

**Universidad Mayor de San Andrés**  
**Facultad de Ciencias Puras y Naturales**  
**Carrera de Informática**



**PROGRAMACIÓN II**  
**INF-121**  
**PRÁCTICA GENERAL I/2024**

AUXILIAR	PARALELO	HORARIO
Univ. Ramirez Luna Roberto Antonio	A	Viernes 14:00-16:00
Univ. Ajhuacho Tarquino Luis David	B	Viernes 12:00 - 14:00
Univ. Tintaya Montecinos Alan Hernan	C	Lunes 14:00-16:00
Univ. Ramirez Castro Alina Erika	D	Por definir

- Debe resolver los problemas que indique el auxiliar que cursa.
- Presentar al auxiliar que cursa los ejercicios resueltos de forma ordenada y a computadora.
- Cada ejercicio debe considerar el problema, su diagrama de clases y la solución de los problemas.

**ESTUDIANTE QUE PRESENTE CUALQUIER PRÁCTICA ESTÁ**  
**INHABILITADO PARA EL EXAMEN DE LIBERACIÓN DE**  
**AUXILIATURA**

# COMPLEJIDAD ALGORÍTMICA

## 1. Algoritmo

```
public static int sumaArreglo(int[] arr) {
    int suma = 0;
    for (int num : arr) {
        suma += num;
    }
    return suma;
}
```

## 2. Algoritmo

```
public static int fibonacciIterativo(int n) {
    if (n <= 1)
        return n;
    int a = 0, b = 1;
    for (int i = 2; i <= n; i++) {
        int temp = a + b;
        a = b;
        b = temp;
    }
    return b;
}
```

1.

## 3. Algoritmo

```
public static int busquedaLineal(int[] arr, int valor) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == valor) {
            return i;
        }
    }
    return -1;
}
```

## 4. Algoritmo

```
public static int busquedaBinaria(int[] arr, int valor) {
    int izquierda = 0;
    int derecha = arr.length - 1;
    while (izquierda <= derecha) {
        int medio = izquierda + (derecha - izquierda) / 2;
        if (arr[medio] == valor) {
            return medio;
        } else if (arr[medio] < valor) {
            izquierda = medio + 1;
        } else {
            derecha = medio - 1;
        }
    }
    return -1;
}
```

## 5. Algoritmo

```
public static void bubbleSort(int[] arr) {
    int n = arr.length;
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}
```

## 6. Algoritmo

```
public static int[][] multMatrices(int[][] A, int[][] B) {
    int n = A.length;
    int[][] resultado = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            for (int k = 0; k < n; k++) {
                resultado[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    return resultado;
}
```

## 7. Algoritmo

```
public static boolean esPrimo(int num) {
    if (num <= 1)
        return false;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0)
            return false;
    }
    return true;
}
```

## 8. Algoritmo

```
public static int buscarSubcad(String cadena, String subcadena) {
    int n = cadena.length();
    int m = subcadena.length();
    for (int i = 0; i <= n - m; i++) {
        int j;
        for (j = 0; j < m; j++) {
            if (cadena.charAt(i + j) != subcadena.charAt(j))
                break;
        }
        if (j == m)
            return i;
    }
    return -1;
}
```

## 9. Algoritmo

```
public static List<Integer> criba(int n) {
    boolean[] esPrimo = new boolean[n + 1];
    Arrays.fill(esPrimo, true);
    esPrimo[0] = false;
    esPrimo[1] = false;

    for (int i = 2; i * i <= n; i++) {
        if (esPrimo[i]) {
            for (int j = i * i; j <= n; j += i) {
                esPrimo[j] = false;
            }
        }
    }

    List<Integer> primos = new ArrayList<>();
    for (int i = 2; i <= n; i++) {
        if (esPrimo[i]) {
            primos.add(i);
        }
    }
    return primos;
}
```

## 10. Algoritmo

```
public static int[][] generarMatrixCaracol(int n) {
    int[][] matriz = new int[n][n];
    int inicioFila = 0, finFila = n - 1;
    int inicioColumna = 0, finColumna = n - 1;
    int contador = 1;

    while (inicioFila <= finFila && inicioColumna <= finColumna) {
        for (int i = inicioColumna; i <= finColumna; ++i) {
            matriz[inicioFila][i] = contador++;
        }
        inicioFila++;

        for (int i = inicioFila; i <= finFila; ++i) {
            matriz[i][finColumna] = contador++;
        }
        finColumna--;

        if (inicioFila <= finFila) {
            for (int i = finColumna; i >= inicioColumna; --i) {
                matriz[finFila][i] = contador++;
            }
            finFila--;
        }

        if (inicioColumna <= finColumna) {
            for (int i = finFila; i >= inicioFila; --i) {
                matriz[i][inicioColumna] = contador++;
            }
            inicioColumna++;
        }
    }
    return matriz;
}
```

## PILAS

1. Mediante pilas, implementar un programa que convierta una expresión infija (por ejemplo, "2 + 3 \* 4") a su forma postfija (notación polaca inversa, ejemplo, "2 3 4 \* +").
2. Sea una pila de Cuadernos < marca, nrohojas, tipo, tamaño >. Se pide:
  - a. ¿Cuántos cuadernos son de la marca X?
  - b. Llevar los cuadernos de la marca X, al final de la pila
  - c. Ordenar la pila de cuadernos por numero de hojas ascendentemente.
3. Dada una pila de información para registro de pacientes en una clinica <nombre, apellido, ci, celular, especialidad > y en otros pila la información de especialidades con su respectivo médico <nomEspecialidad, nomDoctor>. Se pide:
  - a. Mostrar todos los pacientes que fueron por la especialidad X
  - b. Añadir dos especialidades nuevas.
  - c. Mostrar el nombre del doctor que atendió al paciente de nombre X y apellido Y.
4. Se tiene una pila de qué almacenan Zapatos <talla, color, modelo>. Se pide:
  - a. Verificar que los zapatos estén en orden es decir que si después de talla  $a$  esta la talla  $b$ , se debe cumplir que  $(talla_a \leq talla_b)$ .
  - b. Agrupar los zapatos por marca.
5. En una panadería muy famosa la atención al cliente es primordial para su crecimiento. La panadería se toma muy en serio la atención preferencial a adultos mayores y niños. Entonces, Los clientes <nombre, edad, celular>..
  - a. Una vez que ingresan todos a la panadería, serán reordenados y los primeros en salir serán las personas mayores (>60 años) y niños (<12 años). Se pide ordenarlos de esta manera.
  - b. Mostrar a los clientes que su celular comience con el número 6.
  - c. Vaciar la panadería, que ya no tenga clientes para atender.
6. En un restaurante, se lleva un registro de los pedidos de los clientes. Existe una pila de información para registrar los pedidos de los clientes, que incluye los siguientes datos: nombre del cliente, número de mesa, platos solicitados y total a pagar. Además, hay otra pila que contiene información sobre los platos disponibles y sus precios. Se requiere:
  - a. Mostrar todos los pedidos realizados por los clientes en una mesa específica.
  - b. Añadir dos nuevos platos al menú.
  - c. Mostrar el precio total del pedido realizado por el cliente con nombre X en la mesa Y.
7. En una oficina, se gestiona el trabajo de los empleados utilizando tres pilas diferentes: una para las tareas pendientes, otra para los empleados y sus respectivas tareas asignadas, y una tercera para los proyectos en curso.

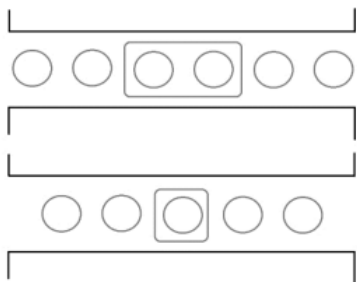
La pila de tareas pendientes contiene información sobre las tareas <nombreTarea, prioridad, fechaLimite>. La pila de empleados y tareas asignadas contiene información sobre los empleados y las tareas asignadas nombreEmpleado, departamento, nombreTarea>. La pila de proyectos en curso contiene información sobre los proyectos en los que están trabajando los empleados <nombreProyecto, estado, nombreEmpleado>. Se pide:

  - a. Actualizar el estado de los proyectos en curso, como "en progreso" o "completado"
  - b. Mostrar todas las tareas pendientes ordenadas por prioridad, de alta a baja.
  - c. Mostrar todas las tareas asignadas a un empleado específico, incluyendo el estado de cada tarea y la fecha límite.

- d. El empleado de nombre X fue despedido, eliminar sus tareas y proyectos donde él estaba trabajando.
8. En una tienda, se lleva un registro de los turnos de trabajo de los empleados. Existe una pila de información para registrar los turnos de los empleados, que incluye los siguientes datos: nombre del empleado, horario de entrada, horario de salida y área asignada. Además, hay otra pila que contiene información sobre los departamentos de la tienda y los empleados asignados a cada departamento. Se requiere:
- a. Mostrar todos los turnos asignados a los empleados en un departamento específico.
  - b. Programar dos nuevos turnos para empleados.
  - c. Mostrar el horario de trabajo del empleado con nombre X en el departamento Y.
9. La unidad del Área de Investigación, gestiona sus investigaciones utilizando una pila para los proyectos de investigación pendientes. Cada proyecto de investigación pendiente contiene información sobre el proyecto, incluyendo el título del proyecto, la descripción, el investigador principal, la fecha de inicio y la fecha de finalización prevista (Formato fecha dd/mm/aaaa). Se pide:
- a. Mostrar todos los proyectos asignados a un investigador específico, incluyendo el estado de cada proyecto y la fecha límite.
  - b. Agregar nuevos proyectos a la pila de proyectos pendientes
  - c. Mostrar los proyectos del año x entre el intervalo de meses  $m_i < m_j$
10. En una tienda de electrónica, se gestiona el inventario de productos utilizando dos pilas diferentes: una para los productos en stock y otra para los productos vendidos. Cada producto en stock <codigo\_prod, nombre, marca, precio,cantidad\_disponible>. Cada producto vendido<nombre\_cliente, codigo\_prod, cantidad, precio>. Se pide:
- a. Se realiza una venta total del producto X, el producto vendido se elimina de la pila de productos en stock y se agrega a la pila de productos vendidos.
  - b. Llegan k productos, se actualiza el stock en la pila de productos en stock para mantener actualizado el inventario de la tienda, si el producto es nuevo y no existe crear un nuevo stock.
  - c. Calcular las ganancias totales obtenidas de todas las ventas registradas en la pila de productos vendidos.

COLAS

1. Dada una cola simple de colores se le pide eliminar el o los elementos de en medio:



2. Dada una cola simple de exámenes<nombre Estudiante, sigla materia, nota> :

- a. Ordenar de mayor a menor según la nota.
- b. Mostrar la nota que más se repite.

3. Dada una cola circular de números A y otra cola circular de números B, intercambiar el i-esimo elemento de A con el j-esimo de B sin usar estructuras auxiliares.

4. Dada una Cola de objetos Libro<título,nro de páginas> se pide eliminar aquellos Libros con un número de páginas impar o si el título del libro empieza en una vocal.

5. Sea una cola simple de Mascotas, y una cola simple de Atenciones de una clínica veterinaria, Mascota <idMascota, nombre, sexo, tipo> , Atención <fecha, idMascota>.

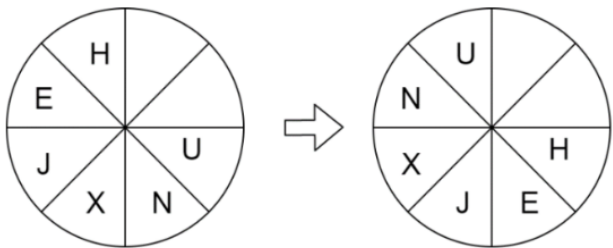
Se pide:

- a. Mostrar el nombre de las mascotas que fueron atendidas en la clínica veterinaria en la fecha X
- b. Eliminar la mascota con el idMascota X si y sólo si su nombre es “Tavo”.
- c. Contar mascotas “hembra” que fueron atendidas en la clínica veterinaria.

6. Sea una cola de Michis<nombre, edad, género> ordenar la cola según la edad ascendentemente sin usar ninguna estructura auxiliar.

7. Sea una cola circular de caracteres se pide:

- a. Ver si existen en la cola todas las vocales.
- b. Invertir el orden de la cola sin el uso de estructuras auxiliares.



8. Sea una Cola Simple de evaluaciones <ci, nota> de la materia INF131 y una cola Circular de estudiantes <ci, nombre>.

- a. Ordenar la Cola Simple de evaluaciones de acuerdo al orden de los estudiantes.
- b. Mostrar el nombre del (los) estudiante(s) con la menor nota acumulada.

9. Sea una cola circular, agregar N elementos antes del i-esimo elemento, sin usar estructuras auxiliares

10. Se maneja una cola simple de transacciones bancarias con los siguientes atributos: Transacción (idTransacción, tipo, monto)

Realizar las siguientes operaciones:

- a. Agregar una nueva transacción a la cola.
- b. Mostrar el tipo de transacción (depósito o retiro) con el monto más alto.
- c. Calcular el monto total de todas las transacciones.
- d. Eliminar la transacción más antigua de la cola y mostrar sus detalles.

## MÚLTIPLES PILAS/COLAS

1. Dada una pila que contiene colas circulares de nombres de persona, se pide:
  - a. Invertir cada cola sin usar estructuras auxiliares
  - b. Ordenar la pila según el número de elementos de las colas, ascendentemente.
2. La veterinaria “El Michi feliz” almacena la información de las mascotas en una cola circular de pilas. Donde cada pila almacena el registro de mascota<nombreMascota, idMascota, tipoMascota, fechaRegistro>.
  - a. Se pide:
  - b. Mover a la primera pila todos los registros de las mascotas de tipo “Gato”.
  - c. Eliminar a la mascota de nombre **X** que se registró en la fecha **Y** (“dd/mm/yyyy”).
  - d. Verificar si existe la mascota de tipo **“Perro”** de nombre **“Odair”**.
3. La OPENINFO de la Carrera expondrá los proyectos que se desarrollan en las diferentes materias, para lo cual se establecerán estructuras para el registro de los proyectos que participaran. La información estará almacenada de la siguiente manera:
  - Cola simple que almacena objetos materia<sigla, Pila de proyectos>
    - Pila de proyectos<título del proyecto, Cola Circular de Participantes>
      - Cola Circular de Participantes<nombre de los participantes>

Se pide:

  - a. Cuántos proyectos se presentarán en cada materia
  - b. Mostrar la sigla de la materia, el título del proyecto que tengan 3 participantes
  - c. Mostrar la sigla de las materias con menos participantes.
4. Dada una múltiple pila de carreras<nombreCarrera, nroEstudiantes, facultad> de la Universidad Mayor de San Andrés.

Se pide:

  - a. Mostrar la(s) carrera(s) con más estudiantes.
  - b. Eliminar todas las carreras de la Facultad X.
5. La biblioteca “El Michi lector” tiene la información de sus libros almacenada a través de una MultiPila de Libros<Título, autor, idLibro, stock>.

Se pide:

  - a. Eliminar de la MultiPila todos los libros con stock igual a 0.
  - b. Mostrar el contenido de todas las pilas en las que existan al menos un libro del autor X.
  - c. Mostrar el número de la pila en la que esté el libro de título X del autor Y (asumiendo que solo puede existir el libro solicitado en una sola pila).
6. Sea una múltiple cola de mascotas<nombreMascota, edad, sexo, tipo>.

Se pide:

  - a. Contar cuantas mascotas del tipo X y del sexo hembra existen.
  - b. Eliminar a todas las mascotas de cada cola que sean menores a 2 años.
  - c. De cada cola llevar a las mascotas de tipo “gato” al principio de esta.
7. Sea una múltiple pila de personas<CI, nombre, edad, género>.

Se pide:

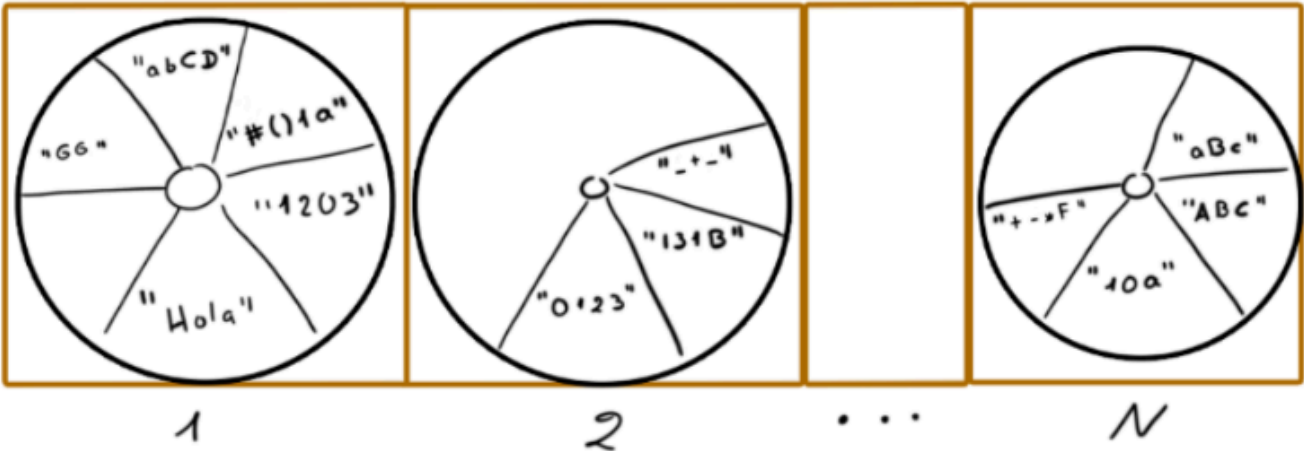
  - a. Mostrar la cantidad de varones y mujeres de cada pila.
  - b. Ordenar los elementos de la pila con respecto a su edad.
  - c. Insertar una nueva persona en la i-ésima posición de la múltiple pila y en la j-ésima posición de la pila i-ésima.
8. Sea una múltiple pila de notas <nombre, nota final>, la nota mínima de aprobación es 51.

Se pide:

  - a. Mostrar la pila que tiene más aprobados y la pila que tiene más reprobados.
  - b. Mostrar el promedio total de TODOS los aprobados.



9. Sea una múltiple cola circular de cadenas.



- Se pide:
- a. De todas las colas circulares mover a la primera cola circular todas las cadenas que solo tengan caracteres entre el '0' al '9', mover a la segunda cola circular todas las cadenas que solo tengan caracteres alfabéticos (tanto mayúsculas como minúsculas) y en la cola circular 3 las cadenas que no cumplan las anteriores condiciones.
  - b. Eliminar todas las cadenas que sean palíndromos de la i-ésima cola.

10. La forma de almacenamiento del banco “EIDinerin”, almacena la información tanto de los clientes como de los servicios de la siguiente manera:
- ColaSimpleCliente<nombre, Ci, edad>
  - MultiPilaServicio<idServicio, tipo> (correspondiendo la primera pila de la MultiPila con el primer cliente en la cola).

Cola simple de clientes

Juan	María	Pedro	Luis
516165	51651	84984	879871
25	18	35	55

MultiPila de servicios

<div>L21, deposito</div> <div>C1, transferencia</div>	<div>P2, consulta</div>	<div>P3, consulta</div> <div>C12, deposito</div>	<div>C3, deposito</div> <div>C3, pagoServicio</div> <div>P5, consulta</div> <div>P5, retiro</div>
---	-------------------------	--	---

- Se pide:
- a. Mostrar a los clientes que accedieron algún servicio de tipo “consulta”.
  - b. Mostrar a los clientes con más servicios.
  - c. Verificar si el cliente de CI X tiene un servicio de tipo “depósito”.

## LISTAS SIMPLES

1. Sea una lista de números enteros:
  - a. Eliminar los números pares
  - b. Insertar un nuevo número después de cada número impar
  - c. Dividir en dos listas, una de números pares y otra de impares. No crear nuevos nodos.
2. Dadas dos listas simples normales ordenadas de números enteros insertar los nodos de la segunda lista en la primera lista, de tal manera que la primera se encuentre siempre ordenada.
3. Sea una lista simple de mensajes recibidos en un celular (#celular, asunto, fecha, hora, texto) que realice las siguientes tareas:
  - a. Mostrar los mensajes recibidos con asunto "ninguno"
  - b. Llevar todos los mensajes recibidos en la fecha X, al principio de la lista
  - c. Eliminar los mensajes recibidos desde el celular con número X.
4. Sean 2 listas simples de nombres, se pide realizar:
  - a. Unir B a la lista A.
  - b. Insertar B después de i-ésimo nodo de A.
  - c. Llevar el último nodo de B al principio de A.
  - d. Llevar los k-últimos nodos de B al principio de A.
5. En una guardería, dada una lista simple circular que controla a los niños, dónde la sección puede ser: pequeñines(desde los más pequeños a 1 año) y niditos(de 2 a 4 años).
  - a. Mostrar a los niños por secciones.
  - b. Separar a los niños por secciones en dos listas distintas.
6. Dada una lista simple enlazada que almacena objetos computadoras que tiene como atributos (marca, modelo, precio). Resolver solo manejando punteros, sin crear listas auxiliares.
  - a. Realizar el diagrama de clases.
  - b. Determinar la computadora más costosa.
  - c. Eliminar la computadora de marca X. d) Insertar una computadora en la posición i-ésima.
  - d. Ordenar la lista por precio.
  - e. Insertar una nueva computadora en el lugar en la posición correspondiente se supone que la lista está ordenada.
  - f. Invertir la lista.
7. Dada una lista simple de palabras, se requiere realizar las siguientes operaciones:
  - a. Eliminar las palabras que contengan menos de tres caracteres.
  - b. Reemplazar cada palabra por su longitud (cantidad de caracteres).
  - c. Dividir la lista en dos, una con palabras que comienzan con vocal y otra con palabras que no.
8. En un sistema de gestión de biblioteca, se tiene una lista simple de libros con los atributos (título, autor, género, año de publicación). Realizar las siguientes operaciones:
  - a. Mostrar todos los libros del autor X.
  - b. Ordenar la lista por año de publicación.
  - c. Eliminar todos los libros de un género específico.
  - d. Insertar un nuevo libro después del libro del año X.

- 9.** Dada una lista simple de empleados de una empresa con atributos (nombre, edad, salario), realizar las siguientes tareas:
- a. Calcular el promedio de salarios de los empleados.
  - b. Encontrar al empleado más joven y al más viejo.
  - c. Aumentar el salario en un porcentaje dado para todos los empleados menores de 30 años.
  - d. Eliminar a los empleados cuyo salario esté por debajo de cierto umbral.
- 10.** En un sistema de gestión de tareas, se tiene una lista simple de tareas pendientes con atributos (nombre de la tarea, descripción, fecha de vencimiento). Realizar las siguientes operaciones:
- a. Mostrar todas las tareas vencidas.
  - b. Insertar una nueva tarea en orden según su fecha de vencimiento.
  - c. Eliminar una tarea específica por su nombre.
  - d. Dividir la lista en dos, una con tareas pendientes para hoy y otra para después de hoy.

## LISTAS DOBLES

1. Se maneja el control de los huéspedes y habitaciones de un hotel de 5 estrellas mediante listas, habiendo dos listas: una lista doble que contiene objetos habitación y una lista simple circular de objetos huésped
  - a. Verificar y mostrar si existe alguna habitación que está excediendo su capacidad.
  - b. La habitación número X queda vacía, entonces se pide eliminar a los huéspedes de dicha habitación.
  - c. Una familia de 3 miembros llegó al hotel y piden la habitación número X, verificar si esta habitación tiene una capacidad de 3 personas, si es así agregar a la familia a la lista de huéspedes.
2. En las salas de un cine se manejan los asientos para una función mediante una lista doble normal con objetos asiento fila = A, B C ..., Columna = 1, 2, 3, ... y otra lista simple normal con objetos cliente.
  - a. Verificar si el asiento de fila X y columna Y está disponible.
  - b. Mover al cliente con nombre X al asiento de su izquierda si existe y está vacante.
3. Considerando que los libros de la biblioteca municipal están organizados por áreas (Infantil, Matemáticas, Ciencias políticas, etc.). Se tiene la Lista Doble Normal que almacena las diferentes áreas<Área, NroLibros> y la Lista Doble Circular que almacena la información de los libros agrupados por áreas según el orden de la lista de las áreas.
  - a. Mostrar cada área y toda la información de sus libros.
  - b. Agregar R libros al área X.
4. Sea una lista doble circular de estudiantes donde cada nodo almacena se pide:
  - a. Hallar el promedio de los estudiantes aprobados
  - b. Ordenar ascendentemente por la nota de cada estudiante.
  - c. Eliminar a los estudiantes reprobados y llevarlos a otra lista.
5. Sea una Lista Simple Circular que almacena la información de Productos
  - a. Dividir por la mitad la lista, obteniendo como resultado dos listas simples circulares. No crear nuevos nodos
  - b. Con los 'k' primeros nodos, formar una lista simple circular, el resto se mantienen en la lista circular. No crear nuevos nodos
6. Sea una lista Doble normal que almacena archivos
  - a. Calcular el tamaño promedio de los archivos del tipo X.
  - b. Verificar si entre los archivos del tipo X existe un archivo con un tamaño mayor a 50 GB.
  - c. Eliminar los archivos del año 2020
7. En una aplicación de gestión de inventario de un almacén, se tiene una lista doble de productos con atributos (nombre, cantidad en stock, precio). Realizar las siguientes operaciones:
  - a. Calcular el valor total del inventario.
  - b. Encontrar el producto más caro y el más barato.
  - c. Incrementar la cantidad en stock de un producto específico.
  - d. Eliminar todos los productos que tengan una cantidad en stock igual a cero.
8. En un sistema de gestión de pedidos en un restaurante, se manejan las órdenes mediante una lista doble de pedidos con atributos (número de pedido, lista de platos). Realizar las siguientes operaciones:
  - a. Mostrar todos los pedidos realizados en una fecha específica.
  - b. Insertar un nuevo pedido después del pedido con número X.
  - c. Eliminar un pedido específico por su número.
  - d. Dividir la lista en dos, una con pedidos para llevar y otra para comer en el lugar.

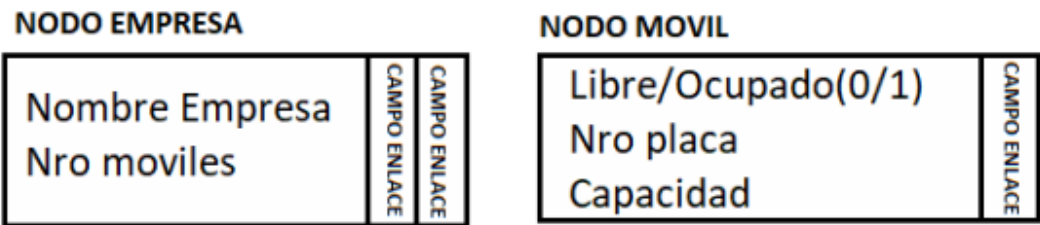
9. En un sistema de gestión de empleados de una empresa, se manejan los datos de los empleados mediante una lista doble con atributos (nombre, edad, departamento). Realizar las siguientes operaciones:
- Mostrar a todos los empleados del departamento X.
  - Cambiar el departamento de un empleado específico.
  - Eliminar a los empleados mayores de cierta edad.
  - Insertar un nuevo empleado después del empleado con nombre X.
10. En una aplicación de gestión de música, se tiene una lista doble de canciones con atributos (título, artista, duración). Realizar las siguientes operaciones:
- Mostrar todas las canciones del artista X.
  - Ordenar la lista por duración de la canción.
  - Eliminar todas las canciones de un artista específico.
  - Insertar una nueva canción después de la canción con título X.

# MÚLTIPLES LISTAS

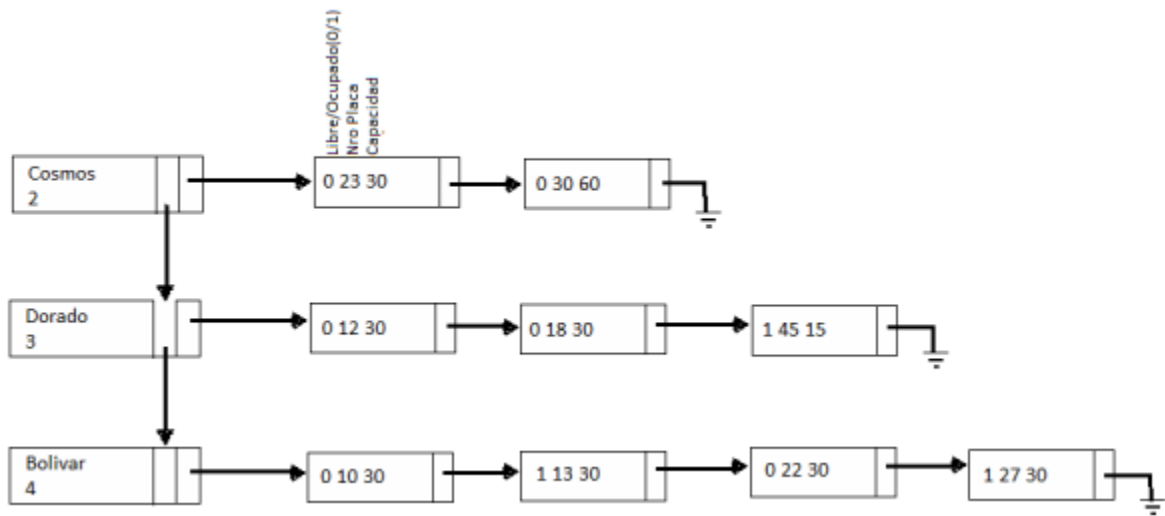
1. Es una Lista Principal donde cada nodo almacena una Lista de letras (las letras forman una palabra). Se pide ordenar la lista principal alfabéticamente por la primera letra de la lista de letras.
2. Sea una Lista simple donde cada nodo contiene una Lista simple de números primos. La primera lista almacena en sus nodos números primos de 1 dígito, la segunda almacena números primos de 2 dígitos y así sucesivamente.
  - a. Generar las listas para números primos de 1 a N dígitos.
  - b. Contar los números de la i-ésima lista que son también Fibonacci.
3. Sean las siguientes listas:
  - Lista simple de estudiantes
  - Lista simple de materias
  - Lista simple de calificaciones

Se pide:

- a. Modelar el diagrama de clases.
  - b. Mostrar los datos de los estudiantes que hayan reprobado, en la materia con nombre X
  - c. Calcular el promedio de todos los estudiantes aprobados de cada materia
4. La Terminal de buses desea manejar la información de las empresas de transporte interdepartamental de la manera más dinámica y operativa posible. Para tal efecto, contrato los servicios de los alumnos de Informática quienes proponen la siguiente estructura para el manejo de los datos.



A continuación se muestra un ejemplo de la lista múltiple.

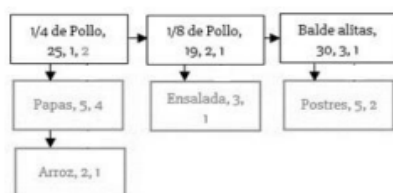


Se pide implementar los siguientes métodos:

- a. Adicionar una Empresa al final de la lista de empresas, adiEmpresa("Empresa").
- b. Adicionar un vehículo en la lista de vehículos de la empresa X en la posición de placa que le corresponda (asumiendo que la lista de vehículos está ordenado por placa) Adivehiculo(X, placa).
- c. Eliminar el primer vehículo ocupado de la empresa X. Eliocupado(X).
- d. Mostrar las placas de los vehículos libres y el nombre de empresa de todos los vehículos con capacidad mayor a W, VehiculoLibre().

5. Se quiere controlar los laboratorios de la carrera de informática mediante listas. Sean las siguientes listas: Lista de Laboratorios (idLaboratorio, ubicación, ListaMaterias, ListaResponsable) Lista de Materias (Sigla, ListaEstudiantes) Lista de responsables (nombreResponsable, ci) Lista de Estudiantes (nombreEstudiante , ci, edad) Realizar:
  - a. Mostrar el idLaboratorio donde se encuentra la materia con sigla "INF121"
  - b. Mostrar la ubicación(es) donde esta el responsable con ci X.
  - c. Mostrar los estudiantes que están en la materia Z.
  - d. Mostrar la sigla de la materia donde se encuentra el estudiante Y.
  
6. Dada una lista doble de estudiantes (nombre, ListaMaterias, semestre, gestión) , Lista simple de materias(materia, 1parcial, 2parcial, exfinal, notaAux ), Materia(sigla,nombre) se pide realizar los siguientes procesos:
  - a. Mostrar los estudiantes que aprobaron sin auxiliatura la materia Y.
  - b. Mostrar el estudiante(s) que obtuvo la mayor nota en la materia "INF131".
  - c. Mostrar el promedio del estudiante con ci X.
  - d. Eliminar a los estudiantes reprobados en más de dos materias.
  
7. Considerando que se quiere controlar la programación semanal de la programación en los canales de Televisión, para lo cual se tiene las siguientes Listas Lista de Canales de TV Objeto Canal: NroCanal, NombreCanal, Dirección, Lista de Programas Lista de Programación de los Canales Objeto Programa: NombreProg, tipo, Duración, Horalnicio, Día Resolver los siguientes problemas:
  - a. Mostrar el NroCanal, NombreCanal y los programas de tipo película que se emitirán el fin de semana en los mismos.
  - b. Mostrar el Nrocanal, el día y el tipo donde se emite el programa X.
  - c. Ordenar ascendentemente los canales de televisión según el número de programas.
  
8. Una cadena de comida rápida decidió contratar un delivery para realizar sus pedidos, entonces se organizó de la siguiente manera:
  - Lista de sucursal: nombre de sucursal, lista de clientes.
  - Lista de clientes: identificación, minutos de lejanía del local, lista de pedido.
  - Lista de pedidos: producto, precio, cantidad, cantidad de extras para el pedido y una lista de extras
  - Lista de extras: nombreExtra, precio, cantidad

ejemplo de Lista de extras



Se pide:

- a. Calcular el precio total para el cliente con identificación X.
  - b. Añadir un postre extra a aquellos clientes que hicieron un pedido de la comida X y al menos añadieron tres comidas extras a su pedido.
  - c. Es la hora X (Ejemplo entrada: 10:00) en la sucursal Y, mostrar a qué hora se llegará a las casas de todos los clientes.
- 
9. Se maneja una aplicación de gestión de eventos deportivos con las siguientes listas:
    - Lista simple de eventos(nombre, lugar, fecha, LDoblePx E)
    - Lista simple de deportes (nombre, categoría, reglas)
    - Lista doble de participantes por evento (nombre, país, edad)

Realizar las siguientes operaciones:

    - a. Mostrar todos los eventos en los que participa un país específico.
    - b. Mostrar todos los participantes de un evento específico.
    - c. Calcular la edad promedio de los participantes por deporte.
    - d. Eliminar un participante específico de un evento

**10.** Se desea crear un sistema de gestión de libros de una biblioteca con las siguientes listas:

- Lista simple de libros (título, autor, género, año de publicación, LDoblePxL)
- Lista simple de autores (nombre, nacionalidad, fecha de nacimiento)
- Lista doble de préstamos por libro (nombre del estudiante, fecha de préstamo, fecha de devolución)

Realizar las siguientes operaciones:

- a. Mostrar todos los libros de un autor específico.
- b. Mostrar todos los libros prestados por un estudiante específico.
- c. Calcular la cantidad promedio de días que un libro está prestado.
- d. Eliminar un préstamo específico de un libro.

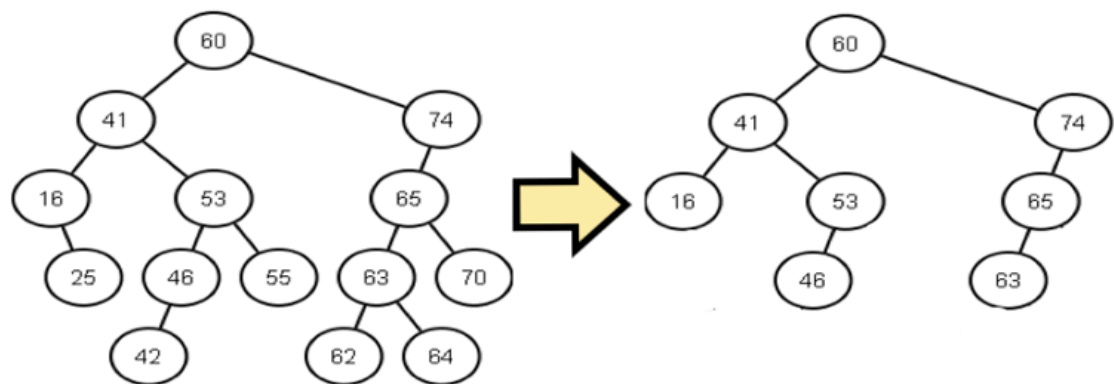


# RECURSIVIDAD

1. Dado un entero N, hallar el Factorial de N de forma recursiva. El factorial de un número entero se define como:  
$$F(n) = n! = 1 * 2 * \dots * (n - 1) * n$$
$$F(n) = n! = n * F(n - 1) \quad \text{con } F(0) = 0! = 1$$
2. Invertir una cadena recursivamente, por ejemplo:  
Programacion → noicamargorP
3. La sucesión de Fibonacci viene definida por la siguiente recurrencia:  
 $F_{n+2} = F_n + F_{n+1}$ . Con valores iniciales  $F_0 = 0$  y  $F_1 = 1$ .
  - a. Calcular el enésimo término de la sucesión recursivamente.
  - b. Realizar la suma de los primeros N números Fibonacci recursivamente.
4. Sea una matriz NxN que almacena números, recursivamente resolver:
  - a. Mostrar el mayor de cada fila.
  - b. Mostrar la Diagonal principal.
  - c. Intercambiar la fila i con la columna i.
5. Dada una Pila de objetos Motocicleta <color, marca, placa> se pide recursivamente:
  - a. Verificar si existe alguna Motocicleta de la marca "BMW".
  - b. Contar a las Motocicletas del color "Rojo".
6. Dada una Cola de objetos Libro <título, género, nroPaginas>, se pide de forma recursiva:
  - a. Cuantas Películas con nroPaginas mayor a X existe.
  - b. Verificar si existe la Película con título X. De no existir, añadir uno al principio de la pila.
7. En los juegos olímpicos sea una Lista de Medallistas <disciplina, nombre, medalla, país>. Se pide recursivamente:
  - a. Mostrar cuántas medallas de "Oro", "Plata" y "Bronce" obtuvo el país X.
  - b. Mostrar a los ganadores de la disciplina Y.
8. Sea una Lista Simple Circular de Contactos de Teléfono <nombre, nroTelefono>. Se pide de forma recursiva:
  - a. Verificar si existe el contacto X, si no existe añadirlo al final de la lista.
  - b. Mostrar el nombre al que pertenece el número telefónico Y.
  - c. Eliminar a todos los contactos de teléfono que su número empiecen por 6.
9. La ciudad de La Paz, controla las diferentes propiedades particulares a través de listas:
  - Lista de Propietarios <ci, nombre, paterno, materno>
  - Lista de Distritos <nro. distrito, ubicación, Lista de Zonas>
  - Lista de Zonas <nombre de zona, superficie y Lista de propiedades>
  - Lista de Propiedades <núm, tipo de propiedad (casa, edificio, terreno), ci del propietario>Se pide de forma recursiva:
  - a. Mostrar los datos de propietarios del distrito Nro N.
  - b. Determinar la cantidad de propiedades tipo X que existe en la zona Y.
10. Sea una Lista de Materias <nombre, sigla, PilaEstudiantes>, la Pila de Estudiantes <nombre, ci, ListaExámenes>, la Lista Exámenes <NroEvaluacion, Nota (sobre 100%)>. Nota: Los estudiantes contarán con 3 exámenes ponderados sobre 30, 35, 35.  
Se pide de forma recursiva:
  - a. Mostrar la nota final de todos los estudiantes.
  - b. Mostrar al(los) mejor(es) estudiantes de cada materia.
  - c. Verificar si el estudiante de nombre X está en la materia Y.

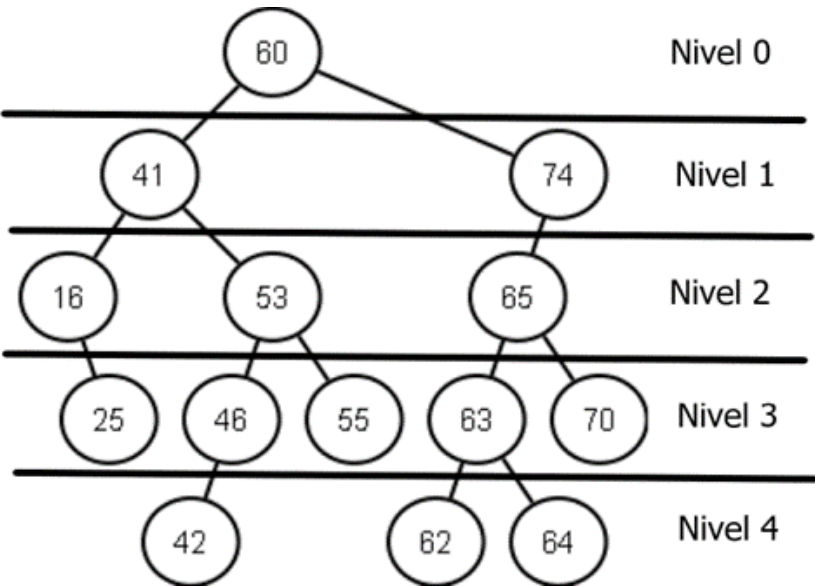
# ÁRBOLES

- 1. El recorrido en inorden de un determinado árbol binario es: SIWTCFRLNONZKy en preorden RCISTWFKLOMNZ. Resolver:
  - a. Dibujar el árbol binario.
  - b. Escribir el recorrido en postorden del árbol construido.
- 2. Dado un árbol binario de números enteros. Se pide:
  - a. Podar hasta el i-ésimo nivel, si existe.
  - b. Podar todas las hojas del árbol. Ejemplo:



- 3. Dado un árbol binario de números. Se pide:
  - a. Verificar si es un “árbol binario completo”.
  - b. Mostrar cuál es el mayor nivel en el cual el árbol recién es un “árbol binario completo”.

Ejemplo:



En el árbol binario de ejemplo, el nivel 1 es el nivel más alto en el que el árbol binario es completo.

- 4. Sean los Árboles binarios A y B de cadenas:
  - a. Verificar si todos los elementos de A están en B.
  - b. Verificar si A y B son iguales.
- 5. Dado un árbol binario de mascotas<nombre, tipo, edad>, se pide:
  - a. Mostrar las mascotas tipo X del subárbol derecho.
  - b. Verificar si existen mascotas con una edad mayor a 12 años.
- 6. Dado un árbol binario que almacena los datos de los estudiantes<Ci, Nombre, nota>. Se pide:
  - a. Mostrar a los estudiantes que tienen la mejor nota.
  - b. Mostrar el nivel donde se encuentran los estudiantes con la menor nota.

7. Dado un árbol binario de búsqueda cuyos nodos almacena productos <nombre, precio>. Se pide:
- Mostrar el producto con el precio X.
  - Mostrar el producto más caro y el más barato.
  - Mostrar los productos con precio mayores a X.
8. Sea un árbol binario de búsqueda de personas <nombre, ci>. Se pide:
- Verificar si existe la persona con ci X.
  - Mostrar el camino que sigue para llegar al ci Y.
  - Mostrar el ci de las personas que su nombre empiece por "A".
9. Sea un Árbol binario de Ciudades<nomCiudad, ArboldeBusquedaTemperatura (registradas en esa ciudad)>. Árbol Binario de Búsqueda de Temperatura<Temperatura>.
- Mostrar las ciudades donde se registró la temperatura X.
  - Verificar si la ciudad X tuvo temperaturas menores a 0 grados.
  - Mostrar las ciudades que tuvieron la temperatura más alta.
10. Sea un Árbol Binario de Bibliotecas<nombre, dirección, ArbolBinarioLibros>. Árbol Binario de Libros<titulo, autor, nroPaginas>. Se pide
- Mostrar cuántos libros hay del autor X.
  - Verificar si la biblioteca X tiene el libro X del autor Z
  - Mostrar los libros que tienen la mayor cantidad de número de páginas.