

# MATRICES Y TABLEROS

# ¿QUÉ ES UNA MATRIZ?

Una matriz, en nuestro contexto, es simplemente un vector de dos dimensiones que tiene el mismo largo en cada uno de sus elementos.

# ¿Y QUE HACEMOS CON LAS MATRICES?

Muchas veces queremos utilizar matrices para representar estructuras como, por ejemplo:

- Matrices de verdad (esas de Álgebra que tienen determinante y esas cosas)
- Tableros (de ajedrez, por ejemplo)
- Mapas (por ejemplo, en cada casillero guardamos la altura del territorio en esas coordenadas, o la cantidad de personas que viven en una determinada manzana)
- Imágenes
- Muchísimos etcéteras

## OPERACIONES (QUE VAMOS A NECESITAR) SOBRE MATRICES

- Declarar una matriz:

```
vector<vector<int>> m;
```

También pueden poner cualquier otro tipo (string, char, etc).

- Inicializar una matriz de m filas x n columnas todas con el mismo valor (x):

```
vector<vector<int>> res(m, vector<int>(n, x));
```

- Agregar una fila:

```
vector<vector<int>> m;  
vector<int> v = {1, 2, 3};  
m.push_back(v);
```

- Inicializar una matriz con valores fijos:

```
vector<vector<int>> mat = {{6, 12, 18},  
                          {7, 14, 21},  
                          {8, 16, 24},  
                          {9, 18, 27}};
```

- Acceder a un elemento en la posición (i, j):

```
m[i][j]
```

## EJEMPLO: ROTACIÓN DE MATRICES

- Implementar una función que cumpla con la siguientes especificación:

```

proc rotar (in mat: seq⟨seq⟨ℤ⟩⟩, in d: ℤ, in a: ℤ, out res: seq⟨seq⟨ℤ⟩⟩)
{
  Pre { |mat| > 0 ∧ (∀i : ℤ) (0 ≤ i < |mat| →L |mat[i]| = |mat[0]|) ∧ a ≥
        0 ∧ d ≥ 0 }
  Post { mismasDimensiones(res, mat) ∧L
        esLaMovidaParaAbajoYDerecha(res, mat) }
}

pred mismasDimensiones (m1: seq⟨seq⟨ℤ⟩⟩, m2: seq⟨seq⟨ℤ⟩⟩) {
  |m1| = |m2| ∧L (∀i : ℤ) (0 ≤ i < |m1| →L |m1[i]| = |m2[i]|)
}

pred esLaMovidaParaAbajoYDerecha ( res: seq⟨seq⟨ℤ⟩⟩, mat:
seq⟨seq⟨ℤ⟩⟩) {
  (∀i, j : ℤ) (0 ≤ i < |mat| ∧L 0 ≤ j < |mat[i]| →L res[(i + a)
mód |mat|][(j + d) mód |mat[i]|] = mat[i][j])
}

```

## EJEMPLO: ROTACIÓN DE MATRICES

```
vector<vector<int>> rotar(vector<vector<int> > mat, int a,
                        int d) {
    int n = mat.size();
    int m = mat[0].size();
    vector<vector<int>> res(n, vector<int>(m));
    int i = 0;
    while(i < n) {
        int j = 0;
        while(j < m) {
            res[(i + a) % n][(j + d) % m] = mat[i][j];
            j++;
        }
        i++;
    }
    return res;
}
```

# MATRICES Y MÁS MATRICES

Durante la carrera verán más ejercicios de matrices hasta el cansancio en:

- Organización del Computador 2: Verán como aplicar filtro a imágenes (como los de Instagram) pero en lenguaje ASM.
- Algoritmos y Estructuras de Datos 3: Ejercicios sobre grafos, programación dinámica, etc.
- Métodos Numéricos: mejor conocida como “Matrices: la materia” (verán algoritmos sobre matrices como las de Álgebra).