MANUAL DE USUARIO

SISTEMA DE PEDIDOS DE COMPRA (SPA)

Vivero Aranjuez V2

Versión del documento: 1.0
Fecha de creación: Febrero 2026
Autor: Sistema de Pedidos Vivero V2

ÍNDICE GENERAL

---

1. INTRODUCCIÓN AL SISTEMA

1.1 Propósito y Objetivos del Sistema

El Sistema de Pedidos de Compra (SPA) es una aplicación desarrollada específicamente para Vivero Aranjuez con el propósito de automatizar completamente el proceso de generación de pedidos de compra semanales. Este sistema representa una solución integral que elimina la necesidad de realizar cálculos manuales tediosos y propensos a errores, permitiendo a los responsables del vivero enfocarse en tareas de mayor valor estratégico.

El objetivo principal del sistema consiste en calcular de manera precisa y automática las cantidades óptimas de pedido para cada artículo, basándose en múltiples factores como las ventas históricas, los objetivos de venta semanales, la clasificación ABC de los artículos, el stock actual disponible y diversos parámetros de configuración que permiten adaptar el sistema a las necesidades específicas del negocio.

Además de la generación de pedidos, el sistema incluye múltiples módulos complementarios que proporcionan funcionalidades adicionales como la clasificación automática de artículos según su importancia relativa (análisis ABC+D), la generación de informes periódicos, el análisis de compras sin autorización, la identificación de artículos no comprados y análisis detallados por categoría. Todas estas funcionalidades trabajan de manera coordinada para proporcionar una visión completa de la gestión de inventarios del vivero.

1.2 Alcance del Software

El alcance del sistema SPA abarca toda la cadena de procesos relacionados con la gestión de pedidos de compra, desde la obtención y procesamiento de datos de ventas históricas hasta la generación final de archivos de pedido listos para su envío a proveedores. El sistema procesa información de múltiples secciones del vivero, incluyendo planta de temporada y floristería, decoración interior y exterior, semillas y bulbos, mascotas (tanto animales vivos como productos manufacturados), plantas de interior, fitosanitarios y abonos, vivero y plantas de exterior, útiles de jardín, tierras y áridos.

El software está diseñado para funcionar de manera completamente autónoma una vez instalado y configurado, ejecutando las tareas programadas de forma automática según el calendario predefinido. Sin embargo, también permite la ejecución manual de cualquiera de sus módulos para situaciones especiales o pruebas.

1.3 Beneficios Principales

La implementación del sistema SPA proporciona múltiples beneficios tangibles para la gestión del vivero. En primer lugar, se logra una significativa reducción del tiempo dedicado a la elaboración de pedidos, pasando de varias horas de trabajo manual a unos

pocos minutos de ejecución automática. Esta eficiencia permite a los empleados dedicar su tiempo a otras tareas importantes del día a día del negocio.

En segundo lugar, el sistema garantiza una mayor precisión en los cálculos, eliminando los errores humanos que pueden ocurrir al realizar cálculos manualmente, especialmente cuando se trabaja con grandes volúmenes de datos y múltiples artículos. Los algoritmos del sistema aplican de manera consistente todos los parámetros de configuración definidos, asegurando que cada pedido se calcule siguiendo los mismos criterios.

En tercer lugar, la automatización del proceso permite mantener una regularidad en la generación de pedidos que manualmente podría verse comprometida por diversas circunstancias. El sistema ejecuta las tareas según el calendario establecido, asegurando que ningún pedido se retrase por olvidos o falta de disponibilidad del personal.

Finalmente, el sistema proporciona una trazabilidad completa de todas las ejecuciones, manteniendo un histórico detallado de los pedidos generados, las ventas reales versus las previsiones, y diversas métricas que permiten analizar la evolución del negocio y realizar ajustes informados a los parámetros de configuración.

1.4 Glosario de Términos Técnicos

Para facilitar la comprensión del presente manual, a continuación se define una serie de términos técnicos utilizados frecuentemente en la documentación y el sistema:

El término **Sección** se refiere a cada una de las áreas temáticas del vivero que se gestionan de manera independiente dentro del sistema, como pueden ser planta de temporada, decoración interior, mascotas, etc. Cada sección tiene sus propios objetivos de venta semanales y sus propios encargados responsables.

Clasificación ABC es una metodología de gestión de inventarios que clasifica los artículos en tres categorías según su importancia relativa, medida normalmente por su contribución al volumen de negocio o a las ventas. Los artículos de categoría A son los más importantes (normalmente representan el 80% del valor), los de categoría B tienen una importancia media, y los de categoría C son los menos importantes.

La **Clasificación ABC+D** añade una categoría D para identificar artículos que requieren atención especial, normalmente aquellos con baja rotación o problemas de rentabilidad.

Los **Períodos (P1, P2, P3, P4)** son divisiones del año utilizadas por el sistema para realizar análisis específicos y generar informes periódicos. Cada período corresponde a una estación del año y tiene una duración aproximada de tres meses.

El término **Objetivo de venta semanal** hace referencia a la cantidad de ventas que se espera alcanzar en una semana determinada para cada sección, expresada en términos monetarios.

El **Stock mínimo** representa la cantidad mínima de un artículo que debe mantenerse en inventario, expresada normalmente como un porcentaje del stock objetivo o como una cantidad fija.

---

2. DESCRIPCIÓN GENERAL DEL SISTEMA

2.1 Arquitectura del Sistema

El sistema SPA está construido sobre una arquitectura modular que permite la ejecución independiente de cada componente mientras mantiene una integración fluida a través de archivos compartidos y una configuración común. La arquitectura se divide en tres capas principales que trabajan conjuntamente para lograr los objetivos del sistema.

La capa de datos es responsable de la obtención, lectura y procesamiento de información proveniente de archivos Excel externos. Esta capa incluye módulos especializados para la carga de datos de ventas, costes, stocks y clasificaciones, aplicando procesos de normalización que permiten manejar variaciones en los nombres de columnas y formatos de los datos de entrada.

La capa de lógica de negocio contiene los motores de cálculo que procesan los datos y generan los resultados esperados. Esta capa incluye el motor de previsión (ForecastEngine) que calcula los pedidos basándose en los algoritmos definidos, el motor de corrección que ajusta los pedidos según el stock actual, y los diversos módulos de análisis que generan los informes complementarios.

La capa de presentación se encarga de generar los archivos de salida en formato Excel, aplicando formatos profesionales que facilitan la lectura e interpretación de los resultados. Esta capa también incluye los módulos de generación de emails y notificaciones.

2.2 Componentes Principales

El sistema está compuesto por múltiples scripts Python que se ejecutan de manera independiente o coordinada según las necesidades. Cada componente tiene una función específica dentro del ecosistema general del sistema.

El componente principal, denominado **main.py**, coordina todo el proceso de generación de pedidos semanales. Este script orchestrar los demás módulos, desde la carga de datos hasta la generación final de los archivos de pedido, pasando por todos los cálculos intermedios.

El script **clasificacionABC.py** es responsable de calcular la clasificación ABC+D de todos los artículos basándose en su histórico de ventas y otros parámetros. Este script se ejecuta con periodicidad mensual y genera archivos de clasificación que son utilizados por el módulo principal para ajustar los pedidos.

El componente **forecast_engine.py** implementa los algoritmos de cálculo de previsiones de venta y determinación de cantidades de pedido. Este motor aplica los factores de crecimiento, los objetivos semanales, los pesos por categoría y todas las demás reglas de negocio definidas en la configuración.

El **order_generator.py** es el componente encargado de generar los archivos Excel de salida con los pedidos calculados. Aplica formatos profesionales y organiza la información de manera que sea fácilmente interpretable por los responsables de compras.

Existen además componentes auxiliares como **state_manager.py** que gestiona la persistencia del estado del sistema entre ejecuciones, **scheduler_service.py** que controla la ejecución programada, **email_service.py** que gestiona el envío de notificaciones, y **alert_service.py** que implements el sistema de alertas.

2.3 Flujo de Trabajo Automatizado

El sistema ejecuta un flujo de trabajo definido que se repite según la programación establecida. Comprender este flujo es fundamental para entender cómo funciona el sistema y qué papel juega cada componente en el proceso general.

El flujo comienza con la carga de la configuración general del sistema desde el archivo config.json, que contiene todos los parámetros operativos incluyendo objetivos de venta, secciones activas y rutas de archivos.

A continuación, se carga el estado persistido del sistema desde el archivo state.json, que contiene información sobre las ejecuciones anteriores, el stock acumulado y otras métricas históricas necesarias para los cálculos.

El sistema determina entonces qué semana debe procesar, calculando automáticamente el número de semana del año en función de la fecha actual y comparando con las semanas ya procesadas almacenadas en el estado.

Con esta información, el sistema procede a leer los archivos de datos de entrada, incluyendo el histórico de ventas, los costes de artículos, la clasificación ABC actual y cualquier otro dato necesario para los cálculos.

El motor de forecast procesa todos estos datos aplicando los algoritmos definidos, los objetivos de venta semanales, los factores de corrección y todas las reglas de negocio configuradas, generando como resultado las cantidades de pedido sugeridas para cada artículo.

Finalmente, el generador de pedidos crea los archivos Excel de salida con los pedidos calculados, aplicando formatos profesionales y organizando la información por secciones y categorías. El sistema actualiza el archivo de estado con la información de la ejecución actual y envía notificaciones por email si está configurado.

---

3. REQUISITOS DEL SISTEMA

3.1 Requisitos de Hardware

El sistema SPA tiene requisitos de hardware modestos, ya que está diseñado para ejecutarse en un ordenador personal convencional. Sin embargo, es importante cumplir con ciertos mínimos para garantizar un funcionamiento adecuado y tiempos de procesamiento razonables.

Se recomienda disponer de al menos 4 GB de memoria RAM para asegurar que el sistema pueda manejar los archivos de datos sin problemas de memoria. Aunque el sistema puede funcionar con menos memoria, los tiempos de procesamiento pueden aumentar significativamente cuando se procesan grandes volúmenes de datos.

En cuanto al espacio en disco, se recomienda disponer de al menos 2 GB de espacio disponible para almacenar el propio sistema, Python portable, las dependencias instaladas y los archivos de datos y salida que se generan con el tiempo. Es importante considerar que los archivos de salida pueden acumularse, por lo que se recomienda realizar limpiezas periódicas de archivos antiguos.

El procesador debe ser compatible con arquitecturas x64 (64 bits) ya que el sistema utiliza Python para arquitectura de 64 bits. Cualquier procesador moderno Intel o AMD de doble núcleo o superior será más que suficiente para las necesidades del sistema.

3.2 Requisitos de Software

El sistema está diseñado específicamente para ejecutarse en sistemas operativos Windows, concretamente en versiones de Windows 10 o Windows 11. No es compatible con sistemas operativos Linux o macOS sin modificaciones significativas.

El sistema incluye su propia instalación de Python portable, por lo que no es necesario tener Python preinstalado en el sistema. Esta característica facilita enormemente la instalación y evita conflictos con otras versiones de Python que puedan estar instaladas.

Es necesario disponer de conexión a internet durante la primera ejecución del sistema para que el instalador pueda descargar Python y las dependencias necesarias. Una vez completada la instalación inicial, el sistema puede funcionar sin conexión a internet, aunque ciertas funcionalidades como el envío de correos electrónicos o la verificación de actualizaciones requerirán conectividad.

Se requiere Microsoft Excel o una alternativa compatible como LibreOffice para poder abrir y visualizar los archivos de entrada y salida que genera el sistema. Los archivos se generan en formato Excel (.xlsx).

3.3 Requisitos de Red e Internet

Para el funcionamiento básico del sistema no se requiere acceso a red local ni a internet, salvo durante el proceso de instalación inicial. Sin embargo, existen funcionalidades opcionales que sí requieren conectividad.

El servicio de envío de correos electrónicos requiere acceso a un servidor SMTP para poder enviar las notificaciones y informes generados. La configuración del servidor de correo se realiza en el archivo de configuración y debe especificarse un servidor SMTP válido con las credenciales apropiadas.

Para recibir soporte técnico o actualizaciones del sistema puede ser necesario disponer de acceso a internet para comunicarse con el equipo de desarrollo o descargar nuevas versiones del software.

3.4 Permisos Necesarios

La instalación y ejecución del sistema requiere ciertos permisos en el sistema operativo Windows. Es importante asegurarse de contar con los permisos necesarios antes de iniciar la instalación.

Se requieren permisos de administrador para ejecutar el instalador SPA.exe, ya que este debe crear las tareas programadas en el Programador de tareas de Windows y potencialmente instalar Python en una ubicación del sistema.

El directorio donde se instale el sistema debe tener permisos de escritura para que el software pueda crear archivos de configuración, logs, datos de salida y cualquier otro archivo necesario para su funcionamiento.

Si se configura el sistema para enviar correos electrónicos, las credenciales del servidor SMTP deben estar correctamente configuradas y el equipo debe tener acceso al servidor de correo designado.

3.5 Compatibilidad con Sistemas Operativos

El sistema está diseñado específicamente para entornos Windows de 64 bits. Las versiones compatibles incluyen Windows 10 (todas las versiones) y Windows 11 (todas las versiones).

No es compatible directamente con sistemas Linux o macOS. Sin embargo, dado que el código está escrito en Python, podría adaptarse para estos sistemas operativos con modificaciones en los paths y en algunos aspectos específicos de Windows como las tareas programadas y el registro de Windows.

Las máquinas virtuales Windows que se ejecuten en entornos de virtualización (VMware, VirtualBox, Hyper-V) son totalmente compatibles siempre que cumplan con los demás requisitos de hardware y software.

---

4. ARCHIVOS DE ENTRADA DEL SISTEMA

4.1 Archivos Necesarios para el Funcionamiento

El sistema SPA requiere varios archivos de datos de entrada para poder funcionar correctamente. Estos archivos contienen la información histórica y actual que el sistema necesita para realizar los cálculos de pedidos. Es fundamental que todos estos archivos estén presentes en la ubicación correcta y con el formato adecuado.

Los archivos de entrada principales que el sistema necesita son los siguientes. En primer lugar, el archivo SPA_ventas.xlsx que contiene el histórico de ventas de todos los artículos del vivero. Este archivo es fundamental ya que constituye la base para calcular las proyecciones de venta futuras y determinar las cantidades óptimas de pedido.

En segundo lugar, el archivo SPA_coste.xlsx que almacena la información de costes unitarios, precios de venta, proveedores y demás datos comerciales de cada artículo. Este archivo se utiliza para calcular el margen de beneficio y determinar la clasificación ABC de los artículos.

En tercer lugar, el archivo SPA_stock_actual.xlsx que refleja el inventario disponible en el momento de ejecutar el sistema. Este dato es esencial para calcular las cantidades de pedido, ya que el sistema debe considerar el stock existente para evitar sobrepedidos.

Finalmente, los archivos CLASIFICACION_ABC+D_*.xlsx que contienen la clasificación de artículos por categoría. Estos archivos se generan periódicamente por el sistema de clasificación y son utilizados por el módulo de pedidos para aplicar los diferentes pesos según la categoría del artículo.

4.2 Origen y Obtención de los Archivos de Entrada

Todos los archivos de entrada que necesita el sistema deben obtenerse del sistema de gestión del vivero o del programa de ventas utilizado. Estos archivos no son generados por el sistema SPA, sino que deben proporcionarse como entrada desde fuentes externas.

4.2.1 SPA_ventas.xlsx — Histórico de Ventas

Este archivo debe exportarse desde el programa de gestión del vivero o desde el sistema de punto de venta. Debe contener registros de todas las ventas realizadas, incluyendo como mínimo la identificación del artículo, la fecha de venta, la cantidad vendida y el importe de la venta.

La estructura típica de este archivo incluye columnas como Código de artículo, Nombre del artículo, Fecha de venta, Cantidad, Importe, Sección a la que pertenece el artículo, y otras columnas adicionales que el sistema puede ignorar si no son relevantes para los cálculos.

La frecuencia de actualización de este archivo depende del uso del sistema. Para la generación semanal de pedidos, es suficiente con actualizar este archivo una vez por semana, antes de cada ejecución del sistema. El sistema automáticamente filtra los datos relevantes según el período que esté procesando.

4.2.2 SPA_coste.xlsx — Costes y Precios

Este archivo contiene la información comercial de cada artículo, incluyendo el coste unitario de compra, el precio de venta recomendado, el proveedor principal, el código de proveedor del artículo, y otros datos de interés para la gestión de compras.

Es importante mantener este archivo actualizado con los últimos precios y costes acordados con los proveedores. Cualquier cambio en los costes de los artículos debe reflejarse en este archivo para que los cálculos de pedidos sean precisos.

La estructura típica incluye columnas como Código de artículo, Nombre del artículo, Coste unitario, Precio de venta, Proveedor principal, Código del proveedor, Unidad de medida, y otras columnas de información comercial.

4.2.3 SPA_stock_actual.xlsx — Stock Actual

Este archivo refleja el inventario disponible en el almacén del vivero en el momento de su lectura. Debe contener la cantidad actual de cada artículo en stock, información esencial para calcular las cantidades de pedido.

El sistema utiliza estos datos para evitar sobrepedir artículos que ya tengan stock suficiente, y para calcular los pedidos de reposición necesarios para mantener niveles de inventario adecuados.

La estructura típica incluye columnas como Código de artículo, Nombre del artículo, Cantidad en stock, Ubicación en almacén, y posiblemente datos adicionales como stock mínimo de seguridad o punto de reorden.

4.2.4 CLASIFICACION_ABC+D_*.xlsx — Clasificación de Artículos

Estos archivos son generados automáticamente por el sistema de clasificación ABC y no requieren obtención manual. Contienen la clasificación de cada artículo en las categorías A, B, C o D según su importancia relativa para el negocio.

El sistema lee automáticamente el archivo de clasificación más reciente para el período que se está procesando, aplicando los pesos correspondientes a cada categoría en los cálculos de pedido.

4.2.5 SPA_compras.xlsx — Datos de Compras

Este archivo se utiliza específicamente para el módulo de clasificación ABC y contiene información sobre las compras realizadas durante el período de análisis. Es necesario para calcular la rotación de los artículos y determinar su clasificación.

Debe contener datos como código del artículo, fecha de compra, cantidad comprada, coste unitario, proveedor, y otros datos relacionados con las transacciones de compra.

4.2.6 encargados.xlsx — Encargados de Secciones

Este archivo contiene la información de los responsables de cada sección del vivero, incluyendo sus datos de contacto como correo electrónico. El sistema utiliza esta información para enviar automáticamente los informes de clasificación ABC a cada encargado.

La estructura típica incluye columnas como Código de sección, Nombre de sección, Nombre del encargado, Correo electrónico, Teléfono de contacto, y otras informações relevantes.

4.3 Actualización Anual de Archivos

Determinados archivos de entrada requieren actualización periódica, normalmente de manera anual, para mantener la precisión de los cálculos del sistema. A continuación se detallan estos archivos y los datos específicos que deben actualizarse.

4.3.1 Archivos que Requieren Actualización Anual

Los archivos que típicamente requieren actualización anual son los siguientes. Los archivos de ventas (SPA_ventas.xlsx) se actualizan continuamente a lo largo del año, pero al comenzar un nuevo año es necesario asegurarse de que el archivo contenga los datos históricos completos del año anterior para los análisis de clasificación ABC.

Los archivos de costes (SPA_coste.xlsx) deben revisarse al inicio de cada año para actualizar cualquier cambio en los precios de proveedores, nuevas tarifas acordadas, o modificaciones en las condiciones comerciales.

El archivo de classification ABC (CLASIFICACION_ABC+D_*.xlsx) se genera automáticamente por el sistema, pero los parámetros utilizados para su cálculo pueden requerir revisión anual.

El archivo de encargados (encargados.xlsx) debe actualizarse siempre que haya cambios en la organización del personal del vivero.

4.3.2 Datos Específicos de Cada Archivo

A continuación se detallan los datos específicos que contiene cada archivo de entrada principal.

SPA_ventas.xlsx contiene los siguientes datos principales: Código de artículo (identificador único del producto), Nombre del artículo, Fecha de venta (en formato date), Cantidad vendidas (número de unidades), Importe total (valor monetario de la venta), Sección (categoría a la que pertenece el artículo), Familia (agrupación adicional del artículo), yany other columns that may be present in the source system.

SPA_coste.xlsx incluye: Código de artículo, Nombre del artículo, Coste unitario (precio de compra por unidad), Precio de venta (precio recomendado), Proveedor (nombre del proveedor principal), Código proveedor (código del artículo en el proveedor), Unidad de medida (kg, unidad, caja, etc.), IVA aplicabel, y otros datos comerciales relevantes.

SPA_stock_actual.xlsx contiene: Código de artículo, Nombre del artículo, Cantidad actual (stock disponible), Ubicación (posición en el almacén), Stock mínimo de seguridad (opcional), Punto de reorden (opcional), y datos adicionales de inventario.

CLASIFICACION_ABC+D_*.xlsx incluye: Código de artículo, Nombre del artículo, Sección, Clasificación (A, B, C o D), Ventas período actual, Ventas período anterior, Rotación, Margen, y datos de análisis adicionales.

4.4 Formato y Estructura de los Archivos de Entrada

El sistema está diseñado para ser flexible con respecto al formato exacto de los archivos de entrada, incorporando funciones de normalización que permiten detectar columnas aunque sus nombres tengan variaciones. Sin embargo, es importante conocer el formato esperado para evitar problemas.

Los archivos deben estar en formato Excel (.xlsx). No se admiten formatos antiguos como .xls ni formatos CSV directamente.

Los nombres de las hojas de cálculo deben ser los esperados por el sistema. Si el sistema busca datos en una hoja específica con un nombre determinado, este debe coincidir o debe modificarse la configuración del sistema.

Los datos deben estar en formato tabular, con la primera fila conteniendo los nombres de columnas y las filas siguientes conteniendo los datos.

El sistema normaliza automáticamente los nombres de columnas, eliminando acentos, puntuación y diferencias de mayúsculas/minúsculas para realizar búsquedas más flexibles. Por ejemplo, las columnas "Coste", "COSTE", "cósté" o "Cóste" serían todas reconocidas como la columna de coste.

4.5 Ubicación de los Archivos de Entrada

Por defecto, el sistema espera encontrar los archivos de entrada en la siguiente ubicación dentro del directorio del proyecto: /data/input/

Esta ubicación puede modificarse en el archivo de configuración config.json en la sección de rutas. El parámetro directorio_entrada permite especificar una ruta alternativa si se desea almacenar los archivos en una ubicación diferente.

Ejemplo de ubicación esperada:

```
SPA/
├── data/
│   ├── input/
│   │   ├── SPA_ventas.xlsx
│   │   ├── SPA_coste.xlsx
│   │   ├── SPA_stock_actual.xlsx
│   │   └── CLASIFICACION_ABC+D_*.xlsx
│   └── output/
│       ├── Pedidos/
│       └── Informes/
```

Es importante mantener esta estructura de carpetas o actualizar la configuración соответственно si se utilizan rutas personalizadas.

---

5. PERÍODOS DE ESTUDIO (P1, P2, P3, P4)

5.1 Definición y Propósito de los Períodos

El sistema SPA divide el año en cuatro períodos de estudio, denominados P1, P2, P3 y P4, cada uno correspondiente a una estación del año. Esta división permite realizar análisis específicos de cada temporada y adaptar los cálculos de pedido a las particularidades de cada época del año.

El propósito principal de esta división es reconocer que las ventas de un vivero varían significativamente según la estación del año. Por ejemplo, la demanda de plantas de temporada es mucho mayor en primavera que en invierno, mientras que los productos de fitosanitarios tienen su pico en los meses de verano.

Al dividir el año en períodos, el sistema puede aplicar objetivos de venta específicos para cada semana dentro del contexto de su período correspondiente, logrando así proyecciones más precisas que si se utilizara un objetivo uniforme para todo el año.

Además, los períodos facilitan la generación de informes y análisis específicos para cada parte del año, permitiendo comparar el rendimiento entre temporadas y detectar tendencias o anomalías.

5.2 Calendario de Períodos

Cada período del sistema corresponde a un rango específico de fechas dentro del año. A continuación se detalla el calendario completo de períodos.

5.2.1 Período P1 — Invierno (Enero-Febrero)

El período P1 corresponde a los meses de enero y febrero, cubriendo aproximadamente las semanas 1 a 8 del año. Este es el período de menor actividad para un vivero, caracterizado por bajas ventas de planta de temporada y productos de jardín.

Durante este período, los productos más vendidos suelen ser los de interior (debido a que la gente passa más tiempo en casa), los alimentos y accesorios para mascotas, y los productos de decoración.

Los objetivos de venta para este período son generalmente los más bajos del año, reflejando la menor actividad del sector. El sistema aplica estos objetivos reducidos al calcular los pedidos de este período.

5.2.2 Período P2 — Primavera (Marzo-Mayo)

El período P2 abarca desde marzo hasta mayo, cubriendo aproximadamente las semanas 9 a 22 del año. Este es el período de mayor crecimiento de ventas, especialmente para plantas de temporada, semillas y productos de jardín.

La primavera marca el inicio de la temporada alta para el vivero, con un incremento significativo en la demanda de plantas de exterior, semillas, tierra, abonos y herramientas de jardín.

Los objetivos de venta para P2 son progresivamente crecientes, alcanzando los niveles más altos hacia finales de mayo. El sistema está configurado para reflejar este crecimiento en los cálculos de pedido.

5.2.3 Período P3 — Verano (Junio-Agosto)

El período P3 corresponde a los meses de junio, julio y agosto, cubriendo aproximadamente las semanas 23 a 35 del año. Este período mantiene un nivel de actividad alto pero con algunas variaciones respecto a la primavera.

Durante el verano, los fitosanitarios y abonos tienen su momento de mayor demanda debido al cuidado de plantas de jardín. También es una época importante para la decoración exterior y los productos de riego.

Los objetivos de venta se mantienen elevados durante este período, aunque pueden experimentar un ligera disminución respecto a mayo debido al efecto del calor en las compras de plantas.

5.2.4 Período P4 — Otoño-Invierno (Septiembre-Diciembre)

El período P4 abarca desde septiembre hasta diciembre, cubriendo aproximadamente las semanas 36 a 53 del año. Este período comienza con un nivel de actividad moderado y va descendiendo hacia finales de año.

En otoño se produce una recuperación de las ventas de plantas de temporada de invierno y productos de decoración tanto interior como exterior. Noviembre y diciembre incluyen el período de navideño, que puede representar ventas adicionales en ciertos productos.

Los objetivos de venta para P4 comienzan siendo moderados en septiembre, aumentan ligeramente en octubre-noviembre, y luego disminuyen en diciembre, reflejando el final del año comercial.

5.3 Cómo Afectan los Períodos a los Cálculos

Los períodos influyen en los cálculos del sistema de varias maneras. En primer lugar, determinan qué objetivos de venta semanales se aplican. Cada semana del año tiene un objetivo específico según su número y el período al que pertenece.

En segundo lugar, los períodos afectan al análisis de clasificación ABC. Al ejecutar el sistema de clasificación, este filtra los datos del año anterior correspondientes al mismo período para realizar comparaciones homogéneas. Por ejemplo, los datos de P1 de 2025 se comparan con P1 de 2024.

En tercer lugar, ciertos parámetros de configuración pueden variar según el período. Por ejemplo, los factores de corrección de stock mínimo o los pesos de categoría pueden ajustarse estacionalmente.

Finalmente, los archivos de salida generados incluyen el período en su nombre para facilitar su identificación y archivo. Por ejemplo, los pedidos del período P2 incluirían esta información en su nomenclatura.

5.4 Archivos Específicos por Período

El sistema genera y utiliza ciertos archivos específicos para cada período. Es importante conocer estos archivos para comprender cómo funciona el sistema.

Los archivos de clasificación ABC se generan con el período en su nombre: CLASIFICACION_ABC+D_{SECCION}P1_2025.xlsx, CLASIFICACION_ABC+D_P2_2025.xlsx, etc.

Los archivos de stock también pueden ser específicos por período: SPA_stock_actual_P1.xlsx, SPA_stock_actual_P2.xlsx, etc., aunque el sistema puede funcionar con un único archivo de stock actualizado.

Los informes generados (presentaciones, análisis, etc.) incluyen el período en su nombre para facilitar su identificación: PRESENTACION_P1_2025.xlsx, INFORME_P2_2025.xlsx, etc.

5.5 Configuración de Períodos en el Sistema

Los períodos están configurados en el archivo config/config.json y pueden modificarse según las necesidades del negocio. La configuración define cómo se mapean las semanas del año a cada período.

La estructura de configuración de períodos en el archivo JSON tiene el siguiente aspecto:

```
{
  "periodos": {
    "P1": {
      "nombre": "Invierno",
      "semanas_inicio": 1,
      "semanas_fin": 8,
      "meses": ["Enero", "Febrero"]
    },
    "P2": {
      "nombre": "Primavera",
      "semanas_inicio": 9,
      "semanas_fin": 22,
      "meses": ["Marzo", "Abril", "Mayo"]
    },
    "P3": {
      "nombre": "Verano",
      "semanas_inicio": 23,
      "semanas_fin": 35,
      "meses": ["Junio", "Julio", "Agosto"]
    },
    "P4": {
      "nombre": "Otoño-Invierno",
      "semanas_inicio": 36,
      "semanas_fin": 53,
      "meses": ["Septiembre", "Octubre", "Noviembre", "Diciembre"]
    }
  }
}
```

Esta configuración puede ajustarse si el negocio requiere una definición diferente de los períodos, por ejemplo si la estacionalidad del vivero difiere del patrón general.

---

6. INSTALACIÓN

6.1 Preparación del Entorno

Antes de iniciar la instalación del sistema SPA, es necesario realizar una serie de preparaciones para asegurar una instalación exitosa.

En primer lugar, debe verificarse que el equipo cumpla con los requisitos mínimos de hardware y software descritos en el capítulo 3 de este manual. Es importante disponer de suficiente espacio en disco y de los permisos necesarios para instalar software.

En segundo lugar, debe decidirse la ubicación donde se instalará el sistema. Se recomienda utilizar una ruta que no contenga espacios ni caracteres especiales, como por ejemplo C:\Sistemas\SPA o D:\Vivero\SPA.

En tercer lugar, debe prepararse la estructura de carpetas necesaria para los datos de entrada. El sistema creará automáticamente las carpetas necesarias, pero es útil preparar los archivos de entrada antes de la primera ejecución.

Finalmente, debe asegurarse una conexión a internet activa durante la primera ejecución, ya que el instalador necesita descargar Python y las dependencias.

6.2 Estructura de Carpetas del Proyecto

Al clonar o descargar el proyecto, la estructura de carpetas debe ser la siguiente:

```
SPA/
├── Python/                  # Python portable (se crea tras primera
ejecución)
├── data/
│   ├── input/              # Archivos de entrada del sistema
│   ├── output/             # Archivos generados por el sistema
│   └── state.json          # Archivo de estado del sistema
├── src/                    # Código fuente del sistema
├── config/
│   └── config.json        # Archivo de configuración principal
├── logs/                   # Archivos de log del sistema
├── dist/                   # Archivos compilados (si aplica)
├── build/                  # Archivos de construcción (si aplica)
├── docs/                   # Documentación del proyecto
├── SPA.exe                 # Ejecutable principal del sistema
├── main.py                 # Script principal de pedidos
├── clasificacionABC.py     # Script de clasificación
├── INFORME.py             # Script de informes
├── PRESENTACION.py        # Script de presentaciones
├── instalar_todo.py       # Script de instalación
├── requirements.txt       # Dependencias de Python
└── README.md              # Archivo README del proyecto
```

Esta estructura puede扩充erse con archivos adicionales según las necesidades del sistema, pero es la estructura base que debe mantenerse.

6.3 Proceso de Instalación Paso a Paso

El proceso de instalación del sistema SPA es automatizado y requiere intervención mínima del usuario. A continuación se detallan los pasos a seguir.

Paso 1: Preparación del directorio

Crear el directorio donde se instalará el sistema y copiar todos los archivos del proyecto en él. Asegurarse de mantener la estructura de carpetas original.

Paso 2: Preparación de archivos de entrada

Copiar los archivos de datos de entrada (SPA_ventas.xlsx, SPA_coste.xlsx, etc.) en la carpeta data/input/. Crear esta carpeta si no existe.

Paso 3: Configuración inicial

Revisar y ajustar si es necesario el archivo config/config.json con los parámetros específicos del negocio, como objetivos de venta, secciones activas, etc.

Paso 4: Ejecución del instalador

Ejecutar el archivo SPA.exe como administrador. El instalador realizará las siguientes acciones automáticamente:

- Verificar si Python portable ya está instalado
- Si no está instalado, descargar e instalar Python 3.11.9 en la carpeta local
- Configurar pip (gestor de paquetes de Python)
- Instalar las dependencias necesarias (pandas, openpyxl, numpy, etc.)
- Crear las tareas programadas en Windows

Paso 5: Verificación de la instalación

Tras la ejecución del instalador, verificar que la instalación fue exitosa comprobando:

- La existencia de la carpeta Python con python.exe dentro
- La existencia de las tareas programadas en el Programador de tareas de Windows
- La capacidad de ejecutar el sistema manualmente

6.4 Primera Ejecución del Sistema

La primera ejecución del sistema después de la instalación es especialmente importante porque establece el estado inicial del sistema y verifica que todos los componentes funcionen correctamente.

Durante la primera ejecución, el sistema realizará las siguientes comprobaciones y acciones:

1. Verificación de la existencia de Python portable
2. Carga del archivo de configuración
3. Creación del archivo de estado inicial (state.json) si no existe
4. Verificación de la existencia de archivos de entrada necesarios
5. Si todo está correcto, ejecución del proceso de cálculo de pedidos

Si durante la primera ejecución se producen errores, el sistema mostrará mensajes de error específicos que indicarán qué componente está fallando. Consultar el capítulo de solución de problemas para resolver incidencias.

6.5 Verificación de la Instalación

Para verificar que la instalación se completó correctamente, pueden realizarse las siguientes comprobaciones.

En primer lugar, verificar la existencia de la carpeta Python en el directorio del proyecto. Esta carpeta debe contener python.exe y una estructura de carpetas con las bibliotecas de Python.

En segundo lugar, abrir el Programador de tareas de Windows (taskschd.msc) y verificar que existen las tareas programadas creadas por el instalador. Las tareas deben aparecer con nombres como Vivero_Main_Pedidos_Semanales, Vivero_ClasificacionABC_P1, etc.

En tercer lugar, intentar ejecutar el sistema manualmente desde la línea de comandos. Abrir una terminal, navegar al directorio del proyecto y ejecutar: python main.py --status

Este comando debe mostrar el estado actual del sistema sin producir errores.

Finalmente, revisar los archivos de log creados en la carpeta logs/ para verificar que no hay errores registrados.

6.6 Instalación en Múltiples Equipos

Si es necesario instalar el sistema en varios equipos, el proceso puede realizarse de manera más eficiente siguiendo ciertos procedimientos.

La forma más sencilla es instalar el sistema en un equipo que servirá como modelo, configurarlo completamente con todos los archivos de entrada y parámetros de configuración, y luego copiar toda la carpeta del proyecto a los demás equipos.

Al copiar la carpeta a nuevos equipos, deben considerarse los siguientes aspectos. Las tareas programadas deben recrearse en cada nuevo equipo ejecutando SPA.exe nuevamente. Las rutas absolutas en la configuración pueden requerir ajustes si los equipos tienen estructuras de carpetas diferentes.

También es importante mantener una estrategia de sincronización de archivos de entrada si varios equipos van a trabajar con los mismos datos. Puede optarse por almacenar los archivos de entrada en una ubicación de red compartida o por actualizar manualmente los archivos en cada equipo.

---

7. CONFIGURACIÓN

7.1 Archivo de Configuración Principal (config.json)

El archivo config.json es el centro de configuración del sistema SPA. Contiene todos los parámetros que determinan cómo funciona el sistema, desde los objetivos de venta hasta las rutas de archivos.

Este archivo se encuentra en la ruta config/config.json dentro del directorio del proyecto y utiliza formato JSON (JavaScript Object Notation), un formato de texto legible que facilita la edición manual.

A continuación se describen las principales secciones del archivo de configuración.

Sección "notas": Contiene información descriptiva sobre el propósito del archivo de configuración y su versión.

Sección "secciones": Define los objetivos de venta semanales para cada sección del vivero. Cada sección tiene su propia subsección con los objetivos para cada semana del año (semanas 1 a 53).

Sección "parametros": Contiene los parámetros generales de cálculo, como el objetivo de crecimiento (porcentaje de incremento sobre ventas anteriores), el stock mínimo como porcentaje, y los pesos asignados a cada categoría ABC.

Sección "festivos": Define factores de ajuste para semanas con festividades o eventos especiales. Por ejemplo, semana 14 (Semana Santa) puede tener un factor de 0.25 (25% de ventas normales).

Sección "secciones_activas": Lista de secciones que serán procesadas por el sistema. Las secciones no incluidas en esta lista serán ignoradas durante la generación de pedidos.

Sección "horario_ejecucion": Esta sección definía anteriormente el día y hora de ejecución automática. Actualmente ya no se utiliza en el código del sistema, ya que el control de horario se gestiona exclusivamente a través de las tareas programadas del sistema operativo (Windows Task Scheduler). Se mantiene en el archivo por compatibilidad y para referencia histórica.

Sección "rutas": Especifica las rutas de directorios para datos de entrada, salida, estado y logs. Permite personalizar la ubicación de los archivos.

Sección "archivos_entrada": Define los nombres esperados de los archivos de entrada, permitiendo personalización si los archivos fuente tienen nombres diferentes.

Sección "parametros_correccion": Configura los parámetros del sistema de corrección que ajusta los pedidos según el stock actual.

Sección "formato_salida": Define el formato de los nombres de archivos de salida.

Sección "env_email": Configura los parámetros para el envío de notificaciones por correo electrónico.

7.2 Configuración de Secciones y Objetivos de Venta

Las secciones del vivero y sus objetivos de venta semanales constituyen la configuración más importante del sistema, ya que determinan directamente las cantidades de pedido calculadas.

Cada sección del vivero debe estar definida en la sección "secciones" del archivo de configuración. La estructura incluye:

```
"secciones": {
    "maf": {
        "descripcion": "Planta de temporada y floristeria",
        "objetivos_semanales": {
            "1": 224.49,
            "2": 410.74,
            ...
        }
    },
    "interior": {
        "descripcion": "Plantas de interior",
        "objetivos_semanales": {
            "1": 907.62,
            ...
        }
    }
}
```

Los objetivos de venta semanales representan el valor de ventas esperado para cada semana del año. Estos objetivos deben establecerse basándose en el historial de ventas del vivero y en las proyecciones de crecimiento.

Para modificar los objetivos de venta, se debe editar el archivo config.json directamente, cambiando los valores numéricos para cada semana de cada sección. Es importante mantener el formato JSON válido (comas entre valores, etc.).

7.3 Parámetros de Cálculo de Pedidos

La sección "parametros" del archivo de configuración controla los algoritmos de cálculo utilizados por el sistema. Estos parámetros determinan cómo se transforman los datos de entrada en pedidos finales.

```
"parametros": {
    "objetivo_crecimiento": 0.05,
    "stock_minimo_porcentaje": 0.30,
    "pesos_categoria": {
        "A": 1.0,
        "B": 0.8,
        "C": 0.6,
        "D": 0.0
    }
}
```

El parámetro **objetivo_crecimiento** (0.05 = 5%) define el porcentaje de crecimiento que se espera sobre las ventas del período anterior. Este factor se aplica para incrementar los pedidos y anticipar el crecimiento del negocio.

El parámetro **stock_minimo_porcentaje** (0.30 = 30%) define el porcentaje del stock objetivo que debe mantenerse como mínimo. Este factor asegura que nunca se quede el almacén sin stock suficiente.

Los **pesos_categoria** definen multiplicadores que se aplican según la clasificación ABC del artículo. Los artículos categoría A (los más importantes) reciben el peso completo (1.0), mientras que los de categoría D (los menos importantes) reciben peso cero, lo que significa que no se pedidos automáticamente.

7.4 Configuración de Festivos y Períodos Especiales

La sección "festivos" permite definir factores de ajuste para semanas que tienen un comportamiento diferente al normal, como semanas con festividades patronales, vacaciones, u otros eventos que afectan a las ventas.

```
"festivos": {
    "14": 0.25,
    "18": 0.00,
    "22": 0.00
}
```

En este ejemplo, la semana 14 tiene un factor de 0.25 (25% de las ventas normales), la semana 18 tiene factor 0.00 (cerrado, sin ventas), y la semana 22 tiene factor 0.00. Estos valores indican al sistema que debe ajustar los objetivos de venta para esas semanas según el factor especificado.

Para modificar esta configuración, agregar o modificar las entradas en la sección "festivos" siguiendo el formato "numero_de_semana": factor. El factor debe ser un número entre 0.0 (sin ventas) y 1.0 (ventas normales), pudiendo tomar valores intermedios como 0.5 (50% de ventas) o 1.5 (150%, ventas incrementadas).

7.5 Rutas de Archivos de Entrada y Salida

La sección "rutas" del archivo de configuración permite personalizar las ubicaciones donde el sistema busca archivos de entrada y donde genera archivos de salida.

```
"rutas": {
    "directorio_base": ".",
    "directorio_entrada": null,
    "directorio_salida": null,
    "directorio_estado": "./data",
    "directorio_logs": "./logs",
    "archivo_estado": "state.json",
    "archivo_config": "config.json"
}
```

El parámetro **directorio_base** establece el directorio raíz del proyecto. Los demás directorios se especifican de forma relativa a este directorio.

Los parámetros **directorio_entrada** y **directorio_salida** están configurados como null, lo que significa que el sistema utilizará las rutas por defecto (data/input y data/output). Si se especifican rutas personalizadas, el sistema utilizará esas rutas en lugar de las predeterminadas.

Los directorios de **estado** y **logs** indican dónde se almacenan los archivos de persistencia del sistema y los archivos de registro respectivamente.

7.6 Programación de Horarios de Ejecución

La sección "horario_ejecucion" ya no se utiliza en el código del sistema. El horario de ejecución se controla exclusivamente a través de las tareas programadas del sistema operativo Windows.

La sección "horario_ejecucion" ya no se utiliza en el código. El sistema ejecuta directamente cuando se llama, siendo la tarea programada del sistema operativo la que controla el horario de ejecución.

7.7 Configuración de Parámetros de Corrección

La sección "parametros_correccion" controla el sistema de corrección que ajusta los pedidos calculados según el stock actual disponible.

```
"parametros_correccion": {
    "habilitar_correccion": true,
    "stock_minimo_por_categoria": {
        "A": 1.5,
        "B": 1.0,
        "C": 0.5,
        "D": 0.0
    },
    "umbral_alerta_stock": 5,
    "permitir_pedidos_negativos": false
}
```

El parámetro **habilitar_correccion** activa o desactiva el sistema de corrección. Cuando está habilitado, el sistema ajustará los pedidos para evitar roturas de stock.

Los valores de **stock_minimo_por_categoria** definen cuántas semanas de stock mínimo se mantienen para cada categoría ABC. Los artículos categoría A mantienen 1.5 semanas, los B mantienen 1 semana, los C mantienen 0.5 semanas, y los D no mantienen stock mínimo.

El **umbral_alerta_stock** define la cantidad mínima de stock que genera una alerta cuando un artículo está por debajo de este nivel.

El parámetro **permitir_pedidos_negativos** determina si se permiten pedidos negativos (devoluciones o ajustes), aunque normalmente debe mantenerse en false.

7.8 Configuración de Alertas y Notificaciones

El sistema incluye un módulo de alertas que puede enviar notificaciones por correo electrónico cuando se producen eventos significativos o errores.

La configuración de correo electrónico se realiza en la sección "env_email" del archivo de configuración:

```
"env_email": {
    "password_var": "EMAIL_PASSWORD",
    "destinatario_alertas": "ivan.delgado@viveverde.es"
}
```

El parámetro **password_var** indica el nombre de la variable de entorno que contiene la contraseña del correo electrónico utilizado para enviar las notificaciones. Por razones de seguridad, la contraseña no se almacena directamente en el archivo de configuración.

El parámetro **destinatario_alertas** especifica la dirección de correo electrónico que recibirá las notificaciones y alertas del sistema.

Para configurar el sistema de correo electrónico, también deben establecerse las variables de entorno correspondientes con la configuración del servidor SMTP. Consultar la documentación técnica para más detalles sobre la configuración completa del sistema de correo.

---

8. TAREAS PROGRAMADAS

8.1 Concepto de Tareas Programadas

Las tareas programadas son funcionalidades del sistema operativo Windows que permiten ejecutar programas de manera automática en momentos específicos, sin necesidad de intervención manual. El sistema SPA utiliza las tareas programadas de Windows para ejecutar sus diferentes módulos de forma periódica.

Al ejecutar el instalador SPA.exe, el sistema automáticamente crea un conjunto de tareas programadas en Windows que ejecutan los diferentes scripts del sistema en los horarios predefinidos. Estas tareas garantizan que los procesos del sistema se ejecuten puntualmente según el calendario establecido.

Las tareas programadas survivecen al cierre de sesión del usuario y se ejecutarán incluso si el equipo está parado pero encendido en el momento programado. Esto asegura que ninguna ejecución se pierda por ausencia del usuario.

8.2 Tareas Programadas Generadas por SPA.exe

Al ejecutar el instalador SPA.exe, se crean las siguientes tareas programadas en el sistema. Cada tarea corresponde a un script específico del sistema y tiene un horario de ejecución predeterminado.

Tareas Semanales (se ejecutan cada semana):

- Vivero_Main_Pedidos_Semanales: Genera los pedidos de compra semanales
- Vivero_Informe_Compras_Sin_Autorizacion: Genera el informe de compras sin autorización
- Vivero_Informe_Articulos_No_Comprados: Genera el informe de artículos no comprados
- Vivero_Analisis_Categoria_CD: Genera el análisis de categoría CD

Tareas Mensuales (se ejecutan cada mes específico):

- Vivero_ClasificacionABC_P1: Genera la clasificación ABC del período P1 (Enero)
- Vivero_ClasificacionABC_P2: Genera la clasificación ABC del período P2 (Febrero)
- Vivero_ClasificacionABC_P3: Genera la clasificación ABC del período P3 (Mayo)
- Vivero_ClasificacionABC_P4: Genera la clasificación ABC del período P4 (Agosto)
- Vivero_Presentacion_Enero: Genera la presentación del período P1
- Vivero_Presentacion_Febrero: Genera la presentación del período P2
- Vivero_Presentacion_Mayo: Genera la presentación del período P3
- Vivero_Presentacion_Agosto: Genera la presentación del período P4
- Vivero_Informe_Enero: Genera el informe del período P1
- Vivero_Informe_Febrero: Genera el informe del período P2
- Vivero_Informe_Mayo: Genera el informe del período P3
- Vivero_Informe_Agosto: Genera el informe del período P4

8.3 Detalle de Cada Tarea Programada

A continuación se describe en detalle cada una de las tareas programadas creadas por el sistema.

8.3.1 Vivero_Main_Pedidos_Semanales

Esta es la tarea principal del sistema, responsable de generar los pedidos de compra semanales. Se ejecuta cada jueves a las 21:50 horas.

Script asociado: main.py

Propósito: Calcular y generar los pedidos de compra para todas las secciones activas del vivero basándose en los objetivos de venta semanales, el histórico de ventas, la clasificación ABC y el stock actual.

Frecuencia: Semanal

Día de ejecución: Jueves

Hora de ejecución: 21:50

Archivos de salida generados: Pedidos en formato Excel en la carpeta data/output/Pedidos/

8.3.2 Vivero_Informe_Compras_Sin_Autorizacion

Esta tarea genera un informe que identifica las compras realizadas que no están autorizadas según los criterios definidos.

Script asociado: informe_compras_sin_autorizacion.py

Propósito: Detectar y reportar compras que se han realizado fuera de los cauces normales de autorización, permitiendo revisar y regularizar estas situaciones.

Frecuencia: Semanal

Día de ejecución: Jueves

Hora de ejecución: 21:10

Archivos de salida generados: Informe en formato Excel

8.3.3 Vivero_Informe_Articulos_No_Comprados

Esta tarea genera un informe de artículos que aparecen en las listas de pedido pero que no se han comprado.

Script asociado: Informe_articulos_no_comprados.py

Propósito: Identificar desalineamientos entre los pedidos generados y las compras reales, permitiendo tomar acciones correctivas.

Frecuencia: Semanal

Día de ejecución: Jueves

Hora de ejecución: 21:10

Archivos de salida generados: Informe en formato Excel

8.3.4 Vivero_Analisis_Categoria_CD

Esta tarea realiza un análisis específico de los artículos categorizados como C y D en la clasificación ABC.

Script asociado: analisis_categoria_cd.py

Propósito: Analizar en profundidad los artículos de baja rotación (categorías C y D) para tomar decisiones sobre si mantenerlos, promocionarlos o eliminarlos del catálogo.

Frecuencia: Semanal

Día de ejecución: Jueves

Hora de ejecución: 21:10

Archivos de salida generados: Análisis en formato Excel

8.3.5 Vivero_ClasificacionABC (P1, P2, P3, P4)

Estas tareas generan la clasificación ABC+D de todos los artículos del vivero para cada período.

Script asociado: clasificacionABC.py con parámetros específicos

Propósito: Calcular la clasificación de artículos según su importancia relativa, utilizada luego por el sistema de pedidos para aplicar los pesos correspondientes.

Frecuencia: Mensual (período específico)

Día de ejecución: Según el período

- P1: Día 1 de Enero a las 12:00
- P2: Día 1 de Febrero a las 09:00

- P3: Día 1 de Mayo a las 09:00
- P4: Día 1 de Agosto a las 09:00

Archivos de salida generados: Archivos de clasificación por sección en formato Excel

8.3.6 Vivero_Presentacion (Enero, Febrero, Mayo, Agosto)

Estas tareas generan presentaciones visuales con los datos y análisis de cada período.

Script asociado: PRESENTACION.py

Propósito: Crear documentos de presentación con los datos más relevantes del período, útiles para reuniones y presentaciones gerenciales.

Frecuencia: Mensual (período específico)

Día de ejecución: Según el período

- Presentación Enero: Día 1 de Enero a las 12:30
- Presentación Febrero: Día 1 de Febrero a las 09:30
- Presentación Mayo: Día 1 de Mayo a las 09:30
- Presentación Agosto: Día 1 de Agosto a las 09:30

Archivos de salida generados: Presentaciones en formato Excel

8.3.7 Vivero_Informe (Enero, Febrero, Mayo, Agosto)

Estas tareas generan informes detallados del análisis ABC+D para cada período.

Script asociado: INFORME.py

Propósito: Crear informes HTML completos con el análisis ABC+D de todos los artículos del vivero, incluyendo métricas detalladas y recomendaciones.

Frecuencia: Mensual (período específico)

Día de ejecución: Según el período

- Informe Enero: Día 1 de Enero a las 12:30
- Informe Febrero: Día 1 de Febrero a las 09:30
- Informe Mayo: Día 1 de Mayo a las 09:30
- Informe Agosto: Día 1 de Agosto a las 09:30

Archivos de salida generados: Informes en formato HTML y Excel

8.4 Configuración Detallada de Cada Tarea

Cada tarea programada tiene una configuración específica que determina cuándo y cómo se ejecuta. Esta configuración se establece durante la instalación y puede modificarse posteriormente.

8.4.1 Frecuencia de Ejecución

La frecuencia de ejecución define cada cuánto tiempo se ejecuta la tarea. El sistema utiliza los siguientes tipos de frecuencia:

- **Semanal**: La tarea se ejecuta una vez por semana, en el día y hora especificados.
- **Mensual**: La tarea se ejecuta una vez al mes, en el día y mes especificados.

8.4.2 Día de la Semana

Para las tareas semanales, el día de la semana determina cuándo se ejecuta la tarea. El sistema está configurado por defecto para ejecutar las tareas semanales los jueves, que es cuando normalmente se preparaban los pedidos de forma manual.

8.4.3 Hora de Ejecución

La hora de ejecución define el momento específico del día en que la tarea se pone en marcha. El sistema utiliza horas de la tarde-noche para evitar afectar el trabajo normal durante las horas laborales.

La hora de la tarea principal (Pedidos Semanales) está configurada a las 21:50, que es cuando presumiblemente el día laboral ha terminado y los datos de ventas del día están completos.

Las demás tareas semanales se ejecutan a las 21:10, 20 minutos antes que la tarea principal, para dar tiempo a que terminen antes de medianoche si hubiera algún problema.

Las tareas mensuales se ejecutan por la mañana, entre las 09:00 y las 12:30, aprovechando las horas de menor actividad del sistema.

8.4.4 Script Asociado

Cada tarea programada ejecuta un script Python específico. A continuación se presenta una tabla resumen de los scripts asociados a cada tarea:

Nombre de Tarea	Script Asociado	Descripción
Vivero_Main_Pedidos_Semanales	main.py	Generación de pedidos semanales
Vivero_Informe_Compras_Sin_Autorizacion	informe_compras_sin_autorizacion.py	Informe de compras no autorizadas
Vivero_Informe_Articulos_No_Comprados	Informe_articulos_no_comprados.py	Informe de artículos pendientes

Nombre de Tarea	Script Asociado	Descripción
Vivero_Analisis_Categoria_CD	analisis_categoria_cd.py	Análisis de categorías C y D
Vivero_ClasificacionABC_P*	clasificacionABC.py -P*	Clasificación ABC del período
Vivero_Presentacion_*	PRESENTACION.py	Presentación del período
Vivero_Informe_*	INFORME.py	Informe del período

8.5 Verificar Tareas Programadas

Para verificar que las tareas programadas están correctamente configuradas en Windows, seguir estos pasos:

1. Abrir el Programador de tareas: Pulsar Win + R y escribir "taskschd.msc", luego presionar Enter.
2. En la ventana del Programador de tareas, expandir la opción "Biblioteca del Programador de tareas" en el panel izquierdo.
3. Buscar las tareas que empiezan por "Vivero_" en la lista de tareas.
4. Al hacer clic en una tarea, en el panel derecho se muestra información detallada incluyendo el estado (habilitada/deshabilitada), la última ejecución, la próxima ejecución, y el historial de ejecuciones.

Si alguna tarea no aparece o está deshabilitada, puede recrearse ejecutando nuevamente el instalador SPA.exe con## 8.6 Modificar o Eliminar Tareas

Para modificar una tarea programada existente, seguir permisos de administrador.

estos pasos:

1. Abrir el Programador de tareas (taskschd.msc).
2. Localizar la tarea a modificar en la lista.
3. Hacer doble clic en la tarea para abrir sus propiedades.
4. En la pestaña "General" pueden modificarse opciones como la cuenta de usuario que ejecuta la tarea.
5. En la pestaña "Desencadenadores" puede modificarse la programación (día, hora, frecuencia).
6. En la pestaña "Acciones" puede modificarse el programa o script que se ejecuta.

Para eliminar una tarea:
1. Abrir el Programador de tareas.

2. Seleccionar la tarea a eliminar.
3. Hacer clic en "Eliminar" en el panel derecho o presionar la tecla Supr.
4. Confirmar la eliminación.

Nota: Si se eliminan tareas, pueden recrearse ejecutando nuevamente SPA.exe.

8.7 Ejecución Manual de Tareas

Además de la ejecución automática según la programación, las tareas pueden ejecutarse manualmente en cualquier momento.

Para ejecutar una tarea manualmente:

1. Abrir el Programador de tareas.
2. Localizar la tarea deseada en la lista.
3. Hacer clic derecho sobre la tarea y seleccionar "Ejecutar".

Alternativamente, puede ejecutarse directamente el script Python asociado desde la línea de comandos. Por ejemplo, para ejecutar la generación de pedidos manualmente:

```
cd C:\ruta\al\proyecto
Python\python.exe main.py
```

Para ejecutar un script con parámetros específicos, por ejemplo la clasificación ABC del período P2:

```
Python\python.exe clasificacionABC.py -P P2 -Y 2025
```

---

9. USO DEL SISTEMA

9.1 Ejecución Manual del Sistema

Aunque el sistema está diseñado para ejecutarse automáticamente mediante tareas programadas, también puede ejecutarse manualmente cuando sea necesario. La ejecución manual permite controlar exactamente qué proceso se ejecuta y cuándo.

Para ejecutar el sistema manualmente, abrir una terminal (símbolo del sistema o PowerShell), navegar al directorio del proyecto y ejecutar el comando correspondiente según el módulo deseado.

La sintaxis general para ejecutar cualquier script del sistema es:

```
Python\python.exe [script.py] [parámetros]
```

Donde [script.py] es el nombre del script a ejecutar y [parámetros] son los argumentos opcionales que modifican el comportamiento del script.

9.2 Parámetros de Línea de Comandos

El sistema soporta varios parámetros de línea de comandos que permiten modificar su comportamiento según las necesidades específicas de cada ejecución.

Parámetros generales del sistema:

- **--semana N**: Procesa una semana específica (N = número de semana). Útil para regenerar pedidos de semanas anteriores o para procesar fuera de secuencia.
- **--continuo**: Ejecuta el sistema en modo continuo. Después de ejecutar, en lugar de cerrarse, vuelve a esperar y ejecutar en la siguiente semana según la tarea programada del sistema operativo.
- **--status**: Muestra el estado actual del sistema sin ejecutar el proceso de generación de pedidos. Útil para verificar la configuración y el estado.
- **--reset**: Borra todo el historial y comienza desde cero. **Precaución:** Esta opción elimina todos los datos acumulados del sistema.
- **--verbose**: Activa el logging detallado para depuración. Muestra mensajes adicionales que ayudan a diagnosticar problemas.

Ejemplos de uso:

```
# Ver el estado del sistema
Python\python.exe main.py --status

# Procesar la semana 15 manualmente
Python\python.exe main.py --semana 15

# Ejecutar en modo continuo
Python\python.exe main.py --continuo

# Ejecutar con logging detallado
Python\python.exe main.py --verbose
```

9.3 Modo Forzado (Procesar Semana Específica)

El modo forzado permite procesar una semana específica sin importar la programación normal del sistema. Esto es útil cuando se necesita regenerar pedidos de una semana pasada o cuando se quiere procesar una semana futura anticipadamente.

Para usar el modo forzado, utilizar el parámetro --semana seguido del número de semana deseado:

```
Python\python.exe main.py --semana 15
```

Este comando procesará la semana 15 independientemente de cuál sea la semana actual del sistema.

El sistema determinará automáticamente qué datos históricos utilizar basándose en el número de semana especificado y generará los pedidos como si fuera la ejecución programada normal.

9.4 Modo Continuo (Espera Programadora)

El modo continuo hace que el sistema permanezca ejecutándose después de cada ejecución. En lugar de cerrarse, vuelve a esperar la siguiente semana para procesar los pedidos según la tarea programada del sistema operativo.

Para ejecutar en modo continuo:

```
Python\python.exe main.py --continuo
```

Este modo es útil cuando se quiere que el sistema funcione como un servicio, ejecutándose automáticamente cada semana sin necesidad de tareas programadas del sistema operativo.

Para detener un sistema en modo continuo, presionar Ctrl + C en la terminal.

9.5 Visualización del Estado del Sistema

El sistema mantiene un archivo de estado (state.json) que registra información sobre las ejecuciones realizadas, las semanas procesadas, el stock acumulado y otras métricas.

Para visualizar el estado actual del sistema sin ejecutar el proceso de generación de pedidos, utilizar el parámetro --status:

```
Python\python.exe main.py --status
```

El sistema mostrará información como:

- Versión del sistema
- Última fecha de ejecución
- Última semana procesada
- Estado de cada sección
- Métricas acumuladas
- Próxima ejecución programada

Esta funcionalidad es útil para verificar que el sistema está funcionando correctamente y para diagnosticar problemas.

9.6 Reset del Sistema

La opción de reset borra todos los datos acumulados del sistema, devolviéndolo a su estado inicial. Esta opción debe usarse con extrema precaución ya que elimina todo el histórico.

Para realizar un reset del sistema:

```
Python\python.exe main.py --reset
```

El sistema mostrará una advertencia antes de proceder con el reset. Confirmar la acción si realmente se desea eliminar todo el historial.

Después del reset, el sistema comenzará como si fuera la primera vez que se ejecuta, sin conocimiento de ejecuciones anteriores ni datos acumulados.

Casos de uso típicos del reset:

- Cambio de año fiscal: Al comenzar un nuevo año, puede ser apropiado reiniciar el estado.
- Problemas graves con los datos: Si los datos de estado se corrompen y no hay forma de recuperarlos.
- Reubicación del sistema: Al instalar el sistema en un nuevo equipo sin necesidad del histórico anterior.

---

10. MÓDULOS DEL SISTEMA

10.1 main.py — Motor de Pedidos Semanales

El script main.py es el módulo principal del sistema y coordina todo el proceso de generación de pedidos semanales. Este script integra todos los componentes del sistema para producir los pedidos de compra.

Funcionalidades principales:

- Carga de configuración desde config.json
- Gestión del estado del sistema
- Coordinación de la ejecución de todos los submódulos
- Control de la programación de ejecuciones
- Generación de notificaciones y alertas
- Logging de todas las operaciones

Uso típico:

```
Python\python.exe main.py
```

Parámetros disponibles:
- --semana N: Procesar semana específica
- --continuo: Modo continuo
- --status: Mostrar estado
- --reset: Reiniciar estado
- --verbose: Modo detallado

10.2 clasificacionABC.py — Clasificación de Artículos

El script clasificacionABC.py implementa el algoritmo de clasificación ABC+D para todos los artículos del catálogo del vivero. Este análisis clasifica los artículos según su importancia relativa basada en las ventas y el margen.

Funcionalidades principales:

- Cálculo de la clasificación ABC+D por artículo
- Análisis de rotación de inventario
- Cálculo de márgenes por artículo
- Generación de archivos de clasificación por sección
- Envío de informes a encargados de sección

Uso típico:

```
# Procesar período actual automáticamente
Python\python.exe clasificacionABC.py

# Procesar período específico
Python\python.exe clasificacionABC.py -P P2 -Y 2025

# Procesar sección específica
Python\python.exe clasificacionABC.py -S maf
```

10.3 PRESENTACION.py — Generación de Presentaciones

El script PRESENTACION.py genera documentos de presentación en formato Excel con los datos y análisis más relevantes de cada período.

Funcionalidades principales:

- Consolidación de datos de ventas por período
- Generación de gráficos y tablas resumen
- Comparativa entre períodos
- Creación de documentos profesionales para reuniones

Uso típico:

```
# Generar presentación del período actual
Python\python.exe PRESENTACION.py

# Generar presentación de período específico
Python\python.exe PRESENTACION.py -P P2 -Y 2025
```

10.4 INFORME.py — Informes Generales

El script INFORME.py genera informes HTML completos con el análisis ABC+D de todos los artículos del vivero.

Funcionalidades principales:

- Análisis detallado de cada artículo
- Generación de informes HTML interactivos
- Inclusión de métricas y recomendaciones
- Envío automático por correo electrónico

Uso típico:

```
# Generar informe del período actual
Python\python.exe INFORME.py

# Generar informe de período específico
Python\python.exe INFORME.py -P P2 -Y 2025
```

10.5 informe_compras_sin_autorizacion.py — Informe de Compras No Autorizadas

Este script genera un informe que identifica compras realizadas fuera de los cauces normales de autorización.

Funcionalidades principales:

- Identificación de compras no autorizadas
- Comparativa entre pedidos y compras reales
- Generación de informe detallado
- Alertas sobre desviaciones significativas

Uso típico:

```
Python\python.exe informe_compras_sin_autorizacion.py
```

10.6 Informe_articulos_no_comprados.py — Artículos Pendientes

Este script analiza los artículos que aparecen en los pedidos pero que no se han comprado efectivamente.

Funcionalidades principales:

- Identificación de artículos pedidos pero no comprados
- Análisis de motivos posibles
- Generación de informe de seguimiento

Uso típico:

```
Python\python.exe Informe_articulos_no_comprados.py
```

10.7 analisis_categoria_cd.py — Análisis de Categoría CD

Este script realiza un análisis específico de los artículos categorizados como C y D en la clasificación ABC.

Funcionalidades principales:

- Análisis detallado de artículos de baja rotación
- Recomendaciones sobre gestión del inventario
- Identificación de artículos a eliminar o promocionar
- Generación de informes especializados

Uso típico:

```
Python\python.exe analisis_categoria_cd.py
```

---

11. INFORMES GENERADOS

11.1 Introducción a los Informes del Sistema

El sistema SPA genera diversos tipos de informes como resultado de sus diferentes módulos. Cada informe tiene un propósito específico y proporciona información valiosa para la gestión del vivero.

Los informes se generan automáticamente según la programación establecida y también pueden generarse manualmente cuando sea necesario. Todos los informes se almacenan en la carpeta de salida configurada.

11.2 Informes de Pedidos

11.2.1 Pedido_Semana_XX_YYYY-MM-DD.xlsx

Este es el informe principal generado por el sistema. Contiene los pedidos de compra calculados para una semana específica.

Propósito: Proporcionar a los proveedores la información necesaria para preparar los pedidos de reposición.

Contenido: Lista de artículos con las cantidades de pedido recomendadas, organizadas por sección y proveedor.

Estructura típica:

Columna	Descripción
Código artículo	Identificador único del producto
Nombre artículo	Denominación del producto
Sección	Sección del vivero a la que pertenece
Proveedor	Nombre del proveedor
Cantidad pedido	Unidades a pedir
Coste unitario	Precio por unidad
Importe total	Valor total del pedido

11.2.2 Pedidos por Sección

Además del pedido consolidado, el sistema genera archivos separados para cada sección del vivero.

Propósito: Facilitar la gestión por parte de los encargados de cada sección.

Contenido: Pedidos específicos de cada sección, con detalle completo.

11.2.3 Resúmenes Consolidados

El sistema puede generar resúmenes consolidados que muestran una visión general de todos los pedidos.

Propósito: Proporcionar una visión global de las necesidades de compra.

Contenido: Resumen de totales por sección, por proveedor, y métricas generales.

11.3 Informes de Clasificación ABC

11.3.1 CLASIFICACION_ABC+D_{SECCION}_{PERIODO}.xlsx

Estos archivos contienen la clasificación ABC+D de los artículos de cada sección para un período específico.

Propósito: Identificar la importancia relativa de cada artículo y aplicar los criterios de gestión apropiados.

Contenido:

Columna	Descripción
Código artículo	Identificador del producto
Nombre artículo	Denominación del producto

Columna	Descripción
Sección	Sección a la que pertenece
Clasificación	Categoría ABC+D asignada
Ventas período	Ventas del período actual
Ventas anterior	Ventas del período anterior
Rotación	Índice de rotación del artículo
Margen	Margen de beneficio

11.3.2 Propósito y Utilidad

La clasificación ABC+D es fundamental para la gestión eficiente del inventario. Permite:

- Priorizar la atención a los artículos más importantes (categoría A)
- Determinar niveles de stock apropiados por categoría
- Identificar artículos que requieren acciones especiales (categoría D)
- Optimizar los recursos de gestión de inventario

11.4 Informes de Presentaciones

11.4.1 PRESENTACION_{PERIODO}.xlsx

Estos informes proporcionan una visión resumida de los datos más relevantes del período.

Propósito: Facilitar la presentación de resultados en reuniones gerenciales.

Contenido:

- Gráficos de ventas por sección
- Comparativa de períodos
- Evolución de métricas clave
- Tablas resumen de clasificación ABC

11.4.2 Contenido y Estructura

Las presentaciones incluyen:

- Portada con información del período
- Resumen ejecutivo
- Gráficos visuales de ventas
- Análisis de clasificación ABC
- Recomendaciones y conclusiones

11.5 Informes de Análisis

11.5.1 Informe de Compras Sin Autorización

Identifica compras que se han realizado sin seguir el proceso normal de autorización.

Propósito: Control y seguimiento de compras irregulares.

Contenido:

- Lista de artículos comprados sin autorización
- Proveedores afectados
- Implicaciones económicas
- Recomendaciones de actuación

11.5.2 Informe de Artículos No Comprados

Analiza los artículos que aparecen en los pedidos generados pero que no se han comprado.

Propósito: Identificar desalineamientos y tomar acciones correctivas.

Contenido:

- Artículos pedidos pero no comprados
- Comparativa con compras reales
- Posibles motivos de la desviación

11.5.3 Análisis de Categoría CD

Análisis específico de los artículos de baja rotación.

Propósito: Decisiones sobre gestión de artículos de bajo rendimiento.

Contenido:

- Lista de artículos categoría C y D
- Análisis de motivos de baja rotación
- Recomendaciones de acción (promocionar, eliminar, mantener)

---

12. ANÁLISIS DETALLADO DE COLUMNAS

12.1 Estructura de las Tablas de Datos

Para comprender completamente cómo funciona el sistema y qué significan los datos generados, es fundamental entender la estructura de las tablas de datos tanto de entrada como de salida. Este capítulo proporciona un análisis detallado de cada columna.

El sistema utiliza un formato de macro tabla con cuatro columnas para documentar cada dato:

- **Columna**: Nombre de la columna o campo de datos
- **Descripción**: Explicación de qué representa ese dato
- **Fórmula**: Cómo se obtiene el valor (si es un cálculo) o "datos de entrada" si proviene de archivos externos
- **Módulo**: Qué parte del sistema utiliza o genera este dato

12.2 Tabla de Descripción de Columnas

A continuación se presentan las tablas detalladas para cada tipo de archivo.

12.2.1 Archivo SPA_ventas.xlsx (Datos de Entrada)

Columna	Descripción	Fórmula	Módulo
Código artículo	Identificador único del producto en el sistema	Datos de entrada (no calculada)	data_loader.py
Nombre artículo	Denominación comercial del producto	Datos de entrada (no calculada)	data_loader.py
Fecha venta	Fecha en que se realizó la venta	Datos de entrada (no calculada)	data_loader.py
Cantidad	Número de unidades vendidas	Datos de entrada (no calculada)	data_loader.py
Importe	Valor monetario total de la venta	Datos de entrada (no calculada)	data_loader.py
Sección	Sección del vivero a la que pertenece el artículo	Datos de entrada (no calculada)	data_loader.py
Familia	Agrupación adicional del artículo	Datos de entrada (no calculada)	data_loader.py
Código barras	Código identificador del artículo	Datos de entrada (no calculada)	data_loader.py

12.2.2 Archivo SPA_coste.xlsx (Datos de Entrada)

Columna	Descripción	Fórmula	Módulo
Código artículo	Identificador único del producto	Datos de entrada (no calculada)	data_loader.py

Columna	Descripción	Fórmula	Módulo
Nombre artículo	Denominación comercial del producto	Datos de entrada (no calculada)	data_loader.py
Coste unitario	Precio de compra por unidad	Datos de entrada (no calculada)	data_loader.py
Precio venta	Precio de venta recomendado	Datos de entrada (no calculada)	data_loader.py
Proveedor	Nombre del proveedor principal	Datos de entrada (no calculada)	data_loader.py
Código proveedor	Código del artículo en el sistema del proveedor	Datos de entrada (no calculada)	data_loader.py
Unidad medida	Unidad en que se mide el producto (ud, kg, caja)	Datos de entrada (no calculada)	data_loader.py
IVA	Porcentaje de IVA aplicable	Datos de entrada (no calculada)	data_loader.py
Stock mínimo	Cantidad mínima de stock de seguridad	Datos de entrada (no calculada)	data_loader.py

12.2.3 Archivo de Pedidos Generados (Datos de Salida)

Columna	Descripción	Fórmula	Módulo
Código artículo	Identificador único del producto	Datos de entrada (no calculada)	order_generator.py
Nombre artículo	Denominación comercial del producto	Datos de entrada (no calculada)	order_generator.py
Sección	Sección del vivero	Datos de entrada (no calculada)	order_generator.py
Proveedor	Nombre del proveedor principal	Datos de entrada (no calculada)	order_generator.py
Stock actual	Cantidad en almacén actualmente	Datos de entrada (no calculada)	order_generator.py
Stock mínimo	Stock mínimo de seguridad configurado	Datos de entrada (no calculada)	order_generator.py
Venta media semanal	Promedio de ventas semanales del artículo	Media de ventas de las últimas N semanas	forecast_engine.py

Columna	Descripción	Fórmula	Módulo
Objetivo sección	Objetivo de venta de la sección para la semana	Configuración de config.json	forecast_engine.py
Factor ponderado	Factor según categoría ABC del artículo	Según peso de categoría en config.json	forecast_engine.py
Cantidad calculada	Cantidad de pedido calculada por el algoritmo	Fórmula: (Objetivo × Factor) - Stock + Stock_mínimo	forecast_engine.py
Cantidad final	Cantidad de pedido tras aplicar correcciones	Cantidad calculada ajustada por correcciones	forecast_engine.py
Coste unitario	Coste por unidad del artículo	Datos de entrada (no calculada)	order_generator.py
Importe pedido	Valor total del pedido	Cantidad final × Coste unitario	order_generator.py
Categoría ABC	Clasificación ABC del artículo	Datos de entrada (no calculada)	order_generator.py

12.2.4 Archivo de Clasificación ABC (Datos de Salida)

Columna	Descripción	Fórmula	Módulo
Código artículo	Identificador único del producto	Datos de entrada (no calculada)	clasificacionABC.py
Nombre artículo	Denominación comercial del producto	Datos de entrada (no calculada)	clasificacionABC.py
Sección	Sección del vivero	Datos de entrada (no calculada)	clasificacionABC.py
Ventas período actual	Ventas acumuladas en el período actual	SUM de ventas del período actual	clasificacionABC.py
Ventas período anterior	Ventas acumuladas en el período anterior	SUM de ventas del período anterior	clasificacionABC.py
Porcentaje ventas	Porcentaje sobre total de ventas	(Ventas artículo / Ventas totales) × 100	clasificacionABC.py
Porcentaje acumulado	Porcentaje acumulado en orden descendente	Suma acumulada de porcentajes	clasificacionABC.py

Columna	Descripción	Fórmula	Módulo
Clasificación	Categoría ABC+D asignada	Según umbrales: A (0-80%), B (80-95%), C (95-99%), D (>99%)	clasificacionABC.py
Rotación	Índice de rotación del artículo	Ventas / Stock medio	clasificacionABC.py
Margen unitario	Margen por unidad	Precio venta - Coste unitario	clasificacionABC.py
Margen total	Margen total del artículo	Margen unitario × Cantidad vendido	clasificacionABC.py
Variación	Cambio respecto al período anterior	((Ventas actual - Ventas anterior) / Ventas anterior) × 100	clasificacionABC.py

12.2.5 Archivo state.json (Estado del Sistema)

Campo	Descripción	Fórmula	Módulo
versión	Versión del sistema	Valor fijo en código	state_manager.py
última_ejecución	Fecha y hora de la última ejecución	datetime de la ejecución	state_manager.py
última_semana_procesada	Número de la última semana procesada	Número de semana	state_manager.py
stock_acumulado	Stock acumulado por artículo	Acumulado de ventas - compras	state_manager.py
histórico_ejecuciones	Registro de todas las ejecuciones	Lista de ejecuciones	state_manager.py
pedidos_generados	Historial de archivos de pedido generados	Lista de archivos	state_manager.py
métricas	Estadísticas acumuladas	Cálculos diversos	state_manager.py

12.3 Glosario de Fórmulas y Cálculos

A continuación se explican las principales fórmulas utilizadas por el sistema en los cálculos de pedidos.

Fórmula de cálculo de cantidad de pedido:

```
Cantidad_pedido = (Objetivo_venta_semanal × Factor_categoría_ABC) -
Stock_actual + Stock_mínimo
```

Esta fórmula básica se complementa con ajustes por:

- Períodos festivos (factor de reducción)
- Tendencias de venta (ajuste por variación)
- Correcciones por stock real (si está habilitado)

Fórmula de clasificación ABC:

```
Porcentaje_artículo = (Ventas_artículo / Ventas_totales) × 100
Porcentaje_acumulado = Σ Porcentaje_artículos_ordenados

Clasificación:
- A: Porcentaje_acumulado ≤ 80%
- B: 80% < Porcentaje_acumulado ≤ 95%
- C: 95% < Porcentaje_acumulado ≤ 99%
- D: Porcentaje_acumulado > 99%
```

Fórmula de rotación:

```
Rotación = Ventas_anuales / Stock_medio
```

---

13. ARCHIVOS DE SALIDA

13.1 Ubicación de Archivos de Salida

Por defecto, el sistema genera todos los archivos de salida en la carpeta data/output/ dentro del directorio del proyecto. Esta ubicación puede modificarse en el archivo de configuración.

La estructura de carpetas de salida es:

```
data/output/
├── Pedidos/                    # Pedidos semanales generados
├── Pedidos_Secciones/          # Pedidos por sección
├── Resumenes/                  # Resúmenes consolidados
├── Clasificaciones/            # Archivos de clasificación ABC
├── Presentaciones/             # Presentaciones de período
├── Informes/                   # Informes detallados
└── Analisis/                   # Análisis específicos
```

13.2 Nomenclatura de Archivos

El sistema utiliza una nomenclatura consistente para los archivos de salida que facilita su identificación:

Pedidos:

- Pedido_Semana_15_2025-04-10.xlsx (Pedido de la semana 15, generado el 10 de abril de 2025)

Clasificaciones:

- CLASIFICACION_ABC+D_maf_P2_2025.xlsx (Clasificación de sección MAF, Período 2, Año 2025)

Presentaciones:

- PRESENTACION_P2_2025.xlsx (Presentación del Período 2, Año 2025)

Informes:

- INFORME_P2_2025.html (Informe del Período 2, Año 2025)

13.3 Tipos de Archivos Generados

El sistema genera varios tipos de archivos dependiendo del módulo que se ejecute:

Archivos Excel (.xlsx): Utilizados para todos los datos tabulares como pedidos, clasificaciones, resúmenes. Formato principal de salida.

Archivos HTML (.html): Utilizados para los informes interactivos que pueden visualizarse en un navegador web.

Archivos JSON (.json): Utilizados para el archivo de estado del sistema y para algunos datos intermedios.

13.4 Organización por Fecha y Período

Los archivos de salida se organizan automáticamente por fecha de generación y período al que pertenecen. Esta organización facilita la búsqueda y recuperación de archivos históricos.

El sistema mantiene una estructura de carpetas que agrupa los archivos por año y período, permitiendo navegar fácilmente por el histórico de operaciones.

---

14. ESTADO Y PERSISTENCIA

14.1 Archivo state.json

El archivo state.json es el elemento central de la persistencia del sistema. Almacena toda la información de estado entre ejecuciones, permitiendo que el sistema mantenga memoria de las operaciones realizadas.

Este archivo se encuentra en la ubicación configurada (por defecto: data/state.json) y se actualiza automáticamente después de cada ejecución del sistema.

14.2 Información del Sistema

El archivo de estado incluye información general del sistema como:

- Número de versión del sistema
- Fecha y hora de última ejecución
- Estado general de funcionamiento
- Configuración activa

14.3 Stock Acumulado

El sistema mantiene un registro del stock acumulado de cada artículo, calculado como:

```
Stock_acumulado = Stock_inicial + Compras - Ventas
```

Este valor se actualiza con cada ejecución y permite al sistema calcular pedidos precisos considerando el movimiento real del inventario.

14.4 Histórico de Ejecuciones

Cada vez que el sistema ejecuta un proceso, registra información sobre la ejecución en el histórico. Esta información incluye:

- Fecha y hora de ejecución
- Semana procesada
- Resultado de la ejecución (éxito/error)
- Archivos generados
- Métricas de la ejecución

Este histórico permite realizar auditorías y diagnósticos del sistema.

14.5 Métricas y Estadísticas

El sistema acumula diversas métricas a lo largo del tiempo:

- Total de pedidos generados
- Importe total de pedidos
- Artículos más pedidos
- Sección con mayor actividad
- Tasa de éxito de ejecuciones

Estas métricas proporcionan información valiosa para la gestión del negocio.

---

15. LOGS Y MONITOREO

15.1 Ubicación de Archivos de Log

El sistema genera archivos de log en la carpeta logs/ dentro del directorio del proyecto. Por defecto, esta ubicación es logs/sistema.log.

Los logs son archivos de texto que registran todas las operaciones realizadas por el sistema, incluyendo mensajes de información, advertencias y errores.

15.2 Niveles de Logging

El sistema utiliza diferentes niveles de logging para clasificar los mensajes:

- **INFO**: Mensajes informativos sobre el funcionamiento normal
- **WARNING**: Situaciones que requieren atención pero no son errores
- **ERROR**: Errores que afectan al funcionamiento
- **DEBUG**: Mensajes detallados para diagnóstico (cuando se usa --verbose)

15.3 Interpretación de Mensajes

Los mensajes de log siguen un formato estándar:

```
2025-02-15 21:50:00 - INFO - Iniciando proceso de generación de pedidos
2025-02-15 21:50:01 - INFO - Cargando configuración desde config.json
2025-02-15 21:50:02 - INFO - Procesando sección: maf
2025-02-15 21:50:05 - INFO - Archivo de pedidos generado:
Pedido_Semana_15_2025-02-15.xlsx
2025-02-15 21:50:06 - INFO - Proceso completado exitosamente
```

15.4 Logging Detallado (Modo Verboso)

Para obtener información más detallada durante la ejecución, usar el parámetro --verbose:

```
Python\python.exe main.py --verbose
```

Este modo activa el nivel de logging DEBUG, que incluye mensajes adicionales sobre cada paso del proceso, facilitando el diagnóstico de problemas.

15.5 Monitoreo del Sistema

Para monitorear el funcionamiento del sistema, se pueden revisar:

1. **Logs de ejecución**: Revisar el archivo de log más reciente para verificar que las ejecuciones son exitosas.
2. **Estado del sistema**: Ejecutar `python main.py --status` para ver el estado actual.
3. **Archivos de salida**: Verificar que se generan los archivos esperados después de cada ejecución.
4. **Tareas programadas**: Revisar el historial de ejecuciones en el Programador de tareas de Windows.

---

16. SOLUCIÓN DE PROBLEMAS

16.1 Problemas de Instalación

Problema: El instalador no se ejecuta o da error

Soluciones posibles:

- Ejecutar el instalador como administrador (clic derecho > Ejecutar como administrador)
- Verificar que el antivirus no esté bloqueando la ejecución
- Comprobar que se tiene espacio suficiente en disco

Problema: Python no se instala correctamente

Soluciones posibles:

- Verificar la conexión a internet
- Descargar e instalar Python manualmente desde python.org
- Comprobar que el directorio de instalación tiene permisos de escritura

16.2 Errores de Python y Dependencias

Problema: Error "No module named 'pandas'"

Solución: Las dependencias no se instalaron correctamente. Ejecutar:

```
Python\python.exe -m pip install -r requirements.txt
```

Problema: Errores de versión de Python

Solución: Verificar que se está usando el Python del proyecto y no otro:

```
Python\python.exe --version
```

16.3 Problemas con Archivos de Entrada

Problema: Error "Archivo no encontrado: SPA_ventas.xlsx"

Soluciones posibles:

- Verificar que el archivo existe en la carpeta data/input/
- Comprobar que el nombre del archivo coincide exactamente con la configuración
- Revisar la configuración de rutas en config.json

Problema: Error al leer datos de archivos Excel

Soluciones posibles:

- Verificar que los archivos no están corruptos
- Comprobar que el formato es .xlsx (no .xls)
- Asegurarse de que los archivos no están abiertos en Excel

16.4 Errores en la Generación de Pedidos

Problema: Los pedidos se generan vacíos o sin datos

Soluciones posibles:

- Verificar que los archivos de entrada contienen datos
- Comprobar que las secciones están configuradas como activas
- Revisar los logs para ver si hay errores de procesamiento

Problema: Los pedidos no se corresponden con las expectativas

Soluciones posibles:

- Revisar los objetivos de venta en config.json
- Verificar que los datos de ventas históricos son correctos
- Comprobar los parámetros de cálculo

16.5 Tareas Programadas No se Ejecutan

Problema: Las tareas programadas no aparecen o no se ejecutan

Soluciones posibles:

- Ejecutar SPA.exe nuevamente como administrador para recrear las tareas
- Verificar que el Programador de tareas de Windows está habilitado
- Comprobar el historial de tareas en el Programador de tareas

16.6 Problemas de Permisos

Problema: Errores de acceso a archivos o carpetas

Soluciones posibles:

- Ejecutar con permisos de administrador
- Verificar los permisos de la carpeta del proyecto
- Comprobar que no hay archivos abiertos por otros programas

16.7 Códigos de Error Comunes

Código	Significado	Solución
E001	Archivo de configuración no encontrado	Verificar config/config.json
E002	Archivo de entrada no encontrado	Verificar archivos en data/input/
E003	Error al leer archivo Excel	Verificar formato y contenido
E004	Error de formato en config.json	Revisar sintaxis JSON

Código	Significado	Solución
E005	Sin permisos de escritura	Ejecutar como administrador
W001	Datos insuficientes para cálculo	Revisar archivos de entrada
W002	Sección sin configuración	Revisar config.json

17. MANTENIMIENTO

17.1 Actualización del Sistema

Para mantener el sistema actualizado:

1. Descargar la nueva versión del repositorio
2. Respaldar la configuración actual (config.json, state.json)
3. Reemplazar los archivos del sistema con los nuevos
4. Restaurar la configuración respaldada
5. Verificar el funcionamiento

17.2 Respaldo de Datos y Configuración

Es importante realizar copias de seguridad periódicas de:

- Archivo de configuración: config/config.json
- Archivo de estado: data/state.json
- Archivos de entrada en data/input/
- Logs en logs/

Se recomienda automatizar estas copias de seguridad o realizarlas al menos mensualmente.

17.3 Restauración del Sistema

Para restaurar el sistema desde una copia de seguridad:

1. Reemplazar los archivos actuales con los respaldados
2. Verificar permisos de archivos
3. Ejecutar el sistema en modo verbose para verificar

17.4 Limpieza de Archivos Temporales

Periódicamente, eliminar archivos antiguos de las carpetas de salida para liberar espacio:

- Archivos de pedidos de hace más de 3 meses
- Logs antiguos
- Archivos temporales de procesos fallidos

17.5 Monitoreo de Espacio en Disco

Vigilar el espacio disponible en el disco donde está instalado el sistema. Si el espacio se agota, el sistema no podrá funcionar correctamente.

---

18. SEGURIDAD

18.1 Permisos de Archivos y Carpetas

El sistema debe tener permisos de lectura y escritura en su directorio de instalación. Los permisos mínimos requeridos son:

- Lectura de archivos de configuración y código fuente
- Escritura en carpetas de datos, salida y logs
- Ejecución de scripts Python

18.2 Ejecución con Privilegios de Administrador

Algunas operaciones requieren privilegios de administrador:

- Instalación inicial (creación de tareas programadas)
- Modificación de tareas programadas
- Acceso a ciertas carpetas del sistema

Para estas operaciones, usar "Ejecutar como administrador".

18.3 Aislamiento del Python Portable

El sistema utiliza Python portable, que está aislado del Python del sistema. Esto significa:

- No afecta otras instalaciones de Python
- No requiere desinstalación para remove el sistema
- Puede coexistir con otras versiones de Python

18.4 Protección de Datos Sensibles

El sistema maneja información sensible del negocio. Para protegerla:

- Restringir permisos de acceso a la carpeta del sistema
- No almacenar contraseñas en archivos de configuración (usar variables de entorno)
- Respaldar y eliminar datos sensibles de equipos compartidos

---

19. REFERENCIA TÉCNICA

19.1 Estructura de Directorios Completa

```
SPA/
├── Python/                    # Python portable (3.11.9)
│   ├── python.exe
│   ├── pythonw.exe
│   ├── Scripts/
│   └── Lib/
├── data/
│   ├── input/                 # Archivos de entrada
│   ├── output/                # Archivos generados
│   │   ├── Pedidos/
│   │   ├── Clasificaciones/
│   │   ├── Presentaciones/
│   │   └── Informes/
│   └── state.json            # Estado del sistema
├── src/                       # Código fuente
│   ├── main.py
│   ├── config_loader.py
│   ├── data_loader.py
│   ├── forecast_engine.py
│   ├── order_generator.py
│   ├── state_manager.py
│   ├── scheduler_service.py
│   ├── email_service.py
│   ├── alert_service.py
│   ├── integracion_alertas.py
│   ├── correction_engine.py
│   ├── correction_data_loader.py
│   ├── date_utils.py
│   ├── paths.py
│   └── __pycache__/
├── config/
│   └── config.json            # Configuración principal
├── logs/                      # Archivos de log
│   └── sistema.log
├── dist/                      # Archivos compilados
├── build/                     # Archivos de build
├── docs/                      # Documentación
├── SPA.exe                    # Ejecutable instalador
├── main.py                    # Script principal
├── clasificacionABC.py        # Clasificación ABC
```

```
├── INFORME.py                  # Informes
├── PRESENTACION.py             # Presentaciones
├── instalar_todo.py           # Instalador completo
├── requirements.txt            # Dependencias
├── README.md                   # README
└── .git/                       # Git repository
```

19.2 Lista de Scripts y Funcionalidades

Script	Funcionalidad	Uso Principal
main.py	Generación de pedidos semanales	Tarea programada semanal
clasificacionABC.py	Clasificación ABC+D	Tarea programada mensual
INFORME.py	Generación de informes	Tarea programada mensual
PRESENTACION.py	Generación de presentaciones	Tarea programada mensual
informe_compras_sin_autorizacion.py	Informe de compras	Tarea programada semanal
Informe_articulos_no_comprados.py	Artículos pendientes	Tarea programada semanal
analisis_categoria_cd.py	Análisis categorías C y D	Tarea programada semanal
instalar_todo.py	Instalación completa	Instalación inicial
crear_tareas_programadas.py	Crear tareas	Configuración

19.3 Dependencias y Versiones Requeridas

El sistema requiere las siguientes dependencias (especificadas en requirements.txt):

- Python 3.11.9 (incluido portable)
- pandas >= 2.0.0
- numpy >= 1.24.0
- openpyxl >= 3.1.0
- python-dateutil >= 2.8.0

Estas dependencias se instalan automáticamente durante la instalación.

19.4 Formato de Archivos JSON de Configuración

El archivo config.json utiliza formato JSON estándar. Algunas consideraciones:

- Las claves deben estar entre comillas
- Los valores numéricos no llevan comillas
- Los valores booleanos son true/false (sin comillas)
- No se permiten comentarios
- Usar comas para separar elementos, no puntos y comas

19.5 Integración con Otras Aplicaciones

El sistema puede integrarse con otras aplicaciones:

- Sistemas de gestión empresarial (exportando/importando datos)
- Servidores de correo (para notificaciones)
- Sistemas de backup (para copias de seguridad automatizadas)

Consultar la documentación técnica para más detalles sobre integraciones específicas.

---

20. PREGUNTAS FRECUENTES (FAQ)

P: ¿Puedo ejecutar el sistema en Linux o Mac?

R: No directamente. El sistema está diseñado para Windows. Sin embargo, el código Python podría adaptarse con modificaciones en paths y tareas programadas.

P: ¿Qué hago si se corrompe el archivo de estado?

R: Ejecutar el sistema con --reset para comenzar desde cero, o restaurar desde una copia de seguridad si se tiene.

P: ¿Cómo puedo modificar los objetivos de venta?

R: Editar el archivo config/config.json, sección "secciones", modificando los valores de "objetivos_semanales" para cada sección.

P: ¿El sistema envía correos automáticamente?

R: Sí, si está configurado correctamente en config.json y las credenciales están definidas en variables de entorno.

P: ¿Puedo ejecutar solo una sección específica?

R: Sí, modificando la lista "secciones_activas" en config.json para incluir solo las secciones deseadas.

P: ¿Qué pasa si no tengo conexión a internet?

R: El sistema funcionará siempre que ya esté instalado. La conexión solo es necesaria para la instalación inicial y para el envío de correos.

P: ¿Cómo puedo ver qué tareas están programadas?

R: Abrir el Programador de tareas de Windows (taskschd.msc) y buscar las tareas con nombre "Vivero_*".

P: ¿Puedo cambiar el día de ejecución de los pedidos?

R: El día y hora de ejecución se controla a través de las tareas programadas del sistema operativo Windows. Para modificar el horario, debe modificar o recrear la tarea programada "Vivero_Main_Pedidos_Semanales" utilizando el script crear_tareas_programadas.py o directamente desde el Programador de tareas de Windows (taskschd.msc).

P: ¿El sistema hace copia de seguridad automáticamente?

R: No automáticamente. Se recomienda hacer copias de seguridad manuales periódicamente.

P: ¿Qué significan los códigos de error en los logs?

R: Consultar la sección 16.7 de este manual para una lista de códigos de error y sus soluciones.

ANEXOS

Anexo A: Ejemplos de Configuración

A.1 Ejemplo de Objetivos de Venta Semanales

```
"maf": {
    "descripcion": "Planta de temporada y floristeria",
    "objetivos_semanales": {
        "1": 224.49,
        "2": 410.74,
        "3": 462.25,
        ...
    }
}
```

A.2 Ejemplo de Parámetros de Corrección

```
"parametros_correccion": {
    "habilitar_correccion": true,
    "stock_minimo_por_categoria": {
        "A": 1.5,
        "B": 1.0,
        "C": 0.5,
        "D": 0.0
    }
}
```

Anexo B: Casos de Uso Típicos

B.1 Regenerar Pedidos de una Semana Pasada

```
python main.py --semana 12
```

B.2 Ver Estado del Sistema

```
python main.py --status
```

B.3 Generar Clasificación ABC de Período Específico

```
python clasificacionABC.py -P P2 -Y 2025
```

Anexo C: Historial de Versiones

Versión	Fecha	Cambios
1.0	Febrero 2026	Versión inicial del manual

Anexo D: Contacto y Soporte Técnico

Para soporte técnico o consultas sobre el sistema:

- Contacto principal: ivan.delgado@viveverde.es
- Documentación adicional: Disponible en la carpeta docs/
- Issues y mejoras: A través del repositorio GitHub

FIN DEL MANUAL

Manual de Usuario del Sistema de Pedidos de Compra (SPA) - Vivero Aranjuez V2