**Sistema de Validación Documental**

**MANUAL TÉCNICO**



SERVICIO NACIONAL DE PATRIMONIO DEL ESTADO SENAPE

LA PAZ – BOLIVIA

2017

INDICE

[1. INTRODUCCIÓN 3](#_Toc478654925)

[2. ANÁLISIS DE REQUERIMIENTOS 4](#_Toc478654926)

[3. ASPECTOS DEL ANÁLISIS 13](#_Toc478654927)

[4. ESPECIFICACIONES TÉCNICAS 13](#_Toc478654928)

[5. CONFIGURACIÓN DEL APLICATIVO 14](#_Toc478654929)

[6. ESTRUCTURA DEL SITEMA 15](#_Toc478654930)

[7. MODELO ENTIDAD RELACION 19](#_Toc478654931)

[8. DICCIONARIO DE DATOS 19](#_Toc478654932)

[9. VISTAS 40](#_Toc478654933)

[10. PROCEDIMIENTOS ALMACENADOS 58](#_Toc478654934)

**Presentación**

El presente documento técnico fue desarrollado con el propósito de proveer la información necesaria sobre el Sistema de Validación Docuemnetal, desarrollado para el área de control y validación dependiente de la dirección de Registro y promoción del Servicio Nacional de Patrimonio del Estado-SENAPE, dicha Área se encarga de validar los datos de la DEJURBE para poder almacenar información confialbe con datos correctos.

El sistema Validación Documental está bajo el modelo MVC (Modelo, Vistas y Controlador) y la arquitectura cliente-servidor, el presente manual muestra una descripción técnica de cada uno de estos componentes de este modelo, explicando diversos detalles, para la instalación, mantenimiento del presente Sistema.

El presente documento ha sido desarrollado manual técnico tiene como finalidad describir el diseño del SISTEMA DE VALIDACIÓN DOCUMENTAL de la DEJURBE 2016 conocer las especificaciones básicas sobre la elaboración de estructura de datos.

Una vez recibida la información por el Sistema de Validación documental se

El manual técnico hace referencia a información necesaria con el fin de orientar al personal en la concepción, planteamiento, análisis, programación e instalación del sistema VALIDACION DOCUMENTAL. Es de anotar que la redacción propia del manual técnico está orientado a personal con conocimientos de sistemas y tecnologías de información, conocimientos de base de datos, responsables del mantenimiento e instalación del sistema en los servidores.

**Resumen**

El manual técnico hace referencia a información necesaria con el fin de orientar al personal en la concepción, planteamiento, análisis, programación e instalación del sistema VALIDACION DOCUMENTAL

El presente manual técnico, tiene la finalidad de mostrar al Sistema de Préstamo de Documentos desde un punto de vista técnico, para instruir al personal encargado en las actividades de instalación, mantenimiento, revisión, solución y configuración del presente Sistema.

Para ello, mostraremos la parte conceptual y la parte técnica, para una mejor administración del presente Sistema.

El sistema de Prestamos de Documentos-SIPRED, está desarrollado con software libre bajo la arquitectura cliente- servidor, para su desarrollo, se enmarca dentro los estándares abiertos de funcionamiento de los sistemas WEB, apoyado con un gestor de base de Datos relacional, bajo el concepto de software libre.

# OBJETIVO

Tener toda la documentación validada y almacenada en una base de datos

# ANÁLISIS DE REQUERIMIENTOS

**Instrucciones que debe seguir en el Sistema de validación Documental**

En el Sistema de validación documental de la DEJURBE, en todos los rubros tienen las siguientes opciones:

* validar documentación (con esta opción accederá directamente a validar el bien con los documentos declarados)
* adicionar documento (+ Documento, utilizará esta opción en caso de que en la DEJURBE de la entidad que esté validando, encuentre un documento que no haya sido declarado, esta acción se guardará en una tabla adicional y no modificará la información registrada)
* adicionar persona (+ Persona, deberá utilizar esta opción en caso de que el documento del bien que esté validando se encuentre a nombre de una persona particular u otra entidad pública, esta opción será utilizada después de validar el documento) y sin documentación (utilizará esta opción cuando la entidad pública declarante no haya declarado y tampoco haya enviado ningún documento de respaldo que demuestre que son propietarios.

**Bienes Inmuebles**

Datos que debe validar en el Folio Real y Tarjeta de Propiedad: Si adjunta o no el documento, si corresponde o no el documento enviado al bien registrado, si el documento enviado es o no legible; asimismo, deberá revisar y comparar los datos del documento de respaldo con los datos registrados y verificar si corresponde o no el Nº de documento, superficie, Dirección y Nº de catastro (opcional) en caso de que estos datos sean incorrectos, su persona es responsable de introducir los datos correctos (estos datos se almacenarán internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar a las entidades públicas en su DEJURBE), opcionalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, la superficie declarada no corresponde a la superficie del documento adjunto, la dirección o ubicación geográfica del bien declarado no coincide con los datos de la documentación adjunta, el registro de este bien está duplicado con los siguientes Id´s, en el Folio Real la superficie del terreno está declarado con valor 0, no se adjuntó Resolución Ministerial, Administrativa o Municipal (baja), y documento adjunto ilegible.

En los documentos Testimonio de Propiedad, Documento de Transferencia, Minuta de Transferencia de Compra Venta, Certificado Catastral, Planos, Acta de Entrega, Avalúo Técnico, Título Ejecutorial, Ley del Estado Plurinacional, Decreto Supremo, Resolución Ministerial, Ordenanza Municipal, Resolución Municipal, Resolución Administrativa, Documento de Compra Venta, Informe Legal, Informe Técnico, Certificado Alodial y Contrato de Comodato, deberá validar lo siguiente:

* si adjunta o no el documento.
* si corresponde o no el documento enviado al bien registrado.
* si el documento enviado es o no legible, deberá revisar si el dato del Nº de documento corresponde o no al bien declarado, en caso de que sea incorrecto este dato.

el responsable de introducir el dato correcto (este dato se almacenará internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), asimismo, puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, la superficie declarada no corresponde a la superficie del documento adjunto, la dirección o ubicación geográfica del bien declarado no coincide con el dato de la documentación adjunta, el registro de este bien está duplicado con los siguientes Id´s, no se adjuntó Resolución Ministerial, Administrativa o Municipal (baja), y documento adjunto ilegible.

**Bienes Vehículos**

Datos que debe validar en el R.U.A.T.:

* si adjunta o no el documento.
* si corresponde o no el documento enviado al bien registrado.
* si el documento enviado es o no legible
* revisar y comparar los datos del documento de respaldo con los datos registrados y verificar si corresponde o no el Nº de documento, tipo, clase, marca, placa, Nº de motor, Nº de chasis, procedencia y modelo (año) del vehículo, en caso de que estos datos sean incorrectos, el responsable de introducir los datos correctos (estos datos se almacenarán internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), finalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, el registro de este bien está duplicado con los siguientes Id´s, no se adjuntó Resolución Ministerial, Administrativa o Municipal (baja), y documento adjunto ilegible.

Cabe aclarar que el número correcto del R.U.A.T., es el número CRPVA (Certificado de Registro de Propiedad – Vehículos Automotor) y no así el número de formulario de este documento, en el siguiente gráfico se señala el lugar en el que se encuentra el número CRPVA.



En el Documento Único de Importación (DUI), P.T.A., Carnet de Propiedad, Póliza de importación, Factura, Ley del Estado Plurinacional, Decreto Supremo, Documento de Transferencia, Minuta de Transferencia de Compra Venta, Resolución Ministerial, Ordenanza Municipal, Resolución Administrativa, Documento de Compra Venta, Acta de Entrega, Avalúo Técnico, Informe Legal, Informe Técnico y Contrato de Comodato, deberá validar lo siguiente: si adjunta o no el documento, si corresponde o no el documento enviado al bien registrado, si el documento enviado es o no legible; asimismo, deberá revisar si el dato del Nº de documento corresponde o no al bien declarado, en caso de que sea incorrecto este dato, su persona es responsable de introducir el dato correcto (este dato se almacenará internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar a las entidades públicas en su DEJURBE), finalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, el registro de este bien está duplicado con los siguientes Id´s, no se adjuntó Resolución Ministerial, Administrativa o Municipal (baja), y documento adjunto ilegible.

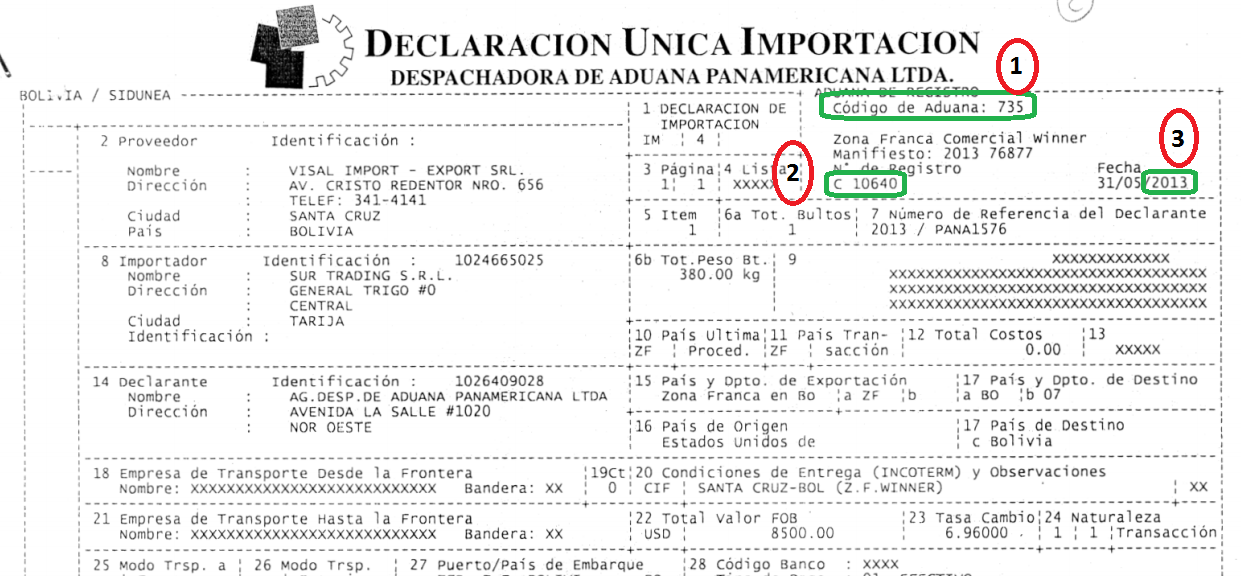
**Bienes Maquinaria Pesada Móvil**

Datos que debe validar en la Póliza de Importación:

* si adjunta o no el documento.
* si corresponde o no el documento enviado al bien registrado.
* si el documento enviado es o no legible; asimismo.
* revisar y comparar los datos del documento de respaldo con los datos registrados y verificar si corresponde o no el Nº de documento, equipo, marca, modelo, Nº de chasis, Nº de motor, Nº de chasis y color, en caso de que estos datos sean incorrectos, su persona es responsable de introducir los datos correctos (estos datos se almacenarán internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), opcionalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, este bien no corresponde ser declarado en la DEJURBE se recomienda dar de baja, el registro de este bien está duplicado con los siguientes ID´s, el documento declarado corresponde a una DUI y no así a una Póliza de Importación, no se adjunta Resolución Ministerial o Administrativa o Municipal (Baja), documento adjunto ilegible.

Datos que debe validar en la Factura, Ley del Estado Plurinacional, Decreto Supremo, Documento de Transferencia, Minuta de Transferencia de Compra Venta, Resolución Ministerial, Ordenanza Municipal, P.T.A., Resolución Administrativa, Documento de Compra Venta, Acta de Entrega, Avalúo Técnico, Informe Legal, Informe Técnico, Contrato de Comodato, Documento Único de Importación (DUI), Carnet de Propiedad, si corresponde o no el documento enviado al bien registrado, si el documento enviado es o no legible; asimismo, deberá revisar si el dato del Nº de documento corresponde o no al bien declarado, en caso de que sea incorrecto este dato, su persona es responsable de introducir el dato correcto (este dato se almacenará internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), finalmente puede observar los siguientes casos: No adjuntaron el documento declarado, este bien corresponde ser declarado en el rubro vehículos, este bien no corresponde ser declarado en la DEJURBE se recomienda dar de baja, el número de chasis declarado no coincide con el número de chasis del documento adjunto, el número de motor declarado no coincide con el número de motor del documento adjunto, el registro de este bien está duplicado con los siguientes Id´s, el documento declarado corresponde a una DUI y no así a una Póliza de Importación, no se adjuntó Resolución Ministerial o Administrativa o Municipal (Baja), documento adjunto ilegible.

Cabe aclarar que el número correcto del Documento Único de Importación – DUI (mayormente las entidades públicas confunden este documento con Póliza de Importación), es la unión de tres datos (se puede observar en el gráfico de la página 7), Código de Aduana (1)/Código C (2)/Fecha (3). Por ejemplo, el número correcto del DUI del gráfico llegaría a ser 735/C10640/2013.



**Bienes Maquinaria y Equipo**

Datos que debe validar en la Factura:

* si adjunta o no el documento.
* si corresponde o no el documento enviado al bien registrado.
* si el documento enviado es o no legible.
* revisar y comparar los datos del documento de respaldo con los datos registrados y verificar si corresponde o no el Nº de documento, equipo, marca, modelo y serie, en caso de que estos datos sean incorrectos, su persona es responsable de introducir los datos correctos (estos datos se almacenarán internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), opcionalmente puede observar los siguientes casos: No adjuntaron el documento declarado, este bien NO tiene un valor mayor o igual al requerido para poder ser declarado en la DEJURBE (D.S. 1991), por lo que se recomienda dar de baja, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, este bien corresponde ser declarado en el rubro maquinaria pesada móvil, este bien no corresponde ser declarado en la DEJURBE, se recomienda dar de baja, el registro de este bien está duplicado con los siguientes ID´s, el documento declarado corresponde a una DUI y no así a una Póliza de Importación, no se adjunta Resolución Ministerial o Administrativa o Municipal (Baja), documento adjunto ilegible.

Datos que debe validar en la Póliza de importación, Ley del Estado Plurinacional, Decreto Supremo, Documento de Transferencia, Minuta de Transferencia de Compra Venta, Resolución Ministerial, Ordenanza Municipal, Resolución Administrativa, Documento de Compra Venta, Acta de Entrega, Avalúo Técnico, Informe Legal, Informe Técnico, Contrato de Comodato, Documento Único de Importación (DUI), Carnet de Propiedad, si corresponde o no el documento enviado al bien registrado, si el documento enviado es o no legible; asimismo, deberá revisar si el dato del Nº de documento corresponde o no al bien declarado, en caso de que sea incorrecto este dato, su persona es responsable de introducir el dato correcto (este dato se almacenará internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar a las entidades públicas en su DEJURBE), finalmente puede observar los siguientes casos: No adjuntaron el documento declarado, este bien NO tiene un valor mayor o igual al requerido para poder ser declarado en la DEJURBE (D.S. 1991), por lo que se recomienda dar de baja, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, este bien corresponde ser declarado en el rubro maquinaria pesada móvil, este bien no corresponde ser declarado en la DEJURBE se recomienda dar de baja, el registro de este bien está duplicado con los siguientes ID´s, el documento declarado corresponde a una DUI y no así a una Póliza de Importación, no se adjunta Resolución Ministerial o Administrativa o Municipal (Baja), documento adjunto ilegible.

**Bienes Inmuebles en Alquiler**

Datos que debe validar en el Contrato de Alquiler del Inmueble:

* si adjunta o no el documento.
* si corresponde o no el documento enviado al bien registrado.
* si el documento enviado es o no legible.
* revisar y comparar los datos del documento de respaldo con los datos registrados y verificar si corresponde o no el Nº de documento, Departamento, Dirección o Ubicación, Inicio del contrato, Conclusión del contrato y canon de alquiler, en caso de que estos datos sean incorrectos, su persona es responsable de introducir los datos correctos (estos datos se almacenarán internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), opcionalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el canon de alquiler descrito en el documento no es igual al canon de alquiler mensual declarado, la dirección y/o ubicación geográfica del bien declarado no coincide con los datos de la documentación adjunta, el registro de este bien está duplicado con los siguientes ID´s, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, la superficie declarada no corresponde a la superficie del documento adjunto, documento adjunto ilegible.

Datos que debe validar en el documento de acuerdo/convenio, si corresponde o no el documento enviado al bien registrado, si el documento enviado es o no legible; asimismo deberá revisar si el dato del Nº de documento corresponde o no al bien declarado, en caso de que sea incorrecto este dato, su persona es responsable de introducir el dato correcto (este dato se almacenará internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), finalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el canon de alquiler descrito en el documento no es igual al canon de alquiler mensual declarado, la dirección y/o ubicación geográfica del bien declarado no coincide con los datos de la documentación adjunta, el registro de este bien está duplicado con los siguientes ID´s, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, la superficie declarada no corresponde a la superficie del documento adjunto, documento adjunto ilegible.

**Bienes Vehículos en Alquiler**

Datos que debe validar en el Contrato de Alquiler del Vehículo:

* si adjunta o no el documento.
* si corresponde o no el documento enviado al bien registrado.
* si el documento enviado es o no legible.
* revisar y comparar los datos del documento de respaldo con los datos registrados y verificar si corresponde o no el Nº de documento, tipo de vehículo, modelo, marca, número de chasis, fecha de inicio de contrato, fecha de finalización del contrato y canon de alquiler, en caso de que estos datos sean incorrectos, su persona es responsable de introducir los datos correctos (estos datos se almacenarán internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), opcionalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, el canon de alquiler descrito en el documento no es igual al canon de alquiler mensual declarado, el número de chasis declarado no coincide con el número de chasis del documento adjunto, el registro de este bien está duplicado con los siguientes ID´s, y documento adjunto ilegible.

Datos que debe validar en el documento Acuerdo o Convenio: Si adjunta o no el documento, si corresponde o no el documento enviado al bien registrado, si el documento enviado es o no legible; asimismo, debe revisar si el dato del Nº de documento corresponde o no al bien declarado, en caso de ser incorrectos estos datos, su persona es responsable de introducir los datos correctos (este dato se almacenará internamente en una tabla y no modificará lo registrado por la entidad pública declarante, servirá para denotar estas observaciones a las entidades públicas en su DEJURBE), opcionalmente puede observar los siguientes casos: No adjuntaron el documento declarado, el documento declarado se encuentra a nombre de otra entidad pública, el documento declarado se encuentra a nombre de una persona particular, el canon de alquiler descrito en el documento no es igual al canon de alquiler mensual declarado, el número de chasis declarado no coincide con el número de chasis del documento adjunto, el registro de este bien está duplicado con los siguientes ID´s, y documento adjunto ilegible.

# ASPECTOS DEL ANÁLISIS

**3.1. Herramientas tecnológicas**

El sistema VALIDADCION DOCUMENTAL fue desarrollado con nuevas tecnologías de programación web: CodeIgniter es un framework PHP para crear aplicaciones web utilizando arquitectura MVC. Permite a los desarrolladores realizar proyectos de manera muy rápida, con una interfaz simple y una lógica muy sencilla.

Trabajar con una estructura MVC facilita la organización del código y poder establecer una división entre el acceso a datos, la lógica de negocio y la capa de presentación de nuestra aplicación.

# 4. ESPECIFICACIONES TÉCNICAS

**Sistema Operativo**: Windows NT/98/Me/2000/2003/XP/Vista/7/8/10.

**Manejador de Base de Datos**: Postgresql 9.2.

**Lenguaje de Programación**: PHP 5.4, HTML5, CSS3, Framework (CodeIgniter -Bootstrap 2.8) y JavaScript.

**Servidor de Aplicaciones**: Servidor Web Apache 2.0, Servidor.

**Navegador Web**: Cualquier versión de Google Chrome (para trabajar en la aplicación) y funciona en todos los exploradores Web.

**Requerimientos mínimos de hardware**

* Procesador : Core
* Memoria RAM: Mínimo: 1 Gigabytes (GB)
* Disco Duro : 500Gb.

# 5. CONFIGURACIÓN DEL APLICATIVO

Se coloca la carpeta llamada VALIDACIONDOCUMENTAL en la ruta donde se encuentra instalado el servidor web local, para el caso de WAMP (64 BITS), sería C:\wamp64\www\VALIDACIONDOCUMENTAL.

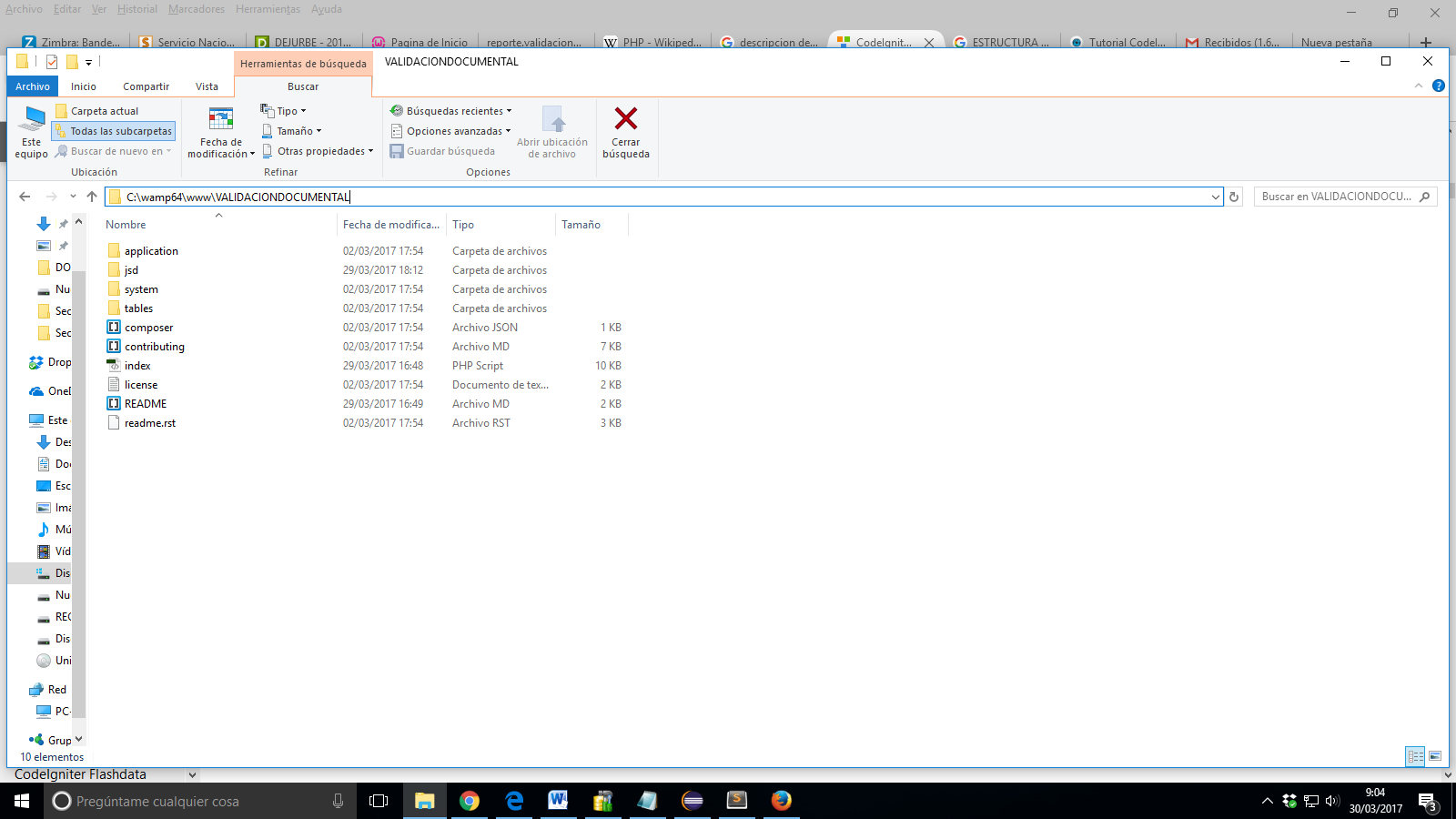


Figura No xxx - WapServer Directorio de la Aplicaciones Web

Luego se configuran los siguientes archivos, tal como se indica a continuación:

**5.1 Configurar archivo local**

Application/config/config.php

$config['base\_url'] = 'http://127.0.0.1:8080/VALIDACIONDOCUMENTAL/';

**5.2 Configurar la base de datos**

Application/config/database.php

$active\_group = 'default';

$query\_builder = **TRUE**;

$db['default'] = **array**(

'dsn' => '',

'hostname' => 'localhost',

'username' => 'postgres',

'password' => '\*\*\*\*\*\*\*\*\*\*\*',

'database' => 'dejurbe11',

'dbdriver' => 'postgre',

'dbprefix' => '',

'pconnect' => **FALSE**,

'db\_debug' => **TRUE**,

'cache\_on' => **FALSE**,

'cachedir' => '',

'char\_set' => 'utf8',

'dbcollat' => 'utf8\_general\_ci',

'swap\_pre' => '',

'encrypt' => **FALSE**,

'compress' => **FALSE**,

'stricton' => **FALSE**,

'failover' => **array**(),

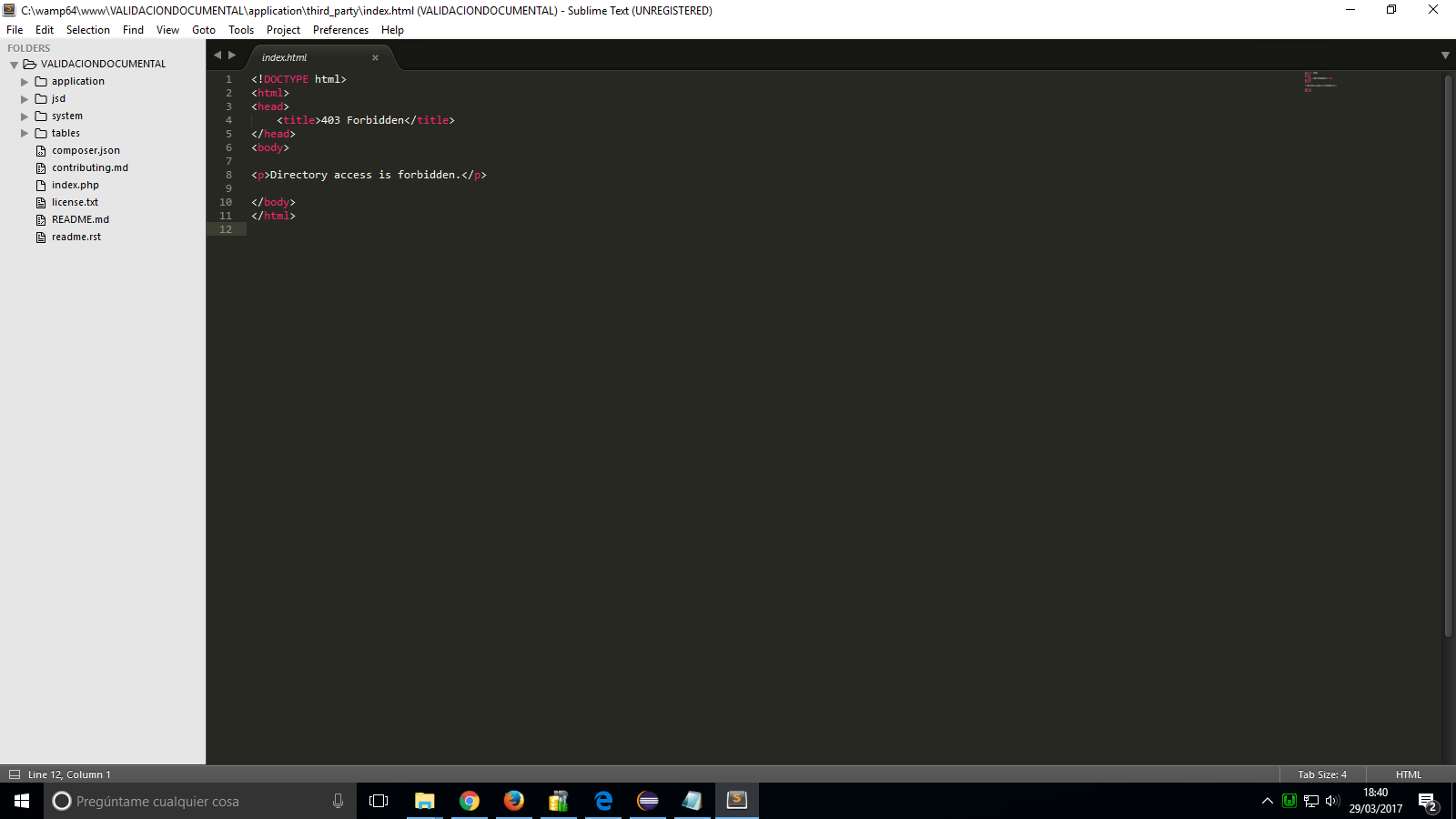
'save\_queries' => **TRUE**

);

# 6**. ESTRUCTURA DEL SITEMA**

El sistema validaciondocumental tiene la siguiente estructura:

|  |  |
| --- | --- |
| Carpeta | Descripción de las funciones |
| application | Se encuentran todos los archivos de nuestro proyecto |
| jsd | Contiene todos los js de la web. |
| system | Aquí se encuentran todas las librerías del Framework. No es nada recomendable modificarlos. |
| table | Se encuentran todos los recursos que necesitaremos para nuestra web : css, js, images y fonts |
| index.php | El index.php sirve como el controlador frontal, inicializando los recursos  básicos que necesita CodeIgniter para ejecutar. |



Dentro de la carpeta **aplication** tenemos las siguientes carpetas:

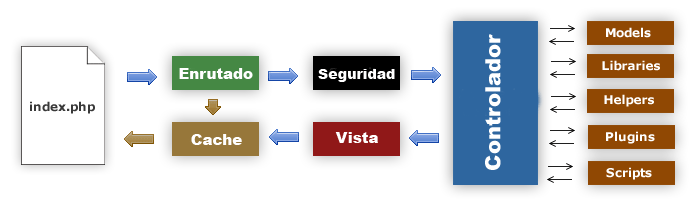


|  |  |
| --- | --- |
| Carpeta | Descripción de las funciones |
| cache | Esta carpeta contiene todas las páginas en caché de la aplicación. Estas páginas en caché aumentará la velocidad total de acceso a las páginas. |
| config | Esta carpeta contiene varios archivos para configurar la aplicación. Con la ayuda de archivo config.php, el usuario puede configurar la aplicación. Usando database.php archivo, el usuario puede configurar la base de datos de la aplicación. |
| controllers | Esta carpeta contiene los controladores de su aplicación. Es la parte fundamental de su aplicación. |
| core | Esta carpeta contendrá clase base de su solicitud. |
| helpers | En esta carpeta, se puede poner clase de ayuda de su aplicación**.** |
| hooks | En esta carpeta alteraríamos el comportamiento que tiene normalmente el framework sin tocar en los archivos los contenidos de la carpeta system. Nosotros lo usaremos para cambiar la seguridad y controlar la entrada de usuarios. |
| language | Esta carpeta contiene los archivos relacionados con el lenguaje. |
| libraries | La usaremos para la creación de funciones complicadas, que involucren varias tablas o que sencillamente no creamos conveniente alojarla en el modelo. La intención de esto es aislar la lógica de la aplicación con el acceso a la base de datos. |
| logs | Esta carpeta contiene los archivos relacionados con el registro del sistema. |
| models | El inicio de sesión de base de datos se encuentra en esta carpeta. |
| third\_party | Serian las librerías que nosotros nos descargamos para ampliar funcionalidades de PHP. Como podría ser leer PDF o generarlos, envió de correos… Se alojarían todos aquellos plugins creados por personas ajenas a nuestra aplicación. |
| views | Los archivos HTML de aplicación se instalan en esta carpeta. |

**Flujo de aplicación de CodeIgniter**

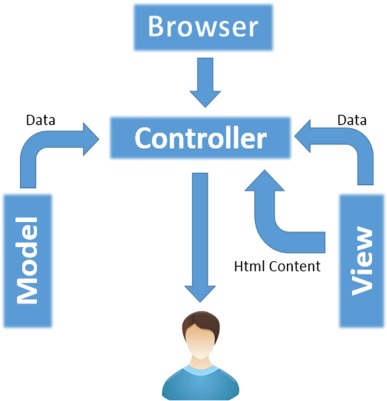
En CodeIgniter existe un procedimiento para atender una solicitud de página del cliente. Este proceso se realiza internamente por el propio CodeIgniter y de manera transparente para nosotros. Durante el proceso participan varios módulos como el enrutamiento de la solicitud, la caché interna, etc. Está bien conocerlo de antemano, aunque algunas de las cosas aun no las entendamos con lo que hemos visto hasta el momento en este manual.

Conviene prestar atención sobre la siguiente imagen, tomada de la cumentación de CodeIgniter, donde hemos traducido simple



**CodeIgniter - MVC Framework**

CodeIgniter se basa en el **patrón de desarrollo del Modelo-Vista-Controlador (MVC).** MVC es un enfoque de software que separa la lógica de aplicación de la presentación. En la práctica, permite que las páginas web que contienen secuencias de comandos mínima ya que la presentación está separado del scripting PHP.



* El **modelo representa las estructuras de datos.** Por lo general, las clases del modelo contendrán las funciones que le ayudan a recuperar, insertar y actualizar la información en su base de datos.
* La **vista es la información que se presenta a un usuario.** Una visión que normalmente será una página web, pero en CodeIgniter, una visión también puede ser un fragmento de la página como un encabezado o pie de página. También puede ser una página de RSS, o cualquier otro tipo de "página".
* El **controlador actúa como un intermediario entre el Modelo, la Vista,** y todos los demás recursos necesarios para procesar la petición HTTP y generar una página web.

A continuación mostramos un diagrama de las vistas, modelo, controladores y archivos Java Script, para un mejor concomiendo del Sistema de **VALIDACIONDOCUMENTAL**.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MODULO DE USUARIO** | | | | | | | | | | | |
| **VISTA** | **ARCHIVOS JAVASCRIPT** | | | **CONTROLADOR** | | | | **MODELO** | | | |
| index.php |  | | |  | | | |  | | | |
| **INGRESO AL SISTEMA DE VALIDACIÓN** | | | | | | | | | | | |
| logued.php |  | | | Usuarios.php | | | | Usuarios\_model.php | | | |
| **INICIO** | | | | | | | | | | | |
| cabecera.php | jquery.js | | | Inicio.php | | | | Entidades\_model.php  Usuarios\_model.php | | | |
| menu.php |
| pie.php |
| validadas.php |
| Validar.php |
| verdocumentos.php |
| verpersonas.php |
| sindocumento.php |
| **VALIDAR IMUEBLES** | | | | | | | | | | | |
| validados.php | jsd/inmuebles.js  jsd/principal.js | | | Inmuebles.php | | | | usuarios\_model  inmuebles\_model  entidades\_model  documentacion\_model | | | |
| validar.php |
| **VALIDAR VEHÍCULOS** | | | | | | | | | | | |
| validadosveh.php | jsd/vehiculos.js  jsd/principal.js | | | Vehiculos.php | | | | usuarios\_model  inmuebles\_model  vehiculos\_model  documentacion\_model | | | |
| validarveh.php |
| **VALIDAR MAQUINARIA** | | | | | | | | | | | |
| validadosmaq.php | jsd/maquinaria.js  jsd/principal.js | | | | Maquinaria.php | | | | usuarios\_model  inmuebles\_model  maquinaria\_model documentacion\_model | | |
| validarmaq.php |
| **VALIDAR MAQUINARIA PESADA** | | | | | | | | | | | |
| maquinariapes.php | jsd/maquinariapesada.js  jsd/principal.js | | | | | Maquinariapesada.php | | | | usuarios\_model  maquinariapesada\_model  documentacion\_model | |
| maquinariatab.php |
| **VALIDAR INMUEBLES ALQUILER** | | | | | | | | | | | |
| inmueblesalquilertab.php | | | jsd/inmueblesalquiler.js  jsd/principal.js | | | | Inmueblesalquiler.php | | | | usuarios\_model  inmueblesalquiler\_model  documentacion\_model |
| inmueblesalquilertab2.php | | |
| **VALIDAR VEHÍCULOS ALQUILER** | | | | | | | | | | | |
| vehiculosalquilertab.php | | jsd/vehiculosalquiler.js | | | | | Vehiculosalquiler.php | | | | usuarios\_model  vehiculosalquiler\_model  documentacion\_model |
| vehiculosalquilertab2.php | |
| **REPORTES** | | | | | | | | | | | |
| reportes.php | | reportes.js  jsd/principal.js | | | | | Reportes.php | | | | usuarios\_model  reportes\_model  entidades\_model |
| reportes\_pdf.php | |

# MODELO ENTIDAD RELACIÓN

# DICCIONARIO DE DATOS

**Id** denominado comoIdentificador: Número secuencial exclusivo (que se incrementa en una unidad) o número aleatorio que Enterprise Transaction Performance asigna cada vez que se añade un nuevo registro a la tabla.

**Fecha**: Correspondiente a la toma de los datos en campo de acuerdo al calendario Gregoriano Dominio de valores: Año – Mes - Día (aaaammdd)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.bien** | | |
| **Descripción** | | Contiene los registros de los bienes que fueron migrados | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial del **bien** |
| identidad | integer | fk |  | Identificador exclusivo obtenido del esquema gobierno tabla **entidades** |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idtipobien | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_activos tabla **tipobien** |
| idestadovalidacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **estado\_validacion** |
| idestadodocumentacion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento de la tabla **estado\_documentacion** |
| idbien | integer |  |  | Identificador exclusivo referenciado del esquema dj\_activos tabla **bien** |
| idsituacion | integer |  |  | Identificador exclusivo referenciado del esquema dj\_activos tabla **situacionbien** |
| idestadobien | integer |  |  | Identificador exclusivo referenciado del esquema dj\_activos tabla **estadobien** |
| iduso | integer |  |  | Identificador exclusivo referenciado del esquema dj\_activos tabla **uso** |
| idclase | integer |  |  | Identificador exclusivo referenciado del esquema dj\_activos tabla **clase** |
| iddepartamento | integer |  |  | Identificador exclusivo referenciado del esquema geografia tabla **departamento** |
| codigoactivo | text |  |  | Código asignado a cada bien |
| departamento | text |  |  | Departamento en la que se registró el bien |
| fechavalidacion | date |  |  | Fecha de la validación |
| idmotivobaja | integer |  |  | Identificador exclusivo obtenido del esquema dj\_activos tabla **motivo** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.bienalquiler** | | |
| **Descripción** | | Contiene los registros de los bienes en alquiler que fueron migrados | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial del **bienalquiler** |
| identidad | Integer | fk |  | Identificador exclusivo obtenido del esquema gobierno tabla **entidades** |
| idgestion | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idtipobien | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_activos tabla **tipobien** |
| idestadovalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **estado\_validacion** |
| idestadodocumenta-cion | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento de la tabla **estado\_documentacion** |
| idbien | Integer |  |  | Identificador exclusivo refenciado del esquema dj\_activos tabla **bienalquiler** |
| idclase | Integer |  |  | Identificador exclusivo referenciado del esquema dj\_activos tabla **clase** |
| iduso | Integer |  |  | Identificador exclusivo obtenido del esquema dj\_activos tabla **uso** |
| fechavalidacionbien | Date |  |  | Fecha de la validación del bien |
| iddepartamento | Integer |  |  | Identificador exclusivo obtenido del esquema geografia tabla **departamento** |
| departamento | Text |  |  | Nombre del departamento registrado del bien en alquiler |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.bitacora** | | |
| **Descripción** | | Contiene los registros de las acciones de los usuarios del sistema | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo secuencial de la bitácora |
| tabla | Text |  |  |  |
| operación | Text |  |  |  |
| datos | Text |  |  |  |
| idusuario | Integer |  |  | Identificador del usuario |
| fecha | Timestamp |  |  | Fecha de la bitácora |
| idregistro | Integer |  |  | Identificador de registro |
| sql | Text |  |  |  |
| ip | Text |  |  |  |
| cliente | Text |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.correspondencia** | | |
| **Descripción** | | Contiene registros de correspondencia | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo de la **correspondencia** |
| descripcion | text |  |  | Descripción de la correspondencia |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.datoadicional** | | |
| **Descripción** | | Contiene registros de las **Persona y/o Entidad propietaria**  cuyo documento de bien está en los documentos presentados en los rubros inmueble, vehículos, maquinaria y maquinaria pesada. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo secuencial de la tabla **datoadicional** |
| descripcion | Text |  |  | Nombre de las **Persona y/o Entidad propietaria** de los documento de bien esta en los documentos presentados |
| idtipodatoadicional | Integer | fk |  | Identificador exclusivo obtenido de la tabla **tipodatoadicional** |
| idbien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| idtipodocumento | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento tabla **documento** |
| eliminado | Boolean |  |  | Campo para almacenar los datos eliminados de |
| idfuncionario | Integer | references |  | Identificador exclusivo obtenido de la tabla **usuario** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.datoadicionalalquiler** | | |
| **Descripción** | | Contiene registros de las **Persona y/o Entidad propietaria** cuyo documento de bien está en los documentos presentados en los rubros inmueble alquiler y vehículos alquiler. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusiva secuencial de la tabla **datoadicionalalquiler** |
| descripcion | Text |  |  | Nombre de las **Persona y/o Entidad propietaria** de los documento de bien esta en los documentos presentados |
| idtipodatoadicional | Integer | fk |  | Identificador exclusivo obtenido de la tabla **tipodatoadicional** |
| idbien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **bienalquiler** |
| idtipodocumento | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento tabla **documento** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.detallevalidacioninmueble** | | |
| **Descripción** | | Contiene registros de la validación de la documentación detallada del inmueble | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial de la tabla **detallevalidacioninmueble** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestioninmueble** |
| nrodocumento | Text |  |  | Número del documento observado |
| correctodocumento | Boolean |  |  | Campo que almacena si el documento es correcto true y si no es false |
| superficieterreno | Text |  |  | Superficie del terreno Observado |
| correctosupterreno | Boolean |  |  | Campo que almacena si la superficie del terreno es correcto |
| catastro | Text |  |  | Censo estadístico de los bienes inmuebles de una determinada población |
| correctocatastro | boolean |  |  | Campo que almacena si dato del campo catastro es correcto al dato Observado |
| denominación | Text |  |  | Denominación del inmueble observado |
| coorectodenominacion | boolean |  |  | Campo que almacena la denominación del inmueble correcto |
| zona | Text |  |  | Zona del inmueble |
| correctozona | boolean |  |  | Campo que almacena si la Zona del inmuebles es correcto |
| dirección | Text |  |  | Dirección del inmueble |
| correctodireccion | boolean |  |  | Campo que almacena si la dirección del inmuebles es correcto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.detallevalidacioninmueblealquiler** | | |
| **Descripción** | | Contiene registros de la validación de la documentación detallada del inmueble alquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo de la tabla **detallevalidacioninmueblealquiler** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestioninmueblealquiler** |
| nrodocumento | Text |  |  | Número del documento observado |
| correctodocumento | boolean |  |  | Campo que almacena si el documento es correcto true y si no es false |
| departamento | Text |  |  | Nombre del departamento registrado el inmueble en alquiler |
| correctodepartamento | boolean |  |  | Campo que almacena si el departamento es correcto true y si no es false |
| zona | Text |  |  | Zona del inmueble |
| correctozona | boolean |  |  | Campo que almacena si la Zona del inmuebles es correcto |
| direccion | Text |  |  | Dirección del inmueble |
| correctodireccion | boolean |  |  | Campo que identifica si la dirección del inmuebles es correcto |
| iniciocontrato | Text |  |  | Fecha de inicio de contrato |
| correctoiniciocontrato | boolean |  |  | Campo donde se almacena si el dato inicio de contrato es correcto o incorrecto |
| fincontrato | Text |  |  | Fecha de finalización del contrato |
| correctofincontrato | boolean |  |  | Campo donde se almacena si el dato de finalización del contrato es correcto o incorrecto |
| canonalquiler | Text |  |  | Total del pago del alquiler definido |
| correctocanonalquiler | boolean |  |  | Campo donde se almacena si el dato de **canonalquiler** es correcto o incorrecto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.detallemaquinaria** | | |
| **Descripción** | | Contiene registros de la validación de la documentación detallada de la maquinaria | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial de la tabla **detallemaquinaria** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionmaquinaria** |
| nrodocumento | Text |  |  | Número del documento observado |
| correctodocumento | boolean |  |  | Campo que almacena si el documento es correcto true y si no es false |
| descripcion | Text |  |  | Descripción del detalle maquinaria |
| correctodescripcion | boolean |  |  | Campo que almacena si la descripción es correcto o es incorrecto |
| marca | Text |  |  | Marca de la maquinaria |
| correctomarca | boolean |  |  | Campo que almacena si la marca de la maquinaria es correcto |
| modelo | Text |  |  | Modelo de la maquinaria |
| correctomodelo | boolean |  |  | Campo que almacena si el modelo de la maquinaria es correcto |
| nroserie | Text |  |  | Número de serie de la maquinaria |
| correctonroserie | boolean |  |  | Campo que almacena si el número de serie es correcto o no |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.detallemaquinariapesada** | | |
| **Descripción** | | Contiene registros de la validación de la documentación detallada de la maquinaria pesada | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo secuencial de la tabla **detallemaquinariapesada** |
| idvalidacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionmaquinariapesada** |
| nrodocumento | Text |  |  | Número del documento observado |
| correctodocumento | boolean |  |  | Campo que almacena si el número documento es correcto true y si no es false |
| descripción | Text |  |  | Descripción del detalle maquinaria |
| correctodescripcion | boolean |  |  | Campo que almacena si la descripción es correcto o es incorrecto |
| marca | Text |  |  | Marca de la maquinaria |
| correctomarca | boolean |  |  | Campo que almacena si la marca de la maquinaria es correcto |
| modelo | Text |  |  | Modelo de la maquinaria |
| correctomodelo | boolean |  |  | Campo que almacena si el modelo de la maquinaria es correcto |
| nrochasis | Text |  |  | Número de serie de la maquinaria |
| correctonrochasis | boolean |  |  | Campo que almacena si el número de serie es correcto o incorrecto |
| nromotor | Text |  |  | Número del motor de la maquina pesada |
| correctonrmotor | boolean |  |  | Campo que almacena si el número del motor es correcto |
| color | Text |  |  | Color de la maquinaria pesada |
| correctocolor | boolean |  |  | Campo que almacena si el color de la maquinaria pesada es correcto o incorrecto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.detallevalidacionvehiculo** | | |
| **Descripción** | | Contiene registros de la validación de la documentación detallada de los vehículos | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo secuencial de la tabla **detallevalidacionvehiculo** |
| idvalidacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionvehiculo** |
| nrodocumento | Text |  |  | Número del documento observado |
| correctodocumento | boolean |  |  | Campo que almacena si el número documento es correcto true y si no es false |
| marca | Text |  |  | Marca del vehículo |
| correctomarca | boolean |  |  | Campo que almacena si la marca del vehículo es correcto o incorrecto |
| idtipovehiculo | Text |  |  | Descripción del tipo de vehículo obtenida del esquema dj\_activos tabla **tipobien** |
| correctovehiculo | boolean |  |  | Campo que almacena si la descripción del vehículo registrado es correcto o incorrecto |
| clase | Text |  |  | Clase de vehículos |
| correctoclase | boolean |  |  | Campo que almacenado si el tipo de vehículo registrado es correcto o incorrecto |
| placa | Text |  |  | Número de la placa del vehículo |
| correctoplaca | boolean |  |  | Campo que almacenado si el número de placa del vehículo registrado es correcto o incorrecto |
| nromotor | Text |  |  | Número del motor del vehículo |
| correctomotor | boolean |  |  | Campo que almacenado si el número del motor del vehículo registrado es correcto o incorrecto |
| nrochasis | Text |  |  | Número de chasis del vehículo |
| correctochasis | boolean |  |  | Campo que almacena si el número de chasis registrado es correcto o incorrecto |
| procedencia | Text |  |  | Lugar de procedencia del vehículo |
| correctoprocedencia | boolean |  |  | Campo que almacena si lugar de procedencia es correcto o es incorrecto |
| modelo | Text |  |  | Nombre del modelo del vehículo |
| correctomodelo | boolean |  |  | Campo que almacena si el nombre del modelo es correcto o incorrecto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.detallevalidacionvehiculoalquiler** | | |
| **Descripción** | | Contiene registros de la validación de la documentación detallada de los vehículos en alquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo de la tabla **detallevalidacionvehiculoalquiler** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionvehiculoalquiler** |
| nrodocumento | Text |  |  | Número del documento observado |
| correctodocuemento | boolean |  |  | Campo que almacena si el documento es correcto true y si no es false |
| departamento | Text |  |  | Nombre del departamento registrado del vehículo en alquiler |
| correctodepartamento | boolean |  |  | Campo que almacena si el departamento es correcto true y si no es false |
| zona | Text |  |  | Zona donde se encuentra el vehículo |
| correctozona | boolean |  |  | Campo que almacena si la Zona del vehículo es correcto |
| dirección | Text |  |  | Dirección del inmueble |
| correctodireccion | boolean |  |  | Campo que identifica si la dirección del inmuebles es correcto |
| iniciocontrato | Text |  |  | Fecha de inicio de contrato |
| correctoiniciocontrato | boolean |  |  | Campo donde se almacena si el dato inicio de contrato es correcto o incorrecto |
| fincontrato | Text |  |  | Fecha de finalización del contrato |
| correctofincontrato | boolean |  |  | Campo donde se almacena si el dato de finalización del contrato es correcto o incorrecto |
| canonalquiler | Text |  |  | Total del pago del alquiler definido |
| correctocanonalquiler | boolean |  |  | Campo donde se almacena si el dato de **canonalquiler** es correcto o incorrecto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.documentobien** | | |
| **Descripción** | | Contiene registros de la documentación de los bienes que fueron migrados. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo secuencial de la tabla **documentobien** |
| idbien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| idtipodocumento | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento tabla **documento** |
| nrodocumento | Text |  |  | Número del documento observado |
| adicionado | Boolean |  |  | Campo que almacena si el documento se adiciono |
| idfuncionario | Integer |  |  | Identificador exclusivo obtenido de la tabla **usuario** |
| fechaadicion | Date |  |  | Fecha de adición del documento |
| gestión | Integer |  |  | El año de la documentación del bien |
| validado | Boolean |  |  | Campo que almacena si el documento esta validado o no |
| antiguo | Boolean |  |  | Campo que almacena si el documento es antiguo o no |
| eliminado | Boolean |  |  | Campo que almacena si el documento esta eliminado o no |
| fechavalidaciondoc | date |  |  | Fecha de la validación del documento |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.documentobienalquiler** | | |
| **Descripción** | | Contiene registros de la documentación de los bienes en alquiler que fueron migrados. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Integer | pk |  | Identificador exclusivo secuencial de la tabla **documentobienalquiler** |
| idbien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **bienalquiler** |
| idtipodocumento | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento tabla **documento** |
| nrodocumento | Text |  |  | Número del documento observado |
| adicionado | Boolean |  |  | Campo que almacena si el documento se adiciono |
| idfuncionario | Integer |  |  | Identificador exclusivo referenciado de la tabla **usuario** referenciando al campo **idfuncionario** |
| fechaadicion | Date |  |  | Fecha de adición del documento |
| gestión | Integer |  |  | El año de la documentación del bien |
| validado | Boolean |  |  | Campo que almacena si el documento esta validado o no |
| eliminado | Boolean |  |  | Campo que almacena si el documento del bien en alquiler fue eliminado o no |
| fechavalidaciondoc | date |  |  | Fecha de la validación del documento |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.documentoseliminados** | | |
| **Descripción** | | Contiene registros de la documentación de los bienes que fueron eliminados. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial de la tabla **documentoseliminados** |
| iddocumento | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_documento tabla **documento** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| código\_digitalizado | carácter varying |  | 250 | Número asignado al documento eliminado |
| nro\_hojas | integer |  |  | Cantidad de hojas del documento eliminado |
| digitalizado | boolean |  |  | Campo que almacena si el documento esta digitalizado o no |
| nro\_documento | text |  |  | Número del documento |
| usuario | integer |  |  | Usuario que elimina del documento |
| usuario\_dj | integer |  |  |  |
| fecha\_inscripcion | timestamp |  |  | Fecha de inscripción |
| observaciones | text |  |  | Observaciones de la documentación a eliminar |
| gestión | integer |  |  | Gestión en que se eliminó el documento |
| validado | boolean |  |  | Campo que almacena si la validación fue validada o no |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.estado\_validacion** | | |
| **Descripción** | | Contiene registros de estado de la validación de los bienes | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial de la tabla **estado\_validacion** |
| descripcion | text |  |  | Estado en al que se encuentra el documento validado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.inmueble** | | |
| **Descripción** | | Contiene registros de los bienes que fueron migrados del rubro inmueble | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | Pk |  | Identificador exclusivo secuencial de la tabla **inmueble** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| superficieterreno | numeric |  | (19,0) | Superficie del terreno en metro cuadrado |
| superficiecontruida | numeric |  | (18,0) | Superficie del inmueble construido |
| zona | text |  |  | Zona del inmueble |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| dirección | text |  |  | Dirección del inmueble |
| denominación | text |  |  | Denominación del inmueble observado |
| nrocatastro | text |  |  | Número registro administrativo dependiente del Estado en el que se describen los bienes inmuebles |
| padre | boolean |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.inmueblealquiler** | | |
| **Descripción** | | Contiene registros de los bienesalquiler que fueron migrados del rubro inmueblealquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial de la tabla **inmueblealquiler** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bienalquiler** |
| denominación | text |  |  | Denominación del inmueble en alquiler observado |
| dirección | text |  |  | Dirección del inmueble en alquiler |
| canonalquiler | numeric (18,0) |  |  | Total del pago del alquiler definido |
| fechainicio | date |  |  | Fecha inicio del contrato de alquiler |
| fechafin | date |  |  | Fecha finalización del contrato de alquiler |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idciudad | integer |  |  | Identificador exclusivo referenciado del esquema geografía tabla departamento |
| zona | text |  |  | Nombre de la zona del inmueble en alquiler |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.maquinaria** | | |
| **Descripción** | | Contiene registros de los bienes que fueron migrados del rubro maquinaria | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | integer | pk |  | Identificador exclusivo secuencial de |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| descripción | text |  |  | Descripción de la maquinaria |
| marca | text |  |  | Nombre de la marca de la maquinaria |
| modelo | text |  |  | Nombre del modelo de la maquinaria |
| nroserie | text |  |  | Número de serie de la maquinaria |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.maquinariapesada** | | |
| **Descripción** | | Contiene registros de los bienes que fueron migrados del rubro maquinaria pesada | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial |  |  | Identificador exclusiva secuencial de la tabla **maquinariapesada** |
| idbien | integer |  |  | Identificador exclusivo obtenido de la tabla **bien** |
| descripción | text |  |  | Nombre de la Maquinaria pesada |
| marca | text |  |  | Marca de la maquinaria pesada |
| modelo | text |  |  | Modelo de la maquinaria pesada |
| nrochasis | text |  |  | Número de chasis de la maquinaria pesada |
| nromotor | text |  |  | Número del motor de la maquinaria pesada |
| color | text |  |  | Color de la maquinaria pesada |
| idgestion | integer |  |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.observacioninmueble** | | |
| **Descripción** | | Contiene registro de las observaciones que se registran de la documentación validada del rubro inmueble | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **observacioninmueble** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestioninmueble** |
| idtipoobservacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **tipoobservacion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.observacioninmueblealquiler** | | |
| **Descripción** | | Contiene registro de las observaciones que se registran de la documentación validada del rubro inmueblealquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| Id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **observacioninmueblealquiler** |
| Idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestioninmueblealquiler** |
| Idtipoobservacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **tipoobservacion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.observacionmaquinaria** | | |
| **Descripción** | | Contiene registro de las observaciones que se registran de la documentación validada del rubro maquinaria | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **observacionmaquinaria** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionmaquinaria** |
| idtipoobservacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **tipoobservacion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.observacionmaquinariapesada** | | |
| **Descripción** | | Contiene registro de las observaciones que se registran de la documentación validada del rubro maquinariapesada | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **observacionmaquinariapesada** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionmaquinariapesasda** |
| idtipoobservacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **tipoobservacion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.observacionvehiculos** | | |
| **Descripción** | | Contiene registro de las observaciones que se registran de la documentación validada del rubro vehículos | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **observacionvehiculos** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionvehiculos** |
| idtipoobservacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **tipoobservacion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.observacionvehiculoalquiler** | | |
| **Descripción** | | Contiene registro de las observaciones que se registran de la documentación validada del rubro vehículo alquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **observaciovehiculoalquiler** |
| idvalidacion | Integer | fk |  | Identificador exclusivo obtenido de la tabla **validacionxgestionvehiculoalquiler** |
| idtipoobservacion | integer | fk |  | Identificador exclusivo obtenido de la tabla **tipoobservacion** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.tipodatoadicional** | | |
| **Descripción** | | Contiene registros del tipo de adición de la documentación. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | Serial | pk |  | Identificador exclusivo secuencial de la tabla **tipodatoadicional** |
| descripcion | Text |  |  | Descripción del tipo de dato adicional |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **Nueva\_validacion.tipoobservacion** | | |
| **Descripción** | | Contiene registro del tipo de observación que se realiza en la validación de la documentación por rubros | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **tipoobservacion** |
| descripción | text |  |  | Descripción de tipo de observación |
| Idrubro | integer | fk |  | Dj activos clase |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.tipopropietarioadicional** | | |
| **Descripción** | | Contiene registro de los tipo de propietarios de la documentación | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **tipopropietarioadicional** |
| descripción | text |  |  | Descripción del tipo propietario adicional de la documentación |
| activo | boolean |  |  | Campo que almacena el tipo propietario adicional está activo o no |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.user\_universo** | | |
| **Descripción** | | Contiene registro de las entidades asignadas a los validadores. | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **user\_universo** |
| idusuario | integer | fk |  | Identificador exclusivo del tabla **usuario**  y se hace una referencia del campo **idfuncionario** |
| identidad | integer | fk |  | Identificador exclusivo del esquema gobierno tabla **entidades** |
| fecha\_asignacion | timestamp |  |  | Fecha de asignación |
| fecha\_desasignacion | timestamp |  |  | Fecha en la que se elimina la asignación |
| gestion | integer |  |  | Año de creación de usuario universal |
| asignado | boolean |  |  | Campo que almacena si fue asignado permisos o no |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.usuario** | | |
| Descripción | | Contiene registros de los usuarios que pueden acceder al sistema | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusiva de la tabla **usuario** |
| usuario | varchar |  | 255 | usuario para ingresar al sistema |
| password | varchar |  | 255 | Contraseña del usuario para acceder al sistema |
| email | varchar |  | 255 | Correo electrónico del usuario |
| nombre | varchar |  | 255 | Nombre completo del usuario |
| corto | varchar |  | 255 | Nombre abreviado del usuario |
| sigla | varchar |  | 255 | Abreviatura del usuario |
| id\_funcionario | integer | references | 255 | Identificador exclusivo obtenido de la tabla **usuario** referenciando al campo **idfuncionario** |
| activo | boolean |  |  | Campo que almacena el tipo propietario adicional está activo o no |
| administrador | boolean |  |  | Campo que almacena si el usuario es asignado como administrador o no |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.validacionxgestioninmueble** | | |
| **Descripción** | | Contiene registros de la verificación, validación y documentación del rubro inmuebles | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **validacionxgestioninmueble** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idcorrespondiencia | integer | fk |  | Identificador exclusivo obtenido de la tabla **correspondencia** |
| adjunta | boolean |  |  | Campo que almacena si el documento a validar tiene documentación adjunta o no |
| legible | boolean |  |  | Campo que almacena si el documento a validar es legible o no |
| observacionesgeneral | text |  |  | Observación de la validación por gestión del rubro inmueble |
| iddocumentobien | integer | fk |  | Identificador exclusivo obtenido de la tabla **documentobien** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.validacionxgestioninmueblealquiler** | | |
| **Descripción** | | Contiene registros de la verificación de la validación de la documentación del rubro inmuebles alquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **validacionxgestioninmueblealquiler** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bienalquiler** |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idcorrespondencia | integer | fk |  | Identificador exclusivo obtenido de la tabla **correspondencia** |
| adjunta | boolean |  |  | Campo que almacena si el documento a validar tiene documentación adjunta o no |
| legible | boolean |  |  | Campo que almacena si el documento a validar es legible o no |
| observacionesgenerales | text |  |  | Observación de la validación por gestión del rubro inmueble alquiler |
| iddocumentobien | integer | fk |  | Identificador exclusivo obtenido de la tabla **documentobienalquiler** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.validacionxgestionmaquinaria** | | |
| **Descripción** | | Contiene registros de la verificación de la validación de la documentación del rubro maquinaria | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| Id | Serial | pk |  | Identificador exclusivo de la tabla **validacionxgestionmaquinaria** |
| Idbien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| Idgestion | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| Idcorrespondencia | Integer | fk |  | Identificador exclusivo obtenido de la tabla **correspondencia** |
| Adjunta | Boolean |  |  | Campo que almacena si el documento a validar tiene documentación adjunta o no |
| Legible | Boolean |  |  | Campo que almacena si el documento a validar es legible o no |
| Observacionesgenerales | text |  |  | Observación de la validación de la documentación del rubro maquinaria |
| iddocumentobien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **documentobien** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.validacionxgestionmaquinariapesada** | | |
| **Descripción** | | Contiene registros de la verificación de la validación de la documentación del rubro maquinaria pesada | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| Id | Serial | pk |  | Identificador exclusivo de la tabla **validacionxgestionmaquinariapesada** |
| Idbien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| Idgestion | Integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| Idcorrespondencia | Integer | fk |  | Identificador exclusivo obtenido de la tabla **correspondencia** |
| Adjunta | Boolean |  |  | Campo que almacena si el documento a validar tiene documentación adjunta o no |
| Legible | Boolean |  |  | Campo que almacena si el documento a validar es legible o no |
| Observacionesgenerales | text |  |  | Observación de la validación de la documentación del rubro maquinaria pesada |
| iddocumentobien | Integer | fk |  | Identificador exclusivo obtenido de la tabla **documentobien** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.validacionxgestionvehiculo** | | |
| **Descripción** | | Contiene registros de la verificación de la validación de la documentación del rubro vehículo | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **validacionxgestionvehiculo** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idcorrespondencia | integer | fk |  | Identificador exclusivo obtenido de la tabla **correspondencia** |
| adjunta | boolean |  |  | Campo que almacena si el documento a validar tiene documentación adjunta o no |
| legible | boolean |  |  | Campo que almacena si el documento a validar es legible o no |
| observacionesgenerales | text |  |  | Observación de la validación de la documentación del rubro vehículo |
| iddocumentobien | integer | fk |  | Identificador exclusivo obtenido de la tabla **documentobien** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.validacionxgestionvehiculoalquiler** | | |
| **Descripción** | | Contiene resgistros de la verificación de la validación de la documentación del rubro vehiculo alquiler | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo secuencial de la tabla **validacionxgestionvehiculo** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idcorrespondencia | integer | fk |  | Identificador exclusivo obtenido de la tabla **correspondencia** |
| adjunta | boolean |  |  | Campo que almacena si el documento a validar tiene documentación adjunta o no |
| legible | boolean |  |  | Campo que almacena si el documento a validar es legible o no |
| observacionesgenerales | text |  |  | Observación de la validación de la documentación del rubro vehículo en alquiler |
| iddocumentobien | integer | fk |  | Identificador exclusivo obtenido de la tabla **documentobienalquiler** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.vehiculo** | | |
| **Descripción** | | Contiene registros de los bienes del rubro vehículos que fueron migrados | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | identificador exclusivo secuencial de la tabla **vehiculo** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bien** |
| idtipovehiculo | integer | fk |  | Descripción del tipo de vehículo obtenida del esquema dj\_activos tabla **tipobien** |
| clase | text |  |  | Clase de vehículo |
| marca | text |  |  | Nombre de la marca del vehículo |
| placa | text |  |  | Nombre de la placa del vehículo |
| nromotor | text |  |  | Número del motor del vehículo |
| nrochasis | text |  |  | Número de chasis del vehículo |
| modelo | text |  |  | Nombre del modelo del vehículo |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| idprocedencia | varchar | fk | 250 | Lugar de procedencia del vehículo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre de la tabla** | | **nueva\_validacion.vehiculoalquiler** | | |
| **Descripción** | | Contiene registros de los bienes del rubro vehículos en alquiler que fueron migrados | | |
| **Nombre del campo** | **Tipo de dato** | **Integridad** | **Longitud** | **Descripción** |
| id | serial | pk |  | Identificador exclusivo de la tabla **vehiculoalquiler** |
| idbien | integer | fk |  | Identificador exclusivo obtenido de la tabla **bienalquiler** |
| marca | text |  |  | Nombre de la marca del vehículo en alquiler |
| modelo | text |  |  | Nombre del modelo del vehículo en alquiler |
| placa | text |  |  | Número de placa del vehículo en alquiler |
| nrochasis | text |  |  | Número de Chasis del vehículo en alquiler |
| color | text |  |  | Color del vehículo en alquiler |
| canonalquiler | integer |  |  | Total del pago del alquiler definido |
| idgestion | integer | fk |  | Identificador exclusivo obtenido del esquema dj\_dejurbe tabla **gestion** |
| fechainicio | date |  |  | Fecha de inicio del contrato de aqluiler |
| fechafin | date |  |  | Fecha fin del contrato de alquiler |
| idciudad | integer |  |  | Identificador exclusivo referenciado del esquema geografía tabla departamento |
| zona | text |  |  | Zona de la entidad que alquila el vehículo |
| dirección | text |  |  | Dirección de la entidad que alquila el vehículo |
| nromotor | text |  |  | Número del motor del vehículo dado en aqluiler |

# VISTAS

|  |  |
| --- | --- |
| **Vista: bienes\_cantdocumentos** | |
| **Nombre** | bienes\_cantdocumentos |
| **Descripción** | Muestra el número de documentos de cada bien utilizando el identificador del bien |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.bienes\_cantdocumentos  AS  SELECT documentobien.idbien  , count(documentobien.id) AS nrodocumentos  FROM nueva\_validacion.documentobien  WHERE (documentobien.eliminado = false)  GROUP BY documentobien.idbien |

|  |  |
| --- | --- |
| **Vista: bienesalquiler\_cantdocumentos** | |
| **Nombre** | bienesalquiler\_cantdocumentos |
| **Descripción** | Muestra el número de documentos de cada bien en alquiler utilizando el identificador del bien |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.bienesalquiler\_cantdocumentos  AS  SELECT documentobienalquiler.idbien  , count(documentobienalquiler.id) AS nrodocumentos  FROM nueva\_validacion.documentobienalquiler  GROUP BY documentobienalquiler.idbien |

|  |  |
| --- | --- |
| **Vista: bienesvalidados** | |
| **Nombre** | bienesvalidados |
| **Descripción** | Muestra el listado del identificador de la entidad y el total de los bienes validados |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.bienesvalidados  AS  SELECT a.identidad  , count(a.bienesvalidados) AS bienesvalidados  FROM (SELECT bien.identidad  , bien.idbien AS bienesvalidados  FROM nueva\_validacion.bien  WHERE (bien.idestadovalidacion = 3) UNION ALL  SELECT bienalquiler.identidad  , bienalquiler.idbien AS bienesvalidados  FROM nueva\_validacion.bienalquiler  WHERE (bienalquiler.idestadovalidacion = 3)) a GROUP BY a.identidad |

|  |  |
| --- | --- |
| **Vista: entidades\_bienes\_nrodocumentos** | |
| **Nombre** | entidades\_bienes\_nrodocumentos |
| **Descripción** | Muestra los datos del número de documentos en base a las entidades y al bien correspondiente |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.entidades\_bienes\_nrodocumentos  AS  SELECT vista\_nrodocumentosbien.idbien  , vista\_nrodocumentosbien.identidad  , vista\_nrodocumentosbien.nrodocumentos  FROM nueva\_validacion.vista\_nrodocumentosbien UNION ALL  SELECT vista\_nrodocumentosbienalquiler.idbien  , vista\_nrodocumentosbienalquiler.identidad  , vista\_nrodocumentosbienalquiler.nrodocumentos  FROM nueva\_validacion.vista\_nrodocumentosbienalquiler |

|  |  |
| --- | --- |
| **Vista: entidades\_cantidades\_validacion** | |
| **Nombre** | entidades \_cantidades\_validacion |
| **Descripción** | Muestra el total de bienes validados de cada entidad en base a su identificador exclusivo |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.entidades\_cantidades\_validacion  AS  SELECT a.identidad  , count(a.idbien) AS total  , b.validados  FROM (nueva\_validacion.bien a  LEFT JOIN (SELECT bien.identidad  , count(bien.idbien) AS validados  FROM nueva\_validacion.bien  WHERE (bien.idestadovalidacion = 3) GROUP BY bien.identidad) b  ON ((b.identidad = a.identidad)))  WHERE (a.identidad IS NOT NULL) GROUP BY a.identidad  , b.validados |

|  |  |
| --- | --- |
| **Vista: entidades\_nbienes\_nvalidados\_ndocumentos** | |
| **Nombre** | entidades\_nbienes \_nvalidados\_ndocumentos |
| **Descripción** | Muestra el total de bienes, bienes validados y el total de documentos de cada entidad con su respectivo identificador y el nombre de la entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.entidades\_nbienes\_nvalidados\_ndocumentos  AS  SELECT b.id  , b.nombre  , d.descripcion AS departamento  , c.totalbienes  , CASE WHEN (a.bienesvalidados IS NULL) THEN (0)::bigint ELSE a.bienesvalidados END AS bienesvalidados  , c.totaldocumentos  FROM (((nueva\_validacion.totales\_bienes\_documentos c JOIN gobierno.entidades b  ON ((c.identidad = b.id))) JOIN geografia.departamento d  ON ((d.id = b.departamento)))  LEFT JOIN nueva\_validacion.bienesvalidados a  ON ((a.identidad = b.id)))  WHERE (b.id <> 27) |

|  |  |
| --- | --- |
| **Vista: entidadesavalidar** | |
| **Nombre** | **entidadesavalidar** |
| **Descripción** | Muestra el total de los bienes y el total de los bienes validados por entidad tomando en cuenta el identificador y el nombre de la entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.entidadesavalidar(  id, nombre, totalbienes, bienesvalidados)  AS  SELECT a.id, a.nombre, b.totalbienes,  CASE  WHEN b.totalvalidados IS NULL THEN 0::numeric  ELSE b.totalvalidados  END AS bienesvalidados  FROM gobierno.entidades a  JOIN nueva\_validacion.total\_validacion\_bienes b ON a.id = b.identidad; |

|  |  |
| --- | --- |
| **View: reportetrabajovalidacionia** | |
| **Nombre** | reportetrabajovalidacionia |
| **Descripción** | Muestra los bienes dado a contrato en alquiler y acuerdo/convenio por cada entidad, tomando en cuenta el identificador del bien y el identificador de la entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.reportetrabajovalidacionia  AS  SELECT crosstab.idbien  , crosstab.identidad  , crosstab."CONTRATO DE ALQUILER"  , crosstab."ACUERDO/CONVENIO"  FROM crosstab('select a.idbien, identidad, idtipodocumento, case when adicionado=false then 1 else -1 end as validadopor  FROM nueva\_validacion.documentobienalquiler a  join nueva\_validacion.bienalquiler b  ON b.idbien=a.idbien  where b.idclase=5  AND a.validado=true order by a.idbien'::text  , 'select iddocumento  FROM dj\_documento.documento\_clasebien  WHERE idclase=5 order by iddocumento'::text) crosstab(idbien integer  , identidad integer  , "CONTRATO DE ALQUILER" integer  , "ACUERDO/CONVENIO" integer) |

|  |  |
| --- | --- |
| **Vista: reportetrabajovalidacioninmuebles** | |
| **Nombre** | reportetrabajovalidacioninmuebles |
| **Descripción** | Muestra la validación del rubro inmuebles de tipo terreno y construcción |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.reportetrabajovalidacioninmuebles  AS  SELECT  crosstab.idbien  , crosstab.identidad  , crosstab."FOLIO REAL"  , crosstab."TARJETA DE PROPIEDAD"  , crosstab."TESTIMONIO DE PROPIEDAD"  , crosstab."LEY DEL ESTADO PLURINACIONAL"  , crosstab."DECRETO SUPREMO"  , crosstab."DOCUMENTO DE TRANSFERENCIA"  , crosstab."MINUTA DE TRANSFERENCIA DE COMPRA VENTA"  , crosstab."RESOLUCION MINISTERIAL"  , crosstab."ORDENANZA MUNICIPAL"  , crosstab."RESOLUCION MUNICIPAL"  , crosstab."RESOLUCION ADMINISTRATIVA"  , crosstab."CERTIFICADO CATASTRAL"  , crosstab."DOCUMENTO DE COMPRA VENTA"  , crosstab."PLANOS"  , crosstab."ACTA DE ENTREGA"  , crosstab."AVALUO TECNICO"  , crosstab."CONTRATO DE COMODATO"  , crosstab."INFORME LEGAL"  , crosstab."TITULO EJECUTORIAL"  , crosstab."INFORME TECNICO"  , crosstab."CERTIFICADO ALODIAL "  , crosstab."RESOLUCION MINISTERIAL  , ADMINISTRATIVA O MUNICIPAL (BAJA)"  FROM crosstab('select a.idbien, identidad, idtipodocumento, case when adicionado=false then 1 else -1 end as validadopor  FROM nueva\_validacion.documentobien a  join nueva\_validacion.bien b  ON b.idbien=a.idbien  where b.idclase in (1, 2)  AND a.validado=true order by idbien'::text  , 'select iddocumento  FROM dj\_documento.documento\_clasebien  WHERE idclase=1 order by iddocumento'::text) crosstab(idbien integer  , identidad integer  , "FOLIO REAL" integer  , "TARJETA DE PROPIEDAD" integer  , "TESTIMONIO DE PROPIEDAD" integer  , "LEY DEL ESTADO PLURINACIONAL" integer  , "DECRETO SUPREMO" integer  , "DOCUMENTO DE TRANSFERENCIA" integer  , "MINUTA DE TRANSFERENCIA DE COMPRA VENTA" integer  , "RESOLUCION MINISTERIAL" integer  , "ORDENANZA MUNICIPAL" integer  , "RESOLUCION MUNICIPAL" integer  , "RESOLUCION ADMINISTRATIVA" integer  , "CERTIFICADO CATASTRAL" integer  , "DOCUMENTO DE COMPRA VENTA" integer  , "PLANOS" integer  , "ACTA DE ENTREGA" integer  , "AVALUO TECNICO" integer  , "CONTRATO DE COMODATO" integer  , "INFORME LEGAL" integer  , "TITULO EJECUTORIAL" integer  , "INFORME TECNICO" integer  , "CERTIFICADO ALODIAL " integer  , "RESOLUCION MINISTERIAL  , ADMINISTRATIVA O MUNICIPAL (BAJA)" integer) |

|  |  |
| --- | --- |
| **Vista: reportetrabajovalidacionpesada** | |
| **Nombre** | reportetrabajovalidacionpesada |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.reportetrabajovalidacionpesada  AS  SELECT crosstab.idbien  , crosstab.identidad  , crosstab."CONTRATO DE COMODATO"  , crosstab."R.U.A.T."  , crosstab."P.T.A."  , crosstab."CARNET DE PROPIEDAD"  , crosstab."POLIZA DE IMPORTACION"  , crosstab."FACTURA"  , crosstab."LEY DEL ESTADO PLURINACIONAL"  , crosstab."DECRETO SUPREMO"  , crosstab."DOCUMENTO DE TRANSFERENCIA"  , crosstab."MINUTA DE TRANSFERENCIA DE COMPRA VENTA"  , crosstab."RESOLUCION MINISTERIAL"  , crosstab."ORDENANZA MUNICIPAL"  , crosstab."RESOLUCION ADMINISTRATIVA"  , crosstab."DOCUMENTO DE COMPRA VENTA"  , crosstab."ACTA DE ENTREGA"  , crosstab."AVALUO TECNICO"  , crosstab."INFORME LEGAL"  , crosstab."INFORME TECNICO"  , crosstab."RESOLUCION MINISTERIAL  , ADMINISTRATIVA O MUNICIPAL (BAJA)"  , crosstab."DOCUMENTO ÚNICO DE IMPORTACIÓN (DUI)"  , crosstab."TESTIMONIO DE PROPIEDAD"  FROM crosstab('select a.idbien, identidad, idtipodocumento, case when adicionado=false then 1 else -1 end as validadopor  FROM nueva\_validacion.documentobien a  join nueva\_validacion.bien b  ON b.idbien=a.idbien  where b.idclase=6  AND a.validado=true order by idbien'::text  , 'select iddocumento  FROM dj\_documento.documento\_clasebien  WHERE idclase=6  order by iddocumento'::text) crosstab(idbien integer  , identidad integer  , "CONTRATO DE COMODATO" integer  , "R.U.A.T." integer  , "P.T.A." integer  , "CARNET DE PROPIEDAD" integer  , "POLIZA DE IMPORTACION" integer  , "FACTURA" integer  , "LEY DEL ESTADO PLURINACIONAL" integer  , "DECRETO SUPREMO" integer  , "DOCUMENTO DE TRANSFERENCIA" integer  , "MINUTA DE TRANSFERENCIA DE COMPRA VENTA" integer  , "RESOLUCION MINISTERIAL" integer  , "ORDENANZA MUNICIPAL" integer  , "RESOLUCION ADMINISTRATIVA" integer  , "DOCUMENTO DE COMPRA VENTA" integer  , "ACTA DE ENTREGA" integer  , "AVALUO TECNICO" integer  , "INFORME LEGAL" integer  , "INFORME TECNICO" integer  , "RESOLUCION MINISTERIAL  , ADMINISTRATIVA O MUNICIPAL (BAJA)" integer  , "DOCUMENTO ÚNICO DE IMPORTACIÓN (DUI)" integer  , "TESTIMONIO DE PROPIEDAD" integer) |

|  |  |
| --- | --- |
| **Vista: reportetrabajovalidacionva** | |
| **Nombre** | reportetrabajovalidacionva |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.reportetrabajovalidacionva  AS  SELECT crosstab.idbien  , crosstab.identidad  , crosstab."CONTRATO DE ALQUILER"  , crosstab."ACUERDO/CONVENIO"  FROM crosstab('select a.idbien, identidad, idtipodocumento, case when adicionado=false then 1 else -1 end as validadopor  FROM nueva\_validacion.documentobienalquiler a  join nueva\_validacion.bienalquiler b  ON b.idbien=a.idbien  where b.idclase=7  AND a.validado=true order by a.idbien'::text  , 'select iddocumento  FROM dj\_documento.documento\_clasebien  WHERE idclase=7 order by iddocumento'::text) crosstab(idbien integer  , identidad integer  , "CONTRATO DE ALQUILER" integer  , "ACUERDO/CONVENIO" integer) |

|  |  |
| --- | --- |
| **Vista: reportetrabajovalidacionvehiculos** | |
| Name | reportetrabajovalidacionvehiculos |
| Comment |  |
| Code | CREATE OR REPLACE VIEW nueva\_validacion.reportetrabajovalidacionvehiculos  AS  SELECT crosstab.idbien  , crosstab.identidad  , crosstab."R.U.A.T."  , crosstab."P.T.A."  , crosstab."CARNET DE PROPIEDAD"  , crosstab."POLIZA DE IMPORTACION"  , crosstab."FACTURA"  , crosstab."DOCUMENTO DE TRANSFERENCIA"  , crosstab."MINUTA DE TRANSFERENCIA DE COMPRA VENTA"  , crosstab."ACTA DE ENTREGA"  , crosstab."AVALUO TECNICO"  , crosstab."LEY DEL ESTADO PLURINACIONAL"  , crosstab."DECRETO SUPREMO"  , crosstab."CONTRATO DE COMODATO"  , crosstab."ORDENANZA MUNICIPAL"  , crosstab."RESOLUCION MINISTERIAL"  , crosstab."RESOLUCION ADMINISTRATIVA"  , crosstab."DOCUMENTO DE COMPRA VENTA"  , crosstab."INFORME LEGAL"  , crosstab."INFORME TECNICO"  , crosstab."RESOLUCION MINISTERIAL  , ADMINISTRATIVA O MUNICIPAL (BAJA)"  , crosstab."DOCUMENTO UNICO DE IMPORTACION (DUI)"  , crosstab."TESTIMONIO DE PROPIEDAD"  FROM crosstab('select a.idbien, identidad, idtipodocumento, case when adicionado=false then 1 else -1 end as validadopor  FROM nueva\_validacion.documentobien a  join nueva\_validacion.bien b  ON b.idbien=a.idbien  where b.idclase=3  AND a.validado=true order by idbien'::text  , 'select iddocumento  FROM dj\_documento.documento\_clasebien  WHERE idclase=3  order by iddocumento'::text) crosstab(idbien integer  , identidad integer  , "R.U.A.T." integer  , "P.T.A." integer  , "CARNET DE PROPIEDAD" integer  , "POLIZA DE IMPORTACION" integer  , "FACTURA" integer  , "DOCUMENTO DE TRANSFERENCIA" integer  , "MINUTA DE TRANSFERENCIA DE COMPRA VENTA" integer  , "ACTA DE ENTREGA" integer  , "AVALUO TECNICO" integer  , "LEY DEL ESTADO PLURINACIONAL" integer  , "DECRETO SUPREMO" integer  , "CONTRATO DE COMODATO" integer  , "ORDENANZA MUNICIPAL" integer  , "RESOLUCION MINISTERIAL" integer  , "RESOLUCION ADMINISTRATIVA" integer  , "DOCUMENTO DE COMPRA VENTA" integer  , "INFORME LEGAL" integer  , "INFORME TECNICO" integer  , "RESOLUCION MINISTERIAL  , ADMINISTRATIVA O MUNICIPAL (BAJA)" integer  , "DOCUMENTO UNICO DE IMPORTACION (DUI)" integer  , "TESTIMONIO DE PROPIEDAD" integer) |

|  |  |
| --- | --- |
| **Vista: totales\_bienes\_documentos** | |
| **Nombre** | totales\_bienes\_documentos |
| **Descripción** | Muestra almacena el total de bienes juntamente con el total de documentos agrupados por entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.totales\_bienes\_documentos  AS  SELECT entidades\_bienes\_nrodocumentos.identidad  , count(entidades\_bienes\_nrodocumentos.idbien) AS totalbienes  , sum(entidades\_bienes\_nrodocumentos.nrodocumentos) AS totaldocumentos  FROM nueva\_validacion.entidades\_bienes\_nrodocumentos GROUP BY entidades\_bienes\_nrodocumentos.identidad ORDER BY entidades\_bienes\_nrodocumentos.identidad |

|  |  |
| --- | --- |
| **Vista: total\_bienesalquiler\_validados** | |
| **Nombre** | total\_bienesalquiler\_validados |
| **Descripción** | Muestra el reporte del total de bienes en alquiler validaos de cada entidad tomando encuenta el identificador de la entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.total\_bienesalquiler\_validados  AS  SELECT a.identidad  , count(a.idbien) AS total  , b.validados  FROM (nueva\_validacion.bienalquiler a  LEFT JOIN (SELECT bienalquiler.identidad  , count(bienalquiler.idbien) AS validados  FROM nueva\_validacion.bienalquiler  WHERE (bienalquiler.idestadovalidacion = 3) GROUP BY bienalquiler.identidad) b  ON ((b.identidad = a.identidad)))  WHERE (a.identidad IS NOT NULL) GROUP BY a.identidad  , b.validados |

|  |  |
| --- | --- |
| **Vista: total\_bienes\_validados** | |
| **Nombre** | total\_bienes\_validados |
| **Descripción** | Muestra el reporte del total de bienes validados de cada entidad tomando encuenta el identificador de la entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.total\_bienes\_validados  AS  SELECT a.identidad  , count(a.idbien) AS total  , b.validados  FROM (nueva\_validacion.bien a  LEFT JOIN (SELECT bien.identidad  , count(bien.idbien) AS validados  FROM nueva\_validacion.bien  WHERE (bien.idestadovalidacion = 3) GROUP BY bien.identidad) b  ON ((b.identidad = a.identidad)))  WHERE (a.identidad IS NOT NULL) GROUP BY a.identidad  , b.validados |

|  |  |
| --- | --- |
| **Vista: total\_bienes\_validados\_general** | |
| **Nombre** | total\_bienes\_validados\_general |
| **Descripción** | Muestra los bienes y los bienes en alquiler validados por el identificador de la entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.total\_bienes\_validados\_general  AS  SELECT total\_bienes\_validados.identidad  , total\_bienes\_validados.total  , total\_bienes\_validados.validados  FROM nueva\_validacion.total\_bienes\_validados **UNION ALL**  SELECT total\_bienesalquiler\_validados.identidad  , total\_bienesalquiler\_validados.total  , total\_bienesalquiler\_validados.validados  FROM nueva\_validacion.total\_bienesalquiler\_validados |

|  |  |
| --- | --- |
| **Vista: total\_validacion\_bienes** | |
| **Nombre** | total\_validacion\_bienes |
| **Descripción** | Es una vista que hace referencia a otra vista “ **total\_bienes\_validados\_general ”** para sacar el total de bienes y el total de bienes validados por entidad |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.total\_validacion\_bienes  AS  SELECT total\_bienes\_validados\_general.identidad  , sum(total\_bienes\_validados\_general.total) AS totalbienes  , sum(total\_bienes\_validados\_general.validados) AS totalvalidados  FROM nueva\_validacion.total\_bienes\_validados\_general GROUP BY total\_bienes\_validados\_general.identidad |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos** | |
| **Nombre** | vista\_cadenadocumentos |
| **Descripción** | Muestra el documento concatedano con su respectivo número condicionando la verificación que no exitan eliminados en cada bien |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM (nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento)))  WHERE (db.eliminado IS FALSE) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos2** | |
| **Nombre** | vista\_cadenadocumentos2 |
| **Descripción** | Muestra el documento concatedano con su respectivo número condicionando la verificación que no exitan eliminados en cada bien |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos2  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS nombres  , array\_to\_string(array\_agg(d.descripcion), ', '::text) AS documentos  , array\_to\_string(array\_agg(db.nrodocumento), ', '::text) AS nro  FROM (nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento)))  WHERE (db.eliminado IS FALSE) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos\_inmueble** | |
| **Nombre** | vista \_cadenadocumentos\_inmueble |
| **Descripción** | Muestra información de los documentos concatenados con su respectivo número, de la clase “TERRENO” |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos\_inmueble  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM ((nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) JOIN dj\_documento.documento\_clasebien cb  ON ((db.idtipodocumento = cb.iddocumento)))  WHERE ((db.eliminado = false)  AND (cb.idclase = 1)) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos\_inmueblealquiler** | |
| **Nombre** | vista \_cadenadocumentos\_inmueblealquiler |
| **Descripción** | Muestra información de los documentos concatenados con su respectivo número, de la clase “ INMUEBLE EN ALQUILER ” |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos\_inmueblealquiler  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM ((nueva\_validacion.documentobienalquiler db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) JOIN dj\_documento.documento\_clasebien cb  ON ((db.idtipodocumento = cb.iddocumento)))  WHERE ((db.eliminado = false)  AND (cb.idclase = 5)) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos\_maquinaria** | |
| **Nombre** | vista \_cadenadocumentos\_maquinaria |
| **Descripción** | Muestra información de los documentos concatenados con su respectivo número, de la clase “MAQUINARIA Y EQUIPO” |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos\_maquinaria  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM ((nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) JOIN dj\_documento.documento\_clasebien cb  ON ((db.idtipodocumento = cb.iddocumento)))  WHERE ((db.eliminado = false)  AND (cb.idclase = 4)) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos\_maquinariapesada** | |
| **Nombre** | vista \_cadenadocumentos\_maquinariapesada |
| **Descripción** | Muestra información de los documentos concatenados con su respectivo número, de la clase “MAQUINARIA PESADA” |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos\_maquinariapesada  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM ((nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) JOIN dj\_documento.documento\_clasebien cb  ON ((db.idtipodocumento = cb.iddocumento)))  WHERE ((db.eliminado = false)  AND (cb.idclase = 6)) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos\_vehiculo** | |
| **Nombre** | vista \_cadenadocumentos\_vehiculo |
| **Descripción** | Muestra información de los documentos concatenados con su respectivo número, de la clase “VEHICULOS AUTOMOTORES” |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos\_vehiculo  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM ((nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) JOIN dj\_documento.documento\_clasebien cb  ON ((db.idtipodocumento = cb.iddocumento)))  WHERE ((db.eliminado = false)  AND (cb.idclase = 3)) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_cadenadocumentos\_vehiculoalquiler** | |
| **Nombre** | vista \_cadenadocumentos\_vehiculoalquiler |
| **Descripción** | Muestra información de los documentos concatenados con su respectivo número, de la clase “VEHICULOS EN ALQUILER ” |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_cadenadocumentos\_vehiculoalquiler  AS  SELECT db.idbien  , array\_to\_string(array\_agg((((d.descripcion)::text || ' Nº: '::text) || db.nrodocumento)), ', '::text) AS documentos  FROM ((nueva\_validacion.documentobienalquiler db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) JOIN dj\_documento.documento\_clasebien cb  ON ((db.idtipodocumento = cb.iddocumento)))  WHERE ((db.eliminado = false)  AND (cb.idclase = 7)) GROUP BY db.idbien |

|  |  |
| --- | --- |
| **Vista: vista\_documentoseliminados** | |
| **Nombre** | vista\_documentoseliminados |
| **Descripción** | Muestra los documentos elimidados con los siguientes campos : identificador del bien, identificador del documento, número del documento, gestión, descripción del documento |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_documentoseliminados  AS  SELECT db.id  , db.idbien  , db.iddocumento  , db.nro\_documento  , db.gestion  , d.descripcion AS documento  FROM (nueva\_validacion.documentoseliminados db JOIN dj\_documento.documento d  ON ((d.id = db.iddocumento))) |

|  |  |
| --- | --- |
| **Vista: vista\_documentosoficiales** | |
| **Nombre** | vista\_documentosoficiales |
| **Descripción** | Muestra los documentos del bien con los siguientes campos: Identificador del bien, identificador del documento, numero del documento, la gestión y la descripción del documento |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_documentosoficiales  AS  SELECT db.id  , db.idbien  , db.idtipodocumento  , db.nrodocumento  , db.gestion  , d.descripcion AS documento  FROM (nueva\_validacion.documentobien db JOIN dj\_documento.documento d  ON ((d.id = db.idtipodocumento))) |

|  |  |
| --- | --- |
| **View: vista\_inmueblealquiler** | |
| **Nombre** | vista\_inmueblealquiler |
| **Descripción** | Muestra los inmuebles en alquiler con los siguientes Campos : identificador del bien, denominación del bien, canonalquiler, fecha inicio, fecha fin, departamento, tipo de bien , nombre de la entidad, estado de la validación, documentos del inmueble en alquiler |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_inmueblealquiler  AS  SELECT a.idbien  , a.denominacion  , a.direccion  , a.canonalquiler  , a.fechainicio  , a.fechafin  , c.descripcion AS departamento  , d.descripcion AS tipobien  , b.identidad  , b.idestadovalidacion  , e.documentos  FROM ((((nueva\_validacion.inmueblealquiler a  LEFT JOIN nueva\_validacion.bienalquiler b  ON ((a.idbien = b.idbien)))  LEFT JOIN geografia.departamento c  ON ((c.id = a.idciudad)))  LEFT JOIN dj\_activos.tipobien d  ON ((d.id = b.idtipobien)))  LEFT JOIN nueva\_validacion.vista\_cadenadocumentos\_inmueblealquiler e  ON ((e.idbien = b.idbien))) |

|  |  |
| --- | --- |
| **Vista: vista\_inmuebles** | |
| **Nombre** | vista\_inmuebles |
| **Descripción** | Muestra los datos de los inmuebles con sus respectivos campos: idenficador del activo, código activo, nombre del inmueble, tipo bien, dirección, superficie del terreno, superficie construida, departamento, nombre de la identidad, estado de la validación, identificador de la situación y los documentos del inmueble |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_inmuebles  AS  SELECT a.idbien  , a.codigoactivo  , b.denominacion  , c.descripcion AS tipobien  , b.direccion  , b.superficieterreno  , b.superficieconstruida  , a.departamento  , a.identidad  , a.idestadovalidacion  , a.idsituacion  , e.documentos  FROM (((nueva\_validacion.bien a JOIN nueva\_validacion.inmueble b  ON ((a.idbien = b.idbien))) JOIN dj\_activos.tipobien c  ON ((c.id = a.idtipobien)))  LEFT JOIN nueva\_validacion.vista\_cadenadocumentos\_inmueble e  ON ((e.idbien = b.idbien))) |

|  |  |
| --- | --- |
| **Vista: vista\_maquinaria** | |
| **Nombre** | vista\_maquinaria |
| **Descripción** | Muestra datos de rubro maquinaria son sus respectivos campos: identifacdor del bien, identificador de la entidad, identificador de la gestión, identificador del tipo del bien etc. |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_maquinaria  AS  SELECT b.id  , b.identidad  , b.idgestion  , b.idtipobien  , b.idestadovalidacion  , b.idestadodocumentacion  , b.idbien  , b.idsituacion  , b.idestadobien  , b.iduso  , b.idclase  , m.descripcion  , m.marca  , m.modelo  , m.nroserie  , sb.descripcion AS situacion  , tb.descripcion AS tipobien  , eb.descripcion AS estadobien  , u.descripcion AS uso  , c.descripcion AS clasebien  , e.nombre AS entidad  , e.sigla  , e.usuario  , vcd.documentos  FROM ((((((((nueva\_validacion.bien b JOIN nueva\_validacion.maquinaria m  ON ((m.idbien = b.idbien))) JOIN dj\_activos.situacionbien sb  ON ((sb.id = b.idsituacion))) JOIN dj\_activos.tipobien tb  ON ((tb.id = b.idtipobien))) JOIN dj\_activos.estadobien eb  ON ((eb.id = b.idestadobien))) JOIN dj\_activos.uso u  ON ((u.id = b.iduso))) JOIN dj\_activos.clase c  ON ((c.id = b.idclase))) JOIN gobierno.entidades e  ON ((e.id = b.identidad)))  LEFT JOIN nueva\_validacion.vista\_cadenadocumentos\_maquinaria vcd  ON ((b.idbien = vcd.idbien))) |

|  |  |
| --- | --- |
| **Vista: vista\_maquinariapesada** | |
| **Nombre** | vista\_maquinariapesada |
| **Descripción** | Muestra los datos de maquinaria pesada con sus repesctivos campos |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_maquinariapesada  AS  SELECT a.idbien  , b.descripcion  , c.descripcion AS tipobien  , b.marca  , b.modelo  , b.nrochasis  , b.nromotor  , b.color  , a.idsituacion  , a.idestadovalidacion  , a.identidad  , d.documentos  FROM (((nueva\_validacion.bien a JOIN nueva\_validacion.maquinariapesada b  ON ((a.idbien = b.idbien))) JOIN dj\_activos.tipobien c  ON ((a.idtipobien = c.id)))  LEFT JOIN nueva\_validacion.vista\_cadenadocumentos\_maquinariapesada d  ON ((d.idbien = b.idbien))) |

|  |  |
| --- | --- |
| **View: vista\_nrodocumentosbien** | |
| **Nombre** | vista\_nrodocumentosbien |
| **Descripción** | Muestra la cantidad de número de documentos almacenados por cada bien |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_nrodocumentosbien  AS  SELECT a.idbien  , a.identidad  , b.nrodocumentos  FROM (nueva\_validacion.bien a  LEFT JOIN nueva\_validacion.bienes\_cantdocumentos b  ON ((a.idbien = b.idbien))) |

|  |  |
| --- | --- |
| **Vista: vista\_nrodocumentosbienalquiler** | |
| **Nombre** | vista \_nrodocumentosbienalquiler |
| **Descripción** | Muestra la cantidad de número de documentos almacenados por cada bien en alquiler |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_nrodocumentosbienalquiler  AS  SELECT a.idbien  , a.identidad  , b.nrodocumentos  FROM (nueva\_validacion.bienalquiler a  LEFT JOIN nueva\_validacion.bienesalquiler\_cantdocumentos b  ON ((a.idbien = b.idbien))) |

|  |  |
| --- | --- |
| **Vista: vista\_users\_universo** | |
| **Nombre** | vista\_users\_universo |
| **Descripción** | Muestra los datos de los usuarios universales |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_users\_universo  AS  SELECT a.identidad  , a.idusuario  , b.nombre  , b.corto  FROM (nueva\_validacion.users\_universo a JOIN nueva\_validacion.usuario b  ON ((a.idusuario = b.id\_funcionario)))  WHERE (a.gestion = 2015) |

|  |  |
| --- | --- |
| **View: vista\_vehiculos** | |
| **Nombre** | vista\_vehiculos |
| **Descripción** | Muestra los datos de del rubro vehiculo con sus repstectivos campos |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_vehiculos  AS  SELECT b.id  , b.identidad  , b.idgestion  , b.idtipobien  , b.idestadovalidacion  , b.idestadodocumentacion  , b.idbien  , b.idsituacion  , b.idestadobien  , b.iduso  , b.idclase  , v.idtipovehiculo  , v.clase  , v.marca  , v.placa  , v.nromotor  , v.nrochasis  , v.modelo  , v.idprocedencia  , tb.descripcion AS tipobien  , vcd.documentos  FROM (((nueva\_validacion.bien b JOIN nueva\_validacion.vehiculo v  ON ((v.idbien = b.idbien))) JOIN dj\_activos.tipobien tb  ON ((tb.id = b.idtipobien)))  LEFT JOIN nueva\_validacion.vista\_cadenadocumentos\_vehiculo vcd  ON ((vcd.idbien = b.idbien))) |

|  |  |
| --- | --- |
| **Vista: vista\_vehiculosalquiler** | |
| **Nombre** | vista\_vehiculosalquiler |
| **Descripción** | Muestra los datos de del rubro vehiculo en alquiler con sus repstectivos campos |
| **Código** | CREATE OR REPLACE VIEW nueva\_validacion.vista\_vehiculosalquiler  AS  SELECT b.id  , b.identidad  , b.idgestion  , b.idtipobien  , b.idestadovalidacion  , b.idestadodocumentacion  , b.idbien  , b.iduso  , b.idclase  , v.marca  , v.placa  , v.nromotor  , v.nrochasis  , v.modelo  , v.color  , tb.descripcion AS tipobien  , v.canonalquiler  , v.fechainicio  , v.fechafin  , u.descripcion AS uso  , c.descripcion AS clasebien  , v.zona  , v.direccion  , e.nombre AS entidad  , e.sigla  , e.usuario  , p.descripcion AS ciudad  , vcd.documentos  FROM (((((((nueva\_validacion.bienalquiler b JOIN nueva\_validacion.vehiculoalquiler v  ON ((v.idbien = b.idbien))) JOIN dj\_activos.tipobien tb  ON ((tb.id = b.idtipobien))) JOIN dj\_activos.uso u  ON ((u.id = b.iduso))) JOIN dj\_activos.clase c  ON ((c.id = b.idclase))) JOIN gobierno.entidades e  ON ((e.id = b.identidad)))  LEFT JOIN geografia.departamento p  ON ((p.id = v.idciudad)))  LEFT JOIN nueva\_validacion.vista\_cadenadocumentos\_vehiculoalquiler vcd  ON ((vcd.idbien = b.idbien))) |

# PROCEDIMIENTOS ALMACENADOS

|  |  |
| --- | --- |
| **Procedimento: actualizarinmuebles** | |
| **Nombre** | actualizarinmuebles |
| **Descripción** | Actualiza los datos del rubro inmuebles |
| **Codigo** | CREATE OR REPLACE FUNCTION nueva\_validacion.actualizarinmuebles (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxbien in select \*from dj\_activos.inmueble LOOP  UPDATE nueva\_validacion.inmueble set superficieterreno = auxbien.superficie\_terreno , superficieconstruida = auxbien.superficie\_construida, denominacion = auxbien.denominacion, nrocatastro= auxbien.codigo\_catastral WHERE idbien = auxbien.idbien;  contador := contador + 1;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: actualizarmaquinaria** | |
| **Nombre** | actualizarmaquinaria |
| **Descripción** | Actualiza los datos del rubro maquinaria |
| **Codigo** | CREATE OR REPLACE FUNCTION nueva\_validacion.actualizarmaquinaria (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxbien in select \*from dj\_activos.maquinaria LOOP  UPDATE nueva\_validacion.maquinaria set  descripcion = auxbien.descripcion,  marca = auxbien.marca,  modelo = auxbien.modelo,  nroserie= auxbien.numero\_serie  WHERE idbien = auxbien.idbien;  contador := contador + 1;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: actualizarmaquinariapesada** | |
| **Nombre** | actualizarmaquinariapesada |
| **Descripción** | Actualiza los datos del rubro maquinaria pesada |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.actualizarmaquinariapesada (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxbien in select \*from dj\_activos.maquinaria LOOP  UPDATE nueva\_validacion.maquinariapesada set  descripcion = auxbien.descripcion,  marca = auxbien.marca,  modelo = auxbien.modelo,  nrochasis = auxbien.chasis,  nromotor = auxbien.motor,  color = auxbien.color  WHERE idbien = auxbien.idbien;  contador := contador + 1;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: actualizarvehiculos** | |
| **Nombre** | actualizarvehiculos |
| **Descripción** | Actualiza los datos del rubro vehículos |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.actualizarvehiculos (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxbien in select \*from dj\_activos.vehiculo LOOP  UPDATE nueva\_validacion.vehiculo set  clase = auxbien.tipo,  marca = auxbien.marca,  placa= auxbien.placa,  nromotor = auxbien.motor,  nrochasis=auxbien.chasis,  modelo = auxbien.modelo,  idprocedencia = auxbien.idprocedencia WHERE idbien = auxbien.idbien;  contador := contador + 1;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedure: adicionar\_documentos\_pasados** | |
| Name | adicionar\_documentos\_pasados |
| Comment | Adicionar documentos pasados |
| Code | CREATE OR REPLACE FUNCTION nueva\_validacion.adicionar\_documentos\_pasados (id\_bien integer) RETURNS integer AS  $BODY$  declare  auxdocumento record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxdocumento in select \* from dj\_documento.documento\_bien where gestion = 2015 LOOP  IF EXISTS (SELECT \* FROM nueva\_validacion.bien WHERE idbien = auxdocumento.idbien)THEN  IF NOT EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxdocumento.idbien and idtipodocumento = auxdocumento.iddocumento and nrodocumento = auxdocumento.nro\_documento )THEN  INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxdocumento.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  contador := contador + 1;  END IF;  END IF;  --contador := contador + 1;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimento : agregarcamposextra** | |
| **Nombre** | agregarcamposextra |
| **Descripción** | Se agrega un campo extra departamento en los bienes |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.agregarcamposextra () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxbien1 record;  declare auxinmueble record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxbien1 in select id, codigo\_activo, case idciudad when 1 then 'CHUQUISACA' when 2 then 'LA PAZ'  when 3 then 'COCHABAMBA' when 4 then 'ORURO' when 5 then 'POTOSI' when 6 then 'TARIJA' when 7 then 'SANTA CRUZ' when 8 then 'BENI'  when 9 then 'PANDO' END as depto from dj\_activos.bien LOOP  update nueva\_validacion.bien set codigoactivo = auxbien1.codigo\_activo, departamento = auxbien1.depto where idbien = auxbien1.id;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: agregarcamposextraalquiler** | |
| **Nombre** | agregarcamposextraalquiler |
| **Descripción** | Se agrega un campo extra departamento en los bienes en alquiler |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.agregarcamposextraalquiler () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxbien1 record;  declare auxinmueble record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxbien1 in select id, idciudad, case idciudad when 1 then 'CHUQUISACA' when 2 then 'LA PAZ'  when 3 then 'COCHABAMBA' when 4 then 'ORURO' when 5 then 'POTOSI' when 6 then 'TARIJA' when 7 then 'SANTA CRUZ' when 8 then 'BENI'  when 9 then 'PANDO' END as depto from dj\_alquiler.bien LOOP  update nueva\_validacion.bienalquiler set iddepartamento = auxbien1.idciudad, departamento = auxbien1.depto  where idbien = auxbien1.id;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: cambiarestado** | |
| **Nombre** | cambiarestado |
| **Descripción** | Función |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.cambiarestado () RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxbien in select \*from nueva\_validacion.bien where idestadovalidacion in (1, 5) and identidad = 721 LOOP  bienes :=(select count(\*) from nueva\_validacion.documentobien where idbien=auxbien.idbien and eliminado=false) as total;  validados := (select count(\*) from nueva\_validacion.documentobien where idbien=auxbien.idbien and eliminado=false and validado=true) as validados;  bien :=auxbien.idbien ;  if bienes = validados then  contador := contador + 1;  end if;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: cambiarestado2** | |
| **Nombre** | cambiarestado2 |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.cambiarestado2 (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  FOR auxbien in select \*from nueva\_validacion.bien where idestadovalidacion in (1, 5) and identidad = id\_bien LOOP  bienes :=(select count(\*) from nueva\_validacion.documentobien where idbien=auxbien.idbien and eliminado=false) as total;  validados := (select count(\*) from nueva\_validacion.documentobien where idbien=auxbien.idbien and eliminado=false and validado=true) as validados;  bien :=auxbien.idbien ;  if bienes > 0 then  if bienes = validados then  UPDATE nueva\_validacion.bien set idestadovalidacion = 3 WHERE idbien = auxbien.idbien;  contador := contador + 1;  end if;  end if;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: cambiar\_estado\_bienes** | |
| **Nombre** | cambiar\_estado\_bienes |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.cambiar\_estado\_bienes () RETURNS void AS  $BODY$  DECLARE  declare auxdocumento record;  BEGIN  FOR auxdocumento in select \* from dj\_documento.documento\_bien where gestion = 2016 LOOP  UPDATE nueva\_validacion.bien set idestadovalidacion = 5 WHERE idbien = auxdocumento.idbien;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: datobaja** | |
| **Nombre** | datobaja |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datobaja () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxbaja record;  BEGIN  FOR auxbaja in select idbien, idmotivo from dj\_activos.bajasbienes LOOP  UPDATE nueva\_validacion.bien set idmotivobaja = auxbaja.idmotivo WHERE idbien = auxbaja.idbien;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: datosmigrados** | |
| **Nombre** | datosmigrados |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datosmigrados () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  declare auxvehiculo record;  declare auxmaquinaria record;  declare auxmaquinariapesada record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  --select \* from dj\_activos.bien where fechainscripcion > '2014-11-17'  --select \* from dj\_activos.inmueble where idbien > 112654  --select \* from dj\_activos.vehiculo where idbien > 112654  --select \* from dj\_activos.maquinaria where idbien > 112654  --select \* from dj\_documento.documento\_bien where idbien > 112654  --delete from nueva\_validacion.bien  --delete from nueva\_validacion.inmueble  --delete from nueva\_validacion.vehiculo  FOR auxbien in select id as idbien, idtipobien, 2014 as idgestion, identidad, idclase, 1 as idestadovalidacion, 4 as idestadodocumentacion, zona, direccion, iduso, idestadobien, idsituacion from dj\_activos.bien where fechainscripcion < '2014-11-30' LOOP  INSERT INTO nueva\_validacion.bien (identidad, idgestion, idtipobien, idestadovalidacion, idestadodocumentacion, idbien, idsituacion, idestadobien, iduso, idclase)  VALUES (auxbien.identidad, auxbien.idgestion, auxbien.idtipobien, auxbien.idestadovalidacion, auxbien.idestadodocumentacion, auxbien.idbien, auxbien.idsituacion, auxbien.idestadobien, auxbien.iduso, auxbien.idclase);  FOR auxdocumento in select \* from dj\_documento.documento\_bien where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  END LOOP;  IF auxbien.idclase = 1 or auxbien.idclase = 2 THEN  FOR auxinmueble in select idbien, superficie\_terreno, superficie\_construida, denominacion, codigo\_catastral, 2014 as idgestion from dj\_activos.inmueble where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.inmueble (idbien, superficieterreno, superficieconstruida, zona, direccion, denominacion, nrocatastro, idgestion)  VALUES (auxinmueble.idbien, auxinmueble.superficie\_terreno, auxinmueble.superficie\_construida, auxbien.zona, auxbien.direccion, auxinmueble.denominacion, auxinmueble.codigo\_catastral, auxinmueble.idgestion);  END LOOP;  END IF;  IF auxbien.idclase = 3 THEN  FOR auxvehiculo in select idbien, tipo as clase, marca, placa, motor, chasis, modelo, 2014 as idgestion, idprocedencia from dj\_activos.vehiculo where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.vehiculo (idbien, idtipovehiculo, clase, marca, placa, nromotor, nrochasis, modelo, idgestion, idprocedencia)  VALUES (auxvehiculo.idbien, auxbien.idtipobien, auxvehiculo.clase, auxvehiculo.marca, auxvehiculo.placa, auxvehiculo.motor, auxvehiculo.chasis, auxvehiculo.modelo, auxvehiculo.idgestion, auxvehiculo.idprocedencia);  END LOOP;  END IF;  IF auxbien.idclase = 4 THEN  FOR auxmaquinaria in select idbien, descripcion, marca, modelo, numero\_serie, 2014 as gestion from dj\_activos.maquinaria where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.maquinaria (idbien, descripcion, marca, modelo, nroserie, idgestion)  VALUES (auxbien.idbien, auxmaquinaria.descripcion, auxmaquinaria.marca, auxmaquinaria.modelo, auxmaquinaria.numero\_serie, auxmaquinaria.gestion);  END LOOP;  END IF;  IF auxbien.idclase = 6 THEN  FOR auxmaquinariapesada in select idbien, descripcion, marca, modelo, chasis, motor, color, 2014 as gestion from dj\_activos.maquinaria where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.maquinariapesada (idbien, descripcion, marca, modelo, nrochasis, nromotor, color, idgestion)  VALUES (auxbien.idbien, auxmaquinariapesada.descripcion, auxmaquinariapesada.marca, auxmaquinariapesada.modelo, auxmaquinariapesada.chasis, auxmaquinariapesada.motor, auxmaquinariapesada.color, auxmaquinariapesada.gestion);  END LOOP;  END IF;  --FOR auxdocumento in select \* from dj\_documento.documento\_bien where idbien = auxbien.idbien LOOP  --INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  --VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  --END LOOP;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimientos: datosmigrados2** | |
| **Nombre** | datosmigrados2 |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datosmigrados2 () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  declare auxvehiculo record;  declare auxmaquinaria record;  declare auxmaquinariapesada record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  --select \* from dj\_activos.bien where fechainscripcion > '2014-11-17'  --select \* from dj\_activos.inmueble where idbien > 112654  --select \* from dj\_activos.vehiculo where idbien > 112654  --select \* from dj\_activos.maquinaria where idbien > 112654  --select \* from dj\_documento.documento\_bien where idbien > 112654  --delete from nueva\_validacion.bien  --delete from nueva\_validacion.inmueble  --delete from nueva\_validacion.vehiculo  FOR auxbien in select id as idbien, idtipobien, 2015 as idgestion, identidad, idclase, 1 as idestadovalidacion, 4 as idestadodocumentacion, zona, direccion, iduso, idestadobien, idsituacion from dj\_activos.bien where fechainscripcion > '2014-11-30' LOOP  INSERT INTO nueva\_validacion.bien (identidad, idgestion, idtipobien, idestadovalidacion, idestadodocumentacion, idbien, idsituacion, idestadobien, iduso, idclase)  VALUES (auxbien.identidad, auxbien.idgestion, auxbien.idtipobien, auxbien.idestadovalidacion, auxbien.idestadodocumentacion, auxbien.idbien, auxbien.idsituacion, auxbien.idestadobien, auxbien.iduso, auxbien.idclase);  FOR auxdocumento in select \* from dj\_documento.documento\_bien where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  END LOOP;  IF auxbien.idclase = 1 or auxbien.idclase = 2 THEN  FOR auxinmueble in select idbien, superficie\_terreno, superficie\_construida, denominacion, codigo\_catastral, 2015 as idgestion from dj\_activos.inmueble where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.inmueble (idbien, superficieterreno, superficieconstruida, zona, direccion, denominacion, nrocatastro, idgestion)  VALUES (auxinmueble.idbien, auxinmueble.superficie\_terreno, auxinmueble.superficie\_construida, auxbien.zona, auxbien.direccion, auxinmueble.denominacion, auxinmueble.codigo\_catastral, auxinmueble.idgestion);  END LOOP;  END IF;  IF auxbien.idclase = 3 THEN  FOR auxvehiculo in select idbien, tipo as clase, marca, placa, motor, chasis, modelo, 2015 as idgestion, idprocedencia from dj\_activos.vehiculo where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.vehiculo (idbien, idtipovehiculo, clase, marca, placa, nromotor, nrochasis, modelo, idgestion, idprocedencia)  VALUES (auxvehiculo.idbien, auxbien.idtipobien, auxvehiculo.clase, auxvehiculo.marca, auxvehiculo.placa, auxvehiculo.motor, auxvehiculo.chasis, auxvehiculo.modelo, auxvehiculo.idgestion, auxvehiculo.idprocedencia);  END LOOP;  END IF;  IF auxbien.idclase = 4 THEN  FOR auxmaquinaria in select idbien, descripcion, marca, modelo, numero\_serie, 2015 as gestion from dj\_activos.maquinaria where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.maquinaria (idbien, descripcion, marca, modelo, nroserie, idgestion)  VALUES (auxbien.idbien, auxmaquinaria.descripcion, auxmaquinaria.marca, auxmaquinaria.modelo, auxmaquinaria.numero\_serie, auxmaquinaria.gestion);  END LOOP;  END IF;  IF auxbien.idclase = 6 THEN  FOR auxmaquinariapesada in select idbien, descripcion, marca, modelo, chasis, motor, color, 2015 as gestion from dj\_activos.maquinaria where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.maquinariapesada (idbien, descripcion, marca, modelo, nrochasis, nromotor, color, idgestion)  VALUES (auxbien.idbien, auxmaquinariapesada.descripcion, auxmaquinariapesada.marca, auxmaquinariapesada.modelo, auxmaquinariapesada.chasis, auxmaquinariapesada.motor, auxmaquinariapesada.color, auxmaquinariapesada.gestion);  END LOOP;  END IF;  --FOR auxdocumento in select \* from dj\_documento.documento\_bien where idbien = auxbien.idbien LOOP  --INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  --VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  --END LOOP;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimientos: datosmigrados3** | |
| **Nombre** | datosmigrados3 |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datosmigrados3 () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  declare auxvehiculo record;  declare auxmaquinaria record;  declare auxmaquinariapesada record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  --select \* from dj\_activos.bien where fechainscripcion > '2014-11-17'  --select \* from dj\_activos.inmueble where idbien > 112654  --select \* from dj\_activos.vehiculo where idbien > 112654  --select \* from dj\_activos.maquinaria where idbien > 112654  --select \* from dj\_documento.documento\_bien where idbien > 112654  --delete from nueva\_validacion.bien  --delete from nueva\_validacion.inmueble  --delete from nueva\_validacion.vehiculo  FOR auxbien in select id as idbien, idtipobien, 2016 as idgestion, identidad, idclase, 1 as idestadovalidacion, 4 as idestadodocumentacion, zona, direccion, iduso, idestadobien, idsituacion from dj\_activos.bien where fechainscripcion > '2015-11-30' LOOP  INSERT INTO nueva\_validacion.bien (identidad, idgestion, idtipobien, idestadovalidacion, idestadodocumentacion, idbien, idsituacion, idestadobien, iduso, idclase)  VALUES (auxbien.identidad, auxbien.idgestion, auxbien.idtipobien, auxbien.idestadovalidacion, auxbien.idestadodocumentacion, auxbien.idbien, auxbien.idsituacion, auxbien.idestadobien, auxbien.iduso, auxbien.idclase);  FOR auxdocumento in select \* from dj\_documento.documento\_bien where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  END LOOP;  IF auxbien.idclase = 1 or auxbien.idclase = 2 THEN  FOR auxinmueble in select idbien, superficie\_terreno, superficie\_construida, denominacion, codigo\_catastral, 2016 as idgestion from dj\_activos.inmueble where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.inmueble (idbien, superficieterreno, superficieconstruida, zona, direccion, denominacion, nrocatastro, idgestion)  VALUES (auxinmueble.idbien, auxinmueble.superficie\_terreno, auxinmueble.superficie\_construida, auxbien.zona, auxbien.direccion, auxinmueble.denominacion, auxinmueble.codigo\_catastral, auxinmueble.idgestion);  END LOOP;  END IF;  IF auxbien.idclase = 3 THEN  FOR auxvehiculo in select idbien, tipo as clase, marca, placa, motor, chasis, modelo, 2016 as idgestion, idprocedencia from dj\_activos.vehiculo where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.vehiculo (idbien, idtipovehiculo, clase, marca, placa, nromotor, nrochasis, modelo, idgestion, idprocedencia)  VALUES (auxvehiculo.idbien, auxbien.idtipobien, auxvehiculo.clase, auxvehiculo.marca, auxvehiculo.placa, auxvehiculo.motor, auxvehiculo.chasis, auxvehiculo.modelo, auxvehiculo.idgestion, auxvehiculo.idprocedencia);  END LOOP;  END IF;  IF auxbien.idclase = 4 THEN  FOR auxmaquinaria in select idbien, descripcion, marca, modelo, numero\_serie, 2016 as gestion from dj\_activos.maquinaria where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.maquinaria (idbien, descripcion, marca, modelo, nroserie, idgestion)  VALUES (auxbien.idbien, auxmaquinaria.descripcion, auxmaquinaria.marca, auxmaquinaria.modelo, auxmaquinaria.numero\_serie, auxmaquinaria.gestion);  END LOOP;  END IF;  IF auxbien.idclase = 6 THEN  FOR auxmaquinariapesada in select idbien, descripcion, marca, modelo, chasis, motor, color, 2016 as gestion from dj\_activos.maquinaria where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.maquinariapesada (idbien, descripcion, marca, modelo, nrochasis, nromotor, color, idgestion)  VALUES (auxbien.idbien, auxmaquinariapesada.descripcion, auxmaquinariapesada.marca, auxmaquinariapesada.modelo, auxmaquinariapesada.chasis, auxmaquinariapesada.motor, auxmaquinariapesada.color, auxmaquinariapesada.gestion);  END LOOP;  END IF;  --FOR auxdocumento in select \* from dj\_documento.documento\_bien where idbien = auxbien.idbien LOOP  --INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  --VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  --END LOOP;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: datosmigradosalquiler** | |
| **Nombre** | datosmigradosalquiler |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datosmigradosalquiler () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  declare auxvehiculo record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxbien in select id as idbien, idtipobien, 2014 as idgestion, identidad, idclase, 1 as idestadovalidacion, 4 as idestadodocumentacion, zona,  direccion, canon\_alquiler, fecha\_inicio\_contrato, fecha\_final\_contrato, iduso, idciudad from dj\_alquiler.bien  where fechainscripcion < '2014-11-30' LOOP  INSERT INTO nueva\_validacion.bienalquiler (identidad, idgestion, idtipobien, idestadovalidacion, idestadodocumentacion, idbien, iduso, idclase)  VALUES (auxbien.identidad, auxbien.idgestion, auxbien.idtipobien, auxbien.idestadovalidacion, auxbien.idestadodocumentacion,  auxbien.idbien, auxbien.iduso, auxbien.idclase);  FOR auxdocumento in select \*, 2014 as gestion from dj\_alquiler.documento\_bien where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.documentobienalquiler(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  END LOOP;  IF auxbien.idclase = 5 THEN  FOR auxinmueble in select denominacion, 2014 as idgestion from dj\_alquiler.inmueble where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.inmueblealquiler (idbien, denominacion, direccion, canonalquiler, fechainicio, fechafin, idgestion, idciudad, zona)  VALUES (auxbien.idbien, auxinmueble.denominacion, auxbien.direccion, auxbien.canon\_alquiler, auxbien.fecha\_inicio\_contrato, auxbien.fecha\_final\_contrato, auxinmueble.idgestion, auxbien.idciudad, auxbien.zona);  END LOOP;  END IF;  IF auxbien.idclase = 7 THEN  FOR auxvehiculo in select idbien, marca, modelo, placa, motor, chasis, color, 2014 as idgestion from dj\_alquiler.vehiculo where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.vehiculoalquiler (idbien, marca, modelo, placa, nromotor, nrochasis, color, canonalquiler, idgestion, fechainicio, fechafin, idciudad, zona, direccion)  VALUES (auxbien.idbien, auxvehiculo.marca, auxvehiculo.modelo, auxvehiculo.placa, auxvehiculo.motor, auxvehiculo.chasis, auxvehiculo.color, auxbien.canon\_alquiler, auxvehiculo.idgestion, auxbien.fecha\_inicio\_contrato, auxbien.fecha\_final\_contrato, auxbien.idciudad, auxbien.zona, auxbien.direccion);  END LOOP;  END IF;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: datosmigradosalquiler2** | |
| **Nombre** | datosmigradosalquiler2 |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datosmigradosalquiler2 () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  declare auxvehiculo record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxbien in select id as idbien, idtipobien, 2015 as idgestion, identidad, idclase, 1 as idestadovalidacion, 4 as idestadodocumentacion, zona,  direccion, canon\_alquiler, fecha\_inicio\_contrato, fecha\_final\_contrato, iduso, idciudad from dj\_alquiler.bien  where extract (year from fecha\_inicio\_contrato)=2015 and extract(year from fecha\_final\_contrato)=2015 LOOP  INSERT INTO nueva\_validacion.bienalquiler (identidad, idgestion, idtipobien, idestadovalidacion, idestadodocumentacion, idbien, iduso, idclase)  VALUES (auxbien.identidad, auxbien.idgestion, auxbien.idtipobien, auxbien.idestadovalidacion, auxbien.idestadodocumentacion,  auxbien.idbien, auxbien.iduso, auxbien.idclase);  FOR auxdocumento in select \*, 2015 as gestion from dj\_alquiler.documento\_bien where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.documentobienalquiler(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  END LOOP;  IF auxbien.idclase = 5 THEN  FOR auxinmueble in select denominacion, 2015 as idgestion from dj\_alquiler.inmueble where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.inmueblealquiler (idbien, denominacion, direccion, canonalquiler, fechainicio, fechafin, idgestion, idciudad, zona)  VALUES (auxbien.idbien, auxinmueble.denominacion, auxbien.direccion, auxbien.canon\_alquiler, auxbien.fecha\_inicio\_contrato, auxbien.fecha\_final\_contrato, auxinmueble.idgestion, auxbien.idciudad, auxbien.zona);  END LOOP;  END IF;  IF auxbien.idclase = 7 THEN  FOR auxvehiculo in select idbien, marca, modelo, placa, motor, chasis, color, 2015 as idgestion from dj\_alquiler.vehiculo where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.vehiculoalquiler (idbien, marca, modelo, placa, nromotor, nrochasis, color, canonalquiler, idgestion, fechainicio, fechafin, idciudad, zona, direccion)  VALUES (auxbien.idbien, auxvehiculo.marca, auxvehiculo.modelo, auxvehiculo.placa, auxvehiculo.motor, auxvehiculo.chasis, auxvehiculo.color, auxbien.canon\_alquiler, auxvehiculo.idgestion, auxbien.fecha\_inicio\_contrato, auxbien.fecha\_final\_contrato, auxbien.idciudad, auxbien.zona, auxbien.direccion);  END LOOP;  END IF;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: datosmigradosalquiler3** | |
| **Nombre** | datosmigradosalquiler3 |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.datosmigradosalquiler3 () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  declare auxvehiculo record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxbien in select id as idbien, idtipobien, 2016 as idgestion, identidad, idclase, 1 as idestadovalidacion, 4 as idestadodocumentacion, zona,  direccion, canon\_alquiler, fecha\_inicio\_contrato, fecha\_final\_contrato, iduso, idciudad from dj\_alquiler.bien  where extract (year from fecha\_inicio\_contrato)=2016 and extract(year from fecha\_final\_contrato)=2016 LOOP  IF EXISTS (SELECT \* FROM nueva\_validacion.bienalquiler WHERE idbien = auxbien.idbien)THEN  UPDATE nueva\_validacion.bienalquiler set idgestion = auxbien.idgestion WHERE idbien = auxbien.idbien;  ELSE  INSERT INTO nueva\_validacion.bienalquiler (identidad, idgestion, idtipobien, idestadovalidacion, idestadodocumentacion, idbien, iduso, idclase)  VALUES (auxbien.identidad, auxbien.idgestion, auxbien.idtipobien, auxbien.idestadovalidacion, auxbien.idestadodocumentacion,  auxbien.idbien, auxbien.iduso, auxbien.idclase);  END IF;  FOR auxdocumento in select \*, 2016 as gestion from dj\_alquiler.documento\_bien where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.documentobienalquiler(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion, eliminado)  VALUES (auxbien.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion, 'FALSE');  END LOOP;  IF auxbien.idclase = 5 THEN  FOR auxinmueble in select denominacion, 2016 as idgestion from dj\_alquiler.inmueble where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.inmueblealquiler (idbien, denominacion, direccion, canonalquiler, fechainicio, fechafin, idgestion, idciudad, zona)  VALUES (auxbien.idbien, auxinmueble.denominacion, auxbien.direccion, auxbien.canon\_alquiler, auxbien.fecha\_inicio\_contrato, auxbien.fecha\_final\_contrato, auxinmueble.idgestion, auxbien.idciudad, auxbien.zona);  END LOOP;  END IF;  IF auxbien.idclase = 7 THEN  FOR auxvehiculo in select idbien, marca, modelo, placa, motor, chasis, color, 2016 as idgestion from dj\_alquiler.vehiculo where idbien = auxbien.idbien LOOP  INSERT INTO nueva\_validacion.vehiculoalquiler (idbien, marca, modelo, placa, nromotor, nrochasis, color, canonalquiler, idgestion, fechainicio, fechafin, idciudad, zona, direccion)  VALUES (auxbien.idbien, auxvehiculo.marca, auxvehiculo.modelo, auxvehiculo.placa, auxvehiculo.motor, auxvehiculo.chasis, auxvehiculo.color, auxbien.canon\_alquiler, auxvehiculo.idgestion, auxbien.fecha\_inicio\_contrato, auxbien.fecha\_final\_contrato, auxbien.idciudad, auxbien.zona, auxbien.direccion);  END LOOP;  END IF;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: documentoelimadocumentobien** | |
| **Nombre** | documentoelimadocumentobien |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.documentoelimadocumentobien () RETURNS void AS  $BODY$  DECLARE  declare auxdocumento record;  declare auxdocumentoelim record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxdocumento in select id, iddocumento, idbien, codigo\_digitalizado, nro\_hojas, digitalizado, nro\_documento, usuario, usuario\_dj, fecha\_inscripcion, observaciones, gestion, validado FROM nueva\_validacion.documentoseliminados LOOP  INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion, validado, eliminado)  VALUES (auxdocumento.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, null, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion, auxdocumento.validado, true);  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: documentoeliminado** | |
| **Nombre** | documentoeliminado |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.documentoeliminado () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxvalidacion record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  --91180  --120539  -- idbien 6053 docs 14, 15 16 21  FOR auxbien in select id from dj\_activos.bien LOOP  FOR auxdocumento in  select a.idbien, b.iddocumento, b.codigo\_digitalizado, b.nro\_hojas, b.digitalizado, b.nro\_documento, b.usuario, b.usuario\_dj, b.fecha\_inscripcion,  b.observaciones, b.gestion  from dj\_documento.documento\_bien a  right join validaciondocumentorecuperado b on a.iddocumento=b.iddocumento and a.idbien=b.idbien  --where b.idbien = 6053  where b.idbien = auxbien.id LOOP  if auxdocumento.idbien is null then  insert into public.documentoseliminados (iddocumento, idbien, codigo\_digitalizado, nro\_hojas, digitalizado, nro\_documento, usuario, usuario\_dj, fecha\_inscripcion, observaciones, gestion)  values(auxdocumento.iddocumento, auxbien.id, auxdocumento.codigo\_digitalizado, auxdocumento.nro\_hojas, auxdocumento.digitalizado, auxdocumento.nro\_documento, 0, 0, auxdocumento.fecha\_inscripcion, auxdocumento.observaciones, auxdocumento.gestion);  raise notice 'uno eliminado';  else  raise notice 'mismos documentos';  end if;  END LOOP;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: documentosadicionadosgestinpast** | |
| **Nombre** | documentosadicionadosgestinpast |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.documentosadicionadosgestinpast () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxbien1 record;  declare auxinmueble record;  declare auxdocumento record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  update nueva\_validacion.documentobien set adicionado = TRUE where idfuncionario is not null;  update nueva\_validacion.documentobienalquiler set adicionado = TRUE where idfuncionario is not null;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: estadodefinitivo** | |
| **Nombre** | estadodefinitivo |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.estadodefinitivo (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  --inmuebles  /\*FOR auxbien in select \*from nueva\_validacion.bien where idestadovalidacion = 3 and idestadodocumentacion in (2, 3, 4) and idclase in(1, 2) LOOP  IF NOT EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 3 WHERE idbien = auxbien.idbien;  ELSE  IF EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien and idtipodocumento in (1, 2))THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 1 WHERE idbien = auxbien.idbien;  contador := contador + 1;  ELSE --contador := contador + 1;  --END IF;  --IF EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien and idtipodocumento <> 1 and idtipodocumento <> 2)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 2 WHERE idbien = auxbien.idbien;  --contador := contador + 1;  END IF;  END IF;  END LOOP; \*/  --vehiculos  /\*FOR auxbien in select \*from nueva\_validacion.bien where idestadovalidacion = 3 and idestadodocumentacion in (2, 3, 4) and idclase in(3) LOOP  IF NOT EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 3 WHERE idbien = auxbien.idbien;  --contador := contador + 1;  ELSE  IF EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien and idtipodocumento = 4)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 1 WHERE idbien = auxbien.idbien;  contador := contador + 1;  ELSE  UPDATE nueva\_validacion.bien set idestadodocumentacion = 2 WHERE idbien = auxbien.idbien;  --contador := contador + 1;  END IF;  END IF;  END LOOP; \*/  --maquinaria  /\* FOR auxbien in select \*from nueva\_validacion.bien where idestadovalidacion = 3 and idestadodocumentacion in (2, 3, 4) and idclase in(4) LOOP  IF NOT EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 3 WHERE idbien = auxbien.idbien;  ELSE  IF EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien and idtipodocumento = 8)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 1 WHERE idbien = auxbien.idbien;  contador := contador + 1;  ELSE  UPDATE nueva\_validacion.bien set idestadodocumentacion = 2 WHERE idbien = auxbien.idbien;  END IF;  END IF;  END LOOP; \*/  --maquinaria pesada  FOR auxbien in select \*from nueva\_validacion.bien where idestadovalidacion = 3 and idestadodocumentacion in (2, 3, 4) and idclase in(6) LOOP  IF NOT EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 3 WHERE idbien = auxbien.idbien;  ELSE  IF EXISTS (SELECT \* FROM nueva\_validacion.documentobien WHERE idbien = auxbien.idbien and idtipodocumento = 7)THEN  UPDATE nueva\_validacion.bien set idestadodocumentacion = 1 WHERE idbien = auxbien.idbien;  contador := contador + 1;  ELSE  UPDATE nueva\_validacion.bien set idestadodocumentacion = 2 WHERE idbien = auxbien.idbien;  END IF;  END IF;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: estadodefinitivoalquiler** | |
| **Nombre** | estadodefinitivoalquiler |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.estadodefinitivoalquiler (id\_bien integer) RETURNS integer AS  $BODY$  DECLARE  declare auxbien record;  consulta4 record;  bien numeric;  bienes numeric;  validados numeric;  contador numeric;  BEGIN  bienes := 0;  validados := 0;  contador := 0;  --inmuebles  FOR auxbien in select \*from nueva\_validacion.bienalquiler where idestadovalidacion = 3 and idclase in(5) LOOP  IF NOT EXISTS (SELECT \* FROM nueva\_validacion.documentobienalquiler WHERE idbien = auxbien.idbien)THEN  UPDATE nueva\_validacion.bienalquiler set idestadodocumentacion = 3 WHERE idbien = auxbien.idbien;  ELSE  IF EXISTS (SELECT \* FROM nueva\_validacion.documentobienalquiler WHERE idbien = auxbien.idbien and idtipodocumento in (23))THEN  UPDATE nueva\_validacion.bienalquiler set idestadodocumentacion = 1 WHERE idbien = auxbien.idbien;  contador := contador + 1;  ELSE  UPDATE nueva\_validacion.bienalquiler set idestadodocumentacion = 2 WHERE idbien = auxbien.idbien;  --contador := contador + 1;  END IF;  END IF;  END LOOP;  return contador ;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: estadodocumentacionbien** | |
| **Nombre** | estadodocumentacionbien |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.estadodocumentacionbien (vidvalidacion integer, vidclase integer) RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxbienvalidadoinmueble record;  declare auxbienvalidadovehiculo record;  declare auxbienvalidadomaquinaria record;  declare auxbienvalidadomaquinariapesada record;  declare auxbienvalidadoinmueblealquiler record;  declare auxbienvalidadovehiculoalquiler record;  declare auxdocumentobieninmueble record;  declare auxdocumentobienvehiculo record;  declare auxdocumentobienmaquinaria record;  declare auxdocumentobienmaquinariapesada record;  declare auxdocumentobieninmueblealquiler record;  declare auxdocumentobienvehiculoalquiler record;  declare cantidad1 integer;  declare estadodoc integer;  declare bandera integer;  BEGIN  if vidclase = 1 then  FOR auxbienvalidadoinmueble in select id, idbien, idcorrespondencia, adjunta, legible, iddocumentobien FROM nueva\_validacion.validacionxgestioninmueble WHERE id = vidvalidacion LOOP  select into estadodoc idestadodocumentacion from nueva\_validacion.bien WHERE idbien = auxbienvalidadoinmueble.idbien;  FOR auxdocumentobieninmueble in select id as iddoc, idbien, idtipodocumento, validado FROM nueva\_validacion.documentobien where id = auxbienvalidadoinmueble.iddocumentobien LOOP  if estadodoc > 1 then  if auxbienvalidadoinmueble.adjunta = 't' and auxbienvalidadoinmueble.idcorrespondencia = 0 and auxbienvalidadoinmueble.legible = 't' then  if auxdocumentobieninmueble.idtipodocumento = 1 then  update nueva\_validacion.bien set idestadodocumentacion = 1 where idbien = auxdocumentobieninmueble.idbien;  else  update nueva\_validacion.bien set idestadodocumentacion = 2 where idbien = auxdocumentobieninmueble.idbien;  end if;  else  update nueva\_validacion.bien set idestadodocumentacion = 3 where idbien = auxdocumentobieninmueble.idbien;  end if;  end if;  END LOOP;  END LOOP;  end if;  if vidclase = 3 then  FOR auxbienvalidadovehiculo in select id, idbien, idcorrespondencia, adjunta, legible, iddocumentobien FROM nueva\_validacion.validacionxgestionvehiculo WHERE id = vidvalidacion LOOP  select into estadodoc idestadodocumentacion from nueva\_validacion.bien WHERE idbien = auxbienvalidadovehiculo.idbien;  FOR auxdocumentobienvehiculo in select id as iddoc, idbien, idtipodocumento, validado FROM nueva\_validacion.documentobien where id = auxbienvalidadovehiculo.iddocumentobien LOOP  if estadodoc > 1 then  if auxbienvalidadovehiculo.adjunta = 't' and auxbienvalidadovehiculo.idcorrespondencia = 1 and auxbienvalidadovehiculo.legible = 't' then  if auxdocumentobienvehiculo.idtipodocumento = 4 then  update nueva\_validacion.bien set idestadodocumentacion = 1 where idbien = auxdocumentobienvehiculo.idbien;  else  update nueva\_validacion.bien set idestadodocumentacion = 2 where idbien = auxdocumentobienvehiculo.idbien;  end if;  else  update nueva\_validacion.bien set idestadodocumentacion = 3 where idbien = auxdocumentobienvehiculo.idbien;  end if;  end if;  END LOOP;  END LOOP;  end if;  if vidclase = 4 then  FOR auxbienvalidadomaquinaria in select id, idbien, idcorrespondencia, adjunta, legible, iddocumentobien FROM nueva\_validacion.validacionxgestionmaquinaria WHERE id = vidvalidacion LOOP  select into estadodoc idestadodocumentacion from nueva\_validacion.bien WHERE idbien = auxbienvalidadomaquinaria.idbien;  FOR auxdocumentobienmaquinaria in select id as iddoc, idbien, idtipodocumento, validado FROM nueva\_validacion.documentobien where id = auxbienvalidadomaquinaria.iddocumentobien LOOP  if estadodoc > 1 then  if auxbienvalidadomaquinaria.adjunta = 't' and auxbienvalidadomaquinaria.idcorrespondencia = 1 and auxbienvalidadomaquinaria.legible = 't' then  if auxdocumentobienmaquinaria.idtipodocumento = 8 then  update nueva\_validacion.bien set idestadodocumentacion = 1 where idbien = auxdocumentobienmaquinaria.idbien;  else  update nueva\_validacion.bien set idestadodocumentacion = 2 where idbien = auxdocumentobienmaquinaria.idbien;  end if;  else  update nueva\_validacion.bien set idestadodocumentacion = 3 where idbien = auxdocumentobienmaquinaria.idbien;  end if;  end if;  END LOOP;  END LOOP;  end if;  if vidclase = 5 then  FOR auxbienvalidadoinmueblealquiler in select id, idbien, idcorrespondencia, adjunta, legible, iddocumentobien FROM nueva\_validacion.validacionxgestioninmueblealquiler WHERE id = vidvalidacion LOOP  select into estadodoc idestadodocumentacion from nueva\_validacion.bienalquiler WHERE idbien = auxbienvalidadoinmueblealquiler.idbien;  FOR auxdocumentobieninmueblealquiler in select id as iddoc, idbien, idtipodocumento, validado FROM nueva\_validacion.documentobienalquiler where id = auxbienvalidadoinmueblealquiler.iddocumentobien LOOP  if estadodoc > 1 then  if auxbienvalidadoinmueblealquiler.adjunta = 't' and auxbienvalidadoinmueblealquiler.idcorrespondencia = 0 and auxbienvalidadoinmueblealquiler.legible = 't' then  if auxdocumentobieninmueblealquiler.idtipodocumento = 23 then  update nueva\_validacion.bienalquiler set idestadodocumentacion = 1 where idbien = auxdocumentobieninmueblealquiler.idbien;  else  update nueva\_validacion.bienalquiler set idestadodocumentacion = 2 where idbien = auxdocumentobieninmueblealquiler.idbien;  end if;  else  update nueva\_validacion.bien set idestadodocumentacion = 3 where idbien = auxdocumentobieninmueblealquiler.idbien;  end if;  end if;  END LOOP;  END LOOP;  end if;  if vidclase = 6 then  FOR auxbienvalidadomaquinariapesada in select id, idbien, idcorrespondencia, adjunta, legible, iddocumentobien FROM nueva\_validacion.validacionxgestionmaquinariapesada WHERE id = vidvalidacion LOOP  select into estadodoc idestadodocumentacion from nueva\_validacion.bien WHERE idbien = auxbienvalidadomaquinariapesada.idbien;  FOR auxdocumentobienmaquinariapesada in select id as iddoc, idbien, idtipodocumento, validado FROM nueva\_validacion.documentobien where id = auxbienvalidadomaquinariapesada.iddocumentobien LOOP  if estadodoc > 1 then  if auxbienvalidadomaquinariapesada.adjunta = 't' and auxbienvalidadomaquinariapesada.idcorrespondencia = 0 and auxbienvalidadomaquinariapesada.legible = 't' then  if auxdocumentobienmaquinariapesada.idtipodocumento = 8 then  update nueva\_validacion.bien set idestadodocumentacion = 1 where idbien = auxdocumentobienmaquinariapesada.idbien;  else  update nueva\_validacion.bien set idestadodocumentacion = 2 where idbien = auxdocumentobienmaquinariapesada.idbien;  end if;  else  update nueva\_validacion.bien set idestadodocumentacion = 3 where idbien = auxdocumentobienmaquinariapesada.idbien;  end if;  end if;  END LOOP;  END LOOP;  end if;  if vidclase = 7 then  FOR auxbienvalidadovehiculoalquiler in select id, idbien, idcorrespondencia, adjunta, legible, iddocumentobien FROM nueva\_validacion.validacionxgestionvehiculoalquiler WHERE id = vidvalidacion LOOP  select into estadodoc idestadodocumentacion from nueva\_validacion.bienalquiler WHERE idbien = auxbienvalidadovehiculoalquiler.idbien;  FOR auxdocumentobienvehiculoalquiler in select id as iddoc, idbien, idtipodocumento, validado FROM nueva\_validacion.documentobienalquiler where id = auxbienvalidadovehiculoalquiler.iddocumentobien LOOP  if estadodoc > 1 then  if auxbienvalidadovehiculoalquiler.adjunta = 't' and auxbienvalidadovehiculoalquiler.idcorrespondencia = 1 and auxbienvalidadovehiculoalquiler.legible = 't' then  if auxdocumentobienvehiculoalquiler.idtipodocumento = 23 then  update nueva\_validacion.bienalquiler set idestadodocumentacion = 1 where idbien = auxdocumentobienvehiculoalquiler.idbien;  else  update nueva\_validacion.bienalquiler set idestadodocumentacion = 2 where idbien = auxdocumentobienvehiculoalquiler.idbien;  end if;  else  update nueva\_validacion.bien set idestadodocumentacion = 3 where idbien = auxdocumentobienvehiculoalquiler.idbien;  end if;  end if;  END LOOP;  END LOOP;  end if;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimeinto: filtradodocumento** | |
| **Nombre** | filtradodocumento |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.filtradodocumento () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxdocumento record;  declare auxdocumento1 record;  declare cantidad1 integer;  declare cantidad2 integer;  BEGIN  FOR auxbien in select id as idbien from dj\_activos.bien where fechainscripcion < '2014-11-30' LOOP  FOR auxdocumento in select id, idbien, idtipodocumento, gestion from nueva\_validacion.documentobien where gestion = 2014 and idbien = auxbien.idbien LOOP  FOR auxdocumento1 in select id, idbien, idtipodocumento, gestion from nueva\_validacion.documentobien where gestion < 2014 and idbien = auxbien.idbien LOOP  IF auxdocumento1.idtipodocumento = auxdocumento.idtipodocumento then  UPDATE nueva\_validacion.documentobien SET eliminado = true where id = auxdocumento1.id;  END IF;  END LOOP;  END LOOP;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: migrardocumentos** | |
| **Nombre** | migrardocumentos |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.migrardocumentos () RETURNS void AS  $BODY$  DECLARE  declare auxdocumento record;  BEGIN  FOR auxdocumento in select db.idbien, db.iddocumento, db.nro\_documento, db.usuario, db.fecha\_inscripcion, db.gestion from dj\_activos.bien b  join dj\_documento.documento\_bien db on db.idbien = b.id  where b.fechainscripcion < '2015-11-30' and db.gestion = 2016 LOOP  INSERT INTO nueva\_validacion.documentobien(idbien, idtipodocumento, nrodocumento, adicionado, idfuncionario, fechaadicion, gestion)  VALUES (auxdocumento.idbien, auxdocumento.iddocumento, auxdocumento.nro\_documento, false, auxdocumento.usuario, to\_char(auxdocumento.fecha\_inscripcion, 'YYYY-mm-dd')::date, auxdocumento.gestion);  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: migrardocumentos2** | |
| **Nombre** | migrardocumentos2 |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.migrardocumentos2 () RETURNS integer AS  $BODY$  DECLARE  declare auxdocumento record;  consulta4 record;  cantidad numeric;  BEGIN  cantidad := 0;  FOR auxdocumento in select db.idbien, db.iddocumento, db.nro\_documento, db.usuario\_dj, db.fecha\_inscripcion, db.gestion from dj\_activos.bien b  join dj\_documento.documento\_bien db on db.idbien = b.id  where b.fechainscripcion < '2015-11-30' and db.gestion = 2016 LOOP  --SELECT count(id)as cant FROM nueva\_validacion.documentobien WHERE idbien = auxdocumento.idbien into consulta4;  cantidad := auxdocumento.idbien; --consulta4.cant;  END LOOP;  return cantidad;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: terrenos** | |
| **Nombre** | terrenos |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.terrenos () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  BEGIN  /\*select count(\*), id from dj\_activos.bien  group by id  having count(\*) > 1  \*/  FOR auxbien in select id as idbien, idpadre from dj\_activos.bien where idpadre > 0 order by idbien LOOP  UPDATE nueva\_validacion.inmueble set padre = TRUE WHERE idbien = auxbien.idbien;  UPDATE dj\_activos.inmueble set padre = TRUE, idpadre = auxbien.idpadre WHERE idbien = auxbien.idbien;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

|  |  |
| --- | --- |
| **Procedimiento: terrenosfinal** | |
| **Nombre** | terrenosfinal |
| **Descripción** |  |
| **Código** | CREATE OR REPLACE FUNCTION nueva\_validacion.terrenosfinal () RETURNS void AS  $BODY$  DECLARE  declare auxbien record;  declare auxinmueble record;  BEGIN  UPDATE nueva\_validacion.inmueble set padre = FALSE;  UPDATE dj\_activos.inmueble set padre = FALSE;  FOR auxbien in select id as idbien, idpadre from dj\_activos.bien where idpadre > 0 order by idbien LOOP  UPDATE nueva\_validacion.inmueble set padre = TRUE WHERE idbien = auxbien.idpadre;  UPDATE dj\_activos.inmueble set padre = TRUE WHERE idbien = auxbien.idpadre;  END LOOP;  END;$BODY$  LANGUAGE 'plpgsql' VOLATILE |

1. **CONLUSIONES Y RECOMENDACIONES**

**<REDACTAR EN FUNCIÒN DE LOS OBJETIVOS ESPECIFICOS DEL SISTEMA>**

**UNA O MAS CONLUSIIONES/ RECOM POR CADA OBJETIVO**