

|                   |          |                   |  |                   |  |
|-------------------|----------|-------------------|--|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |  | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   |  | Legajo            |  |

## ALGORITMOS Y ESTRUCTURA DE DATOS EXAMEN FINAL

Una agencia de venta de autos que comercializa hasta 15 modelos diferentes con un plantel de 10 vendedores dispone de los siguientes archivos de registros:

**VENDEDORES.dat** (sin orden)

| idVendedor | apeYNom       |
|------------|---------------|
| 1..10      | 20 caracteres |

**MODELOS.dat** (sin orden)

| idModelo     | descripcion   |
|--------------|---------------|
| 4 caracteres | 20 caracteres |

**VENTAS.dat** (ordenado por fecha)

| idModelo | idVendedor | fecha       | cliente       |
|----------|------------|-------------|---------------|
|          |            | <i>mmdd</i> | 20 caracteres |

### Se pide

1 - Emitir el siguiente listado ordenado por fecha, idVendedor, idModelo:

Fecha: mm/dd, Cantidad ventas del día: 99

Vendedor: XXXXXXXXXXXX , Modelo: XXXXXXXXXXXXXXXXXXXX

| Orden | Cliente    |
|-------|------------|
| 1     | XXXXXXXXXX |
| 2     | XXXXXXXXXX |
| 3     | XXXXXXXXXX |

2 - Al final del proceso indique el nombre del vendedor con la menor cantidad de ventas (entre los que vendieron) y la fecha de mayor cantidad de ventas. (ambos datos de tipo byte)

### Recursos y restricciones

- Memoria para *arrays*: máximo 2000 bytes.
- Memoria dinámica: nodos de 25 bytes.
- Accesos a disco: 1 acceso secuencial a cada registro de cada archivo.

### Nota para el alumno

- Se considerará especialmente la correcta aplicación de la metodología *top-down*.
- Escribir las hojas en una sola carilla, con letra clara y trazo firme.

|                   |          |                   |  |                   |  |
|-------------------|----------|-------------------|--|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |  | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   |  | Legajo            |  |

### Qué se evalúa

Este examen evalúa que el alumno sepa aprovechar el orden original con el que se presentan los datos. Es decir: dado que el archivo de movimientos (VENTAS.dat) está ordenado por fecha, debe recorrerlo con con corte de control para poder emitir el listado que se pide.

Además, el alumno debe saber aplicar matrices para establecer una relación entre los vendedores y los modelos y combinar estructuras de datos, estáticas y dinámicas para vincular a cada vendedor/modelo la lista de clientes que adquirieron un vehículo.

|                   |          |                   |  |                   |  |
|-------------------|----------|-------------------|--|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |  | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   |  | Legajo            |  |

## ALGORITMOS Y ESTRUCTURA DE DATOS EXAMEN FINAL

Una agencia de venta de autos que comercializa hasta 15 modelos diferentes con un plantel de 10 vendedores dispone de los siguientes archivos de registros:

**VENDEDORES.dat** (sin orden)

| idVendedor | apeYNom       |
|------------|---------------|
| 1..10      | 20 caracteres |

**MODELOS.dat** (sin orden)

| idModelo     | descripcion   |
|--------------|---------------|
| 4 caracteres | 20 caracteres |

**VENTAS.dat** (ordenado por fecha)

| idModelo | idVendedor | fecha | nroReserva | cliente       |
|----------|------------|-------|------------|---------------|
|          |            | mmdd  | 4 dígitos  | 20 caracteres |

### Se pide

1 – Emitir el siguiente listado ordenado por idModelo, idVendedor y fecha.

Modelo: XXXXXXXXXXXX, Cantidad ventas totales: 999

Vendedor: XXXXXXXXXXXX , Cantidad de ventas de este vendedor (para este modelo)

| Orden | Fecha  | Cliente    |
|-------|--------|------------|
| 1     | mm/dd, | XXXXXXXXXX |
| 2     | mm/dd, | XXXXXXXXXX |

2 – Al finalizar el proceso grabar el archivo PERFORMANCE.dat cuya estructura se detalla a continuación con los nombres de los vendedores y la cantidad total de unidades vendidas por cada uno de ellos. Este archivo debe salir ordenado de menor a mayor según esta cantidad.

**PERFORMANCE.dat** (ordenado por cantidad)

| apeYNom | cantidad  |
|---------|-----------|
|         | 4 dígitos |

### Recursos y restricciones

- Memoria para *arrays*: máximo 2000 bytes.
- Memoria dinámica: nodos de 6 bytes.
- Accesos a disco: 1 acceso secuencial a cada registro de cada archivo + 1 acceso directo a cada registro del archivo de ventas.

### Nota para el alumno

- Se considerará especialmente la correcta aplicación de la metodología *top-down*.
- Escribir las hojas en una sola carilla, con letra clara y trazo firme.

|                   |          |                   |  |                   |  |
|-------------------|----------|-------------------|--|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |  | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   |  | Legajo            |  |

### Qué evalúa este examen

Se busca evaluar que el alumno pueda utilizar estructuras de datos combinadas para alterar el orden original de los datos. Como se pide emitir el listado con un orden diferente al del archivo de ventas, el alumno deberá redistribuir adecuadamente los registros para luego, accesos directos mediante, emitir la salida requerida.

|                   |          |                   |  |                   |  |
|-------------------|----------|-------------------|--|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |  | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   |  | Legajo            |  |

## ALGORITMOS Y ESTRUCTURA DE DATOS EXAMEN FINAL

Una agencia de venta de autos que comercializa hasta 15 modelos diferentes con un plantel de 10 vendedores dispone de los siguientes archivos de registros:

**VENDEDORES.dat** (sin orden)

| idVendedor | apeYNom       |
|------------|---------------|
| 1..10      | 20 caracteres |

**MODELOS.dat** (sin orden)

| idModelo     | descripcion   |
|--------------|---------------|
| 4 caracteres | 20 caracteres |

**VENTAS2011.dat, VENTAS2012.dat**

(ambos con el mismo diseño, ordenados por `idModelo` y luego por `idVendedor`)

| idModelo | idVendedor | nroReserva | fecha  | cliente       |
|----------|------------|------------|--------|---------------|
|          |            | 4 dígitos  | aaaamm | 20 caracteres |

**Se pide**

1 – Grabar un archivo ordenado por `idModelo` con los totales de unidades vendidas de cada modelo durante los años 2011 y 2012. El archivo debe tener el siguiente diseño:

**COMPARATIVA.dat**

| idModelo     | descripcion   | vend2011  | vend2012  |
|--------------|---------------|-----------|-----------|
| 4 caracteres | 20 caracteres | 4 dígitos | 4 dígitos |

2 – Emitir un listado ordenado por `idVendedor`, `idModelo` y *año/mes* indicando las cantidades totales vendidas. El usuario ingresará por teclado los datos: *año/mes desde* y *año/mes hasta*.

Código de Vendedor (`idVendedor`): 99

Modelo (`descripcion`): XXXXXXXXXX

| Año/Mes | Cantidad |
|---------|----------|
| 2011/09 | 43       |
| 2011/10 | 30       |
| 2011/11 | 21       |
| 2011/12 | 10       |
| 2012/01 | 31       |
| 2012/02 | 05       |

:

:

Nota: este listado corresponde al siguiente ingreso de datos: *Año/Mes desde*: 201109, *Año/Mes hasta*: 201202.

### Recursos y restricciones

- Archivos: un acceso secuencial a cada uno de ellos.
- Memoria dinámica: 0 bytes. Memoria en disco: sólo para grabar el archivo del punto 1.
- Memoria para *arrays*: 7650 bytes.

### Nota para el alumno

- Se considerará especialmente la correcta aplicación de la metodología *top-down*.
- Escribir las hojas en una sola carilla, con letra clara y trazo firme.

|                   |          |                   |  |                   |  |
|-------------------|----------|-------------------|--|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |  | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   |  | Legajo            |  |

### Qué se evalúa con este examen

Este examen evalúa que el alumnos sepa distinguir entre datos son útiles y datos que son irrelevantes como ser, por ejemplo, el archivo de vendedores.

Se pretende también evaluar si el alumno asocia la comparativa entre los dos años en cuestión con la técnica de apareamiento de archivos.

Finalmente se busca que el alumno pueda asociar la relación modelo/vendedor/fecha a una estructura estática de tipo “cubo” o *array* de 3 dimensiones.

|                   |          |                   |        |                   |  |
|-------------------|----------|-------------------|--------|-------------------|--|
| Fecha:            | 15-12-12 | Firmó la libreta: |        | Cantidad de Hojas |  |
| Apellido y Nombre |          |                   | Legajo |                   |  |

Funciones de biblioteca que puede utilizar, en caso de ser necesario, sin desarrollar. Tenga en cuenta que debe ser preciso con el nombre del módulo y con los parámetros.

| Archivos   |
|--|
| Procedure <b>lecturaEspecial</b> (var Archivo: tipoArchivo; var Registro: tipoRegistro; var Fin: Boolean)<br>Function <b>leerEspecial</b> (var Archivo: tipoArchivo; var Registro: tipoRegistro: boolean<br>Function <b>busquedaBinariaA</b> (var Archivo: tipoArchivo; N:Entero; clave:tipoInfo): Entero<br>Procedure <b>busqBinA</b> (var Archivo: tipoArchivo; clave:tipoInfo; var Registro: tipoRegistro)  |
| Array  |
| Procedure <b>ordenarVector</b> (var Vector: tipoVector; Clave: tipoInfo;N: Entero)<br>Procedure <b>cargarSinRepetir</b> (var Vector:tipoVector; var Pos, N; var Inserto:Boolean; Clave: tipoInfo)<br>Function <b>busquedaBinariaV</b> (var Vector: tipoVector; N:Entero; clave:tipoInfo): Entero<br>Procedure <b>busqBinV</b> (var Vector: tipoVector; N:Entero; clave:tipoInfo; var Pos: Entero)  |
| Estructuras Enlazadas  |
| Procedure <b>meter</b> (var pila: tipoPuntero; valor : tipoInfo)<br>Procedure <b>sacar</b> (var pila: tipoPuntero; var valor : tipoInfo)<br>Procedure <b>agregar</b> (var colaFte, colaFin: tipoPuntero; valor : tipoInfo)<br>Procedure <b>suprimir</b> (var colaFte, colaFin: tipoPuntero; var valor : tipoInfo)<br>Procedure <b>insertaNodo</b> (var lista: tipoPuntero; valor : tipoInfo)<br>Procedure <b>suprimeNodo</b> (var lista: tipoPuntero; valor : tipoInfo)<br>Procedure <b>buscaInserta</b> (var lista, ptr : tipoPuntero; valor : tipoInfo)<br>Procedure <b>insertaPrimero</b> (var lista: tipoPuntero; valor : tipoInfo)<br>Procedure <b>insertaDelante</b> (var lista: tipoPuntero; valor : tipoInfo)<br>Procedure <b>insertaEnMedio</b> (var lista: tipoPuntero; valor : tipoInfo)<br>Procedure <b>insertaAlFinal</b> (var lista: tipoPuntero; valor : tipoInfo)<br>Function <b>buscaNodo</b> (lista: tipoPuntero; valor : tipoInfo): tipoPuntero |