

# Manual de configuración Perfil de Clientes

---



## Manual de Consulta de Productos Bancarios

Este manual explica cómo utilizar una estructura JSON para generar dinámicamente tablas de productos bancarios a partir de servicios REST. El archivo organiza los datos en secciones, productos, endpoints y configuraciones de visualización.

---



### 1. Estructura General

El JSON contiene una propiedad principal:

- `"secciones"`: Lista de secciones con productos financieros agrupados (ej. Captaciones, Deuda Directa, etc.).

Cada sección incluye:

- `"tituloSeccion"`: Nombre de la categoría.
- `"productos"`: Lista de productos financieros que pertenecen a esa sección.

---




### 2. Estructura de un Producto


Cada producto tiene las siguientes propiedades clave:

###  ``tituloProducto``

Nombre visible del producto (ej. "Cheques", "Cajas de Seguridad").

###  ``iconoProducto`` `*(opcional)*`

Ícono para representar el producto visualmente (ej. ``account_balance``).

###  ``endpoint``

Configuración para consultar la lista principal del producto:

- ``url``: Ruta del servicio.
- ``metodo``: `"GET"` o `"POST"`.
- ``parametros``: Query params que pueden incluir variables globales (ej. ``$cliente``).
- ``body``: Solo para métodos POST, también puede incluir variables globales.

### 🔍 `endpointDetalle` \*(opcional)\*

Servicio para obtener detalles adicionales por cada ítem:

- - `url`: Incluye variables como `\$numeroCuenta`, `\$descripcionSubProducto`, etc.
- - `variables`: Lista de campos necesarios para formar el URL dinámico.

### 📌 `condiciones` \*(opcional)\*

Filtrado adicional para mostrar solo ciertos productos según:

- - `tipoProducto`
- - `relacion` (ej. solo productos donde el cliente es TITULAR)

---



### 3. Columnas de la Tabla

### 📊 `columnas`

Lista de columnas a mostrar en la tabla con las siguientes propiedades:

- - `titulo`: Encabezado visible.
- - `claveDato`: Clave identificadora.
- - `ruta`: Ruta del dato dentro de la respuesta (ej. ``\$data.numeroCuenta``).
- - `iconos` \*(opcional)\*: Mapeo de valores a íconos según estado (ej. ACTIVA → ✅).
- - `style` \*(opcional)\*: Estilos CSS por columna.
- - `formato` \*(opcional)\*: Para fechas (`"date"`) o montos (`"currency"`).
- - `interceptors` \*(opcional)\*: Diccionario para traducir valores internos a legibles.

---



### 4. Mapeo de Datos

### 🗺️ `mapeoDatos.rutaPrincipal`

Define la ruta dentro de la respuesta JSON donde está el arreglo de datos.

- - Si es `""`, se toma el objeto raíz.
- - Si es ``\$data.data``, significa que los datos están anidados.

---



### 5. Paginación

### 📄 `paginacionServerSide`


- - `true`: Se espera paginación por el servidor.
- - `false`: Toda la información se obtiene en una sola llamada.

Si es `true`, se deben usar variables como:

- - `\$numeroPagina`
- - `\$cantidadItems`

---


## 6. Seleccionabilidad

###  `seleccionable`

Define si el usuario puede seleccionar filas en la tabla.

---

## 7. Pie de Tabla (Footer)

###  `footer` \*(opcional)\*

Sección para mostrar totales en columnas:

- - `mostrar`: `true/false`
- - `titulo`: Texto que se muestra en el footer.
- - `columnasSumar`: Índices de columnas a sumar.
- - `styleGeneral` y `styleIndex`: Estilos visuales.

---

## 8. Variables Globales

Se encuentran en la clave `"variablesGlobales"` y se utilizan para llenar dinámicamente URLs y cuerpos:

```
{  
"$cliente": "",  
"$noCaja": "",  
"$numeroPagina": "1",  
"$cantidadItems": "10",  
"$excluir": "0"  
}
```

Estas variables deben ser llenadas antes de hacer cualquier consulta.

---

## Ejemplo de Flujo de Uso

1. **\*\*Mostrar Secciones y Productos:\*\***

- Agrupar por `"tituloSeccion"`.
- Listar productos por `"tituloProducto"`.

2. **\*\*Llenar Variables Globales:\*\***

3. **\*\*Ejecutar el ``endpoint`` del producto.\*\***

- Si ``metodo`` = GET, incluir ``parametros`` en el query string.
- Si ``metodo`` = POST, enviar ``body``.

4. **\*\*Parsear Respuesta:\*\***

- Seguir `"mapeoDatos.rutaPrincipal"` para obtener los datos.
- Usar `"columnas"` para construir la tabla.

5. **\*\*Si hay ``endpointDetalle``,\*\*** habilitar navegación o visualización de más datos.

---

## Ejemplo Real - Producto "Ahorros"

### Endpoint de consulta:

GET <https://plataforma-qa.bytesw.cloud/api/plataforma-services/v1/api/cuentas?documento=1000581&excluir=0>

### Respuesta:

[

{

"numeroCuenta": "21-302-017418-1",

"nombre": "VASQUEZ LOPEZ JULIETA",

"tipoProducto": 2,

"descripcionSubProducto": "CUENTA DE AHORROS MONEDA NACIONAL",

"moneda": "LPS",

"estado": "C",

"descripcionEstado": "CANCELADA",

```
"relacion": "TITULAR"
```


```
}
```



```
]
```

### Tabla renderizada:

Estado	Número de cuenta	Nombre del cliente	Relación	Sub producto	Moneda
--------	------------------	--------------------	----------	--------------	--------

-----	-----	-----	-----	-----	-----
-------	-------	-------	-------	-------	-------

	21-302-017418-1	VASQUEZ LOPEZ JULIETA	TITULAR	CUENTA DE AHORROS	MONEDA NACIONAL   LPS
---	-----------------	-----------------------	---------	-------------------	-----------------------

 El icono  proviene del valor `estado: C`, que se interpreta como `CANCELADA` y se asigna al ícono correspondiente en la columna.