por motivos de fallecimiento.

Administración de Elecciones

- Permitir a los administradores crear nuevas elecciones.
- Configurar fechas y horarios.
- Eliminar elecciones.
- Gestionar candidatos.

Acceso a Votantes al Sistema

- Autenticación segura al sistema para el usuario.
- Emitir votos media vez sea válido.

Diagrama de Casos de Uso

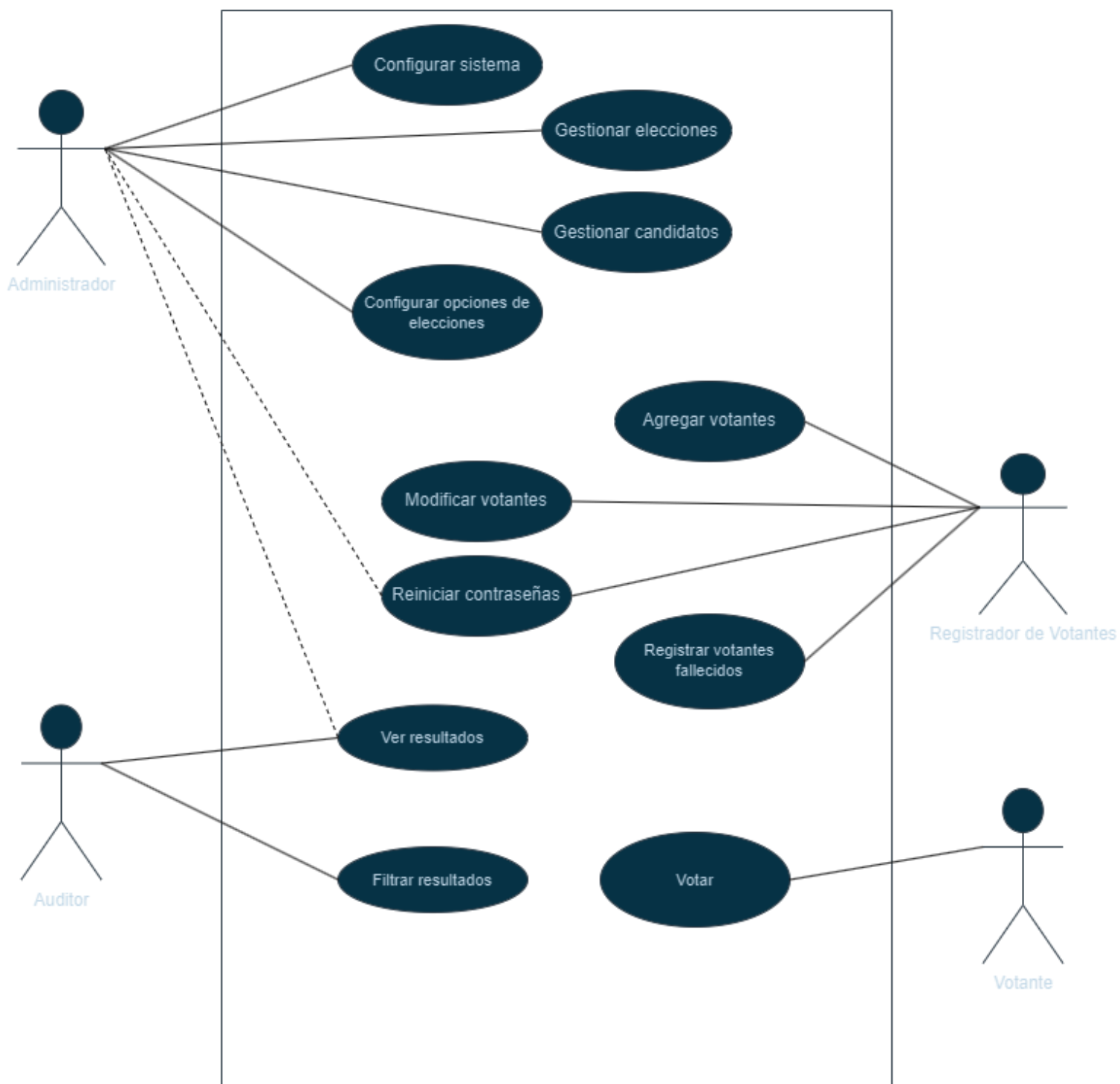


Diagrama de Clases



RegistroUsuarios

- USERS
- AUDITORS
- REGISTERS

+ HomeRegistroUsuario()
+ RegistroUsuarioCrear()
+ RegistroUsuarioEditar()
+ RegistroUsuarioDeshabilitar()

VotanteRegistro

- USERS
- REGISTRO_VOTOS

+ RegistroVotos()
- existeRegistro()
- agregarInformacion()
- crearNuevoRegistro()

ConfigElecciones

- ELECTIONS

+ HomeConfigElecciones()
- HoraVotacion()
- Mostrar()
- establecerFechasVotacion()
- establecerFechasInscripcion()
- HorasInscripcion()

Candidatos

- ELECTIONS
+ CANDIDATOS
+ CANDIDATOS_ELECCION

+ GestionCandidatos()
+ GestionarCandidatosCrear()
+ GestionarCandidatosEliminar()
+ GestionarCandidatosAsignar()

Administrador

- ELECTIONS

+ VerificarAdmin
+ LoginAdmin
+ HomeAdmin

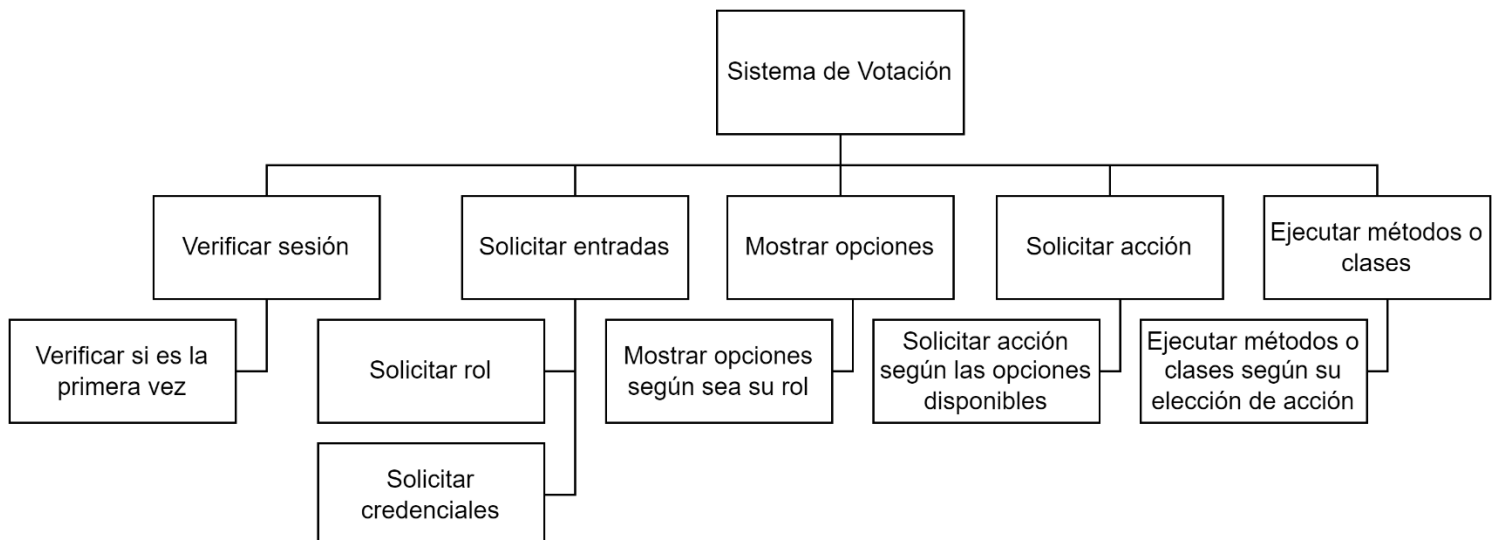
Algoritmo de la Solución

Algoritmo Principal

Análisis del Problema

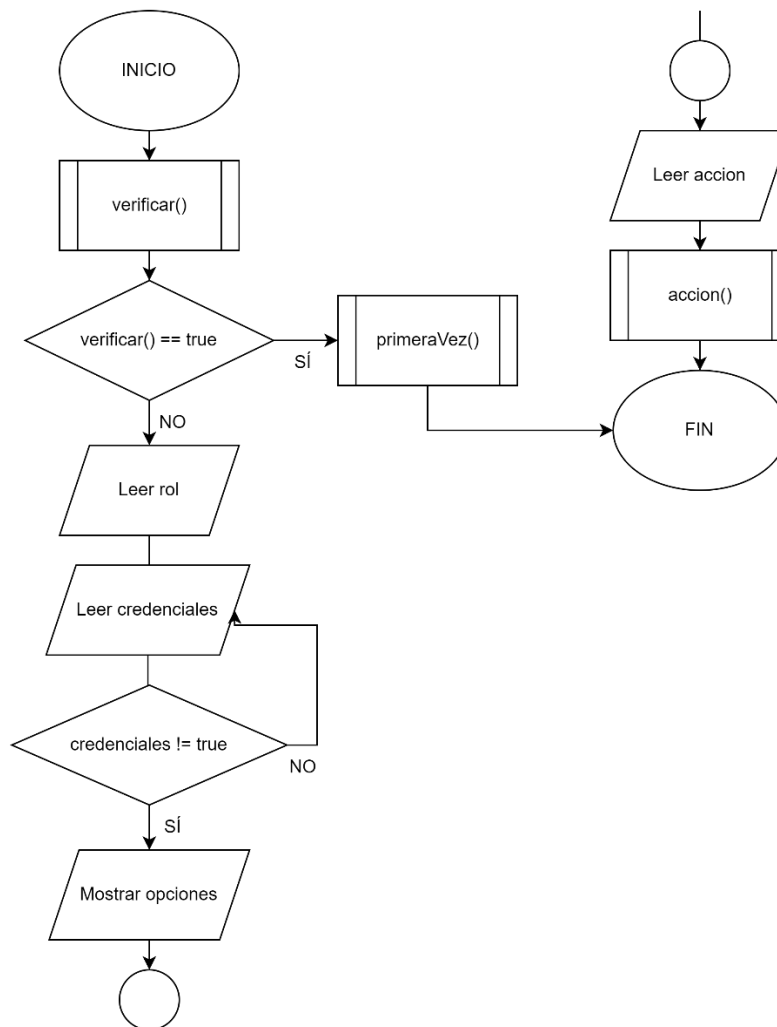
1. ¿Qué entradas se requieren?
 - a. Rol para ejercer
 - b. Credenciales del rol
2. ¿Cuál es la salida deseada?
 - a. Opciones para ejercer una actividad
3. ¿Qué método produce la salida deseada?
 - a. Verificar si es la primer sesión, si es así ejecutar primeraVez()
 - b. Si no es la primera vez, solicitar rol
 - c. Luego del rol seleccionado, mostrar opciones y solicitar acción a realizar
4. Requerimientos adicionales y/o restricciones
 - a. Solo se solicitará la contraseña de administrador si es la primera vez que se ejecuta el programa
 - b. La contraseña del administrador es inmutable
 - c. Tener instalado Java JDK 17 o superior

Diseño descendente



Algoritmo

1. Entradas: rol, credenciales
2. Salidas: Opciones para ejercer una actividad
3. Proceso
 - a. Inicio
 - b. Ver si es la primera vez que se ejecuta el programa ejecutando verificar()
 - c. Si es la primera vez, ejecutar primeraVez()
 - d. Si no lo es, solicitar el rol a ejercer
 - e. Solicitar las credenciales del rol
 - f. Si las credenciales no son las mismas a las del archivo, volver a solicitar
 - g. Mostrar opciones posibles por el rol
 - h. Solicitar acción a realizar
 - i. Ejecutar método dependiendo del caso que se presente
 - j. Fin

Diagrama de Flujo

Sub-Algoritmo Home

Análisis del problema

1. ¿Qué entradas se requieren?
 - a. Opción de la acción a realizar el usuario
2. ¿Cuál es la salida deseada?
 - a. Acción solicitada por el usuario
3. ¿Qué método produce la salida deseada?
 - a. Solicitar opción al usuario y ejecutar el método correspondiente
5. Requerimientos adicionales y/o restricciones
 - a. Un votante no puede ingresar a otros roles

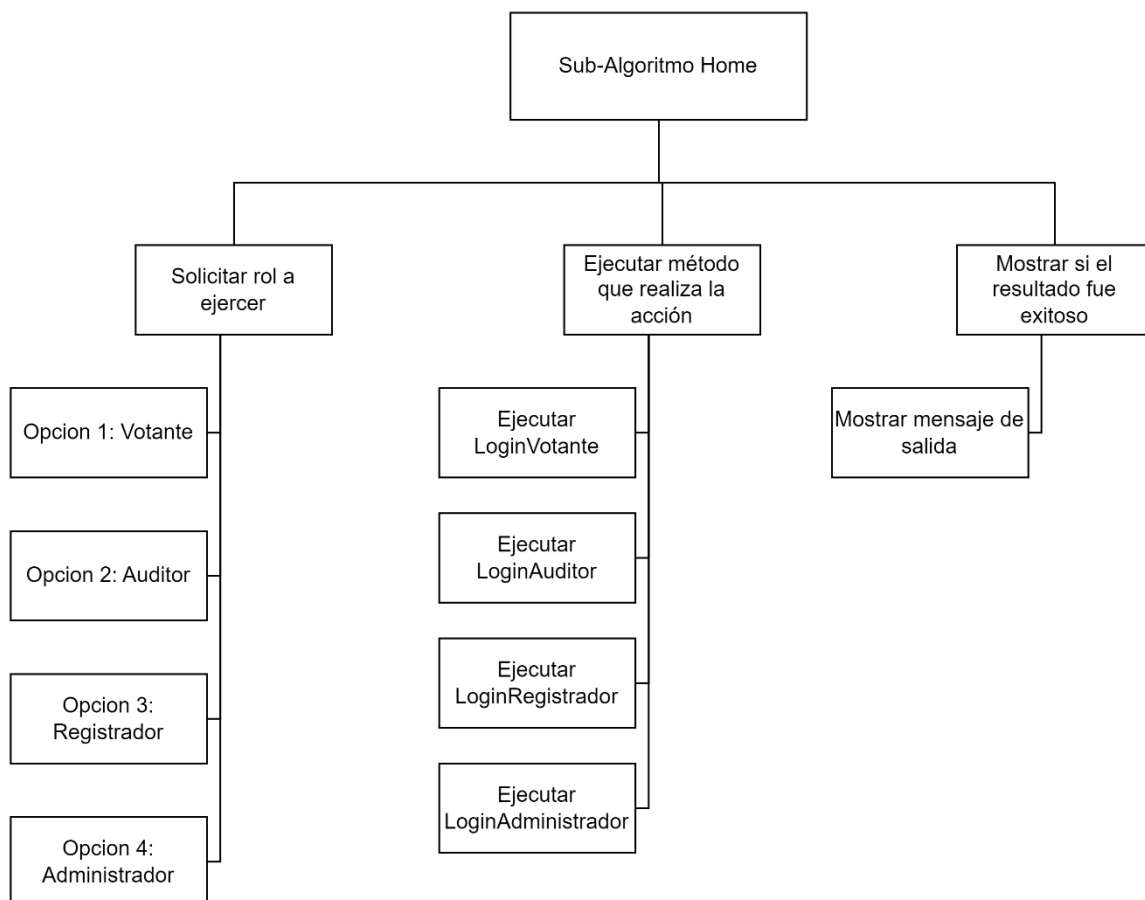
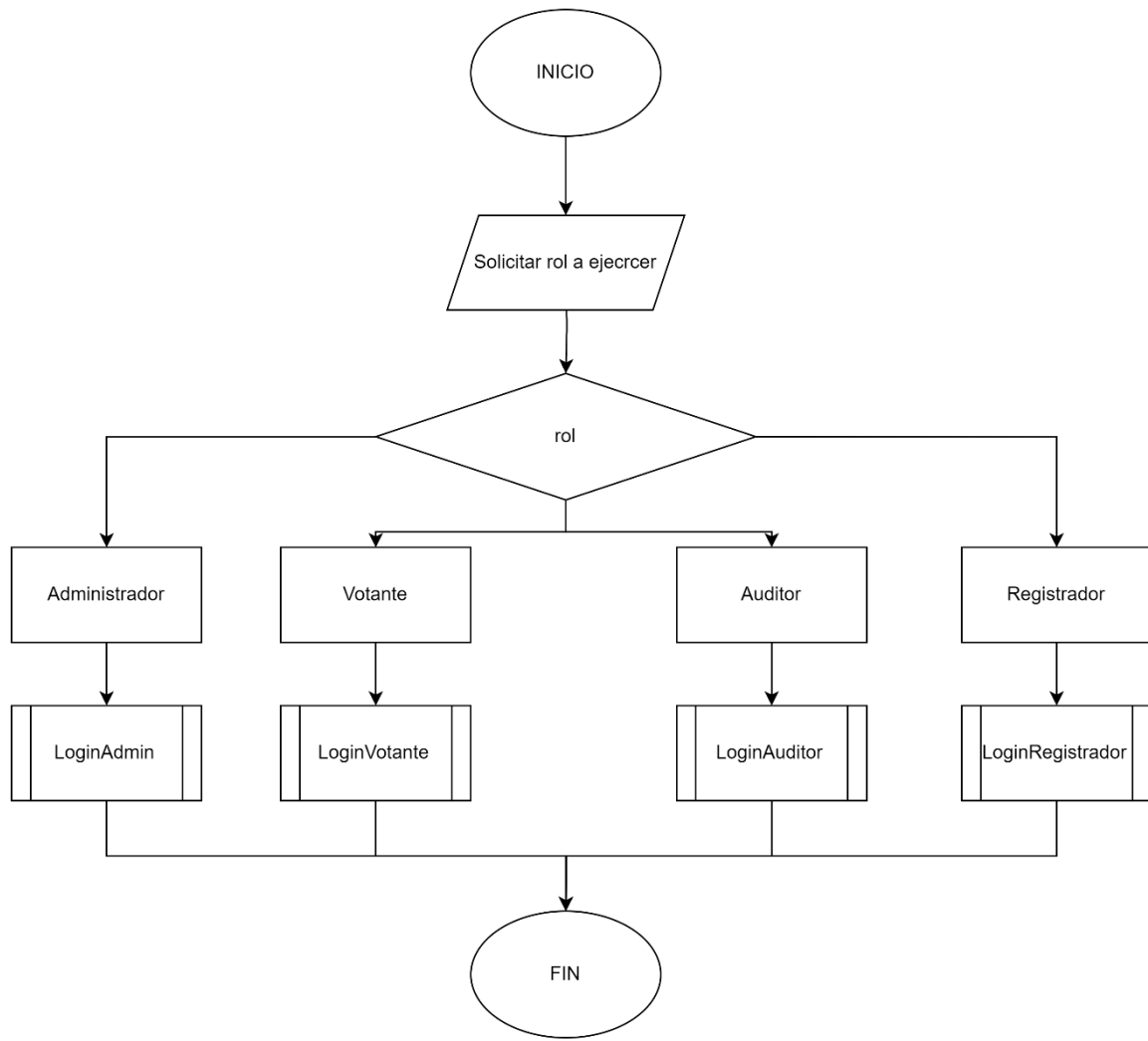


Diagrama de Flujo



Sub-Algoritmo Administrador

Análisis del problema

4. ¿Qué entradas se requieren?
 - a. Contraseña
 - b. Opción de la acción a realizar
5. ¿Cuál es la salida deseada?
 - a. Iniciar sesión
 - b. Realizar la acción deseada
6. ¿Qué método produce la salida deseada?
 - a. Solicitar la contraseña del administrador y si es correcta solicitar la acción a realizar y ejecutar el método destinado
6. Requerimientos adicionales y/o restricciones
 - a. Contraseña

Diseño descendente

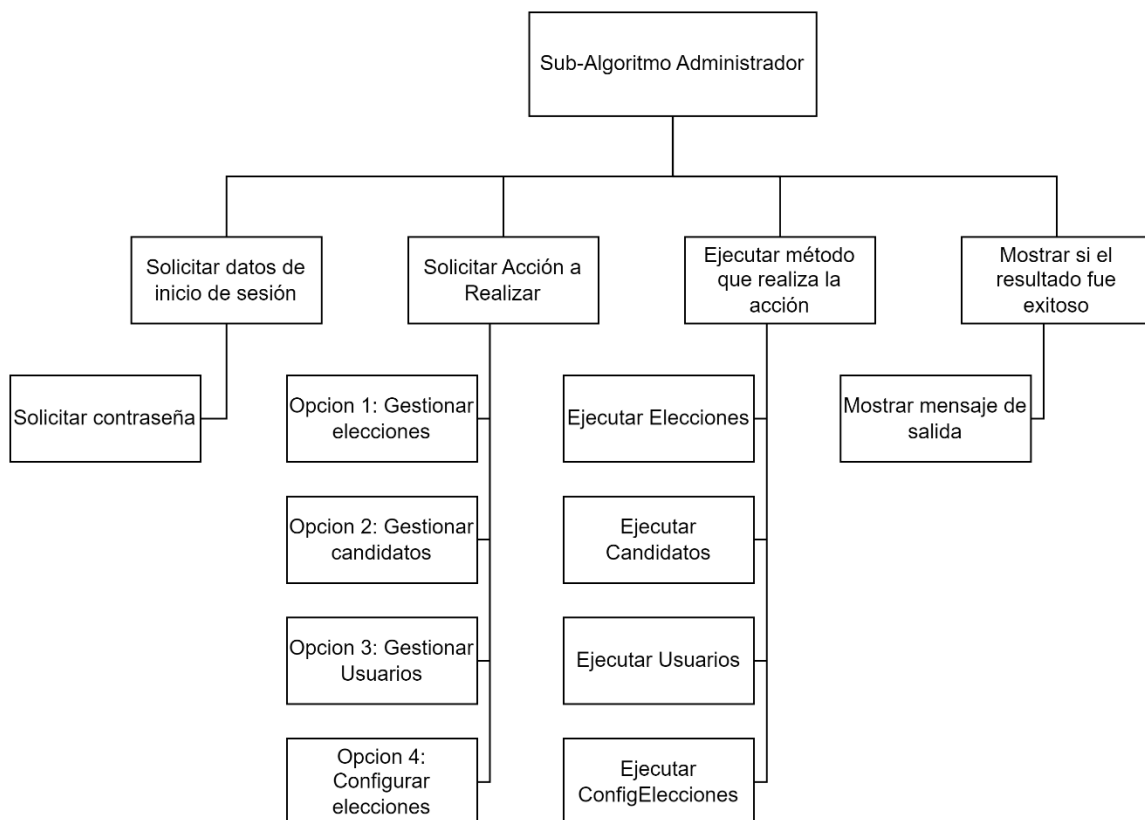
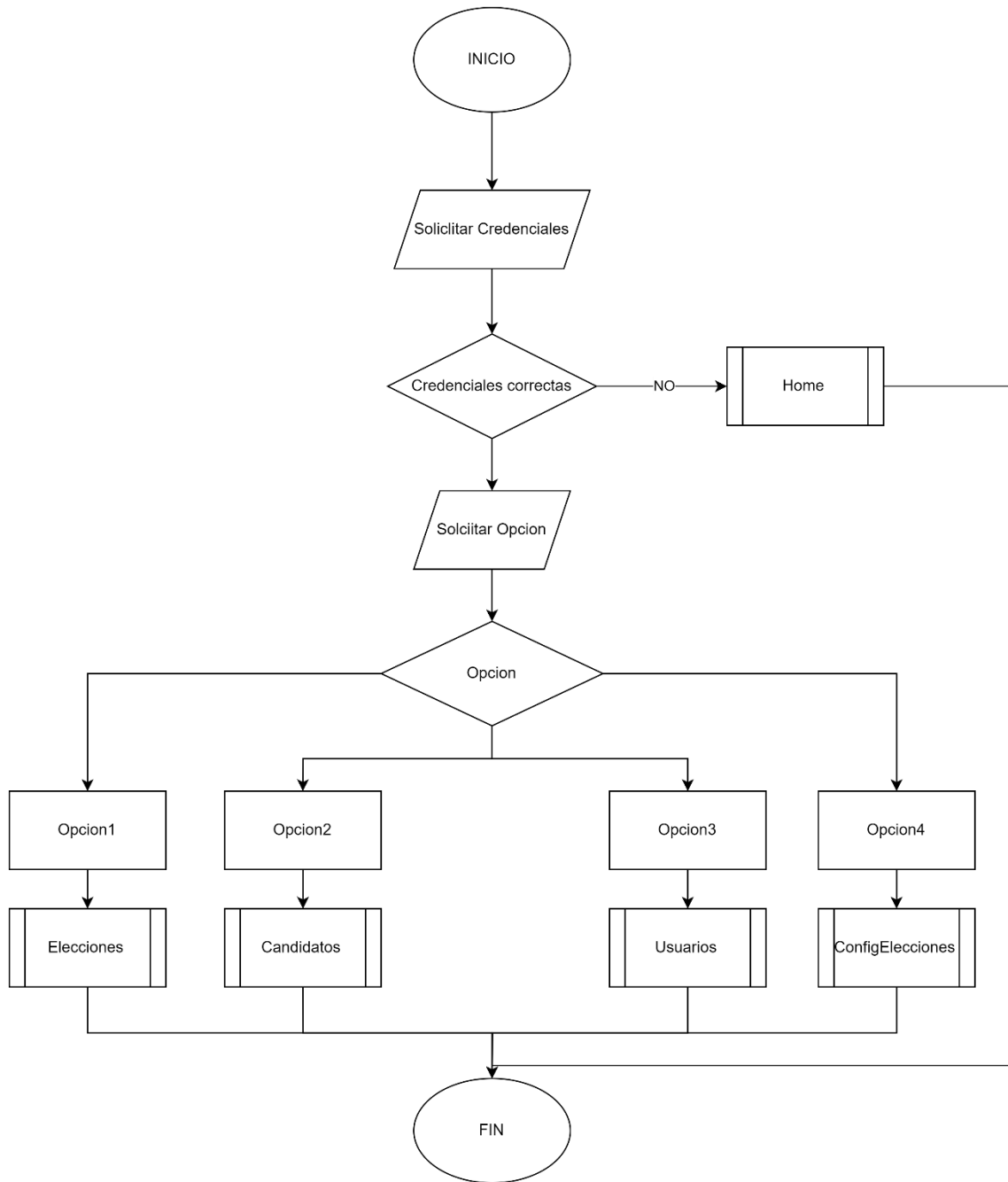


Diagrama de Flujo



Sub-Algoritmo Votante

Análisis del problema

7. ¿Qué entradas se requieren?
 - a. Datos de inicio de sesión
 - b. Código de la elección
 - c. Código del candidato a votar
8. ¿Cuál es la salida deseada?
 - a. Emitir un voto a una elección o varias
9. ¿Qué método produce la salida deseada?
 - a. Solicitar credenciales de inicio de sesión y votar
7. Requerimientos adicionales y/o restricciones
 - a. Solo puede votar a las que haya solicitado votar

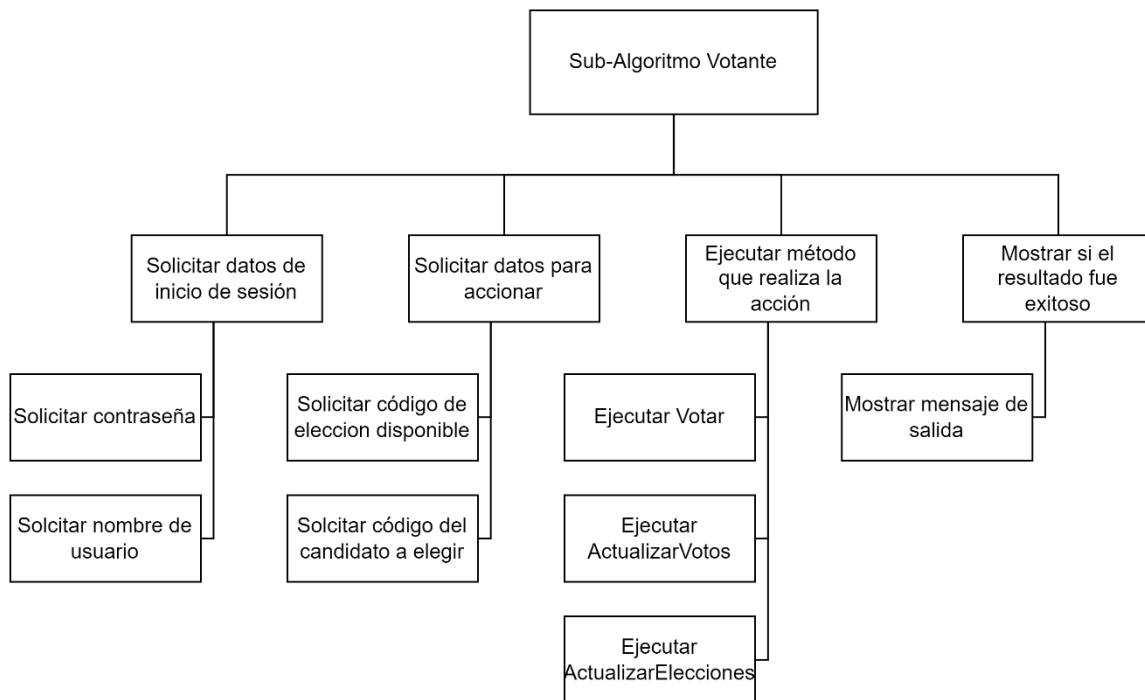
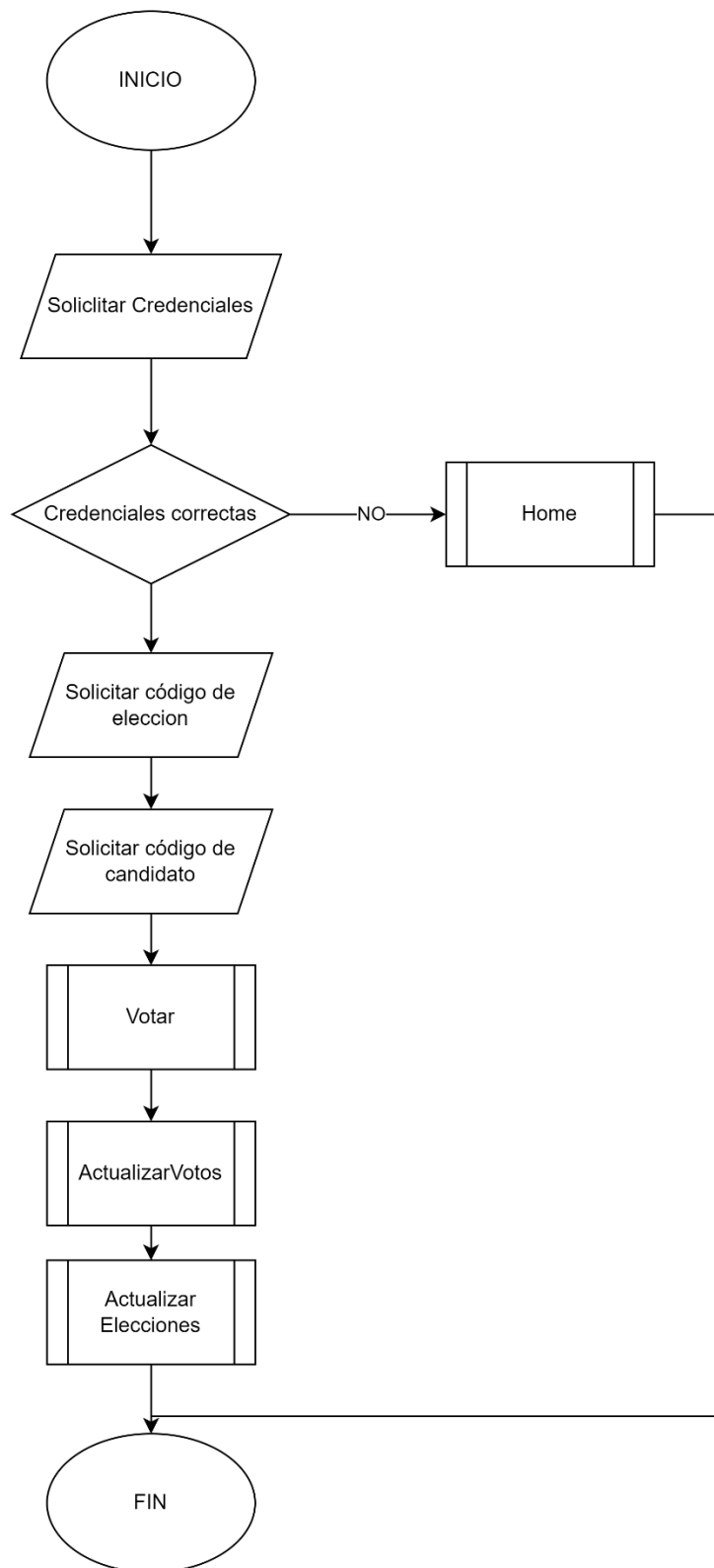


Diagrama de Flujo

Sub-Algoritmo Registrador

Análisis del problema

10. ¿Qué entradas se requieren?
 - a. Solicitar credenciales de inicio de sesión
 - b. Solicitar opción a realizar
11. ¿Cuál es la salida deseada?
 - a. Acción que desea solicitar el usuario
12. ¿Qué método produce la salida deseada?
 - a. Solicitar las credenciales y la opción a realizar
8. Requerimientos adicionales y/o restricciones
 - a. Un Registrador no puede tener otros roles

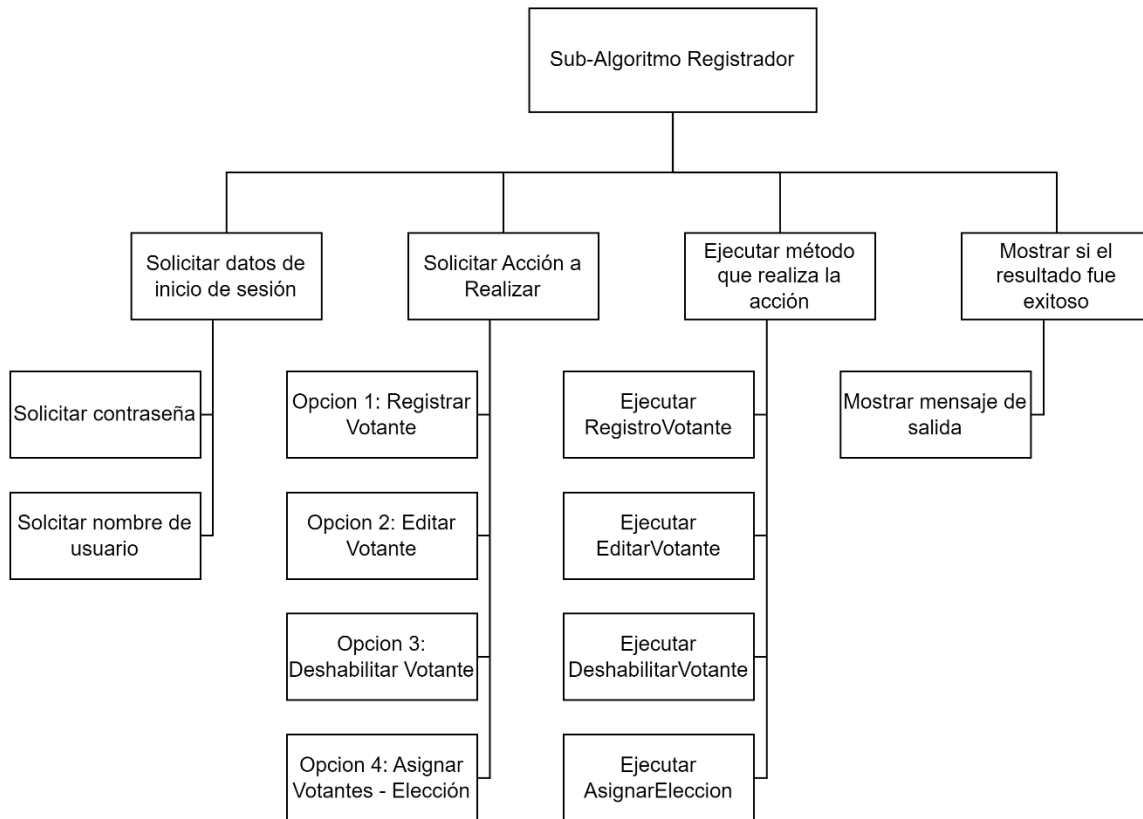
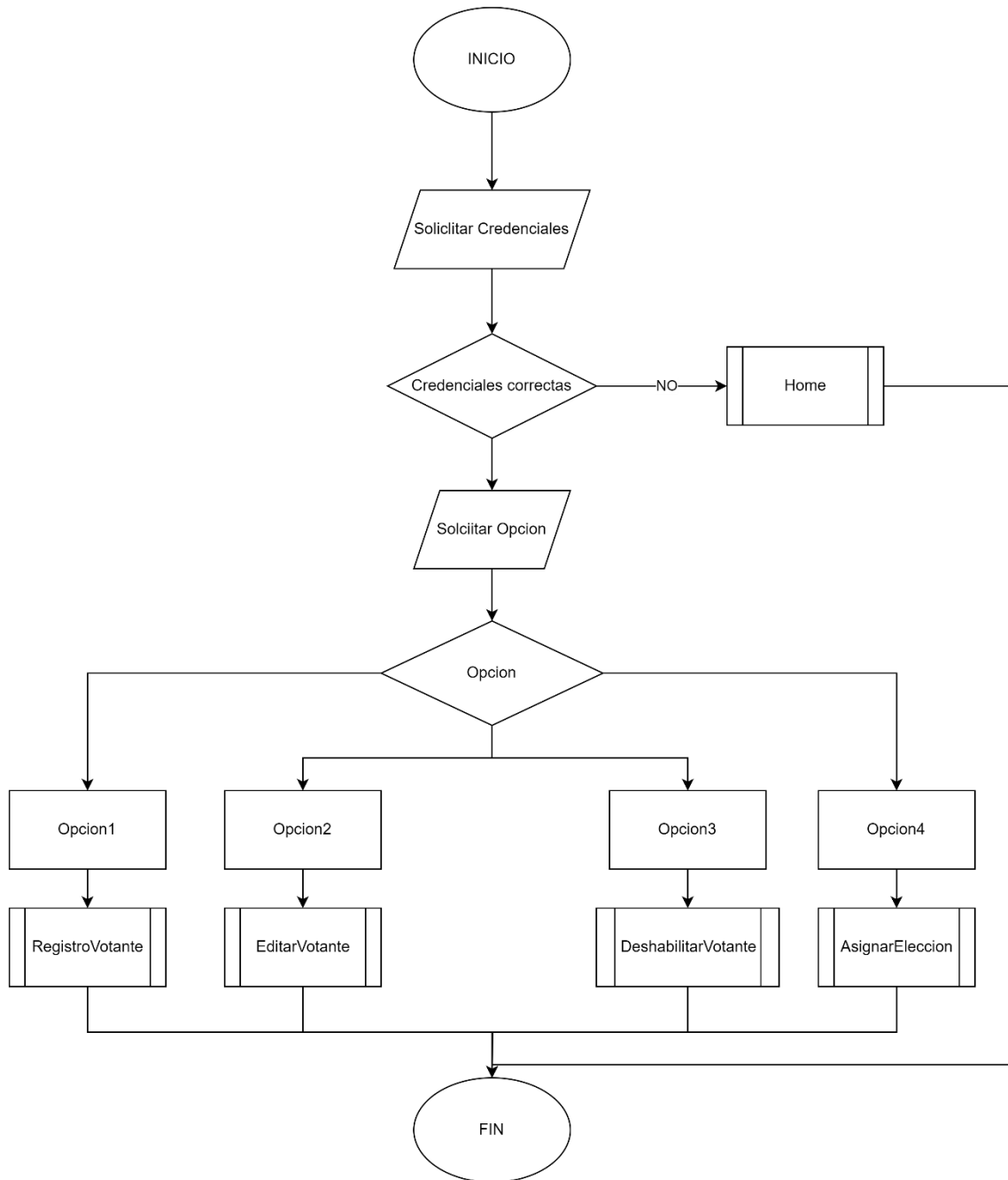


Diagrama de Flujo



Sub-Algoritmo Auditor

Análisis del problema

13. ¿Qué entradas se requieren?
 - a. Solicitar credenciales para poder ingresar
 - b. Solicitar elección para visualizar sus estadísticas
14. ¿Cuál es la salida deseada?
 - a. Visualizar estadísticas
15. ¿Qué método produce la salida deseada?
 - a. Solicitar credenciales y solicitar elección
9. Requerimientos adicionales y/o restricciones
 - a. Un Auditor no tiene acceso a los demás roles

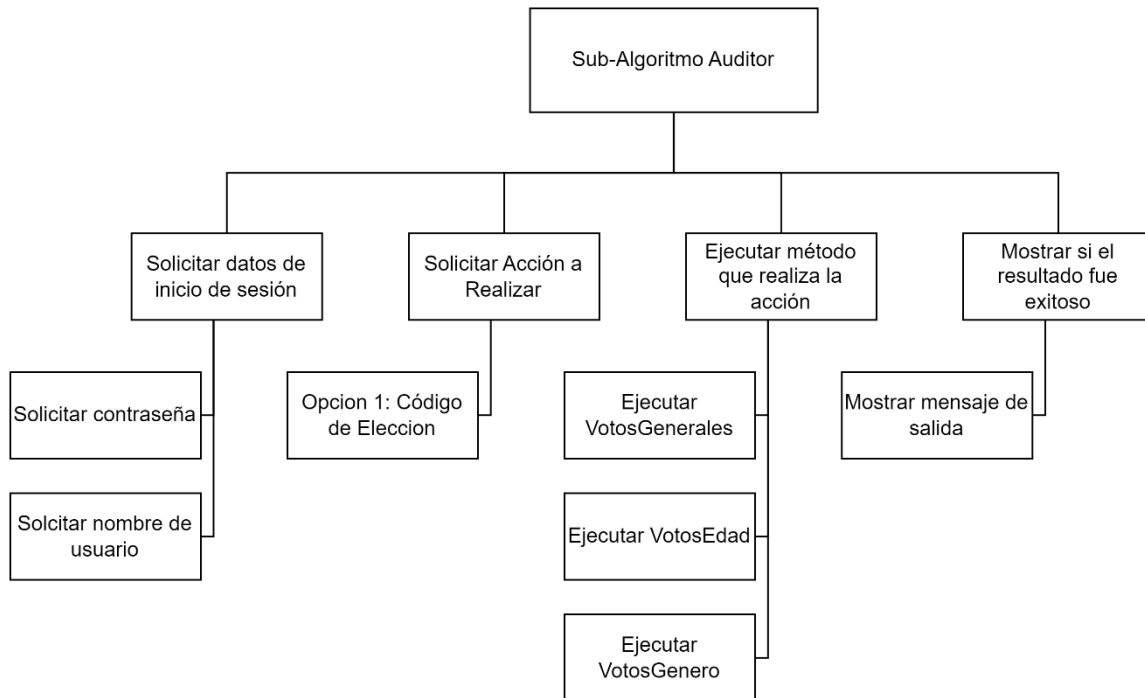
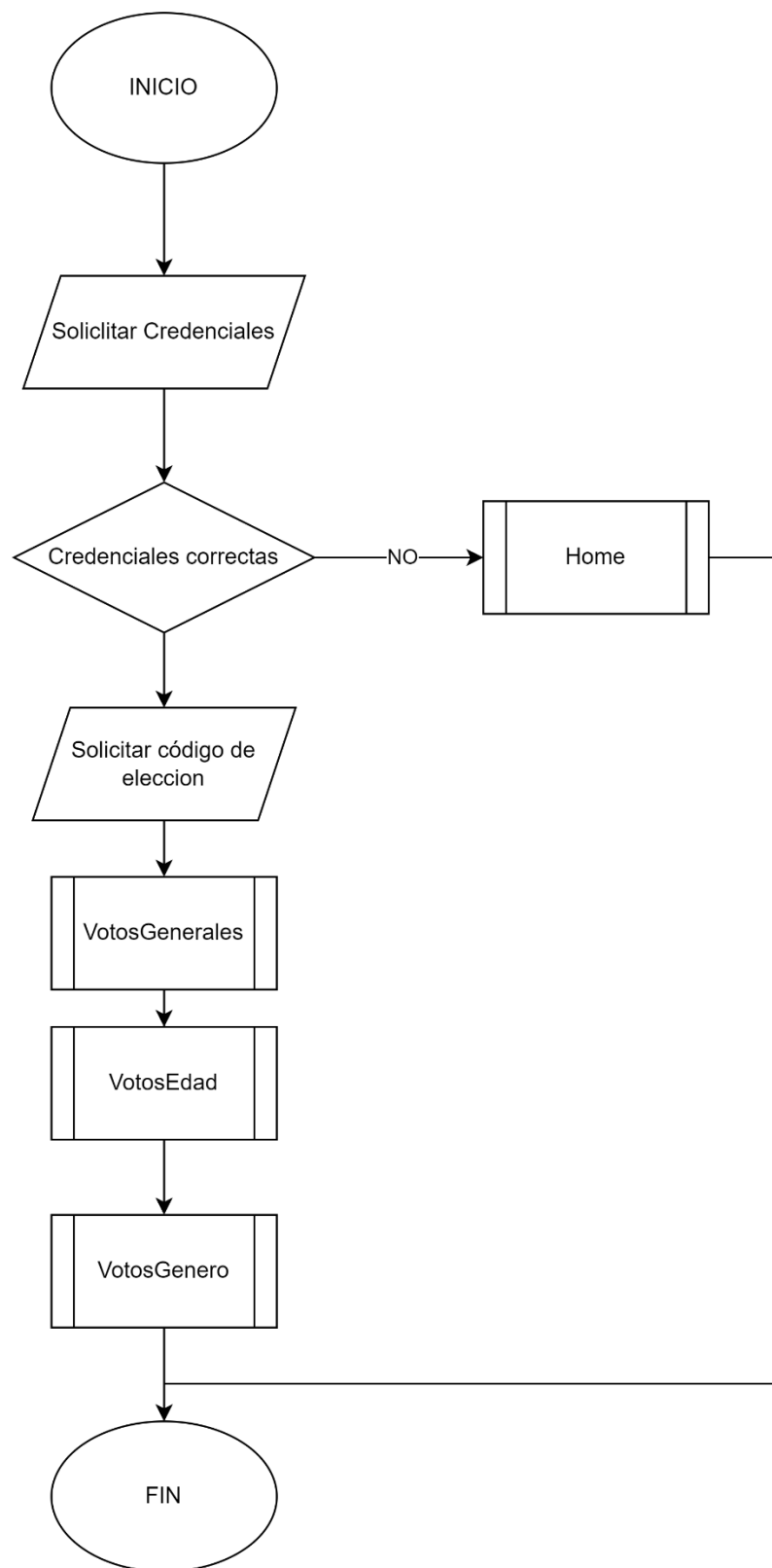


Diagrama de Flujo

Tecnologías Utilizadas

Para este proyecto se han utilizado distintas tecnologías para el funcionamiento más óptimo y eficaz posible, estas tecnologías se rigen bajo la licencia MIT o Public Domain, así mismo se dejan los créditos correspondientes en la sección de referencias.

- [Java](#)

Instrucciones de Configuración del Entorno de Trabajo

Requisitos Técnicos

- JDK Versión 17 o superior.
- IntelliJ IDEA (Recomendado, no indispensable).
- Windows 10 o superior.
- Última versión del código fuente alojado en el repositorio de GitHub.

Clonación del Proyecto

Para comenzar a modificar el código de nuestro proyecto deberás clonarlo de nuestro repositorio de GitHub de la siguiente manera:

```
git clone https://github.com/estuardodev/ProyectoAlgoritmosUMG.git
```

Una vez clonado el repositorio deberás de cargarlo en tu IDE que estes utilizando, como recomendación utiliza **IntelliJ IDEA**.

Si en dado caso ya tienes clonado el repositorio, pero lo que quieres es actualizar el código, ejecuta el siguiente comando:

```
git pull origin master
```

Media vez ya realizaste algún cambio y lo deseas mandar nuevamente al repositorio deberás de seguir estos pasos:

1. Construir el archivo .jar, primero deberás de construir el .jar con los cambios y este deberá de estar en la carpeta de JARS con la siguiente estructura “SistemaVotacionUMG_VersionXX.jar” y deberás de actualizar el archivo que se encuentra en la raíz del proyecto con el formato de “SistemaVotacionUMG.jar”
2. En el mensaje del commit deberás de especificar que archivo modificaste y si es cual es el número de IUSSE que realizas o si es una funcionalidad aparte.

Proceso de Compilación del Proyecto

Si deseas probar el proyecto una vez ya configurado en tu entorno de desarrollo integrado, simplemente puedes hacer un clic en el botón de ejecutar de tu entorno de desarrollo.

El software tiene la capacidad de crear los archivos que utilice para poder trabajar con total tranquilidad y no tengas que preocuparte de nada mas que probar tus nuevas funcionalidades.

Solución de Problemas Comunes

El software está diseñado para tratar de lidiar con la mayoría de los problemas que puedan lograr suceder, siempre y cuando ya exista un parche para cada problema. Pero puede llegar a suceder que te encuentres en alguna situación especial como algunas de las siguiente:

1. **No saber cómo regresar:** Si en dado caso estas en un menú donde no encuentres como regresar al menú anterior o dejar de ejecutar la acción que estas realizando puedes presionar el número 0 y luego enter, para que así el programa detecte que deseas salir o regresar y te regrese al menú más cercano.
2. **No saber cómo finalizar el programa:** Para finalizar el programa es tan fácil como presionar **CONTROL + C** al mismo tiempo y esto finalizara por completo el proceso.

NOTA: Si en dado caso te topas con algún problema como los anteriores te recomendamos iniciar una IUSSES en GitHub para tenerlo en cuenta y así poder trabajar en ella.

Información de Contacto

Este producto software tiene como fines educativos y por ningún motivo debería de ser utilizado en un entorno profesional o lega. NO ME HAGO RESPONSABLE POR EL USO QUE LE PUEDAN LLEGAR A DAR.

Si deseas contactar con mi persona tienes las siguientes opciones para poder hacerlos:

Portafolio: estuardo.dev

Correo Electrónico Personal: estuardo@estuardo.dev

Correo Electrónico de Negocios: contacto@estuardodev.com

Twitter/X: <https://twitter.com/estuardodev>

Glosario

1. **Sistema informático para votaciones:** Plataforma electrónica diseñada para facilitar y mejorar el proceso de votaciones en diversos contextos, como ámbitos políticos, empresariales, u otras formas de elecciones democráticas.
2. **Algoritmo de la Solución:** Conjunto de pasos ordenados y lógicos que describe cómo el sistema aborda y resuelve un problema específico.
3. **Roles:** Funciones o categorías asignadas a los usuarios del sistema, como administrador, registrador de votantes, votante y auditor, cada uno con sus propias responsabilidades y permisos.
4. **Autenticación segura:** Proceso mediante el cual el sistema verifica la identidad del usuario antes de permitir el acceso, asegurando que solo las personas autorizadas puedan interactuar con el sistema.
5. **Registro de Votantes:** Proceso de ingreso y gestión de información detallada de los votantes, incluyendo la generación de contraseñas seguras y la posibilidad de dar de baja votantes por motivos específicos, como fallecimiento.
6. **Administración de Elecciones:** Funcionalidad que permite a los administradores crear, configurar y gestionar elecciones, incluyendo la gestión de candidatos, fechas y horarios.
7. **Diagrama de Casos de Uso:** Representación gráfica que describe cómo los usuarios interactúan con el sistema, mostrando diferentes escenarios de uso.
8. **Diagrama de Clases:** Representación gráfica de la estructura y relaciones entre las clases (componentes) del sistema, utilizado en la programación orientada a objetos.
9. **Java:** Lenguaje de programación utilizado en el desarrollo del sistema, conocido por su portabilidad y amplio uso en aplicaciones empresariales.
10. **JDK (Java Development Kit):** Conjunto de herramientas necesarias para desarrollar aplicaciones Java, incluyendo el compilador y otras utilidades.
11. **IntelliJ IDEA:** Entorno de desarrollo integrado (IDE) utilizado para escribir, depurar y compilar código Java.
12. **Repositorio de GitHub:** Plataforma en línea para almacenar y gestionar versiones de código fuente, permitiendo la colaboración entre desarrolladores.
13. **Clonación del Proyecto:** Copia local del repositorio de GitHub en la máquina del desarrollador para facilitar la modificación y prueba del código.
14. **Compilación del Proyecto:** Proceso de traducción del código fuente a un formato ejecutable, listo para ser ejecutado en la máquina del usuario.
15. **IUSSE (Issue):** Problema o tarea específica registrada en GitHub para el seguimiento y resolución por parte del equipo de desarrollo.

Referencias

Freepik. (s.f.). *Imagen*. Obtenido de Flaticon: https://www.flaticon.es/icono-gratis/levantar-la-mano_2633926?term=democracia&page=1&position=3&origin=search&related_id=2633926

srip. (s.f.). *Imagen*. Obtenido de Flaticon: https://www.flaticon.es/icono-gratis/manual_3246731?term=manual&page=1&position=25&origin=search&related_id=3246731