

Structs AEDD25

☰ AEDD

El struct es una estructura que agrupa tipos de datos relacionados, suele utilizarse cuando un dato involucra más tipos y resulta conveniente agruparlos.

Basicamente es crear un tipo de dato propio, como si la clase **Reales** incluyese todos los tipos de números posibles.

```
struct Natural{
    int numeroRandom;
};
struct Entero {
    int numeroRandom;
    Natural numeroNatural;
};
struct Racional{
    int numeroRandom;
    Entero numeroEntero;
};
struct Irracional {
    int numeroRandom;
};

struct Reales {
    Irracional numeroIrracional;
    Racional numeroRacional;

};
```

Dentro de las estructuras existen infinidad de usos, o estilos de usos. Los datos dentro de estos **structs** son accesibles mediante la notación de **.**

```
struct ejemplo {
    int arreglo[100];
    int tlArreglo;
```

```

    double promedio;
};

int main(){
    ejemplo datoA;
    datoA.arreglo[0]=10;
    datoA.tlArreglo=0;
    datoA.tlArreglo++;
    datoA.promedio=datoA.arreglo[0]/(2*datoA.tlArreglo);
};

```

Además es posible crear tantos `structs` como necesitemos, incluso anidarlos (incluir uno dentro de otro).

```

struct tipoDeDato {
    int a;
    double b;
    char c;
    string d;
    float e[100];
};

struct datoDeOtroStruct {
    tipoDeDato nombreDato1;
    tipoDeDato nombreDato2;
};

struct Nombre {
    tipoDeDato nombreDato1;
    tipoDeDato nombreDato2;
    tipoDeDato nombreDato3;
    datoDeOtroStruct nombreStruct1;
};

int main(){
    Nombre datoA;
    datoDeOtroStruct datoB;
    tipoDeDato datoC;
}

```

Cuando son usadas como parámetros de funciones se tratan de igual forma que los `int`, `double`, `floats`, `arreglos`etc. Para que se afecte el dato, sin que sea un arreglo, deben ser pasados por referencia.