

Informe de Módulo de Formación Presencial

BOGOTÁ, 2025

601 716 7812 - 321 754 3845
sales@3dstudioweb.com
Bogotá - Colombia
3dstudioweb.com

TABLA DE CONTENIDO

1.Introducción 3

2. Objetivo 3

3. Desarrollos implementados 3

1.Introducción

Este documento presenta un detalle de cada acción y desarrollo realizado para el módulo de formación presencial de acuerdo a las Historias de usuario entregadas por la SDDE.

2. Objetivo

Mostrar la trazabilidad de cada uno de los requerimientos entregados desde lo desarrollado desde el fronend y backend de la plataforma de Bogota trabaja y con esto garantizar la calidad y funcionalidad de la plataforma.

3. Desarrollos implementados

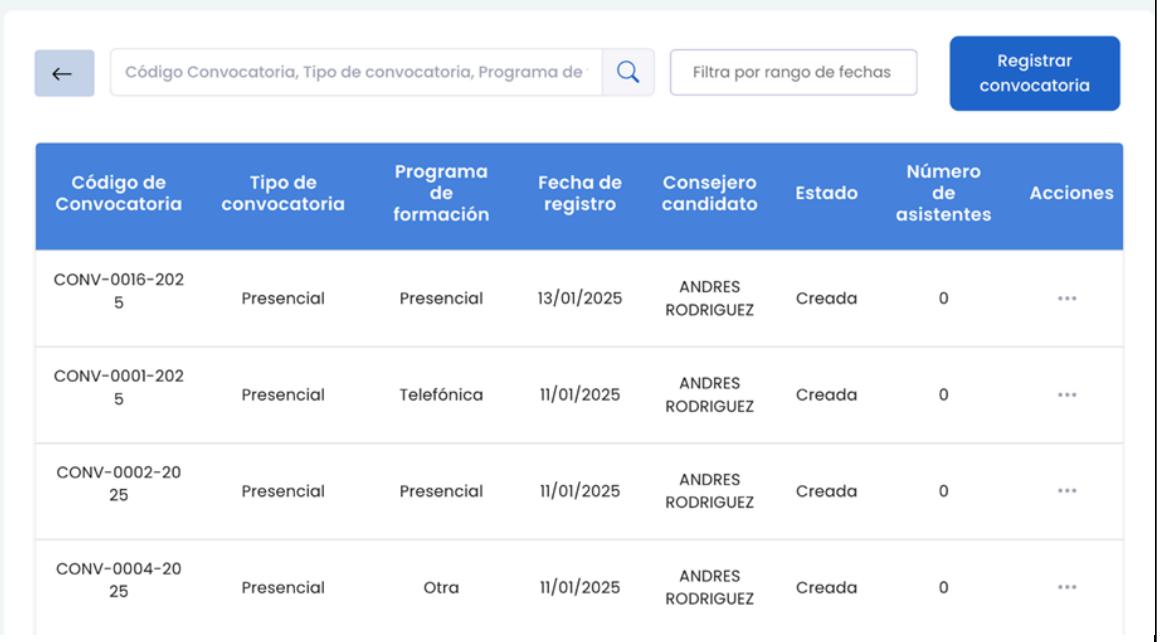
A continuación se muestra el detalle que el equipo de 3DStudio realizó para dar solución y luego liberar para que el equipo funcional de la SDDE realice pruebas y cierre del mismo.

Clave	Resumen	Estado	Sprint
SCRUM-66	1 [28] La creación de una herramienta que se llame convocatoria que permite la identificación el programa, fecha de registro entre otras variables	Finalizada	Tablero Sprint 10
Backend			

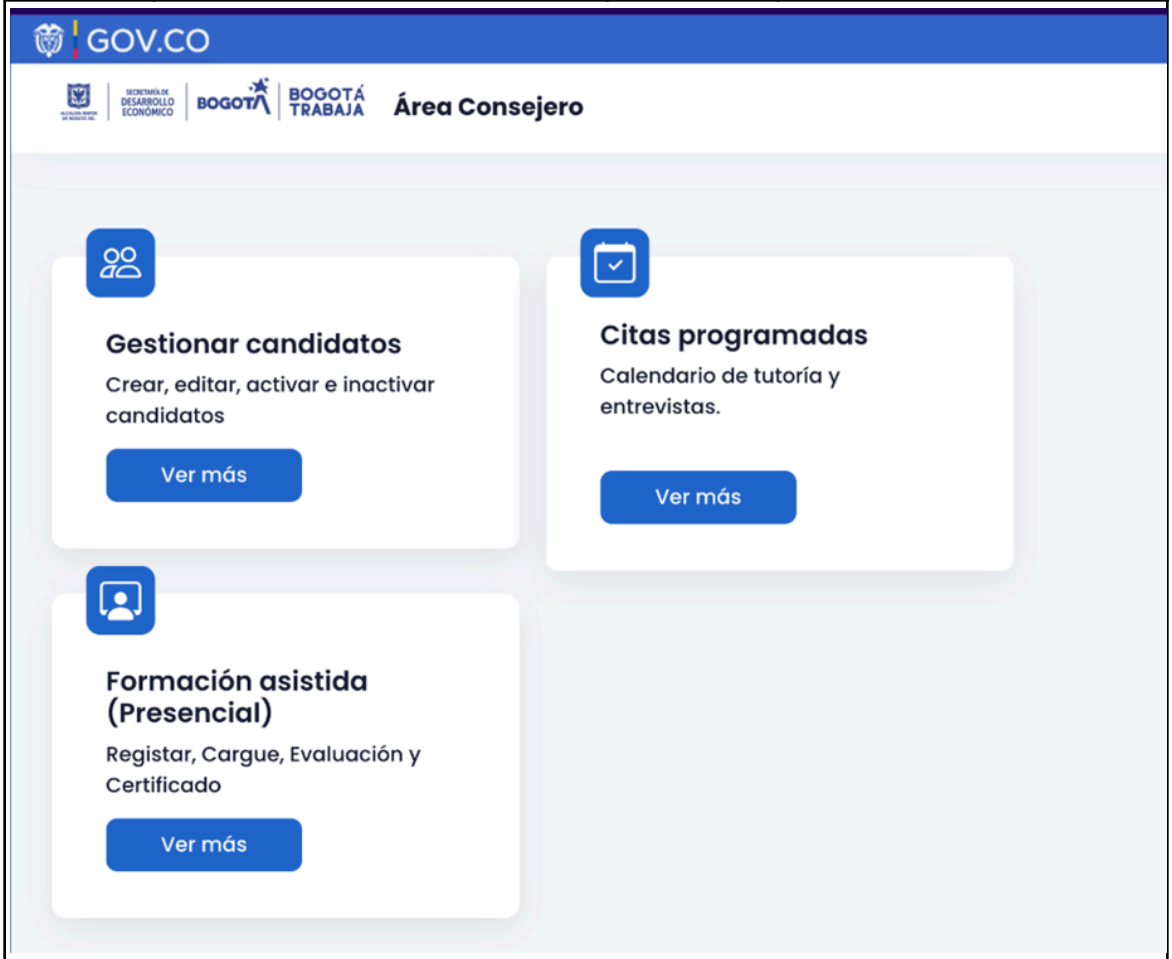
Clave	Resumen	Estado	Sprint

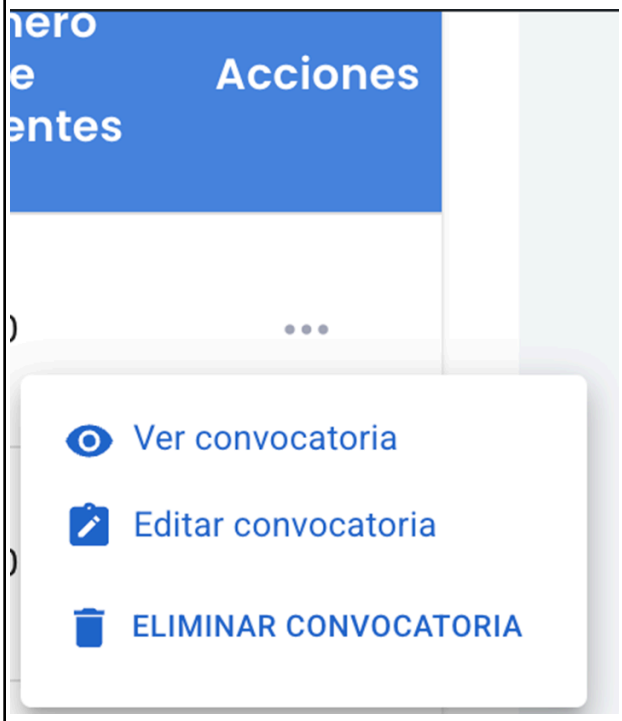
```
// Convocatoria
// Crear convocatoria
routes.post( path: `${contextPath()}/convocatoria`, scopes( requiredScopes: [Roles.COUNSELOR_CANDIDATE.name]), ConvocatoriaController.store);
// Actualizar convocatoria
routes.put( path: `${contextPath()}/convocatoria/:id`, scopes( requiredScopes: [Roles.COUNSELOR_CANDIDATE.name]), ConvocatoriaController.update);
// Obtener convocatoria
routes.get( path: `${contextPath()}/convocatoria/:id`, scopes( requiredScopes: [Roles.COUNSELOR_CANDIDATE.name]), ConvocatoriaController.one);
// Eliminar convocatoria
routes.delete( path: `${contextPath()}/convocatoria/:id`, scopes( requiredScopes: [Roles.COUNSELOR_CANDIDATE.name]), ConvocatoriaController.delet
// Listar convocatoria
routes.get( path: `${contextPath()}/convocatorias`, scopes( requiredScopes: [Roles.COUNSELOR_CANDIDATE.name]), ConvocatoriaController.all);
```

Frontend



Código de Convocatoria	Tipo de convocatoria	Programa de formación	Fecha de registro	Consejero candidato	Estado	Número de asistentes	Acciones
CONV-0016-2025	Presencial	Presencial	13/01/2025	ANDRES RODRIGUEZ	Creada	0	...
CONV-0001-2025	Presencial	Telefónica	11/01/2025	ANDRES RODRIGUEZ	Creada	0	...
CONV-0002-2025	Presencial	Presencial	11/01/2025	ANDRES RODRIGUEZ	Creada	0	...
CONV-0004-2025	Presencial	Otra	11/01/2025	ANDRES RODRIGUEZ	Creada	0	...

Clave	Resumen	Estado	Sprint
			
SCRUM-66	2 [28] Relación Rol, Estado y acciones permitidas en convocatorias	Finalizada	Tablero Sprint 10

Clave	Resumen	Estado	Sprint
Frontend 			
SCRUM-66	2.1 Visualización de la convocatoria	Finalizada	Tablero Sprint 10

Clave	Resumen	Estado	Sprint
Frontend 			
SCRUM-66	2,2 [28] Modificación de la convocatoria.	Finalizada	Tablero Sprint 10
Frontend  Backend			

Clave	Resumen	Estado	Sprint

```
/**
 * Actualiza una convocatoria
 * @param req
 * @param res
 * @return {Promise<void>}
 */
async update(req, res) : Promise<void> { Show usages  Brayan Salgado
  try {
    const {id} = req.params;
    const {
      codigo_convocatoria,
      id_programa_formacion,
      id_tipo_convocatoria,
      id_consejero_candidato,
      estado,
      numero_personas_inscritas
    } = req.body;

    const convocatoria : Convocatoria = await Convocatoria.findByPk(id);
    if (!convocatoria) {
      return sendErrorResponse(res, statusCode: 404, message: "Convocatoria no encontrada");
    }

    // Validar que el estado exista
    if (!Object.values(EstadoConvocatoria).includes(estado)) {
      return sendErrorResponse(res, statusCode: 400, message: "El estado no es válido");
    }

    // Validar que el programa de formación exista
    const programaFormacion : ProgramaFormacion = await ProgramaFormacion.findByPk(id_programa_formacion);
    if (!programaFormacion) {
```

SCRUM-66	2,3 [28] Eliminación de la convocatoria	Finalizada	Tablero Sprint 10
Backend			

Clave	Resumen	Estado	Sprint

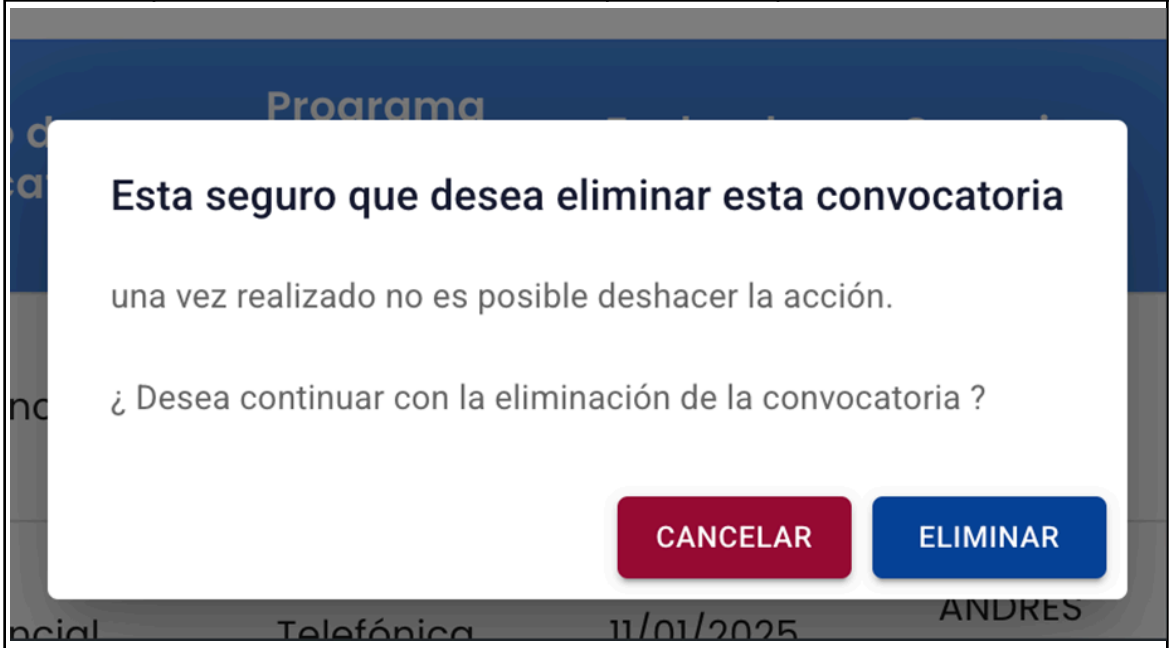
```
/**
 * Elimina una convocatoria
 * @param req
 * @param res
 * @return {Promise<void>}
 */
async delete(req, res) : Promise<void> { Show usages  Brayan Salgado
  try {
    const { id } = req.params;
    const userId = req.userId;

    const convocatoria : Convocatoria = await Convocatoria.findByPk(id);
    if (!convocatoria) {
      return sendErrorResponse(res, statusCode: 404, message: "Convocatoria no encontrada");
    }

    await convocatoria.update( keys: {
      deletedAt: new Date(),
      deletedBy: userId
    });

    return sendSuccessResponse(res, data: "Convocatoria eliminada exitosamente");
  } catch (error) {
    return sendErrorResponse(res, statusCode: 500, message: "Error procesando la solicitud", error);
  }
}
```

Frontend

Clave	Resumen	Estado	Sprint
			
SCRUM-66	3 [29] Realizar registro: debe permitir a través de cargue masivo, registrar las personas inscritas al programa de formación, siempre y cuando estén registradas en la plataforma y tengan la hoja de vida al 100%.	Finalizada	Tablero Sprint 10
Frontend			

Clave	Resumen	Estado	Sprint



Backend

Clave	Resumen	Estado	Sprint
SCRUM-66	4. [28] Visualización de los asistentes a la convocatoria	Finalizada	Tablero Sprint 10

```
/**
 * Registrar participante a una convocatoria
 * @param req
 * @param res
 * @return {Promise<void>}
 */
async store(req, res) : Promise<void> { Show usages  Brayan Salgado *
  try {
    const file = req.file;
    const {convocatoria_id} = req.body

    if (!file) {
      return sendErrorResponse(res, statusCode: 400, message: "No se ha proporcionado ningún archivo.");
    }

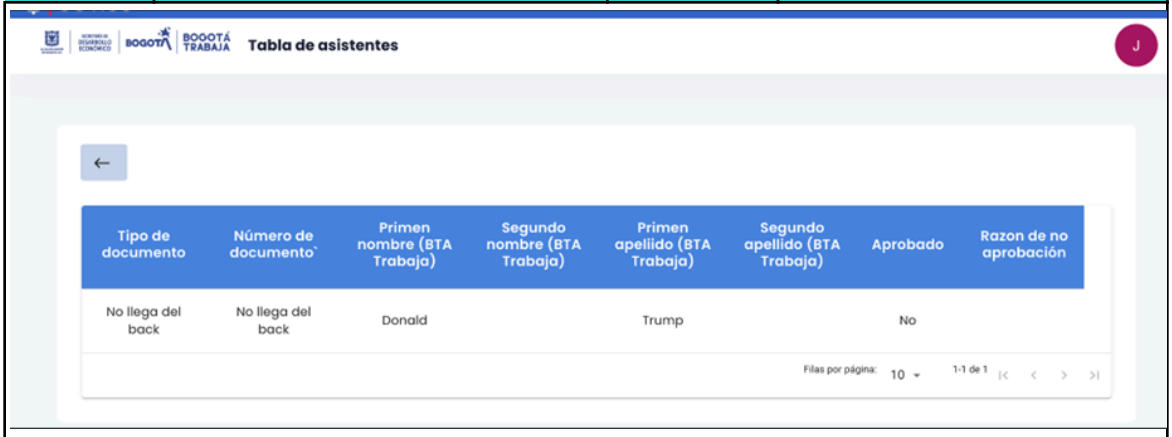
    const convocatoria : Convocatoria = await Convocatoria.findByPk(convocatoria_id)

    if (!convocatoria) {
      return sendErrorResponse(res, statusCode: 404, message: "Convocatoria no encontrada");
    }

    if (convocatoria.estado === EstadoConvocatoria.AsistentesCargados) {
      return sendErrorResponse(
          res,
          statusCode: 400,
          message: "El proceso de registro de usuarios ya fue cerrado, no es posible realizar más registros."
      );
    }

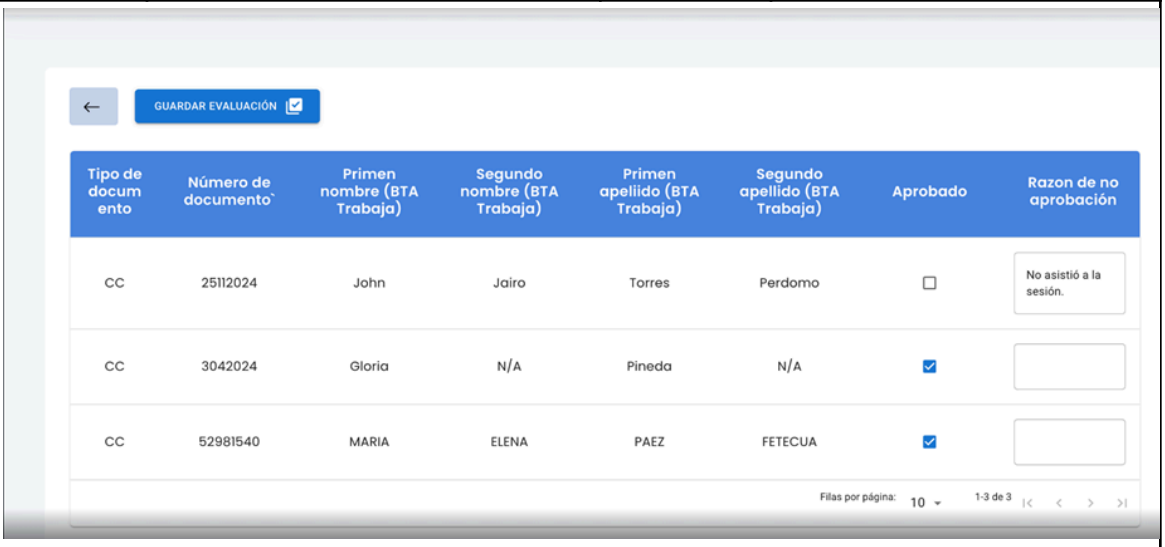
    if (convocatoria.estado === EstadoConvocatoria.Finalizada) {
      return sendErrorResponse(res, statusCode: 400, message: "Convocatoria finalizada.");
    }
```

Frontend

Clave	Resumen	Estado	Sprint
			
[SCRUM-66](#)	5 [30] Realizar proceso evaluación: marcar como formados los registros de las personas que culminaron el proceso de formación.	Finalizada	Tablero Sprint 10

Frontend

Clave	Resumen	Estado	Sprint



Backend

```
/**
 * Aprobar o desaprobar múltiples participantes de una convocatoria
 * @param req
 * @param res
 * @return {Promise<void>}
 */
async aprobar(req, res) : Promise<void> { Show usages  Brayan Salgado
  try {
    const {convocatoriaId} = req.params;
    const {aspirantes} = req.body;

    // Validación inicial: verificar estructura del array recibido
    if (!Array.isArray(aspirantes) || aspirantes.length === 0) {
      return sendErrorResponse(res, statusCode: 400, message: "Debe proporcionar un array de aspirantes con sus estados de aprobación.");
    }

    // Verificar si la convocatoria existe
    const convocatoria : Convocatoria = await Convocatoria.findByPk(convocatoriaId);
    if (!convocatoria) {
      return sendErrorResponse(res, statusCode: 404, message: "Convocatoria no encontrada.");
    }

    if (convocatoria.estado === EstadoConvocatoria.Finalizada) {
      return sendErrorResponse(res, statusCode: 400, message: "Convocatoria finalizada.");
    }

    const cantidadRegistrados : GroupedCountResultItem[] = await ConvocatoriaParticipante.count( options: {where: {convocatoria_id: convocatoriaId}});
```

Clave	Resumen	Estado	Sprint
SCRUM-66	6 [31] Generar certificado de formación: opción para generar tanto individual como masivamente, los certificados de formación para las personas que culminaron el proceso	Finalizada	Tablero Sprint 10

Frontend



Clave	Resumen	Estado	Sprint



Tipo de documento	Número de documento	Primen nombre (BTA Trabaja)	Segundo nombre (BTA Trabaja)	Primen apeliido (BTA Trabaja)	Segundo apellido (BTA Trabaja)	Aprobado	Razon de no aprobación	Certificado
CC	25112024	John	Jairo	Torres	Perdomo	No	N/A	—
CC	3042024	Gloria	N/A	Pineda	N/A	No	N/A	—
CC	52981540	MARIA	ELENA	PAEZ	FETECUA	No	N/A	—

Filas por página: 10 1-3 de 3

Control de cambios

Versión	Elaborado por	Fecha de creación	Fecha de aprobación	Aprobado	Descripción
1	3DStudio	06/12/2024	10/02/2025	Gerente de proyecto	Creación del documento