

## Práctica #2 - Complejidad en espacio

1. Considerando las siguientes definiciones de tipo.

```
typedef bool arr[10];  
typedef arr mat[8];  
mat aux;
```

- Indique el costo en memoria de la variable aux.
- Indique la cardinalidad del tipo mat.
- Calcule la fórmula de acceso de aux[i]
- Calcule la fórmula de acceso de aux[i][j]

2. Considerando la siguiente estructura de datos.

```
struct e1 {  
    float x, y;  
    int z[15];  
};  
e1 aux[100];
```

- Indique el costo en memoria para la estructura del objeto aux.
- Indique la cardinalidad de aux.
- Calcule la fórmula de acceso de aux[i].x
- Calcule la fórmula de acceso de aux[i].z[j]

3. Considerando el siguiente bloque de programa.

```
enum sex = {  
    FEMENINO,  
    MASCULINO  
};  
struct cancion {  
    string nombre, autor;  
    int año;  
};  
struct cantante {  
    string nombre;  
    char pais[9];  
    cancion canciones[3];  
    int puntos;  
    sex sexo;  
};  
cantante concurso[25];
```

- Indique el costo en memoria para la estructura del objeto concurso.
- Indique la cardinalidad de concurso.
- Calcule la fórmula de acceso de concurso[i].canciones[j].nombre

4. Considerando la siguiente estructura.

```
struct vivienda {  
    float costo, altura;  
    string direccion;  
    int pisos, metros;  
    bool vigilancia;  
};  
vivienda urbanizacion[15];
```

- Indique el costo en memoria para la estructura del objeto urbanizacion
- Indique la cardinalidad de urbanizacion
- Calcule la fórmula de acceso de urbanizacion[i].direccion

5. Considerando el siguiente bloque de programa.

```
struct fecha {
    int dia, año;
    string mes;
};
struct entradaDeDirectorio {
    string nombre;
    fecha creacion;
    int tamañoDeArchivo, numeroDeArchivos;
    bool compartido;
};
entradaDeDirectorio directorio[365];
```

- Indique el costo en memoria para la estructura del objeto directorio.
- Indique la cardinalidad de directorio.
- Calcule la fórmula de acceso de directorio[i].creacion.mes

6. Considerando las siguientes estructura de datos.

```
enum tipoMateria = {
    OBLIGATORIA,
    ELECTIVA,
    COMPLEMENTARIA,
    LABORATORIO
};
struct fecha {
    int dia, año;
    string mes;
};
struct evaluacion {
    string nombre;
    fecha inicio, fin;
    float notaPromedio, notaMaxima;
    int numeroDeAprobados, numeroDeReprobados;
};
struct materia {
    int codigo, numeroCreditos, semestre, notaMaxima;
    tipoMateria tipo;
    string nombre, descripcion;
    float notaPromedio;
    switch (nombre) {
        case "AYP":
            int horasLaboratorio;
            evaluacion evaluaciones[10];
            break;
        case "AYED":
            evaluacion evaluaciones[7];
            break;
        case "ICG":
            bool disponible;
            evaluacion evaluaciones[5];
            break;
    }
};
materia pensum[32];
```

- Indique el costo en memoria para la estructura del objeto pensum.
- Indique la cardinalidad de pensum.

7. Considerando el siguiente bloque de programa.

```
struct persona {
    char nombre[50], direccionTrabajo[100], rif[15];
    bool estadoCivil, sexo;
    int edad, ci;
};
typedef persona juntaCondominio[10];
struct edificio {
    char nombre[50], direccion[101];
    int cantidadApartamentos;
    juntaCondominio junta;

};
edificio avenida[25];
```

- Indique el costo en memoria para la estructura del objeto avenida.
- Indique la cardinalidad de avenida.
- Calcule la fórmula de acceso de avenida[i].junta[j].estadoCivil

8. Considerando las siguientes estructura de datos.

```
struct punto3D {
    float x, y, z;
};
struct modelo3D {
    string nombre, formato;
    punto3D vertices[250], normales[250];
    bool visible;
};
struct escena3D {
    string nombre;
    modelo3D modelos[20];
};
escena3D mapa3D[25];
```

- Indique el costo en memoria para la estructura del objeto mapa3D.
- Indique la cardinalidad de mapa3D.
- Calcule la fórmula de acceso de mapa3D[i].modelos[j].vertices[k].x

9. Considerando las siguientes definiciones de tipo.

```
struct producto {
    int codigo;
    string nombre, descripcion;
    float costo;
    bool tipo;
    switch (tipo) {
        case true:
            float iva;
            break;
        case false:
            string descripcion;
            break;
    }
};
typedef producto almacen[100];
almacen supermercado[10];
```

- Indique el costo en memoria para la estructura del objeto supermercado.
- Indique la cardinalidad de supermercado.

10. Considerando las siguientes estructura de datos.

```
struct futbolista {
    string nombre, nacionalidad;
    int edad;
    float salario;
    switch (nombre) {
        case "Messi":
            bool balonDeOro;
            float agilidad;
            break;
        case "Ronaldo":
            bool campeonLiga;
            float salto, potencia;
            break;
        case "Mbappe":
            bool balonDeOroSub21;
            float velocidad;
            break;
    }
};

struct estadio {
    string nombre;
    int antigüedad;
    float valor;
    bool catedral;
};

struct equipo {
    string nombre;
    int antigüedad, copas;
    futbolista jugadores[23];
    estadio campo;
    switch (nombre) {
        case "Barcelona":
            int bajas, altas;
            break;
        case "RealMadrid":
            futbolista fichajes[8];
            break;
        case "Valencia":
            bool campeonCopa;
            break;
    }
};

equipo laLiga[30];
```

- Indique el costo en memoria para la estructura del objeto laLiga.
- Indique la cardinalidad de laLiga.