

# Documento de integración JSON API REST

NIMACH S.R.L.

versión 5.9

Fecha: 09/05/2025

Lista de cambios.....	3
Autenticación.....	6
Campo “action”.....	8
Categorías.....	9
Clientes.....	9
Información de cuentas de usuario.....	10
Sucursales de clientes.....	10
Alícuotas por cliente.....	10
Rutas de entrega.....	11
Grupos de categorías de productos.....	11
Listas de precios.....	11
Marcas.....	11
Precios.....	12
Impuestos internos.....	12
Productos.....	13
Combos.....	14
Stock por producto.....	14
Proveedores.....	14
Tipos de cliente.....	14
Comerciales.....	15
Promociones.....	15
Reglas de cobertura.....	16
Venta condicionada.....	16
Realizar consultas sobre objetos.....	17
Logotipos de marcas.....	19
Obtener lista de logotipos de marcas.....	19
Banners de marcas.....	20
Obtener lista de banners de marcas.....	20
Imágenes de productos.....	21
Obtener lista de imágenes de productos.....	22
Banners de categorías.....	22

Obtener lista de banners de categorías .....	23
<b>Pedidos .....</b>	<b>24</b>
Estructura de cabecera .....	26
Estructura de items .....	27
Estructura de impuestos .....	27
<b>Marcar un pedido exportado.....</b>	<b>28</b>
<b>Cambio de estado de pedido .....</b>	<b>28</b>
<b>End point: ABM clientes/sucursales .....</b>	<b>30</b>
<b>Subir comprobantes de facturación electrónica .....</b>	<b>31</b>
<b>Quiebres (Backorder) .....</b>	<b>32</b>
<b>Función de cálculo de fecha de entrega.....</b>	<b>33</b>
<b>Venta facturada.....</b>	<b>34</b>
<b>Cuenta corriente .....</b>	<b>35</b>

## Lista de cambios

Versión	Descripción
2.0	<ul style="list-style-type: none"><li>Realizar consultas sobre objetos: <i>Cambios en parámetro de filtro, léase ejemplo.</i></li><li>Logotipos de marcas: <i>Agregado</i></li><li>Banners de marcas: <i>Agregado</i></li><li>Banners de categorías: <i>Agregado</i></li></ul>
2.1	<ul style="list-style-type: none"><li>Productos: <i>Se agregó campo 3tems33c_pro</i></li><li>Clientes: <i>Se quitaron campos referentes a la dirección de entrega</i></li><li>Sucursales: <i>Se creó tabla de sucursales para habilitar múltiples puntos de entrega por código de cliente</i></li></ul>
2.2	<ul style="list-style-type: none"><li>Ejemplo de autenticación</li><li>Modificación de ejemplos PHP CURL</li></ul>
2.3	<ul style="list-style-type: none"><li>Modificación de estructura de promociones</li></ul>
2.4	<ul style="list-style-type: none"><li>Incorporación de meta-tags por categoría, marca y producto</li></ul>
2.5	<ul style="list-style-type: none"><li>Se agregó campo de dirección de correo electrónico de cliente</li></ul>
2.6	<ul style="list-style-type: none"><li>Se agregó campo para determinar si el cliente posee cuenta corriente</li></ul>
2.7	<ul style="list-style-type: none"><li>Se agregó acción “reset” para borrar todos los registros de una tabla</li><li>Se creó tabla de rutas de entrega para cálculo de fecha</li></ul>
2.8	<ul style="list-style-type: none"><li>Se modificó tabla pedidos:<ul style="list-style-type: none"><li>Se reemplazó total_iibb_ped por otal_impuestos_ped</li><li>Se eliminó campo obs_ped</li><li>Se crearon campo obs_preparacion_ped y obs_logistica_ped</li><li>Se agregó campo de 3tems3_lp_ped</li></ul></li><li>Se modificó tabla pedidos_items<ul style="list-style-type: none"><li>Se redefinieron campos y tipos de datos (Ver tabla)</li></ul></li></ul>
2.9	<ul style="list-style-type: none"><li>Tabla pedidos<ul style="list-style-type: none"><li>Se agregaron campos para la discriminación de cargo por envío</li><li>Se agregaron campos de tipo de pedido y tipo de logística</li><li>Se agregó campo con la IP del comprador</li></ul></li></ul>
3.0	<ul style="list-style-type: none"><li>Se modificó tabla de promociones<ul style="list-style-type: none"><li>Cambio en tipo de dato referente a cantidades</li><li>Se anexaron campos para soporte de promoción “Llevando 3 paga 2” (Cantidades configurables)</li></ul></li></ul>
3.1	<ul style="list-style-type: none"><li>Cambio de tipo de dato sobre columnas referentes a cantidad de tabla productos</li><li>Se agregó campo para la codificación de promociones</li><li>Se normalizó el dato “descrip_um_pro” de tabla productos</li></ul>
3.2	<ul style="list-style-type: none"><li>Se agregó estructura de impuestos a consulta de pedidos</li></ul>
3.3	<ul style="list-style-type: none"><li>Se agregaron campos de descuento (% , NETO y TOTAL) por línea en selección de pedidos</li></ul>
3.4	<ul style="list-style-type: none"><li>Se agregó campo CUIT de proveedor</li></ul>

- 3.5 • Se creó API para obtener la información de las cuentas de usuario
- 3.6 • Se quitó campo de correo electrónico de tabla de sucursales
- 3.7 • Se creó end point para el alta de clientes/sucursales
- 3.8 • Se creó función para subir comprobantes de facturación electrónica
- 3.9 • Se creó función para marcar pedido cómo entregado
- 3.9 • Se agregó campo de pago electrónico en clientes
- 4.0 • Se agregaron campos de transacción de pago electrónico en pedidos
- 4.0 • Se agregó campo de fecha de notificación de promoción
- 4.1 • Se creó tabla de comerciales
- 4.1 • Se agregó campo comercial a la tabla de clientes
- 4.2 • Se agregó marca de producto alimento sin T.A.C.C.
- 4.2 • Se agregó tabla de venta condicionada
- 4.3 • Se agregó costo unitario neto de producto
- 4.4 • Se agregó pago electrónico requerido en tipos de clientes
- 4.5 • Se agregó campo para clasificar productos de temperatura controlada
- 4.6 • Se creó tabla de impuestos internos
- 4.6 • Se agregó campo relacional de impuesto interno a productos
- 4.7 • Se agregó fecha de última compra a tabla de productos
- 4.8 • Se modificaron unidades de medida
- 4.9 • Se agregaron filtros por lista de precios y tipo de cliente a tabla de promociones
- 5.0 • Se agregaron filtros por comercial a la tabla de promociones
- 5.1 • Se agregó campo para definición de orden de aparición de promociones
- 5.2 • Se agregó bandera de liquidación de stock
- 5.3 • Se agregó API de cambio de estado de pedido
- 5.3 • Se eliminó API de marca de entrega (Reemplazada por punto anterior)
- 5.4 • Se agregó campo accionado\_pre a tabla de precios
- 5.5 • Se agregó campo accion\_auto\_pro a tabla de productos
- 5.6 • Se agregó tabla de quiebres (backorder)
- 5.6 • Se agregó API función de cálculo de fecha de entrega
- 5.6 • Se agregó API de venta facturada
- 5.6 • Se agregó combos a API de productos
- 5.6 • Se agregó API de cuenta corriente

- 5.7
  - Se agregó campo de información adicional en tabla de productos
- 5.8
  - Se agregó API de reglas de cobertura
  - Se agregó tabla de zonas
  - Se agregó campo de zona a sucursales de clientes
  - Se agregó campo zona a reglas de cobertura
- 5.9
  - Se agregó campo de precio sin descuento a detalle de ítems de combos



```
$result = curl_exec($ch);  
$result = json_decode($result);  
  
curl_close($ch);
```

### Salida en caso de éxito:

```
stdClass Object  
(  
    [code] => 203  
    [error] => false  
    [data] => Operación exitosa  
)
```

### Salida en caso de error:

```
stdClass Object  
(  
    [code] => 1234  
    [error] => true  
    [data] => Descripción del error  
)
```

## Campo “action”

<b>insert</b>	Inserta un registro en la tabla
<b>update</b>	Actualiza registro pre existente
<b>delete</b>	Elimina un registro pre existente
<b>insert+update</b>	Inserta un registro en la tabla, si ya existe actualiza sus datos
<b>reset</b>	Elimina todos los registros de la tabla (No requiere especificar campos claves)

Se deberá definir la estructura de datos en un array json y direccionar al objeto según corresponda.

Los campos mencionados con fuente **negrita** son obligatorios.

A continuación se detallan los objetos disponibles:



## Categorías

**Objeto:** categorias

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cat	Texto	9	
descrip_cat	Texto	255	
codigo_gca_cat	Texto	9	Código relacional de grupo de categorías
metatags_cat	Texto	512	Meta etiquetas de búsqueda separadas por coma.

## Clientes

**Objeto:** clientes

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli	Texto	30	
razonsocial_cli	Texto	255	
nombre_fantasia_cli	Texto	255	
codigo_tic_cli	Texto	9	Código relacional de tipo de cliente
tipo_doc_cli	Texto	5	Tipo de documento: DNI, CUIT, CUIL
numero_doc_cli	Número	30	N.º de documento
telefono_cli	Texto	255	Teléfono(s) de contacto
whatsapp_cli	Texto	50	Teléfono celular para comunicación WhatsApp
email_cli	Texto	255	Correo electrónico
consumidor_final_cli	Texto	1	¿Es consumidor final? S, N
empleado_cli	Texto	1	¿Es empleado? S, N
codigo_lp_cli	Texto	9	Código de relacional de lista de precios asignada
exento_cli	Texto	1	¿Es exento de I.V.A.? S, N
limite_de_credito_cli	Decimal	13E+2D	Importe en pesos de límite de crédito disponible
email_cli	Texto	255	Dirección de correo electrónico
cuenta_corriente_cli	Texto	1	¿El cliente posee cuenta corriente? S, N
pago_electronico_cli	Texto	1	¿Pago electrónico? S, N (O = Obligatorio)
codigo_com_cli	Numérico	9	Código del comercial

## Información de cuentas de usuario

**Descripción:** Devuelve un listado con la información actualizada de las cuentas de usuario.

**Objeto:** usuarios/select

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli_usr	Texto	30	Código relacional de cliente
alias_usr	Texto	20	Cuenta de usuario
email_usr	Texto	255	Dirección de correo electrónico
email_verificado_usr	Texto	1	Cuenta de correo electrónico verificada (S/N)
telefono_usr	Texto	50	Teléfono de línea
whatsapp_usr	Texto	30	Número de WhatsApp
web_usr	Texto	255	Sitio Web

## Sucursales de clientes

**Objeto:** clientes\_sucursales

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli_suc	Texto	30	Código relacional de cliente
numero_suc	Numérico	4	Número de sucursal (único dentro del código de cliente)
direccion_suc	Texto	255	
pais_suc	Texto	30	
provincia_suc	Texto	50	
localidad_suc	Texto	50	
cp_suc	Texto	10	Código postal
telefono_suc	Texto	255	Teléfono(s) de contacto
whatsapp_suc	Texto	50	Teléfono celular para comunicación WhatsApp
codigo_zon_suc	Texto	15	Código de zona

## Alícuotas por cliente

**Objeto:** clientes\_alicuotas

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli_ali	Texto	30	Código relacional de cliente
descrip_ali	Texto	30	Descripción impositiva, ejemplo: IIBB CABA
coeficiente_ali	Decimal	3E+2D	Tasa %

## Rutas de entrega

**Descripción:** Tabla para cálculo de fecha de entrega basado rutas pre programadas.

**Objeto:** clientes\_rutas

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli_rut	Texto	30	Código relacional de cliente
codigo_suc_rut	Numérico	4	Código relacional de sucursal
dia_rut	Texto	3	Día de entrega según ruta de reparto.
	<b>Valor</b>		<b>Descripción</b>
	dom		Domingos
	lun		Lunes
	mar		Martes
	mie		Miércoles
	jue		Jueves
	vie		Viernes
	sab		Sábados
	*		Todos los días de la semana

## Grupos de categorías de productos

**Objeto:** grupos\_de\_categorias

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_gca	Texto	9	
descrip_gca	Texto	255	

## Listas de precios

**Objeto:** listas\_de\_precios

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_lp	Texto	9	Código/N° de Lista
descrip_lp	Texto	255	Descripción referencia de lista
consumidor_final_lp	Texto	1	¿Es la lista por defecto para consumidores finales? S, N
iva_incluido_lp	Texto	1	¿El I.V.A. de los precios de ésta lista está incluido en los precios? S, N

## Marcas

**Objeto:** marcas

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_mar	Texto	9	
descrip_mar	Texto	255	
metatags_mar	Texto	512	Meta etiquetas de búsqueda separadas por coma.

## Precios

**Objeto:** precios

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_lp_pre	Texto	9	Código relacional de lista de precios
codigo_pro_pre	Texto	30	Código relacional de producto
precio_pre	Decimal	13E+2D	Precio unitario
accionado_pre	Texto	1	¿El precio está accionado? S/N

## Impuestos internos

**Objeto:** impuestos\_internos

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_ii	Texto	9	Código de impuesto
descrip_ii	Texto	255	Descripción
tasa_ii	Decimal	3E+2D	Tasa

## Productos

**Objeto:** productos

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_pro	Texto	30	
descrip_larga_pro	Texto	255	Descripción larga del producto
descrip_corta_pro	Texto	100	Descripción corta
codigo_mar_pro	Texto	9	Código relacional de marca
codigo_prv_pro	Texto	9	Código relacional de proveedor
codigo_cat_pro	Texto	9	Código relacional de categoría
ean13_pro	Texto	20	Código de barras EAN13
costo_uni_netto_pro	Decimal	13E+2D	Costo unitario neto de producto
tasa_iva_pro	Decimal	3E+2D	Porcentaje de I.V.A.
activo_pro	Texto		¿El producto está activo? S, N
disponible_pro	Texto		¿El producto está disponible para la venta? S, N
cant_min_vta_pro	Numérico	11	Cantidad mínima por venta
cant_max_vta_pro	Numérico	11	Cantidad máxima por venta
venta_uni_sueltas_pr	Texto	1	¿Permite vender unidades sueltas? S, N
cant_x_bulto_pro	Numérico	11	Cantidad de unidades por bulto
cant_fraccion_pro	Numérico	11	Venta en fracciones de.. Ejemplo: Si define 3 puede comprar 3, 6, 9, 12, etc..
pesable_pro	Texto	1	¿El producto se vende por kilo? S, N
peso_x_unidad_pro	Decimal	13E+2D	Peso en kilogramos por unidad (En productos pesables refiere a peso de la pieza, horma, etc.)
descrip_um_pro	Texto	25	Unidad de venta del producto <b>Unidades disponibles:</b> UNIDADES CAJAS HORMAS PIEZAS LATAS SOBRES BOLSAS PACKS KILOS
creacion_pro	Fecha Hora	20	aaaa-mm-dd hh:mm:ss - Fecha y hora de creación del producto en el sistema local. SE UTILIZA PARA RECONOCER CUALES SON LAS NOVEDADES Y DESTACARLAS.
metatags_pro	Texto	512	Meta etiquetas de búsqueda separadas por coma.
alimento_sin_tacc_pro	Texto	1	¿Producto es un alimento sin T.A.C.C.? S, N
temp_controlada_pro	Texto	1	¿Producto requiere temperatura controlada? S, N
codigo_ii_pro	Texto	9	Código de impuesto interno (Solo si aplica)

fecha_uc_pro	Fecha	10	Fecha de última compra (opcional)
accion_auto_pro	Texto	1	¿Activar acciones automáticas sobre este producto? S, N
info_pro	Texto	999999	Información adicional del producto

## Combos

**Descripción:** Para crear combos se debe agregar la propiedad de tipo array “combo” a la estructura de registro de productos.

Campo	Tipo	Longitud	Descripción
codigo_pro	Texto	30	Código relacional de producto
cantidad	Decimal	13E+2D	Cantidad de este producto dentro del combo
descuento	Decimal	5E+2D	Porcentaje de descuento sobre el precio de lista del producto (Aplicable a este producto dentro del combo)
precio_sin_dto	Decimal	13E+2D	Precio del producto sin el descuento aplicado

## Stock por producto

**Objeto:** productos\_stock

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_pro_stk	Texto	30	Código relacional de producto
cantidad_stk	Decimal	13E+2D	Cantidad en stock disponible para la venta
liquidar_stk	Texto	1	Liquidar stock S/N (Omite validaciones de volumen por pedido)

## Proveedores

**Objeto:** proveedores

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_prv	Texto	30	
razonsocial_prv	Texto	255	
cuit_prv	Numérico	30	N.º de CUIT del proveedor

## Tipos de cliente

**Objeto:** tipos\_de\_cliente

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_tic	Texto	9	
descrip_tic	Texto	255	
pago_electronico_tic	Texto	1	¿Pago electrónico requerido? S, N

## Comerciales

**Objeto:** comerciales

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_com	Texto	9	
nombre_com	Texto	255	Nombre y apellido del comercial
whatsapp_com	Texto	50	Teléfono celular para comunicación WhatsApp
email_com	Texto	255	Correo electrónico

## Promociones

**Objeto:** promociones

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_prm	Texto	9	Código de la promoción
codigo_pro_prm	Texto	30	Código relacional de producto
desde_prm	Fecha	10	(aaaa-mm-dd) Precio promocional vigente a partir de fecha.
vigencia_prm	Fecha	10	(aaaa-mm-dd) Precio promocional vigente hasta la fecha inclusive.
precio_prm	Decimal	13E+2D	Precio promocional
iva_incluido_prm	Texto	1	¿El I.V.A. de los precios de ésta lista está incluido en los precios? S, N
cant_min_prm	Numérico	11	Cantidad mínima para acceder a la promoción (0 sin mínimo de compra)
cant_max_prm	Numérico	11	Cantidad máxima para acceder a la promoción (0 sin máximo de compra)
cant_paga_prm	Numérico	11	Cantidad que paga cada N cantidad que compra. (0 para omitir)
cant_compra_prm	Numérico	11	Variable N de campo cant_paga_prm. (0 para omitir)
notif_fecha_prm	Fecha	10	(aaaa-mm-dd) Fecha de notificación de la promoción
codigo_lp_prm	Texto	512	Códigos de lista de precios habilitadas separadas por coma
codigo_tic_prm	Texto	512	Códigos de tipos de cliente habilitados separados por coma
codigo_com_prm	Texto	512	Códigos de comerciales habilitados separados por coma
orden_prm	Numérico	11	Orden de aparición en página principal

\* *Clave única secundaria: codigo\_pro\_prm + precio\_prm + iva\_incluido\_prm*

\* *No especificar lista de precios / tipo de cliente si no es necesario el filtro*

## Reglas de cobertura

**Descripción:** Tabla para habilitar o deshabilitar venta de productos por combinación de parámetros.

**Objeto:** reglas\_de\_cobertura

**Método:** PUT

Campo	Tipo	Longitud	Descripción	
pais_rco	Texto	3	Código de país	
			<table border="1"> <thead> <tr> <th>Código</th> <th>Nombre</th> </tr> </thead> <tbody> <tr> <td>AR</td> <td>Argentina</td> </tr> </tbody> </table>	Código
Código	Nombre			
AR	Argentina			
provincia_rco	Texto	5	Letra de provincia	
localidad_rco	Texto	50	Nombre de localidad	
cp_rco	Texto	10	Código postal	
codigo_prv_rco	Texto	30	Código de proveedor	
codigo_mar_rco	Texto	9	Código de marca	
codigo_cat_rco	Texto	9	Código de categoría	
codigo_pro_rco	Texto	30	Código de producto	
codigo_tic_rco	Texto	9	Código de tipo de cliente	
codigo_cli_rco	Texto	9	Código de cliente	
codigo_com_rco	Texto	9	Código de comercial	
codigo_zon_rco	Texto	15	Código de zona	
prioridad_rco	Numérico	1	Prioridad de la regla (0 a 20) (0 para máxima prioridad)	
habilitado_rco	Numérico	1	1 = Venta permitida 0 = Venta no permitida	

## Venta condicionada

**Objeto:** venta\_condicionada

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_vta_vco	Texto	34	Código de producto 1
codigo_con_vco	Texto	34	Código de producto a comprar en forma obligatoria para poder comprar producto 1



## Realizar consultas sobre objetos

### Método: PUT

Para realizar consultas sobre los objetos, es decir obtener una lista de los registros exportados hacia la misma se deberá agregar **/select** al final del mismo. Por ejemplo para obtener la lista de las categorías definidas en la plataforma debe ejecutar el api `"https://www.pedifull.com/api/categorias/select/"`

### Ejemplo PHP CURL:

```
$ch = curl_init();

$data_json = json_encode([
    "filter" => [
        [
            /* Nombre del campo */
            "field" => "codigo_cat",

            /* Comparador lógico [= | < | > | <> | LIKE] */
            /* Para campos de texto siempre debe usarse LIKE, soporta % */
            "comparator" => "=",

            /* Valor de filtro */
            "value" => "8889",

            /* Siguiete comparador [AND | OR]*/
            "next_filter" => "OR"
        ],
        [
            "field" => "codigo_cat",
            "comparator" => "<>",
            "value" => "9999"
        ]
    ] /* ← Filtro opcional */
]);

curl_setopt($ch, CURLOPT_USERPWD , "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL, "https://www.pedifull.com/api/object/");
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLINFO_HEADER_OUT, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json', 'Content-Length: ' . strlen($data_json)));

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

### Salida:

```
stdClass Object
(
    [error] =>
    [results] => Array
        (

```

```
[0] => stdClass Object
(
    [codigo_cat] => 0102
    [descrip_cat] => HAMBURGUESAS
    [codigo_gca_cat] => 01
    [log_cat] => 2020-08-31 20:17:14
)

[1] => stdClass Object
(
    [codigo_cat] => 0104
    [descrip_cat] => PESCADOS
    [codigo_gca_cat] => 01
    [log_cat] => 2020-08-31 20:17:14
)

[2] => stdClass Object
(
    [codigo_cat] => 0106
    [descrip_cat] => PREFRITOS POLLO
    [codigo_gca_cat] => 01
    [log_cat] => 2020-08-31 20:17:14
)

)

)
```

## Errores

En caso de ocurrir un error la variable **error** estará en **True** y se añadirá a su mismo nivel la propiedad “**error\_desc**” que detallará el motivo del mismo.

## Salida:

```
stdClass Object
(
    [error] => 1
    [error_desc] => You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near...
)
```

## Logotipos de marcas

**Objeto:** marcas\_logotipos

**Método:** POST

Se deben subir de a una las imágenes de los marcas indicando su código. En el siguiente ejemplo PHP CURL se detalla la estructura JSON que debe acompañar dicha subida:

```
$file = "C:\Archivos\marca123.jpg"; /* <- Ruta local del archivo */
$mime = mime_content_type($file);
$info = pathinfo($file);
$name = $info['basename'];
$upload_file = new CURLFile($file, $mime, $name);

$data = array(
    "123" => $upload_file, /* <- Código de marca */
    "data" => json_encode([
        'action' => 'insert+update',
        'codigo_mar' => "123", /* <- Código de marca */
    ])
);

$data_json = json_encode($data);

$ch = curl_init();
curl_setopt($ch, CURLOPT_USERPWD, "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL, "https://www.pedifull.com/api/marcas_logotipos/");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

## Obtener lista de logotipos de marcas

**Objeto:** marcas\_logotipos/select

**Método:** PUT

**Nota:** Se debe utilizar esta función para determinar la necesidad de actualizar el archivo de imagen con el fin de evitar subirlo recursivamente.

Campo	Tipo	Longitud	Descripción
codigo_mar_img	Texto	9	Código relacional de marca
log_img	Fecha Hora	20	Fecha y hora de actualización. <b>SE RECOMIENDA COMPARAR LA FECHA DE MODIFICACIÓN DEL ARCHIVO LOCAL CONTRA ESTE VALOR PARA EVALUAR SI LA IMAGEN DEBE SER ACTUALIZADA.</b>

## Banners de marcas

**Objeto:** marcas\_banners

**Método:** POST

Las imágenes deben ser subidas de una indicando el código de marca a la cual responden. En el siguiente ejemplo PHP CURL se detalla la estructura JSON que debe acompañar dicha subida:

```
$file = "C:\Archivos\banner_marca123.png"; /* <- Ruta local del archivo */
$mime = mime_content_type($file);
$info = pathinfo($file);
$name = $info['basename'];
$upload_file = new CURLFile($file, $mime, $name);

$data = array(
    "123" => $upload_file, /* <- Código de marca */
    "data" => json_encode([
        'action' => 'insert+update',
        'codigo_mar' => "123", /* <- Código de marca */
    ])
);

$data_json = json_encode($data);

$ch = curl_init();
curl_setopt($ch, CURLOPT_USERPWD, "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL, "https://www.pedifull.com/api/marcas_banners/");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

## Obtener lista de banners de marcas

**Objeto:** marcas\_banners/select

**Método:** PUT

**Nota:** Se debe utilizar esta función para determinar la necesidad de actualizar el archivo de imagen con el fin de evitar subirlo recursivamente.

Campo	Tipo	Longitud	Descripción
codigo_mar_img	Texto	9	Código relacional de marca
log_img	Fecha Hora	20	Fecha y hora de actualización. <b>SE RECOMIENDA COMPARAR LA FECHA DE MODIFICACIÓN DEL ARCHIVO LOCAL CONTRA ESTE VALOR PARA EVALUAR SI LA IMAGEN DEBE SER ACTUALIZADA.</b>

## Imágenes de productos

**Objeto:** productos\_imagenes

**Método:** POST

Se deben subir de a una las imágenes de los productos indicando su EAN13. En el siguiente ejemplo PHP CURL se detalla la estructura JSON que debe acompañar dicha subida:

```
$file = "C:\Archivos\producto1234.jpg"; /* <- Ruta local del archivo */
$mime = mime_content_type($file);
$info = pathinfo($file);
$name = $info['basename'];
$upload_file = new CURLFile($file, $mime, $name);

$data = array(
    "1234567891000" => $upload_file, /* <- Código de barras EAN13 */
    "data" => json_encode([
        'action' => 'insert+update',
        'codigo' => "SKU1234", /* <- Código de producto */
        'ean13' => "1234567891000" /* <- Código de barras EAN13 */
    ])
);

$data_json = json_encode($data);

$ch = curl_init();
curl_setopt($ch, CURLOPT_USERPWD, "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL,
    "https://www.pedifull.com/api/productos_imagenes/");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

## Obtener lista de imágenes de productos

**Objeto:** productos\_imagenes/select

**Método:** PUT

**Nota:** Se debe utilizar esta función para determinar la necesidad de actualizar el archivo de imagen con el fin de evitar subirlo recursivamente.

Las imágenes son compartidas mediante su EAN13 con otras tiendas que vendan el mismo producto y carezcan de la misma. Esta colaboración es bidireccional por lo que podrá también omitir la subida del archivo sobre los productos en caso de que se indique que se dispone de tal colaboración.

Campo	Tipo	Longitud	Descripción
codigo_pro_img	Texto	30	Código relacional de producto
propia_img	Texto	1	¿Tiene imagen definida por mi? S, N
compartida_img	Texto	1	¿Tiene imagen definida por otra tienda que vende el mismo producto? S, N
log_img	Fecha Hora	20	Fecha y hora de actualización (aaaa-mm-dd hh:mm:ss) <i>SE RECOMIENDA COMPARAR LA FECHA DE MODIFICACIÓN DEL ARCHIVO LOCAL CONTRA ESTE VALOR PARA EVALUAR SI LA IMAGEN DEBE SER ACTUALIZADA.</i>
log_cmp	Fecha Hora	20	Fecha y hora de actualización efectuada por otra tienda (aaaa-mm-dd hh:mm:ss)

## Banners de categorías

**Objeto:** categorias\_banners

**Método:** POST

Las imágenes deben ser subidas de una indicando el código de categoría a la cual responden. En el siguiente ejemplo PHP CURL se detalla la estructura JSON que debe acompañar dicha subida:

```
$file = "C:\Archivos\banner_categoria_123ABC.png"; /* <- Ruta local del archivo */
$mime = mime_content_type($file);
$info = pathinfo($file);
$name = $info['basename'];
$upload_file = new CURLFile($file, $mime, $name);

$data = array(
    "123ABC" => $upload_file, /* <- Código de categoría */
    "data" => json_encode([
        'action' => 'insert+update',
        'codigo_cat' => "123ABC", /* <- Código de categoría */
    ])
);

$data_json = json_encode($data);

$ch = curl_init();
curl_setopt($ch, CURLOPT_USERPWD, "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
```

```
curl_setopt($ch, CURLOPT_URL,
"https://www.pedifull.com/api/categorias_banners/");
curl_setopt($ch, CURLOPT_POST,1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);

$result = curl_exec($ch);
$result = json_decode($result);

curl_close ($ch);
```

## Obtener lista de banners de categorías

**Objeto:** categorias\_banners/select

**Método:** PUT

**Nota:** Se debe utilizar esta función para determinar la necesidad de actualizar el archivo de imagen con el fin de evitar subirlo recursivamente.

Campo	Tipo	Longitud	Descripción
codigo_cat_img	Texto	9	Código relacional de categoría
log_img	Fecha Hora	20	Fecha y hora de actualización. <i>SE RECOMIENDA COMPARAR LA FECHA DE MODIFICACIÓN DEL ARCHIVO LOCAL CONTRA ESTE VALOR PARA EVALUAR SI LA IMAGEN DEBE SER ACTUALIZADA.</i>

## Pedidos

**Objeto:** pedidos/select

**Método:** PUT

Para obtener un listado de pedidos cargados en la plataforma debe aplicar la misma metodología de consulta indicada en **“Realizar consultas sobre objetos”**.

A diferencia del resto de las tablas, esta expresa sus items en modo árbol de datos:

**Salida:**

```
stdClass Object
(
    [error] =>
    [results] => Array
        (
            [0] => stdClass Object
                (
                    [numero_ped] => 1
                    [codigo_cli_ped] => 00700
                    [codigo_lp_ped] => 01
                    [fecha_ped] => 2020-09-03
                    [obs_preparacion_ped] => SI NO HAY X 6 MANDARME X 12
                    [obs_logistica_ped] => DE 09:00 A 18:00 PREGUNTAR POR JUAN
                    [total_ng_ped] => 1342.00
                    [total_ne_ped] => 0.00
                    [total_iva_ped] => 21.00
                    [total_impuestos_ped] => 0.00
                    [total_envio_ng_ped] => 0.00
                    [total_envio_iva_ped] => 0.00
                    [total_ped] => 1623.82
                    [ent_fecha_ped] => 2020-09-04
                    [ent_direccion_ped] => CALLE 99
                    [ent_pais_ped] => AR
                    [ent_provincia_ped] => BS AS
                    [ent_localidad_ped] => CABA
                    [ent_cp_ped] => 1
                    [ent_tipo_doc_ped] => DNI
                    [ent_numero_doc_ped] => 32143412
                    [ent_telefono_ped] => 1143694425
                    [ent_whatsapp_ped] => 1133624428
                    [tipo_de_pedido_ped] => b2b
                    [tipo_de_logistica_ped] => normal
                    [exportado_ped] => N
                    [creado_log_ped] => 2020-09-03 18:59:16
                    [exportado_log_ped] =>
                    [creado_log_ped] => 2020-09-03 18:59:16
                    [forma_de_pago_ped] => CR
                    [pago_pendiente_ped] => N
                    [pago_transaccion_ped] => null
                    [impuestos] => Array
                        (
                            [0] => stdClass Object
                                (
                                    [descrip_imp] => INGRESOS BRUTOS PROV. BS. AS.
                                    [tasa_imp] => 1.60
                                    [importe_imp] => 48.00
                                )
                            )
                    [items] => Array
                        (
```



```
[0] => stdClass Object
(
    [item_itm] => 1
    [codigo_pro_itm] => 1000
    [precio_net0_itm] => 50.00
    [tasa_iva_itm] => 21.00
    [precio_final_con_iva_itm] => 60.50
    [oferta_itm] => N
    [cantidad_itm] => 1
    [pesable_itm] => N
    [kilos_itm] => 0.00
    [dto_porc_itm] => 0.00
    [dto_net0_itm] => 0.00
    [dto_total_itm] => 0.00
)
[1] => stdClass Object
(
    [item_itm] => 2
    [codigo_pro_itm] => 1001
    [precio_net0_itm] => 50.00
    [tasa_iva_itm] => 21.00
    [precio_final_con_iva_itm] => 60.50
    [oferta_itm] => N
    [cantidad_itm] => 1
    [pesable_itm] => N
    [kilos_itm] => 0.00
)
[2] => stdClass Object
(
    [item_itm] => 2
    [codigo_pro_itm] => 5005
    [precio_net0_itm] => 138.00
    [tasa_iva_itm] => 21.00
    [precio_final_con_iva_itm] => 166.98
    [oferta_itm] => N
    [cantidad_itm] => 2
    [pesable_itm] => S
    [kilos_itm] => 9.00
    [dto_porc_itm] => 0.00
    [dto_net0_itm] => 0.00
    [dto_total_itm] => 0.00
)
)
)
)
```

## Estructura de cabecera

Campo	Tipo	Longitud	Descripción						
numero_ped	Numérico	20	Número de pedido generado						
codigo_cli_ped	Texto	30	Código relacional de cliente						
codigo_lp_ped	Texto	9	Código relacional de lista de precios de cliente						
fecha_ped	Fecha	10	Fecha de pedido						
obs_preparacion_ped	Texto	255	Observaciones relacionadas a la preparación del pedido						
obs_logistica_ped	Texto	255	Observaciones para repartidor						
total_ng_ped	Decimal	13E+2D	Neto gravado total						
total_ne_ped	Decimal	13E+2D	Neto exento total						
total_iva_ped	Decimal	13E+2D	I.V.A. total						
total_impuestos_ped	Decimal	13E+2D	Total de impuestos calculados según tabla clientes_alicutas						
total_envio_ng_ped	Decimal	13E+2D	Neto gravado de cargo por envío						
total_envio_iva_ped	Decimal	13E+2D	I.V.A. de cargo por envío						
total_ped	Decimal	13E+2D	Importe total						
ent_fecha_ped	Fecha	10	Fecha de entrega calculada						
ent_direccion_ped	Texto	255	Dirección de entrega						
ent_pais_ped	Texto	30	País de entrega						
ent_provincia_ped	Texto	50	Provincia de entrega						
ent_localidad_ped	Texto	50	Localidad de entrega						
ent_cp_ped	Texto	10	Código postal de entrega						
ent_tipo_doc_ped	Texto	5	Tipo de documento (CUIL, CUIT, DNI)						
ent_numero_doc_ped	Texto	30	N.º de documento						
ent_telefono_ped	Texto	255	Teléfonos						
ent_whatsapp_ped	Texto	50	Teléfono WhatsApp						
tipo_de_pedido_ped	Texto	15	<table border="1"> <thead> <tr> <th>Tipo</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>b2b</td> <td>Negocio a negocio</td> </tr> <tr> <td>b2c</td> <td>Negocio a cliente</td> </tr> </tbody> </table>	Tipo	Descripción	b2b	Negocio a negocio	b2c	Negocio a cliente
Tipo	Descripción								
b2b	Negocio a negocio								
b2c	Negocio a cliente								
tipo_de_logistica_ped	Texto	15	<table border="1"> <thead> <tr> <th>Tipo</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>normal</td> <td>Envío normal</td> </tr> <tr> <td>pickup</td> <td>Retira comprador</td> </tr> </tbody> </table>	Tipo	Descripción	normal	Envío normal	pickup	Retira comprador
Tipo	Descripción								
normal	Envío normal								
pickup	Retira comprador								
network_ip_ped	Texto	20	IP del comprador						
exportado_ped	Texto	1	¿El pedido ya fue exportado? S, N						
creado_log_ped	Fecha Hora	20	Fecha y hora de creación (aaaa-mm-dd hh:mm:ss)						
exportado_log_ped	Fecha Hora	20	Fecha y hora de exportación (aaaa-mm-dd hh:mm:ss)						

## Estructura de ítems

Campo	Tipo	Longitud	Descripción
item_itm	Numérico	11	Número de línea
codigo_pro_itm	Texto	30	Código relacional de producto
precio_netto_itm	Decimal	12E+3D	Precio unitario sin I.V.A. <i>(En pesables refiere a peso por kilogramo)</i>
tasa_iva_itm	Decimal	3E+2D	Porcentaje de I.V.A.
precio_final_con_iva_itm	Decimal	12E+3D	Precio unitario con I.V.A. <i>(En pesables refiere a peso por kilogramo)</i>
oferta_itm	Texto	1	¿Es una precio promocional? S, N
cantidad_itm	Numérico	11	Cantidad de unidades pedidas <i>(En pesables refiere a cantidad de piezas)</i>
pesable_itm	Texto	1	¿Es un producto pesable? S, N
kilos_itm	Decimal	13E+2D	Peso total de unidades pedidas (Sólo para productos pesables)
dto_porc_itm	Decimal	3E+2D	Porcentaje de descuento aplicado
dto_total_netto_itm	Decimal	13E+2D	Importe total neto del descuento aplicado
dto_total_con_iva_itm	Decimal	13E+2D	Importe total con I.V.A. del descuento aplicado

## Estructura de impuestos

Campo	Tipo	Longitud	Descripción
descrip_imp	Texto	255	Descripción del impuesto
tasa_imp	Decimal	3E+2D	Alícuota aplicada
importe_imp	Decimal	13E+2D	Importe

## Marcar un pedido exportado

Los pedidos generados en la plataforma nacen con su campo **exportado\_ped** en “N” el cual puede ser utilizado de filtro para obtener aquellos pedidos pendientes de integración en su sistema local. Para marcar este pedido cómo exportado a fin de que ya no entre en dicho filtro se debe hacer la llamada a “<https://www.pedifull.com/api/pedidos/exportado/>”

Ejemplo PHP CURL:

```
$ch = curl_init();

$filter = json_encode([
    "numero_ped" => 1234
]);

curl_setopt($ch, CURLOPT_USERPWD , "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL,
"https://www.pedifull.com/api/pedidos/exportado/");
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLINFO_HEADER_OUT, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, $filter);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json','Content-Length: ' . strlen($filter)));

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

Salida en caso de éxito:

```
stdClass Object
(
    [code] => 203
    [error] => false
    [data] => Operación exitosa
)
```

Salida en caso de error:

```
stdClass Object
(
    [code] => 1234
    [error] => true
    [data] => Descripción del error
)
```

## Cambio de estado de pedido

Para cambiar el estado de un pedido se debe hacer la llamada a “<https://www.pedifull.com/api/pedidos/estado/>”

Estado	Descripción
pendiente	Pendiente de preparación
en_preparacion	En preparación

preparado	Preparado listo para despachar
facturado	Pedido facturado
transito	En tránsito (camino a la entrega)
entregado	El pedido fue entregado
rechazado	El pedido fue rechazado
cancelado	El pedido fue cancelado
reprogramado	La entrega fue reprogramada

## Ejemplo PHP CURL:

```
$ch = curl_init();

$filter = json_encode([
    "numero_ped" => 1234, /* ← Número de pedido */
    "fecha" => "2021-06-03", /* ← Fecha de cambio de estado AAAA-MM-DD */
    "hora" => "05:40", /* ← Hora de cambio de estado HH:MM */
    "estado" => "entregado" /* ← Estado */
]);

curl_setopt($ch, CURLOPT_USERPWD, "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL, "https://www.pedifull.com/api/pedidos/estado/");
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLINFO_HEADER_OUT, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, $filter);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json','Content-Length: ' . strlen($filter)));

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

## End point: ABM clientes/sucursales

**Objeto:** Configurable

**Método:** PUT

Al crear, modificar o eliminar un cliente/sucursal el backend puede enviar un paquete de datos a un end point configurado en mi tienda.

Ejemplo JSON:

```
{
  "funcion": "alta",
  "datos": {
    "codigo_cli": "00123",
    "numero_suc": 1,
    "razonsocial_cli": "JUAN RAMON AYALA",
    "direccion_suc": "AV. ESTONIA 5193",
    "cp_suc": "1499",
    "localidad_suc": "CAPITAL FEDERAL",
    "tipo_doc_cli": "CUIT",
    "numero_doc_cli": "20999999997",
    "telefono_suc": "49991234",
    "whatsapp_suc": "1149999999",
    "email_cli": "juan@email.com.ar",
    "codigo_tic_cli": "07",
    "consumidor_final_cli": "N",
    "limite_de_credito_cli": 15000
  }
}
```

En las funciones de **alta**, **modificación** y **baja** debe retornarse el resultado de la operación en el sistema externo que captura el evento.

En la función **obtener\_nuevo\_codigo** debe devolverse el siguiente código disponible para el alta de un cliente dentro del sistema que va a capturar dicho evento.

Función	Retorno esperado
<b>alta</b>	<pre>{   "result": true/false }</pre>
<b>modificación</b>	
<b>baja</b>	<pre>{   "result": true/false   "data": {     "codigo_cli": "00123"   } }</pre>
<b>obtener_nuevo_codigo</b>	

## Subir comprobantes de facturación electrónica

**Objeto:** comprobantes\_electronicos

**Método:** POST

Se deben subir de a uno los documentos en formato pdf acompañados por un hash de identificación. En el siguiente ejemplo PHP CURL se detalla la estructura JSON que debe acompañar dicha subida:

```
$file = "C:\Archivos\FACA0001-00003625.pdf"; /* <- Ruta local del archivo */
$mime = mime_content_type($file);
$info = pathinfo($file);
$name = $info['basename'];
$upload_file = new CURLFile($file, $mime, $name);

$data = array(
    "ca6dd89d1c367877dd859745db836242" => $upload_file, /* <- Hash */
    "data" => json_encode([
        'action' => 'insert',
        'hash' => "ca6dd89d1c367877dd859745db836242", /* <- Hash */
        'numero_ped_cbte' => 35148, /* Pedido pedíFull */
        'tipo_cbte' => 'F', /* [F=FACTURA/C=CREDITO/D=DEBITO] */
        'letra_cbte' => 'A', /* [A/B/C/M/E] */
        'sucursal_cbte' => '0000',
        'numero_cbte' => 3625
    ])
);

$data_json = json_encode($data);

$ch = curl_init();
curl_setopt($ch, CURLOPT_USERPWD, "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL,
    "https://www.pedifull.com/api/comprobantes_electronicos/");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

## Quiebres (Backorder)

**Descripción:** Actualización de cantidades enviadas a quiebre al procesar pedido en sistema de facturación local.

**Objeto:** backorder

**Método:** PUT

Campo	Tipo	Longitud	Descripción
numero_ped	Numérico	20	Número de pedido en pedíFull
codigo_pro	Texto	30	Código de producto
backorder	Decimal	13E+2D	Cantidad enviada a quiebre al procesar el pedido



## Función de cálculo de fecha de entrega

**Descripción:** Devuelve fecha de entrega calculada partiendo del parámetro de fecha de alta.

**Objeto:** funciones/calcular\_fecha\_de\_entrega\_desde\_alta

**Método:** PUT

Ejemplo PHP:

```
$data = array(
    "params" => array(
        [
            "name" => "codigo_cli", /* <- Código de cliente */
            "value" => "1234"
        ],
        [
            "name" => "numero_suc", /* <- Número de succursal */
            "value" => "1"
        ],
        [
            "name" => "fecha_hora_alta", /* Fecha base de cálculo */
            "value" => "2023-09-27"
        ]
    )
);

$data_json = json_encode($data);

$ch = curl_init();

curl_setopt($ch, CURLOPT_USERPWD , "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL,
"https://www.pedifull.com/api/funciones/calcular_fecha_de_entrega_desde_alta/");
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLINFO_HEADER_OUT, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json','Content-Length: ' . strlen($data_json)));

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

**Salida:**

```
stdClass Object
(
    [results] => Array
        (
            [0] => stdClass Object
                (
```

```
[fecha] => "2023-09-30"
)
)
)
```

## Venta facturada

**Descripción:** Permite subir la facturación de todos los canales de venta con el fin de ser considerada en los algoritmos que crean cupones de acciones comerciales inteligentes.

**Objeto:** venta\_facturada

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli_fac	Texto	30	Código de cliente
codigo_pro_fac	Texto	30	Código de producto
fecha_fac	Fecha	10	Fecha de facturación
costo_uni_fac	Decimal	13E+2D	Costo unitario del producto
precio_uni_fac	Decimal	13E+2D	Precio de venta unitario
cantidad_fac	Decimal	13E+2D	Cantidad facturada

Ejemplo (Reseteo de información) PHP:

```
$data = array(
    "action" => "reset", /* <- Indicación de resetear */
    "reset_from_date" => "2023-09-20" /* <- Fecha desde la cuál se desea
eliminar las facturación previamente subida */
);

$data_json = json_encode($data);

$ch = curl_init();

curl_setopt($ch, CURLOPT_USERPWD , "USER:APIKEY");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_URL,
"https://www.pedifull.com/api/funciones/calcular_fecha_de_entrega_desde_alta/");
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HEADER_OUT, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_json);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json','Content-Length: ' . strlen($data_json)));

$result = curl_exec($ch);
$result = json_decode($result);

curl_close($ch);
```

## Cuenta corriente

**Descripción:** Permite subir facturas pendientes de cancelación en cuenta corriente.

**Objeto:** cuenta\_corriente

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_cli_cta	Texto	30	Código de cliente
comprobante_cta	Texto	50	Comprobante
fecha_cta	Fecha	10	Fecha de comprobante
vencimiento_cta	Fecha	10	Fecha de vencimiento
importe_cta	Decimal	13E+2D	Importe del comprobante
saldo_cta	Decimal	13E+2D	Saldo pendiente de cancelación

## Zonas

**Descripción:** Zonas geográficas de cobertura

**Objeto:** zonas

**Método:** PUT

Campo	Tipo	Longitud	Descripción
codigo_zon	Texto	15	Código
descrip_zon	Texto	255	Nombre