



OPERACIONES BÁSICAS CON VECTORES

1. BÚSQUEDA DE UN ELEMENTO EN UN VECTOR:

- ✦ **Búsqueda Secuencial:** Para buscar un valor determinado en un vector, se recorre el vector, por cada elemento, hasta que se encuentre el valor buscado, estando el vector ordenado o no. La desventaja que tiene esta búsqueda, es que si el valor buscado está en la última posición o no está, se recorre todo el vector.

```
Procedimiento Bus_Sec (Entero: vec[], N, valor)
  Entero: i ← 0

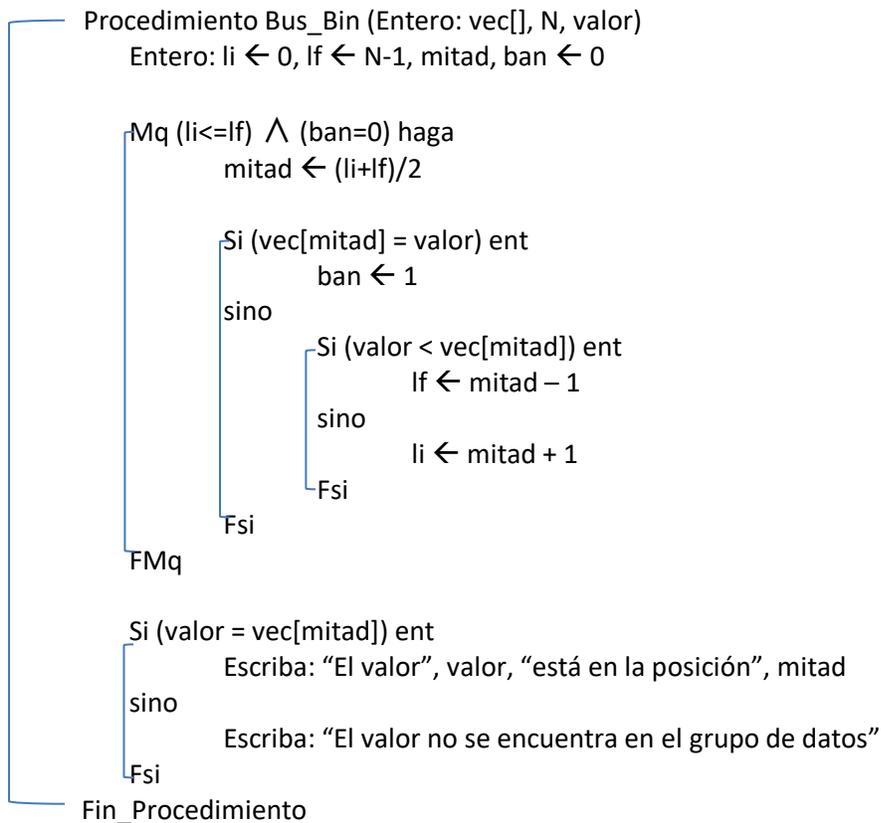
  Mq (i<N) ∧ (valor <>vec[i]) haga
    i ← i + 1
  FMq

  Si (i<N) ent
    Escriba: "El valor", valor, "está en la posición", i
  sino
    Escriba: "El valor no se encuentra en el grupo de datos"
  Fsi

Fin_Procedimiento
```

/*Este procedimiento funciona adecuadamente, en el caso de que todos los elementos del vector sean diferentes. Cuando existen elementos iguales, el procedimiento muestra el primer valor que encuentra, a partir del primer elemento. */

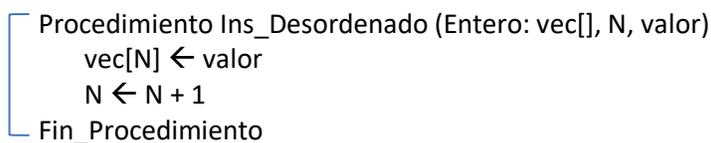
- ✦ **Búsqueda Binaria:** Para buscar un valor determinado en un vector que está ordenado, se divide el vector en dos partes y se compara el valor buscado con el elemento que está en la mitad del vector; sino coincide, se mira si el valor está en la primera o segunda mitad y se vuelve a realizar el proceso hasta que se encuentre el valor buscado.



/*El anterior procedimiento funciona adecuadamente, en el caso de que todos los elementos del vector estén ordenados y sean diferentes. Cuando existen elementos iguales, el procedimiento muestra el primer valor que encuentra, a partir de la búsqueda binaria. */

2. INSERTAR UN ELEMENTO EN UN VECTOR:

- ✦ **Insertar un elemento en un vector desordenado:** Cuando el vector está desordenado simplemente se inserta el nuevo elemento al final del vector (N) y se incrementa en uno el número de elementos, teniendo en cuenta antes de llamar al procedimiento, que no se sobrepase el tamaño del vector.





- ✦ **Insertar un elemento en un vector ordenado:** Para esto, se debe buscar el lugar dentro del vector donde debe insertarse el valor de forma ordenada y correr todos los demás elementos una posición hacia la derecha, para darle espacio al nuevo valor y ahí insertarlo. Igual que el anterior procedimiento, se incrementa en uno el número de elementos, y se debe tener en cuenta antes de llamar al procedimiento, que no se sobrepase el tamaño del vector.

```
Procedimiento Ins_Ordenado (Entero: vec[], N, valor)
  Entero: i ← 0, j
  Mq (i <= N-1) ∧ (valor > vec[i]) haga
    i ← i + 1
  FMq
  Para (j = N, i+1, -1) ent
    vec[j] ← vec[j-1]
  Fpara
  vec[i] ← valor
  N ← N + 1
Fin_Procedimiento
```

3. ELIMINAR UN ELEMENTO EN UN VECTOR:

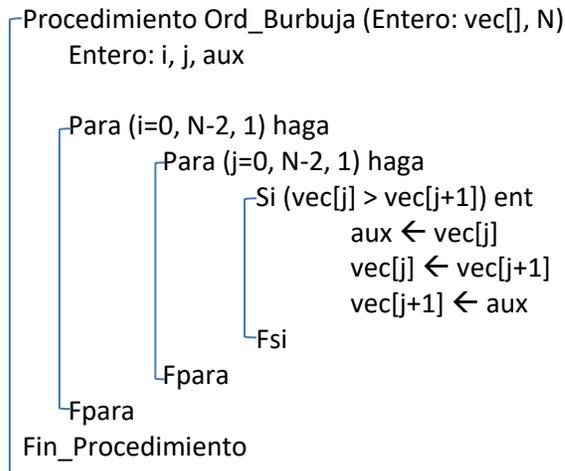
Para eliminar un elemento en un vector, se debe verificar antes de llamar al procedimiento, que el vector no esté vacío. Posteriormente, se busca la posición de elemento a borrar y luego se corren los elementos una posición a la izquierda, a partir de la posición donde se encuentra el elemento a borrar y finalmente se disminuye el número de elementos del vector.

```
Procedimiento Eliminar (Entero: vec[], N, valor)
  Entero: i ← 0, j
  Mq (i < N) ∧ (valor <> vec[i]) haga
    i ← i + 1
  FMq
  Si (i < N) ent
    Para (j = i, N-2, 1) haga
      vec[j] ← vec[j+1]
    Fpara
    N ← N-1
    vec[N] ← 0
  sino
    Escriba: "El valor", valor, "no está en el vector"
  Fsi
Fin_Procedimiento
```

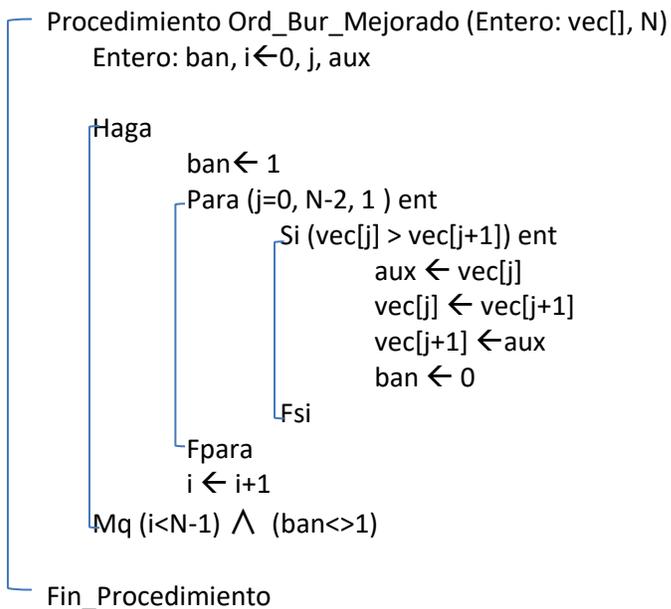


4. ORDENAMIENTO DE LOS ELEMENTOS DE UN VECTOR (Orden Ascendente):

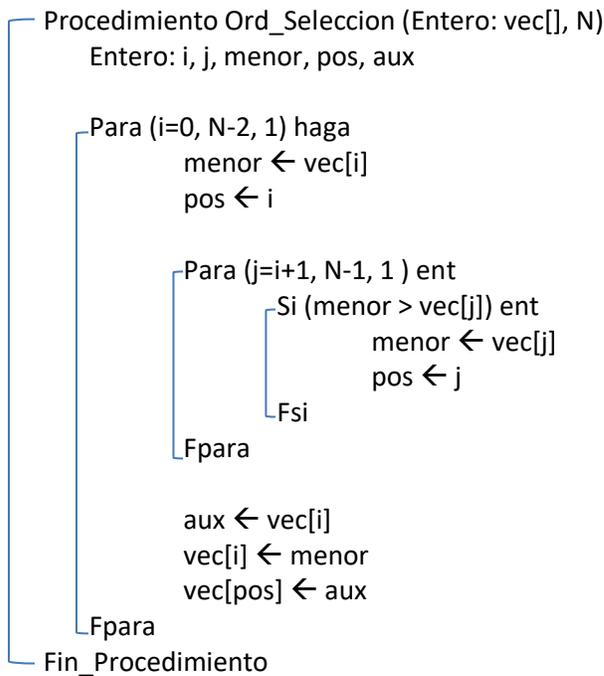
- ✦ **Ordenamiento por Burbuja:** El ordenamiento lo realiza con base en la comparación de los elementos seguidos del vector y si están desordenados los va intercambiando, y así lo realiza en varias iteraciones (N-2) hasta que el vector quede ordenado. La desventaja que tiene es que si el vector queda organizado antes, de igual forma sigue haciendo las iteraciones.



- ✦ **Ordenamiento por Burbuja Mejorado:** Es parecido al de la Burbuja, a diferencia que no realiza todas las iteraciones; si el vector ya está ordenado, no realiza más iteraciones.



- ✦ **Ordenamiento por Selección:** En N-2 iteraciones, este método en cada iteración busca el menor elemento y lo cambia de posición por el primero que se asume como menor, en la segunda iteración por el segundo y así sucesivamente.



- ✦ **Ordenamiento por Inserción o Baraja:** Consiste en ordenar el vector a partir del segundo elemento, haciendo el intercambio respectivo si están desordenados, así como lo realizan los jugadores de cartas, corriendo todos los elementos hacia la derecha una posición, a partir del elemento dos.

