Contrato de prestación de servicios objeto de seguimiento y control técnico:							ASM No. 022/2023						
Contratista:	David Steven Torres Figueroa												
Objeto del contrato de prestación de servicios:													
Prestar por sus propios medios con plena autonomía técnica los servicios profesionales para realizar el soporte y mantenimientos sobre las herramientas implementadas en la lógica de negocio (Back-End) del Sistema de solicitudes de trámites catastrales de la Asociación de Municipios del Catatumbo, provincia de Ocaña y sur del Cesar- Asomunicipios.													
Fecha suscripción contrato de prestación de servicios:	dd	07	mm	02	aaaa	2023							
Datos del profesional contratista que realizará el Informe de seguimiento y control técnico													
Contratista:	Yolimar Bermúdez Labarca												
Contrato de prestación de servicios:							ASM No. 007/2023						
Objeto del contrato de prestación de servicios:													
Prestar por sus propios medios, con plena autonomía técnica y administrativa los servicios profesionales como coordinadora del proceso de conservación catastral y Sistema de Información Geográfica – SIG Catatumbo en el marco del Convenio Interadministrativo No.00128 de 2021 con la Gobernación de Norte de Santander y Asomunicipios.													
Desarrollo del seguimiento y control técnico del contrato de prestación de servicios													
Informe de actividades N°: 1	Fecha del Informe:	dd	07	mm	03	aaaa	2023						
Periodo del informe N° 1 de actividades objeto de seguimiento y control:													
del	dd	07	mm	02	aaaa	2023	al	dd	07	mm	03	aaaa	2023

1. Revisión, seguimiento y validación técnica del (los) producto(s) a entregar de acuerdo al objeto contractual y correspondiente(s) al Informe N° 1

El contratista de la Coordinación Técnica de la (EPSEA – Catastro) de Asomunicipios, realizó la respectiva revisión, seguimiento y validación técnica de la(s) actividad(es) y/o producto(s) entregado(s) en el desarrollo del objeto del contrato ASM 022/2023 correspondiente al periodo citado en el presente documento, contenidos en el informe N° 01, de acuerdo a la siguiente relación:

Producto 1: Apoyo en la estructuración del código (Back End).

Validado técnicamente satisfactorio, según compromiso contractual correspondiente al periodo relacionado en el presente informe y de acuerdo con el informe N° 01

Producto 2: Creación y configuración del código (Back End).

Validado técnicamente satisfactorio, según compromiso contractual correspondiente al periodo relacionado en el presente informe y de acuerdo con el informe N° 01

Producto 3: Generar y documentar código fuente módulo de registro de trámites catastrales (Back End).

Validado técnicamente satisfactorio, según compromiso contractual correspondiente al periodo relacionado en el presente informe y de acuerdo con el informe N° 01

Producto 4: Realizar el despliegue de la aplicación en ambiente de producción.

Validado técnicamente satisfactorio, según compromiso contractual correspondiente al periodo relacionado en el presente informe y de acuerdo con el informe N° 01, donde se realizó una reunión virtual donde se observó la implementación de los cambios en la plataforma

Contrato de prestación de servicios objeto de seguimiento y control técnico:	ASM No. 022/2023
Contratista:	David Steven Torres Figueroa
Producto 5: Generar manuales de instalación de aplicación.	
Validado técnicamente satisfactorio, según compromiso contractual correspondiente al periodo relacionado en el presente informe y de acuerdo con el informe N° 01	

2. Cierre del Informe de Seguimiento y Control Técnico

Yo, Yolimar Bermúdez, en mi calidad de contratista Asomunicipios para realizar la Coordinación de Conservación Catastral – Catastro, informo a la Supervisión que el (la) señor(a) **David Steven Torres Figueroa** quien tiene contrato vigente con Asomunicipios # ASM 022/2023; que de acuerdo a mi responsabilidad contractual, he realizado el seguimiento y validación técnica a conformidad de los productos del informe de actividades N° 1 del contrato en mención, como consta en los numeral 1 del presente documento.

Por lo anteriormente expuesto y con base en el desarrollo del presente documento contemplados en el numeral 1, se firma para dejar constancia y evidencia del seguimiento y control al contrato de prestación de servicios.

Yolimar Bermúdez
Contratista
Coordinador EPSEA – Catastro Asomunicipios

Señores:
Asociación de Municipios del Catatumbo, Provincia de Ocaña y Sur del Cesar - Asomunicipios
Atn; Hugo Armando Arévalo Franco – Director Ejecutivo y/o Dalgie Esperanza Torrado Rizo –Profesional Administrativo y Financiero
Ocaña, Norte de Santander

Ref. Entrega de Informe de actividades en el marco del Contrato de Prestación de Servicios profesionales ASM No 022 del 2023 correspondiente al periodo 07 de febrero al 07 de marzo de 2023.

En cumplimiento del compromiso adquirido en valor y forma del contrato en mención, el cual tiene por objeto: Prestar por sus propios medios con plena autonomía técnica los servicios profesionales para realizar el soporte y mantenimientos sobre las herramientas implementadas en la lógica de negocio (Back-End) del Sistema de solicitudes de trámites catastrales de la Asociación de Municipios del Catatumbo, provincia de Ocaña y sur del Cesar- Asomunicipios. me permito hacer entrega del primer Informe de actividades, documento equivalente No.0548, pago de seguridad social, lo anterior para su revisión y aprobación.

ASOMUNICIPIOS
Presentado por: David Steven
HORA
13 MAR. 2023 5:11
Angelica T. Herrera
FIRMA FUNCIONARIO
00000173

Atentamente,

David Steven Torres Figueroa
C.C 1.013.611.554

Documento soporte de costos y gastos en operaciones con no obligados a expedir factura o documento equivalente

Artículo 1.6.1.4.12 Decreto Único reglamentario en materia tributaria 1625 de 2016, y el numeral 5 el artículo 55 de la Res. 00042 del 5 de mayo de 2020
Sustituido por el Decreto 358 de 2020

ASOCIACIÓN DE MUNICIPIOS DEL CATATUMBO, PROVINCIA DE OCAÑA Y SUR DEL CESAR ASOMUNICIPIOS

NIT. 800.244.263-1

Calle 12 No. 11-76 Piso 2 Apto. 201 Ocaña, N. de S.

DS 0548

Autorización de numeración DIAN No 18764030132715
Fecha: 15-06-2022 Autorizacion: DS 301 al DS 600
Vigencia: 12 meses

Fecha de operación: Marzo de 2023

Vendedor o quien presta el servicio:	David Steven Torres Figueroa	NIT.	1013611554
Dirección:	Diagonal 35B 2B 16 San Cristobal	Tel.	3620417

CANTIDAD	DETALLE	VR UNITARIO	VR TOTAL
1	Pago portal CPSP No. 022/2023		6'000.000=
		SUBTOTAL	6'000.000=
		RETEFUENTE	—
		TOTAL	6'000.000=

ACEPTO QUE NO SOY RESPONSABLE DE IVA

ELABORADO POR

Lito Torcoroma - Alfredo Numa. NIT. 88.138.209-6
Cel. 315 6205179

Informe de Actividades Contrato de Prestación de Servicios ASM No. 022/2023													
Informe N°: 01			De fecha:	dd	13	mm	03	aaaa	2023				
Periodo del informe:													
del	dd	07	mm	02	aaaa	2023	al	dd	07	mm	03	aaaa	2023
Contratista:	David Steven Torres Figueroa												
Objeto del contrato de prestación de servicios:													
Prestar por sus propios medios con plena autonomía técnica los servicios profesionales para realizar el soporte y mantenimientos sobre las herramientas implementadas en la lógica de negocio (Back-End) del Sistema de solicitudes de trámites catastrales de la Asociación de Municipios del Catatumbo, provincia de Ocaña y sur del Cesar- Asomunicipios.													
Fecha suscripción contrato				dd	07	mm	02	aaaa	2023				
Informe Presentado a:	Dalgie Esperanza Torrado Rizo: Subdirección Administrativa y Financiera - ASOMUNICIPIOS												

Producto(s) Entregable(s)

Producto 1: Apoyo en la estructuración del código (Back End).

De acuerdo con los requerimientos de Asomunicipios, se procedió a generar las nuevas funcionalidades/herramientas que permiten dar solución a los requerimientos planteados. Se adjuntan capturas de pantalla del código fuente.

Búsqueda por ID de solicitud

```
if (usuarioEncontrado.get().getTipoUsuario().equals("I")) {

    List<SolicitudesInterfaceDTO> solicitudesUsuarioInterno = new ArrayList<SolicitudesInterfaceDTO>();

    if (idSolicitud != null) {
        solicitudesUsuarioInterno = solicitudesRepository.buscarSolicitudId(idSolicitud);
    } else {
        solicitudesUsuarioInterno = solicitudesRepository.buscarSolicitudRadicado(nRadicado);
    }

    for (SolicitudesInterfaceDTO solicitud : solicitudesUsuarioInterno) {

        SolicitudesConsultaDTO solicitudConsultaDTO = modelMapper.map(solicitud, SolicitudesConsultaDTO.class);

        ValoresLista dominioNombreTramite = domainRepository.findByValor(solicitudConsultaDTO.getNombreTramite());
        ValoresLista dominioTipoTramite = domainRepository.findByValor(solicitudConsultaDTO.getTipoTramite());
        EstadosSolicitudInterfaceDTO estadoInterfaceDTO = estadosRepository.buscarUltimoEstadoIdSolicitud(
                solicitudConsultaDTO.getIdSolicitud());
        ValoresLista dominioNombreEstado = domainRepository.findByValor(estadoInterfaceDTO.getestado());

        solicitudConsultaDTO.setNombreTramite(dominioNombreTramite.getDescripcionValor());
        solicitudConsultaDTO.setTipoTramite(dominioTipoTramite.getDescripcionValor());
        solicitudConsultaDTO.setEstado(dominioNombreEstado.getDescripcionValor());

        solicitudesDTOList.add(solicitudConsultaDTO);
    }
```

Informe de Actividades Contrato de Prestación de Servicios ASM No. 022/2023

Paginación de elementos

```
Optional<Usuarios> usuarioEncontrado = usersRepository.findById(idUsuario);
List<SolicitudesConsultaDTO> solicitudesDTOList = new ArrayList<SolicitudesConsultaDTO>();

if (usuarioEncontrado.get().getTipoUsuario().equals("I")) {

    Pageable paginaElementos = PageRequest.of(pageable.getPageNumber(), pageable.getPageSize(), pageable.getSort());
    Page<SolicitudesInterfaceDTO> numeroPaginasSolicitudesUsuarioInternos = solicitudesRepository.buscarSolicitudesInternoNumeroPag(paginaElementos);
    nPaginas = numeroPaginasSolicitudesUsuarioInternos.getTotalPages()-1;
    totalRegistros = numeroPaginasSolicitudesUsuarioInternos.getTotalElements();

    List<SolicitudesInterfaceDTO> solicitudesUsuarioInterno = solicitudesRepository.buscarSolicitudesInternoPaginadoLista(paginaElementos);

    for (SolicitudesInterfaceDTO solicitud : solicitudesUsuarioInterno) {

        SolicitudesConsultaDTO solicitudConsultaDTO = modelMapper.map(solicitud, SolicitudesConsultaDTO.class);

        ValoresLista dominioNombreTramite = domainRepository.findByValor(solicitudConsultaDTO.getNombreTramite());
        ValoresLista dominioTipoTramite = domainRepository.findByValor(solicitudConsultaDTO.getTipoTramite());
        EstadosSolicitudInterfaceDTO estadoInterfaceDTO = estadosRepository.buscarUltimoEstadoIdSolicitud(
                solicitudConsultaDTO.getIdSolicitud());
        ValoresLista dominioNombreEstado = domainRepository.findByValor(estadoInterfaceDTO.getestado());

        solicitudConsultaDTO.setNombreTramite(dominioNombreTramite.getDescripcionValor());
        solicitudConsultaDTO.setTipoTramite(dominioTipoTramite.getDescripcionValor());
        solicitudConsultaDTO.setEstado(dominioNombreEstado.getDescripcionValor());

        solicitudesDTOList.add(solicitudConsultaDTO);
    }
```

Configuración servicios web desarrollados

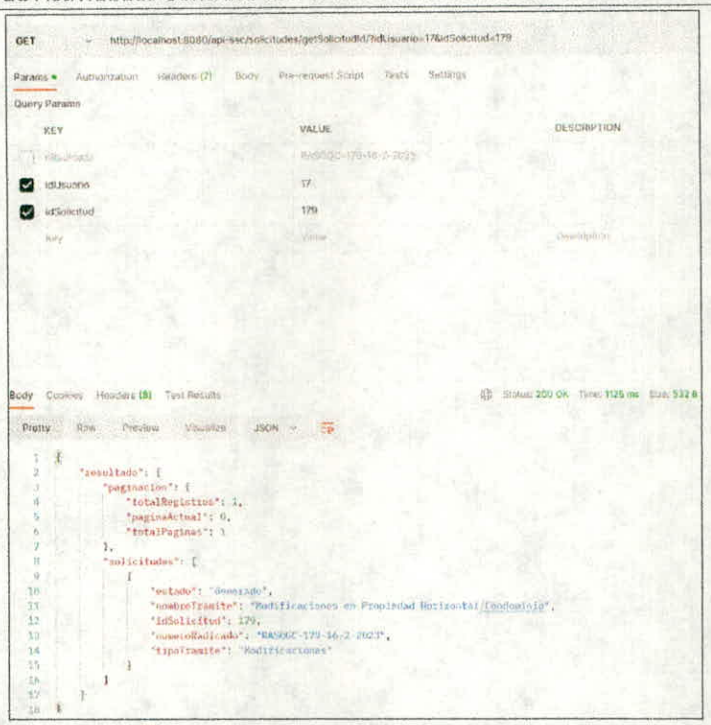
```
@GetMapping(path="/getSolicitudId")
public ResponseEntity<Object> getSolicitudId(@RequestParam("idUsuario") Long idUsuario, @RequestParam(value ="idSolicitud", required = false) Long idSolicitud,
        @RequestParam(value ="nRadicado", required = false) String nRadicado, @RequestHeader(name = "token") String token){
    return solicitudesService.getSolicitudId(idUsuario, idSolicitud, nRadicado, token);
}
```

Producto 2: Creación y configuración del código (Back End).

Se actualizaron los Endpoint que debían intervenirse en el proyecto, esto con el fin de satisfacer los nuevos requerimientos del sistema, a continuación, capturas de pantalla de las respuestas de los servicios web actualizados

Servicio Web para consulta de solicitudes por ID:

Informe de Actividades Contrato de Prestación de Servicios ASM No. 022/2023



Servicio web para obtener los registros paginados por 10 registros en cada pagina

Informe de Actividades Contrato de Prestación de Servicios ASM No. 022/2023

CATATUMBO / SOLICITUD / **OBTENER SOLICITUDES PARA UN USUARIO PAGINADO**

Save

GET http://localhost:8080/api-ssc/solicitudes/getSolicitudesUsuario/17?pag=5&tam=10

Params • Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
pag	5	
tam	10	
Key	Value	Description

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 3.50 s Size: 1.13 KB

Pretty Raw Preview Visualize JSON


```
{
    "resultado": {
        "paginacion": {
            "totalRegistros": 55,
            "paginaActual": 5,
            "totalPaginas": 5
        },
        "solicitudes": [
            {
                "estado": "Generado",
                "nombreTramite": "Modificaciones en Propiedad Horizontal/Condominio",
                "idSolicitud": 179,
                "numeroRadicado": "RASOGC-179-16-2-2023",
                "tipoTramite": "Modificaciones"
            },
            {
                "estado": "Radicado",
                "nombreTramite": "Inscripción de predio",
```


Servicio web que consulta los estados de una solicitud, con la actualización de que están ordenados por fecha de creación

Informe de Actividades Contrato de Prestación de Servicios ASM No. 022/2023

CATATUMBO / SOLICITUD / **OBTENER DETALLE SOLICITUD POR ID**

Save

GET http://localhost:8080/api-ssc/solicitudes/getDetalleSolicitud/179

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 6.30 s Size: 6.51 KB Save R

Pretty Raw Preview Visualize JSON


```
            "observaciones": "hfhfghfgh",
            "idSolicitud": 179,
            "fechaEstado": 1676748988444,
            "idEstadoSolicitud": 165
        },
        {
            "estado": "Asignado",
            "observaciones": "testiando",
            "idSolicitud": 179,
            "fechaEstado": 1676668888034,
            "idEstadoSolicitud": 164
        },
        {
            "estado": "Radicado",
            "observaciones": "Creación de la solicitud",
            "idSolicitud": 179,
            "fechaEstado": 1676594453719,
```

Producto 3: Generar y documentar código fuente módulo de registro de trámites catastrales (Back End).

Se agregan los diferentes comentarios al código fuente, estos solo son visibles directamente sobre los componentes del proyecto, por este motivo, se adjuntan capturas de los comentarios dentro del código

```
/**
 * Metodo que permite realizar la actualizacion de la informacion de una solicitud
 * @param informacion a actualizar de la solicitud
 * @param token de seguridad
 * @return json con el resultado
 */
@Override
public ResponseEntity<Object> updateEstadoSolicitud(EstadosSolicitudDTO estadoSolicitudDTO, String token) {
    ResponseEntity<Object> exception = null;

    try {
        ResponseEntity<Object> validarToken = utils.validarToken(token);
        String tokenValidado = validarToken.getBody().toString();

        if (tokenValidado.indexOf("OK") != -1) {

            estadoSolicitudDTO.setFechaEstado(new Date());
            EstadosSolicitud estadoActualizar = modelMapper.map(estadoSolicitudDTO, EstadosSolicitud.class);
            estadosRepository.save(estadoActualizar);

            Optional<Solicitudes> solicitud = solicitudesRepository.findById(estadoSolicitudDTO.getIdSolicitud());
            Optional <Usuarios> user = usersRepository.findById(solicitud.get().getIdSolicitante().getIdUsuario());
            UsuariosDTO userDto = modelMapper.map(user.get(), UsuariosDTO.class);

            ValoresLista dominioActualizarEstado = domainRepository.findByValor(estadoSolicitudDTO.getEstado());

            JSONObject item = new JSONObject();
            item.put("mensaje", "Se actualizo correctamente la solicitud");


/**
 * Metodo que permite realizar la creacion de una solicitud
 * @param informacion de la solicitud
 * @param archivo comprimido que contiene la informacion que sustenta la solicitud
 * @param token de seguridad
 * @return json con el resultado
 */
@Override
public ResponseEntity<Object> createSolicitud(String solicitudJson, MultipartFile file, String token) throws SQLException {
    ResponseEntity<Object> exception = null;

    try {
        ResponseEntity<Object> validarToken = utils.validarToken(token);
        String tokenValidado = validarToken.getBody().toString();

        if (tokenValidado.indexOf("OK") != -1) {

            String mensaje = "Buen dia estimado solicitante,"
                    + "\n\n De acuerdo con su petición se creó su solicitud."
                    + "\n\n Este es un mensaje autogenerado, no responder a esta dirección de correo."
                    + "\n\n Atentamente,"
                    + "\n Asomunicipios";

            SolicitudesDTO solicitudDTO = objectMapper.readValue(solicitudJson, SolicitudesDTO.class);
            Solicitudes solicitud = modelMapper.map(solicitudDTO, Solicitudes.class);
```

```java
/**
 * Metodo que consulta los diferentes tipos con los que se pueden crear solicitudes
 * @param criterio de busqueda
 * @param token de seguridad
 * @return json con el resultado
 */
@Override
public ResponseEntity<Object> getTiposSolicitud(String valorConsultar, String token) {
    ResponseEntity<Object> exception = null;

    try {

        ResponseEntity<Object> validarToken = utils.validarToken(token);
        String tokenValidado = validarToken.getBody().toString();

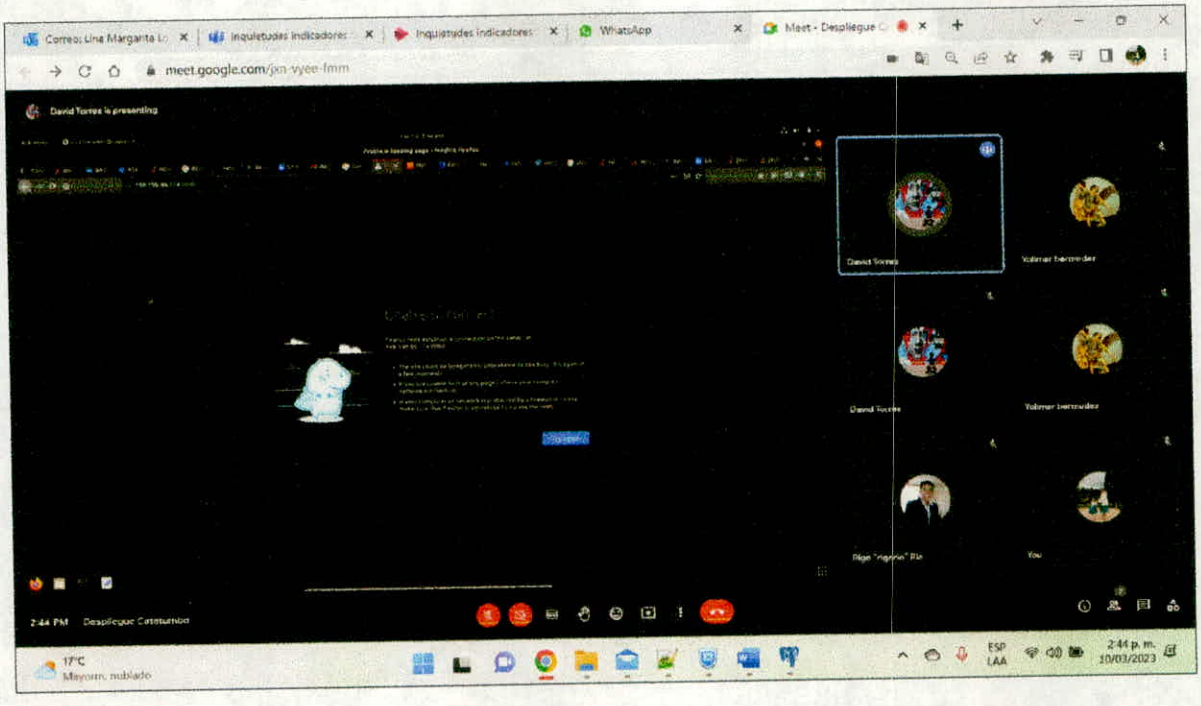
        if (tokenValidado.indexOf("OK") != -1) {

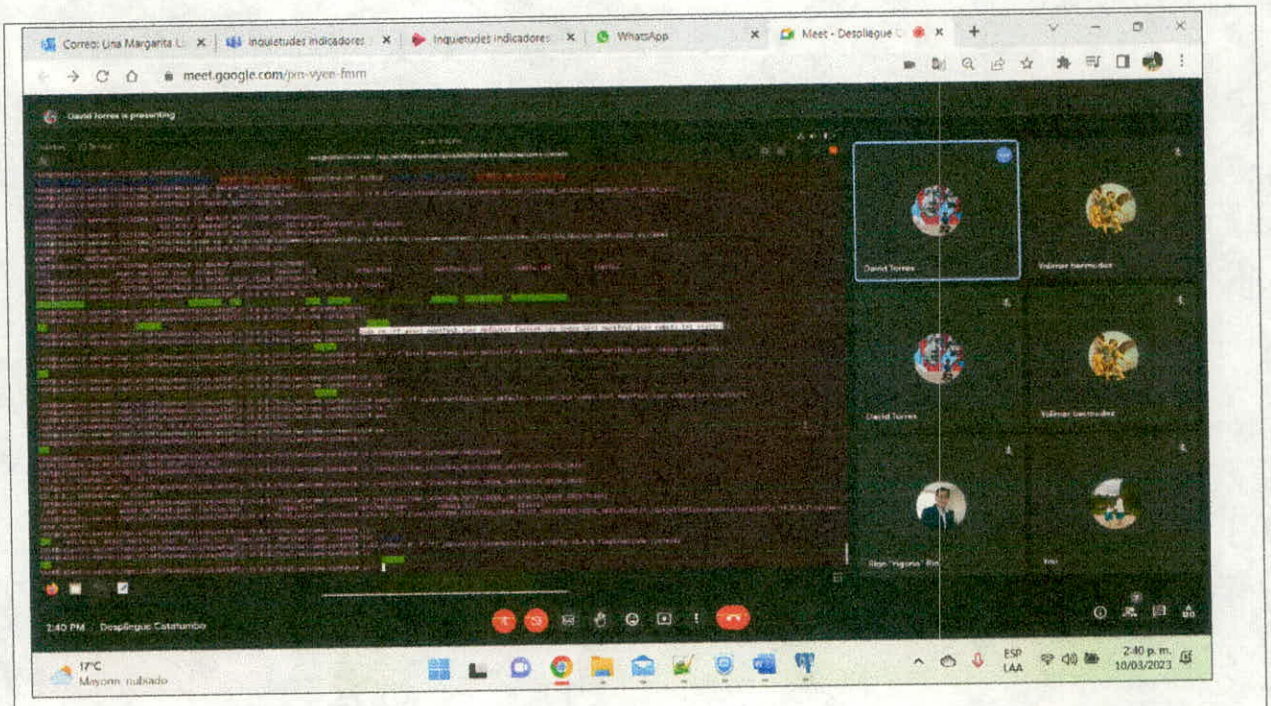
            List<ValoresLista> domains = domainRepository.findTipoSolicitud(valorConsultar);
            List<ValoresListaDTO> domainDTOList = new ArrayList<ValoresListaDTO>();

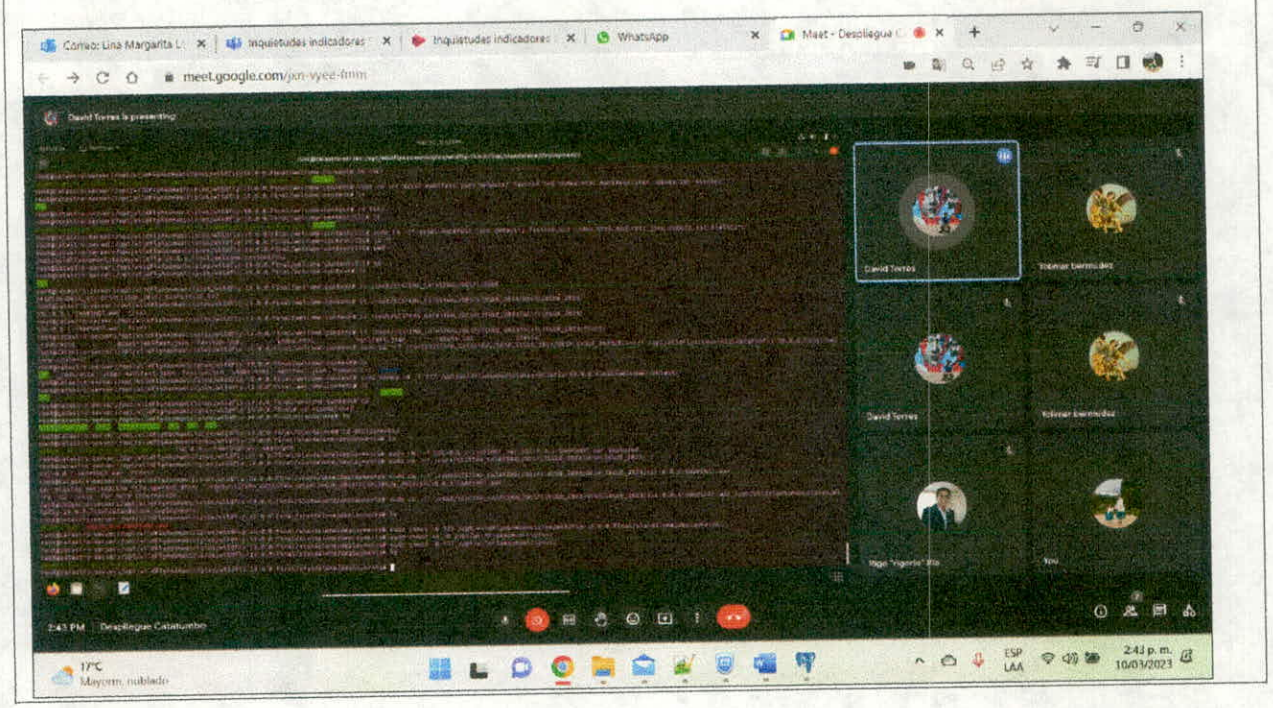
            for (ValoresLista domain : domains) {
                ValoresListaDTO domainDto = modelMapper.map(domain, ValoresListaDTO.class);
                domainDTOList.add(domainDto);
            }
```

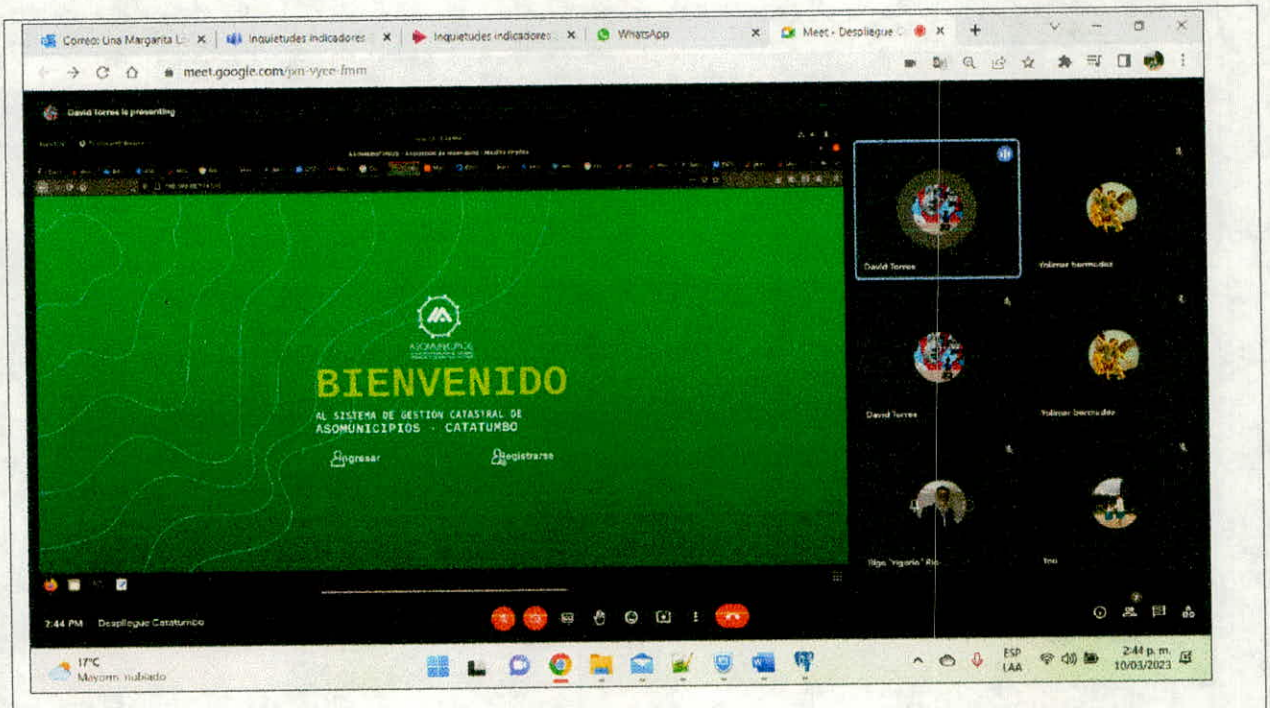
Producto 4: Realizar el despliegue de la aplicación en ambiente de producción.

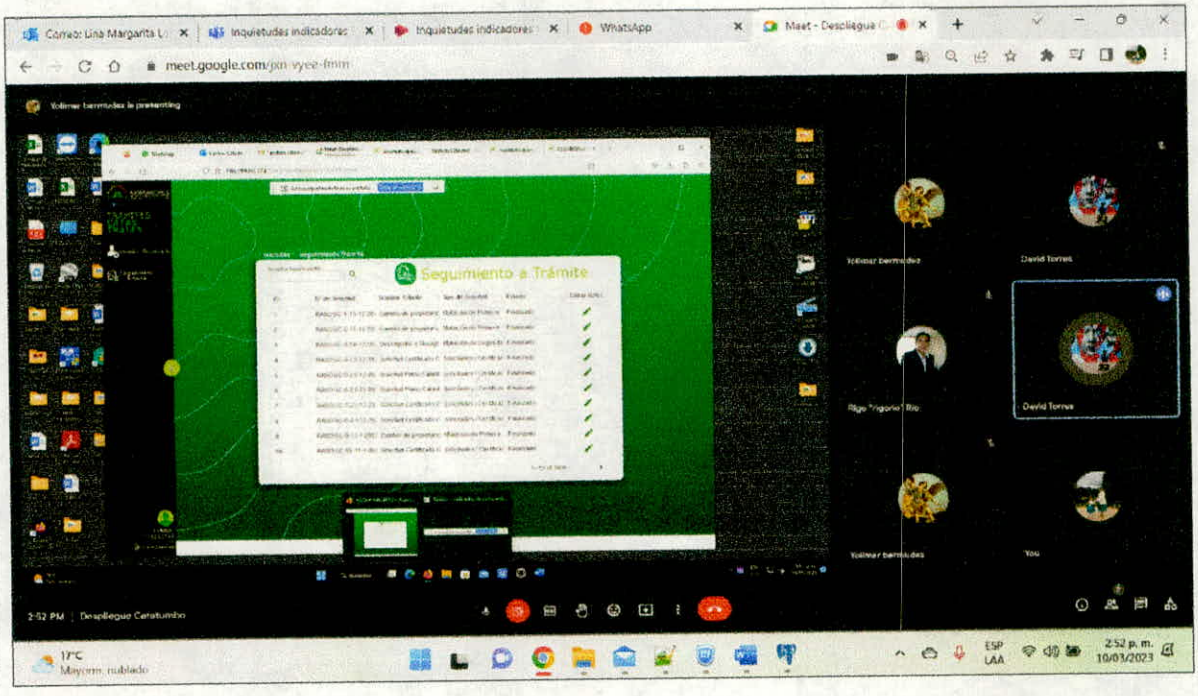
Se realiza implementación de la nueva versión de la aplicación sobre el servidor dispuesto por Asomunicipios con IP 198.199.80.174.

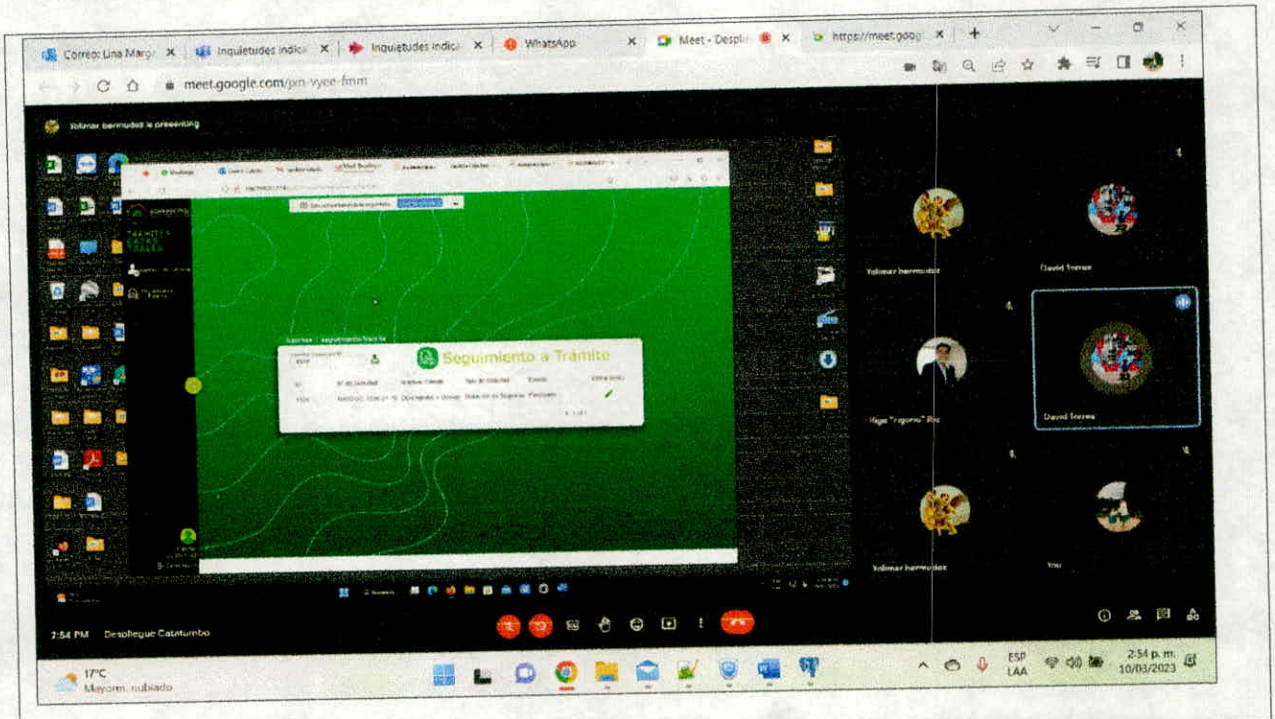



meet.google.com/jxn-vyee-fmm
David Torres is presenting
2:40 PM Despliegue Catatumbo
17°C
Mayorm. nublado
ESP
LAA
2:40 p. m.
10/03/2023


meet.google.com/jxn-vyee-fmm
David Torres is presenting
2:43 PM Despliegue Catatumbo
17°C
Mayorm. nublado
ESP
LAA
2:43 p. m.
10/03/2023


meet.google.com/pxn-vyee-fmm
David Torres is presenting
BIENVENIDO
AL SISTEMA DE GESTION CATASTRAL DE
ASOMUNICIPIOS - CATATUMBO
Ingresar
Registrarse
Despliegue Catatumbo
2:44 PM


meet.google.com/pxn-vyee-fmm
Seguimiento a Trámite
Despliegue Catatumbo
2:52 PM



Producto 5: Generar manuales de instalación de aplicación.

Se realiza el manual de instalación para la aplicación en entornos Linux, como evidencia se adjunta el documento que contiene la información

MANUAL DE IMPLEMENTACION PI

Firma del Prestador del Servicio	David Steven Torres Figueroa						
	Documento: 1.013.611.554						
Firma recibo del Supervisor del Contrato ASM No. 022/2023							
	Fecha:	dd	13	mm	03	aaaa	2023

Información general de la reunión							
Fecha:	Día	Mes	Año	Hora de inicio	2:00 pm	Hora de finalización	3:00 pm
	15	02	2023				
Lugar de la reunión:				Sala de juntas Asociación de Municipios - virtual			
Objetivo de la reunión:				Revisión comportamiento anómalo del sistema de radicación y seguimiento trámites catastrales y revisión de datos ingresados en base de datos.			

Participantes			
No.	Nombre	Entidad	Cargo / Contratista©
1	Yolimar Rodríguez	Asomunicipios	Contratista
2	David Steven Torres	Asomunicipios	Contratista
3	Lina Margarita Lora	Asomunicipios	Contratista
4			
Nota: Después de cinco participantes, utilizar listado de asistencia			
*Persona(s) conectada(s) virtualmente			

Orden del día		
No.	Tema	Responsable
1.	Saludo y apertura de la reunión	Yolimar Rodríguez
2.	Revisión comportamiento anómalo del sistema de radicación y seguimiento trámites catastrales y revisión de datos ingresados en base de datos	Todos los asistentes

Desarrollo de la reunión

Inicia la reunión con el fin de dar a conocer con ejemplos los trámites catastrales que generan un comportamiento anómalo al realizar navegación por las distintas opciones de radicación del trámite, se tomó como ejemplo los id 1811, 318, 391, donde se identificó que para el tipo de trámite de mutación cuarta, donde se realiza una revisión de avalúo catastral, un valor que aparecía en la lista de valores del campo Revisión Busca (IN - Incrementar) se repetía en otra lista de valores, por este motivo el sistema tomaba una referencia que no era y no permitía seguir con el proceso normal en el sistema.

Para esto desde base de datos se realizó una actualización a los valores de la lista de valores llamada REVISION_BUSCA con los siguientes cambios INC - Incrementar, DIS – Disminuir.

```
UPDATE public.valores_listas
   SET valor = 'INC'
 WHERE nombre_lista = 'REVISION_BUSCA'
   AND id_valor_lista = 1199;

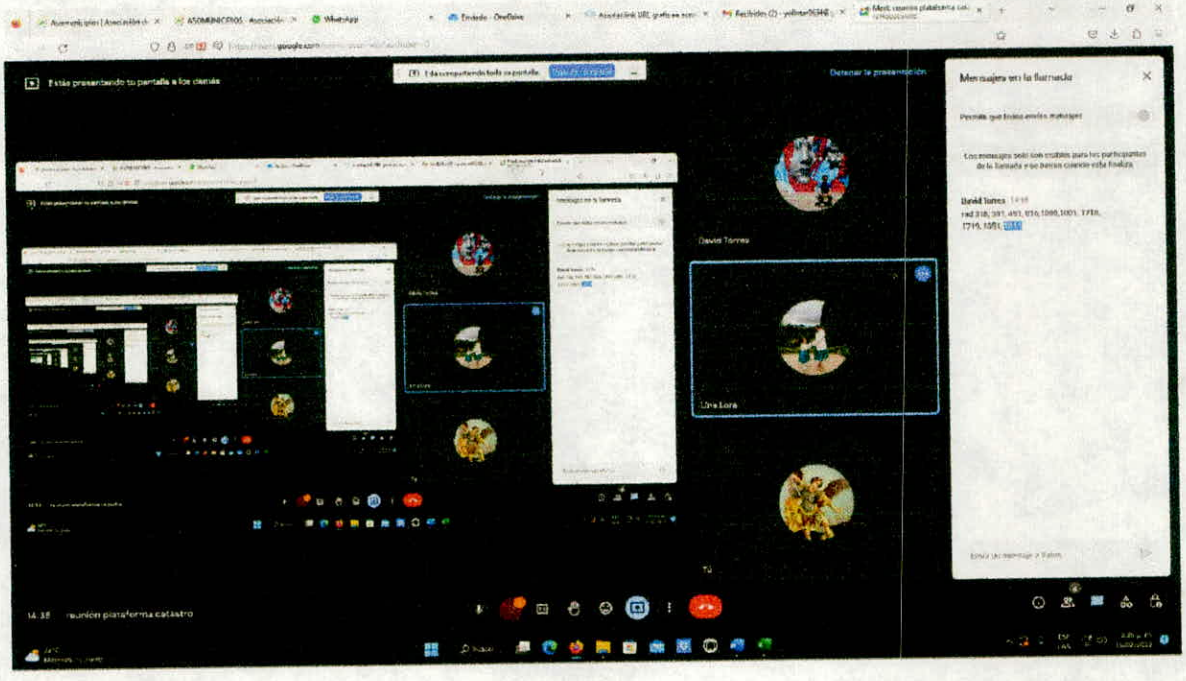
UPDATE public.valores_listas
   SET valor = 'DIS'
 WHERE nombre_lista = 'REVISION_BUSCA'
   AND id_valor_lista = 1200;
```

También se habló acerca de los datos que se están guardando actualmente desde la plataforma, se identificó que el campo Nombre Trámite estaba mostrando el mismo valor que el cambo Tipo Trámite, para esto se revisará a nivel de desarrollo y base de datos la inserción de datos y variables correspondientes por cada tipo de trámite y así realizar el ajuste a nivel de aplicación y a nivel de base de datos.

Finalmente se definieron las consultas que se debían generar a partir de vistas y funciones dentro de la base de datos como lo son:

1. Consulta de Tramites mensuales
2. Consulta de Tramites finalizados
3. Consulta de Tipos de tramites por Municipio mensual, cuantos ingresaron y cuantos finalizaron.

Se adjunta evidencia de la reunión virtual.



Compromiso(s)		
Actividad(es)	Responsable(s) Nombre(s) / entidad	Fecha cumplimiento dd/mm/aaaa
Revisar a nivel de desarrollo y base de datos la inserción de datos y variables correspondientes por cada tipo de trámite.	David Torres, Lina Lora, Rigoberto Ríos	03/03/2023

Anexo(s)	
Nombre del anexo	Número de folios

Elaboró el acta:

Firma(s)	
	
Lina Margarita Lora Contratista	Yolimar Rodríguez Contratista
	
David Steven Torres Contratista	

Información general de la reunión							
Fecha:	Día	Mes	Año	Hora de inicio	3:00 pm	Hora de finalización	4:00 pm
	03	03	2023				
Lugar de la reunión:	Sala de juntas Asociación de Municipios - virtual						
Objetivo de la reunión:	Entrega de resultados en desarrollo de los ajustes en el sistema de trámites catastrales.						

Participantes			
No.	Nombre	Entidad	Cargo / Contratista
1	Yolimar Rodríguez	Asomunicipios	Contratista
2	Lina Margarita Lora	Asomunicipios	Contratista
3	David Steven Torres	Asomunicipios	Contratista
4	Rigoberto Ríos Hueso	Asomunicipios	Contratista
5	Eliana Rodríguez	Asomunicipios	Contratista
6	Juan Camilo Alsina	Asomunicipios	Contratista
7	Dalgie Esperanza Torrado Rizo	Asomunicipios	Supervisora
Nota: Después de cinco participantes, utilizar listado de asistencia			
*Persona(s) conectada(s) virtualmente			

Orden del día		
No.	Tema	Responsable
1.	Saludo y apertura de la reunión	Yolimar Rodríguez
2.	Entrega de resultados en desarrollo de los ajustes en el sistema de trámites catastrales.	Todos los asistentes

Desarrollo de la reunión

Inicia la reunión con el fin de entregar los resultados en desarrollo de los ajustes en el sistema de trámites catastrales en ambiente de pruebas para la posterior implementación.

Se tomó en cuenta los ajustes solicitados y planteados en las actividades contractuales así:

1. Realizar el cambio de correo de contacto para la recuperación de contraseña.

Para este requerimiento se hizo el ajuste en el sistema y se cambió el correo de contacto en los archivos de configuración asociados

2. Para el formulario de detalle de solicitudes revisar y ajustar el comportamiento anómalo de la funcionalidad al descargar un archivo con la documentación de la solicitud, así como funcionalidad de cambio de estado de la solicitud.

Para este requerimiento se generó reunión el pasado 15 de febrero donde se identificó que, para el tipo de trámite de mutación cuarta, donde se realiza una revisión de avalúo catastral, un valor que aparecía en la lista de valores del campo Revisión Busca (IN - Incrementar) se repetía en otra lista de valores, por este motivo el sistema tomaba una referencia que no era y no permitía seguir con el proceso normal en el sistema.

Para esto desde base de datos se realizó una actualización a los valores de la lista de valores llamada REVISION_BUSCA con los siguientes cambios INC - Incrementar, DIS – Disminuir.

```
UPDATE public.valores_listas
   SET valor = 'INC'
 WHERE nombre_lista = 'REVISION_BUSCA'
   AND id_valor_lista = 1199;

UPDATE public.valores_listas
   SET valor = 'DIS'
 WHERE nombre_lista = 'REVISION_BUSCA'
   AND id_valor_lista = 1200;
```

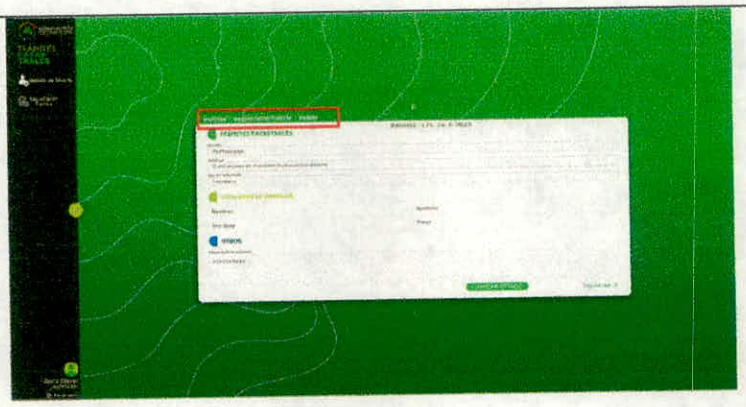
3. Para todo el sistema revisar el tiempo de cierre de sesión automático por inactividad, ya que actualmente el sistema cierra sesión automáticamente cada 5 minutos si no existe ninguna interacción con el sistema.

Para este requerimiento se hizo el ajuste en el sistema en los archivos de configuración para que la sesión tenga mas tiempo de espera de actividad.

4. Implementar en el sistema, un esquema de navegación tipo "Miga de Pan", esto quiere decir que al ingresar a los diferentes componentes se pueda regresar progresivamente de acuerdo con el flujo trabajado.

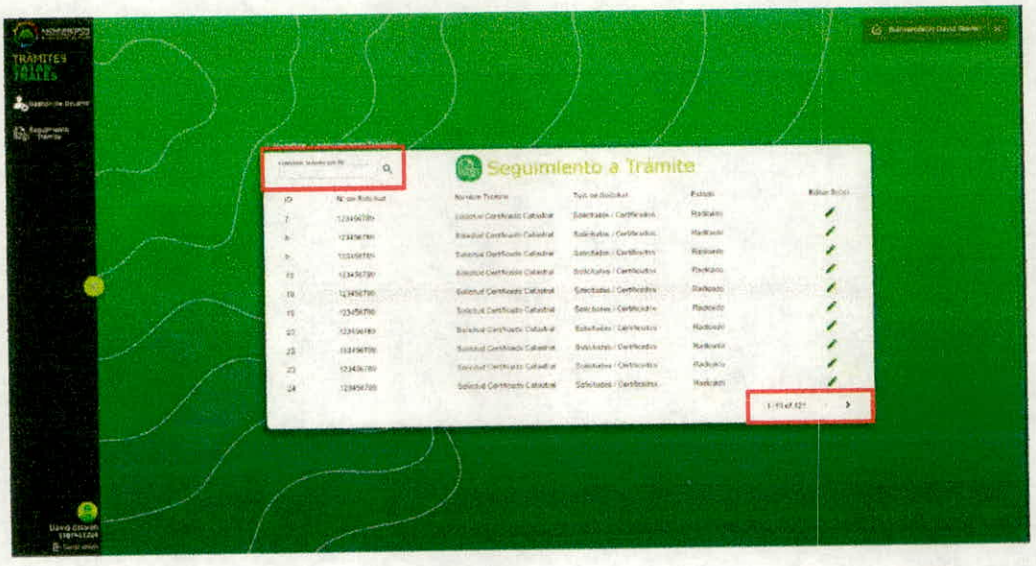
Para este requerimiento se mostró que se ajustó para que a nivel de presentación se tenga una ruta de sitios donde se encuentra en el momento, esto con el fin de poder navegar en las diferentes ventanas a partir de esta ruta.





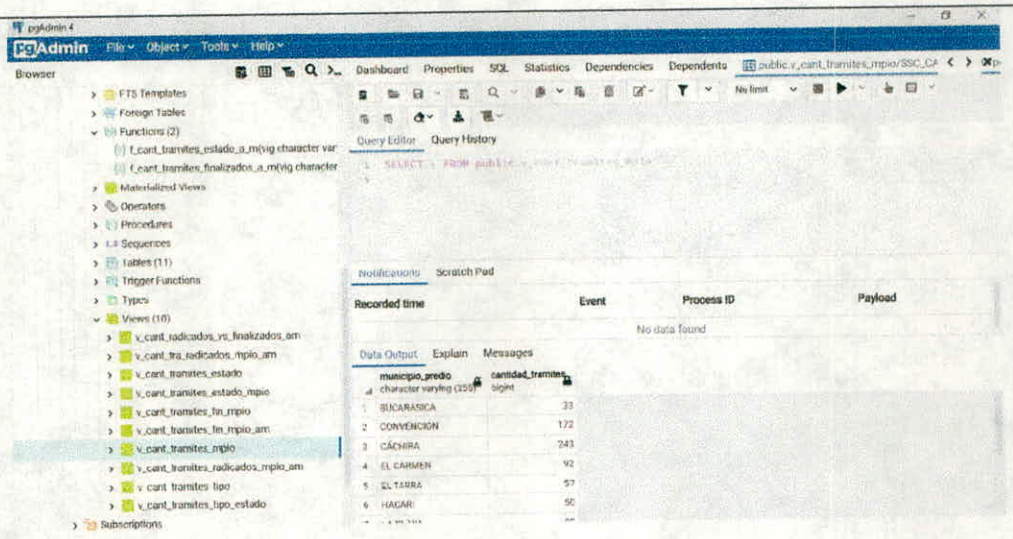
5. En el formulario de consulta de solicitudes realizar la optimización para que al cargar los registros de base de datos no presente largos tiempos de respuesta

Para este requerimiento se realizó ajuste de paginación del sistema con el fin de realizar consulta de los datos a la base de datos con un filtro de registros, esto permite una optimización en tiempo de respuesta de consulta, adicionalmente se creó un campo búsqueda que permite realizar filtros de los trámites por el id del trámite o por el número de radicado del trámite.



6. Revisar y optimizar las vistas que generan las diferentes consultas de trámites en base de datos.

Para este requerimiento se generó reunión el pasado 22 de febrero donde se realiza entrega de consultas, vistas y funciones para elaborar los diferentes reportes descriptivos como seguimiento al estado de los trámites catastrales dentro de la base de datos; así mismo en esta reunión se da muestra del desarrollo de base de datos.



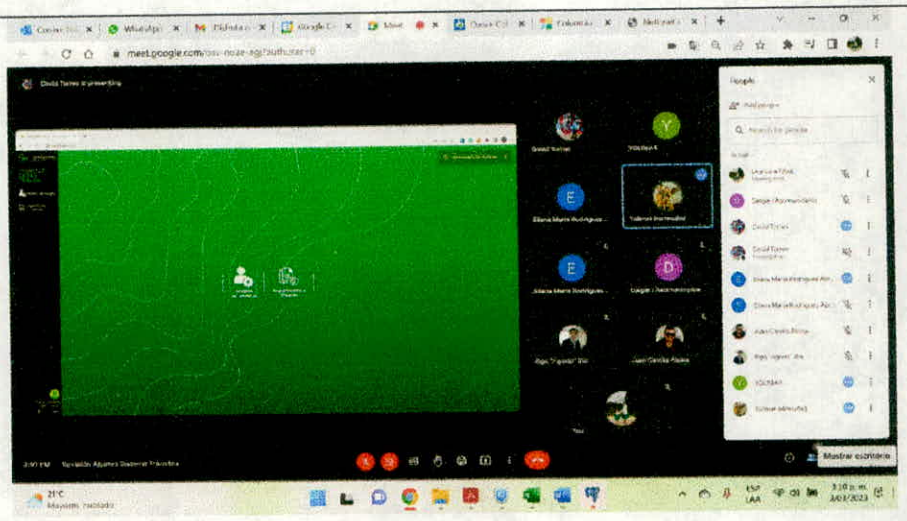
7. Para el formulario de estados de solicitudes, ampliar el número de caracteres del campo observación

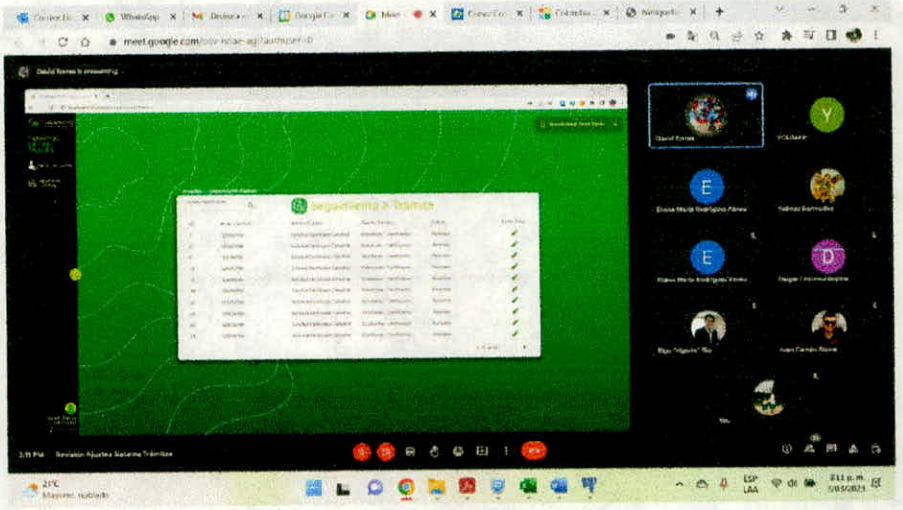
Para este requerimiento se realizó ajuste tanto en base de datos como en el sistema ampliando los caracteres al campo observaciones y que este campo va incluido en el correo electrónico que el sistema envía.

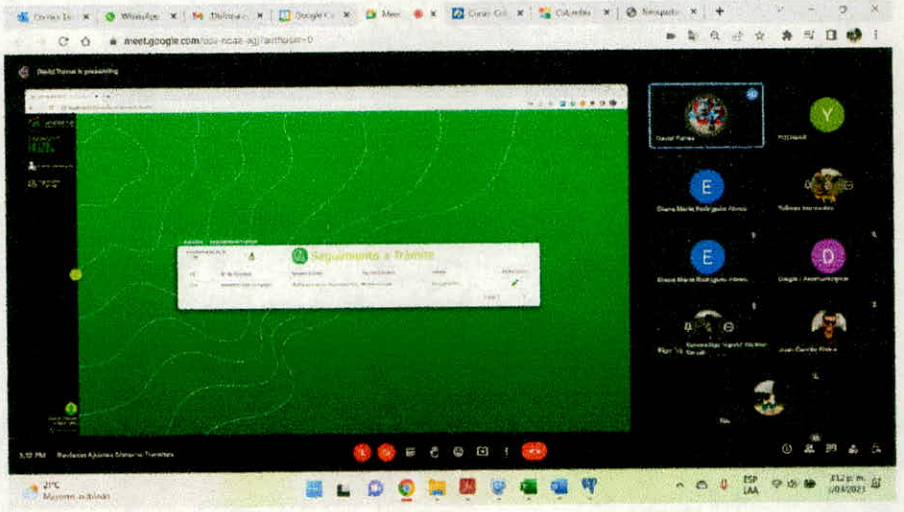


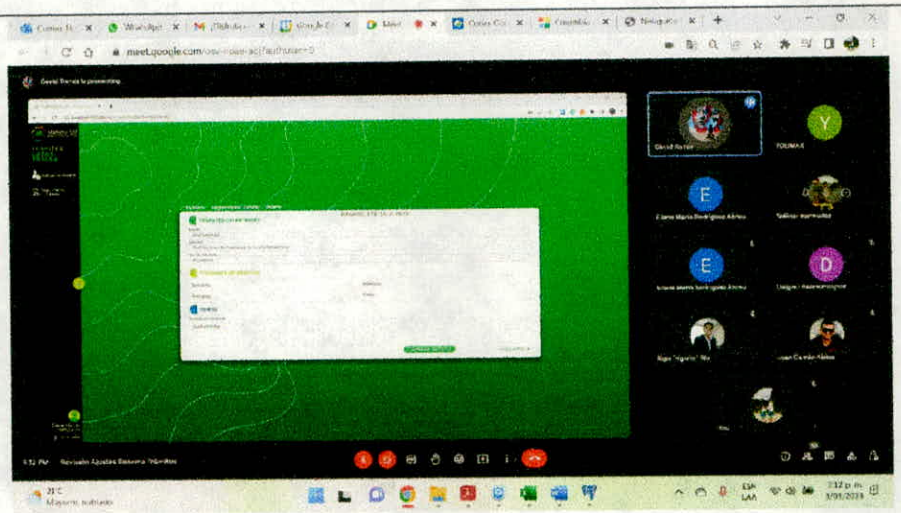
Con esto se da por validado y aprobado los ajustes realizados, como respuesta a las actividades contractuales de los contratos de David Steven Torres, Rigoberto Ríos Hueso y Lina Margarita Lora Matiz

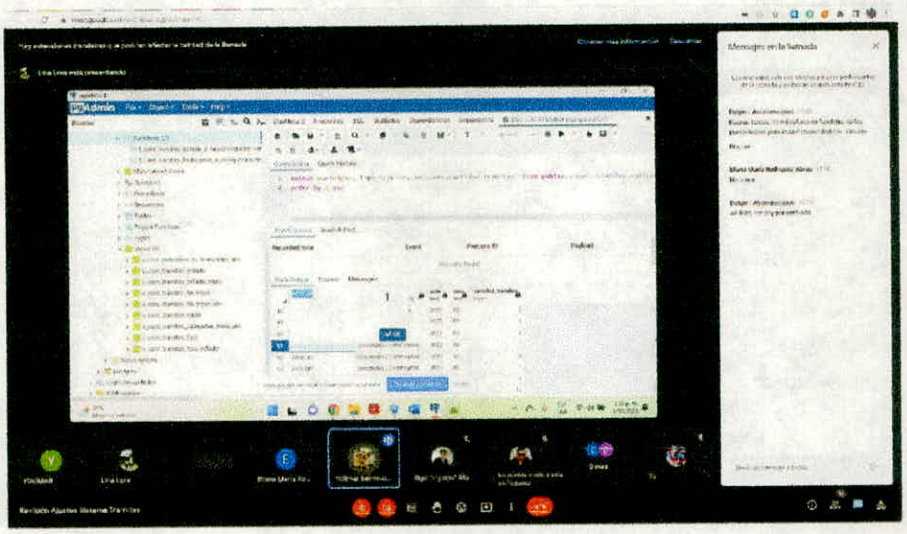
Se adjunta evidencia de la reunión virtual.

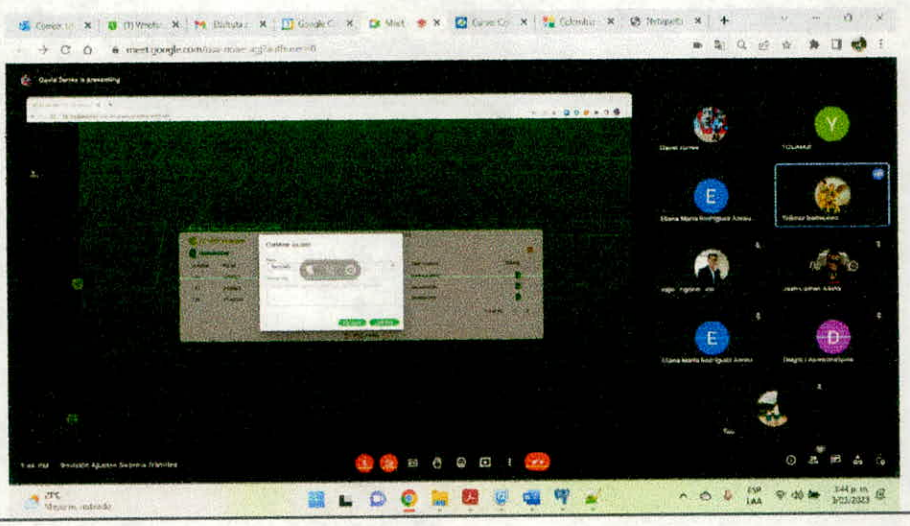












Compromiso(s)		
Actividad(es)	Responsable(s) Nombre(s) / entidad	Fecha cumplimiento dd/mm/aaaa

Anexo(s)	
Nombre del anexo	Número de folios

Elaboró el acta:

Firma(s)	
Yolimar Rodríguez Contratista	Lina Margarita Lora Contratista
David Steven Torres Contratista	Rigoberto Ríos Hueso Contratista
Eliana Rodríguez Contratista	Juan Camilo Alsina Contratista
Dalgie Esperanza Torrado R Supervisora	

MANUAL DE IMPLEMENTACION PROYECTO SSC EN ENTORNOS LINUX

Versión 1.0

Marzo 2023

TABLA DE CONTENIDO

ALCANCE 5

INTRODUCCIÓN 5

1. *DESCARGA E INSTALACIÓN JDK 11* 6
2. *DESCARGA SERVIDOR WILDFLY 19* 8
3. *EJECUCIÓN MANUAL SERVIDOR WILDFLY 19* 10
4. *CONFIGURACIÓN SERVIDOR WILDFLY 19 PARA INICIO CON SISTEMA OPERATIVO* 11
5. *EJECUCIÓN APLICACIÓN SSC* 14

LISTADO DE FIGURAS

Figura 1. Instalación vacia Java 6
Figura 2. Instalación JDK 11 7
Figura 3. Versión de Java Instalada 7
Figura 4. Descarga Servidor WildFly 8
Figura 5. Descarga WildFly 19 8
Figura 6. Ejecución Manual WildFly 10
Figura 7. Archivo Launch.sh 11
Figura 8. Unidad de Servicio 12
Figura 9. Estado Proceso Servidor Aplicaciones 13
Figura 10. Ejecución Web WildFly 13
Figura 11. Ejecución Aplicación SSC 14
Figura 12. Ingreso Aplicación SSC 15

INFORMACIÓN DEL DOCUMENTO

Proyecto:	Sistema de Solicitudes Catastrales
Nombre del Documento:	Manual de implementación del proyecto SSC en ambientes Linux.

CONTROL DE VERSIONES DEL DOCUMENTO

Versión	Fecha	Descripción del cambio	Elaborado por
1.0	2023-03-07	Creación del documento	David Steven Torres Figueroa

ALCANCE

Presentar el manual de implementación sobre ambientes Linux para el proyecto SSC en infraestructura del cliente, de acuerdo con el contrato 022/2023.

INTRODUCCIÓN

El presente manual proporciona las instrucciones y procedimientos necesarios para realizar la instalación de cada uno de los componentes de software que hacen parte del proyecto SSC.

El procedimiento descrito a continuación es acompañado por capturas de pantalla que servirán de guía para la configuración/despliegue en el ambiente definitivo.

1. DESCARGA E INSTALACIÓN JDK 11

¿QUE ES WILDFLY?

Es un servidor de aplicaciones Java de código abierto y multiplataforma, compatible con cualquier sistema operativo en el que se encuentre disponible la máquina virtual de Java, y permite la ejecución y administración de aplicaciones desarrolladas bajo el lenguaje de programación Java, es adaptable a múltiples entornos y bastante flexible en su configuración.

La aplicación SSC está diseñada con el lenguaje de programación JAVA, por ende, para poder ejecutar aplicaciones de este tipo es necesario contar con la instalación del JDK en su versión 11, para este caso se realiza el procedimiento para la versión open source llamada OpenJDK, puesto que para entornos productivos la versión de Oracle necesita ser licenciada.

Para corroborar que no se cuenta con ninguna versión instalada se debe ejecutar el comando:

```
java -version
```

Debería mostrarse lo siguiente:

```
No se ha encontrado la orden «java», pero se puede instalar con:

sudo apt install openjdk-11-jre-headless  # version 11.0.8+10-0ubuntu1~20.04, or
sudo apt install default-jre              # version 2:1.11-72
sudo apt install openjdk-8-jre-headless   # version 8u265-b01-0ubuntu2~20.04
sudo apt install openjdk-13-jre-headless  # version 13.0.3+3-1ubuntu2
sudo apt install openjdk-14-jre-headless  # version 14.0.1+7-1ubuntu1
```

Figura 1. Instalación vacia Java

Para realizar la instalación se debe realizar con el siguiente comando:

```
sudo apt install openjdk-11-jdk
```

Esto empezará con la descarga e instalación del JDK

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc
  openjdk-11-demo openjdk-11-source visualvm fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk
  openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 21 newly installed, 0 to remove and 0 not upgraded.
Need to get 275 MB of archives.
After this operation, 431 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Figura 2. Instalación JDK 11

Después de este proceso ya es posible obtener la información de la versión instalada, para esto se debe ejecutar nuevamente el comando `java -version` y en esta ocasión el resultado será el siguiente:

```
openjdk 11.0.9.1 2020-11-04
OpenJDK Runtime Environment (build 11.0.9.1+1-Ubuntu-0ubuntu1.20.04)
OpenJDK 64-Bit Server VM (build 11.0.9.1+1-Ubuntu-0ubuntu1.20.04, mixed mode, sh
aring)
```

Figura 3. Versión de Java Instalada

2. DESCARGA SERVIDOR WILDFLY 19

Para realizar el despliegue de la aplicación SSC se debe contar con una instancia del servidor de aplicaciones WildFly en su versión 19.0.0.Final, este componente se hace necesario para poder ejecutar aplicaciones web desarrolladas con el lenguaje Java, para realizar la descarga se debe hacer lo siguiente:

Ingresar a la URL https://www.wildfly.org/ y dirigirse a la sección de descargas

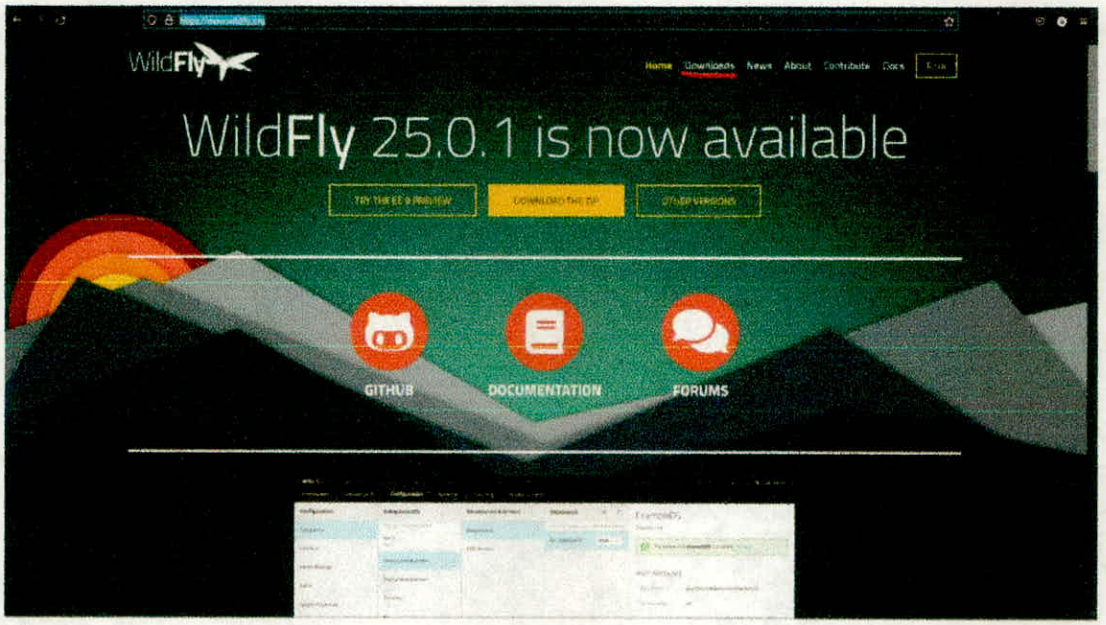


Figura 4. Descarga Servidor WildFly

Se debe buscar y descargar la versión en cuestión desde aquí

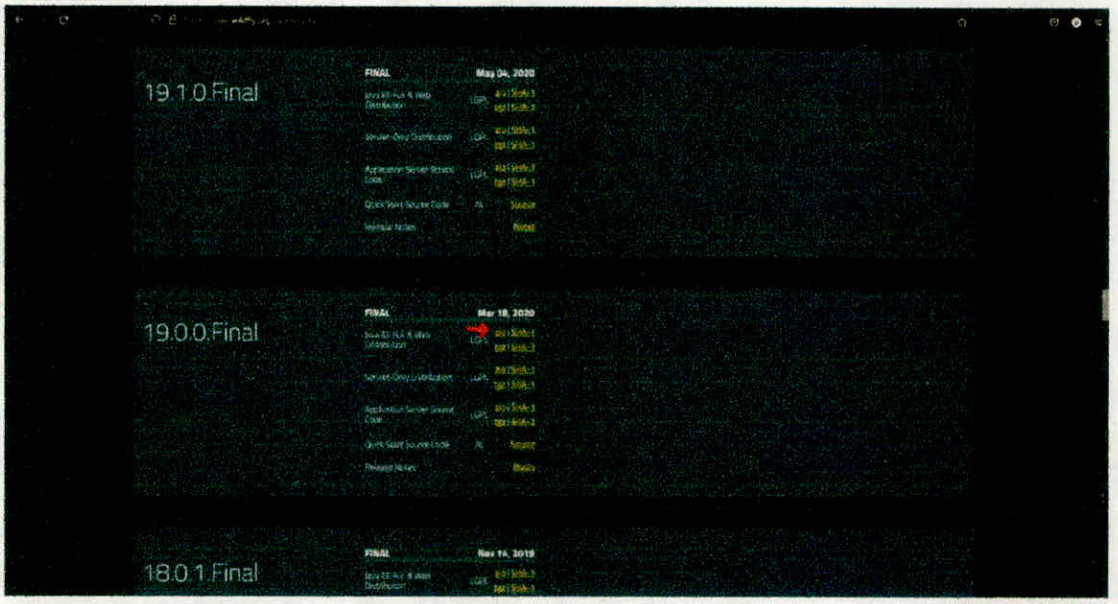


Figura 5. Descarga WildFly 19

Una vez este descargado se debe crear un directorio en la siguiente ruta mediante el comando:

```
sudo mkdir /opt/wildFlyAsomunicipios
```

Dependiendo de donde se haya realizado la descarga del archivo en formato zip de WildFly se debe copiar el archivo descargado a la ruta creada anteriormente:

```
sudo mv wildfly-19.0.0.Final.zip /opt/wildFlyAsomunicipios/
```

Seguido a esto se debe descomprimir el archivo sobre esa ruta con los siguientes comandos:

```
cd /opt/wildFlyAsomunicipios/
```

```
sudo tar xf wildfly-19.0.0.Final.zip
```

o el siguiente comando es equivalente

```
sudo unzip wildfly-19.0.0.Final.zip
```

3. EJECUCIÓN MANUAL SERVIDOR WILDFLY 19

Para ejecutar el servidor es necesario que el directorio posea permisos de lectura/escritura y ejecución para que no se tengan inconvenientes, para esto se deben ejecutar los siguientes comandos:

```
sudo chmod -R 777 /opt/wildFlyAsomunicipios/wildfly-19.0.0.Final/
```

```
sudo chmod -R +X /opt/wildFlyAsomunicipios/wildfly-19.0.0.Final/
```

Luego ubicarse dentro de la ruta `/opt/wildFlyAsomunicipios/wildfly-19.0.0.Final/bin/` y dentro de esta se debe ejecutar lo siguiente:

```
nohup ./startup.sh -b 0.0.0.0 &
```

Si la ejecución es exitosa le ejecución debería ser accesible desde un navegador web accediendo a la URL `http://localhost:8080` y debería visualizarse la siguiente interfaz:



Figura 6. Ejecución Manual WildFly

Esta ejecución funciona mientras la maquina se encuentre encendida, pero si esta es reiniciada el proceso no inicia de forma automática y se debe nuevamente iniciar de forma manual, para configurar el proceso como un servicio del sistema operativo en el capitulo 4 se explica este proceso.

4. CONFIGURACIÓN SERVIDOR WILDFLY 19 PARA INICIO CON SISTEMA OPERATIVO

Para este caso particular la configuración del servidor de aplicaciones es la que se maneja por defecto, la cual es que el servidor de aplicaciones se ejecuta en modo standalone y por el puerto 8080.

Luego de descomprimir el archivo que contiene WildFly se debe crear un directorio en `etc` para almacenar los archivos que contienen la configuración del servidor de aplicaciones, esto se realiza con el siguiente comando:

```
sudo mkdir -p /etc/wildfly
```

A este directorio se debe enviar el siguiente archivo de configuración:

```
sudo cp /opt/wildFlyAsomunicipios/wildfly-
19.0.0.Final/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
```

Como se enuncio anteriormente en este caso la configuracion no se modificó y se aplicó la que está por defecto, por ende, no se realizaron modificaciones en este archivo, y no deben modificarse a menos que se requiera algún requerimiento especifico.

Luego de esto se debe el archivo que permite realizar la ejecución del servidor de aplicaciones, para esto se debe ejecutar el comando:

```
sudo cp /opt/wildFlyAsomunicipios/wildfly-
19.0.0.Final/docs/contrib/scripts/systemd/launch.sh
/opt/wildFlyAsomunicipios/wildfly-19.0.0.Final/bin/
```

En este caso debe modificarse el contenido del archivo `launch.sh` para que el valor de la variable `WILDFLY_HOME` corresponda a `/opt/wildFlyAsomunicipios/wildfly-19.0.0.Final`

```
#!/bin/bash

if [ "x$WILDFLY_HOME" = "x" ]; then
    WILDFLY_HOME="/opt/wildflyTadeo/wildfly-19.0.0.Final"
fi

if [[ "$1" == "domain" ]]; then
    $WILDFLY_HOME/bin/domain.sh -c $2 -b $3
else
    $WILDFLY_HOME/bin/standalone.sh -c $2 -b $3
fi
```

Figura 7. Archivo Launch.sh

Paso seguido se debe crear el archivo que va a contener la unidad de servicio para que el servidor pueda ser ejecutado junto al inicio del sistema operativo y a si mismo pueda recibir y ejecutar señales de inicio/reinicio y finalización.

Se debe ejecutar el siguiente comando:

```
sudo cp /opt/wildFlyAsomunicipios/wildfly-
19.0.0.Final/docs/contrib/scripts/systemd/wildfly.service
/etc/systemd/system/
```

Sobre el contenido de este archivo se debe modificar lo siguiente:

- Ubicar la línea que contenga `ExecStart` y modificar por lo siguiente `/opt/wildFlyAsomunicipios/wildfly-19.0.0.Final/bin/launch.sh/`
- Modificar User por root.

El archivo debería quedar de la siguiente manera:

```
[Unit]
Description=The WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=-/etc/wildfly/wildfly.conf
User=root
LimitNOFILE=102642
PIDFile=/var/run/wildfly/wildfly.pid
ExecStart=/opt/wildflyTadeo/wildfly-19.0.0.Final/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG $WILDFLY_BIND
StandardOutput=null

[Install]
WantedBy=multi-user.target
```

Figura 8. Unidad de Servicio

Seguido a esto se debe realizar la actualización del proceso que maneja los servicios que se ejecutan en el sistema operativo, se debe ejecutar el comando:

```
sudo systemctl daemon-reload
```

En este punto si la configuración se realizó la configuración de manera correcta ya es posible iniciar el servicio de WildFly y este estará ejecutándose sobre el puerto por defecto que es el 8080, esto se realiza con el comando:

```
sudo systemctl start wildfly
```

Para habilitar el servicio con el inicio del sistema operativo se debe ejecutar:

```
sudo systemctl enable wildfly
```

Para obtener el estado en el que se encuentra el servicio se debe ejecutar el siguiente comando

```
sudo systemctl status wildfly
```

Si todo ha funcionado correctamente se debe visualizar lo siguiente:



Figura 9. Estado Proceso Servidor Aplicaciones

Por último, se puede visualizar la ejecución del servidor accediendo a la URL http://localhost:8080 y debería visualizarse la siguiente interfaz:



Figura 10. Ejecución Web WildFly

5. EJECUCIÓN APLICACIÓN SSC

Para poder ejecutar la aplicación SSC en ambos casos, ya sea en ejecución manual o como servicio del sistema operativo es necesario ubicar el archivo .war que contiene la aplicación en cuestión, este nombre es `SSC-0.0.1-SNAPSHOT.war` con este archivo simplemente debe ser ubicado dentro de la ruta `/opt/wildFlyAsomunicipios/wildfly-19.0.0.Final/standalone/deployments/` siempre y cuando este no se encuentre ya en esta ruta.

Al realizar la ejecución por cualquiera de las dos alternativas el resultado debería ser el siguiente:



Figura 11. Ejecución Aplicación SSC

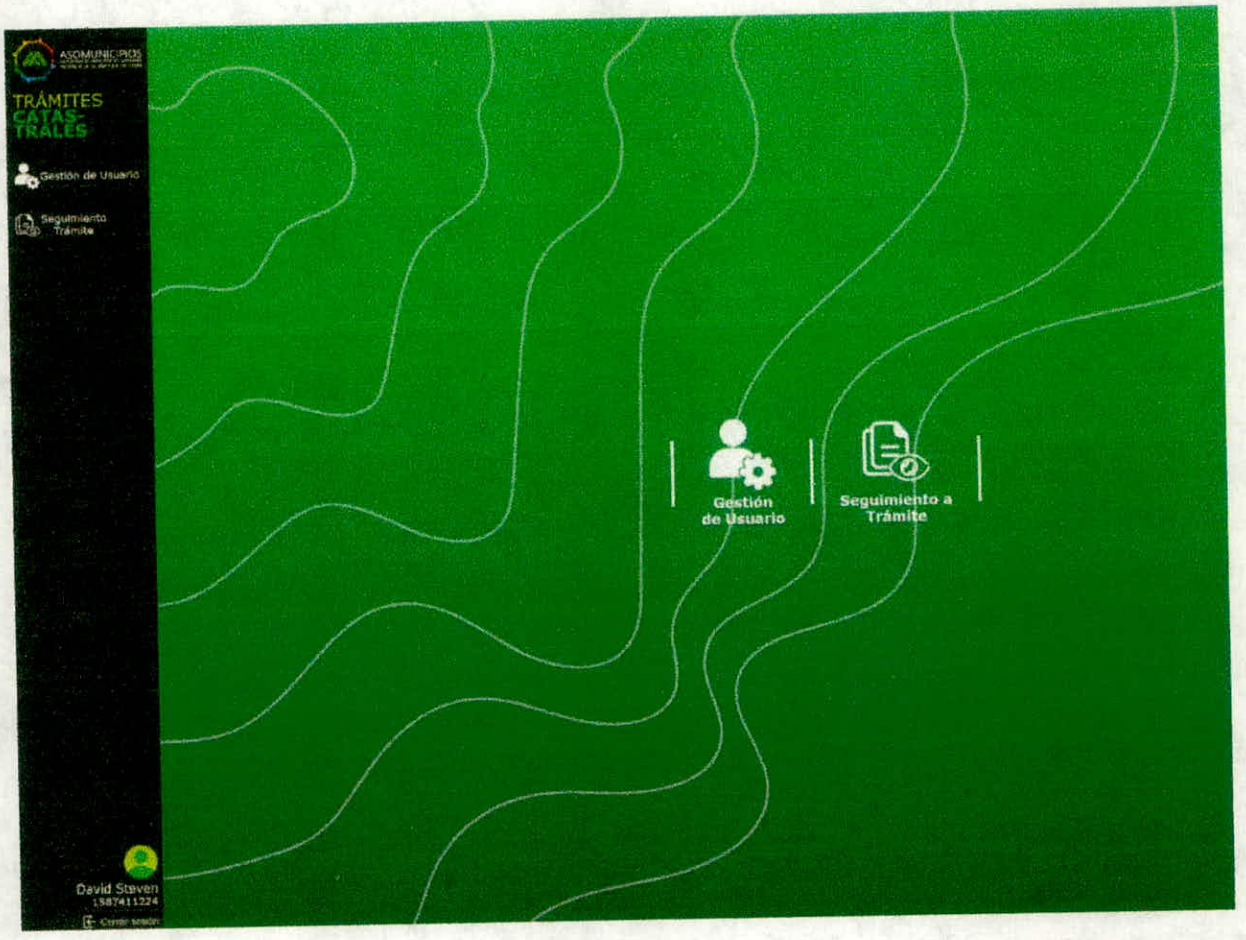


Figura 12. Ingreso Aplicación SSC

compensar | miplanilla.com

PLANILLA INTEGRADA DE AUTOLIQUIDACION DE APORTES

DATOS DEL APORTANTE						
TIPO	NÚMERO	NOMBRE APORTANTE	DIRECCIÓN	TELÉFONO	CORREO	EXONERADO PAGO PARAFISCALES Y SALUD
CC	1013611554	DAVID STEVEN TORRES FIGUEROA	DG 35B SUR 2B- 16	3620417	DAVIDS182009@GMAIL.COM	
FORMA PRESENTACIÓN	CLASE APORTANTE	NOMBRE SUCURSAL / CÓDIGO	DEPARTAMENTO	CIUDAD / MUNICIPIO		
ÚNICA	I – Independiente		BOGOTÁ D. C.	BOGOTÁ, D.C.		SI

DATOS DE LA PLANILLA						
PLANILLA ASOCIADA	FECHA PAGO ASOCIADA (DIA/MES/AÑO)	TIPO PLANILLA	FECHA PAGO (DIA/MES/AÑO)	NÚMERO PLANILLA	CANTIDAD	
					EMPLEADOS	UPC
PERIODO SALUD	PERIODO PENSIONES				1	0
					TOTAL A PAGAR	
2023-02	2023-02	I	27/02/2023	65846090	$1.025.800	

TOTALES POR SUBSISTEMAS

TOTALES SALUD												
Código EPS	Nombre	NIT	Cotización Obligatoria	UPC Adicional	Incapacidades		Licencia Maternidad		Días Mora	Valor Mora Cotización	Valor Mora UPC	Total a Pagar	No. Afiliados
					No. Autorización	Valor	No. Autorización	Valor					
EPS008	Compensar EPS	860066942-7	441.800	0		0		0	0	0	0	441.800	1

TOTALES PENSIÓN												
Código AFP	Nombre	NIT	Cotización Obligatoria	Aporte Voluntario Afiliado	Aporte Voluntario Aportante	Aporte FSP - Solidaridad	Aporte FSP - Subsistencia	Días Mora	Valor Mora Cotización	Valor Mora FSP	Total a Pagar	No. Afiliados
25-14	Colpensiones	900336004-7	565.500	0	0	0	0	0	0		565.500	1

TOTALES RIESGOS LABORALES															
Código ARL	Nombre	NIT	Cotización Obligatoria	Incapacidades		Aportes Otros Sistemas	Valor Neto Cotización	Días Mora	Valor Mora Cotización	Subtotal Cotización	No. Radicado Saldo a Favor	Valor Saldo a Favor	Fondo Solidaridad	Total a Pagar	No. Afiliados
				No. Autorización	Valor										
14-23	Positiva Seguros	860011153-6	18.500				18.500	0	0	18.500			185	18.500	1

TOTALES CAJAS							
Código CCF	Nombre	NIT	Valor Aporte	Días Mora	Valor Mora Aporte	Total a Pagar	No. Afiliados

TOTALES PARAFISCALES				
Valor Aporte	Días Mora	Valor Mora Aporte	Total a Pagar	No. Afiliados
SENA				
0	0	0	0	0
ICBF				
0	0	0	0	0
ESAP				
MEN				

TOTALES POR SUBSISTEMA			
Tipo Administradora	No. Administradoras Reportadas	Valor antes de IGE, LMA, IRP y Mora	Total a Pagar
Salud	1	441.800	441.800
Pensión	1	565.500	565.500
Riesgos Laborales	1	18.500	18.500
CCF	0	0	0
ESAP	0	0	0
ICBF	0	0	0
MEN	0	0	0
SENA	0	0	0
TOTALES	3	1.025.800	1.025.800

PAGADA

Este documento está clasificado como PRIVADO por parte de Compensar Operador de Información

compensar | miplanilla.com

PLANILLA INTEGRADA DE AUTOLIQUIDACION DE APORTES

DATOS DEL APORTANTE						
TIPO	NÚMERO	NOMBRE APORTANTE	DIRECCIÓN	TELÉFONO	CORREO	EXONERADO PAGO PARAFISCALES Y SALUD
CC	1013611554	DAVID STEVEN TORRES FIGUEROA	DG 35B SUR 2B- 16	3620417	DAVIDS182009@GMAIL.COM	
FORMA PRESENTACIÓN	CLASE APORTANTE	NOMBRE SUCURSAL / CÓDIGO	DEPARTAMENTO	CIUDAD / MUNICIPIO		
ÚNICA	I – Independiente		BOGOTÁ D. C.	BOGOTÁ, D.C.		SI

DATOS DE LA PLANILLA						
PLANILLA ASOCIADA	FECHA PAGO ASOCIADA (DIA/MES/AÑO)	TIPO PLANILLA	FECHA PAGO (DIA/MES/AÑO)	NÚMERO PLANILLA	CANTIDAD	
					EMPLEADOS	UPC
					1	0
PERIODO SALUD	PERIODO PENSIONES				TOTAL A PAGAR	
2023-02	2023-02	I	27/02/2023	65846090	$1.025.800	

DETALLE POR COTIZANTE

INFORMACIÓN COTIZANTE								INFORMACIÓN NOVEDADES																PENSIÓN							SALUD			RIESGOS LABORALES				CCF			PARAFISCALES				
No.	Tipo	No. de Identificación	Apellidos y Nombres	Cotizante	Subtipo	Extranjero	Colombiano en el exterior	Exonerado	IGE	RET	TDE	TAE	TDP	TAP	VSP	VST	SLN	IGE	LMA	VAC	AVP	VCT	IRL	CORRECCIÓN	Cód. AFP	IBC AFP	Cotización	Voluntario Afiliado	Voluntario Aportante	Fondo pensional de solidaridad	Fondo pensional de subsistencia	Cód. EPS	IBC EPS	Cotización / Valor UPC	Cód. ARL	IBC ARL	Clase de Riesgo	Cotización	Código CCF	IBC CCF	Aporte CCF	IBC otros parafiscales	Aporte SENA	Aporte ICBF	Aporte ESAP	Aporte MEN
1	CC	1013611554	TORRES FIGUEROA DAVID STEVEN	59	0			N																	25-14	3.534.204	565.500	0	0	0	0	EPS008	3.534.204	441.800	14-23	3.534.204	1	18.500		0	0	0	0	0	0	0	0

PAGADA

Este documento está clasificado como PRIVADO por parte de Compensar Operador de Información