





El futuro digital  
es de todos

MinTIC

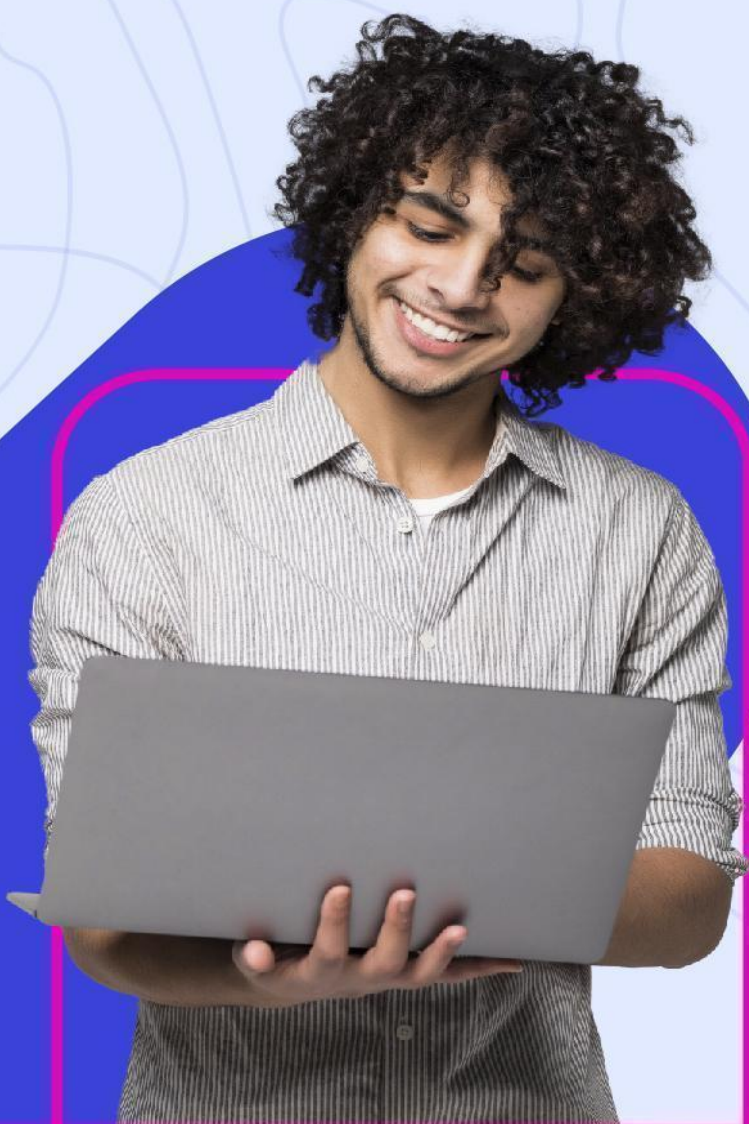


# Proyecto

## Ciclo 4a

### Programación Web

#### Universidad Nacional de Colombia



## Hechos

QUE

CONECTAN



## PROYECTO PROGRAMACION WEB

### VOTACIONES 2022

La **REGISTRADURÍA NACIONAL DE COLOMBIA** es una entidad con autonomía administrativa, contractual y presupuestal, organizada de manera desconcentrada, que tiene a su cargo el registro de la vida civil e identificación de los colombianos y la realización de los procesos electorales y los mecanismos de participación ciudadana, con plenas garantías para los colombianos.



Se ha encomendado a los beneficiarios del módulo de Programación Web de Misión TIC 2022 de la Universidad Nacional de Colombia, que lleven a cabo el **proceso de implementación del módulo de registro de resultados de las elecciones al senado bajo la modalidad de voto preferente**. Es de aclarar que el voto preferente se da cuando el partido deja que los ciudadanos voten no solo por el partido, sino también por cada uno de sus candidatos individualmente.

Para este caso, lograrán una curul o puesto quienes obtengan mayores votaciones sin importar el orden que ocupaban dentro de la lista.

Para ello, se han realizado reuniones para formalizar los requisitos, obteniendo la siguiente información:

1. La plataforma por seguridad debe de poseer un módulo de gestión de usuarios, de los cuales es necesario conocer su nombre de usuario (seudónimo), correo y contraseña.
2. Cada uno de los usuarios posee un rol dentro de la plataforma, de cada rol se debe conocer el nombre. Los roles definidos en inicio son: Administrador, Jurado de votación y ciudadano. Mas adelante se describe las funcionalidades a las cuales tiene acceso
3. Con el fin de controlar el acceso a cada una de las funcionalidades de la plataforma, se ha visto la necesidad de implementar un módulo de administración de permisos, para determinar que roles tienen acceso a cada una de las

funcionalidades de sistema (micro servicios). De los permisos se debe almacenar el identificador, la url y el método (GET, POST, PATCH, DELETE).

4. La administración de las mesas de votación, son de vital importancia ya que son la unidad básica en el proceso electoral, de estas se debe almacenar el número de mesa (que corresponde al identificador único) y número de cédulas inscritas.
5. En la plataforma se debe permitir la creación de los partidos políticos de los cuales se debe de almacenar, el nombre y el lema bajo el cual se fundamenta.
6. Cada uno de los partidos políticos, se compone de candidatos de los cuales se debe almacenar su número de resolución que lo acredita como candidato, cédula, nombre y apellido.
7. La razón de ser de la plataforma es la administración de los resultados, los cuales son emitidos por cada una de las mesas, es de aclarar que en cada mesa se registran los votos a los diferentes candidatos y que cada candidato puede ser votado en diferentes mesas.
8. Por último, es muy importante la visualización de reportes de las elecciones, tales como;
  - a. Listado de los votos obtenidos por todos los candidatos con el nombre del partido político al que pertenecen ordenados de mayor a menor de forma general por todas más mesas y también por una mesa en específica.
  - b. Listado de mesas con mayor participación ciudadana (suma de todos los votos de la mesa) ordenados de menor a mayor.
  - c. Listado de partidos políticos con cantidad de votos sufragados a su favor ordenados de mayor a menor de forma general y con opción de filtrar por mesa.
  - d. Distribución porcentual por partido político del nuevo congreso de la república. Tenga en cuenta que el congreso para esta edición será compuesto por los primeros 15 senadores mas votados en todas las mesas.

Teniendo en cuenta lo anterior se han planteado el siguiente diagrama de casos de uso. Ver imagen 1.

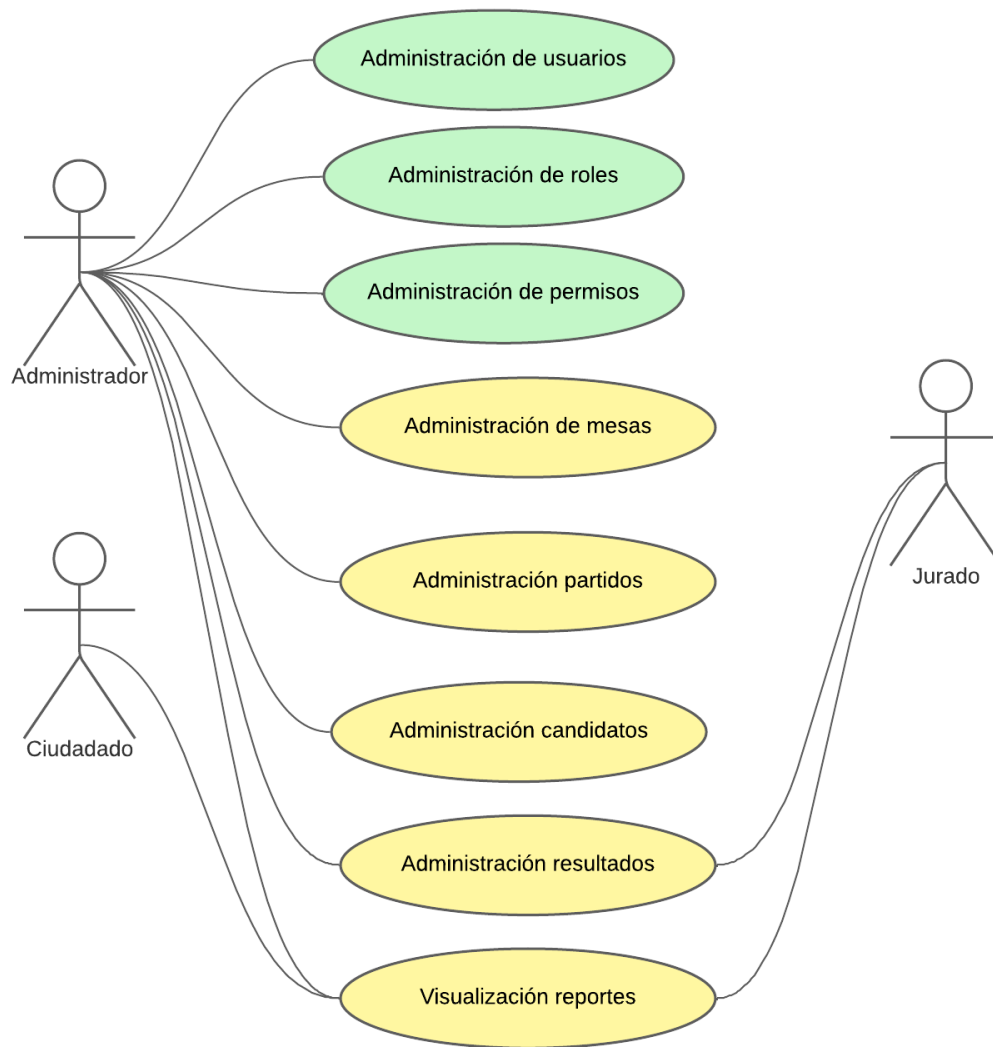


Imagen 1. Diagrama de casos de uso de la plataforma

**NOTA:** Todos los casos de uso que se refieren a tareas relacionadas con **“Administración”** implica llevar a cabo procesos de Creación, Listado, Visualización, Actualización y Eliminación (CRUD). Por ejemplo, el caso de uso “Administración de candidatos” en la implementación de código debe, permitir crear un candidato, visualizarlo, modificar y eliminarlo teniendo en cuenta un atributo como por ejemplo su cédula y por último la funcionalidad de listar todos los candidatos existentes en la plataforma.

Por recomendaciones generales de expertos en plataformas de gestión de votaciones a nivel mundial, se ha recomendado que el sistema tenga una arquitectura orientada a micro servicios y se ha planteado el esquema de la imagen 2.

Como se puede evidenciar el sistema se divide en 2 partes inicialmente, cliente – servidor. En el cliente (frontend) se requiere llevar a cabo la implementación utilizando el framework “Vue JS” y se darán ampliarán los detalles más adelante. Por el lado del servidor (backend), se encuentra la arquitectura distribuida en 3 partes fundamentalmente, el primer lugar se encuentra el “api gateway” el cual es el encargado de recibir todas las peticiones realizadas por el cliente, y su función principal es llevar a cabo el proceso de redirección de las peticiones ya sea al backend de seguridad o al backend de los resultados, a su vez actúa como interceptor “middleware” para validar si determinado usuario con su respectivo rol, tiene acceso a la petición solicitada. Por ejemplo, en caso que un ciudadano cualquiera quiera modificar un resultado de alguna mesa, el middleware debería denegarle la transacción ya que este no posee el permiso suficiente para llevar a cabo esta tarea.

En segundo lugar se encuentra el backend de seguridad, el cual por recomendaciones de seguridad se de implementar utilizando Java Spring Boot, en este se llevará a cabo toda la gestión de usuarios, roles y permisos. Esto quiere decir que los casos de uso 1,2 y 3 descritos anteriormente (los que aparecen en color verde en la imagen 1 “Diagrama de casos de uso de la plataforma”) se deben de implementar allí.

En tercer lugar se encuentra el backend de resultados, el cual se debe implementar Python Flask, allí se encontrará el desarrollo de los casos de uso 4,5,6,7 y 8, descritos anteriormente y que están en color amarillo en la imagen 1.

Es de aclarar que tanto el backend de seguridad como el de resultados tendrán acceso a la capa de persistencia. En esta parte se utilizará una base de datos no relacional usando Mongo DB. Se utilizará el formato JSON para las comunicaciones entre los diferentes módulos del sistema.

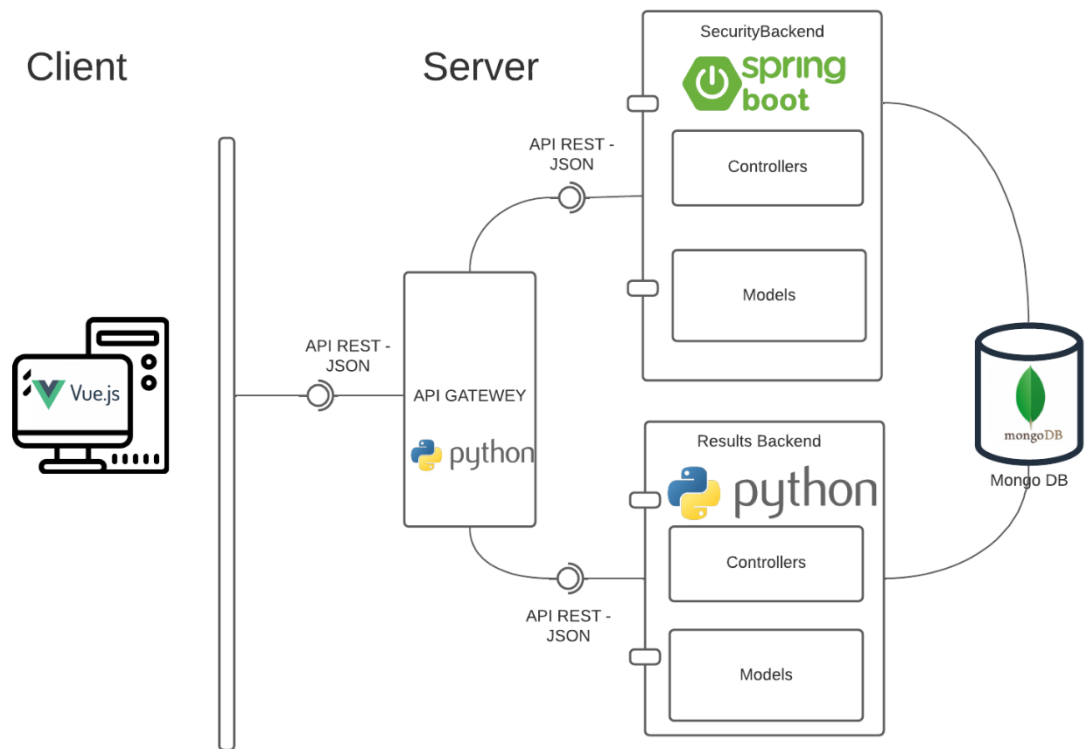


Imagen 2. Diagrama arquitectónico de la plataforma.

Para la implementación del sitio web (Cliente o Frontend) se deben de tener en cuenta las siguientes consideraciones:

- Se debe diseñar un portal web, una página inicial, la cual describe la información general de la registraduría, con secciones tales como “quienes somos”, “misión”, “visión”,
- Por seguridad**, en el sistema de información se debe realizar un formulario de identificación de usuarios, donde a través de las credenciales de acceso “login” (usuario y contraseña, el usuario será el email de la persona) tendrán acceso al sistema. La contraseña deberá ser almacenada de forma cifrada. Solo los usuarios identificados podrán acceder a las funcionalidades del sistema.
- Roles:** Debido a que existen diferentes grupos de personas que interactúan con el sistema, es necesario realizar la administración usuarios, roles y permisos. Los roles que se han planteado inicialmente en la plataforma son
  - Administrador de la plataforma: Tiene acceso a todos los módulos.
  - Jurado: Tiene acceso a módulo de gestión de resultados y reportes.
  - Ciudadano: Solo tiene acceso al módulo de reportes de los resultados.
- Según se el rol correspondiente, se debe poder visualizar cada uno de las vistas para la manipulación de todos los casos de uso descritos anteriormente,

**Requisitos técnicos:**

- Vistas de alto rendimiento considerando el funcionamiento asíncrono mediante promesas.
- Usar un framework de CSS para el desarrollo de la GUI. La cual debe considerar principios de usabilidad, tener apariencia agradable y ser responsiva. •
- Implementar el sistema mediante tecnologías de desarrollo web Backend (utilizando Python-Flask y Java-Spring Boot) y Frontend (utilizando Angular S)
- Se pueden implementar otros aspectos que supongan profundizar en alguna de las tecnologías vistas durante el módulo. Dichos aspectos deberán ser autorizados por el profesor.
- Comentar el código de manera formal.
- Se debe entregar un documento con el manual de usuario del sistema y un documento con el análisis y la definición de la base de datos.
- El control de versiones debe ser administrado con Git, y el control de actividades debe ser llevado a cabo con una herramienta en línea que lo permita como Trello. En ambos casos deben notificar y agregar al profesor.



## Entregas

**Sprint 0 (Análisis y diseño): Inicial** En este sprint, es para dar claridad al proyecto, al final se entrega la Base de Datos por parte del profesor. todos los miembros del equipo actuarán como analistas de requerimientos y arquitectos de software.

1. Documento de análisis y diseño de base de datos. Recuerde que son 2 esquemas, el primero corresponde a las tablas que serán gestionadas por el backend de seguridad, y el segundo a las tablas que serán gestionadas por el backend de resultados.
2. Diseño de cada uno de los microservicios que ofrecerá la plataforma, el cual sigue los lineamientos de la siguiente tabla:

Ejemplo:

Método	GET	
URL	https://url-server/hello/123	
Parámetros		
Campo	Tipo	Descripción
cedula	string	Cédula de la persona que se quiere saludar
Respuesta		
Campo	Tipo	Descripción
mensaje	string	Mensaje en el cual se saluda con nombre y apellido al usuario correspondiente a la cédula enviada
Ejemplo respuesta		
<pre>HTTP/1.1 200 OK {   "mensaje": "Hola usuario Juan Sanchez con CC 123", }</pre>		

3. Diseño de los mockups de cada una de las vistas que tendrá la plataforma.
4. Cronograma estimado de actividades
5. Implementación del tablero scrum con el detalle de cada una de las tareas a realizar en el producto backlog.

**Sprint 1 (Desarrollo y Pruebas - Backend Resultados Parte 1):** En este sprint todos los miembros del equipo actuarán como desarrolladores y testers.

1. Implementación de código de todos los end points correspondientes a los siguientes casos de uso.
  - a. Administración de mesas
  - b. Administración de partidos
  - c. Administración de candidatos
  - d. Administración de resultados

**Nota:** Se debe anexar el enlace del github correspondiente

2. Pruebas a todos los end points implementados en el punto anterior. En este punto se recomienda pegar capturas de imagen tanto de la petición como de la respuesta obtenida, se recomienda utilizar postman.
3. Captura de pantalla del tablero actualizado de trello en el cual se evidencie las tareas que llevó a cabo cada participante del grupo.

**Sprint 2 (Desarrollo y Pruebas - Backend Resultados Parte 2 y Backend Seguridad ):** En este sprint todos los miembros del equipo actuarán como desarrolladores y testers.

1. Implementación de código de todos los end points correspondientes a los siguientes casos de uso:
  - a. Visualización de reportes
  - b. Administración de usuarios
  - c. Administración de roles
  - d. Administración de servicios

**Nota:** Se debe anexar el enlace del github correspondiente

2. Pruebas a todos los end points implementados en el punto anterior. En este punto se recomienda pegar capturas de imagen tanto de la petición como de la respuesta obtenida, se recomienda utilizar postman.
3. Captura de pantalla del tablero actualizado de trello en el cual se evidencie las tareas que llevó a cabo cada participante del grupo.

**Sprint 3 (Desarrollo y Pruebas - Api Gateway):** En este sprint todos los miembros del equipo actuarán como desarrolladores y testers.

1. Implementación de código del api Gateway. Envío del enlace del github correspondiente
2. Implementación de los middlewares. Envío del enlace del github correspondiente
3. Pruebas al middleware en donde se evidencie como los usuarios con determinados roles asignados previamente pueden acceder a los servicios propios de su rol.
4. Pruebas al middleware en donde se evidencie como los usuarios que no tienen permisos se les prohíba continuar con las transacciones que no son propias de su rol
5. Se solicita que para el punto 3 y 4 de esta entrega se anexen las capturas de imagen tanto de la petición como de la respuesta obtenida, se recomienda utilizar postman.
6. Captura de pantalla del tablero actualizado de trello en el cual se evidencie las tareas que llevó a cabo cada participante del grupo.

**Sprint 4 (Desarrollo – Frontend ):** En este sprint todos los miembros del equipo actuarán como desarrolladores.

1. Implementación de código correspondiente a los CRUD en la vista de los siguientes casos de uso:
  - a. Home
  - b. Login
  - c. Administración de mesas.
  - d. Administración de candidatos
  - e. Administración de resultados
  - f. Visualización de reportes
2. Video donde se evidencie el funcionamiento de todos los puntos realizados en el literal anterior. Nota: Todos los integrantes del grupo deben exponer los componentes realizados.
3. Captura de pantalla del tablero actualizado de trello en el cual se evidencie las tareas que llevó a cabo cada participante del grupo.