



Algoritmos y Estructura de Datos

Unidad 2: Arreglo lineal

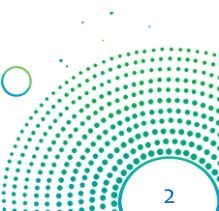
Tema 7: Artificios y operaciones especiales

Semana 07





Tema 7: Artificios y operaciones especiales





Índice

2.3 Tema 7: Artificios y operaciones especiales

- 2.3.1 Método privado buscar
- 2.3.2 Método privado intercambiar
- 2.3.3 Método privado eliminar
- 2.3.4 Operaciones públicas complementarias
- 2.3.5 Ejemplo





Capacidades

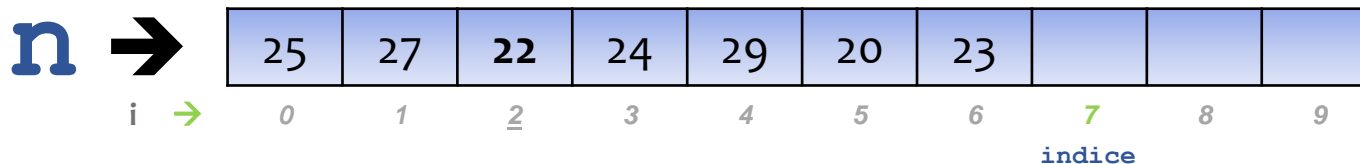
- Identifica los nuevos recursos en programación.
- Diseña programas utilizando arreglos lineales.





2.3.1 Método privado buscar

Artificio 1: Recibe un número y retorna la posición. Si no lo encuentra retorna -1



```
private int buscar(int numero) {  
    for (int i=0; i<indice; i++)  
        if (n[i] == numero)  
            return i;  
    return -1;  
}
```

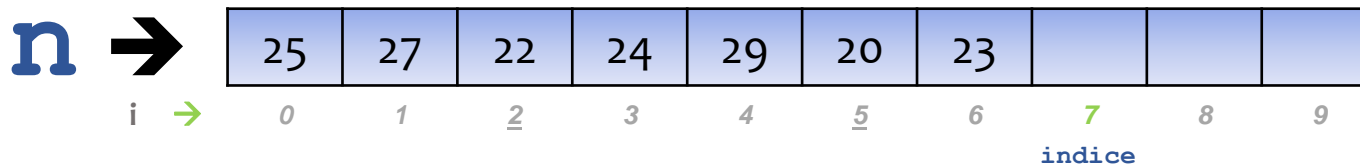
Ejemplo:

```
int pos1 = buscar(22); // pos1 ← 2  
int pos2 = buscar(28); // pos2 ← -1
```

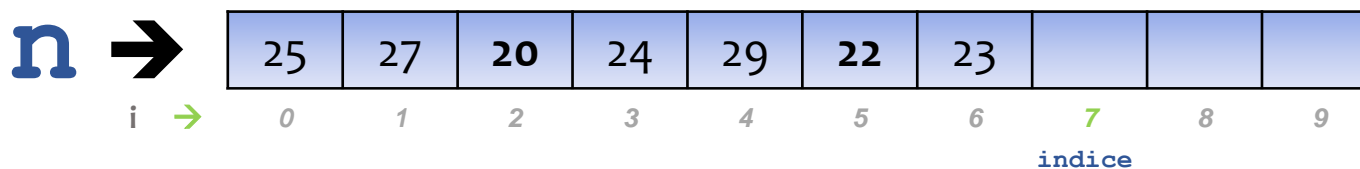


2.3.2 Método privado intercambiar

Artificio 2: Recibe dos posiciones e intercambia sus contenidos



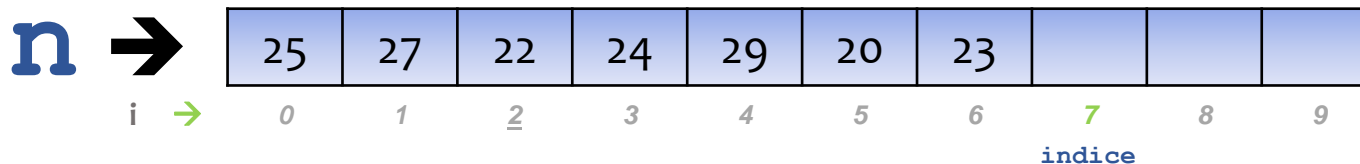
```
private void intercambiar(int pos1, int pos2) {  
    int aux = n[pos1];  
    n[pos1] = n[pos2];  
    n[pos2] = aux;  
}
```



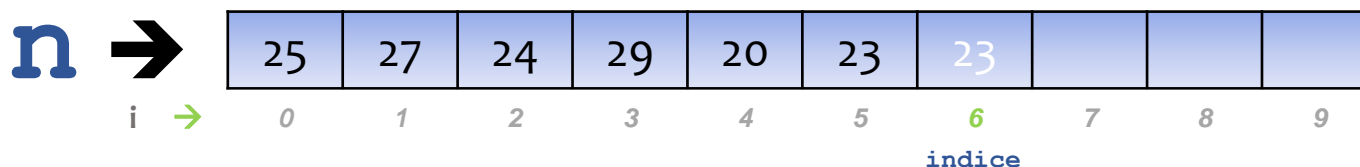


2.3.3 Método privado eliminar

Artificio 3: Recibe una posición y elimina del arreglo el valor ubicado en dicha posición



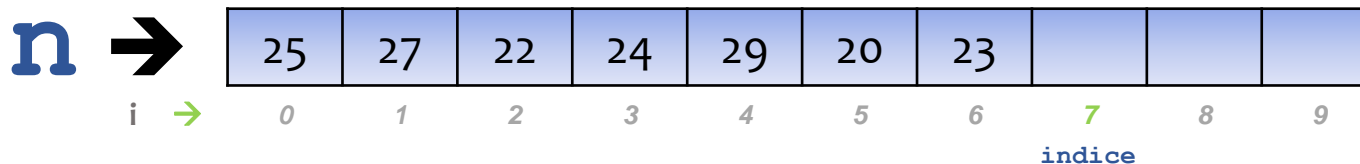
```
private void eliminar(int pos) {  
    for (int i=pos; i<indice-1; i++)  
        n[i] = n[i+1];  
    indice --;  
}
```





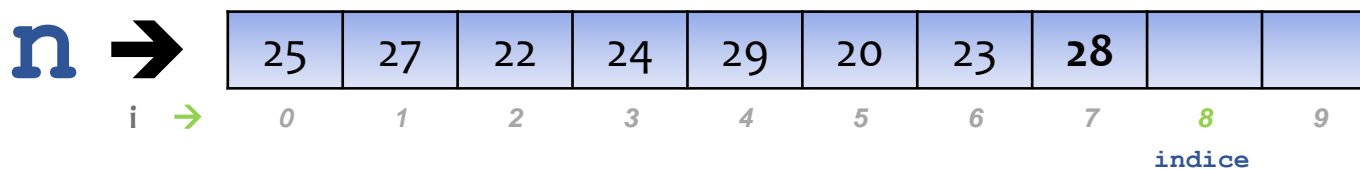
2.3.4 Operaciones públicas complementarias

Artificio 4: Recibe un número, lo busca, y si no está registrado lo adiciona



```
public void buscarAdicionar(int numero) {  
    int pos = buscar(numero);  
    if (pos == -1)  
        adicionar(numero);  
}
```

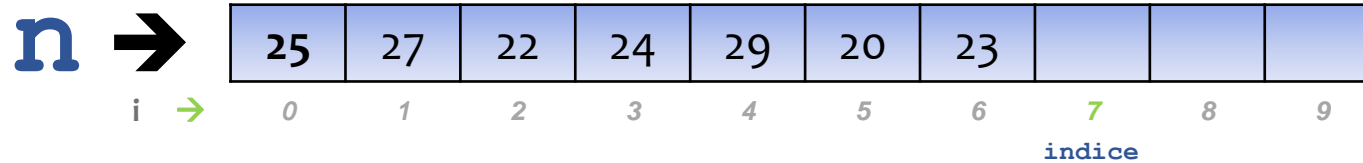
28



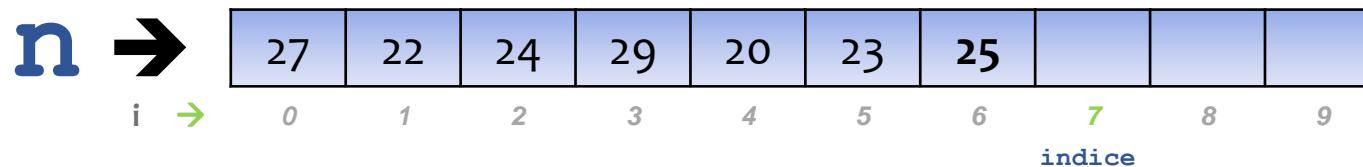


2.3.4 Operaciones públicas complementarias

Artificio 5: Traslada el primer número al final



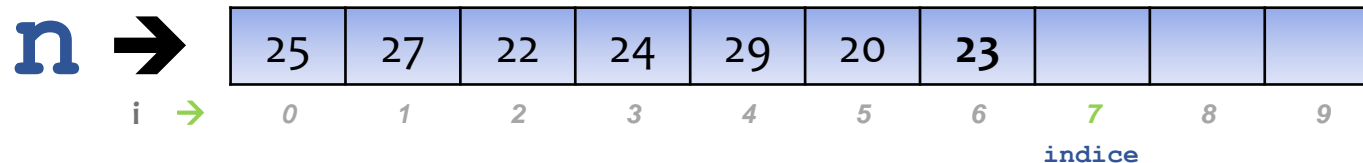
```
public void primeroAlFinal() {  
    for (int i=0; i<índice-1; i++)  
        intercambiar(i, i+1);  
}
```



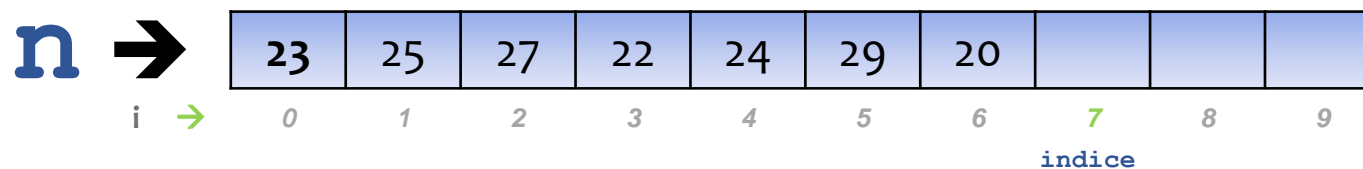


2.3.4 Operaciones públicas complementarias

Artificio 6: Traslada el último número al inicio



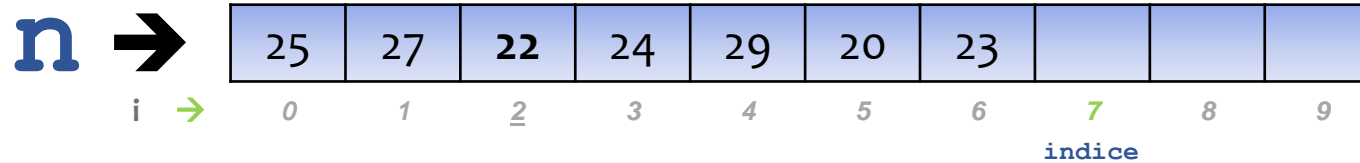
```
public void ultimoAlInicio() {  
    for (int i=índice-1; i>0; i--)  
        intercambiar(i, i-1);  
}
```





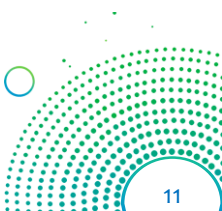
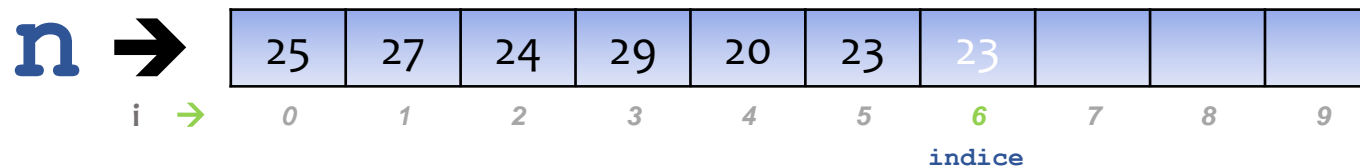
2.3.4 Operaciones públicas complementarias

Artificio 7: Recibe un número, lo busca, y si está registrado lo elimina



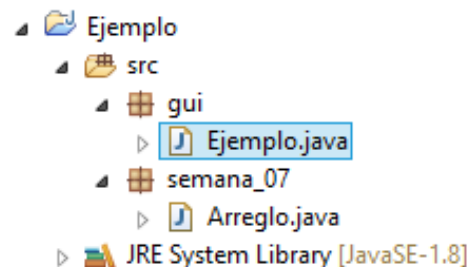
```
public void buscarEliminar(int numero) {  
    int pos = buscar(numero);  
    if (pos != -1)  
        eliminar(pos);  
}
```

22
↓





2.3.5 Ejemplo



- a) Implementa la clase **Arreglo** en el paquete **semana_07**
 - atributos privados **int n** (arreglo lineal) y contador **indice**.
 - constructor que reserva en **n** diez espacios e inicializa el **indice** en cero.
- b) Implementa operaciones públicas básicas
 - método **tamano** que retorna la cantidad de números ingresados hasta ese momento.
 - método **obtener** que recibe la posición y retorna el número registrado en dicha posición.
 - método privado **ampliarArreglo** que extiende el arreglo en diez espacios más.
 - método privado **adicionar** que recibe un número y lo registra en la posición que corresponde.
 - método privado **buscar** que recibe un número y retorna la posición. Si no lo encuentra retorna -1.
 - método privado **intercambiar** que recibe dos posiciones e intercambia sus contenidos.
 - método privado **eliminar** que recibe una posición y elimina del arreglo el valor ubicado en dicha posición.
- c) Implementa operaciones públicas complementarias
 - método **buscarAdicionar** que recibe un número, lo busca y si no está registrado lo adiciona.
 - método **primeroAlFinal** que traslada el primer número al final.
 - método **ultimoAlInicio** que traslada el último número al inicio.
 - método **buscarEliminar** que recibe un número, lo busca y si está registrado lo elimina.
- d) En la clase **Ejemplo** declara y crea como variable global un objeto de tipo **Arreglo**, implementa un método **listar**, que visualiza los números; y a la pulsación de los botones respectivos invoca a las operaciones públicas complementarias.



Conclusiones

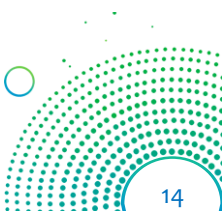
- La habilidad y creatividad nos permitirá generar una serie de trucos y artificios.





Referencias bibliográficas

- **Joyanes Aguilar Luis.** *Fundamentos de programación: algoritmos, estructuras de datos y objetos.* Madrid, España: McGraw-Hill (005.1 JOYA/A 2021)
- **Lewis John.** *Estructuras de datos con Java: diseño de estructuras y algoritmos.* Madrid, Pearson Educación (005.73 LEWI/E 2021)
- **Deitel Harvey.** *Cómo programar en Java.* México, D.F.: Pearson Educación (005.133J DEIT 2021)



GRACIAS



SEDE MIRAFLORES

Calle Díez Canseco Cdra 2 / Pasaje Tello
Miraflores – Lima
Teléfono: 633-5555

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE LIMA CENTRO

Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535