Barranquilla, 12 de septiembre 2023.

YULLY CATHERINE ROJAS MORENO

Contrato No CD-06-2023-2147

Secretaria de Planeación.

Alcaldía Distrital de Barranquilla. Ciudad.

Asunto: **QUINTO INFORME DE GESTIÓN.**

En cumplimiento del objeto del contrato de prestación de servicios No. CD-06- 2022-1476,que usted supervisa, me permito rendirle informe de las actividades realizadas en la **PRESTACION DE SERVICIOS PROFESIONALES PARA APOYAR A LA SECRETARIA DISTRITAL DE PLANEACION EN EL DESARROLLO DE LOS PROGRAMAS Y PROYECTOS RELACIONADOS CON GESTION (12) DE SEPTIEMBRE DE DOS MILVEINTIDOS (2023).**

1. Desarrollar software para la secretaria Distrital de Planeación.

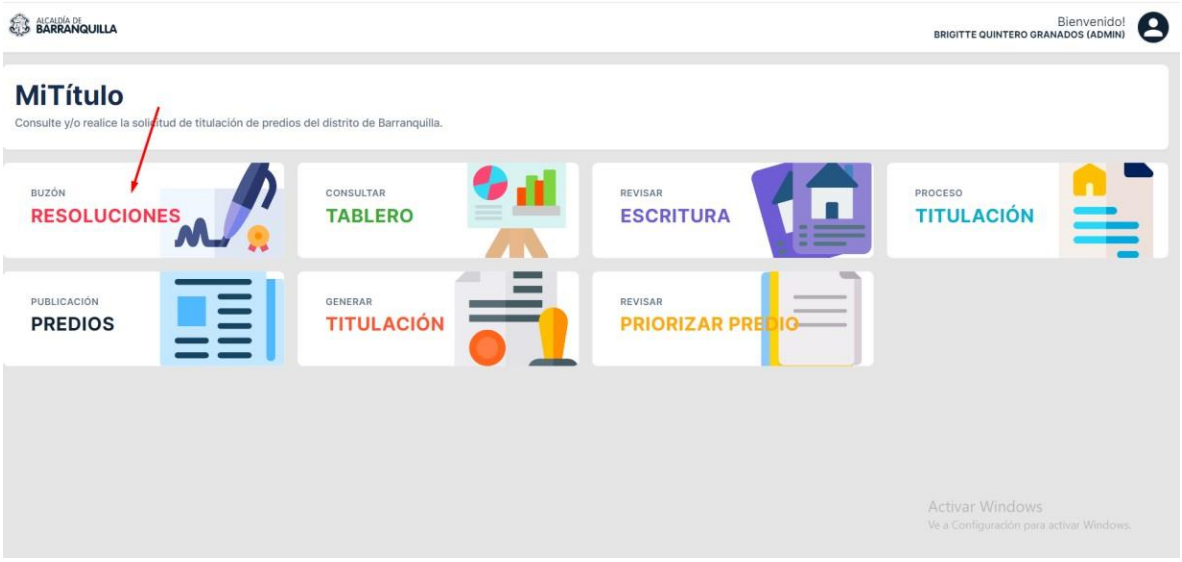
2. Actualizar funcionalidades y desarrollo del sistema de Panorama Urbano, Fichas de Barrio, y Mi Título.

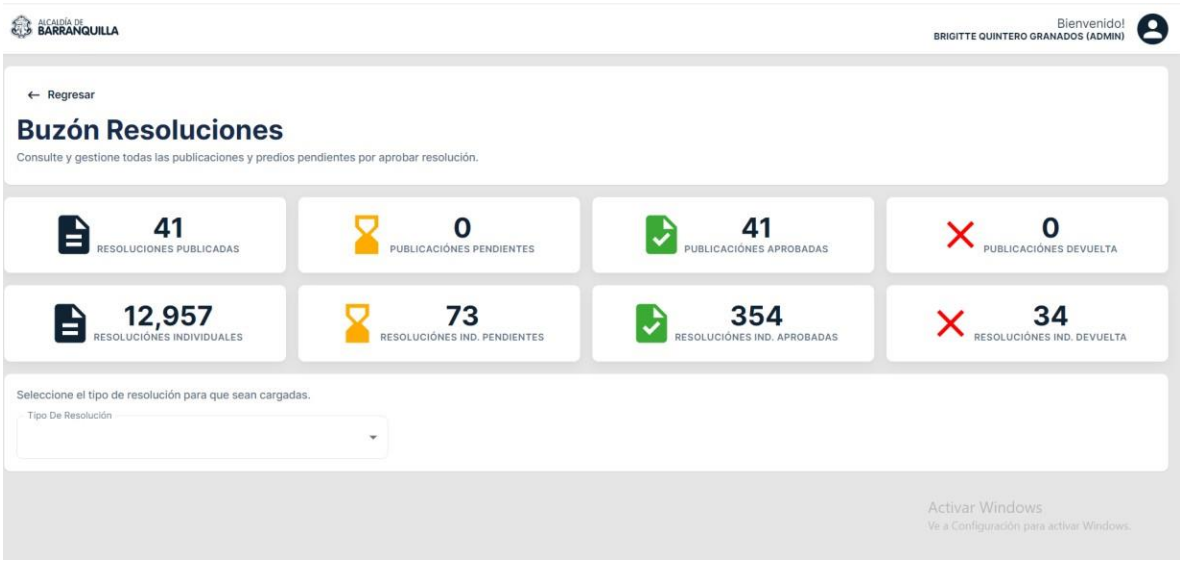
3. Dar soporte técnico a las bases de datos y a los sistemas desarrollados para el manejo de la información de la secretaria Distrital de Planeación.

- **Mititulo.**

<u>Evidencias funcionales:</u>

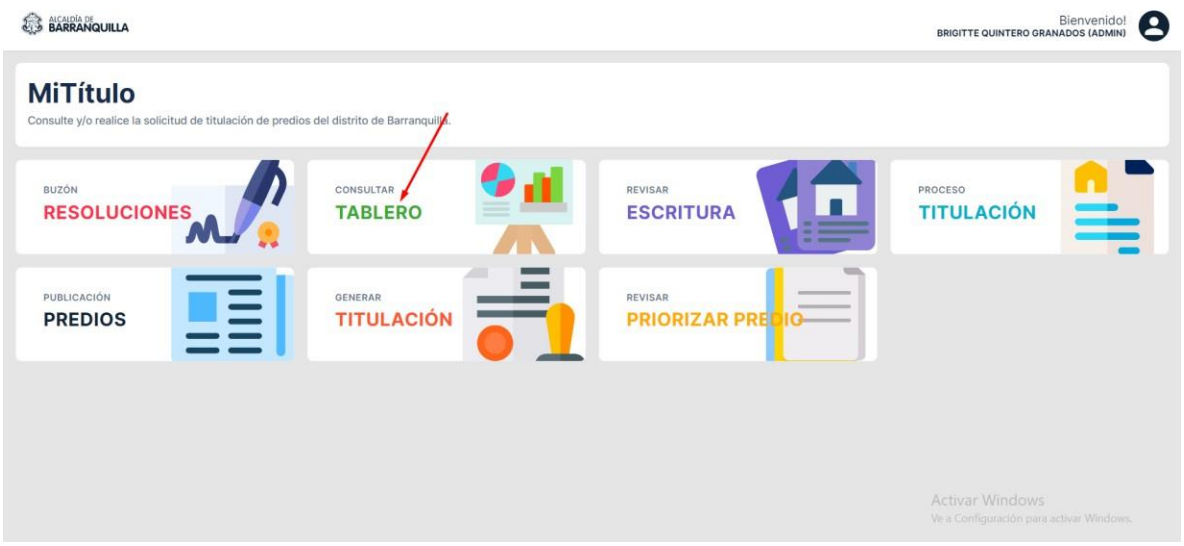
Buzón de Resoluciones.
Se creo el botón de resoluciones como Resoluciones públicas, publicaciones pendientes, publicaciones aprobadas, publicaciones devueltas, resoluciones individuales y publicaciones devueltas.

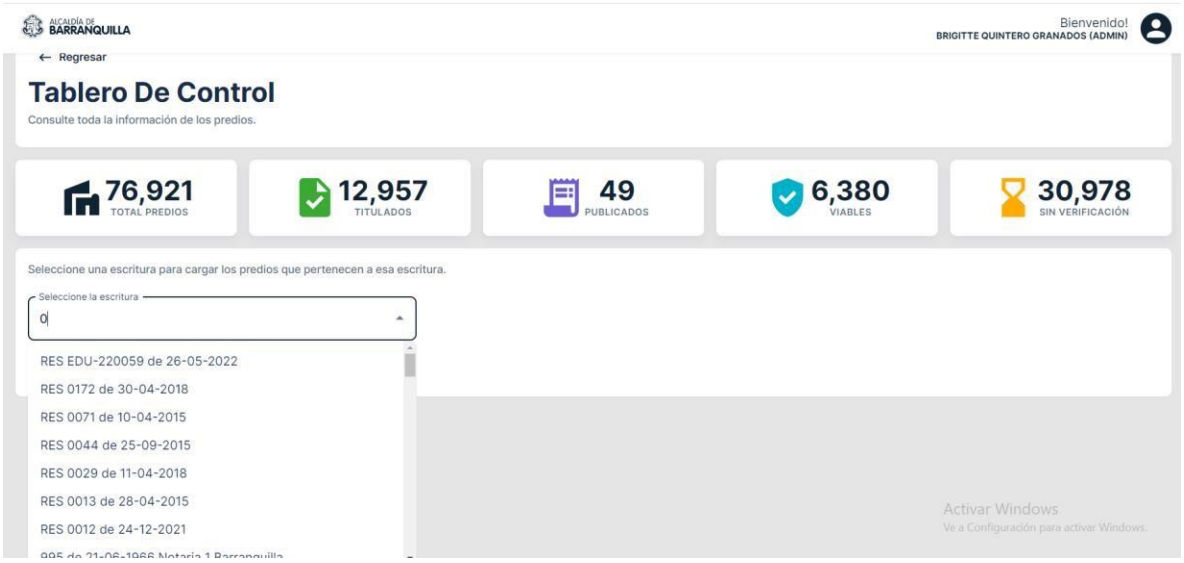




Tablero de consultas.

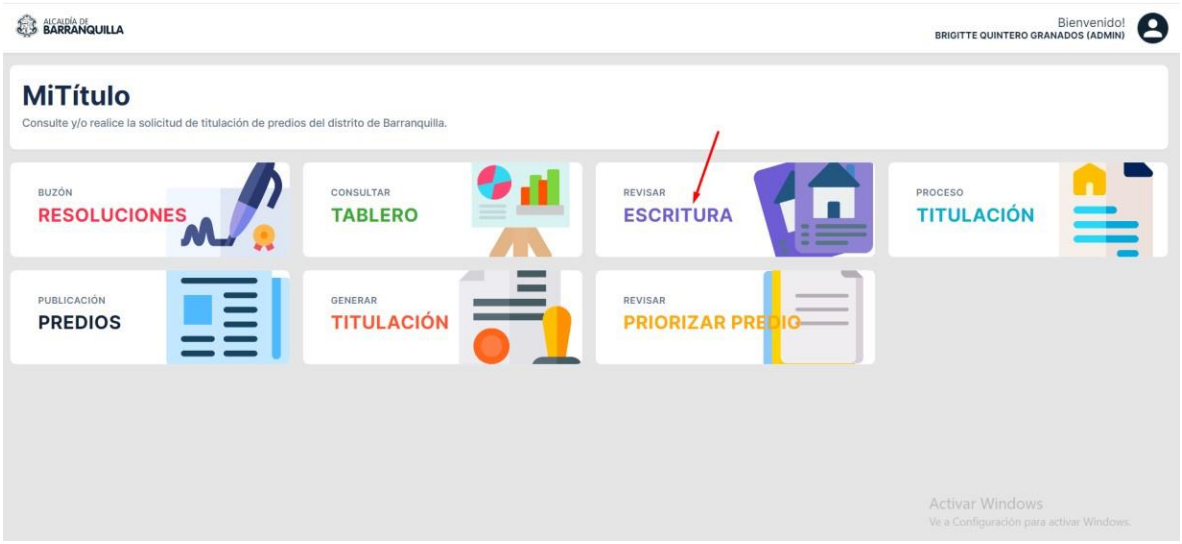
Tablero de control permite realizar búsqueda de toda la información de los predios.

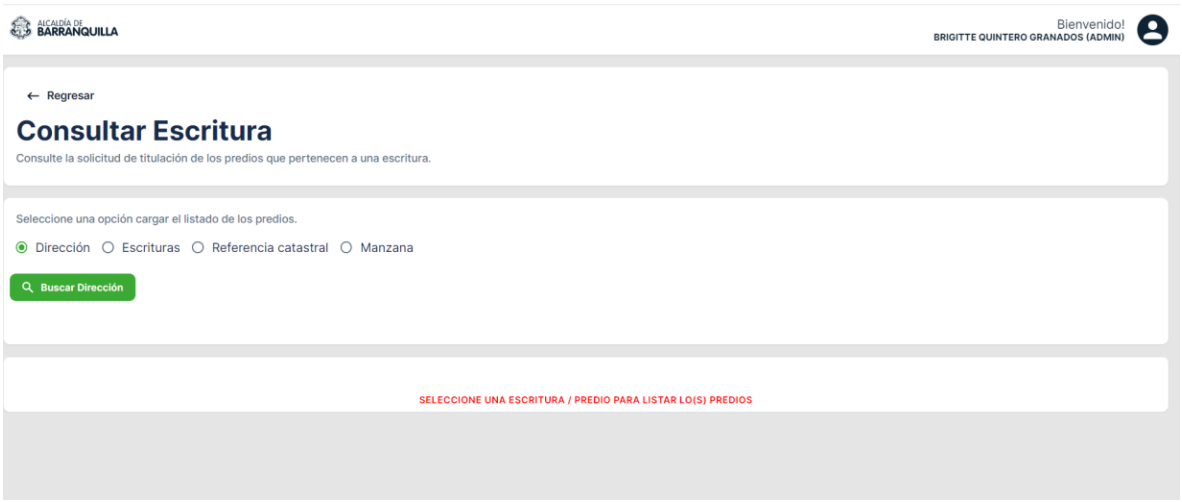




Revisar escritura.

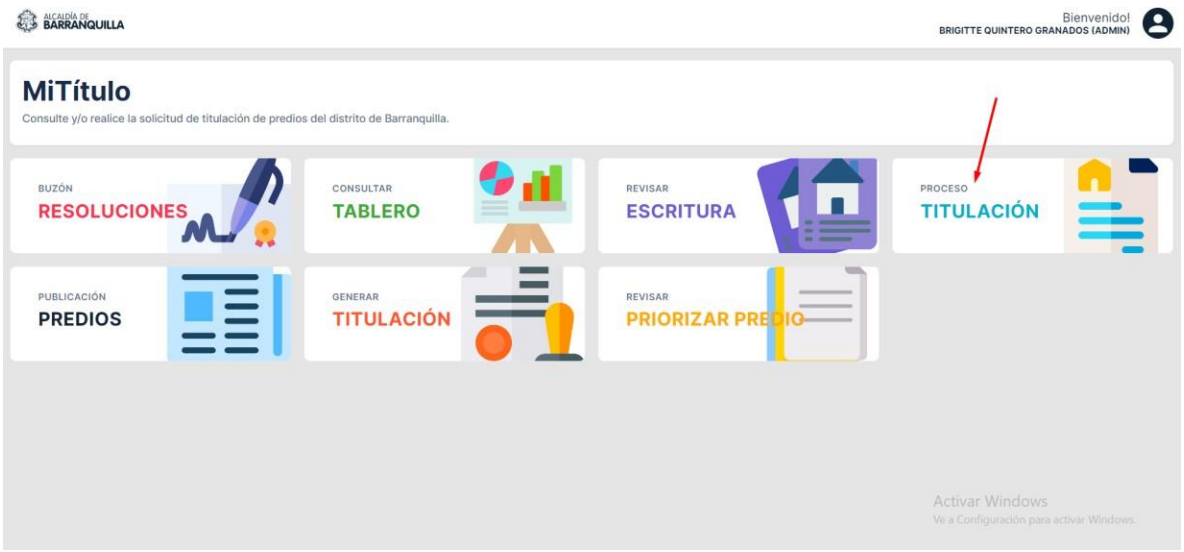
Es consultar la solicitud de titulación de los predios del distrito de Barranquilla.

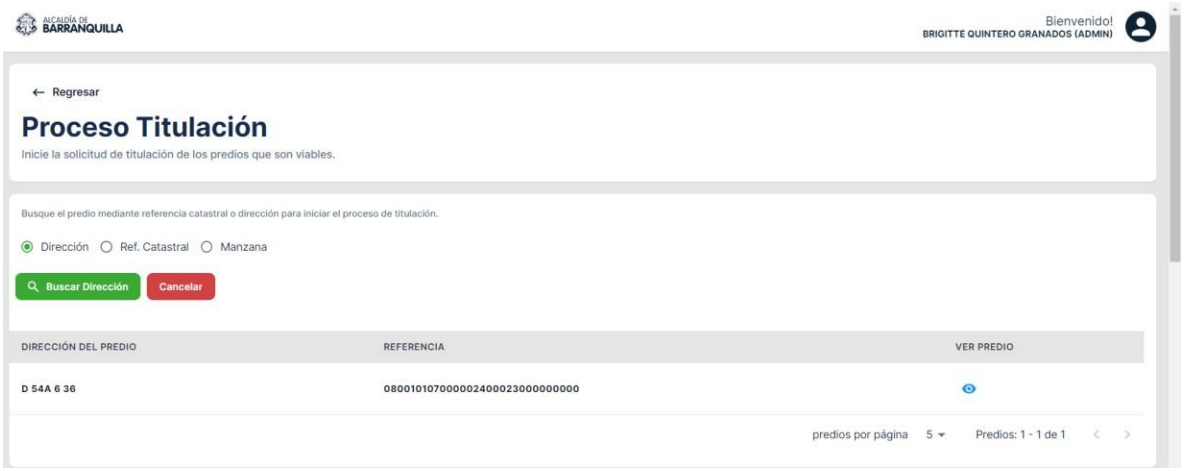




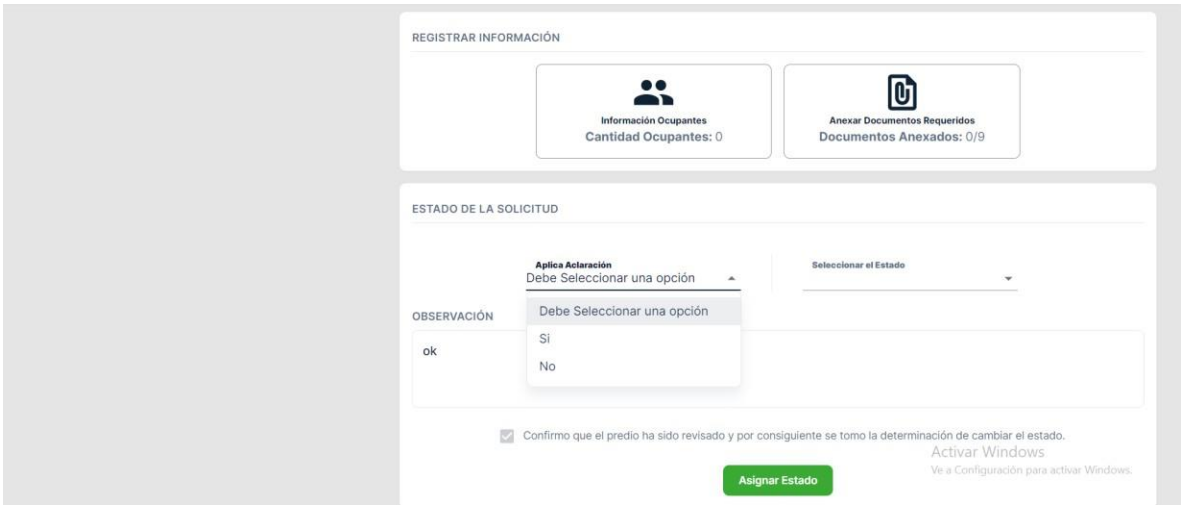
Proceso de titulación.

Realizar búsquedas de predios mediante referencia catastrales o dirección para iniciar el proceso de titulación.

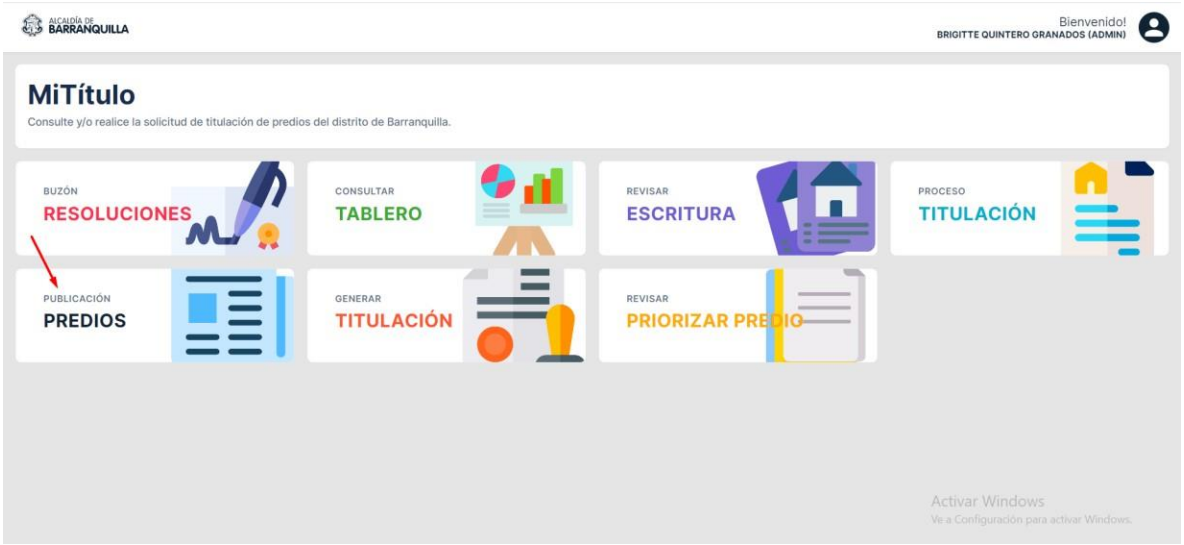

ALCALDÍA DE BARRANQUILLA
Bienvenido!
BRIGITTE QUINTERO GRANADOS (ADMIN)
MiTítulo
Consulte y/o realice la solicitud de titulación de predios del distrito de Barranquilla.
BUZÓN
RESOLUCIONES
CONSULTAR
TABLERO
REVISAR
ESCRITURA
PROCESO
TITULACIÓN
PUBLICACIÓN
PREDIOS
GENERAR
TITULACIÓN
REVISAR
PRIORIZAR PREDIO
Activar Windows
Ve a Configuración para activar Windows.


ALCALDÍA DE BARRANQUILLA
Bienvenido!
BRIGITTE QUINTERO GRANADOS (ADMIN)
← Regresar
Proceso Titulación
Inicie la solicitud de titulación de los predios que son viables.
Busque el predio mediante referencia catastral o dirección para iniciar el proceso de titulación.
Dirección
Ref. Catastral
Manzana
Buscar Dirección
Cancelar
DIRECCIÓN DEL PREDIO
REFERENCIA
VER PREDIO
D 54A 6 36
08001010700000240002300000000
predios por página 5
Predios: 1 - 1 de 1

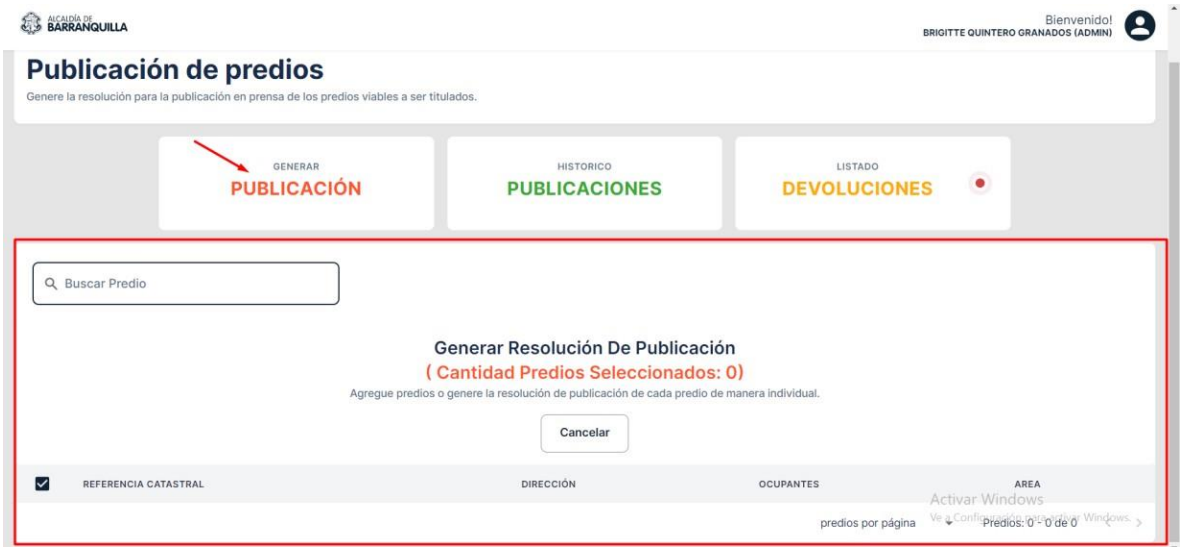

BITÁCORA DEL PREDIO
Nuevo Evento
1
FECHA
2023-3-8
USUARIO GESTIÓN
BRIGITTE QUINTERO GRANADOS (ADMIN)
OBSERVACIÓN
ok - Paso a Estado EN PROCESO / VIABLE
INFORMACIÓN DEL PREDIO
Matrícula:
NULL
Referencia Catastral:
08001010700000240002300000000
Dirección:
D 54A 6 36
Barrio:
LA SIERRITA
Amenaza:
ALTA, BAJA, MEDIA, MUY ALTA
Usos:
AREA DE RESERVA Y PROTECCION
Destino Económico:
NO REGISTRA
Escritura:
995 DE 21-06-1966 NOTARIA 1 BARRANQUILLA
Estado:
EN PROCESO / VIABLE
Fecha Viabilidad:
08/03/23
Área:
456 M²



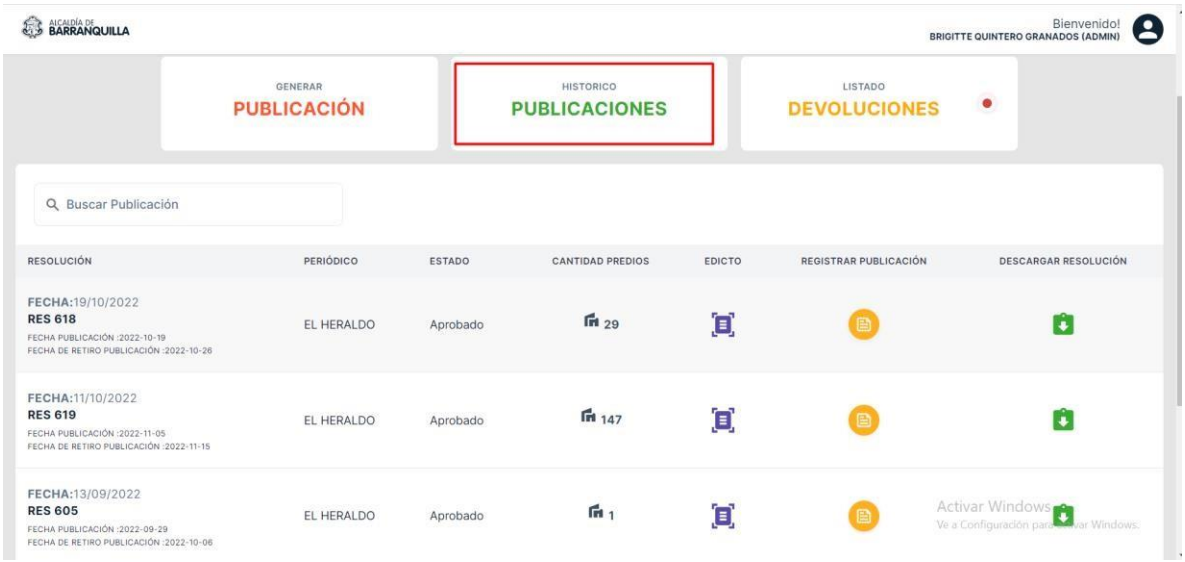
Publicación de predios.



Generar publicaciones.



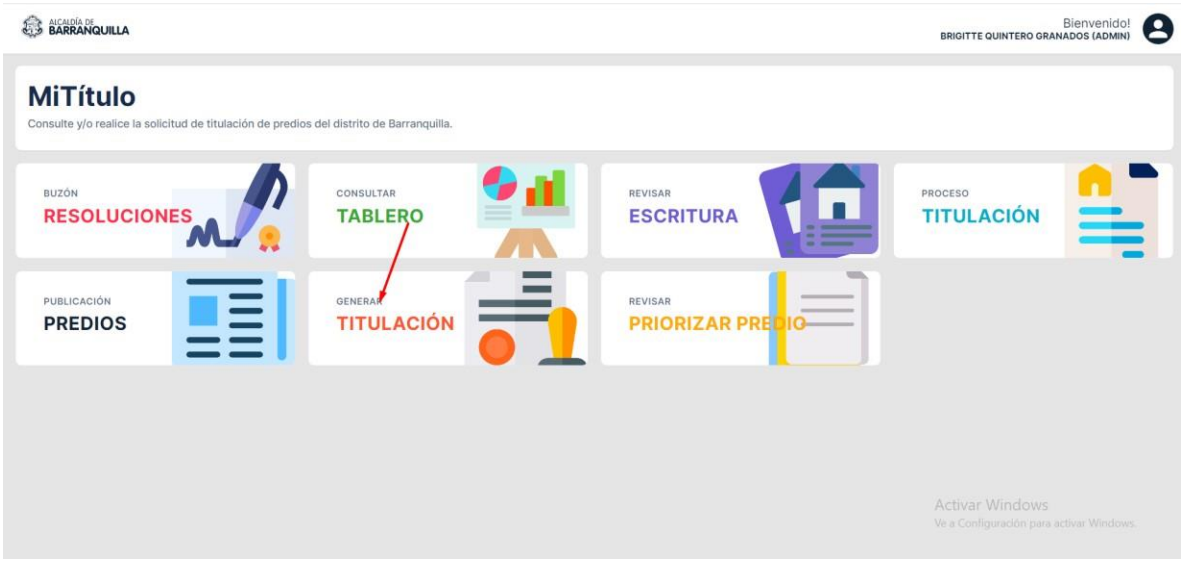
Se visualiza el histórico de publicaciones.



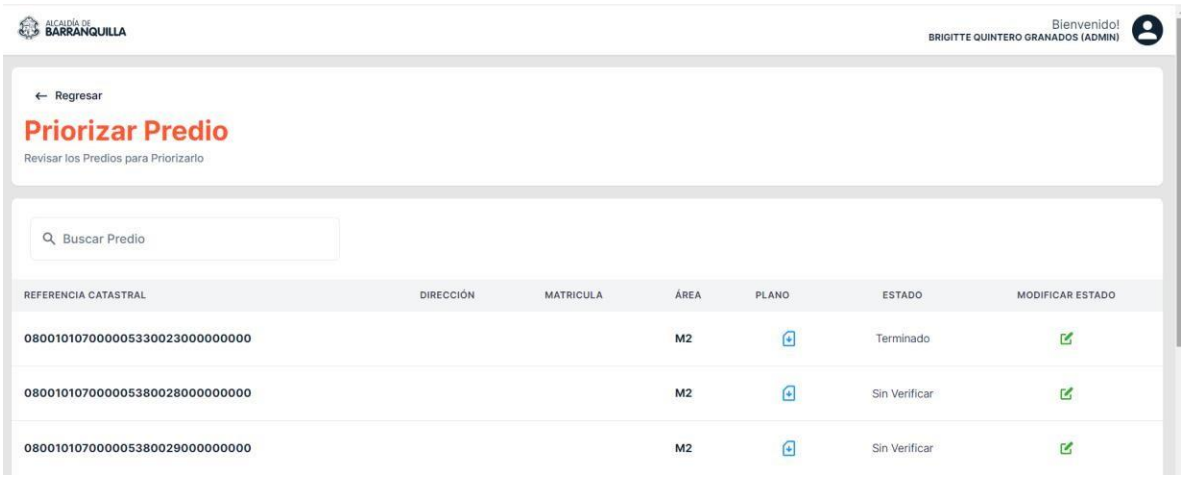
Listado de devoluciones.



Generar titulación.



Priorizar predio.

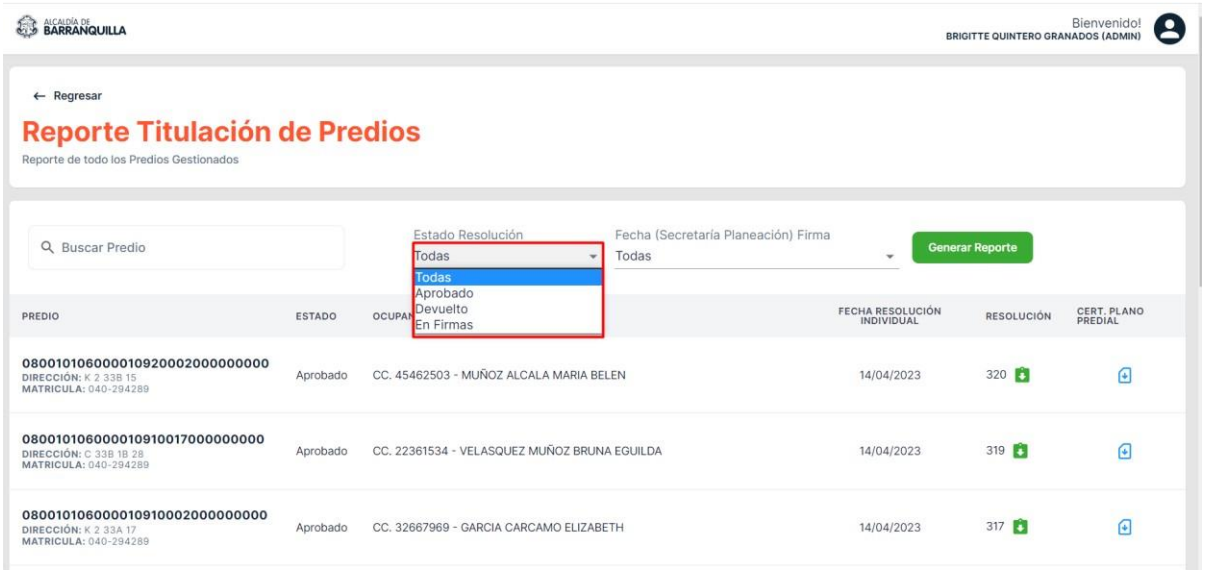


Titulación de reporte.



Reportes para generar predios gestionados.

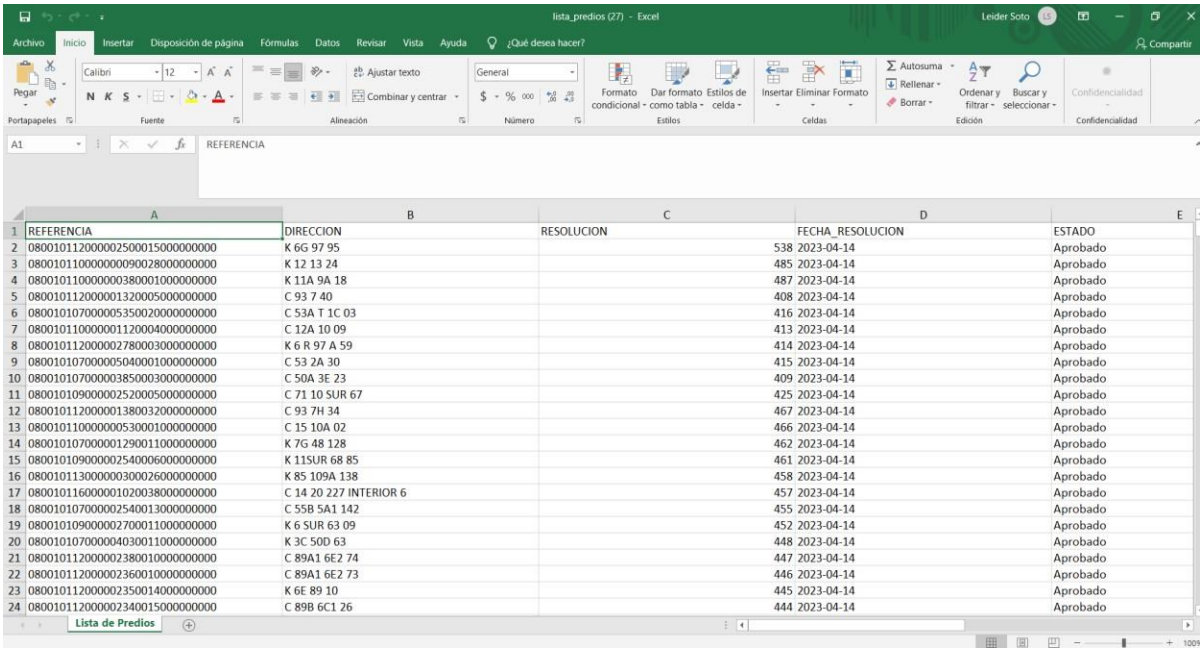
Estado de la resolución, **todos.**



Generar reporte.



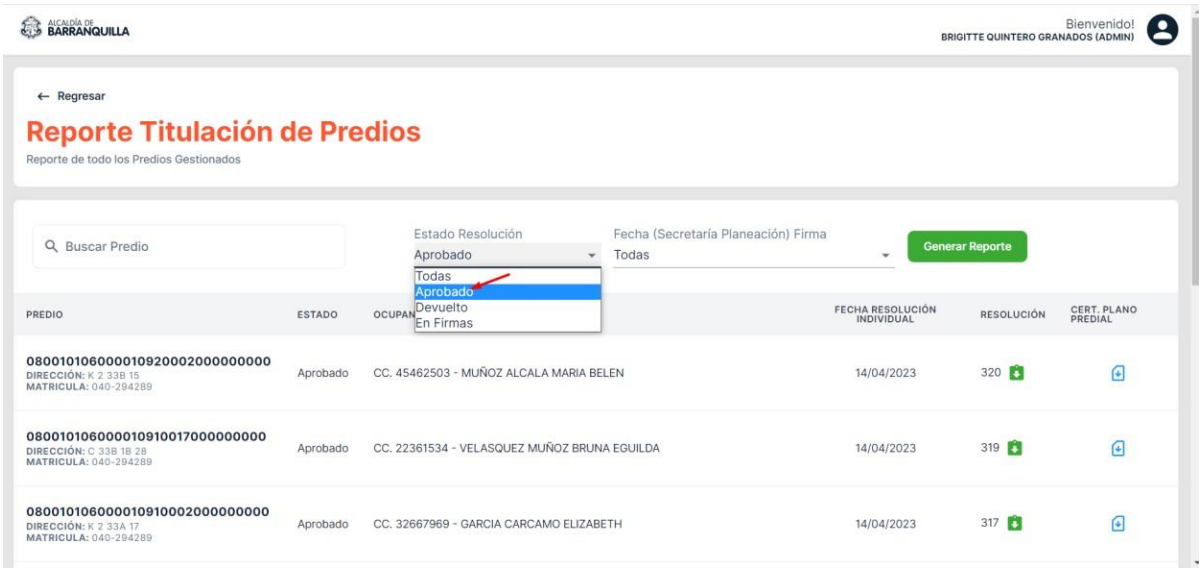
Se visualiza la información de la gestión mediante la descarga del Excel.



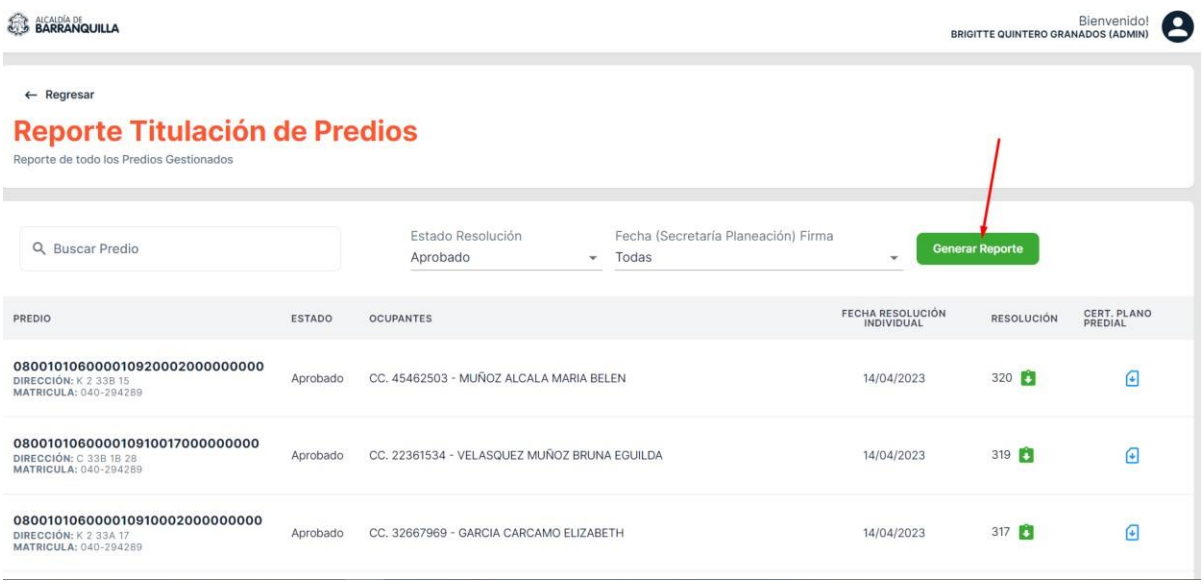
lista_predios (27) - Excel

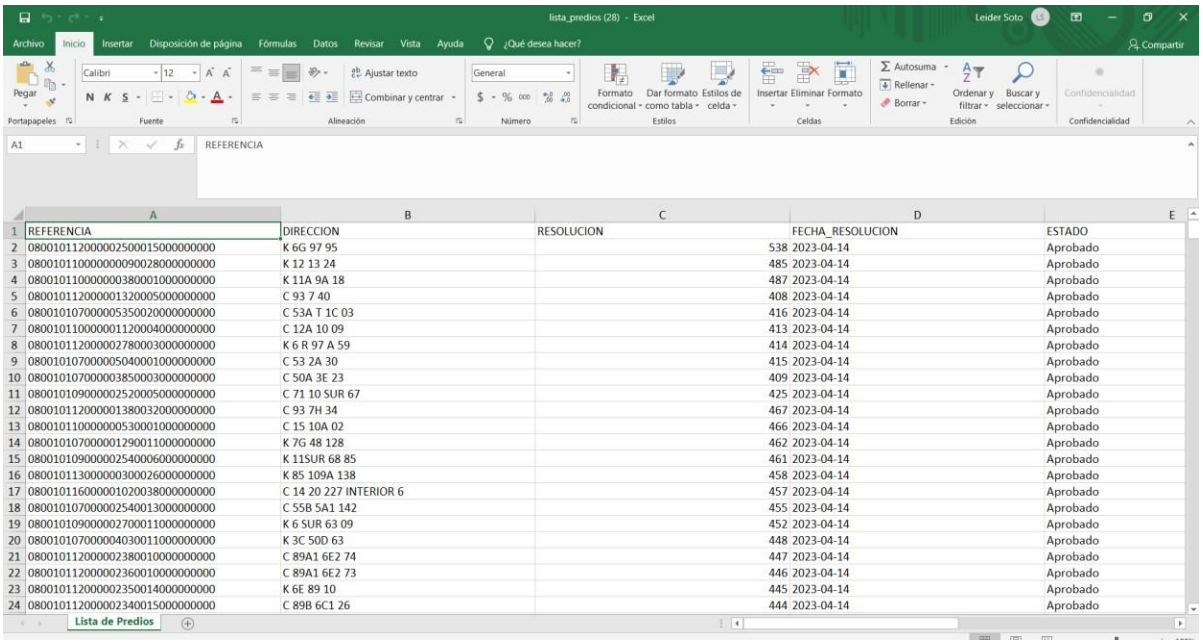
	A	B	C	D	E
1	REFERENCIA	DIRECCION	RESOLUCION	FECHA_RESOLUCION	ESTADO
2	08001011200000250001500000000	K 6G 97 95	538	2023-04-14	Aprobado
3	08001011000000009002800000000	K 12 13 24	485	2023-04-14	Aprobado
4	08001011000000038000100000000	K 11A 9A 18	487	2023-04-14	Aprobado
5	08001011200000132000500000000	C 93 7 40	408	2023-04-14	Aprobado
6	08001010700000535002000000000	C 53A T 1C 03	416	2023-04-14	Aprobado
7	08001011000000112000400000000	C 12A 10 09	413	2023-04-14	Aprobado
8	08001011200000278000300000000	K 6 R 97 A 59	414	2023-04-14	Aprobado
9	08001010700000504000100000000	C 53 2A 30	415	2023-04-14	Aprobado
10	08001010700000385000300000000	C 50A 3E 23	409	2023-04-14	Aprobado
11	08001010900000252000500000000	C 71 10 SUR 67	425	2023-04-14	Aprobado
12	08001011200000138003200000000	C 93 7H 34	467	2023-04-14	Aprobado
13	08001011000000053000100000000	C 15 10A 02	466	2023-04-14	Aprobado
14	08001010700000129001100000000	K 7G 48 128	462	2023-04-14	Aprobado
15	08001010900000254000600000000	K 11SUR 68 85	461	2023-04-14	Aprobado
16	08001011300000030002600000000	K 85 109A 138	458	2023-04-14	Aprobado
17	08001011600000102003800000000	C 14 20 227 INTERIOR 6	457	2023-04-14	Aprobado
18	08001010700000254001300000000	C 55B 5A1 142	455	2023-04-14	Aprobado
19	08001010900000270001100000000	K 6 SUR 63 09	452	2023-04-14	Aprobado
20	08001010700000403001100000000	K 3C 50D 63	448	2023-04-14	Aprobado
21	08001011200000238001000000000	C 89A1 6E2 74	447	2023-04-14	Aprobado
22	08001011200000236001000000000	C 89A1 6E2 73	446	2023-04-14	Aprobado
23	08001011200000235001400000000	K 6E 89 10	445	2023-04-14	Aprobado
24	08001011200000234001500000000	C 89B 6C1 26	444	2023-04-14	Aprobado

Estado de la resolución, **aprobado.**



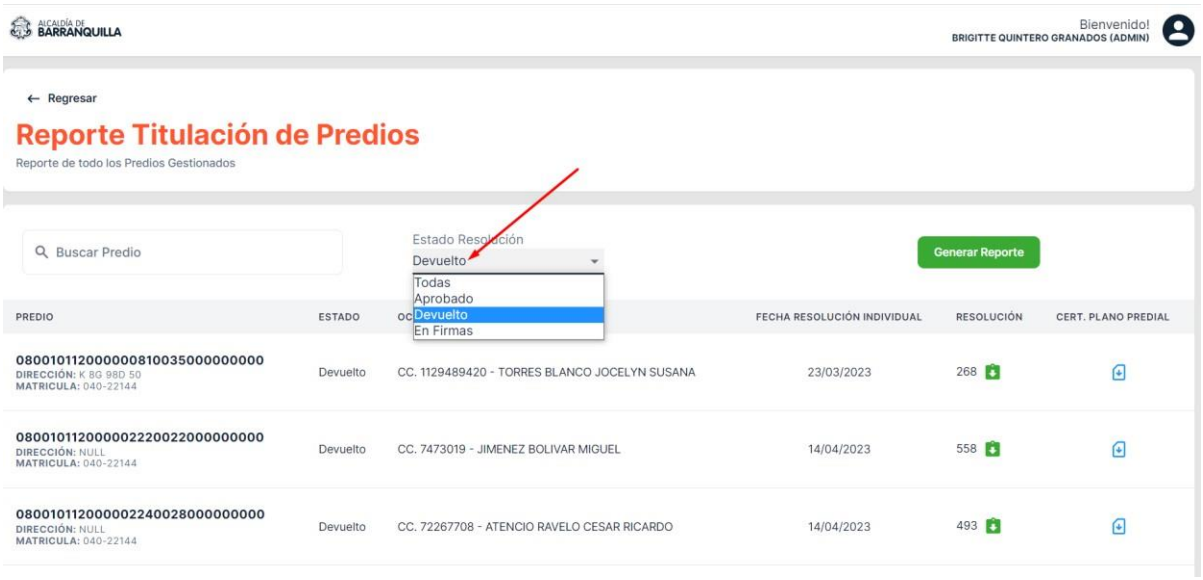
General Excel.



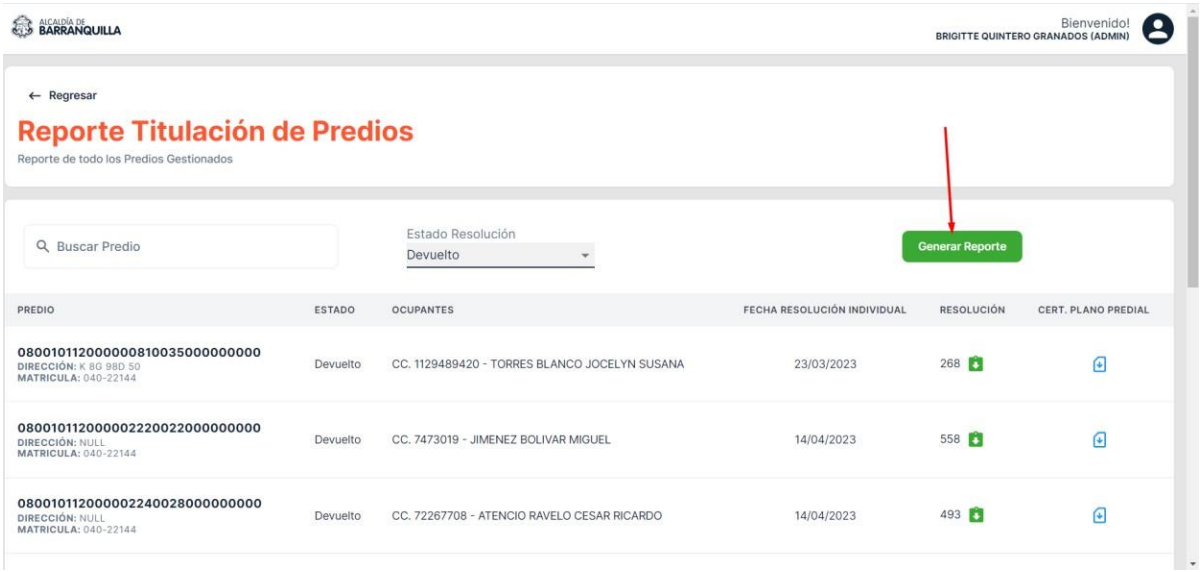


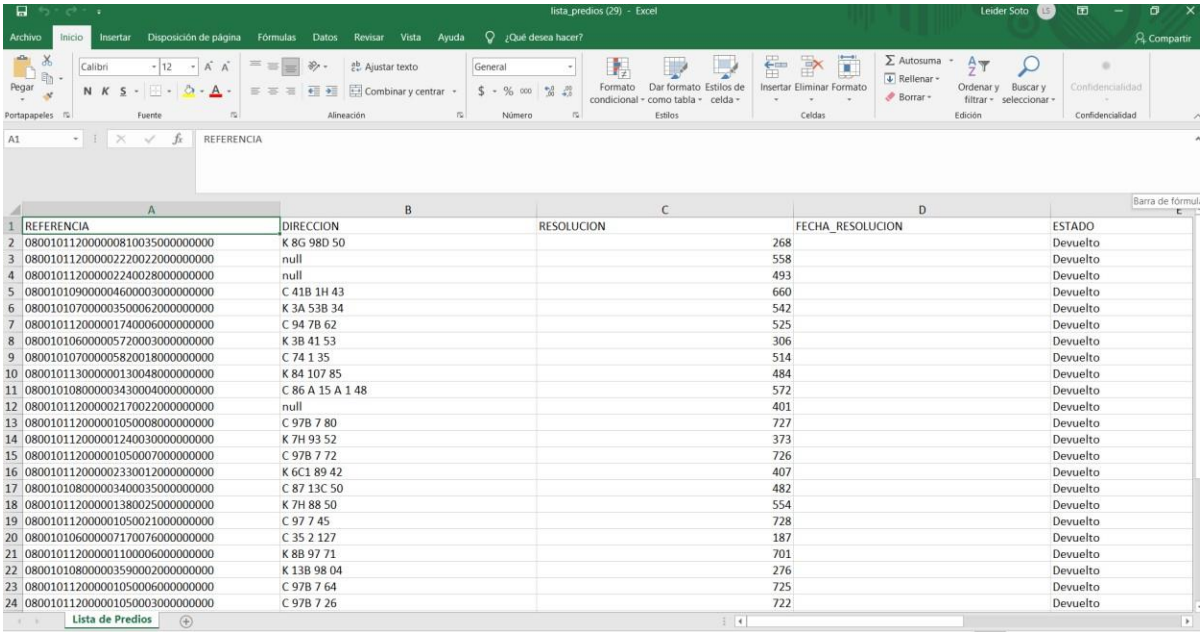
REFERENCIA	DIRECCION	RESOLUCION	FECHA_RESOLUCION	ESTADO
08001011200000250001500000000	K 6G 97 95	538	2023-04-14	Aprobado
08001011000000009002800000000	K 12 13 24	485	2023-04-14	Aprobado
08001011000000038000100000000	K 11A 9A 18	487	2023-04-14	Aprobado
08001011200000132000500000000	C 93 7 40	408	2023-04-14	Aprobado
08001010700000535002000000000	C 53A T 1C 03	416	2023-04-14	Aprobado
08001011000000112000400000000	C 12A 10 09	413	2023-04-14	Aprobado
08001011200000278000300000000	K 6 R 97 A 59	414	2023-04-14	Aprobado
08001010700000050400010000000	C 53 2A 30	415	2023-04-14	Aprobado
08001010700000385000300000000	C 50A 3E 23	409	2023-04-14	Aprobado
08001010900000252000500000000	C 71 10 SUR 67	425	2023-04-14	Aprobado
08001011200000138003200000000	C 93 7H 34	467	2023-04-14	Aprobado
08001011000000053000100000000	C 15 10A 02	466	2023-04-14	Aprobado
08001010700000012900110000000	K 7G 48 128	462	2023-04-14	Aprobado
08001010900000254000600000000	K 11SUR 68 85	461	2023-04-14	Aprobado
08001011300000030002600000000	K 85 109A 138	458	2023-04-14	Aprobado
08001011600000102003800000000	C 14 20 227 INTERIOR 6	457	2023-04-14	Aprobado
08001010700000254001300000000	C 55B 5A1 142	455	2023-04-14	Aprobado
08001010900000270001100000000	K 6 SUR 63 09	452	2023-04-14	Aprobado
08001010700000403001100000000	K 3C 50D 63	448	2023-04-14	Aprobado
08001011200000238001000000000	C 89A1 6E2 74	447	2023-04-14	Aprobado
08001011200000236001000000000	C 89A1 6E2 73	446	2023-04-14	Aprobado
08001011200000235001400000000	K 6E 89 10	445	2023-04-14	Aprobado
08001011200000234001500000000	C 89B 6C1 26	444	2023-04-14	Aprobado

Estado de la devolución devuelta.



Generar reporte.

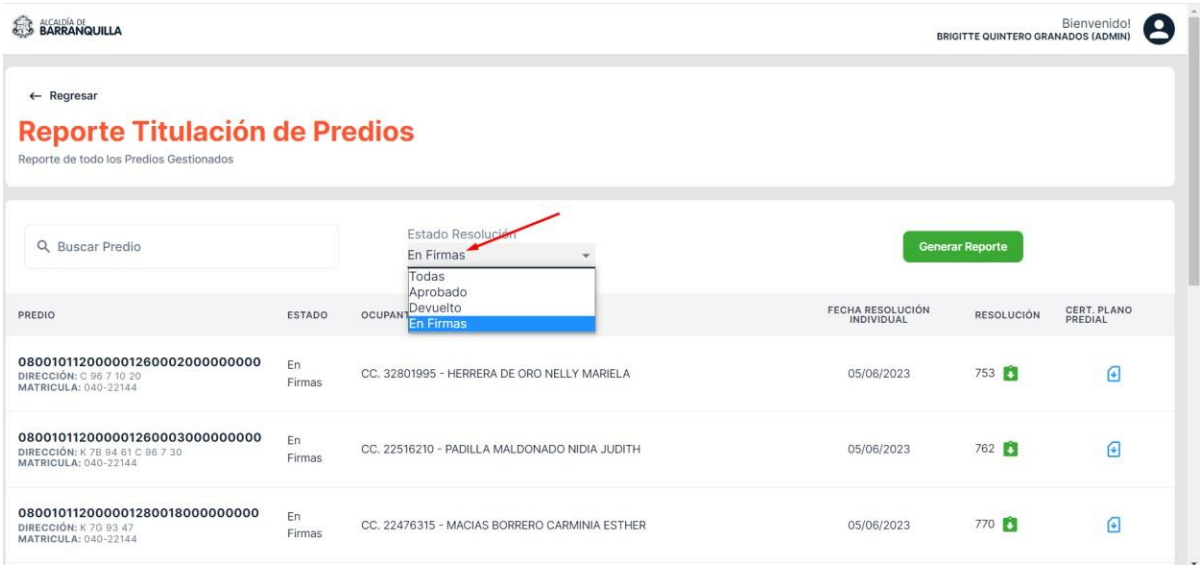




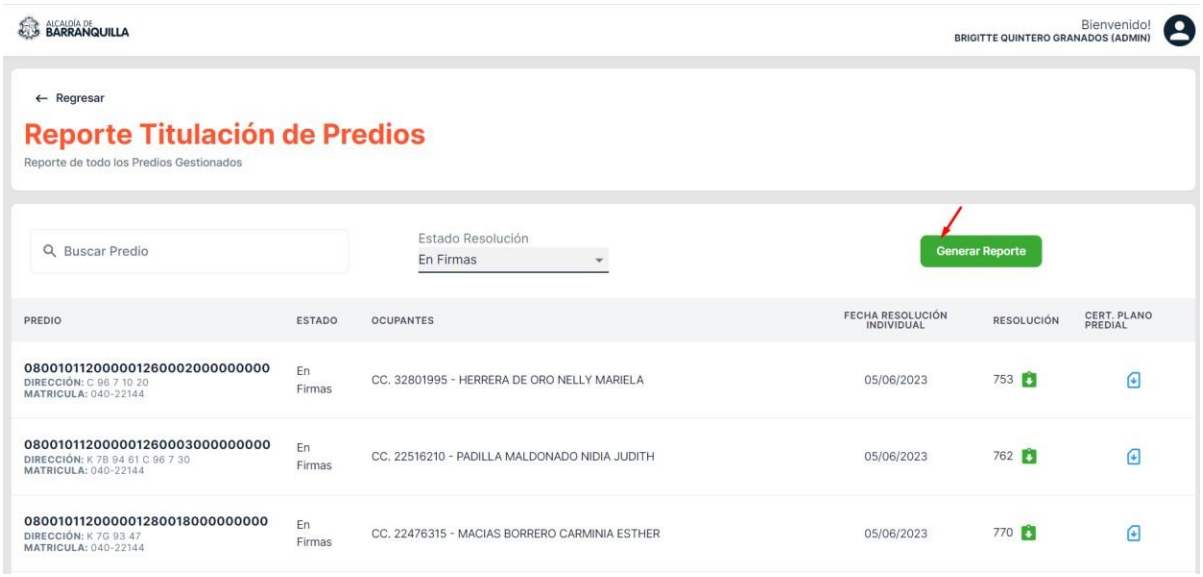
lista_predios (29) - Excel

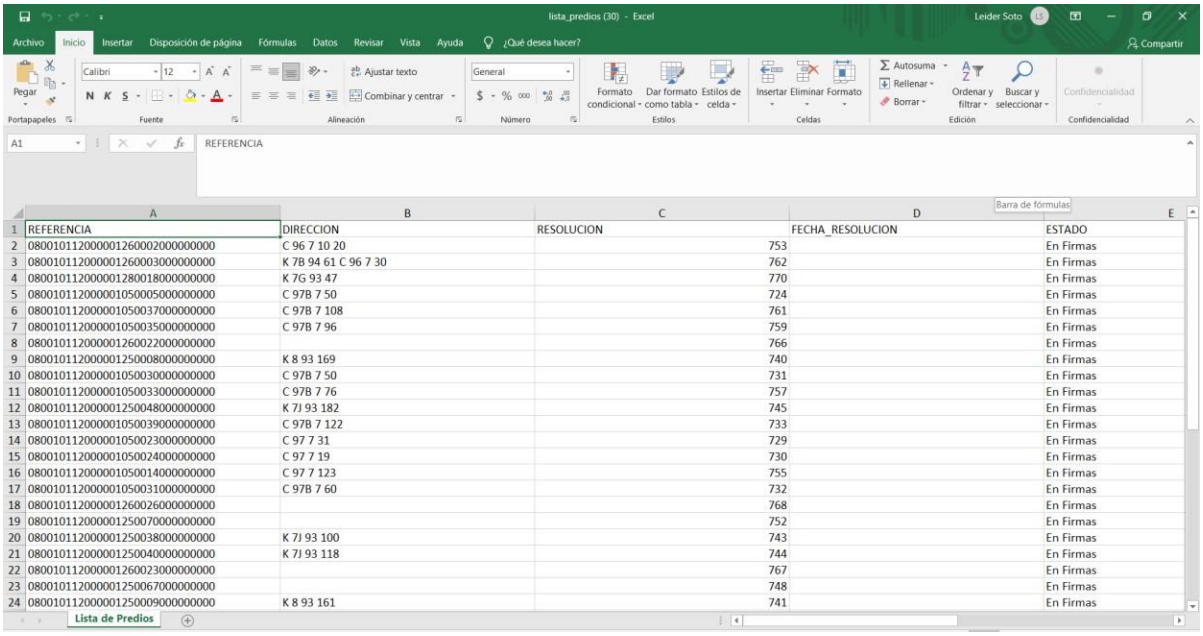
	A	B	C	D	E
1	REFERENCIA	DIRECCION	RESOLUCION	FECHA_RESOLUCION	ESTADO
2	08001011200000081003500000000	K 8G 98D 50	268		Devuelto
3	08001011200000222002200000000	null	558		Devuelto
4	08001011200000224002800000000	null	493		Devuelto
5	08001010900000460000300000000	C 41B 1H 43	660		Devuelto
6	08001010700000350006200000000	K 3A 53B 34	542		Devuelto
7	08001011200000174006600000000	C 94 7B 62	525		Devuelto
8	08001010600000572003000000000	K 3B 41 53	306		Devuelto
9	08001010700000582001800000000	C 74 1 35	514		Devuelto
10	08001011300000013004800000000	K 84 107 85	484		Devuelto
11	08001010800000343000400000000	C 86 A 15 A 1 48	572		Devuelto
12	08001011200000217002200000000	null	401		Devuelto
13	08001011200000105000800000000	C 97B 7 80	727		Devuelto
14	08001011200000124003000000000	K 7H 93 52	373		Devuelto
15	08001011200000105000700000000	C 97B 7 72	726		Devuelto
16	08001011200000233001200000000	K 6C1 89 42	407		Devuelto
17	08001010800000340003500000000	C 87 13C 50	482		Devuelto
18	08001011200000138002500000000	K 7H 88 50	554		Devuelto
19	08001011200000105002100000000	C 97 7 45	728		Devuelto
20	08001010600000717007600000000	C 35 2 127	187		Devuelto
21	08001011200000110000600000000	K 8B 97 71	701		Devuelto
22	08001010800000359000200000000	K 13B 98 04	276		Devuelto
23	08001011200000105000600000000	C 97B 7 64	725		Devuelto
24	08001011200000105000300000000	C 97B 7 26	722		Devuelto

Estado de la resolución, **En firma.**



Generar reporte.

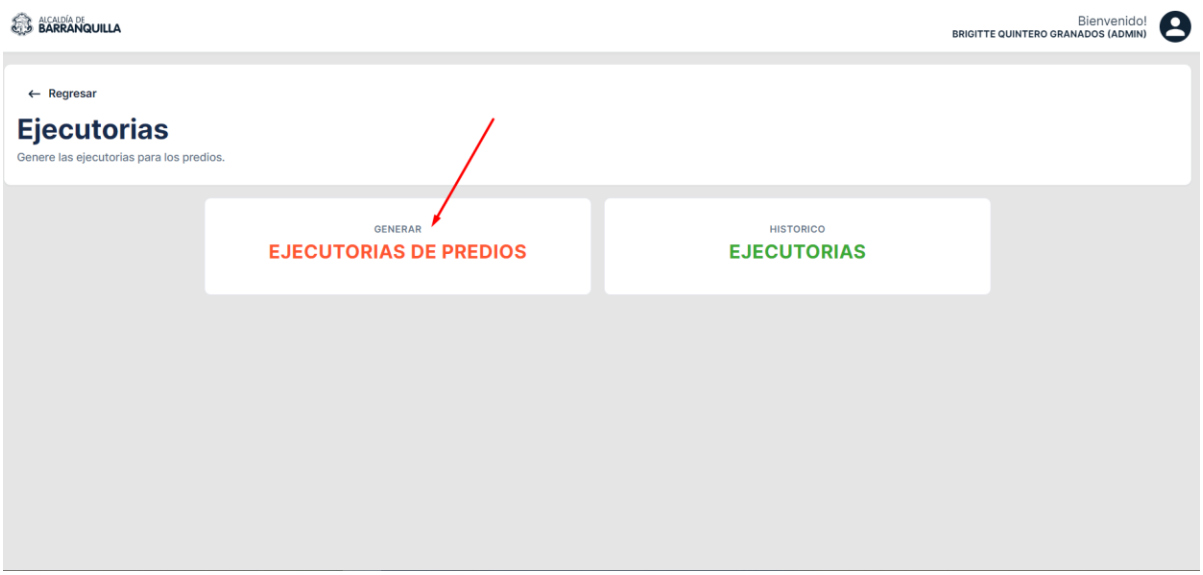




	A	B	C	D	E
1	REFERENCIA	DIRECCION	RESOLUCION	FECHA_RESOLUCION	ESTADO
2	08001011200000126002000000000	C 96 7 10 20	753		En Firmas
3	08001011200000126000300000000	K 7B 94 61 C 96 7 30	762		En Firmas
4	08001011200000128001800000000	K 7G 93 47	770		En Firmas
5	08001011200000105000500000000	C 97B 7 50	724		En Firmas
6	08001011200000105003700000000	C 97B 7 108	761		En Firmas
7	08001011200000105003500000000	C 97B 7 96	759		En Firmas
8	08001011200000126002200000000		766		En Firmas
9	08001011200000125000800000000	K 8 93 169	740		En Firmas
10	08001011200000105003000000000	C 97B 7 50	731		En Firmas
11	08001011200000105003300000000	C 97B 7 76	757		En Firmas
12	08001011200000125004800000000	K 7J 93 182	745		En Firmas
13	08001011200000105003900000000	C 97B 7 122	733		En Firmas
14	08001011200000105002300000000	C 97 7 31	729		En Firmas
15	08001011200000105002400000000	C 97 7 19	730		En Firmas
16	08001011200000105001400000000	C 97 7 123	755		En Firmas
17	08001011200000105003100000000	C 97B 7 60	732		En Firmas
18	08001011200000126002600000000		768		En Firmas
19	08001011200000125007000000000		752		En Firmas
20	08001011200000125003800000000	K 7J 93 100	743		En Firmas
21	08001011200000125004000000000	K 7J 93 118	744		En Firmas
22	08001011200000126002300000000		767		En Firmas
23	08001011200000125006700000000		748		En Firmas
24	08001011200000125000900000000	K 8 93 161	741		En Firmas

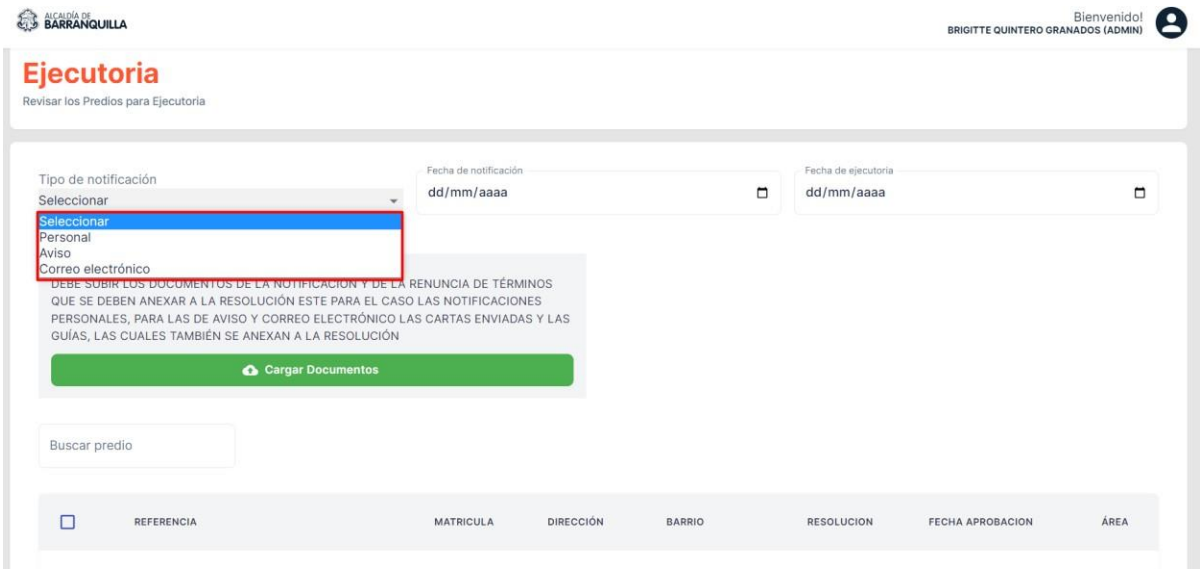
Generar ejecutoria de predios.

Tiene como objetivo generar las ejecutorias para los predios.

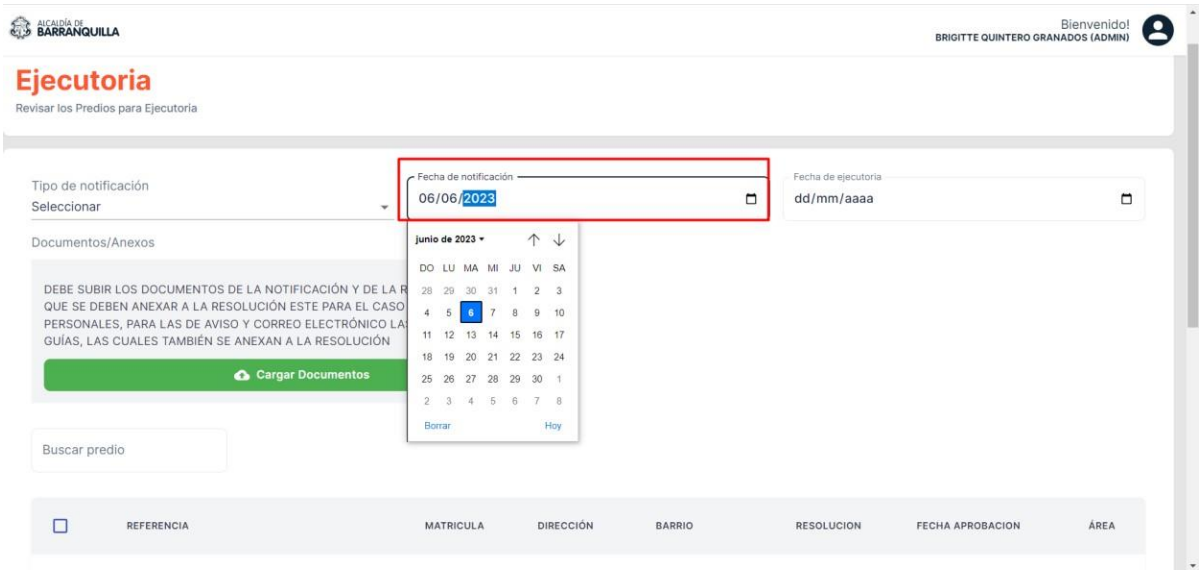


Se visualiza el combo de tipo de notificación.

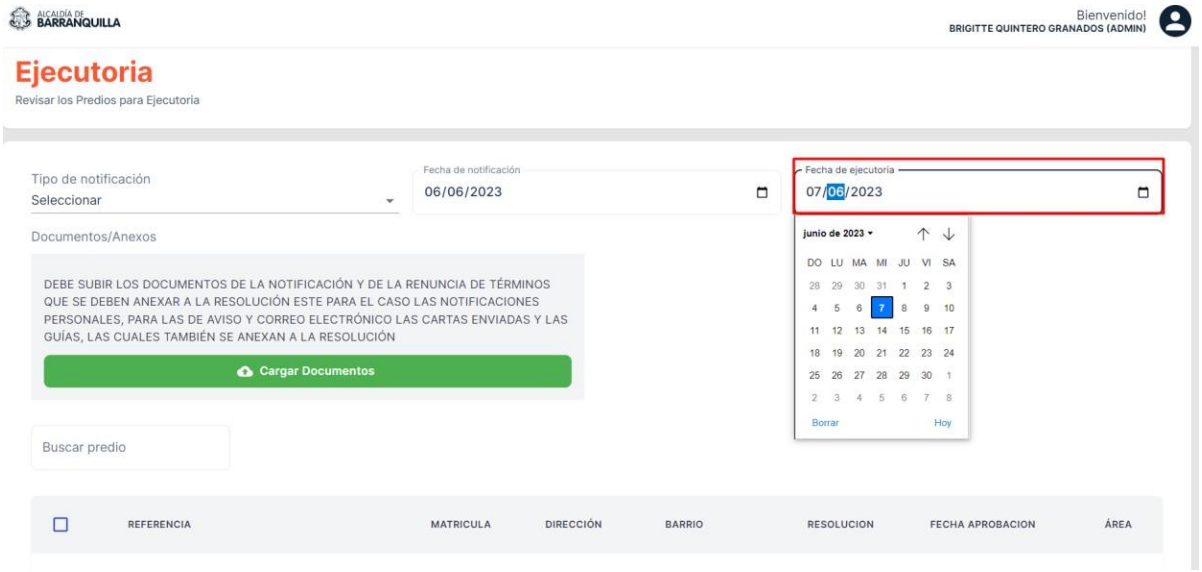
Seleccionar ya sea: personal, aviso o correo electrónico.



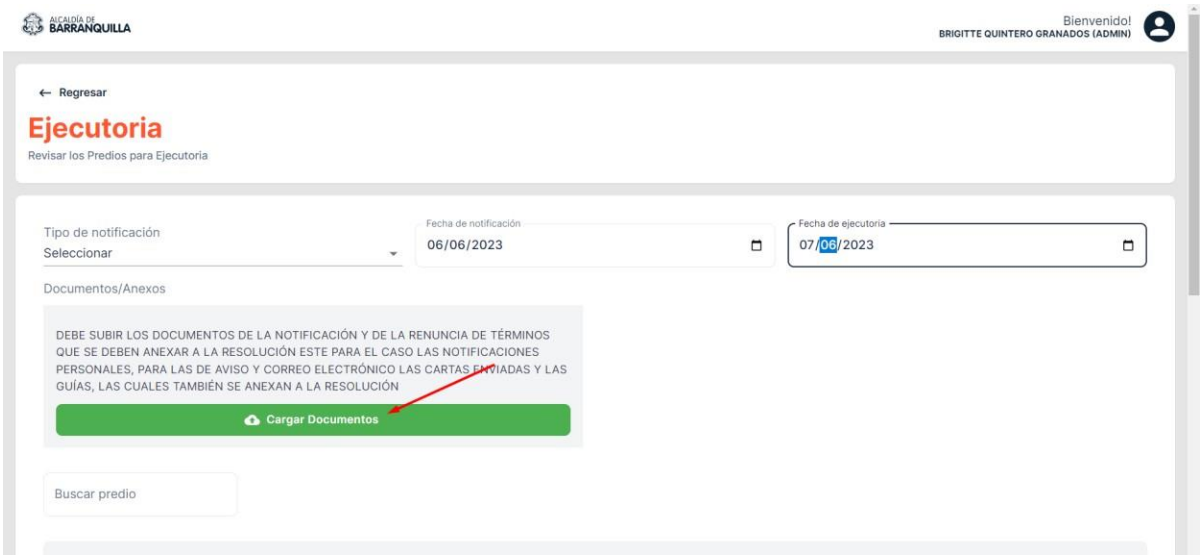
Fecha de notificación.



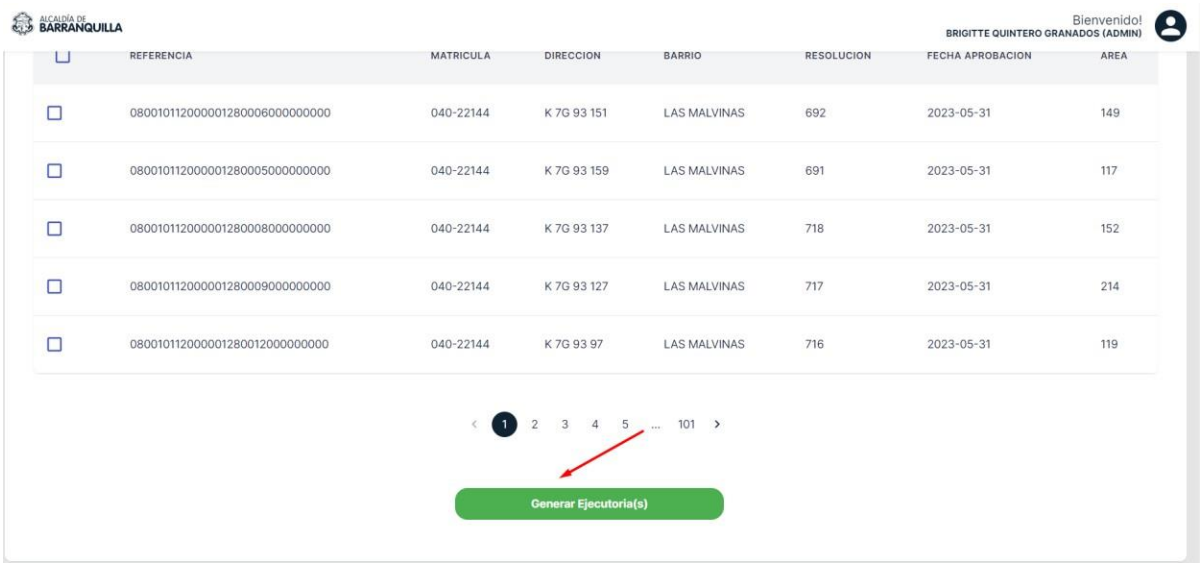
Fecha de ejecutoria.



Se carga los documentos

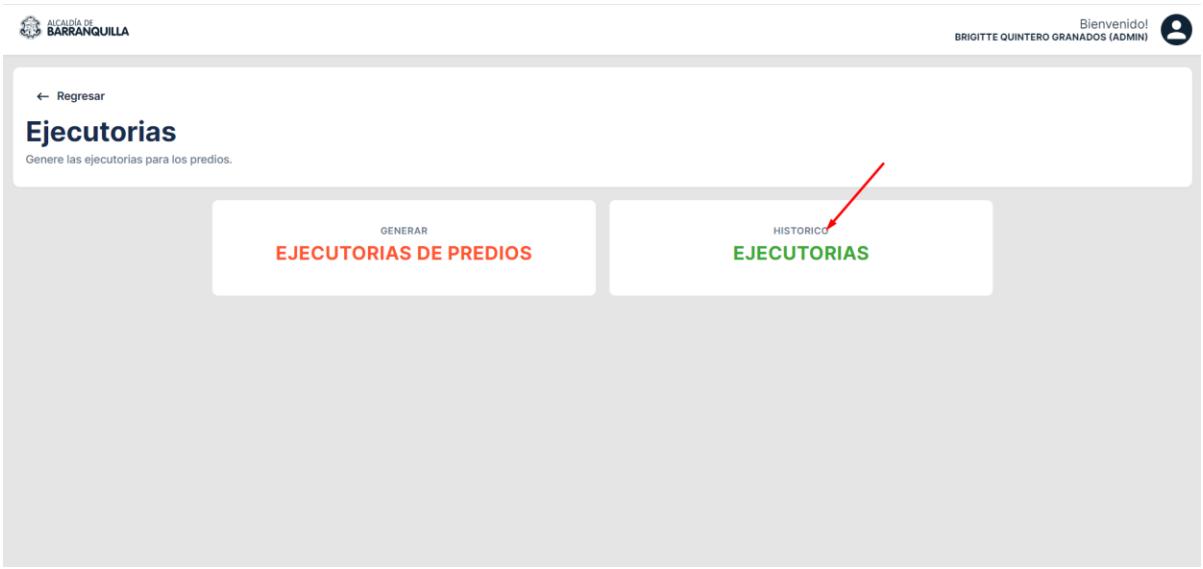


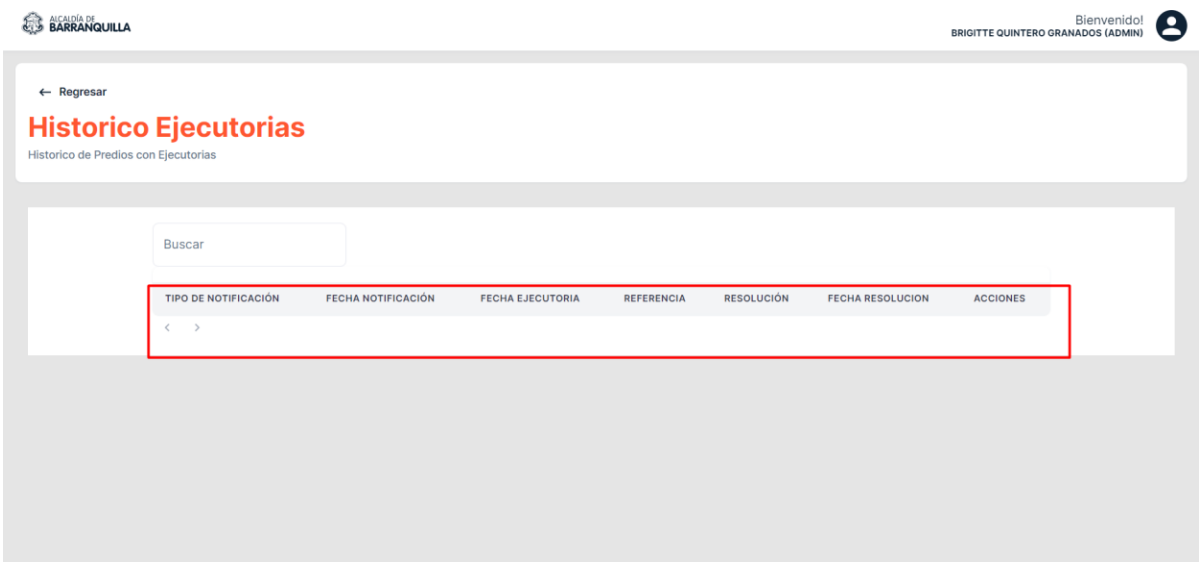
Generar ejecutoria.



Histórico de ejecutorias.

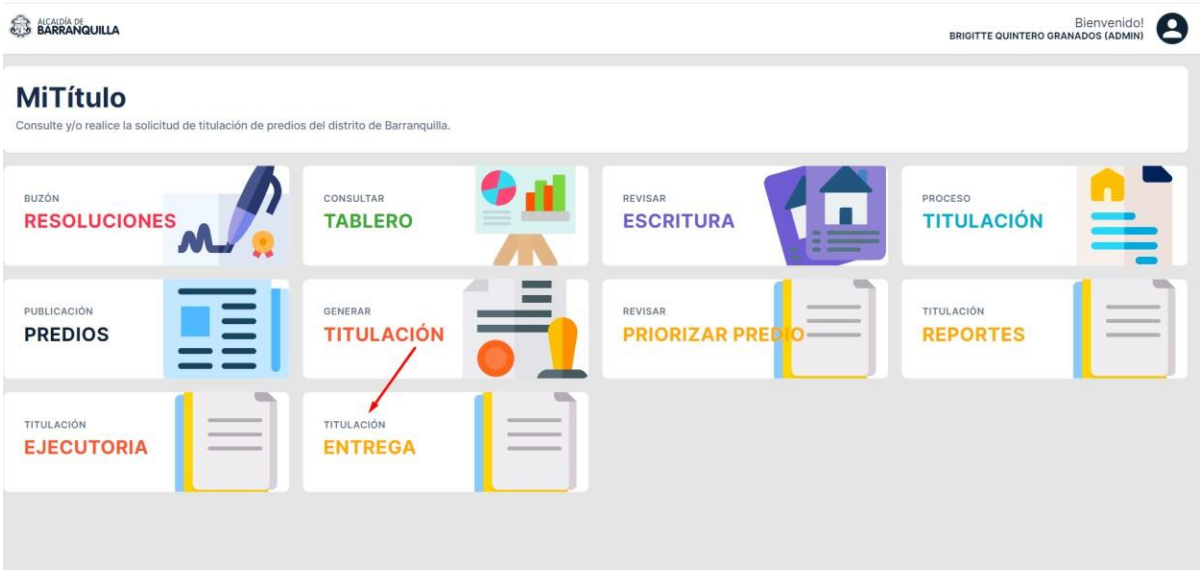
Permite descargar en un solo archivo PDF certificado expedientes, comprobantes de notificación ya sea por medio de aviso, correo y personal.





Entregada de titulación,

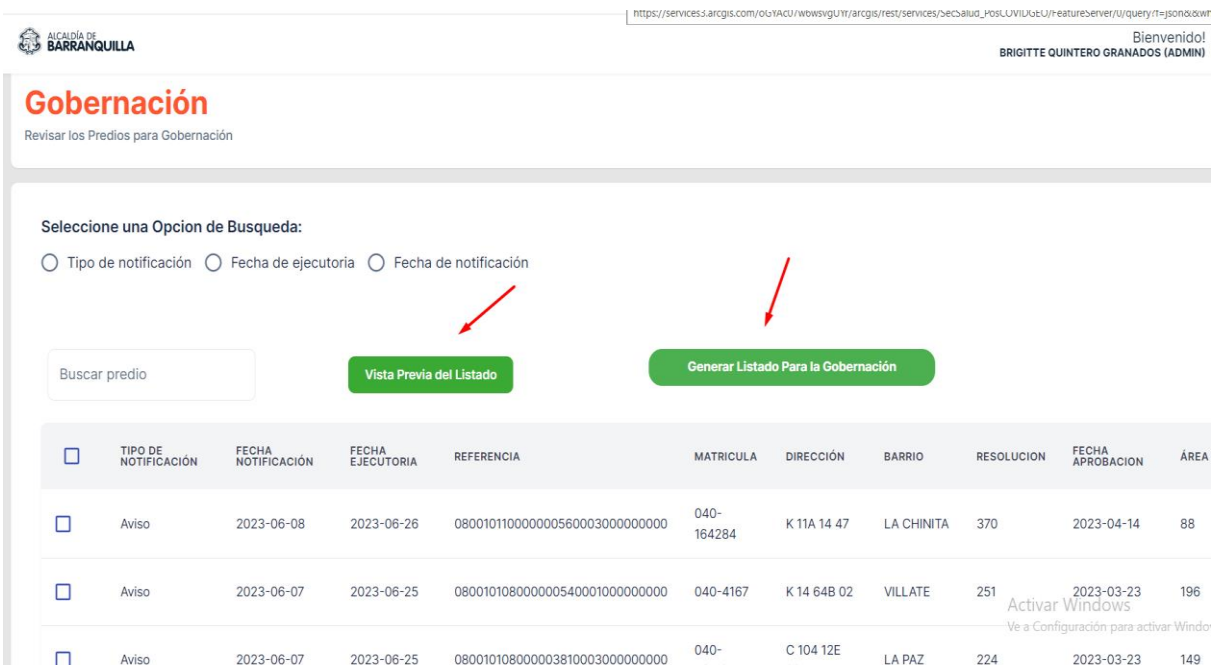
Esta opción permite que los ciudadanos beneficiados, realicen búsqueda de la información del número de mesa, código y el lugar donde será entregado el título del predio.



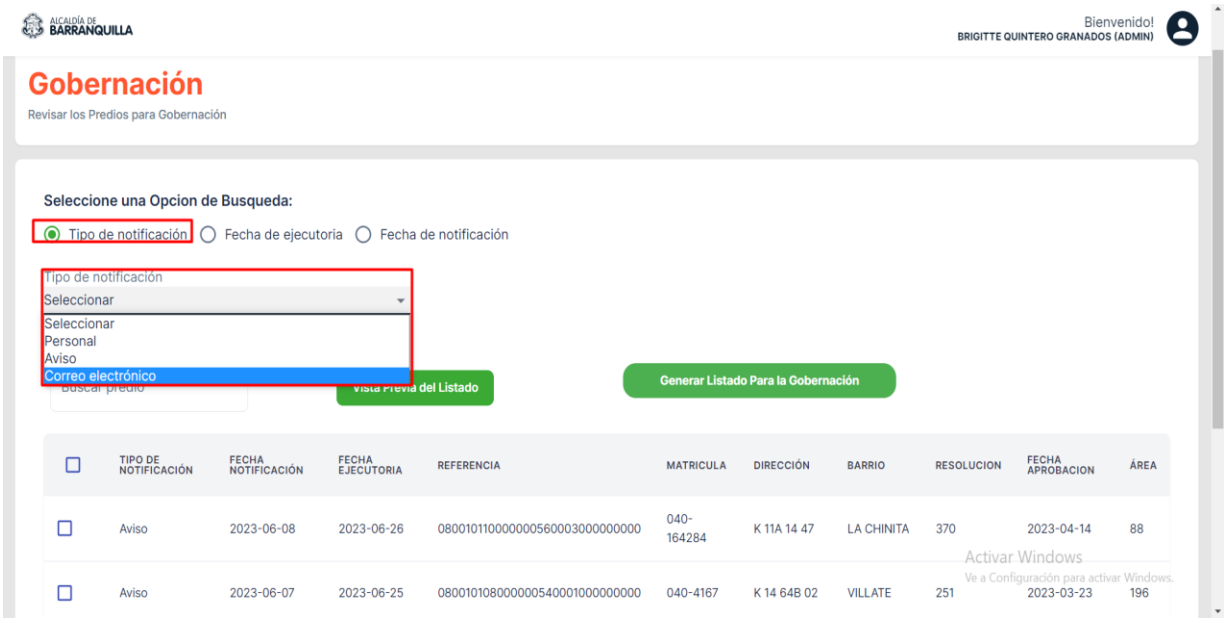
Generar Gobernación de predios.



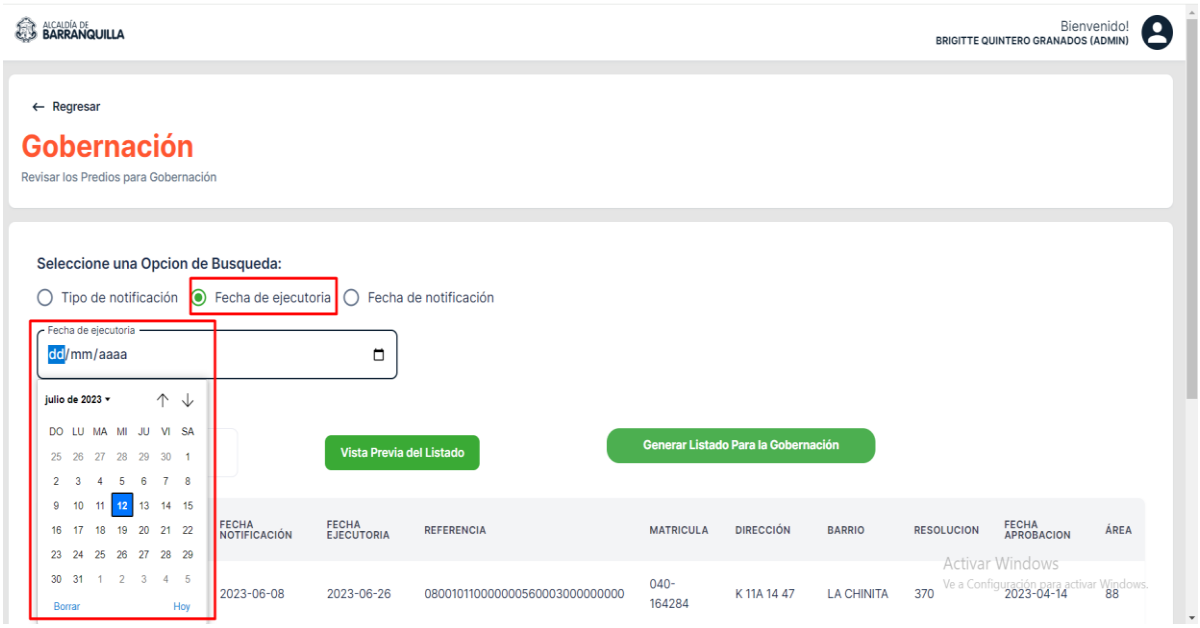
Vista de previa del listado , generar listado para la gobernación.



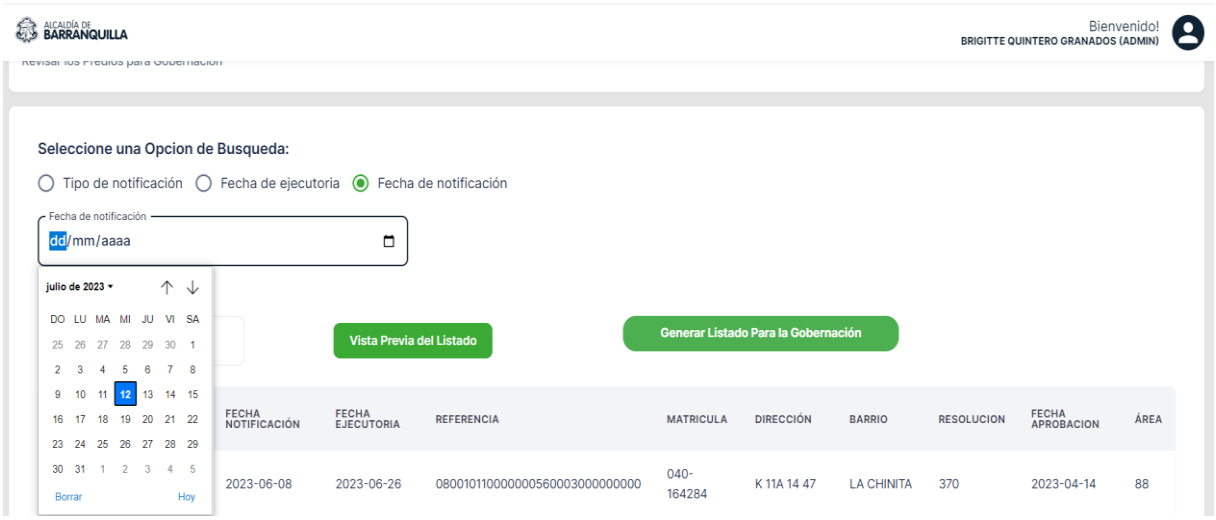
Tipo de notificación.



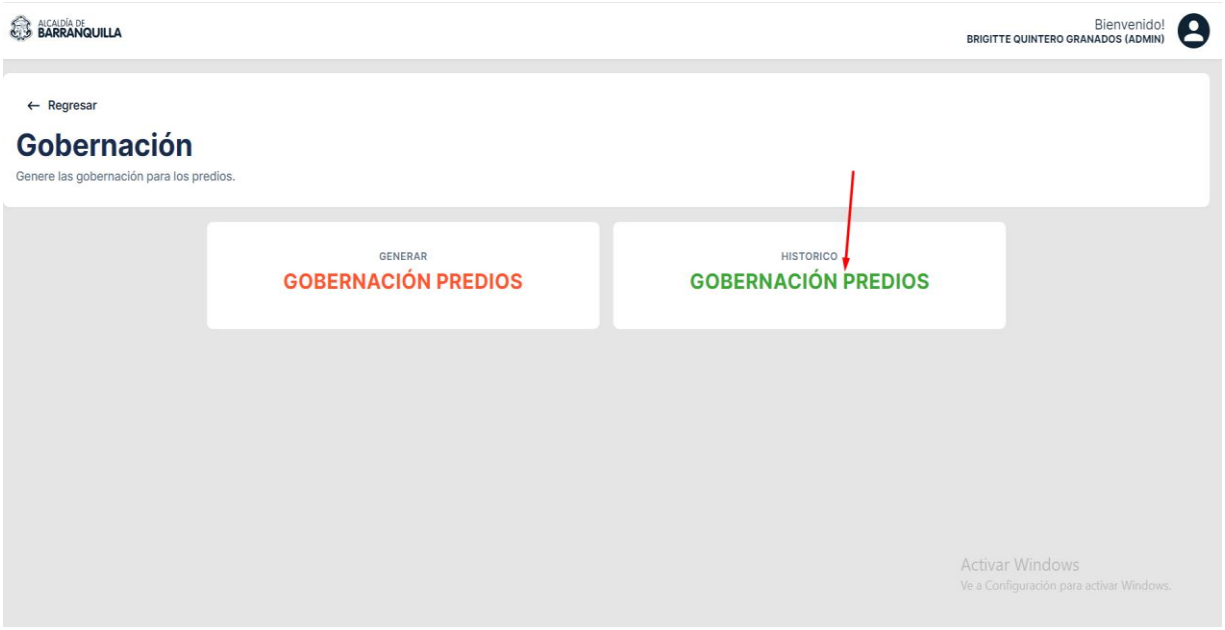
Fecha de ejecutoria.



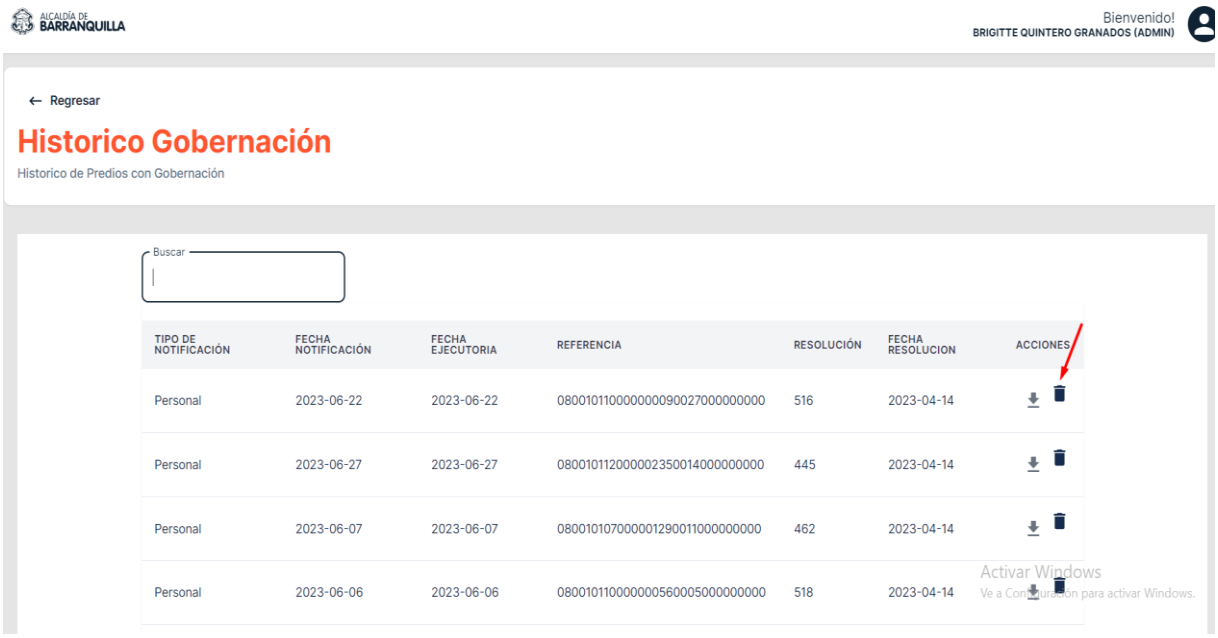
Fecha de notificación.



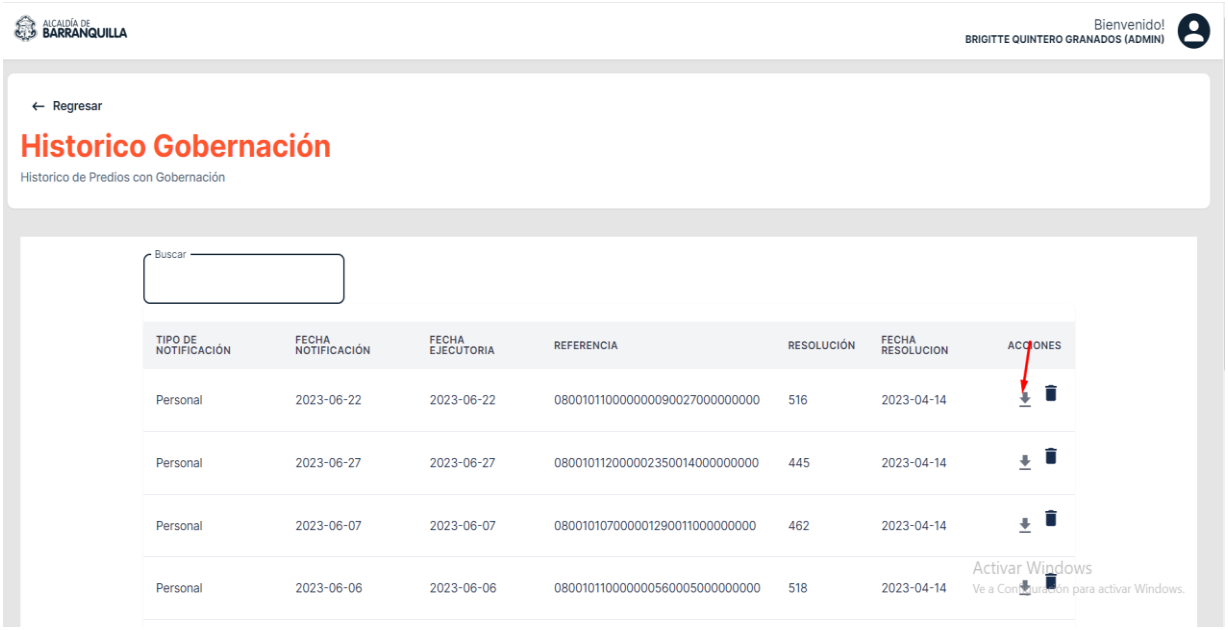
Histórico Gobernación de predios.



Opción permite eliminar.



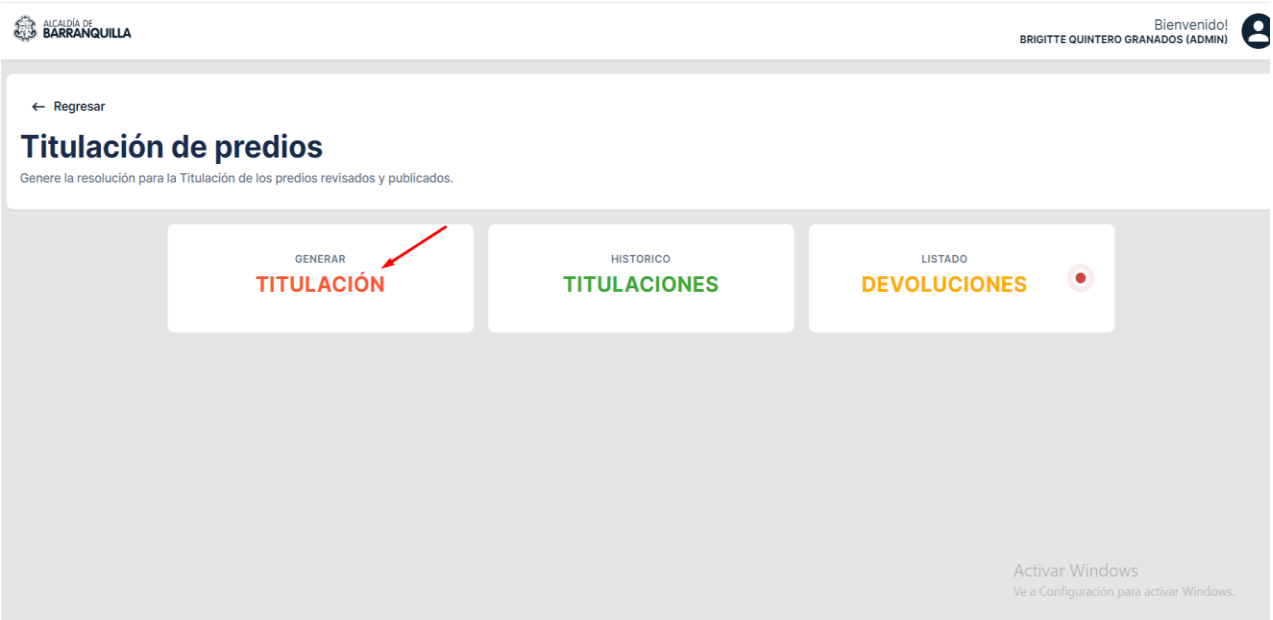
Acción de descargar.



Generar titulación.



Generar titulación.



Asignar por devoluciones.



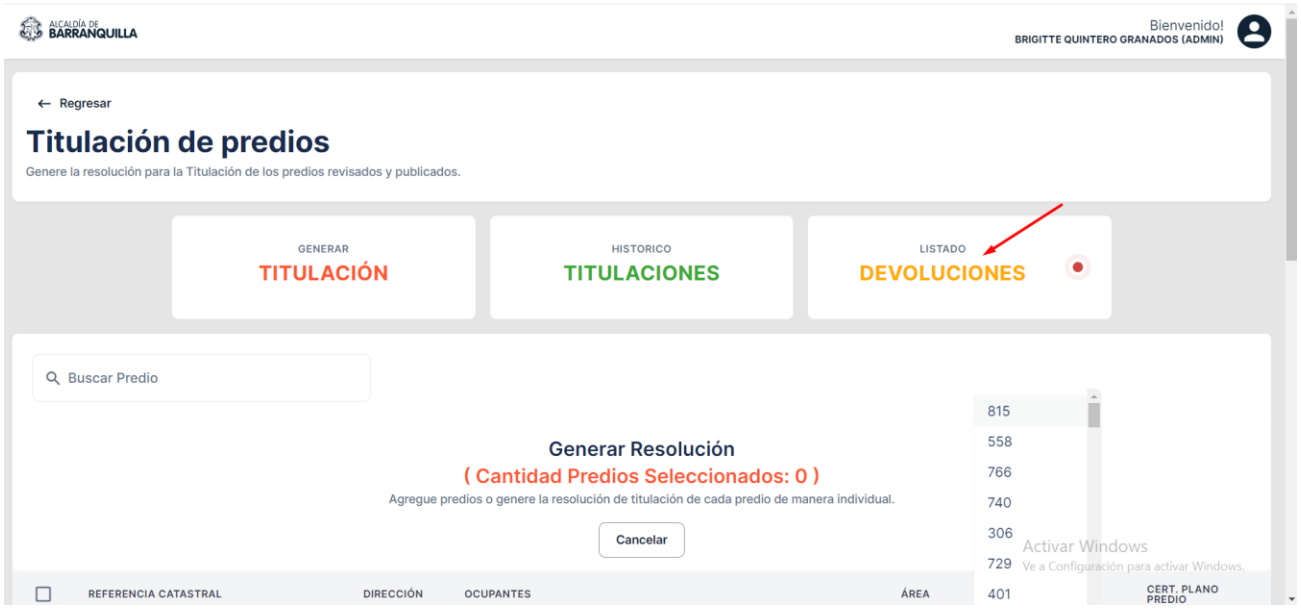
Botón para asignar.



Descargar.



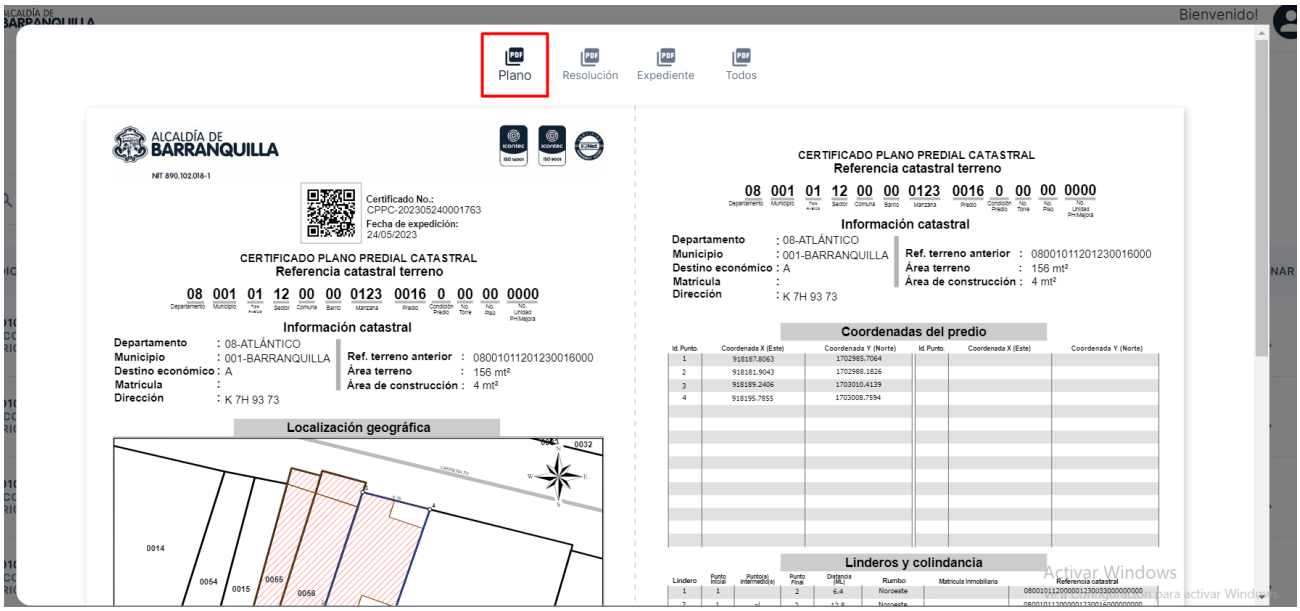
Listado devolución.



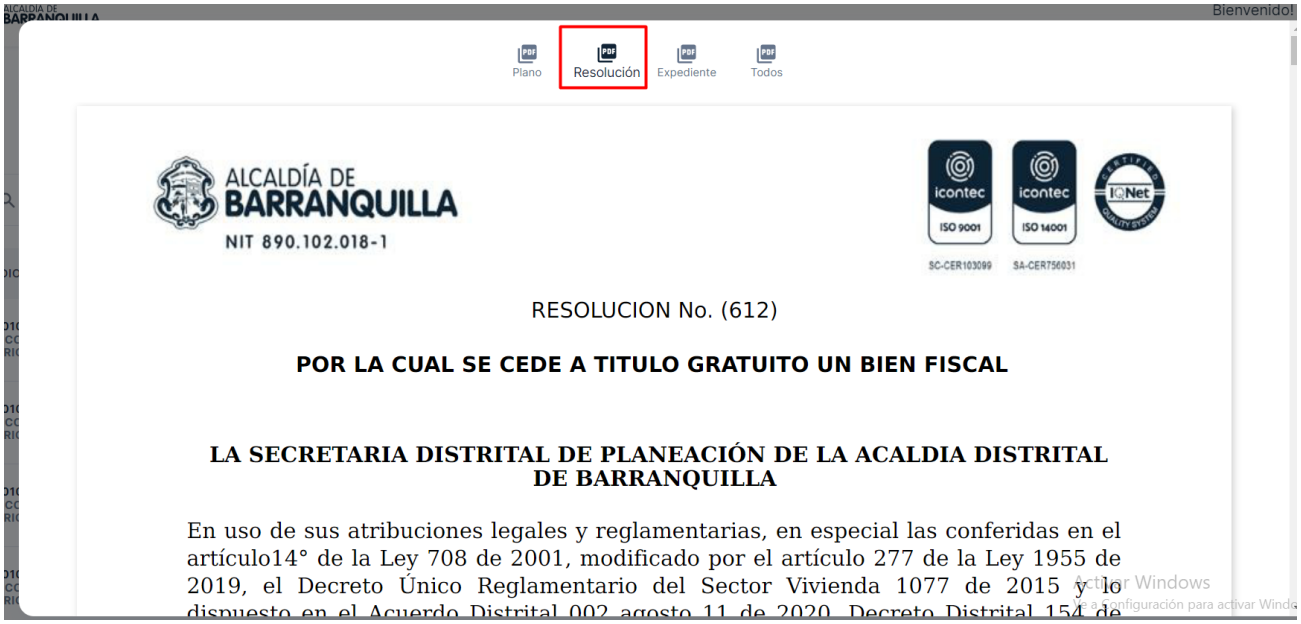
Regenerar resolución.



Plano.



Resolución.



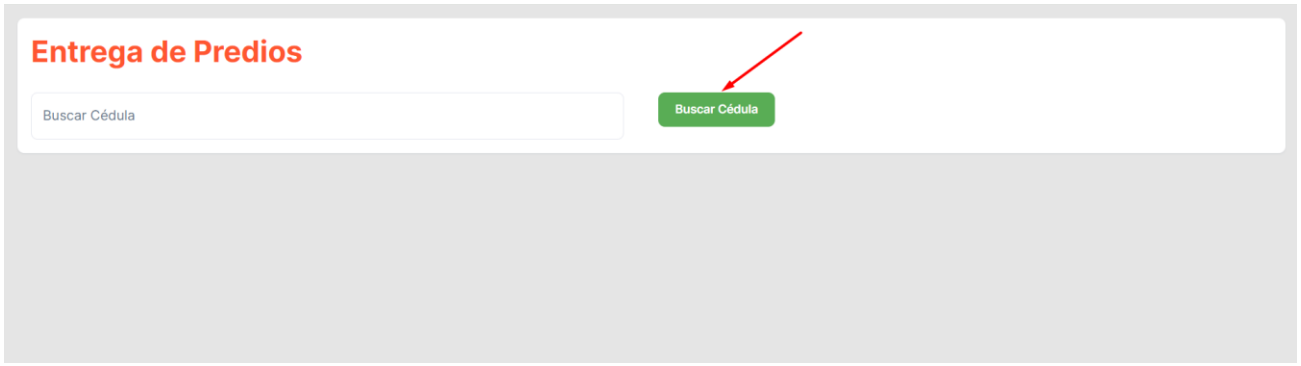
Expediente.



Todos.

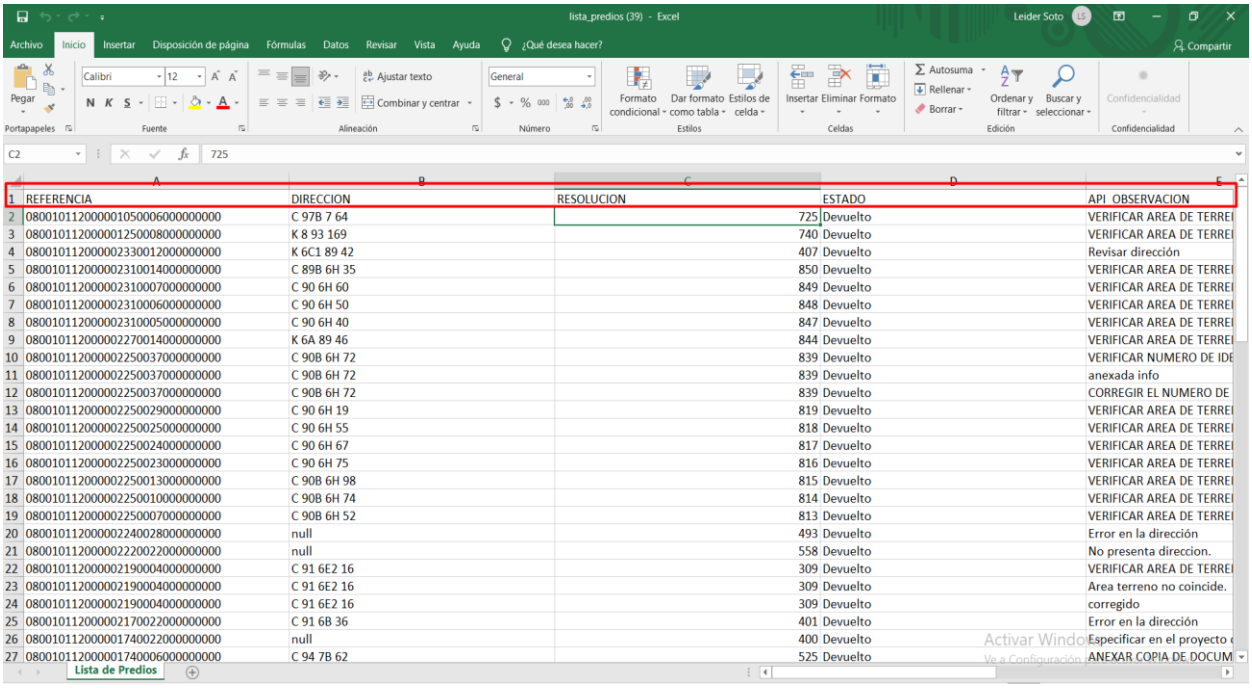


Permite realizar búsquedas numero de cedula para entrega de predios.



Generar reporte resolución devuelta.





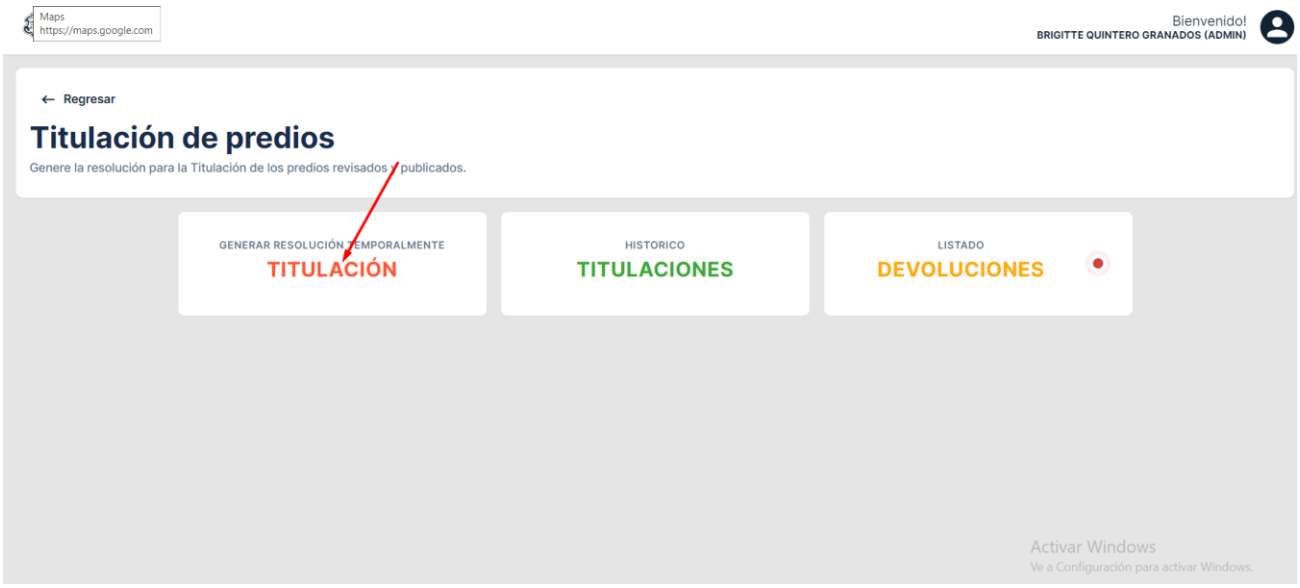
lista_predios (39) - Excel

	REFERENCIA	DIRECCION	RESOLUCION	ESTADO	API_OBSERVACION
2	08001011200000105000600000000	C 97B 7 64	725	Devuelto	VERIFICAR AREA DE TERREI
3	08001011200000112500080000000	K 8 93 169	740	Devuelto	VERIFICAR AREA DE TERREI
4	08001011200000233001200000000	K 6C1 89 42	407	Devuelto	Revisar dirección
5	08001011200000231001400000000	C 89B 6H 35	850	Devuelto	VERIFICAR AREA DE TERREI
6	08001011200000231000700000000	C 90 6H 60	849	Devuelto	VERIFICAR AREA DE TERREI
7	08001011200000231000600000000	C 90 6H 50	848	Devuelto	VERIFICAR AREA DE TERREI
8	08001011200000231000500000000	C 90 6H 40	847	Devuelto	VERIFICAR AREA DE TERREI
9	08001011200000227001400000000	K 6A 89 46	844	Devuelto	VERIFICAR AREA DE TERREI
10	08001011200000225003700000000	C 90B 6H 72	839	Devuelto	VERIFICAR NUMERO DE IDE
11	08001011200000225003700000000	C 90B 6H 72	839	Devuelto	anexada info
12	08001011200000225003700000000	C 90B 6H 72	839	Devuelto	CORREGIR EL NUMERO DE
13	08001011200000225002900000000	C 90 6H 19	819	Devuelto	VERIFICAR AREA DE TERREI
14	08001011200000225002500000000	C 90 6H 55	818	Devuelto	VERIFICAR AREA DE TERREI
15	08001011200000225002400000000	C 90 6H 67	817	Devuelto	VERIFICAR AREA DE TERREI
16	08001011200000225002300000000	C 90 6H 75	816	Devuelto	VERIFICAR AREA DE TERREI
17	08001011200000225001300000000	C 90B 6H 98	815	Devuelto	VERIFICAR AREA DE TERREI
18	08001011200000225001000000000	C 90B 6H 74	814	Devuelto	VERIFICAR AREA DE TERREI
19	08001011200000225000700000000	C 90B 6H 52	813	Devuelto	VERIFICAR AREA DE TERREI
20	08001011200000224002800000000	null	493	Devuelto	Error en la dirección
21	08001011200000222002200000000	null	558	Devuelto	No presenta direccion.
22	08001011200000219000400000000	C 91 6E2 16	309	Devuelto	VERIFICAR AREA DE TERREI
23	08001011200000219000400000000	C 91 6E2 16	309	Devuelto	Area terreno no coincide.
24	08001011200000219000400000000	C 91 6E2 16	309	Devuelto	corregido
25	08001011200000217002200000000	C 91 6B 36	401	Devuelto	Error en la dirección
26	08001011200000174002200000000	null	400	Devuelto	Especificar en el proyecto
27	08001011200000174000600000000	C 94 7B 62	525	Devuelto	ANEXAR COPIA DE DOCUM

Generar titulación.



Permite genera titulación temporalmente



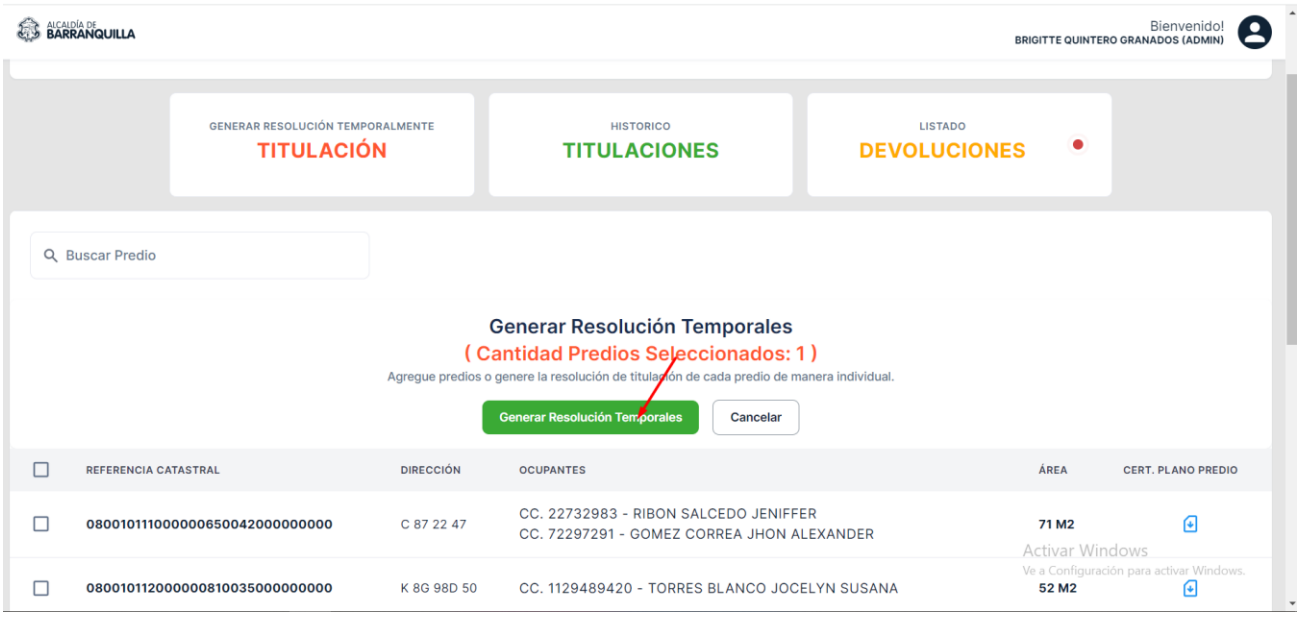
Se visualiza la información de referencia catastral, dirección, ocupante, área y CERT. Plano predio



Descargar Excel.



Generar resolución temporalmente.



Consecutivo resolución de predio.



Aquí se hace la asignación de consecutivo resolución de manera definitiva.



Titulación Ejecutoria.



Generar ejecutoria de predios.



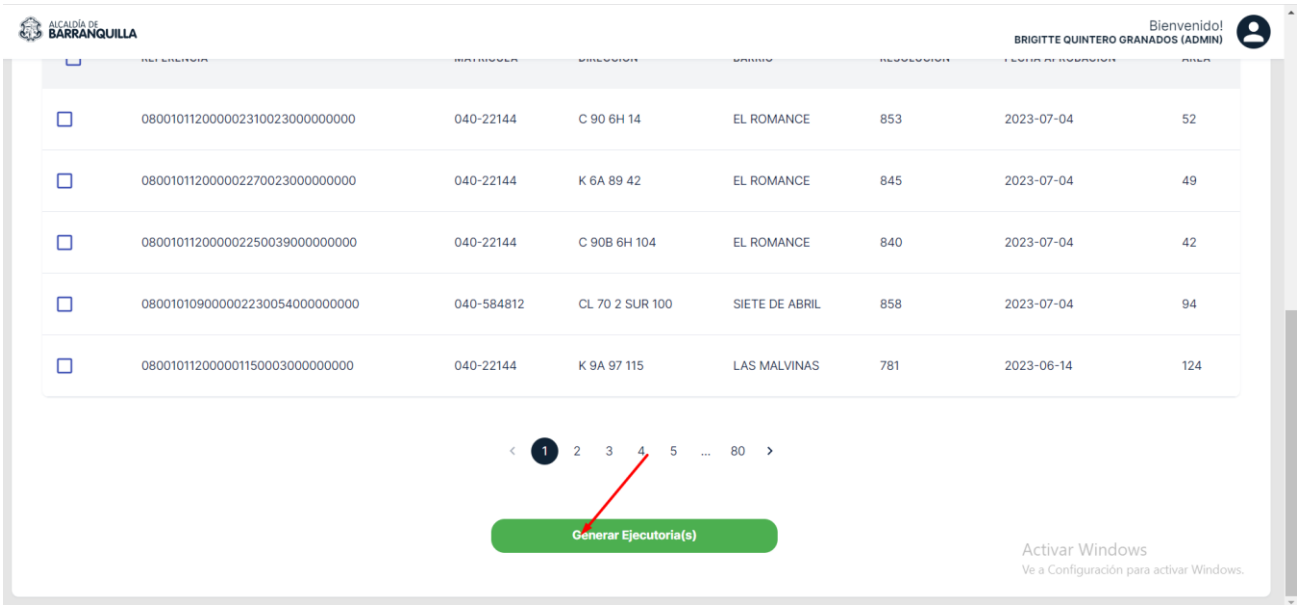
Tipo de notificación.



Cargar documentos.



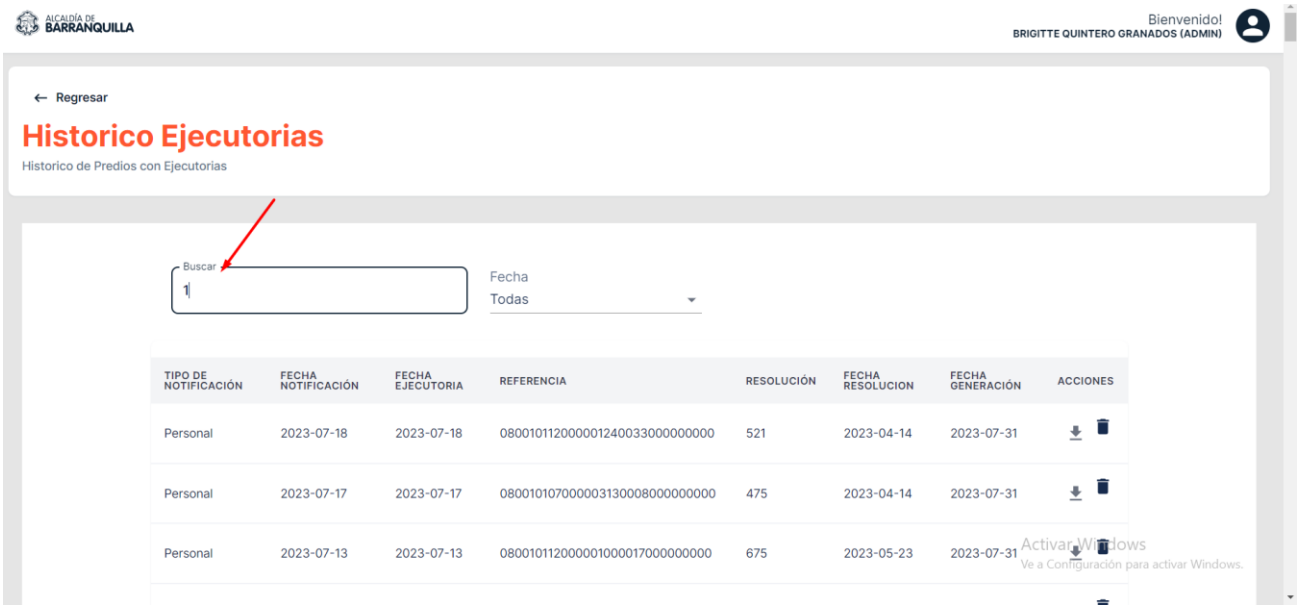
Generar ejecutoria.



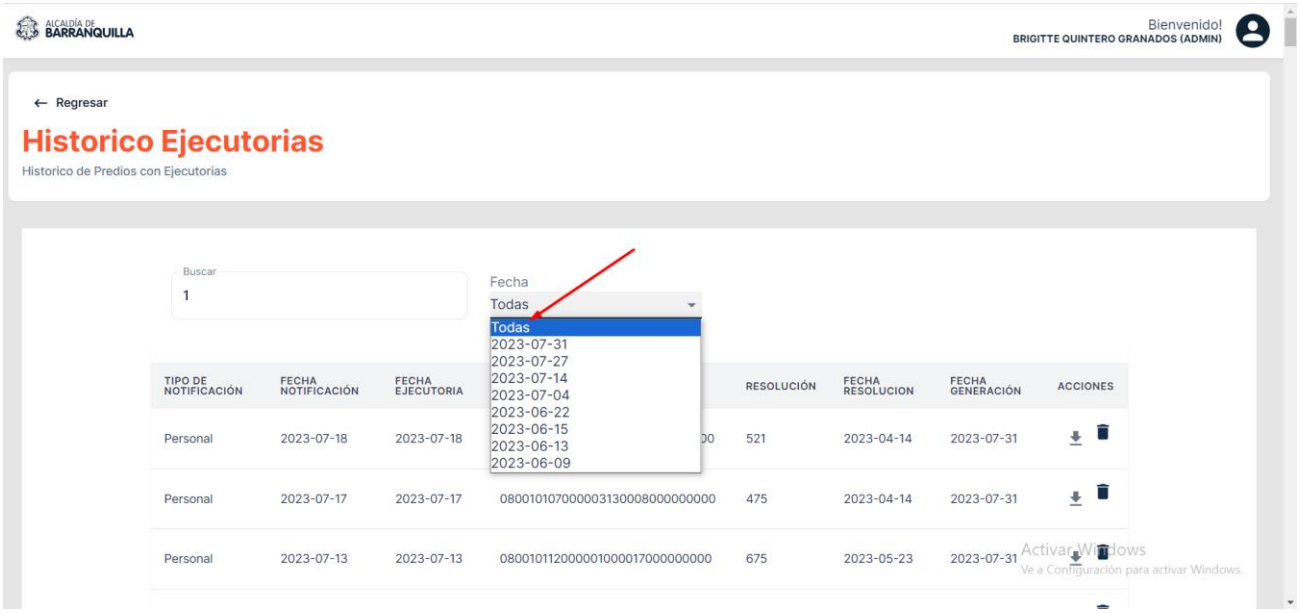
Histórico ejecutoria.



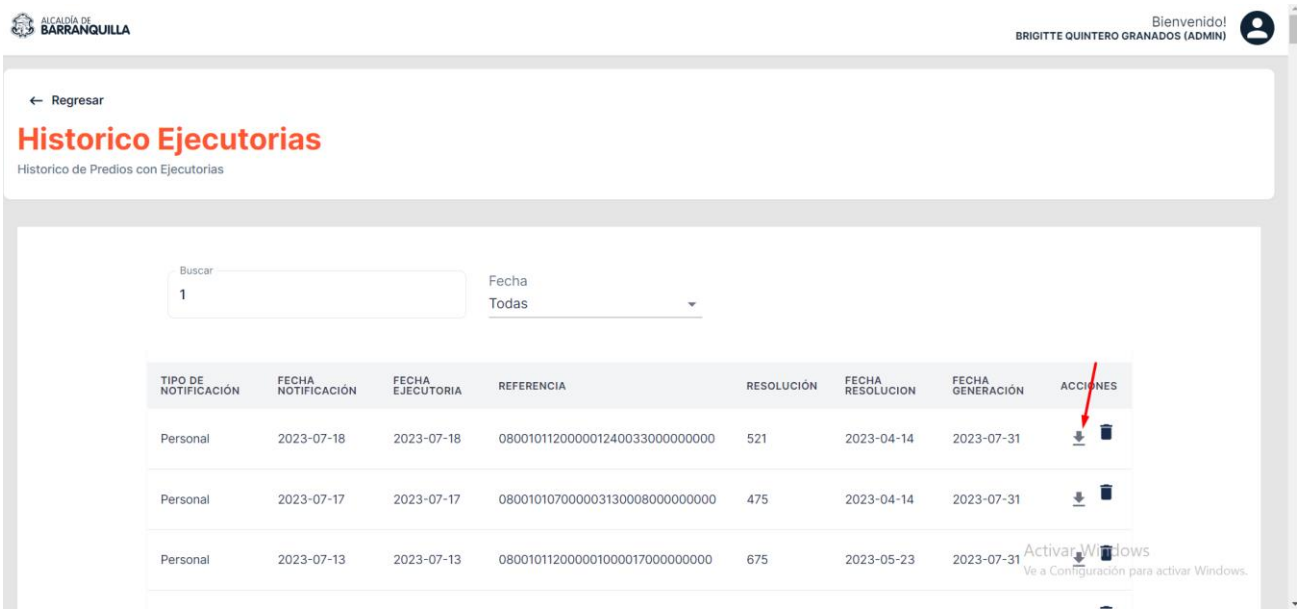
Se observa el buscador.



Seleccionar las fechas.



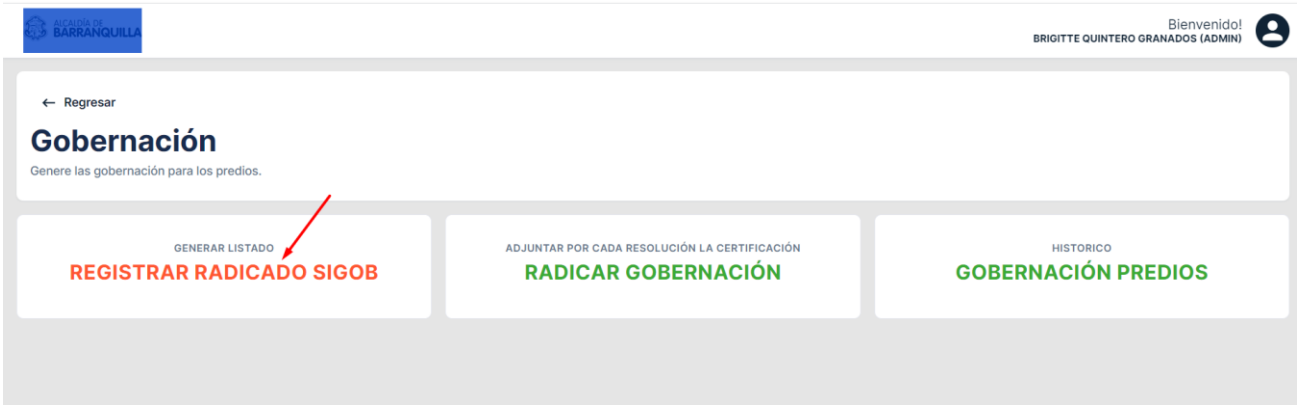
Descargar accines.

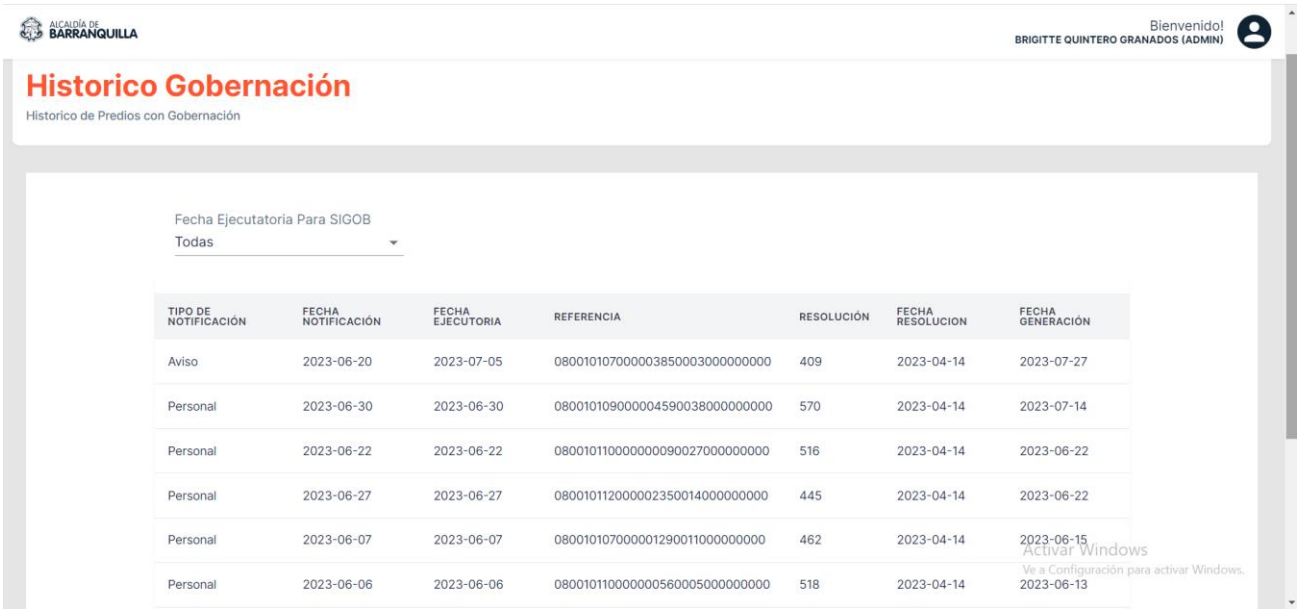


Titulación gobernación.

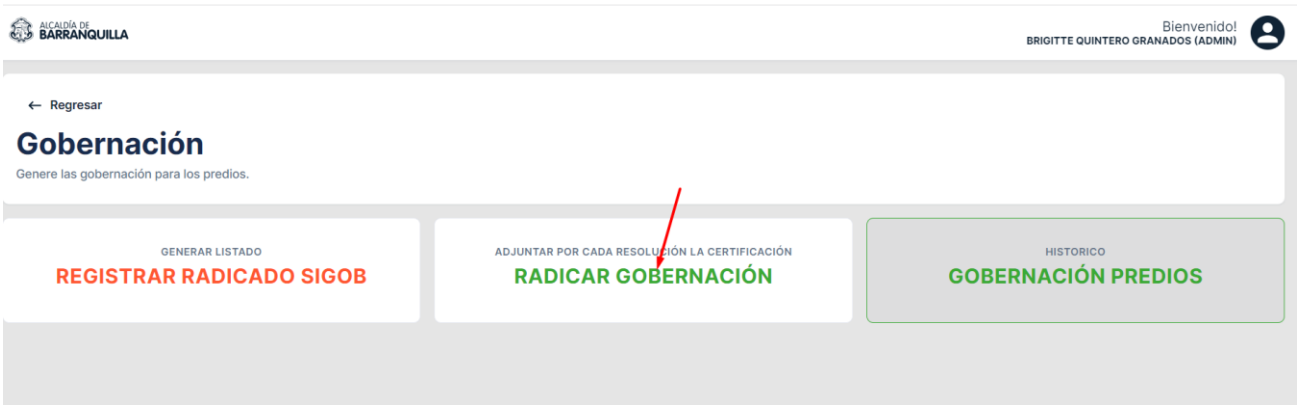


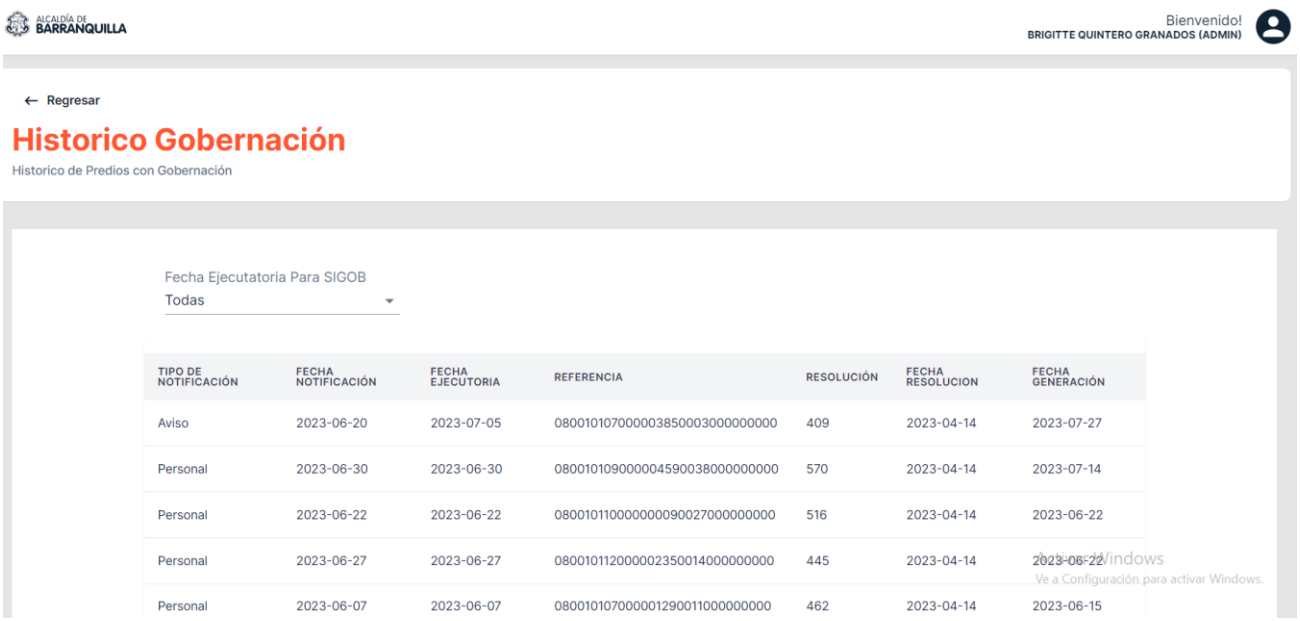
Registrar radicación SIGOB.



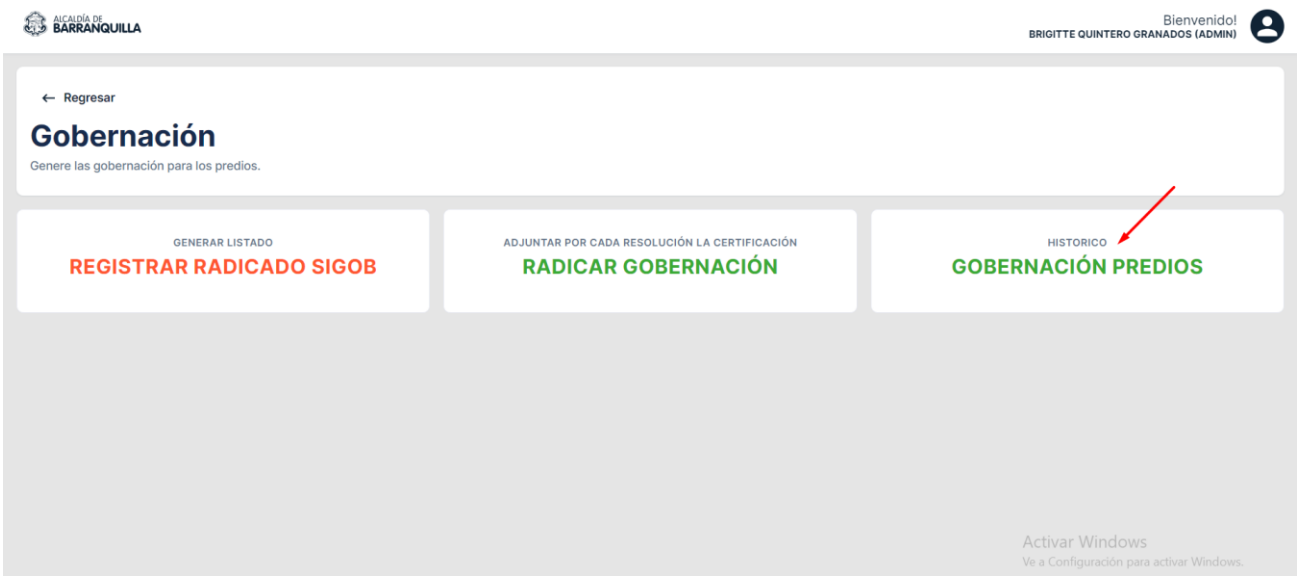


Radicar gobernación.





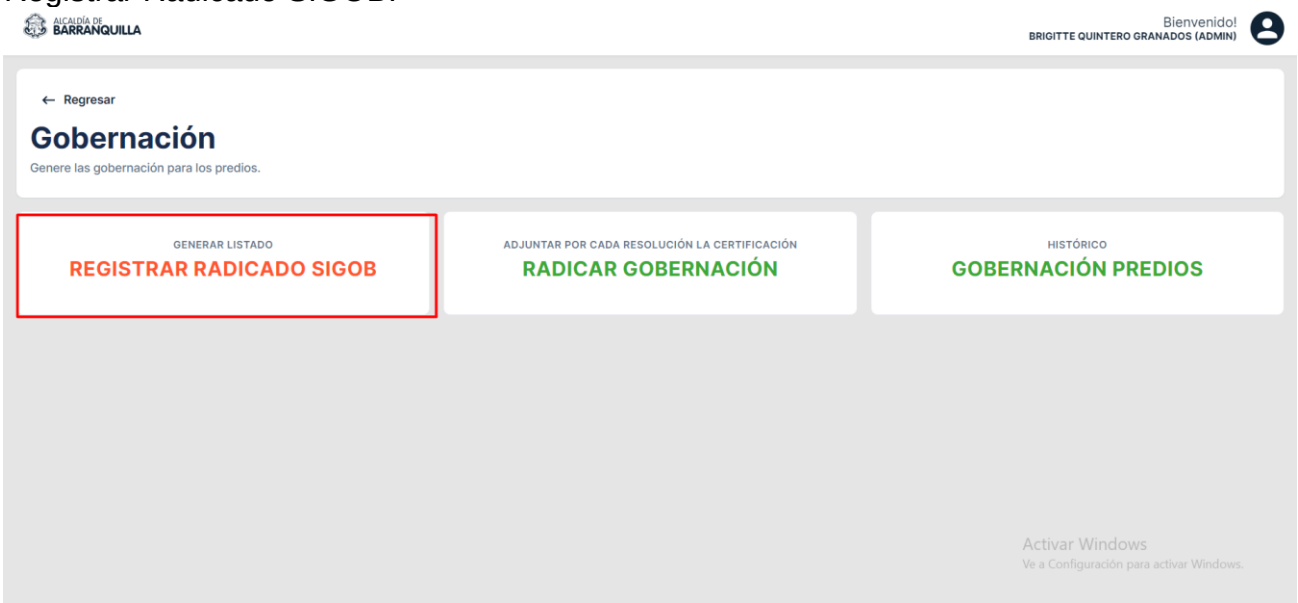
Gobernación Predios.



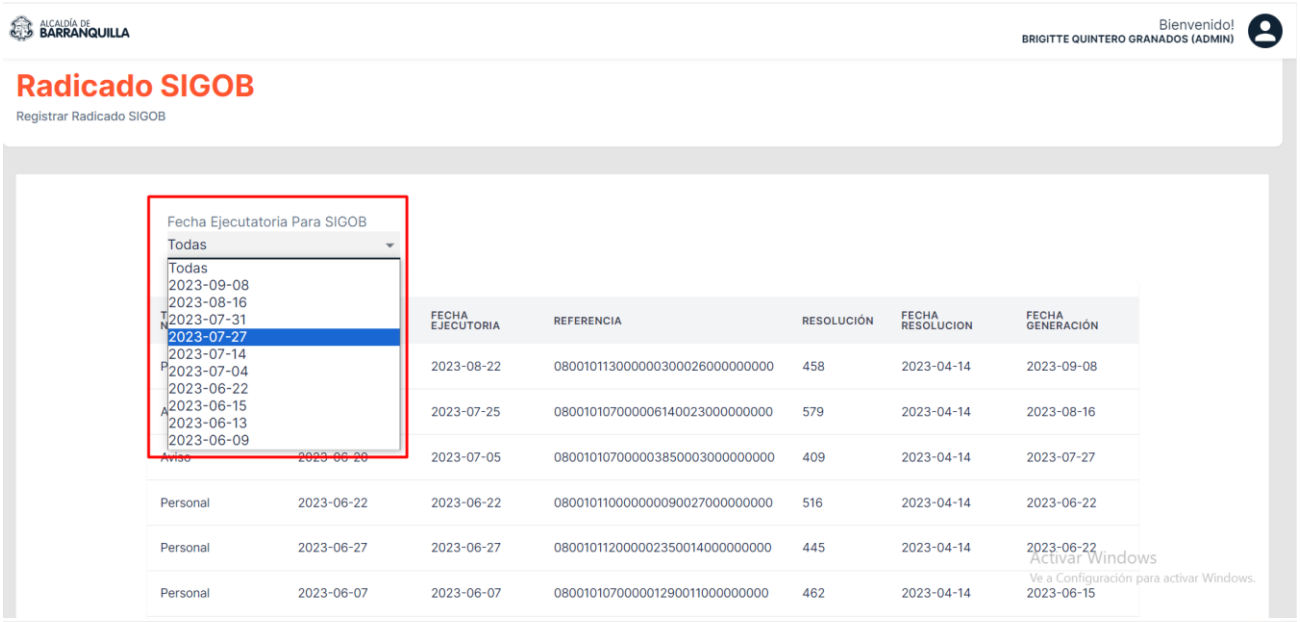
Módulo de Gobernación, el cual ya se encuentra a en productivo.



Registrar Radicado SIGOB.

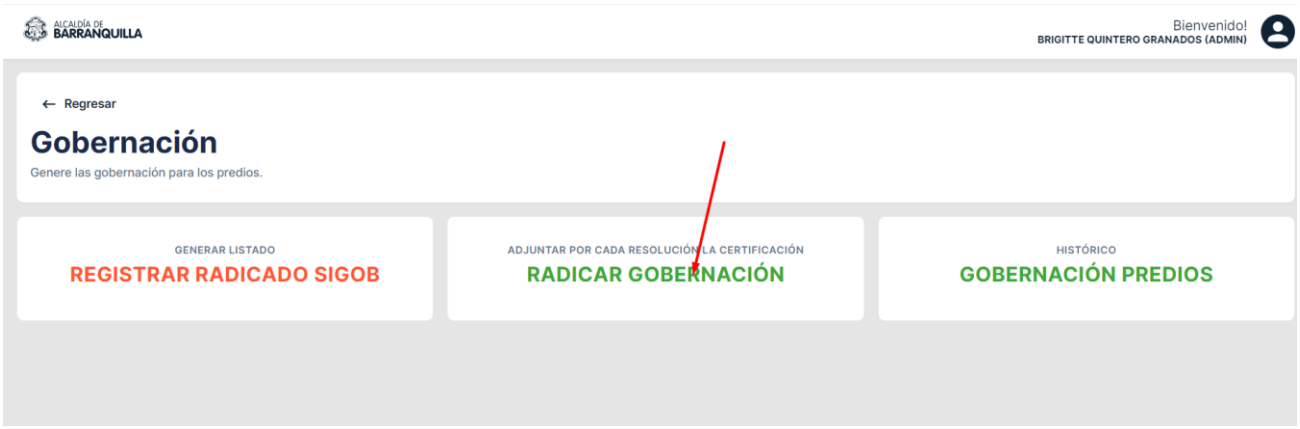


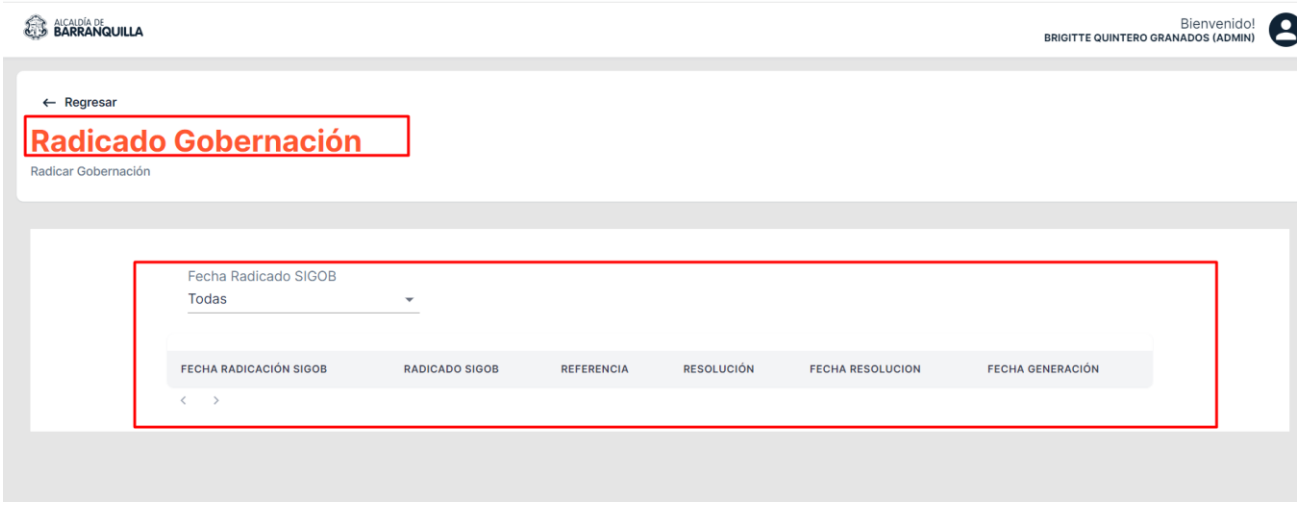
Seleccionar fecha ejecutoria para SIGOB.



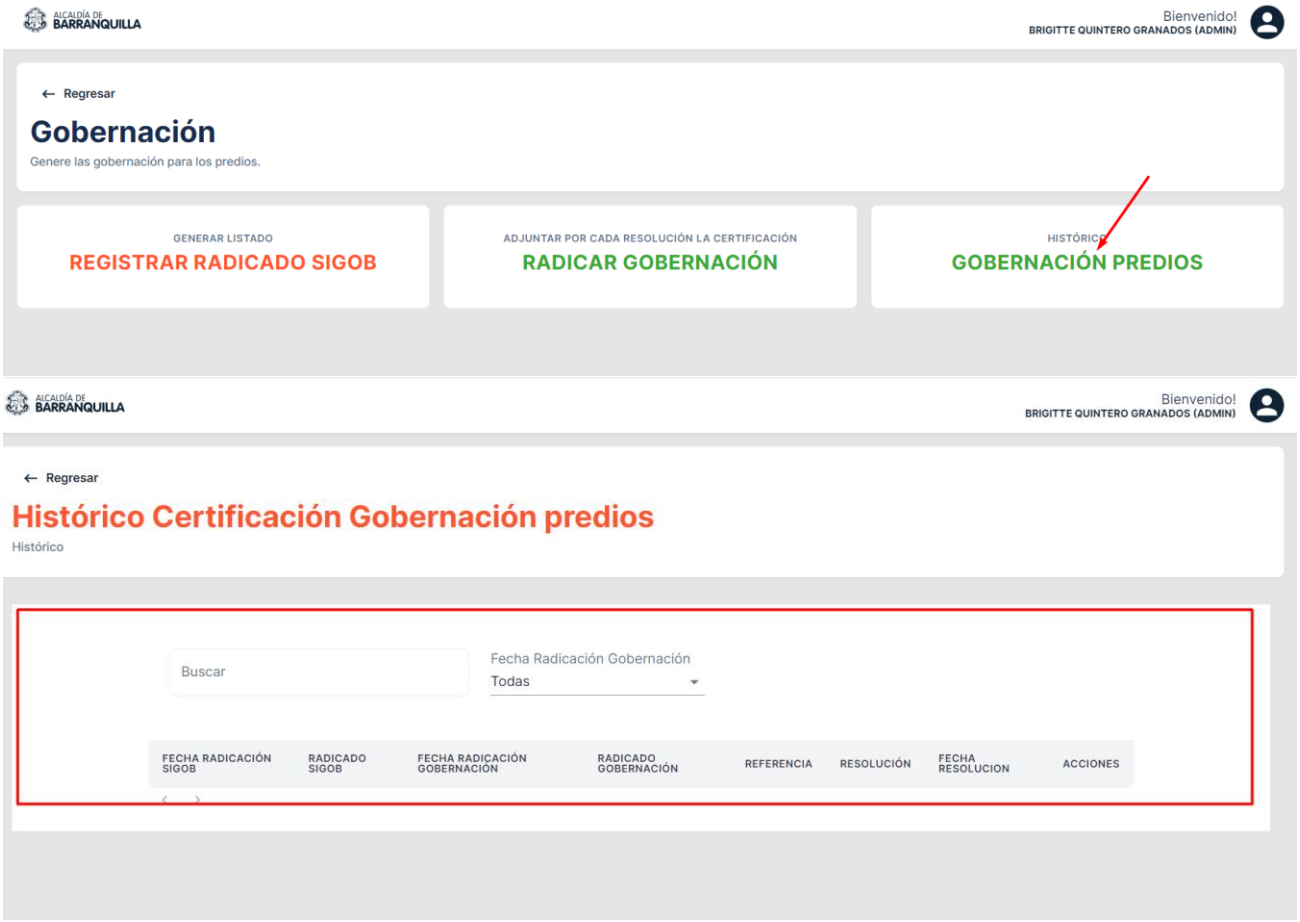


Radicar Gobernación.





Gobernación Predio.

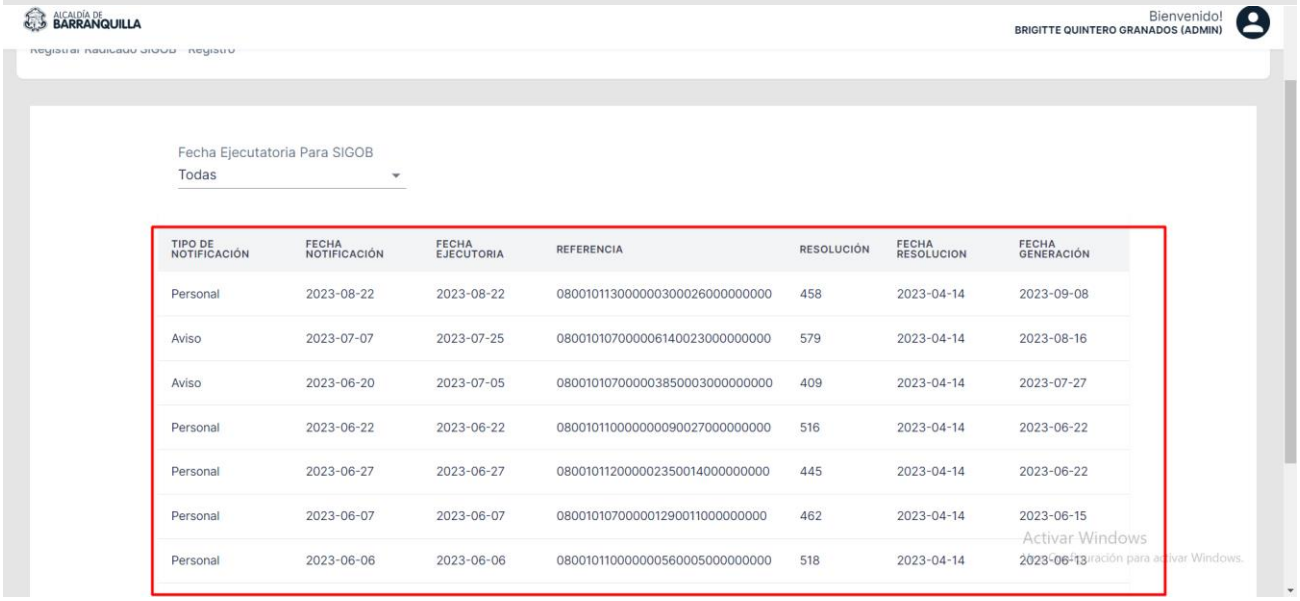


Avance de Titulación de Predio, el cual se encuentra con un porcentaje de avances de un 40%

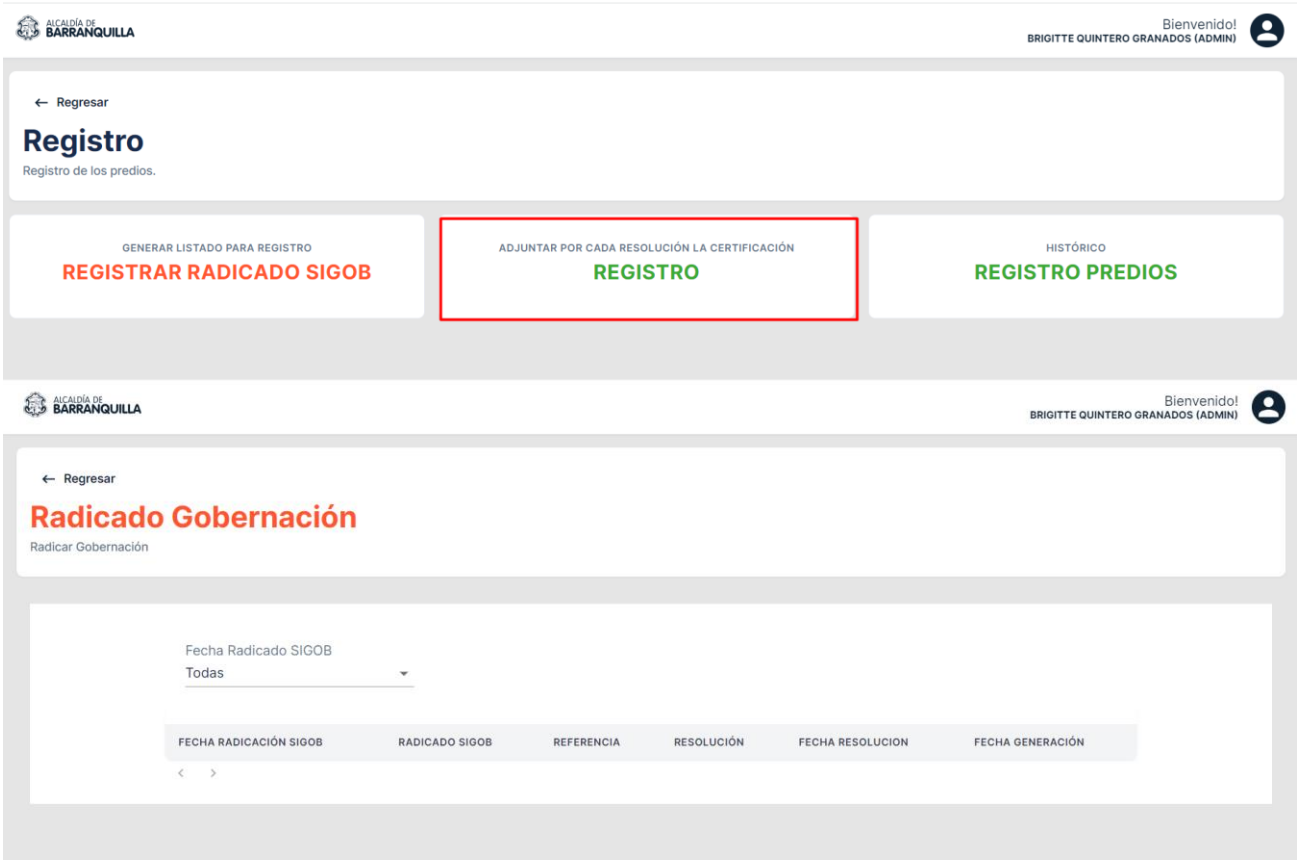


Registrar Radicado SIGOB.

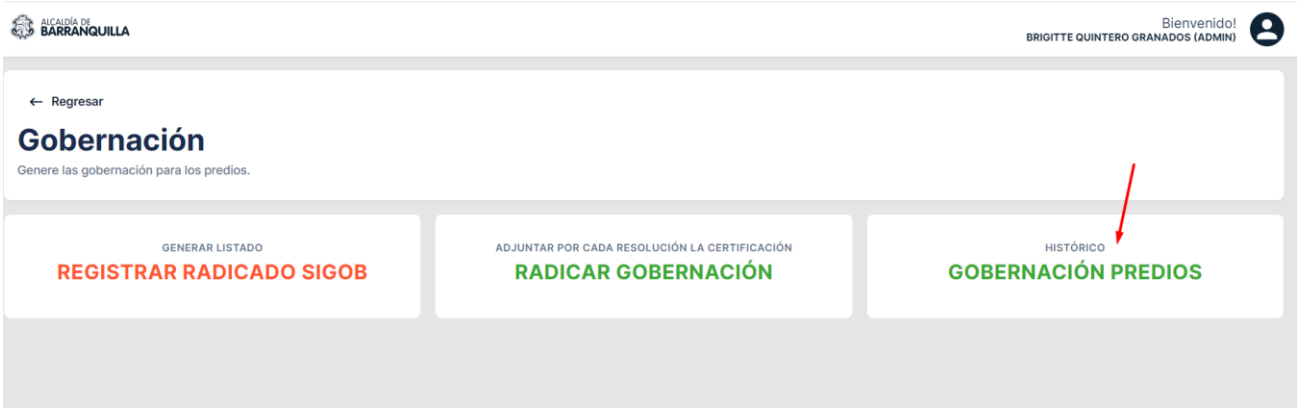


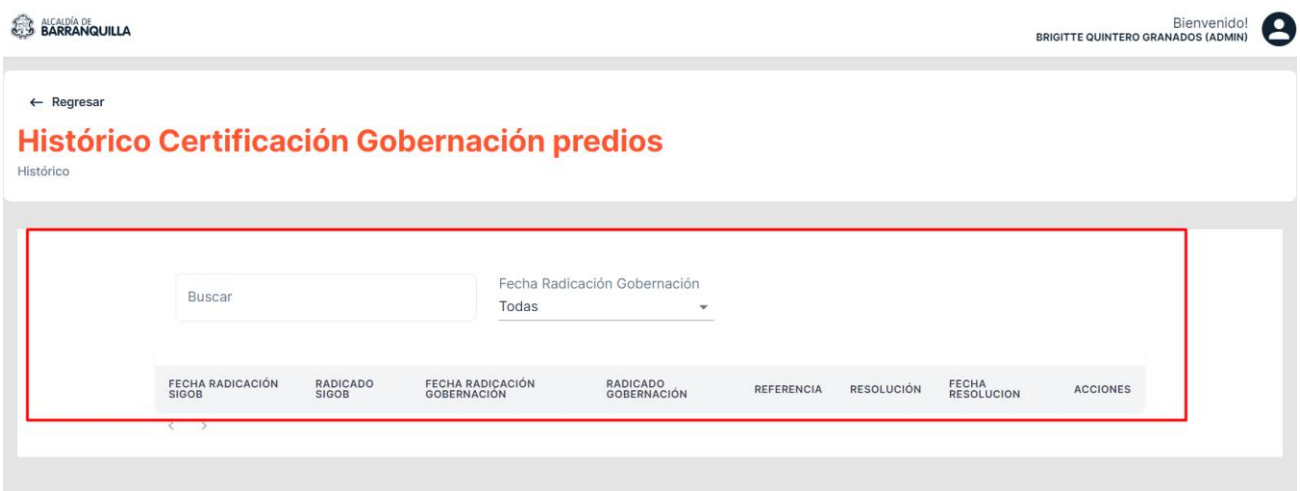


Registro.

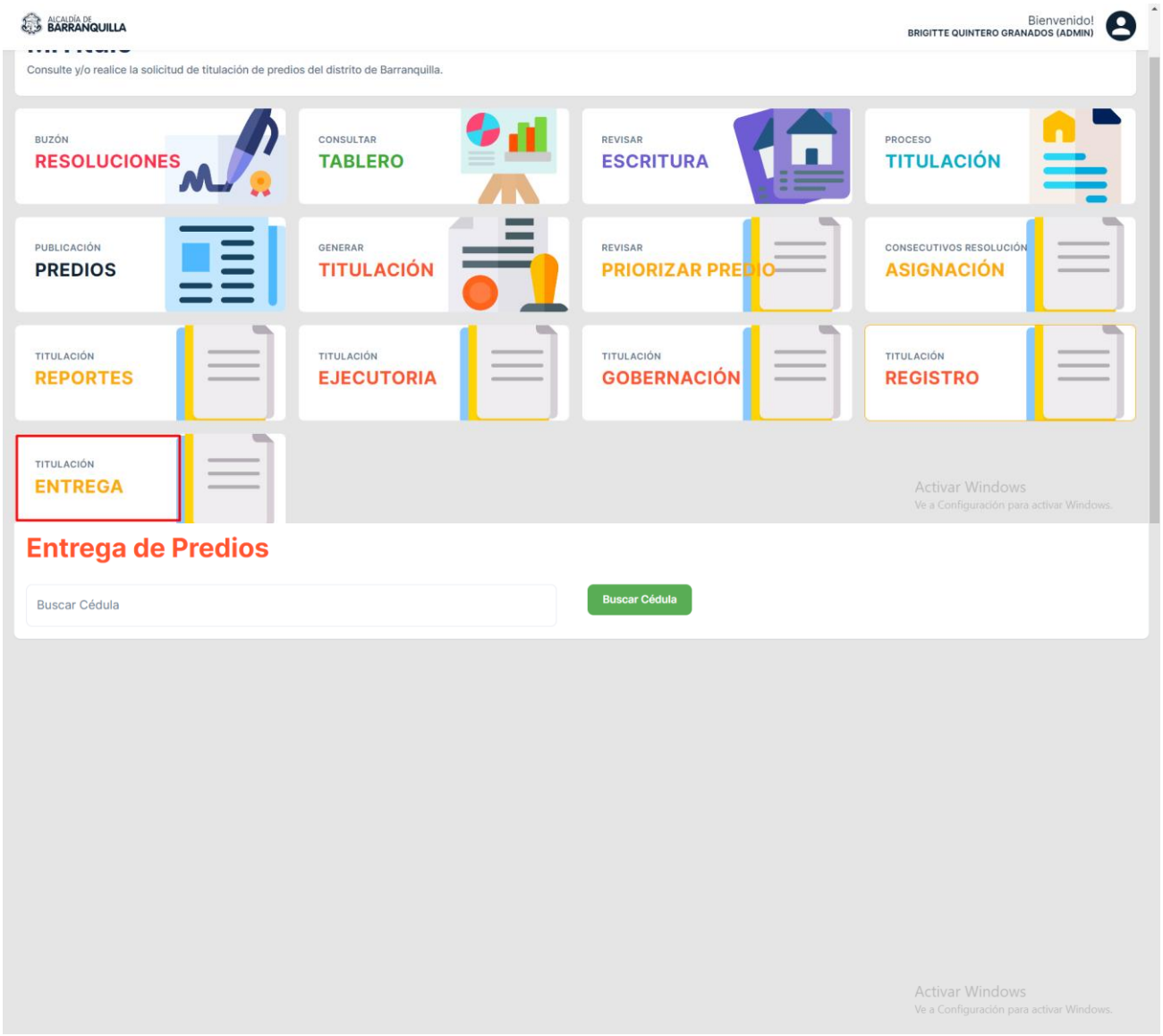


Gobernación de predios.





Titulación de Predio.



Evidencias técnicas.

src/components/dashboard/predio/predio-lista-proceso.js

```
@@ -346,7 +346,7 @@ export const ListaPredioProceso = (props) => {
                                Agregue predios o genere la resolución de publicación de cada predio de manera individual.
                            </Typography>
                        </Grid>
-                       <Grid item>
+                       <Grid item sx={{ display: cantidad > 0 ? '' : 'none' }}>
                            <Button variant="contained" color="secondary" size="large"
                                onClick={
                                    () => setModalId(1)
@@ -636,7 +636,7 @@ export const ListaPredioProceso = (props) => {
                    </Table>
                </Scrollbar>
            </Grid>
-           <Grid item>
+           <Grid item sx={{ display: cantidad > 0 ? '' : 'none' }}>
                <Button variant="contained" color="secondary"
                    onClick={
                        () => GenerarResolcuion()
```

src/components/dashboard/predio/predio-lista-titulacion.js

```
@@ -320,7 +320,7 @@ const GenerarResolucion = async () => {
              </Typography>
            </Grid>
            <Grid
-             item
+             item sx={{ display: cantidad > 0 ? '' : 'none' }}
            >
              <Button
                variant="contained"
@@ -638,8 +638,8 @@ const GenerarResolucion = async () => {
              </Table>
            </Scrollbar>
          </Grid>
-         <Grid item>
-           <Button variant="contained" color="secondary" onClick={() => GenerarResolucion()}>Generar Resolución</Button>
+         <Grid item sx={{ display: cantidad > 0 ? '' : 'none' }}>
+           <Button variant="contained" color="secondary" onClick={() => GenerarResolucion()}>Generar Resolución</Button>
          </Grid>
          <Grid item>
            <Button variant="contained" onClick={() => setModalId(0)}>Cancelar</Button>
```

src/components/dashboard/predio/predio-lista-validacion.js

```
@@ -81,10 +81,10 @@ const modalStyle = {
  transform: 'translate(-50%, -50%)',
  width: {
    sm: '90%',
-   md: '70%'
+   md: '90%'
  },
  overflowY:'scroll',
- maxHeight: '80%',
+ maxHeight: '90%',
  bgcolor: 'background.paper',
  borderRadius: 2,
  boxShadow: 24,
@@ -137,11 +137,79 @@ export const ListaPredioValidacion = (props) => {
    const [propietariOtrosOcupante, setPropietariOtrosOcupante] = useState(0);
    const [parentescoOcupante, setParentescoOcupante] = useState('');

+   const [TipoEstadoCivilLista, setEstadoCivilLista]    = useState([]);
+   const [TipoParentescoLista,  setTipoParentescoLista] = useState([]);
+   const [TipoPoblacionLista,   setTipoPoblacionLista]  = useState([]);
+   const [TipoSubsidioLista,    setTipoSubsidioLista]   = useState([]);
+
+
+   const ListaEstadoCivilLista = async () => {
+     try {
+       axios
+         .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-estados-civil`, {}, config)
+         .then(({ data: res }) => {
+           setEstadoCivilLista(res);
+         })
+         .catch(error => {
+           console.log(error);
        });
    } catch (err) {
      console.error(err);
    }
  }
  const ListaTipoParentescoLista = async () => {
    try {
      axios
        .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-tipo-parentesco`, {}, config)
        .then(({ data: res }) => {
          setTipoParentescoLista(res);
        })
        .catch(error => {
          console.log(error);
        });
    } catch (err) {
      console.error(err);
    }
  }
  const ListaTipoPoblacionLista = async () => {
    try {
      axios
        .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-tipo-poblacion`, {}, config)
        .then(({ data: res }) => {
          setTipoPoblacionLista(res);
        })
        .catch(error => {
          console.log(error);
        });
    } catch (err) {
      console.error(err);
    }
  }
```

```
    const ListaTipoSubsidioLista = async () => {
      try {
        axios
          .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-tipo-subsidio`, {}, config)
          .then(({ data: res }) => {
            setTipoSubsidioLista(res);
          })
          .catch(error => {
            console.log(error);
          });
      } catch (err) {
        console.error(err);
      }
    }


  // Reset selected customers when customers change
  useEffect(() => {
      if (selectedCustomers.length) {
        setSelectedCustomers([]);
      }
      ListaEstadoCivilLista();
      ListaTipoParentescoLista();
      ListaTipoPoblacionLista();
      ListaTipoSubsidioLista();
    },
    // eslint-disable-next-line react-hooks/exhaustive-deps
    [customers]);
@@ -972,10 +1040,16 @@ const ListarCategoriaRelacion = async (idprediosel) => {
                      autoComplete="off"
                      disabled
                    >
-                      <MenuItem value="1">Soltero</MenuItem>
-                      <MenuItem value="2">Casado</MenuItem>
-                      <MenuItem value="3">Unión Libre</MenuItem>
-                      <MenuItem value="4">Viudo</MenuItem>
+                      {
+                          TipoEstadoCivilLista.map((item, i) => (
+                            <MenuItem
+                              key={i}
+                              value={item.ECV_ID}
+                            >
+                              {item.ECV_NOM_ESTADO}
+                            </MenuItem>
+                          ))
+                        }
                    </TextField>
                  </Grid>
                  <Grid item>
@@ -1008,12 +1082,16 @@ const ListarCategoriaRelacion = async (idprediosel) => {
                      disabled
                      fontSize={10}
                    >
-                      <MenuItem value="1">Fonvivienda</MenuItem>
-                      <MenuItem value="2">Inurbe</MenuItem>
-                      <MenuItem value="3">Cajas De Compensación Familiar</MenuItem>
-                      <MenuItem value="4">Caja De Vivienda Militar</MenuItem>
-                      <MenuItem value="5">Banco Agrario</MenuItem>
-                      <MenuItem value="6">Otros</MenuItem>
+                      {
+                          TipoSubsidioLista.map((item, i) => (
+                            <MenuItem
+                              key={i}
+                              value={item.SUB_ID}
+                            >
+                              {item.SUB_NOM_SUBSIDIO}
```

Activar Window
Ve a Configuración pa

```
+                            </MenuItem>
+                          ))
+                        }
                    </TextField>
                  </Grid>
                  <Grid item>
@@ -1032,10 +1110,17 @@ const ListarCategoriaRelacion = async (idprediosel) => {
                      select
                      autoComplete="off"
                    >
-                     <MenuItem value="1">Población Desplazada</MenuItem>
-                     <MenuItem value="2">Negritudes</MenuItem>
-                     <MenuItem value="3">Indigena</MenuItem>
-                     <MenuItem value="4">Raizal</MenuItem>
+                     {
+                         TipoPoblacionLista.map((item, i) => (
+                           <MenuItem
+                             key={i}
+                             value={item.POB_ID}
+                           >
+                             {item.POB_DESC_POBLACION}
+                           </MenuItem>
+                         ))
+                       }
+
                    </TextField>
                  </Grid>
                  <Grid item>
```

src/components/dashboard/predio/predio-radicar.js

```
@@ -201,20 +201,13 @@ const PredioRadicar = () => {
  const [tipoPoblacionOcupante, setTipoPoblacionOcupante] = useState(0);
  const [propietariOtrosOcupante, setPropietariOtrosOcupante] = useState(0);
  const [parentescoOcupante, setParentescoOcupante] = useState('');
-
  const [cantidadObligatorioCat, setCantidadObligatorioCat] = useState(0);
  const [cantidadDocGestionados, setCantidadDocGestionados] = useState(0);
-
  const [cantidadDocObligaorior, setCantidadDocObligaorior] = useState(0);
  const [cantidadDocGesObligatorios, setCantidadDocGesObligatorios] = useState(0);
-
-
  const [aplicaAclaracion, setAplicaAclaracion] = useState(0);
-
-
  const [enable, setEnable] = useState(1);
  const { user } = useAuth();
-
  const [letraPrin, setLetraPrin] = useState('Ninguna');
  const [surPrin, setSurPrin] = useState('No');
  const [bisPrin, setBisPrin] = useState('No');
@@ -241,13 +234,74 @@ const PredioRadicar = () => {
  const [listaDirecciones, setListaDirecciones] = useState({ 'data': [{ "REFERENCIA": '', "DIRECCION": '' }] });
  const [cargadoPredioTodos, setCargadoPredioTodos] = useState(0);
  const [selectedValue, setSelectedValue] = useState('a');
-
  const [buscarManzana, setBuscarManaza] = useState('');

-  const [page, setPage] = useState(2);
```

Activar Windows
Ve a Configuración pa

```
  const [buscarManzana, setBuscarManaza] = useState('');

-  const [page, setPage] = useState(2);
+  const [page, setPage] = useState(0);
  const [rowsPerPage, setRowsPerPage] = useState(5);

+  const [TipoEstadoCivilLista, setEstadoCivilLista]    = useState([]);
+  const [TipoParentescoLista,  setTipoParentescoLista] = useState([]);
+  const [TipoPoblacionLista,   setTipoPoblacionLista]  = useState([]);
+  const [TipoSubsidioLista,    setTipoSubsidioLista]   = useState([]);
+

+  const ListaEstadoCivilLista = async () => {
+    try {
+      axios
+        .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-estados-civil`, {}, config)
+        .then(({ data: res }) => {
+          setEstadoCivilLista(res);
+        })
+        .catch(error => {
+          console.log(error);
+        });
+    } catch (err) {
+      console.error(err);
+    }
+  }
+  const ListaTipoParentescoLista = async () => {
+    try {
+      axios
+        .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-tipo-parentesco`, {}, config)
        setTipoParentescoLista(res);
      })
      .catch(error => {
        console.log(error);
      });
  } catch (err) {
    console.error(err);
  }
}
const ListaTipoPoblacionLista = async () => {
  try {
    axios
      .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-tipo-poblacion`, {}, config)
      .then(({ data: res }) => {
        setTipoPoblacionLista(res);
      })
      .catch(error => {
        console.log(error);
      });
  } catch (err) {
    console.error(err);
  }
}
const ListaTipoSubsidioLista = async () => {
  try {
    axios
      .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_ocupante/get-tipo-subsidio`, {}, config)
      .then(({ data: res }) => {
        setTipoSubsidioLista(res);
      })
      .catch(error => {
```

Activar Windo
Ve a Configuración
Activar Window
Ve a Configuración pa

```
+    } catch (err) {
+      console.error(err);
+    }
+  }
+
  // LISTA DE PREDIOS

  const lista_predios_operador = useCallback(async () => {
@@ -660,8 +714,14 @@ const PredioRadicar = () => {
          text: 'El Anexo no debe ser mayor a 4 MG!',
        })
      } else {
+       Swal.fire({
+         icon:  'info',
+         title: 'Cargando Anexo... ',
+         text:  'Por Favor Espere unos Segundos... Recuerde Que debe confirmar en el Botón Guardar',
+         showConfirmButton: false,
+         timer: 3000
+       })
        setFilescategoria([filescategoria]);
-
      }

    } else {
@@ -762,8 +822,11 @@ const PredioRadicar = () => {
    }

    lista_predios_operador();
-
    CantiddaCategoriaObligatoria();
+    ListaEstadoCivilLista();
+    ListaTipoParentescoLista();


  },
@@ -893,41 +956,182 @@ const PredioRadicar = () => {
  }

}
const GuardarOcupantePredio = async () => {
  if (idOcupante == 0) {


    if (cedulaOcupante != "undefined"

      && nombreOcupante != "undefined" && cedulaOcupante != ""

      && estadocivilOcupante != "undefined" && estadocivilOcupante != ""

      && telefonoOcupante != "undefined" && telefonoOcupante != ""

      && emailOcupante != "undefined" && emailOcupante != ""

      && edadOcupante != "undefined" && edadOcupante != ""




const GuardarOcupantePredio = async () => {
let mensaje =edadOcupante+' ';
let array_fecha=mensaje.split(' ');
var result_mes = array_fecha[1].replace(/Jan/g, "01");
result_mes = array_fecha[1].replace(/Jan/g, "01");
```

Activar Windows
Ve a Configuración pa

```
                >
                  <TextField
-                   fullWidth
-                   label="Parentesco"
-                   value={parentescoOcupante}
-                   onChange={handleParentescoOcupanteChange}
-                   type="text"
-                   autoComplete="off"
-                  />
+                   fullWidth
+                   label="Parentesco"
+                   value={parentescoOcupante}
+                   onChange={handleParentescoOcupanteChange}
+                   select
+                   autoComplete="off"
+                  >
+                    {
+                        TipoParentescoLista.map((item, i) => (
+                          <MenuItem
+                            key={i}
+                            value={item.PAR_PARENTESCO}
+                          >
+                            {item.PAR_PARENTESCO}
+                          </MenuItem>
+                        ))
+                      }
+
+                  </TextField>
                 </Grid>
                 <Grid
                   item
                     autoComplete="off"
                   >
-                    <MenuItem value="1">Soltero</MenuItem>
-                    <MenuItem value="2">Casado</MenuItem>
-                    <MenuItem value="3">Unión Libre</MenuItem>
-                    <MenuItem value="4">Viudo</MenuItem>
+                    {
+                        TipoEstadoCivilLista.map((item, i) => (
+                          <MenuItem
+                            key={i}
+                            value={item.ECV_ID}
+                          >
+                            {item.ECV_NOM_ESTADO}
+                          </MenuItem>
+                        ))
+                      }
+

                   </TextField>

                 </Grid>
@@ -3428,12 +3647,17 @@ const PredioRadicar = () => {
                     select
                     autoComplete="off"
                   >
-                    <MenuItem value="1">Fonvivienda</MenuItem>
-                    <MenuItem value="2">Inurbe</MenuItem>
-                    <MenuItem value="3">Cajas De Compensación Familiar</MenuItem>
-                    <MenuItem value="4">Caja De Vivienda Militar</MenuItem>
-                    <MenuItem value="5">Banco Agrario</MenuItem>
-                    <MenuItem value="6">Otros</MenuItem>
+                    {
+                        TipoSubsidioLista.map((item, i) => (
```

Activar Wind
Ve a Configuració
Activar Windo
Ve a Configuración p

```
                            value={item.SUB_ID}
                          >
                            {item.SUB_NOM_SUBSIDIO}
                          </MenuItem>
                        ))
                      }

                    </TextField>
                  </Grid>
                  <Grid
@@ -3450,10 +3674,17 @@ const PredioRadicar = () => {
                      select
                      autoComplete="off"
                    >
                      <MenuItem value="1">Población Desplazada</MenuItem>
                      <MenuItem value="2">Negritudes</MenuItem>
                      <MenuItem value="3">Indigena</MenuItem>
                      <MenuItem value="4">Raizal</MenuItem>
                       {
                          TipoPoblacionLista.map((item, i) => (
                            <MenuItem
                              key={i}
                              value={item.POB_ID}
                            >
                              {item.POB_DESC_POBLACION}
                            </MenuItem>
                          ))
                       }

                    </TextField>
```

Activar Windows
Ve a Configuración par

src/components/widgets/forms/form-8.js config.js

```
-
-
var oracledb = require('oracledb');
var connAttrs = {
    user: "DBS_SEG_PROYECTOS",
```

controllers/mi-plan/plan-desarrollos.js

```
+ // controllers/planDesarrolloController.js
+
+ const PlanDesarrollo = require('../models/planDesarrolloModel');
+
+ // Obtener todos los planes de desarrollo
+ exports.getAllPlanesDesarrollo = async (req, res) => {
+   try {
+     const planesDesarrollo = await PlanDesarrollo.find();
+     res.json(planesDesarrollo);
+   } catch (error) {
+     console.error('Error al obtener los planes de desarrollo:', error);
+     res.status(500).json({ error: 'Ocurrió un error al obtener los planes de desarrollo' });
+   }
+ };
+
+ // Obtener un plan de desarrollo por ID
+ exports.getPlanDesarrolloById = async (req, res) => {
+   try {
+     const planDesarrollo = await PlanDesarrollo.findById(req.params.id);
+     if (!planDesarrollo) {
+       return res.status(404).json({ error: 'Plan de Desarrollo no encontrado' });
+     }
+     res.json(planDesarrollo);
+   } catch (error) {
+     console.error('Error al obtener el plan de desarrollo:', error);
+     res.status(500).json({ error: 'Ocurrió un error al obtener el plan de desarrollo' });
+   }
+ };
+
+ // Crear un nuevo plan de desarrollo
+ exports.createPlanDesarrollo = async (req, res) => {
+   try {
+     const planDesarrollo = new PlanDesarrollo(req.body);
+     const nuevoPlanDesarrollo = await planDesarrollo.save();
+     res.status(201).json(nuevoPlanDesarrollo);
```

Activar Windows
Ve a Configuración para

```
  } catch (error) {
    console.error('Error al crear el plan de desarrollo:', error);
    res.status(500).json({ error: 'Ocurrió un error al crear el plan de desarrollo' });
  }
};

// Actualizar un plan de desarrollo por ID
exports.updatePlanDesarrollo = async (req, res) => {
  try {
    const planDesarrollo = await PlanDesarrollo.findByIdAndUpdate(
      req.params.id,
      req.body,
      { new: true }
    );
    if (!planDesarrollo) {
      return res.status(404).json({ error: 'Plan de Desarrollo no encontrado' });
    }
    res.json(planDesarrollo);
  } catch (error) {
    console.error('Error al actualizar el plan de desarrollo:', error);
    res.status(500).json({ error: 'Ocurrió un error al actualizar el plan de desarrollo' });
  }
};

// Eliminar un plan de desarrollo por ID
exports.deletePlanDesarrollo = async (req, res) => {
  try {
    const planDesarrollo = await PlanDesarrollo.findByIdAndDelete(req.params.id);
    if (!planDesarrollo) {
      return res.status(404).json({ error: 'Plan de Desarrollo no encontrado' });
    }
    res.json({ message: 'Plan de Desarrollo eliminado correctamente' });
  } catch (error) {
    console.error('Error al eliminar el plan de desarrollo:', error);
    res.status(500).json({ error: 'Ocurrió un error al eliminar el plan de desarrollo' });
  }
};
```

Activar Windows
Ve a Configuración par

```
exports.updatePlanDesarrollo = async (req, res) => {
  try {
    const planDesarrollo = await PlanDesarrollo.findByIdAndUpdate(
      req.params.id,
      req.body,
      { new: true }
    );
    if (!planDesarrollo) {
      return res.status(404).json({ error: 'Plan de Desarrollo no encontrado' });
    }
    res.json(planDesarrollo);
  } catch (error) {
    console.error('Error al actualizar el plan de desarrollo:', error);
    res.status(500).json({ error: 'Ocurrió un error al actualizar el plan de desarrollo' });
  }
};

// Eliminar un plan de desarrollo por ID
exports.deletePlanDesarrollo = async (req, res) => {
  try {
    const planDesarrollo = await PlanDesarrollo.findByIdAndDelete(req.params.id);
    if (!planDesarrollo) {
      return res.status(404).json({ error: 'Plan de Desarrollo no encontrado' });
    }
    res.json({ message: 'Plan de Desarrollo eliminado correctamente' });
  } catch (error) {
    console.error('Error al eliminar el plan de desarrollo:', error);
    res.status(500).json({ error: 'Ocurrió un error al eliminar el plan de desarrollo' });
  }
};
```

controllers/titulacion/mit_predio.js

```
@@ -18,6 +18,8 @@ const multer = require('multer')
const fs = require('fs');
var pdf = require('html-pdf');
const { PDFDocument } = require('pdf-lib');
+const AdmZip = require('adm-zip');
+const zlib = require('zlib');

// const rutaServerUplod ='C:/servicios-web'; // window
 const rutaServerUplod ='/servicios-web'; // centos
@@ -2451,6 +2453,231 @@ const EliminarEjecutoria = async (req, res) => {
    });
}

+const descargarZip = async (req, res) => {
+ oracledb.getConnection(connAttrs, async function (err, connection) {
+    if (err) {
+        // Error connecting to DB
+        res.set('Content-Type', 'application/json');
+        res.status(500).send(JSON.stringify({
+            status: 500,
+            message: "Error connecting to DB",
+            detailed_message: err.message
+        }));
+        return;
+    }
+    let ruta_principal=rutaServerUplod+'/uploads/titulacion/resolucion_individuales/';
+
+    let {ESTADO,FECHA} = req.body
+    var sqlQuery = ``;
+    var CAMPOS=`VWS_PREDIOS_GESTIONADOS.TEMP_CONSECUTIVO AS RESOLUCION`;
+
+    if(FECHA==0){
+        sqlQuery = `SELECT `+CAMPOS+`  FROM VWS_PREDIOS_GESTIONADOS WHERE VWS_PREDIOS_GESTIONADOS.EAP_ID=`+ESTADO+` ORDER BY VWS_PREDIOS_GESTIONADOS.EAP_ESTADO ASC`;
+    }else{
+        sqlQuery = `SELECT `+CAMPOS+` FROM VWS_PREDIOS_GESTIONADOS WHERE VWS_PREDIOS_GESTIONADOS.EAP_ID=`+ESTADO+`
+            const filesToCompress = [];
+
+            result = await connection.execute(
+                sqlQuery,
+            [],
+            { resultSet: true, outFormat: oracledb.OUT_FORMAT_OBJECT });
+
+            const rs = result.resultSet;
+            let row;
+
+
+            while ((row = await rs.getRow())) {
+
+                if (fs.existsSync(ruta_principal+row.RESOLUCION+'_resolucion.pdf')) {
+
+                filesToCompress.push(ruta_principal+row.RESOLUCION+'_resolucion.pdf');
+                }
+
+            }
+
+        // Ruta de salida para el archivo ZIP
+        const zipPath = ruta_principal+'salida.zip';
+        if (fs.existsSync(zipPath)) {
+        fs.unlink(zipPath, (unlinkErr) => {
+            if (unlinkErr) {
+            console.error('Error al eliminar el archivo ZIP:', unlinkErr);
+            } else {
+            console.log('Archivo ZIP eliminado con éxito.');
+            }
+        });
+    }
```

Activar Window
Ve a Configuración pa

```
            }
        });
    }


        // Crear el archivo vacío
        fs.closeSync(fs.openSync(zipPath, 'w'));

        // Otorgar permisos de escritura
        fs.chmod(zipPath, '666', (err) => {
        if (err) {
            console.error('Error al otorgar permisos de escritura:', err);
            return;
        }
        console.log('Permisos de escritura otorgados con éxito.');
        });



         // Otorgar permisos de escritura
         fs.chmod(zipPath, '777', (err) => {
            if (err) {
            console.error('Error al otorgar permisos de escritura:', err);
            return;
            }
        });
        // Comprimir los archivos y guardar como ZIP
        compressFiles(filesToCompress, zipPath);
        const fileName = 'archivo.zip';

        const fileStream = fs.createReadStream(zipPath);
        const stat = fs.statSync(zipPath);

        res.setHeader('Content-Disposition', `attachment; filename="${fileName}"`);
        res.setHeader('Content-Type', 'application/zip');
        res.setHeader('Content-Length', stat.size);
```

Activar Windows
Ve a Configuración para

```
const compressFiles = (files, zipPath) => {
  try {
    // Crear una instancia de AdmZip
    const zip = new AdmZip();

    // Agregar cada archivo al archivo ZIP
    files.forEach((filePath) => {
      const fileData = fs.readFileSync(filePath);
      const fileName = filePath.substring(filePath.lastIndexOf('/') + 1);
      zip.addFile(fileName, fileData);
    });

    // Guardar el ZIP en disco
    zip.writeZip(zipPath);

    console.log('Compresión completada.');
  } catch (error) {
    console.error('Error al comprimir los archivos:', error);
  }
};


const HistoricoEjecutoriasParaGobernacion = async (req, res) => {
    oracledb.getConnection(connAttrs, function (err, connection) {
        if (err) {
            // Error connecting to DB
            res.set('Content-Type', 'application/json');
            res.status(500).send(JSON.stringify({
                status: 500,
                message: "Error connecting to DB",
                detailed_message: err.message
            }));
            return;
```

Activar Windo
Ve a Configuración

```
            return;
        }

        // const {search} = req.query;

        let whereClause = '';

        /*if (search.trim()) {
        whereClause = ` WHERE PRE_REFERENCIA LIKE '%${search.trim()}%' OR PRE_MATRICULA LIKE '%${search.trim()}%' OR TEMP_CONSECUTIVO LIKE
 '%${search.trim()}%' `; // Agregar los campos de búsqueda necesarios
        }*/

        const sqlQuery =  `SELECT
        VWS_HISTORICO_EJECUTORIAS.PRE_DIRECCION,
        VWS_HISTORICO_EJECUTORIAS.PRE_MATRICULA,
        VWS_HISTORICO_EJECUTORIAS.TEMP_CONSECUTIVO,
        VWS_HISTORICO_EJECUTORIAS.PRE_AREA_M2,
        VWS_HISTORICO_EJECUTORIAS.FECHA,
        VWS_HISTORICO_EJECUTORIAS.FECHA_FIRMA,
        VWS_HISTORICO_EJECUTORIAS.EJE_TIPO_NOTIFICACION,
        VWS_HISTORICO_EJECUTORIAS.EJE_FECHA_NOTIFICACION,
        VWS_HISTORICO_EJECUTORIAS.EJE_FECHA_EJECUTORIA,
        VWS_HISTORICO_EJECUTORIAS.EJE_ID,
        VWS_HISTORICO_EJECUTORIAS.PRE_REFERENCIA,
        VWS_HISTORICO_EJECUTORIAS.PRE_ID,
        VWS_PREDIOS_GESTIONADOS.PRE_DIRECCION,
        VWS_PREDIOS_GESTIONADOS.PRE_FK_ESCRITURA,
        VWS_PREDIOS_GESTIONADOS.PRE_MATRICULA,
        VWS_PREDIOS_GESTIONADOS.PRE_RESOLUCION,
        VWS_PREDIOS_GESTIONADOS.PRE_FEC_TITULACION,
        VWS_PREDIOS_GESTIONADOS.TEMP_GENERADA,
        VWS_PREDIOS_GESTIONADOS.PRE_DESTINO,
        VWS_PREDIOS_GESTIONADOS.PRE_OBSERVACION,
        VWS_PREDIOS_GESTIONADOS.EST_NOMBRE,
        VWS_PREDIOS_GESTIONADOS.ESC_ID,
        VWS_PREDIOS_GESTIONADOS.ESC_NUM_ESCRI,
        VWS_PREDIOS_GESTIONADOS.PRE_BARRIO,
        VWS_PREDIOS_GESTIONADOS.PRE_ULT_UPDATE,
         VWS_PREDIOS_GESTIONADOS.TEMP_CONSECUTIVO,
         VWS_PREDIOS_GESTIONADOS.TEMP_FECHA,
         VWS_PREDIOS_GESTIONADOS.TEMP_DESDE_REF,
         VWS_PREDIOS_GESTIONADOS.PRE_ID,
         VWS_PREDIOS_GESTIONADOS.PRE_REFERENCIA,
         VWS_PREDIOS_GESTIONADOS.PRE_AREA_M2,
         VWS_PREDIOS_GESTIONADOS.CERT,
         VWS_PREDIOS_GESTIONADOS.FECHA,
         VWS_PREDIOS_GESTIONADOS.FECHA_FIRMA
         FROM
         VWS_HISTORICO_EJECUTORIAS
         INNER JOIN VWS_PREDIOS_GESTIONADOS ON VWS_HISTORICO_EJECUTORIAS.PRE_ID = VWS_PREDIOS_GESTIONADOS.PRE_ID
         ORDER BY
         VWS_HISTORICO_EJECUTORIAS.EJE_ID DESC
          `;

        binds = [];

        connection.execute(sqlQuery,binds, {
            outFormat: oracledb.OBJECT // Return the result as Object
        }, function (err, result) {
            if (err) {
                res.set('Content-Type', 'application/json');
                res.status(500).send(JSON.stringify({
                    status: 500,
                    message: "Error al obtener dba_tablespaces",
                    detailed_message: err.message
                }));
            } else {
                res.header('Access-Control-Allow-Origin','*');
                res.header('Access-Control-Allow-Headers','Content-Type');
                res.header('Access-Control-Allow-Methods','GET,PUT,POST,DELETE,OPTIONS');
                res.contentType('application/json').status(200);
                res.send(JSON.stringify(result.rows));

            }
            // Release the connection
            connection.release(
```

Activar Windows
Ve a Configuración para

Activar Windows
Ve a Configuración para

```
+            connection.release(
+                function (err) {
+                    if (err) {
+                        console.error(err.message);
+                    } else {
+                        console.log("GET /sendTablespace : Connection released");
+                    }
+                });
+        });
+
+    });
+}
+
+
+
 module.exports.Guardar= Guardar;
 module.exports.Modificar= Modificar;
 module.exports.ModificarEstado= ModificarEstado;
 @@ -2490,6 +2717,7 @@ module.exports.test= test;
 module.exports.GenerarEjecutoriaPredios= GenerarEjecutoriaPredios;
 module.exports.HistoricoEjecutorias= HistoricoEjecutorias;
 module.exports.EliminarEjecutoria= EliminarEjecutoria;
+module.exports.descargarZip= descargarZip;

-
+module.exports.HistoricoEjecutoriasParaGobernacion= HistoricoEjecutoriasParaGobernacion;
```

controllers/titulacion/mit_resolucion.js

```
 @@ -890,8 +890,8 @@ async function GeneracionResolucionDoc(req, res) {
                 var html_cabeza        = rowsPlantilla['rows'][0][3]; //TEXTO CABEZA
                 var html_pie           = rowsPlantilla['rows'][0][4]; //TEXTO PIE

-                var html_sin_afectacion = rowsPlantilla['rows'][0][5]; //TEXTO SIN AFECTACION
-                var html_con_afectacion = rowsPlantilla['rows'][0][6]; //TEXTO CON AFECTACION
+                var html_sin_afectacion = rowsPlantilla['rows'][0][6]; //TEXTO SIN AFECTACION
+                var html_con_afectacion = rowsPlantilla['rows'][0][5]; //TEXTO CON AFECTACION
                 var html_escritura      = rowsPlantilla['rows'][0][7]; //TEXTO SI ES ESCRITURA
                 var html_resolucion     = rowsPlantilla['rows'][0][8]; //TEXTO SI ES RESOLUCION

 @@ -1184,37 +1184,29 @@ async function GeneracionResolucionDoc(req, res) {
             console.log('------------------------');


-    // AFEACTACION SI APLICA
-
+    // AFEACTACION SI/NO APLICA
         var ap='';
-
         const rows_aplica_afectacion_no_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
-         WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``
+         WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(3,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``
         );
-         console.log( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
-         WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);
-
-         console.log(rows_aplica_afectacion_no_afectacion['rows'][0][0]);
-
-         if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){
+         if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){ //VERIFICAMOS SI NO TIENE AFECTACION // SOLTERO(A)  VIUDO(A)
             const rows_aplica_afectacion_si_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
-            WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(2,3) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);
```

Activar Windows
Ve a Configuración para

```
-            if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){
+            if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){ // CASADO(A)  UNIÓN LIBRE
                 ap= html_con_afectacion;
-
             }
-
         }else{
-            if(rows_aplica_afectacion_no_afectacion['rows'][0][0]>0){
-            ap=html_sin_afectacion;
+            if(rows_aplica_afectacion_no_afectacion['rows'][0][0]>0){ //SI NO TIENE AFECTACION
+                ap=html_sin_afectacion;

             }
         }
-
+        if(MIT_PLANTILLAS_PLA_ID==4){
+            ap = ap.replace(/(TERCERO.)/g, 'CUARTO.');
+           }
         html = html.replace(/(--afectacion--)/g,ap);
-
-        // AFEACTACION SI APLICA
+        // AFEACTACION SI/NO APLICA

         var url_qr='http://secplaneacion.barranquilla.gov.co:9000/mit_predio/descarga-anexo/'+inicio+'_resolucion.pdf/resolucion_individuales';
         // async
@@ -2709,8 +2701,8 @@ async function GeneracionResolucionDocFirmas(PRE_ID) {
            var html_cabeza         = rowsPlantilla['rows'][0][3]; //TEXTO CABEZA
            var html_pie            = rowsPlantilla['rows'][0][4]; //TEXTO PIE
            var PLA_FIRMAR          = rowsPlantilla['rows'][0][9]; //TEXTO FIRMA
-           var html_sin_afectacion = rowsPlantilla['rows'][0][5]; //TEXTO SIN AFECTACION
-           var html_con_afectacion = rowsPlantilla['rows'][0][6]; //TEXTO CON AFECTACION
+           var html_sin_afectacion = rowsPlantilla['rows'][0][6]; //TEXTO SIN AFECTACION
+           var html_con_afectacion = rowsPlantilla['rows'][0][5]; //TEXTO CON AFECTACION
            var html_escritura      = rowsPlantilla['rows'][0][7]; //TEXTO SI ES ESCRITURA
            var html_resolucion     = rowsPlantilla['rows'][0][8]; //TEXTO SI ES RESOLUCION

@@ -2937,38 +2929,29 @@ async function GeneracionResolucionDocFirmas(PRE_ID) {
            html_cabeza = html_cabeza.replace(/(--resolucion_individual--)/g, resolucion_individual_sel);

        // AFEACTACION SI APLICA

            var ap='';
        // AFEACTACION SI/NO APLICA
        var ap='';
        const rows_aplica_afectacion_no_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(3,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``
        );
        if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){ //VERIFICAMOS SI NO TIENE AFECTACION // SOLTERO(A)  VIUDO(A)
        const rows_aplica_afectacion_si_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,2) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);

            const rows_aplica_afectacion_no_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
                WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``
                );
                console.log( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
                WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);
        if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){ // CASADO(A)  UNIÓN LIBRE
                 ap= html_con_afectacion;
        }
        }else{
            if(rows_aplica_afectacion_no_afectacion['rows'][0][0]>0){ //SI NO TIENE AFECTACION
                ap=html_sin_afectacion;

                console.log(rows_aplica_afectacion_no_afectacion['rows'][0][0]);

                if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){
                    const rows_aplica_afectacion_si_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
                    WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(2,3) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);

                    if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){
                        ap= html_con_afectacion;

                    }
```

Activar Windc
Ve a Configuración

Activar Windc
Ve a Configuración

```
                    }
                }

                html = html.replace(/(--afectacion--)/g,ap);
        // AFEACTACION SI APLICA


            }
        }
        if(MIT_PLANTILLAS_PLA_ID==4){
            ap = ap.replace(/(TERCERO.)/g, 'CUARTO.');
        }
        html = html.replace(/(--afectacion--)/g,ap);
        // AFEACTACION SI/NO APLICA

        var url_qr='http://secplaneacion.barranquilla.gov.co:9000/mit_predio/descarga-
anexo/'+row2.TEMP_CONSECUTIVO+'_resolucion.pdf/resolucion_individuales';
        // async
@@ -3053,16 +3036,12 @@ async function aplica_afectacion(OCU_FK_PREDIO){


        const rows_aplica_afectacion_no_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+OCU_FK_PREDIO+``
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(3,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+OCU_FK_PREDIO+``
        );
        console.log( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+OCU_FK_PREDIO+``);

        console.log(rows_aplica_afectacion_no_afectacion['rows'][0][0]);

        if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){
            const rows_aplica_afectacion_si_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
            WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(2,3,5) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+OCU_FK_PREDIO+``);
            WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,2) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+OCU_FK_PREDIO+``);

            if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){
                ap= `<p style='margin-right:2.55pt;margin-left:0cm;font-size:12px;font-family:"Calibri",sans-serif;margin:0cm;text-
align:justify;'>
```

Activar Windows
Ve a Configuración para

```
       FROM VWS_RESOLUCIONES_TITULA WHERE VWS_RESOLUCIONES_TITULA.TEMP_GENERADA IN(3)`;
       FROM VWS_RESOLUCIONES_TITULA WHERE VWS_RESOLUCIONES_TITULA.TEMP_GENERADA IN(3,4)`;

        result = await connection.execute(
            sqlQuery,
@@ -3378,8 +3357,8 @@ async function GeneracionResolucionDocFirmasParaCorregirDev(PRE_ID) {
           var html_cabeza         = rowsPlantilla['rows'][0][3]; //TEXTO CABEZA
           var html_pie            = rowsPlantilla['rows'][0][4]; //TEXTO PIE
           var PLA_FIRMAR          = rowsPlantilla['rows'][0][9]; //TEXTO FIRMA
           var html_sin_afectacion = rowsPlantilla['rows'][0][5]; //TEXTO SIN AFECTACION
           var html_con_afectacion = rowsPlantilla['rows'][0][6]; //TEXTO CON AFECTACION
           var html_sin_afectacion = rowsPlantilla['rows'][0][6]; //TEXTO SIN AFECTACION
           var html_con_afectacion = rowsPlantilla['rows'][0][5]; //TEXTO CON AFECTACION
           var html_escritura      = rowsPlantilla['rows'][0][7]; //TEXTO SI ES ESCRITURA
           var html_resolucion     = rowsPlantilla['rows'][0][8]; //TEXTO SI ES RESOLUCION

@@ -3598,37 +3577,29 @@ async function GeneracionResolucionDocFirmasParaCorregirDev(PRE_ID) {
            html_cabeza = html_cabeza.replace(/(--resolucion_individual--)/g, resolucion_individual_sel);


            // AFEACTACION SI APLICA

                var ap='';

                const rows_aplica_afectacion_no_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
                    WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``
                    );
                    console.log( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
                    WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);

                    console.log(rows_aplica_afectacion_no_afectacion['rows'][0][0]);

                    if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){
                        const rows_aplica_afectacion_si_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
                        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(2,3) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);

                        if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){
                            ap= html_con_afectacion;
```

Activar Windows
Ve a Configuración para

```
                }else{
                    if(rows_aplica_afectacion_no_afectacion['rows'][0][0]>0){
                    ap=html_sin_afectacion;

                    }
                }

                html = html.replace(/(--afectacion--)/g,ap);
      // AFEACTACION SI APLICA
        // AFEACTACION SI/NO APLICA
        var ap='';
        const rows_aplica_afectacion_no_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(3,4) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``
        );
        if(rows_aplica_afectacion_no_afectacion['rows'][0][0]==0){ //VERIFICAMOS SI NO TIENE AFECTACION // SOLTERO(A)  VIUDO(A)
        const rows_aplica_afectacion_si_afectacion = await connection.execute( `SELECT COUNT(*) AS CANTIDAD FROM MIT_OCUPANTE
        WHERE MIT_OCUPANTE.OCU_FK_ESTADO_CIVIL IN(1,2) AND MIT_OCUPANTE.OCU_FK_PREDIO=`+row2.PRE_ID+``);

        if(rows_aplica_afectacion_si_afectacion['rows'][0][0]>0){ // CASADO(A)  UNIÓN LIBRE
                ap= html_con_afectacion;
        }
        }else{
            if(rows_aplica_afectacion_no_afectacion['rows'][0][0]>0){ //SI NO TIENE AFECTACION
                ap=html_sin_afectacion;

            }
        }
        if(MIT_PLANTILLAS_PLA_ID==4){
         ap = ap.replace(/(TERCERO.)/g, 'CUARTO.');
        }
        html = html.replace(/(--afectacion--)/g,ap);
        // AFEACTACION SI/NO APLICA


        var url_qr='http://secplaneacion.barranquilla.gov.co:9000/mit_predio/descarga-
anexo/'+row2.TEMP_CONSECUTIVO+'_resolucion.pdf/resolucion_individuales';
@@ -3699,6 +3670,41 @@ async function GeneracionResolucionDocFirmasParaCorregirDev(PRE_ID) {

// fin firma individual resolucion


const CorreccionSinConFirmaAprobar= async (req, res) => {
    oracledb.getConnection(connAttrs, async function (err, connection) {
        if (err) {
            // Error connecting to DB
            res.set('Content-Type', 'application/json');
            res.status(500).send(JSON.stringify({
                status: 500,
                message: "Error connecting to DB",
                detailed_message: err.message
            }));
            return;
        }
        let {PRE_ID,FIRMA} = req.body

        if(FIRMA==1){
                GeneracionResolucionDocFirmas(PRE_ID);// SI YA ESTA APROBADA POR LA SECRETARIA
        }else{
            if(FIRMA==3 || FIRMA==4){
                GeneracionResolucionDocFirmasParaCorregirDev(PRE_ID); // LA RESOLUCION ESTA EN FIRMAS O DEVUELTAS
            }

        }
        res.header('Access-Control-Allow-Origin','*');
        res.header('Access-Control-Allow-Headers','Content-Type');
        res.header('Access-Control-Allow-Methods','GET,PUT,POST,DELETE,OPTIONS');
        res.contentType('application/json').status(200);
        res.send('OK');
```

Activar Wir

```
module.exports.Guardar= Guardar;
module.exports.Modificar= Modificar;
module.exports.GetLineaRegistro= GetLineaRegistro;
@@ -3734,6 +3740,8 @@ module.exports.FirmarTodasResolucionesPendientes= FirmarTodasResolucionesPendien
module.exports.CorreccionFirmarTodasResolucionesPendientes= CorreccionFirmarTodasResolucionesPendientes;


+module.exports.CorreccionSinConFirmaAprobar= CorreccionSinConFirmaAprobar;
+
```

node_modules/.package-lock.json

```
@@ -200,9 +200,9 @@
      }
    },
    "node_modules/adm-zip": {
-     "version": "0.5.9",
-     "resolved": "https://registry.npmjs.org/adm-zip/-/adm-zip-0.5.9.tgz",
-     "integrity": "sha512-s+3fXLkeeLjZ2kLjCBwQufpI5fuN+kIGBxu6530nVQZGVol0d7Y/M88/xw9HGGUcJjKf8LutN3VPRUBq6N7Ajg==",
+     "version": "0.5.10",
+     "resolved": "https://registry.npmjs.org/adm-zip/-/adm-zip-0.5.10.tgz",
+     "integrity": "sha512-x0HvcHqVJNTPk/Bw8JbLWlWoo6Wwnsug0fnYYro1HBrjxZ3G7/AZk7Ahv8JwDe1uIcz8eBqvu86FuF1POiG7vQ==",
      "engines": {
        "node": ">=6.0"
      }
```

node_modules/adm-zip/adm-zip.js

```
@@ -266,8 +266,9 @@ module.exports = function (/**String*/ input, /** object */ options) {
         * @param zipPath optional path inside zip
         * @param filter optional RegExp or Function if files match will
         *               be included.
+        * @param {number | object} attr - number as unix file permissions, object as filesystem Stats object
         */
-        addLocalFolder: function (/**String*/ localPath, /**String=*/ zipPath, /**=RegExp|Function*/ filter) {
+        addLocalFolder: function (/**String*/ localPath, /**String=*/ zipPath, /**=RegExp|Function*/ filter, /**=number|object*/ attr) {
             // Prepare filter
             if (filter instanceof RegExp) {
                 // if filter is RegExp wrap it
@@ -299,9 +300,9 @@ module.exports = function (/**String*/ input, /** object */ options) {
                         if (filter(p)) {
                             var stats = filetools.fs.statSync(filepath);
                             if (stats.isFile()) {
-                                self.addFile(zipPath + p, filetools.fs.readFileSync(filepath), "", stats);
+                                self.addFile(zipPath + p, filetools.fs.readFileSync(filepath), "", attr ? attr : stats);
                             } else {
-                                self.addFile(zipPath + p + "/", Buffer.alloc(0), "", stats);
+                                self.addFile(zipPath + p + "/", Buffer.alloc(0), "", attr ? attr : stats);
                             }
                         }
                     });
@@ -375,7 +376,9 @@ module.exports = function (/**String*/ input, /** object */ options) {
                                 }
                             });
                         } else {
-                            next();
+                            process.nextTick(() => {
+                                next();
+                            });
```

```
+                                process.nextTick(() => {
+                                    next();
+                                });
                             }
                         } else {
                             callback(true, undefined);
 @@ -441,24 +444,22 @@ module.exports = function (/**String*/ input, /** object */ options) {
             var fileattr = entry.isDirectory ? 0x10 : 0; // (MS-DOS directory flag)

             // extended attributes field for Unix
-            if (!Utils.isWin) {
-                // set file type either S_IFDIR / S_IFREG
-                let unix = entry.isDirectory ? 0x4000 : 0x8000;
-
-                if (isStat) {
-                    // File attributes from file stats
-                    unix |= 0xfff & attr.mode;
-                } else if ("number" === typeof attr) {
-                    // attr from given attr values
-                    unix |= 0xfff & attr;
-                } else {
-                    // Default values:
-                    unix |= entry.isDirectory ? 0o755 : 0o644; // permissions (drwxr-xr-x) or (-r-wr--r--)
-                }
+            // set file type either S_IFDIR / S_IFREG
+            let unix = entry.isDirectory ? 0x4000 : 0x8000;

-                fileattr = (fileattr | (unix << 16)) >>> 0; // add attributes
+            if (isStat) {
+
+                fileattr = (fileattr | (unix << 16)) >>> 0; // add attributes
+
             }

             entry.attr = fileattr;

             entry.setData(content);
 @@ -633,12 +634,14 @@ module.exports = function (/**String*/ input, /** object */ options) {
          * @param callback The callback will be executed when all entries are extracted successfully or any error is thrown.
          */
         extractAllToAsync: function (/**String*/ targetPath, /**Boolean*/ overwrite, /**Boolean*/ keepOriginalPermission, /**Function*/ callback)
 {
-            if (!callback) {
-                callback = function () {};
-            }
             overwrite = get_Bool(overwrite, false);
             if (typeof keepOriginalPermission === "function" && !callback) callback = keepOriginalPermission;
             keepOriginalPermission = get_Bool(keepOriginalPermission, false);
+            if (!callback) {
+                callback = function (err) {
+                    throw new Error(err);
+                };
+            }
             if (!_zip) {
                 callback(new Error(Utils.Errors.NO_ZIP));
                 return;
```

node_modules/adm-zip/headers/mainHeader.js

```
 @@ -76,7 +76,7 @@ module.exports = function () {
                 // total number of entries
                 _totalEntries = Utils.readBigUInt64LE(data, Constants.ZIP64TOT);
                 // central directory size in bytes
-                _size = Utils.readBigUInt64LE(data, Constants.ZIP64SIZ);
+                _size = Utils.readBigUInt64LE(data, Constants.ZIP64SIZE);
                 // offset of first CEN header
                 _offset = Utils.readBigUInt64LE(data, Constants.ZIP64OFF);

 @@ -127,3 +127,4 @@ module.exports = function () {
         }
     };
 };
+ // Misspelled
 \ No newline at end of file
```

node_modules/adm-zip/package.json

```
{
  "name": "adm-zip",
-  "version": "0.5.9",
+  "version": "0.5.10",
  "description": "Javascript implementation of zip for nodejs with support for electron original-fs. Allows user to create or extract zip files
both in memory or to/from disk",
  "scripts": {
    "test": "mocha -R spec",
@@ -41,7 +41,7 @@
  },
  "devDependencies": {
    "chai": "^4.3.4",
-    "mocha": "^8.3.2",
+    "mocha": "^10.2.0",
    "prettier": "^2.2.1",
    "rimraf": "^3.0.2"
  }
```

controllers/titulacion/mit_predio.js

```
@@ -3587,6 +3587,65 @@ const descargarZipEjecutorias = async (req, res) => {
  }


+  const Gobernacion = async  (req, res) => {
+   oracledb.getConnection(connAttrs, async function (err, connection) {
+       if (err) {
+           // Error connecting to DB
+           res.set('Content-Type', 'application/json');
+           res.status(500).send(JSON.stringify({
+               status: 500,
+               message: "Error connecting to DB",
+               detailed_message: err.message
+           }));
+           return;
+          }
+           try {
+              let binds = [];
+             // Obtener los datos del formulario desde req.body
+             const { GOB_RAD_SIGOB, GOB_FECHA_RAD_SIGOB,FECHA } = req.body;
+
+           // Confirmar la transacción
+
+
+           const sqlQuery =
+             `SELECT * FROM (
+               SELECT
+               PRE_ID,
+               EJE_ID
+               FROM VWS_HISTORICO_EJECUTORIAS  WHERE TO_CHAR(EJE_FECHA_REGISTRO, 'YYYY-MM-DD')='`+FECHA+`'
+             ) ORDER BY EJE_ID DESC`;
+
+          console.log(sqlQuery);
+
+           result = await connection.execute(
+               sqlQuery,
+           [],
+           { resultSet: true, outFormat: oracledb.OUT_FORMAT_OBJECT });
```

Activar Windows
Ve a Configuración para

```
            { resultSet: true, outFormat: oracledb.OUT_FORMAT_OBJECT });

            const rs2 = result.resultSet;
            let row2;
            /* LISTA DE PREDIOS */
            while ((row2 = await rs2.getRow()))
            {
                await connection.execute( `INSERT INTO MIT_GOBERNACIONES
                (GOB_ID, GOB_PRED_ID,GOB_RAD_SIGOB,GOB_FECHA_RAD_SIGOB) VALUES
                 (seq_gobernacion.NEXTVAL,`+row2.PRE_ID+`,'`+GOB_RAD_SIGOB+`','`+GOB_FECHA_RAD_SIGOB+`')`, binds, {autoCommit: true});
              console.log(`INSERT INTO MIT_GOBERNACIONES(GOB_ID, GOB_PRED_ID,GOB_RAD_SIGOB,GOB_FECHA_RAD_SIGOB) VALUES
(seq_gobernacion.NEXTVAL,`+row2.PRE_ID+`,'`+GOB_RAD_SIGOB+`','`+GOB_FECHA_RAD_SIGOB+`'`);
            }


              // Liberar la conexión
              await connection.close();

              res.json({ success: true, message: 'guardado exitosamente' });
            } catch (error) {
               console.log(error);
              res.status(500).json({ success: false, message: 'Error al guardar' });
            }
          });


}

module.exports.Guardar= Guardar;
module.exports.Modificar= Modificar;
@@ -3643,3 +3702,5 @@ module.exports.GetFechaFirmasEjecutoria=GetFechaFirmasEjecutoria

module.exports.descargarZipEjecutorias=descargarZipEjecutorias

module.exports.Gobernacion=Gobernacion
```

Activar Windows
Ve a Configuración para

routes/titulacion/mit_predio.js

```
 @@ -66,7 +66,6 @@ router.get('/get-historico-ejecutorias', titulacion_router.HistoricoEjecutorias)

 router.delete('/eliminar-ejecutoria/:id', titulacion_router.EliminarEjecutoria);

-
 router.post('/descargar-zip', titulacion_router.descargarZip);

 router.get('/get-historico-ejecutorias-para-gobernacion', titulacion_router.HistoricoEjecutoriasParaGobernacion);
 @@ -83,6 +82,7 @@ router.get('/lista-fechas-aprobadas-ejecutoria', titulacion_router.GetFechaFirma

 router.post('/descargar-ejecutorias-zip', titulacion_router.descargarZipEjecutorias);

+router.post('/registrar-radicado-gobernacion', titulacion_router.Gobernacion);

 module.exports=router;
```

controllers/titulacion/mit_predio.js

```
    }

+   const Gobernacion = async  (req, res) => {
+    oracledb.getConnection(connAttrs, async function (err, connection) {
+        if (err) {
+            // Error connecting to DB
+            res.set('Content-Type', 'application/json');
+            res.status(500).send(JSON.stringify({
+                status: 500,
+                message: "Error connecting to DB",
+                detailed_message: err.message
+            }));
+            return;
+           }
+            try {
+               let binds = [];
+              // Obtener los datos del formulario desde req.body
+              const { GOB_RAD_SIGOB, GOB_FECHA_RAD_SIGOB,FECHA } = req.body;
+
+            // Confirmar la transacción
+
+
+            const sqlQuery =
+              `SELECT * FROM (
+                SELECT
+                PRE_ID,
+                EJE_ID
+                FROM VWS_HISTORICO_EJECUTORIAS  WHERE TO_CHAR(EJE_FECHA_REGISTRO, 'YYYY-MM-DD')='`+FECHA+`'
+              ) ORDER BY EJE_ID DESC`;
+
+           console.log(sqlQuery);
+
+            result = await connection.execute(
+                sqlQuery,
+            [],
+            { resultSet: true, outFormat: oracledb.OUT_FORMAT_OBJECT });
+            console.log(sqlQuery);
+
+             result = await connection.execute(
+                 sqlQuery,
+              [],
+              { resultSet: true, outFormat: oracledb.OUT_FORMAT_OBJECT });
+
+              const rs2 = result.resultSet;
+              let row2;
+              /* LISTA DE PREDIOS */
+              while ((row2 = await rs2.getRow()))
+              {
+                  await connection.execute( `INSERT INTO MIT_GOBERNACIONES
+                  (GOB_ID, GOB_PRED_ID,GOB_RAD_SIGOB,GOB_FECHA_RAD_SIGOB) VALUES
+                   (seq_gobernacion.NEXTVAL,`+row2.PRE_ID+`,'`+GOB_RAD_SIGOB+`','`+GOB_FECHA_RAD_SIGOB+`')`, binds, {autoCommit: true});
+                console.log(`INSERT INTO MIT_GOBERNACIONES(GOB_ID, GOB_PRED_ID,GOB_RAD_SIGOB,GOB_FECHA_RAD_SIGOB) VALUES
  (seq_gobernacion.NEXTVAL,`+row2.PRE_ID+`,'`+GOB_RAD_SIGOB+`','`+GOB_FECHA_RAD_SIGOB+`')`);
+              }
+
+
+                // Liberar la conexión
+                await connection.close();
+
+                res.json({ success: true, message: 'guardado exitosamente' });
+              } catch (error) {
+                  console.log(error);
+                res.status(500).json({ success: false, message: 'Error al guardar' });
+              }
+            });
+
+
+ }

  module.exports.Guardar= Guardar;
  module.exports.Modificar= Modificar;
  @@ -3643,3 +3702,5 @@ module.exports.GetFechaFirmasEjecutoria=GetFechaFirmasEjecutoria

  module.exports.descargarZipEjecutorias=descargarZipEjecutorias

  module.exports.Guardar= Guardar;
  module.exports.Modificar= Modificar;
  @@ -3643,3 +3702,5 @@ module.exports.GetFechaFirmasEjecutoria=GetFechaFirmasEjecutoria

  module.exports.descargarZipEjecutorias=descargarZipEjecutorias

+ module.exports.Gobernacion=Gobernacion
+
```

Activar Windows
Ve a Configuración para

routes/titulacion/mit_predio.js

```
@@ -66,7 +66,6 @@ router.get('/get-historico-ejecutorias', titulacion_router.HistoricoEjecutorias)

router.delete('/eliminar-ejecutoria/:id', titulacion_router.EliminarEjecutoria);

-
router.post('/descargar-zip', titulacion_router.descargarZip);

router.get('/get-historico-ejecutorias-para-gobernacion', titulacion_router.HistoricoEjecutoriasParaGobernacion);
@@ -83,6 +82,7 @@ router.get('/lista-fechas-aprobadas-ejecutoria', titulacion_router.GetFechaFirma

router.post('/descargar-ejecutorias-zip', titulacion_router.descargarZipEjecutorias);

+router.post('/registrar-radicado-gobernacion', titulacion_router.Gobernacion);

module.exports=router;
```

src/components/dashboard/gobernacion/historico_gobernacion.js

```
@@ -21,6 +21,7 @@ import { format } from 'date-fns';
import DeleteIcon from '@mui/icons-material/Delete';
import Swal from 'sweetalert2';
import {ExportToExcel} from '../../ExportToExcel'
+import SaveIcon from '@mui/icons-material/Save';

const CustomCheckbox = withStyles({
  root: {
@@ -69,7 +70,7 @@ const useStyles = makeStyles((theme) => ({
  },
}));

-export const ListaEjecutorias = () => {
+export const ListaGobernacion = () => {
  const [ejecutorias, setEjecutorias] = useState([]);
  const [loading, setLoading] = useState(true);
  const [searchTerm, setSearchTerm] = useState('');
@@ -86,6 +87,25 @@ export const ListaEjecutorias = () => {
  const classes = useStyles();
  const [resolEstado, setResolEstado] = useState(0);

+  const [formData, setFormData] = useState({
+    GOB_PRED_ID: '',
+    GOB_RAD_SIGOB: '',
+    GOB_FECHA_RAD_SIGOB: '',
+    GOB_RAD_GOBERNACION: '',
+    GOB_FECHA_RAD_GOB: '',
+    GOB_FECHA_REGIS: '',
+    GOB_USUARIO_GESTION: 1,
+  });
+
+  const handleInputChange = (event) => {
+    const { name, value } = event.target;
+    setFormData({
+      ...formData,
+      [name]: value,
+    });
```

Activar Windows
Ve a Configuración para

```
      [name]: value,
    });
  };

/*
  const descargarZip = useCallback(async (FECHA) => {
    let data ={
      'FECHA':FECHA
@@ -115,11 +135,12 @@ export const ListaEjecutorias = () => {


  }, [])

  const handleDownload = () => {
    const fileURL = `${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/descarga-anexo/salida_ejecutoria.zip/documento-predio-completo`;
    window.location.href = fileURL;
  };

*/
  const ListaFechas = async () => {
    try {

@@ -171,42 +192,70 @@ export const ListaEjecutorias = () => {
    }
  };

  const eliminarEjecutoria = async (id) => {
  const SIGOBAntesGobernacion = async () => {
    try {
      setLoading(true);
      await axios.delete(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/eliminar-ejecutoria/${id}`);
      // Actualizar la lista de ejecutorias después de eliminar una
      fetchData(0);

      Swal.fire({
        title: 'Eliminado',
        text: 'La ejecutoria ha sido eliminada correctamente',
        icon: 'success',
      });
+         url: `${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/registrar-radicado-gobernacion`,
+         method: 'post',
+         data: JSON.stringify(data),
+         headers: {
+           'Content-Type': 'application/json'
+         }
+       }).then(({ data: res }) => {
+         Swal.fire({
+           title: 'Eliminado',
+           text: 'Se Registro el numero de radicado y fecha de SIGOB correctamente',
+           icon: 'success',
+         });
+         fetchData(0);
+
+       })
+         .catch(error => {
+           console.log(error);
+         });
+
+

    } catch (error) {
      console.log(error);
      // Mostrar una notificación de error
      Swal.fire({
        title: 'Error',
-       text: 'No se pudo eliminar la ejecutoria',
+       text: 'No se pudo registrar la informacioón',
        icon: 'error',
      });
    }
  };

-   const confirmarEliminacion = (id) => {
+   const radicarSIGOBGobernacion = () => {

+     if(formData.GOB_RAD_SIGOB!='' && formData.GOB_RAD_SIGOB!=' ' && formData.GOB_FECHA_RAD_SIGOB!=''){
      Swal.fire({
-       title: 'Confirmar eliminación',
```

Activar Windows
Ve a Configuración para

```
        text: '¿Estás seguro que este es numero de radicado y fecha de SIGOB ?',
        icon: 'warning',
        showCancelButton: true,
        confirmButtonText: 'Eliminar',
        confirmButtonText: 'Confirmar',
        cancelButtonText: 'Cancelar',
      }).then((result) => {
        if (result.isConfirmed) {
          eliminarEjecutoria(id);
          SIGOBAntesGobernacion();
        }
      });
    }else{
      Swal.fire({
        title: 'Error',
        text: 'Indique el numero de radicado y fecha de SIGOB',
        icon: 'error',
      });

    }
    };

  const handleSearch = (event) => {
@@ -250,7 +299,7 @@ export const ListaEjecutorias = () => {
              p: 3
            }}
          >
-           <Grid
+           {/* <Grid
              item
              sm={4}
              xs={12}
@@ -264,26 +313,24 @@ export const ListaEjecutorias = () => {
                  sx={{  minWidth: 350}}

              />
-           </Grid>
+           </Grid> */}
          */}
+             sm={3}
+             xs={3}
+             style={{ paddingRight: '5px' }}
            >

-             <InputLabel variant="standard" htmlFor="uncontrolled-native">
-                 Fecha
+             <InputLabel variant="standard" htmlFor="uncontrolled-native">
+              Fecha Ejecutatoria Para SIGOB
              </InputLabel>
                <NativeSelect
                label="Filtrar por Fecha"
                select
                fullWidth
                value={resolFecha}
-               onChange={handleChangeResolFecha}
-
-               >
+               onChange={handleChangeResolFecha}>
                <option value={0}>Todas</option>
                  {resolFechaLista.length>0 && resolFechaLista.map((option) => (
                    <option value={option.FECHA}>
@@ -294,21 +341,46 @@ export const ListaEjecutorias = () => {
            </Grid>
            <Grid
              item
-             sm={1}
-             xs={12}
+             sm={3}
+             xs={3}
              sx={{
-               minWidth: 250,
+               minWidth: 255,
              }}
              display={ocultar == 0 ? 'none' : ''}
+             style={{ paddingRight: '10px' }}
              >
-             <ExportToExcel   color="#FF5630" opt={3} origen={1} fileName={'lista-predios-gobernacion'} numero_columna={10} responsable={0}
nombre_boton='Generar Listado' variant="subtitle2" estado={110} fecha={resolFecha} />
```

Activar Windows
Ve a Configuración par
Activar Windows
Ve a Configuración para a

src/pages/gobernacion/gobernacion_gestion.js

```
@@ -65,7 +65,7 @@ const gobernacion_gestion = () => {
              >
                <Grid
                  item
-                 xs={8}
+                 xs={12}
                >
                  <Grid
                    item
```

src/pages/gobernacion/historico.js

```
@@ -12,7 +12,7 @@ import {
import KeyboardBackspaceIcon from '@mui/icons-material/KeyboardBackspace';
import { AuthGuard } from '../../components/authentication/auth-guard';
import { DashboardLayout } from '../../components/dashboard/dashboard-layout';
-import { ListaEjecutorias } from '../../components/dashboard/gobernacion/historico_gobernacion';
+import { ListaGobernacion } from '../../components/dashboard/gobernacion/historico_gobernacion';
import { useAuth } from '../../hooks/use-auth';

const HistoricoEjecutoriaGestion = () => {
@@ -44,6 +44,7 @@ const HistoricoEjecutoriaGestion = () => {
        sx={{
          flexGrow: 1
        }}
+       xs={12}
      >
        <Grid
          container
@@ -103,12 +104,12 @@ const HistoricoEjecutoriaGestion = () => {

          <Grid
            item
-           lg={12}
+
            md={12}
            sm={12}
            xs={12}
          >
-           <ListaEjecutorias />
+           <ListaGobernacion />
          </Grid>

        </Grid>
```

Activar Windows

controllers/titulacion/mit_predio.js

```
@@ -1317,8 +1317,8 @@ const login = async (req, res) => {
                const json = JSON.parse(body);
                let binds = [];

-               if(json.data.universocatastrou && json.data.universocatastrou.AREA_GEO!=null){
-                var valor=json.data.universocatastrou.AREA_GEO;
+               if(json.data.universocatastrou && json.data.universocatastrou.AREA_TERRENO!=null){
+                var valor=json.data.universocatastrou.AREA_TERRENO;
                 var direccion=json.data.universocatastrou.DIRECCION;
                 var matricula=json.data.universocatastrou.MATRICULA;

@@ -1338,10 +1338,12 @@ const login = async (req, res) => {

                        }

-                       console.log(PRM_MATRICULA_RES+' ojo  '+matricula);
-                       var porciones = valor.split(',');
-                       console.log( `UPDATE MIT_PREDIO SET PRE_AREA_M2='`+porciones[0]+`' ,PRE_DIRECCION='`+direccion+`' WHERE PRE_REFERENCIA='`+REFERENCIA+`'`);
-                       await connection.execute( `UPDATE MIT_PREDIO SET PRE_AREA_M2='`+porciones[0]+`' ,PRE_DIRECCION='`+direccion+`' WHERE PRE_REFERENCIA='`+REFERENCIA+`'`, binds, {autoCommit: true});
+                       console.log(PRM_MATRICULA_RES+'---- ojo --- '+matricula);
+                       var originalString = valor;
+                       var porciones = originalString.replace(",", ".");
+                       //var porciones = valor.split(',');
+                       console.log( `UPDATE MIT_PREDIO SET PRE_AREA_M2='`+porciones+`' ,PRE_DIRECCION='`+direccion+`' WHERE PRE_REFERENCIA='`+REFERENCIA+`'`);
+                       await connection.execute( `UPDATE MIT_PREDIO SET PRE_AREA_M2='`+porciones+`' ,PRE_DIRECCION='`+direccion+`' WHERE PRE_REFERENCIA='`+REFERENCIA+`'`, binds, {autoCommit: true});
                        connection.close();
                }

@@ -1363,25 +1365,45 @@ const UpdateTodosRegistroPorEscritura = async (req, res) => {
        let TIPO = req.params.TIPO
        let sqlQuery='';
       if(TIPO==1){
-                sqlQuery = `SELECT DISTINCT VWS_PREDIOS.* ,
+                sqlQuery = `SELECT DISTINCT VWS_PREDIOS.* ,
                (SELECT
                CONCAT(
                CONCAT(CONCAT(CEP_NUM_CERTIFICADO,'_'),CEP_REFERENCIA),'.pdf') AS NUM_CER FROM MIT_CERTIFICADO_PREDIAL
                WHERE MIT_CERTIFICADO_PREDIAL.CEP_FK_ID_PREDIO=VWS_PREDIOS.PRE_ID AND CEP_NUM_CERTIFICADO IS NOT NULL AND ROWNUM<=1) AS CERT
                FROM VWS_PREDIOS
                WHERE VWS_PREDIOS.PRE_FK_ESTADO IN (6) AND VWS_PREDIOS.PRE_ID NOT IN(SELECT TEMP_PRE_ID FROM MIT_RESOLUCION_IND_TEMP )`;

    }else{
         if(TIPO==2){
                sqlQuery = `SELECT VWS_RESOLUCIONES_TITULA.*,

    }else if(TIPO==2){
                /*sqlQuery = `SELECT VWS_RESOLUCIONES_TITULA.*,
                (   SELECT
                CONCAT(
                CONCAT(CONCAT(CEP_NUM_CERTIFICADO,'_'),CEP_REFERENCIA),'.pdf') AS NUM_CER FROM MIT_CERTIFICADO_PREDIAL
                WHERE MIT_CERTIFICADO_PREDIAL.CEP_FK_ID_PREDIO=VWS_RESOLUCIONES_TITULA.PRE_ID AND  ROWNUM<=1) AS CERT
                FROM VWS_RESOLUCIONES_TITULA WHERE VWS_RESOLUCIONES_TITULA.TEMP_GENERADA IN(SELECT MIT_ESTADO_APROBACION.EAP_ID FROM MIT_ESTADO_APROBACION )
                AND VWS_RESOLUCIONES_TITULA.EAP_ID IN(4,3)`;
           */
       }else if(TIPO==3){
                sqlQuery = `SELECT
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_PRE_ID,
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_GENERADA,
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_CONSECUTIVO,
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_FECHA,
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_DESDE_REF,
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_USUARIO_LEGAL_GES,
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_IS,
                DBS_IDETEST.MIT_PREDIO.PRE_ID,
                DBS_IDETEST.MIT_PREDIO.PRE_AREA_M2,
                DBS_IDETEST.MIT_PREDIO.PRE_REFERENCIA
                FROM
                DBS_IDETEST.MIT_RESOLUCION_IND_TEMP
                INNER JOIN DBS_IDETEST.MIT_PREDIO ON DBS_IDETEST.MIT_RESOLUCION_IND_TEMP.TEMP_PRE_ID = DBS_IDETEST.MIT_PREDIO.PRE_ID
                WHERE
```

Activar Windov
Ve a Configuración p
Activar Windows
Ve a Configuración para

controllers/titulacion/mit_resolucion.js

```
@@ -27,6 +27,7 @@ var url_catastro='http://catastro.barranquilla.gov.co:9090/';

var QR = require("qrcode-base64");

+ const { InformacionGeneralPredioCatastroUpdate } = require("../../controllers/titulacion/mit_predio");

  // const rutaServerUplod ='C:/servicios-web'; // window
   const rutaServerUplod ='/servicios-web'; // centos
@@ -885,7 +886,7 @@ async function GeneracionResolucionDoc(req, res) {
                SQL_SECUENCIA_TEMP     =`select resolucion_temp.nextval AS SECUENCIA  from dual`;
                rowSQL_SECUENCIA_TEMP  = await connection.execute(SQL_SECUENCIA_TEMP);
                inicio                 = rowSQL_SECUENCIA_TEMP['rows'][0][0];
-
+              //  InformacionGeneralPredioCatastroUpdate(row2.PRE_ID);

                /* PLANTILLA*/
                var MIT_PLANTILLAS_PLA_ID=row2.PRE_ACLARACION==1?4:3;
```

src/components/dashboard/registro/historico_registro.js

```
+ import React, { useState, useEffect,useCallback } from 'react';
+ import axios from 'axios';
+ import {
+   Container,
+   Table,
+   TableBody,
+   TableCell,
+   TableContainer,
+   TableHead,
+   TableRow,
+   Paper,
+   TextField,
+   Pagination,
+   IconButton,
+   CircularProgress,
+   Link,Grid,NativeSelect,InputLabel,Box,Button
+ } from '@mui/material';
+ import { Download as DownloadIcon } from '@mui/icons-material';
+ import { withStyles, makeStyles } from '@mui/styles';
+ import { format } from 'date-fns';
+ import DeleteIcon from '@mui/icons-material/Delete';
+ import Swal from 'sweetalert2';
+ import {ExportToExcel} from '../../ExportToExcel'
+ import SaveIcon from '@mui/icons-material/Save';
+ import BeatLoader from "react-spinners/BeatLoader";
+
+ const CustomCheckbox = withStyles({
+   root: {
+     color: '#3f51b5',
+     '&$checked': {
+       color: '#3f51b5',
+     },
+   },
+   checked: {},
+ })((props) => <Checkbox color="default" {...props} />);
+
+ const useStyles = makeStyles((theme) => ({
```

Activar Windows
Ve a Configuración para

```
  formControl: {
    minWidth: 200,
  },
  tableContainer: {
    marginTop: theme.spacing(2),
    marginBottom: theme.spacing(2),
  },
  pagination: {
    marginTop: theme.spacing(2),
    display: 'flex',
    justifyContent: 'center',
  },
  submitButton: {
    marginTop: theme.spacing(2),
  },
  roundedButton: {
    borderRadius: theme.spacing(2),
    textTransform: 'none',
    fontWeight: 'bold',
    height: '40px', // Ajusta la altura deseada aquí
    transition: 'background-color 0.3s'
  },
  primaryButton: {
    backgroundColor: theme.palette.primary.main,
    color: theme.palette.primary.contrastText, // Agrega una transición suave para el cambio de color de fondo
    height: '40px'
  },
  secondaryButton: {
    backgroundColor: theme.palette.secondary.main,
    color: theme.palette.secondary
  },
}));

export const ListaRegistro = () => {
  const [ejecutorias, setEjecutorias] = useState([]);
  const [loading, setLoading] = useState(true);
  const [searchTerm, setSearchTerm] = useState('');
  const [currentPage, setCurrentPage] = useState(1);
  const [totalPages, setTotalPages] = useState(1);
  };
  const [ocultar, setOcultar] = useState(0);
  const classes = useStyles();
  const [resolEstado, setResolEstado] = useState(0);

  const [formData, setFormData] = useState({
    GOB_PRED_ID: '',
    GOB_RAD_SIGOB: '',
    GOB_FECHA_RAD_SIGOB: '',
    GOB_RAD_GOBERNACION: '',
    GOB_FECHA_RAD_GOB: '',
    GOB_FECHA_REGIS: '',
    GOB_USUARIO_GESTION: 1,
  });

  const handleInputChange = (event) => {
    const { name, value } = event.target;
    setFormData({
      ...formData,
      [name]: value,
    });
  };


  const ListaFechas = async () => {
    try {

      const response = await axios.get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/Lista-fechas-aprobadas-gobernacion`, {
      });
      console.log(response.data);
     setResolFechaLista( response.data);


    } catch (err) {
      console.error(err);
    }
  }
```

Activar Windows
Ve a Configuración para

Activar Windows
Ve a Configuración para

```
const fetchData = async () => {
  // Cancelar la solicitud anterior si existe
  if (cancelTokenSource) {
    cancelTokenSource.cancel('Nueva solicitud realizada');
  }
  // Crear una nueva fuente de cancelación
  cancelTokenSource = axios.CancelToken.source();

  try {
    setLoading(true);
    const response = await axios.get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/get-historico-ejecutorias-para-gobernacion?
page=${currentPage}&search=${searchTerm}&fecha=${resolFecha}`, {
    });
    setEjecutorias(response.data.ejecutorias);
    setTotalPages(response.data.totalPages);
    setLoading(false);



  } catch (error) {
    if (axios.isCancel(error)) {
      console.log('Solicitud cancelada:', error.message);
    } else {
      console.log('Error:', error.message);
    }
  }
};

const SIGOBAntesGobernacion = async () => {
  try {
    setLoading(true);
    let data = {
      'GOB_RAD_SIGOB':  formData.GOB_RAD_SIGOB,
      'GOB_FECHA_RAD_SIGOB':  formData.GOB_FECHA_RAD_SIGOB,
      'FECHA':  resolFecha
    }
```

Activar Windows
Ve a Configuración para

```
      axios({
        url: `${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/registrar-radicado-gobernacion`,
        method: 'post',
        data: JSON.stringify(data),
        headers: {
          'Content-Type': 'application/json'
        }
      }).then(({ data: res }) => {
        Swal.fire({
          title: 'Registrado...',
          text: 'Se Registro el numero de radicado y fecha de SIGOB correctamente',
          icon: 'success',
        });
        setOcultar(0);
        ListaFechas();
        setResolFecha(0);
        setLoading(false);
        fetchData();



      })
        .catch(error => {
          console.log(error);
        });


  } catch (error) {
    console.log(error);
    // Mostrar una notificación de error
    Swal.fire({
      title: 'Error',
      text: 'No se pudo registrar la informacioón',
      icon: 'error',
    });
  }
```

Activar Windows
Ve a Configuración para

```
      icon: 'warning',
      showCancelButton: true,
      confirmButtonText: 'Confirmar',
      cancelButtonText: 'Cancelar',
    }).then((result) => {
      if (result.isConfirmed) {
        SIGOBAntesGobernacion();
        setFormData({
          ...formData,
          ['GOB_RAD_SIGOB']: '',['GOB_FECHA_RAD_SIGOB']: '',
        });
      }
    });
  }else{
    Swal.fire({
      title: 'Error',
      text: 'Indique el numero de radicado y fecha de SIGOB',
      icon: 'error',
    });

  }
  };


  const formatDate = (date) => {
    return format(new Date(date), 'yyyy-MM-dd');
  };
  const handlePageChange = (event, page) => {
    setCurrentPage(page);
  };


  const handleChangeResolFecha= (event) => {
    if(event.target.value=='0'){
      setOcultar(0);
      setResolEstado(0);
    }else{
                value={resolFecha}
                onChange={handleChangeResolFecha}>
                <option value={0}>Todas</option>
                  {resolFechaLista.length>0 && resolFechaLista.map((option) => (
                    <option value={option.FECHA}>
                      {option.FECHA}
                    </option>
                     ))}
                </NativeSelect>
            </Grid>
            <Grid
              item
              sm={3}
              xs={3}
              sx={{
                minWidth: 255,
              }}
              display={ocultar == 0 ? 'none' : ''}
              style={{ paddingRight: '10px' }}
              >
              <ExportToExcel   color="#FF5630" opt={3} origen={1} fileName={'lista-predios-gobernacion'} numero_columna={10} responsable={0}
nombre_boton='Generar Listado para Oficio' variant="subtitle2" estado={110} fecha={resolFecha} />
            </Grid>
                <Grid item xs={2}  style={{ paddingRight: '10px' }}  display={ocultar == 0 ? 'none' : ''}>
                  <TextField
                    label="RADICADO SIGOB"
                    type="text"
                    name="GOB_RAD_SIGOB"
                    value={formData.GOB_RAD_SIGOB}
                    onChange={handleInputChange}
                    fullWidth
                  />
                </Grid>
                <Grid item xs={2}  style={{ paddingRight: '10px' }}  display={ocultar == 0 ? 'none' : ''} >
                  <TextField
                    type="date"
                    name="GOB_FECHA_RAD_SIGOB"
                    value={formData.GOB_FECHA_RAD_SIGOB}
                    onChange={handleInputChange}
```

Activar Windows
Ve a Configuración para

Activar Windows
Ve a Configuración para

```
          </TableRow>
        </TableHead>
        <TableBody>
          {loading ? (
            <TableRow>
              <TableCell colSpan={12} align="center">
                <CircularProgress />
              </TableCell>
            </TableRow>
          ) : (
            ejecutorias.map((ejecutoria) => (
              <TableRow key={ejecutoria.EJE_ID}>
                <TableCell>{tipoNotificacionLabels[ejecutoria.EJE_TIPO_NOTIFICACION]}</TableCell>
                <TableCell>{formatDate(ejecutoria.EJE_FECHA_NOTIFICACION)}</TableCell>
                <TableCell>{formatDate(ejecutoria.EJE_FECHA_EJECUTORIA)}</TableCell>
                <TableCell>{ejecutoria.PRE_REFERENCIA}</TableCell>
                <TableCell>{ejecutoria.TEMP_CONSECUTIVO}</TableCell>
                <TableCell>{ejecutoria.FECHA}</TableCell>
                <TableCell>{formatDate(ejecutoria.EJE_FECHA_REGISTRO)}</TableCell>

              </TableRow>
            ))
          )}
        </TableBody>
      </Table>
    </TableContainer>
    <Pagination
      count={totalPages}
      page={currentPage}
      onChange={handlePageChange}
      color="primary"
    />
  </Container>
  </div>


  );
};
```

Activar Windows
Ve a Configuración para

src/components/dashboard/registro/historico_registro_radicado.js

```
import React, { useState, useEffect,useCallback } from 'react';
import axios from 'axios';
import {
  Container,
  Table,
  TableBody,
  TableCell,
  TableContainer,
  TableHead,
  TableRow,
  Paper,
  TextField,
  Pagination,
  IconButton,
  CircularProgress,
  Link,Grid,NativeSelect,InputLabel,Box,Button,Typography
} from '@mui/material';
import { Download as DownloadIcon } from '@mui/icons-material';
import { withStyles, makeStyles } from '@mui/styles';
import { format } from 'date-fns';
import DeleteIcon from '@mui/icons-material/Delete';
import Swal from 'sweetalert2';
import {ExportToExcel} from '../../ExportToExcel'
import SaveIcon from '@mui/icons-material/Save';
import CloudUploadIcon from '@mui/icons-material/CloudUpload';
import BeatLoader from "react-spinners/BeatLoader";

const CustomCheckbox = withStyles({
  root: {
    color: '#3f51b5',
    '&$checked': {
      color: '#3f51b5',
    },
  },
  checked: {},
})((props) => <Checkbox color="default" {...props} />);
```

Activar Windows
Ve a Configuración para

```
+ }))((props) => <Checkbox color="default" {...props} />);
+
+ const useStyles = makeStyles((theme) => ({
+   root: {
+     margin: theme.spacing(2),
+   },
+   formControl: {
+     minWidth: 200,
+   },
+   tableContainer: {
+     marginTop: theme.spacing(2),
+     marginBottom: theme.spacing(2),
+   },
+   pagination: {
+     marginTop: theme.spacing(2),
+     display: 'flex',
+     justifyContent: 'center',
+   },
+   submitButton: {
+     marginTop: theme.spacing(2),
+   },
+   roundedButton: {
+     borderRadius: theme.spacing(2),
+     textTransform: 'none',
+     fontWeight: 'bold',
+     height: '40px', // Ajusta la altura deseada aquí
+     transition: 'background-color 0.3s'
+   },
+   primaryButton: {
+     backgroundColor: theme.palette.primary.main,
+     color: theme.palette.primary.contrastText, // Agrega una transición suave para el cambio de color de fondo
+     height: '40px'
+   },
+   secondaryButton: {
+     backgroundColor: theme.palette.secondary.main,
+     color: theme.palette.secondary
+   },
+ }));
+
+ export const ListaGobernacion = () => {
+   const [ejecutorias, setEjecutorias] = useState([]);
+   const [loading, setLoading] = useState(true);
+   const [searchTerm, setSearchTerm] = useState('');
+   const [currentPage, setCurrentPage] = useState(1);
+   const [totalPages, setTotalPages] = useState(1);
+   const [resolFechaLista, setResolFechaLista] = useState([]);
+   const [resolFecha, setResolFecha] = useState(0);
+   const tipoNotificacionLabels = {
+     1: 'Personal',
+     2: 'Aviso',
+     3: 'Correo electrónico',
+   };
+   const [ocultar, setOcultar] = useState(0);
+   const classes = useStyles();
+   const [resolEstado, setResolEstado] = useState(0);
+   const [selectedDocumentos, setSelectedDocumentos] = useState([]);
+
+   const [formData, setFormData] = useState({
+     GOB_PRED_ID: '',
+     GOB_RAD_SIGOB: '',
+     GOB_FECHA_RAD_SIGOB: '',
+     GOB_RAD_GOBERNACION: '',
+     GOB_FECHA_RAD_GOB: '',
+     GOB_FECHA_REGIS: '',
+     GOB_USUARIO_GESTION: 1,
+     documentos: selectedDocumentos,
+     FECHA:''
+   });
+
+   const handleInputChange = (event) => {
+     const { name, value } = event.target;
+     setFormData({
+       ...formData,
+       [name]: value,
+     });
+   };
```

Activar Windows
Ve a Configuración para

```
    try {

      const response = await axios.get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/lista-fechas-radicado-sigob`, {
      });
     setResolFechaLista( response.data);


    } catch (err) {
      console.error(err);
    }
  }


  let cancelTokenSource = null;

  useEffect(() => {
    ListaFechas();
    fetchData();

  }, [currentPage, searchTerm,resolFecha]);


  const fetchData = async () => {
    // Cancelar la solicitud anterior si existe
    if (cancelTokenSource) {
      cancelTokenSource.cancel('Nueva solicitud realizada');
    }
    // Crear una nueva fuente de cancelación
    cancelTokenSource = axios.CancelToken.source();

    try {
      setLoading(true);
      const response = await axios.get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/get-historico-radicar-gobernacion?
page=${currentPage}&search=${searchTerm}&fecha=${resolFecha}`, {
      });
      setEjecutorias(response.data.ejecutorias);
      setTotalPages(response.data.totalPages);
      setLoading(false);

+      setLoading(false);
+
+    } catch (error) {
+      if (axios.isCancel(error)) {
+        console.log('Solicitud cancelada:', error.message);
+      } else {
+        console.log('Error:', error.message);
+      }
+    }
+  };
+
+  const handleSubmit = async (event) =>
+  {
+    event.preventDefault();
+    if (selectedDocumentos.length > 0 && formData.GOB_RAD_GOBERNACION!='' && formData.GOB_RAD_GOBERNACION!=' ' && formData.GOB_FECHA_RAD_GOB!='')
+    {
+          try {
+
+            Swal.fire({
+              title: 'Confirmar Radicado de Gobernación',
+              text: '¿Estás seguro de registrar el Número de radicado y fecha de Radicado de Gobernación ?',
+              icon: 'warning',
+              showCancelButton: true,
+              confirmButtonText: 'Registrar Cerificado Gobernación ',
+              cancelButtonText: 'Cancelar',
+            }).then(async (result)  => {
+              if (result.isConfirmed) {
+                setLoading(true);
+
+                formData.documentos=selectedDocumentos[0];
+                formData.FECHA=resolFecha;
+
+                // Aquí se realiza el llamado al servicio web para guardar la información
+                // y enviar los documentos al servidor
+                const response = await axios.post(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/subir-certificado-gobernacion-
 radicado`, formData, {
+                  headers: {
+                    'Content-Type': 'multipart/form-data'
+                  },
```

Activar Windows
Ve a Configuración para

```
          fetchData(0);


            setLoading(false);
            formData.GOB_RAD_GOBERNACION='';
            formData.GOB_FECHA_RAD_GOB='';
            setResolFecha(0);
            setOcultar(0);
            setSelectedDocumentos([]);


          }
        });
        } catch (error) {
          Swal.fire({
            icon: 'error',
            title: 'Error',
            text: 'Ocurrió un error al guardar la información.',
          });
        }

    //FIN PARA GENERAR UNO A UNO


  }else{
    Swal.fire({
      icon: 'error',
      title: 'Error',
      text: 'Debe diligenciar todo los campos por favor.',
    });

    }


};

const formatDate = (date) => {
```

Activar Windows
Ve a Configuración para

```
const handlePageChange = (event, page) => {
  setCurrentPage(page);
};

const downloadArchivo = (url) => {
  window.open(url, '_blank');
};



const handleChangeResolFecha= (event) => {
  if(event.target.value=='0'){
    setOcultar(0);
    setResolEstado(0);
  }else{
    setOcultar(1);
    setResolEstado(0);

  }
  setResolFecha(event.target.value);
};

const handleDocumentoChange = (event) => {
  const selectedFiles = Array.from(event.target.files);

  setSelectedDocumentos(selectedFiles);


};
const handleDeleteDocumento = (documento) => {
  setSelectedDocumentos((prevSelectedDocumentos) =>
    prevSelectedDocumentos.filter((doc) => doc.name !== documento.name)
  );
};


return (

  <div className={classes.root}  style={{ backgroundColor: '#ffffff', padding: '20px' }}>
```

Activar Windows
Ve a Configuración para

```
    <Box
            sx={{
              alignItems: 'center',
              display: 'flex',
              flexWrap: 'wrap',
              p: 3
            }}
          >

           <Grid
              item
              sx={{  minWidth: 150}}
              sm={3}
              xs={3}
              style={{ paddingRight: '5px' }}
            >
              <InputLabel variant="standard" htmlFor="uncontrolled-native">
               Fecha Radicado SIGOB
              </InputLabel>
                <NativeSelect
                label="Filtrar por Fecha"
                select
                fullWidth
                value={resolFecha}
                onChange={handleChangeResolFecha}>
                <option value={0}>Todas</option>
                  {resolFechaLista.length>0 && resolFechaLista.map((option) => (
                    <option value={option.FECHA}>
                      {option.FECHA}
                    </option>
                     ))}
                </NativeSelect>
            </Grid>
            <Grid
              item
              sm={3}
              xs={3}
              sx={{
                minWidth: 225
             Certificado Gobernación
            </Button>
            </InputLabel>
            <input
              type="file"
              id="documento"
              accept=".pdf"
              onChange={handleDocumentoChange}
              style={{ display: 'none' }}

            />


            </Grid>


                <Grid item xs={2}  style={{ paddingRight: '10px' }}  display={ocultar == 0 ? 'none' : ''}>
                  <TextField
                    label="RADICADO GOBERNACIÓN"
                    type="text"
                    name="GOB_RAD_GOBERNACION"
                    value={formData.GOB_RAD_GOBERNACION}
                    onChange={handleInputChange}
                    fullWidth
                  />
                </Grid>
                <Grid item xs={2}  style={{ paddingRight: '10px' }}  display={ocultar == 0 ? 'none' : ''} >
                  <TextField
                    type="date"
                    name="GOB_FECHA_RAD_GOB"
                    value={formData.GOB_FECHA_RAD_GOB}
```

Activar Windows
Ve a Configuración para

Activar Windows
Ve a Configuración par

```
                <Button variant="contained" color="secondary"  type="submit" startIcon={<SaveIcon />}>
                  { loading &&
                  <BeatLoader color={'#FF304F'} size={12}  />
                  }

                  Guardar</Button>


                </Grid>


            <Grid item xs={12} fullWidth>

             <Table fullWidth>

                  <TableBody fullWidth>
                    {selectedDocumentos.map((documento, index) => (

                        <TableRow >
                        <TableCell>

                        <Typography variant="body1">
                          {documento.name}
                        </Typography>
                        </TableCell>
                        <TableCell>
                        <Button
                          variant="outlined"
                          color="error"
                          size="small"
                          startIcon={<DeleteIcon />}
                          onClick={() => handleDeleteDocumento(documento)}
                          sx={{ borderRadius: '9999px' }}
                        >
                          Eliminar
                        </Button>
                        </TableCell>
                        </TableRow>

        </TableHead>
        <TableBody>
          {loading ? (
            <TableRow>
              <TableCell colSpan={12} align="center">
                <CircularProgress />
              </TableCell>
            </TableRow>
          ) : (
            ejecutorias.map((ejecutoria) => (
              <TableRow key={ejecutoria.EJE_ID}>
                <TableCell>{formatDate(ejecutoria.GOB_FECHA_RAD_SIGOB)}</TableCell>
                <TableCell>{ejecutoria.GOB_RAD_SIGOB}</TableCell>
                <TableCell>{ejecutoria.PRE_REFERENCIA}</TableCell>
                <TableCell>{ejecutoria.TEMP_CONSECUTIVO}</TableCell>
                <TableCell>{ejecutoria.FECHA}</TableCell>
                <TableCell>{formatDate(ejecutoria.GOB_FECHA_REGIS)}</TableCell>

              </TableRow>
            ))
          )}
        </TableBody>
      </Table>
    </TableContainer>
    <Pagination
      count={totalPages}
      page={currentPage}
      onChange={handlePageChange}
      color="primary"
    />
  </Container>
  </form>
  </div>


  );
};
```

Activar Windows
Ve a Configuración para

Activar Windows
Ve a Configuración para

src/components/dashboard/registro/registro_form.js

```jsx
/* eslint-disable react/jsx-max-props-per-line */
import { useEffect, useState } from 'react';
import axios from 'axios';
import {
  Button,
  Checkbox,
  FormControlLabel,
  InputLabel,
  Paper,
  CircularProgress,
  Table,
  TableBody,
  TableCell,
  TableContainer,
  TableHead,
  TableRow,
  TextField,NativeSelect,Typography ,Card,Grid ,FormControl, Radio, RadioGroup
} from '@mui/material';
import CloudUploadIcon from '@mui/icons-material/CloudUpload';
import DeleteIcon from '@mui/icons-material/Delete';
import Swal from 'sweetalert2';
import { withStyles, makeStyles } from '@mui/styles';
import Pagination from '@mui/lab/Pagination';
import { format } from 'date-fns';
import {ExportToExcel} from '../../ExportToExcel'

const CustomCheckbox = withStyles({
  root: {
    color: '#3f51b5',
    '&$checked': {
      color: '#3f51b5',
    },
  },
  checked: {},
})((props) => <Checkbox color="default" {...props} />);

const useStyles = makeStyles((theme) => ({
  root: {
    minWidth: 200,
  },
  tableContainer: {
    marginTop: theme.spacing(2),
    marginBottom: theme.spacing(2),
  },
  pagination: {
    marginTop: theme.spacing(2),
    display: 'flex',
    justifyContent: 'center',
  },
  submitButton: {
    marginTop: theme.spacing(2),
  },
  roundedButton: {
    borderRadius: theme.spacing(2),
    textTransform: 'none',
    fontWeight: 'bold',
    height: '40px', // Ajusta la altura deseada aquí
    transition: 'background-color 0.3s'
  },
  primaryButton: {
    backgroundColor: theme.palette.primary.main,
    color: theme.palette.primary.contrastText, // Agrega una transición suave para el cambio de color de fondo
    height: '40px'
  },
  secondaryButton: {
    backgroundColor: theme.palette.secondary.main,
    color: theme.palette.secondary
  },
}));

export const GobernacionForm = () =>
{

  const [tipoNotificacion, setTipoNotificacion] = useState('');
  const [selectedPreIds, setSelectedPreIds] = useState([]);
  const [fechaNotificacion, setFechaNotificacion] = useState('');
  const [fechaEjecutoria, setFechaEjecutoria] = useState('');
```

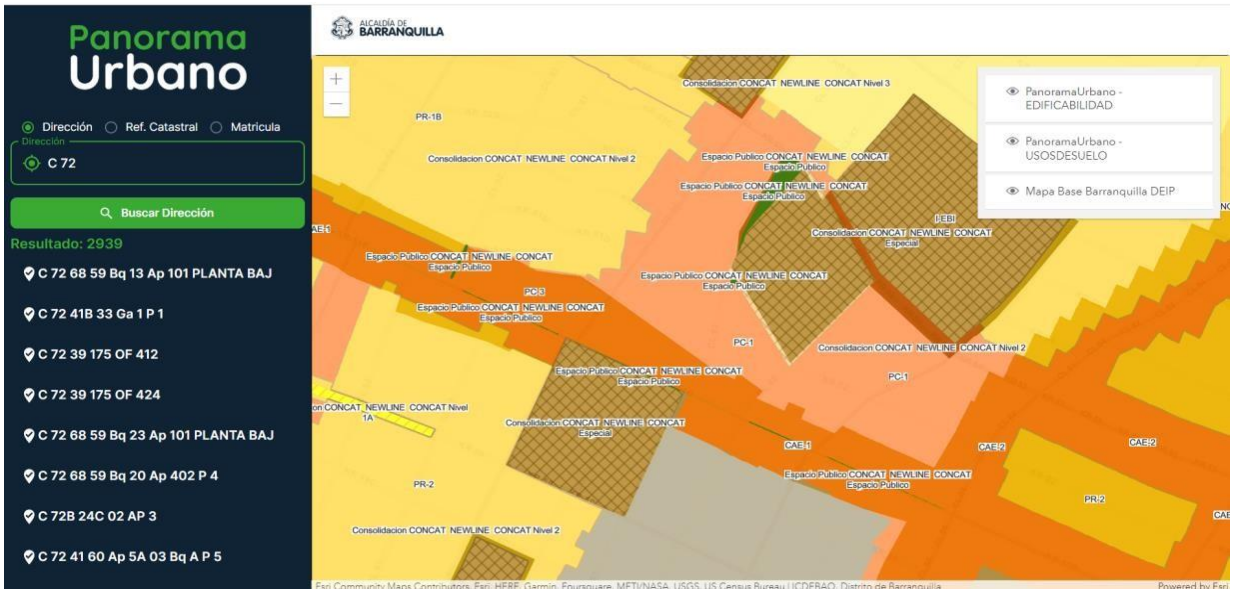
Activar Windows
Ve a Configuración para

```
+  const [loading, setLoading] = useState(true);
+  const [selectedOption, setSelectedOption] = useState(null);
+  const handleOptionChange = (event) => {
+    setSelectedOption(event.target.value);
+  };
+  const classes = useStyles();
+
+  const [formErrors, setFormErrors] = useState({
+    tipoNotificacion: false,
+    fechaNotificacion: false,
+    fechaEjecutoria: false,
+    documentos: false,
+    predioIds: false,
+  });
+
+
+ function listar_predios(){
+    setLoading(true);
+      axios({
+        url:`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/get-historico-ejecutorias-para-gobernacion`,
+        method:'get'
+      }).then(({ data: res }) => {
+
+      if (res.length > 0) {
+        setPredios(res)
+        setFilteredPredios(res)
+
+      }else{
+        Swal.fire("No hay Predio ", {
+          icon: "error"
+        });
+
+      }
+      setLoading(false);
+    })
+
+ }
+
+
```

Activar Windows
Ve a Configuración par

- **Panorama Urbano.**

Evidencias funcionales:



Evidencias técnicas.

src/components/dashboard/dashboard-navbar.js

```
@@ -27,7 +27,7 @@ import { NotificationsPopover } from './notifications-popover';
import { LanguagePopover } from './language-popover';
import { LogoAlcaldia2 } from '../../images/logo-alcaldia.svg';

- const LogoAlcaldia = '/static/logo-alcaldia.svg'
+ const LogoAlcaldia = '/panoramaurbano/static/logo-alcaldia.svg'

const languages = {
  en: '/static/icons/uk_flag.svg',
```

src/components/dashboard/dashboard-sidebar.js

```
@@ -48,7 +48,7 @@ import axios from 'axios';
import ClipLoader from "react-spinners/ClipLoader";
import BeatLoader from "react-spinners/BeatLoader";

- const LogoAlcaldia = '/static/planeacion-white.png'
+ const LogoAlcaldia = '/panoramaurbano/static/planeacion-white.png'

const override = {
  display: "block",
@@ -156,9 +156,11 @@ export const DashboardSidebar = (props) => {
  const [numBisGen, setNumBisGen] = useState('');
  const [complemento, setComplemento] = useState('');
  const [direccion, setDireccion] = useState('');
+  const [predioReferencia, setPredioReferencia] = useState({"data":{}});
  const [referencia, setReferencia] = useState('');
  const [destino, setDestino] = useState('');
  const [selectedActividad, setSelectedActividad] = useState('');
+  const [entra, setEntra] = useState(false);

  const handleChange = (event) => {setSelectedActividad(event.target.value)};
  const handleOpenDireccion = () => setOpenModalDireccion(true);
@@ -285,10 +285,10 @@ export const DashboardSidebar = (props) => {
    return [...new Map(arr.map(item => [item[key], item])).values()]
  }

-  const buscarRef = () => {
+  const buscarRef = () => {
    setLoading(true)
    setInfoDisplay('none')
-    setReferencia(document.getElementById('referencia_sel').value);
+    buscarRefButt(document.getElementById('referencia_sel').value);
    buscarReferencia(document.getElementById('referencia_sel').value, function(result){
```

Activar Windc
Ve a Configuración

```
@@ -307,14 +309,6 @@ export const DashboardSidebar = (props) => {
        setDireccion(predio.attributes.direccion)
        setDestino(predio.attributes.destino)

-       // let listaActividad = []
-
-                           // data.features.forEach(function(feature){
-                           //      let act = { "REFERENCIA":'',"ACTIVIDAD":feature.attributes.ACTIVIDAD, "NOMBRE_USO":
 feature.attributes.NOMBRE_USO, "POLIGONO": feature.attributes.POLIGONO}
-       //   setListaActividades(current => [...current, act]);
-                           // });
-       // console.log(predio)
-       //setListaActividades(eliminarRepetidosLista(listaActividades, 'POLIGONO'))

        mapa.goTo({
          geometry:predio.geometry, zoom:18
@@ -354,11 +348,38 @@ export const DashboardSidebar = (props) => {
        return;
      }
      setListaActividades([{ "REFERENCIA":'',"ACTIVIDAD":'Ninguna', "NOMBRE_USO": '', "POLIGONO": ''}]);
+     const config = {
+       headers: {
+         'Access-Control-Allow-Origin': '*',
+         'Content-Type': 'application/json',
+       },
+     };
+     axios
+     .get(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN2}consulta/referencia/productos/${referencia}`, {}, config)
+     .then(({ data: res }) => {
+       console.log(res.data.universocatastrou.DIRECCION);
+       setPredioReferencia(res);
+       setEntra(true);
+       setReferencia(referencia);
+
```

Activar Window
Ve a Configuración pa

```
+                   id="radio_select_referencia"
+                   value="b"
+                   name="radio-buttons"
+                   inputProps={{ 'aria-label': 'B' }}
+                   sx={{ '& .MuiSvgIcon-root': { fontSize: 18 } }}
+                 />
+               }
+               label={<Typography fontSize={14}>Matricula</Typography>}
+             />
            </RadioGroup>
          </Grid>
          <Grid
@@ -667,6 +706,8 @@ export const DashboardSidebar = (props) => {
                </Typography>
              )}
            </Grid>
+
+
            {
              listaDirecciones.data.map((item, i) => (
              <Grid
@@ -674,13 +715,18 @@ export const DashboardSidebar = (props) => {
                xs={12}
                key={item.REFERENCIA}
              >
+               {/*
+
+               lista de direcciones
+
+               */}
                <Button
                  variant="text"
                  size="large"
                  color="secondary"
                  startIcon={<WhereToVoteIcon />}
```

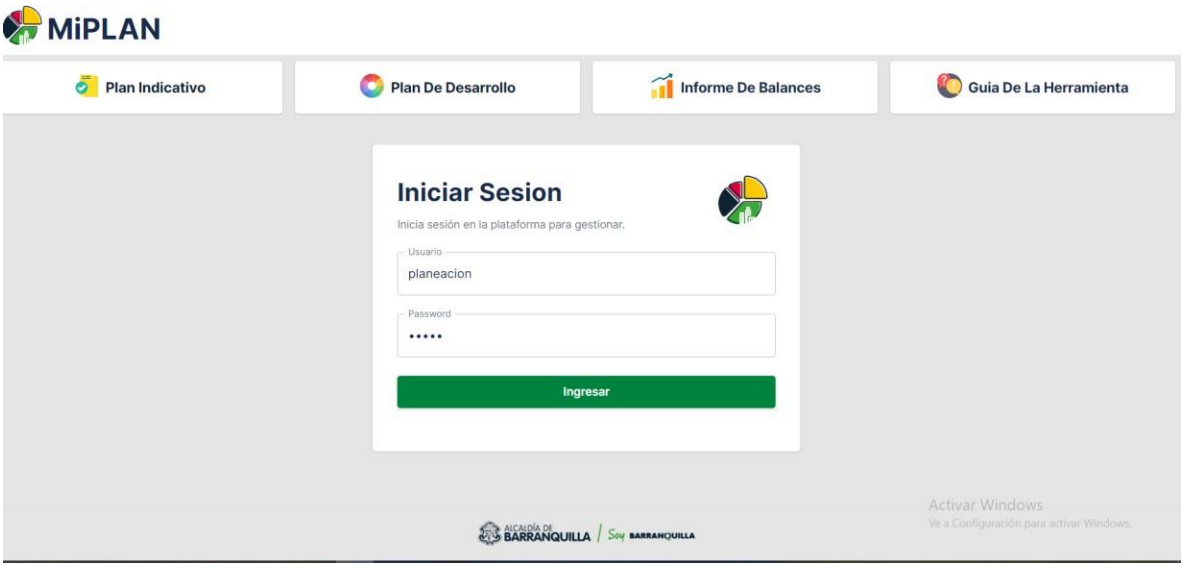
Activar Window
Ve a Configuración pa

```
                <Button
                  variant="text"
                  size="large"
                  color="secondary"
                  startIcon={<WhereToVoteIcon />}
                  sx={{ display: resultado === 'Cargando lista de dirección encontradas…' ? 'none' : '', fontSize: 16, color: 'white', "&
.MuiButton-startIcon": { marginRight: "0px", color: 'white' } }}
-                 onClick={() => {buscarRefButt(item.REFERENCIA.substring(5,30))}}
+                 onClick={() => {buscarRefButt(item.REFERENCIA.substring(5,30))}}
                >
                  {item.DIRECCION}
                </Button>
@@ -765,6 +811,7 @@ export const DashboardSidebar = (props) => {
                </Typography>
              </Grid>
            </Grid>
+
            <Grid
              container
              direction="row"
@@ -773,6 +820,8 @@ export const DashboardSidebar = (props) => {
              sx={{ px: 1 }}
              display={infoDisplay}
            >
+
+
              <Grid item>
                <Typography
                  fontSize={12}
@@ -785,7 +834,10 @@ export const DashboardSidebar = (props) => {
                  fontWeight="bold"
                  fontSize={14}
                >
-                 CL 123 45 67
+
+                 predioReferencia.data.universocatastrou.DIRECCION
+                }
                </Typography>
              </Grid>
              <Grid item>
@@ -800,7 +852,10 @@ export const DashboardSidebar = (props) => {
                  fontWeight="bold"
                  fontSize={14}
                >
-                 {referencia}
+                 {entra &&
+
+                   referencia
+                 }
                </Typography>
              </Grid>
              <Grid item>
@@ -808,13 +863,17 @@ export const DashboardSidebar = (props) => {
                  fontSize={12}
                  color="textSecondary"
                >
-                 Barrio:
+                 Destino:
                </Typography>
                <Typography
                  variant="body2"
                  fontWeight="bold"
                  fontSize={14}
                >
+                  {entra &&
+                      predioReferencia.data.universocatastrou.DESTINO
+                   }
+
                </Typography>
              </Grid>
```

- **Miplan.**

Evidencias funcionales:

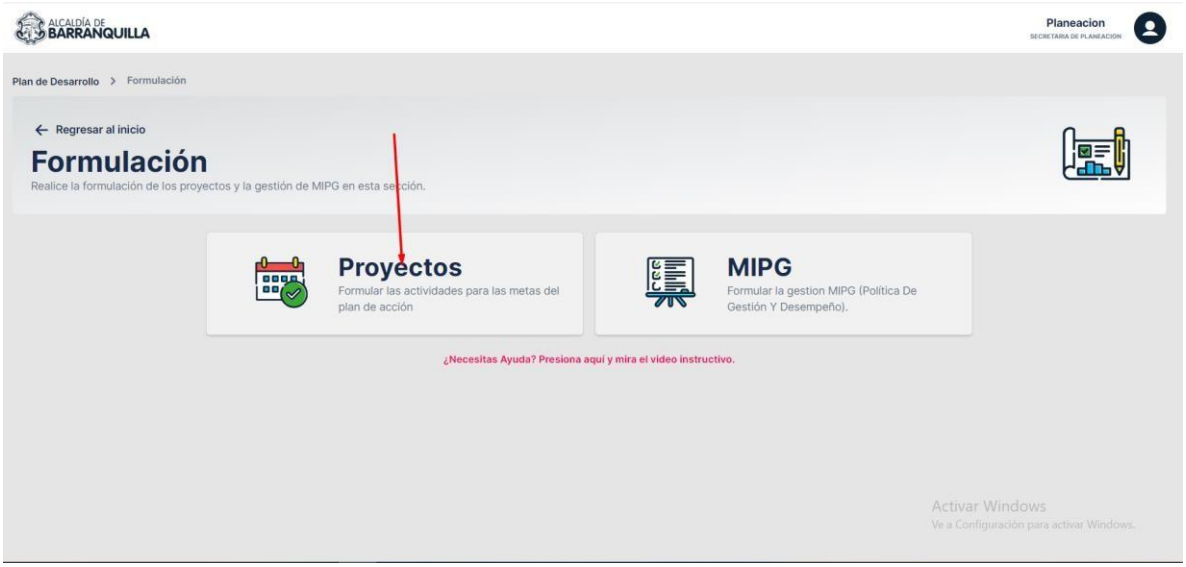
Se ingresa con el usuario.



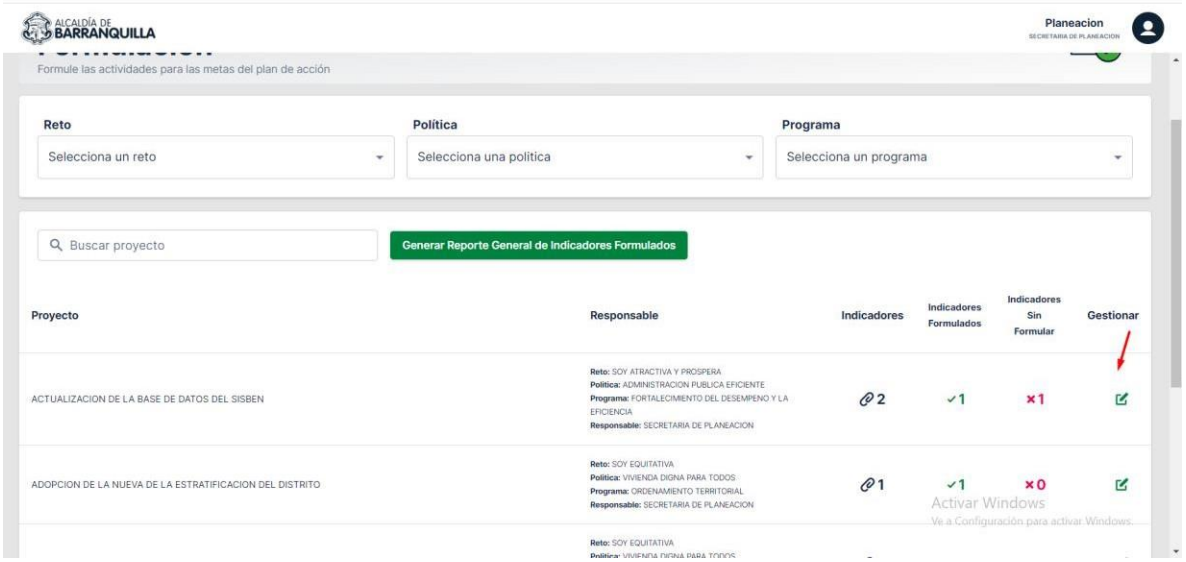
Se selecciona la opción de formulación.



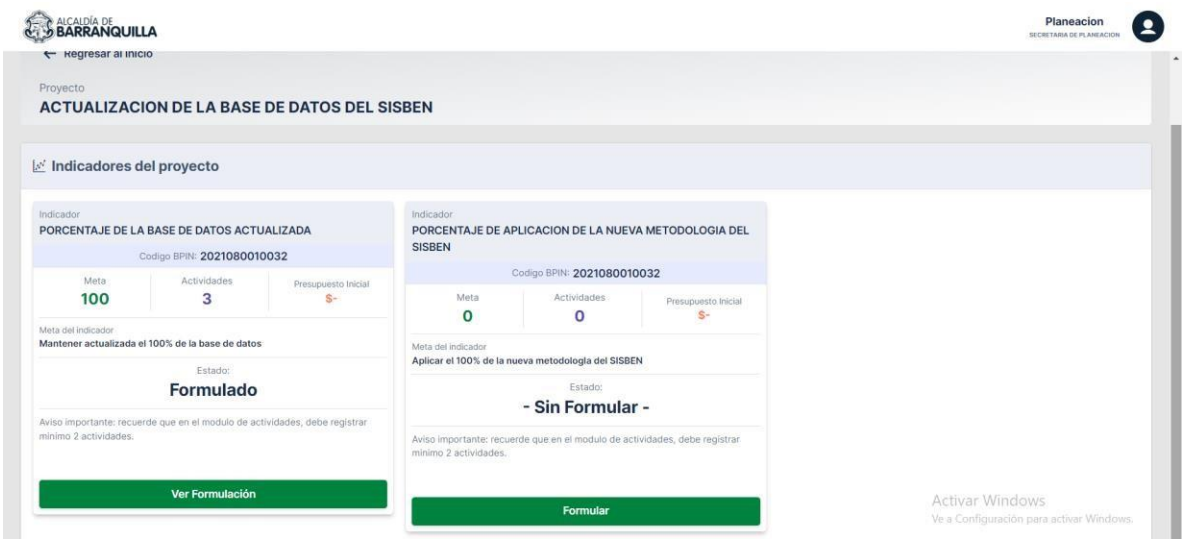
Realizar la formulación de los proyectos.



Click en seleccionar.



Indicadores de proyecto.

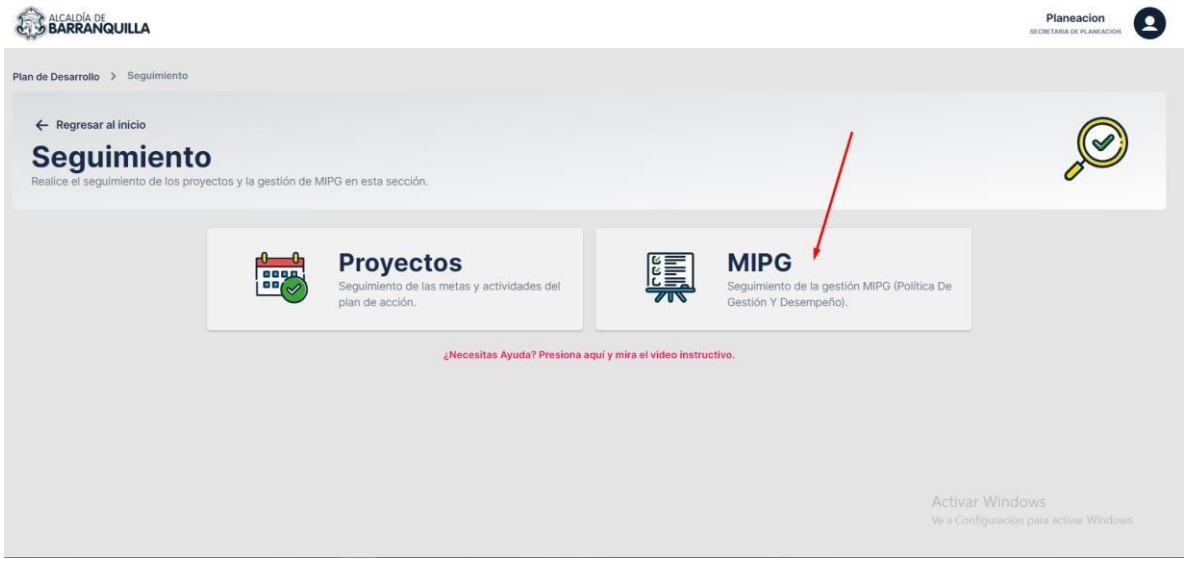


Formulación de proyecto.

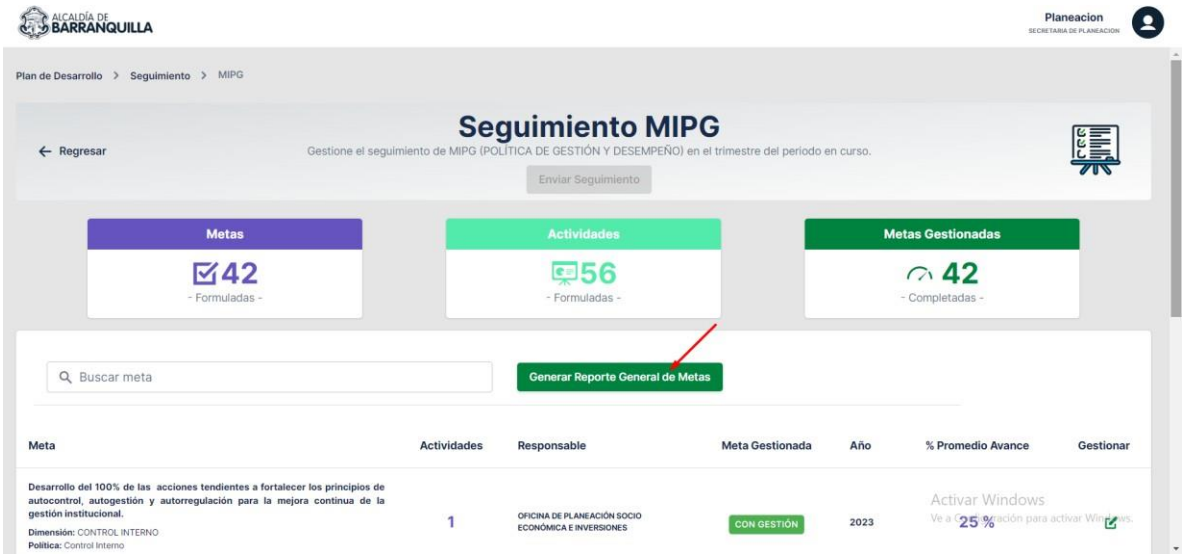


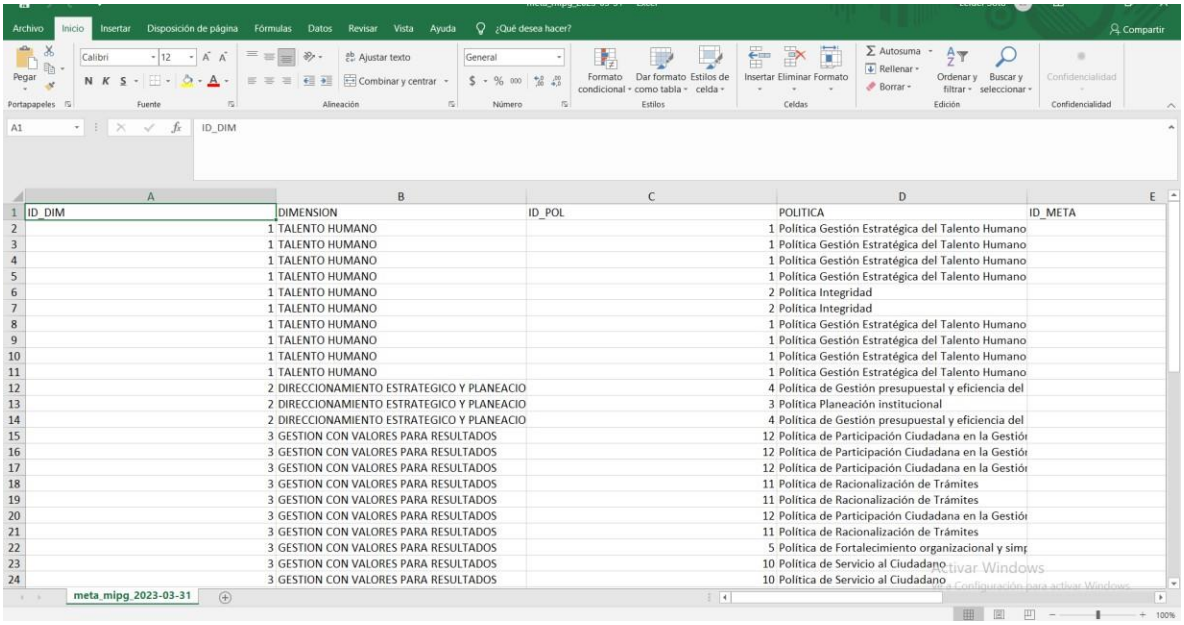


MIPEG.



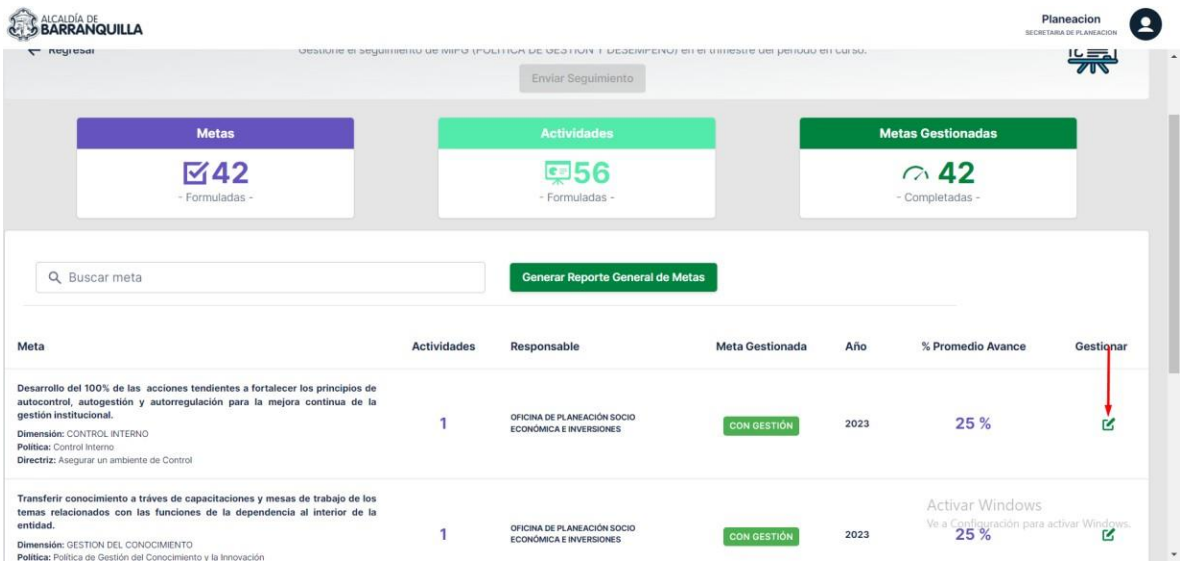
Generar reporte general de metas.



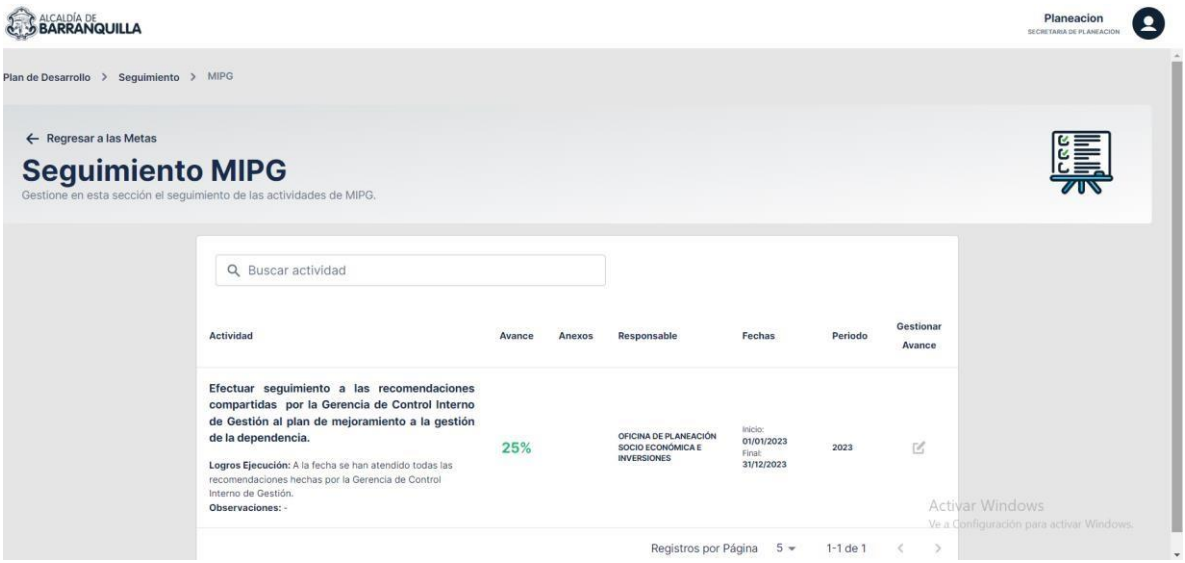


	A	B	C	D	E
1	ID_DIM	DIMENSION	ID_POL	POLITICA	ID_META
2		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
3		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
4		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
5		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
6		1 TALENTO HUMANO	2	Política Integridad	
7		1 TALENTO HUMANO	2	Política Integridad	
8		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
9		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
10		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
11		1 TALENTO HUMANO	1	Política Gestión Estratégica del Talento Humano	
12		2 DIRECCIONAMIENTO ESTRATEGICO Y PLANEACIO	4	Política de Gestión presupuestal y eficiencia del	
13		2 DIRECCIONAMIENTO ESTRATEGICO Y PLANEACIO	3	Política Planeación institucional	
14		2 DIRECCIONAMIENTO ESTRATEGICO Y PLANEACIO	4	Política de Gestión presupuestal y eficiencia del	
15		3 GESTION CON VALORES PARA RESULTADOS	12	Política de Participación Ciudadana en la Gestió	
16		3 GESTION CON VALORES PARA RESULTADOS	12	Política de Participación Ciudadana en la Gestió	
17		3 GESTION CON VALORES PARA RESULTADOS	12	Política de Participación Ciudadana en la Gestió	
18		3 GESTION CON VALORES PARA RESULTADOS	11	Política de Racionalización de Trámites	
19		3 GESTION CON VALORES PARA RESULTADOS	11	Política de Racionalización de Trámites	
20		3 GESTION CON VALORES PARA RESULTADOS	12	Política de Participación Ciudadana en la Gestió	
21		3 GESTION CON VALORES PARA RESULTADOS	11	Política de Racionalización de Trámites	
22		3 GESTION CON VALORES PARA RESULTADOS	5	Política de Fortalecimiento organizacional y simp	
23		3 GESTION CON VALORES PARA RESULTADOS	10	Política de Servicio al Ciudadano	
24		3 GESTION CON VALORES PARA RESULTADOS	10	Política de Servicio al Ciudadano	

Gestionar.



Se gestiona en esta sección el seguimiento de las actividades.



Evidencias técnicas.

controllers/mi-plan/pd-actividades.js

```
@@ -201,18 +201,18 @@ const ActualizarActividadSeguimiento= async (req, res) => {
        ACT_DIA=``+ACT_DIA+`'
        WHERE ACT_ID=`+ACT_ID+``;

-
+       console.log(sqlQuery);
        binds = [];
-       await connection.execute( sqlQuery, binds, {autoCommit: true});
+       // await connection.execute( sqlQuery, binds, {autoCommit: true});


-       res.header('Access-Control-Allow-Origin','*');
+       /*res.header('Access-Control-Allow-Origin','*');
        res.header('Access-Control-Allow-Headers','Content-Type');
        res.header('Access-Control-Allow-Methods','GET,PUT,POST,DELETE,OPTIONS');
        res.contentType('application/json').status(200);
        res.send(JSON.stringify(200));
-
-       /* connection.execute(sqlQuery, binds, {autoCommit: true}, function (err) {
+  */
+       connection.execute(sqlQuery, binds, {autoCommit: true}, function (err) {
            if (err) {
                console.log(err)
                res.set('Content-Type', 'application/json');
@@ -240,7 +240,7 @@ const ActualizarActividadSeguimiento= async (req, res) => {
                }
            });
-       }); */
+       });
    });
```

Activar Windows
Ve a Configuración pa

controllers/mi-plan/pd-dependencias.js

```
@@ -320,10 +320,11 @@ const GetDependenciasRevisionMipg = async (req, res) => {
        `;
        const rowsPlantilla = await connection.execute(SQL_TRIMESTRE);
        year       = rowsPlantilla['rows'][0][0];
-
+        let CTR_MES = rowsPlantilla['rows'][0][1];
+        let CTR_DIA = rowsPlantilla['rows'][0][2];

        let query='SELECT (SELECT COUNT(PDG_CTROL_INT.CTR_FK_POLITICA)  FROM  PDG_CTROL_INT WHERE '+
-            'PDG_CTROL_INT.CTR_FK_RESPONSABLE=RESP.RES_ID AND PDG_CTROL_INT.CTR_ANO=2022 AND PDG_CTROL_INT.CTR_MES=6 AND
PDG_CTROL_INT.CTR_DIA=30 ) AS CAT_POL_AUD ,'+
+            'PDG_CTROL_INT.CTR_FK_RESPONSABLE=RESP.RES_ID AND PDG_CTROL_INT.CTR_ANO='+year+' AND PDG_CTROL_INT.CTR_MES='+CTR_MES+' AND
PDG_CTROL_INT.CTR_DIA='+CTR_DIA+' ) AS CAT_POL_AUD ,'+
            'RESP.RES_ID, RESP.RES_NOMBRE, ' +
            '(SELECT COUNT(PDG_META.MTA_ID) AS CANTIDAD_META FROM PDG_META ' +
            'INNER JOIN PD_RESPONSABLE ON PD_RESPONSABLE.RES_ID = PDG_META.MTA_FKRESPONSABLE ' +
```

controllers/mi-plan/pd-vista-general.js

```
@@ -1865,6 +1865,7 @@ const GetReporteAuditoriaInd = async (req, res) => {
        `;
        const rowsPlantilla = await connection.execute(SQL_TRIMESTRE);
        year       = rowsPlantilla['rows'][0][0];
+        let mes       = rowsPlantilla['rows'][0][1];

        if(RES_ID === idAdmin || RES_ID ===controlinterno){
        sqlQueryFiltro = " WHERE PD_PROY_INDICADOR.PIN_ANO = "+year+" ";
@@ -1880,14 +1881,14 @@ const GetReporteAuditoriaInd = async (req, res) => {
        ACT_FECHA_FINAL,

        (SELECT CTI_OBSERVACION FROM PD_CONTROL_INT WHERE PD_CONTROL_INT.CTI_FK_PROYINDICADOR=B.ID_INDI
-        AND PD_CONTROL_INT.CTI_FK_RESPONSABLE=ID_RESP AND PD_CONTROL_INT.CTI_FK_PORYECTO_ID=ID_PROY) AS OBSERVACION,
+        AND PD_CONTROL_INT.CTI_FK_RESPONSABLE=ID_RESP AND PD_CONTROL_INT.CTI_FK_PORYECTO_ID=ID_PROY AND PD_CONTROL_INT.CTI_MES=`+mes+`) AS
OBSERVACION,

        (SELECT CTI_RECOMENDACION FROM PD_CONTROL_INT WHERE PD_CONTROL_INT.CTI_FK_PROYINDICADOR=B.ID_INDI
-        AND PD_CONTROL_INT.CTI_FK_RESPONSABLE=ID_RESP AND PD_CONTROL_INT.CTI_FK_PORYECTO_ID=ID_PROY) AS RECOMENDACION,
+        AND PD_CONTROL_INT.CTI_FK_RESPONSABLE=ID_RESP AND PD_CONTROL_INT.CTI_FK_PORYECTO_ID=ID_PROY AND PD_CONTROL_INT.CTI_MES=`+mes+`) AS
RECOMENDACION,

        (SELECT PD_ESTADO_AUDITORIA.EAU_NOMBRE_ESTADO FROM PD_CONTROL_INT,PD_ESTADO_AUDITORIA WHERE PD_CONTROL_INT.CTI_FK_PROYINDICADOR=B.ID_INDI
        AND PD_CONTROL_INT.CTI_FK_RESPONSABLE=ID_RESP AND PD_CONTROL_INT.CTI_FK_PORYECTO_ID=ID_PROY AND
-        PD_ESTADO_AUDITORIA.EAU_ID=PD_CONTROL_INT.CTI_FK_ESTADO
+        PD_ESTADO_AUDITORIA.EAU_ID=PD_CONTROL_INT.CTI_FK_ESTADO AND PD_CONTROL_INT.CTI_MES=`+mes+`
        ) AS ESTADO
        FROM(
```

Activar Window
Ve a Configuración pa

controllers/mi-plan/pd_indicador.js

```
@@ -207,6 +207,10 @@ async function getFormulaIndicador(req, res) {
        console.log(Resultado3);
        console.log(MetaAno3);
        console.log('--------------------');*/
+
+
+
+
         var resultados = new Array();

         var logro_anual=Number((Number(eval(rowResultado['rows'][0][0]))*100).toFixed(1));
@@ -218,6 +222,8 @@ async function getFormulaIndicador(req, res) {
             }

         }
+
+

         var avance_cuatrenio=Number((Number(eval(rowResultado['rows'][0][1]))*100).toFixed(1));
         if(avance_cuatrenio<0){
@@ -232,6 +238,11 @@ async function getFormulaIndicador(req, res) {
         }

         }
+
+        if(MetaAno==0){
+            logro_anual=0;
+        }
+
         //MIRAMOS LOS AÑOS ANTERIORES 2021
         if(avance_cuatrenio==0){
             const sqlResultado21 = "select
VWS_FORMULAS.FORMULA_LOGRO_ANUAL,VWS_FORMULAS.FORMULA_CUATRENIO,VWS_FORMULAS.MET_ID,VWS_FORMULAS.RESULTADO,VWS_FORMULAS.ID_FORMULA_CUATRENIO from
VWS_FORMULAS where VWS_FORMULAS.PIN_ID="+ano2+" AND ANO =2021";
VWS_FORMULAS where VWS_FORMULAS.PIN_ID="+ano2+" AND ANO =2021";
@@ -266,7 +277,16 @@ async function getFormulaIndicador(req, res) {
          resultados.push('0%'); // PROMEDIO AVANCE ACTIVIDAD

         }
-
+      // 68,5
+
+        binds_rs = [];
+        avance_cuatrenio = (isNaN(avance_cuatrenio) ? 0 : avance_cuatrenio);
+        logro_anual = (isNaN(logro_anual) ? 0 : logro_anual);
+
+        console.log('UPDATE PD_METAS SET MET_LOGRO_ANUAL= '+logro_anual+' , MET_AVANCE_META_CUATRE= '+avance_cuatrenio+' WHERE
MET_ID='+rowResultado['rows'][0][2]);
+        await connection.execute('UPDATE PD_METAS SET MET_LOGRO_ANUAL= '+logro_anual+' , MET_AVANCE_META_CUATRE= '+avance_cuatrenio+' WHERE
MET_ID='+rowResultado['rows'][0][2], binds_rs, {autoCommit: true});
+
+
         resultados.push(rowResultado['rows'][0][0]); //FORMULA LOGRO ANUAL
         resultados.push(rowResultado['rows'][0][1]); //FORMULA AVANCE CUATRENIO
       //  resultados.push(Number(resultado_total.toFixed(1))); // RESULTADO TOTAL
@@ -758,7 +778,9 @@ async function sincronizar(req, res) {
         });
     }

-
+
+
+
 module.exports.ActualizarResultadoMetaProyecto= ActualizarResultadoMetaProyecto;
 module.exports.ActualizarResultadoMetaProyectoPorcentaje= ActualizarResultadoMetaProyectoPorcentaje;
 module.exports.getUnidadMedida= getUnidadMedida;
```

Activar Windov
Ve a Configuración p

controllers/titulacion/mit_predio.js

```
@@ -1971,7 +1971,7 @@ const ListaPredioParaEjecutoria = async (req, res) => {

         sqlQuery = `SELECT `+CAMPOS+`  FROM VWS_PREDIOS_GESTIONADOS
         WHERE VWS_PREDIOS_GESTIONADOS.EAP_ID=1 AND VWS_PREDIOS_GESTIONADOS.PRE_ID NOT IN(SELECT MIT_EJECUTORIAS.EJE_PRE_ID FROM MIT_EJECUTORIAS)
-        ORDER BY VWS_PREDIOS_GESTIONADOS.EAP_ESTADO ASC `;
+        ORDER BY VWS_PREDIOS_GESTIONADOS.FECHA DESC `;

         binds = [];

@@ -2081,14 +2081,14 @@ const GenerarEjecutoriaPredios = async  (req, res) => {
                 //GENENERAR LA EJECUTORIA
                 oracledb.fetchAsString = [ oracledb.CLOB ];

-                let plantilla=5;
+                let plantilla=7;
                 if(tipoNotificacion==1){
-                    plantilla=5;
+                    plantilla=7;
                  }else{
                       if(tipoNotificacion==2){
-                        plantilla=6;
+                        plantilla=5;
                       }else{
-                        plantilla=7;
+                        plantilla=6;
                       }
                  }
                 const sqlplantilla = `SELECT  TO_CHAR(PLA_TEXT_RES), TO_CHAR(PLA_TEXT_MAT), TO_NCLOB(PLA_TXT_PRINCIPAL), TO_NCLOB(PLA_CABEZA),
 TO_NCLOB(PLA_PIE),TO_NCLOB(PLA_FIRMA) FROM MIT_PLANTILLAS WHERE MIT_PLANTILLAS.PLA_ID=`+plantilla; //EXTRAER  LOS TEXTO DE LA PLATILLA
 TO_NCLOB(PLA_PIE),TO_NCLOB(PLA_FIRMA) FROM MIT_PLANTILLAS WHERE MIT_PLANTILLAS.PLA_ID=`+plantilla; //EXTRAER  LOS TEXTO DE LA PLATILLA
@@ -2135,6 +2135,7 @@ const GenerarEjecutoriaPredios = async  (req, res) => {
                 html = html.replace(/(--fecha_resolucion--)/g, rowPredioResolucionFecha);
                 html = html.replace(/(--ocupantes--)/g, nombre);
                 html = html.replace(/(--cedulas--)/g, cedula);
+                html = html.replace(/(--fecha_noti--)/g, fechaNotificacion);
                 html = html.replace(/(--fecha_ejecutoria--)/g, fechaEjecutoria);
                 html = html.replace(/--firma--/g, PLA_FIRMAR);

@@ -2316,12 +2317,14 @@ const HistoricoEjecutorias = async (req, res) => {
             const offset = (page - 1) * pageSize; // Cálculo del offset

             let whereClause = '';
-            if (search) {
-              whereClause = ` WHERE PRE_REFERENCIA LIKE '%${search}%' OR PRE_MATRICULA LIKE '%${search}%' OR TEMP_CONSECUTIVO LIKE '%${search}%'
 `; // Agregar los campos de búsqueda necesarios
+            let wherePag = '';
+
+            if (search.trim()) {
+              whereClause = ` WHERE PRE_REFERENCIA LIKE '%${search.trim()}%' OR PRE_MATRICULA LIKE '%${search.trim()}%' OR TEMP_CONSECUTIVO LIKE
 '%${search.trim()}%' `; // Agregar los campos de búsqueda necesarios
+            }else{
+                wherePag=` WHERE rn > ${offset} AND ROWNUM <= ${pageSize}`
             }

-
-
             const result = await connection.execute(
               `SELECT * FROM (
                 SELECT
@@ -2336,10 +2339,10 @@ const HistoricoEjecutorias = async (req, res) => {
                 EJE_TIPO_NOTIFICACION,
                 EJE_FECHA_NOTIFICACION,
                 EJE_FECHA_EJECUTORIA , PRE_ID,
```

Activar Windows
Ve a Configuración para activar Win
Activar Windows
Ve a Configuración para activar W

```
+              ) ${wherePag}`
             );
-
+
             const ejecutorias = result.rows.map((row) => ({
               EJE_ID:        row[0],
               PRE_DIRECCION: row[1],
@@ -2351,7 +2354,8 @@ const HistoricoEjecutorias = async (req, res) => {
               FECHA_FIRMA: row[7],
               EJE_TIPO_NOTIFICACION: row[8],
               EJE_FECHA_NOTIFICACION: row[9],
-              EJE_FECHA_EJECUTORIA: row[10]
+              EJE_FECHA_EJECUTORIA: row[10],
+              PRE_ID: row[11]
             }));

             const totalCount = await connection.execute(`SELECT COUNT(*) FROM VWS_HISTORICO_EJECUTORIAS ${whereClause}`);
@@ -2369,6 +2373,84 @@ const HistoricoEjecutorias = async (req, res) => {
 }


+const EliminarEjecutoria = async (req, res) => {
+    oracledb.getConnection(connAttrs, async function (err, connection) {
+        if (err) {
+            // Error connecting to DB
+            res.set('Content-Type', 'application/json');
+            res.status(500).send(JSON.stringify({
+                status: 500,
+                message: "Error connecting to DB",
+                detailed_message: err.message
+            }));
+            return;
+        }
+        try {
+
+                let SqlDocEjecutorias = `SELECT MIT_EJECUTORIAS.EJE_PRE_ID  FROM MIT_EJECUTORIAS WHERE EJE_ID=`+id; //EXTRAER EJECUTORIAS
+                let rowsSqlDocEjecutoria = await connection.execute(SqlDocEjecutorias);
+                let PRE_ID = rowsSqlDocEjecutoria['rows'][0][0]; //PREDIO
+
+
+
+
+
+               let secuenciaEjeId=id;
+
+
+            //ELIMINAR SOPORTE EJECUTORIA
+            let sqlResulDocsEje = `SELECT DOC_NOMBRE
+            FROM MIT_DOCUMENTO_EJECUTORIAS
+            WHERE DOC_EJE_ID=`+secuenciaEjeId+``;
+            result2 = await connection.execute(
+                sqlResulDocsEje,
+            [],
+            { resultSet: true, outFormat: oracledb.OUT_FORMAT_OBJECT });
+
+            let rs_sqlResulDocsEje = result2.resultSet;
+            let row_sqlResulDocsEje;
+
+            while ((row_sqlResulDocsEje = await rs_sqlResulDocsEje.getRow())) {
+                if (fs.existsSync(rutaServerUplod+'/uploads/titulacion/anexos-ejecutorias/'+row_sqlResulDocsEje.DOC_NOMBRE+'')) {
+                    fs.unlinkSync(rutaServerUplod+'/uploads/titulacion/anexos-ejecutorias/'+row_sqlResulDocsEje.DOC_NOMBRE+''); //EXPEDIENTE
+                }
+            }
+            //ELIMINAR SOPORTE EJECUTORIA
+            if (fs.existsSync(rutaServerUplod+'/uploads/titulacion/anexos-ejecutorias/ejecutorias/'+secuenciaEjeId+'_predio_ejecutoria.pdf')) {
+
+                fs.unlinkSync(rutaServerUplod+'/uploads/titulacion/anexos-ejecutorias/ejecutorias/'+secuenciaEjeId+'_predio_ejecutoria.pdf');
 //UNIR LA EJECUTORIA
+            }
```

Activar Windows
Ve a Configuración para activar Windows

```
            }
            if (fs.existsSync(rutaServerUplod+'/uploads/titulacion/documento-predio-completo/'+PRE_ID+'_documento-predio-completo.pdf')) {
                fs.unlinkSync(rutaServerUplod+'/uploads/titulacion/documento-predio-completo/'+PRE_ID+'_documento-predio-completo.pdf')
            }


            const result3 = await connection.execute(
                `DELETE FROM MIT_EJECUTORIAS WHERE EJE_ID = :id`,
                [secuenciaEjeId],
                { autoCommit: true }
              );

              const result4 = await connection.execute(
                `DELETE FROM MIT_DOCUMENTO_EJECUTORIAS WHERE MIT_DOCUMENTO_EJECUTORIAS.DOC_EJE_ID=:id`,
                [secuenciaEjeId],
                { autoCommit: true }
              );


            res.status(200).json({ success: true, message: 'Ejecutoria eliminada correctamente' });
          } catch (error) {
            console.error(error);
            res.status(500).json({ success: false, message: 'Error al eliminar la ejecutoria' });
          }



    });
}

module.exports.Guardar= Guardar;
module.exports.Modificar= Modificar;
module.exports.ModificarEstado= ModificarEstado;
```

Activar Windows
Ve a Configuración para activar Win

```
@@ -64,6 +64,9 @@ router.post('/generar-ejecutoria',upload.array('documentos'), titulacion_router.

router.get('/get-historico-ejecutorias', titulacion_router.HistoricoEjecutorias);

+ router.delete('/eliminar-ejecutoria/:id', titulacion_router.EliminarEjecutoria);
+
+

module.exports=router;
```

routes/titulacion/mit_predio.js

```
@@ -64,6 +64,9 @@ router.post('/generar-ejecutoria',upload.array('documentos'), titulacion_router.

router.get('/get-historico-ejecutorias', titulacion_router.HistoricoEjecutorias);

router.delete('/eliminar-ejecutoria/:id', titulacion_router.EliminarEjecutoria);


module.exports=router;
```

src/components/dashboard/ejecutoria/ejecutoria_form.js

```
@@ -8,7 +8,7 @@ import {
  Grid,
  InputLabel,
  Paper,
-  Select,
+  CircularProgress,
  Table,
  TableBody,
  TableCell,
@@ -84,6 +84,7 @@ export const EjecutoriaForm = () =>
  const [itemsPerPage] = useState(5);
  const [selectedDocumentos, setSelectedDocumentos] = useState([]);
  const [selectedDocumentosApi, setSelectedDocumentosApi] = useState([]);
+  const [loading, setLoading] = useState(true);

  const classes = useStyles();

@@ -97,6 +98,7 @@ export const EjecutoriaForm = () =>


 function listar_predios(){
+    setLoading(true);
     axios({
       url:`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/get-lista-predios-para-ejecutoria`,
       method:'get'
@@ -105,12 +107,14 @@ export const EjecutoriaForm = () =>
       if (res.length > 0) {
         setPredios(res)
         setFilteredPredios(res)
+
       }else{
         Swal.fire("No hay Predio ", {
           icon: "error"
         });
         setPredios(res)
         setFilteredPredios(res)
+
       }else{
         Swal.fire("No hay Predio ", {
           icon: "error"
         });

       }
+      setLoading(false);
     })

   }
@@ -461,16 +465,23 @@ export const EjecutoriaForm = () =>
                     </TableCell>
                     <TableCell>REFERENCIA</TableCell>
                     <TableCell>MATRICULA</TableCell>
-                    <TableCell>DIRECCION</TableCell>
+                    <TableCell>DIRECCIÓN</TableCell>
                     <TableCell>BARRIO</TableCell>
                     <TableCell>RESOLUCION</TableCell>
                     <TableCell>FECHA APROBACION</TableCell>
-                    <TableCell>AREA</TableCell>
+                    <TableCell>ÁREA</TableCell>

                   </TableRow>
                 </TableHead>
                 <TableBody>
-                  {currentItems.map((predio) => (
+                {loading ? (
+                  <TableRow>
+                    <TableCell colSpan={8} align="center">
+                      <CircularProgress />
+                    </TableCell>
+                  </TableRow>
```

Activar Windows
Ve a Configuración para activar Win

```
+                {loading ? (
+                  <TableRow>
+                    <TableCell colSpan={8} align="center">
+                      <CircularProgress />
+                    </TableCell>
+                  </TableRow>
+                  ) : (
+                  currentItems.map((predio) => (
                     <TableRow key={predio.PRE_ID}>
                       <TableCell>
                         <FormControlLabel
@@ -503,7 +514,10 @@ export const EjecutoriaForm = () =>


                     </TableRow>
-                  ))}
+                  ))
+                 )
+                }
+
                 {formErrors.predios && (
                 <Typography variant="body2" color="error">
                 Debes seleccionar al menos un predio.
```

src/components/dashboard/ejecutoria/historico_ejecutorias.js

```
@@ -15,7 +15,7 @@ import {
  IconButton,
  Tooltip,
  CircularProgress,
-  Link,Card
+  Link,Card,Grid
} from '@mui/material';
import { Download as DownloadIcon } from '@mui/icons-material';
import { withStyles, makeStyles } from '@mui/styles';
@@ -114,7 +114,7 @@ export const ListaEjecutorias = () => {
  };

  return (
-    <Card>
+
    <div className={classes.root}  style={{ backgroundColor: '#ffffff', padding: '20px' }}>
    <Container >

@@ -123,8 +123,10 @@ export const ListaEjecutorias = () => {
        value={searchTerm}
        onChange={handleSearch}
        variant="outlined"
+
      />
-      <TableContainer component={Paper} style={{ width: '100%' }}>
+
+      <TableContainer component={Paper} style={{ width: '100%',paddingTop: '20px' }}>
      <Table style={{ width: '100%' }}>
        <TableHead>
          <TableRow>
```

Activar Windows
Ve a Configuración para activar Wind

```
    <div className={classes.root}  style={{ backgroundColor: '#ffffff', padding: '20px' }}>
    <Container >

@@ -123,8 +123,10 @@ export const ListaEjecutorias = () => {
        value={searchTerm}
        onChange={handleSearch}
        variant="outlined"
+
      />
-      <TableContainer component={Paper} style={{ width: '100%' }}>
+
+      <TableContainer component={Paper} style={{ width: '100%',paddingTop: '20px' }}>
       <Table style={{ width: '100%' }}>
          <TableHead>
            <TableRow>
@@ -132,7 +134,6 @@ export const ListaEjecutorias = () => {
              <TableCell>Fecha Notificación</TableCell>
              <TableCell>Fecha Ejecutoria</TableCell>
              <TableCell>Referencia</TableCell>
-
              <TableCell>Resolución</TableCell>
              <TableCell>Fecha Resolucion</TableCell>
              <TableCell>Acciones</TableCell>
@@ -141,7 +142,7 @@ export const ListaEjecutorias = () => {
          <TableBody>
            {loading ? (
              <TableRow>
-              <TableCell colSpan={4}>
+              <TableCell colSpan={7} align="center">
                <CircularProgress />
              </TableCell>
            </TableRow>
@@ -154,7 +155,6 @@ export const ListaEjecutorias = () => {
              <TableCell>{ejecutoria.PRE_REFERENCIA}</TableCell>
              <TableCell>{ejecutoria.TEMP_CONSECUTIVO}</TableCell>
              <TableCell>{ejecutoria.FECHA}</TableCell>
-
              <TableCell>
                <Tooltip title="Descargar Archivos">
                  <IconButton onClick={() => downloadArchivo(ejecutoria.EJE_ARCHIVO_URL)}>
@@ -176,7 +176,8 @@ export const ListaEjecutorias = () => {
     </Container>
     </div>
-    </Card>
+
+
   );
 };
```

Activar Windows
Ve a Configuración para activar Win

src/pages/ejecutoria/historico.js

```
@@ -85,7 +85,7 @@ const HistoricoEjecutoriaGestion = () => {
          >
            <Grid
              item
-             xs={8}
+             xs={12}
            >
              <Grid
                item
@@ -103,6 +103,9 @@ const HistoricoEjecutoriaGestion = () => {

        <Grid
          item
+         lg={12}
+         md={12}
+         sm={12}
          xs={12}
        >
          <ListaEjecutorias />
```

package-lock.json

```
@@ -40,22 +40,26 @@
    "@fullcalendar/timegrid": "^5.10.1",
    "@fullcalendar/timeline": "^5.10.1",
    "@material-ui/core": "^4.12.4",
    "@material-ui/icons": "^4.11.3",
    "@material-ui/styles": "^4.11.5",
    "@mui/icons-material": "^5.11.0",
    "@mui/icons-material": "^5.11.16",
    "@mui/lab": "^5.0.0-alpha.119",
    "@mui/material": "^5.11.8",
    "@mui/system": "^5.11.8",
    "@mui/x-date-pickers": "^5.0.17",
    "@react-pdf-viewer/core": "^3.12.0",
    "@react-pdf-viewer/default-layout": "^3.12.0",
    "@react-pdf-viewer/zoom": "^3.12.0",
    "@react-pdf/renderer": "^3.1.3",
    "@reduxjs/toolkit": "^1.9.2",
    "apexcharts": "^3.37.0",
    "aws-amplify": "^5.0.13",
    "axios": "^1.3.2",
    "crypto-js": "^4.1.1",
    "date-fns": "^2.29.3",
    "draft-js": "^0.11.7",
    "esri-loader": "^3.7.0",
    "file-saver": "^2.0.5",
    "firebase": "^9.17.1",
    "formik": "^2.2.9",
    "gray-matter": "^4.0.3",
@@ -66,6 +70,7 @@
    "nprogress": "^0.2.0",
    "numeral": "^2.0.6",
    "prop-types": "^15.8.1",
+   "qrcode.react": "^3.1.0",
    "react": "^17.0.2",
    "react-apexcharts": "^1.4.0",
    "react-beautiful-dnd": "^13.1.1",
@@ -80,6 +85,7 @@
    "react-pdf-viewer": "^0.1.0",
    "react-quill": "2.0.0",
    "react-redux": "^8.0.5",
+   "react-router-dom": "^6.11.1",
    "react-spinners": "^0.13.8",
    "react-syntax-highlighter": "^15.5.0",
    "redux": "^4.2.1",
@@ -92,6 +98,7 @@
    "sweetalert": "^2.1.2",
    "sweetalert2": "^11.7.1",
    "typescript": "^4.9.5",
+   "xlsx": "^0.18.5",
    "yup": "^1.0.0"
  },
  "devDependencies": {
```

Activar Windows
Ve a Configuración para activar Win

src/components/ExportToExcel.js

```
+ import * as FileSaver from "file-saver";
+ import * as XLSX from "xlsx";
+
+ import axios from 'axios';
+ import swal from 'sweetalert';
+ import BeatLoader from "react-spinners/BeatLoader";
+ import {Button} from '@mui/material';
+ export const ExportToExcel  = ({ opt, fileName,nombre_boton,numero_columna,responsable,origen,estado,fecha }) => {
+
+   const [bt, setBt] = useState(nombre_boton);
+   const [nombre_botonTemp, setNombre_botonTemp] = useState(nombre_boton);
+   let ws_name='Lista de Predios';
+   const fileType =
+     "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet;charset=UTF-8";
+   const fileExtension = ".xlsx";
+   const exportToCSV = async (opt, fileName,responsable,origen) => {
+   var vista='get-generar-reporte-indicadores-proyecto';
+
+
+     setBt('Procesando el Reporte...' );
+     const response = await axios.post(`${process.env.NEXT_PUBLIC_AUTH0_DOMAIN_SERVER}mit_predio/lista-resoluciones-individual-gestionadas`,
+     {
+       'ESTADO':estado,
+       'FECHA':fecha
+     });
+
+     if(response.data.length>0){
+         const apiData=response.data;
+
+     const wb = {SheetNames: [], Sheets: {}};
+
    /* create worksheet: */
    const ws = XLSX.utils.json_to_sheet(apiData);

    /* the range object is used to keep track of the range of the sheet */
    const range = { s: { c: 0, r: 0 }, e: { c: 0, r: 0 } };
    /* Iterate through each element in the structure */
    for (let R = 0; R < apiData.length; R++) {  //FILA
         if (range.e.r < R) {
           range.e.r = R;
         }
    for (let C = 0; C < numero_columna; C++) { //COLUMNA
        if (range.e.c < C) {
          range.e.c = C;
        }
            // tslint:disable-next-line:max-line-length
            const cell = { v: apiData[R][C], s: { alignment: { textRotation: 90 }, font: { sz: 16, bold: true, color: '#FF00FF' }, fill: {
bgColor: '#FFFFFF' } }, t: 's' };
            if (cell.v == null) {
              continue;
            }
            /* create the correct cell reference */
            const cell_ref = XLSX.utils.encode_cell({ c: C, r: R });
            /* determine the cell type */
            if (typeof cell.v === 'number') {
              cell.t = 'n';
            } else if (typeof cell.v === 'boolean') {
              cell.t = 'b';
            } else {
              cell.t = 's';
            }

       /* add to structure */
```

Activar Windows
Ve a Configuración para activar Win
Activar Windows
Ve a Configuración para activar

```
        /* add to structure */
        ws[cell_ref] = cell;
      }

    }

    const wscols = [
      { wch: 40 }, // "characters"
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 40 },
      { wch: 15 },
      { wch: 15 },
      { wch: 20 },
      { wch: 20 },
      { wch: 20 }
    ];


    ws['!cols'] = wscols;
    wb.SheetNames.push(ws_name);

    ws['!margins'] = { left: 0.25, right: 0.25, top: 0.75, bottom: 0.75, header: 0.3, footer: 0.3 };
    wb.Sheets[ws_name] = ws;

    const excelBuffer = XLSX.write(wb, { bookType: "xlsx", type: "array" });
  setBt(nombre_botonTemp);
 }
};


  if(origen===1){

    if (bt !== "Procesando el Reporte...") {
      return (
        <Button onClick={(e) => exportToCSV(opt, fileName,responsable,origen)} variant="contained" color='success'>{bt}</Button>
      );
     }else{
      return (
        <Button onClick={(e) => exportToCSV(opt, fileName,responsable,origen)} variant="contained" color='success'> <BeatLoader color={'#ffffff'}
 size={10} />{bt}</Button>
      );
     }

  }

};
 \ No newline at end of file
```

Activar Windows
Ve a Configuración para activar Win

src/components/dashboard/predio/predio-lista-reporte.js

```
+ /* eslint-disable react/jsx-max-props-per-line */
+ import { useEffect, useState } from 'react';
+ import locale from 'date-fns/locale/es';
+ import PropTypes from 'prop-types';
+ import { format} from 'date-fns';
+ import AssignmentReturnedIcon from '@mui/icons-material/AssignmentReturned';
+
+ import axios from 'axios';
+ import {
+   Box,
+   Button,
+   Divider,
+   Grid,
+   IconButton,
+   Link,
+   Modal,
+   TextField,
+   Table,
+   TableBody,
+   TableCell,
+   TableHead,
+   TablePagination,
+   TableRow,
+   Typography,
+   MenuItem,
+   Select
+ } from '@mui/material';
+ import { PencilAlt as PencilAltIcon } from '../../../icons/pencil-alt';
+
+ import RemoveCircleIcon from '@mui/icons-material/RemoveCircle';
+ import { Scrollbar } from '../../scrollbar';
+ import { MobileDatePicker } from '@mui/x-date-pickers'
+ import Swal from 'sweetalert2'
+
+   transform: 'translate(-50%, -50%)',
+   width: {
+     xs: '90%',
+     sm: '90%',
+     md: '70%'
+   },
+   overflowY:'auto',
+   maxHeight: '80%',
+   bgcolor: 'background.paper',
+   borderRadius: 2,
+   boxShadow: 24,
+   p: 4,
+ };
+
+ export const ListaPredioList = (props) => {
+   const {
+     customers,
+     customersCount,
+     onPageChange,
+     onRowsPerPageChange,
+     page,
+     rowsPerPage,
+     getCustomers,
+     ...other
+   } = props;
+   const [selectedCustomers, setSelectedCustomers] = useState([]);
+
+   const [modalId, setModalId] = useState(0);
+   const [estadoProrizar, setEstadoProrizar] = useState([]);
+   const [valuestadoProrizar, setValueEstadoProrizar] = useState(4);
+   const [predioId, setValuePredioId] = useState(4);
+   const [seleccionPredio, setSeleccionPredio] = useState([]);
```

Activar Windows
Ve a Configuración para activar Wi

Activar Windows
Ve a Configuración para activar Wi

```
    },
    // eslint-disable-next-line react-hooks/exhaustive-deps
    [customers]);

 const handleChangeProrizar = (event) => {
  setValueEstadoProrizar(event.target.value);
}

const CambiarEstadoModal = async (predio) => {
setModalId(1)
setValuePredioId(predio);
SelectPredio(predio);
}
const CambiarEstado = async () => {
        Swal.fire({
          title: 'Cambiar Estado',
          text: "Se va Cambiar Estado, esta seguro?",
          icon: 'warning',
          showCancelButton: true,
          confirmButtonColor: '#3085d6',
          cancelButtonColor: '#d33',
          confirmButtonText: 'Si, Esta seguro!'
        }).then((result) => {
          if (result.isConfirmed) {
            let data ={
              'PRE_ID':predioId,
              'PRE_CAMPO_ADI':valuestadoProrizar
            }
```

Activar Windows
Ve a Configuración para activar Wind

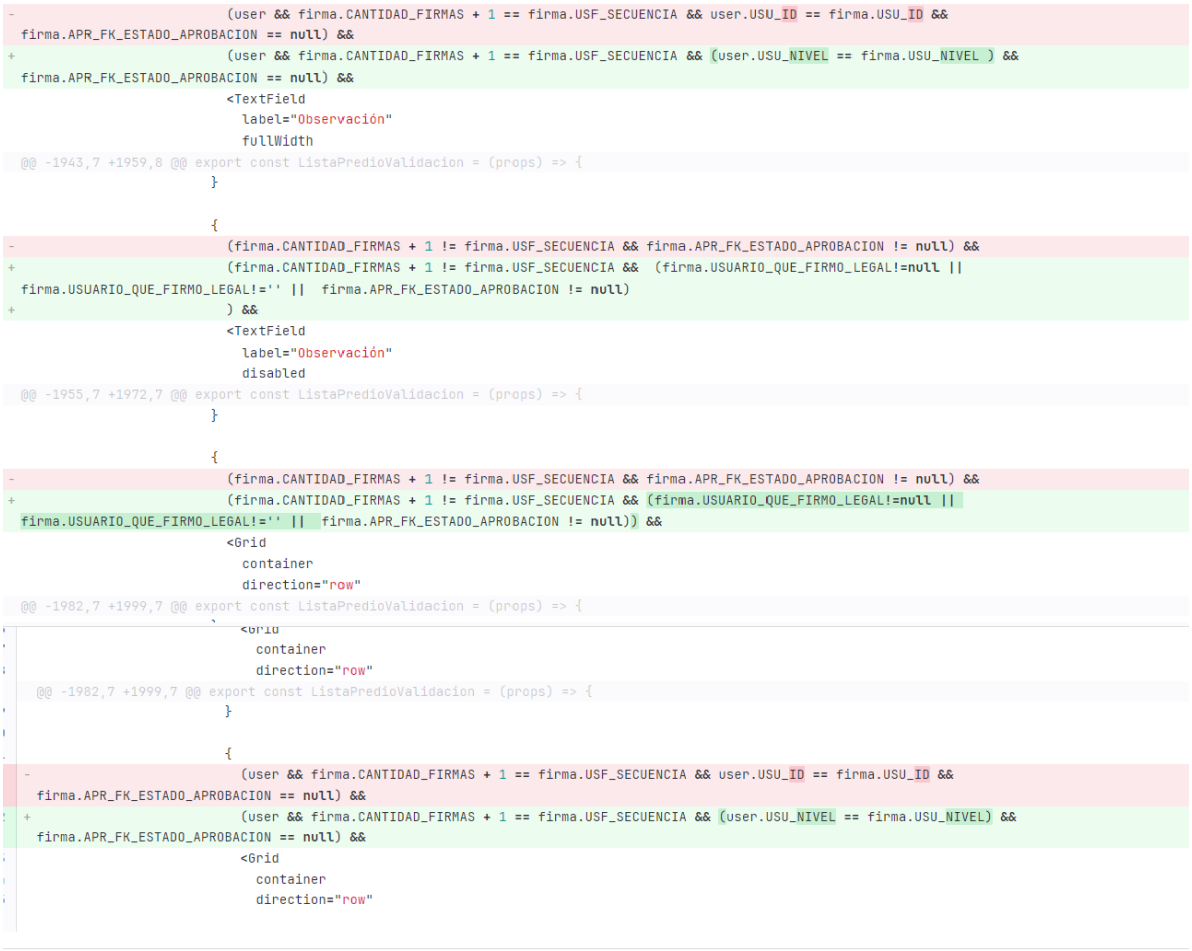
src/components/dashboard/predio/predio-lista-validacion.js

```
@@ -1927,10 +1927,26 @@ export const ListaPredioValidacion = (props) => {
                        xs={10}
                        align="center"
                      >
-                       <Typography variant="overline" color="textSecondary" fontSize={16}>{firma.USU_NOMBRE} ({firma.USF_CARGO}) </Typography>
+                       <Typography variant="overline" color="textSecondary" fontSize={16}>
+                         {user && firma && (firma.USUARIO_QUE_FIRMO_LEGAL==null || firma.USUARIO_QUE_FIRMO_LEGAL=='') && firma.USU_NIVEL==2  &&
+                           user.USU_NOMBRE
+                         }
+                         {firma && (firma.USU_NIVEL!=2 ) &&
+                          firma.USU_NOMBRE
+                         }
+                         {firma && firma.USU_NIVEL==2 && (firma.USUARIO_QUE_FIRMO_LEGAL!=null || firma.USUARIO_QUE_FIRMO_LEGAL!='') &&
+                          firma.USUARIO_QUE_FIRMO_LEGAL
+                         }

+                         {firma && (firma.USU_NIVEL!=2 ) &&
+                         <Typography variant="overline" color="textSecondary" fontSize={16}>
+                           ( {firma.USF_CARGO} )
+                          </Typography>
+                         }
+                        </Typography>

                        {
```

Activar Windows
Ve a Configuración para activar Wi



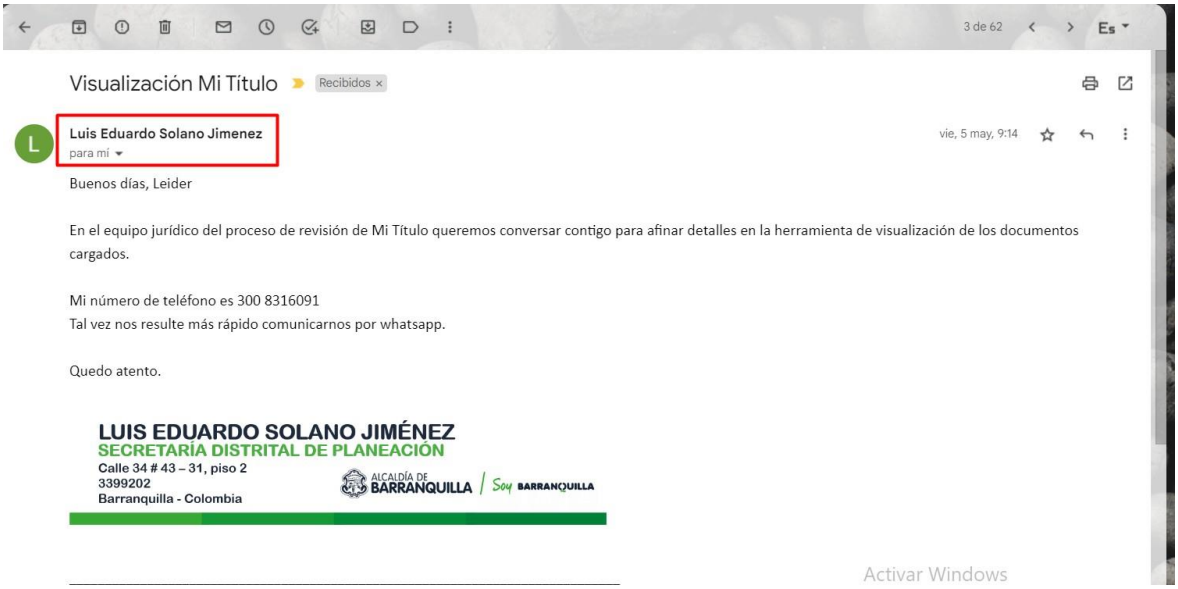
4.Capacitar sobre el manejo de nuevas plataformas y desarrollos de los sistemas de la secretaria Distrital de Planeación.

No lo realice porque no fue necesario.

5.Realizar la recolección, normalización, transformación y cargue de las diferentes fuentes de información a una base de datos de los programas de la Secretaría Distrital de Planeación.

Evidencias:

Se realizo reunión con el usuario Luis **Eduardo Solano** de la secretaria de planeación**,** con el objetivo de afinar detalles de la herramienta mititulo



Visualización Mi Título Recibidos ×

3 de 62

Luis Eduardo Solano Jimenez
para mí

vie, 5 may, 9:14

Buenos días, Leider

En el equipo jurídico del proceso de revisión de Mi Título queremos conversar contigo para afinar detalles en la herramienta de visualización de los documentos cargados.

Mi número de teléfono es 300 8316091
Tal vez nos resulte más rápido comunicarnos por whatsapp.

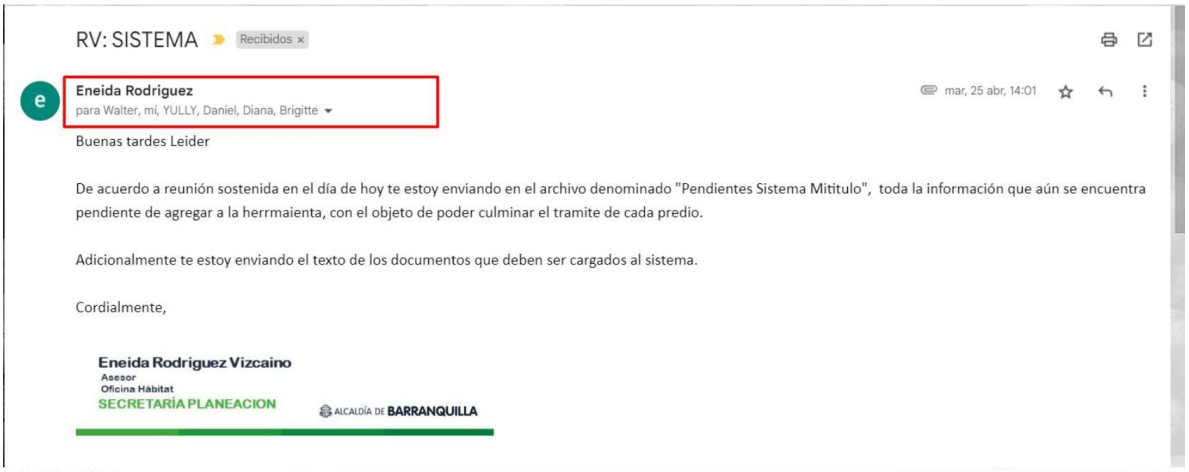
Quedo atento.

LUIS EDUARDO SOLANO JIMÉNEZ
SECRETARÍA DISTRITAL DE PLANEACIÓN
Calle 34 # 43 – 31, piso 2
3399202
Barranquilla - Colombia

ALCALDÍA DE BARRANQUILLA / Soy BARRANQUILLA

Activar Windows

Se realizo reunión con el usuario **Eneida Rodríguez,** de la oficina de hábitat, con el objetivo de poder definir los nuevo requerimiento de la herramienta mititulo.



RV: SISTEMA Recibidos ×

Eneida Rodriguez
para Walter, mi, YULLY, Daniel, Diana, Brigitte

mar, 25 abr, 14:01

Buenas tardes Leider

De acuerdo a reunión sostenida en el día de hoy te estoy enviando en el archivo denominado "Pendientes Sistema Mititulo", toda la información que aún se encuentra pendiente de agregar a la herrmaienta, con el objeto de poder culminar el tramite de cada predio.

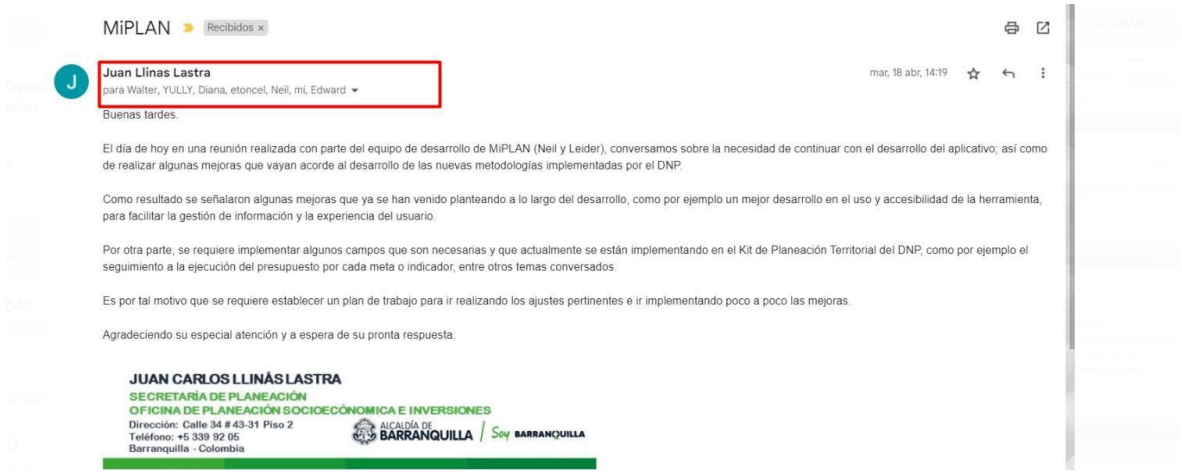
Adicionalmente te estoy enviando el texto de los documentos que deben ser cargados al sistema.

Cordialmente,

Eneida Rodriguez Vizcaino
Asesor
Oficina Hábitat
SECRETARÍA PLANEACION

ALCALDÍA DE BARRANQUILLA

Se realizo reunión con el usuario **Juan Llinás Lastra** de la oficina de planeación socioeconómica e inversiones, con el objetivo de analizar los nuevo requerimiento de la herramienta miplan.



MiPLAN Recibidos ×

Juan Llinas Lastra
para Walter, YULLY, Diana, etoncel, Neil, mí, Edward

mar, 18 abr, 14:19

Buenas tardes.

El día de hoy en una reunión realizada con parte del equipo de desarrollo de MiPLAN (Neil y Leider), conversamos sobre la necesidad de continuar con el desarrollo del aplicativo; así como de realizar algunas mejoras que vayan acorde al desarrollo de las nuevas metodologías implementadas por el DNP.

Como resultado se señalaron algunas mejoras que ya se han venido planteando a lo largo del desarrollo, como por ejemplo un mejor desarrollo en el uso y accesibilidad de la herramienta, para facilitar la gestión de información y la experiencia del usuario.

Por otra parte, se requiere implementar algunos campos que son necesarias y que actualmente se están implementando en el Kit de Planeación Territorial del DNP, como por ejemplo el seguimiento a la ejecución del presupuesto por cada meta o indicador, entre otros temas conversados.

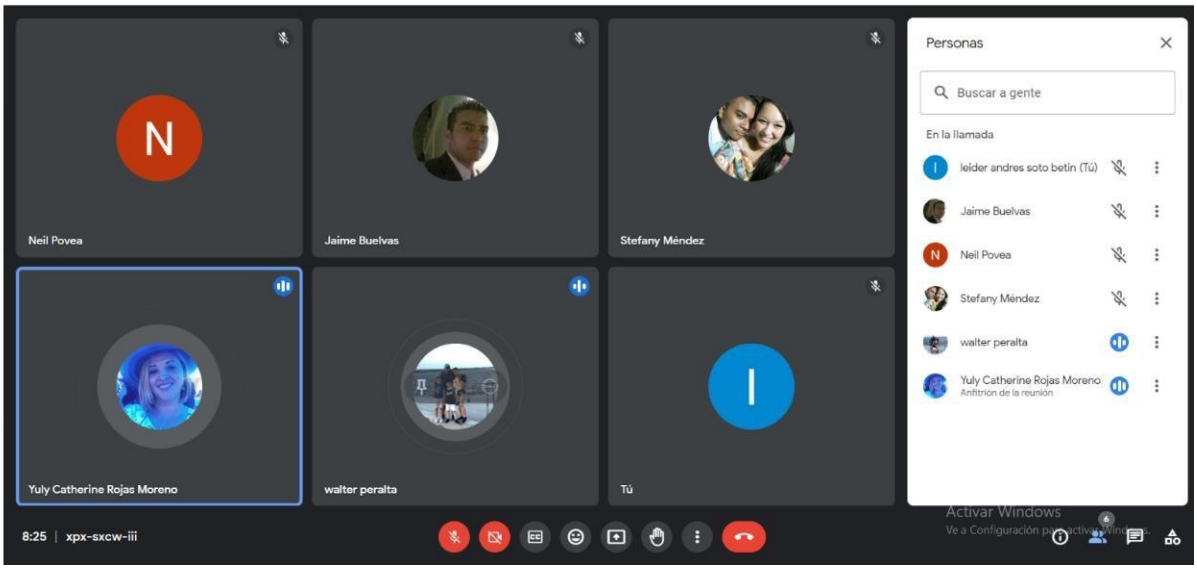
Es por tal motivo que se requiere establecer un plan de trabajo para ir realizando los ajustes pertinentes e ir implementando poco a poco las mejoras.

Agradeciendo su especial atención y a espera de su pronta respuesta.

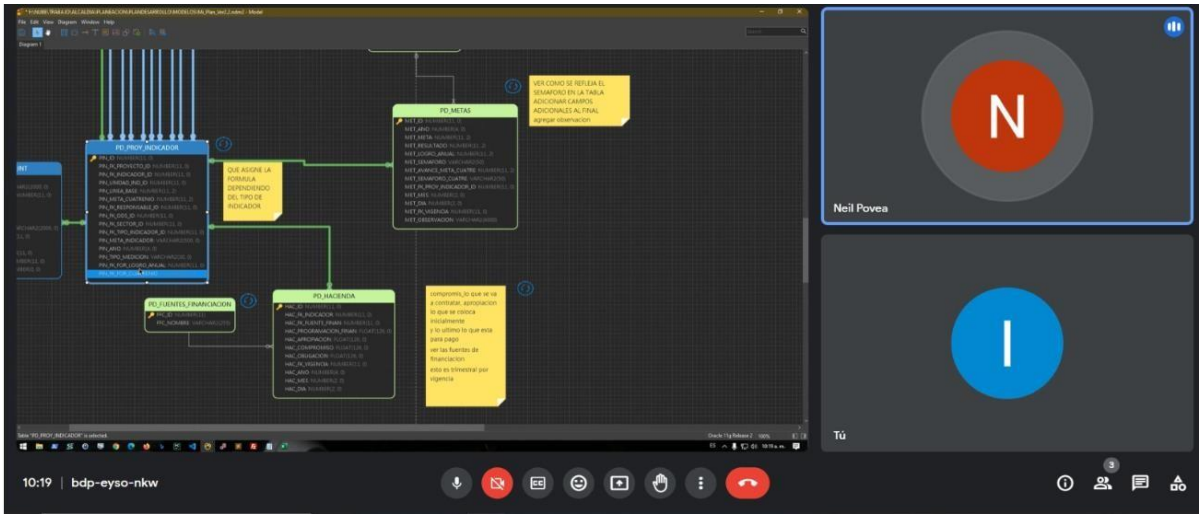
JUAN CARLOS LLINÁS LASTRA
SECRETARÍA DE PLANEACIÓN
OFICINA DE PLANEACIÓN SOCIOECÓNOMICA E INVERSIONES
Dirección: Calle 34 # 43-31 Piso 2
Teléfono: +5 339 92 05
Barranquilla - Colombia

ALCALDÍA DE BARRANQUILLA | Soy BARRANQUILLA

Se realizo reunión con la jefe **Yuly Rojas,** y demás compañeros con el fin de mirar los avances del proyecto.



Se realizo reunión con el compañero **Neil Poveda, con** el fin de afinar el modelo entidad relación de miplan.



6.Recolectar y transformar lainformación alfanumérica.

No lo realice porque no fue necesario.

leider soto

LEIDER ANDRES SOTO BETIN

CC: 78.300.913 de Montelíbano.