Barranquilla, 30 de noviembre

YULLYCATHERINE ROJAS MORENO

Contrato No CD- 06-2024-4862

Secretaria de Planeación

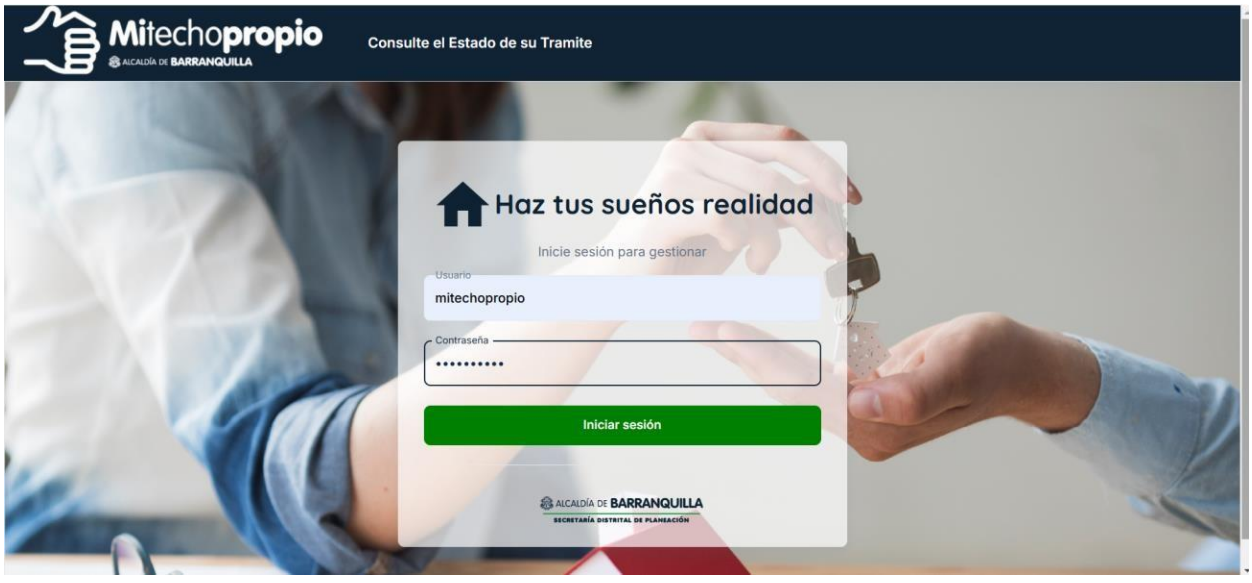
Asunto: **SEGUNDO INFORME DE GESTIÓN.**

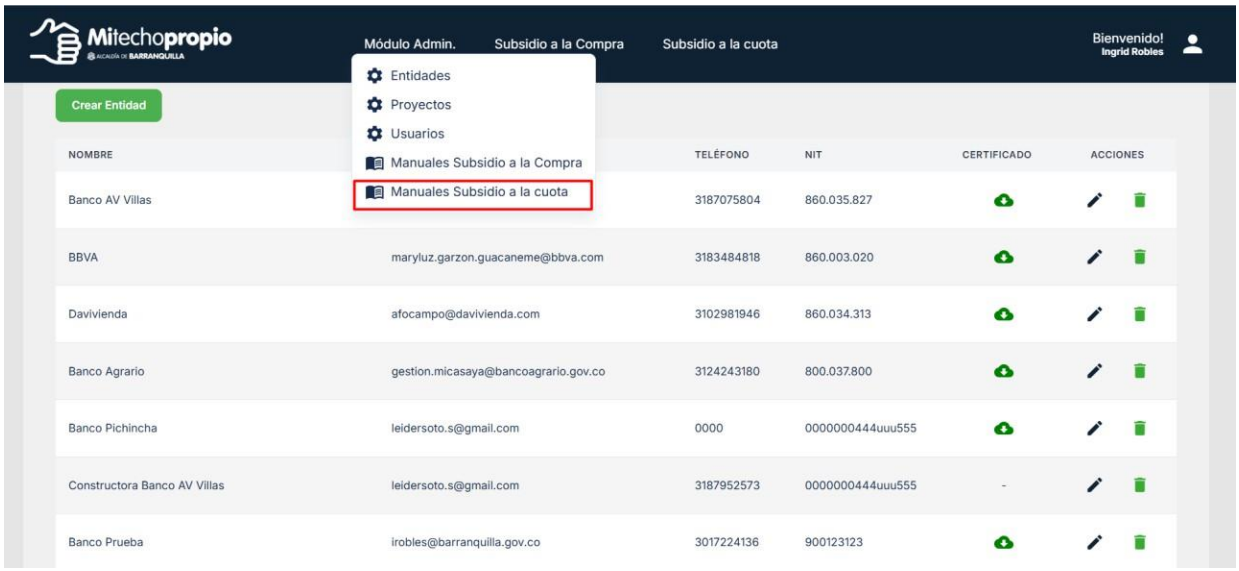
En cumplimiento del objeto del contrato de prestación de servicios No CD- 06-2024-4862, que usted supervisa, me permito rendirle informe de las actividades realizadas

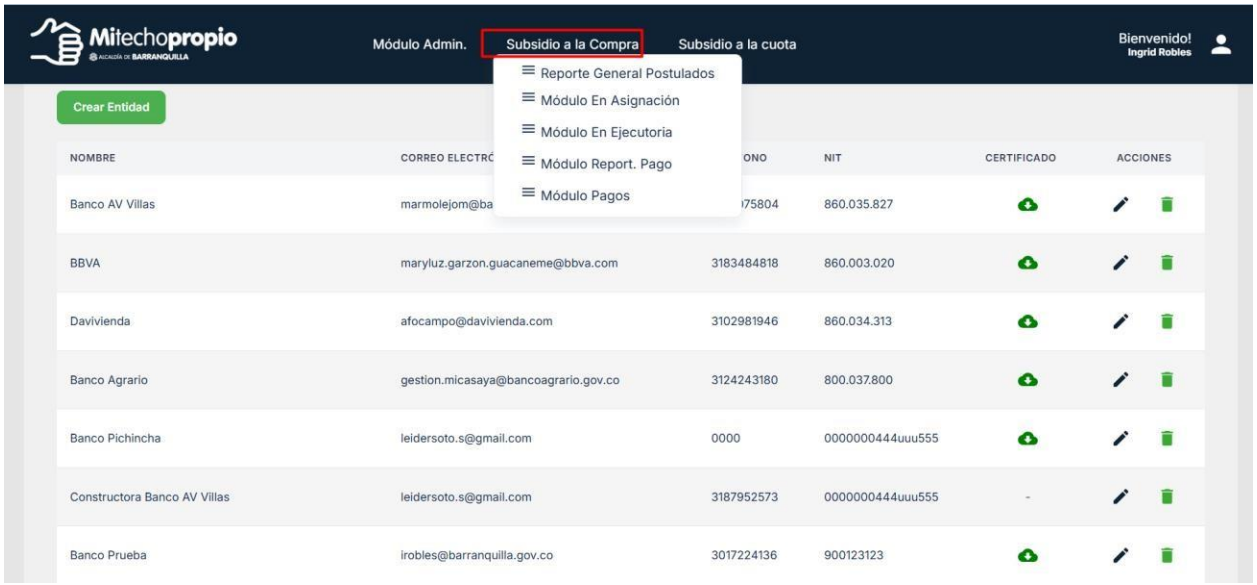
PRESTACIÓN DE SERVICIOS PROFESIONALES PARA ASESORAR A LA SECRETARIA DISTRITAL DE PLANEACIÓN EN EL DESARROLLO DE LOS PROGRAMAS Y PROYECTOS RELACIONADOS CON SU GESTIÓN.

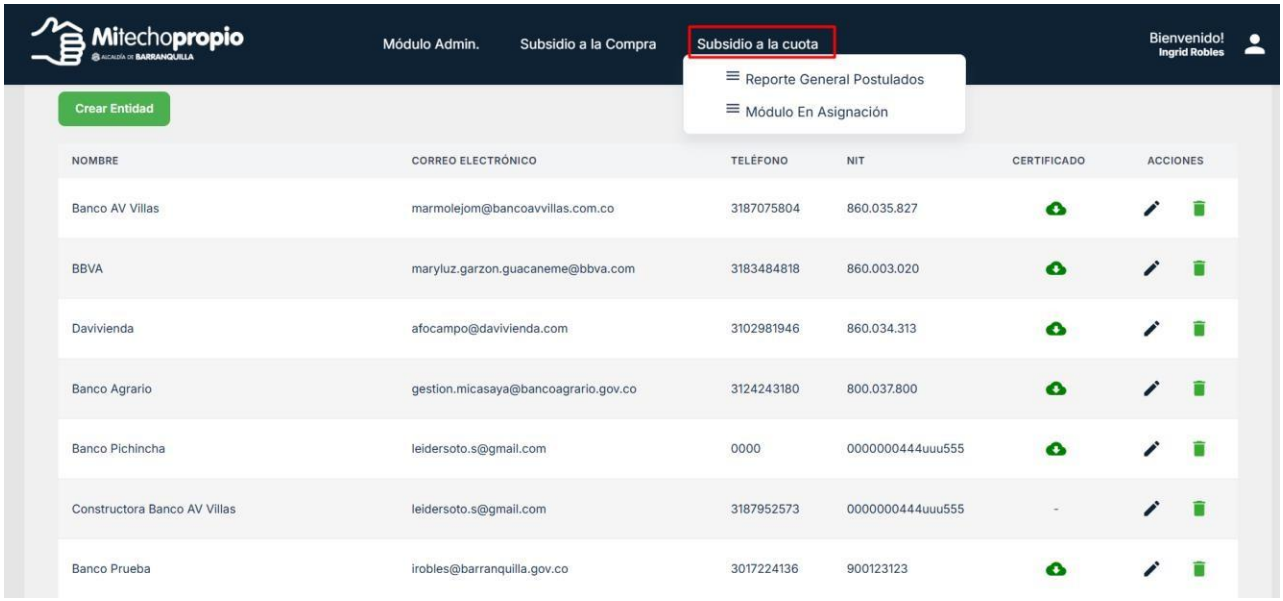
1. Desarrollar software para la secretaria Distrital de Planeación.Herramienta mitechopropio.
   - Modulo Proyecto.
   - Modulo Entidad.
   - Modulo Postulante.
   - Modulo Usuarios.
2. Actualizar funcionalidades y desarrollo de los sistemas de la secretaria Distritalde Planeación.
3. Socializar y capacitar sobre el manejo de nuevas plataformas y desarrollos delos sistemas de la secretaria Distrital de Planeación.
4. Dar soporte técnico a las bases de datos y a los sistemas desarrollados parael manejo de la información de la secretaria Distrital de Planeación.
5. Realizar la recolección, normalización, transformación y cargue de las diferentes fuentes de información a una base de datos de los programas de laSecretaría Distrital de Planeación.
6. Recolectar y transformar la información alfanumérica.

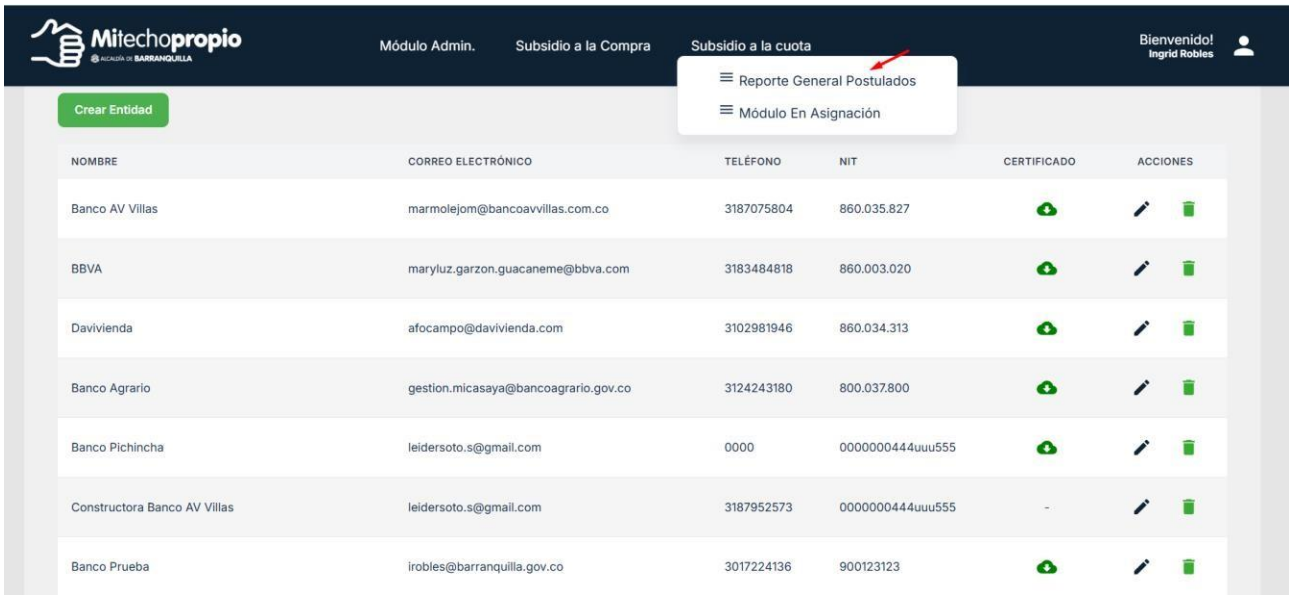
Evidencias funcionales:



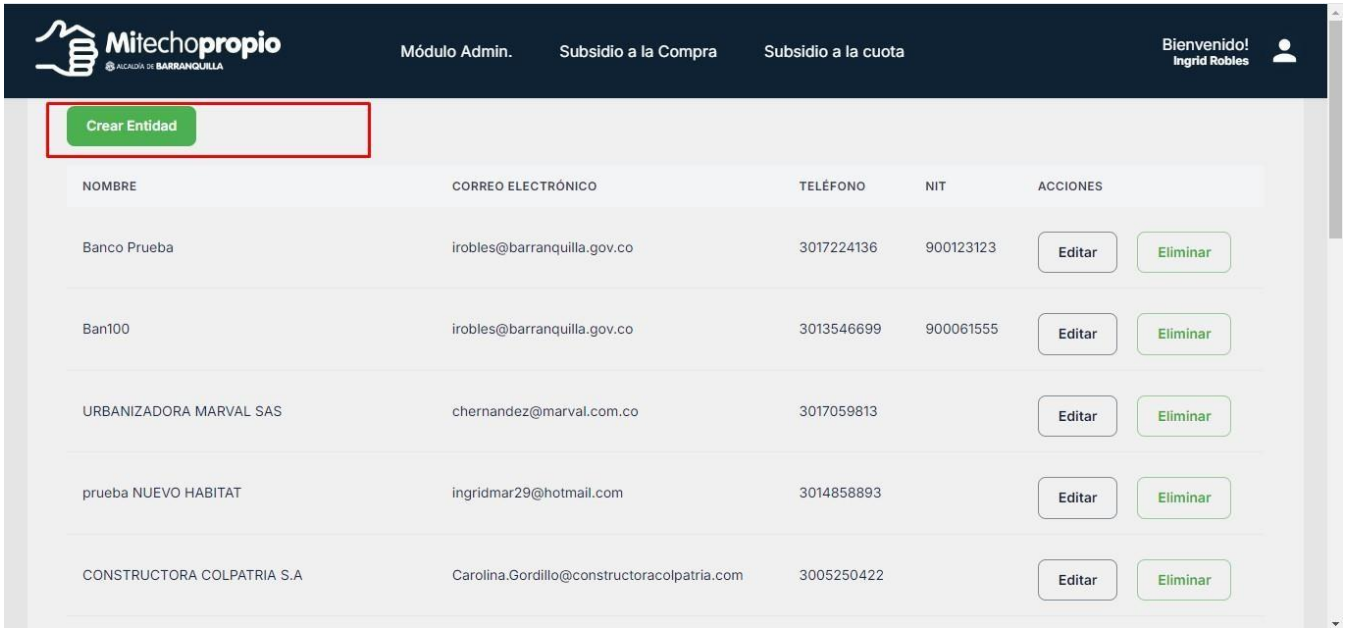

Mitechopropio
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
Entidades
Proyectos
Usuarios
Manuales Subsidio a la Compra
Manuales Subsidio a la cuota
Crear Entidad
NOMBRE
TELÉFONO
NIT
CERTIFICADO
ACCIONES
Banco AV Villas
3187075804
860.035.827
BBVA
maryluz.garzon.guacaneme@bbva.com
3183484818
860.003.020
Davivienda
afocampo@davivienda.com
3102981946
860.034.313
Banco Agrario
gestion.micasaya@bancoagrario.gov.co
3124243180
800.037.800
Banco Pichincha
leidersoto.s@gmail.com
0000
0000000444uuu555
Constructora Banco AV Villas
leidersoto.s@gmail.com
3187952573
0000000444uuu555
Banco Prueba
irobles@barranquilla.gov.co
3017224136
900123123

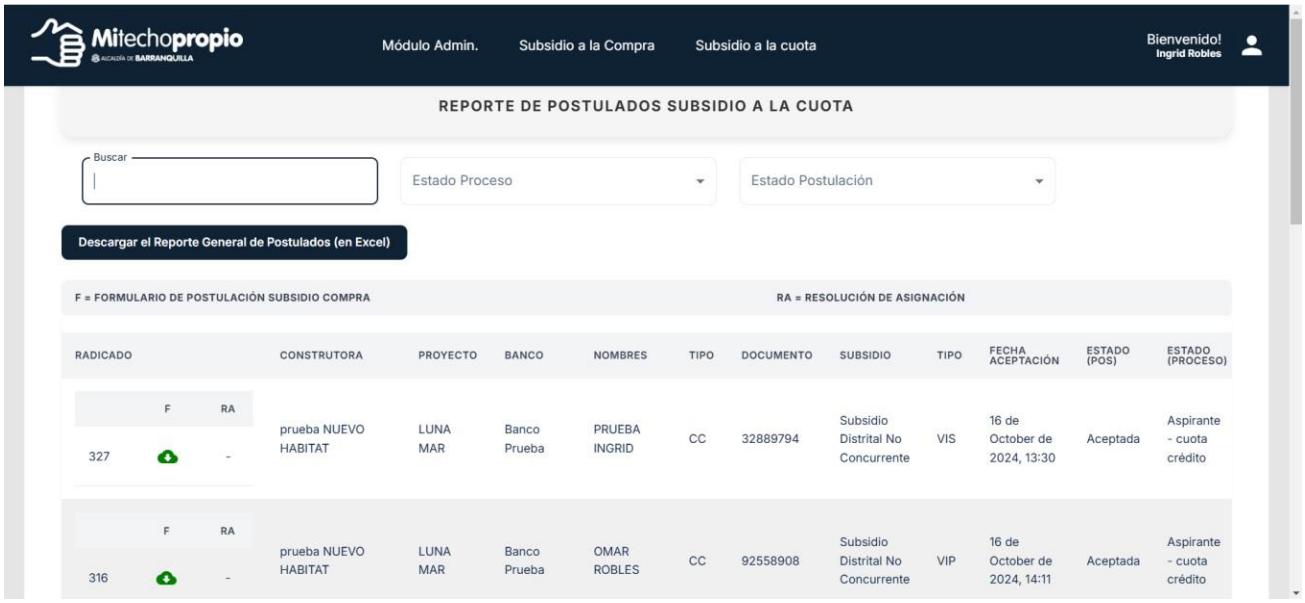

Mitechopropio
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
Reporte General Postulados
Módulo En Asignación
Módulo En Ejecutoria
Módulo Report. Pago
Módulo Pagos
Crear Entidad
NOMBRE
CERTIFICADO
ACCIONES
Banco AV Villas
860.035.827
BBVA
maryluz.garzon.guacaneme@bbva.com
3183484818
860.003.020
Davivienda
afocampo@davivienda.com
3102981946
860.034.313
Banco Agrario
gestion.micasaya@bancoagrario.gov.co
3124243180
800.037.800
Banco Pichincha
leidersoto.s@gmail.com
0000
0000000444uuu555
Constructora Banco AV Villas
leidersoto.s@gmail.com
3187952573
0000000444uuu555
Banco Prueba
irobles@barranquilla.gov.co
3017224136
900123123

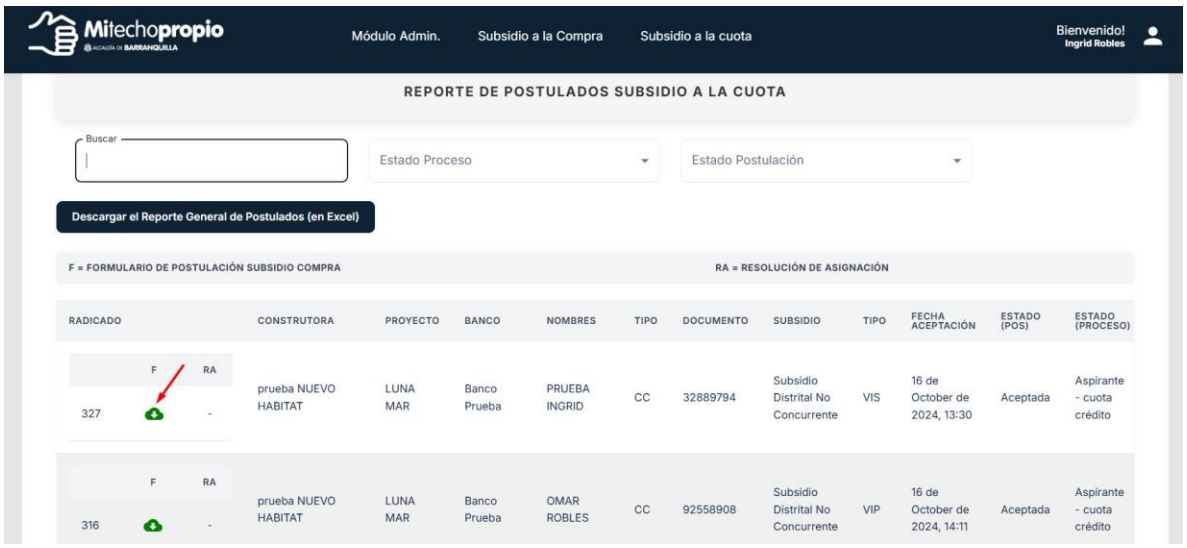

Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Reporte General Postulados
Módulo En Asignación
Bienvenido!
Ingrid Robles
Crear Entidad
NOMBRE
CORREO ELECTRÓNICO
TELÉFONO
NIT
CERTIFICADO
ACCIONES
Banco AV Villas
marmolejom@bancoavvillas.com.co
3187075804
860.035.827
BBVA
maryluz.garzon.guacaneme@bbva.com
3183484818
860.003.020
Davivienda
afocampo@davivienda.com
3102981946
860.034.313
Banco Agrario
gestion.micasaya@bancoagrario.gov.co
3124243180
800.037.800
Banco Pichincha
leidersoto.s@gmail.com
0000
0000000444uuu555
Constructora Banco AV Villas
leidersoto.s@gmail.com
3187952573
0000000444uuu555
-
Banco Prueba
irobles@barranquilla.gov.co
3017224136
900123123

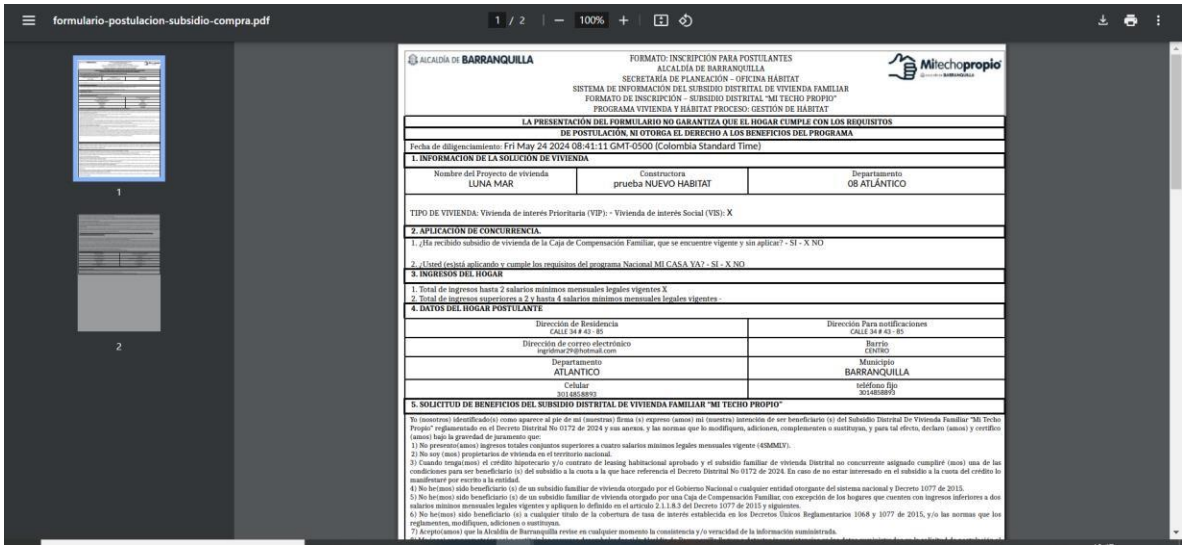

Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Reporte General Postulados
Módulo En Asignación
Bienvenido!
Ingrid Robles
Crear Entidad
NOMBRE
CORREO ELECTRÓNICO
TELÉFONO
NIT
CERTIFICADO
ACCIONES
Banco AV Villas
marmolejom@bancoavvillas.com.co
3187075804
860.035.827
BBVA
maryluz.garzon.guacaneme@bbva.com
3183484818
860.003.020
Davivienda
afocampo@davivienda.com
3102981946
860.034.313
Banco Agrario
gestion.micasaya@bancoagrario.gov.co
3124243180
800.037.800
Banco Pichincha
leidersoto.s@gmail.com
0000
0000000444uuu555
Constructora Banco AV Villas
leidersoto.s@gmail.com
3187952573
0000000444uuu555
-
Banco Prueba
irobles@barranquilla.gov.co
3017224136
900123123

Crear entidad

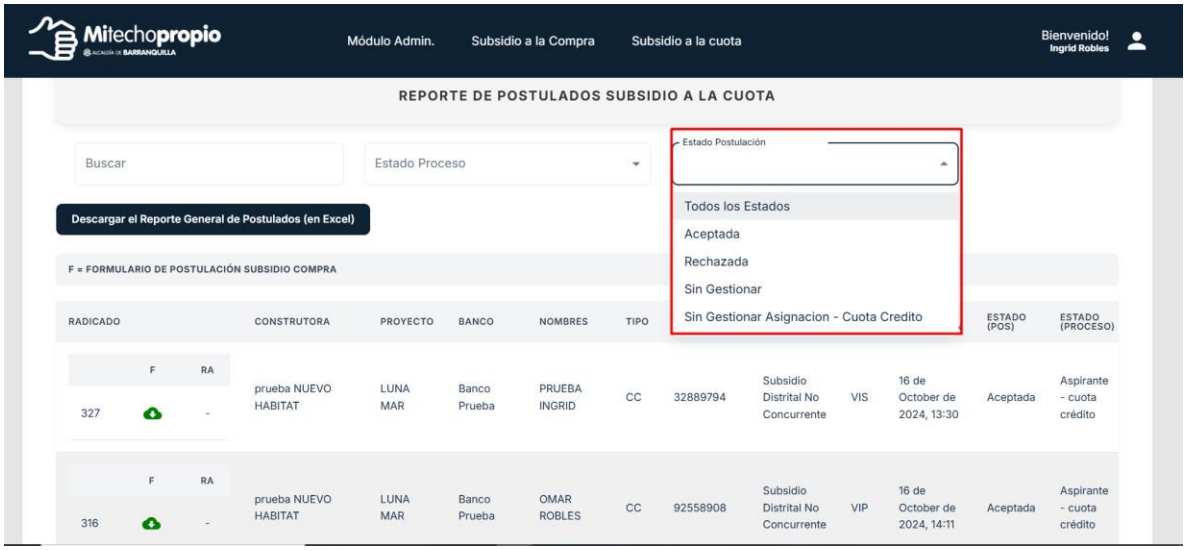




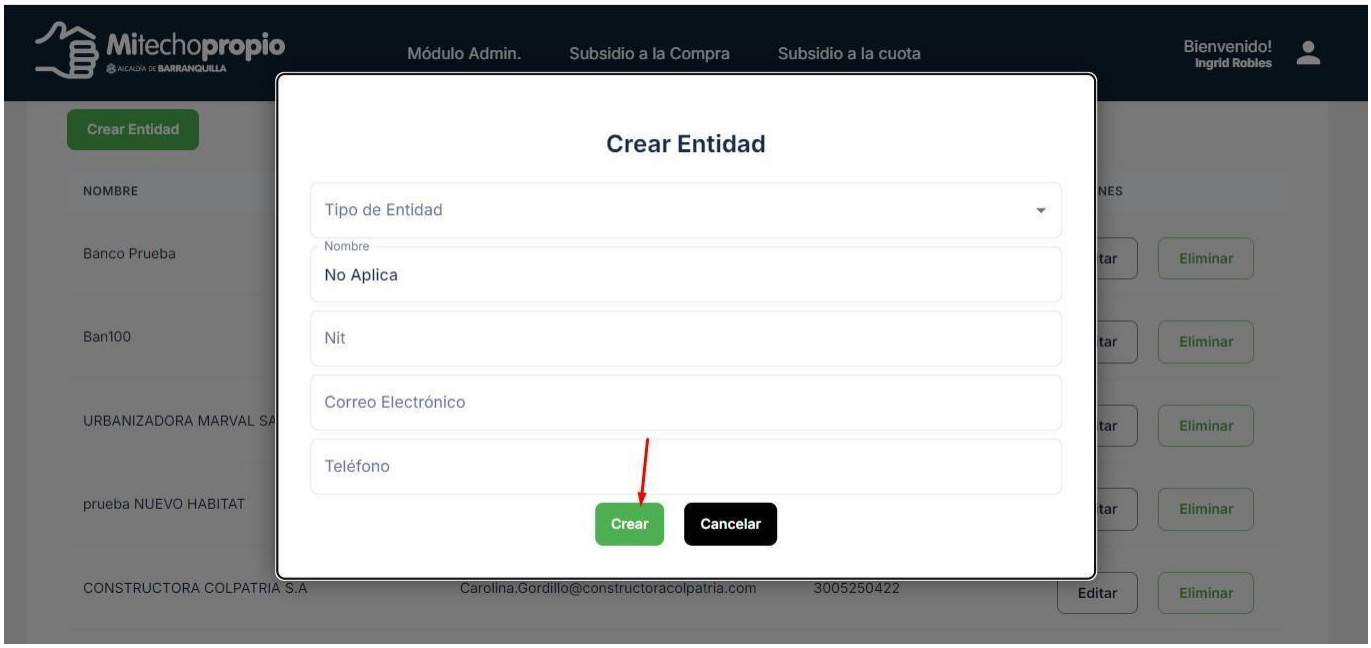

Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
REPORTE DE POSTULADOS SUBSIDIO A LA CUOTA
Buscar
Estado Proceso
Estado Postulación
Descargar el Reporte General de Postulados (en Excel)
F = FORMULARIO DE POSTULACIÓN SUBSIDIO COMPRA
RA = RESOLUCIÓN DE ASIGNACIÓN
RADICADO
F
RA
CONSTRUTORA
PROYECTO
BANCO
NOMBRES
TIPO
DOCUMENTO
SUBSIDIO
TIPO
FECHA ACEPTACIÓN
ESTADO (POS)
ESTADO (PROCESO)
327
prueba NUEVO HABITAT
LUNA MAR
Banco Prueba
PRUEBA INGRID
CC
32889794
Subsidio Distrital No Concurrente
VIS
16 de October de 2024, 13:30
Aceptada
Aspirante - cuota crédito
316
prueba NUEVO HABITAT
LUNA MAR
Banco Prueba
OMAR ROBLES
CC
92558908
Subsidio Distrital No Concurrente
VIP
16 de October de 2024, 14:11
Aceptada
Aspirante - cuota crédito


formulario-postulacion-subsidio-compra.pdf
1 / 2
100%
ALCALDÍA DE BARRANQUILLA
FORMATO: INSCRIPCIÓN PARA POSTULANTES
ALCALDÍA DE BARRANQUILLA
SECRETARÍA DE PLANEACIÓN - OFICINA HÁBITAT
SISTEMA DE INFORMACIÓN DEL SUBSIDIO DISTRITAL DE VIVIENDA FAMILIAR
FORMATO DE INSCRIPCIÓN - SUBSIDIO DISTRITAL "MI TECHO PROPIO"
PROGRAMA VIVIENDA Y HABITAT PROCESO: GESTIÓN DE HABITAT
LA PRESENTACIÓN DEL FORMULARIO NO GARANTIZA QUE EL HOGAR CUMPLE CON LOS REQUISITOS
DE POSTULACIÓN, NI OTORGA EL DERECHO A LOS BENEFICIOS DEL PROGRAMA
Fecha de diligenciamiento: Fri May 24 2024 08:41:11 GMT-0500 (Colombia Standard Time)
1. INFORMACION DE LA SOLUCIÓN DE VIVIENDA
Nombre del Proyecto de vivienda
LUNA MAR
Constructora
prueba NUEVO HABITAT
Departamento
08 ATLÁNTICO
TIPO DE VIVIENDA: Vivienda de interés Prioritaria (VIP): - Vivienda de interés Social (VIS): X
2. APLICACIÓN DE CONCURRENCIA.
1. ¿Ha recibido subsidio de vivienda de la Caja de Compensación Familiar, que se encuentre vigente y sin aplicar? - SI - X NO
2. ¿Usted (es)tá aplicando y cumple los requisitos del programa Nacional MI CASA YA? - SI - X NO
3. INGRESOS DEL HOGAR
1. Total de ingresos hasta 2 salarios mínimos mensuales legales vigentes X
2. Total de ingresos superiores a 2 y hasta 4 salarios mínimos mensuales legales vigentes -
4. DATOS DEL HOGAR POSTULANTE
Dirección de Residencia
CALLE 34 # 43 - 85
Dirección Para notificaciones
CALLE 34 # 43 - 85
Dirección de correo electrónico
ingridmar29@hotmail.com
Barrio
CENTRO
Departamento
ATLANTICO
Municipio
BARRANQUILLA
Celular
3014858893
teléfono fijo
3014858893
5. SOLICITUD DE BENEFICIOS DEL SUBSIDIO DISTRITAL DE VIVIENDA FAMILIAR "MI TECHO PROPIO"


Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
REPORTE DE POSTULADOS SUBSIDIO A LA CUOTA
Buscar
Estado Postulación
Descargar el Reporte General de Postulados (en Excel)
F = FORMULARIO DE POSTULACIÓN SUBSIDIO COMPRA
RA = RESOLUCIÓN DE ASIGNACIÓN
Todos los Estados
Esperando aceptación - cuota crédito
Registrado - cuota crédito
Aspirante - cuota crédito
En asignación - cuota crédito
Asignado - cuota crédito
En espera Etapa-1 - Cuota crédito
Subsanado Etapa-1 - Cuota crédito
Proceso de pago - cuota credito
Subsanado Etapa-1 - Cuota crédito
Terminación anticipada cuota crédito
Pago final - cuota crédito
Rechazado cuota credito
Perdida definitiva de Subsidio
No Gestionado
RADICADO
CONSTRUTORA
DOCUMENTO
SUBSIDIO
TIPO
FECHA ACEPTACIÓN
ESTADO (POS)
ESTADO (PROCESO)
327
prueba NUEVO HABITAT
32889794
Subsidio Distrital No Concurrente
VIS
16 de October de 2024, 13:30
Aceptada
Aspirante - cuota crédito
316
prueba NUEVO HABITAT
92558908
Subsidio Distrital No Concurrente
VIP
16 de October de 2024, 14:11
Aceptada
Aspirante - cuota crédito



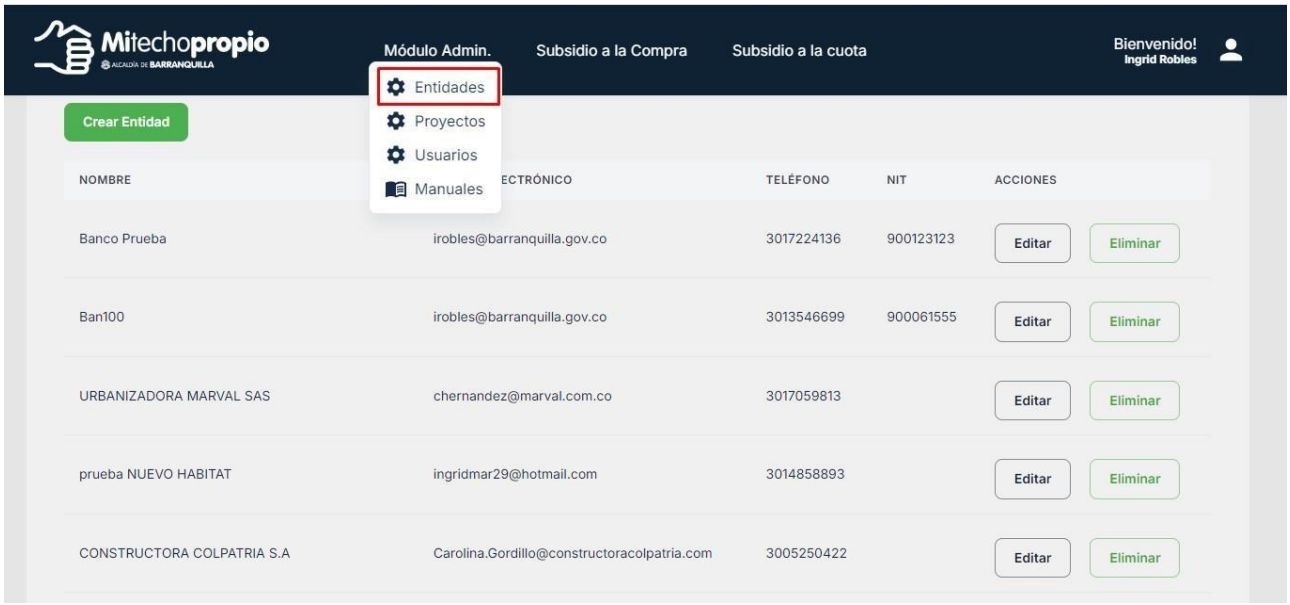
Crear



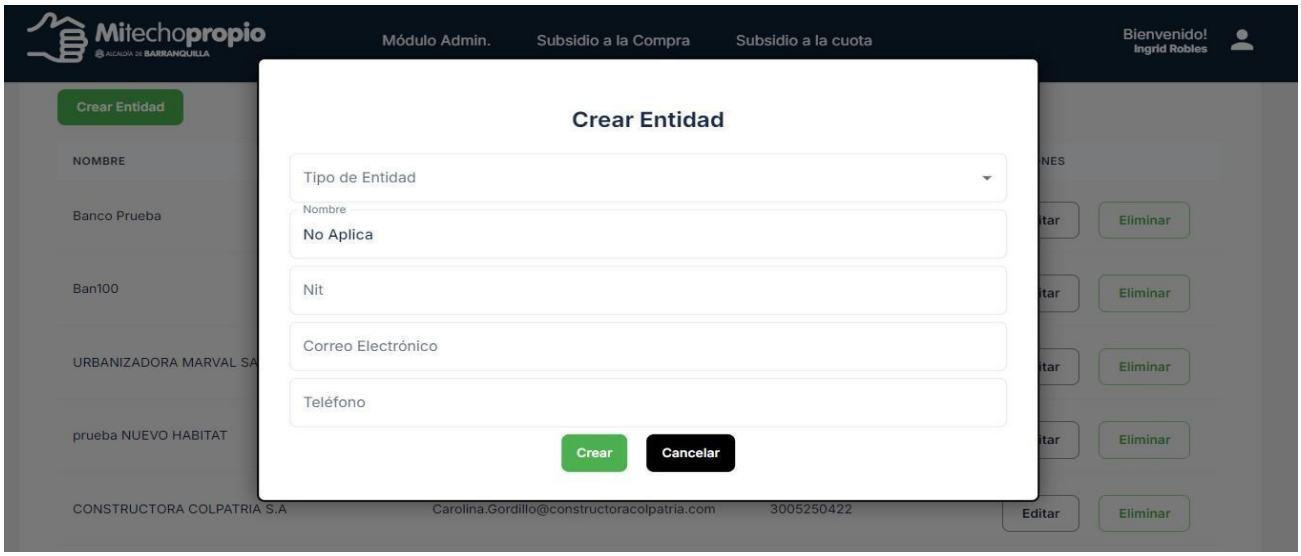
Modulo admin


Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
Crear Entidad
NOMBRE
CORREO ELECTRÓNICO
TELÉFONO
NIT
ACCIONES
Banco Prueba
irobles@barranquilla.gov.co
3017224136
900123123
Editar
Eliminar
Ban100
irobles@barranquilla.gov.co
3013546699
900061555
Editar
Eliminar
URBANIZADORA MARVAL SAS
chernandez@marval.com.co
3017059813
Editar
Eliminar
prueba NUEVO HABITAT
ingridmar29@hotmail.com
3014858893
Editar
Eliminar
CONSTRUCTORA COLPATRIA S.A
Carolina.Gordillo@constructoracolpatria.com
3005250422
Editar
Eliminar

Entidad


Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
Entidades
Proyectos
Usuarios
Manuales
Crear Entidad
NOMBRE
ECTRÓNICO
TELÉFONO
NIT
ACCIONES
Banco Prueba
irobles@barranquilla.gov.co
3017224136
900123123
Editar
Eliminar
Ban100
irobles@barranquilla.gov.co
3013546699
900061555
Editar
Eliminar
URBANIZADORA MARVAL SAS
chernandez@marval.com.co
3017059813
Editar
Eliminar
prueba NUEVO HABITAT
ingridmar29@hotmail.com
3014858893
Editar
Eliminar
CONSTRUCTORA COLPATRIA S.A
Carolina.Gordillo@constructoracolpatria.com
3005250422
Editar
Eliminar

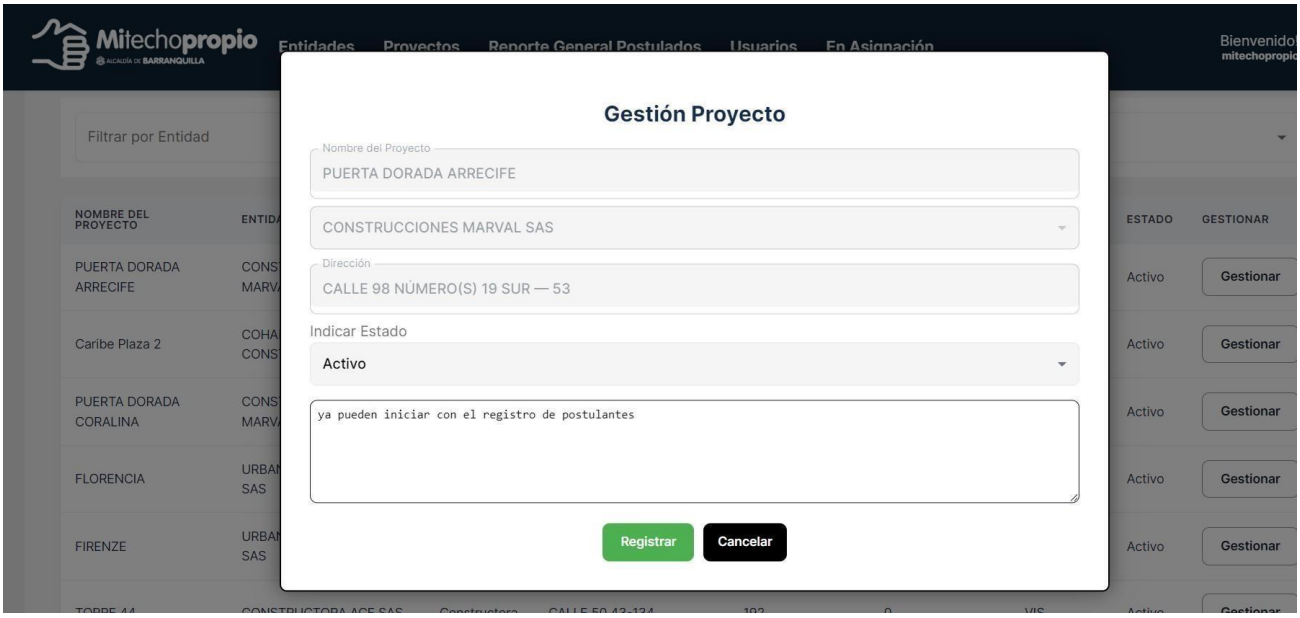


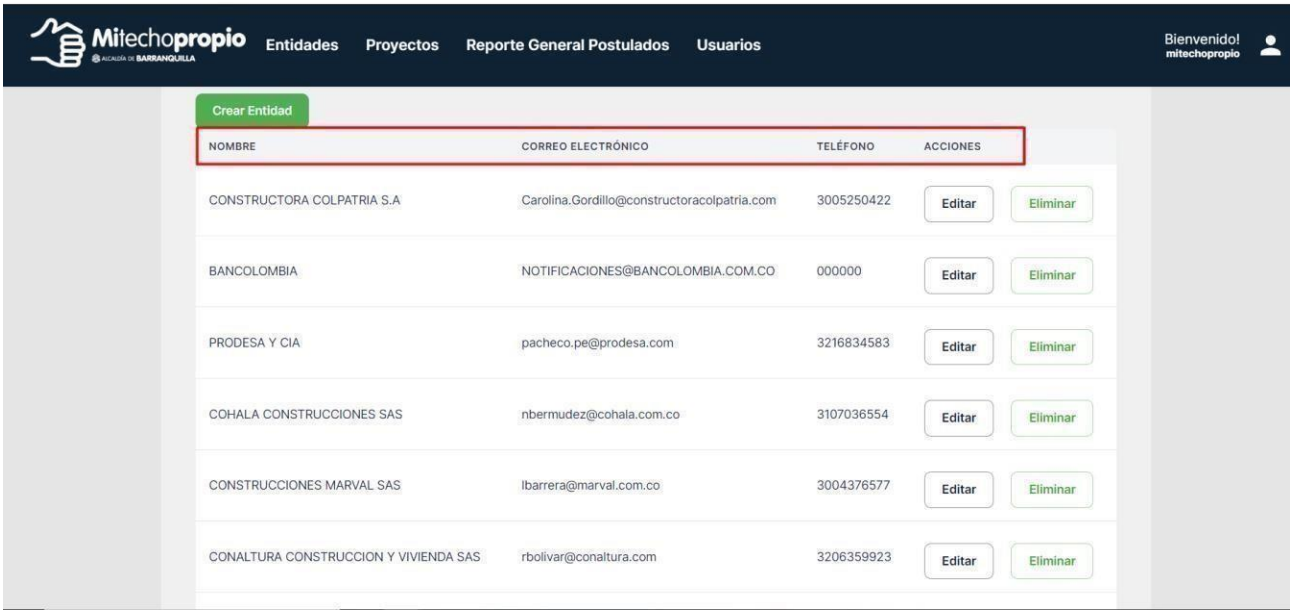


Visualizan los filtros

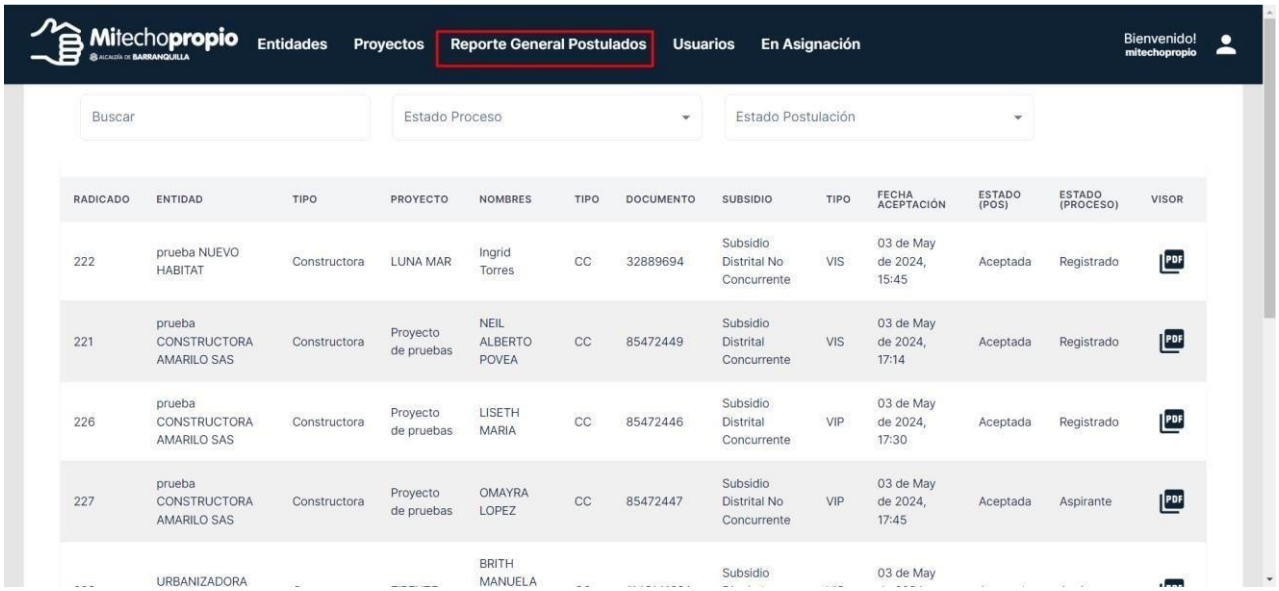


Gestión proyecto



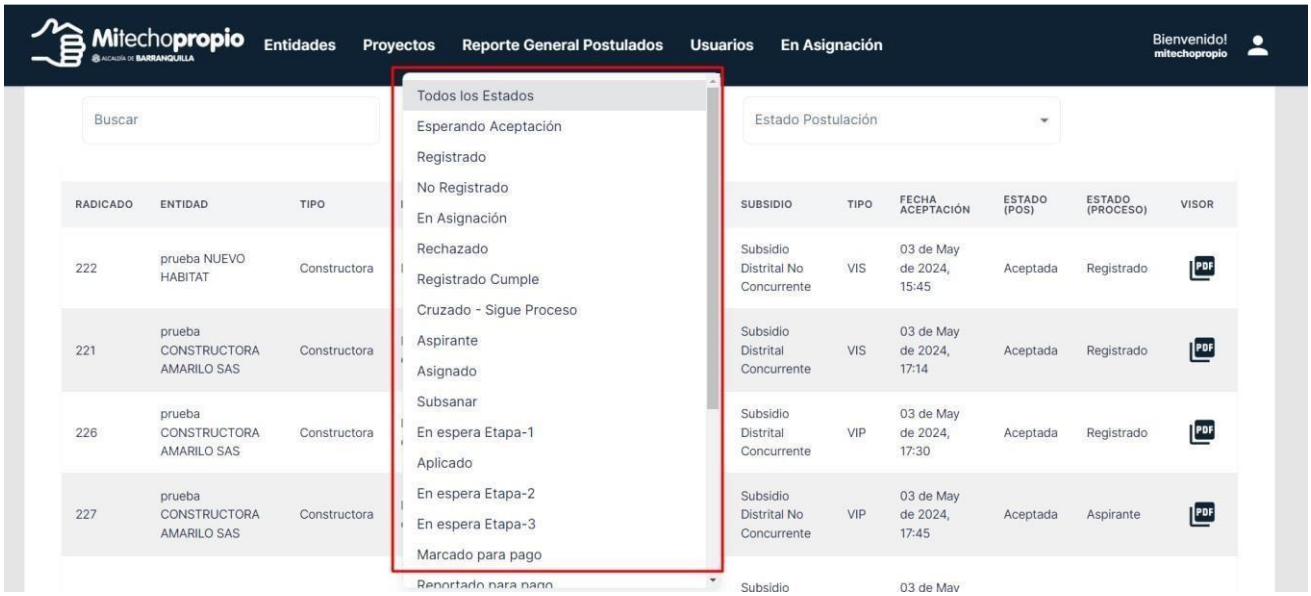


Reporte general Postulados

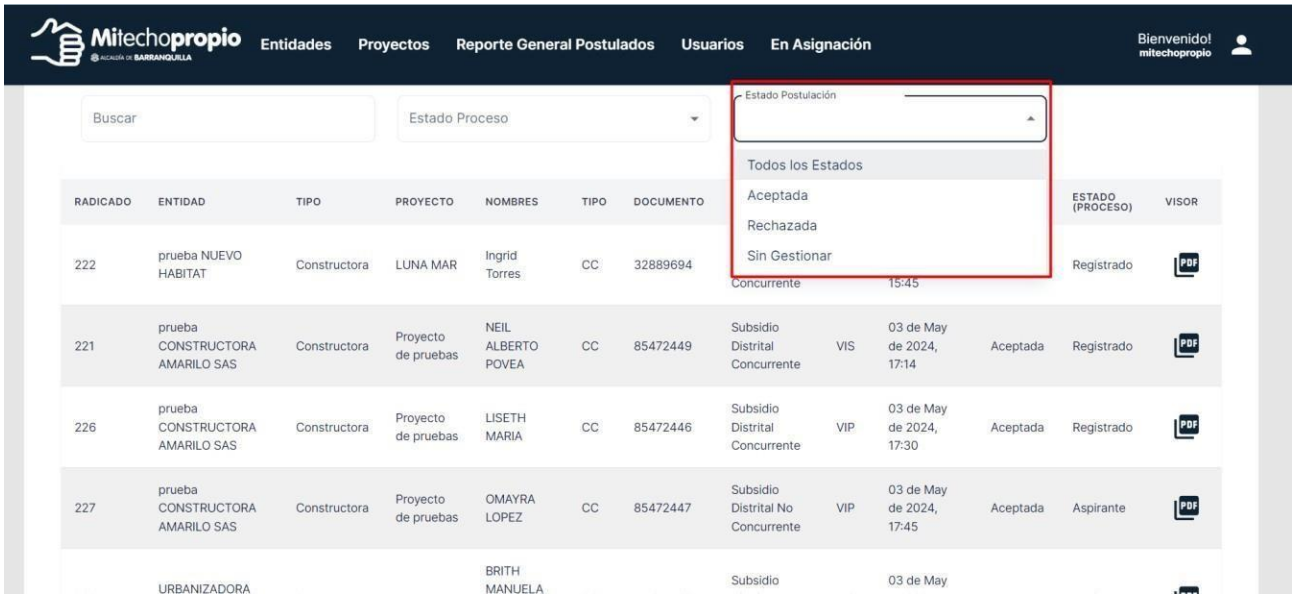




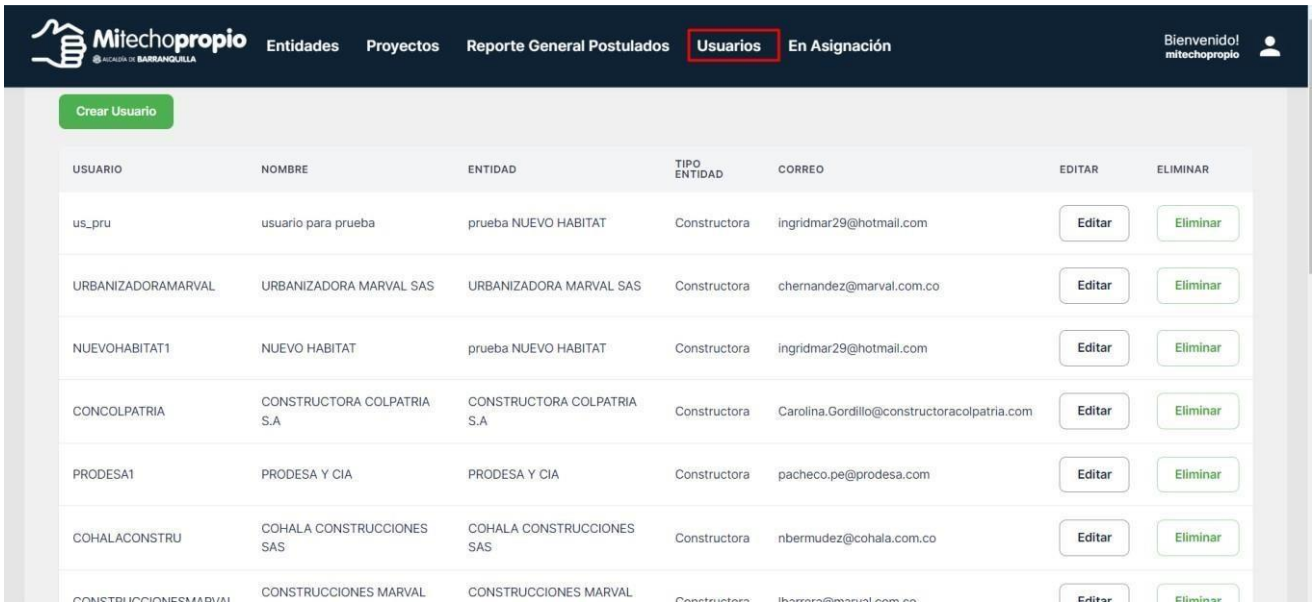
Todos los estados



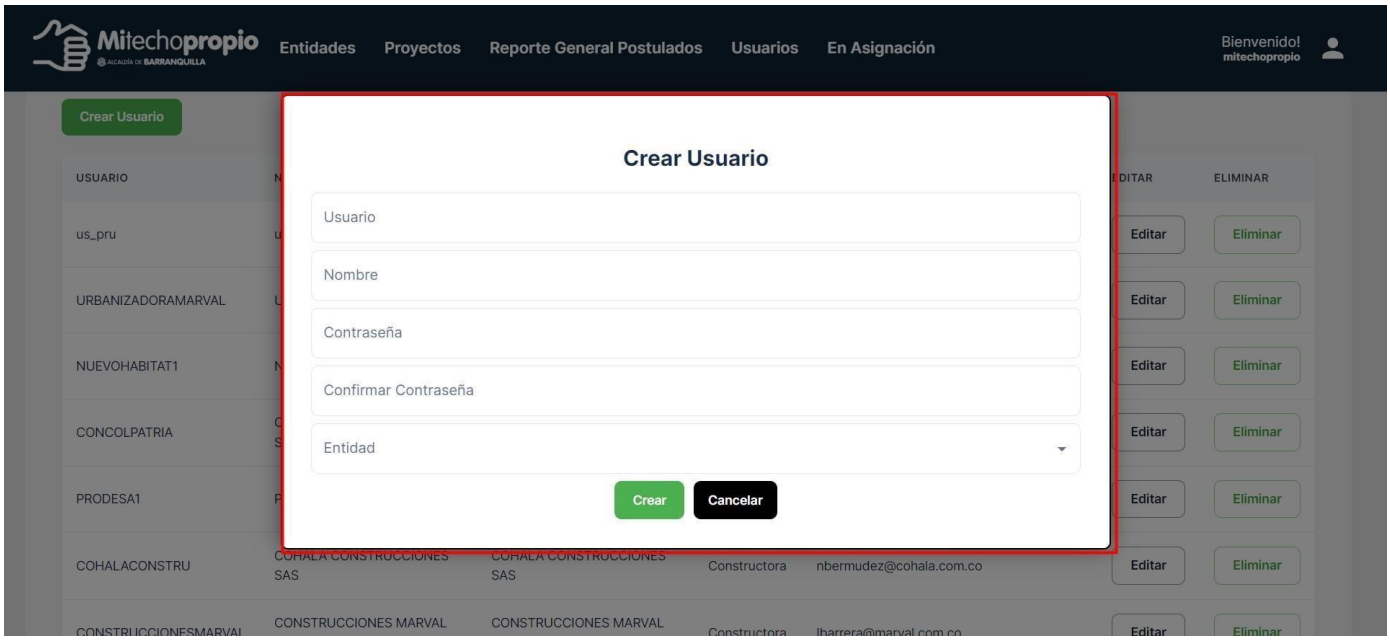
En asignación



Usuarios



Crear usuario



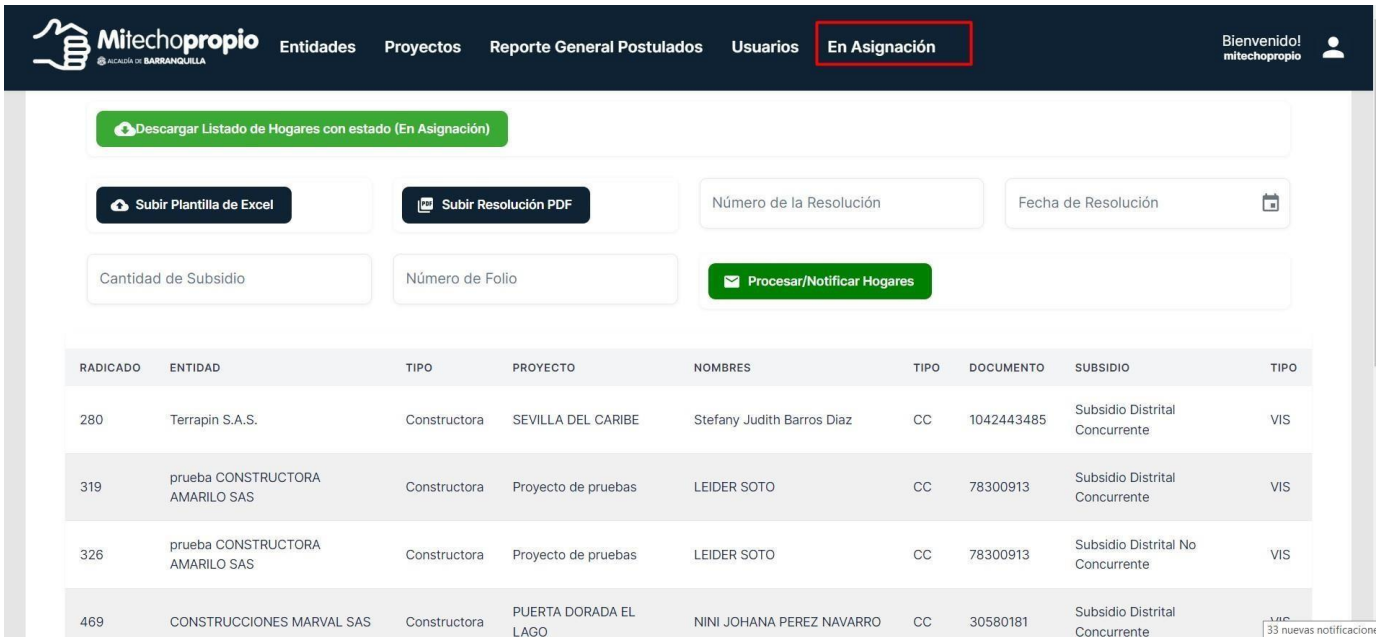
Editar



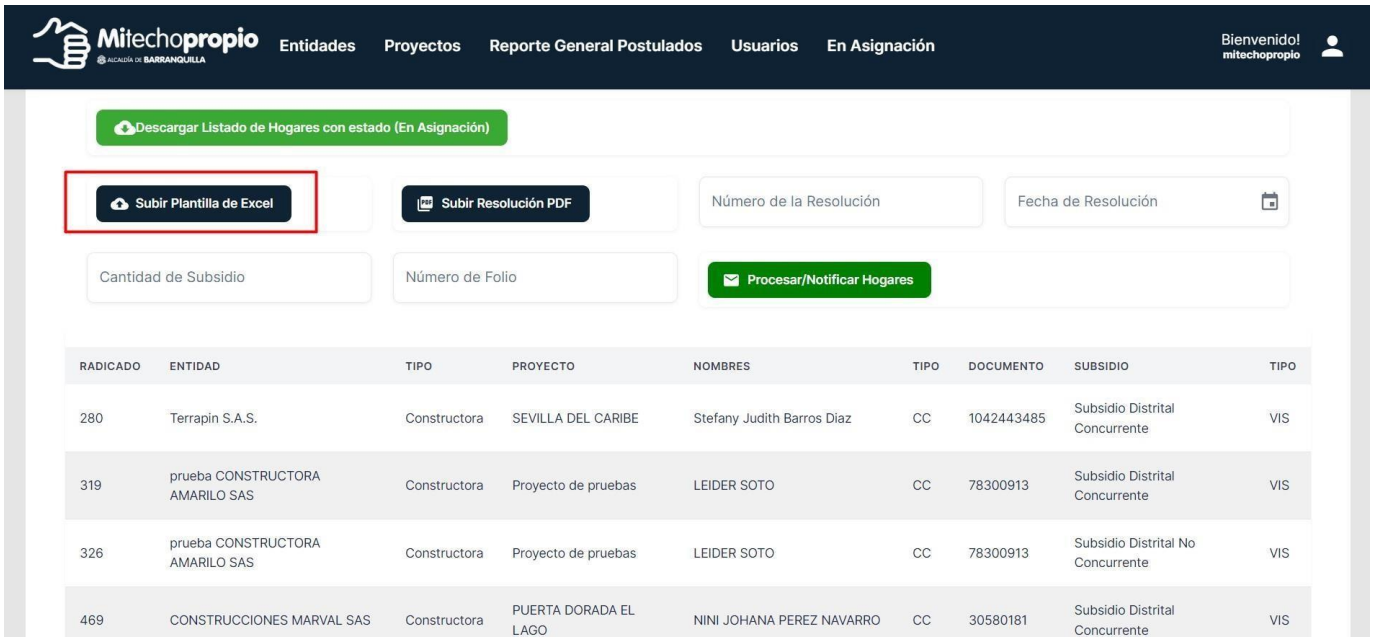
Eliminar



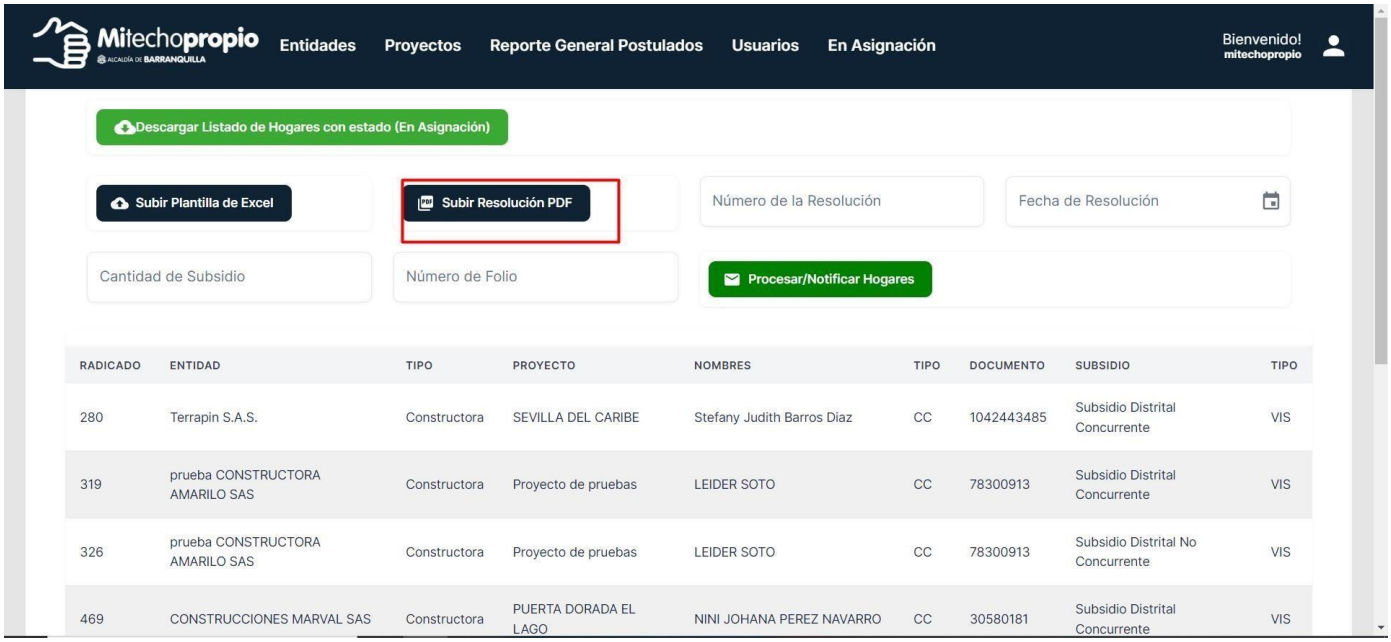
En asignación



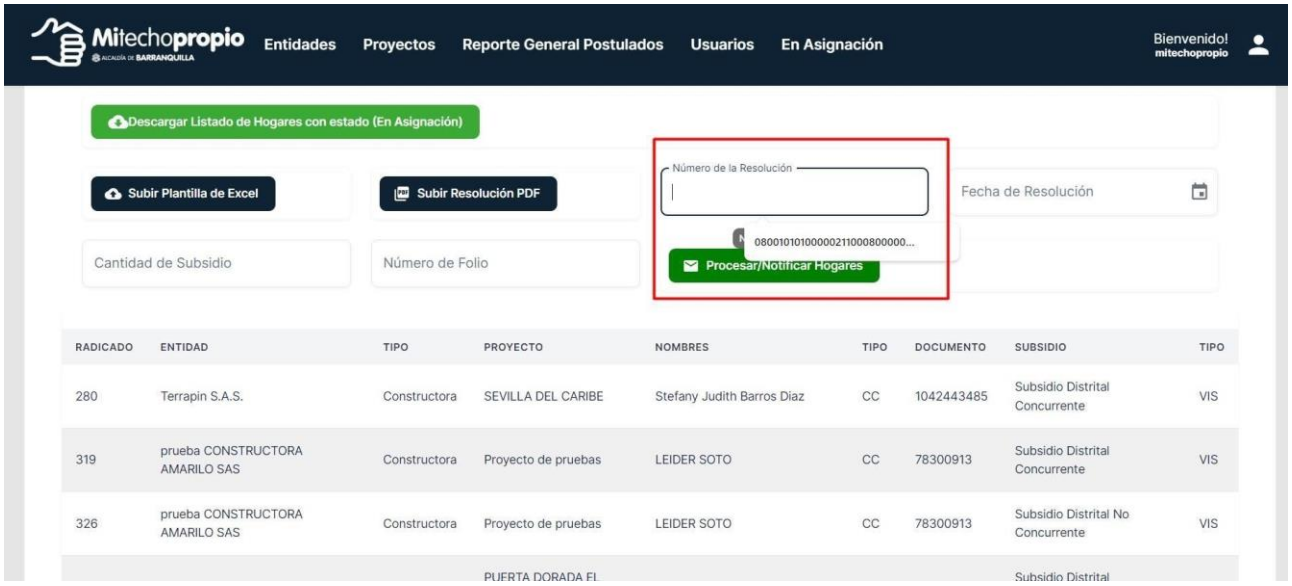
Subir plantilla

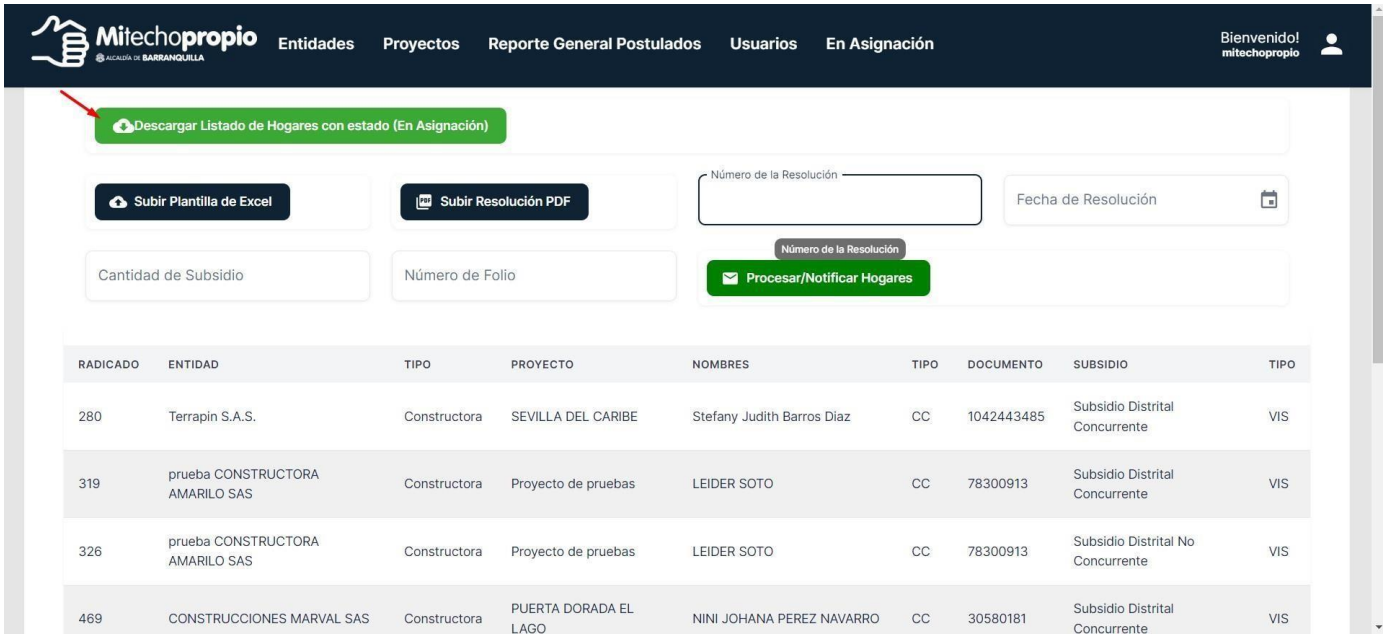


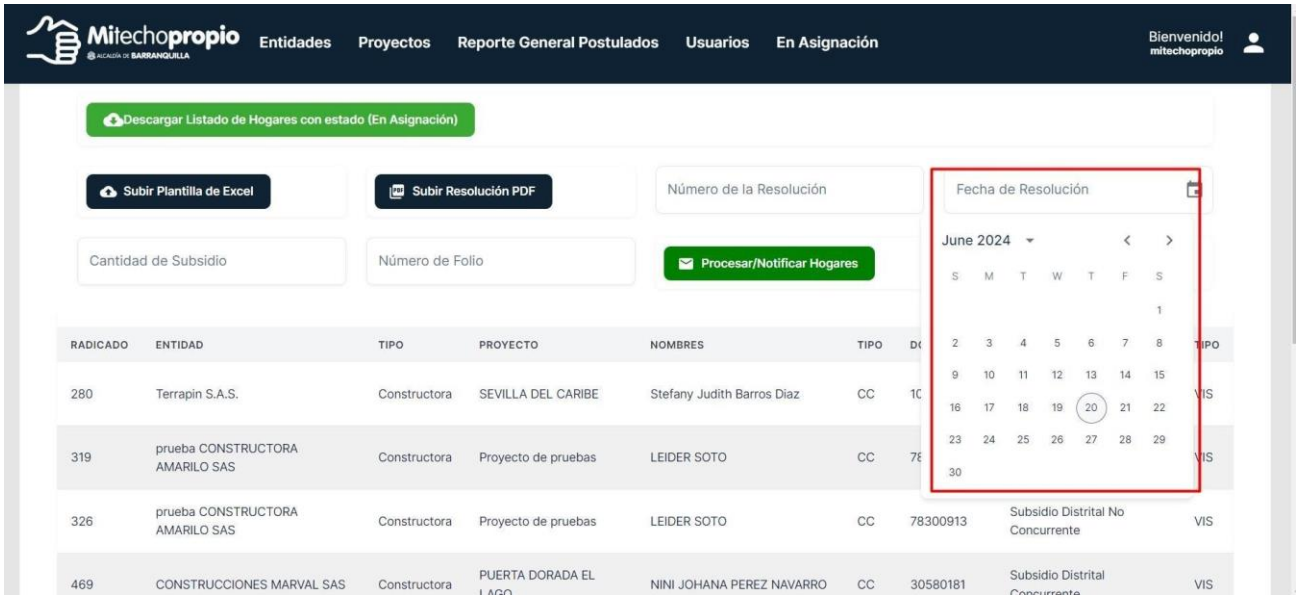
Subir resolución



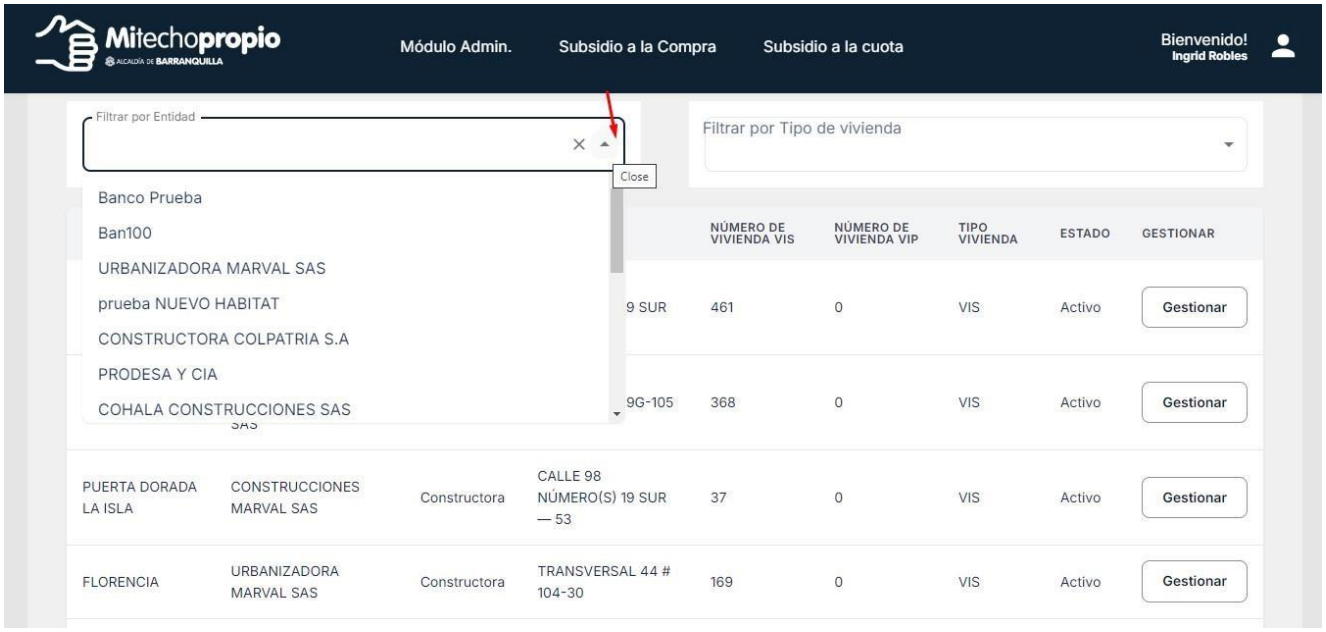
Numero resolución



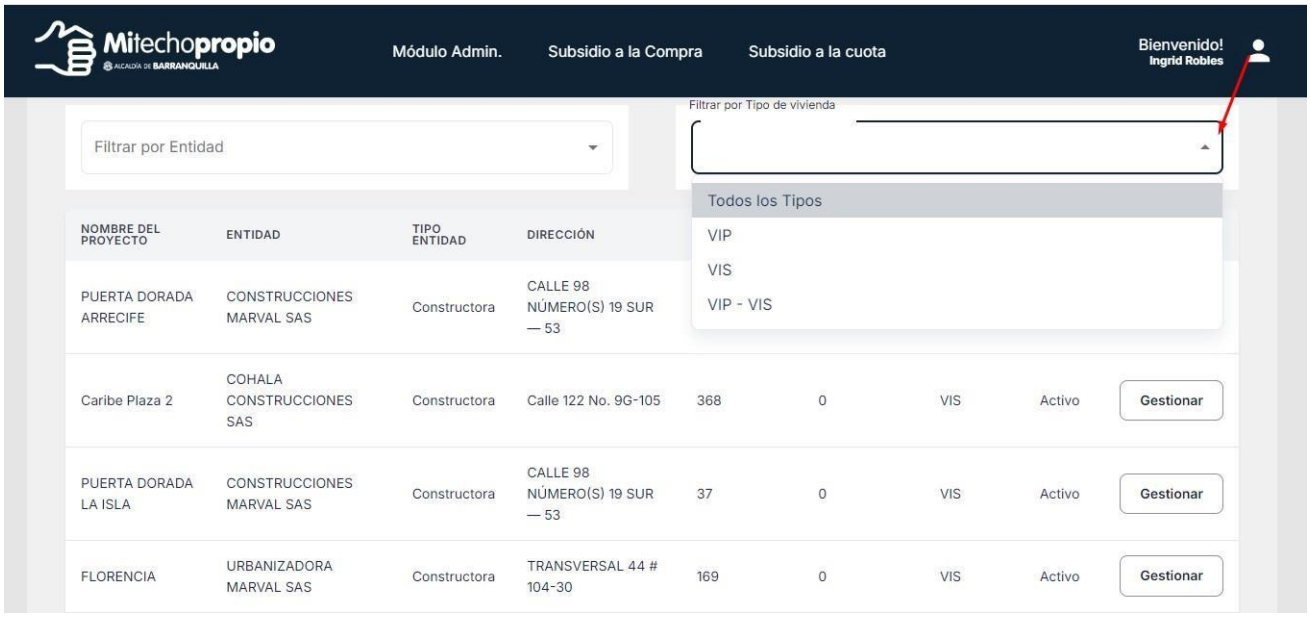




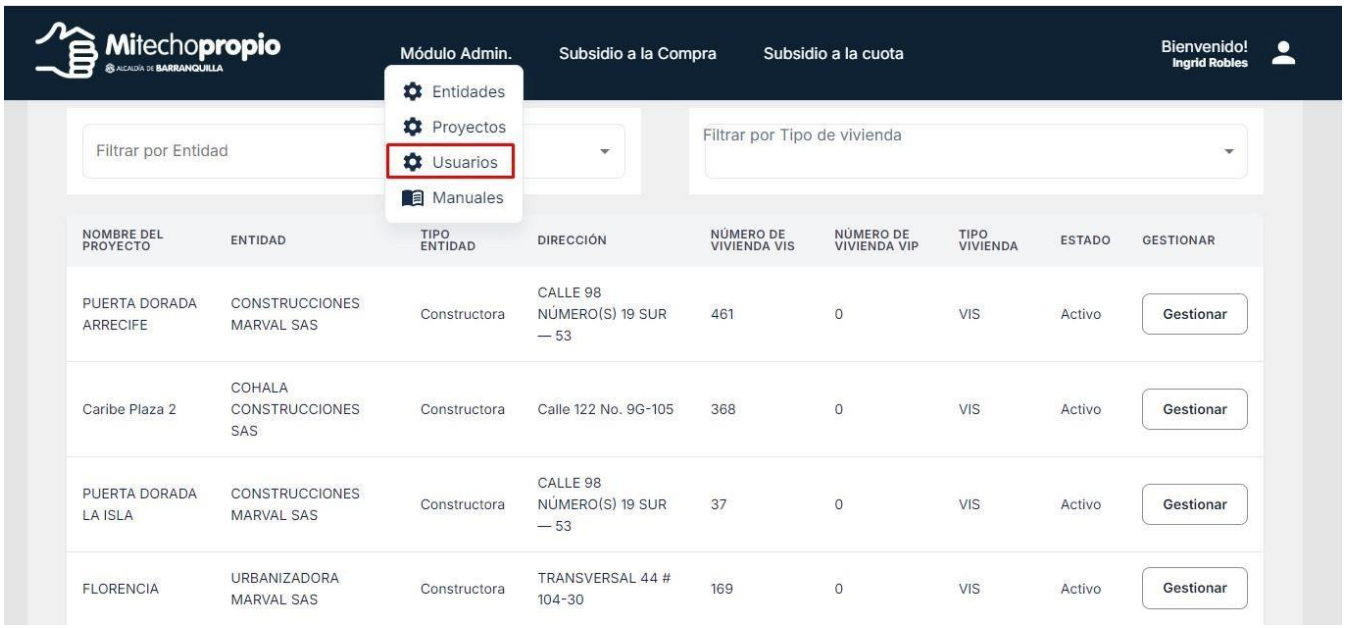
Filtros por entidades



Filtros por vivienda



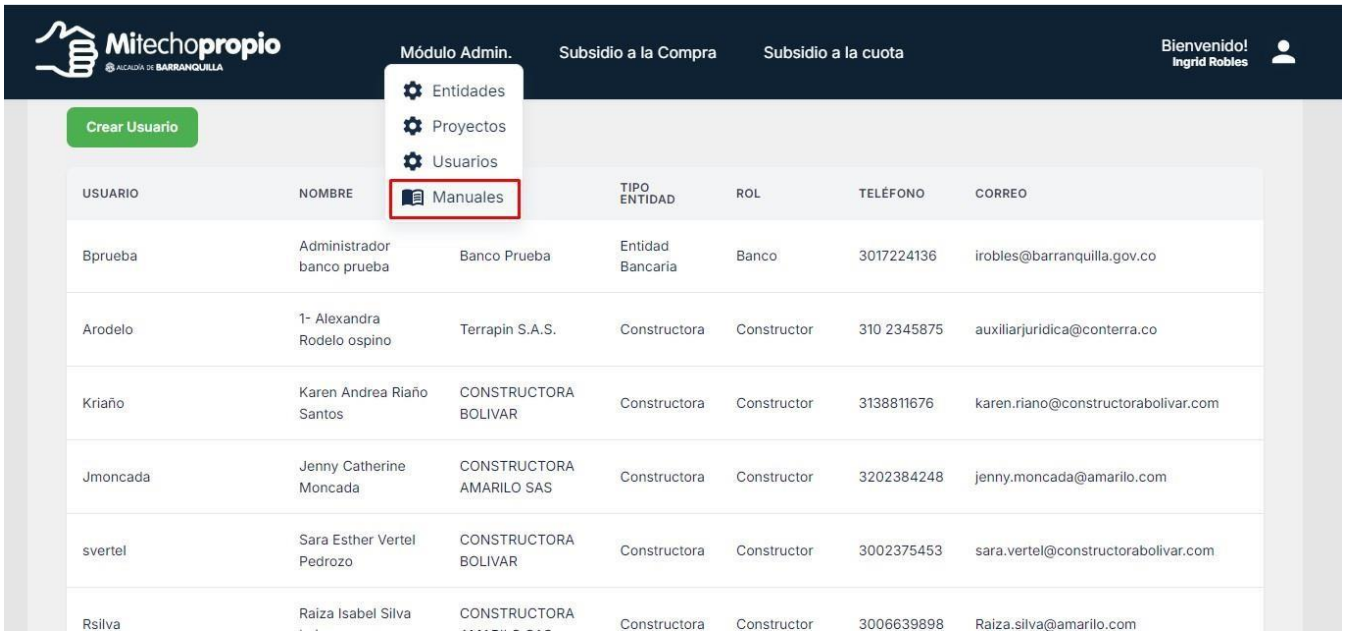
Usuario



Crear usuario



Manual

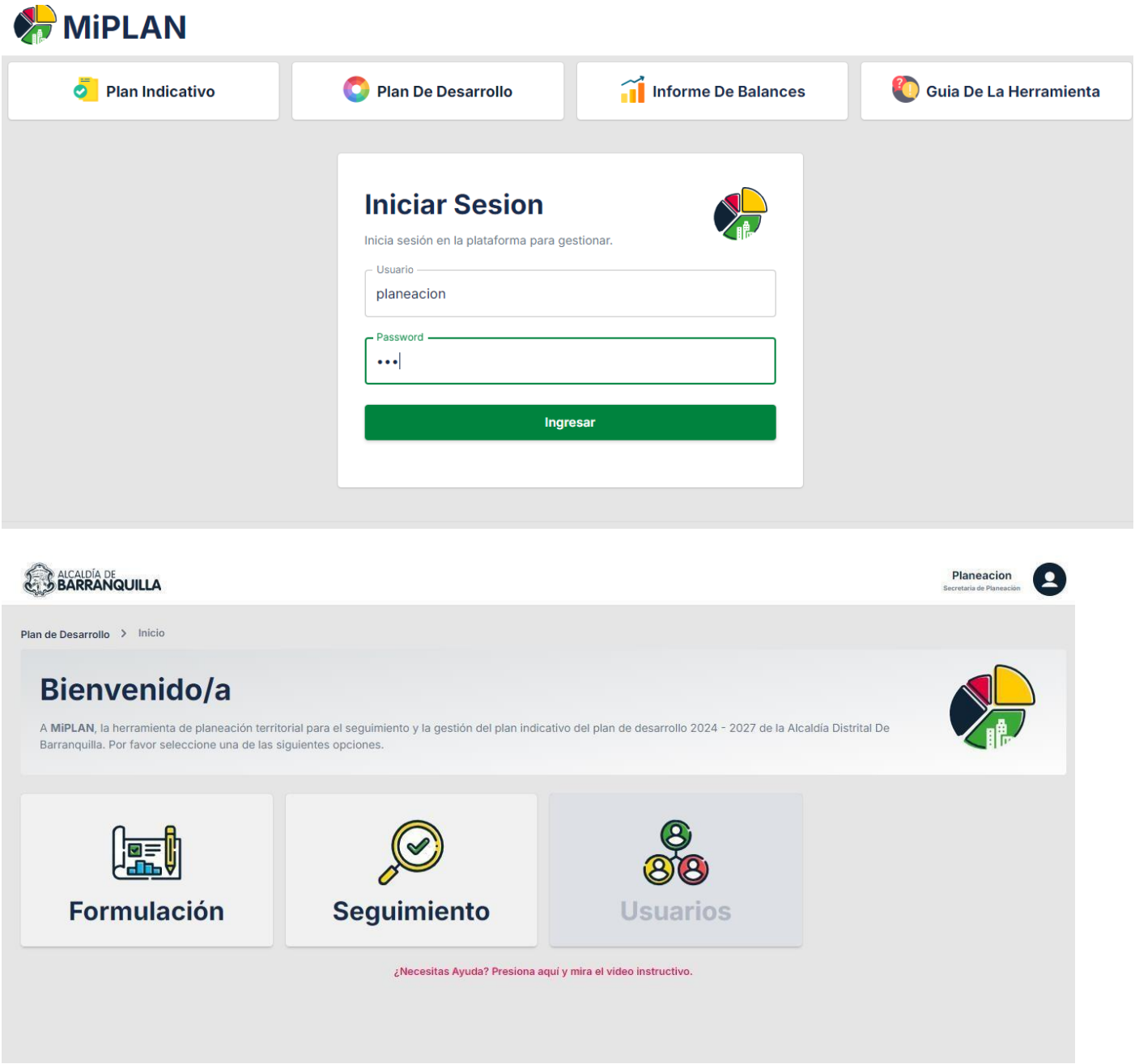


Reporte de apostolados subsidio a la compra

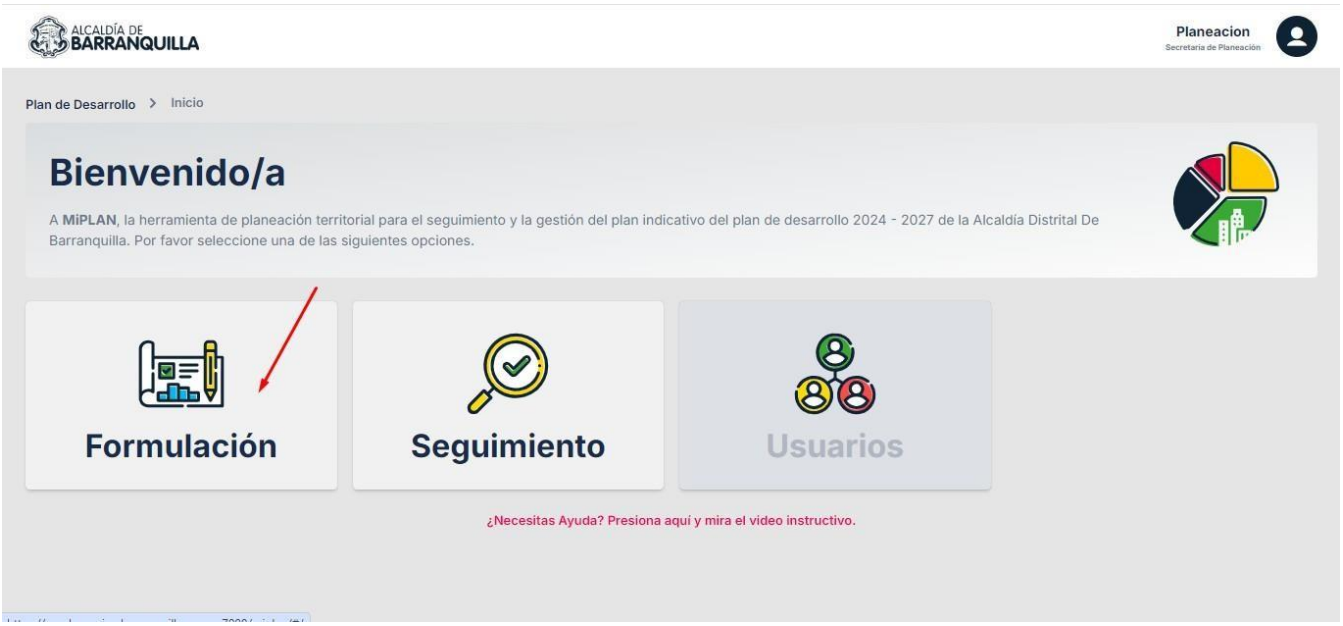

Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
REPORTE DE POSTULADOS SUBSIDIO A LA COMPRA
Buscar
Estado Proceso
Estado Postulación
Descargar el Reporte General de Postulados (en Excel)
RADICADO
ENTIDAD
PROYECTO
NOMBRES
TIPO
DOCUMENTO
SUBSIDIO
TIPO
FECHA ACEPTACIÓN
ESTADO (POS)
R
E
RP
P
222
prueba NUEVO HABITAT
LUNA MAR
Ingrid Torres
CC
32889694
Subsidio Distrital No Concurrente
VIS
03 de May de 2024, 15:45
Aceptac
221
prueba CONSTRUCTORA AMARILO SAS
Proyecto de pruebas
NEIL ALBERTO POVEA
CC
85472449
Subsidio Distrital Concurrente
VIS
03 de May de 2024, 17:14
Aceptac


Mitechopropio
ALCALDÍA DE BARRANQUILLA
Módulo Admin.
Subsidio a la Compra
Subsidio a la cuota
Bienvenido!
Ingrid Robles
REPORTE DE POSTULADOS SUBSIDIO A LA COMPRA
Buscar
Estado Proceso
Estado Postulación
Descargar el Reporte General de Postulados (en Excel)
RADICADO
ENTIDAD
PROYECTO
NOMBRES
TIPO
DOCUMENTO
SUBSIDIO
TIPO
FECHA ACEPTACIÓN
ESTADO (POS)
R
E
RP
P
222
prueba NUEVO HABITAT
LUNA MAR
Ingrid Torres
CC
32889694
Subsidio Distrital No Concurrente
VIS
03 de May de 2024, 15:45
Aceptac
221
prueba CONSTRUCTORA AMARILO SAS
Proyecto de pruebas
NEIL ALBERTO POVEA
CC
85472449
Subsidio Distrital Concurrente
VIS
03 de May de 2024, 17:14
Aceptac

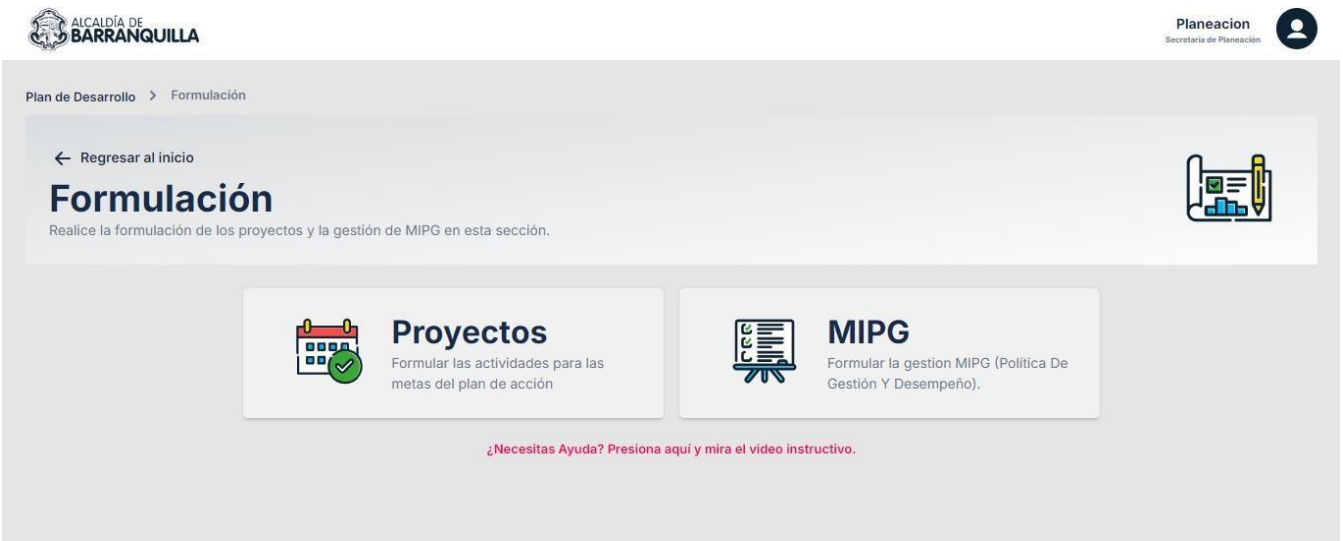
MiPLAN

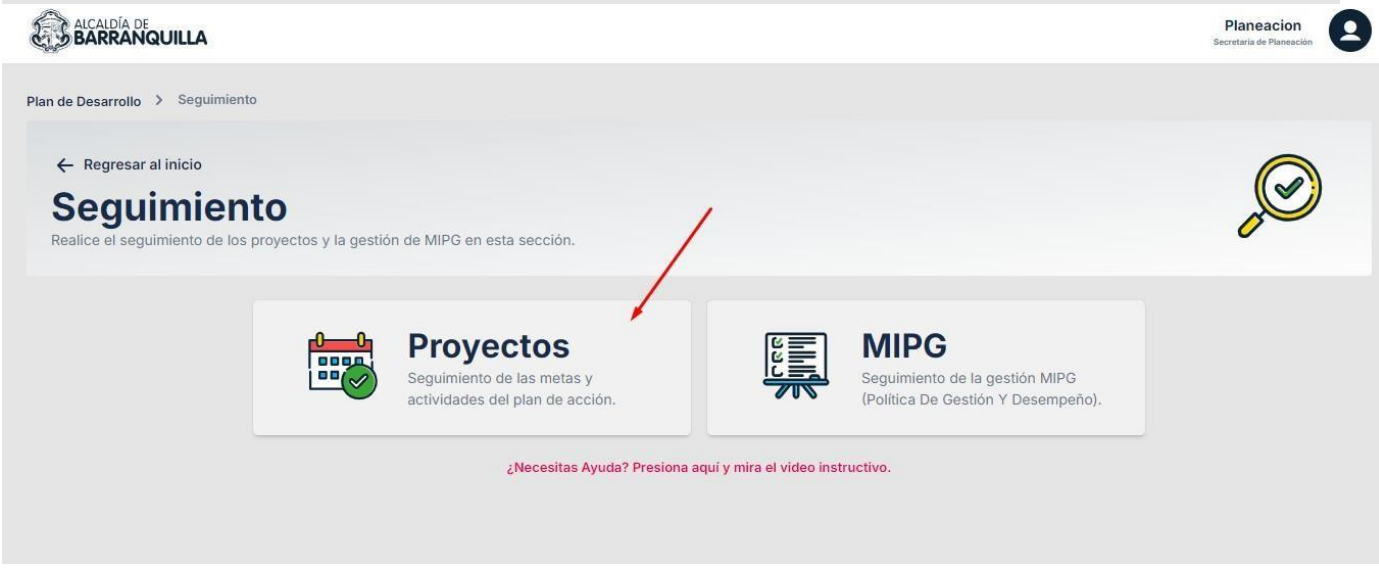


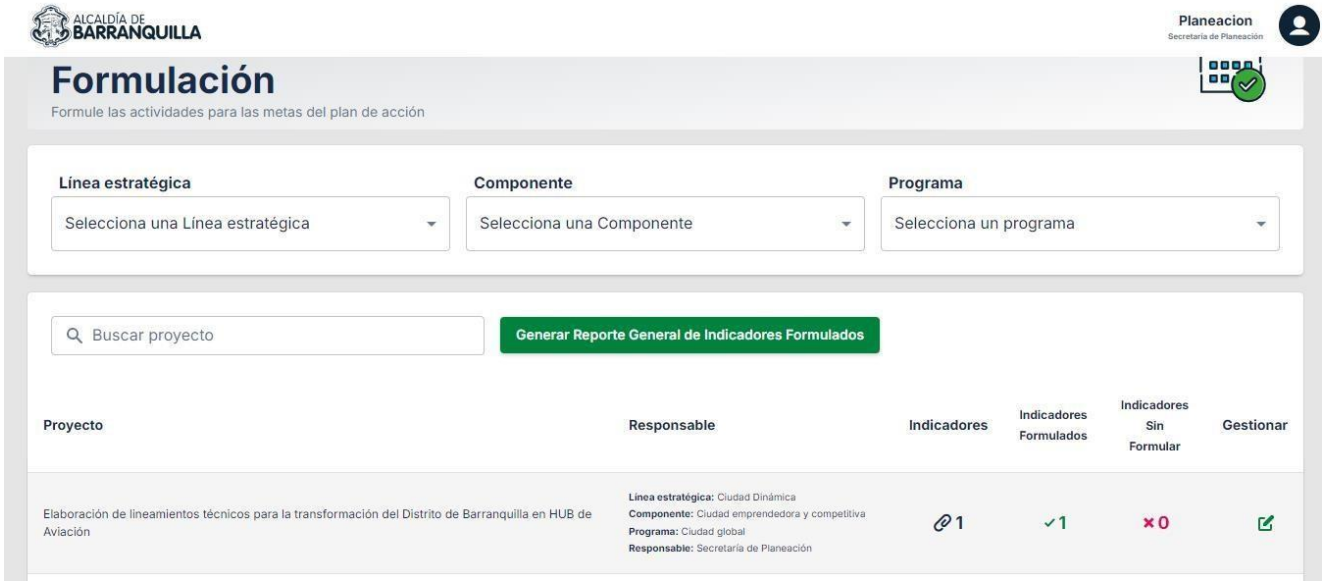
Formulación


ALCALDÍA DE BARRANQUILLA
Planeacion
Secretaría de Planeación
Plan de Desarrollo > Inicio
Bienvenido/a
A MiPLAN, la herramienta de planeación territorial para el seguimiento y la gestión del plan indicativo del plan de desarrollo 2024 - 2027 de la Alcaldía Distrital De Barranquilla. Por favor seleccione una de las siguientes opciones.
Formulación
Seguimiento
Usuarios
¿Necesitas Ayuda? Presiona aquí y mira el video instructivo.

Proyecto


ALCALDÍA DE BARRANQUILLA
Planeacion
Secretaría de Planeación
Plan de Desarrollo > Formulación
← Regresar al inicio
Formulación
Realice la formulación de los proyectos y la gestión de MIPG en esta sección.
Proyectos
Formular las actividades para las metas del plan de acción
MIPG
Formular la gestion MIPG (Política De Gestión Y Desempeño).
¿Necesitas Ayuda? Presiona aquí y mira el video instructivo.

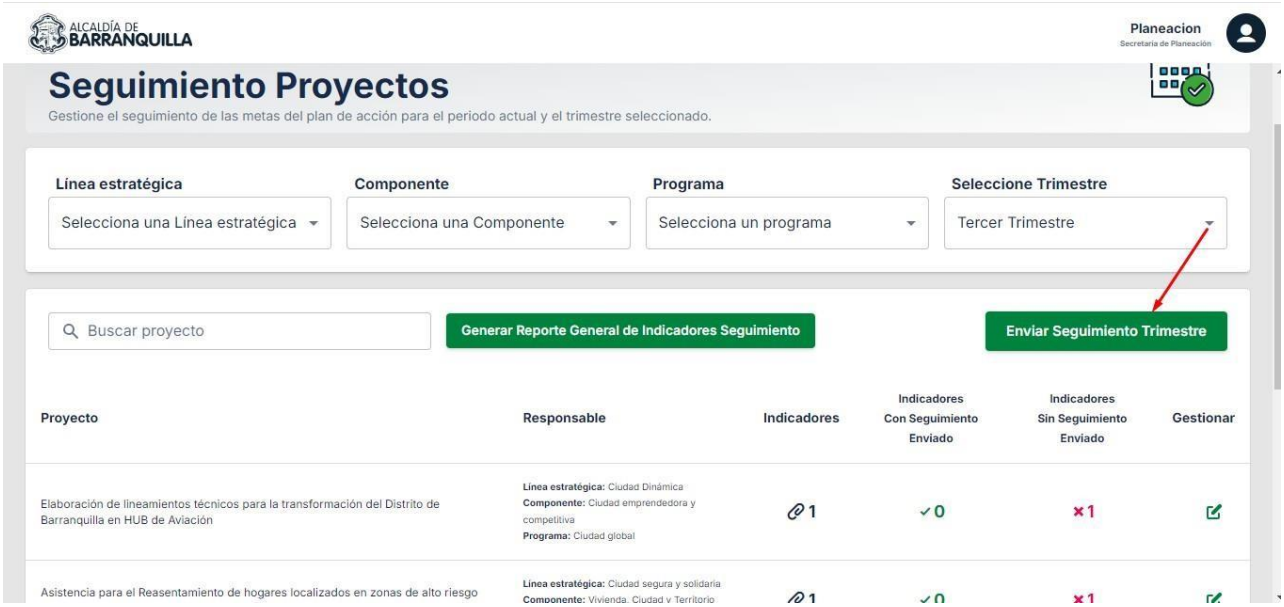

ALCALDÍA DE BARRANQUILLA
Planeacion
Secretaría de Planeación
Plan de Desarrollo > Seguimiento
← Regresar al inicio
Seguimiento
Realice el seguimiento de los proyectos y la gestión de MIPG en esta sección.
Proyectos
Seguimiento de las metas y actividades del plan de acción.
MIPG
Seguimiento de la gestión MIPG (Política De Gestión Y Desempeño).
¿Necesitas Ayuda? Presiona aquí y mira el video instructivo.



Seguimiento




Seguimiento Proyectos
Gestione el seguimiento de las metas del plan de acción para el periodo actual y el trimestre seleccionado.
Línea estratégica
Selecciona una Línea estratégica
Componente
Selecciona una Componente
Programa
Selecciona un programa
Seleccione Trimestre
Tercer Trimestre
Buscar proyecto
Generar Reporte General de Indicadores Seguimiento
Enviar Seguimiento Trimestre
Proyecto
Responsable
Indicadores
Indicadores Con Seguimiento Enviado
Indicadores Sin Seguimiento Enviado
Gestionar
Elaboración de lineamientos técnicos para la transformación del Distrito de Barranquilla en HUB de Aviación
Línea estratégica: Ciudad Dinámica
Componente: Ciudad emprendedora y competitiva
Programa: Ciudad global
1
0
1
Asistencia para el Reasentamiento de hogares localizados en zonas de alto riesgo
Línea estratégica: Ciudad segura y solidaria
Componente: Vivienda, Ciudad y Territorio
1
0
1



MIPG

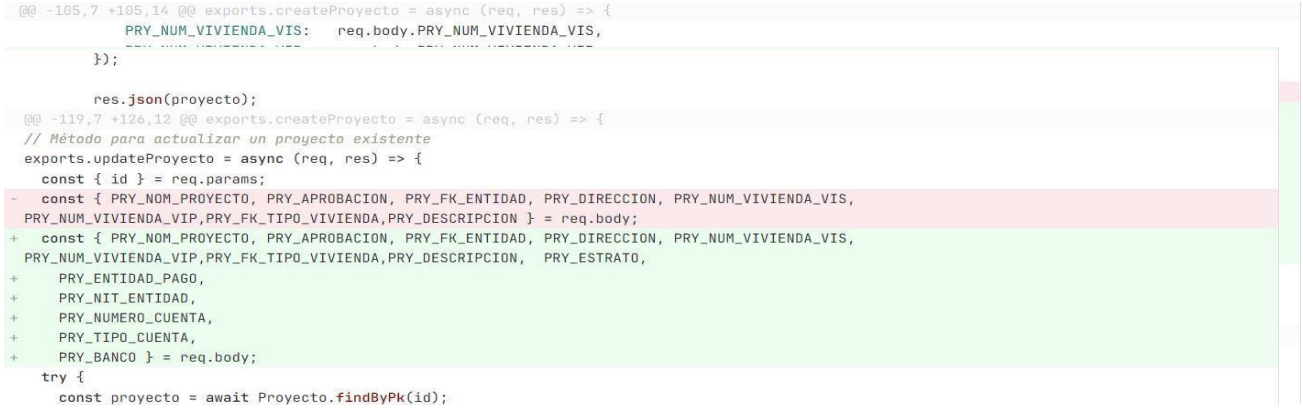


Seguimiento a MIPG


ALCALDÍA DE BARRANQUILLA
Planeacion
Secretaría de Planeación
Plan de Desarrollo > Seguimiento > MIPG
Seguimiento MIPG
Regresar
Gestione el seguimiento de MIPG (POLÍTICA DE GESTIÓN Y DESEMPEÑO) en el trimestre del periodo en curso.
Enviar Seguimiento Trimestre
Metas
24
- Formuladas -
Actividades
39
- Formuladas -
Actividades Gestionadas
1
- Completadas -
Buscar meta
Generar Reporte General de Metas Seguimiento
Meta
Actividades
Responsable
Actividades Gestionadas
Año
% Promedio Avance
Gestionar
Desarrollo del 100% de las acciones tendientes a fortalecer los principios de autocontrol, autogestión y autorregulación para la


ALCALDÍA DE BARRANQUILLA
Planeacion
Secretaría de Planeación
Plan de Desarrollo > Seguimiento > MIPG
Seguimiento MIPG
Regresar
Gestione el seguimiento de MIPG (POLÍTICA DE GESTIÓN Y DESEMPEÑO) en el trimestre del periodo en curso.
Enviar Seguimiento Trimestre
Metas
24
- Formuladas -
Actividades
39
- Formuladas -
Actividades Gestionadas
1
- Completadas -
Buscar meta
Generar Reporte General de Metas Seguimiento
Meta
Actividades
Responsable
Actividades Gestionadas
Año
% Promedio Avance
Gestionar
Desarrollo del 100% de las acciones tendientes a fortalecer los principios de autocontrol, autogestión y autorregulación para la

Evidencias técnicas:
Controllers/mtp/ProyectoController.



```
@@ -105,7 +105,14 @@ exports.createProyecto = async (req, res) => {
            PRY_NUM_VIVIENDA_VIS:   req.body.PRY_NUM_VIVIENDA_VIS,
        });

        res.json(proyecto);
@@ -119,7 +126,12 @@ exports.createProyecto = async (req, res) => {
// Método para actualizar un proyecto existente
exports.updateProyecto = async (req, res) => {
  const { id } = req.params;
-  const { PRY_NOM_PROYECTO, PRY_APROBACION, PRY_FK_ENTIDAD, PRY_DIRECCION, PRY_NUM_VIVIENDA_VIS,
PRY_NUM_VIVIENDA_VIP,PRY_FK_TIPO_VIVIENDA,PRY_DESCRIPCION } = req.body;
+  const { PRY_NOM_PROYECTO, PRY_APROBACION, PRY_FK_ENTIDAD, PRY_DIRECCION, PRY_NUM_VIVIENDA_VIS,
PRY_NUM_VIVIENDA_VIP,PRY_FK_TIPO_VIVIENDA,PRY_DESCRIPCION,  PRY_ESTRATO,
+    PRY_ENTIDAD_PAGO,
+    PRY_NIT_ENTIDAD,
+    PRY_NUMERO_CUENTA,
+    PRY_TIPO_CUENTA,
+    PRY_BANCO } = req.body;
  try {
    const proyecto = await Proyecto.findByPk(id);
```

```
        acerca del cumplimiento de las relaciones y obligaciones que he (mos) tenido con el sector financiero y sobre las
        relaciones y obligaciones que hacía futuro adquiera con dicho sector; (iv) Enviar información de novedades o
@@ -2017,11 +2017,17 @@ exports.HistoricoPostulacionesCuotas = async (req, res) => {
exports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {

try {
-  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos} = req.query;
+  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos,masivo} = req.query;
   const pageSize = 10; // Número de registros por página
   const offset = (page - 1) * pageSize; // Cálculo del offset

   let whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91`;
+  if(masivo==2){
+    whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91 AND HAE_ESTADOCUOTA_ID=2`;
+  }
+
+
+
   let wherePag = '';
   if(page!=0){
     wherePag=` WHERE rn > ${offset} AND ROWNUM <= ${pageSize}`
@@ -2134,8 +2140,8 @@ try {
        EPO_ID: row.EPO_ID,
        GRP_OBSERVACION:row.GRP_OBSERVACION,
        RESOLUCION_CONCATENADA:row.RESOLUCION_CONCATENADA,
+       RESOLUCION_ASIG:row.RESOLUCION_ASIG,
        USUARIO_EJECUTORIA :row.USUARIO_EJECUTORIA,
-       RESOLUCION_ASIG:row.RESOLUCION_ASIG,
        RESOLUCION_CONT_PAGO :row.RESOLUCION_CONT_PAGO,
        MTP_RESOLUCION_PAGO :row.MTP_RESOLUCION_PAGO,
```

```
-   const { PRY_NOM_PROYECTO, PRY_APROBACION, PRY_FK_ENTIDAD, PRY_DIRECCION, PRY_NUM_VIVIENDA_VIS,
 PRY_NUM_VIVIENDA_VIP,PRY_FK_TIPO_VIVIENDA,PRY_DESCRIPCION } = req.body;
+   const { PRY_NOM_PROYECTO, PRY_APROBACION, PRY_FK_ENTIDAD, PRY_DIRECCION, PRY_NUM_VIVIENDA_VIS,
 PRY_NUM_VIVIENDA_VIP,PRY_FK_TIPO_VIVIENDA,PRY_DESCRIPCION,  PRY_ESTRATO,
+     PRY_ENTIDAD_PAGO,
+     PRY_NIT_ENTIDAD,
+     PRY_NUMERO_CUENTA,
+     PRY_TIPO_CUENTA,
+     PRY_BANCO } = req.body;
    try {
      const proyecto = await Proyecto.findByPk(id);
      if (proyecto) {
 @@ -131,7 +143,13 @@ exports.updateProyecto = async (req, res) => {
         PRY_NUM_VIVIENDA_VIS,
         PRY_NUM_VIVIENDA_VIP,
         PRY_FK_TIPO_VIVIENDA,
-        PRY_DESCRIPCION
+        PRY_DESCRIPCION,
+        PRY_ESTRATO,
+        PRY_ENTIDAD_PAGO,
+        PRY_NIT_ENTIDAD,
```

```
-   const { PRY_NOM_PROYECTO, PRY_APROBACION, PRY_FK_ENTIDAD, PRY_DIRECCION, PRY_NUM_VIVIENDA_VIS,
 PRY_NUM_VIVIENDA_VIP,PRY_FK_TIPO_VIVIENDA,PRY_DESCRIPCION } = req.body;
+   const { PRY_NOM_PROYECTO, PRY_APROBACION, PRY_FK_ENTIDAD, PRY_DIRECCION, PRY_NUM_VIVIENDA_VIS,
 PRY_NUM_VIVIENDA_VIP,PRY_FK_TIPO_VIVIENDA,PRY_DESCRIPCION,  PRY_ESTRATO,
+     PRY_ENTIDAD_PAGO,
+     PRY_NIT_ENTIDAD,
+     PRY_NUMERO_CUENTA,
+     PRY_TIPO_CUENTA,
+     PRY_BANCO } = req.body;
    try {
      const proyecto = await Proyecto.findByPk(id);
      if (proyecto) {
 @@ -131,7 +143,13 @@ exports.updateProyecto = async (req, res) => {
         PRY_NUM_VIVIENDA_VIS,
         PRY_NUM_VIVIENDA_VIP,
         PRY_FK_TIPO_VIVIENDA,
-        PRY_DESCRIPCION
+        PRY_DESCRIPCION,
+        PRY_ESTRATO,
+        PRY_ENTIDAD_PAGO,
+        PRY_NIT_ENTIDAD,
+        PRY_NUMERO_CUENTA,
+        PRY_TIPO_CUENTA,
+        PRY_BANCO
```

controllers/mtp/UsuarioController.js

```
 @@ -37,8 +37,10 @@ exports.createUsuario = async (req, res) => {
       USU_FK_ENTIDAD: req.body.USU_FK_ENTIDAD,
       USU_FK_PROYECTO: req.body.USU_FK_PROYECTO,
       USU_USUARIO: req.body.USU_USUARIO,
-      USU_FK_ROL: 2,
-      USU_NOMBRE: req.body.USU_NOMBRE
+      USU_FK_ROL:  req.body.USU_FK_ROL,
+      USU_NOMBRE: req.body.USU_NOMBRE,
+      USU_TELEFONO: req.body.USU_TELEFONO,
+      USU_EMAIL: req.body.USU_EMAIL
     });

     // Crear la relación de usuario y rol
 @@ -48,7 +50,7 @@ exports.createUsuario = async (req, res) => {
       {
         replacements: {
           usuId: siguienteValorSecuencia,
-          rolId: 2
+          rolId: req.body.USU_FK_ROL
         }
```

```
  filename: (req, file, cb) => {
    // Renombrar el archivo usando el ID proporcionado y conservar la extensión original
    const { id } = req.params;
    const ext = path.extname(file.originalname); // Obtiene la extensión del archivo original
    cb(null, `${id}${ext}`); // Renombra el archivo con el ID + extensión
  }
});

// Configurar el middleware de Multer
const upload = multer({
  storage: storage,
  limits: { fileSize: 5 * 1024 * 1024 }, // Límite opcional de 5MB por archivo
}).single('file'); // Solo sube un archivo llamado 'file'

// Controlador para subir el archivo
const uploadFile = async (req, res) => {
  const { id } = req.params;

  upload(req, res, async (err) => {
    if (err) {
      // Manejo de errores de Multer
      return res.status(500).send({ message: 'Error al subir el archivo.', error: err.message });
    }

    if (!req.file) {
      // Si no se subió ningún archivo
      return res.status(400).send({ message: 'No se ha subido ningún archivo.' });
    }
      // Si no se subió ningún archivo
      return res.status(400).send({ message: 'No se ha subido ningún archivo.' });
    }

    try {
      const sqlQuery = `UPDATE MTP_ENTIDAD SET CRT_NOMBRE_CERTIFICADO = :nombreArchivo WHERE CTR_ID = :id`;

      // Ejecutar la consulta
      await sequelize.query(sqlQuery, {
        replacements: {
          nombreArchivo: req.file.filename,
          id: id
        }
      });

      // Si el archivo se subió correctamente, enviamos la respuesta con la información del archivo
      res.send({
        message: 'Archivo subido exitosamente.',
        filename: req.file.filename, // Nombre del archivo renombrado
        path: `/uploads/mitechopropio/certificacion-entidad/${req.file.filename}` // Ruta donde se encuentra el archivo
      });
    } catch (error) {
      // Manejo de errores al ejecutar la consulta
      res.status(500).send({
        status: 500,
        message: "Error al realizar el update",
        detailed_message: error.message
      });
    }


// Controlador para la descarga de archivos
const downloadFile = (req, res) => {
  const filename = req.params.id;
  const filePath = path.join(rutaServerUplod, 'uploads/mitechopropio/certificacion-entidad/', filename);

  console.log('Ruta del archivo:', filePath); // Verifica la ruta

  fs.access(filePath, fs.constants.F_OK, (err) => {
    if (err) {
      return res.status(404).send({ message: 'Archivo no encontrado.' });
    }

    res.download(filePath, (err) => {
      if (err) {
        return res.status(500).send({ message: 'Error al descargar el archivo.' });
      }
    });
  });
};
```

```
+      }
+    });
+  });
+};
+
+
+
+
 module.exports = {
   getAllEntidades,
   getEntidadById,
   addEntidad,
   updateEntidad,
-  toggleEntidadEstado
+  toggleEntidadEstado,
+  uploadFile,
+  downloadFile
```

controllers/mtp/HogarController.js

```
@ -450,7 +450,7 @@ async function  EnviarNotificacion(body,radicado)  {
       irrevocablemente a la Constructora y al Distrito de Barranquilla, para que utilice los datos que he(mos) suministrado
       o que llegue(mos) a suministrar con los siguientes fines: (i) Compartirlos a la Alcaldía de Barranquilla para las
       validaciones que considere necesarias, para realizar el otorgamiento de los beneficios, conforme lo establecido en el
-      Decreto Distrital No 0172 de 2024, y demás normas que lo modifiquen, adicionen o sustituyan, (ii) Consultar mis
+      Decreto Distrital No 0811 de 2024, y demás normas que lo modifiquen, adicionen o sustituyan, (ii) Consultar mis
       (nuestros) datos ante las centrales de información, tanto por el Constructor o como por Alcaldía, (iii) Consultar
       acerca del cumplimiento de las relaciones y obligaciones que he (mos) tenido con el sector financiero y sobre las
       relaciones y obligaciones que hacía futuro adquiera con dicho sector; (iv) Enviar información de novedades o
@@ -2017,11 +2017,17 @@ exports.HistoricoPostulacionesCuotas = async (req, res) => {
 xports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {

 ry {
-  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos} = req.query;
+  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos,masivo} = req.query;
   const pageSize = 10; // Número de registros por página
   const offset = (page - 1) * pageSize; // Cálculo del offset

   let whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91`;
+  if(masivo==2){
+    whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91 AND HAE_ESTADOCUOTA_ID=2`;
+  }
+
+
+
   let wherePag = '';
   if(page!=0){
```

```
       acerca del cumplimiento de las relaciones y obligaciones que he (mos) tenido con el sector financiero y sobre las
       relaciones y obligaciones que hacía futuro adquiera con dicho sector; (iv) Enviar información de novedades o
@@ -2017,11 +2017,17 @@ exports.HistoricoPostulacionesCuotas = async (req, res) => {
 exports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {

 try {
-  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos} = req.query;
+  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos,masivo} = req.query;
   const pageSize = 10; // Número de registros por página
   const offset = (page - 1) * pageSize; // Cálculo del offset

   let whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91`;
+  if(masivo==2){
+    whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91 AND HAE_ESTADOCUOTA_ID=2`;
+  }
+
+
+
   let wherePag = '';
   if(page!=0){
     wherePag=` WHERE rn > ${offset} AND ROWNUM <= ${pageSize}`
@@ -2134,8 +2140,8 @@ try {
       EPO_ID: row.EPO_ID,
       GRP_OBSERVACION:row.GRP_OBSERVACION,
       RESOLUCION_CONCATENADA:row.RESOLUCION_CONCATENADA,
+      RESOLUCION_ASIG:row.RESOLUCION_ASIG,
       USUARIO_EJECUTORIA :row.USUARIO_EJECUTORIA,
-      RESOLUCION_ASIG:row.RESOLUCION_ASIG,
       RESOLUCION_CONT_PAGO :row.RESOLUCION_CONT_PAGO,
       MTP_RESOLUCION_PAGO :row.MTP_RESOLUCION_PAGO,
```

```
    });

    // Crear la relación de usuario y rol
@@ -48,7 +50,7 @@ exports.createUsuario = async (req, res) => {
      {
        replacements: {
          usuId: siguienteValorSecuencia,
-         rolId: 2
+         rolId: req.body.USU_FK_ROL
        },
        type: sequelize.QueryTypes.INSERT
      }
@@ -67,23 +69,9 @@ exports.getAllUsuarios = async (req, res) => {
  try {
    const usuarios = await sequelize.query(`
      SELECT
-       MTP_USUARIO.USU_ID,
-       MTP_USUARIO.USU_USUARIO,
-       MTP_USUARIO.USU_NOMBRE,
-       MTP_USUARIO.USU_PASSWORD,
-       MTP_USUARIO.USU_FK_ROL,
-       MTP_USUARIO.USU_ESTADO,
-       MTP_USUARIO.USU_FK_PROYECTO,
-       MTP_USUARIO.USU_FK_ENTIDAD,
-       MTP_TIPO_ENTIDAD.ENT_TIPO_ENTIDAD,
-       MTP_ENTIDAD.CTR_NOMBRE,
-       MTP_ENTIDAD.CTR_CORREO_ELECTRONICO,
-       MTP_ENTIDAD.ENT_FK_TIPO_ENTIDAD
+       *

-       MTP_USUARIO
-       INNER JOIN MTP_ENTIDAD ON MTP_USUARIO.USU_FK_ENTIDAD = MTP_ENTIDAD.CTR_ID
-       INNER JOIN MTP_TIPO_ENTIDAD ON MTP_ENTIDAD.ENT_FK_TIPO_ENTIDAD = MTP_TIPO_ENTIDAD.ENT_ID
-       ORDER BY MTP_USUARIO.USU_ID DESC
+       VIEW_LISTA_USUARIOS
    `, {

      type: QueryTypes.SELECT
@@ -115,6 +103,31 @@ exports.updateUsuario = async (req, res) => {
    const usuario = await Usuario.findByPk(req.params.id);
    if (usuario) {
      await usuario.update(req.body);
+
+      // Primero realiza el DELETE
+      const deleteResult = await sequelize.query(
+        'DELETE FROM "MTP_USUARIO_ROL" WHERE "USU_ID" = :usuId',
+        {
+          replacements: {
+            usuId: req.params.id  // El ID del usuario que quieres eliminar
+          },
+          type: sequelize.QueryTypes.DELETE
+        }
+      );
+
+      // Luego realiza el INSERT
+      const insertResult = await sequelize.query(
+        'INSERT INTO "MTP_USUARIO_ROL" ("USU_ID", "ROL_ID") VALUES (:usuId, :rolId)',
+        {
+          replacements: {
```

```
          type: sequelize.QueryTypes.DELETE
        }
      );

      // Luego realiza el INSERT
      const insertResult = await sequelize.query(
        'INSERT INTO "MTP_USUARIO_ROL" ("USU_ID", "ROL_ID") VALUES (:usuId, :rolId)',
        {
          replacements: {
            usuId: req.params.id,
            rolId: req.body.USU_FK_ROL  // El ID del rol que deseas insertar
          },
          type: sequelize.QueryTypes.INSERT
        }
      );


    res.json({ mensaje: 'Usuario actualizado correctamente' });
  } else {
    res.status(404).json({ error: 'Usuario no encontrado' });
@@ -226,6 +239,62 @@ exports.loginValida = async (req, res) => {
};


exports.cambioContrasena = async (req, res) => {
  try {
    const { USU_ID, USU_PASSWORD, NUEVA_PASSWORD } = req.body;


  }

  if (NUEVA_PASSWORD.length < 6) {
    return res.status(400).json({ error: 'La nueva contraseña debe tener al menos 6 caracteres' });
  }

  // Verificar si el usuario y la contraseña actual son correctos
  const user = await sequelize.query(`
    SELECT
      USU_ID,
      USU_USUARIO,
      USU_NOMBRE,
      PERMISOS,
      USU_FK_ENTIDAD
    FROM
      VIEW_PERMISO_USUARIO
    WHERE
      USU_ESTADO=1 AND
      USU_ID = :USU_ID
  `, {
    replacements: { USU_ID },
    type: QueryTypes.SELECT
  });

  if (user.length === 0) {
    return res.status(401).json({ error: 'Usuario o contraseña incorrectos' });
  }
```

```
    await sequelize.query(
      UPDATE MTP_USUARIO SET
        USU_PASSWORD = :USU_PASSWORD
      WHERE
        USU_ID = :USU_ID
    `, {
      replacements: { USU_PASSWORD, USU_ID },
      type: QueryTypes.UPDATE
    });


    res.status(200).json({ message: 'Contraseña cambiada con éxito' });

  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Error al cambiar la contraseña' });
  }
};
\ No newline at end of file
```

controllers/mtp/UtilCo

```
@@ -381,6 +381,8 @@ function obtenerDirectorio(tipo) {
      return rutaServerUplod + 'uploads/mitechopropio/en-asignacion/';
    case 4:
      return rutaServerUplod + 'uploads/mitechopropio/en-ejecutorias/';
+   case 5:
+     return rutaServerUplod + 'uploads/mitechopropio/en-marcado-pago/';
    default:
      // Ajusta esto según tus necesidades si el tipo no es 1 ni 3
      return rutaServerUplod + 'uploads/mitechopropio/';
@@ -1431,6 +1433,18 @@ exports.EnAsigancion = async (req, res) => {
        const cellRef = xlsx.utils.encode_cell({ r: rowNum, c: 0 }); // Columna A (c: 0)
        // Obtiene el contenido de la celda en la primera columna
        const cellValue = sheet[cellRef] ? sheet[cellRef].v : null;
+
+
+       const cellRefC = xlsx.utils.encode_cell({ r: rowNum, c: 9 }); // Columna J (c: 9)
+       // Obtiene el contenido de la celda en la primera columna
+       const cellValueCantidad = sheet[cellRefC] ? sheet[cellRefC].v : null;
+
+       const cellRefV = xlsx.utils.encode_cell({ r: rowNum, c: 10 }); // Columna J (c: 9)
+       // Obtiene el contenido de la celda en la primera columna
+       const cellValueValor = sheet[cellRefV] ? sheet[cellRefV].v : null;
+
+
+
        // Agrega el contenido de la celda al array de datos de la primera columna
```

```
       // Agrega el contenido de la celda al array de datos de la primera columna
       if(cellValue>0){

@@ -1453,13 +1467,15 @@ exports.EnAsigancion = async (req, res) => {
         if (!existingRecord || existingRecord.length === 0) {  console.log('LEIDER ANDRES SOTO');
         // Si no existe, insertar el nuevo registro
         await sequelize.query(
-          `INSERT INTO MTP_RESOLUCION_HOGAR (RHO_FK_RESOLUCION, RHO_FK_HOGAR, RHO_USUARIO, RHO_ESTADO, RHO_FECHA)
-          VALUES (:resolucionId, :hogarId, :usuario, :estado, SYSDATE)`,
+          `INSERT INTO MTP_RESOLUCION_HOGAR (RHO_FK_RESOLUCION, RHO_FK_HOGAR, RHO_USUARIO, RHO_ESTADO,
RHO_FECHA,RHO_NUM_SMLMV,RHO_VALOR_SUBSIDIO)
          VALUES (:resolucionId, :hogarId, :usuario, :estado, SYSDATE, :cantidad, :valor)`,
          {
            replacements: {
              resolucionId: secuenciaValue,
              hogarId: cellValue,
              usuario: usuario_id,
+             cantidad: cellValueCantidad,
+             valor: cellValueValor,
              estado: 1
            }
          }
@@ -1983,4 +1999,166 @@ exports.getEscrituraHogar = async (req, res) => {
    console.error(error);
```

controllers/mtp/UsuarioController.js

```
@@ -37,8 +37,10 @@ exports.createUsuario = async (req, res) => {
      USU_FK_ENTIDAD: req.body.USU_FK_ENTIDAD,
      USU_FK_PROYECTO: req.body.USU_FK_PROYECTO,
      USU_USUARIO: req.body.USU_USUARIO,
-     USU_FK_ROL: 2,
-     USU_NOMBRE: req.body.USU_NOMBRE
+     USU_FK_ROL:  req.body.USU_FK_ROL,
+     USU_NOMBRE: req.body.USU_NOMBRE,
+     USU_TELEFONO: req.body.USU_TELEFONO,
+     USU_EMAIL: req.body.USU_EMAIL
    });

    // Crear la relación de usuario y rol
@@ -48,7 +50,7 @@ exports.createUsuario = async (req, res) => {
      {
        replacements: {
          usuId: siguienteValorSecuencia,
-         rolId: 2
+         rolId: req.body.USU_FK_ROL
        }
```

```
        // Agrega el contenido de la celda al array de datos de la primera columna
        if(cellValue>0){

@@ -1453,13 +1467,15 @@ exports.EnAsigancion = async (req, res) => {
          if (!existingRecord || existingRecord.length === 0) {  console.log('LEIDER ANDRES SOTO');
          // Si no existe, insertar el nuevo registro
          await sequelize.query(
-           `INSERT INTO MTP_RESOLUCION_HOGAR (RHO_FK_RESOLUCION, RHO_FK_HOGAR, RHO_USUARIO, RHO_ESTADO, RHO_FECHA)
-           VALUES (:resolucionId, :hogarId, :usuario, :estado, SYSDATE)`,
+           `INSERT INTO MTP_RESOLUCION_HOGAR (RHO_FK_RESOLUCION, RHO_FK_HOGAR, RHO_USUARIO, RHO_ESTADO,
RHO_FECHA,RHO_NUM_SMLMV,RHO_VALOR_SUBSIDIO)
+           VALUES (:resolucionId, :hogarId, :usuario, :estado, SYSDATE, :cantidad, :valor)`,
            {
              replacements: {
                resolucionId: secuenciaValue,
                hogarId: cellValue,
                usuario: usuario_id,
+               cantidad: cellValueCantidad,
+               valor: cellValueValor,
                estado: 1
              }
            }
@@ -1983,4 +1999,166 @@ exports.getEscrituraHogar = async (req, res) => {
    console.error(error);
```

```
    });

    // Crear la relación de usuario y rol
@@ -48,7 +50,7 @@ exports.createUsuario = async (req, res) => {
      {
        replacements: {
          usuId: siguienteValorSecuencia,
-         rolId: 2
+         rolId: req.body.USU_FK_ROL
        },
        type: sequelize.QueryTypes.INSERT
      }
@@ -67,23 +69,9 @@ exports.getAllUsuarios = async (req, res) => {
  try {
    const usuarios = await sequelize.query(`
        SELECT
-       MTP_USUARIO.USU_ID,
-       MTP_USUARIO.USU_USUARIO,
-       MTP_USUARIO.USU_NOMBRE,
-       MTP_USUARIO.USU_PASSWORD,
-       MTP_USUARIO.USU_FK_ROL,
-       MTP_USUARIO.USU_ESTADO,
-       MTP_USUARIO.USU_FK_PROYECTO,
-       MTP_USUARIO.USU_FK_ENTIDAD,
-       MTP_TIPO_ENTIDAD.ENT_TIPO_ENTIDAD,
-       MTP_ENTIDAD.CTR_NOMBRE,
-       MTP_ENTIDAD.CTR_CORREO_ELECTRONICO,
-       MTP_ENTIDAD.ENT_FK_TIPO_ENTIDAD
+       *
```

```
       acerca del cumplimiento de las relaciones y obligaciones que he (mos) tenido con el sector financiero y sobre las
       relaciones y obligaciones que hacía futuro adquiera con dicho sector; (iv) Enviar información de novedades o
@@ -2017,11 +2017,17 @@ exports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {
exports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {

try {
-  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos} = req.query;
+  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos,masivo} = req.query;
  const pageSize = 10; // Número de registros por página
  const offset = (page - 1) * pageSize; // Cálculo del offset

  let whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91`;
+  if(masivo==2){
+    whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91 AND HAE_ESTADOCUOTA_ID=2`;
+  }
+
+
+
  let wherePag = '';
  if(page!=0){
     wherePag=` WHERE rn > ${offset} AND ROWNUM <= ${pageSize}`
@@ -2134,8 +2140,8 @@ try {
        EPO_ID: row.EPO_ID,
        GRP_OBSERVACION:row.GRP_OBSERVACION,
        RESOLUCION_CONCATENADA:row.RESOLUCION_CONCATENADA,
+       RESOLUCION_ASIG:row.RESOLUCION_ASIG,
        USUARIO_EJECUTORIA :row.USUARIO_EJECUTORIA,
-       RESOLUCION_ASIG:row.RESOLUCION_ASIG,
        RESOLUCION_CONT_PAGO :row.RESOLUCION_CONT_PAGO,
        MTP_RESOLUCION_PAGO :row.MTP_RESOLUCION_PAGO,
```

controllers/mi-plan/pd-actividades.js

```
@@ -737,13 +737,56 @@ const AnexoProyectoMetaIndicador = async (req, res) => {
        }

        try {
+           /*
+
            // Consulta SQL para actualizar el nombre del anexo
            const sqlQuery_EXE = `UPDATE PD_ACTIVIDADES SET ACT_NOMBRE_ANEXO = :name_file WHERE ACT_ID = :idActividad`;
-           const binds = { name_file, idActividad };
+
            // Definir los valores que se van a enlazar (asegúrate de que name_file y idActividad están definidas)
            const binds = {
            name_file: name_file,
            idActividad: idActividad
            };

            // Ejecutar la consulta
            await connection.execute(sqlQuery_EXE, binds, { autoCommit: true });
+
           */
+
+           const sqlQuery = `UPDATE PD_ACTIVIDADES SET ACT_NOMBRE_ANEXO = :1 WHERE ACT_ID = :2`;
+
+           binds = [name_file,idActividad];
+
+           connection.execute(sqlQuery, binds, {autoCommit: true}, function (err) {
+               if (err) {
+                   res.set('Content-Type', 'application/json');
+                   res.status(500).send(JSON.stringify({
+                       status: 500,
```

```
const sqlQuery = `UPDATE PD_ACTIVIDADES SET ACT_NOMBRE_ANEXO = :1 WHERE ACT_ID = :2`;

binds = [name_file,idActividad];

connection.execute(sqlQuery, binds, {autoCommit: true}, function (err) {
    if (err) {
        res.set('Content-Type', 'application/json');
        res.status(500).send(JSON.stringify({
            status: 500,
            message: "Error al realizar el update",
            detailed_message: err.message
        }));
    } else {
        connection.commit();
         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"));

    }
    // Release the connection
    connection.release(
        function (err) {
            if (err) {
                console.error(err.message);
            } else {
                console.log("PUT /sendTablespace : Connection released");
```

```
                });
            });


            // Guardar el archivo en el servidor
            fs.writeFile(filePath, buffer, (err) => {
                if (err) {
-860,7 +903,47 @@ const storage = multer.diskStorage({
    // Si no se subió ningún archivo
    return res.status(400).send({ message: 'No se ha subido ningún archivo.' });
  }

' Establecer conexión con Oracle
oracledb.getConnection(connAttrs, async (err, connection) => {
 if (err) {
    // Error al conectar a la base de datos
    console.error("Error al conectar a la base de datos:", err);
    return res.status(500).json({
        status: 500,
        message: "Error al conectar a la base de datos",
        detailed_message: err.message
    });
 }

  const sqlQuery = `UPDATE PD_ACTIVIDADES SET ACT_NOMBRE_ANEXO = :1 WHERE ACT_ID = :2`;

  binds = [req.file.filename,req.body.id];

    const sqlQuery = `UPDATE PD_ACTIVIDADES SET ACT_NOMBRE_ANEXO = :1 WHERE ACT_ID = :2`;

    binds = [req.file.filename,req.body.id];

    connection.execute(sqlQuery, binds, {autoCommit: true}, function (err) {
        if (err) {
            res.set('Content-Type', 'application/json');
            res.status(500).send(JSON.stringify({
                status: 500,
                message: "Error al realizar el update",
                detailed_message: err.message
            }));
        } else {
            connection.commit();

        }
        // Release the connection
        connection.release(
            function (err) {
                if (err) {
                    console.error(err.message);
                } else {
                    console.log("PUT /sendTablespace : Connection released");
                }
```

controllers/mtp/EntidadController.js

```
 const sequelize = require('../../config/config_mtp.js');
 const { QueryTypes } = require('sequelize');
+const multer = require('multer');
+const path = require('path');
+const fs = require('fs');
+
+require('dotenv').config();
+const rutaServerUplod =process.env.RUTA_SERVICIOS_UPLOAD;
 /*
 async function getAllEntidades(req, res) {
   try {
 @@ -49,6 +55,8 @@ async function getEntidadById(req, res) {
 }
 async function addEntidad(req, res) {
     try {
+      console.log(req.body);
+
         // Validar que los campos requeridos no estén vacíos
         const requiredFields = ['CTR_NOMBRE','CTR_CORREO_ELECTRONICO','ENT_FK_TIPO_ENTIDAD','CTR_TELEFONO'];
         for (const field of requiredFields) {
 @@ -70,10 +78,14 @@ async function addEntidad(req, res) {
         const nextVal = sequenceResult[0].NEXTVAL;

         let nombreEntidad = req.body.CTR_NOMBRE;
+        let TipoCuentaEntidad =4;
         if(req.body.ENT_FK_TIPO_ENTIDAD==2){
           nombreEntidad = await obtenerNombreEntidadPorId(req.body.CTR_FK_ENTIDAD_FIN);
```

```
+       console.log(req.body);
+
           // Validar que los campos requeridos no estén vacíos
          const requiredFields = ['CTR_NOMBRE','CTR_CORREO_ELECTRONICO','ENT_FK_TIPO_ENTIDAD','CTR_TELEFONO'];
          for (const field of requiredFields) {
 @@ -70,10 +78,14 @@ async function addEntidad(req, res) {
          const nextVal = sequenceResult[0].NEXTVAL;

          let nombreEntidad = req.body.CTR_NOMBRE;
+         let TipoCuentaEntidad =4;
          if(req.body.ENT_FK_TIPO_ENTIDAD==2){
            nombreEntidad = await obtenerNombreEntidadPorId(req.body.CTR_FK_ENTIDAD_FIN);
+           TipoCuentaEntidad =3;
          }

+
+
         // Crear la entidad con el próximo valor de la secuencia y los datos recibidos en el cuerpo de la solicitud
         const entidad = await Entidad.create({
             CTR_ID: nextVal,
 @@ -82,7 +94,10 @@ async function addEntidad(req, res) {
             CTR_TELEFONO: req.body.CTR_TELEFONO,
             ENT_FK_TIPO_ENTIDAD: req.body.ENT_FK_TIPO_ENTIDAD,
             CTR_NIT: req.body.CTR_NIT,
-            CTR_FK_ENTIDAD_FIN: req.body.CTR_FK_ENTIDAD_FIN
+            CTR_FK_ENTIDAD_FIN: req.body.CTR_FK_ENTIDAD_FIN,
+            CTR_TIPO_CUENTA: TipoCuentaEntidad,
            CTR_TELEFONO: req.body.CTR_TELEFONO,
            ENT_FK_TIPO_ENTIDAD: req.body.ENT_FK_TIPO_ENTIDAD,
            CTR_NIT: req.body.CTR_NIT,
            CTR_FK_ENTIDAD_FIN: req.body.CTR_FK_ENTIDAD_FIN
            CTR_FK_ENTIDAD_FIN: req.body.CTR_FK_ENTIDAD_FIN,
            CTR_TIPO_CUENTA: TipoCuentaEntidad,
            CRT_NUMERO_CUENTA: req.body.CRT_NUMERO_CUENTA,
            CRT_NUM_PORTAFOLIO: req.body.CRT_NUM_PORTAFOLIO

       });

@@ -103,15 +118,13 @@ function validateEmail(email) {

async function updateEntidad(req, res) {
  const { id } = req.params;
  let  { CTR_NOMBRE, CTR_CORREO_ELECTRONICO, CTR_TELEFONO,CTR_NIT,CTR_FK_ENTIDAD_FIN,ENT_FK_TIPO_ENTIDAD } = req.body;

  let  { CTR_NOMBRE, CTR_CORREO_ELECTRONICO,
CTR_TELEFONO,CTR_NIT,CTR_FK_ENTIDAD_FIN,ENT_FK_TIPO_ENTIDAD,CTR_TIPO_CUENTA,CRT_NUMERO_CUENTA,CRT_NUM_PORTAFOLIO } = req.body;


  // Si el tipo de entidad es 2, obtener el nombre de la entidad financiera
  if (ENT_FK_TIPO_ENTIDAD == 2) {
    try {
```

controllers/mtp/UsuarioController.js

```
 @@ -37,8 +37,10 @@ exports.createUsuario = async (req, res) => {
      USU_FK_ENTIDAD: req.body.USU_FK_ENTIDAD,
      USU_FK_PROYECTO: req.body.USU_FK_PROYECTO,
      USU_USUARIO: req.body.USU_USUARIO,
-     USU_FK_ROL: 2,
-     USU_NOMBRE: req.body.USU_NOMBRE
+     USU_FK_ROL:  req.body.USU_FK_ROL,
+     USU_NOMBRE: req.body.USU_NOMBRE,
+     USU_TELEFONO: req.body.USU_TELEFONO,
+     USU_EMAIL: req.body.USU_EMAIL
    });

    // Crear la relación de usuario y rol
 @@ -48,7 +50,7 @@ exports.createUsuario = async (req, res) => {
      {
        replacements: {
          usuId: siguienteValorSecuencia,
-         rolId: 2
+         rolId: req.body.USU_FK_ROL
        }
```

```
    });

    // Crear la relación de usuario y rol
@@ -48,7 +50,7 @@ exports.createUsuario = async (req, res) => {
      {
        replacements: {
          usuId: siguienteValorSecuencia,
-         rolId: 2
+         rolId: req.body.USU_FK_ROL
        },
        type: sequelize.QueryTypes.INSERT
      }
@@ -67,23 +69,9 @@ exports.getAllUsuarios = async (req, res) => {
  try {
    const usuarios = await sequelize.query(`
        SELECT
-       MTP_USUARIO.USU_ID,
-       MTP_USUARIO.USU_USUARIO,
-       MTP_USUARIO.USU_NOMBRE,
-       MTP_USUARIO.USU_PASSWORD,
-       MTP_USUARIO.USU_FK_ROL,
-       MTP_USUARIO.USU_ESTADO,
-       MTP_USUARIO.USU_FK_PROYECTO,
-       MTP_USUARIO.USU_FK_ENTIDAD,
-       MTP_TIPO_ENTIDAD.ENT_TIPO_ENTIDAD,
-       MTP_ENTIDAD.CTR_NOMBRE,
-       MTP_ENTIDAD.CTR_CORREO_ELECTRONICO,
-       MTP_ENTIDAD.ENT_FK_TIPO_ENTIDAD
+       *

-       MTP_USUARIO
-       INNER JOIN MTP_ENTIDAD ON MTP_USUARIO.USU_FK_ENTIDAD = MTP_ENTIDAD.CTR_ID
-       INNER JOIN MTP_TIPO_ENTIDAD ON MTP_ENTIDAD.ENT_FK_TIPO_ENTIDAD = MTP_TIPO_ENTIDAD.ENT_ID
-       ORDER BY MTP_USUARIO.USU_ID DESC
+       VIEW_LISTA_USUARIOS
    `, {

      type: QueryTypes.SELECT
@@ -115,6 +103,31 @@ exports.updateUsuario = async (req, res) => {
    const usuario = await Usuario.findByPk(req.params.id);
    if (usuario) {
      await usuario.update(req.body);
+
+      // Primero realiza el DELETE
+      const deleteResult = await sequelize.query(
+        'DELETE FROM "MTP_USUARIO_ROL" WHERE "USU_ID" = :usuId',
+        {
+          replacements: {
+            usuId: req.params.id  // El ID del usuario que quieres eliminar
+          },
+          type: sequelize.QueryTypes.DELETE
+        }
+      );
+
+      // Luego realiza el INSERT
+      const insertResult = await sequelize.query(
+        'INSERT INTO "MTP_USUARIO_ROL" ("USU_ID", "ROL_ID") VALUES (:usuId, :rolId)',
+        {
+          replacements: {
```

```
          type: sequelize.QueryTypes.DELETE
        }
      );

      // Luego realiza el INSERT
      const insertResult = await sequelize.query(
        'INSERT INTO "MTP_USUARIO_ROL" ("USU_ID", "ROL_ID") VALUES (:usuId, :rolId)',
        {
          replacements: {
            usuId: req.params.id,
            rolId: req.body.USU_FK_ROL  // El ID del rol que deseas insertar
          },
          type: sequelize.QueryTypes.INSERT
        }
      );


    res.json({ mensaje: 'Usuario actualizado correctamente' });
  } else {
    res.status(404).json({ error: 'Usuario no encontrado' });
@@ -226,6 +239,62 @@ exports.loginValida = async (req, res) => {
};


exports.cambioContrasena = async (req, res) => {
  try {
    const { USU_ID, USU_PASSWORD, NUEVA_PASSWORD } = req.body;


  }

  if (NUEVA_PASSWORD.length < 6) {
    return res.status(400).json({ error: 'La nueva contraseña debe tener al menos 6 caracteres' });
  }

  // Verificar si el usuario y la contraseña actual son correctos
  const user = await sequelize.query(`
    SELECT
      USU_ID,
      USU_USUARIO,
      USU_NOMBRE,
      PERMISOS,
      USU_FK_ENTIDAD
    FROM
      VIEW_PERMISO_USUARIO
    WHERE
      USU_ESTADO=1 AND
      USU_ID = :USU_ID
  `, {
    replacements: { USU_ID },
    type: QueryTypes.SELECT
  });

  if (user.length === 0) {
    return res.status(401).json({ error: 'Usuario o contraseña incorrectos' });
  }
```

```
    await sequelize.query(
      UPDATE MTP_USUARIO SET
        USU_PASSWORD = :USU_PASSWORD
      WHERE
        USU_ID = :USU_ID
    `, {
      replacements: { USU_PASSWORD, USU_ID },
      type: QueryTypes.UPDATE
    });


    res.status(200).json({ message: 'Contraseña cambiada con éxito' });

  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Error al cambiar la contraseña' });
  }
};
\ No newline at end of file
```

controllers/mtp/UtilController.js

```
@@ -381,6 +381,8 @@ function obtenerDirectorio(tipo) {
      return rutaServerUplod + 'uploads/mitechopropio/en-asignacion/';
    case 4:
      return rutaServerUplod + 'uploads/mitechopropio/en-ejecutorias/';
+   case 5:
+     return rutaServerUplod + 'uploads/mitechopropio/en-marcado-pago/';
    default:
      // Ajusta esto según tus necesidades si el tipo no es 1 ni 3
      return rutaServerUplod + 'uploads/mitechopropio/';
@@ -1431,6 +1433,18 @@ exports.EnAsigancion = async (req, res) => {
        const cellRef = xlsx.utils.encode_cell({ r: rowNum, c: 0 }); // Columna A (c: 0)
        // Obtiene el contenido de la celda en la primera columna
        const cellValue = sheet[cellRef] ? sheet[cellRef].v : null;
+
+
+       const cellRefC = xlsx.utils.encode_cell({ r: rowNum, c: 9 }); // Columna J (c: 9)
+       // Obtiene el contenido de la celda en la primera columna
+       const cellValueCantidad = sheet[cellRefC] ? sheet[cellRefC].v : null;
+
+       const cellRefV = xlsx.utils.encode_cell({ r: rowNum, c: 10 }); // Columna J (c: 9)
+       // Obtiene el contenido de la celda en la primera columna
+       const cellValueValor = sheet[cellRefV] ? sheet[cellRefV].v : null;
+
+
+
        // Agrega el contenido de la celda al array de datos de la primera columna
```

```
       // Agrega el contenido de la celda al array de datos de la primera columna
       if(cellValue>0){

@@ -1453,13 +1467,15 @@ exports.EnAsigancion = async (req, res) => {
         if (!existingRecord || existingRecord.length === 0) {  console.log('LEIDER ANDRES SOTO');
         // Si no existe, insertar el nuevo registro
         await sequelize.query(
-          `INSERT INTO MTP_RESOLUCION_HOGAR (RHO_FK_RESOLUCION, RHO_FK_HOGAR, RHO_USUARIO, RHO_ESTADO, RHO_FECHA)
-          VALUES (:resolucionId, :hogarId, :usuario, :estado, SYSDATE)`,
+          `INSERT INTO MTP_RESOLUCION_HOGAR (RHO_FK_RESOLUCION, RHO_FK_HOGAR, RHO_USUARIO, RHO_ESTADO,
RHO_FECHA,RHO_NUM_SMLMV,RHO_VALOR_SUBSIDIO)
          VALUES (:resolucionId, :hogarId, :usuario, :estado, SYSDATE, :cantidad, :valor)`,
          {
            replacements: {
              resolucionId: secuenciaValue,
              hogarId: cellValue,
              usuario: usuario_id,
+             cantidad: cellValueCantidad,
+             valor: cellValueValor,
              estado: 1
            }
          }
@@ -1983,4 +1999,166 @@ exports.getEscrituraHogar = async (req, res) => {
    console.error(error);
```

```
                estado: 1
              }
            }
@@ -1983,4 +1999,166 @@ exports.getEscrituraHogar = async (req, res) => {
    console.error(error);
    res.status(500).json({ error: 'Error al obtener los escritura' });
  }
- };
\ No newline at end of file
+ };
+
+
+ exports.getRolUsuario = async (req, res) => {
+   try {
+     const escritura = await sequelize.query(`
+     SELECT * FROM MTP_ROL WHERE ROL_ESTADO=1
+     `, {
+       replacements: {
+         id_hogar: req.params.hogar
+        },
+       type: QueryTypes.SELECT
+     });
+
+     res.json(escritura);
+
+   } catch (error) {
```

```
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Error al obtener los Rol Usuario' });
  }
};


exports.getAllTipoCuenta = async (req, res) => {
  try {
    const usuarios = await sequelize.query(`
    SELECT *  FROM MTP_TIPO_CUENTA WHERE TPA_ESTADO=1 ORDER BY TPA_NOMBRE_CTA ASC
 `, {
      type: QueryTypes.SELECT
    });

    res.json(usuarios);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Error al obtener las entidades financieras' });
  }
};



exports.EnMarcadoPago = async (req, res) => {
  try {
    // Obtiene los datos del formulario
```

```
      type: QueryTypes.SELECT
    });

    res.json(usuarios);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Error al obtener las entidades financieras' });
  }
};



exports.EnMarcadoPago = async (req, res) => {
  try {
    // Obtiene los datos del formulario
    const { resolutionNumber, resolutionDate, numeroSubsidio, numeroFolio,usuario_id } = req.body;

    // Accede a los archivos subidos desde req.files
    const excelFile = req.files.excelFile[0];
    const pdfFile = req.files.pdfFile[0];
    // Formatea la fecha a formato de fecha local
    const localDate = new Date(resolutionDate);

    // Obtiene el año, mes y día de la fecha
    const year = localDate.getFullYear();
    const month = localDate.getMonth() + 1; // Los meses en JavaScript van de 0 a 11
    const day = localDate.getDate();
```

```
    // Obtiene el valor de la secuencia
    const secuenciaValue = secuencia[0][0].NEXTVAL;

    // Crea el nombre de archivo utilizando el prefijo secuenciaValue
    const excelFileName = `${secuenciaValue}.xlsx`; // Cambia el nombre y la extensión según sea necesario
    const pdfFileName = `${secuenciaValue}.pdf`; // Cambia el nombre y la extensión según sea necesario

    // Mueve los archivos renombrados al destino final
    await moveFile(excelFile.path, rutaServerUplod+'uploads/mitechopropio/en-marcado-pago/' + excelFileName);
    await moveFile(pdfFile.path, rutaServerUplod+'uploads/mitechopropio/en-marcado-pago/' + pdfFileName);


   // Inserta los valores en la tabla MTP_RESOLUCION
   await sequelize.query(
    `INSERT INTO MTP_RESOLUCION_PAGO
    (RSP_ID, RSP_RESOLUCION, RSP_NUM_SUBSIDIOS,RSP_NUM_FOLIOS, RSP_DIA, RSP_MES, RSP_ANO, RSP_REGISTRO)
     VALUES (:resId, :resResolucion, :resNumSubsidios, :resNumFolios, :resDia, :resMes, :resAno, SYSDATE)`,
    {
      replacements: {
        resId: secuenciaValue,
        resResolucion: resolutionNumber,
        resNumSubsidios: numeroSubsidio,
        resNumFolios: numeroFolio,
        resDia: day,
        resMes: month,
        resAno: year
```

```
// Ruta al archivo Excel
const filePath = path.join(rutaServerUplod+'uploads/mitechopropio/en-marcado-pago/', excelFileName);

// Carga el libro de Excel
const workbook = xlsx.readFile(filePath);

// Obtiene la primera hoja de cálculo
const sheetName = workbook.SheetNames[0];
const sheet = workbook.Sheets[sheetName];

// Importar PQueue dinámicamente
const { default: PQueue } = await import('p-queue');

// Configura la cola con un límite de concurrencia
const queue = new PQueue({ concurrency: 3 });


// Define el rango de celdas de la primera columna
const range = xlsx.utils.decode_range(sheet['!ref']);
const columnData = [];

// Itera sobre las celdas de la primera columna
for (let rowNum = range.s.r; rowNum <= range.e.r; rowNum++) {
  // Crea el identificador de la celda de la primera columna (por ejemplo, A1, A2, etc.)
  const cellRef = xlsx.utils.encode_cell({ r: rowNum, c: 0 }); // Columna A (c: 0)
  // Obtiene el contenido de la celda en la primera columna
  const cellValue = sheet[cellRef] ? sheet[cellRef].v : null;
```

```
        console.error('Error al obtener el nombre de la entidad financiera:', error);
        return res.status(500).json({ error: 'Error al obtener el nombre de la entidad financiera' });
@@ -127,7 +140,10 @@ async function updateEntidad(req, res) {
          CTR_TELEFONO,
          CTR_NIT,
          CTR_FK_ENTIDAD_FIN,
-         ENT_FK_TIPO_ENTIDAD
+         ENT_FK_TIPO_ENTIDAD,
+         CTR_TIPO_CUENTA,
+         CRT_NUMERO_CUENTA,
+         CRT_NUM_PORTAFOLIO
        },
        {
          where: { CTR_ID: id },
@@ -212,10 +228,100 @@ async function obtenerNombreEntidadPorId(enfId) {
}


+// Configurar el almacenamiento de Multer
+const storage = multer.diskStorage({
+  destination: (req, file, cb) => {
+    // Carpeta donde se guardarán los archivos
+    cb(null, path.join(rutaServerUplod, 'uploads/mitechopropio/certificacion-entidad/')); // Asegúrate de que la carpeta 'uploads' exista
+  },
+  filename: (req, file, cb) => {
+    // Renombrar el archivo usando el ID proporcionado y conservar la extensión original
+    const { id } = req.params;
```

```
  filename: (req, file, cb) => {
    // Renombrar el archivo usando el ID proporcionado y conservar la extensión original
    const { id } = req.params;
    const ext = path.extname(file.originalname); // Obtiene la extensión del archivo original
    cb(null, `${id}${ext}`); // Renombra el archivo con el ID + extensión
  }
});

// Configurar el middleware de Multer
const upload = multer({
  storage: storage,
  limits: { fileSize: 5 * 1024 * 1024 }, // Límite opcional de 5MB por archivo
}).single('file'); // Solo sube un archivo llamado 'file'

// Controlador para subir el archivo
const uploadFile = async (req, res) => {
  const { id } = req.params;

  upload(req, res, async (err) => {
    if (err) {
      // Manejo de errores de Multer
      return res.status(500).send({ message: 'Error al subir el archivo.', error: err.message });
    }

    if (!req.file) {
      // Si no se subió ningún archivo
      return res.status(400).send({ message: 'No se ha subido ningún archivo.' });
    .
      // Si no se subió ningún archivo
      return res.status(400).send({ message: 'No se ha subido ningún archivo.' });
    }

    try {
      const sqlQuery = `UPDATE MTP_ENTIDAD SET CRT_NOMBRE_CERTIFICADO = :nombreArchivo WHERE CTR_ID = :id`;

      // Ejecutar la consulta
      await sequelize.query(sqlQuery, {
        replacements: {
          nombreArchivo: req.file.filename,
          id: id
        }
      });

      // Si el archivo se subió correctamente, enviamos la respuesta con la información del archivo
      res.send({
        message: 'Archivo subido exitosamente.',
        filename: req.file.filename, // Nombre del archivo renombrado
        path: `/uploads/mitechopropio/certificacion-entidad/${req.file.filename}` // Ruta donde se encuentra el archivo
      });
    } catch (error) {
      // Manejo de errores al ejecutar la consulta
      res.status(500).send({
        status: 500,
        message: "Error al realizar el update",
        detailed_message: error.message
      }).
```

```
// Controlador para la descarga de archivos
const downloadFile = (req, res) => {
  const filename = req.params.id;
  const filePath = path.join(rutaServerUplod, 'uploads/mitechopropio/certificacion-entidad/', filename);

  console.log('Ruta del archivo:', filePath); // Verifica la ruta

  fs.access(filePath, fs.constants.F_OK, (err) => {
    if (err) {
      return res.status(404).send({ message: 'Archivo no encontrado.' });
    }

    res.download(filePath, (err) => {
      if (err) {
        return res.status(500).send({ message: 'Error al descargar el archivo.' });
      }
    });
  });
};
```

```
+      }
+    });
+  });
+};
+
+
+
+
 module.exports = {
   getAllEntidades,
   getEntidadById,
   addEntidad,
   updateEntidad,
-  toggleEntidadEstado
+  toggleEntidadEstado,
+  uploadFile,
+  downloadFile
```

controllers/mtp/HogarController.js

```
@ -450,7 +450,7 @@ async function  EnviarNotificacion(body,radicado)  {
       irrevocablemente a la Constructora y al Distrito de Barranquilla, para que utilice los datos que he(mos) suministrado
       o que llegue(mos) a suministrar con los siguientes fines: (i) Compartirlos a la Alcaldía de Barranquilla para las
       validaciones que considere necesarias, para realizar el otorgamiento de los beneficios, conforme lo establecido en el
-      Decreto Distrital No 0172 de 2024, y demás normas que lo modifiquen, adicionen o sustituyan, (ii) Consultar mis
+      Decreto Distrital No 0811 de 2024, y demás normas que lo modifiquen, adicionen o sustituyan, (ii) Consultar mis
       (nuestros) datos ante las centrales de información, tanto por el Constructor o como por Alcaldía, (iii) Consultar
       acerca del cumplimiento de las relaciones y obligaciones que he (mos) tenido con el sector financiero y sobre las
       relaciones y obligaciones que hacia futuro adquiera con dicho sector; (iv) Enviar información de novedades o
@@ -2017,11 +2017,17 @@ exports.HistoricoPostulacionesCuotas = async (req, res) => {
xports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {

ry {
-  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos} = req.query;
+  const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos,masivo} = req.query;
   const pageSize = 10; // Número de registros por página
   const offset = (page - 1) * pageSize; // Cálculo del offset

   let whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91`;
+  if(masivo==2){
+    whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91 AND HAE_ESTADOCUOTA_ID=2`;
+  }
+
+
+
   let wherePag = '';
   if(page!=0){
```

```
          acerca del cumplimiento de las relaciones y obligaciones que he (mos) tenido con el sector financiero y sobre las
          relaciones y obligaciones que hacia futuro adquiera con dicho sector; (iv) Enviar información de novedades o
@@ -2017,11 +2017,17 @@ exports.HistoricoPostulacionesCuotas = async (req, res) => {
 exports.HistoricoPostulacionesEntidadCuotas = async (req, res) => {

 try {
-    const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos} = req.query;
+    const { page, search,entidad ,filtroEstadoPostulacion, filtroEstadoProcesos,masivo} = req.query;
     const pageSize = 10; // Número de registros por página
     const offset = (page - 1) * pageSize; // Cálculo del offset

     let whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91`;
+    if(masivo==2){
+      whereClause = ` WHERE BANCO=${entidad} AND HAE_ESTADO_ACEPTACION<>91 AND HAE_ESTADOCUOTA_ID=2`;
+    }
+
+
+
     let wherePag = '';
     if(page!=0){
       wherePag=` WHERE rn > ${offset} AND ROWNUM <= ${pageSize}`
@@ -2134,8 +2140,8 @@ try {
          EPO_ID: row.EPO_ID,
          GRP_OBSERVACION:row.GRP_OBSERVACION,
          RESOLUCION_CONCATENADA:row.RESOLUCION_CONCATENADA,
+         RESOLUCION_ASIG:row.RESOLUCION_ASIG,
          USUARIO_EJECUTORIA :row.USUARIO_EJECUTORIA,
-         RESOLUCION_ASIG:row.RESOLUCION_ASIG,
          RESOLUCION_CONT_PAGO :row.RESOLUCION_CONT_PAGO,
          MTP_RESOLUCION_PAGO :row.MTP_RESOLUCION_PAGO,
```

leider soto

LEIDER ANDRES SOTO BETIN
CC: 78.300.913 de Montelíbano