

PROGRAMACIÓN I

TP 4

TRABAJANDO CON LISTAS

UNIDAD 2
ESTRUCTURAS DINÁMICAS 1

Autor de contenidos:
Nicolás Battaglia



OBJETIVOS

Realizar programas utilizando listas y recursividad

ENUNCIADO

Recursividad

- 1) Implementación de un método recursivo que reciba un parámetro de tipo entero y luego llame en forma recursiva con el valor del parámetro menos 1. Deberá mostrar por consola : "10 9 8 7 6 5 4 3 2 1 "
- 2) Obtener el factorial de un número. Recordar que el factorial de un número es el resultado que se obtiene de multiplicar dicho número por el anterior y así sucesivamente hasta llegar a uno. Ej. el factorial de 4 es $4 * 3 * 2 * 1$ es decir 24.
- 3) Obtener una función recursiva para ordenar los elementos de un vector. Se deberá mostrar desordenado, ordenarlo y mostrarlo ordenado.

Listas

Ejercicio 1:

Se tiene una lista de pacientes de un hospital y se desea realizar un sistema sobre registro de los mismos.

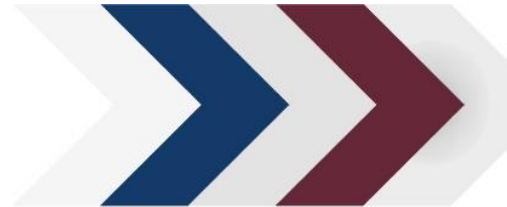
Los datos de los pacientes son código, nombres, apellido, dirección y teléfono.

El sistema debe permitir:

- a. Registrar un nuevo paciente
- b. Eliminar paciente (en cualquier posición)
- c. Actualizar Pacientes (modificar algún dato)
- d. Agregar después del seleccionado.
- e. Mostrar Listado

Ejercicio 2:





Realice un sistema que permita el registro de alumnos de una institución que permita registrar, modificar, eliminar los mismos. Los datos de los alumnos son nombres, apellido, DNI, fecha de nacimiento, dirección y teléfono.

El sistema debe permitir:

- a. Registrar un nuevo alumno
- b. Eliminar alumno (en cualquier posición)
- c. Actualizar alumno (modificar algún dato)
- d. Agregar después del seleccionado
- e. Agregar antes del seleccionado
- f. Mostrar Listado de los alumnos actuales

Ejercicio 3:

Un cine pequeño nos propone hacer una aplicación para controlar las personas que ingresa al mismo, para los grandes estrenos de películas.

Unos conjuntos de personas esperaran para sacar una entrada, tendremos que calcular la entrada según la edad de la persona (mínimo 5 años).

Las edades de las personas se generan aleatoriamente entre 5 y 60 años.

Al final, deberemos mostrar la **cantidad total recaudada**. El número de personas de la lista se elige al azar entre 0 y 50.

Además, se debe permitir:

- a. Llevar un registrar de las nuevas personas que ingresan a la lista
- b. Poder eliminar a la persona que se va de la lista (en cualquier posición)
- c. Actualizar los datos de la persona (modificar algún dato)
- e. Agregar después del seleccionado
- d. Mostrar Listado de las personas.

Ejercicio 4:

En el jardín hay torneo del juego el pato ñato. Para la inscripción del mismo se requiere saber el nombre, apellido, edad y DNI.

Se va a llevar un registro sistémico:

- a. Registrar de los chicos que participan del juego.





- b. Poder eliminar a los chicos que van perdiendo (en cualquier posición)
- c. Actualizar los datos de los chicos (modificar algún dato)
- d. Agregar después del seleccionado.
- e. Mostrar Listado de los chicos.

Nota: Se requiere jugar en forma circular

Ejercicio 5:

Se tiene una lista de pacientes de un hospital y se desea realizar un sistema sobre registro de los mismos.

Los datos de los pacientes son código, nombres, apellido, dirección y teléfono.

El sistema debe permitir:

- a. Registrar un nuevo paciente
- b. Eliminar paciente (en cualquier posición)
- c. Actualizar Pacientes (modificar algún dato)
- d. Mostrar Listado

NOTA: Realizar el ejercicio en forma de lista doblemente enlazada

Ejercicio 6:

Un cine pequeño nos propone hacer una aplicación para controlar las personas que ingresa al mismo, para los grandes estrenos de películas.

Unos conjuntos de personas esperaran para sacar una entrada, tendremos que calcular la entrada según la edad de la persona (mínimo 5 años).

Las edades de las personas se generan aleatoriamente entre 5 y 60 años.

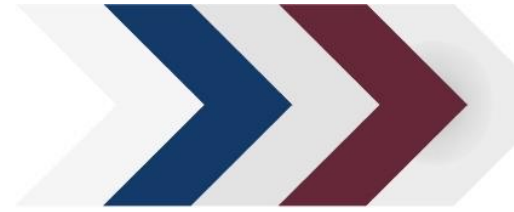
Al final, deberemos mostrar la **cantidad total recaudada**. El número de personas de la lista se elige al azar entre 0 y 50-

Además, se debe permitir:

- a. Llevar un registrar de las nuevas personas que ingresan a la lista
- b. Poder eliminar a la persona que se va de la lista (en cualquier posición)
- c. Actualizar los datos de la persona (modificar algún dato)
- d. Mostrar Listado de las personas.

NOTA: Realizar el ejercicio en forma de lista doblemente enlazada





Ejercicio 7:

Completar la implementación de las funciones no desarrolladas en el ejemplo provisto en la unidad, a saber:

1. `public void IntercambiarDerecha(int numero)`
2. `public void IntercambiarIzquierda(int numero)`
3. `public void Intercambiar(int numero1, int numero2)`