



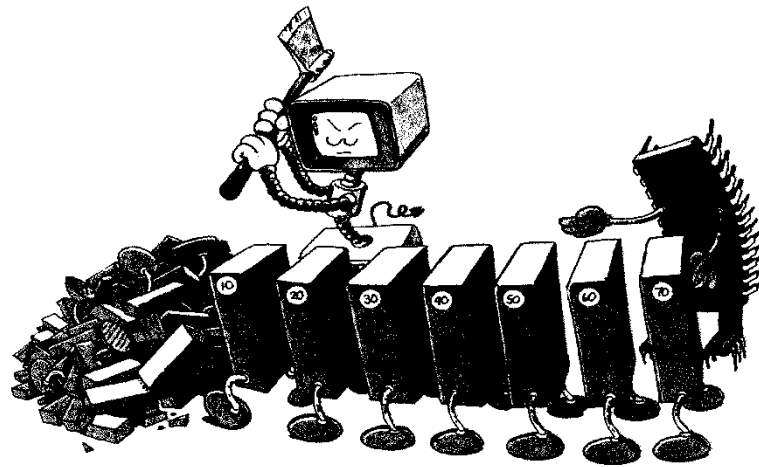
Esquemas algorítmicos de tratamiento secuencial





Secuencias.

El tipo de problemas más frecuente con el que podemos enfrentarnos implica la manipulación de una secuencia de valores de algún tipo



S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
-------	-------	-------	-------	-------	-------	-------	-------



Tipos de secuencia

- Secuencia numerada.
 - El número de elementos es conocido a priori.
 - Para recorrerlos sólo hace falta contar.

Secuencia de 10 elementos:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

- Secuencia marcada
 - El número de elementos es desconocido a priori.
 - El final de la secuencia se determina por una marca o valor especial que sigue al último elemento o está asociado a él.

Secuencia hasta encontrar el 0:

1, 5, 60, 23, ..., 36, 1, 54, 99, **0**



Esquema básico de recorrido

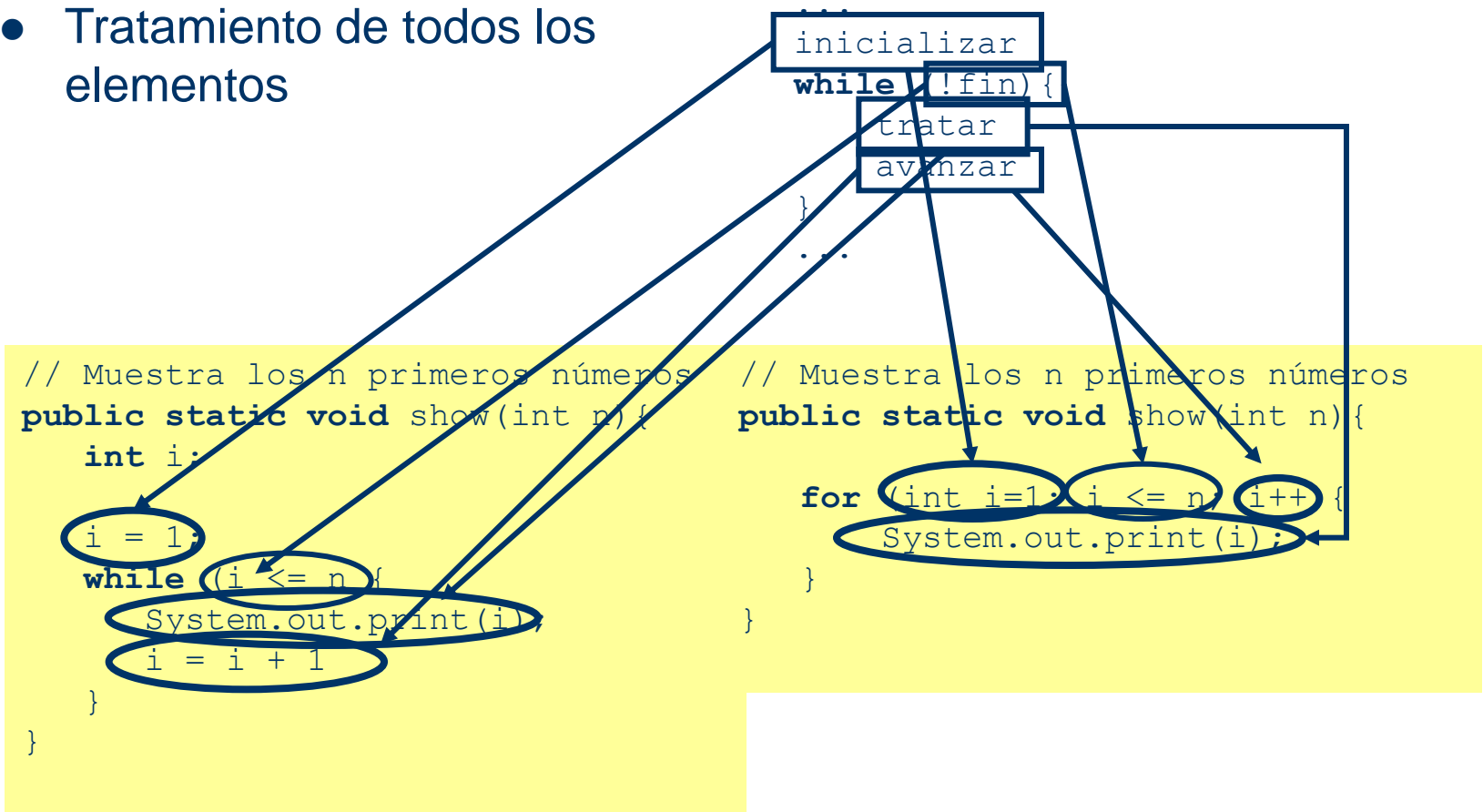
```
...  
inicializar  
while (!fin) {  
    avanzar  
}  
...
```

```
...  
int n = 0  
while (n < m) {  
    n = n + 1  
}  
...
```



Esquema de recorrido con tratamiento

- Tratamiento de todos los elementos





Tratamiento de los elementos que cumplen una propiedad

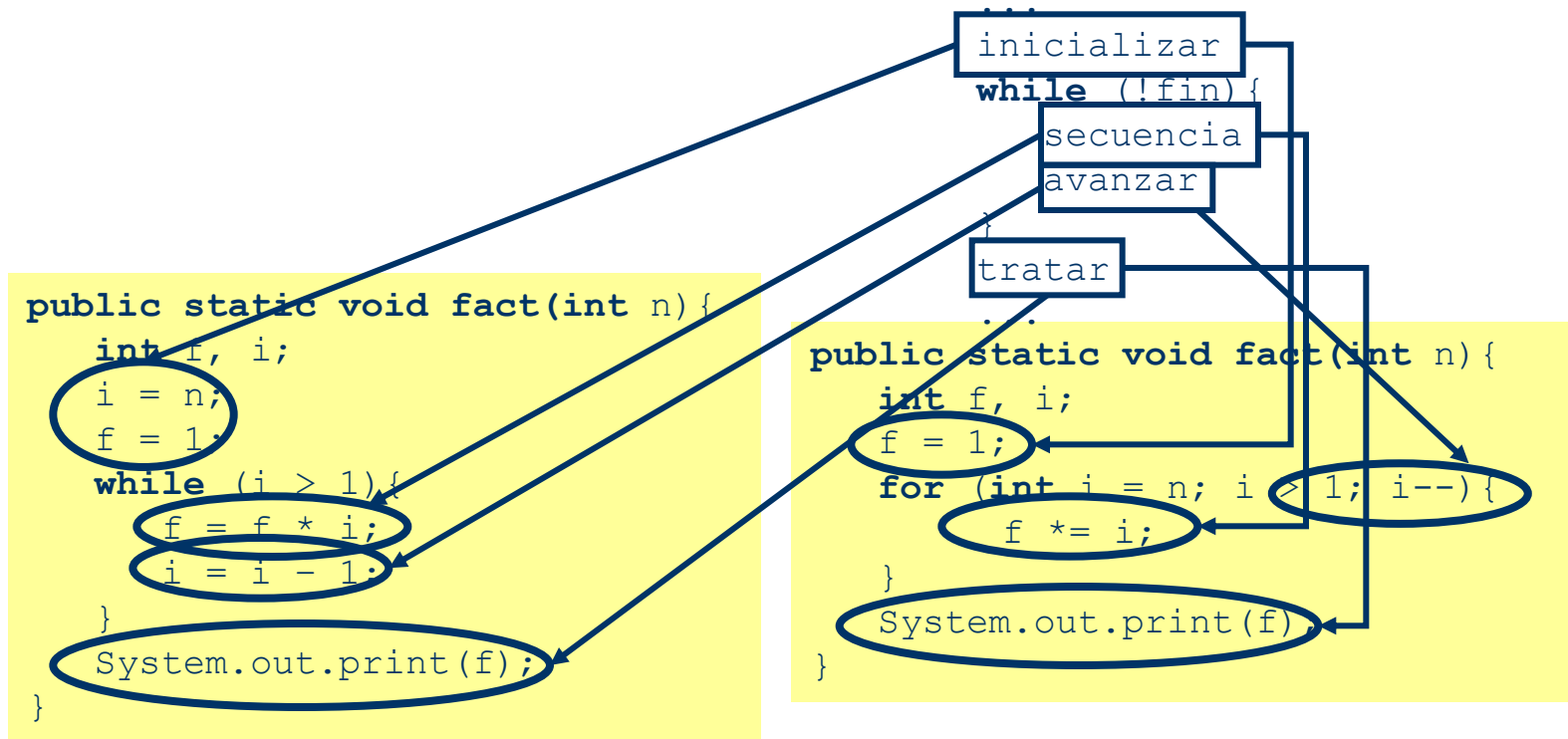
- Tratamiento de todos los elementos que cumplen una propiedad

```
...
inicializar
while (!fin){
    if (p(x)){
        tratar(x)
    }
    avanzar
}
...
```

```
...
n = 1
while (n < m){
    if (isPrime(n)){
        System.out.print(n);
    }
    n = n + 1
}
...
```



Tratamiento del último elemento de una secuencia





Esquema de búsqueda

- Tratamiento del primer elemento que cumple una condición

```
...
inicializar
while (!fin && !encontrado)
    avanzar
}
if (encontrado) {
    tratar
} else {
    acciones cuando no encontrado
}
...
```

```
public static boolean isPrime(p)
    int d;
    d = 2;
    while ((d < p) && (p % d != 0)) {
        d = d + 1;
    }
    if (d < p) {
        return false;
    } else {
        return true;
    }
}
```




Combinación de esquemas

- Esquema de recorrido y de búsqueda
- Combinación secuencial

```
...  
while (!fin && !encontrado){  
    avanzar en N  
}  
for (int i=1; i <= n; i++){  
    tratar  
}  
...
```

- Combinación anidada

```
...  
for (int i=0; i < final; i++){  
    for (int j=0; j < final; j++){  
        if (condición){  
            tratar  
        }  
    }  
}  
...
```



Conclusión

