

```
woocommerce-aliexpress-dropshipping/
|
|— assets/
|   |— css/
|       |— admin.css
|       |— js/
|           |— order-manager.js
|       |— images/
|           |— logo.png
|
|— includes/
|   |— class-wc-alidropship-order-manager.php
|   |— class-wc-alidropship-product-fields.php
|   |— admin/
|       |— class-wc-alidropship-admin.php
|
|
|— woocommerce-aliexpress-dropshipping.php
```

admin.css

```
/* Estilos para el panel de administración del plugin */
```

```
/* Panel de información del producto */
```

```
.aliexpress-customer-data,
.aliexpress-order-products {
    margin-bottom: 15px;
}
```

```
.aliexpress-customer-data h4,
.aliexpress-order-products h4 {
    margin-top: 0;
    padding-bottom: 5px;
    border-bottom: 1px solid #eee;
    font-size: 14px;
    font-weight: 600;
}
```

```
.aliexpress-customer-data table {
    margin-bottom: 10px;
    width: 100%;
    border-collapse: collapse;
}
```

```
.aliexpress-customer-data table td {
```

```
padding: 5px;
vertical-align: top;
border-bottom: 1px solid #f5f5f5;
}
```

```
.aliexpress-customer-data table td:first-child {
width: 30%;
font-weight: 500;
}
```

```
/* Lista de productos */
.order-products-list {
margin: 0;
padding: 0;
list-style: none;
}
```

```
.order-products-list li {
margin-bottom: 10px;
padding: 10px;
background: #f9f9f9;
border: 1px solid #e5e5e5;
border-radius: 3px;
}
```

```
.product-name {
display: block;
font-weight: bold;
margin-bottom: 5px;
font-size: 13px;
}
```

```
.product-options {
margin-bottom: 10px;
}
```

```
.product-options span {
display: inline-block;
margin-right: 5px;
margin-bottom: 5px;
font-size: 12px;
background: #e9e9e9;
padding: 2px 5px;
border-radius: 3px;
}
```

```
}
```

```
.aliexpress-actions {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  margin-top: 8px;  
}
```

```
.aliexpress-id {  
  font-size: 12px;  
  color: #777;  
}
```

```
.aliexpress-missing {  
  margin-top: 5px;  
}
```

```
/* Botón copiar datos */  
#copy-customer-data {  
  display: inline-block;  
  margin-top: 5px;  
  background-color: #2271b1;  
  border-color: #2271b1;  
  color: #fff;  
  text-shadow: none;  
}
```

```
#copy-customer-data:hover {  
  background-color: #135e96;  
  border-color: #135e96;  
}
```

```
#copy-status {  
  display: inline-block;  
  vertical-align: middle;  
  font-size: 13px;  
  margin-left: 10px;  
  font-weight: 500;  
}
```

```
/* Campos en pantalla de edición de producto */  
.options_group h4 {  
  margin-top: 10px;
```

```
margin-bottom: 10px;
color: #23282d;
font-size: 14px;
font-weight: 600;
}
```

```
.extract-aliexpress-data {
margin-right: 10px !important;
}
```

```
.aliexpress-extraction-status {
display: inline-block;
vertical-align: middle;
font-size: 13px;
}
```

```
/* Análisis de beneficio */
.analysis-profit {
margin: 10px;
padding: 15px;
background: #f8f8f8;
border-left: 3px solid #2271b1;
}
```

```
.analysis-profit h4 {
margin-top: 0 !important;
margin-bottom: 10px !important;
font-size: 14px !important;
}
```

```
.analysis-profit p {
margin: 5px 0;
font-size: 13px;
}
```

```
.analysis-profit .positive {
color: #388e3c;
font-weight: 500;
}
```

```
.analysis-profit .negative {
color: #d32f2f;
font-weight: 500;
}
```

```
/* Compatibilidad con HPOS */
.woocommerce-page.edit-order .aliexpress-customer-data,
.woocommerce-page.edit-order .aliexpress-order-products {
    margin-top: 15px;
}
```

```
/* Página de configuración */
.wc-alidropship-howto {
    margin-top: 15px;
    margin-bottom: 15px;
}
```

```
.wc-alidropship-howto ol {
    margin-left: 20px;
}
```

```
.wc-alidropship-howto li {
    margin-bottom: 8px;
}
```

```
/* Ajustes para modo oscuro (dark mode) */
.admin-color-dark .aliexpress-customer-data h4,
.admin-color-dark .aliexpress-order-products h4 {
    border-bottom-color: #555;
}
```

```
.admin-color-dark .order-products-list li {
    background: #32373c;
    border-color: #555;
}
```

```
.admin-color-dark .product-options span {
    background: #444;
    color: #eee;
}
```

```
.admin-color-dark .analysis-profit {
    background: #32373c;
}
```

```
/* Ajustes para dispositivos móviles */
@media screen and (max-width: 782px) {
    .aliexpress-customer-data table td {
```

```
display: block;
width: 100%;
}
```

```
.aliexpress-customer-data table td:first-child {
font-weight: bold;
padding-bottom: 0;
border-bottom: none;
}
```

```
.aliexpress-actions {
flex-direction: column;
align-items: flex-start;
}
```

```
.aliexpress-id {
margin-top: 5px;
}
}
```

order-manager.js

```
/**
 * JavaScript para la gestión de pedidos con AliExpress
 */
jQuery(document).ready(function($) {
// Botón para copiar datos del cliente
$('#copy-customer-data').on('click', function(e) {
e.preventDefault();

const orderId = $(this).data('order-id');
const copyStatus = $('#copy-status');

// Mostrar estado procesando
copyStatus.html('<span style="color: #0073aa;">Procesando...</span>');

// Hacer la petición AJAX
$.ajax({
type: 'POST',
url: wcAlidropship.ajaxUrl,
data: {
action: 'aliexpress_copy_customer_data',
nonce: wcAlidropship.nonce,
order_id: orderId
}
```

```

    },
    success: function(response) {
        if (response.success && response.data) {
            // Obtener el JSON como string y formatearlo para copiar
            const jsonData = JSON.stringify(response.data.data);
            console.log('Datos a copiar:', jsonData);

            // Método 1: API moderna (Chrome, Firefox, Edge)
            if (navigator.clipboard && navigator.clipboard.writeText) {
                navigator.clipboard.writeText(jsonData)
                    .then(function() {
                        copyStatus.html('<span style="color: #4CAF50;">' +
                            wcAlidropship.copySuccess + '</span>');
                        setTimeout(function() {
                            copyStatus.html("");
                        }, 3000);
                    })
                    .catch(function(err) {
                        console.error('Error clipboard API:', err);
                        // Intentar método 2 como fallback
                        copyToClipboardFallback(jsonData);
                    });
            } else {
                // Método 2: Fallback para navegadores antiguos
                copyToClipboardFallback(jsonData);
            }
        } else {
            copyStatus.html('<span style="color: #F44336;">' + (response.data ?
                response.data.message : wcAlidropship.copyError) + '</span>');
            setTimeout(function() {
                copyStatus.html("");
            }, 3000);
        }
    },
    error: function() {
        copyStatus.html('<span style="color: #F44336;">' +
            wcAlidropship.copyError + '</span>');
        setTimeout(function() {
            copyStatus.html("");
        }, 3000);
    }
});
});
});

```

```

// Método de fallback para copiar al portapapeles
function copyToClipboardFallback(text) {
  // Crear elemento textarea temporal
  const textarea = document.createElement('textarea');
  textarea.value = text;

  // Hacer que el textarea no sea visible pero sí esté en el DOM
  textarea.style.position = 'fixed';
  textarea.style.top = '0';
  textarea.style.left = '0';
  textarea.style.width = '2em';
  textarea.style.height = '2em';
  textarea.style.padding = '0';
  textarea.style.border = 'none';
  textarea.style.outline = 'none';
  textarea.style.boxShadow = 'none';
  textarea.style.background = 'transparent';

  document.body.appendChild(textarea);
  textarea.focus();
  textarea.select();

  let successful = false;
  try {
    successful = document.execCommand('copy');
  } catch (err) {
    console.error('Error en fallback copy:', err);
  }

  document.body.removeChild(textarea);

  const copyStatus = $('#copy-status');
  if (successful) {
    copyStatus.html('<span style="color: #4CAF50;">' +
wcAlidropship.copySuccess + '</span>');
  } else {
    copyStatus.html('<span style="color: #F44336;">' + wcAlidropship.copyError
+ '</span>');
  }

  setTimeout(function() {
    copyStatus.html("");
  }, 3000);
}

```



```
// Implementar función para extraer ID de productos de AliExpress
```

```
function extractAliExpressItemId(url) {
```

```
    // Diferentes patrones de URL de AliExpress
```

```
    var patterns = [
```

```
        /\itemV(\d+)\.html/,
```

```
        /\(\d+)\.html/,
```

```
        /\_(\d+)\.html/,
```

```
        /aliexpress\.com\itemV(\d+)/,
```

```
        /aliexpress\.com\iV(\d+)\.html/,
```

```
        /\productV(\d+)\.html/
```

```
    ];
```

```
    for (var i = 0; i < patterns.length; i++) {
```

```
        var match = url.match(patterns[i]);
```

```
        if (match && match[1]) {
```

```
            return match[1];
```

```
        }
```

```
    }
```

```
    return "";
```

```
}
```

```
// Autoextraer IDs en la página de edición de producto
```

```
$('#_aliexpress_product_url').on('change', function() {
```

```
    var url = $(this).val();
```

```
    if (url) {
```

```
        var itemId = extractAliExpressItemId(url);
```

```
        if (itemId) {
```

```
            $('#_aliexpress_item_id').val(itemId);
```

```
        // Limpiar cualquier mensaje de estado anterior
```

```
        $('#.aliexpress-extraction-status').html("");
```

```
        // Actualizar análisis de beneficio si los campos de precio están completos
```

```
        updateProfitAnalysis();
```

```
    }
```

```
}
```

```
});
```

```
// Botón para verificar URL y extraer datos
```

```
$('#.extract-aliexpress-data').on('click', function() {
```

```
    var url = $('#_aliexpress_product_url').val();
```

```
    var statusElem = $('#.aliexpress-extraction-status');
```

```

        if (!url) {
            statusElem.html('<span style="color: #f44336;">Por favor, ingresa una URL
de AliExpress</span>');
            return;
        }

        // Mostrar estado de procesamiento
        statusElem.html('<span style="color: #2196F3;">Extrayendo datos...</span>');

        // Extraer ID del producto
        var itemId = extractAliExpressItemId(url);
        if (itemId) {
            $('#_aliexpress_item_id').val(itemId);
            statusElem.html('<span style="color: #4CAF50;">ID extraído correctamente: '
+ itemId + '</span>');

            // Simular cálculo de beneficio actualizando los campos
            updateProfitAnalysis();
        } else {
            statusElem.html('<span style="color: #FF9800;">No se pudo extraer el ID.
Verifica la URL.</span>');
        }
    });

    // Actualizar análisis de beneficio al cambiar precio o costo
    $('#_aliexpress_price, #_aliexpress_shipping, #_regular_price').on('change',
function() {
        updateProfitAnalysis();
    });

    // Función para actualizar el análisis de beneficio
    function updateProfitAnalysis() {
        var regularPrice = parseFloat($('#_regular_price').val()) || 0;
        var aliPrice = parseFloat($('#_aliexpress_price').val()) || 0;
        var aliShipping = parseFloat($('#_aliexpress_shipping').val()) || 0;

        if (regularPrice > 0 && aliPrice > 0) {
            var profit = regularPrice - aliPrice - aliShipping;
            var profitMargin = regularPrice > 0 ? (profit / regularPrice) * 100 : 0;

            // Si ya existe el div de análisis, actualizarlo
            if ($('#_analysis-profit').length) {
                $('#_analysis-profit').remove();
            }
        }
    }

```

```
}
```

```
    // Crear div de análisis
    var profitClass = profit > 0 ? 'positive' : 'negative';
    var analysisHtml = '<div class="analysis-profit">';
    analysisHtml += '<h4>Análisis de beneficio</h4>';
    analysisHtml += '<p><strong>Precio de venta:</strong> ' +
formatCurrency(regularPrice) + '</p>';
    analysisHtml += '<p><strong>Costo de producto:</strong> ' +
formatCurrency(aliPrice) + '</p>';
    analysisHtml += '<p><strong>Costo de envío:</strong> ' +
formatCurrency(aliShipping) + '</p>';
    analysisHtml += '<p><strong>Beneficio:</strong> <span class="' +
profitClass + '">' + formatCurrency(profit) + ' (' + profitMargin.toFixed(2) +
'%)</span></p>';
    analysisHtml += '</div>';
```

```
    // Añadir después del último campo del grupo
    $('#options_group').last().append(analysisHtml);
  }
}
```

```
    // Función para formatear moneda
    function formatCurrency(amount) {
      return new Intl.NumberFormat('es-ES', { style: 'currency', currency: 'EUR'
}).format(amount);
    }
```

```
    // Ejecutar al cargar si hay datos
    if ($('#_aliexpress_price').length && $('#_aliexpress_price').val()) {
      setTimeout(updateProfitAnalysis, 100);
    }
```

```
    // Registrar eventos de depuración para ayudar a solucionar problemas
    console.log('WC AliExpress Dropshipping: Scripts cargados');
```

```
    // Verificar si el script se está ejecutando en las páginas correctas
    if ($('#copy-customer-data').length) {
      console.log('WC AliExpress Dropshipping: Botón de copia detectado');
    }
```

```
    if ($('#_aliexpress_product_url').length) {
      console.log('WC AliExpress Dropshipping: Campo URL de producto
detectado');
```

```
}  
});
```

```
class-wc-alidropship-order-manager.php
```

```
<?php
```

```
/**
```

```
 * Clase para gestionar pedidos y su integración con AliExpress
```

```
 */
```

```
class WC_AliDropship_Order_Manager {
```

```
    /**
```

```
     * Constructor
```

```
     */
```

```
    public function __construct() {
```

```
        // Añadir metabox en la página de pedido
```

```
        add_action('add_meta_boxes', array($this, 'add_order_meta_box'));
```

```
        // AJAX para copiar datos del cliente
```

```
        add_action('wp_ajax_alieexpress_copy_customer_data', array($this,  
'ajax_copy_customer_data'));
```

```
        // Añadir estilos y scripts necesarios
```

```
        add_action('admin_enqueue_scripts', array($this, 'enqueue_admin_scripts'));
```

```
    }
```

```
    /**
```

```
     * Añadir metabox en la página de pedido
```

```
     */
```

```
    public function add_order_meta_box() {
```

```
        // Detectar si HPOS está activo y ajustar la pantalla según corresponda
```

```
        $screen = 'shop_order';
```

```
        // Compatibilidad con HPOS
```

```
        if
```

```
(class_exists('\Automattic\WooCommerce\Internal\DataStores\Orders\CustomOrders  
TableController')) {
```

```
            try {
```

```
                if (method_exists(wc_get_container(), 'get') &&
```

```
                    wc_get_container()->get('\Automattic\WooCommerce\Internal\DataStores\Orders\Cu  
stomOrdersTableController')->custom_orders_table_usage_is_enabled()) {
```

```
                        $screen = wc_get_page_screen_id('shop-order');
```

```
                    }
```

```

    } catch (Exception $e) {
        // Mantener pantalla predeterminada si hay error
    }
}

```

```

add_meta_box(
    'wc_alieexpress_dropship_data',
    __('Datos para AliExpress', 'wc-alieexpress-dropship'),
    array($this, 'render_order_meta_box'),
    $screen,
    'side',
    'high'
);
}

```

```

/**
 * Renderizar el contenido del metabox de pedido
 */
public function render_order_meta_box($post_or_order_obj) {
    // Obtener el ID del pedido
    $order_id = $post_or_order_obj;
    if (is_object($post_or_order_obj)) {
        if (method_exists($post_or_order_obj, 'get_id')) {
            // Es un objeto WC_Order (HPOS)
            $order_id = $post_or_order_obj->get_id();
        } else if (isset($post_or_order_obj->ID)) {
            // Es un objeto WP_Post (Legacy)
            $order_id = $post_or_order_obj->ID;
        }
    }
}

```

```

    // Obtener la orden
    $order = wc_get_order($order_id);
    if (!$order) {
        echo '<p>' . esc_html__( 'No se pudo cargar la información del pedido.',
'wc-alieexpress-dropship' ) . '</p>';
        return;
    }

```

```

    // Datos del cliente
    $this->render_customer_data_section($order);

```

```

    // Productos del pedido
    $this->render_products_section($order);

```

```
}
```

```
/**
```

```
 * Renderizar sección de datos del cliente
```

```
 */
```

```
private function render_customer_data_section($order) {
```

```
    // Obtener los datos de dirección
```

```
    $address = $order->get_address('shipping');
```

```
    if (empty($address) || empty($address['address_1'])) {
```

```
        $address = $order->get_address('billing');
```

```
    }
```

```
    // Asegurar que tenemos un número de teléfono
```

```
    $phone = !empty($address['phone']) ? $address['phone'] :
```

```
$order->get_billing_phone();
```

```
    // Obtener prefijo telefónico
```

```
    $phone_prefix = get_option('wc_alidropship_phone_prefix', '+34');
```

```
    // Formatear datos para AliExpress
```

```
    $customer_data = array(
```

```
        'firstName' => $address['first_name'] ?? "",
```

```
        'lastName'  => $address['last_name'] ?? "",
```

```
        'phonePrefix' => $phone_prefix,
```

```
        'phoneNumber' => preg_replace('/[^0-9]/', "", $phone),
```

```
        'streetAddress' => $address['address_1'] ?? "",
```

```
        'apartment'    => $address['address_2'] ?? "",
```

```
        'zipCode'      => $address['postcode'] ?? "",
```

```
        'city'         => $address['city'] ?? "",
```

```
        'state'        => $address['state'] ?? "",
```

```
        'country'      => $address['country'] ?? "",
```

```
        'email'        => $order->get_billing_email()
```

```
    );
```

```
    // Codificar los datos para JSON
```

```
    $customer_json = htmlspecialchars(json_encode($customer_data,  
JSON_PRETTY_PRINT), ENT_QUOTES, 'UTF-8');
```

```
    ?>
```

```
<div class="aliexpress-customer-data">
```

```
    <h4><?php esc_html_e('Datos del Cliente', 'wc-aliexpress-dropship');  
?></h4>
```

```
    <input type="hidden" id="aliexpress-customer-json" value="<?php echo  
$customer_json; ?>">
```

```

<table class="widefat">
  <tbody>
    <tr>
      <td><strong><?php esc_html_e('Nombre:', 'wc-aliexpress-dropship');
?></strong></td>
      <td><?php echo esc_html($address['first_name'] . ' ' .
$address['last_name']); ?></td>
    </tr>
    <tr>
      <td><strong><?php esc_html_e('Dirección:',
'wc-aliexpress-dropship'); ?></strong></td>
      <td>
        <?php echo esc_html($address['address_1']); ?>
        <?php if (!empty($address['address_2'])) echo '<br>' .
esc_html($address['address_2']); ?>
      </td>
    </tr>
    <tr>
      <td><strong><?php esc_html_e('Ciudad:', 'wc-aliexpress-dropship');
?></strong></td>
      <td><?php echo esc_html($address['city']); ?></td>
    </tr>
    <tr>
      <td><strong><?php esc_html_e('CP:', 'wc-aliexpress-dropship');
?></strong></td>
      <td><?php echo esc_html($address['postcode']); ?></td>
    </tr>
    <tr>
      <td><strong><?php esc_html_e('País:', 'wc-aliexpress-dropship');
?></strong></td>
      <td><?php echo
esc_html(WC()->countries->get_countries())[$address['country']] ??
$address['country']; ?></td>
    </tr>
    <tr>
      <td><strong><?php esc_html_e('Teléfono:', 'wc-aliexpress-dropship');
?></strong></td>
      <td><?php echo esc_html($phone); ?></td>
    </tr>
  </tbody>
</table>

<p>

```

```

        <button id="copy-customer-data" class="button button-primary"
data-order-id="<?php echo esc_attr($order->get_id()); ?>">
        <?php esc_html_e('Copiar datos', 'wc-aliexpress-dropship'); ?>
    </button>
    <span id="copy-status" style="margin-left: 10px;"></span>
</p>
</div>
<?php
}

```

```

/**
 * Renderizar sección de productos
 */
private function render_products_section($order) {
    global $wpdb;

    // Tabla personalizada
    $table_name = $wpdb->prefix . 'wc_alidropship_products';

    // Obtener items del pedido
    $order_items = $order->get_items();

    if (empty($order_items)) {
        echo '<p>' . esc_html__('Este pedido no contiene productos.',
'wc-aliexpress-dropship') . '</p>';
        return;
    }
    ?>
    <div class="aliexpress-order-products">
        <h4><?php esc_html_e('Productos para Comprar', 'wc-aliexpress-dropship');
?></h4>

```

```

        <ul class="order-products-list">
            <?php foreach ($order_items as $item_id => $item) :
                $product_id = $item->get_product_id();
                $variation_id = $item->get_variation_id();
                $target_id = $variation_id ? $variation_id : $product_id;
                $product = wc_get_product($target_id);

                if (!$product) continue;

                // Obtener datos de AliExpress de la tabla personalizada
                $aliexpress_data = $wpdb->get_row(

```



```
$wpdb->prepare("SELECT * FROM $table_name WHERE product_id  
= %d", $product_id)  
);
```

```
// O desde los metadatos del producto  
if (!$aliexpress_data) {  
    $aliexpress_url = $product->get_meta('_aliexpress_product_url');  
    $aliexpress_item_id = $product->get_meta('_aliexpress_item_id');  
} else {  
    $aliexpress_url = $aliexpress_data->aliexpress_url;  
    $aliexpress_item_id = $aliexpress_data->aliexpress_item_id;  
}
```

```
// Opciones/atributos del producto  
$product_options = array();
```

```
// Si es una variación, obtener los atributos  
if ($variation_id) {  
    $variation = wc_get_product($variation_id);  
    if ($variation) {  
        $attributes = $variation->get_attributes();  
        if (!empty($attributes)) {  
            foreach ($attributes as $attribute_name => $attribute_value) {  
                $product_options[] =  
wc_attribute_label(str_replace('attribute_', '', $attribute_name)) . ':' .  
$attribute_value;  
            }  
        }  
    }  
}
```

```
// Metadatos adicionales del ítem  
$item_meta_data = $item->get_formatted_meta_data();  
if (!empty($item_meta_data)) {  
    foreach ($item_meta_data as $meta) {  
        if (!empty($meta->display_key) && !empty($meta->display_value))  
{  
            $product_options[] = strip_tags($meta->display_key) . ':' .  
strip_tags($meta->display_value);  
        }  
    }  
}  
?>  
<li>
```

```
<div class="product-info">
    <span class="product-name">
        <?php echo esc_html($item->get_name()); ?> × <?php echo
esc_html($item->get_quantity()); ?>
    </span>
```

```
<?php if (!empty($product_options)) : ?>
    <div class="product-options">
        <?php foreach ($product_options as $option) : ?>
            <span><?php echo esc_html($option); ?></span>
        <?php endforeach; ?>
    </div>
<?php endif; ?>
```

```
<?php if (!empty($aliexpress_url)) : ?>
    <div class="aliexpress-actions">
        <a href="<?php echo esc_url($aliexpress_url); ?>" target="_blank"
class="button">
            <?php esc_html_e('Ir al Producto', 'wc-aliexpress-dropship'); ?>
        </a>
```

```
<?php if (!empty($aliexpress_item_id)) : ?>
    <span class="aliexpress-id">
        ID: <?php echo esc_html($aliexpress_item_id); ?>
    </span>
<?php endif; ?>
</div>
<?php else : ?>
    <div class="aliexpress-missing">
        <span style="color: #f44336;">
            <?php esc_html_e('No hay enlace de AliExpress para este
producto.', 'wc-aliexpress-dropship'); ?>
        </span>
    </div>
<?php endif; ?>
</div>
</li>
<?php endforeach; ?>
</ul>
</div>
<?php
}
```

```
/**
```

```

* Cargar scripts y estilos necesarios
*/
public function enqueue_admin_scripts($hook) {
    // Solo cargar en la página de edición de pedidos
    $is_order_page = false;

    // Verificar si estamos en la página de edición de pedidos
    if ('post.php' === $hook) {
        global $post;
        if ($post && 'shop_order' === $post->post_type) {
            $is_order_page = true;
        }
    }

    // Compatibilidad con HPOS
    if ('woocommerce_page_wc-orders' === $hook && isset($_GET['action']) &&
'edit' === $_GET['action']) {
        $is_order_page = true;
    }

    if (!$is_order_page) {
        return;
    }

    // Registrar y cargar scripts
    wp_enqueue_script(
        'wc-alidropship-admin',
        WC_ALIDROPSHIP_PLUGIN_URL . 'assets/js/order-manager.js',
        array('jquery'),
        WC_ALIDROPSHIP_VERSION,
        true
    );

    // Localizar script
    wp_localize_script('wc-alidropship-admin', 'wcAlidropship', array(
        'ajaxUrl' => admin_url('admin-ajax.php'),
        'nonce' => wp_create_nonce('wc_alidropship_nonce'),
        'copySuccess' => __('Datos copiados', 'wc-aliexpress-dropship'),
        'copyError' => __('Error al copiar', 'wc-aliexpress-dropship')
    ));

    // Estilos
    wp_enqueue_style(
        'wc-alidropship-admin',

```

```

        WC_ALIDROPSHIP_PLUGIN_URL . 'assets/css/admin.css',
        array(),
        WC_ALIDROPSHIP_VERSION
    );
}

/**
 * AJAX para copiar datos del cliente
 */
public function ajax_copy_customer_data() {
    // Verificar nonce
    if (!isset($_POST['nonce']) || !wp_verify_nonce($_POST['nonce'],
'wc_alidropship_nonce')) {
        wp_send_json_error(array('message' => __('Error de seguridad.',
'wc-aliexpress-dropship')));
    }

    // Verificar permiso
    if (!current_user_can('edit_shop_orders')) {
        wp_send_json_error(array('message' => __('No tienes permiso para esta
acción.', 'wc-aliexpress-dropship')));
    }

    // Verificar ID del pedido
    if (!isset($_POST['order_id']) || empty($_POST['order_id'])) {
        wp_send_json_error(array('message' => __('ID de pedido no válido.',
'wc-aliexpress-dropship')));
    }

    $order_id = intval($_POST['order_id']);
    $order = wc_get_order($order_id);

    if (!$order) {
        wp_send_json_error(array('message' => __('Pedido no encontrado.',
'wc-aliexpress-dropship')));
    }

    // Obtener los datos de dirección
    $address = $order->get_address('shipping');
    if (empty($address) || empty($address['phone'])) {
        $address = $order->get_address('billing');
    }

    // Asegurarnos de tener un número de teléfono

```

```
$phone = !empty($address['phone']) ? $address['phone'] :  
$order->get_billing_phone();
```

```
// Obtener prefijo telefónico
```

```
$phone_prefix = get_option('wc_alidropship_phone_prefix', '+34');
```

```
// Procesar el campo de provincia para España
```

```
if (isset($address['country']) && $address['country'] === 'ES') {
```

```
    // Mapa de códigos de provincia de España a nombres completos
```

```
    $spanish_states = array(  
        'C' => 'A Coruña',  
        'VI' => 'Álava',  
        'AB' => 'Albacete',  
        'A' => 'Alicante',  
        'AL' => 'Almería',  
        'O' => 'Asturias',  
        'AV' => 'Ávila',  
        'BA' => 'Badajoz',  
        'PM' => 'Islas Baleares',  
        'B' => 'Barcelona',  
        'BU' => 'Burgos',  
        'CC' => 'Cáceres',  
        'CA' => 'Cádiz',  
        'S' => 'Cantabria',  
        'CS' => 'Castellón',  
        'CR' => 'Ciudad Real',  
        'CO' => 'Córdoba',  
        'CU' => 'Cuenca',  
        'GI' => 'Girona',  
        'GR' => 'Granada',  
        'GU' => 'Guadalajara',  
        'SS' => 'Gipuzkoa',  
        'H' => 'Huelva',  
        'HU' => 'Huesca',  
        'J' => 'Jaén',  
        'LO' => 'La Rioja',  
        'GC' => 'Las Palmas',  
        'LE' => 'León',  
        'L' => 'Lleida',  
        'LU' => 'Lugo',  
        'M' => 'Madrid',  
        'MA' => 'Málaga',  
        'MU' => 'Murcia',  
        'NA' => 'Navarra',
```

```

        'OR' => 'Ourense',
        'P' => 'Palencia',
        'PO' => 'Pontevedra',
        'SA' => 'Salamanca',
        'TF' => 'Santa Cruz de Tenerife',
        'SG' => 'Segovia',
        'SE' => 'Sevilla',
        'SO' => 'Soria',
        'T' => 'Tarragona',
        'TE' => 'Teruel',
        'TO' => 'Toledo',
        'V' => 'Valencia',
        'VA' => 'Valladolid',
        'BI' => 'Vizcaya',
        'ZA' => 'Zamora',
        'Z' => 'Zaragoza',
        'CE' => 'Ceuta',
        'ML' => 'Melilla'
    );

```

```

    $province_found = false;

```

```

    // Prioridad 1: Intentar obtener el nombre completo de WooCommerce
    if (!empty($address['state']) && strlen($address['state']) <= 3) { // Si parece un
código
        if (function_exists('WC') && method_exists(WC(), 'countries')) {
            $wc_states = WC()->countries->get_states($address['country']);
            if (isset($wc_states[$address['state']])) {
                $address['state'] = $wc_states[$address['state']];
                $province_found = true;
                error_log('Provincia obtenida de WooCommerce: ' . $address['state']);
            }
        }
    }
}

```

```

    // Prioridad 2: Si lo anterior falla, buscar en los datos formateados de dirección
    if (!$province_found) {
        // Extraer la provincia de los datos postales mostrados en pantalla
        $shipping_address_text = trim($order->get_formatted_shipping_address());
        $billing_address_text = trim($order->get_formatted_billing_address());
    }

```

```

    // Buscar líneas que podrían contener la provincia
    $address_text = $shipping_address_text ? $billing_address_text;
    $address_lines = explode("\n", $address_text);

```

```

        foreach ($address_lines as $line) {
            $line = trim($line);
            // Si la línea no contiene el código postal ni la ciudad, podría ser la
            provincia
            if (!empty($line) && !preg_match('/\d{5}/', $line) && isset($address['city'])
            && $line != $address['city']) {
                // Verificar si alguna provincia conocida está en la línea
                foreach ($spanish_states as $code => $province_name) {
                    if (strpos($line, $province_name) !== false) {
                        $address['state'] = $province_name;
                        $province_found = true;
                        error_log('Provincia encontrada en dirección formateada: ' .
                        $address['state']);
                        break 2; // Salir de ambos bucles
                    }
                }
            }
        }
    }
}

```

```

        // Prioridad 3: Si todo lo anterior falla, convertir el código a nombre completo
        if (!$province_found && !empty($address['state']) &&
        isset($spanish_states[$address['state']])) {
            $address['state'] = $spanish_states[$address['state']];
            error_log('Provincia convertida de código a nombre: ' . $address['state']);
        }
    }
}

```

```

// Depuración: mostrar en el log lo que estamos obteniendo
error_log('Datos de dirección para AliExpress: ' . print_r($address, true));

```

```

// Formatear datos para AliExpress
$customer_data = array(
    'firstName' => $address['first_name'] ?? "",
    'lastName' => $address['last_name'] ?? "",
    'phonePrefix' => $phone_prefix,
    'phoneNumber' => preg_replace('/[^0-9]/', "", $phone),
    'streetAddress' => $address['address_1'] ?? "",
    'apartment' => $address['address_2'] ?? "",
    'zipCode' => $address['postcode'] ?? "",
    'city' => $address['city'] ?? "",
    'state' => $address['state'] ?? "",
    'country' => $address['country'] ?? "",

```

```

        'email'      => $order->get_billing_email(),
        'countryCode' => $address['country'] ?? 'ES',
    );

    // Marcar el pedido como procesado para dropshipping
    $order->add_order_note(__( 'Datos copiados para proceso de dropshipping en
AliExpress.', 'wc-aliexpress-dropship' ));

    // Guardar metadatos en el pedido (compatible con HPOS)
    $order->update_meta_data('_aliexpress_dropship_processed', 'yes');
    $order->update_meta_data('_aliexpress_dropship_processed_at',
current_time('mysql'));
    $order->save();

    wp_send_json_success(array(
        'message' => __( 'Datos copiados exitosamente.', 'wc-aliexpress-dropship' ),
        'data' => $customer_data
    ));
} // Cierre del método ajax_copy_customer_data

} // Cierre de la clase WC_AliDropship_Order_Manager

```

class-wc-alidropship-product-fields.php

```

<?php
/**
 * Clase para gestionar campos adicionales en productos
 */
class WC_AliDropship_Product_Fields {

    /**
     * Constructor
     */
    public function __construct() {
        // Añadir campos en editor de producto
        add_action('woocommerce_product_options_general_product_data',
array($this, 'add_aliexpress_fields'));

        // Guardar campos
        add_action('woocommerce_process_product_meta', array($this,
'save_aliexpress_fields'));
    }

    /**

```



```

* Añadir campos para AliExpress en la pestaña general del producto
*/
public function add_aliexpress_fields() {
    global $post;

    echo '<div class="options_group">';

    // Título de la sección
    echo '<h4 style="padding-left: 10px;">' . esc_html__( 'Información de
AliExpress', 'wc-aliexpress-dropship' ) . '</h4>';

    // URL del producto en AliExpress
    woocommerce_wp_text_input(
        array(
            'id'      => '_aliexpress_product_url',
            'label'    => __( 'URL del producto', 'wc-aliexpress-dropship' ),
            'placeholder' => 'https://www.aliexpress.com/item/...',
            'desc_tip'  => true,
            'description' => __( 'Ingresa la URL completa del producto en AliExpress.',
'wc-aliexpress-dropship' )
        )
    );

    // ID del producto en AliExpress (se extraerá automáticamente)
    woocommerce_wp_text_input(
        array(
            'id'      => '_aliexpress_item_id',
            'label'    => __( 'ID del producto', 'wc-aliexpress-dropship' ),
            'placeholder' => 'Se extrae automáticamente de la URL',
            'desc_tip'  => true,
            'description' => __( 'ID único del producto en AliExpress.',
'wc-aliexpress-dropship' )
        )
    );

    // Precio del producto en AliExpress
    woocommerce_wp_text_input(
        array(
            'id'      => '_aliexpress_price',
            'label'    => __( 'Precio en AliExpress', 'wc-aliexpress-dropship' ),
            'placeholder' => '0.00',
            'desc_tip'  => true,
            'description' => __( 'Precio de compra en AliExpress (para cálculo de
beneficios)', 'wc-aliexpress-dropship' ),

```

```

        'type'      => 'number',
        'custom_attributes' => array(
            'step' => '0.01',
            'min' => '0'
        )
    );
};

```

```

// Costo de envío desde AliExpress
woocommerce_wp_text_input(
    array(
        'id'      => '_aliexpress_shipping',
        'label'    => __('Costo de envío', 'wc-aliexpress-dropship'),
        'placeholder' => '0.00',
        'desc_tip' => true,
        'description' => __('Costo de envío desde AliExpress.',
'wc-aliexpress-dropship'),
        'type'      => 'number',
        'custom_attributes' => array(
            'step' => '0.01',
            'min' => '0'
        )
    )
);

```

```

// Botón para verificar URL y extraer datos
echo '<p class="form-field">';
echo '<label>&nbsp;</label>';
echo '<button type="button" class="button button-secondary
extract-aliexpress-data">' . esc_html__( 'Extraer datos de la URL',
'wc-aliexpress-dropship') . '</button>';
echo '<span class="aliexpress-extraction-status" style="margin-left:
10px;"></span>';
echo '</p>';

```

```

// Cálculo de beneficio
$product = wc_get_product($post->ID);
if ($product) {
    $regular_price = $product->get_regular_price();
    $aliexpress_price = $product->get_meta('_aliexpress_price', true);
    $aliexpress_shipping = $product->get_meta('_aliexpress_shipping', true);

    if ($regular_price && $aliexpress_price) {
        $profit = $regular_price - $aliexpress_price - $aliexpress_shipping;
    }
}

```

```
$profit_margin = $regular_price > 0 ? ($profit / $regular_price) * 100 : 0;
```

```
    echo '<div style="margin: 10px; padding: 10px; background: #f8f8f8;
border-left: 3px solid #2271b1;">';
    echo '<h4 style="margin-top: 0;">' . esc_html__( 'Análisis de beneficio',
'wc-aliexpress-dropship' ) . '</h4>';
    echo '<p><strong>' . esc_html__( 'Precio de venta:',
'wc-aliexpress-dropship' ) . '</strong>' . wc_price($regular_price) . '</p>';
    echo '<p><strong>' . esc_html__( 'Costo de producto:',
'wc-aliexpress-dropship' ) . '</strong>' . wc_price($aliexpress_price) . '</p>';
    echo '<p><strong>' . esc_html__( 'Costo de envío:',
'wc-aliexpress-dropship' ) . '</strong>' . wc_price($aliexpress_shipping) . '</p>';
    echo '<p><strong>' . esc_html__( 'Beneficio:', 'wc-aliexpress-dropship' ) .
'</strong>' . wc_price($profit) . ' (' . number_format($profit_margin, 2) . '%)</p>';
    echo '</div>';
    }
}
```

```
echo '</div>';
```

```
// Script para extraer ID y otros datos de la URL
?>
<script type="text/javascript">
jQuery(document).ready(function($) {
    // Función para extraer ID de producto de la URL de AliExpress
    function extractAliExpressItemId(url) {
        // Diferentes patrones de URL de AliExpress
        var patterns = [
            /\bitemV(\d+)\.html/,
            /\bV(\d+)\.html/,
            /\b_(\d+)\.html/,
            /\baliexpress\.com\bitemV(\d+)\.
        ];

        for (var i = 0; i < patterns.length; i++) {
            var match = url.match(patterns[i]);
            if (match && match[1]) {
                return match[1];
            }
        }

        return "";
    }
}
```

```

        // Evento para extraer datos cuando se ingresa una URL
        $('#_aliexpress_product_url').on('change', function() {
            var url = $(this).val();
            if (url) {
                var itemId = extractAliExpressItemId(url);
                if (itemId) {
                    $('#_aliexpress_item_id').val(itemId);
                }
            }
        });

        // Botón para verificar URL y extraer datos
        $('.extract-aliexpress-data').on('click', function() {
            var url = $('#_aliexpress_product_url').val();
            var statusElem = $('.aliexpress-extraction-status');

            if (!url) {
                statusElem.html('<span style="color: #f44336;">Por favor, ingresa una URL de AliExpress</span>');
                return;
            }

            // Mostrar estado de procesamiento
            statusElem.html('<span style="color: #2196F3;">Extrayendo datos...</span>');

            // Extraer ID del producto
            var itemId = extractAliExpressItemId(url);
            if (itemId) {
                $('#_aliexpress_item_id').val(itemId);
                statusElem.html('<span style="color: #4CAF50;">ID extraído correctamente: ' + itemId + '</span>');
            } else {
                statusElem.html('<span style="color: #FF9800;">No se pudo extraer el ID. Verifica la URL.</span>');
            }
        });
    });
</script>
<?php
}

/**
 * Guardar los campos de AliExpress

```

```

*/
public function save_alieexpress_fields($product_id) {
    global $wpdb;

    $product = wc_get_product($product_id);
    if (!$product) {
        return;
    }

    // URL del producto
    $aliexpress_url = isset($_POST['aliexpress_product_url']) ?
        sanitize_text_field($_POST['aliexpress_product_url']) : "";
    $product->update_meta_data('aliexpress_product_url', $aliexpress_url);

    // ID del producto
    $aliexpress_item_id = isset($_POST['aliexpress_item_id']) ?
        sanitize_text_field($_POST['aliexpress_item_id']) : "";
    $product->update_meta_data('aliexpress_item_id', $aliexpress_item_id);

    // Precio
    $aliexpress_price = isset($_POST['aliexpress_price']) ?
        wc_format_decimal($_POST['aliexpress_price']) : "";
    $product->update_meta_data('aliexpress_price', $aliexpress_price);

    // Envío
    $aliexpress_shipping = isset($_POST['aliexpress_shipping']) ?
        wc_format_decimal($_POST['aliexpress_shipping']) : "";
    $product->update_meta_data('aliexpress_shipping', $aliexpress_shipping);

    // Guardar los cambios
    $product->save();

    // Actualizar o insertar en la tabla personalizada
    $table_name = $wpdb->prefix . 'wc_alidropship_products';

    $exists = $wpdb->get_var(
        $wpdb->prepare("SELECT id FROM $table_name WHERE product_id =
%d", $product_id)
    );

    if ($exists) {
        // Actualizar
        $wpdb->update(
            $table_name,

```



```
// Añadir metabox de cómo usar
add_action('add_meta_boxes', array($this, 'add_how_to_use_metabox'));
}
```

```
/**
 * Añadir menú de administración
 */
public function add_admin_menu() {
    add_submenu_page(
        'woocommerce',
        __('AliExpress Dropshipping', 'wc-aliexpress-dropship'),
        __('AliExpress Dropshipping', 'wc-aliexpress-dropship'),
        'manage_woocommerce',
        'wc-alidropship-settings',
        array($this, 'settings_page')
    );
}
```

```
/**
 * Registrar configuraciones
 */
public function register_settings() {
    register_setting('wc_alidropship_settings', 'wc_alidropship_phone_prefix');
    register_setting('wc_alidropship_settings', 'wc_alidropship_extension_url');
}
```

```
/**
 * Renderizar página de configuración
 */
public function settings_page() {
    ?>
    <div class="wrap">
        <h1><?php esc_html_e('WooCommerce AliExpress Dropshipping',
'wc-aliexpress-dropship'); ?></h1>

        <div class="notice notice-info" style="margin: 15px 0;">
            <p><?php esc_html_e('Este plugin te ayuda a automatizar el proceso de
dropshipping entre WooCommerce y AliExpress.', 'wc-aliexpress-dropship'); ?></p>
        </div>

        <div class="card" style="max-width: 800px;">
            <h2><?php esc_html_e('Configuración General', 'wc-aliexpress-dropship');
?></h2>
```

```

        <form method="post" action="options.php">
            <?php settings_fields('wc_alidropship_settings'); ?>
            <?php do_settings_sections('wc_alidropship_settings'); ?>

            <table class="form-table">
                <tr valign="top">
                    <th scope="row"><?php esc_html_e('Prefijo telefónico
predeterminado', 'wc-aliexpress-dropship'); ?></th>
                    <td>
                        <input type="text" name="wc_alidropship_phone_prefix"
value="<?php echo esc_attr(get_option('wc_alidropship_phone_prefix', '+34')); ?>" />
                        <p class="description"><?php esc_html_e('Se usará cuando no
se pueda determinar el prefijo a partir del número de teléfono.',
'wc-aliexpress-dropship'); ?></p>
                    </td>
                </tr>

                <tr valign="top">
                    <th scope="row"><?php esc_html_e('URL de la extensión de
navegador', 'wc-aliexpress-dropship'); ?></th>
                    <td>
                        <input type="text" name="wc_alidropship_extension_url"
class="regular-text" value="<?php echo
esc_attr(get_option('wc_alidropship_extension_url', '')); ?>" />
                        <p class="description"><?php esc_html_e('Enlace para
descargar la extensión de navegador (opcional).', 'wc-aliexpress-dropship'); ?></p>
                    </td>
                </tr>
            </table>

            <?php submit_button(); ?>
        </form>
    </div>

    <div class="card" style="max-width: 800px; margin-top: 20px;">
        <h2><?php esc_html_e('Instrucciones de Uso', 'wc-aliexpress-dropship');
?></h2>

        <h3><?php esc_html_e('Paso 1: Configurar productos',
'wc-aliexpress-dropship'); ?></h3>
        <p><?php esc_html_e('Para cada producto en tu tienda WooCommerce:',
'wc-aliexpress-dropship'); ?></p>
        <ol>

```



```
<li><?php esc_html_e('Ve a la página de edición del producto',  
'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Encuentra la sección "Información de  
AliExpress"', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Introduce la URL del producto en AliExpress',  
'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Haz clic en "Extraer datos de la URL" para  
obtener el ID automáticamente', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Opcionalmente, ingresa el precio y costo de  
envío para calcular beneficios', 'wc-aliexpress-dropship'); ?></li>  
</ol>
```

```
<h3><?php esc_html_e('Paso 2: Instalar la extensión de navegador',  
'wc-aliexpress-dropship'); ?></h3>  
<p><?php esc_html_e('Para autocompletar formularios en AliExpress:',  
'wc-aliexpress-dropship'); ?></p>  
<ol>  
<li><?php esc_html_e('Descarga e instala la extensión de navegador  
desde el enlace proporcionado', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('La extensión permitirá autocompletar los datos  
del cliente en AliExpress', 'wc-aliexpress-dropship'); ?></li>  
</ol>
```

```
<h3><?php esc_html_e('Paso 3: Procesar pedidos',  
'wc-aliexpress-dropship'); ?></h3>  
<p><?php esc_html_e('Cuando recibas un pedido en WooCommerce:',  
'wc-aliexpress-dropship'); ?></p>  
<ol>  
<li><?php esc_html_e('Ve a la página de edición del pedido',  
'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('En el panel lateral encontrarás los "Datos para  
AliExpress"', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Haz clic en "Copiar datos" para guardar la  
información del cliente', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('En la sección de productos, haz clic en "Ir al  
Producto" para abrir AliExpress', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Usa la extensión para autocompletar los datos  
de envío', 'wc-aliexpress-dropship'); ?></li>  
<li><?php esc_html_e('Completa la compra en AliExpress',  
'wc-aliexpress-dropship'); ?></li>  
</ol>
```

```
<div style="margin-top: 20px; padding: 10px; background-color: #f8f8f8;  
border-left: 4px solid #2271b1;">
```

```

        <p><strong><?php esc_html_e('Nota:', 'wc-aliexpress-dropship');
?></strong> <?php esc_html_e('La extensión de navegador solo funciona en los
navegadores Google Chrome, Microsoft Edge y Firefox.', 'wc-aliexpress-dropship');
?></p>
    </div>
</div>
</div>
<?php
}

```

```

public function add_how_to_use_metabox() {
    // Detectar si HPOS está activo y ajustar la pantalla según corresponda
    $screen = 'shop_order';

    // Compatibilidad con HPOS
    if
(class_exists('\Automattic\WooCommerce\Internal\DataStores\Orders\CustomOrders
TableController')) {
        try {
            if (method_exists(wc_get_container(), 'get') &&
wc_get_container()->get('\Automattic\WooCommerce\Internal\DataStores\Orders\Cu
stomOrdersTableController')->custom_orders_table_usage_is_enabled()) {
                $screen = wc_get_page_screen_id('shop-order');
            }
        } catch (Exception $e) {
            // Mantener pantalla predeterminada si hay error
        }
    }
}

```

```

add_meta_box(
    'wc_alidropship_how_to_use',
    ('Cómo usar AliExpress Dropshipping', 'wc-aliexpress-dropship'),
    array($this, 'render_how_to_use_metabox'),
    $screen,
    'normal',
    'default'
);
}

```

```

/**
 * Renderizar metabox de instrucciones
 */
public function render_how_to_use_metabox($post) {

```

```

?>
<div class="wc-alidropship-howto">
    <p><?php esc_html_e('Para realizar pedidos en AliExpress con los datos del
cliente:', 'wc-aliexpress-dropship'); ?></p>

    <ol>
        <li>
            <strong><?php esc_html_e('Copia los datos del cliente',
'wc-aliexpress-dropship'); ?></strong><br>
            <?php esc_html_e('Usa el panel lateral "Datos para AliExpress" y haz
clic en el botón "Copiar datos".', 'wc-aliexpress-dropship'); ?>
        </li>
        <li>
            <strong><?php esc_html_e('Abre el producto en AliExpress',
'wc-aliexpress-dropship'); ?></strong><br>
            <?php esc_html_e('Utiliza los botones "Ir al Producto" para cada
artículo en la sección "Dropshipping con AliExpress".', 'wc-aliexpress-dropship'); ?>
        </li>
        <li>
            <strong><?php esc_html_e('Autocompletar formulario',
'wc-aliexpress-dropship'); ?></strong><br>
            <?php esc_html_e('En la página de AliExpress, usa la extensión de
navegador para rellenar automáticamente los datos de envío.',
'wc-aliexpress-dropship'); ?>
        </li>
        <li>
            <strong><?php esc_html_e('Completa la compra',
'wc-aliexpress-dropship'); ?></strong><br>
            <?php esc_html_e('Finaliza el pedido en AliExpress. Recuerda que
enviarás directamente el producto al cliente.', 'wc-aliexpress-dropship'); ?>
        </li>
    </ol>

    <?php $extension_url = get_option('wc_alidropship_extension_url', ""); ?>
    <?php if ($extension_url) : ?>
        <p><a href="<?php echo esc_url($extension_url); ?>" target=" blank"
class="button"><?php esc_html_e('Descargar Extensión de Navegador',
'wc-aliexpress-dropship'); ?></a></p>
    <?php else : ?>
        <p><?php esc_html_e('Si aún no tienes la extensión de navegador,
consulta la página de configuración del plugin.', 'wc-aliexpress-dropship'); ?></p>
    <?php endif; ?>
</div>
<?php

```

```
}  
}
```

woocommerce-aliexpress-dropshipping.php

```
<?php  
/**  
 * Plugin Name: WooCommerce AliExpress Dropshipping  
 * Description: Facilita el dropshipping desde WooCommerce hacia AliExpress  
 automatizando el flujo de pedidos.  
 * Version: 1.0.0  
 * Author: CosasUtilesParaCasa  
 * Text Domain: wc-aliexpress-dropship  
 * Domain Path: /languages  
 * WC requires at least: 3.0.0  
 * WC tested up to: 7.0.0  
 */
```

```
// Evitar acceso directo  
if (!defined('ABSPATH')) {  
    exit;  
}
```

```
// Declarar compatibilidad con características avanzadas de WooCommerce  
add_action('before_woocommerce_init', function() {  
    if (class_exists('\Automattic\WooCommerce\Utilities\FeaturesUtil')) {  
        // HPOS - High Performance Order Storage
```

```
\Automattic\WooCommerce\Utilities\FeaturesUtil::declare_compatibility('custom_order_tables', __FILE__, true);
```

```
        // Cart and Checkout Blocks
```

```
\Automattic\WooCommerce\Utilities\FeaturesUtil::declare_compatibility('cart_checkout_blocks', __FILE__, true);
```

```
        // Product Block Editor
```

```
\Automattic\WooCommerce\Utilities\FeaturesUtil::declare_compatibility('product_block_editor', __FILE__, true);  
    }  
});
```

```
/**
```

```

* Clase principal del plugin
*/
class WC_AliExpress_Dropshipping {

    /**
     * Versión del plugin
     */
    public $version = '1.0.0';

    /**
     * Instancia única (patrón singleton)
     */
    protected static $_instance = null;

    /**
     * Obtener instancia única - Singleton Pattern
     */
    public static function instance() {
        if (is_null(self::$_instance)) {
            self::$_instance = new self();
        }
        return self::$_instance;
    }

    /**
     * Constructor
     */
    public function __construct() {
        // Definir constantes
        $this->define_constants();

        // Inicializar hooks
        $this->init_hooks();

        // Inicializar funcionalidades básicas
        add_action('admin_init', array($this, 'check_requirements'));
        add_action('woocommerce_init', array($this, 'init'));
        add_action('init', array($this, 'load_textdomain'));

        // Comprobar configuración del sistema
        add_action('admin_init', array($this, 'check_system_configuration'));

        // Añadir acción de activación
        register_activation_hook(__FILE__, array($this, 'activate'));

```

```
}
```

```
/**
```

```
 * Definir constantes
```

```
 */
```

```
 private function define_constants() {
```

```
     $this->define('WC_ALIDROPSHIP_VERSION', $this->version);
```

```
     $this->define('WC_ALIDROPSHIP_PLUGIN_DIR', plugin_dir_path( __FILE__ ));
```

```
     $this->define('WC_ALIDROPSHIP_PLUGIN_URL', plugin_dir_url( __FILE__ ));
```

```
 }
```

```
/**
```

```
 * Inicializar hooks
```

```
 */
```

```
 private function init_hooks() {
```

```
     // Añadir enlaces a la página de plugins
```

```
     add_filter('plugin_action_links_' . plugin_basename( __FILE__ ), array($this, 'plugin_action_links'));
```

```
 }
```

```
/**
```

```
 * Definir constante si no está ya definida
```

```
 */
```

```
 private function define($name, $value) {
```

```
     if (!defined($name)) {
```

```
         define($name, $value);
```

```
     }
```

```
 }
```

```
/**
```

```
 * Inicializar el plugin
```

```
 */
```

```
 public function init() {
```

```
     if (class_exists('WooCommerce')) {
```

```
         // Verificar actualizaciones
```

```
         $this->update();
```

```
         // Cargar archivos necesarios
```

```
         $this->includes();
```

```
     }
```

```
 }
```

```
/**
```

```
 * Cargar archivos necesarios
```

```

*/
private function includes() {
    // Core
    if (file_exists(WC_ALIDROPSHIP_PLUGIN_DIR .
'includes/class-wc-alidropship-product-fields.php')) {
        include_once WC_ALIDROPSHIP_PLUGIN_DIR .
'includes/class-wc-alidropship-product-fields.php';
        if (class_exists('WC_AliDropship_Product_Fields')) {
            new WC_AliDropship_Product_Fields();
        }
    }

    if (file_exists(WC_ALIDROPSHIP_PLUGIN_DIR .
'includes/class-wc-alidropship-order-manager.php')) {
        include_once WC_ALIDROPSHIP_PLUGIN_DIR .
'includes/class-wc-alidropship-order-manager.php';
        if (class_exists('WC_AliDropship_Order_Manager')) {
            new WC_AliDropship_Order_Manager();
        }
    }

    // Admin
    if (is_admin() && file_exists(WC_ALIDROPSHIP_PLUGIN_DIR .
'includes/admin/class-wc-alidropship-admin.php')) {
        include_once WC_ALIDROPSHIP_PLUGIN_DIR .
'includes/admin/class-wc-alidropship-admin.php';
        if (class_exists('WC_AliDropship_Admin')) {
            new WC_AliDropship_Admin();
        }
    }

    /**
     * Cargar traducciones
     */
    public function load_textdomain() {
        load_plugin_textdomain('wc-aliexpress-dropship', false,
dirname(plugin_basename(__FILE__)) . '/languages');
    }

    /**
     * Comprobar requisitos
     */
    public function check_requirements() {

```

```

        // Verificar si WooCommerce está activo
        if (!class_exists('WooCommerce')) {
            add_action('admin_notices', array($this, 'woocommerce_missing_notice'));
        }
    }
}

```

```

/**
 * Aviso de WooCommerce faltante
 */
public function woocommerce_missing_notice() {
    ?>
    <div class="error">
        <p><?php esc_html_e('WooCommerce AliExpress Dropshipping requiere
que WooCommerce esté instalado y activado.', 'wc-aliexpress-dropship'); ?></p>
    </div>
    <?php
}

```

```

/**
 * Añadir enlaces a la página de plugins
 */
public function plugin_action_links($links) {
    $plugin_links = array(
        '<a href="' . admin_url('admin.php?page=wc-alidropship-settings') . '">' .
esc_html__( 'Configuración', 'wc-aliexpress-dropship') . '</a>'
    );
    return array_merge($plugin_links, $links);
}

```

```

/**
 * Activación del plugin
 */
public function activate() {
    // Crear tablas de base de datos necesarias
    $this->create_tables();
}

```

```

    // Guardar versión actual
    update_option('wc_alidropship_version', $this->version);
}

```

```

    // Otras acciones de activación
    do_action('wc_alidropship_activated');
}

```

```

/**

```



```

* Actualizar el plugin
*/
public function update() {
    $current_version = get_option('wc_alidropship_version', '0.0.0');

    if (version_compare($current_version, $this->version, '<')) {
        // Crear tablas
        $this->create_tables();

        // Registrar metadatos
        if (class_exists('\Automattic\WooCommerce\Utilities\FeaturesUtil')) {
            $meta_keys = array(
                '_aliexpress_dropship_processed',
                '_aliexpress_dropship_processed_at'
            );

            foreach ($meta_keys as $meta_key) {
                if (method_exists('\Automattic\WooCommerce\Utilities\FeaturesUtil',
                    'register_order_meta_key_for_hpos')) {

\Automattic\WooCommerce\Utilities\FeaturesUtil::register_order_meta_key_for_hpos
($meta_key);
                }
            }
        }

        // Actualizar versión guardada
        update_option('wc_alidropship_version', $this->version);
    }
}

/**
 * Crear tablas de base de datos
 */
private function create_tables() {
    global $wpdb;

    $wpdb->hide_errors();

    require_once ABSPATH . 'wp-admin/includes/upgrade.php';

    $collate = "";

    if ($wpdb->has_cap('collation')) {

```

```
        $collate = $wpdb->get_charset_collate();
    }
}
```

```
// Tabla para almacenar relaciones de productos
$tables = "
CREATE TABLE {$wpdb->prefix}wc_alidropship_products (
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
    product_id BIGINT UNSIGNED NOT NULL,
    aliexpress_url VARCHAR(500) DEFAULT '',
    aliexpress_item_id VARCHAR(100) DEFAULT '',
    aliexpress_shop_id VARCHAR(100) DEFAULT '',
    aliexpress_price DECIMAL(10,2) DEFAULT 0,
    aliexpress_shipping DECIMAL(10,2) DEFAULT 0,
    last_update DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id),
    UNIQUE KEY product_id (product_id)
) $collate;
";
```

```
dbDelta($tables);
```

```
// Registrar metadatos personalizados para HPOS
if (class_exists('\Automattic\WooCommerce\Utilities\FeaturesUtil')) {
    // Registrar metadatos personalizados para HPOS
    $meta_keys = array(
        '_aliexpress_dropship_processed',
        '_aliexpress_dropship_processed_at'
    );
```

```
    foreach ($meta_keys as $meta_key) {
        if (method_exists('\Automattic\WooCommerce\Utilities\FeaturesUtil',
            'register_order_meta_key_for_hpos')) {
```

```
\Automattic\WooCommerce\Utilities\FeaturesUtil::register_order_meta_key_for_hpos
($meta_key);
    }
}
}
}
```

```
/**
 * Comprobar la configuración del sistema
 */
public function check_system_configuration() {
```

```

        // Verificar si se está utilizando HPOS y si somos compatibles
        if (
            class_exists('\Automattic\WooCommerce\Utilities\OrderUtil') &&
            method_exists('\Automattic\WooCommerce\Utilities\OrderUtil',
'custom_orders_table_usage_is_enabled') &&
            \Automattic\WooCommerce\Utilities\OrderUtil::custom_orders_table_usage_is_enabled()
        ) {
            add_action('admin_notices', function() {
                ?>
                <div class="notice notice-success is-dismissible">
                    <p><?php esc_html_e('WooCommerce AliExpress Dropshipping es compatible con el Almacenamiento de Pedidos de Alto Rendimiento (HPOS) de WooCommerce.', 'wc-aliexpress-dropship'); ?></p>
                </div>
                <?php
            });
        }

```

```

        // Verificar que existan las carpetas necesarias para assets
        if (!file_exists(WC_ALIDROPSHIP_PLUGIN_DIR . 'assets/css/admin.css') || !file_exists(WC_ALIDROPSHIP_PLUGIN_DIR . 'assets/js/order-manager.js')) {
            add_action('admin_notices', function() {
                ?>
                <div class="notice notice-warning is-dismissible">
                    <p><?php esc_html_e('Faltan archivos de recursos en WooCommerce AliExpress Dropshipping. Por favor, verifica que existan las carpetas assets/css y assets/js con sus respectivos archivos.', 'wc-aliexpress-dropship'); ?></p>
                </div>
                <?php
            });
        }
    }
}

```

```

/**
 * Función principal para acceder a la instancia del plugin
 */
function WC_AliExpress_Dropshipping() {
    return WC_AliExpress_Dropshipping::instance();
}

```

```

// Iniciar el plugin

```

```
$GLOBALS['wc_aliepress_dropshipping'] = WC_AliExpress_Dropshipping();
```

```
// Desactivar temporalmente la verificación de archivos  
add_filter('wc_alidropship_check_assets', 'return_false');
```

```
// Crear carpetas y archivos si no existen  
add_action('admin_init', 'wc_alidropship_create_assets');  
function wc_alidropship_create_assets() {  
    // Rutas de directorios  
    $assets_dir = WC_ALIDROPSHIP_PLUGIN_DIR . 'assets';  
    $css_dir = $assets_dir . '/css';  
    $js_dir = $assets_dir . '/js';
```

```
    // Crear directorios si no existen  
    if (!file_exists($assets_dir)) {  
        wp_mkdir_p($assets_dir);  
    }
```

```
    if (!file_exists($css_dir)) {  
        wp_mkdir_p($css_dir);  
    }
```

```
    if (!file_exists($js_dir)) {  
        wp_mkdir_p($js_dir);  
    }
```

```
    // Crear archivos si no existen  
    $css_file = $css_dir . '/admin.css';  
    if (!file_exists($css_file)) {  
        $css_content = "/* Estilos para el panel de administración */  
.aliexpress-customer-data, .aliexpress-order-products { margin-bottom: 15px; }  
.aliexpress-customer-data h4, .aliexpress-order-products h4 { margin-top: 0;  
padding-bottom: 5px; border-bottom: 1px solid #eee; }  
.aliexpress-customer-data table { margin-bottom: 10px; width: 100%; }  
.aliexpress-customer-data table td { padding: 5px; vertical-align: top; }  
.order-products-list { margin: 0; padding: 0; list-style: none; }  
.order-products-list li { margin-bottom: 10px; padding: 10px; background: #f9f9f9;  
border: 1px solid #e5e5e5; border-radius: 3px; }  
.product-name { display: block; font-weight: bold; margin-bottom: 5px; }  
.product-options { margin-bottom: 10px; }  
.product-options span { display: inline-block; margin-right: 10px; font-size: 12px;  
background: #e9e9e9; padding: 2px 5px; border-radius: 3px; }  
.aliexpress-actions { display: flex; align-items: center; justify-content: space-between;  
margin-top: 8px; }
```

```
.aliexpress-id { font-size: 12px; color: #777; }
.aliexpress-missing { margin-top: 5px; }";
```

```
file_put_contents($css_file, $css_content);
}
```

```
$js_file = $js_dir . '/order-manager.js';
if (!file_exists($js_file)) {
    $js_content = "/*
 * JavaScript para la gestión de pedidos
 */
jQuery(document).ready(function($) {
    // Botón para copiar datos del cliente
    $('#copy-customer-data').on('click', function(e) {
        e.preventDefault();
```

```
const orderId = $(this).data('order-id');
const copyStatus = $('#copy-status');
```

```
// Mostrar estado procesando
copyStatus.html('<span style=\"color: #0073aa;\">Procesando...</span>');
```

```
// Hacer la petición AJAX
$.ajax({
    type: 'POST',
    url: wcAlidropship.ajaxUrl,
    data: {
        action: 'aliexpress_copy_customer_data',
        nonce: wcAlidropship.nonce,
        order_id: orderId
    },
    success: function(response) {
        if (response.success && response.data) {
            // Obtener el JSON como string
            const jsonData = JSON.stringify(response.data.data);
            console.log('Datos a copiar:', jsonData);

            // Método 1: API moderna (Chrome, Firefox, Edge)
            if (navigator.clipboard && navigator.clipboard.writeText) {
                navigator.clipboard.writeText(jsonData)
                    .then(function() {
                        copyStatus.html('<span style=\"color: #4CAF50;\">' +
                            wcAlidropship.copySuccess + '</span>');
                        setTimeout(function() {
```

```

        copyStatus.html("");
    }, 3000);
    })
    .catch(function(err) {
        console.error('Error clipboard API:', err);
        // Intentar método 2 como fallback
        copyToClipboardFallback(jsonData);
    });
} else {
    // Método 2: Fallback para navegadores antiguos
    copyToClipboardFallback(jsonData);
}
} else {
    copyStatus.html('<span style=\'color: #F44336;\'>' + (response.data ?
response.data.message : wcAlidropship.copyError) + '</span>');
    setTimeout(function() {
        copyStatus.html("");
    }, 3000);
}
},
error: function() {
    copyStatus.html('<span style=\'color: #F44336;\'>' +
wcAlidropship.copyError + '</span>');
    setTimeout(function() {
        copyStatus.html("");
    }, 3000);
}
});
});

```

```

// Método de fallback para copiar al portapapeles
function copyToClipboardFallback(text) {
    // Crear elemento textarea temporal
    const textarea = document.createElement('textarea');
    textarea.value = text;

    // Hacer que el textarea no sea visible pero sí esté en el DOM
    textarea.style.position = 'fixed';
    textarea.style.top = '0';
    textarea.style.left = '0';
    textarea.style.width = '2em';
    textarea.style.height = '2em';
    textarea.style.padding = '0';
    textarea.style.border = 'none';
}

```

```

        textarea.style.outline = 'none';
        textarea.style.boxShadow = 'none';
        textarea.style.background = 'transparent';

        document.body.appendChild(textarea);
        textarea.focus();
        textarea.select();

        let successful = false;
        try {
            successful = document.execCommand('copy');
        } catch (err) {
            console.error('Error en fallback copy:', err);
        }

        document.body.removeChild(textarea);

        const copyStatus = $('#copy-status');
        if (successful) {
            copyStatus.html('<span style=\\"color: #4CAF50;\\">' +
wcAlidropship.copySuccess + '</span>');
        } else {
            copyStatus.html('<span style=\\"color: #F44336;\\">' + wcAlidropship.copyError
+ '</span>');
        }

        setTimeout(function() {
            copyStatus.html("");
        }, 3000);
    });
});";

    file_put_contents($js_file, $js_content);
}

// Registrar y cargar recursos (CSS y JS)
add_action('admin_enqueue_scripts', 'wc_alidropship_load_admin_resources');

function wc_alidropship_load_admin_resources($hook) {
    // Solo cargar en páginas específicas
    $screen = get_current_screen();
    if (!$screen || !in_array($screen->id, array('product', 'shop_order',
'woocommerce_page_wc-orders'))) {

```

```

    return;
}

// Estilos
if (file_exists(WC_ALIDROPSHIP_PLUGIN_DIR . 'assets/css/admin.css')) {
    wp_enqueue_style(
        'wc-alidropship-admin',
        WC_ALIDROPSHIP_PLUGIN_URL . 'assets/css/admin.css',
        array(),
        WC_ALIDROPSHIP_VERSION
    );
}

// Scripts
if (file_exists(WC_ALIDROPSHIP_PLUGIN_DIR . 'assets/js/order-manager.js')) {
    wp_enqueue_script(
        'wc-alidropship-admin',
        WC_ALIDROPSHIP_PLUGIN_URL . 'assets/js/order-manager.js',
        array('jquery'),
        WC_ALIDROPSHIP_VERSION,
        true
    );
}

// Localizar script
wp_localize_script('wc-alidropship-admin', 'wcAlidropship', array(
    'ajaxUrl' => admin_url('admin-ajax.php'),
    'nonce' => wp_create_nonce('wc_alidropship_nonce'),
    'copySuccess' => __('Datos copiados', 'wc-aliexpress-dropship'),
    'copyError' => __('Error al copiar', 'wc-aliexpress-dropship')
));
}
}

```

la extension se divide en
content.js

```

/**
 * Content Script optimizado para las páginas de AliExpress
 * Se enfoca en detectar y completar formularios de dirección
 * Versión 3.0 - Mejorada para formularios específicos con clases numeradas
 */

```

```

// Escuchar mensajes del popup
chrome.runtime.onMessage.addListener(function(request, sender, sendResponse) {

```



```

if (request.action === 'fillAliExpressForm') {
  try {
    console.log('Iniciando autocompletado con datos:', request.data);

    // Usar un timeout para asegurar que todos los elementos están disponibles
    setTimeout(() => {
      // Detectar y completar el formulario adecuado
      if (document.querySelector('.deliver-address-wrap')) {
        console.log('Detectado formulario de dirección AliExpress');
        const result = fillAliExpressForm(request.data);
        sendResponse(result);
      } else {
        console.warn('No se detectó un formulario de dirección compatible');
        sendResponse({
          success: false,
          message: 'No se detectó un formulario de dirección compatible en esta
página'
        });
      }
    }, 300);
  } catch (error) {
    console.error('Error en autocompletado:', error);
    sendResponse({success: false, message: 'Error: ' + error.message});
  }

  return true; // Mantener conexión abierta para respuesta asíncrona
}
});

```

```

// Función principal para completar formulario
function fillAliExpressForm(data) {
  try {
    // Verificar el tipo de formulario que tenemos
    const isNumericForm = document.querySelector('.one-textinput-placeholder-1')
    !== null;

    if (isNumericForm) {
      return fillNumericFormAliExpress(data);
    } else {
      return fillGenericAliExpressForm(data);
    }
  } catch (error) {
    console.error('Error detectando tipo de formulario:', error);
  }
}

```

```
    return {success: false, message: 'Error al detectar tipo de formulario: ' +  
error.message};  
  }  
}
```

// Función especializada para formularios con clases numéricas (el que has
compartido)

```
function fillNumericFormAliExpress(data) {  
  console.log('Usando método para formulario con clases numéricas');
```

```
  // Resultados
```

```
  const completedFields = [];
```

```
  const failedFields = [];
```

```
  try {
```

```
    // 1. Campo de nombre (no modificar país que es placeholder-0)
```

```
    const nameField = document.querySelector('.one-textinput-placeholder-1');
```

```
    if (nameField && data.firstName) {
```

```
      if (fillInputField(nameField, data.firstName)) {
```

```
        completedFields.push('Nombre');
```

```
        console.log('✓ Campo nombre completado: ' + data.firstName);
```

```
      } else {
```

```
        failedFields.push('Nombre');
```

```
        console.warn('✗ No se pudo completar el campo nombre');
```

```
    }
```

```
  }
```

```
  // 2. Campo de apellido
```

```
  const lastNameField = document.querySelector('.one-textinput-placeholder-2');
```

```
  if (lastNameField && data.lastName) {
```

```
    if (fillInputField(lastNameField, data.lastName)) {
```

```
      completedFields.push('Apellido');
```

```
      console.log('✓ Campo apellido completado: ' + data.lastName);
```

```
    } else {
```

```
      failedFields.push('Apellido');
```

```
      console.warn('✗ No se pudo completar el campo apellido');
```

```
    }
```

```
  }
```

```
  // Nota: No modificamos el campo de prefijo telefónico (placeholder-3)
```

```
  // 3. Campo de teléfono
```

```
  const phoneField = document.querySelector('.one-textinput-placeholder-4');
```

```
  if (phoneField && data.phoneNumber) {
```

```
    if (fillInputField(phoneField, data.phoneNumber)) {
```

```

    completedFields.push('Teléfono');
    console.log('✓ Campo teléfono completado: ' + data.phoneNumber);
  } else {
    failedFields.push('Teléfono');
    console.warn('✗ No se pudo completar el campo teléfono');
  }
}
}

```

```

// 4. Campo de dirección (tipo especial)
const addressField = document.querySelector('.autocomplete-input');
if (addressField && data.streetAddress) {
  if (fillInputField(addressField, data.streetAddress)) {
    completedFields.push('Dirección');
    console.log('✓ Campo dirección completado: ' + data.streetAddress);
  } else {
    failedFields.push('Dirección');
    console.warn('✗ No se pudo completar el campo dirección');
  }
}
}

```

```

// 5. Campo de complemento/apartamento
const aptField = document.querySelector('.one-textinput-placeholder-5');
if (aptField && data.apartment) {
  if (fillInputField(aptField, data.apartment)) {
    completedFields.push('Complemento');
    console.log('✓ Campo complemento completado: ' + data.apartment);
  } else {
    failedFields.push('Complemento');
    console.warn('✗ No se pudo completar el campo complemento');
  }
}
}

```

// Nota: No modificamos los campos de provincia y municipio (placeholder-6 y placeholder-7)

```

// 6. Campo de código postal
const zipField = document.querySelector('.one-textinput-placeholder-8');
if (zipField && data.zipCode) {
  if (fillInputField(zipField, data.zipCode)) {
    completedFields.push('Código postal');
    console.log('✓ Campo código postal completado: ' + data.zipCode);
  } else {
    failedFields.push('Código postal');
    console.warn('✗ No se pudo completar el campo código postal');
  }
}
}

```

```
}  
}
```

```
// Intento especial para el campo de dirección que a veces es problemático  
if (addressField && data.streetAddress && !completedFields.includes('Dirección'))  
{  
  console.log('Intentando método alternativo para el campo de dirección...');  
  setTimeout(() => {  
    try {  
      fillInputFieldAggressive(addressField, data.streetAddress);  
    } catch (e) {  
      console.warn('Error en método alternativo para dirección:', e);  
    }  
  }, 500);  
}
```

```
// Generar mensaje con resultado  
let message = "";  
if (completedFields.length > 0) {  
  message = `Se completaron ${completedFields.length} campos:  
${completedFields.join(', ')}. `;  
}  
if (failedFields.length > 0) {  
  message += `No se pudieron completar ${failedFields.length} campos:  
${failedFields.join(', ')}. `;  
}
```

```
message += 'Por favor verifica y completa manualmente los campos faltantes.';
```

```
// Mostrar notificación  
showCompletionNotification(message, completedFields.length > 0);
```

```
return {  
  success: completedFields.length > 0,  
  message: message,  
  completedFields: completedFields,  
  failedFields: failedFields  
};  
} catch (error) {  
  console.error('Error en el proceso de autocompletado:', error);  
  showCompletionNotification('Error al completar el formulario: ' + error.message,  
false);
```

```
return {
```

```

    success: false,
    message: 'Error: ' + error.message
  };
}
}
}

```

```

// Función para formularios genéricos (como respaldo para otras páginas)
function fillGenericAliExpressForm(data) {
  console.log('Usando método para formulario genérico');

```

```

  const completedFields = [];
  const failedFields = [];

```

```

  try {
    // Buscar campos por placeholders y atributos comunes

```

```

    // 1. Campo de nombre
    const nameSelectors = [
      'input[placeholder*="Nombre"]',
      'input[placeholder*="nombre"]',
      'input[name*="firstName"]',
      'input[id*="firstName"]'
    ];

```

```

    let nameField = findFieldBySelectors(nameSelectors);
    if (nameField && data.firstName) {
      if (fillInputField(nameField, data.firstName)) {
        completedFields.push('Nombre');
      } else {
        failedFields.push('Nombre');
      }
    } else if (data.firstName) {
      failedFields.push('Nombre');
    }

```

```

    // 2. Campo de apellido
    const lastNameSelectors = [
      'input[placeholder*="Apellido"]',
      'input[placeholder*="apellido"]',
      'input[name*="lastName"]',
      'input[id*="lastName"]'
    ];

```

```

    let lastNameField = findFieldBySelectors(lastNameSelectors);

```

```

    if (lastNameField && data.lastName) {
      if (fillInputField(lastNameField, data.lastName)) {
        completedFields.push('Apellido');
      } else {
        failedFields.push('Apellido');
      }
    } else if (data.lastName) {
      failedFields.push('Apellido');
    }
  }
}

```

```

// 3. Campo de teléfono
const phoneSelectors = [
  'input[placeholder*="Teléfono"]',
  'input[placeholder*="teléfono"]',
  'input[placeholder*="móvil"]',
  'input[name*="phone"]',
  'input[id*="phone"]'
];

```

```

let phoneField = findFieldBySelectors(phoneSelectors);
if (phoneField && data.phoneNumber) {
  if (fillInputField(phoneField, data.phoneNumber)) {
    completedFields.push('Teléfono');
  } else {
    failedFields.push('Teléfono');
  }
} else if (data.phoneNumber) {
  failedFields.push('Teléfono');
}

```

```

// 4. Campo de dirección
const addressSelectors = [
  'input[placeholder*="Calle"]',
  'input[placeholder*="calle"]',
  'input[placeholder*="dirección"]',
  'input[placeholder*="Dirección"]',
  '.autocomplete-input',
  'input[role="combobox"]'
];

```

```

let addressField = findFieldBySelectors(addressSelectors);
if (addressField && data.streetAddress) {
  if (fillInputField(addressField, data.streetAddress)) {
    completedFields.push('Dirección');
  }
}

```

```

    } else {
        failedFields.push('Dirección');
    }
    } else if (data.streetAddress) {
        failedFields.push('Dirección');
    }
}

```

```

// 5. Campo de código postal
const zipSelectors = [
    'input[placeholder*="Código postal"]',
    'input[placeholder*="código postal"]',
    'input[placeholder*="CP"]',
    'input[name*="zip"]',
    'input[id*="zip"]'
];

```

```

let zipField = findFieldBySelectors(zipSelectors);
if (zipField && data.zipCode) {
    if (fillInputField(zipField, data.zipCode)) {
        completedFields.push('Código postal');
    } else {
        failedFields.push('Código postal');
    }
} else if (data.zipCode) {
    failedFields.push('Código postal');
}

```

```

// Generar mensaje con resultado
let message = "";
if (completedFields.length > 0) {
    message = `Se completaron ${completedFields.length} campos:
    ${completedFields.join(', ')}. `;
}
if (failedFields.length > 0) {
    message += `No se pudieron completar ${failedFields.length} campos:
    ${failedFields.join(', ')}. `;
}

```

```

message += 'Por favor verifica y completa manualmente los campos faltantes.';

```

```

// Mostrar notificación
showCompletionNotification(message, completedFields.length > 0);

```

```

return {

```

```
    success: completedFields.length > 0,  
    message: message,  
    completedFields: completedFields,  
    failedFields: failedFields  
  };  
} catch (error) {  
  console.error('Error en el proceso de autocompletado genérico:', error);  
  showCompletionNotification('Error al completar el formulario: ' + error.message,  
false);
```

```
  return {  
    success: false,  
    message: 'Error: ' + error.message  
  };  
}  
}
```

```
// Función para rellenar un campo de entrada con un valor
```

```
function fillInputField(element, value) {  
  if (!element || value === undefined || value === null) return false;
```

```
  try {  
    // Asegurarnos de que el elemento está visible  
    if (!isVisibleElement(element)) {  
      console.warn('El elemento no es visible:', element);  
      return false;  
    }
```

```
    // 1. Dar foco al elemento y activarlo  
    element.click();  
    element.focus();
```

```
    // 2. Limpiar el valor actual  
    const originalValue = element.value;  
    element.value = '';
```

```
    // 3. Disparar eventos para notificar que el campo está vacío  
    element.dispatchEvent(new Event('input', { bubbles: true }));  
    element.dispatchEvent(new Event('change', { bubbles: true }));
```

```
    // Pequeña pausa entre limpiar y escribir  
    setTimeout(() => {  
      // 4. Asignar el nuevo valor  
      element.value = value;
```



```
// 5. Disparar eventos para notificar el cambio
element.dispatchEvent(new Event('input', { bubbles: true }));
element.dispatchEvent(new Event('change', { bubbles: true }));
element.dispatchEvent(new Event('blur', { bubbles: true }));
```

```
// 6. Manejar placeholder si existe
handlePlaceholder(element);
}, 50);
```

```
return true;
} catch (error) {
  console.error('Error al rellenar campo:', error);
```

```
// Método de emergencia: asignar directamente
try {
  element.value = value;
  element.dispatchEvent(new Event('input', { bubbles: true }));
  element.dispatchEvent(new Event('change', { bubbles: true }));
  return true;
} catch (e) {
  return false;
}
}
```

```
// Función más agresiva para campos problemáticos
```

```
function fillInputFieldAggressive(element, value) {
  if (!element || !value) return false;
```

```
  try {
    // Método 1: Utilizar el descriptor del prototipo
    const descriptor =
      Object.getOwnPropertyDescriptor(HTMLInputElement.prototype, 'value');
    if (descriptor && descriptor.set) {
      descriptor.set.call(element, value);
    }
  }
```

```
  // Método 2: Usar definición directa
  Object.defineProperty(element, 'value', {
    value: value,
    writable: true
  });
```

```
  // Método 3: Eventos de teclado simulados
```

```
element.focus();
```

```
// Limpiar primero
```

```
for (let i = 0; i < element.value.length; i++) {  
  element.dispatchEvent(new KeyboardEvent('keydown', { key: 'Backspace',  
keyCode: 8 }));  
  element.dispatchEvent(new KeyboardEvent('keyup', { key: 'Backspace',  
keyCode: 8 }));  
}
```

```
// Escribir nuevo valor
```

```
for (let i = 0; i < value.length; i++) {  
  const char = value.charAt(i);
```

```
  element.dispatchEvent(new KeyboardEvent('keydown', { key: char }));  
  element.value = value.substring(0, i + 1);  
  element.dispatchEvent(new KeyboardEvent('keyup', { key: char }));  
}
```

```
// Disparar eventos finales
```

```
element.dispatchEvent(new Event('input', { bubbles: true }));  
element.dispatchEvent(new Event('change', { bubbles: true }));  
element.dispatchEvent(new Event('blur', { bubbles: true }));
```

```
return true;
```

```
} catch (error) {
```

```
  console.error('Error en método agresivo:', error);
```

```
  return false;
```

```
}  
}
```

```
// Función para manejar el placeholder de un campo
```

```
function handlePlaceholder(element) {
```

```
  try {
```

```
    // Buscar placeholder en diferentes ubicaciones
```

```
    // 1. En el elemento padre
```

```
    let parent = element.parentElement;
```

```
    if (parent) {
```

```
      const placeholder = parent.querySelector('.placeholder-puppt');
```

```
      if (placeholder && element.value) {
```

```
        placeholder.classList.add('placeholder-small');
```

```
        return true;
```

```
      }  
    }
```

```

// 2. Buscar en abuelo
parent = parent.parentElement;
if (parent) {
    const placeholder = parent.querySelector('.placeholder-puppt');
    if (placeholder && element.value) {
        placeholder.classList.add('placeholder-small');
        return true;
    }
}
}
}

```

```

// 3. Buscar como hermano
const siblings = Array.from(element.parentElement.children);
for (const sibling of siblings) {
    if (sibling.classList && sibling.classList.contains('placeholder-puppt')) {
        if (element.value) {
            sibling.classList.add('placeholder-small');
            return true;
        }
    }
}
}
}

```

```

return false;
} catch (error) {
    console.warn('Error al manejar placeholder:', error);
    return false;
}
}
}

```

```

// Función para encontrar un campo por múltiples selectores
function findFieldBySelectors(selectors) {
    for (const selector of selectors) {
        const element = document.querySelector(selector);
        if (element && isVisibleElement(element)) {
            return element;
        }
    }
    return null;
}
}

```

```

// Función para verificar si un elemento es visible
function isVisibleElement(element) {
    if (!element) return false;

```

```
const style = window.getComputedStyle(element);
return style.display !== 'none' &&
    style.visibility !== 'hidden' &&
    style.opacity !== '0' &&
    element.offsetParent !== null;
}
```

```
// Función auxiliar para esperar
function sleep(ms) {
    return new Promise(resolve => setTimeout(resolve, ms));
}
```

```
// Función para mostrar notificación sobre el resultado
function showCompletionNotification(message, isSuccess) {
    // Eliminar notificación anterior si existe
    const existingNotification = document.getElementById('aliexpress-notification');
    if (existingNotification) {
        existingNotification.remove();
    }
}
```

```
// Crear elemento de notificación
const notification = document.createElement('div');
notification.id = 'aliexpress-notification';
notification.textContent = message;
```

```
// Aplicar estilos según el resultado
notification.style.cssText = `
    position: fixed;
    top: 70px;
    left: 50%;
    transform: translateX(-50%);
    background-color: ${isSuccess ? '#4CAF50' : '#F44336'};
    color: white;
    padding: 12px 20px;
    border-radius: 4px;
    font-family: Arial, sans-serif;
    font-size: 14px;
    font-weight: bold;
    box-shadow: 0 2px 8px rgba(0,0,0,0.3);
    z-index: 2147483647;
    max-width: 80%;
    text-align: center;
    opacity: 0;
    transition: opacity 0.3s ease;
`;
```

```
`;
```

```
// Añadir al DOM
```

```
document.body.appendChild(notification);
```

```
// Animar entrada
```

```
setTimeout(() => {
```

```
  notification.style.opacity = '1';
```

```
}, 10);
```

```
// Ocultar después de 8 segundos
```

```
setTimeout(() => {
```

```
  notification.style.opacity = '0';
```

```
  setTimeout(() => {
```

```
    notification.remove();
```

```
  }, 300);
```

```
}, 8000);
```

```
}
```

```
// Crear botón flotante para autocompletar
```

```
function createAutofillButton() {
```

```
  // Verificar si ya existe
```

```
  if (document.getElementById('aliexpress-autofill-button')) {
```

```
    return;
```

```
  }
```

```
// Crear elemento de botón
```

```
const button = document.createElement('button');
```

```
button.id = 'aliexpress-autofill-button';
```

```
button.textContent = 'Autocompletar Datos';
```

```
// Estilos del botón
```

```
button.style.cssText = `
```

```
  position: fixed;
```

```
  top: 20px;
```

```
  right: 20px;
```

```
  background-color: #ff6600;
```

```
  color: white;
```

```
  padding: 10px 15px;
```

```
  border: none;
```

```
  border-radius: 4px;
```

```
  font-family: Arial, sans-serif;
```

```
  font-size: 14px;
```

```
  font-weight: bold;
```

```
cursor: pointer;
z-index: 2147483647;
box-shadow: 0 2px 8px rgba(0,0,0,0.3);
transition: all 0.2s;
```

```
// Estilos de hover
```

```
button.addEventListener('mouseover', () => {
  button.style.backgroundColor = '#e55c00';
});
```

```
button.addEventListener('mouseout', () => {
  button.style.backgroundColor = '#ff6600';
});
```

```
// Evento de clic
```

```
button.addEventListener('click', () => {
  chrome.storage.local.get('lastCustomerData', result => {
    if (result.lastCustomerData) {
      console.log('Ejecutando autocompletado con datos almacenados:',
result.lastCustomerData);
      fillAliExpressForm(result.lastCustomerData);
    } else {
      showCompletionNotification('No hay datos de cliente almacenados. Copia los
datos desde WooCommerce primero.', false);
    }
  });
});
```

```
// Añadir al DOM
```

```
document.body.appendChild(button);
}
```

```
// Inicializar la extensión
```

```
function initializeExtension() {
  console.log('Inicializando extensión AliExpress Autocompletador...');
```

```
// Verificar si estamos en AliExpress
```

```
if (window.location.href.includes('aliexpress.com')) {
  console.log('Sitio AliExpress detectado');
```

```
// Crear botón flotante
```

```
createAutofillButton();
```

```

// Intentar solucionar campos problemáticos cada cierto tiempo
setInterval(() => {
  troubleshootProblematicFields();
}, 5000);
}
}

```

```

// Función para intentar arreglar campos problemáticos
function troubleshootProblematicFields() {
  // Intentar solucionar el problema del campo de teléfono si está presente y vacío
  const phoneField = document.querySelector('.one-textinput-placeholder-4');
  if (phoneField && phoneField.value === "") {
    // Cargar datos almacenados y completar solo el teléfono si está vacío
    chrome.storage.local.get('lastCustomerData', result => {
      if (result.lastCustomerData && result.lastCustomerData.phoneNumber) {
        fillInputFieldAggressive(phoneField, result.lastCustomerData.phoneNumber);
      }
    });
  }
}

```

```

// Ejecutar cuando el DOM esté listo
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', initializeExtension);
} else {
  initializeExtension();
}

```

```

// Observar cambios en el DOM para detectar formularios cargados dinámicamente
const observer = new MutationObserver((mutations) => {
  for (const mutation of mutations) {
    if (mutation.addedNodes.length) {
      for (const node of mutation.addedNodes) {
        if (node.nodeType === Node.ELEMENT_NODE) {
          // Si se añade un formulario de dirección
          if (node.querySelector && (
            node.querySelector('.deliver-address-wrap') ||
            node.classList && node.classList.contains('deliver-address-wrap')
          )) {
            console.log('Formulario de dirección detectado dinámicamente');
            initializeExtension();
            return;
          }
        }
      }
    }
  }
}

```

```
}  
}  
}  
});
```

```
observer.observe(document.body, { childList: true, subtree: true });
```

```
// También ejecutar periódicamente para manejar cambios dinámicos en la página  
setInterval(() => {  
  if (!document.getElementById('aliexpress-autofill-button') &&  
      window.location.href.includes('aliexpress.com')) {  
    createAutofillButton();  
  }  
, 2000);
```

```
popup.js  
document.addEventListener('DOMContentLoaded', function() {  
  const fillFormButton = document.getElementById('fillFormButton');  
  const pasteFromClipboardButton =  
document.getElementById('pasteFromClipboardButton');  
  const customerDataInput = document.getElementById('customerDataInput');  
  const statusMessage = document.getElementById('statusMessage');
```

```
// Función para mostrar mensajes de estado  
function showStatus(message, type) {  
  statusMessage.textContent = message;  
  statusMessage.className = 'status ' + type;  
  statusMessage.style.display = 'block';
```

```
  setTimeout(() => {  
    statusMessage.style.display = 'none';  
  }, 3000);  
}
```

```
// Botón para pegar desde el portapapeles  
pasteFromClipboardButton.addEventListener('click', function() {  
  navigator.clipboard.readText()  
    .then(text => {  
      try {  
        // Intentar formatear como JSON si es posible  
        const data = JSON.parse(text);  
        customerDataInput.value = JSON.stringify(data, null, 2);  
      } catch (e) {  
        // Si no es JSON válido, usar el texto tal cual
```



```

        customerDataInput.value = text;
    }
    showStatus('Datos pegados correctamente', 'success');
})
.catch(err => {
    showStatus('Error al leer el portapapeles: ' + err.message, 'error');
});
});

```

```

// Botón para completar el formulario
fillFormButton.addEventListener('click', function() {
    const customerDataText = customerDataInput.value.trim();

```

```

    if (!customerDataText) {
        showStatus('Por favor, ingresa los datos del cliente', 'error');
        return;
    }

```

```

    let customerData;
    try {
        customerData = JSON.parse(customerDataText);
    } catch (e) {
        showStatus('Error al analizar los datos JSON', 'error');
        return;
    }

```

```

    // Almacenar los datos para uso futuro
    chrome.storage.local.set({
        lastCustomerData: customerData
    });

```

```

    // Enviar datos al content script
    chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {
        chrome.tabs.sendMessage(tabs[0].id, {
            action: 'fillAliExpressForm',
            data: customerData
        }, function(response) {
            if (response && response.success) {
                showStatus('¡Formulario completado correctamente!', 'success');
            } else {
                showStatus(response ? response.message : 'Error al comunicarse con la
                página. Asegúrate de estar en AliExpress.', 'error');
            }
        });
    });

```

```
});  
});
```

```
// Cargar los últimos datos utilizados al abrir  
chrome.storage.local.get('lastCustomerData', function(result) {  
  if (result.lastCustomerData) {  
    customerDataInput.value = JSON.stringify(result.lastCustomerData, null, 2);  
  }  
});
```

```
// Al abrir el popup, intentar leer del portapapeles automáticamente  
navigator.clipboard.readText()  
  .then(text => {  
    try {  
      // Verificar si el texto es un JSON válido con la estructura esperada  
      const data = JSON.parse(text);  
      if (data.firstName && data.lastName) {  
        customerDataInput.value = JSON.stringify(data, null, 2);  
        showStatus('Datos detectados y cargados automáticamente', 'success');  
      }  
    } catch (e) {  
      // No es un JSON válido, no hacer nada  
    }  
  })  
  .catch(err => {  
    // Error al leer el portapapeles, no hacer nada  
  });  
});
```

manifest.json

```
{  
  "manifest_version": 3,  
  "name": "AliExpress Autocompletador para Dropshipping",  
  "version": "1.0",  
  "description": "Autocompleta los formularios de AliExpress con datos de clientes de  
WooCommerce",  
  "permissions": [  
    "activeTab",  
    "storage",  
    "clipboardRead"  
  ],  
  "action": {  
    "default_popup": "popup.html"  
  },  
  "content_scripts": [  
    {  
      "matches": ["*://*/*"],  
      "js": ["script.js"]  
    }  
  ]  
}
```

```
{
  "matches": ["*://*.aliexpress.com/*"],
  "js": ["content.js"]
}
]
}
popup.html
```

Resumen de cómo recoge AliExpress los datos

1. País/región

- `input` dentro de un `` (valor: "Spain", código país "ES").

2. Información personal

- Nombre: `<input placeholder="Nombre*">`
- Apellido: `<input placeholder="Apellido*">`
- Teléfono:
 - Prefijo: `<input value="+34">`
 - Número: `<input placeholder="Teléfono móvil*">`

3. Dirección

- `<input placeholder="Buscar por código postal, calle, o dirección">`
- Luego un botón para "Rellenar manualmente" permite el input de dirección, calle, código postal, etc., en campos adicionales (que aparecen al hacer clic, normalmente).

4. Check para dirección predeterminada

5. Botones Confirmar / Cancelar

estos son todos los elementos de aliexpress

```
<input placeholder="Calle, casa/apartamento*" height="100%" autocomplete="off" value="" data-spm-anchor-id="a2g0o.placeorder.0.i20.9b367851cCH7dO">
```

```
<input placeholder="Apto., suite, unidad, etc. (opcional)" height="100%" autocomplete="off" value="" data-spm-anchor-id="a2g0o.placeorder.0.i18.9b367851cCH7dO">
```

estos son todos los campos iniciales de aliexpress

```
<form role="form" class="next-form next-medium deliver-address-form" data-spm-anchor-id="a2g0o.placeorder.0.i2.9b367851cCH7dO"><div class="form-section "><div class="section-title ">País/región</div><div
```

```
class="section-body"><div dir="ltr" role="row" class="next-row form-row-wrap "
style="margin-left: -8px; margin-right: -8px;"
data-spm-anchor-id="a2g0o.placeorder.0.i3.9b367851cCH7dO"><div dir="ltr"
role="gridcell" class="next-col next-col-8"><div class="next-form-item next-top
has-success next-medium default-select-wrap"><div
class="next-form-item-control"><span class="next-select next-select-trigger
next-select-single next-medium next-inactive next-has-search"
aria-haspopup="true"><span class="next-input next-medium
next-select-inner"><label class="next-input-label"><i class="sdk-country-flag
country-flag-y2023 ES"></i></label><span class="next-select-values
next-input-text-field"><em title="Spain">Spain</em><span
class="next-select-trigger-search"><input aria-label="select" role="combobox"
tabindex="0" aria-expanded="false" aria-disabled="false" height="100%" size="1"
autocomplete="off" value=""><span aria-hidden="true"><span></span><span
style="display: inline-block; width:
1px;">&nbsp;</span></span></span></span><span
class="next-input-control"><span aria-hidden="true" class="next-select-arrow"><i
class="next-icon next-icon-arrow-down next-medium
next-select-symbol-fold"></i></span></span></span><span class="next-sr-only"
aria-live="polite"></span></span> <div class="next-form-item-help"></div>
</div></div></div></div></div></div><div class="form-section "><div
class="section-title "
data-spm-anchor-id="a2g0o.placeorder.0.i4.9b367851cCH7dO">Información
personal</div><div class="section-body"><div dir="ltr" role="row" class="next-row
form-row-wrap " style="margin-left: -8px; margin-right: -8px;"><div dir="ltr"
role="gridcell" class="next-col"><div class="next-form-item next-top has-success
next-medium default-input-wrap"><div class="next-form-item-control"><span
class="next-input next-medium"><input placeholder="Nombre*" height="100%"
autocomplete="off" value=""></span> <div class="next-form-item-help"></div>
</div></div></div><div dir="ltr" role="gridcell" class="next-col"><div
class="next-form-item next-top has-success next-medium default-input-wrap"><div
class="next-form-item-control"><span class="next-input next-medium"><input
placeholder="Apellido*" height="100%" autocomplete="off" value=""></span> <div
class="next-form-item-help"></div> </div></div></div><div dir="ltr" role="gridcell"
class="next-col"><div class="next-form-item next-top has-success next-medium
combine-input-wrap"><div class="next-form-item-control"><span
class="next-input-group"><span class="next-input-group-addon next-before"><span
class="next-input next-medium pre-part "><input height="100%" autocomplete="off"
value="+34"></span></span><span class="next-input next-medium main-part
"><input placeholder="Teléfono móvil*" height="100%" autocomplete="off"
value=""></span></span> <div class="next-form-item-help"><span
class="help-inner _tip-wrap"><span>Tu número de teléfono estará
seguro</span>&nbsp;</span></div> </div></div></div></div><div dir="ltr" role="row"
class="next-row form-row-wrap " style="margin-left: -8px; margin-right:
```

```

-8px;"></div></div></div><div class="form-section "><div class="section-title
">Dirección</div><div class="section-body"><div dir="ltr" role="row" class="next-row
form-row-wrap " style="margin-left: -8px; margin-right: -8px;"><div dir="ltr"
role="gridcell" class="next-col next-col-12"><div class="next-form-item next-top
has-success next-medium search-input-wrap"><div
class="next-form-item-control"><span class="next-select next-select-auto-complete
next-size-medium" aria-haspopup="true"><span class="next-input
next-medium"><span class="next-input-inner next-before"><i class="next-icon
next-icon-search next-medium"></i></span><input role="combobox"
aria-autocomplete="list" aria-disabled="false" aria-expanded="false"
placeholder="Buscar por código postal, calle, o dirección" height="100%"
autocomplete="off" value=""></span><span class="next-sr-only"
aria-live="polite"></span></span> <div class="next-form-item-help"></div>
</div></div></div><div dir="ltr" role="gridcell" class="next-col next-col-12"><div
class="next-form-item next-medium"><div class="next-form-item-control"><div
class="text-button-wrap" style="margin-top: 15px;"><div
class="text-button-container">Rellenar manualmente</div><span type="arrow-down"
class="comet-icon comet-icon-arrowdown "><svg viewBox="0 0 1024 1024"
width="1em" height="1em" fill="currentColor" aria-hidden="false"
focusable="false"><path d="M97.6 308.032a35.925333 35.925333 0 0 0-4.128
49.813333l1.408 1.632 355.232 371.914667a85.333333 85.333333 0 0 0
123.381333 0.032l355.626667-371.946667a35.936 35.936 0 0
0-2.730667-51.445333 37.674667 37.674667 0 0 0-50.944 1.130667l-1.504
1.546667L527.253333 674.986667a21.333333 21.333333 0 0 1-30.922666
0L150.058667 310.698667a37.653333 37.653333 0 0
0-52.448-2.666667z"></path></svg></span></div> </div></div></div><div
dir="ltr" role="row" class="next-row form-row-wrap " style="margin-left: -8px;
margin-right: -8px;"></div><div dir="ltr" role="row" class="next-row form-row-wrap "
style="margin-left: -8px; margin-right: -8px;"></div></div><div
class="form-section "><div class="section-body"><div dir="ltr" role="row"
class="next-row form-row-wrap " style="margin-left: -8px; margin-right: -8px;"><div
dir="ltr" role="gridcell" class="next-col"><div class="next-form-item has-success
next-medium checkbox-wrap "><div class="next-form-item-control"><div
class="check-box-wrap" style="display: inline-block;"><label
class="next-checkbox-wrapper"><span class="next-checkbox"><span
class="next-checkbox-inner"><i class="next-icon next-icon-select next-xs
next-checkbox-select-icon"></i></span><input type="checkbox"
aria-checked="false" class="next-checkbox-input" value=""></span><span
class="next-checkbox-label"><span>Establecer como dirección de envío
predeterminada</span></span></label></div> <div
class="next-form-item-help"></div> </div></div></div></div></div><div
class="form-section "><div class="section-body"><button type="button"
class="comet-btn comet-btn-primary comet-btn-large
form-item-button"><span>Confirmar</span></button><button type="button"

```

```
class="comet-btn comet-btn-normal comet-btn-large  
form-item-button"><span>Cancelar</span></button></div></div></form>
```

Estos son los campos que se despliegan en adición cuando le doy a rellenar
manualmente al boton <input role="combobox" aria-autocomplete="list"
aria-disabled="false" aria-expanded="false" placeholder="Buscar por código postal,
calle, o dirección" height="100%" autocomplete="off" value="">

```
<input placeholder="Calle, casa/apartamento*" height="100%" autocomplete="off"  
value="" data-spm-anchor-id="a2g0o.placeorder.0.i20.9b367851cCH7dO">
```

```
<input placeholder="Apto., suite, unidad, etc. (opcional)" height="100%"  
autocomplete="off" value=""  
data-spm-anchor-id="a2g0o.placeorder.0.i18.9b367851cCH7dO">  
<span class="next-select-values next-input-text-field"><em title="A Coruna">A  
Coruna</em><span class="next-select-trigger-search"><input aria-label="select"  
role="combobox" tabindex="0" aria-expanded="false" aria-disabled="false"  
height="100%" size="1" autocomplete="off" value=""><span  
aria-hidden="true"><span></span><span style="display: inline-block; width:  
1px;">&nbsp;</span></span></span></span>
```

```
<span class="next-select-values next-input-text-field"><em title="A Baña">A  
Baña</em><span class="next-select-trigger-search"><input aria-label="select"  
role="combobox" tabindex="0" aria-expanded="false" aria-disabled="false"  
height="100%" size="1" autocomplete="off" value=""><span  
aria-hidden="true"><span></span><span style="display: inline-block; width:  
1px;">&nbsp;</span></span></span></span></span>  
<input placeholder="Código postal*" height="100%" autocomplete="off" value="">
```

para que entiendas, en el plugin que recoge la info en mi página web para copiar la
recoje así, esto es solo un ejemplo no quiero que guardes la info,

```
<input type="hidden" id="aliexpress-customer-json" value="{  
  &quot;firstName&quot;: &quot;Francileny&quot;,,  
  &quot;lastName&quot;: &quot;Ramirez parra&quot;,,  
  &quot;phonePrefix&quot;: &quot;+34&quot;,,  
  &quot;phoneNumber&quot;: &quot;676941280&quot;,,  
  &quot;streetAddress&quot;: &quot;Av blas infante 8&quot;,,  
  &quot;apartment&quot;: &quot;1-4&quot;,,  
  &quot;zipCode&quot;: &quot;29631&quot;,,  
  &quot;city&quot;: &quot;Benalmadena&quot;,,  
  &quot;state&quot;: &quot;MA&quot;,,  
  &quot;country&quot;: &quot;ES&quot;,,
```

"email": "franci_col@hotmail.com"
}">

El problema esta en que no autocompleta en aliexpress el apt,suite,unidad o la informacion sobrante que no se haya pegado es todo lo que no coincida con

Nombre

Apellido

Teléfono móvil

Calle, casa/apartamento*

ciudad

provincia

codigo postal

ahora mismo mi implementacion hace bien en aliexpress el nombre, apellido, el pais, el numero de telefono

quiero cambiar que ponga la info de mi pagina web que recoge cuando copio en portapapeles y pego en la extension la ponga como este orden acorde al formato aliexpress para cuando lo autocomplete en aliexpress.

"firstName": "Francileny",,
"lastName": "Ramirez parra",,
"phonePrefix": "+34",,
"phoneNumber": "676941280",,
codigo postal, calle/ direccion , ciudad, ciudad/ provincia/ pueblo

he pensado dividir el pegado en 2 botones, primer boton seria darle a autocompletar en aliexpress la info relacionada con estos campos

"firstName": "Francileny",,
"lastName": "Ramirez parra",,
"phonePrefix": "+34",,
"phoneNumber": "676941280",,
codigo postal, calle/ direccion , ciudad, ciudad/ provincia/ pueblo

y el segundo botón sería para darle a autocompletar este apartado con la info correspondiente copiada que es la relacionada con esto

mi pagina info copiada

"apartment": "",,

aliexpress field

```
<input type="text" class="input-text " name="billing_address_2"
```

```
id="billing_address_2" placeholder="Apartamento, habitación, etc. (opcional)"
```

```
value="" autocomplete="address-line2" data-placeholder="Apartamento, habitación,  
etc. (opcional)">
```

lo ideal para mi seria que no hubiera dos botones si no secuencias, que cuando le
de a autocompletar se active el primer paso, y que espere 3 segundos y se active el
2 paso.