

# **CODDNS**

## **RESOLUCIÓN DINÁMICA DE NOMBRES DE DOMINIO**

**Proyecto Final ASIR**

Fco de Borja Sánchez Soto

IES Tierno Galván (Parla)

Junio de 2015

# INTRODUCCIÓN

Los sistemas de configuración automática facilitan la conexión con el exterior, uniéndonos a la nube masiva de dispositivos que conforman Internet.



Muy sencillo, con **etiquetas** que siempre apunten en la **dirección correcta**.

# CODDNS

## RESOLUCIÓN DINÁMICA DE NOMBRES DE DOMINIO

### MOTIVACIÓN

Supongamos que queremos, ya sea por necesidad o por ocio, alguna de las siguientes posibilidades:

- Mostrar una página web o un blog.
- Guardar o recoger archivos de mi ordenador personal.
- Ver contenido multimedia.
- Imprimir documentos en mi impresora desde cualquier sitio.
- Poner la calefacción de mi domicilio mientras estoy yendo hacia él.
- Ver cómo está la casa cuando me voy de vacaciones

Hay multitud de aplicaciones y servicios que proveen esas posibilidades, y cada vez son más sencillas de configurar. Pero siempre tendremos el mismo problema: Lo puedo utilizar cuando estoy en casa, pero... **¿y cuando no lo esté? ¿Qué pasa cuando intento acceder desde el exterior?**

### CUBRIENDO UNA NECESIDAD

Todo equipo conectado a Internet recibe una dirección IP para comunicarse con el resto de equipos de la red.

Para conectar con nuestro equipo solo tenemos que conocer dicha dirección, lo que plantea dos problemas con la dirección IP:

1. Es un conjunto de números. Siempre es más sencillo recordar una cadena de texto.
2. No hay suficientes direcciones (en la versión 4) para que todos los equipos se conecten de manera simultánea, por lo que se estableció desde las operadoras de red, que la asignación de direcciones sería dinámica. Esto quiere decir que ya no nos vale con recordar la dirección, sino que tenemos que estar pendientes de las nuevas direcciones que se vayan asignando.

**Como conclusión:** Necesitamos algún sistema que **asocie una IP dinámica a una etiqueta de texto** fácil de recordar.

## EXPONIENDO NUESTRA RED

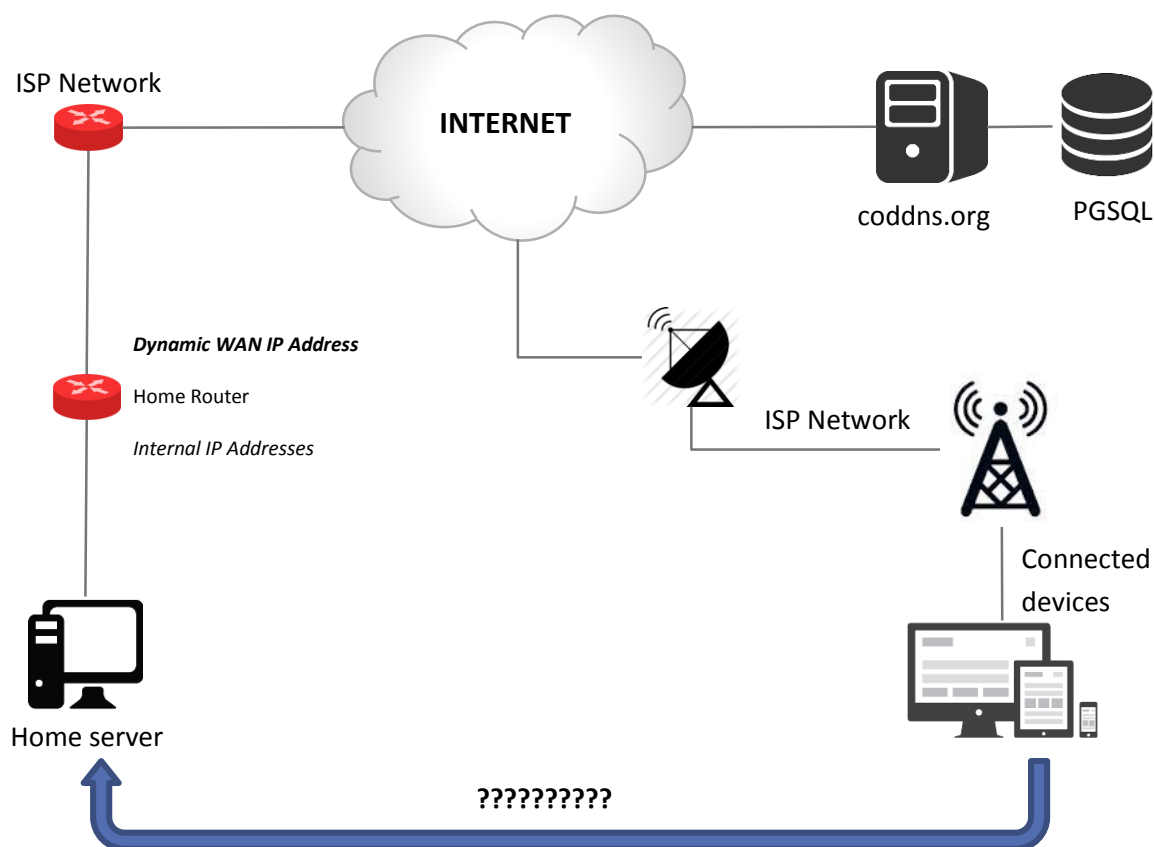
Tenemos que tener en cuenta que todas las soluciones que se pueden aportar para conectar con nuestra red, requieren una configuración avanzada de nuestro punto de acceso a internet.

Esto es, necesitamos tener claro qué servicios queremos ofrecer al exterior (tanto de uso privado como público) y configurar nuestro *router* para que redireccione las solicitudes recibidas contra nuestro servidor interno.

¿Cómo lo hago? Es el famoso “cómo abrir los puertos de mi router” de tantos artículos de Internet.

## ESQUEMA DE FUNCIONAMIENTO DE CODDNS

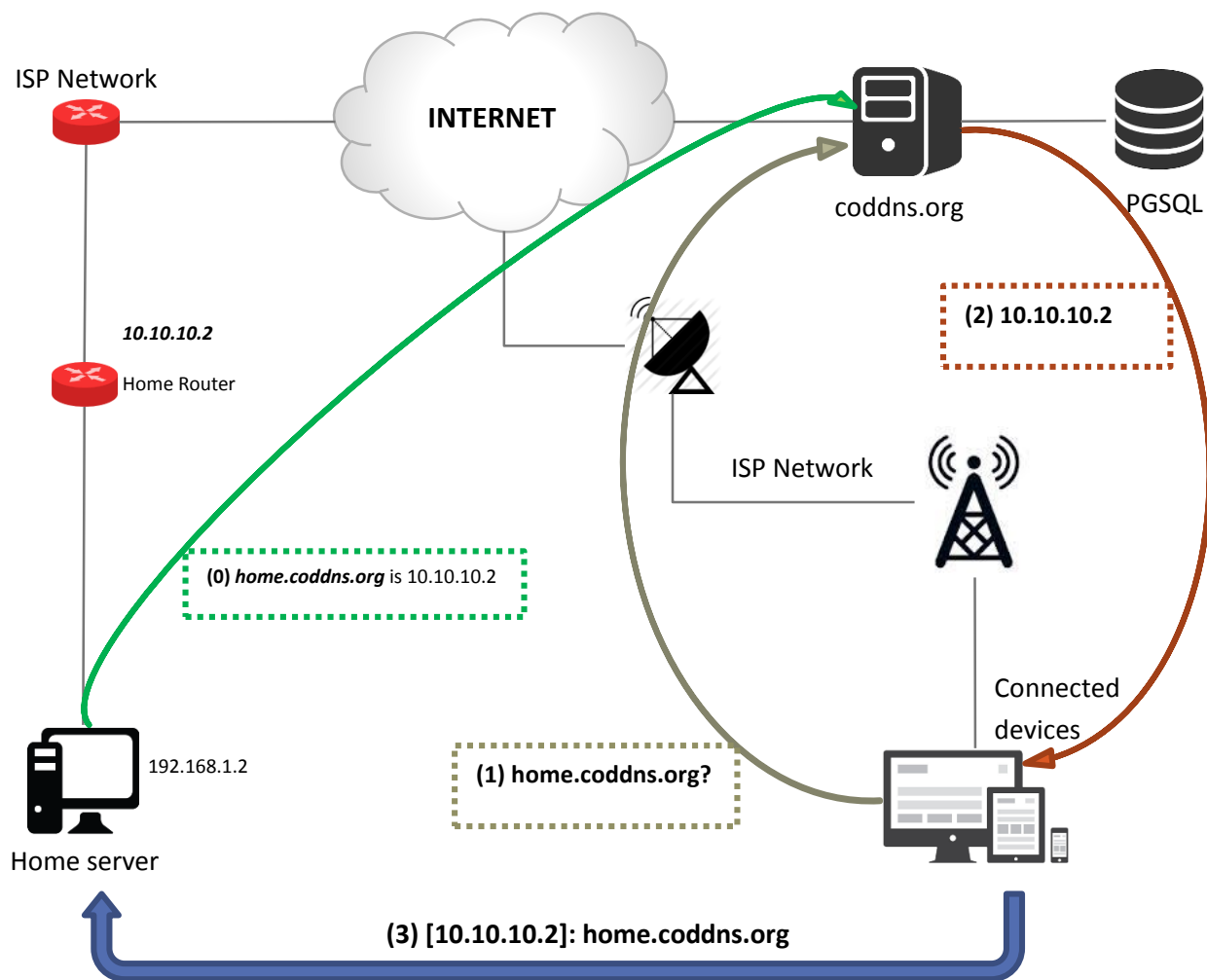
En el esquema planteado a continuación vemos claramente que el problema reside en encontrar el camino a nuestra red privada.



La solución propuesta es utilizar *coddns.org* para identificar nuestra red en Internet asociando a la dirección IP pública de nuestro router la etiqueta que elijamos.

Será *coddns.org* quien se encargue de realizar las traducciones para los equipos de internet.

## (Continuación) Esquema de funcionamiento CODDNS



- (0) Registramos nuestro servidor de casa (Home server) en coddns.org mediante un usuario, contraseña y una etiqueta de nuestra elección (en este ejemplo *home.coddns.org*).
  - a. En el momento del registro, coddns.org recogerá automáticamente nuestra dirección IP pública (WAN IP Address) para realizar la asociación entre etiqueta e IP.
  - b. Posteriormente será nuestro propio servidor, a través de la utilidad *ddns\_updater*, el que comunique al servidor los cambios en su IP pública.
- (1) El equipo conectado a Internet que desee acceder a nuestra red, realizará una consulta contra su servidor DNS preguntando por la dirección IP asociada a la etiqueta que hayamos elegido (en el ejemplo *home.coddns.org*)
- (2) El servidor DNS reenviará esa solicitud contra coddns.org, que resolverá la consulta respondiendo con la IP que el servidor le facilitó en su momento.
- (3) Accedemos a nuestra red a través de la dirección facilitada por el sistema coddns.

# CODDNS

## RESOLUCIÓN DINÁMICA DE NOMBRES DE DOMINIO

### OBJETIVO

Nuestro objetivo principal es garantizar la resolución de la IP pública del cliente sea cual sea esta, independientemente de que sea variable o estática.

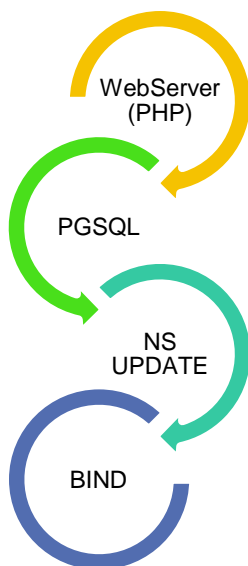
### ¿CÓMO LLEVARLO A CABO?

Para ofrecer el servicio coddns necesitaremos los siguientes elementos:

- Un servidor en Internet con una IP estática (o dinámica resolviendo de un sistema coddns primario)
- Servidor basado en Linux (ya sea basado en Debian o RHL)
- Servicio de resolución de nombres de dominio Bind
- Servicio de páginas web (apache, nginx, etc.)
- Servicio de aplicaciones PHP
- Servicio de base de datos (recomendado PostgreSQL)

No tiene por qué estar en el mismo servidor, puede estar en otro diferente, aunque por motivos de seguridad es conveniente que nunca esté expuesto.

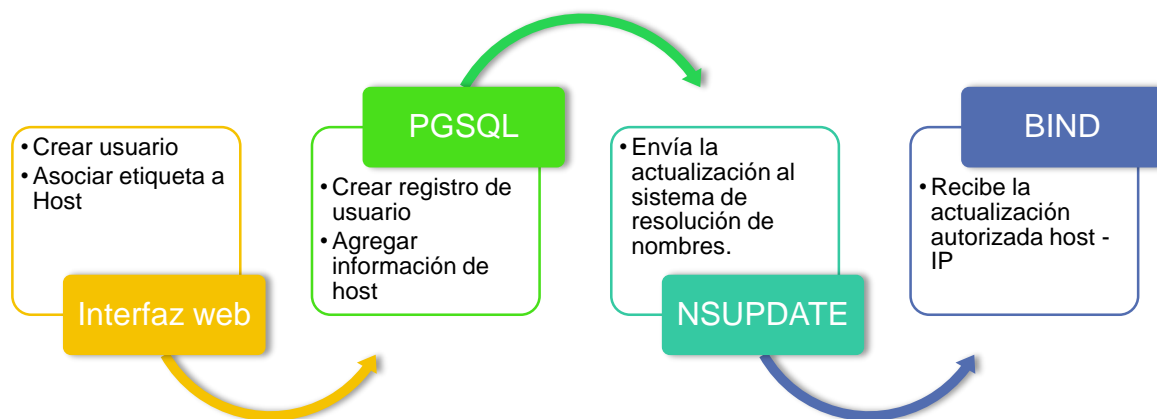
- Utilidades de envío de correo (notificaciones, cambios en las cuentas, etc.)



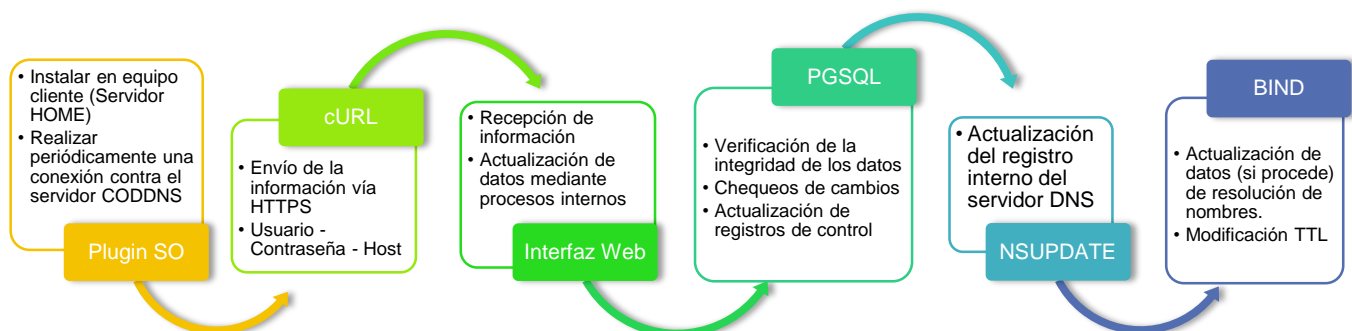
El servidor web proporcionará la interfaz para agregar la información inicial del cuarteto [usuario, contraseña, host, IP]

Esa información se almacenará en la base de datos y se agregará el registro correspondiente en el sistema de resolución de nombres DNS.

## ANÁLISIS: CREACIÓN DE NUEVO REGISTRO



## ANÁLISIS: ACTUALIZACIÓN DINÁMICA DE REGISTRO EXISTENTE



## ANÁLISIS: ELIMINACIÓN DE REGISTRO EXISTENTE





# DISEÑO BASADO EN CAPAS

## Capa de presentación

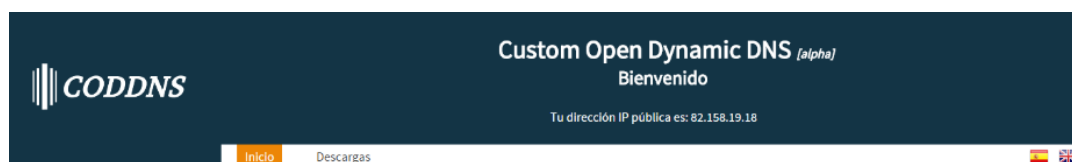
La interfaz basada en HTML5 y CSS3 es generada por los ficheros PHP. Se ha dispuesto de componentes visuales divididos en procesamiento cliente y servidor.

Se compone de las siguientes páginas:

### Interfaz pública

Cada una de las páginas contiene tres partes:

- una **cabecera**, en la que mostramos una interfaz simple con una el título del proyecto, un subtítulo mostrando la IP actual origen del cliente, y un panel de navegación que nos indica en qué sección de la aplicación estamos y permite cambiar el idioma de muestra al usuario.



- Un **cuerpo** cambiante en función de la página en que nos encontremos:
  - o Página principal. Contiene el mensaje de bienvenida y descripción general de las capacidades de la aplicación. También presenta formularios de acceso o de registro y formularios de recuperación de credenciales.

¡Hola!

¿Necesitas acceso a tu **servidor privado** en casa? ¿o quieres ver las cámaras de vigilancia de tu domicilio? Es posible que prefieras **acceder a tus contenidos multimedia desde cualquier sitio** sin tener que cargar con un disco duro extraíble o contratar una IP estática, o tener que confirmar tu dirección de correo cada mes para mantener el servicio...

¡Para eso estamos aquí!

Con CODDNS tendrás siempre una etiqueta a través de la cual **podrás acceder a la red de tu casa**, sin tener que estar preocupandote de los cambios en la IP del router. Simplemente asocia una etiqueta disponible a tu dirección IP, instala el actualizador y accede a tu equipo desde cualquier parte de Internet.

Bienvenidos

---

**Acceder**

|  |   |
|--|---|
| E-mail:                                | <input type="text" value="correo electrónico"/> |
| Password:                              | <input type="password" value="password"/>       |
| <a href="#">¿Olvidó su contraseña?</a> | <input type="button" value="Enviar"/>           |

**Registrarme**

|                    |  |
|--------------------|--|
| E-mail:            | <input type="text" value="correo electrónico"/>    |
| Password:          | <input type="password" value="password"/>          |
| Confirma password: | <input type="password" value="confirma password"/> |
|                    | <input type="button" value="Enviar"/>              |



- Página de recuperación de contraseñas: Nos permitirá recuperar una contraseña perdida mediante el envío de un token de seguridad con tiempo de caducidad al correo electrónico del usuario:

#### Contraseña olvidada

Si has olvidado tu contraseña, introduce tu dirección de correo a continuación.  
Se te enviará un código con un enlace para que modifiques tu contraseña

correo electrónico

enviar

- Página de descargas, con enlaces a las aplicaciones cliente que los usuarios deberán instalar para mantener la información del servidor actualizada.

#### Descargas

##### Linux

Puedes descargar el cliente de actualización de DNS dinámico para Linux de [aquí](#)

##### Windows

Puedes descargar el instalador del cliente de actualización de DNS dinámico para Windows de [aquí](#)

[Volver](#)

- Un **pie**, en el que mostramos los enlaces a las políticas de privacidad y protección de datos, las condiciones de uso, así como un enlace al perfil público del autor. (**Nota:** En vez del autor, una vez presentado, se mostrará un enlace al repositorio GIT del proyecto).

Contactar/Políticas

Contactar/Políticas



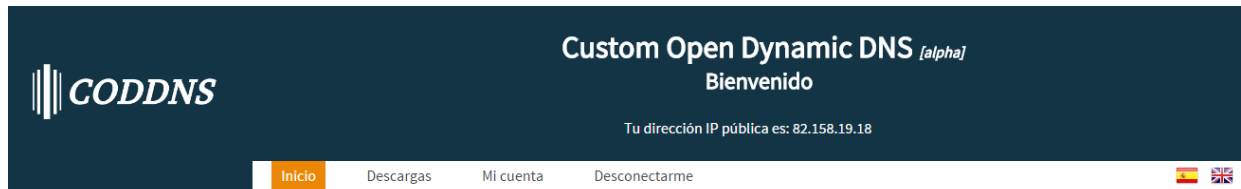
Política de cookies

Condiciones de uso

## Interfaz privada

Una vez hemos nos hemos autenticado en la página, encontraremos ciertos cambios en la interfaz.

Dispondremos de nuevas opciones en la cabecera, como la edición de los datos de acceso y la opción de desconexión.



El cuerpo de la página principal pasará a tener un listado con los hosts que vayamos registrando, y provee una interfaz para agregarlos:

**Gestor de etiquetas**

Aquí puedes agregar nuevas etiquetas para tus dispositivos (hosts) o administrar las existentes. Recuerda que la responsabilidad sobre los contenidos que abras a Internet es sólo tuya.

Etiqueta:

Dirección IP:

**Mis hosts**

| Nombre de host        | IP             | Ops. |
|-----------------------|----------------|------|
| admin.coddns.org      | 217.160.143.23 |      |
| main.coddns.org       | 217.160.143.23 |      |
| prueba.coddns.org     | 62.174.24.75   |      |
| padres.coddns.org     | 87.221.203.210 |      |
| porritas.coddns.org   | 217.160.143.23 |      |
| subnetting.coddns.org | 217.160.143.23 |      |

La capa visual se genera en el lado del servidor, transfiriéndose al cliente junto con las herramientas para comunicar las respuestas sin necesidad de recargar la página completamente.

Por ejemplo, en las comprobaciones previas a la hora de agregar un host a la lista de nombres de dominio resueltos incluyen un **chequeo de disponibilidad** de etiqueta vía AJAX:

Etiqueta:

Disponibile

Dirección IP:

Una vez agregado el host, podemos modificar la IP a la que direcciona en la página (actualizando directamente el registro en el DNS) a través de la página generada al pulsar el botón “editar” en operaciones.

[Volver](#)

Host: admin.coddns.org

IP actual: 217.160.143.23













Nueva IP:

[Coger mi IP actual](#)

Actualizar

También podemos proceder con la eliminación de la etiqueta pulsando el icono rojo, en la columna de operaciones sobre hosts, en la página principal:

Mis hosts

| Nombre de host        | IP             | Ops.  |
|-----------------------|----------------|---|
| admin.coddns.org      | 217.160.143.23 |       |
| main.coddns.org       | 217.160.143.23 |       |
| prueba.coddns.org     | 62.174.24.75   |       |
| padres.coddns.org     | 87.221.203.210 |       |
| porritas.coddns.org   | 217.160.143.23 |   |
| subnetting.coddns.org | 217.160.143.23 |   |

## Capa lógica

El componente PHP relaciona la capa de presentación con el sistema, controlando las órdenes que se emiten contra el servidor de nombres de dominio, el sistema base y la base de datos.

### Librerías incluidas:

#### **Ipv4.php**

Provee al sistema de la función `_ip()` la cual devuelve la dirección IP (v4) más probable de un equipo cliente.

#### **Responsive.php**

Incluye las funciones:

`check_user_agent`

Indica el tipo de navegador que está usando un cliente

`isOverHTTPS`

Indica si el cliente está navegando usando el protocolo de navegación segura HTTPS

#### **Pgclient.php**

Define la clase PgClient, que utilizaremos para interactuar con la capa de datos. Dispone de los siguientes métodos:

`<Constructor>`

Construye un nuevo objeto de tipo PgClient con los datos de conexión configurados en la propia clase.

`connect`

Conecta el objeto con la capa de datos, devuelve el socket a través del cual se realizarán las comunicaciones

`exeq`

Conecta el objeto con la capa de datos, devuelve el socket a través del cual se realizarán las comunicaciones

`Lq_serror`

Devuelve el error (en caso de haberse producido) correspondiente a la última consulta ejecutada.

`Lq_nresults`

Devuelve el número de resultados producidos por la última consulta ejecutada.

`disconnect`

Desconecta el socket previamente abierto con método `connect`.

`date_checker`

Comprueba si una fecha recibida es correcta.

`datetime_checker`

Transforma una fecha recibida vía formulario (Chrome // Firefox // Otros) en un formato local interpretable por el objeto.

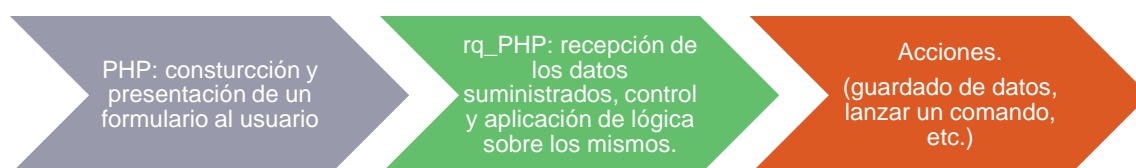
*prepare*

Uno de los métodos más importantes, permite controlar todo lo que el usuario nos remite de la forma que sea, filtrando consultas maliciosas, y preparando el contenido para el tipo de dato esperado.

*decode*

El almacenamiento en base de datos de cadenas de texto se realiza codificada en formato URL. Para la devolución y muestra de los datos, se decodifica este valor para una muestra correcta al usuario.

El resto de componentes PHP están contruidos según el siguiente esquema de trabajo:



A la hora de presentar la página principal, se realiza una comprobación de argumentos recibidos en index.php de la siguiente manera:

```
//...
<section id="main_section">
<?php

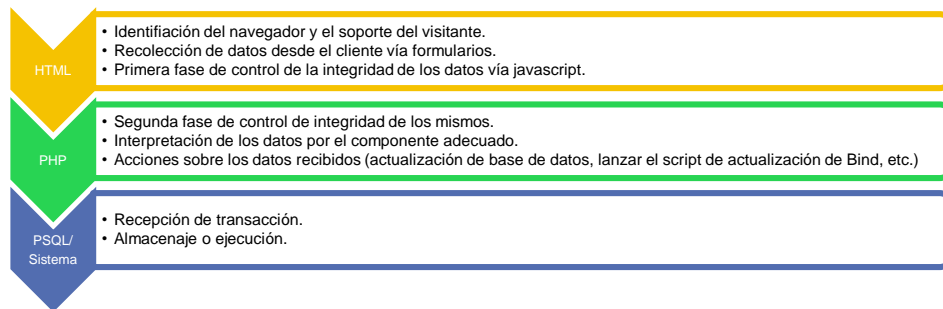
if (! isset ($_GET["z"]))
    include ("main.php");
else {
    switch ($_GET["z"]){
        case "hosts":
            include ("usr/hosts.php");
            break;
        case "mod":
            include ("usr/modhost.php");
            break;
        case "del":
            include ("usr/delhost.php");
            break;
        case "remember":
            include ("usr/remember.php");
            break;
        case "newpassword":
            include ("usr/newpass.php");
            break;
        case "downloads":
            include ("downloads.php");
            break;
        case "usermod":
            include ("usr/user_actions.php");
            break;
        default:
            include ("main.php");
            break;
    }
}

?>
</section>
//...
```

En esta muestra del código podemos observar la forma de mostrar al usuario cada una de las secciones de la página.

Cada uno de los ficheros contiene sus propios chequeos en cuanto a sesiones activas, qué se debe mostrar y qué no, filtrados específicos, etc.

En todo proceso de recogida de datos aplicamos el siguiente procedimiento:



## LISTADO DE COMPONENTES DESPLEGADOS Y SU UTILIDAD:

```
$HTML_ROOT_DIR..... Directorio contenedor de documentos HTML.
├── cliupdate.php..... Escucha a ddns_updater (cliente) para la actualización del DNS.
├── cpolicy.html..... Política de cookies.
├── downloads
│   ├── ddns_linux.tar.gz..... Cliente ddns_updater Linux.
│   ├── ddns_windows.zip..... Cliente ddns_updater Windows.
│   └── index.html
├── downloads.php..... Muestra las descargas al usuario (ampliable a gestor interno).
├── err40X.html..... Redirección de páginas de error.
├── favicon.ico
├── include..... Directorio para configuración.
│   └── config.php..... Configuración del sistema CODDNS.
├── index.php..... Núcleo de la interfaz. Alterna vistas por GET.
├── ip.php..... Servicio REST, indica la IP del cliente.
├── lib..... Directorio de librerías.
│   ├── ipv4.php..... Librería IPv4.
│   ├── pgclient.php..... Librería PgClient.
│   └── responsive.php..... Librería Responsive.
├── logout.php..... Desconecta a un usuario.
├── main.php..... Muestra la pantalla principal dinámicamente según el usuario.
├── rest_host.php..... Indica si una etiqueta está disponible o no.
├── rs..... Directorio de recursos.
│   ├── css
│   │   ├── m.css..... CSS estilo para dispositivos móviles.
│   │   └── pc.css..... CSS estilo para equipos con pantallas grandes.
│   ├── img..... Directorio de imágenes.
│   └── js
│       └── util.js..... Librería de utilidades JS // AJAX // Efectos.
├── terms.html..... Condiciones de uso del sistema.
├── usr..... Directorio de gestión.
│   ├── delhost.php..... Componente de eliminación de host.
│   ├── hosts.php..... Componente de listado y creación de hosts y opciones.
│   ├── modhost.php..... Formulario de modificación de hosts.
│   ├── newpass.php..... Formulario para nueva contraseña (con token).
│   ├── remember.php..... Formulario para nuevo token (sin sesión activa).
│   ├── rq_login.php..... Componente de identificación de usuario.
│   ├── rq_modhost.php..... Componente de recepción: modificación de host.
│   ├── rq_newpass.php..... Componente de recepción: modificación de contraseña.
│   ├── rq_nhost.php..... Componente de recepción: creación de nuevo host.
│   ├── rq_signin.php..... Componente de creación de cuentas de usuario.
│   ├── rq_ua.php..... Componente de mod. de contraseña (con sesión activa)
│   ├── sendtoken.php..... Componente de asignación de token para cuenta inaccesible.
│   └── user_actions.php..... Formulario para nueva contraseña (con sesión activa).
├── /opt..... Directorio contenedor de scripts.
│   ├── ddns
│   │   ├── ddns_updater.sh..... Copia del actualizador ddns_updater para clientes.
│   │   └── dnsmgr.sh..... Script de actualización de bind9 (nuevo, eliminar, actualizar).
│   └── srv
│       └── maintenance.sh..... Script de mantenimiento. (Copias de seguridad PSQL).
├── /usr/share/backups..... Directorio contenedor de backups de la base de datos.
├── /etc
├── zones..... Directorio contenedor de las zonas DNS.
│   ├── db.coddns.org..... Fichero con la información de zona.
│   └── db.coddns.org.jnl..... Fichero jnl con la nueva información de zona.
```

## SCRIPTS

### DNS manager

```
#!/bin/bash
#--- Fco de Borja Sanchez
#-----
tmp_file="ddns_operation_`date +%d%m%Y%H%M%S`_"$RANDOM
server="127.0.0.1"
TTL="8640"
KEY="/share/ddns/rndc.key"

# $1 - host name
# $2 - mode
# $3 - IP
prepare_addRow(){
    echo "server "$server > /tmp/$tmp_file
    echo "update add "$1" "$TTL" "$2" "$3 >> /tmp/$tmp_file
    echo "send" >> /tmp/$tmp_file
    echo "quit" >> /tmp/$tmp_file
}

# $1 - host name
# $2 - host type
prepare_deleteRow(){
    echo "server "$server > /tmp/$tmp_file
    echo "update delete "$1" "$2 >> /tmp/$tmp_file
    echo "send" >> /tmp/$tmp_file
    echo "quit" >> /tmp/$tmp_file
}

launch(){
    nsupdate -k $KEY < /tmp/$tmp_file 2>&1
}

#----- MAIN --
if [ $# -lt 3 ] || [ $# -gt 4 ]; then
    echo "nARGS ERR"
    exit 1;
fi

#-- $1 [a|d]
#-- $2 [hostname]
#-- $3 [type]
#-- $4 [ip]

case $1 in
    "a")
        prepare_addRow $2 $3 $4
        launch
        ;;
    "d")
        prepare_deleteRow $2 $3
        launch
        ;;
    *)
        echo "ARGS MALFORMED"
        exit 2;
    ;;
esac
```

Este script se encarga de mantener la zona *coddns.org* actualizada. Se invoca desde el PHP de manera dinámica cada vez que un usuario crea/ modifica/ elimina un registro:

```
/* Nuevo host. Fichero: usr/rq_nhost.php */
$out = shell_exec("dnsmgr a " . $host . " A " . $ip);

/* Eliminación de host. Fichero: usr/delhost.php */
$out = shell_exec("dnsmgr d " . $host . " A");

/* Actualización de datos de host. Fichero: usr/rq_modhost.php */
// -- erase
$out = shell_exec("dnsmgr d " . $host . " A");
// -- add
$out = shell_exec("dnsmgr a " . $host . " A " . $ip);
```



## Maintenance

```
#!/bin/bash
# Realiza copias de seguridad de la base
# de datos.
# Autor: Fco de Borja Sanchez
#-----
path="/usr/share/backups"

if [[ `whoami` != "postgres" ]]; then
    echo "no soy el usuario correcto "`whoami`
    exit 1
fi

cd $path
pg_dump h123_ddnsp > $path/pgdump_coddns.sql
tar -czf `date +%Y%m%d%H%M`_coddns.tar.gz pgdump_coddns.sql 2>&1 > /dev/null
rm pgdump_coddns.sql 2>&1 > /dev/null

exit 0
```

Script de mantenimiento, realiza volcados de seguridad de la base de datos.

Queda programado en */etc/crontab*:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed

31 2,6,10,14,18,22 * * * root /opt/ddns/ddns_updater.sh > /dev/null 2>&1
10 3 * * 0 postgres /opt/srv/maintenance.sh > /dev/null 2>&1
```

## SCRIPTS CLIENTE

### Linux ddns\_updater

```
#!/bin/bash
# Script de actualizacion de datos IP de cliente
# Programar al inicio del sistema / cada 4h
# Autor: Fco de Borja Sanchez
#-----

# datos

ddnsconf=`cat /etc/passwd | grep $USER | head -1 | cut -f6 -d':'`"/.userdata"
global_ddnsconf="/usr/share/ddns/userdata"
dest="http://coddns.org/cliupdate.php"
release=""
installer=""

# funciones
check_release(){
    if [ "`cat /etc/*-release |grep NAME | head -1 | cut -f 2 -d'"'"'" = "Fedora" ]; then
        release="fedora"
        installer="yum"
    elif [ "`cat /etc/*-release |grep DISTRIB | head -1 | cut -f 2 -d'"'"'" = "Ubuntu" ]; then
        release="ubuntu"
        installer="apt-get"
    elif [ "`cat /etc/*-release |grep NAME | tail -1 | cut -f 2 -d'"'"'" = "\"Debian GNU/Linux\"" ]; then
        release="debian"
        installer="apt-get"
    else
        #read -p "introduzca el comando que utiliza para instalar paquetes: " installer
        echo " [ERROR]: Sistema no soportado"
        exit 4;
    fi
}

#-----
# MAIN
#-----
# comprobaciones previas

curl --version > /dev/null 2>&1
if [ $? != 0 ]; then
    echo "cURL es necesario para la correcta ejecucion del script."
    check_release
    echo " Instalando... usando $installer"
    (yes | $installer install curl) > /dev/null 2>&1
    if [ $? != 0 ]; then
        echo " [ERROR]: Error al instalar curl. Lance este script con permisos de super usuario"
        exit 1;
    else
        echo " instalado correctamente."
    fi
fi

if [ ! -f $ddnsconf ]; then
    ddnsconf=$global_ddnsconf
elif [ ! -f $ddnsconf ]; then
    echo " [ERROR]: datos de conexion no encontrados, por favor, reinstale"
    exit 2;
fi

# inicio solicitudes
usuario=`cat $ddnsconf | grep usuario | cut -f 2 -d':'`
password=`cat $ddnsconf | grep password | cut -f 2 -d':'`
host=`cat $ddnsconf | grep host | cut -f 2 -d':'`

echo " datos de conexion: "
echo " "$usuario"/"$password
echo " HOST: "$host
echo " conectando con servidor ddns"
r=`curl --data 'u='$usuario'&p='$password'&h='$host $dest 2>/dev/null`
if [ $? -eq 0 ]; then
    echo " conexion completada con exito"
    echo " Recibido mensaje: "$r
else
    echo " [ERROR]: error en la conexion"
    exit 3;
fi
```

Emite una orden de actualización contra **cliupdater.php** (en el servidor de CODDNS), vía cURL, manteniendo el registro correspondiente actualizado en el sistema DNS.

## Windows ddns\_updater

```
' ddns_updater
' Author: Fco de Borja Sanchez
' DD: 01/06/2015
'
' Updates the data in http://coddns.org with the detected IP and
' the data contained in the configuration file passed as first argument
'-----

Function HTTPPost(sUrl, sRequest)
    set o = CreateObject("Microsoft.XMLHTTP")
    o.open "POST", sUrl,false
    o.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
    o.setRequestHeader "Content-Length", Len(sRequest)
    o.send sRequest
    HTTPPost = o.responseText
End Function

Function remove_blanks (line)
    sLast=""
    sInit=line
    Do Until sLast = sInit
        sLast = sInit
        sInit = Replace(sInit, " ", "")
    Loop
    remove_blanks=sLast
End Function

Function ReadConfiguration (config_path)
    Set objFS = CreateObject("Scripting.FileSystemObject")
    If Not objFS.FileExists(config_path) Then
        WScript.Quit 2
    End If

    Set objFile = objFS.OpenTextFile(config_path, 1)
    user = ""
    pass = ""
    host = ""

    Do Until objFile.AtEndOfStream
        line = remove_blanks(objFile.ReadLine)

        If Left(line,InStr(line,":")-1) = "usuario" Then
            user = "u=" & Mid(line,InStr(line,":")+1, Len(line))
        ElseIf Left(line,InStr(line,":")-1) = "password" Then
            pass = "p=" & Base64Encode(Mid(line,InStr(line,":")+1, Len(line)))
        ElseIf Left(line,InStr(line,":")-1) = "host" Then
            host = "h=" & Mid(line,InStr(line,":")+1, Len(line))
        End If
    Loop
    objFile.Close

    If ( Len(user) = 0 OR Len(pass)=0 OR Len (host) = 0 ) Then
        WScript.Quit 3
    End If

    ReadConfiguration = user & "&" & pass & "&" & host
End Function

' MAIN PROCEDURAL

if WScript.Arguments.Count = 0 then
    WScript.Echo "Missing parameters"
    WScript.Quit 1
end if

sUrl = "http://coddns.org/cliupdate.php"
sRequest = ReadConfiguration (WScript.Arguments(0))
response = HTTPPost (sUrl, sRequest)
WScript.echo response

If Left(response,InStr(response,":")-1) = "ERR" then
    WScript.Quit 4
End If
WScript.Quit 0
```

Se utiliza la funcionalidad Base64Encode, obtenida de <http://stackoverflow.com> de la contribución del usuario Antonin Foller.

Realiza las mismas acciones que el script bash de Linux, pero en Visual Basic Script de Windows.

Requiere un fichero de configuración con el contenido siguiente:

```
usuario:ejemplo@coddns.org  
password:su_password  
host:ejemplo.coddns.org
```

Utilizaremos SHTASKS para programar la tarea que actualizará el registro en el servidor CODDNS

```
> SHTASKS /Create /RU SYSTEM /SC HOURLY /TN "DDNS_UPDATER" /TR "cscript /B  
C:\CODDNS\ddns\ddns_updater.vbs C:\CODDNS\ddns\ddns_userdata.conf"
```

## Capa de datos

Almacenamos los datos en PostgreSQL 9.X utilizando la librería desarrollada en PHP.

Con la base de datos llevaremos un control de los usuarios que tienen acceso al sistema, así como de las etiquetas que vayan registrando y los cambios en estas.

Solo se guarda información de la última IP registrada **no se lleva un control sobre un histórico de IP**, con esto ahorramos espacio, aumentamos el rendimiento del sistema y protegemos la privacidad del usuario.

La estructura donde se almacena la información es extremadamente simple y se compone de únicamente dos tablas:

### Usuarios

| Column  | Type                     | Table "public.usuarios" | Modifiers   | Storage  | Description |
|---|--------------------------|-------------------------|---|----------|-------------|
| id  | bigint                   |                         | not null default nextval('usuarios_id_seq'::regclass) | plain    |             |
| mail  | character varying(250)   |                         | not null  | extended |             |
| pass  | text                     |                         | not null  | extended |             |
| last_login  | timestamp with time zone |                         |   | plain    |             |
| first_login   | timestamp with time zone |                         | default now()   | plain    |             |
| ip_last_login   | inet                     |                         |   | main     |             |
| ip_first_login  | inet                     |                         |   | main     |             |
| hash  | text                     |                         |   | extended |             |
| max_time_valid_hash   | timestamp with time zone |                         |   | plain    |             |
| Indexes:  |                          |                         |   |          |             |
| "pkey_usuarios" PRIMARY KEY, btree (id)   |                          |                         |   |          |             |
| "const_usuarios_unique_mail" UNIQUE CONSTRAINT, btree (mail)                                    |                          |                         |   |          |             |
| "usuarios_hash_key" UNIQUE CONSTRAINT, btree (hash)   |                          |                         |   |          |             |
| Referenced by:  |                          |                         |   |          |             |
| TABLE "hosts" CONSTRAINT "fkey_host_owner" FOREIGN KEY (oid) REFERENCES usuarios(id) MATCH FULL |                          |                         |   |          |             |
| Has OIDs: no  |                          |                         |   |          |             |

Tenemos, para almacenar la información, los campos:

- id (identificador numérico único para cada registro).
- la dirección de correo electrónico almacenada en el campo "mail", la contraseña codificada vía hash con algoritmo SHA y semilla.
- un registro del último acceso al sistema almacenando IP y marca de tiempo del primer y último acceso.
- Un código hash para la recuperación de contraseña en caso de pérdida.
- Una marca de tiempo de validez máxima para el token para cambio de contraseña.

## Hosts

| Table "public.hosts" |                          |  |  |  |          |             |
|----------------------|--------------------------|--|--|--|----------|-------------|
| Column               | Type                     | Modifiers  |  |  | Storage  | Description |
| id                   | bigint                   | not null default nextval('hosts_id_seq'::regclass) |  |  | plain    |             |
| oid                  | bigint                   | not null   |  |  | plain    |             |
| tag                  | character varying(200)   | not null   |  |  | extended |             |
| ip                   | inet                     |  |  |  | main     |             |
| created              | timestamp with time zone | default now()                                      |  |  | plain    |             |
| last_updated         | timestamp with time zone | default now()                                      |  |  | plain    |             |

Indexes:

"pkey\_host" PRIMARY KEY, btree (id)

"const\_hosts\_unique\_tag" UNIQUE CONSTRAINT, btree (tag)

Foreign-key constraints:

"fkey\_host\_owner" FOREIGN KEY (oid) REFERENCES usuarios(id) MATCH FULL

Has OIDs: no

Para gestionar las etiquetas que los usuarios vayan agregando al sistema, y mantenerlo actualizado, se lleva un registro de cada una de ellas, su propietario, cuándo se crearon y la marca de tiempo de la última actualización.

- **id** (identificador numérico único para cada registro).
- **Oid** (*owner identifier*) identificador de propietario, coincidirá con el identificador del usuario almacenado en la tabla *usuarios*.
- **Tag**: almacenará la cadena de texto con la que el usuario desea identificar su IP en la red.
- **IP**: obviamente también se almacenará la última IP registrada. Este campo se utiliza para su gestión en la interfaz del sistema y como dato de control en el proceso de actualización de los registros de DNS.
- **Last\_updated**: Una marca de tiempo de la última actualización del registro.

La descrita es la información básica de cómo trabaja CODDNS. Puede encontrar más información en el fichero **coddns\_pgsql.sql** que contiene la estructura completa para preparar un entorno SQL limpio.

# PROCESO DE DESPLIEGUE EN UN SERVIDOR LINUX

El sistema CODDNS es compatible con cualquier distribución Linux.

## Pasos a seguir (Ejemplo CentOS)

### INSTALACIÓN DE BIND9

Agregar los repositorios EPEL para CentOS

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
yum install -y epel-release-6-8.noarch.rpm
yum makecache
```

Instalación de bind9 y utilidades:

```
yum install bind bind-utils
```

Creación de claves para actualización dinámica:

```
# Crear una clave rndc.key y almacenarla en /share/ddns/
$ mkdir -p /share/ddns
$ rndc-confgen |head -5 > /share/ddns/rndc.key
$ chown -R apache:root /share
$ chmod 755 -R /share
$ chmod 644 /share/ddns/rndc.key
```

Configuración de zonas de resolución directa y opciones del servidor de nombres de dominio:

```
//Contenido de /etc/named.conf

// Opciones: (en sistemas basados en Debian, se encuentra en /etc/bind/named.conf.options)
include "/share/ddns/rndc.key";

options {
    listen-on port 53 { any; };
    listen-on-v6 port 53 { any; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query { any; };
    allow-recursion { localhost; };
    recursion yes;

    dnssec-enable yes;
    dnssec-validation yes;
    dnssec-lookaside auto;
};

// Zonas: (en sistemas basados en Debian se encuentra en /etc/bind/named.conf.local)
zone "coddns.org" IN {
    type master;
    file "/etc/zones/db.coddns.org";
    allow-update { key "rndc-key"; };
};
```



Contenido de zona: coddns.org (fichero **/etc/zones/db.coddns.org** → SI NO EXISTE CREELO)

```
$ORIGIN .
$TTL 604800 ; 1 week
coddns.org.      IN SOA  ns1.coddns.org. admin.coddns.org. (
    2015050628 ; serial
    604800     ; refresh (1 week)
    86400      ; retry (1 day)
    2419200    ; expire (4 weeks)
    604800     ; minimum (1 week)
)
NS ns1.coddn.org.
A  217.160.143.23
$ORIGIN coddns.org.
n2      A  217.160.143.23
ns1     A  217.160.143.23
// Aquí debajo, se agregarán las etiquetas personalizadas de los usuarios del sistema CODDNS
```

## ¡IMPORTANTE!

El agente registrador que contrate, debe redireccionar **todas** las peticiones DNS sobre el dominio que elija (en el ejemplo coddns.org) contra **su servidor**.

**Observación:** Si su dominio se va a llamar, por ejemplo, **micasa.lan**, llame al fichero **/etc/zones/db.micasa.lan**

**Observación2:** Recuerde modificar todas las referencias a dominios en named.conf y en los archivos de zona.

Comprobación de zona:

Ejecute:

```
named-checkzone coddns.org /etc/zones/db.coddns.org
```

Ajuste a sus nombres de dominio y ficheros.

Debe recibir el resultado siguiente:

```
# named-checkzone coddns.org. db.coddns.org
zone coddns.org/IN: loaded serial 2015050628
OK
```

Si no recibe ese resultado, por favor, verifique todos los parámetros de configuración estén bien definidos.

Iniciamos el servicio de resolución de nombres de dominio Bind9

```
service named start
```

## INSTALACIÓN DE HTTPD CON SOPORTE A PHP Y POSTGRESQL

### Instalación de HTTPD con soporte a PHP y PostgreSQL

```
yum install httpd php php-pgsql
```

#### NOTA CRÍTICA:

Recuerde configurar correctamente su firewall y las políticas de seguridad de SELinux.

Si no sabe cómo hacerlo consulte en la web o deshabilite siguiendo los siguientes pasos:

```
$ setenforce 0
$ service firewalld stop

# para deshabilitar totalmente selinux -> /etc/selinux/config -> SELINUX=disabled
# para deshabilitar el firewall:
chconfig firewalld off
```

#### Opcional: Servidores virtuales

Agregue el siguiente contenido a “/etc/httpd/conf/httpd.conf” o a la sección de servidores virtuales que corresponda, sustituyendo coddns.org por el nombre de dominio elegido:

```
NameVirtualHost coddns.org:80

<VirtualHost coddns.org:80>
    ServerAdmin admin@coddns.org
    DocumentRoot /var/www/html/coddns
    ServerName coddns.org
    ErrorLog /var/log/coddns/error_log
    CustomLog /var/log/coddns/access_log common
</VirtualHost>
```

Esto le permitirá cargar la página del sistema CODDNS simplemente poniendo el nombre de dominio elegido.

Creamos el directorio /var/log/coddns y le asignamos como propietario a apache, para la salida de logs:

```
mkdir -p /var/log/coddns
chown apache /var/log/coddns
```

## INSTALACIÓN DE POSTGRESQL

Proceso de instalación en sistema CentOS:

```
wget http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-centos94-9.4-1.noarch.rpm
yum install -y pgdg-centos94-9.4-1.noarch.rpm
yum makecache
yum install postgresql94-server
```

Una vez instalado, inicializamos la base de datos y el servicio PostgreSQL:

```
service postgresql-9.4 initdb

# si el comando no funciona, usar
/usr/pgsql-9.4/bin/postgresql94-setup initdb

# Iniciamos PostgreSQL
service postgresql-9.4 start
```

A continuación, con todo preparado, preparamos para cargar el SQL de inicio del sistema CODDNS.

#### Paso 1: Creación del esquema que alojará la base de datos ddns

```
cp ~/coddns_pgsql.sql /tmp/
su postgres
psql
postgres-# \i /tmp/coddns_pgsql.sql
postgres-# create user ddns;
postgres-# grant all on database db_ddns to ddns;
postgres-# grant all on schema sch_ddns to ddns;
postgres-# grant all on all tables in schema sch_ddns to ddns;
postgres-# grant all on all sequences in schema sch_ddns to ddns;
postgres-# alter user ddns with password 'p4ssw0rd';
```

Una vez cargado el script SQL y creado el usuario y la contraseña, procedemos a habilitar el acceso a PSQL vía autenticación con contraseña en el fichero `/var/lib/pgsql/9.4/data/pg_hba.conf`

| #  | TYPE | DATABASE | USER | ADDRESS      | METHOD |
|--|------|----------|------|--------------|--------|
| # "local" is for Unix domain socket connections only |      |          |      |              |        |
| local  | all  |          | all  |              | ident  |
| # IPv4 local connections:                            |      |          |      |              |        |
| host   | all  |          | all  | 127.0.0.1/32 | md5    |
| # IPv6 local connections:                            |      |          |      |              |        |
| host   | all  |          | all  | :::1/128     | md5    |

Al indicar md5 para conexiones locales, permitimos la conexión del PDO PHP-PGSQL contra el motor de la base de datos.

#### DESPLIEGUE Y CONFIGURACIÓN DE LA PÁGINA WEB DEL SISTEMA CODDNS

Extraemos el contenido de `coddns_console.tar.gz` en /

```
cd /
tar xvfz ~/coddns_console.tar.gz
chown apache:apache -R /var/www/html/coddns
```

Verifique que el destino `/var/www/html` coincide con su `$ROOT_HTML_DIR`, en caso contrario corríjalo.

Verifique que el usuario correspondiente a su gestor de páginas web es `apache`, en caso contrario asigne los permisos en base al usuario correcto.

Modificamos las credenciales con las elegidas en el fichero y el domainname a resolver:

#### `$ROOT_HTML_DIR/coddns/include/config.php`

```
<?php
/*
 * Database configuration
 */
$pg_config = array("username"=>"ddns",
                  "password"=>"p4ssw0rd",
                  "hostname"=>"localhost",
                  "port"    =>"5432",
                  "name"    =>"db_ddns",
                  "schema"  =>"sch_ddns");

$db_type = "pgsql";
$db_config = $pg_config;
$config = array("domainname" => "example.lan");
defined("MIN_USER_LENGTH") or define("MIN_USER_LENGTH", 4);
defined("MIN_PASS_LENGTH") or define("MIN_PASS_LENGTH", 4);
$salt = "*****";
?>
```

## DESPLIEGUE DE SCRIPTS DE ADMINISTRACIÓN DE BIND9

Extraemos el contenido de `coddns_bind_management.tar.gz` a /

```
cd /  
tar xvfz ~/coddns_bind_management.tar.gz  
chmod 755 -R /opt/srv  
chmod 755 -R /opt/ddns
```

Verificamos que tienen los permisos correctos (755 para todos los scripts).

A continuación crearemos el enlace simbólico al gestor de dns en `/usr/bin` mediante:

```
ln -s /opt/ddns/dnsmgr.sh /usr/bin/dnsmgr
```

## FAQ

**Q: No funciona nada...**

A: Siga las instrucciones descritas paso a paso.

**Q: En el log de apache recibo una salida “access to /coddns/index.php denied (filesystem path '/var/www/html/coddns/index.php') because search permissions are missing on a component of the path”**

A: Es posible que tenga habilitada la seguridad SELinux, por favor configúrela correctamente o deshabilítala siguiendo los pasos indicados en el manual.

**Q: No se están agregando los hosts a la zona, ni aparece un archivo .jnl**

A: Es posible que la configuración de bind no sea correcta, o que esté usando una clave diferente. Recuerde que la clave debe encontrarse en `/share/ddns/rndc.key` y debe de tener permisos de lectura para todos los usuarios.

**Q: No consigo crear un usuario, sale un mensaje diciendo Wooops, contacte con el administrador.**

A: Es posible que no haya ajustado correctamente los permisos para el usuario que conecta desde PHP a PostgreSQL, verifique que ha seguido al pie de la letra las instrucciones provistas en el apartado “Instalación de PostgreSQL” en esta misma guía.

**Q: Al crear un host nuevo se va la pantalla a la página de bienvenida de instalación de HTTPD... ¿qué está pasando?**

A: Como se indicaba en la sección de “Instalación de HTTPD con soporte a PHP y PostgreSQL” se recomienda utilizar la directiva **Virtualhost** para direccionar las solicitudes contra la página. En caso de que haya decidido prescindir de la recomendación, siéntase libre de mover el contenido de `/var/www/html/coddns/*` a `/var/www/html/*`

## BIBLIOGRAFÍA

Servidor BIND9 manual de uso y técnicas de actualización: <http://www.zytrax.com/books/dns/ch7/xfer.html>

Servidor Apache manual de uso y configuración: <http://httpd.apache.org/docs/2.2/>

PostgreSQL manual de uso y configuración: <http://www.postgresql.org/docs/9.1/static/>

PHP manual de instalación, configuración y guía de desarrollo: <http://php.net/manual/es/index.php>