

Carrera INGENIERIA EN INFORMATICA
Asignatura 3629 – Programación Estructurada Básica
Tema: Manejo de archivos secuenciales
Unidad 3
Objetivo: Que el alumno sea capaz de manejar datos persistentes en archivos binarios secuenciales para lectura y grabación de registros, así como también pueda exportar datos en archivos de texto.
Competencia/s a desarrollar manejo de archivos secuenciales
<p>Descripción de la Actividad</p> <p>1-tiempo estimado de resolución: cada ejercicio tiene un tiempo estimado de resolución distinto, siendo los primeros, introductorios, de resolución rápida y con un aumento de complejidad gradual. En cualquier caso, el tiempo esperado de resolución máxima de un ejercicio complejo es de 2h.</p> <p>2-metodología: siempre comenzar en papel planteando la estrategia de resolución, analizando el problema completo y decidiendo que archivos, estructuras de datos y algoritmos se deben emplear para resolver cada problema. Luego una vez definida la estrategia se procede a la codificación en lenguaje C para su posterior prueba pensando en los distintos casos posibles. La codificación puede realizarse en papel o computadora con un compilador de c (codeblocks, devcpp, etc). También puede realizarse en celulares o tablets (coding c para Android) o mediante un compilador on line.</p> <p>3-forma de entrega: No se requiere entrega</p> <p>4-metodología de corrección y feedback al alumno: El alumno puede consultar al docente las dudas en clase, también dispone de foros en miel para interacción entre alumnos y clases de consulta. Algunos ejercicios son explicados en clase por el docente para que el alumno pueda comparar con su propia resolución.</p> <p>Bibliografía En miel se dispone de una serie de videos complementarios para poder reforzar los temas y ejemplos de aplicación.</p>



Guía de Ejercicios Prácticos

Unidad 3 – Archivos Secuenciales

3.1 Los datos de los alumnos de una comisión de Computación Transversal son:

- Número de DNI (entero)
 - Nombre y Apellido (80 caracteres)
 - Nota1, Nota2 (entero)
 - Nota Promedio (real, calculado según Nota1 y Nota2)
- a. Declarar un tipo de dato que contenga la información del alumno.
 - b. Con la información indicada para los alumnos. Grabar los datos en el archivo "ALUMNOS.dat". Esta información de grabación finaliza con DNI cero.
 - c. Leer los datos del archivo, mediante la **Función LECTURA**.

3.2 Tomando como base el archivo ALUMNOS.DAT creado en el ejercicio anterior se desea desarrollar un programa que, basado en la información del archivo, genere 3 archivos:

- Uno con los alumnos PROMOCIONADOS (ambas notas mayores o igual a 7).
- Otro archivo con los alumnos en condición CURSADO (ambas notas mayores o igual a 4 pero no promocionados).
- Y un tercer archivo con los REPROBADOS (al menos una nota, menor que 4).

Para comprobar los datos guardados realizar una función que reciba por parámetro el nombre del archivo a mostrar, lo abre y muestre su contenido. Se debe invocar a la función 3 veces para mostrar todos los archivos generados.

3.3 Se desea realizar un programa que permita actualizar la lista de precios de una empresa en forma individual a cada uno de sus productos.

La empresa trabaja con a lo sumo 100 productos que están guardados en el archivo **productos.dat** con la siguiente estructura:

- Código (entero)
- Precio (float)
- Descripción (de hasta 50 caracteres)

Para modificar un precio se ingresa por teclado el código del producto y el nuevo precio. La modificación de precios finaliza con un código de producto igual a 0.

Al finalizar:

- a. Actualizar el archivo de productos con los nuevos precios
- b. Exportar a un archivo con formato de texto (.csv) para que la lista de precios pueda ser visualizada directamente utilizando un programa de planillas de cálculo como por ejemplo el Excel.

3.4 Realizar un programa que permite actualizar una lista de precios en forma masiva, ingresando un porcentaje de incremento. El archivo original se llama **precios.dat** y fue generado utilizando la siguiente estructura:

- Código (entero)
- Precio (float)
- Descripción (de hasta 50 caracteres)

No se sabe la cantidad de registros del archivo.

Se solicita generar un nuevo archivo llamado **Precios_actualizados.dat**.

3.5 Se dispone de un archivo llamado Stock.dat que contiene la información de los 10 productos que vende una fábrica. En el archivo se guarda:

- Código de artículo (entero)
- Descripción (50 caracteres máximo)
- Stock (entero)

Luego se ingresan por teclado las ventas a realizar indicando:

- Código de artículo
- Cantidad

La carga por teclado de las ventas finaliza con un código de artículo igual a 0.

Por cada venta se debe controlar si hay stock suficiente y si lo hay, restar el stock de dicho producto, sino hay stock se debe vender lo que quede disponible y grabar un registro en un archivo Faltantes.dat con la cantidad que no pudo venderse, dicho registro debe contener:

- Código de artículo
- Cantidad

Se genera un registro por cada venta que no pudo concretarse, pudiendo quedar en el archivo varios registros faltantes del mismo producto.

Al finalizar actualizar el archivo Stock.dat con el stock actualizado luego de procesar las ventas.

3.6 Realizar un programa que tome el archivo Faltantes.dat del ejercicio anterior y unifique las cantidades sumariéndolas por artículo (la empresa trabaja con 10 artículos en total, pero no todos pueden estar en falta). Generar una planilla Excel para enviar al proveedor con el pedido a realizar. Para ello se debe crear un nuevo archivo llamado pedido.csv que contenga los registros por artículo conteniendo:

- Código de artículo (entero)
- Cantidad (entero)

3.7 Se dispone de un archivo que contiene información de los vuelos realizados por las distintas aerolíneas a lo largo del mes. El archivo se denomina Vuelos.dat y guarda los registros con la siguiente estructura:

- Código Aerolínea (alfanumérico de 10 caracteres máximo)
- Día (entero)
- Número de Vuelo (entero)
- Costo del pasaje (real)
- Pasajeros (inicialmente en el archivo viene en 0)

El mes tiene 30 días y se sabe que como máximo cada aerolínea realiza 5 vuelos diarios. Luego se dispone de un segundo archivo llamado Pasajeros.dat que incluye la información de los viajeros del mes para la aerolínea con código "Aero1", el archivo contiene los siguientes campos:

- DNI (entero)
- Número de Vuelo (entero)

Se desea realizar un programa que actualice la cantidad de pasajeros de la aerolínea con código "Aero1" con la información de los pasajeros que realizar los viajes. Al finalizar, grabar la información en un nuevo archivo llamado Aero1.dat. Agregar la opción para exportar el archivo Aero1 a formato csv.

3.8 Se dispone de un archivo denominado INSCRIPTOS.dat que contiene la información de los inscriptos a un curso de programación. El curso tiene una capacidad máxima para 60 personas. El archivo aún no está completo ya que la inscripción se realiza por partes. Por cada inscripto se tiene la siguiente información:

- DNI (entero)
- Apellido y Nombres (texto de 20 caracteres máximo)
- Pagado (campo entero donde 1 indica que pagó y 0 que aún adeuda la matrícula)

Se desea realizar un programa que cumpla con dos funciones:

- a. Registrar los pagos de los alumnos ya inscriptos.
- b. Si hay cupo, agregar nuevos inscriptos.

El programa solicitará el ingreso del DNI y lo buscará entre los inscriptos, en caso de que lo encuentre dará la opción para registrar el pago (si es que no está pago ya). En caso de que no lo encuentre dará la opción para inscribirlo al curso (el pago se realiza luego). En el momento que ya no quede cupo en el curso si llega un nuevo inscripto se le preguntará si desea quedar registrado para un curso futuro, y en caso afirmativo se le solicitará el teléfono y se guardará el nombre, DNI y teléfono en un archivo Interesados.dat.

El ingreso de datos finaliza con un DNI negativo. 2

Al finalizar se debe actualizar el archivo de inscriptos con los cambios realizados.

3.9 Una fábrica produce como máximo 20 productos diferentes y necesita un reporte de la producción realizada durante el año. Para ello se dispone del archivo "ARTICULOS.dat", el cual contiene:

- Código de artículo (entero de 4 cifras no correlativo)
- Descripción del artículo (30 caracteres máximo)
- Cantidad de stock remanente del año anterior (entero)

A continuación, se procesa por teclado, la información correspondiente a todo el año:

- Número del mes de producción. Ingresar y controlar mediante la Función **LEE_CONTROL**.
- Código de artículo (entero de 4 cifras no correlativo) Ingresar y controlar un número de 4 cifras. Función **LEE_CONTROL**.
- Cantidad de unidades que se produjeron en ese mes. Controlar si el número es mayor que 0. Función **MAYOR_CERO**.

Esta información no se encuentra ordenada, y termina con número de mes igual a 0.

Se solicita construir un programa que determine e informe:

- Para cada artículo, la cantidad de unidades producidas en cada uno de los 12 meses del año, de acuerdo con el siguiente modelo:

Código Artículo	Descripción	Ene.	Feb.	...	Oct.	Nov.	Dic.
XXX	XXXXXXXXXX	XX	XX	...	XX	XX	XX
XXX	XXXXXXXXXX	XX	XX	...	XX	XX	XX

- Confeccionar un listado con el/los códigos y descripción de aquellos artículos que NO se produjeron en todo el año.
- Generar el archivo actualizado llamado "ARTI_ACTUALIZADO.dat" con la suma del stock remanente del año anterior más la cantidad producida en el primer trimestre del presente año.

3.10 La empresa TODOLIMPIO que comercializa un máximo de 25 productos, dispone del archivo "LIMPIEZA.dat", donde cada registro contiene:

- Número de Artículo (entero, 5 cifras)
- Unidades en stock (entero)
- Precio unitario (real)

Cada vez que se realiza una venta se ingresa por teclado, los siguientes datos:

- Número de Cliente (1 a 20) Función **LEE_CONTROL**
- Número de Artículo (entero, 5 cifras). Función **LEE_CONTROL**
- Cantidad de Unidades Vendidas. Función **MAYORCERO**.

Esta información termina con número de cliente igual a 0.

Si la cantidad de unidades vendidas es superior al stock existente de cada uno de los productos, NO realizar la venta y marcarla para un posterior reporte.

Se solicita confeccionar un programa que actualice el archivo "PRODUCTOS.dat" y e informar:

- La facturación total por cliente.
- La cantidad de veces que cada cliente realizó una compra.
- Emitir un listado ordenado por monto de facturación de forma descendente de acuerdo con el siguiente diseño:

CLIENTE	MONTO FACTURADO	CANTIDAD DE VECES QUE REALIZÓ UNA COMPRA
XXX	XXX,XX	XX

- Emitir un listado con el número de artículos que no se pudo realizar las ventas por falta de stock.
- Actualizar el archivo "LIMPIEZA.dat", en UNIDADES en STOCK según las ventas ingresadas.

3.11 La municipalidad de Marina del Mar desea efectuar un control de la cobranza del impuesto Municipal en los 6 bimestres del año. Existen 50 contribuyentes, numerados de 1 a 50.

- El primer dato para ingresar es el número del año a procesar (xxxx)

Se suministra el archivo "IMPORTES.dat", ordenado por año-bimestre, con 18 ítems, que contienen los importes de los impuestos de los últimos 3 años, cada ítem contiene:

- Año del impuesto (entero)
- Bimestre (entero)
- Importe (real)

Se dispone además de un segundo archivo, llamado "CONTROL.dat", ordenado por número de contribuyente, con un registro por cada cobro efectuado, con los siguientes datos (solo considerar los del año en estudio).

- Número de contribuyente (entero)
- Año del impuesto (entero)
- Bimestre (entero)

Se solicita confeccionar un programa para determinar e informar:

- El importe total recaudado en el año.
- Las deudas de cada contribuyente según el siguiente formato de impresión:

CONTRIBUYENTE	DEUDAS REGISTRADAS EN EL AÑO XXX						IMPORTE ADEUDADO	
	BIMESTRES	1	2	3	4	5		6
XXXXX			X				X	\$ XXXXXX,XX

3.12 Un Laboratorio de medicamentos quiere actualizar las ventas acumuladas a la fecha, dispone para este trabajo de un archivo llamado "VENTAS.dat" con las ventas acumuladas del año, hasta la fecha de proceso con los siguientes datos:

- Número de visitador médico de 1 a 15 (entero)
- Código del medicamento de 1 a 30 (entero)
- Cantidad de unidades vendidas (entero)

Este archivo está clasificado y ordenado en forma ascendente por número de visitador médico y luego por el código del medicamento. Pueden NO estar todos los códigos de medicamentos ni todos los visitadores médicos "activos".

La información que se registra por teclado de las ventas del día NO está ordenada y para finalizar se ingresa un número de cliente igual a 0. Los datos disponibles son:

- Número de Cliente (entero entre 1 y 50000)
- Número de Visitador Médico (entero)
- Código de medicamento (entero)

- Cantidad vendida (entero)
- Importe facturado (float)

Se solicita escribir un programa que permita:

- Leer el archivo "VENTAS.dat" y los cargue en la matriz TOTAL de 15 por 30 elementos.
- Ingresar por teclado, la fecha de proceso (día, mes, año).
- Procesar las ventas del día.
- Actualizar los acumulados de la matriz TOTAL y al finalizar actualizar el archivo "VENTAS.dat".
- Imprimir un listado con el siguiente diseño:

VENTAS DEL XX/XX/XX	
Código de medicamento	Importe de la Facturación
XX	\$ XXXXXX, XX
-----	-----
TOTAL VENDIDO	\$ XXXXXX, XX

- Al finalizar el proceso informe el número de Visitador Médico que en el día vendió el mayor importe. (Total Facturado).

3.13 Se desean procesar las transacciones bancarias de las Cuentas Corrientes del BANCO NACIÓN. Se solicita:

- Leer los archivos "SALDOS.dat" con el saldo del día anterior de cada cliente y lo almacene en la memoria. El archivo está ordenado por número de cuenta y está formado por registros con el siguiente diseño (máximo 300 clientes)
 - Número de cuenta (8 dígitos)
 - Razón Social (28 caracteres)
 - Crédito máximo (real)
 - Saldo (real, que puede contener signo -)
- Ingresar la fecha del día de proceso, controlando que sean valores "lógicos", confeccionando y utilizando una función, llamada LeerValorLogico y utilizarla para el ingreso y validación. (día, mes y Año). Utilizar una estructura.
- También se dispone de otro archivo "MOVIM.dat" con los movimientos del día para actualizar los saldos de las cuentas. Cada registro consta de:
 - Número de Cuenta
 - Código de Operación (R: retiro / D: depósito)
 - Importe (real)
- Si el cliente no se encuentra en el archivo, puede ser una CUENTA NUEVA (Nro. De cuenta mayor al último número existente en el archivo Saldos.dat.) o debido a un error. Esta verificación de búsqueda, realizarla confeccionando y utilizando una Función llamada VeriCuenta. Si hay error se muestra y rechaza la operación, si es cuenta nueva, solo puede ser un depósito, en cuyo caso se solicita por teclado el ingreso de la Razón Social y se lo agrega al final, con el doble del importe como crédito máximo y con el importe como saldo.
- Listar las Cuentas Nuevas, con el siguiente diseño:

CUENTAS NUEVAS DEL XX/XX/XXXX		
Número de Cuenta	Crédito Máximo	Saldo
xxxxxxx	xxxxxxx, xx	xxxxxxx, xx
xxxxxxx	xxxxxxx, xx	xxxxxxx, xx

3.14 Se dispone de un archivo "EXPENSAS.dat" con los importes según cada uno de las 5 categorías de un complejo de edificios. Este archivo consta de un (1) solo registro con los 5 importes correspondientes en forma correlativa a la categoría 1, categoría 2 hasta la 5.

Además, se dispone del archivo "PROPIETARIO.dat" con los datos de cada propietario:

- DNI (entero)
- Apellido y Nombre (50 caracteres)
- Domicilio (50 caracteres)

Se sabe que existen como MÁXIMO 100 PROPIETARIOS.

También se dispone del archivo "PAGOS.dat" con los efectuados por cada propietario que ha pagado la expensa:

- DNI (entero)
- Mes de pago (entero)
- Categoría (entero de 1 a 5)

Se solicita:

a. Informar el siguiente listado, con "x" el mes o meses que NO pagó.

Apellido y Nombre	Mes 1	Mes 2 ...	Mes 12
xxxxxxx, xxxxxxxx	x	x	
xxxxxxx, xxxxxxxx		x	x

b. Total pagado por cada propietario, ordenando en forma ascendente por total pagado.

Apellido y Nombre	Total Pagado
xxxxxxx, xxxxxxxx	\$ xxx,xx
xxxxxxx, xxxxxxxx	\$ xxx,xx

c. Generar el archivo "DEUDAS.dat" con aquellos propietarios que NO han pagado más de 6 meses, con igual diseño que el archivo "PROPIETARIOS.dat".

d. Cuando NO se encuentra el DNI, grabar en el archivo "ERROR.dat" la información contenida en el archivo "PAGOS.dat".

3.15 Desarrollar un programa que permita:

- Solicitar el ingreso desde el teclado del código de uno de los 3 departamento de la empresa AZ CONSULTORES, confeccionando y utilizando una función con parámetros que controles su validez (los códigos de departamento pueden ser 100, 200 o 300), llamarla **LeeCodigoDeparta**.
- Leer luego desde un archivo secuencial, llamado "PERSONAL.dat", los datos que sean necesarios de los empleados que corresponden al departamento seleccionado, reservándolos en memoria y utilizando una adecuada estructura de datos.

El diseño de los registros es:

- Departamento (entero)
- Legajo (Número, entero NO correlativo de 4 cifras)
- Apellido y Nombres (60 caracteres)
- Años de Antigüedad (entero)

Cada departamento tiene como máximo 300 empleados

- Ingresar desde el teclado los sueldos de dichos empleados:

- Legajo (Número, entero NO correlativo de 4 cifras)
- Sueldo (real)

El ingreso se tiene que finalizar con un número de legajo negativo.

- Controlar la validez del legajo (existencia dentro del Departamento) confeccionando y utilizando una Función llamada **ControlLega**. Si se detecta un legajo inexistente se debe informar Error y solicitarlo nuevamente. Grabar un archivo "ERROR.dat" solamente con el número de legajo.
- Confeccionar un listado con los empleados cuyo sueldo es menor que el promedio general de sueldos del departamento en estudio, informando el sueldo y el nuevo sueldo con el suplemento por antigüedad incluido. El suplemento por antigüedad (sobre el sueldo ingresado en el punto c.) es de un 1% para menos de 10 años y 15% para 10 o más años.

REMUNERACIONES MENORES AL PROMEDIO			
LEGAJO	SUELDO	ANTIGÜEDAD	NUEVO SUELDO
XXX	XXXXX,XX	XXX	XXXXX,XX
...
XXX	XXXXX,XX	XXX	XXXXX,XX

3.16 Se dispone de un archivo ventas.dat con el siguiente formato:

- Código de producto (alfanumérico 10 caracteres)
- Código de vendedor (entero 1 a 5)
- Cantidad (entero)

El archivo puede contener más de una venta para el mismo producto/vendedor. Luego se dispone de un segundo archivo llamado productos.dat que contiene:

- Código de producto (alfanumérico 10 caracteres)
- Precio de venta (float)

Como máximo hay 15 productos.

- Se desea mostrar la cantidad de ventas realizada por cada vendedor de cada producto (utilizar una matriz).

Producto	vendedor 1	vendedor 2	vendedor 3	vendedor 4	vendedor 5
Prod1	9	0	1	4	1
Prod2	19	10	0	0	1

- Registrar nuevas ventas ingresando por teclado:

- Código de producto
- Código de vendedor
- Cantidad

(el ingreso por teclado finaliza con vendedor igual a 99)

- Mostrar nuevamente el listado del punto a actualizado con las nuevas ventas
- Mostrar un listado ordenado de mayor a menor por importe de ventas mostrando código de vendedor e importe total vendido.
- Mostrar el / los productos que vendieron menor cantidad de unidades
- Actualizar el archivo ventas.dat.