

Entrega 1: Hasta la clase de 08/05/2025

- Identificar los recursos y diseñar las APIs con estilo REST para responder a las funcionalidades que se necesitan.
- Indicar endpoint (URL) y método a utilizar para cada acción.
- De cada endpoint que parámetros debe recibir por la URL, cuales por el cuerpo y que debe responder en el cuerpo de la respuesta y posibles códigos de estados http.

Recursos identificados en proyecto de votación electrónica:

- ★ Candidato.
- ★ Elección.
- ★ Lista.
- ★ Persona.
- ★ Resultado.
- ★ Voto.

API REST:

1. Recurso: Candidatos (/candidatos)

Endpoint: GET /candidatos

Descripción: Lista todos los candidatos.

Parámetros query opcionales: Se podría consultar por cualquier campo.

Cuerpo de la respuesta:

```
[  
  {  
    "id": 1,  
    "nombreCandidato": "string",  
    "puestoCandidato": "string",  
    "idLista": 1  
  }  
]
```

Códigos de respuesta:

- 200 OK: Consulta exitosa, se devuelven los candidatos.
- 204 No Content: No se encontraron candidatos que coincidan con los parámetros de búsqueda.
- 400 Bad Request: Parámetros de consulta inválidos o mal formateados.
- 404 Not Found: No se encontraron candidatos con los parámetros dados.

Endpoint: POST /candidatos

Descripción: Crea un nuevo candidato.

Cuerpo de petición:

```
{  
  "nombreCandidato": "string",  
  "puestoCandidato": "string",  
  "idLista": 1  
}
```

Cuerpo de la respuesta:

```
{  
  "id": 1  
}
```

Códigos de respuesta:

- 201 Created: Candidato creado exitosamente.
 - 400 Bad Request: Faltan campos obligatorios o el formato no es válido.
 - 404 Not Found: No se encuentra la lista con ese ID.
-

Endpoint: GET /candidatos/{id}

Descripción: Obtiene un candidato específico por su ID.

Cuerpo de la respuesta:

```
{  
  "id": 1,  
  "nombreCandidato": "string",  
  "puestoCandidato": "string",  
  "idLista": 1  
}
```

Códigos de respuesta:

- 200 OK: Se encontró el candidato y se devuelve su información.
 - 404 Not Found: No existe ningún candidato con ese ID.
-

Endpoint: PUT /candidatos/{id}

Descripción: Actualiza un candidato existente por su ID.

Cuerpo de petición:

```
{  
  "nombreCandidato": "string",  
  "puestoCandidato": "string",  
  "idLista": 1  
}
```

Cuerpo de respuesta:

```
{  
  "id": 1,  
  "nombreCandidato": "string",  
  "puestoCandidato": "string",  
  "idLista": 1  
}
```

Códigos de respuesta:

- 200 OK : Candidato actualizado exitosamente.
 - 400 Bad Request: Datos inválidos o faltantes.
 - 404 Not Found: No se encontró un candidato con el ID especificado.
-

Endpoint: DELETE /candidatos/{id}

Descripción: Elimina un candidato por su ID.

Códigos de respuesta:

- 204 No Content: Candidato eliminado exitosamente. No se devuelve contenido en la respuesta.
- 404 Not Found: No existe ningún candidato con el ID especificado.

2. Recurso: Personas (/personas)

Endpoint: GET /personas

Descripción: Lista todas las personas.

Parámetros query opcionales: Se podría consultar por cualquier campo.

Cuerpo de la respuesta:

```
[
  {
    "id": 1,
    "nombrePersona": "string",
    "nroIdentificacionPersona": "string",
    "apellidoPersona": "string",
    "contraseniaPersona": "string",
    "tipoDocumentoPersona": "string"
  }
]
```

Códigos de respuesta:

- 200 OK: Consulta exitosa, se devuelven las personas.
- 204 No Content: No se encontraron personas que coincidan con los parámetros de búsqueda.
- 400 Bad Request: Parámetros de consulta inválidos o mal formateados.

Endpoint: POST /personas

Descripción: Crea una nueva persona.

Cuerpo de petición:

```
{
  "nombrePersona": "string",
  "nroIdentificacionPersona": "string",
  "apellidoPersona": "string",
  "contraseniaPersona": "string",
  "tipoDocumentoPersona": "string"
}
```

Cuerpo de la respuesta:

```
{
  "id": 1
}
```

Códigos de respuesta:

- 201 Created: Persona creada exitosamente.
 - 400 Bad Request: Faltan campos obligatorios o el formato no es válido.
 - 409 Conflict: Ya existe una persona con ese número de identificación.
-

Endpoint: GET /personas/{id}

Descripción: Obtiene una persona específica por su ID.

Cuerpo de la respuesta:

```
{  
  "id": 1,  
  "nombrePersona": "string",  
  "nroIdentificacionPersona": "string",  
  "apellidoPersona": "string",  
  "contraseniaPersona": "string",  
  "tipoDocumentoPersona": "string"  
}
```

Códigos de respuesta:

- 200 OK: Se encontró la persona y se devuelve su información.
 - 404 Not Found: No existe ninguna persona con ese ID.
-

Endpoint: PUT /personas/{id}

Descripción: Actualiza una persona existente por su ID.

Cuerpo de petición:

```
{  
  "nombrePersona": "string",  
  "nroIdentificacionPersona": "string",  
  "apellidoPersona": "string",  
  "contraseniaPersona": "string",  
  "tipoDocumentoPersona": "string"  
}
```

Cuerpo de respuesta:

```
{  
  "id": 1,  
  "nombrePersona": "string",  
  "nroIdentificacionPersona": "string",  
  "apellidoPersona": "string",  
  "contraseniaPersona": "string",  
  "tipoDocumentoPersona": "string"  
}
```

Códigos de respuesta:

- 200 OK : Persona actualizada exitosamente.
 - 400 Bad Request: Datos inválidos o faltantes.
 - 404 Not Found: No se encontró una persona con el ID especificado.
-

Endpoint: DELETE /personas/{id}

Descripción: Elimina una persona por su ID.

Códigos de respuesta:

- 204 No Content: Persona eliminada exitosamente. No se devuelve contenido en la respuesta.
- 404 Not Found: No existe ninguna persona con el ID especificado.

3. Recurso: Elecciones (/elecciones)

Endpoint: GET /elecciones

Descripción: Lista todas las elecciones.

Parámetros query opcionales: Se podría consultar por cualquier campo.

Cuerpo de la respuesta:

```
[  
  {  
    "id": 1,  
    "nombreEleccion": "string",  
    "descripcionEleccion": "string",  
    "cantidadListas": 0,  
    "fechaInicioEleccion": "2025-05-01T08:00:00",  
    "fechaFinEleccion": "2025-05-01T08:00:00"  
  }  
]
```

Códigos de respuesta:

- 200 OK: Consulta exitosa, se devuelven las elecciones.
- 204 No Content: No se encontraron elecciones que coincidan con los parámetros de búsqueda.
- 400 Bad Request: Parámetros de consulta inválidos o mal formateados.

Endpoint: POST /elecciones

Descripción: Crea una nueva elección.

Cuerpo de petición:

```
{  
  "nombreEleccion": "string",  
  "descripcionEleccion": "string",  
  "cantidadListas": 0,  
  "fechaInicioEleccion": "2025-05-01T08:00:00",  
  "fechaFinEleccion": "2025-05-01T08:00:00"  
}
```

Cuerpo de la respuesta:

```
{  
  "id": 1  
}
```

Códigos de respuesta:

- 201 Created: Elección creada exitosamente.
 - 400 Bad Request: Faltan campos obligatorios o el formato no es válido.
 - 409 Conflict: Ya existe una elección con el mismo nombre y fechas.
-

Endpoint: GET /elecciones/{id}

Descripción: Obtiene una elección específica por su ID.

Cuerpo de la respuesta:

```
{
  "id": 1,
  "nombreEleccion": "string",
  "descripcionEleccion": "string",
  "cantidadListas": 0,
  "fechaInicioEleccion": "2025-05-01T08:00:00",
  "fechaFinEleccion": "2025-05-01T08:00:00"
}
```

Códigos de respuesta:

- 200 OK: Se encontró la elección y se devuelve su información.
 - 404 Not Found: No existe ninguna elección con ese ID.
-

Endpoint: PUT /elecciones/{id}

Descripción: Actualiza una elección existente por su ID.

Cuerpo de petición:

```
{
  "nombreEleccion": "string",
  "descripcionEleccion": "string",
  "cantidadListas": 0,
  "fechaInicioEleccion": "2025-05-01T08:00:00",
  "fechaFinEleccion": "2025-05-01T08:00:00"
}
```

Cuerpo de respuesta:

```
{
  "id": 1,
  "nombreEleccion": "string",
  "descripcionEleccion": "string",
  "cantidadListas": 0,
  "fechaInicioEleccion": "2025-05-01T08:00:00",
  "fechaFinEleccion": "2025-05-01T08:00:00"
}
```

Códigos de respuesta:

- 200 OK : Elección actualizada exitosamente.
 - 400 Bad Request: Datos inválidos o faltantes.
 - 404 Not Found: No se encontró una elección con el ID especificado.
-

Endpoint: DELETE /elecciones/{id}

Descripción: Elimina una elección por su ID.

Códigos de respuesta:

- 204 No Content: Elección eliminada exitosamente. No se devuelve contenido en la respuesta.
- 404 Not Found: No existe ninguna elección con el ID especificado.

4. Recurso: Listas (/listas)

Endpoint: POST /listas

Descripción: Crea una nueva lista.

Códigos de respuesta:

POST /listas

```
{
  "nombreLista": "string",
  "descripcionLista": "string",
  "CantidadIntegrantes": "int",
  "IdEleccion": "int",
}
```

Cuerpo de la respuesta:

```
{
  "id": 1
}
```

Códigos de respuesta:

- 201 Created: Lista creada exitosamente.
- 400 Bad Request: Faltan campos obligatorios o el formato no es válido.

Endpoint: GET /listas/{id}

Descripción: Obtiene una lista específica por su ID.

Cuerpo de la respuesta:

```
{
  "id": 1,
  "nombreLista": "string",
  "IdEleccion": "Int",
  "cantidadIntegrantes": int,
  "votos": [],
  "candidatos": []
}
```

Códigos de respuesta:

- 200 OK: Se encontró la lista y se devuelve su información.
- 404 Not Found: No existe ninguna lista con ese ID.

Endpoint: PUT /listas/{id}

Descripción: Actualiza una lista existente por su ID.

Cuerpo de petición:

```
{
  "nombreLista": "string",
}
```

```
"descripcionLista": "string",  
"CantidadIntegrantes": "int",  
"IdEleccion": "int"  
}
```

Cuerpo de respuesta:

```
{  
  "id": 1,  
  "nombreLista": "string",  
  "IdEleccion": "Int",  
  "cantidadIntegrantes": int,  
  "votos": [],  
  "candidatos": []  
}
```

Códigos de respuesta:

- 200 OK : Lista actualizada exitosamente.
- 400 Bad Request: Datos inválidos o faltantes.
- 404 Not Found: No se encontró una lista con el ID especificado.

Endpoint: DELETE /listas/{id}

Descripción: Elimina una lista por su ID.

Códigos de respuesta:

- 204 No Content: Lista eliminada exitosamente. No se devuelve contenido en la respuesta.
- 404 Not Found: No existe ninguna lista con el ID especificado.

5. Recurso: Resultados (/resultados)

Endpoint: POST /resultados

Descripción: Crea un nuevo resultado.

Cuerpo de petición:

```
{  
  "eleccionId": 1,  
  "listaId": 1  
}
```

Cuerpo de respuesta:

```
{  
  "id": 1  
}
```

Códigos de respuesta:

- 201 Created: Resultado registrado exitosamente.
- 400 Bad Request: Faltan campos obligatorios o el formato no es válido.
- 404 Not Found: No se encuentra la elección o la lista con el ID especificado

Endpoint: GET /resultados/{eleccionId}

Descripción: Obtiene los resultados de una elección específica por su ID.

Cuerpo de la respuesta:

```
{  
  "listId": 1,  
  "totalVotos": 0  
}
```

Códigos de respuesta:

- 200 OK: Consulta exitosa, se devuelven los resultados.
- 400 Bad Request: ID de elección inválido.
- 404 Not Found: No se encontraron resultados para la elección especificada.

Endpoint: GET /resultados

Descripción: Obtiene todos los resultados registrados.

Cuerpo de la respuesta:

```
[  
  {  
    "id": 1,  
    "eleccionId": 1,  
    "listId": 1,  
    "totalVotos": 100  
  },  
  {  
    "id": 2,  
    "eleccionId": 2,  
    "listId": 3,  
    "totalVotos": 150  
  }  
]
```

Códigos de respuesta:

- 200 OK: Consulta exitosa, se devuelven los resultados.
- 404 Not Found: No se encontraron resultados.

Endpoint: DELETE /resultados/{id}

Descripción: Elimina un resultado por su ID.

Códigos de respuesta:

- 204 No Content: Eliminación exitosa.
- 404 Not Found: No se encontró el resultado con el ID especificado.

6. Recurso: Votos (/votos)

Endpoint: POST /votos

Descripción: Registra un nuevo voto.

Cuerpo de petición:

```
{
```

```
"eleccionId": 1,  
"listId": 1  
}  
Cuerpo de la respuesta:  
{  
  "mensaje": "Voto registrado exitosamente."  
}
```

Códigos de respuesta:

- 200 OK: Voto registrado exitosamente.
- 400 Bad Request: La persona no está autorizada o ya ha votado en esta elección.
- 404 Not Found: No se encuentra la elección o la lista con el ID especificado.

Endpoint: DELETE /votos/{id}

Descripción: Elimina un voto por su ID.

Códigos de respuesta:

- 204 No Content: Voto eliminado exitosamente. No se devuelve contenido en la respuesta.
- 404 Not Found: No existe ningún voto con el ID especificado.