



# **QUIPUX-COMUNIDAD**

## **Versión 4**

# **INSTALACIÓN Y CONFIGURACIÓN DE BASE DE DATOS Y SISTEMA QUIPUX**

**Secretaría de la Administración  
Pública**

Por  
José David Gamboa Vega

Autor: José David Gamboa Vega  
david.gamboa@administracionpublica.gob.ec  
josedavo@gmail.com

2014-2015

## ÍNDICE

1 INTRODUCCIÓN.....	3
2 ESTRUCTURA DE LA BASE DE DATOS.....	3
3 BASE DE DATOS TRANSACCIONAL.....	4
4 RELACIÓN DE TABLAS PARA DOCUMENTOS.....	5
5 PROCESO DE REGISTRO DE DOCUMENTOS.....	11
6 ESTRUCTURA DE LA BASE DE DATOS DOCUMENTAL.....	11
7 CREACIÓN DE TABLESPACE.....	13
8 INSTALACIÓN DE QUIPUX Y HERRAMIENTAS PARA PUESTA A PRODUCCIÓN.....	13
9 INSTALACIÓN DE PAQUETES NECESARIOS.....	14
10 INSTALACIÓN DE LA BASE DE DATOS DE QUIPUX.....	15
11 INSTALACIÓN DE PAQUETES NECESARIOS.....	17
12 CONFIGURACIÓN DEL SISTEMA HTML_A_PDF.....	18

## 1. INTRODUCCIÓN

La gestión documental en toda organización es el conjunto de normas técnicas y prácticas usadas para administrar el flujo de los documentos, permite la recuperación de la información de ellos, determinar el tiempo que deben guardarse, y “eliminar” aquellos que no sirven, aplicando principios de racionalización y economía.

Desde el año 2008 la Subsecretaría de Gobierno Electrónico, antes Subsecretaría de Informática, ha realizado modificaciones al Sistema de Gestión Documental Quipux, cuya versión original es Orfeo (Colombia), adaptándolo a los requerimientos de la gestión documental de las Instituciones Públicas. QUIPUX es un sistema informático con acceso web que la Subsecretaría de Gobierno Electrónico a través de la Secretaría Nacional de la Administración Pública, pone a disposición a las entidades o instituciones públicas para la gestión de documentos electrónicos, permite el registro, control, circulación y organización de los documentos digitales y/o físicos que se envían y reciben, estos documentos pueden ser internos (documentos generados en la institución entre las diferentes áreas), externos (documentos generados entre instituciones y ciudadanos).

### Carga de documentos

El Sistema de Gestión Documental Quipux tiene una carga aproximada de:

NRO INSTITUCIONES	317
NRO USUARIOS	1762482
NRO DOCUMENTOS GENERADOS	24416518

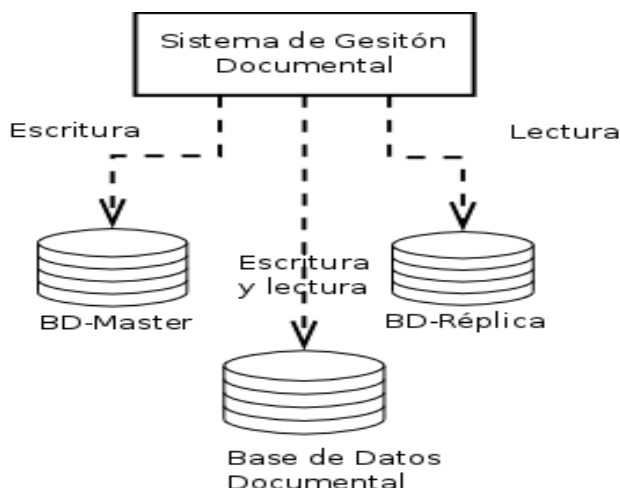
**Tabla 1.- Carga de la Base de datos Quipux**

## 2. ESTRUCTURA DE LA BASE DE DATOS

En la figura 1, podemos observar que existe 3 bases de datos, según la estructura de Quipux, tenemos 2 Bases de datos Principales BD-Master en donde se registra toda las transacciones que genera el Sistema, y una Base de datos Documental en donde se registra toda la documentación generada por Quipux, además almacena los anexos que se registran en el Sistema, los anexos

pueden ser pdf, xls, doc, la base de datos Réplica que puede ser una opción que se puede implementar dependiendo de la magnitud que tenga Quipux en dar su servicio.

En la nueva version de Quipux, la Base de datos trabaja bajo la siguiente estructura:



**Figura 1.- Estructura de Bases de datos de Quipux**

### 3. BASE DE DATOS TRANSACCIONAL

Como mencionamos anteriormente la base de datos transaccional se encarga de registrar toda la metadata de Quipux, por poner un ejemplo: Instituciones, Areas, Usuarios, Tipos de Documentos, etc.

A continuación describimos la base de datos transaccional con su estructura:

En Linux, el comando que nos ayuda a generar la documentación de una base de datos en postgres es el siguiente, antes de ejecutar el comando, instalamos el paquete de postgres *postgresql\_autodoc*.

```
postgresql_autodoc -d quipux_comunidad -f /carpeta_en_donde_guardar/ -h 127.0.0.1 -p 5432 -u postgres --password=postgres
```

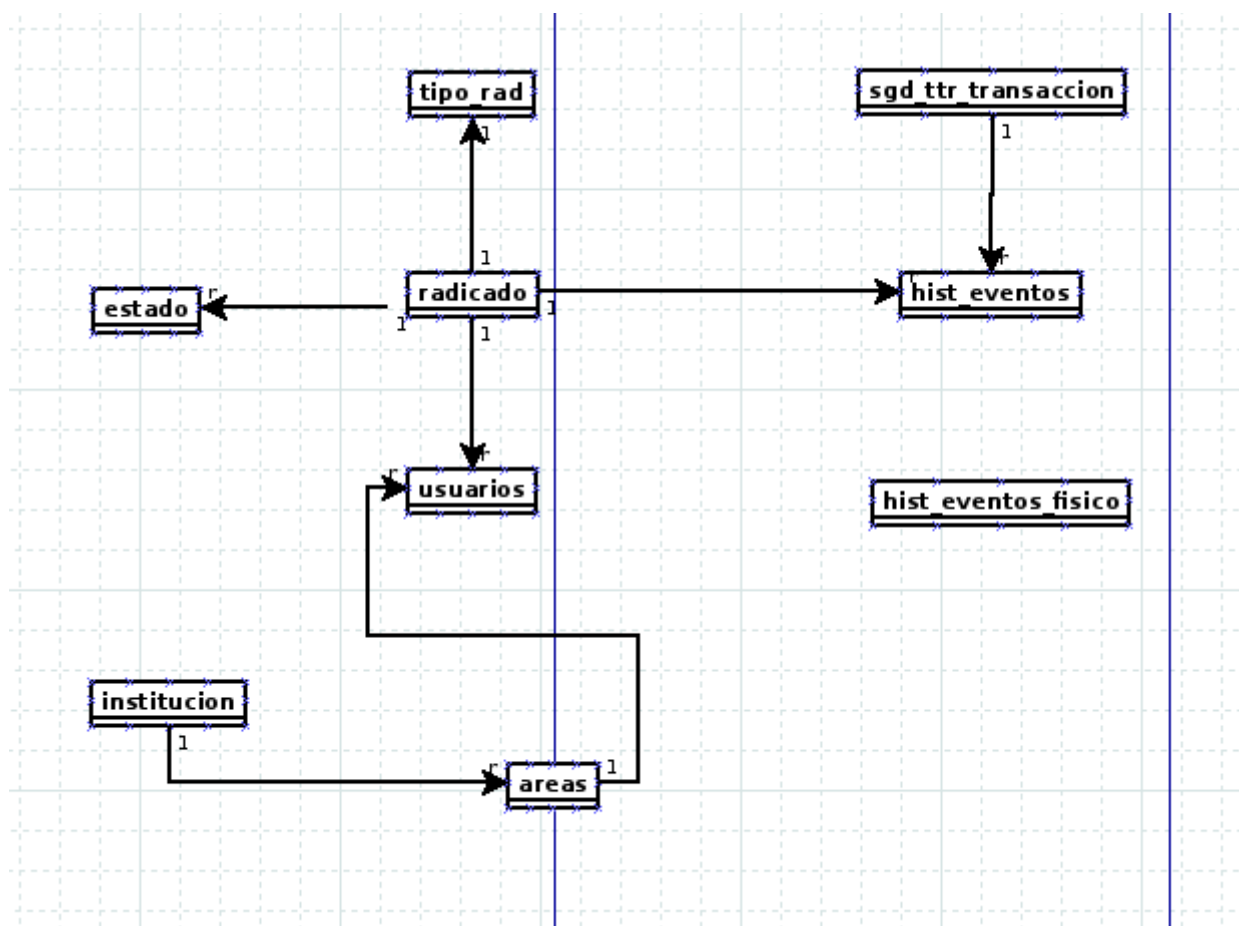
El comando genera los siguientes archivos:

```
-rw-r--r-- 1 root    root    884052 dic 10 12:25 .dia
-rw-r--r-- 1 root    root    151807 dic 10 12:25 .dot
-rw-r--r-- 1 root    root    343716 dic 10 12:25 .html
-rw-r--r-- 1 root    root     29006 dic 10 12:25 .neato
-rw-r--r-- 1 root    root    401473 dic 10 12:25 .xml
-rw-r--r-- 1 root    root    907512 dic 10 12:25 .zigzag.dia
```

**Figura 2.- Archivos generados (estructura de la base de datos Quipux)**

La documentación generada se encuentra creada dentro del repositorio de este documento.

#### 4. RELACIÓN DE TABLAS PARA DOCUMENTOS



**Figura 2.- Tablas que procesan la información de los documentos**

En la tabla radicado se registra la metadata de los documentos, entre la información principal que se registra podemos citar:

- Número del documento
- Tipo de documento
- Usuario Remitente
- Usuario Destinatario
- Usuarios con copia
- Estado del Documento
- Institución en donde se encuentra el documento
- código del documento físico que está registrado en la base de datos documental

**tabla 2.- radicado**

TABLA	TIPO, descripción
radi_num_radi	numeric(20,0) NOT NULL, -- Id del documento
radi_num_text	character varying(50), -- Número del documento según el formato definido en la institución
radi_num_temp	numeric(20,0) NOT NULL, -- Id del documento padre (desde el que se generan las copias para cada destinatario)
radi_fech_radi	timestamp with time zone NOT NULL, -- Fecha en la que se creó el documento
radi_fech_ofic	timestamp with time zone, -- Fecha en la que se firma y se envía el documento o fecha de referencia en el caso de documentos externos
radi_num_der	numeric(20,0), -- Id del documento al cual se encuentra asociado el documento actual (responder)
radi_path	character varying(150), -- Path donde se encuentra el archivo PDF en la bodega
esta_codi	smallint, -- Estado en el que se encuentra el documento
radi_usua_actu	integer, -- Id del usuario actual del documento
radi_fech_asig	timestamp with time zone, -- Fecha máxima para realización de trámite cuando se reasigna un documento
radi_leido	smallint DEFAULT 0, -- bandera que indica si el documento ya fue leído
radi_fech_agend	timestamp with time zone, -- Campo en desuso
radi_cca	character varying, -- Lista de usuarios para enviar copias, se separan por guiones (-id1--id2-)
radi_cuentai	character varying(50), -- Numero de referencia del documento
radi_asunto	character varying(350), -- Asunto del documento
radi_resumen	character varying(1000), -- Notas adicionales al documento
radi_desc_anex	character varying(100), -- Descripción general de los anexos
radi_flag_impr	smallint, -- Campo en desuso

radi_texto	integer, -- Id de la version del texto del documento que se está utilizando
radi_tipo	smallint, -- Tipo de documento (memo, oficio, etc.)
radi_usua_rem	character varying, -- Lista de usuarios remitentes del documento; se separan por guiones (-id1--id2-)
radi_usua_ante	integer, -- Id del usuario anterior del documento
radi_usua_dest	character varying, -- Lista de usuarios destinatarios del documento; se separan por guiones (-id1--id2-)
radi_usua_radi	integer, -- Id del usuario que registro el documento
radi_permiso	smallint DEFAULT 0, -- nivel de seguridad del documento (publico o confidencial)
radi_nomb_usua_firma	character varying, -- Datos de la firma electrónica del documento
radi_fech_firma	timestamp with time zone, -- Fecha en que se firmó electrónicamente el documento (cuando se validó en quipux)
radi_inst_actu	integer, -- Institucion actual del documento
radi_archivo	smallint DEFAULT 0, -- Indica si el documento se encuentra archivado físicamente
usar_plantilla	integer DEFAULT 0, -- Bandera que indica si el documento se generará con una plantilla o en una hoja en blanco
ajust_texto	integer DEFAULT 100, -- Determina si el archivo se comprime o se expande (Tamaño de letra)
radi_tipo_impresion	character varying(1) DEFAULT 1, -- Opciones de impresión - Modo de impresión de los datos del destinatario (combo)
radi_lista_dest	character varying, -- Listado de las listas de usuarios seleccionadas para el envío de los documentos
radi_tipo_archivo	smallint DEFAULT 0, -- Define si el archivo de la imagen del documento (almacenado en radi_path) es temporal (generada por el sistema y no firmada) o definitiva.
cod_codi	bigint DEFAULT 0, -- Id de la codificación del documento (tipificación) cat_codi bigint DEFAULT 0, -- Id de la categoría del documento
radi_ocultar_recorrido	smallint DEFAULT 0, -- Indica si se ocultará el recorrido del documento
radi_usua_redirigido	bigint DEFAULT 0, -- Id del usuario al que se redirigirá el documento (registro de documentos externos)
radi_text_temp	character varying(50), -- Número temporal del documento que se le asigno mientras estaba en elaboración
radi_ume_asoc	numeric(20,0), -- Id del documento antecedente (documentos asociados)
arch_codi	bigint DEFAULT 0, -- Código del archivo almacenado en la BDD de documentos
arch_codi_firma	bigint DEFAULT 0, -- Código del archivo firmado electrónicamente, almacenado en la BDD de documentos
radi_imagen	character varying(50), -- Código del anexo cargado como imagen digitalizada

En la tabla hist\_eventos se registra todas las acciones que se realiza sobre un documento, puede ser un comentario, reasignación, creación, informar entre otros.

**tabla 3.- hist\_eventos**

TABLA	TIPO, descripción
hist_fech	timestamp with time zone NOT NULL, -- Fecha de la transacción
usua_codi_ori	integer NOT NULL, -- Usuario que realizó la transacción
radi_num_radi	numeric(20,0) NOT NULL, -- Id del documento
hist_obse	character varying(600) NOT NULL, -- Observaciones usua_codi_dest integer, -- Código del usuario destino, en caso que la transacción involucre a más de un usuario
sgd_ttr_codigo	smallint, -- Id de la transacción hist_codi bigint NOT NULL DEFAULT nextval('hist_eventos_hist_codi_seq'::regclass), -- Id
hist_referencia	character varying(50),

La tabla sgd\_ttr\_transaccion, es un catálogo que está relacionada con la tabla hist\_eventos que permite registrar las acciones sobre un documento.

**tabla 4.- sgd\_ttr\_transaccion**

TABLA	TIPO, descripción
sgd_ttr_codigo	smallint NOT NULL, -- Id de la transacción
sgd_ttr_descrip	character varying(100) NOT NULL, -- Detalle de la transacción

El estado del documento tiene los siguientes tipos:

- En Edición
- En Tramite
- No Enviado (Electrónicamente)
- No Enviado (Manualmente)
- Enviado
- Eliminado
- Archivado
- No Enviado (Original)
- Eliminado Total
- Pendientes ciudadanos

**tabla 5- estado**

TABLA	TIPO, descripción
-------	-------------------



esta_codi	smallint NOT NULL, -- ESTA_CODI
esta_desc	character varying(100) NOT NULL, -- ESTA_DESC

Los usuarios, esta tabla nos permite registrar a los usuarios que usan Quipux, para centralizar la información con instituciones y dependencias (áreas), tenemos una vista materializada la cual se encarga de registrar en la tabla usuario toda la información de la persona: Institucion, area, usuario.

**tabla 6.- usuarios**

usua_login	character varying(50), -- Login del usuario (deben comenzar con "U"); existen usuarios especiales que comienzan con 'UUSR' y 'UADM'
usua_pasw	character varying(35), -- Contraseña del usuario en md5
usua_nomb	character varying(200), -- Nombre del usuario
usua_cedula	character varying(50), -- Número de cédula
usua_email	character varying(500), -- Email, pueden ser varios separados por comas
usua_titulo	character varying(100), -- Tratamiento o título académico
usua_abr_titulo	character varying(30), -- Abreviacion del titulo
usua_esta	smallint DEFAULT 1, -- Estado del usuario, activo o inactivo
usua_codi	integer NOT NULL DEFAULT nextval('usuarios_usua_codi_seq'::regclass), -- Id del usuario cargo_tipo smallint DEFAULT 0, -- 0 normal 1 jefe 2 asistente
depe_codi	integer, -- Área a la que pertenece el usuario
usua_nuevo	smallint DEFAULT 1, -- Determina si el usuario ya cambió su clave del sistema o si se debe enviar el email para cambio de clave
usua_tipo	smallint DEFAULT 2, -- si el usuario es interno o externo
usua_cargo	character varying(200), -- Cargo del usuario
inst_codi	integer, -- Institución a la que pertenece el usuario
usua_apellido	character varying(200), -- Apellido del usuario
cargo_id	integer, -- Campo en desuso

usua_obs	text, -- Observaciones sobre el usuario
ciu_codi	integer, -- Id de la ciudad a la que pertenece el usuario
usua_genero	character(1),
usua_firma_path	character varying, -- Path en el que se encuentra la imagen escaneada de la firma
usua_direccion	character varying, -- Dirección domiciliaria
usua_telefono	character varying, -- Número telefónico
usua_codi_actualiza	integer, -- Id del usuario que realizó la ultima modificación de los datos
usua_fecha_actualiza	timestamp with time zone, -- Fecha en la que se realizó la última modificación de los datos
usua_obs_actualiza	character varying, -- Cambios realizados durante la última modificación del usuario
usua_cargo_cabecera	character varying(200), -- Cargo que se muestra cuando se selecciona al usuario como destinatario
usua_sumilla	character varying(50), -- Iniciales del usuario utilizadas cuando este tiene responsabilidad en la elaboración de un documento
usua_responsable_area	integer DEFAULT 0, -- Indica que el usuario es responsable del area, razón por la cual la inicial de sus sumilla se...
inst_nombre	character varying(200), -- Nombre de la institución a la que pertenece el usuario
usua_tipo_certificado	smallint DEFAULT 0, -- Id del tipo de certificado digital que posee
visible_sub	integer DEFAULT 1, -- Indica si el usuario ha sido subrogado
usua_subrogado	integer, -- Id del usuario subrogado
usua_celular	character varying, -- No. del teléfono celular
tipo_identificacion	integer DEFAULT 0, -- 0 cedula 1 pasaporte

La tabla institucion nos permite registrar las instituciones que usan Quipux.

#### **tabla 7.- institucion**

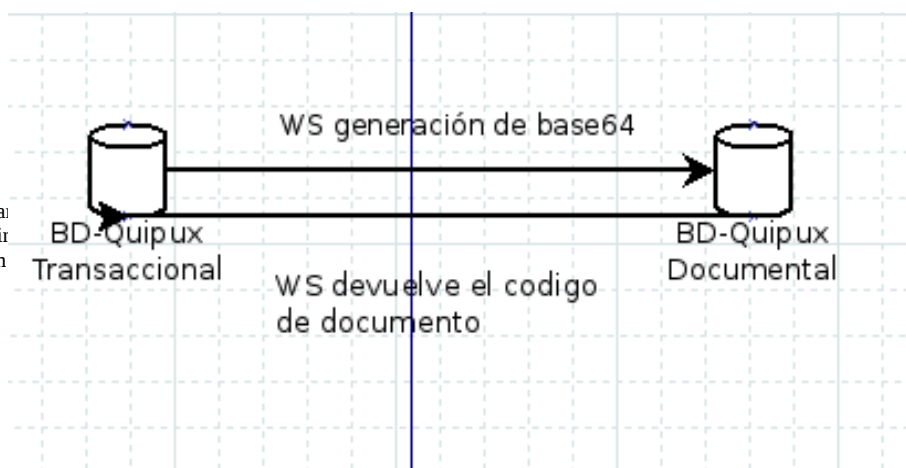
<b>TABLA</b>	<b>TIPO, descripción</b>
inst_ruc	character varying(14), -- RUC de la Institución
inst_nombre	character varying(200), -- Nombre de la institución

inst_logo	character varying(100), -- Path donde se encuentra la imagen con el logo institucional
inst_sigla	character varying(10), -- Siglas de la institución
inst_pie1	character varying(150), -- Campo en desuso
inst_pie2	character varying(150), -- Campo en desuso
inst_pie3	character varying(150), -- Campo en desuso
inst_codi	integer NOT NULL, -- Id
inst_estado	integer, -- Estado, activa o inactiva
inst_coordinador	smallint DEFAULT 0, -- Id del ministerio coordinador
inst_telefono	character varying(30), -- Número telefónico
inst_despedida_ofi	character varying, -- Frase de despedida por defecto que saldrá en los documentos (Ejm: Dios, Patria y Libertad)
inst_email	character varying(50), -- email para soporte institucional
inst_ws_wsdl	character varying(500),
inst_ws_usuario	character varying(100),
inst_ws_contrasena	character varying(100),

## 5. PROCESO DE REGISTRO DE DOCUMENTOS

El proceso de registro de documentos se realiza de la siguiente manera:

1.- Se registra la metadata del documento en la tabla radicado de la base de datos Transaccional, el contenido del documento, por cuestiones de seguridad no se describe el registro del documento en este documento; Luego de esto se ejecuta un servicio web el cual se encarga de enviar el documento en base64 hasta la base de datos Documental la cual ejecuta la función `func_grabar_archivo`, la cual permite grabar el archivo en la base de datos documental, luego de esto, la misma función se encarga de enviar el código del archivo a la base de datos transaccional.



**Figura 3.- Estructura de Bases de datos de Quipux**

## **6. ESTRUCTURA DE LA BASE DE DATOS DOCUMENTAL**

La base de datos documental es el repositorio central de la documentación de Quipux, esta base de datos está manejada bajo la administración manual de creación de repositorios con tablespace y en la administración del sistema se asigna a que tablespace se registrará los documentos.

En la tabla archivo se registra la metadata del documento, el tamaño, el archivo en md5, es necesario describir el campo indi\_codi que indica el índice que hace referencia al tablespace de la base de datos en donde se encuentra almacenado físicamente.

**tabla 8.- archivo**

<b>TABLA</b>	<b>TIPO, descripción</b>
arch_codi	bigint NOT NULL,
nombre	character varying(500),
fecha_creacion	timestamp with time zone,
tamano	bigint,
arch_md5	character(32),
indi_codi	integer,
estado	smallint DEFAULT 1,

La tabla índice permite registrar los tablespace creados, los cuales se registra mediante el sistema, el sistema lee los tablespace de la base de datos documental y se crea el registro en la administración del sistema, esta tarea se realiza a través del Superadministrador del Sistema.

**tabla 9.- índice**

<b>TABLA</b>	<b>TIPO, descripción</b>
--------------	--------------------------

indi_codi integer NOT NULL,	integer NOT NULL, -- Id, código de la tabla
arch_codi_inicio	bigint DEFAULT 0, codigo del archivo donde inicia el registro de los documentos
arch_codi_fin	bigint DEFAULT 0, codigo del archivo donde finaliza el registro de los documentos
tamano	bigint DEFAULT 0, Tamaño del documento
tamano_maximo	bigint DEFAULT 2097152,
esta_codi	smallint DEFAULT 0, estado del índice del tablespace asignado
nombre_tabla	character varying(100), nombre del tablespaces
nombre_tablespace	character varying(100), nombre físico del tablespaces
fecha_creacion	timestamp with time zone DEFAULT now(), fecha de creacion
fecha_activacion	fecha de activación, timestamp with time zone,
fecha_cierre	fecha de cierre del tablespaces, timestamp with time zone,
usua_codi_crea	integer, usuario quien crea el tablespace en el sistema
usua_codi_activa	integer, usuario que activa el tablespace en el sistema
usua_codi_cierra	integer, usuario quien cierra el tablespace en el sistema

La tabla archivo\_xx es en donde se registra los documentos, los cuales a su vez están en el repositorio que ubicamos en la tabla antes descrita archivo con el campo. indi\_codi

## tabla 10.- archivo\_xx

TABLA	TIPO, descripción
arch_codi	bigint NOT NULL, id de la tabla
archivo	character varying, archivo en base 65 de quipux

## 7. CREACIÓN DE TABLESPACE

La base de datos documental de Quipux utiliza tablespace para almacenar los documentos, a continuación se procede a una breve explicación sobre la creación de tablespace en un mismo servidor:

Dependiendo de la distribución de Linux en nuestro caso tenemos FEDORA, ingresamos a la carpeta:

En donde está la data de postgres: `/var/lib/pgsql/data`

procedemos a crear el tablespace

```
[root@localhost data]# sudo mkdir documentos_1
```

procedemos a dar los permisos necesarios a la carpeta de tablespace

```
[root@localhost data]# sudo chown -R postgres: documentos_1
```

Luego de crear físicamente, nos vamos a postgres a su gusto, y procedemos a crear el tablespace en la base de datos, el comando que ejecutaremos será el siguiente:

```
CREATE TABLESPACE documentos_1  
OWNER postgres  
LOCATION '/var/lib/pgsql/data/documentos_1';
```

## **8. INSTALACIÓN DE QUIPUX Y HERRAMIENTAS PARA PUESTA A PRODUCCIÓN**

Quipux trabaja con el motor de base de datos POSTGRESQL, muy robusta y con miles de desarrolladores en todo el mundo.

Para la instalación del Postgres como servidor se ejecuta el siguiente comando por consola linux como root:

```
yum -y install postgresql-server postgresql-lib postgresql-contrib
```

Para iniciar el servicio de Postgres tenemos:

```
/etc/init.d/postgresql restart
```

Para administrar la base de datos postgres es necesario cambiar la contraseña por defecto la cual nos permite con el siguiente comando:

```
su postgres  
psql  
alter user postgres with password 'postgres';
```

Listo, con los comandos anteriores tenemos listo la base de datos para administrar, lo siguiente que debemos hacer es dar los permisos necesarios de accesos a Postgres (necesario).

1.- Ingresamos a la instalación de base de datos:

Ingresamos al archivo y editamos: **vim /var/lib/pgsql/data/postgresql.conf**

aquí configuramos lo siguiente:

buscamos la línea: `listen_adress='*'` que por defecto se encuentra comentada, esta tenemos que descomentarla.

2.- Permisos por Ip a la base de datos:

Ingresamos al archivo y editamos: **`vim /var/lib/pgsql/data/pg_hba.conf`**

En la parte de accesos agregamos nuestra ip o la ip del servidor que se conectara con Postgres, si están en el mismo servidor la aplicación y la base de datos (no recomendable): agregamos la siguiente línea:

**`host all all 127.0.0.1/24 md5`**

## **9. INSTALACIÓN DE PAQUETES NECESARIOS**

Como sabemos, el Quipux funciona en lenguaje PHP y con servidor Apache, existen algunas funcionalidades de Quipux en donde se requiere algunas librerías, con el fin de tener todo actualizado recomiendo el siguiente comando a ejecutar.

Los comandos de este documento, están bajo la distribución de Fedora 20, los comandos cambian dependiendo de la distribución de Linux del servidor.

```
yum --enablerepo=remi install httpd php php-common  
yum --enablerepo=remi install php-pecl-apc php-cli php-pear php-pdo php-mysqlnd  
php-pgsql php-pecl-mongo php-sqlite php-pecl-memcache php-pecl-memcached php-gd  
php-mbstring php-mcrypt php-xml  
yum --enablerepo=remi install php-soap
```

Iniciamos los servicios instalados, y desde luego configuramos para que cuando se reinicie el servidor que se suban por sí solos, lo cual es recomendable.

```
chkconfig --level 2345 httpd on  
chkconfig --level 2345 postgresql on
```

## **10. INSTALACIÓN DE LA BASE DE DATOS DE QUIPUX**

Instalamos en nuestra pc el pgadminIII para ejecutar los script para realizar de manera más cómoda.

**Base de datos Transaccional:**

```
CREATE DATABASE quipux_comunidad  
WITH OWNER = postgres  
ENCODING = 'UTF8'  
TABLESPACE = pg_default  
LC_COLLATE = 'es_EC.UTF-8'  
LC_CTYPE = 'es_EC.UTF-8'  
CONNECTION LIMIT = -1;
```

### Base de datos Documental:

```
CREATE DATABASE quipux_combodega
WITH OWNER = postgres
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'es_EC.UTF-8'
LC_CTYPE = 'es_EC.UTF-8'
CONNECTION LIMIT = -1;
```

### ejecutamos el script ubicado en:

/home/usuario/QUIPUX\_COMUNIDAD/quipux\_comunidad\_v4/

### Ejecutar en este orden:

- 1.- quipux.sql
- 2.- quipux\_bodega.sql
- 3.- particionamiento\_log\_acceso.sql
- 4.- La siguiente función se debe ejecutar dentro de la base de datos documental:

### tabla 11.- script base de datos

#### **FUNCIÓN GUARDAR ARCHIVOS DE QUIPUX, BASE DE DATOS DOCUMENTAL**

```
CREATE OR REPLACE FUNCTION func_grabar_archivo(var_nombre_archivo text, var_archivo_base_64 text)
RETURNS integer AS
$BODY$
DECLARE
    var_sql text;
    var_md5 text;
    var_tamano numeric;
    var_arch_codi bigint;
    var_nombre_tabla text;
    var_indi_codi integer;
    var_recordset record;
    arr_tablas text[];
    var_num_tablas integer := 0;
    var_num_tabla_rr integer;
BEGIN
    BEGIN
        var_md5 := md5(var_archivo_base_64);
        SELECT arch_codi from archivo where arch_md5=var_md5 and estado=1 limit 1 INTO var_recordset;
        IF var_recordset is not null THEN return var_recordset.arch_codi; END IF;

        FOR var_recordset IN select indi_codi, nombre_tabla from indice where esta_codi = 2 order by indi_codi asc LOOP
            IF arr_tablas is null THEN
```



```

arr_tablas := ARRAY[[var_recordset.indi_codi::text, var_recordset.nombre_tabla]];

ELSE
  arr_tablas := array_cat(arr_tablas, ARRAY[var_recordset.indi_codi::text, var_recordset.nombre_tabla]);
END IF;
var_num_tablas := var_num_tablas + 1;
END LOOP;
IF var_num_tablas=0 THEN return 0; END IF;

var_tamano := length(var_archivo_base_64)/8*6;
var_arch_codi := nextval('sec_archivo'::regclass);

-- Calculamos la tabla en la que se va a insertar el registro (tipo round robin) y validamos que esté activa
var_num_tabla_rr := (var_arch_codi % var_num_tablas) + 1;
IF NOT func_validar_bloqueo_tabla(arr_tablas[var_num_tabla_rr][2]) THEN
  var_num_tabla_rr := 1;
  WHILE var_num_tabla_rr <= var_num_tablas and NOT func_validar_bloqueo_tabla(arr_tablas[var_num_tabla_rr][2]) LOOP
    var_num_tabla_rr := var_num_tabla_rr + 1;
  END LOOP;
END IF;
IF trim(arr_tablas[var_num_tabla_rr][2]) is null THEN return 0; END IF;
var_nombre_tabla := arr_tablas[var_num_tabla_rr][2];
var_indi_codi := arr_tablas[var_num_tabla_rr][1];

var_sql := 'INSERT INTO archivo (arch_codi, indi_codi, nombre, fecha_creacion, tamano, arch_md5)
VALUES ('||var_arch_codi::text||', '||var_indi_codi::text||', '
||quote_literal(var_nombre_archivo)||', now(), '
||var_tamano::text||', '||quote_literal(var_md5)||')';
EXECUTE var_sql;

var_sql := 'INSERT INTO '||var_nombre_tabla||' (arch_codi, archivo)
VALUES ('||var_arch_codi::text||', '||quote_literal(var_archivo_base_64)||')';
EXECUTE var_sql;

return var_arch_codi;
EXCEPTION WHEN OTHERS THEN
  PERFORM func_log_archivo (var_sql, SQLERRM);
return 0;
END;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION func_grabar_archivo(text, text)
OWNER TO postgres;

```

## 11. INSTALACIÓN DE PAQUETES NECESARIOS

En la carpeta QUIPUX\_COMUNIDAD encontramos el sistema en la carpeta quipux\_comunidad\_v4 la carpeta del sistema es quipux, esta debemos copiar en la carpeta del servidor /var/www/html/ el comando que permite ejecutar esta acción es el siguiente:

**cp -R /home/usuario/QUIPUX\_COMUNIDAD/quipux\_comunidad\_v4/quipux /var/www/html**

además de Quipux, necesitamos instalar la aplicación html\_a\_pdf la cual nos permite generar los documentos en el servidor.

necesitamos ejecutar el siguiente comando:

**cp -R**

**/home/usuario/QUIPUX\_COMUNIDAD/quipux\_comunidad\_v4/html\_a\_pdf /var/www/html**

## **CONFIGURACIÓN**

En la raíz de la carpeta del sistema quipux, tenemos el archivo de configuración config2.php del cual debe ser creado un archivo llamado config.php, de esta manera:

**cp quipux/config2.php quipux/config.php**

Editamos el archivo config.php y realizamos el cambio en las siguientes variables:

### **Configuración de la base de datos Transaccional**

```
// Configuración de la conexión con la BDD
$usuario = "postgres";
$contrasena = "postgres";
$servidor = "127.0.0.1:5432";
$driver = "postgres";
$db = "quipux_comunidad";
```

### **Configuración de la base de datos Documental**

```
$usuario_bodega = "postgres";
$contrasena_bodega = "postgres";
$servidor_bodega = "127.0.0.1:5432";
$db_bodega = "quipux_combodega";
```

Si deseamos podemos cambiar los nombres de las bases de datos a su gusto.

## **12. CONFIGURACIÓN DEL SISTEMA HTML\_A\_PDF**

De la misma manera el sistema html\_a\_pdf está desarrollado en php, lo cual hemos explicado el montaje de un servidor apache, y solo procederemos a configurar el sistema.

Ingresamos a la carpeta html\_a\_pdf y escribimos el siguiente comando:

**cp html\_a\_pdf/config2.php html\_a\_pdf/config.php**

El config.php editamos en el servidor donde está alojado.

```
$nombre_servidor = "http://IP_SERVIDOR_PDF/html_a_pdf";  
$tipo_sistema = "Produccion";
```

El siguiente paso es la comunicación, entonces configuramos en el sistema html\_a\_pdf el archivo:  
**html\_a\_pdf.wsdl**

Sólo editaremos la parte que nos corresponde a la interconexión del Servicio Web.

```
<service name="soapapihtml_a_pdfService">  
  <port name="soapapihtml_a_pdfPort" binding="typens:soapapihtml_a_pdfBinding">  
    <soap:address location="http://IP_SERVIDOR_PDF/html_a_pdf/html_a_pdf.php"/>  
  </port>  
</service>
```

Luego, necesitamos que el sistema Quipux se conecte con el sistema html\_a\_pdf por servicio web, para esto nos vamos al sistema Quipux y editamos el archivo config.php la variable servidor\_pdf.

```
$servidor_pdf = "http://IP_SERVIDOR__QUIPUX/html_a_pdf";
```

Algo importante a destacar, que en la explicación anterior tenemos en servidores diferentes los sistemas Quipux y html\_a\_pdf, si tenemos en un mismo servidor sería en resumen lo siguiente:

## **CONFIGURACIÓN SISTEMA HTML\_A\_PDF**

```
$nombre_servidor = "http://IP_SERVIDOR_PDF/html_a_pdf";  
$tipo_sistema = "Produccion";
```

## **CONFIGURACIÓN SERVICIO WEB EN SERVIDOR HTML\_A\_PDF**

```
<service name="soapapihtml_a_pdfService">  
  <port name="soapapihtml_a_pdfPort" binding="typens:soapapihtml_a_pdfBinding">  
    <soap:address location="http://IP_SERVIDOR_PDF/html_a_pdf/html_a_pdf.php"/>  
  </port>  
</service>
```

## **CONFIGURACIÓN SISTEMA HTML\_A\_PDF EN EL SISTEMA QUIPUX**

`$servidor_pdf = "http://IP_SERVIDOR__QUIPUX/html_a_pdf";`

## **CONFIGURACIONES ADICIONALES DE SERVIDORES**

Verificar las siguientes variables en el php.ini

`register_globals = Off`

`short_open_tag = On`

`display_errors = Off`

`upload_max_filesize = 2M (DEPENDIENDO DE LA CAPACIDAD DE CADA SERVIDOR)`

## **INSTALACIÓN EL PAQUETE HTML2PS**

`yum install html2ps`