

**Perfil de Fabricante –
Importación de Descripciones –
Conexión de Kalstein Plus con Woocommerce
(documentación)**

**Por: ing. José Velandia
Maracay, edo. Aragua 16-10-2023**

Panorama General del Perfil Fabricante

Registro de productos, con las siguientes características:

- Nombre
- Modelo
- Descripción
- Tabla técnica (opcional)
- Categoría
- Imagen
- Cantidad en stock
- Precio unitario
- Divisa
- Medidas del producto
- Medidas del packing (y tipo de paquete)
- Catálogo (PDF / opcional)
- Manual técnico (PDF / opcional)
- Descuentos por mayoreo
- Accesorios (múltiple):
 - Nombre
 - Modelo
 - Descripción (opcional)
 - Imagen (opcional)
 - Medidas de Producto
 - Medidas de Packing
 - Precio Unitario
 - Divisa

Descripción de directorios:

```
.
├── kalsteinPerfiles/
│   ├── php/
│   │   └── manufacturer
│   └── src/
│       ├── client
│       ├── ...
│       ├── ...
│       ├── manufacturer/
│       │   ├── css
│       │   └── js
│       ├── ...
│       ├── ...
│       └── templates-php/
│           └── manufacturer/
│               ├── dashboard.php
│               ├── stock.php
│               ├── productForm.php
│               ├── ...
│               └── ...
```

Siendo la raíz del proyecto kalsteinPerfiles, dentro de php/manufacturer se encuentran todos los archivos del Backend, incluyendo la subida de producto al servidor, la validación de cotizaciones, la recuperación de información para el Dashboard y el Reporte de Ventas, aprobación de cotizaciones, entre otros.

Dentro de src/manufacturer están los recursos de CSS y JavaScript, siendo la mayoría de las llamadas al backend mediante la función `ajax().done()` de jQuery.

Dentro de src/templates-php/manufacturer están las plantillas php del frontend, las tablas como la de "stock" y "órdenes" se encuentran aquí, así como la plantilla de formulario de registro de productos, con algunas excepciones de plantillas que se encuentran fuera:

Las siguientes plantillas heredan casi todo su código php del directorio `src/templates-php/client`:

- Inbox (en manufacturer) -> `inbox.php` (en client)
- Tienda -> `generate-quote.php`
- Catálogos -> `catalogs.php`
- Configuración de Cuenta -> `settings.php`
- Preview del producto (el ojo de la table de stock) -> `wooPreview.php`

El php del motor de búsqueda de la tienda se encuentra en el módulo `kalsteinCotizacion`, dependiente de `kalsteinPerfiles`, concretamente en: `kalsteinCotizacion/classes/templates-php/resultPages.php`.

El php de la calculadora de precios de envío en la pestaña de stock se encuentra en `kalsteinPerfiles/src/templates-php/distributor/`

Algunas partes cruciales del funcionamiento:

- **Validación de datos pre-envío al servidor:** justo al principio del archivo `product.request.js` y en el archivo `validateProductData.php` del backend. Este último también es el que recupera los datos mandados por Ajax y los almacena en sus respectivas variables como `$pName` o `$pModel` para mandarlos a la base de datos.
- **Subida y Actualización de producto:** `uploadProductData.php` y `updateProductData.php` en la carpeta de backend. La subida de accesorios es en `php/manufacturer/accessories/uploadProductData.php`.
- **Recuperación de información del reporte de ventas:** `getChartInfoSales.php` en el backed
- **Comportamiento dinámico de la tienda y barra de búsqueda:** `script.cot2.js` y `script.cot2.light.manu.js` dentro de `kalsteinCotizacion/assets/js`.

- **Comportamiento del inbox:** `inbox.pages.js` en `kalsteinPerfiles/src/js` (sin incluir el subdirectorio `manufacturer`, directamente en `src`).
- **La mayoría del comportamiento dinámico del perfil:** `product.request.js`, `fetch.data.js`, `accessorie.requests.js`.

Detalles en la subida de Accesorios:

Al adjuntar un accesorios en el editor se crea un `div.card`, este `div` tiene:

- Un `textarea` oculto que contiene un JSON con la información del accesorio.
- Un atributo `data` de estado que indica si es un producto añadido allí mismo (`draft`) o un producto recuperado de la base de datos (`uploaded`).
- Otro atributo `data` que le da un `id` único, que en caso de ser el `div` del tipo `uploaded`, representa el `id` del producto.
- Una referencia en el JSON a una variable global `currentAccessoryFiles` que contiene las imágenes, debido a que los objetos de tipo `File` no se pueden guardar en el JSON.

Los accesorios al subirse caen a la misma tabla de productos, pero sin los datos que no se piden en el formulario, con el campo `product_group` con el valor "1", y con el campo `product_parent` igual al `product_aid` del producto padre. La diferencia entre el "draft" y el "uploaded" es:

- El `draft` manda a añadir el nuevo accesorio con un `INSERT INTO`.
- El `uploaded` manda a actualizar el accesorio con un `UPDATE SET`.

Conexión con woocommerce

Uno de los atractivos importantes de la plataforma es que los productos registrados caen también a su propia publicación en el módulo de woocommerce del dominio principal (e.j. si hablamos de la copia del sitio alojada en platform.kalstein.us, el dominio principal es kalstein.us), para esto, el archivo `uploadProductData.php` cuenta al final con una serie de instrucciones para llevar el producto interno (de la tabla `wp_k_products`) al post de woocommerce (en la tabla `wp_post`) que se listan:

- Crear el post en `wp_post` de tipo “product” con el estatus en borrador (en espera de validación por el moderador).
- Recuperar la categoría del producto, transformarla en su slug (importante para la ubicación en el filtro de woocommerce), y asociar al post insertado esta categoría alojada en `wp_terms` mediante las tablas `wp_terms_relationships` y `wp_terms_taxonomy`.
- Colocar la metadata del precio en `wp_postmeta`.
- Agregar otro post en `wp_post` para la imagen, que es de tipo “attachment”.
- Asociar este thumbnail al post del producto en `wp_postmeta`.
- Por último, añadir el id del producto en `wp_k_products` y el id del producto en `wp_post` a la tabla de `wp_product_id_woo`, para tener una forma de relacionar los dos registros.

Para actualizar el producto:

- Se guarda el slug generado en `wp_post` para la publicación.
- Se elimina la misma al seleccionar su ID de `wp_product_id_woo`.
- Se elimina también la relación de `wp_product_id_woo` puesto que el post ya no existe.
- Se crea una publicación nueva con el producto actualizado, siguiendo el mismo procedimiento de la página anterior. **Importante:** Se usa el mismo slug que se recuperó para mantener la consistencia en los links.

**Importación de descripciones
desde woocommerce hasta la
plataforma kalsteinPlus**

Debido a que la publicación de productos KALSTEIN en woocommerce es algo anterior al desarrollo de la propia plataforma, surge la necesidad de recuperar la información de la descripción y tablas técnicas para que se visualicen en la aplicación de kalsteinPerfiles, para ello se ha desarrollado una herramienta que a continuación se explica cómo usar:

- Ir al dominio con el idioma que se desea importar las categorías, ir a *productos>todos los productos>exportar*, Seleccionar los campos de: ID, nombre, descripción y atributos, posterior a eso exportar la categoría del producto como un archivo .csv, (recomendado renombrarlo con el nombre de la categoría luego de descargarlo).
- Almacenarlo en algún subdirectorío de `src/templates-php/retrieve-products-src/`.
- Ir a `retrieve-rescription.php` dentro de `templaes-php/` y modificar la variable `let name` del primer evento 'click' del javascript incrustado para incluir el subdirectorío con los csv (si existe), e.j. si tu director es '`src/templates-php/retrieve-products-src/categorias-2020/`' la variable debe quedar:

```
let name = `categorias-2020/${$('#numero').val()}`;
```

- Asegurarse estar seleccionando los atributos de la variable `product` en el idioma correcto, e.j. Entre las líneas 165 y 190 están las sentencias `product.Name`, `product.Description` y `product['Attribute 1 value(s)']`, asegurarse de que estén en el idioma correcto, e.j. `product.Nombre`, `product.Descripcion`, `product['Valor(es) de atributo 1']`. **Importante reemplazar todos y cada uno**. Puedes crear una copia de este archivo y llamarla independientemente en WordPress si crees que necesitarás cambiar constantemente de idioma.
- Asegurarse de que los campos de la base de datos del archivo `save-description.php` (en el mismo directorío que el archivo anterior), para ello, fijarse en el sufijo (`_es`, `_fr`, etc) de `product_description` y

product_technical_description. NOTA: los campos de inglés son sin sufijo, ya que son los “originales” por así decirlo.

- Una vez verificados los dos puntos anteriores:
 - Abrir “dominio.kalstein.actual”/recover-data/
 - Colocar el nombre del archivo csv en el campo texto
 - Presionar en el botón
 - Revisar la consola JavaScript para tener más información del estado de la subida.

NOTA: se recomienda respaldar la tabla wp_k_products antes de realizar esto, debido a que, si se sube por error una descripción a los campos de un idioma que no es el correspondiente, poder tener el punto de guardado.

NOTA 2: En la consola saldrán frecuentemente mensajes de {data-not-found}, esto es totalmente normal y no es un error, será más común mientras más productos en serie contenga el csv, un ejemplo es la categoría de autoclaves.

NOTA 3: En caso extremo de un producto que no esté siendo seleccionado correctamente por el algoritmo, imprimir la consulta SQL del producto, colocarle el modelo requerido, y ejecutarla manualmente en el phpMyAdmin del servidor.