

### **Descripción del Problema:**

El Supermercado "Bit Market" desea organizar y persistir en un archivo la información de stock de la casa central. Para ello, mantiene el stock de sus artículos en un archivo [CSV](#) con los siguientes campos: Sección, Código de artículo, Nombre del artículo, Cantidad en stock, Precio unitario, Fecha de adquisición, Fecha de última venta, Fecha de caducidad y un campo que indica si el artículo está o no de alta.

Cada artículo se identifica de manera única mediante los campos Sección y Código de artículo (campos clave).

[Enlace al archivo CSV aquí.](#)

### **Desarrolla un algoritmo que permita:**

- **Generar un arreglo ORDENADO de artículos:** Consiste en **levantar el archivo CSV a un arreglo y ordenarlo por Sección y Código de artículo** (campos clave). *Esta tarea, debe realizarse al inicio del algoritmo y luego de las inicializaciones y validaciones que considere necesarias.*
- **Generar un archivo ORDENADO de artículos, llamado INVENTARIO.DAT** que contenga un registro por cada artículo. Para ello será necesario:
  - a) Contar con el arreglo del punto 1 (ordenado por clave).
  - b) Escribir cada uno de los registros del arreglo **en un archivo de registros**, considerando que:
    - i) Si un artículo está de baja, no se tiene en cuenta (no se copia al archivo).
    - ii) Si un artículo está vencido, no se tiene en cuenta (no se copia al archivo).Observar que el archivo se generará ordenado por clave siempre que el arreglo también lo esté.
- Una vez conseguido lo anterior, el algoritmo debe ofrecer un menú de opciones que permita:
  1. **Dar de alta un artículo en el archivo:** Debe solicitar al usuario los datos de un nuevo artículo. Además, debe guardar esta información en el archivo **INVENTARIO.DAT**, teniendo en cuenta que *no debe existir (ni de alta ni de baja) el nuevo artículo* y que *el archivo debe permanecer ordenado luego de añadir el nuevo artículo*.
  2. **Modificar un artículo de alta en el archivo:** Debe solicitar al usuario la clave del artículo que se desea modificar. En caso de existir, debe permitir modificar todos sus datos (excepto las componentes del campo clave y campo de alta).
  3. **Eliminar un artículo del archivo:** Debe solicitar al usuario la clave del artículo a eliminar. En caso de existir, debe *eliminarlo lógicamente del archivo* (marcarlo de baja), manteniendo el orden del mismo.
  4. **Activar un artículo de baja en el archivo:** Sólo permitirá activar un artículo marcado de baja. Para ello, debe solicitar al usuario la clave del artículo que desea activar y marcarlo de alta.
  5. **Mostrar un artículo del archivo:** Debe solicitar al usuario la clave del artículo a mostrar. En caso de existir, mostrará por pantalla toda su información.
  6. **Listar** todos los artículos de una sección, separándolos por:
    - a. Artículo de alta
    - b. Artículos de baja
  7. **Exportar a CSV el archivo INVENTARIO.DAT:** Debe permitir exportar toda la información del archivo de inventario, a formato [CSV](#). Para ello, debe generar un **archivo de texto** llamado **NUEVO\_INVENTARIO.CSV** con una línea por artículo, en donde cada campo estará separado por coma.

El algoritmo funcionará, mientras el usuario lo desee (puede incluir en el menú una opción para salir).

### **Consideraciones de importancia:**

- Debes utilizar el archivo [SUCURSAL\\_CENTRO.CSV](#) provisto por la cátedra.
- Puedes escribir pseudocódigo y código fuente en paralelo. Es posible comenzar con **pequeñas porciones de pseudocódigo** e ir **codificando en Pascal**, para avanzar a paso seguro respecto a lo que estás

haciendo en papel. Se recomienda comenzar por el menú principal, para continuar con cada una de las opciones.

- Puedes reutilizar aquellos subalgoritmos del TP 1 que consideres de utilidad.
- Debes documentar pseudocódigo y código, **especificando cada subalgoritmo** (qué hace, precondiciones, poscondiciones) y añadiendo comentarios en donde lo consideres necesario.
- Para volcar el pseudocódigo a código fuente, debes usar el lenguaje de programación Pascal. En el aula virtual se encuentra disponible el **IDE Dev-Pascal**.
- El trabajo es grupal (**2 a 4 integrantes**). **No se aceptarán trabajos individuales, salvo justificación formal.**
- En pseudocódigo puedes asumir la existencia de las siguientes operaciones:
  - **Procedimiento limpiarPantalla()**  
**Qué hace:** Permite limpiar la pantalla. Su equivalente en Pascal es *ClrScr()*.
  - **Función enteroACadena (n: entero): cadena**  
**Qué hace:** Convierte un número en una cadena. Su equivalente en Pascal es *intToStr()*.  
**Prec:** N = n  
**Posc:** C = enteroACadena
  - **Función CadenaAentero (cad: cadena): entero**  
**Qué hace:** Convierte una cadena íntegramente formada por dígitos en una cadena. Su equivalente en Pascal es *StrToInt()*.  
**Prec:** C = cad. C es una cadena que sólo contiene dígitos.  
**Posc:** N = CadenaAentero
  - **Función CadenaAreal (cad: cadena): real**  
**Qué hace:** Convierte una cadena numérica a un valor real Su equivalente en Pascal es *StrToFloat()*.  
**Prec:** C = cad. C es una cadena numérica  
**Posc:** R = CadenaAreal
  - **Función pos (c: carácter, cad: cadena): entero**  
**Qué hace:** Busca la primera ocurrencia de un carácter en una cadena. Su equivalente en Pascal *pos()*.  
**Prec:** C = c, D = cad.  
**Posc:** pos = -1 si C no está en D. Caso contrario pos = N si C está en D u D[N] = C

Para contar con estas operaciones en Pascal, debes incluir a tu código, las librerías Crt y Sysutils.

### **Fecha y forma de entrega:**

La entrega del trabajo consta de dos etapas.

- **PRIMERA ETAPA (Pre-entrega opcional):**  
El día **06/11/2025** deberás entregar **como mínimo PSEUDOCÓDIGO** (puedes acompañar con CÓDIGO FUENTE si así lo deseas). Esta pre-entrega es **opcional** y debes entregar lo que tengas funcionando hasta ese momento. El objetivo es que recibas observaciones y/o correcciones por parte de la cátedra. Todas estas observaciones y/o correcciones deberán ser aplicadas en la entrega final. Es importante que sigas avanzando en tu proyecto mientras esperas las observaciones de la persona encargada de revisar

tu entrega. Ten en cuenta que si bien esta instancia es opcional, representa la primera entrega del TP, siendo la siguiente fecha, **la ÚLTIMA instancia disponible para la aprobación del mismo.**

- **SEGUNDA ETAPA (Entrega final y defensa oral):**

El día **18/11/2025** deberás entregar pseudocódigo y código fuente en Pascal funcionando. Ese mismo día se realizará la **Defensa Oral del Trabajo** en horario de clase. **Todos** los integrantes del grupo deben estar presentes y participar de la defensa. La ausencia de un integrante resultará en considerarlo **ausente**.

Las entregas de la primera y segunda etapa se realizan en la [Zona de entrega del Trabajo Práctico 2](#).

El trabajo resuelto debe ser presentado en formato **.zip** o **.rar** conteniendo:

- **Archivo .docx** con la solución propuesta y una **carátula**, que indique:
  - Nombre de la materia
  - Fecha de entrega y año en que cursa
  - Nombre de integrantes del grupo
  - Enunciado del trabajo práctico

Además, la resolución en **pseudocódigo**, debidamente documentada.

- **Archivo .pas** correspondiente al **código fuente Pascal**, fiel al diseño propuesto en Pseudocódigo (**sólo el archivo .pas**. *No incluir el resto de los archivos generados en el proyecto en Dev-Pascal*).

Tanto el pseudocódigo como el código fuente **DEBEN ESTAR DOCUMENTADOS**.

### **Forma de evaluación:**

El equipo de cátedra evaluará el pseudocódigo, el código fuente y probará el programa. El grupo realizará una defensa oral el día 18/11/2025. **Todos/as los y las integrantes del grupo deberán participar de la defensa. Se considerará ausente a quien no participe de esta instancia.**

La calificación surgirá del promedio entre la nota grupal y la nota individual (fijada en la defensa).

### **Criterios de evaluación:**

Para evaluar el Trabajo Práctico se tendrá en cuenta:

- La capacidad de trabajo en equipo.
- La correctitud y robustez de la solución propuesta.
- La modularización de la solución implementada y la capacidad de dividir un problema en subproblemas.
- La legibilidad de la solución (indentación, uso de nombres significativos).
- El paso de pseudocódigo a código fuente.
- La documentación de pseudocódigo y código fuente (especificaciones, comentarios, etc.).
- El uso de herramientas provistas por el lenguaje y por el IDE Dev-Pascal (u otro).

### **Referencias de interés:**

Formato de archivo CSV

[https://es.wikipedia.org/wiki/Valores\\_separados\\_por\\_comas](https://es.wikipedia.org/wiki/Valores_separados_por_comas)

Documentación Pascal

<https://www.freepascal.org/docs-html/rtl/system/index-5.html>