

EVALUACION	Examen	GRUPO		FECHA	
MATERIA	Algoritmos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	<ul style="list-style-type: none">- Puntos: Máximo: Mínimo:- Duración: 2 horas- Sin material				
Nombre	Nro estudiante		Nota		

Ejercicio 1

Implementar un algoritmo que reciba un vector ordenado y haga una búsqueda binaria
El algoritmo debe ser resuelto en forma **recursiva**.

Utilizar la siguiente firma:

```
public int buscar(int[] números, int número, int inicio, int fin) {
```

solución

```
public int buscar(int[] números, int número, int inicio, int fin) {  
  
    int centro = (inicio + fin) / 2;  
  
    if (fin < inicio) {  
        return -1;  
    }  
  
    if (número < números[centro]) {  
        return buscar(números, número, inicio, centro - 1);  
    }  
  
    if (número > números[centro]) {  
        return buscar(números, número, centro + 1, fin);  
    }  
  
    if (número == números[centro]) {  
        return centro;  
    }  
  
    return -1;  
}
```

Ejercicio 2

Dado el siguiente vector

```
int v[] = {64, 34, 25, 12, 22, 11, 90};
```

a) Indique a que método de ordenamiento corresponde la siguiente secuencia

- 34- 25- 12- 22- 11- 64- 90
- 25- 12- 22- 11- 34- 64- 90
- 12- 22- 11- 25- 34- 64- 90
- 12- 11- 22- 25- 34- 64- 90
- 11- 12- 22- 25- 34- 64- 90
- 11- 12- 22- 25- 34- 64- 90

Solución: Burbuja

b) Implemente el método de ordenamiento en forma recursiva.

// Solución

```
static void burbuja (int v[], int n)
```

```
{
```

```
// Base case
```

```
if (n == 1)
```

```
return;
```

```
for (int i=0; i<n-1; i++)
```

```
if (v[i] > v[i+1])
```

```
{
```

```
int temp = v[i];
```

```
v[i] = v[i+1];
```

```
v[i+1] = temp;
```

```
}
```

```
Burbuja(v, n-1);
```

```
}
```

```
}
```

```
}
```

Ejercicio 3

Implemente un método de búsqueda por bipartición en forma recursiva que retorne -1 si no encuentra el valor x recibido como parámetro y si lo encuentra retorne la posición dentro del vector.

Firma a utilizar `int busqueda(int v[], int X, int i, int j)`

Ejemplo

`Int v[] = { 11, 12,22, 25, 34, 64, 90};`

<code>System.out.println(busqueda(v, 64, 0, v.length-1));</code>	retorna 5
<code>System.out.println(busqueda(v, 34, 0, v.length-1));</code>	retorna 4
<code>System.out.println(busqueda(v, 200, 0, v.length-1));</code>	retorna -1

// Solución

```
static int busqueda(int v[], int X, int i, int j) {
    int medio;
    if (i > j) {
        return -1;
    }
    medio = (i + j) / 2;
    if (v[medio] < X) {
        return busqueda(v, X, medio + 1, j);
    } else {
        if (v[medio] > X) {
            return busqueda(v, X, i, medio - 1);
        } else {
            return medio;
        }
    }
}
```