carrier enabler

FUNCIONAMIENTO INTERNO

Contenido

[1. CONEXIÓN A BASES DE DATOS (connections.php) 3](#_Toc4756199)

[2. FUNCIÓN PARA DESENCRIPTAR CADENA DE CONEXIÓN (decrypt.php) 4](#_Toc4756200)

[3. INICIO DE SESIÓN (login.php) 5](#_Toc4756201)

[3.1 VALIDACIONES 5](#_Toc4756202)

[3.2 PROCESO DE INICIO DE SESIÓN 6](#_Toc4756203)

[3.3. RENOVACIÓN DE CONTRASEÑA 11](#_Toc4756204)

[3.4. CHEQUEO DE NAVEGADOR 13](#_Toc4756205)

[3.5. ORDEN DE FUNCIONES DE INICIO DE SESIÓN 13](#_Toc4756206)

[4. ASPECTOS COMUNES 14](#_Toc4756207)

[4.1. INICIO DE PÁGINAS 14](#_Toc4756208)

[4.2. PÁGINAS DEL PORTAL 15](#_Toc4756209)

[4.3. PÁGINAS FRAME 15](#_Toc4756210)

[4.3.1. Sidebar 15](#_Toc4756211)

[4.3.2. Header 15](#_Toc4756212)

[4.3.3. Footer 15](#_Toc4756213)

[4.3.4. Chat 15](#_Toc4756214)

[4.4. USO DE WEB SERVICES 15](#_Toc4756215)

[5. TIPOS DE CUENTAS 15](#_Toc4756216)

[6. PERFIL Y CUENTAS (profile.php) 15](#_Toc4756217)

[7. PRODUCTOS Y SOLICITUDES 15](#_Toc4756218)

[7.1. CONCEPTO DE PRODUCTO 15](#_Toc4756219)

[7.2. CONCEPTO DE SOLICITUD 16](#_Toc4756220)

[7.3. TABLA DE PRODUCTOS 17](#_Toc4756221)

[7.4. DISPONIBILIDAD DE PRODUCTOS 18](#_Toc4756222)

[7.5. PARÁMETROS DE PRODUCTOS Y SOLICITUDES 19](#_Toc4756223)

[7.5.1. Lista de parámetros 19](#_Toc4756224)

[7.5.2. Tabla de solicitudes 26](#_Toc4756225)

[7.5.3. Valores de parámetros 26](#_Toc4756226)

[7.5.4. Visualización de parámetros 27](#_Toc4756227)

[7.6. INCOMPATIBILIDADES Y DEPENDENCIAS ENTRE PRODUCTOS 27](#_Toc4756228)

[7.7. OTRAS CARACTERÍSTICAS DE PRODUCTOS 28](#_Toc4756229)

[7.8. ESTADOS DE UNA SOLICITUD DE UN PRODUCTO 28](#_Toc4756230)

[7.8.1. Concepto de estado y tipos de estados 28](#_Toc4756231)

[7.8.2. Estados de provisión 29](#_Toc4756232)

[7.8.3. Cambios de estado asociados a un producto [provisión / administrativos] 30](#_Toc4756233)

[7.8.4. Cambios de estado de modificación 36](#_Toc4756234)

[7.9. CREACIÓN DE PRODUCTOS 36](#_Toc4756235)

[7.10. NUEVA SOLICITUD (newrequest.php) 36](#_Toc4756236)

[7.10.1. Consideraciones iniciales 36](#_Toc4756237)

[7.10.2. Nueva solicitud primaria 37](#_Toc4756238)

[7.10.3. Solicitud de servicio adicional sobre servicio principal ya existente 37](#_Toc4756239)

[7.10.4. Datos de los clientes asociados a la solicitud 37](#_Toc4756240)

[7.10.5. Confirmación de la solicitud 37](#_Toc4756241)

[7.11 BÚSQUEDA DE SOLICITUDES DE PRODUCTOS (searchrequests.php) 37](#_Toc4756242)

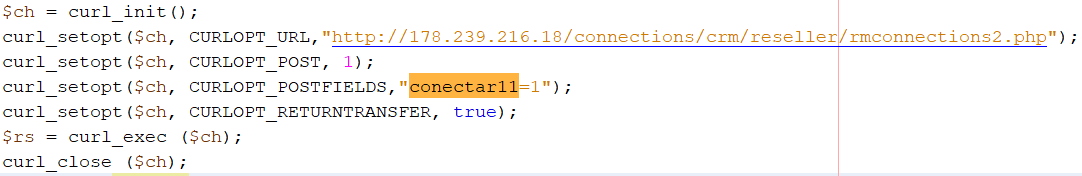
# CONEXIÓN A BASES DE DATOS (connections.php)

Esta página realiza las conexiones a todas las BBDD necesarias. El funcionamiento es el siguiente:

* Según a qué BBDD se quiere conectar, se llama a una de las funciones de esta página. Por ejemplo, conectar11() es la función que conecta con la BBDD del portal. Al llamar a cualquiera de estas funciones, es necesario enviarle una variable ($file), la cual contiene la ruta de la clave para desencriptar la cadena de conexión que recibiremos más adelante (Ver punto 2).



* La función ejecutada realiza una llamada vía CURL a la dirección: <http://178.239.216.18/connections/crm/reseller/rmconnections2.php>, enviándole una variable que indica a qué BBDD se quiere conectar. Por ejemplo, en el caso de la función conectar11(), la variable enviada es: conectar11=1



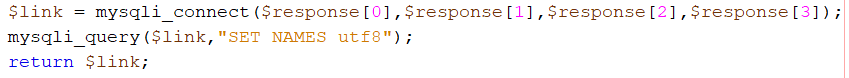
* La página rmconnections2.php del servidor 178.239.216.18, devuelve una cadena encriptada, que contiene todos los datos necesarios para la conexión a la BBDD separados por “//”, siguiendo esta estructura:

IP Servidor + // + Usuario + // + Contraseña + // + Nombre BBDD

* Se desencripta la cadena mediante la función decrypt() (de la página decryptCad.php > Ver punto 2), y se separan sus componentes mediante la función explode, interna de PHP.



* Se realiza la conexión a la BBDD una vez obtenidos los datos necesarios. Dependiendo del tipo de BBDD habrá que conectarse con el driver MySQLi, dblib, etc., en función de si la BBDD en cuestión es de tipo MySQL, SQL Server, etc. Por último, la función devuelve (return) la variable con la conexión ya realizada, con la que podemos seguir trabajando en la página desde la que hemos invocado la función actual.



# FUNCIÓN PARA DESENCRIPTAR CADENA DE CONEXIÓN (decrypt.php)

El objetivo de esta página es desencriptar una cadena ($string), utilizando una clave de desencriptación previamente indicada, la cual se encuentra en un archivo ($file).

En realidad, no la cadena con la que se trabaja no está encriptada como tal, sino ofuscada en base64, aunque utilicemos el término encriptar. En todo caso, el resultado es el mismo: evitar que accesos externos a nuestros servidores puedan ojear las contraseñas de acceso a nuestras BBDD.

El funcionamiento de la función es el siguiente:

1. Al llamar a la función decrypt(), se le envían los parámetros:
   1. $string: la cadena a desencriptar, recibida de la página rmconnections2.php del servidor 178.239.216.18.
   2. $file: la ruta del archivo que contiene la clave de desencriptación. La ruta habitual es: ‘/var/www/html/workflow4 /unknown/k.txt’.

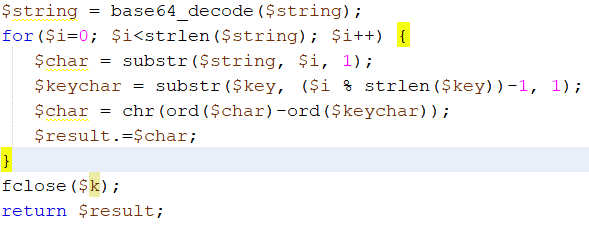


NOTA: Las cadenas recibidas desde el servidor 178.239.216.18 están encriptadas con una clave, alojada en la ruta: ‘/var/www/connections/unknown/k.txt’. La clave que contenga este archivo debe ser la misma que la indicada en el archivo enviado a la función decrypt(), para que la cadena de conexión pueda desencriptarse adecuadamente. De lo contrario, se encriptaría con una cadena, y se desencriptaría con otra diferente, imposibilitando su lectura correcta.

1. Se abre el archivo que contiene la clave de desencriptación, y se lee dicha clave.



1. Se desencripta la cadena recibida, con la clave obtenida, y se devuelve a la página desde la cual se ha llamado a la función actual (connections.php).



# INICIO DE SESIÓN (login.php)

Por razones de seguridad, esta página, aunque es la de inicio, no se nombra como “index.php”. Básicamente, tiene como finalidad permitir a los usuarios loguearse en el portal, identificarlos, y asignarles sus permisos correspondientes, mediante los cuales tendrán acceso a unas funciones u otras dentro del mismo.

Esta página tiene 2 funciones:

1. Permitir al usuario loguearse e identificarse, mediante una serie de validaciones.
2. Modificar la contraseña de inicio de sesión periódicamente.
3. Chequeo de navegador.

## VALIDACIONES

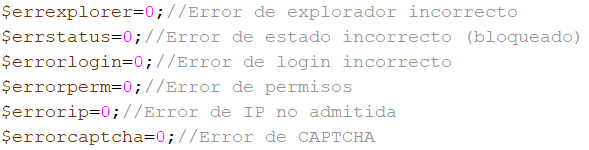
A continuación, se indican todos los procesos que se ejecutan una vez introducidos los datos de acceso. La página realiza 6 comprobaciones inicialmente:

1. Verificar que el navegador es compatible, y la versión del navegador es relativamente reciente. Para ello, la función detect() comprueba estos datos (ver punto 3.4).

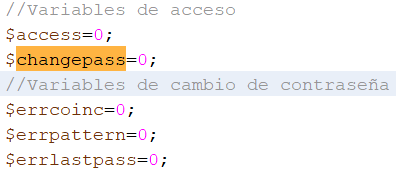


1. Las credenciales introducidas son correctas, es decir, existen en la tabla resellers\_accesos.
2. La IP desde la que el usuario está intentando conectarse está permitida, es decir, está contemplada en la tabla resellers\_accesos\_ips.
3. El usuario que está intentando acceder al portal no está bloqueado.
4. El usuario que está intentando acceder al portal tiene permisos (páginas habilitadas).
5. El CAPTCHA introducido es correcto. Esta comprobación no se realizará en caso de que la IP del usuario esté previamente introducida en una whitelist (también en la tabla resellers\_accesos\_ips).

Las variables relacionadas con estas comprobaciones son las siguientes:

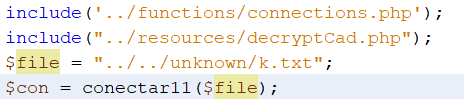


Existen otras variables relativas a otras validaciones, pero se explicarán más adelante:

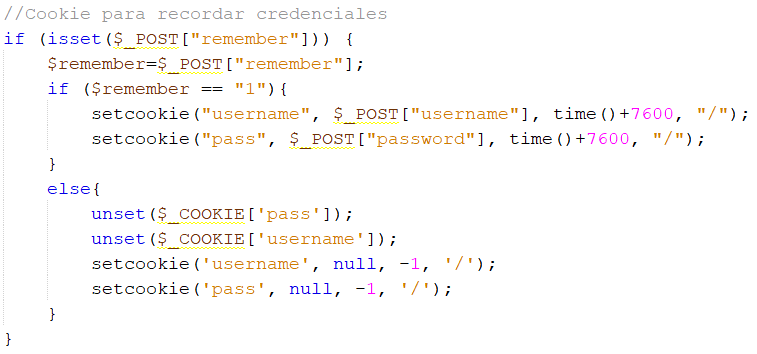


## PROCESO DE INICIO DE SESIÓN

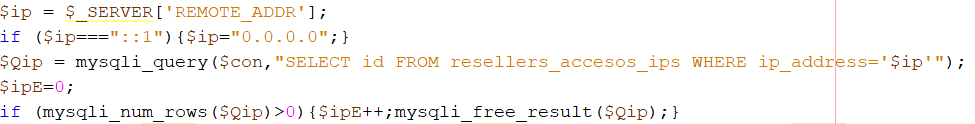
En primer lugar, tras declarar la función detect() relativa al navegador, la página se conecta a la BBDD del portal mediante la función conectar11() (Ver puntos 1 y 2):



Si el usuario final, al introducir los datos, ha marcado la casilla “Recordar”, se genera una Cookie con su usuario y contraseña.



A continuación, se comprueba la IP desde la que el usuario final está intentando conectarse. En caso de que dicha IP ya esté previamente introducida en la tabla resellers\_accesos\_ips, no será necesario que el usuario rellene el CAPTCHA de inicio.

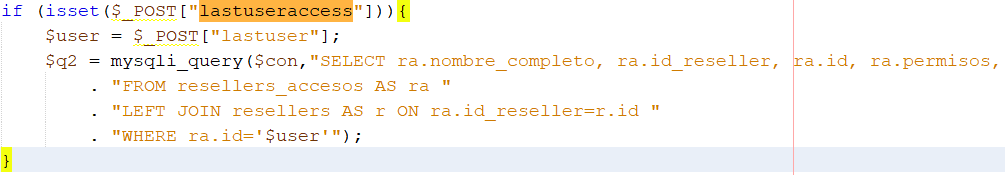


El portal ofrece la posibilidad de acceder como otro usuario. Es decir, desde Carrier, podemos acceder como un usuario principal de otro reseller. Sin tener que indicar su usuario y contraseña. Esta opción está disponible desde el apartado “Perfil y cuentas” (Ver punto 4).

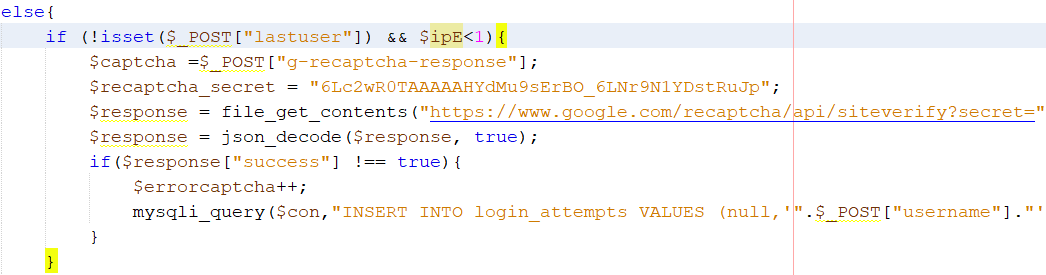
De modo que, la comprobación que se realiza ahora, es la siguiente: si el usuario ya ha introducido su login y contraseña (caso 1), o desde Carrier hemos accedido con el usuario de otro reseller (variable “lastuser” – caso 2) y ahora deseamos recuperar nuestra sesión anterior, continuamos:



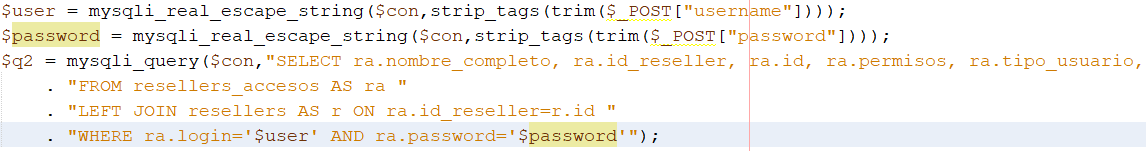
Ahora, si nos encontramos en el caso 2, y desde Carrier hemos accedido al usuario de otro reseller, pero deseamos recuperar nuestra sesión, se validan los permisos de nuestro usuario de Carrier:



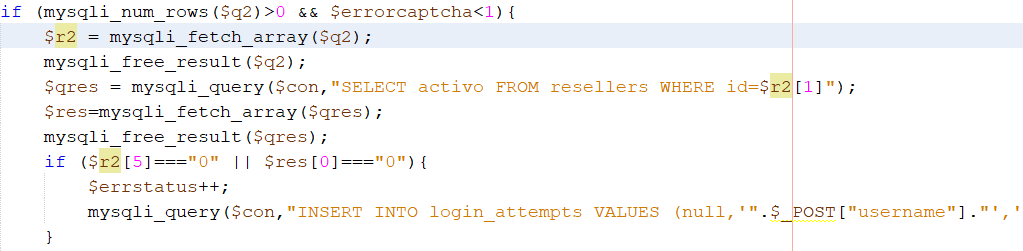
En caso contrario, es decir, estamos en el caso 1 en el que un usuario habitual está intentando acceder al portal, simplemente se validan sus datos de acceso. Primero, si su IP no estaba incluida en la tabla resellers\_accesos\_ips, se valida que el CAPTCHA introducido sea correcto. Si no lo es, la variable $errorcaptcha se incrementa en 1, y se añade un registro a la tabla login\_attempts:



Después, se verifica en la base de datos si el usuario y la contraseña introducidos existen:



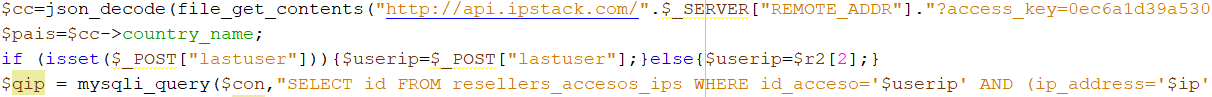
En caso de que el usuario y contraseña introducidos existan, y el CAPTCHA introducido también sea correcto (o no haya sido necesario introducirlo), se verifica que el usuario en cuestión no esté bloqueado. En caso de que lo esté, la variable $errstatus se incrementa, y se añade un registro en la tabla login\_attempts:



Si el usuario está activo, a continuación, se valida la IP desde la que está intentando acceder. Anteriormente ya se comprobó si la IP existía en la tabla resellers\_accesos\_ips, para determinar si el usuario tiene que hacer la comprobación CAPTCHA o no. En este caso, se comprueba que el usuario tiene acceso desde el país al que pertenece su IP.

En caso de que el usuario tenga establecida la IP 0.0.0.0, y como país “Spain” en la tabla resellers\_accesos\_ips, quiere decir que puede acceder desde cualquier IP de España. También se pueden indicar IPs específicas en esta tabla, o todas las IPs (0.0.0.0) de cualquier otro país.

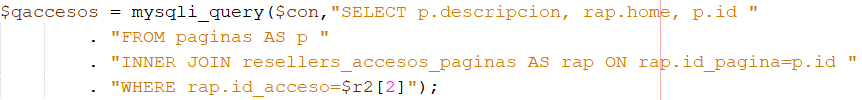
El país al que pertenece la IP del cliente se obtiene mediante la web api.ipstack.com. Una vez obtenido el país, se realiza la comprobación anterior:



En caso de que la IP o el país desde el que está intentando acceder al portal no estén habilitados, la variable $errorip se establecerá en 1, y se añadirá un registro en la tabla login\_attempts:



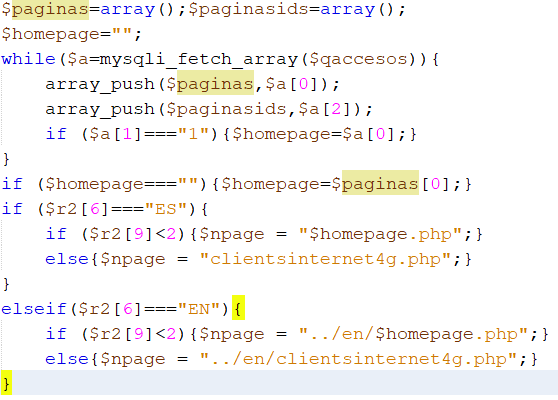
En caso contrario, se pasarán a validar los permisos del usuario. Se leerán las páginas del portal a las que tiene acceso de la tabla resellers\_accesos\_paginas, mediante la siguiente consulta:



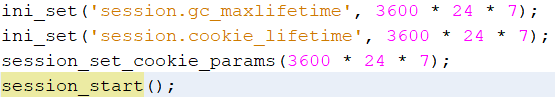
Si no tiene habilitada ninguna página, no podrá acceder al portal. Se establecerá en 1 la variable $errorperm, y se añadirá el registro correspondiente en la tabla login\_attempts.



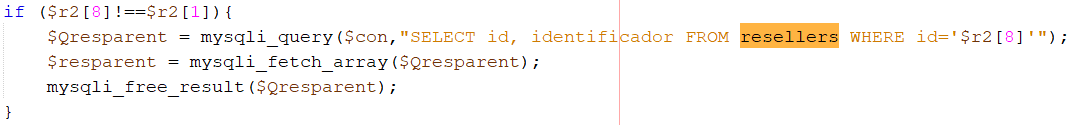
SI tiene páginas habilitadas, se determinará cuál de ellas es la página de inicio (campo “home” de la tabla resellers\_accesos\_páginas). En este caso, existe una excepción para distribuidores (no resellers), que siempre accederán por defecto a la página clientsinternet4g.php:



Seguidamente, se inicializa una sesión que se mantendrá abierta mientras estemos en el portal:



Se obtiene el id y el identificador del reseller de nivel superior. Para todos los resellers, el reseller de nivel superior es Carrier. Para los distribuidores, es su reseller asociado:

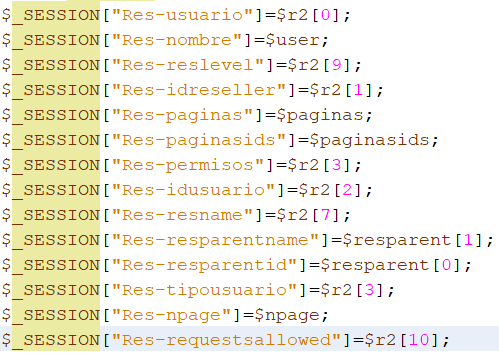


En caso de que, desde Carrier, estemos accediendo a la cuenta de otro reseller, guardamos la variable $lastuser (es decir, el ID de nuestro usuario en Carrier), en una variable de sesión (Res-lastuser). De esa manera, cuando queramos recuperar nuestra sesión anterior, podremos hacerlo leyendo de la variable de sesión. Sino, la variable de sesión “Res-lastuser” se establece en cero:

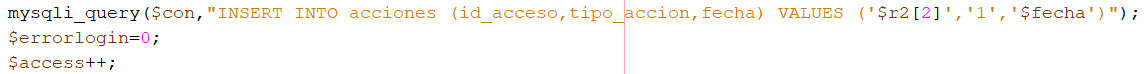


Se crean otras variables de sesión:

* Res-usuario: ID del usuario obtenido de la tabla resellers\_accesos
* Res-nombre: Nombre del usuario introducido en el formulario de inicio
* Res-reslevel: Nivel del reseller obtenido de la tabla resellers, en la consulta $q2. El nivel de Carrier es 0, el nivel de sus resellers es 1, y el de los distribuidores es 2. Si en el futuro se incluyeran nuevos niveles, cada uno sería inferior al de su reseller asociado (reseller parent).
* Res-idreseller: ID del reseller del usuario, obtenido de la tabla resellers\_accesos
* Res-paginas: Nombres de las páginas a las que el usuario tiene acceso, obtenidas de la consulta $qaccesos.
* Res-paginasids: IDs de las páginas a las que el usuario tiene acceso, obtenidas de la consulta $qaccesos.
* Res-permisos: Permisos del usuario, obtenidos de la tabla resellers\_accesos. Los permisos disponibles son: usuario, administrador, provisión. Este campo se puede editar desde la sección “Perfil y cuentas” (ver punto 4).
* Res-resname: Nombre (identificador) del reseller al que pertenece el usuario, obtenido de la tabla resellers.
* Res-resparentname: Nombre (identificador) del reseller de nivel superior (reseller parent), obtenido de la consulta $Qresparent. Para la mayoría de usuarios, el nombre de su reseller parent es Carrier. Para los usuarios de distribuidores, el nombre de su reseller parent es el reseller asociado, el cual es a su vez cliente de Carrier.
* Res-resparentid: ID del reseller de nivel superior (reseller parent).
* Res-tipousuario: Mismo campo que el utilizado en la variable Res-permisos.
* Res-npage: Página de inicio, obtenida de la consulta $qaccesos. Anteriormente se explica el proceso.
* Res-requestsallowed: Posibilidad de realizar o no solicitudes. Esta opción normalmente está establecida en 1 (Sí), aunque existen algunas excepciones, de resellers que no tienen posibilidad de realizar peticiones de servicios. En ese caso, la opción se encontrará en 0 (No). Este campo se lee de la tabla resellers.



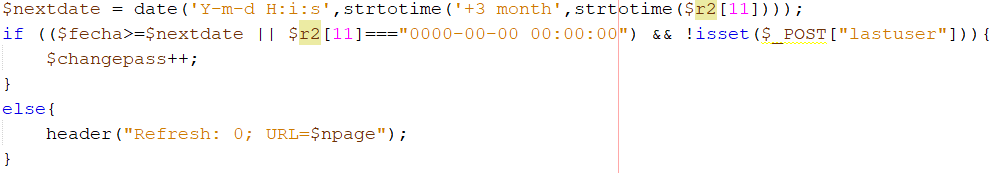
Dado que el inicio de sesión ha sido correcto, se da el valor correspondiente a las variables $errorlogin y $access, y se inserta un registro en la tabla login\_attempts. En este punto, la sesión ya habrá sido iniciada:



## 3.3. RENOVACIÓN DE CONTRASEÑA

Uno de los datos que también se obtiene al iniciar sesión, es la última vez que la contraseña del usuario fue modificada. Por seguridad, el sistema está diseñado para que cada usuario modifique este dato cada 3 meses. La fecha de la última modificación de contraseña se obtiene de la tabla resellers\_accesos.

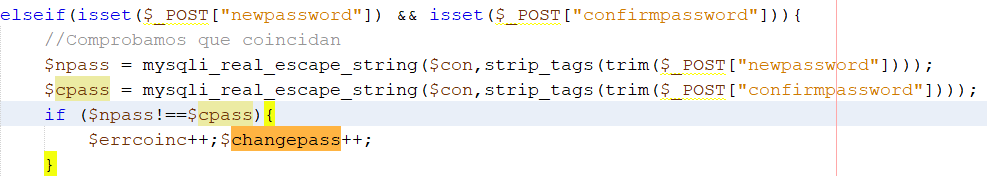
En caso de que hayan pasado más de 3 meses desde la última modificación de contraseña, se le solicitará un cambio al usuario que ha iniciado la sesión, tras realizarse todas las validaciones de inicio. Únicamente no se solicitará la modificación si es desde Carrier que estamos intentando acceder a la cuenta de dicho usuario (la variable “lastuser” no existe):



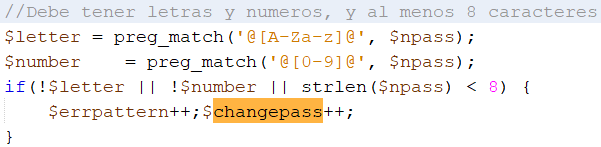
Como se muestra en la imagen anterior, si se requiere un cambio de contraseña, se incrementará en uno la variable $changepass. Sino, el sistema redirigirá al usuario a su página de inicio.

En caso de que el usuario haya de modificar su contraseña de acceso al portal, le aparecerá en pantalla el formulario oportuno.

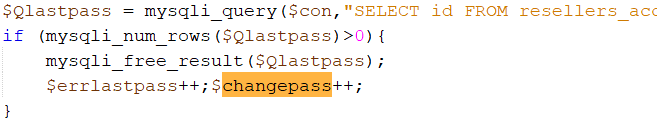
Al confirmar el cambio de contraseña, se enviarán vía POST las variables newpassword y confirmpassword, que deben coincidir. Si no es así, la variable $errcoinc, mostrada anteriormente, se incrementará en 1.



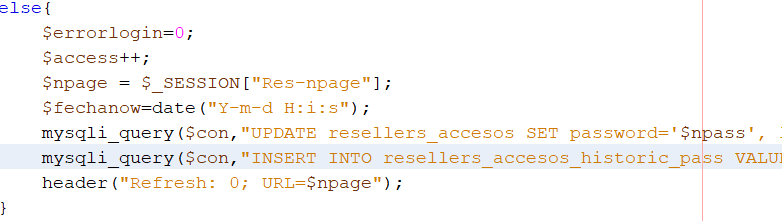
Si coinciden, se realizan otras validaciones. La contraseña debe tener letras y números, y, al menos, 8 caracteres. Si no se superan estas validaciones, la variable $errpattern, mostrada anteriormente, se incrementará en 1:



Otra validación que se realiza en este paso, es que la contraseña introducida no sea igual que ninguna de las anteriores para este usuario. Esto se comprueba en la tabla resellers\_accesos\_historic\_pass. En caso afirmativo, la variable $errlastpass se incrementará en 1, y se le requerirá de nuevo el cambio de contraseña al usuario:



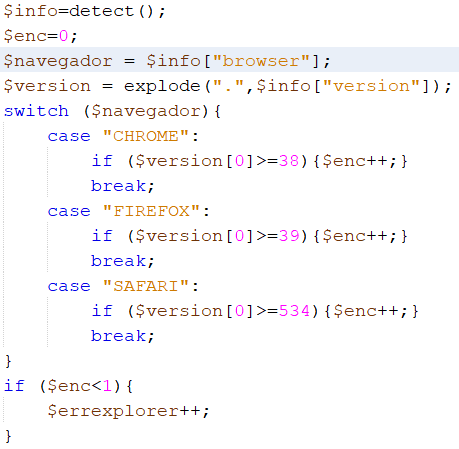
Si todo es correcto, se actualiza la contraseña de acceso del usuario en la tabla resellers\_accesos, se inserta el nuevo registro en la tabla resellers\_accesos\_historic\_pass, y se redirige al usuario a su página de inicio:



## CHEQUEO DE NAVEGADOR

En puntos anteriores se comentó la función detect(), mediante la cual podemos obtener la versión de navegador, entre otros datos, del usuario que intenta acceder al portal.

Antes de que el usuario pueda introducir sus datos de acceso, se chequeará que el navegador desde el que intenta acceder al portal es relativamente reciente, mediante el siguiente proceso:



Si usa Chrome en una versión superior a la 38, Firefox en una versión superior a la 39, o Safari en una versión superior a la 534, podrá acceder al portal.

En caso contrario, el formulario para acceder al mismo estará inhabilitado, y se mostrará en pantalla un mensaje indicando este motivo.

## ORDEN DE FUNCIONES DE INICIO DE SESIÓN

El orden de todas las funciones implicadas en esta página es el siguiente:

1. El usuario accede a la página login.php.
2. Se comprueba su navegador. Si no es correcto, se muestra un mensaje de error.
3. Si el navegador es correcto, introduce los datos, los cuales también se validan.
4. En caso de que todos los datos (anteriormente explicados) sean correctos, se validará la antigüedad de la contraseña.
5. Si la contraseña es antigua, se solicitará la renovación, que también se validará.
6. Se accederá a la página de inicio si los datos de acceso son correctos y la contraseña es reciente.
7. Si se accede a la sesión de un usuario desde Carrier, directamente se iniciará la sesión, a no ser que el acceso en cuestión esté bloqueado.

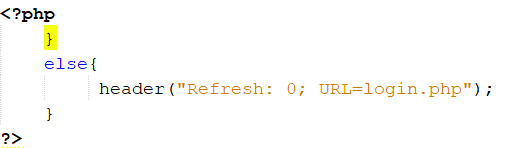
# ASPECTOS COMUNES

## 4.1. INICIO DE PÁGINAS

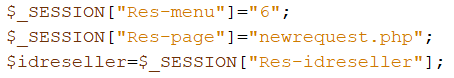
Al acceder a esta página, tal como al resto, se valida que la sesión se ha iniciado correctamente y esta no ha expirado:



En caso negativo, se recargará la página de login.php (el código de refresco está en la parte final de la página):



Si la sesión está activa, se inicializan las variables de sesión oportunas. Las variables “Res-menu” y “Res-page” tienen la finalidad de interactuar con las páginas sidebar.php y header.php, que realizan la carga del menú y cabecera del portal, respectivamente:

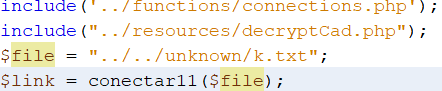


La variable “Res-menu” identifica el ID del menú en que nos encontramos, para marcarlo en el sidebar con un color diferente.

La variable “Res-page” permite identificar el nombre de la página .php en la que nos encontramos, de manera que, al cambiar de idioma de español a inglés, o viceversa, el sistema pueda redirigirnos a esa misma página, en el directorio correspondiente (“es” o “en”).

Por último, la variable de sesión “Res-idreseller”, permite obtener nuestro ID de reseller, previamente cargado en la página login.php, al iniciar sesión. Esta variable permite que, todas las acciones que se lleven a cabo en el portal, se asocien a la cuenta de reseller del usuario activo, independiente de su nivel de reseller (Carrier/Reseller/Distribuidor/etc.).

Adicionalmente, la mayoría de páginas realiza su inicio una conexión a la BBDD del portal, para trabajar con esta a lo largo de la carga de la página (Ver puntos 1 y 2):



Por último, las páginas que en alguna ocasión han dado problemas de carga, tienen establecido el tiempo máximo de carga en 0 (infinito). De esa manera se evita que la página colapse al superar los 30 segundos de carga. Al establecer esta constante en cero, eliminamos dicha restricción:



## 4.2. PÁGINAS DEL PORTAL

## 4.3. PÁGINAS FRAME

### 4.3.1. Sidebar

### 4.3.2. Header

### 4.3.3. Footer

### 4.3.4. Chat

## 4.4. USO DE WEB SERVICES

# TIPOS DE CUENTAS

# PERFIL Y CUENTAS (profile.php)

# PRODUCTOS Y SOLICITUDES

## CONCEPTO DE PRODUCTO

Se entiende por producto cualquier servicio que Carrier ofrece a sus resellers. La mayoría de productos son facturables, es decir, computan para las facturas que Carrier emite a sus resellers. Ejemplo: ADSL. Tiene coste mensual.

Otros simplemente se solicitan para acceder a ciertas funciones que no tienen coste en sí, pero cuyo uso sí genera un consumo que repercute en la facturación mensual. Ejemplo: IP de terminación. El servicio es gratuito. Pero las llamadas que se generan desde la IP del reseller, computan para la facturación.

Los productos que pueden solicitarse son de 2 tipos:

1. Productos principales: se trata de servicios que no dependen o cuelgan de ningún otro.
2. Productos adicionales: son servicios que no pueden funcionar sin la existencia de otro servicio de carácter superior, del cual dependen. Es necesario solicitar primero el servicio principal, antes del adicional. En algunos casos, será necesario que el servicio principal esté totalmente finalizado para poder solicitar el servicio adicional. En otros casos, será posible será posible solicitar servicios adicionales antes de que el principal esté provisionado.

## CONCEPTO DE SOLICITUD

Todos los servicios que ofrece Carrier son solicitados por los resellers a través de esta sección. En un servicio, existen 2 tipos de datos:

1. Datos de la solicitud: son todos los parámetros que componen el servicio en cuestión.
2. Datos del cliente: nombre, documento, y otra información relativa al cliente final.

Por otra parte, existen diferentes situaciones en las que un reseller debe interactuar con estos datos:

1. Al realizar una nueva solicitud, debe rellenar ambos tipos de datos.
2. Si se le solicita por nuestra parte una modificación de datos de la solicitud o del cliente (sea de forma automática o manual), el reseller deberá aplicar los cambios oportunos para continuar con la provisión del servicio.
3. Al iniciar un cambio de titular, el reseller debe indicar todos los datos del nuevo cliente asociado al servicio en cuestión.
4. Al ejecutar un cambio de estado de una solicitud, es posible que, en algunos casos, se le soliciten al reseller los datos del cliente final. Este caso apenas se presenta.
5. Al realizar búsquedas de servicios en la página de búsqueda de solicitudes, el reseller podrá identificar un servicio por los datos de la solicitud, y los del cliente.

Aparte de estos dos tipos de datos, existen algunos parámetros comunes a todas las solicitudes, y transparentes para el reseller. Se tratarán más adelante.

Después, otro programa (Check de facturación) lee todos los datos de los servicios activos, para añadirlos en el programa de facturación (Eclipse), que emite las facturas a los resellers. De manera que, los datos de clientes finales, únicamente son para uso del reseller. Nosotros no facturamos a clientes finales, sino a los resellers. Los datos del cliente final se solicitan porque así lo exige la ley en la mayoría de casos, y para facilitarles a los resellers las búsquedas de servicios en el portal.

Por tanto, es de vital importancia que todos los cambios que deban hacerse en los datos de servicio o del cliente se realicen, siempre que sea posible, a través del mismo portal, no directamente desde la BBDD, para evitar fallos en las búsquedas y en la facturación.

## TABLA DE PRODUCTOS

Todos los productos disponibles en el portal se encuentran en la tabla productos. Esta tabla define las características principales de los productos, así como el propietario de estos. La estructura de la tabla es la que sigue a continuación:

* id: Identificador único del producto.
* tipo\_servicio: Ámbito sobre el cual funciona el servicio. Existen 3 tipos:
  + 1: Voz
  + 2: Datos
  + 3: Hardware
* descripcion: Nombre del producto en español.
* descripcion\_eng: Nombre del producto en inglés.
* id\_reseller: Identificador del reseller creador y propietario del producto.
* adicional: Este campo determina si el producto es un servicio principal o adicional. Si vale 0, es un servicio principal. Si vale 1, es un servicio adicional.
* ofrecer: Puede valer 0 (No) o 1 (Sí). En caso de que valga 0, el producto no estará disponible para ser solicitado por el portal. Si vale 1, podrán solicitarlo todos aquellos resellers que tengan el producto asignado (Ver punto 6.3).
* finalizada: Esta opción solo es válida para los servicios adicionales, no para principales. Si vale 1, quiere decir que el servicio adicional solo se puede solicitar en productos que ya están provisionados y facturando. Más adelante se explicarán estos conceptos (Ver punto 6.7).
* auto: Este campo es válido en productos que no dependen de proveedores externos, y están programados para provisionarse automáticamente en las diferentes plataformas. De manera que, cuando el reseller los solicita desde el portal, el sistema buscará el camino más rápido entre los distintos cambios de estado disponibles, para provisionar totalmente el servicio, llevando el producto hacia los estados administrativos “activado” y “facturando” (Ver punto 6.7).
* tipo\_cuenta: Identificador de la cuenta sobre la cual se provisiona este producto. Si vale 0, quiere decir que este servicio no precisa de ninguna cuenta de voz, datos, u otras, por parte del reseller, para poder funcionar.
* link: Identificador de la página del portal que hace referencia a este producto, y en la cual se puede acceder a la configuración avanzada del servicio en cuestión. Si vale 0, quiere decir que este servicio no está relacionado ninguna página de configuración.
* help\_text: Texto en idioma español, en formato HTML, que sirve de ayuda para la configuración del servicio por parte del reseller. Este texto se mostrará en la sección “Solicitudes > Buscar”, al pulsar sobre el botón de detalles de la solicitud (Ver punto ).
* help\_text\_eng: Igual que el campo anterior, pero en idioma inglés.
* paginas: Este campo ya no está en uso [Obsoleto].
* defecto: Puede valer 0 (No) o 1 (Sí). Este campo delimita qué productos tienen todos los resellers por defecto. Es decir, cuando se crea un nuevo reseller desde el portal comercial (Ver punto ), a la hora de determinar qué productos tiene disponibles el reseller, los que tengan este campo en 1 vendrán habilitados por defecto.
* limite: Número máximo de solicitudes no desactivadas que un reseller puede tener de este producto. Este campo no se utiliza en ningún producto actualmente, aunque no está obsoleto.
* limite\_order: Número máximo de solicitudes por pedido que un reseller puede realizar de este producto.
* allow\_quantity: Este campo ya no está en uso [Obsoleto].

## DISPONIBILIDAD DE PRODUCTOS

La tabla resellers\_productos indica qué productos están disponibles para cada reseller. Consta de la siguiente estructura:

* id: Identificador único del registro.
* id\_producto: Identificador del producto.
* id\_reseller: Identificador del reseller.
* pedido\_minimo: Número mínimo de solicitudes que el reseller debe realizar de un producto por cada pedido.
* tipo\_pago: Este campo ya no está en uso [Obsoleto].

## PARÁMETROS DE PRODUCTOS Y SOLICITUDES

### Lista de parámetros

Cada producto contiene una serie de parámetros, los cuales definen sus características, configuración, y otros detalles. Cuando un reseller solicita un producto a través del portal (Ver punto ), y cuando se realiza un cambio de estado de la solicitud relativa al producto por parte del reseller o del seller (Ver punto ), se deben rellenar una serie de parámetros, definidos en la tabla productos\_dependencias\_solicitudes2, cuya estructura es la siguiente:

* id: Identificador único de la tabla
* id\_producto: Identificador del producto en cuestión.
* campo: Parámetro cuyo valor va a ser definido o modificado. Dicho campo debe formar parte de estructura de la tabla solicitudes, es decir, debe ser una de las columnas de dicha tabla. Aunque generalmente se identifican los parámetros (columnas) con un nombre, no es necesario, ya que dichos parámetros pueden ser reutilizados en varios productos.
* value\_default: Valor por defecto que tomará el parámetro. Los valores pueden ser de diferentes tipos, se explican en el punto 7.4.2.
* descripcion: Nombre del parámetro en inglés.
* dependencia\_campo: Algunos parámetros dependen de otros, tanto en su existencia como en sus diferentes valores. Es decir, dependiendo del valor que tome un parámetro, otro puede mostrarse en pantalla o no (existencia), o tomar diferentes valores. En este campo se indican los campos de los que depende el parámetro actual, separados por coma. Este campo está muy relacionado con el campo “valor\_dependencia”, que se explicará más adelante.
* valor: Valores que puede tomar el parámetro, en español. Los valores pueden ser de diferentes tipos, se explican en el punto 7.4.2.
* valor\_eng: Valores que puede tomar el parámetro, en inglés. Los valores pueden ser de diferentes tipos, se explican en el punto 7.4.2.
* descripcion\_valor: Este campo tiene utilidad solo en el caso de parámetros informativos. No son parámetros en sí, ya que no tienen efecto alguno en la BBDD, sino que se trata de información que aparece en pantalla a la hora de realizar una nueva solicitud, o un cambio de estado de una solicitud. El único caso en que se usa este campo es en las portabilidades, mostrándole al reseller que al rellenar la solicitud acepta ciertas condiciones. El idioma en que aparece este campo es inglés.
* valor\_dependencia: Tiene estrecha relación con el campo “dependencia\_campo”, antes explicado. Sirve para indicar qué valores deben tener los campos de los que depende el campo actual, para existir, o para tomar unos valores u otros. Es decir, dependiendo de los valores que tomen los campos definidos en el campo “dependencia\_campo” (separados por coma), el campo actual podrá existir (mostrarse en pantalla) o no, o tomará distintos valores. Por ejemplo:

|  |  |  |
| --- | --- | --- |
| **dependencia\_campo** | **valor\_dependencia** | **Explicación** |
| telefono\_analogico,id\_existente | 1,1 | El campo teléfono\_analogico debe valer 1, y el campo valor\_dependencia debe valer 1, para que el campo actual exista (se muestre en pantalla). |
| tipo\_sip | 1|3|4|5|6 | El campo tipo\_sip debe valer 1, o 2, o 3, o 4, o 5, o 6, para que el campo actual pueda existir. |
| tipo\_adsl | 1 | El campo tipo\_adsl debe valer 1 para que el campo actual pueda existir. |
| hunt\_lists,provincia | 1, | El campo hunt\_lists debe valer 1 para que el campo actual exista. Además, el campo actual podrá tomar diferentes valores en función del valor que tenga el campo provincia. |
| tipo\_sip,id\_existente,config\_file,extensiones | 0,0,, | Los campos tipo\_sip y id\_existente deben valer 1 para que el campo actual pueda existir. Además, el campo actual tomará diferentes valores en función de los valores que tengan los campos config\_file y extensiones. |

* mostrar: Este campo define si el parámetro en cuestión se mostrará en pantalla o no. Puede valer 0 (No) o 1 (Sí). En caso de que valga 0, nunca aparecerá en pantalla el parámetro, independientemente de los valores indicados en los campos “dependencia\_campo” y “valor\_dependencia”.
* priority: Orden en que aparecerá el parámetro en pantalla, en caso de que se muestre.
* type: Tipo de parámetro. Está relacionado con el campo “valor”, ya que, dependiendo de este, el parámetro en cuestión podrá ser de un tipo u otro. Por ejemplo, si el campo “valor” es de tipo “All”, quiere decir que el parámetro acepta cualquier tipo de valores. O, si el campo “valor” es de tipo “@V:”, quiere decir que el parámetro acepta una serie de valores predefinidos, los cuales son seleccionables.

Ahora bien, este campo (type) determina la naturaleza del parámetro que aparecerá en pantalla, que puede ser:

* + datetimepicker: Campo de tipo datetime (calendario).
  + files: Campo de tipo archivos (subida de archivos).
  + inputselect: Campo de tipo input autorellenable. Generalmente, cuando el campo “valor” es de tipo “@V:”, los valores predefinidos a seleccionar aparecen en un campo de tipo select. Sin embargo, si el campo type está definido como inputselect, aparecerá un input autorellenable en lugar de un select.
  + numeric: Campo de tipo numérico.
  + textarea: Campo de tipo textarea.
* descripcion\_esp: Nombre del parámetro en español.
* descripcion\_valor\_esp: Igual que el campo descripcion\_valor, pero en idioma español.
* obligatorio: Determina si el parámetro es obligatorio o no. Puede valer 0 (No) o 1 (Sí). En algunos casos, el que un parámetro sea obligatorio o no depende de ciertas condiciones, establecidas en los campos “obligatorio\_dependencia”, “obligatorio\_valor”, “obligatorio\_todas\_condiciones” y “obligatorio\_una\_condicion”. En esos casos, el campo actual (obligatorio) se establecerá en 0, y se establecerán dichas condiciones en los campos correspondientes.
* max\_length: Máxima longitud que puede tomar el valor del parámetro.
* min\_value: Valor mínimo puede tomar un parámetro. No necesariamente debe ser un valor fijo. Puede depender de ciertas condiciones (valor de tipo “@V:”, ver punto 7.4.2).
* max\_value: Valor máximo puede tomar un parámetro. No necesariamente debe ser un valor fijo. Puede depender de ciertas condiciones (valor de tipo “@V:”, ver punto 7.4.2).
* modificable: Este campo ya no está en uso [Obsoleto].
* informativo: Al igual que los campos descripcion\_valor y descripcion\_valor\_esp, este campo solo se utiliza en parámetros que no tienen efecto alguno en la BBDD, pero muestran información en pantalla a la hora de realizar una solicitud, o cambiar el estado de la misma. Para que los campos descripcion\_valor y descripcion\_valor\_esp se muestren en pantalla de la manera oportuna, es necesario que este campo esté establecido con valor 1. Actualmente solo se utiliza en el caso de portabilidades, igual que los dos campos antes mencionados.
* restar: Está relacionado con el campo max\_value. En el caso de algunos parámetros de tipo numérico (type=numeric), su valor máximo depende de ciertas condiciones. Por tanto, el valor de este campo será de tipo “@V:”. En el punto 7.4.2 se explica con mayor detalle en qué consisten este tipo de valores. Actualmente este campo solo se utiliza en el producto Tenant. Algunos de los parámetros de este producto dependen de cuánto uso se haya dado a la Cloud PBX de la cual depende. Por ejemplo:

Cloud PBX 50: Tiene un máximo de 50 extensiones, de las cuales ya ha ocupado 41 entre sus distintos tenants activos.

Al solicitar un nuevo tenant para esta Cloud PBX, el valor máximo del parámetro “extensiones” dependerá del campo max\_value (en este caso, 50), y del campo actual (restar), el cual determinará que al campo max\_value hay que restarle las 41 extensiones ya activas, de manera que el valor máximo del parámetro “extensiones” será de 9.

En resumen, el campo “restar”, resta (valga la redundancia) una cantidad determinada al campo “max\_value”.

* comparacion: Este campo ya no está en uso [Obsoleto].
* id\_reseller: Mediante este campo podemos limitar un parámetro a unos resellers u otros. Actualmente no tiene uso, aunque no está obsoleto. Generalmente tiene el valor “All”, lo que significa que el parámetro aparecerá en pantalla en solicitudes de cualquier reseller.
* min\_length: Máxima longitud que puede tomar el valor del parámetro.
* readonly: Puede valer 0 (No) o 1 (Sí). En caso de que valga 1, el parámetro en cuestión no será modificable, únicamente de solo lectura.
* pattern: Patrón regex que debe cumplir el parámetro.
* obligatorio\_dependencia: Campo[s] del que depende el parámetro para ser obligatorio, separados por coma.
* obligatorio\_valor: Valor que debe tener cada campo de los que depende el parámetro para ser obligatorio. Si hay dependencia de varios campos, los valores deben estar separados por coma. Pueden ser valores concretos, o tomados de una lista. Por ejemplo:

1 = Debe valer 1

1|6|25|31 = Puede valer 1, 6, 25 o 31.

* obligatorio\_todas\_condiciones: Actualmente este campo no se encuentra en uso, aunque no está obsoleto. Puede valer 0 (No) o 1 (Sí). En caso de que valga 1, quiere decir que deben cumplirse todas las condiciones definidas entre los campos “obligatorio\_valor” y “obligatorio\_dependencia”, es decir, que todos los campos definidos en la columna “obligatorio\_dependencia” tengan los valores indicados en la columna “obligatorio\_valor”.
* obligatorio\_una\_condicion: Actualmente este campo no se encuentra en uso, aunque no está obsoleto. Puede valer 0 (No) o 1 (Sí). En caso de que valga 1, quiere decir que, al menos, debe cumplirse una condición de las definidas entre los campos “obligatorio\_valor” y “obligatorio\_dependencia”, es decir, que al menos uno de los campos definidos en la columna “obligatorio\_dependencia” tenga el valor indicado en la columna “obligatorio\_valor”.
* informacion: Texto informativo que aparece en pantalla al pulsar sobre el botón de información del parámetro, en español.
* informacion\_eng: Texto informativo que aparece en pantalla al pulsar sobre el botón de información del parámetro, en inglés.
* id\_dependencia\_estado: Este campo determina el momento en que será necesario rellenar el parámetro en cuestión. Puede tomar valor 0, o cualquiera de los IDs de la tabla productos\_dependencias\_estados, del producto en cuestión. Si vale 0, quiere decir que el parámetro se rellenará al solicitar el producto por primera vez. En caso contrario, se rellenará al realizar un cambio de estado en la solicitud relativa al producto. Dicho cambio de estado corresponderá al ID de la tabla productos\_dependencias\_estados.

Es posible que algunos parámetros hayan de ser rellenados en diversos momentos, por cada uno se deberá generar un registro en la tabla actual (productos\_dependencias\_solicitudes2), indicando en este campo el momento adecuado del proceso de provisión de una solicitud.

* showIfNotEmpty: Dado que existe la posibilidad de que un parámetro pueda ser rellenado en diversos momentos del proceso de provisión de una solicitud, y también existe la posibilidad de que un parámetro sea obligatorio, o no, este campo permite definir que, si el parámetro en cuestión ya está rellenado (not empty), no aparezca en pantalla. Puede valer 0 (No) o 1 (Sí).

De modo que, si vale 1, aunque el parámetro en cuestión ya haya sido rellenado previamente, aparecerá igualmente en pantalla para rellenarlo (si se cumplen las condiciones de dependencia con otros campos, y el campo mostrar también vale 1). Si vale 0, en caso de que el parámetro en cuestión ya haya sido rellenado, no aparecerá en pantalla para rellenarlo de nuevo, aunque el campo mostrar valga 1 y se cumplan las condiciones de dependencia con otros campos.

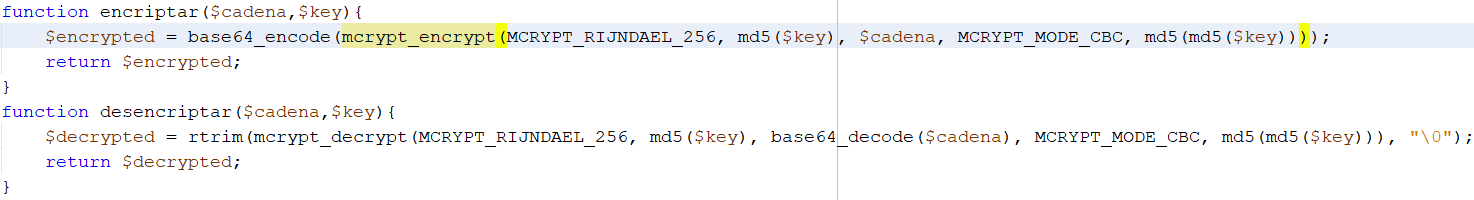
Este campo no tiene validez cuando el producto se está solicitando por primera vez, sino únicamente en cambios de estado de solicitudes.

* existencia\_valores: El que un parámetro aparezca en pantalla o no, no es lo mismo que el que un parámetro exista o no. Un parámetro puede no mostrarse en pantalla, estando oculto, pero sí tener efecto a nivel de BBDD. Los únicos campos que permiten definir cuándo un parámetro existe (afecta a la BBDD) son: “dependencia\_campo” y “valor\_dependencia”, así como “existencia\_valores” y “existencia\_condiciones”. Los dos primeros definen que un parámetro existirá o no en función del valor de otro[s] parámetro[s]. Los dos últimos definen que un parámetro existirá o no en base a ciertas condiciones (valor de tipo “@V:”, ver punto 7.4.2). En caso de que deban cumplirse varias condiciones para que el parámetro exista, estas vendrán separadas por “//”.

Actualmente solo se utiliza en el caso de DIDs que cuelgan del producto Carrier SIP Entrante.

existencia\_condiciones: Condiciones que deben cumplirse para que el parámetro exista. Es decir, los valores obtenidos del campo “existencia\_valores”, deben ser los que se indican en el campo actual. En el caso anterior, las condiciones que deben cumplirse son:

* + El producto principal del DID debe ser Carrier SIP Entrante (@V:75 = 63).
  + El reseller que solicita el DID debe tener varios Carrier SIP Entrante finalizados, para poder balancear el tráfico entrante del DID entre estos (@V:74 = >1).
* comprobacion: Actualmente solo se utiliza para validar IPs. Es preferible utilizar el campo “pattern”, explicado antes, para validar cualquier patrón de una cadena. No obstante, el portal tiene implementadas en la programación algunas funciones para validar ciertos tipos de cadenas:
  + ip: para validar direcciones IP
  + mac: para validar direcciones MAC
  + email: para validar direcciones de email.
* mindays: En el caso de parámetros de tipo datetimepicker, es posible definir la fecha mínima disponible para seleccionar en el calendario. Este campo determina cuántos días (en periodos 24h) deben haber pasado entre la fecha actual y la fecha mínima seleccionable. Actualmente solo tiene utilidad para seleccionar la fecha de VC en portabilidades.
* hoursdisabled: En el caso de parámetros de tipo datetimepicker, este campo permite definir qué horas concretas no será posible seleccionar en el calendario. Actualmente solo tiene utilidad para seleccionar la fecha de VC en portabilidades, las cuales solo permiten seleccionar hora para VC a las 8.00h, 11.00h, 15.00h y 17.00h.
* daysdisabled: En el caso de parámetros de tipo datetimepicker, este campo permite definir qué horas concretas no será posible seleccionar en el calendario. Actualmente solo tiene utilidad para seleccionar la fecha de VC en portabilidades, las cuales solo permiten seleccionar día para VC de lunes a viernes, sábados y domingos no están habilitados (días 0 y 6).
* encriptar: Este campo determina los parámetros cuyos valores se encriptarán a la hora de guardarlos en la BBDD, en base a unos criterios definidos en la programación del portal. Se usa únicamente para encriptar contraseñas de servicios SIP. Dichas contraseñas se encriptarán con el plugin MCRYPT, que utilizará la cadena de cifrado “CarrierWPass$!” (sin comillas), en formato MD5. Después se ofuscan en BASE64, antes de almacenarlas en el portal. La desencriptación de las cadenas sigue el proceso inverso. A continuación se muestran las funciones que realizan dichos procesos, las cuales se encuentran en la ruta: /var/www/workflow4/functions/webFunctions.php:



### Tabla de solicitudes

Siempre que se solicita un producto mediante el portal, todos los valores de sus parámetros se almacenan en la tabla solicitudes, la cual posteriormente se valida a nivel de facturación.

Dicha tabla contiene multitud de columnas, las cuales contienen 2 tipos de datos:

1. Valores comunes: Estos valores son genéricos para cualquier tipo de solicitud. Es decir, independientemente del producto solicitado, cada solicitud contiene dichos valores, entre los cuales se incluyen:

* id: Identificador único de la solicitud.
* id\_reseller: Identificador del reseller *que ha realizado la solicitud*. Si el reseller es de nivel 2 en adelante (nivel distribuidor), se guarda por defecto el nivel superior. Aunque esto desencadenaría problemas en distribuidores de nivel superior a 2. Pero todavía no se ha dado el caso.

Este dato afecta a la facturación mensual. Por esta razón, en este campo siempre debe haber guardado el valor del reseller de nivel 1 (directamente relacionado con Carrier) al que pertenece la solicitud.

Dado que el portal permite la posibilidad de realizar una solicitud en nombre de otro reseller (mediante el campo “Distribuidor” en el formulario de nueva solicitud), si se diera el caso, el campo que se guardaría en la tabla corresponderá al reseller que ha realizado la solicitud, y no al que pertenece dicha solicitud. En casos así, será necesario modificar este campo id\_ por el del reseller de nivel 1.

Por ejemplo: Carrier realiza una solicitud en nombre de otro reseller (Telitec). En la solicitud, se guardará como id\_reseller=19, que corresponde a Carrier. Esto es incorrecto. Habrá que acudir a la BBDD y manualmente modificar el campo id\_reseller por el de Telitec. De lo contrario, la facturación se vería afectada negativamente.

Realmente, este campo simplemente sirve para identificar al reseller que ha realizado la solicitud. No se pensó en un primer momento con fines de facturación. Para ello, debería utilizarse la tabla solicitudes\_datos (Ver punto ), que indica las relaciones correctas entre solicitudes y sus resellers. Sin embargo, el programa que concilia la facturación lee directamente de la tabla solicitudes, en lugar de solicitudes\_datos, por lo que es necesario que en esta tabla se índice el id\_reseller de nivel 1 al que pertenece la solicitud, en lugar del que la realizó.

* id\_cliente:
* id\_producto:
* fecha\_estado:
* id\_estado:
* solicitud\_asociada:
* activada/deactivated/processing\_activation/processing\_deactivation/facturando:
* fecha\_cambio:
* fecha\_act:
* tipo\_preseleccion:

1. Valores paramétricos: Son los valores de cada uno de los parámetros de

### Orígenes de valores paramétricos

Muchos de los parámetros de la tabla productos\_dependencias\_solicitudes2, explicada antes, toman sus valores de otras tablas, en base a ciertas condiciones.

La tabla que maneja los orígenes de dichos valores es solicitudes\_valores\_campos, cuya estructura se explica a continuación:

* id: Identificador único de la lectura de valores. Este campo es el que se indica en los valores de parámetros de tipo “@V:”.
* campo\_lectura: En el caso de parámetros de tipo select, o inputselect, los textos que aparecen para seleccionar en pantalla difieren de sus valores, los cuales están ocultos. En este caso, el campo actual identifica la columna cuyos valores se leerán de una tabla determinada, pero no se mostrarán en pantalla, a no ser que el parámetro sea de tipo numeric, text, textarea, etc., en cuyo caso el valor del mismo no se puede ocultar, por lo que debe ser el mismo que el indicado en campo\_mostrar.
* campo\_mostrar: A diferencia del campo anterior, este campo determina el texto que aparecerá en pantalla para ser seleccionado por el cliente en el caso de parámetros de tipo select o inputselect, pero que puede diferir de su valor, el cual está oculto. Este campo únicamente identifica la columna de la tabla en cuestión cuyo texto aparecerá en pantalla, pero no el valor de este.

Si el campo es de tipo numeric, text, textarea, etc., en esos casos el valor del campo\_lectura coincide con el campo\_mostrar.

* solicitud: Cuando este campo está rellenado, significa que el valor o valores a buscar serán leídos de la tabla solicitudes. Puede ser de 3 tipos diferentes:
  + PARENT: Se lee el valor (campo\_lectura) y el texto a mostrar (campo\_mostrar) de la solicitud principal asociada a la actual.
  + PARQUE: Se lee el valor (campo\_lectura) y el texto a mostrar (campo\_mostrar) de todas las solicitudes asociadas al reseller.
  + SCALEASSOC-N: Dado que un producto puede depender de otro producto, que a su vez depende de otro, en infinitos niveles, N identifica el número de veces que escalaremos el nivel de solicitud, a partir de la solicitud del producto principal asociado al actual. Por ejemplo:

|  |  |
| --- | --- |
| **Nivel de escalado** | **Descripción** |
| SCALEASSOC-1 | Solicitud de la que cuelga la solicitud principal del servicio actual |
| SCALEASSOC-2 | Solicitud de la que cuelga la solicitud principal de la solicitud principal del servicio actual |
| SCALEASSOC-3 | Solicitud de la que cuelga la solicitud principal de la solicitud principal de la solicitud principal del servicio actual |

* elemento:
* elemento\_externo:
* webservice:
* c\_solicitudes:
* c\_elementos:
* c\_elementos\_externos:
* c\_webservices:
* check\_all\_assoc:
* exclude:
* result\_selectable:
* cambio\_estado:
* only\_active:
* orden:
* paginacion:

Además, las lecturas de valores que se

SOLICITUDES\_VALORES\_CONDICIONES

ELEMENTOS

ELEMENTOS EXTERNOS

### Visualización de parámetros

PRODUCTOS\_VISUAL\_SOLICITUDES2

## INCOMPATIBILIDADES Y DEPENDENCIAS ENTRE PRODUCTOS

PRODUCTOS\_DEPENDENCIAS\_ADT

PRODUCTOS\_COMPATIBILIDADES

PRODUCTOS\_INCOMPATIBILIDADES

PRODUCTOS\_DEPENDENCIAS\_PRODUCTOS

PRODUCTOS\_DEPENDENCIAS\_HERENCIA

## OTRAS CARACTERÍSTICAS DE PRODUCTOS

PRODUCTOS\_ACCIONES

PRODUCTOS\_TRANSFERIBLES

## ESTADOS Y CAMBIOS DE ESTADO

### Concepto de estado y tipos de estados

Desde que un cliente solicita un producto en nuestro portal, hasta que este está totalmente operativo, en la mayoría de casos es necesario un tiempo de provisión. Para que el reseller sea consciente de los avances en la provisión del producto, la solicitud relativa al mismo pasará por diferentes estados.

Existen 3 tipos diferentes de estados:

1. Estados de provisión: Cuando hablemos de estados, generalmente nos referiremos a este tipo, a no ser que se indique lo contrario. Son todos aquellos estados que permiten conocer en qué punto del proceso de provisión se encuentra la solicitud. Cuando cambia el estado de una solicitud, también puede cambiar su estado administrativo (cuya definición se indica a continuación), por lo que también tienen efecto en la facturación mensual, así como en las distintas plataformas activas.
2. Estados administrativos: Estos estados son transparentes a nivel de portal, es decir, funcionan de forma secundaria y no son alterables directamente, a no ser por vía BBDD. Existen 5 tipos de estados administrativos:
   1. *Procesando activación*: El servicio asociado a la solicitud todavía no está activo, ya que se encuentra en proceso de provisión.
   2. *Activado*: El servicio asociado a la solicitud está totalmente provisionado.
   3. *Procesando desactivación*: El servicio asociado a la solicitud está provisionado, pero en proceso de baja.
   4. *Desactivado*: El servicio asociado a la solicitud ha sido totalmente desprovisionado.
   5. *Facturando*: El servicio asociado a la solicitud está computando para la facturación mensual. Este estado administrativo es independiente del resto, ya que un servicio puede estar totalmente provisionado (activado) y no estar facturando todavía. O estar facturando sin estar totalmente aprovisonado, aunque este caso rara vez se produce.
3. Estados de modificación: Son independientes de los estados de provisión y administrativos, y pueden solaparse con estos. A lo largo del proceso de provisión, pueden darse situaciones en las que sea necesario que el reseller modifique ciertos datos de la solicitud. En estos casos, la solicitud adoptará un estado modificable. A diferencia de los estados de provisión y administrativos, que todas las solicitudes tienen de forma obligatoria, los estados modificables no siempre son necesarios ni tienen por qué estar aplicados. No tienen efecto en la facturación mensual ni en las distintas plataformas activas.

### Estados de provisión

La lista de estados disponibles se encuentra en la tabla solicitudes\_estados, que tiene la siguiente estructura:

* id: Identificador único del estado
* descripcion: Nombre del estado en idioma español.
* descripcion\_eng: Nombre del estado en idioma inglés.
* label: Etiqueta CSS. La plantilla del portal dispone de etiquetas de diferentes colores, las cuales se nombran como “label + color”. Se pueden crear nuevas etiquetas añadiéndolas al CSS de la plantilla, en la ruta: assets/global/css/components-rounded.css
* style: Código CSS embebido. Dado que el portal realiza envíos de emails a los resellers, indicándoles los cambios de estado que se producen en las solicitudes de productos, en estos emails también se diferencian los estados con colores, ya que el email no dispone de plantilla CSS. La cadena contenida en esta columna permite realizar esta diferenciación.

Cada producto tiene habilitados unos estados diferentes. Asimismo, cada estado determina si la solicitud relativa a un producto está o no computando en la facturación mensual. Es decir, un producto que todavía no está operativo, no computará para la facturación. Y un producto que ha sido dado de baja, tampoco.

Además, al ir pasando de un estado a otro, en un momento determinado (cambio de estado) es posible que la solicitud sea provisionada en las diferentes plataformas, o desprovisionada.

Por ejemplo: ADSL. Se provisiona en la plataforma Radius (creación del usuario) al pasar a estado “Solicitada a NP”. Y se desprovisiona al entrar en estado “Baja”.

### Cambios de estado asociados a un producto [provisión / administrativos]

La tabla que determina por qué estados puede pasar un producto, su orden, y el estado administrativo (facturable o no, provisionado o no) en que este se encuentra, la encontramos en productos\_dependencias\_estados, cuya estructura es la que sigue:

* id: Identificador único del cambio de estado
* id\_producto: Identificador del producto en cuestión
* id\_estado\_new: Identificador del nuevo estado (según la tabla solicitudes\_estados) al que es posible pasar el producto.
* id\_estado\_ant: Identificador del estado anterior (según la tabla solicitudes\_estados) desde el que es posible pasar el producto al nuevo estado.
* funcion: Trigger que se ejecuta al realizar el cambio de estado. Se identifica con un número. Las funciones trigger relativas a cambios de estado se encuentran en el archivo .php cuya ruta es: /var/www/html/workflow4/functions/triggerFunctions.php.
* activada: Puede valer 0 (No) o 1 (Sí). Este campo determina si el producto, al realizar este cambio de estado, se activa en las diferentes plataformas o no. Si está en 1 (Sí), al realizar el cambio de estado, se ejecutarán las funciones de activación, que se encuentran en el archivo .php con ruta: /var/www/html/workflow4/functions/createFunctions.php
* allowed\_reseller: Puede valer 0 (No) o 1 (Sí). Este campo habilita o no a un reseller para realizar este cambio de estado. Si está en 0 (No), el reseller no tendrá la posibilidad de aplicar este cambio de estado en el producto en cuestión.
* msj: Siempre que se ejecuta un cambio de estado, se envía un email al reseller al que pertenece la solicitud del producto, notificándole del cambio. Este campo permite, adicionalmente, añadir un mensaje a dicha notificación. El campo actual determina qué mensaje se enviará si el idioma del reseller es español.
* msj\_eng: Igual que el campo anterior. Pero, en este caso, determina qué mensaje se enviará al reseller si su idioma es inglés.
* accion: A nivel de portal, el reseller tiene la posibilidad de cambiar el estado de una solicitud mediante un botón en la tabla de búsqueda de solicitudes. Más adelante se indicarán todas las funciones de esta tabla. Al pulsar sobre este botón, aparecerán en pantalla todos los estados disponibles a los que es posible pasar la solicitud, por ejemplo: Cancelada, Tramitando, Baja. Este campo permite facilitar que un reseller entienda qué acción va a llevar a cabo con dicho cambio de estado. Por ejemplo, en el caso del estado “Baja”, al rellenar este campo con el texto “Solicitar baja”. De esa manera, al reseller le aparecerá la opción “Solicitar baja” en lugar de “Baja”, que quizás es menos intuitivo. Si se deja vacío, aparecerá el nombre del estado por defecto.
* accion\_eng: Igual que el campo anterior. Pero, en este caso, determina qué estado le aparecerá al reseller en la lista de cambios de estado disponibles, si su idioma es inglés.
* processing\_activation: Puede valer 0 (No) o 1 (Sí). Este campo determina si el producto, al realizar este cambio de estado, se encuentra en proceso de activación. Es decir, todavía no está totalmente provisionado. Aunque este campo no tiene ninguna validez a nivel de plataformas, es necesario rellenarlo correctamente para efectuar posteriores validaciones, que explicaremos más adelante.
* deactivated: Puede valer 0 (No) o 1 (Sí). Este campo determina si el producto, al realizar este cambio de estado, se desactiva en las diferentes plataformas o no. Si está en 1 (Sí), al realizar el cambio de estado, se ejecutarán las funciones de desactivación, que se encuentran en el archivo .php con ruta: /var/www/html/workflow4/functions/createFunctions.php.
* processing\_deactivation: Puede valer 0 (No) o 1 (Sí). Este campo determina si el producto, al realizar este cambio de estado, se encuentra en proceso de desactivación. Es decir, todavía no está totalmente desprovisionado. Aunque este campo no tiene ninguna validez a nivel de plataformas, es necesario rellenarlo correctamente para efectuar posteriores validaciones, que explicaremos más adelante.
* depend\_adt: Este campo ya no está en uso [Obsoleto].
* allowed\_seller: Puede valer 0 (No) o 1 (Sí). Este campo habilita o no al seller de un producto, es decir, el propietario de un producto (Carrier en el 100% de casos actualmente) para realizar este cambio de estado. Si está en 0 (No), ningún usuario del seller (Carrier) tendrá la posibilidad de aplicar este cambio de estado en el producto en cuestión.
* orden: Este campo es **imprescindible** para los productos que se provisionan de forma automática. En este tipo de productos, cuando el reseller los solicita por la web, el sistema busca el camino más rápido entre los distintos cambios de esta disponibles, para provisionar la solicitud totalmente, manteniendo el servicio activado y facturando. Además, permite visualizar de forma rápida el orden de cambios de estado de un producto. Por ejemplo:

Producto de AMLT

|  |  |  |
| --- | --- | --- |
| **Estado anterior** | **Nuevo estado** | **Orden** |
| 11 (Ninguno) | 5 (Pendiente) | 0 |
| 5 (Pendiente) | 1 (Solicitada a NP) | 1 |
| 1 (Solicitad a NP) | 4 (Finalizada) | 2 |
| 4 (Finalizada) | 8 (Baja solicitada) | 3 |
| 8 (Baja solicitada) | 9 (Baja solicitada a NP) | 4 |
| 9 (Baja solicitada a NP) | 10 (Baja) | 5 |

El ejemplo anterior no es totalmente correcto. Se trata de un caso óptimo, en el que la solicitud no ha sido denegada por ningún motivo, ni ha ocurrido problema alguno en la provisión del servicio. Sin embargo, en la mayoría de casos, pueden existir estados intermedios que también deban ser contemplados. Este campo permite tener trazabilidad del orden de los distintos estados por los que pasa una solicitud de un producto determinado.

* facturando: Puede valer 0 (No) o 1 (Sí). Este campo determina si el producto, al realizar este cambio de estado, se encuentra en estado de facturación, es decir, computa para la facturación mensual. Por tanto, un producto puede estar provisionado en las distintas plataformas (campo “activada”), y no estar facturando todavía. O estar desprovisionado (campo “deactivated”), y seguir facturando, aunque este último caso rara vez se produce.
* check\_n\_date: Puede valer 0 (No) o 1 (Sí). El portal ofrece la posibilidad de cambiar el estado de una solicitud con una fecha de aplicación (fecha de estado) posterior a la actual. En caso de que este campo valga 1, significa que la fecha de hoy debe superar a la de la aplicación del estado para que este estado esté disponible. Es decir, si un producto está en estado “Finalizada”, por ejemplo, con fecha de mañana a las 8.00AM, y este campo está en 1, no será posible realizar este cambio de estado hasta mañana a las 8.00AM, en adelante.
* check\_l\_date: Puede valer 0 (No) o 1 (Sí). En caso de que este campo valga 1, significa que la fecha de hoy debe ser inferior a la de la aplicación del estado para que este estado esté disponible. Es decir, si un producto está en estado “Finalizada”, por ejemplo, con fecha de mañana a las 8.00AM, y este campo está en 1, solo será posible realizar este cambio de estado hasta mañana a las 8.00AM. De ahí en adelante, no aparecerá en la lista de cambios de estado disponibles.
* datos\_cliente: Puede valer 0 (No) o 1 (Sí). En caso de que este campo valga 1, significa que, al realizar este cambio de estado, el reseller tendrá que rellenar en pantalla los datos del cliente final. En algunos productos, como las SIMs 4G (Acceso 4G), cuando el reseller los solicita por primera vez, desconoce los datos del cliente final. El reseller solicita una cantidad determinada de SIMs, para venderlas posteriormente. Una vez las vende, cambia la solicitud de estado “Enviada” a “Petición activación”, y asigna los datos del cliente final. Entonces, desde Carrier procedemos a la activación de la SIM y realizamos el proceso de provisión correspondiente. Este caso puede darse en otros productos similares.
* r\_exp\_date: Puede valer 0 (No) o 1 (Sí). Este campo solo se utiliza en productos que cambiarán de estado en la fecha de expiración del periodo de facturación de su servicio principal. Actualmente solo se utiliza en los productos de Ampliación 4G, de manera que utilizaremos este producto para explicar el funcionamiento de este parámetro.

Cuando un reseller solicita una SIM 4G, puede hacerlo en modo prepago o postpago. Las SIMs postpago se facturan de día 1 del mes hasta el día 1 del mes siguiente, cuando expira su periodo de facturación. Las SIMs prepago se facturan de mes en mes, comenzando por el día en que están operativas. De modo que, el periodo de facturación de las SIMs prepago no es necesariamente el día 1 del mes, sino el día correspondiente de cada mes, desde que comienzan a funcionar.

Las ampliaciones de gigas de las SIMs 4G, únicamente están activas hasta que finaliza el periodo de facturación mensual de la SIM. Por ejemplo, si la SIM es postpago, y la ampliación de gigas se solicita el día 20 del mes, dicha ampliación será válida hasta el día 1 del próximo mes, momento en el cual se dará de baja automáticamente. Aunque también puede hacerse manualmente.

Si es el caso (cambio de estado manual) este campo establece que el cambio de estado debe aplicarse obligatoriamente cuando termine el periodo de facturación del producto principal asociado. No se permitirá escoger otra fecha para el cambio de estado.

* multiple\_allowed: Puede valer 0 (No) o 1 (Sí). En algunos casos, es útil cambiar el estado de varias solicitudes al mismo tiempo. Si este campo vale 1, será posible aplicar el cambio de estado actual de forma múltiple, teniendo en cuenta que el producto y estado actual debe coincidir en todas las solicitudes.
* dias\_disponible: Este campo determina por cuántos días está disponible el cambio de estado actual, desde el último cambio de estado de la solicitud. Si vale 0, el número de días es infinito.
* maximo\_veces: Actualmente este campo solo tiene validez para las SIMs 4G. Este servicio, aunque sea dado de baja, permite la posibilidad de ser reactivado antes de 60 días desde la baja (dias\_disponible). Es decir, si una SIM se da de baja, durante 60 días es posible reactivarla de nuevo (cambio de estado de “Baja” a “Reactivación solicitada”. Sin embargo, este proceso solo puede realizarse una sola vez. Esa es la finalidad de este campo, limitar el máximo de veces que este cambio de estado se puede aplicar en una solicitud. En el caso de las SIMs 4G, si ya han sido reactivadas y posteriormente dadas de baja, no tendrán posibilidad de reactivarlas de nuevo.
* minimo\_dias: Este campo ya no está en uso [Obsoleto].
* minimo\_meses: Mínimo de meses que han debido de pasar desde el cambio de estado anterior para que esté disponible este cambio de estado. Actualmente no hay ningún cambio de estado que utilice este campo. No obstante, no está obsoleto.
* minimo\_meses\_activada: Mínimo de meses que han debido pasar desde que la solicitud está activada en las diferentes plataformas (campo “activada”). Actualmente este caso solo tiene utilidad en las SIMs 4G, ya que solo pueden ser dadas de baja, como mínimo 1 mes después de su activación.
* nsd: Puede valer 0 (No) o 1 (Sí). Si vale 1, significa que, en este cambio estado, no habrá posibilidad de escoger la fecha de aplicación. Es decir, el cambio de estado solo podrá aplicarse de forma inmediata, no con una fecha anterior o posterior a la actual. En pantalla no aparecerá la posibilidad de escoger fecha de aplicación.
* r\_act\_date: Este campo ya no está en uso [Obsoleto].
* r\_choose\_date: Puede valer 0 (No) o 1 (Sí). En caso de que valga 1, el reseller, al aplicar este cambio de estado, no tendrá posibilidad de escoger la fecha de aplicación de este cambio de estado. Es una función similar al campo “nsd”, pero solo tiene efecto para el reseller, no al seller (Carrier).
* ldatetonow: Puede valer 0 (No) o 1 (Sí). En caso de que valga 1, al realizar un retroceso de cambio de estado (devolver solicitud al estado anterior), la fecha de estado de la solicitud no será la que tenía en su estado anterior, sino la actual. Es decir, una solicitud que esté cancelada hace 3 meses, al retroceder al estado anterior (Pendiente, por ejemplo), recuperaría la fecha en la que estaba en estado “Pendiente”, de hace 3 meses. Sin embargo, si este campo está en 1 en el cambio de estado de “Pendiente” a “Cancelada”, al realizar el retroceso de cambio de estado, la fecha establecida será la actual, no la de hace 3 meses.
* r\_choose\_date\_r: Puede valer 0 (No) o 1 (Sí). En caso de que valga 1, el seller (Carrier), al aplicar este cambio de estado, no tendrá posibilidad de escoger la fecha de aplicación de este cambio de estado. Es una función similar al campo “nsd”, pero solo tiene efecto para el seller (Carrier), no a sus resellers.
* check\_pmethod: Este campo ya no está en uso [Obsoleto].
* activo: En algunos casos, es posible que un cambio de estado haya de ser inhabilitado. Bajo ningún concepto deben eliminarse de esta tabla cambios de estado, a no ser que estos no se hayan utilizado nunca en ninguna solicitud. Si el proceso de provisión de un producto ha cambiado, y se han creado nuevos cambios de estado, quedando otros obsoletos, estos últimos no deben eliminarse. El campo actual permite inhabilitarlos para que no sean utilizados posteriormente. Para inhabilitarlos, debe establecerse el valor de este campo en 0. Mientras valgan 1, estarán disponibles para su uso.
* repeat\_func: Este campo ya no está en uso [Obsoleto].
* parent\_status: Este campo determina que el cambio de estado actual sólo estará disponible si el servicio principal asociado a la solicitud en cuestión está en un estado concreto. Puede estar compuesto por 2 partes:

ID estado administrativo + [ “:” + ID modificable]

Por ejemplo:

* + 1:rq\_2 = El cambio de estado solo estará disponible si el servicio principal se encuentra en estado: Solicitada a NP + Modificación requerida.
  + 4 = El cambio de estado solo estará disponible si el servicio principal se encuentra en estado: Finalizada
* set\_date: Mediante este campo, podemos valernos de un parámetro de la solicitud para establecer la fecha de estado del cambio de estado. Actualmente solo se utiliza en servicios FTTH NEBA. Cuando TESA envía la finalización del servicio, también indica la fecha de comienzo de servicio del mismo, la cual puede ser anterior a la notificación. Esta fecha de comienzo de servicio, se guarda en un parámetro de la solicitud de FTTH. Y, al aplicar el cambio de estado de “Solicitada a NP” a “Finalizada”, la fecha de estado que se establece no es la actual, sino la de comienzo de servicio, que previamente ha sido guardada en el parámetro “fecha\_reject” de la solicitud. De esa forma, se evita perder días de facturación del servicio.
* last\_day\_min\_date: Mínima hora del día anterior para que este cambio de estado esté disponible. Actualmente solo se utiliza en el caso de portabilidades. Una vez han sido confirmadas por la AOP (Finalizadas), solo es posible cancelarlas hasta el día antes de la fecha de finalización, a la hora indicada en este campo. Una vez superada dicha fecha y hora, no será posible cancelarlas.

Esta tabla no tiene relación con los estados de modificación, ya que estos se pueden producir en cualquier momento, independientemente del estado de provisión o administrativo de la solicitud.

### Propiedad de una solicitud

Solicitudes\_datos

### Cambios de estado de una solicitud

Solicitudes\_cambios

### Cambios de estado de modificación

## CREACIÓN DE PRODUCTOS

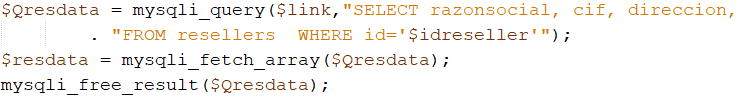
## NUEVA SOLICITUD (newrequest.php)

### Consideraciones iniciales

Este apartado permite realizar nuevas solicitudes de servicios. Al igual que otras páginas, al iniciar:

1. Verifica que la sesión sigue activa
2. Inicializa las variables de sesión: $idreseller
3. Establece el tiempo máximo de carga de la página en 0 (infinito)
4. Conecta con la BBDD del portal mediante la función conectar11()

También, antes de comenzar con la carga de la página, se obtienen todos los datos del reseller asociado a la cuenta del usuario activo:



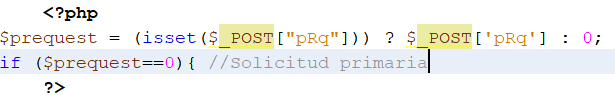
Algunos de los servicios que el reseller puede solicitar desde el portal, son para el reseller en sí como empresa, no para ningún cliente final. Por esta razón, los datos del reseller al que pertenece la cuenta del usuario activo se obtienen y guardan una sola vez en la variable $resdata. Más adelante, al realizar solicitudes, el usuario tendrá la posibilidad de indicar que la solicitud no es para un cliente final, sino para el reseller. Al marcar esta opción, se rellenarán automáticamente los datos del cliente con los obtenidos en la consulta anterior.

### Nueva solicitud primaria

Al acceder a la página actual, siempre aparecerá por defecto el formulario para solicitar un servicio principal, a no ser que tratemos de solicitar un servicio adicional sobre un servicio principal ya existente. Este último caso se explicará después, en el punto 9.1.3.

Llamamos a este proceso “Solicitud primaria” y no “Solicitud de servicio principal” ya que en este caso es posible solicitar al mismo tiempo servicios principales, así como servicios adicionales (en el caso de estos últimos, siempre y cuando no sea necesario que el servicio principal asociado esté finalizado).

En caso de que se trate de una solicitud primaria, al acceder a esta página no recibiremos ninguna variable $pRq, de tipo POST. Esta variable determina sobre qué servicio principal se va a realizar la solicitud de servicios adicionales. Al tratarse de una solicitud primaria, en la que no estamos tratando de añadir servicios a otro servicio principal ya existente. Verificaremos que se trata de una solicitud primaria con el siguiente código:



### 7.10.3. Solicitud de servicio adicional sobre servicio principal ya existente

### 7.10.4. Datos de los clientes asociados a la solicitud

### 7.10.5. Confirmación de la solicitud

## 7.11 BÚSQUEDA DE SOLICITUDES DE PRODUCTOS (searchrequests.php)