

### Ejercicio 6 Repetitiva Mientras (While)

Diseñar un algoritmo que muestre por pantalla la tabla de multiplicación del número que ingrese el usuario. Para definir hasta que número desea que muestre la tabla de multiplicación el usuario también deberá ingresar este valor. La tabla de multiplicación a mostrar debe empezar en la multiplicación por 1.

### Ejercicio 7 Repetitiva Hacer Mientras (Do While)

Diseñar un algoritmo que muestre por pantalla la tabla de multiplicación del número que ingrese el usuario. Para definir hasta que número desea que muestre la tabla de multiplicación el usuario también deberá ingresar este valor. La tabla de multiplicación a mostrar debe empezar en la multiplicación por 1.

### Ejercicio 8 Repetitiva Para (For)

Diseñar un algoritmo que realice el promedio de 4 números. Los números podrán ser decimales y serán ingresados por pantalla por el usuario.

## Ejercicios sobre Estructuras de Datos

### Ejercicio 9 Arreglo Booleano

Diseñar un algoritmo que recorra las butacas de una sala de cine y determine cuántas butacas desocupadas hay en la sala. Suponga que inicialmente tiene un array (arreglo) con valores booleanos que si es verdadero(verdadero) implica que está ocupada y si es falso(falso) la butaca está desocupada. Tenga en cuenta que el array deberá ser creado e inicializado al principio del algoritmo.

### Ejercicio 10 Dos Arreglos

Una escuela tiene un total de 3 aulas con la siguiente capacidad:

Identificador Aula	Cantidad de Bancos del Aula
Azul	40
Verde	35
Amarillo	30

Sabiendo la cantidad de bancos de cada aula, el usuario deberá ingresar la cantidad de alumnos inscriptos para cursar tercer grado y el sistema deberá determinar qué aula es la indicada para la cantidad ingresada. La escuela ya sabe que la máxima capacidad de sus aulas es de 40 alumnos, por lo tanto, la cantidad de alumnos inscriptos que ingresa el usuario siempre será un número menor o igual a 40.

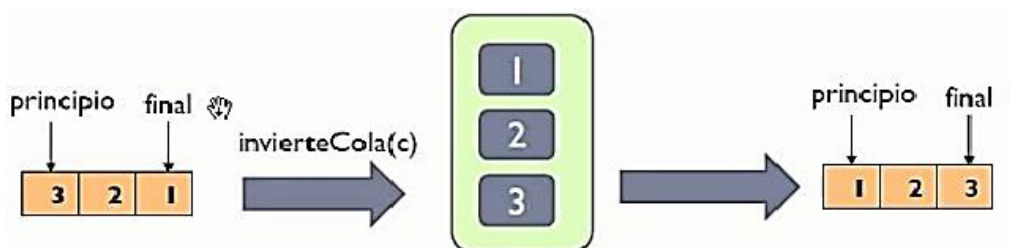
Listas necesarias para resolver el problema:

40	35	30
0	1	2
Azul	Verde	Amarillo
0	1	2

### Ejercicio 11 Pila

**Enunciado:**

Diseñar un algoritmo que a partir de una pila inicial de tres elementos devuelva una pila invertida. La pila contiene números enteros como se muestra en la figura. Al comienzo la pila está vacía, se deben apilar los siguientes elementos: 1,2,3 y luego invertir su orden.



## Ejercicios Algoritmos Fundamentales

### Ejercicio 12 Ordenamiento por Inserción

<https://www.youtube.com/watch?v=5kVQ8kf52K4>

Escribir el pseudocódigo y las pruebas de escritorio para realizar el ordenamiento de un vector con 5 números enteros. El usuario ingresa los números que él desea, cree un vector para guardar temporalmente dichos datos y luego realice el ordenamiento del mismo por inserción

### Ejercicio 13 Ordenamiento de la Burbuja

<https://www.youtube.com/watch?v=L3d48etbseY>

Escribir el pseudocódigo y las pruebas de escritorio para realizar el ordenamiento de un vector con 5 números enteros. El usuario ingresa los números que él desea, cree un vector para guardar temporalmente dichos datos y luego realice el ordenamiento del mismo a través del método de la burbuja.

### Ejercicio 14 Ordenamiento por Selección

<https://www.youtube.com/watch?v=IOYwcUJB3vo>

Escribir el pseudocódigo y las pruebas de escritorio para realizar el ordenamiento de un vector con 5 números enteros. El usuario ingresa los números que él desea, cree un vector para guardar

temporalmente dichos datos y luego realice el ordenamiento del mismo a través del método de la burbuja.

### Ejercicio 15 Búsqueda Secuencial

Escribir el pseudocódigo y las pruebas de escritorio para realizar la búsqueda del nombre de un cliente en un vector que contiene 5 clientes en total. El cliente a buscar será ingresado por pantalla por el usuario. El algoritmo deberá devolver, en caso de que ese nombre exista, la posición en donde se encuentra dicho cliente dentro del vector.