Algoritmo del Servicio WooService

Objetivo General  
El servicio WooService tiene como objetivo procesar los pedidos de WooCommerce en estado 'Empacando', almacenarlos en una base de datos local y sincronizarlos con Dynamics AX para su facturación y despacho. Además, se implementará un manejo robusto de errores, notificaciones de posibles soluciones, y se monitoreará la eficiencia del servicio.

# Fases del Algoritmo

## 1. Descarga de Pedidos desde WooCommerce

- Conexión al API REST de WooCommerce: El servicio se conecta al API REST de WooCommerce, alojado en Webifica, utilizando las credenciales (customer\_key, secret\_key).  
- Recuperación de Pedidos: Descargar todos los pedidos en estado 'Empacando' en formato JSON.  
- Deserialización del JSON: Convertir el listado de pedidos en objetos de C# para su procesamiento.  
- Manejo de Errores de Conexión: Si se produce un error en la conexión al API de WooCommerce, notificar al administrador con las posibles causas y pasos de resolución, como verificar las credenciales o la conectividad del servidor.

## 2. Procesamiento del Cliente

- Extracción de Datos del Cliente: Para cada pedido, extraer nombre, direcciones (facturación y envío), municipio, departamento, NIT, y correo electrónico.  
- Validación del Cliente:  
 - Comprobación de Cliente: Verificar si el cliente ya existe en la base de datos local 'arimany-woocommerce'. Si no existe, crear un nuevo registro en la tabla 'Clientes'. Si ya existe, actualizar la información.  
 - Comprobación de Direcciones: Verificar si la dirección ya está en la base de datos 'arimany-woocommerce' y 'arimany-order' (municipios y departamentos). Si no, crear una nueva entrada. Si la dirección de envío no está especificada, utilizar la dirección de facturación.  
 - Comprobación de Correo Electrónico: Validar que el correo electrónico tenga el formato adecuado usando una expresión regular.

Código en C# para validar correo electrónico:  
  
bool IsValidEmail(string email) {  
 var emailPattern = @'^[^@\s]+@[^@\s]+\.[^@\s]+$';  
 return Regex.IsMatch(email, emailPattern);  
}

- Manejo de Errores de Validación: Si los datos del cliente o las direcciones no son válidos, notificar al administrador del portal con una posible solución.

## 3. Registro de Pedido en Base de Datos Local

- Insertar Encabezado del Pedido: Registrar el pedido en la tabla 'Woo\_Pedidos'.  
- Sincronización del Cliente con Dynamics AX: Sincronizar o actualizar el cliente en Dynamics AX a través de un servicio WCF/AIF. En caso de errores, notificar con un detalle específico del error y sugerencias para corregirlo.

## 4. Procesamiento de Productos

- Verificación de Productos: Comprobar que cada producto tenga un código válido, cantidad mayor que cero y precio mayor que cero.  
- Registro de Productos: Insertar los productos en la tabla 'WooPedido\_producto'. Si el pedido incluye recargos por envío, añadir el producto especial (código: 007239) para reflejar estos cargos.  
- Registro en Tabla de Sincronización: Registrar el pedido en la tabla 'Pedidos' con el estado 'No sincronizado en AX'.

## 5. Sincronización de Pedidos con Dynamics AX

- Obtener Pedidos Pendientes: El servicio se conecta a la base de datos local 'arimany-woocommerce' para obtener todos los pedidos con estado 'No sincronizado en AX'.  
- Sincronización con Dynamics AX: Para cada pedido pendiente, enviar el encabezado y los productos en formato JSON al servicio WCF/AIF de Dynamics AX. Si la sincronización es exitosa, actualizar el estado del pedido en la tabla 'Pedidos' como 'Sincronizado en AX'.  
- Manejo de Errores de Sincronización: Si ocurre un error durante la sincronización, registrar el error en la bitácora y notificar al administrador, indicando una posible solución.

## 6. Actualización de Datos en Dynamics AX

- Registrar/Actualizar Cliente en AX: Deserializar los datos del cliente (nombre, NIT, dirección) desde el JSON y registrarlo en Dynamics AX, o actualizar el registro existente.  
- Registrar Pedido en AX: Insertar el encabezado del pedido en la tabla 'SalesTable' de Dynamics AX y los productos en la tabla 'SalesLine'. Manejar cualquier error de validación de productos y notificar a WooService.

## 7. Manejo Global de Errores y Notificaciones

- Registro en Bitácora: Todos los errores deben ser registrados en una bitácora para su posterior revisión.  
- Notificaciones con Soluciones: Incluir una posible solución en cada notificación de error.  
- Reintentos: Implementar un sistema de reintentos automáticos en caso de errores temporales.

## 8. Monitoreo de Eficiencia del Servicio

- Métricas de Rendimiento: Registrar el tiempo de respuesta de cada operación y almacenarlas en una base de datos separada o enviarlas a un sistema de monitoreo como Prometheus o Grafana.  
- Monitoreo de Fallos: Contabilizar el número de fallos de sincronización y su frecuencia para identificar patrones que puedan optimizarse.

# Librerías Sugeridas para Mejorar Comprobaciones

- Newtonsoft.Json (Json.NET) para deserializar JSON.  
- System.Text.RegularExpressions para validación de correos electrónicos.  
- Serilog para registrar eventos y errores, alternativa moderna a log4net.