

# Programación Orientada a Objetos

## Tarea 2.3. La clase matriz

José Fidel Argudo Argudo    Francisco Palomo Lozano  
Inmaculada Medina Bulo    Gerardo Aburrizaga García  
Pedro Delgado Pérez



Versión 2.0



## Tarea 2.3. Cuestiones

### Ejercicio 1

Clasifique las funciones y operadores miembro de la clase `matriz` en diferentes categorías: **constructores** (identificando sus diferentes tipos), **destructores**, **observadores** y **modificadores**.

¿Cuántas hay en cada categoría?

### Ejercicio 2

Si existen los siguientes constructores, escriba una instrucción en cada caso en la que se invoque al mismo:

- Constructor predeterminado
- Constructor de lista inicializadora
- Constructor de copia
- Constructor de movimiento

A continuación, estudie el programa de prueba de las transparencias de la clase `matriz` y diga si estos constructores se utilizan en el mismo y dónde.

### Ejercicio 3

- Describa los errores que hay en el siguiente código:

```
1 matriz A = 3;  
2 matriz B = matriz(5);  
3 matriz C(3);  
4 B = 2;  
5 A = matriz(5);
```

- ¿Por qué se declara `explicit` el primer constructor de la clase `matriz`? ¿Podría causar algún problema si no se hiciera así?

En tal caso, ¿se podría evitar ese problema definiendo un operador de conversión de `int` a `matriz`? Razone la respuesta.

## Tarea 2.3. Cuestiones

### Ejercicio 4

¿Qué tipos devuelven los operadores de signo + y -? ¿Por qué devuelven tipos distintos?

### Ejercicio 5

¿El operador - de cambio de signo es miembro de la clase `matriz`?

¿Se podría definir de la otra forma? En caso afirmativo, escriba la declaración. ¿Qué ventajas e inconvenientes tendría?

## Tarea 2.3. Cuestiones

### Ejercicio 6

¿Es correcto definir el operador \*= de matriz como sigue?

```
1 inline matriz& matriz::operator *=(const matriz& a)
2 {
3     n = a.columnas();
4     x *= a.x;
5     return *this;
6 }
```