

Manual del Sistema de Gestión de Pedidos de Compra

Vivero Aranjuez - Sistema Automatizado V2

Versión del Documento: 2.0

Fecha de Actualización: 6 de febrero de 2026

Autor: Sistema de Pedidos Vivero V2

1. Introducción

Este documento describe el funcionamiento completo del Sistema de Gestión de Pedidos de Compra automatizado para Vivero Aranjuez. El sistema está diseñado para gestionar el proceso de generación de pedidos de compra de manera eficiente, utilizando clasificación ABC+D y forecast de ventas. El sistema se ejecuta en dos niveles complementarios que trabajan juntos para proporcionar una gestión integral del inventario y los pedidos. Por un lado, existe un proceso trimestral que genera las clasificaciones ABC+D para cada sección, analizando el desempeño de todos los artículos durante un período determinado. Por otro lado, hay un proceso semanal que genera los pedidos específicos basándose en las clasificaciones ABC+D y los datos de ventas más recientes.

El objetivo principal del sistema es automatizar el proceso de generación de pedidos de compra para las diferentes secciones del vivero. Esto incluye el análisis de ventas históricas, la clasificación de artículos por importancia utilizando categorías A, B, C y D, y la generación de pedidos optimizados basados en forecast y corrección por tendencia de ventas. La automatización de estos procesos permite reducir errores manuales, optimizar los niveles de inventario y garantizar la disponibilidad de los productos más importantes para el negocio.

Todos los scripts del sistema comparten una estructura de directorios común que organiza los datos de entrada y salida de manera clara y sistemática. Esta organización facilita el mantenimiento del sistema y permite mantener un histórico completo de todas las operaciones realizadas. El sistema está diseñado para ser robusto y tolerante a fallos, con mecanismos de backup y recuperación que garantizan la integridad de los datos en todo momento.

2. Arquitectura del Sistema

2.1 Estructura de Directorios

La estructura de directorios del sistema está diseñada para separar claramente los datos por período mientras se mantienen ciertos archivos compartidos que son comunes a todos los procesos. Esta organización permite mantener un histórico completo de todas las operaciones y facilita la gestión de los diferentes períodos de análisis a lo largo del año.

El directorio raíz del proyecto es PROGRAMA_PEDIDOS, que contiene todos los componentes del sistema. En este nivel superior se encuentran los scripts principales que pueden ser ejecutados directamente por el usuario. El directorio config/ almacena los archivos de configuración del sistema, incluyendo la configuración general, la configuración común y la información de los encargados de cada sección. El directorio src/ contiene todos los módulos Python que implementan la lógica del sistema, organizados por funcionalidad específica. El directorio data/ es el más importante para el funcionamiento del sistema, ya que almacena todos los datos de entrada y salida organizados de manera eficiente.

La estructura completa del sistema es la siguiente:

```

PROGRAMA_PEDIDOS/
├── clasificacionABC.py          # Script principal - Clasificación
ABC+D (4 veces/año)
├── main.py                      # Script principal - Generación de
pedidos (semanal)
├── INFORME.py                  # Generador de informes detallados
└── PRESENTACION.py            # Generador de presentaciones
├── generar_informe_html.py    # Generador de informes HTML
└── run.bat                     # Archivo por lotes para ejecución
    rápida
├── requirements.txt            # Dependencias Python
└── README.md                   # Documentación inicial

|
├── config/
│   ├── config.json             # Configuración del sistema
│   ├── config_comun.json      # Configuración general
│   └── encargados.json        # Configuración adicional
                                # Encargados de secciones

|
├── src/
│   ├── config_loader.py        # Módulos del sistema (compartidos)
    centralizado
│   │   # Cargador de configuración
│   ├── data_loader.py          # Carga y procesamiento de datos
│   ├── forecast_engine.py     # Motor de forecast de ventas
│   ├── correction_engine.py   # Motor de corrección de pedidos
│   ├── order_generator.py     # Generador de pedidos
│   ├── correction_data_loader.py # Cargador de datos de corrección
│   ├── email_service.py       # Servicio de envío de emails
│   ├── scheduler_service.py   # Servicio de programación
│   └── state_manager.py       # Gestor de estado del sistema

|
├── data/
│   ├── input/
│   │   ├── Coste.xlsx          # Datos del sistema
                                # Archivos de entrada
                                # Costes unitarios (UNO SOLO -
atemporal)
|   |   ├── Ventas.xlsx        # Ventas de TODO el año (se filtra
por fechas)
|   |   ├── Compras.xlsx       # Compras de TODO el año (se filtra
por fechas)
|   |   ├── Stock_P1.xlsx      # Stock inicio Período 1 (ene-feb)
|   |   ├── Stock_P2.xlsx      # Stock inicio Período 2 (mar-may)
|   |   ├── Stock_P3.xlsx      # Stock inicio Período 3 (jun-ago)
|   |   ├── Stock_P4.xlsx      # Stock inicio Período 4 (sep-dic)

```

```

|   |   └ CLASIFICACION_ABC+D_*.xlsx # Archivos generados por
clasificacionABC
|
|   |
|   └ output/                      # Archivos de salida
|       └ Pedido_Semana_*.xlsx
|       └ Resumen_Pedidos_CONSOLIDADO_*.xlsx
|
|   └ COMPARTIDO/                  # Recursos compartidos
|       └ state.json               # Estado del sistema
|       └ state.json.backup        # Backup del estado
|       └ logs/                   # Logs de ejecución
|
└ Documentacion/                 # Documentación del sistema
    └ Manual_Sistema_Pedidos_V2.pdf # Este documento

```

2.2 Definición de Períodos

El sistema está diseñado para trabajar con cuatro períodos de análisis a lo largo del año. Cada período tiene una duración específica y se corresponde con una época del año diferente, lo que permite capturar las variaciones estacionales en las ventas y el comportamiento de compra de los clientes. Esta división en períodos es fundamental para el proceso de clasificación ABC+D, que se ejecuta cuatro veces al año, una vez al inicio de cada período.

El primer período abarca desde el 1 de enero hasta el 28 de febrero, con una duración de 59 días. Este período cubre la temporada de inicio de año, que en el sector de jardinería es generalmente de menor actividad debido al clima invernal. Sin embargo, es importante para analizar el comportamiento post-navidad y preparar la transición hacia la primavera. Durante estos meses, las ventas suelen ser más bajas en comparación con otras épocas del año, lo que permite identificar los artículos de temporada de invierno y planificar las compras necesarias para la primavera.

El segundo período va desde el 1 de marzo hasta el 31 de mayo, con una duración de 92 días. Este es uno de los períodos más importantes del año, ya que coincide con la primavera, que es la temporada alta de ventas en el sector de jardinería y plantas. Durante estos meses se produce el mayor volumen de ventas de plantas de temporada, sustratos y productos relacionados con el jardín. La clasificación ABC+D de este período es crucial para identificar los productos estrella de primavera y garantizar su disponibilidad durante toda la temporada.

El tercer período comprende desde el 1 de junio hasta el 31 de agosto, también con 92 días de duración. Este período incluye el verano, donde las ventas pueden verse afectadas por el calor extremo y las vacaciones. Es un período de transición donde se debe gestionar adecuadamente el inventario de plantas de temporada de verano y preparar la llegada del otoño. La clasificación ABC+D de este período ayuda a identificar

los productos que mantienen demanda durante el verano y aquellos que requieren estrategias de promoción para evitar la merma.

El cuarto y último período va desde el 1 de septiembre hasta el 31 de diciembre, con una duración de 122 días. Este es el período más largo del año e incluye la vuelta al cole, el otoño y la temporada navideña. Es un período estratégico para analizar las tendencias de cierre de año y planificar el siguiente ciclo. Durante estos meses se venden productos de otoño como plantas de temporada otoñal, sustratos específicos y productos para el cuidado del jardín en otoño. También incluye la temporada navideña, donde hay demanda de artículos específicos para decoración y regalos relacionados con la jardinería.

Período	Fechas	Días	Ejecución	Mes Principal
Período 1	1 enero - 28 febrero	59 días	Enero	Enero
Período 2	1 marzo - 31 mayo	92 días	Marzo	Marzo
Período 3	1 junio - 31 agosto	92 días	Junio	Junio
Período 4	1 septiembre - 31 diciembre	122 días	Septiembre	Septiembre

2.3 Archivos de Datos por Período

Una de las características más importantes del sistema es la gestión diferenciada de los archivos de datos según su naturaleza temporal. Esta aproximación simplifica significativamente la estructura del sistema y reduce la duplicación de archivos, manteniendo al mismo tiempo la flexibilidad necesaria para los diferentes procesos del sistema.

El archivo Coste.xlsx es atemporal y contiene los costes unitarios de todos los artículos del catálogo. Este archivo permanece en la carpeta data/input/ como un único archivo que se utiliza tanto para el proceso de clasificación ABC+D como para la generación semanal de pedidos. Al ser un archivo de referencia que contiene información fija sobre los costes de los artículos, no es necesario actualizarlo periódicamente ni mantener versiones por período.

El archivo Ventas.xlsx contiene las ventas de TODO el año y se mantiene en data/input/. El script clasificacionABC.py filtra este archivo por las fechas del período que se está procesando, extrayendo únicamente las ventas correspondientes a ese período específico. Por su parte, el script main.py también utiliza este archivo pero lo filtra por semana específica para el forecast semanal. Esta aproximación evita la duplicación de datos y simplifica el trabajo del ERP, que solo necesita generar un archivo de ventas con todas las transacciones del año.

El archivo Compras.xlsx funciona de manera similar a Ventas.xlsx. Contiene las compras de TODO el año y se mantiene en data/input/. El script clasificacionABC.py filtra este archivo por las fechas del período correspondiente para calcular las métricas de rotación

y beneficio de cada artículo. Esta configuración permite que el ERP genere un único archivo de compras anual sin necesidad de segmentarlo por períodos.

El archivo Stock.xlsx es el único que requiere gestión por períodos debido a su naturaleza. El stock representa la situación del inventario en un momento específico, y para el proceso de clasificación ABC+D es necesario conocer el stock al inicio de cada período. Por esta razón, el sistema utiliza cuatro archivos diferentes: Stock_P1.xlsx para el período 1 (enero-febrero), Stock_P2.xlsx para el período 2 (marzo-mayo), Stock_P3.xlsx para el período 3 (junio-agosto) y Stock_P4.xlsx para el período 4 (septiembre-diciembre). Cada archivo contiene el inventario registrado al inicio del período correspondiente.

Archivo	Ubicación	Razón	Script que lo usa
Coste.xlsx	data/input/ (uno solo)	Atemporal	clasificacionABC.py, main.py
Ventas.xlsx	data/input/ (uno anual)	Se filtra por fechas	clasificacionABC.py, main.py
Compras.xlsx	data/input/ (uno anual)	Se filtra por fechas	clasificacionABC.py
Stock_P1.xlsx	data/input/	Stock inicio Período 1	clasificacionABC.py -- periodo 1
Stock_P2.xlsx	data/input/	Stock inicio Período 2	clasificacionABC.py -- periodo 2
Stock_P3.xlsx	data/input/	Stock inicio Período 3	clasificacionABC.py -- periodo 3
Stock_P4.xlsx	data/input/	Stock inicio Período 4	clasificacionABC.py -- periodo 4

3. Descripción de Scripts

3.1 clasificacionABC.py

El script clasificacionABC.py es el componente fundamental del sistema para la clasificación de artículos. Su función principal es analizar los datos de compras, ventas, stock y costes de un período determinado para clasificar cada artículo en una categoría ABC+D. Esta clasificación es esencial para la gestión eficiente del inventario, ya que permite identificar los artículos más importantes (categoría A) y aquellos que no generan ventas (categoría D), facilitando así la toma de decisiones sobre compras, descuentos y gestión del espacio en tienda.

La frecuencia de ejecución de este script es de 4 veces al año, una vez por cada período de análisis. Esta periodicidad está diseñada para capturar las variaciones estacionales en las ventas y adaptar las clasificaciones a las diferentes épocas del año. Es importante ejecutar el script al inicio de cada período para disponer de clasificaciones actualizadas que guíen las decisiones de compra durante los siguientes meses.

Los datos de entrada del script son cuatro archivos que deben estar disponibles en la ubicación correcta. El archivo compras.xlsx contiene todos los movimientos de compras del año, y el script filtra por las fechas del período que se está procesando. El archivo Ventas.xlsx registra todas las transacciones de venta del año, también filtrado por fechas del período. El archivo Stock_Px.xlsx (donde x es el número del período) proporciona el inventario al inicio del período correspondiente. Finalmente, el archivo Coste.xlsx contiene los costes unitarios de cada artículo, necesarios para calcular el beneficio real de las ventas.

Los datos de salida del script son 11 archivos Excel, uno para cada sección del vivero. Cada archivo contiene la clasificación ABC+D de todos los artículos de esa sección, con información detallada sobre ventas, stock, beneficio, riesgo de merma y acciones sugeridas. Estos archivos se generan con un nombre que incluye el período de análisis, por ejemplo: CLASIFICACION_ABC+D_MAF_20260101-20260228.xlsx. Los archivos generados deben copiarse a la carpeta data/input/ para que main.py pueda utilizarlos en la generación semanal de pedidos.

La clasificación ABC+D divide los artículos en cuatro categorías basadas en su contribución al beneficio total. La categoría A incluye los artículos que representan el 80% del beneficio, estos son los productos estrellas que no deben faltar nunca en el almacén debido a su alta rotación y margen de contribución. La categoría B comprende los artículos que representan el siguiente 15% del beneficio, estos forman el complemento de gama y requieren una gestión adecuada para mantener su disponibilidad. La categoría C contiene los artículos con menor contribución, representando el 5% restante, y son productos de presencia mínima en las ventas que pueden requerir estrategias de promoción o reducción de stock. Finalmente, la categoría D incluye todos los artículos que no han tenido ventas durante el período de análisis, estos requieren revisión para decidir si deben mantenerse en catálogo, liquidarse o descontinuarse.

Comandos de ejecución:

El script clasificacionABC.py acepta el parámetro --periodo para especificar qué archivo de stock utilizar y qué período procesar. A continuación se presentan los comandos de ejecución disponibles:

```

# Ejecutar para el Período 1 (Enero-Febrero)
python clasificacionABC.py --periodo 1

# Ejecutar para el Período 2 (Marzo-Mayo)
python clasificacionABC.py --periodo 2

# Ejecutar para el Período 3 (Junio-Agosto)
python clasificacionABC.py --periodo 3

# Ejecutar para el Período 4 (Septiembre-Diciembre)
python clasificacionABC.py --periodo 4

# Ejecutar modo verbose para ver detalles
python clasificacionABC.py --periodo 1 --verbose

# Procesar solo una sección específica
python clasificacionABC.py --periodo 1 --seccion vivero

```

Proceso interno:

Cuando se ejecuta el script, sigue una secuencia de pasos claramente definida. Primero, carga los cuatro archivos de datos desde las ubicaciones correspondientes, seleccionando el archivo Stock correcto según el período indicado. Segundo, normaliza los datos, homogenizando códigos de artículo, nombres, tallas y colores. Tercero, filtra los artículos con códigos válidos (mínimo 10 dígitos) y elimina registros con unidades cero o valores nulos. Cuarto, para cada sección, procesa los artículos calculando métricas como ventas, beneficio, stock, tasa de rotación y riesgo de merma. Quinto, aplica la clasificación ABC+D basada en el beneficio acumulado. Sexto, genera las acciones sugeridas según el escenario de cada artículo. Séptimo, crea los archivos Excel con formato profesional y los guarda en la carpeta de salida del período. Finalmente, envía un email automáticamente al encargado de cada sección con el archivo correspondiente adjunto.

3.2 main.py

El script main.py es el motor de generación de pedidos semanales. Su función es crear los pedidos de compra para cada sección basándose en las clasificaciones ABC+D generadas por clasificacionABC.py, los datos de ventas actuales y los algoritmos de forecast y corrección. Este script es el componente operativo del sistema que se ejecuta regularmente para mantener el inventario actualizado y garantizar la disponibilidad de productos.

La frecuencia de ejecución de este script es semanal, preferentemente los domingos a las 15:00 cuando el sistema está configurado en modo automático. Esta frecuencia permite

mantener los niveles de inventario actualizados en función de las tendencias de venta más recientes y anticipar las necesidades de reposición para la semana siguiente. El script puede ejecutarse manualmente en cualquier momento utilizando los parámetros de línea de comandos disponibles.

Los datos de entrada del script utilizan múltiples fuentes de datos para generar los pedidos. Los archivos CLASIFICACION_ABC+D_.xlsx proporcionan la clasificación de artículos y sus parámetros de stock mínimo y máximo. El archivo Ventas.xlsx contiene los datos de ventas de la semana actual y anteriores, utilizados para el forecast. El archivo Stock_actual.xlsx proporciona el inventario actual de cada artículo. El archivo Coste.xlsx contiene los costes unitarios para calcular el importe de los pedidos. El archivo compras.xlsx registra las compras recientes, útil para evitar duplicidades. Adicionalmente, se pueden usar archivos de corrección opcionales como Ventas_semana_.xlsx y Compras_semana_.xlsx para la FASE 2 de corrección.

Los datos de salida del script generan múltiples archivos organizados por sección y semana. Los archivos Pedido_Semana_XX_YYYY-MM-DD_sección.xlsx contienen el detalle de pedidos para cada sección, incluyendo código de artículo, nombre, unidades a pedir, precio y proveedor. El archivo Resumen_Pedidos_CONSOLIDADO_YYYY-MM-DD.xlsx contiene un resumen consolidado de todos los pedidos de la semana, facilitando la visión global de las necesidades de compra. Adicionalmente, se pueden generar archivos corregidos cuando se aplica la corrección FASE 2.

Fases del proceso:

El proceso de generación de pedidos se divide en dos fases complementarias que trabajan juntas para producir pedidos optimizados y adaptados a la realidad operativa del negocio.

La FASE 1 - Forecast es la etapa donde el sistema analiza las ventas históricas de cada artículo para predecir las ventas de la próxima semana. El algoritmo considera múltiples factores, incluyendo las ventas de semanas anteriores que proporcionan la base para la predicción, la estacionalidad del artículo basada en su familia que determina la rotación esperada, y los días de cobertura objetivo según la categoría ABC del artículo que influyen en los niveles de stock objetivo. El resultado es un pedido teórico que cubre las necesidades predichas de cada artículo, calculando cantidades óptimas basadas en el histórico de ventas y los parámetros de cada categoría.

La FASE 2 - Corrección es la etapa donde el sistema ajusta el pedido teórico basándose en la realidad operativa actual. Se consideran factores como el stock real actual que puede estar por encima o por debajo del mínimo recomendado, las ventas reales de la semana anterior que pueden indicar tendencias de aumento o disminución, las compras recibidas recientemente que afectan la disponibilidad actual, y las tendencias detectadas de aumento de ventas que requieren incrementar los pedidos. La corrección por tendencia de ventas es especialmente importante: si un artículo está vendiendo por encima de lo esperado y ha consumido parte de su stock mínimo, el sistema incrementa el pedido para compensar y evitar rupturas de stock.

Comandos de ejecución:

```
# Ejecución automática (domingo 15:00)
python main.py

# Forzar semana específica
python main.py --semana 14

# Con corrección habilitada
python main.py --semana 14 --con-correccion

# Sin corrección (solo FASE 1)
python main.py --semana 14 --sin-correccion

# Modo continuo (espera hasta el horario de ejecución)
python main.py --continuo

# Verificar configuración de email
python main.py --verificar-email

# Ver estado del sistema
python main.py --status

# Mostrar ayuda
python main.py --help
```

Parámetros importantes:

El parámetro `--semana` o `-s` permite especificar el número de semana a procesar (1-52). Este parámetro fuerza el procesamiento de una semana específica, ignorando el cálculo automático del sistema. Es útil para reprocesar semanas anteriores o para pruebas.

El parámetro `--con-correccion` habilita la FASE 2 de corrección del pedido. Si no se especifica, el sistema usa la configuración por defecto del archivo `config.json`. Esta opción permite forzar la corrección incluso si está deshabilitada en la configuración.

El parámetro `--verbose` o `-v` activa el logging detallado, mostrando información de depuración durante la ejecución. Es útil para diagnosticar problemas o para entender el comportamiento del sistema en detalle.

El parámetro `--log` permite especificar un archivo de log personalizado. Por defecto, los registros se guardan en el archivo `logs/sistema.log`. Este parámetro es útil para separar los registros de diferentes ejecuciones.

3.3 INFORME.py

El script INFORME.py genera informes detallados de ventas para análisis gerencial. Este script es útil para revisar el desempeño de cada sección y artículo durante un período específico, proporcionando métricas y visualizaciones que facilitan la toma de decisiones estratégicas. La generación de informes puede realizarse bajo demanda cuando se requiere un análisis detallado del rendimiento del negocio.

Comandos de ejecución:

```
# Generar informe del Período 1  
python INFORME.py --periodo 1  
  
# Generar informe verbose  
python INFORME.py --periodo 1 --verbose
```

3.4 PRESENTACION.py

El script PRESENTACION.py genera presentaciones en formato HTML para reuniones de equipo o presentaciones gerenciales. Las presentaciones incluyen gráficos y métricas clave del desempeño de cada sección, diseñadas para comunicar de manera efectiva los resultados y las recomendaciones derivadas del análisis de datos. Este script se ejecuta bajo demanda, típicamente para reuniones o presentaciones formales donde se requiere visualizar la información de manera profesional.

Comandos de ejecución:

```
# Generar presentación del Período 1  
python PRESENTACION.py --periodo 1
```

4. Configuración del Sistema

4.1 Archivo config.json

El archivo config.json contiene la configuración general del sistema y define todos los parámetros necesarios para el funcionamiento de los diferentes scripts. A continuación se describen las secciones principales de este archivo, organizadas de manera que facilitan la comprensión y modificación de la configuración según las necesidades del negocio.

La configuración de rutas define las rutas relativas para los directorios de entrada, salida y otros recursos del sistema. Estas rutas son relativas al directorio del script que se está ejecutando y pueden ajustarse según la estructura de carpetas deseada. Los directorios

principales incluyen el directorio_base que indica la ubicación base del proyecto, el directorio_entrada donde se encuentran los archivos de datos, el directorio_salida donde se generan los archivos de resultados, y el directorio_logs donde se almacenan los registros de ejecución.

La configuración de parámetros incluye valores globales utilizados por los algoritmos de forecast y corrección. El objetivo_crecimiento define el porcentaje de crecimiento esperado en las ventas (por defecto 5%), que se utiliza para ajustar los pedidos en función de las tendencias de crecimiento. El stock_minimo_porcentaje establece el porcentaje del stock objetivo que se considera como nivel mínimo (por defecto 30%). Los pesos_categoria definen multiplicadores específicos para cada categoría ABC que afectan los cálculos de stock mínimo y máximo.

La configuración de secciones define las secciones activas del vivero y sus parámetros específicos. Cada sección incluye una descripción que identifica el tipo de productos que gestiona, y objetivos_semanales que contienen los importes objetivo de venta para cada una de las 53 semanas del año. Estos objetivos permiten capturar la estacionalidad de las ventas y ajustar los pedidos en función de las expectativas de cada época del año.

La configuración de email define los parámetros del servidor SMTP para el envío automático de emails, incluyendo el servidor y puerto de correo, las credenciales del remitente, y las direcciones de los destinatarios por sección. Esta configuración permite que el sistema envíe automáticamente los pedidos generados a los responsables de cada sección, facilitando la comunicación y el seguimiento de las operaciones.

La configuración de corrección define los parámetros de la FASE 2 de corrección, incluyendo si la corrección está habilitada por defecto, las políticas de stock mínimo por categoría ABC que determinan los niveles de seguridad del inventario, y otros umbrales utilizados para detectar situaciones que requieren ajustes en los pedidos.

4.2 Archivo encargados.json

El archivo encargados.json mapea cada sección con su encargado correspondiente, incluyendo nombre y email. Este archivo es utilizado por el sistema para enviar automáticamente los informes de clasificación ABC+D y los pedidos semanales a cada encargado. La estructura del archivo permite definir múltiples encargados por sección si es necesario, facilitando la distribución de responsabilidades en la gestión del inventario.

La estructura del archivo es la siguiente:

```
{  
    "encargados": {  
        "maf": {  
            "nombre": "Nombre del encargado",  
            "email": "email@ejemplo.com"  
        },  
        "interior": {  
            "nombre": "Nombre del encargado",  
            "email": "email@ejemplo.com"  
        }  
    }  
}
```

4.3 Configuración de Variables de Entorno

Para el envío de emails, el sistema requiere la contraseña del remitente configurada como variable de entorno. Esta aproximación protege las credenciales evitando almacenarlas en archivos de configuración que podrían ser accedidos inadvertidamente. La configuración de la variable de entorno varía según el sistema operativo utilizado.

En sistemas Windows (PowerShell), la configuración se realiza de la siguiente manera:

```
$env:EMAIL_PASSWORD="tu_contraseña_aquí"
```

En sistemas Linux/macOS (bash), la configuración se realiza así:

```
export EMAIL_PASSWORD="tu_contraseña_aquí"
```

Es importante configurar esta variable de entorno antes de ejecutar los scripts que requieren envío de emails, ya que de lo contrario el sistema no podrá autenticarse en el servidor de correo y los envíos fallarán.

5. Flujo de Trabajo del ERP

5.1 Envío de Archivos para Clasificación ABC+D (4 veces al año)

El sistema ERP debe seguir un procedimiento específico para enviar los archivos de datos al sistema de pedidos. Este proceso debe ejecutarse al final de cada período de análisis, antes de ejecutar clasificacionABC.py, para garantizar que el sistema disponga de datos actualizados para la clasificación de artículos.

Paso 1: Generación de archivos

El ERP genera los cuatro archivos necesarios con los datos del período correspondiente. Es importante que los archivos Ventas.xlsx y Compras.xlsx contengan los datos de TODO el año, ya que el sistema filtrará por las fechas del período que se está procesando. El archivo Stock_Px.xlsx debe contener el inventario al inicio del período correspondiente, y el archivo Coste.xlsx contiene los costes unitarios actuales de todos los artículos.

Paso 2: Envío a la ubicación correcta

El ERP coloca los archivos en las ubicaciones correspondientes dentro de la estructura de datos del sistema. El archivo Stock_Px.xlsx debe enviarse a data/input/ con el nombre correcto según el período (Stock_P1.xlsx para período 1, Stock_P2.xlsx para período 2, etc.). Los archivos Ventas.xlsx, Compras.xlsx y Coste.xlsx se mantienen en data/input/ ya que contienen datos anuales y se utilizan para todos los períodos.

Paso 3: Ejecución del script

Una vez enviados los archivos, el usuario ejecuta el script de clasificación para el período correspondiente, especificando el número de período como parámetro:

```
python clasificacionABC.py --periodo 1
```

Paso 4: Verificación de resultados

El script genera los archivos de clasificación en la carpeta de salida correspondiente. El usuario debe verificar que todos los archivos se generaron correctamente y que no hubo errores durante el procesamiento.

Paso 5: Copia a la carpeta de entrada para main.py

Los archivos ABC+D generados deben copiarse a la carpeta data/input/ para que main.py pueda utilizarlos en la generación semanal de pedidos. Esta copia es necesaria porque main.py lee los archivos ABC+D desde la carpeta de entrada, no desde las carpetas específicas de cada período.

```
# Copiar todos los archivos ABC+D al directorio de entrada  
cp data/output/CLASIFICACION_ABC+D_*.xlsx data/input/
```

5.2 Envío de Archivos Semanales

Para la generación semanal de pedidos, el ERP debe mantener actualizados los siguientes archivos en la carpeta data/input/. Estos archivos son utilizados por main.py para generar los pedidos de cada semana y deben reflejar la situación actual del negocio.

El archivo Ventas.xlsx debe actualizarse semanalmente con las ventas de la semana actual. El sistema lo utiliza para el forecast de la próxima semana, por lo que es importante que contenga los datos más recientes de transacciones. Este archivo es acumulativo y debe contener las ventas de TODO el año, no solo de la semana actual.

El archivo Stock_actual.xlsx debe reflejar el inventario actual del almacén. Se utiliza en la FASE 2 de corrección para ajustar los pedidos según el stock real disponible. Es importante mantener este archivo actualizado para garantizar que los pedidos generados reflejen las necesidades reales de reposición.

El archivo Coste.xlsx contiene los costes unitarios de cada artículo y debe mantenerse actualizado. Se utiliza para calcular el importe de los pedidos y determinar los márgenes de beneficio. Aunque este archivo cambia con menos frecuencia que los otros, es importante actualizarlo cuando hay cambios en los precios de coste de los proveedores.

El archivo compras.xlsx registra las compras recientes y se utiliza para evitar duplicidades en los pedidos. Al mantener un registro de las compras recientes, el sistema puede identificar artículos que ya están en camino y ajustar los pedidos en consecuencia.

Archivos de corrección opcionales:

Para la FASE 2 avanzada, se pueden proporcionar archivos adicionales que proporcionan información más detallada sobre las operaciones de la semana. El archivo Ventas_semana_XX.xlsx contiene las ventas específicas de la semana actual y permite al sistema detectar tendencias de aumento o disminución en la demanda. El archivo Compras_semana_XX.xlsx registra las compras recibidas durante la semana, lo que permite ajustar los pedidos considerando el inventario que ya está en camino.

6. Algoritmos del Sistema

6.1 Clasificación ABC+D

El algoritmo de clasificación ABC+D es fundamental para la gestión eficiente del inventario. Se basa en el principio de Pareto (80/20), donde un pequeño número de artículos genera la mayor parte del beneficio del negocio. Este principio reconoce que no todos los artículos tienen la misma importancia para el negocio, y que concentrando los esfuerzos de gestión en los artículos más valiosos se puede obtener el mayor retorno de la inversión en tiempo y recursos.

Paso 1: Filtrado de artículos con ventas

El primer paso del algoritmo es separar los artículos que han tenido ventas durante el período de aquellos que no han vendido nada. Los artículos sin ventas se clasifican automáticamente como categoría D, ya que no han contribuido al beneficio durante el período analizado. Este filtrado es importante para enfocar el análisis en los artículos relevantes y evitar sesgar los cálculos con artículos inactivos.

Paso 2: Cálculo de beneficio por artículo

Para cada artículo con ventas, se calcula el beneficio total generado durante el período. El beneficio se calcula como el importe de ventas menos el coste de las mercancías vendidas. Este cálculo considera el margen real de cada artículo después de deducir los costes directos asociados a su venta.

Paso 3: Ordenamiento por beneficio

Los artículos se ordenan de mayor a menor beneficio generado, lo que permite identificar claramente los artículos más importantes del catálogo. Este ordenamiento es la base para el cálculo de los porcentajes acumulados que determinan las categorías ABC.

Paso 4: Cálculo de porcentaje acumulado

Se calcula el porcentaje de beneficio de cada artículo respecto al total de beneficio de todos los artículos con ventas. Luego se calcula el porcentaje acumulado, que representa la suma de los porcentajes de todos los artículos desde el más importante hasta el actual.

Paso 5: Asignación de categorías

Con base en los porcentajes acumulados, se asignan las categorías según los umbrales definidos:

- **Categoría A:** Artículos con porcentaje acumulado menor o igual al 80%. Estos son los productos estrellas que generan la mayor parte del beneficio y requieren especial atención para garantizar su disponibilidad permanente.
- **Categoría B:** Artículos con porcentaje acumulado mayor al 80% y menor o igual al 95%. Estos artículos complementan la oferta principal y requieren una gestión adecuada para mantener su presencia en el catálogo.
- **Categoría C:** Artículos restantes con porcentaje acumulado mayor al 95%. Estos productos tienen una contribución menor al beneficio y pueden requerir estrategias de promoción o reducción de stock.
- **Categoría D:** Artículos sin ventas durante el período. Estos artículos requieren revisión para decidir si deben mantenerse, liquidarse o descontinuarse del catálogo.

6.2 Forecast de Ventas (FASE 1)

El algoritmo de forecast predice las ventas de la próxima semana basándose en datos históricos y en las características específicas de cada artículo. El objetivo es generar un pedido teórico que cubra las necesidades predichas de inventario, optimizando los niveles de stock para evitar tanto rupturas como excesos de inventario.

Factores considerados:

El algoritmo considera múltiples factores para generar el forecast. Las ventas históricas del artículo proporcionan la base para predecir las ventas futuras, considerando patrones de venta y estacionalidad. La familia del artículo determina la rotación esperada, expresada en días de cobertura de stock, que varía según el tipo de producto. La categoría ABC del artículo influye en los objetivos de disponibilidad, con niveles más estrictos para artículos de categorías superiores. Los días de cobertura objetivo varían según la categoría, siendo más estrictos para artículos A que requieren mayor disponibilidad.

Cálculo de stock mínimo y máximo:

El sistema calcula el stock mínimo y máximo para cada artículo según su categoría y familia, utilizando fórmulas específicas basadas en la rotación esperada de cada familia de productos:

Para rotación de 7 días: Stock mínimo = ventas_día × 3.5, Stock máximo = ventas_día × 10.5

Para rotación de 15 días: Stock mínimo = ventas_día × 7.5, Stock máximo = ventas_día × 22.5

Para rotación de 30 días: Stock mínimo = ventas_día × 15, Stock máximo = ventas_día × 45

Para rotación de 60 días: Stock mínimo = ventas_día × 30, Stock máximo = ventas_día × 90

Para rotación de 90 días: Stock mínimo = ventas_día × 45, Stock máximo = ventas_día × 135

Estas fórmulas permiten adaptar los niveles de stock a las características de cada producto, garantizando disponibilidad adecuada para artículos de alta rotación mientras se evita el exceso de inventario para artículos de baja rotación.

6.3 Corrección de Pedidos (FASE 2)

La FASE 2 ajusta el pedido teórico de la FASE 1 basándose en la realidad operativa actual del negocio. Esta corrección es esencial para adaptar los pedidos a las condiciones reales del inventario y las tendencias de venta, evitando tanto rupturas de stock como excesos de inventario.

Fórmula principal:

La corrección se aplica mediante la siguiente fórmula:

```
Pedido_Corregido = max(0, Pedido_Generado + (Stock_Mínimo - Stock_Real))
```

Donde Pedido_Generado es el resultado del algoritmo de forecast (FASE 1), Stock_Mínimo es el stock mínimo objetivo del artículo calculado según su categoría ABC, Stock_Real es el stock físico actual en el almacén, y la función max(0, ...) garantiza que no se generen pedidos negativos.

Esta fórmula asegura que el pedido final considere tanto las necesidades predichas por el forecast como la situación actual del inventario. Si el stock real está por encima del mínimo, el pedido se reduce; si está por debajo, el pedido se incrementa para compensar.

Corrección por tendencia de ventas:

Cuando las ventas reales de un artículo superan las ventas objetivo, el sistema detecta una tendencia de aumento y aplica un incremento adicional al pedido. Este ajuste es crucial para artículos que están experimentando un aumento en la demanda, ya que permite anticipar las necesidades de reposición antes de que se agote el stock.

El cálculo de la corrección por tendencia es:

```
Porcentaje_Consumido = (Ventas_Reales - Ventas_Ojetivo) /  
Ventas_Ojetivo
```

```
Incremento_Tendencia = Ventas_Ojetivo × Porcentaje_Consumido
```

```
Pedido_Final = Pedido_Corregido + Incremento_Tendencia
```

Este ajuste proporcional permite incrementar el pedido de manera proporcional al exceso de ventas detectado, adaptándose a diferentes magnitudes de tendencia sin sobreactuar ni infravalorar las necesidades reales.

7. Secciones del Sistema

El sistema está diseñado para gestionar las siguientes secciones del vivero, cada una con sus propias características de codificación, rotación de inventario y encargado responsable. Esta organización permite adaptar los algoritmos de forecast y corrección a las particularidades de cada tipo de producto, optimizando la gestión del inventario de manera integral.

MAF (Mesas, Asientos, Fuentes): Sección dedicada a mobiliario de jardín. Los códigos de artículo comienzan con el dígito 7. Esta sección incluye mesas, sillas, bancos, fuentes decorativas y otros elementos de mobiliario para exteriores. La rotación de esta categoría es generalmente más lenta que la de plantas debido a la naturaleza de los productos.

DECO_INTERIOR (Decoración Interior): Artículos de decoración para interiores. Los códigos de artículo comienzan con el dígito 6. Incluye jarrones, cuadros, esculturas, textiles decorativos y otros elementos para la decoración del hogar. La rotación depende en gran medida de las tendencias de decoración y las temporadas.

SEMILLAS: Sección de semillas y productos relacionados con la siembra. Los códigos de artículo comienzan con el dígito 5. Incluye semillas de flores, hortalizas, Césped y otros productos de siembra. La demanda de esta sección tiene una fuerte estacionalidad, concentrándose en primavera y otoño.

MASCOTAS_VIVO: Productos vivos para mascotas (animales). Los códigos de artículo son específicos y comienzan con códigos de 4 dígitos (2104, 2204, etc.). Incluye peces, aves, reptiles y otros animales domésticos. Esta sección requiere condiciones especiales de almacenamiento y manejo.

MASCOTAS_MANUFACTURADO: Productos manufacturados para mascotas. Los códigos de artículo comienzan con el dígito 2 pero no están en la lista de códigos de vivos. Incluye alimentos, accesorios, jaulas y productos para el cuidado de mascotas. Esta sección suele tener alta rotación debido al consumo recurrente.

INTERIOR: Plantas de interior. Los códigos de artículo comienzan con el dígito 1. Incluye plantas decorativas de interior con diferentes necesidades de luz y riego. La demanda de plantas de interior varía según las tendencias de decoración y las estaciones.

FITOS (Fitosanitarios): Productos fitosanitarios y de protección de plantas. Los códigos de artículo comienzan con el dígito 3 pero no son de tierra o áridos (códigos 31, 32). Incluye pesticidas, fungicidas, abonos y productos para el cuidado de plantas. Esta sección requiere especial atención a la normativa de productos fitosanitarios.

VIVERO: Plantas de vivero y árboles. Los códigos de artículo comienzan con el dígito 8. Incluye árboles, arbustos, plantas perennes y otros productos de vivero. La rotación de esta sección es variable según la temporada y el tipo de planta.

UTILES_JARDIN (Utiles de Jardín): Herramientas y útiles de jardín. Los códigos de artículo comienzan con el dígito 4. Incluye herramientas de mano, mangueras, tijeras, guantes y otros productos para el cuidado del jardín. Esta sección tiene una rotación más lenta pero márgenes generalmente más altos.

TIERRAS_ARIDOS (Tierras y Áridos): Tierras, sustratos y áridos. Los códigos de artículo comienzan con 31 o 32. Incluye tierras vegetales, sustratos, grava, arena y otros materiales para jardinería. Esta sección tiene una fuerte estacionalidad concentrada en primavera.

DECO_EXTERIOR (Decoración Exterior): Artículos de decoración para exteriores. Los códigos de artículo comienzan con el dígito 9. Incluye macetas, figuras decorativas, iluminación exterior y otros elementos para jardines y terrazas.

8. Troubleshooting y Mantenimiento

8.1 Problemas Comunes

Error: No se encontró el archivo

Si el sistema reporta que no encuentra un archivo, verificar que la ruta sea correcta y que el archivo exista. Para clasificacionABC.py, verificar que el archivo Stock_Px.xlsx esté en data/input/ y que el número de período sea correcto. Para main.py, verificar que los archivos ABC+D estén en data/input/ y que los archivos de datos principales (Ventas.xlsx, Stock_actual.xlsx, Coste.xlsx) estén en sus ubicaciones correspondientes.

Error: Faltan columnas en el archivo

El sistema busca columnas específicas en cada archivo. Si falta una columna, revisar que el archivo Excel tenga las columnas esperadas. El sistema tiene lógica de normalización de columnas que es insensible a mayúsculas y minúsculas y acentos, pero requiere que las columnas existan con nombres similares a los esperados. Verificar la documentación de formatos de archivo en los anexos de este manual.

Error: SMTP al enviar emails

Verificar que la variable de entorno EMAIL_PASSWORD esté configurada correctamente. Verificar la configuración del servidor SMTP en config.json. Usar el comando python main.py --verificar-email para diagnosticar problemas de configuración de email.

No hay datos de ventas para la semana

Esto puede ocurrir si se especifica una semana futura o si el archivo Ventas.xlsx no contiene datos para esa semana. Verificar que el archivo Ventas.xlsx tenga registros para la semana que se intenta procesar. Asegurarse de que el archivo Ventas.xlsx contenga los datos de TODO el año y no solo de un período específico.

8.2 Backup y Recuperación

Respaldo del estado del sistema:

El archivo state.json en data/COMPARTIDO/ contiene el estado actual del sistema, incluyendo el histórico de ejecuciones y el stock acumulado. Se recomienda hacer backup de este archivo regularmente, especialmente antes de ejecutar procesos importantes como la clasificación ABC+D o la generación de pedidos de fin de año.

Recuperación después de un error:

Si el sistema falla durante la ejecución, verificar los logs en data/COMPARTIDO/logs/ para identificar la causa del error. Los archivos de log contienen información detallada sobre cada ejecución, incluyendo errores, advertencias y métricas de rendimiento. Corregir el problema identificado y re-ejecutar el script con la opción --semana para forzar el reprocesamiento de la semana o período afectado.

8.3 Actualización del Sistema

Actualización de código:

Si se actualiza el código del sistema, verificar que todos los scripts estén sincronizados. Las configuraciones en config/ son compartidas, por lo que cualquier cambio afectará a todos los scripts. Se recomienda probar los cambios en un entorno de desarrollo antes de aplicarlos en producción.

Cambio de período:

Para cambiar de período de análisis, simplemente ejecutar clasificacionABC.py con el nuevo número de período. El sistema utilizará automáticamente el archivo Stock correcto (Stock_P1.xlsx, Stock_P2.xlsx, etc.) según el período especificado. No es necesario mover archivos ni modificar la estructura de carpetas.

9. Anexos

9.1 Formato de Archivos de Entrada

Formato de compras.xlsx:

Campo	Tipo	Descripción
Artículo	Texto	Código del artículo
Nombre artículo	Texto	Descripción del artículo

Campo	Tipo	Descripción
Fecha	Fecha	Fecha de la compra
Unidades	Número	Cantidad comprada
Precio	Número	Precio unitario

Formato de Ventas.xlsx:

Campo	Tipo	Descripción
Artículo	Texto	Código del artículo
Nombre artículo	Texto	Descripción del artículo
Fecha	Fecha	Fecha de la venta
Unidades	Número	Cantidad vendida
Importe	Número	Importe total de la venta
Tipo registro	Texto	Tipo de registro (usar "Detalle")

Formato de Stock.xlsx:

Campo	Tipo	Descripción
Artículo	Texto	Código del artículo
Nombre artículo	Texto	Descripción del artículo
Talla	Texto	Talla del artículo (si aplica)
Color	Texto	Color del artículo (si aplica)
Unidades	Número	Stock disponible

Formato de Coste.xlsx:

Campo	Tipo	Descripción
Artículo	Texto	Código del artículo
Talla	Texto	Talla del artículo
Color	Texto	Color del artículo
Coste	Número	Coste unitario
Últ. Compra	Fecha	Fecha de última compra

9.2 Códigos de Familia y Rotación

El sistema utiliza códigos de familia para determinar la rotación esperada de cada artículo. Los códigos de familia se determinan según los primeros dígitos del código del artículo:

Familia	Nombre	Rotación (días)
10-19	Plantas Interior	30
20-29	Mascotas	30
31-32	Tierras y Áridos	60
33-39	Fitosanitarios	90
40-49	Útiles Jardín	90
50-59	Semillas	60
60-69	Deco Interior	90
70-79	MAF	90
80-89	Vivero	60
90-99	Deco Exterior	90

9.3 Matriz de Escenarios

El sistema utiliza una matriz de escenarios para determinar las acciones sugeridas para cada artículo. Cada escenario considera tres factores: ventas reales versus objetivo, compras reales versus sugerido, y stock real versus mínimo. La matriz completa de 26 escenarios proporciona acciones específicas desde descuento máximo y reducción de compras hasta reposición inmediata y aumento de stock.

10. Glosario de Términos

ABC+D: Método de clasificación de artículos basado en su importancia, donde A representa el 80% del beneficio, B el 15%, C el 5% y D los artículos sin ventas.

Forecast: Predicción de ventas futuras basada en datos históricos y algoritmos de análisis.

Rotación: Número de días que tarda un artículo en venderse completamente. Determina los niveles de stock mínimo y máximo.

Stock mínimo: Nivel de inventario mínimo recomendado para un artículo, calculado según su categoría y familia.

Stock máximo: Nivel de inventario máximo recomendado para un artículo, calculado según su categoría y familia.

Tasa de venta: Porcentaje del stock disponible que se ha vendido durante un período determinado.

Período de análisis: Intervalo de tiempo utilizado para la clasificación ABC+D, con una duración de 2-3 meses.

Semana de pedido: Semana específica para la cual se genera un pedido de compra.

FASE 1: Etapa de forecast donde se genera el pedido teórico basándose en ventas históricas.

FASE 2: Etapa de corrección donde se ajusta el pedido teórico según la realidad operativa.

Corrección por tendencia: Ajuste adicional al pedido cuando las ventas reales superan las ventas objetivo, anticipando posibles incrementos de demanda.

Documento generado automáticamente por el Sistema de Pedidos Vivero V2

Para dudas o sugerencias, consultar al departamento de sistemas.