

Módulo Imperativo

Práctica Inicial

1. – Implementar un programa que procese la información de los alumnos de la Facultad de Informática.

a) Implementar un módulo que lea y retorne, en una estructura adecuada, la información de todos los alumnos. De cada alumno se lee su apellido, número de alumno, año de ingreso, cantidad de materias aprobadas (a lo sumo 36) y nota obtenida (sin contar los aplazos) en cada una de las materias aprobadas. La lectura finaliza cuando se ingresa el número de alumno 11111, el cual debe procesarse.

b) Implementar un módulo que reciba la estructura generada en el inciso a) y retorne número de alumno y promedio de cada alumno.

c) Analizar: ¿qué cambios requieren los puntos a y b, si no se sabe de antemano la cantidad de materias aprobadas de cada alumno, y si además se desean registrar los aplazos? ¿cómo puede diseñarse una solución modularizada que requiera la menor cantidad de cambios?

2. – Implementar un programa que procese información de propiedades que están a la venta en una inmobiliaria.

Se pide:

a) Implementar un módulo para almacenar en una estructura adecuada, las propiedades agrupadas por zona. Las propiedades de una misma zona deben quedar almacenadas ordenadas por tipo de propiedad. Para cada propiedad debe almacenarse el código, el tipo de propiedad y el precio total. De cada propiedad se lee: zona (1 a 5), código de propiedad, tipo de propiedad, cantidad de metros cuadrados y precio del metro cuadrado. La lectura finaliza cuando se ingresa el precio del metro cuadrado -1.

b) Implementar un módulo que reciba la estructura generada en a), un número de zona y un tipo de propiedad y retorne los códigos de las propiedades de la zona recibida y del tipo recibido.

3. – Implementar un programa que procese las ventas de un supermercado. El supermercado dispone de una tabla con los precios y stocks de los 1000 productos que tiene a la venta.

a) Implementar un módulo que retorne, en una estructura de datos adecuada, los tickets de las ventas. De cada venta se lee código de venta y los productos vendidos. Las ventas finalizan con el código de venta -1. De cada producto se lee código y cantidad de unidades solicitadas. Para cada venta, la lectura de los productos a vender finaliza con cantidad de unidades vendidas igual a 0. El ticket debe contener:

- Código de venta

- Detalle (código de producto, cantidad y precio unitario) de los productos que se pudieron vender. En caso de no haber stock suficiente, se venderá la máxima cantidad posible.

- Monto total de la venta.

c) Implementar un módulo que reciba la estructura generada en el inciso a) y un código de producto y retorne la cantidad de unidades vendidas de ese código de producto.

Módulo Imperativo

Práctica Ordenación

1.- Descargar el programa **ImperativoEjercicioClase1.pas** que contiene parte del código del siguiente enunciado a resolver. Implementar lo indicado en el código.

Se desea procesar la información de las ventas de productos de un comercio (como máximo 50).

Implementar un programa que invoque los siguientes módulos:

- Un módulo que retorne la información de las ventas en un vector. De cada venta se conoce el día de la venta, código del producto (entre 1 y 15) y cantidad vendida (como máximo 99 unidades). El código debe generarse automáticamente (random) y la cantidad se debe leer. El ingreso de las ventas finaliza con el día de venta 0 (no se procesa).
- Un módulo que muestre el contenido del vector resultante del punto a).
- Un módulo que ordene el vector de ventas por código.
- Un módulo que muestre el contenido del vector resultante del punto c).
- Un módulo que elimine, del vector ordenado, las ventas con código de producto entre dos valores que se ingresan como parámetros.
- Un módulo que muestre el contenido del vector resultante del punto e).
- Un módulo que retorne la información (ordenada por código de producto de menor a mayor) de cada código par de producto junto a la cantidad total de productos vendidos.
- Un módulo que muestre la información obtenida en el punto g).

2.- El administrador de un edificio de oficinas cuenta, en papel, con la información del pago de las expensas de dichas oficinas.

Implementar un programa que invoque a módulos para cada uno de los siguientes puntos:

- Genere un vector, sin orden, con a lo sumo las 300 oficinas que administra. De cada oficina se ingresa el código de identificación, DNI del propietario y valor de la expensa. La lectura finaliza cuando se ingresa el código de identificación -1, el cual no se procesa.
- Ordene el vector, aplicando el método de inserción, por código de identificación de la oficina.
- Ordene el vector aplicando el método de selección, por código de identificación de la oficina.

3.- Netflix ha publicado la lista de películas que estarán disponibles durante el mes de diciembre de 2022. De cada película se conoce: código de película, código de género (1: acción, 2: aventura, 3: drama, 4: suspenso, 5: comedia, 6: bélico, 7: documental y 8: terror) y puntaje promedio otorgado por las críticas.

Implementar un programa que invoque a módulos para cada uno de los siguientes puntos:

- Lea los datos de películas, los almacene por orden de llegada y agrupados por código de género, y retorne en una estructura de datos adecuada. La lectura finaliza cuando se lee el código de la película -1.

- b. Genere y retorne en un vector, para cada género, el código de película con mayor puntaje obtenido entre todas las críticas, a partir de la estructura generada en a).
- c. Ordene los elementos del vector generado en b) por puntaje utilizando alguno de los dos métodos vistos en la teoría.
- d. Muestre el código de película con mayor puntaje y el código de película con menor puntaje, del vector obtenido en el punto c).

4.- Una librería requiere el procesamiento de la información de sus productos. De cada producto se conoce el código del producto, código de rubro (del 1 al 8) y precio.

Implementar un programa que invoque a módulos para cada uno de los siguientes puntos:

- a. Lea los datos de los productos y los almacene ordenados por código de producto y agrupados por rubro, en una estructura de datos adecuada. El ingreso de los productos finaliza cuando se lee el precio 0.
- b. Una vez almacenados, muestre los códigos de los productos pertenecientes a cada rubro.
- c. Genere un vector (de a lo sumo 30 elementos) con los productos del rubro 3. Considerar que puede haber más o menos de 30 productos del rubro 3. Si la cantidad de productos del rubro 3 es mayor a 30, almacenar los primeros 30 que están en la lista e ignore el resto.
- d. Ordene, por precio, los elementos del vector generado en c) utilizando alguno de los dos métodos vistos en la teoría.
- e. Muestre los precios del vector resultante del punto d).
- f. Calcule el promedio de los precios del vector resultante del punto d).

Módulo Imperativo

Práctica Recursión

1.- Descargar el programa **ImperativoEjercicioClase2.pas** que contiene parte del código del siguiente enunciado a resolver. Implementar lo indicado en el código.

Implementar un programa que invoque a los siguientes módulos.

- Un módulo recursivo que retorne un vector de a lo sumo 15 números enteros “random” mayores a 10 y menores a 155 (incluidos ambos). La carga finaliza con el valor 20.
- Un módulo no recursivo que reciba el vector generado en a) e imprima el contenido del vector.
- Un módulo recursivo que reciba el vector generado en a) e imprima el contenido del vector.
- Un módulo recursivo que reciba el vector generado en a) y devuelva la suma de los valores pares contenidos en el vector.
- Un módulo recursivo que reciba el vector generado en a) y devuelva el máximo valor del vector.
- Un módulo recursivo que reciba el vector generado en a) y un valor y devuelva verdadero si dicho valor se encuentra en el vector o falso en caso contrario.
- Un módulo que reciba el vector generado en a) e imprima, para cada número contenido en el vector, sus dígitos en el orden en que aparecen en el número. Debe implementarse un módulo recursivo que reciba el número e imprima lo pedido. Ejemplo si se lee el valor 142, se debe imprimir 1 4 2.

2.- Escribir un programa que:

- Implemente un módulo recursivo que genere y retorne una lista de números enteros “random” en el rango 100-200. Finalizar con el número 100.
- Un módulo recursivo que reciba la lista generada en a) e imprima los valores de la lista en el mismo orden que están almacenados.
- Implemente un módulo recursivo que reciba la lista generada en a) e imprima los valores de la lista en orden inverso al que están almacenados.
- Implemente un módulo recursivo que reciba la lista generada en a) y devuelva el mínimo valor de la lista.
- Implemente un módulo recursivo que reciba la lista generada en a) y un valor y devuelva verdadero si dicho valor se encuentra en la lista o falso en caso contrario.

3.- Implementar un programa que invoque a los siguientes módulos.

- Un módulo recursivo que retorne un vector de 20 números enteros “random” mayores a 300 y menores a 1550 (incluidos ambos).

b. Un módulo que reciba el vector generado en a) y lo retorne ordenado. (*Utilizar lo realizado en la práctica anterior*)

c. Un módulo que realice una búsqueda dicotómica en el vector, utilizando el siguiente encabezado:

Procedure busquedaDicotomica (v: vector; ini,fin: indice; dato:integer; var pos: indice);

Nota: El parámetro “pos” debe retornar la posición del dato o -1 si el dato no se encuentra en el vector.

Desafío...

4.- Realizar un programa que lea números y que utilice un módulo recursivo que escriba el equivalente en binario de un número decimal. El programa termina cuando el usuario ingresa el número 0 (cero).

Ayuda: Analizando las posibilidades encontramos que: Binario (N) es N si el valor es menor a 2. ¿Cómo obtenemos los dígitos que componen al número? ¿Cómo achicamos el número para la próxima llamada recursiva? Ejemplo: si se ingresa 23, el programa debe mostrar: 10111.

Módulo Imperativo

Práctica Árboles 1

1. Descargar el programa **ImperativoEjercicioClase3.pas** que contiene parte del código del siguiente enunciado a resolver. Implementar lo indicado en el código.

Escribir un programa que:

- a. Implemente un módulo que almacene información de socios de un club en un árbol binario de búsqueda. De cada socio se debe almacenar número de socio, nombre y edad. La carga finaliza con el número de socio 0 y el árbol debe quedar ordenado por número de socio. La información de cada socio debe generarse aleatoriamente.
- b. Una vez generado el árbol, realice módulos independientes que reciban el árbol como parámetro y que :
 - i. Informar los datos de los socios en orden creciente.
 - ii. Informar los datos de los socios en orden decreciente.
 - iii. Informar el número de socio con mayor edad. Debe invocar a un módulo recursivo que retorne dicho valor.
 - iv. Aumentar en 1 la edad de los socios con edad impar e informar la cantidad de socios que se les aumentó la edad.
 - vi. Leer un nombre e informar si existe o no existe un socio con ese nombre. Debe invocar a un módulo recursivo que reciba el nombre leído y retorne verdadero o falso.
 - vii. Informar la cantidad de socios. Debe invocar a un módulo recursivo que retorne dicha cantidad.
 - viii. Informar el promedio de edad de los socios. Debe invocar a un módulo recursivo que retorne el promedio de las edades de los socios.

2. Escribir un programa que:

- a. Implemente un módulo que genere aleatoriamente información de ventas de un comercio. Para cada venta generar código de producto, fecha y cantidad de unidades vendidas. Finalizar con el código de producto 0. Un producto puede estar en más de una venta. Se pide:

- i. Generar y retornar un árbol binario de búsqueda de ventas ordenado por código de producto. Los códigos repetidos van a la derecha.
- ii. Generar y retornar otro árbol binario de búsqueda de productos vendidos ordenado por código de producto. Cada nodo del árbol debe contener el código de producto y la cantidad total de unidades vendidas.
- iii. Generar y retornar otro árbol binario de búsqueda de productos vendidos ordenado por código de producto. Cada nodo del árbol debe contener el código de producto y la lista de las ventas realizadas del producto.

Nota: El módulo debe retornar TRES árboles.

- b. Implemente un módulo que reciba el árbol generado en **i.** y una fecha y retorne la cantidad total de productos vendidos en la fecha recibida.

c. Implemente un módulo que reciba el árbol generado en **ii.** y retorne el código de producto con mayor cantidad total de unidades vendidas.

c. Implemente un módulo que reciba el árbol generado en **iii.** y retorne el código de producto con mayor cantidad de ventas.

3. Implementar un programa que contenga:

a. Un módulo que lea información de los finales rendidos por los alumnos de la Facultad de Informática y los almacene en una estructura de datos. La información que se lee es legajo, código de materia, fecha y nota. La lectura de los alumnos finaliza con legajo 0. La estructura generada debe ser eficiente para la búsqueda por número de legajo y para cada alumno deben guardarse los finales que rindió en una lista.

b. Un módulo que reciba la estructura generada en a. y retorne la cantidad de alumnos con legajo impar.

c. Un módulo que reciba la estructura generada en a. e informe, para cada alumno, su legajo y su cantidad de finales aprobados (nota mayor o igual a 4).

c. Un módulo que reciba la estructura generada en a. y un valor real. Este módulo debe retornar los legajos y promedios de los alumnos cuyo promedio supera el valor ingresado.

Módulo Imperativo

Práctica Árboles 2

1. Descargar el programa ImperativoEjercicioClase4.pas. y completar los módulos indicados en el código.
 - a. Almacenar los productos vendidos en una estructura eficiente para la búsqueda por código de producto. De cada producto deben quedar almacenados su código, la cantidad total de unidades vendidas y el monto total. De cada venta se cargan código de venta, código del producto vendido, cantidad de unidades vendidas y precio unitario. El ingreso de las ventas finaliza cuando se lee el código de venta 0.
 - b. Imprimir el contenido del árbol ordenado por código de producto.
 - c. Retornar el menor código de producto.
 - d. Retornar la cantidad de códigos que existen en el árbol que son menores que un valor que se recibe como parámetro.
 - e. Retornar el monto total entre todos los códigos de productos comprendidos entre dos valores recibidos (sin incluir) como parámetros.
2. Descargar el programa ImperativoEjercicioClase3.pas de la clase anterior e incorporar lo necesario para:
 - i. Informar el número de socio más grande. Debe invocar a un módulo recursivo que retorne dicho valor.
 - ii. Informar los datos del socio con el número de socio más chico. Debe invocar a un módulo recursivo que retorne dicho socio.
 - iii. Leer un valor entero e informar si existe o no existe un socio con ese valor. Debe invocar a un módulo recursivo que reciba el valor leído y retornar verdadero o falso.
 - iv. Leer e informar la cantidad de socios cuyos códigos se encuentran comprendidos entre los valores leídos. Debe invocar a un módulo recursivo que reciba los valores leídos y retorne la cantidad solicitada.
3. Implementar un programa modularizado para una librería. Implementar módulos para:
 - a. Almacenar los productos vendidos en una estructura eficiente para la búsqueda por código de producto. De cada producto deben quedar almacenados su código, la cantidad total de unidades vendidas y el monto total. De cada venta se lee código de venta, código del producto vendido, cantidad de unidades vendidas y precio unitario. El ingreso de las ventas finaliza cuando se lee el código de venta -1.
 - b. Imprimir el contenido del árbol ordenado por código de producto.
 - c. Retornar el código de producto con mayor cantidad de unidades vendidas.
 - d. Retornar la cantidad de códigos que existen en el árbol que son menores que un valor que se recibe como parámetro.

- e. Retornar el monto total entre todos los códigos de productos comprendidos entre dos valores recibidos (sin incluir) como parámetros.
4. Una biblioteca nos ha encargado procesar la información de los préstamos realizados durante el año 2021. De cada préstamo se conoce el ISBN del libro, el número de socio, día y mes del préstamo y cantidad de días prestados. Implementar un programa con:
- a. Un módulo que lea préstamos y retorne 2 estructuras de datos con la información de los préstamos. La lectura de los préstamos finaliza con ISBN 0. Las estructuras deben ser eficientes para buscar por ISBN.
 - i. En una estructura cada préstamo debe estar en un nodo. Los ISBN repetidos insertarlos a la derecha.
 - ii. En otra estructura, cada nodo debe contener todos los préstamos realizados al ISBN. (prestar atención sobre los datos que se almacenan).
 - b. Un módulo recursivo que reciba la estructura generada en i. y retorne el ISBN más grande.
 - c. Un módulo recursivo que reciba la estructura generada en ii. y retorne el ISBN más pequeño.
 - d. Un módulo recursivo que reciba la estructura generada en i. y un número de socio. El módulo debe retornar la cantidad de préstamos realizados a dicho socio.
 - e. Un módulo recursivo que reciba la estructura generada en ii. y un número de socio. El módulo debe retornar la cantidad de préstamos realizados a dicho socio.
 - f. Un módulo que reciba la estructura generada en i. y retorne una nueva estructura ordenada ISBN, donde cada ISBN aparezca una vez junto a la cantidad total de veces que se prestó.
 - g. Un módulo que reciba la estructura generada en ii. y retorne una nueva estructura ordenada ISBN, donde cada ISBN aparezca una vez junto a la cantidad total de veces que se prestó.
 - h. Un módulo recursivo que reciba la estructura generada en g. y muestre su contenido.
 - i. Un módulo recursivo que reciba la estructura generada en i. y dos valores de ISBN. El módulo debe retornar la cantidad total de préstamos realizados a los ISBN comprendidos entre los dos valores recibidos (incluidos).
 - j. Un módulo recursivo que reciba la estructura generada en ii. y dos valores de ISBN. El módulo debe retornar la cantidad total de préstamos realizados a los ISBN comprendidos entre los dos valores recibidos (incluidos).

Módulo Imperativo

Práctica Adicionales

1. El administrador de un edificio de oficinas tiene la información del pago de las expensas de dichas oficinas. Implementar un programa con:
 - a) Un módulo que retorne un vector, sin orden, con a lo sumo las 300 oficinas que administra. Se deben cargar, para cada oficina, el código de identificación, DNI del propietario y valor de la expensa. La lectura finaliza cuando llega el código de identificación 0.
 - b) Un módulo que reciba el vector retornado en a) y retorne dicho vector ordenado por código de identificación de la oficina. Ordenar el vector aplicando uno de los métodos vistos en la cursada.
 - c) Un módulo que realice una búsqueda dicotómica. Este módulo debe recibir el vector generado en b) y un código de identificación de oficina. En el caso de encontrarlo, debe retornar la posición del vector donde se encuentra y en caso contrario debe retornar 0. Luego el programa debe informar el DNI del propietario o un cartel indicando que no se encontró la oficina.
 - d) Un módulo recursivo que retorne el monto total de las expensas.
2. Una agencia dedicada a la venta de autos ha organizado su stock y, tiene la información de los autos en venta. Implementar un programa que:
 - a) Genere la información de los autos (patente, año de fabricación (2010..2018), marca y modelo, finalizando con marca 'MMM') y los almacene en dos estructuras de datos:
 - i. Una estructura eficiente para la búsqueda por patente.
 - ii. Una estructura eficiente para la búsqueda por marca. Para cada marca se deben almacenar todos juntos los autos pertenecientes a ella.
 - b) Invoque a un módulo que reciba la estructura generado en a) i y una marca y retorne la cantidad de autos de dicha marca que posee la agencia.
 - c) Invoque a un módulo que reciba la estructura generado en a) ii y una marca y retorne la cantidad de autos de dicha marca que posee la agencia.
 - d) Invoque a un módulo que reciba el árbol generado en a) i y retorne una estructura con la información de los autos agrupados por año de fabricación.
 - e) Invoque a un módulo que reciba el árbol generado en a) i y una patente y devuelva el modelo del auto con dicha patente.
 - f) Invoque a un módulo que reciba el árbol generado en a) ii y una patente y devuelva el modelo del auto con dicha patente.
3. Un supermercado requiere el procesamiento de sus productos. De cada producto se conoce código, rubro (1..10), stock y precio unitario. Se pide:

- a) Generar una estructura adecuada que permita agrupar los productos por rubro. A su vez, para cada rubro, se requiere que la búsqueda de un producto por código sea lo más eficiente posible. El ingreso finaliza con el código de producto igual a 0.
 - b) Implementar un módulo que reciba la estructura generada en a), un rubro y un código de producto y retorne si dicho código existe o no para ese rubro.
 - c) Implementar un módulo que reciba la estructura generada en a), y retorne, para cada rubro, el código y stock del producto con mayor código.
 - d) Implementar un módulo que reciba la estructura generada en a), dos códigos y retorne, para cada rubro, la cantidad de productos con códigos entre los dos valores ingresados.
4. Una oficina requiere el procesamiento de los reclamos de las personas. De cada reclamo se ingresa código, DNI de la persona, año y tipo de reclamo. El ingreso finaliza con el código de igual a 0. Se pide:
- a) Un módulo que retorne estructura adecuada para la búsqueda por DNI. Para cada DNI se deben tener almacenados cada reclamo y la cantidad total de reclamos que realizó.
 - b) Un módulo que reciba la estructura generada en a) y un DNI y retorne la cantidad de reclamos efectuados por ese DNI.
 - c) Un módulo que reciba la estructura generada en a) y dos DNI y retorne la cantidad de reclamos efectuados por todos los DNI comprendidos entre los dos DNI recibidos.
 - d) Un módulo que reciba la estructura generada en a) y un año y retorne los códigos de los reclamos realizados en el año recibido.