

Índice

1. Introducción

- 1.1. Objetivos del Proyecto**
- 1.2. Alcance del Proyecto**

2. Visión General del Proyecto

- 2.1. Descripción del Proyecto**
- 2.2. Componentes**
- 2.3. Organización y descripción de directorios**

3. Diseño Básico del Sistema

4. Manual del Usuario

- 4.1. Guía de conexión el servidor mediante protocolo SSH**
- 4.2. Guía de ejecución manual**
- 4.3. Configuración de ejecución automática**
- 4.4. Solución de Problemas**

5. Documentación Técnica

- 5.1. Tecnologías utilizadas**
- 5.2. Accesos**
- 5.3. Formato de archivos**
 - 5.3.1. Archivos CSV**
 - 5.3.2. Imágenes**
- 5.4. Entornos**
- 5.5. Diagramas**

6. Recomendaciones para el Futuro

6.1. Mantenimiento

- 6.1.1. Imagenes basura**
- 6.2.1. Copias de Seguridad**

6.2. Mejoras Potenciales

- 6.2.1. Marcas de Productos**
- 6.2.1. Backups automaticos**

1. Introducción

1.1. Objetivos del Proyecto

El objetivo primordial de este proyecto reside en la carga y actualización de las categorías, jerarquías y productos, incluyendo imágenes, disponibilidad en stock, precios y otros detalles relevantes, en la plataforma de comercio electrónico del cliente. Estos datos se obtendrán de archivos CSV proporcionados por el sistema de gestión del cliente, lo que permitirá una integración entre el sistema de gestión y la plataforma de comercio electrónico. Además de la actualización de datos, el proyecto se enfoca en la automatización de este proceso mediante tareas programadas previamente configuradas en el hosting contratado, con el propósito de sincronizar la replicación de los datos en los dos sistemas.

1.2. Alcance del Proyecto

- **Carga y Actualización de Categorías:** Implementación de funcionalidades que permitan la carga inicial y la actualización periódica de las categorías de productos en el sistema de comercio electrónico del cliente.
- **Carga y Actualización de Jerarquías:** Implementación de funcionalidades que permitan la carga inicial y la actualización periódica de las jerarquías de productos.
- **Carga y Actualización de Productos:** Se llevará a cabo la carga inicial y la actualización continua de productos en la plataforma del cliente, incluyendo información detallada como imágenes, disponibilidad en stock, precios y otros atributos relevantes.
- **Eliminación de Productos Restantes:** En el proceso de actualización de productos, se identificarán y eliminarán automáticamente los productos que no se encuentren en los archivos CSV proporcionados por el sistema de gestión del cliente.
- **Tareas Programadas:** Se implementarán tareas programadas que ejecuten de manera automática y periódica el proceso de carga y actualización, asegurando la sincronización adecuada de los datos con las fuentes de origen.

2. Visión General del Proyecto

2.1. Descripción del Proyecto

El proyecto se ha desarrollado empleando el lenguaje de programación PHP y se presenta como un directorio organizado que contiene todo el código fuente, estructurado en diversas carpetas. Su función principal consiste en la extracción de datos a partir de los archivos proporcionados por el cliente y de los datos existentes en la plataforma de comercio electrónico. Posteriormente, se lleva a cabo una minuciosa comparación entre los datos extraídos y los datos preexistentes.

En el proceso de comparación, se ejecutan las siguientes acciones:

- Si se detecta una nueva categoría en los datos proporcionados, esta se carga automáticamente.
- Si se identifica una jerarquía previamente no registrada, esta se carga automáticamente.
- En caso de hallar un producto nuevo en los datos extraídos, se realiza su incorporación en el catálogo de productos del comercio electrónico.
- Cuando se detecta que un producto ya registrado presenta cambios en sus datos, se efectúa la correspondiente actualización para reflejar la información más reciente.
- Al culminar el proceso de actualización y carga de datos, se procede a la eliminación de aquellos productos preexistentes en el ecommerce que no figuren en el archivo .csv proporcionado por el cliente.

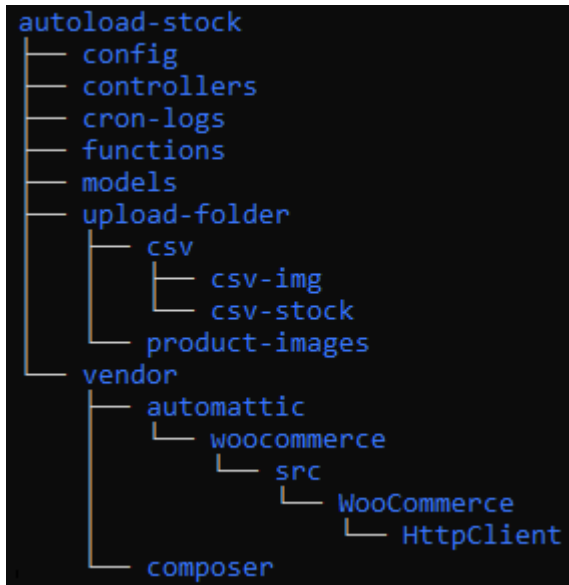
2.2. Componentes

- **Sistema de Gestión del Cliente.** Su rol principal radica en proporcionar los archivos de actualización necesarios para el sistema de comercio electrónico. Este componente se encarga de facilitar la actualización de datos al entregar los archivos de actualización en los horarios preestablecidos por parte del cliente.

Opera como el punto de origen de los archivos de actualización necesarios para mantener la información del comercio electrónico al día. El cliente, en su calidad de administrador del sistema, tiene la responsabilidad de generar y proporcionar estos archivos en formatos específicos y estructurados de acuerdo con las necesidades del proyecto.

- **Actualización/Carga de Productos (Núcleo del Proyecto).** Se trata del proyecto en sí mismo. Es el encargado de extraer los datos e información tanto del ecommerce como de los archivos proporcionados por el sistema del cliente, darles una organización, estructura y formato válido para interactuar con ellos a través del código fuente, y realizar las acciones de carga, actualización y borrado pertinentes.
- **Ecommerce.** Es la página web (Wordpress) donde se van a publicar las diferentes categorías, jerarquías y productos. Sobre ella se deben reflejar los cambios de carga, actualización y borrado, según los archivos que proporcione el sistema cliente.
- **Servidor.** Además de alojar el sitio web, es donde va a estar almacenado e instalado el proyecto. También se encarga de configurar los horarios de ejecución de los procesos para que se ejecuten de manera automática.

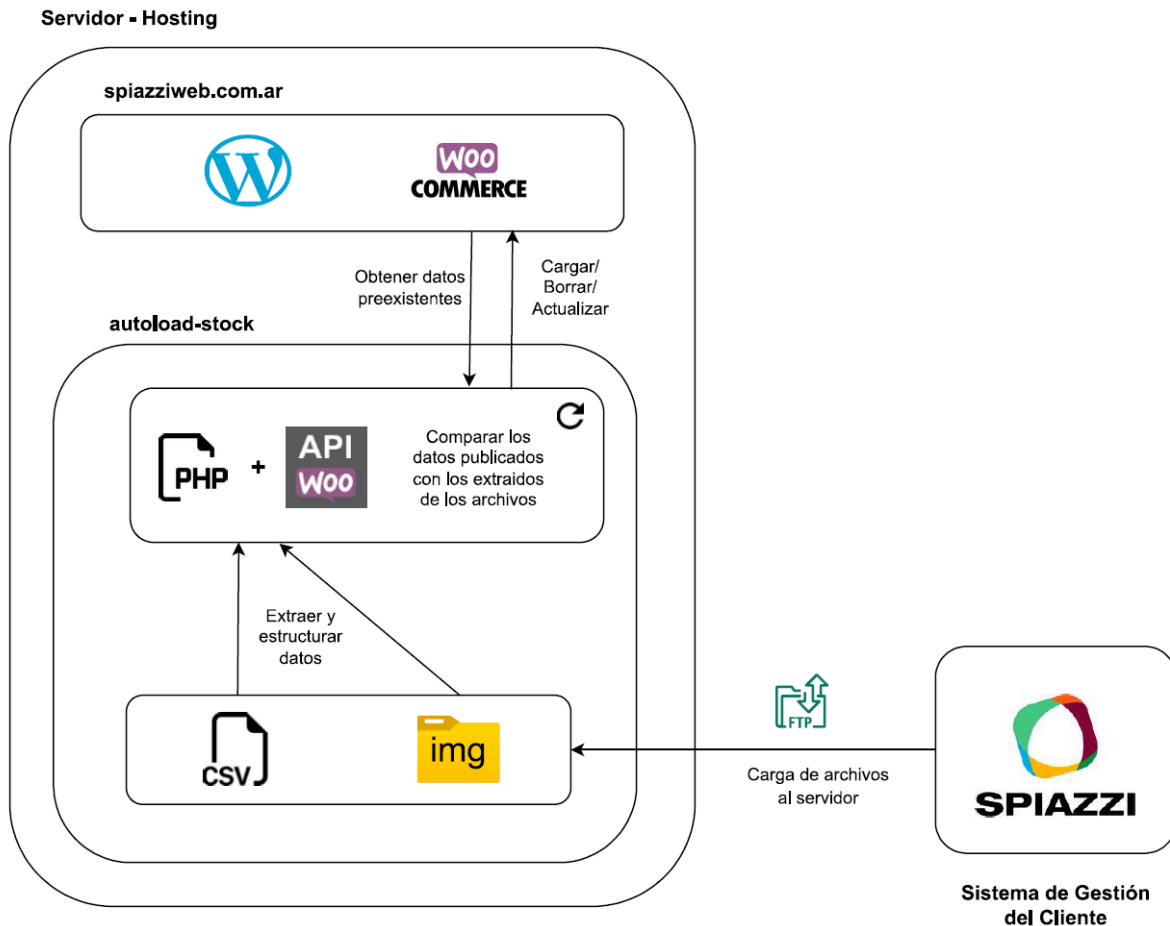
2.3. Organización y descripción de directorios



- **autoload-stock**: Directorio que contiene todos los archivos del proyecto.
- **config**: Archivos de configuración PHP del proyecto. Contiene la configuración de conexión a la API de WooCommerce, rutas, variables y “requires”.
- **controllers**: Contiene las clases que funcionan como controladores.
- **cron-logs**: Almacena los archivos “.log”; registros de ejecución.
- **models**: Contiene las clases que funcionan como modelos y estructuras de datos.
- **upload-folder**: Almacena los archivos que el cliente transfiere por ftp periódicamente al servidor.
 - **csv**: Almacena los archivos csv’s
 - **csv-img**: Almacena el archivo csv que se utiliza para extraer la información de las imágenes de productos.
 - **csv-stock**: Almacena el archivo csv que se utiliza para extraer la información de los productos a actualizar.
- **product-images**: Almacena las imágenes de los diferentes productos.
- **vendor**: Carpeta utilizada para el funcionamiento de las dependencias de Composer del proyecto.

- **functions:** Funciones del código fuente adicionales que no pertenecen a una clase específica.

3. Diseño Básico del Sistema



4. Manual del Usuario

4.1. Guía de conexión el servidor mediante protocolo SSH

A pesar de haber establecido previamente que el proceso de carga y actualización de productos se lleva a cabo mediante tareas programadas, puede surgir la necesidad de ejecutar el proceso de manera manual en cualquier momento.

Para iniciar el proceso de ejecución manual, se debe abrir la interfaz de línea de comandos en el sistema Windows, que comúnmente se conoce como CMD (Símbolo del Sistema) o consola de Windows.

Luego, se procederá a ejecutar el siguiente comando de referencia, sustituyendo los valores encerrados entre llaves “{ }” con los datos proporcionados por el proveedor de hosting:

```
ssh -p{puerto} {usuario}@{ip}
```

Ejemplo con los datos de acceso actuales:

```
C:\Users>ssh -p5933 root@200.58.98.196
```

Una vez ejecutado, se deberá ingresar la contraseña y presionar enter (al ingresar la contraseña no se mostrarán los caracteres escritos en pantalla).

```
C:\Users>ssh -p5933 root@200.58.98.196  
root@200.58.98.196's password:
```

Una vez ingresados los datos con éxito, ya nos encontraremos conectados remotamente con el servidor.

4.2. Guía de ejecución manual

Primero tendremos que estar conectados al servidor mediante SSH. Luego nos posicionarnos sobre el directorio del proyecto con el comando “cd”, ejemplo:

```
cd /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock
```

```
[root@vps-3086172-x ~] # cd /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock  
[root@vps-3086172-x /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock] #
```

Una vez posicionados en la carpeta del proyecto debemos ejecutar el comando que ejecutará el proceso de carga y actualización. Se puede ejecutar de dos maneras:

- Si se quiere ver en pantalla la ejecución del proceso en tiempo real (básicamente ver por la consola cómo se escribe el log de ejecución), se debe ejecutar el comando de la siguiente manera:

```
php run.php
```

```
[root@vps-3086172-x /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock] # php run.php
```

- Por otro lado, también se puede ejecutar el proceso y guardar todos sus registros en un archivo de tipo “log”. Esto no mostrará la ejecución en pantalla mientras el proceso se ejecute, sino que, guardará tanto los registros, como los errores en el archivo .log con el nombre especificado. Se debe hacer de la siguiente manera:

```
php run.php >> archivo.log 2>&1
```

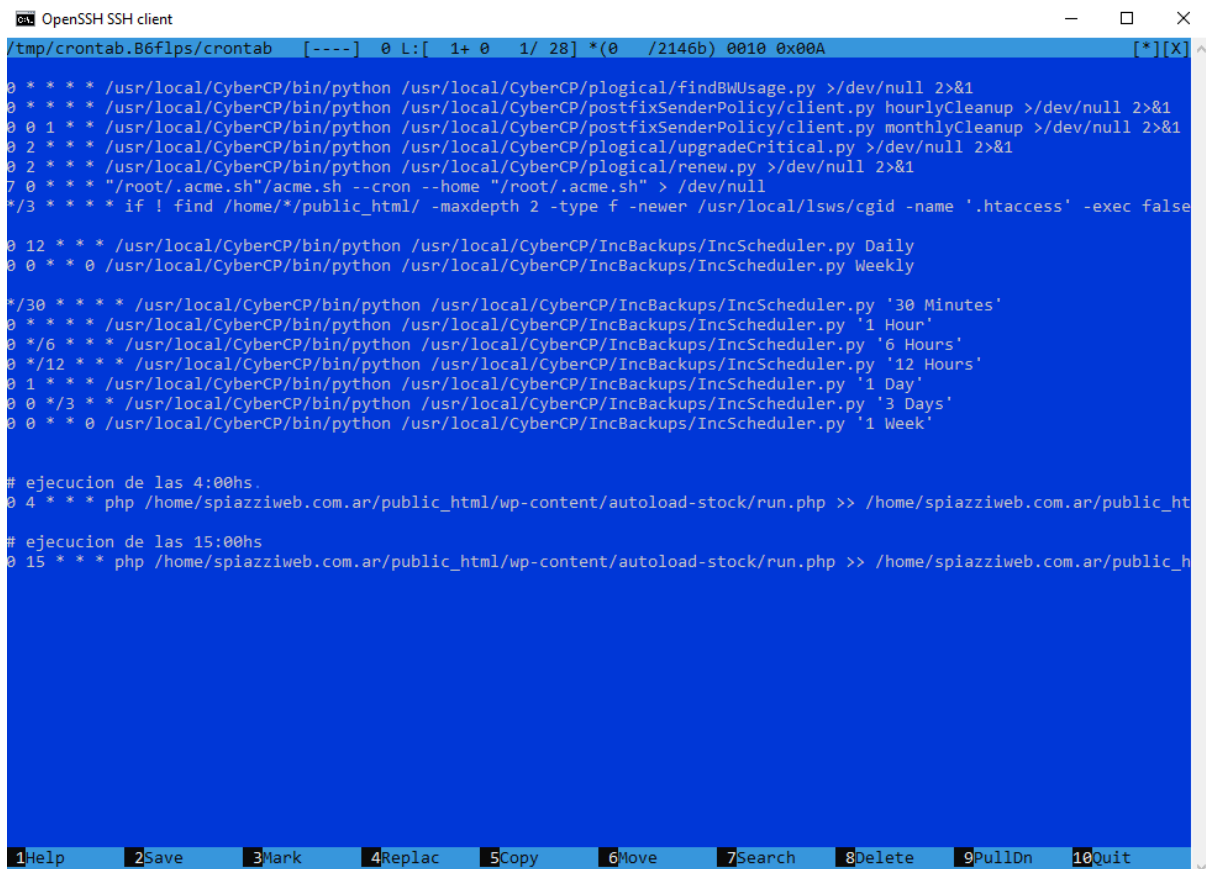
```
[root@vps-3086172-x /home/spiazziweb.com.ar/public_html/wp-content/autoload-stock] # php run.php >> archivo.log 2>&1
```

4.3. Configuración de ejecución automática

Conéctese al servidor mediante SSH. Una vez que esté conectado, abra el archivo de configuración de los “cronjobs” ejecutando el siguiente comando en la consola:

```
crontab -e
```

Esto abrirá el archivo utilizado para configurar los cronjobs en modo de edición.



```
OpenSSH SSH client
/tmp/crontab.B6flps/crontab  [----]  0 L: [ 1+ 0 1/ 28] *(0 /2146b) 0010 0x00A  [*][X]
0 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/plogical/findBWUsage.py >/dev/null 2>&1
0 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/postfixSenderPolicy/client.py hourlyCleanup >/dev/null 2>&1
0 0 1 * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/postfixSenderPolicy/client.py monthlyCleanup >/dev/null 2>&1
0 2 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/plogical/upgradeCritical.py >/dev/null 2>&1
0 2 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/plogical/renew.py >/dev/null 2>&1
7 0 * * * "/root/.acme.sh"/acme.sh --cron --home "/root/.acme.sh" > /dev/null
*/3 * * * if ! find /home/*/public_html/ -maxdepth 2 -type f -newer /usr/local/lsws/cgid -name '.htaccess' -exec false
0 12 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py Daily
0 0 * * 0 /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py Weekly

*/30 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '30 Minutes'
0 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '1 Hour'
0 */6 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '6 Hours'
0 */12 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '12 Hours'
0 1 * * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '1 Day'
0 0 */3 * * /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '3 Days'
0 0 * * 0 /usr/local/CyberCP/bin/python /usr/local/CyberCP/IncBackups/IncScheduler.py '1 Week'

# ejecucion de las 4:00hs.
0 4 * * * php /home/spiazziweb.com.ar/public_html/wp-content/autoload-stock/run.php >> /home/spiazziweb.com.ar/public_h

# ejecucion de las 15:00hs
0 15 * * * php /home/spiazziweb.com.ar/public_html/wp-content/autoload-stock/run.php >> /home/spiazziweb.com.ar/public_h

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

En el archivo de configuración, podrás escribir líneas de código para programar cuándo, en qué días deseas que se ejecute el proceso, el archivo que deseas ejecutar y el archivo de “.log” que se va a generar cuando se ejecute el proceso. Cada línea de código en este archivo tiene un formato específico, ejemplo:

```
*****{php comando_a_ejecutar.php} >> {archivo.txt} 2>&1
```

Vamos a explicar esta línea de código por partes, primero los asteriscos:

- El primer asterisco representa el minuto en que deseas que se ejecute el comando (0-59). Si se utiliza un asterisco en este campo, el comando se ejecutará en cada minuto de la hora especificada.
- El segundo asterisco representa la hora en que deseas que se ejecute el comando (0-23). Cuando se emplea un asterisco en este campo, el comando se ejecutará en cada hora del día.
- El tercer asterisco representa el día del mes en que deseas que se ejecute el comando (1-31). Si decides utilizar un asterisco aquí, el comando se ejecutará en el día especificado de cada mes.
- El cuarto asterisco representa el mes en que deseas que se ejecute el comando (1-12). Al utilizar un asterisco en este campo, el comando se ejecutará en el mes especificado.
- El quinto asterisco representa el día de la semana en que deseas que se ejecute el comando (0-6, donde 0 es domingo y 6 es sábado). Si se coloca un asterisco en este campo, el comando se ejecutará en el día de la semana especificado.

Por ejemplo, si deseas que el proceso de actualización se ejecute todos los días a las 3:30 PM, la línea se vería así:

```
0 15 *** {php comando_a_ejecutar.php} >> {archivo.txt} 2>&1
```

Una vez definida la hora y los días de ejecución, hay que agregar a la línea de código del archivo que queremos que se ejecute. Para ejecutar el archivo que corre el proceso de actualización primero se debe anteponer la palabra “php” y luego la ruta absoluta del archivo que queremos ejecutar, en este caso para el entorno de producción es “/home/spiazzweb.com.ar/public_html/wp-content/autoload-stock/run.php”.

```
0 15 *** php /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock/run.php >> {archivo.txt} 2>&1
```

Para finalizar, debemos configurar donde queremos que se escriban y guarden los registros de ejecución. Después de definir el archivo a ejecutar, debemos escribir “>>” y luego la ruta absoluta donde queremos que se guarde el archivo, la cual debe ser la carpeta “cron-logs” del proyecto en cuestión, la cual es:

```
/home/spiazzweb.com.ar/public_html/wp-content/autoload-stock/cron-logs
```

Seguido del nombre del archivo, el cual se va a definir con un formato de fecha y hora específico:


```
autoload_stock_$(date +%Y-%m-%d_%H-%M-%S).log
```

Para finalizar, escribiremos “2>&1” al final de la línea de configuración, quedando como resultado la siguiente línea de código:

```
0 15 * * * /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock/run.php  
>> /home/spiazzweb.com.ar/public_html/wp-content/autoload-stock/cron-logs/autoload_stock_$(date +%Y-%m-%d_%H-%M-%S).log 2>&1
```

El nuevo cronjob quedará programado y comenzará a ejecutarse según la programación que hayas establecido, en este caso el archivo “run.php” se ejecutará todos los días a las 15:00hs y se registrará en el archivo “autoload_stock_(fecha)_(hora).log”.

4.4. Solución de Problemas

Dentro de la carpeta “cron-logs” se encuentran los archivos “.log” que se generan como resultado de las tareas programadas. Al examinar estos registros, es posible deducir si el proceso se ejecutó sin inconvenientes o si se presentaron errores. En caso de detectar indicios de problemas en los datos registrados, es fundamental realizar un seguimiento detallado de los mismos.

Cuando el proceso se ha desarrollado con éxito, se podrá visualizar el siguiente mensaje al final del log de ejecución.

```
-> BORRANDO PRODUCTOS RESTANTES  
-> No hay productos para eliminar  
FIN DE EJECUCIÓN  
[root@vps-3086172-x /home/test.spiazzweb.]
```

No obstante, si el registro no incluye este mensaje, puede indicar que se ha producido un problema durante la ejecución del proceso.

Se recomienda encarecidamente llevar a cabo la siguiente verificación con el fin de abordar los problemas que puedan surgir:

- Verificar si la ejecución se interrumpió durante la actualización de una categoría, jerarquía o producto. En este caso, se aconseja examinar los datos relativos al elemento en cuestión en los archivos CSV que contienen su información. Esto permitirá identificar posibles problemas relacionados con los datos, como errores en nombres, caracteres especiales, formatos u otros aspectos que puedan estar afectando la correcta ejecución del proceso.
- Analizar el error que el Lenguaje PHP imprime en pantalla. En la mayoría de los casos de error, PHP mostrará un mensaje de error en pantalla que detalla

específicamente cuál fue el error a nivel técnico y la línea de código donde se produjo el mismo.

5. Documentación Técnica

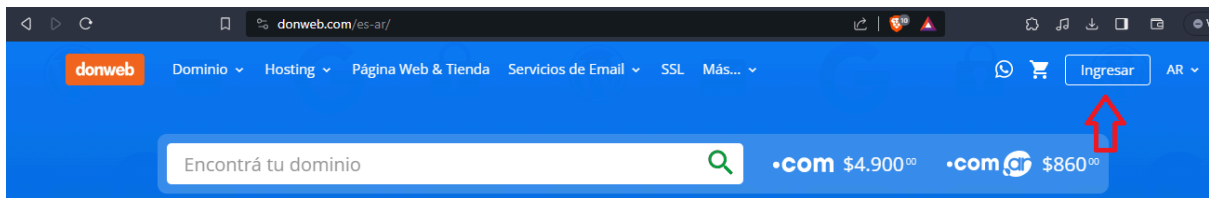
5.1. Tecnologías

- **Ubuntu v20.04** -> Sistema operativo que corre el servidor (Hosting DonWeb).
- **PHP v7.4.3** -> Lenguaje de Programación en el que está desarrollado el proyecto.
- **WooCommerce REST API v3.1.0** -> API utilizada para interactuar con la plataforma de ecommerce.
- **Composer v2.5.7** -> Dependencias para instalar la API REST de WooCommerce.

5.2. Accesos

Los datos de acceso se pueden consultar y cambiar desde la página web del servicio de hosting contratado (DonWeb).

Deberá loguearse en el sitio con los datos correspondiente y acceder al panel administrador.





Acceder

¿No tienes cuenta aún? [Créala desde aquí.](#)

Continuar con Facebook

Email o ID de Cliente

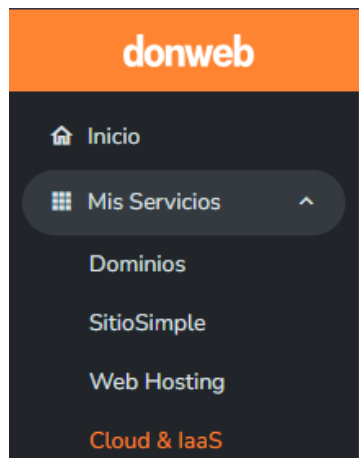
Contraseña

Acceder

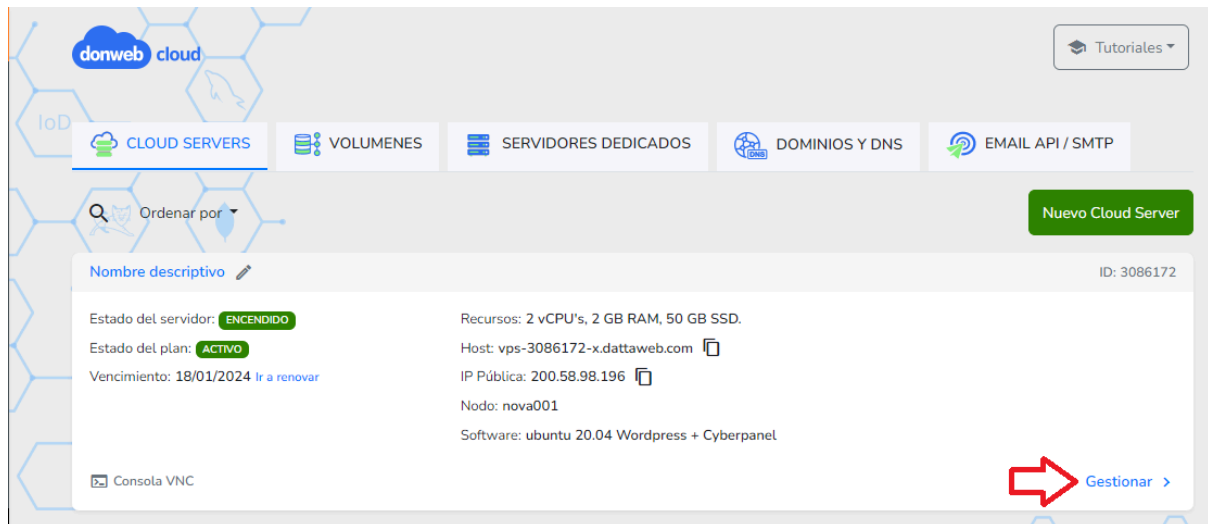
[Olvidé mi contraseña.](#)

Al utilizar una cuenta en DonWeb.com, estás aceptando los [Terminos y Condiciones](#) y la [Política de Privacidad](#).

Una vez logueado, accederemos mediante las opciones del menú lateral izquierdo a la sección “Mis Servicios” > “Cloud & IaaS”.



Cuando estemos dentro de esta sección se mostrarán los datos del servidor y servicios contratados, luego se debe acceder al apartado de “Gestionar”.



Una vez dentro accederemos a la opción del menú que dice “Software y Accesos”.



Finalmente, podrá consultar los datos de acceso en la parte inferior del panel, tanto al panel administrador del servidor, como por protocolo SSH.

Información

Software y Accesos

Estadísticas

Firewall

Pago y renovación

Cambio de servicio

Solicitud de baja

SOFTWARE

SNAPSHOTS

LICENCIAS

Sistema operativo

Plataforma: Linux

Sistema operativo: Ubuntu 20.04

Versión de SO: 20.04

Arquitectura: 64 bits

Aplicación o Stack

Aplicación: Wordpress + Cyberpanel

Imagen: Ubuntu2004-64-cyberpanel-wp

Conocer primeros pasos

Recrear

Vaciar

WORDPRESS

CYBERPANEL

SSH

Datos de conexión

URL de acceso: [https://192.168.1.100:8080](#)

Usuario: admin

Contraseña:

Encuentra tutoriales de Cyberpanel [aquí](#).

Ingresar al panel

5.3. Formato de archivos

5.3.1 Archivos CSV

Como se mencionó anteriormente, el proyecto necesita de dos archivos csv para realizar la actualización de los datos de los productos.

Uno tiene como nombre "upload-stock.csv". El mismo contiene las categorías y los datos de los productos que se desean cargar/actualizar. Su ubicación es

/home/spiazzziweb.com.ar/public_html/wp-content/autoload-stock/upload-folder/csv/csv-stock

LINEA	FAMILIA	TIPO_PRODUCTO	SKU	NOMBRE	DESCRIPCION_CORTA	PRECIO	STOCK	MARCA
TECNOLOGIA	FOTOGRAFIA	ACCESORIOS DE FO	40004	ESTUCHE	Estuche para cámara com	\$846,74	4	CASE LOGIC
MUEBLES	MUEBLES PAR	CUNAS	40221	CUNA FIJA	Fabricado en melamina / F	\$54.787,03	2	LA ESPERANZA
MUEBLES	MUEBLES DOR	MESA DE LUZ	40233	MESA DE I	Fabricada en melamina / 3	\$39.675,28	3	LA ESPERANZA
MUEBLES	MUEBLES DOR	CHIFONIER/CAJONEF	40255	CHIFONIEI	Fabricado en melamina cc	\$63.404,40	5	PLATINUM
MUEBLES	MUEBLES DOR	ROPERO/PLACARD	40259	PLACARD	Fabricado en melamina cc	\$54.121,37	3	PLATINUM
MUEBLES	OTROS MUEBL	TABLA PLANCHAR/GA	40280	GABINETE	Fabricado en melamina cc	\$34.660,27	8	PLATINUM
MUEBLES	OTROS MUEBL	TABLA PLANCHAR/GA	40281	GABINETE	Fabricado en melamina cc	\$37.278,03	2	PLATINUM
FERRETERIA	HERRAMIENTA	CORTADORAS DE CE	40634	CORTADO	Eléctrica / 1700W de pote	\$178.662,73	8	PETRI
MUEBLES	MUEBLES COM	SILLAS COMEDOR	40658	SILLA S35	Silla de caño 1 1/4 pulgad	\$24.985,88	18	DIELFE
COLCHONERIA	COLCHONES	COLCHON 2 PLAZAS	40887	COLCHON	Exclusiva espuma de poli	\$213.127,24	9	PIERO
MUEBLES	MUEBLES OFIC	PUESTOS DE TRABA	40954	PUESTO D	Fabricado en melamina 18	\$116.253,15	2	PIRO
MUEBLES	MUEBLES OFIC	PUESTOS DE TRABA	40955	PUESTO D	Fabricado en melamina 18	\$116.253,15	3	PIRO
MUEBLES	MUEBLES DOR	CHIFONIER/CAJONEF	41048	CHIFONIEI	Fabricado en melamina cc	\$52.889,60	3	PLATINUM
MUEBLES	MUEBLES DOR	ROPERO/PLACARD	41054	PLACARD	Fabricado en melamina cc	\$128.712,12	2	PLATINUM
MUEBLES	MUEBLES DOR	ROPERO/PLACARD	41065	PLACARD	Fabricado en melamina cc	\$54.121,37	4	PLATINUM
CLIMATIZACION	CALEFACCION	CALEFACTOR A GAS	41081	CALEFACT	Tiro balanceado / Multigas	\$236.609,87	12	LONGVIE
CLIMATIZACION	CALEFACCION	CALEFACTOR A GAS	41099	CALEFACT	Tiro balanceado / Multigas	\$198.858,97	2	LONGVIE
TIEMPO LIBRE	JARDIN	SILLAS JARDIN	41160	SILLON FIE	Sillón apilable con brazos	\$10.367,69	90	VOSS
CLIMATIZACION	VENTILACION	VENTILADOR DE TEC	41168	VENT TEC	75W de potencia / 4 palas	\$42.344,28	3	CODINI
CLIMATIZACION	VENTILACION	VENTILADOR DE TEC	41169	VENT TEC	Motor tradicional / Carcaza	\$55.439,81	6	CODINI

La lógica del código fuente identifica las diferentes categorías y los datos de los productos mediante el número de columnas. Por lo tanto es de vital importancia que se mantenga el siguiente orden para el correcto funcionamiento del proyecto.

- Categoría 1 (categoría padre pura) = Columna n°.1
- Categoría 2 (hija de la Categoría 1) = Columna n°.2
- Categoría 3 (hija de la Categoría 2) = Columna n°.3
- SKU = Columna n°.4
- Nombre del Producto = Columna n°.5
- Descripción corta = Columna n°.6
- Precio = Columna n°.7
- Stock = Columna n°.8
- Marca = Columna n°.9 (en desuso)

Se recomienda tener cuidado con el uso de caracteres especiales que puedan llegar a contener los datos (como “/”), ya que puede ocasionar problemas en el funcionamiento.

También se hará uso de un archivo CSV con el nombre de “upload-img.csv”, el cual contendrá la información necesaria para que el proceso de actualización identifique y cargue/actualice las imágenes pertenecientes a un producto. Su ubicación es

/home/spiazzziweb.com.ar/public_html/wp-content/autoload-stock/upload-folder/csv/csv-img

SKU	NOMBRE	DESCRIPCION_CORTA
40004	290176_40004- ESTUCHE P.CAM. DIG. CASE LOGIC DCB-302.png	1
40004	290176_40004- ESTUCHE P.CAM. DIG. CASE LOGIC DCB-302.png	2
40221	290796_40221- CUNA FIJA JUNIOR 3010 BCA.ROSA L.E.png	1
40233	290982_40233- MESA DE LUZ BABY 3134 3CAJ. L.E.png	1
40255	162525_40255-CHIFONIER CEDRO PLATINUM.png	1
40259	227402_40259-PLACARD 912 2P 2C TABACO PLATINUM.png	1
40259	227402_MEDIDAS 912 PLATINUM.png	2
40280	208446_40280-GABINETE DE PLANCHADO 3080 BCO PLATINUM.png	1
40281	208446_40281-GABINETE DE PLANCHADO 3081 BCO PLATINUM.png	1
40281	208446_40281-GABINETE DE PLANCHADO 3081 BCO PLATINUMb.png	2
40634	172212_40634-CORTADORA CESPED ELECTRICA MAX 5003 1HP CREC	1
40658	290796_40658- SILLA S3500 DIELFE.png	1
40887	206446_40887-COLCHON PARAISO REAL II 1.40X1.90 PIERO.png	1
40954	289047_108216_40954-8.jpg-01.png	1
40955	289047_108216_40955-8.jpg-01.png	1
41048	223439_41048-CHIFONIER 920 CEDRO PLATINUM.png	1
41054	211821_41054-PLACARD 916 6P 4C CEDRO PLATINUM.png	1
41054	211821_MEDIDAS 916 PLATINUM.png	2
41065	289759_41065- PLACARD 912 BLANCO PLATINUM-01.png	1
41065	289759_MEDIDAS 912 PLATINUM.png	2

Al igual que la lógica del anterior archivo, este también identifica el tipo de dato por el número de columna:

- SKU = Columna n°.1
- Nombre del archivo de la imagen = Columna n°.2

La columna numero n° 3 no tiene uso. Puede omitirse y hasta eliminarse del formato del archivo sin problemas.

Recordar de nuevo el tener cuidado al utilizar caracteres especiales en los nombres de las imágenes.

5.3.2. Imágenes

La ubicación

`/home/spiazzweb.com.ar/public_html/wp-content/autoload-stock/upload-folder/product-images` es la que contiene los archivos de las imágenes de los productos.

Para garantizar el correcto funcionamiento del proceso, al momento de cargar o actualizar una imagen, el archivo de la misma se debe encontrar almacenada en la ubicación antes mencionada y al mismo tiempo el nombre del archivo de la misma debe estar registrado en el archivo "upload-img.csv" con el SKU del producto correspondiente.

De nuevo se recuerda tener especial cuidado con el nombre de los archivos de las imágenes, ya que podría causar errores el uso de caracteres especiales o malos formatos de nombre.

5.4. Entornos

- Producción
 - Ubicación:
 - URL: <https://spiazzweb.com.ar/>
 - Nombre Base de Datos:
- Testing
 - Ubicación:
 - URL:
 - Nombre Base de Datos:

5.4. Diagramas

Diagrama de Clases

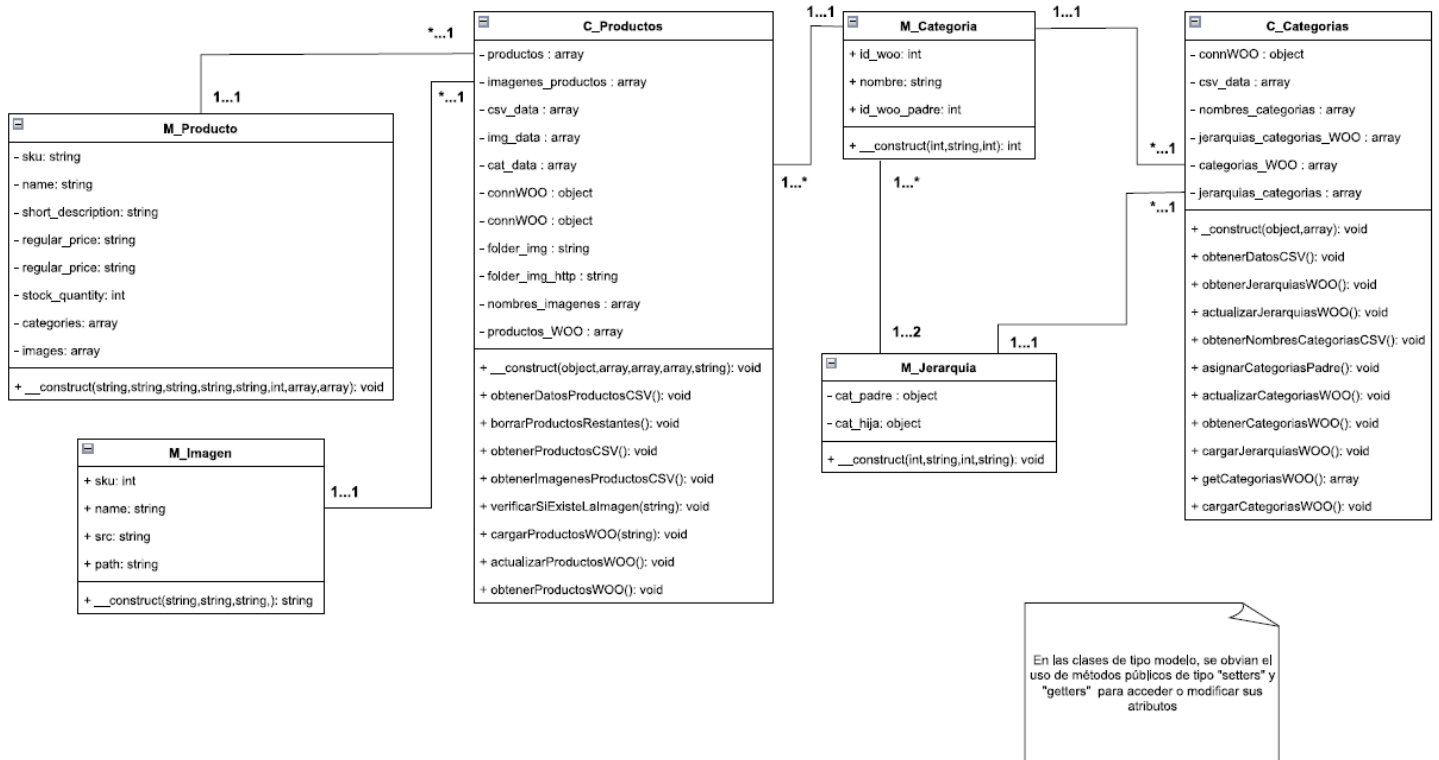
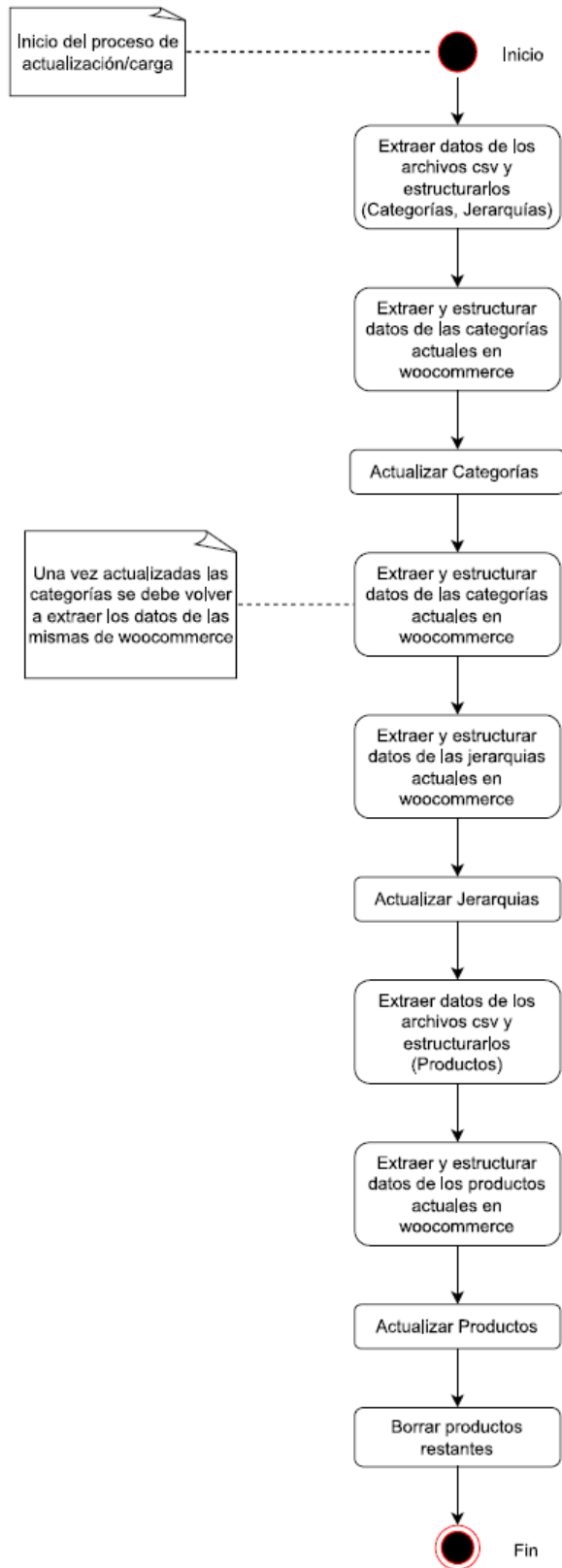


Diagrama de Actividad (Ejecución del proceso)



6.1. Mantenimiento

6.1.1. Imágenes basura

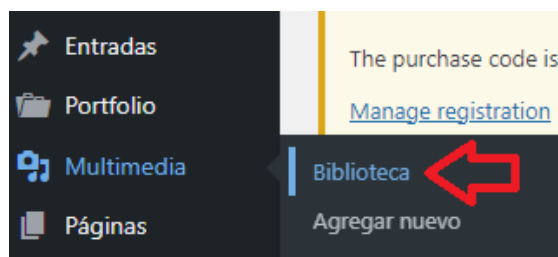
Cuando el proceso de actualización borra un producto o las imágenes de un producto, estas imágenes solo se eliminan de la información del producto en cuestión en el apartado del plugin de WooCommerce, pero estas aún quedan almacenadas en la sección de medios de WordPress, ocupando almacenamiento en el servidor.

Si bien este problema no es algo que afecte en el rendimiento del ecommerce o del proceso de actualización, a la larga puede comprometer el almacenamiento disponible en el servidor por la acumulación de imágenes basura.

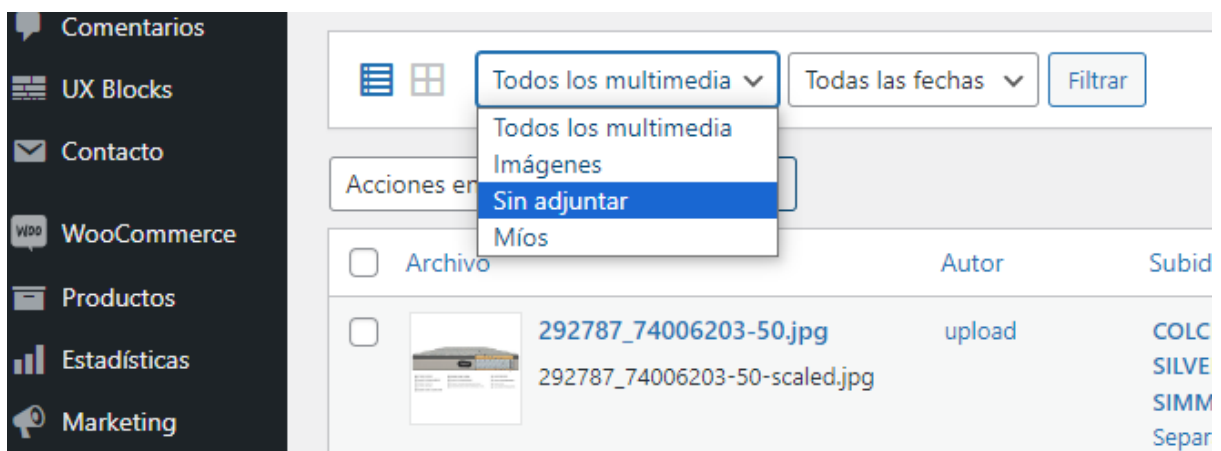
Por lo tanto, se recomienda que regularmente se borren estas imágenes de la galería de medios de WordPress. Puede hacer esto de dos formas.

- Borrar las imágenes manualmente.

Para hacer esto deberá ingresar a la galería de medios de wordpress.

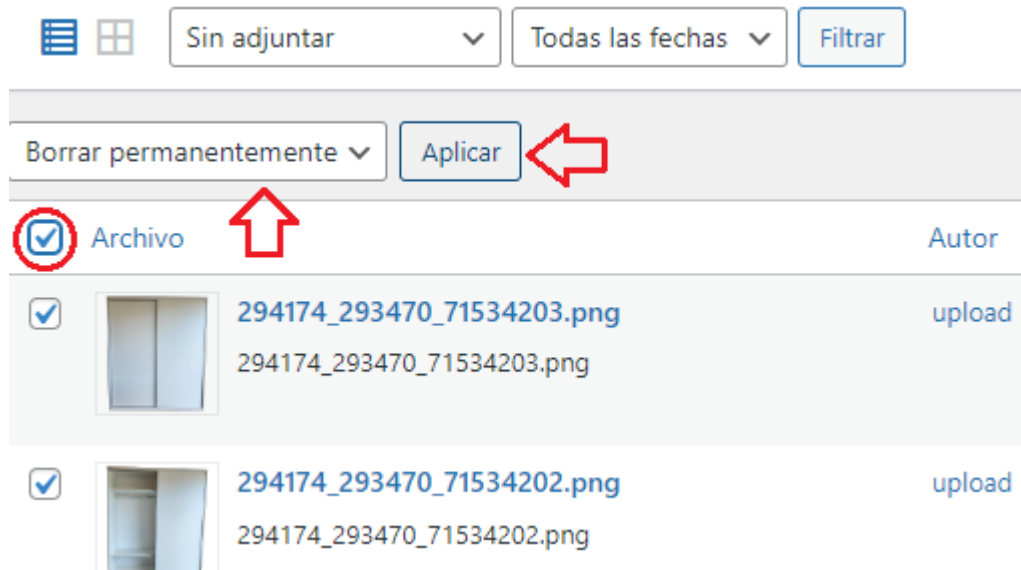


Luego filtramos todos los archivos multimedia como "Sin adjuntar".



Una vez hecho esto se listaran en pantalla todas las imágenes que no tienen asignado un producto.

Para borrarlas, se deberá tildar el checkbox que selecciona todos los archivos de la lista, seleccionar la acción “Borrar Permanentemente” y hacer click en el botón “Aplicar”.



Esto borrará todas las imágenes que no tienen un producto adjuntado en WOO.

- Utilizar algún plugin de WordPress. Se encuentran disponibles varias extensiones o plugins que ofrecen soluciones a esta problemática. La elección de la opción más apropiada, depende de la preferencia del cliente o del desarrollador, de acuerdo a sus necesidades particulares.

6.1.2. Copias de Seguridad

Es de extrema importancia tener en cuenta que el proyecto se encuentra instalado y configurado dentro del directorio del sitio WordPress en sí. Por lo tanto, si se borra o pierde el directorio del sitio web (/home/spiazzweb.com.ar), también se pierde todo el proyecto (/home/spiazzweb.com.ar/public_html/wp-content/autoload-stock).

Se recomienda encarecidamente generar copias de seguridad del directorio del proyecto (autoload-stock). Se pueden hacer las mismas descargando el directorio que contiene todos los archivos mediante:

- Protocolo FTP/SFTP
- Desde el panel administrador del servidor (CyberPanel) en el gestor de archivos

También se puede hacer una copia del directorio en otro lugar del servidor que no sea el ecommerce (tener en cuenta que si se pierden los datos del servidor se pierde también el proyecto). Cabe aclarar que se va a dejar una copia de seguridad en el servidor en la ubicación “/home/autoload-stock-bkk” con la última versión del entorno de producción.

Por último, se recomienda tener una copia de seguridad en algún medio físico (discos rígidos, pen drive, memorias externas, etc).

6.2 Mejoras Potenciales

6.2.1. Marcas de productos

Se pretende que el proyecto también sea capaz de actualizar o cargar la marca específica de un producto. El inconveniente es que la API WooCommerce no tiene funcionalidades para manipular este tipo de dato. Esto se debe a que la incorporación de la marca de los productos no es propia de las funcionalidades estándar de WooCommerce, sino que se incorpora a través de una extensión llamada “WooCommerce Brands”.

A continuación, detallaremos algunas recomendaciones para abordar este requerimiento:

- Utilizar la documentación oficial. La misma indica cómo conectarse y cómo se utilizan las diferentes funcionalidades de la API REST de WooCommerce Brands. La misma puede encontrarse en <https://woo.com/document/woocommerce-brands/#:~:text=The%20Brands%20REST%20API%20allows.of%20product%20categories%20REST%20API>
- Programar el código fuente con el mismo enfoque que se utiliza en el proyecto. Cada elemento (productos, categorías, jerarquías, imágenes, etc) posee una clase de tipo “modelo” o una clase de tipo “controlador” o ambas. La clase modelo debería estructurar los datos necesarios para interactuar con la API REST, y la clase controlador debe encargarse de las conexiones, las validaciones y la interacción con la API entre otras cosas. En este caso se recomienda crear ambas, una clase modelo “M_Marca” y una clase controladora “C_Marcas”. Podría solamente crearse una sola clase modelo para las marcas y extender la lógica de la clase controladora “C_Productos” pero esto terminaría complejizando la misma.
- Lo esencial es conseguir listar todas las marcas de los productos publicados en el sitio web y cargar/actualizar la marca a un producto específico a través de la API REST de WooCommerce Brands. Cuando consultamos todas las marcas de los productos publicados, tenemos disponible la información para comparar y verificar si hay que cargar o actualizar una marca en base a los archivos. En el caso que esto sea afirmativo, se cargará o actualizará la marca del producto en cuestión. Todo lo mencionado debe hacerse con las funcionalidades que prevea la API.
- Una vez conseguido el requerimiento actualizar la documentación.

6.2.2. Backups automaticos

Se recomienda configurar el servidor para realizar copias de seguridad automáticas periódicas del proyecto. Esto es esencial para garantizar la seguridad de los datos y la continuidad del proyecto en caso de incidentes. La frecuencia, el alcance, la ubicación y la



retención de estas copias de seguridad deben ser definidos cuidadosamente para cumplir con los requisitos y disponibilidad del proyecto