PROGRAMACION II

Tipo de Datos Abstracto (TDA)

Código Asignatura 6A4

Año 2021

Abstracción

La *abstracción consiste en captar las características esenciales de una entidad u objeto, así como su comportamiento*. Por ejemplo, pensemos en automóviles, ¿Qué características podemos abstraer de los automóviles? O lo que es lo mismo: ¿Qué características semejantes tienen todos los automóviles? Todos tendrán una marca, un modelo, número de chasis, peso, llantas, puertas, ventanas, etc. Y en cuanto a su comportamiento todos los automóviles podrán acelerar, frenar, retroceder, etc.

Encapsulamiento

El *encapsulamiento se utiliza para ocultar o esconder las características esenciales de una entidad u objeto*, de manera que no pueda ser alterado por otros, en cierto modo proveen un efecto de caja negra donde la interacción debe darse por medio de su interface (operaciones, funciones) y no directamente.

EJEMPLO TDA COMPLEJO

INTERFACE del TDA COMPLEJO (complejo.h)

```
typedef struct{
    float r, i;} complex;

void suma (complex c1, complex c2, complex *c3);
void ingresa (complex *c);
void muestra (complex c);
float creal (complex c);
float cimag (complex c);
```

DESARROLLO del TDA COMPLEJO (complejo.c)

```
#include "complejo.h"
#include <stdio.h>

void suma (complex c1, complex c2, complex *c3){
  c3->r=c1.r+c2.r;
  c3->i=c1.i+c2.i;
}

void ingresa (complex *c){
printf("Ingrese las componente de un complejo");
scanf("%f %f",&(c->r), &(c->i));
}

void muestra (complex c){
    printf("%5.2f",c.r );
    (c.i > 0) ? printf("+%5.2f i",c.i ): printf("%5.2f i",c.i );
}

float creal (complex c){
 return c.r;
}

float cimag (complex c){
 return c.i;
}
```

UTILIZACION del TDA COMPLEJO (main.c)

```
#include <stdio.h>
#include <stdlib.h>
#include "complejo.h"

void main()
{
  complex c1, c2, c3;
  ingresa(&c1);
  ingresa(&c2);
  suma(c1,c2,&c3);
  muestra(c1);
  muestra(c2);
  printf("%5.2f",creal(c3));
 (cimag(c3)>0)?printf("+%5.2fi",cimag(c3)):printf("%5.2f i",cimag(c3));

  /* Que ocurriría con las siguientes sentencias? */
  printf("%5.2f",c3.r);
 (c3.i>0)?printf("+%5.2fi",c3.i):printf("%5.2f i",c3.i);

}
```