

# Semana 1 – Grupo Teórico

## Los POOsexuales

Escuela de Ingeniería en Sistemas  
poo@ues.edu.sv

## Comunismo

Universidad de El Salvador  
comunismo@ues.edu.sv

### Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent convallis orci arcu, eu mollis dolor. Aliquam eleifend suscipit lacinia. Maecenas quam mi, porta ut lacinia sed, convallis ac dui. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse potenti.

**2012 ACM Subject Classification** Replace ccsdesc macro with valid one, e.g., ‘Information systems → Question answering’

**Keywords and phrases** keyword one, keyword 2, three

**Digital Object Identifier** 10.4230/LIPIcs.CVIT.2023.1

### Tabla de Contenidos

<b>1</b>	<b>Clase Lunes</b>	<b>2</b>
1.1	Generalidades del curso . . . . .	2
1.1.1	Evaluaciones . . . . .	2
1.1.2	Acerca de la modalidad . . . . .	2
<b>2</b>	<b>Clase Miércoles</b>	<b>2</b>
2.1	Unidad 1 – TAD . . . . .	2
2.1.1	¿Qué es una abstracción? . . . . .	2
2.1.2	Concepto de abstracción . . . . .	2
2.1.3	TAD . . . . .	3
2.1.4	Estructura de Datos . . . . .	3
2.1.5	Componentes de una estructura . . . . .	3
2.1.6	Declaración de una estructura . . . . .	3
2.1.7	Definición de variables de estructuras . . . . .	4
2.1.8	Inicialización . . . . .	4
2.1.9	Acceso a los componentes . . . . .	5
2.1.10	Objetos → son TAD . . . . .	5
2.1.11	Ventajas . . . . .	5
2.1.12	Clasificación de las operaciones . . . . .	5
2.1.13	Especificación de los TAD . . . . .	6



© John Q. Public and Joan R. Public;  
licensed under Creative Commons License CC-BY  
42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 1; pp. 1:1–1:7



Leibniz International Proceedings in Informatics  
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Clase Lunes

### 1.1 Generalidades del curso

→ Pequeña introducción.

Programa de la asignatura esta disponible en el campus.

No nos interesa el lenguaje de programación, trabajaremos con

■ C

■ C++

■ Java

Nos dirán cuando empezar la tarea → es en grupos.

#### 1.1.1 Evaluaciones

Evaluación	Porcentaje
Parcial 1	15%
Parcial 2	20%
Práctica Evaluada 1	25%
Práctica Evaluada 2	25%
Tarea ex aula	15%

► Comentario. EL GL-06 se cierra.

#### 1.1.2 Acerca de la modalidad

Se está manejando que los teóricos sean virtuales y que las discusiones y evaluaciones eventualmente sean presenciales.

## 2 Clase Miércoles

► Nota 1. Entre tarde ~ 10 mins.

■ Calendarización = fechas tentativas (Jueves/Miércoles)

Las clases serán virtuales todo el ciclo, laboratorios esta en veremos (van a esperar 3 meses).

### 2.1 Unidad 1 – TAD

#### 2.1.1 ¿Qué es una abstracción?

Usando una abstracción es que se puede conducir sin conocer cómo esta fabricado.

#### 2.1.2 Concepto de abstracción

Capacidad de manejar un objeto (tema o idea) como un concepto general, sin considerar la enorme cantidad de detalles que pueden estar asociados con dicho objeto.

■ Beneficio principal: concede al programador la facilidad de pensar acerca del problema a resolver.

### 2.1.3 TAD

Una declaración de datos empaquetada junto con las operaciones que son significativas para el tipo de dato. → similar a clases.

Un TAD lo definimos nosotros

Tipo de dato → tipo `int` que tiene definidas sus operaciones.

Se encapsulan los datos y las operaciones y se ocultan de la vista del usuario.

### 2.1.4 Estructura de Datos

► Comentario 2. Joyanes → escritor de libros de programación.

Es una implementación física de un TAD.

se refiere a los datos almacenados en la memoria principal de la computadora.

■ Estructura archivo: almacenamiento periférico.

En C se puede utilizar una estructura.

► Nota 3. ¿Los structs en C#? → similar

En POO → la clase es una estructura.

### 2.1.5 Componentes de una estructura

■ Miembros: componentes individuales

Pueden contener valores de diferente tipo de datos.

### 2.1.6 Declaración de una estructura

■ Listing 1 Declaración de una estructura en C

```
81 struct <nombre>
82 {
83     <tipo de dato miembro 1> <nombre miembro 1>;
84     <tipo de dato miembro 2> <nombre miembro 2>;
85     <tipo de dato miembro 3> <nombre miembro 3>;
86     ...
87     <tipo de dato miembro n> <nombre miembro n>;
88 }
89
90
```

■ Listing 2 Ejemplo de una estructura en C

```
91 struct Libro
92 {
93     char titulo[30];
94     int anyo;
95     ...
96 }
97
98
```

Esto solo es la forma de la estructura, solo con esto no basta.

### 2.1.7 Definición de variables de estructuras

A una estructura se acceso utilizando una variable o variables que se deben definir después de la declaración de la estructura.

► **Definición 4** (Declaración). *Especifica el nombre y el tipo de datos de la estructura.*

► **Definición 5** (Definición). *Crea un área de memoria para el almacenamiento de los datos.*

Hay dos formas

1. Listandolas después de la llave de cierre de la declaración de la estructura.

#### ■ Listing 3 Forma 1

```

107 struct InfoAtleta
108 {
109     char nombre[30];
110     int edad;
111     char sexo;
112     char categoria[20];
113     float tiempo;
114     }Atleta1, Atleta2, Atleta3;
115

```

2. Listando el tipo de estructura creada seguida por las variables correspondientes en cualquier lugar del programa

#### ■ Listing 4 Forma 2

```

119 struct InfoAtleta Atleta1, Atleta2, Atleta3;
120
121

```

### 2.1.8 Inicialización

→ Valores iniciales.

Es posible inicializar una estructura de dos formas. La primera dentro de la sección de código del programa, y la segunda como parte de la definición.

#### ■ Listing 5 Forma 1

```

126 struct infoLibro
127 {
128     char titulo[60];
129     char autor[30];
130     char editorial[30];
131     int anyo;
132     }Libro1={"La investigación","Elssy Bonilla", "Alfaomega", 2009};
133
134

```

#### ■ Listing 6 Forma 2

```

135 struct infoLibro Libro1={"La investigación","Elssy Bonilla", "Alfaomega", 2009};
136
137

```

## 2.1.9 Acceso a los componentes

Utilizando el operador punto (.)

Camino directo

### Listing 7 Forma 1

```
Atleta1.edad=23;

strcpy(Atleta1.nombre,"Juan"); //strcpy es string copy

printf("Nombre del atleta");
gets(Atleta1.nombre)
```

Utilizando el operador puntero (->)

Sirve para acceder a los datos de la estructura a partir de un puntero. Para utilizar este operador primero se debe definir una variable puntero para direccionar hacia la estructura.

### Listing 8 Forma 2

```
struct InfoAtleta*ptrAtleta;

//Asignar la dirección de memoria de la variable Atleta 1 a la variable ptrAtleta
ptrAtleta = &Atleta1;

//Asignando valores directamente
ptrAtleta -> edad = 23;

strcpy(ptrAtleta1 -> nombre, "Juan");
```

► Comentario 6. No hemos visto punteros, se verá más adelante.

► Comentario 7. Los structs son parecidos a union, solo que el union solo puede guardar un miembro. No hay memoria para más.

## 2.1.10 Objetos → son TAD

Combinan en una sola unidad datos y funciones que operan sobre esos datos.

Usando el paradigma de la POO.

Clases en JAVA.

Paquetes → subdirectorios.

► Comentario 8. Static: No es necesario instanciar la clase para usarlo.

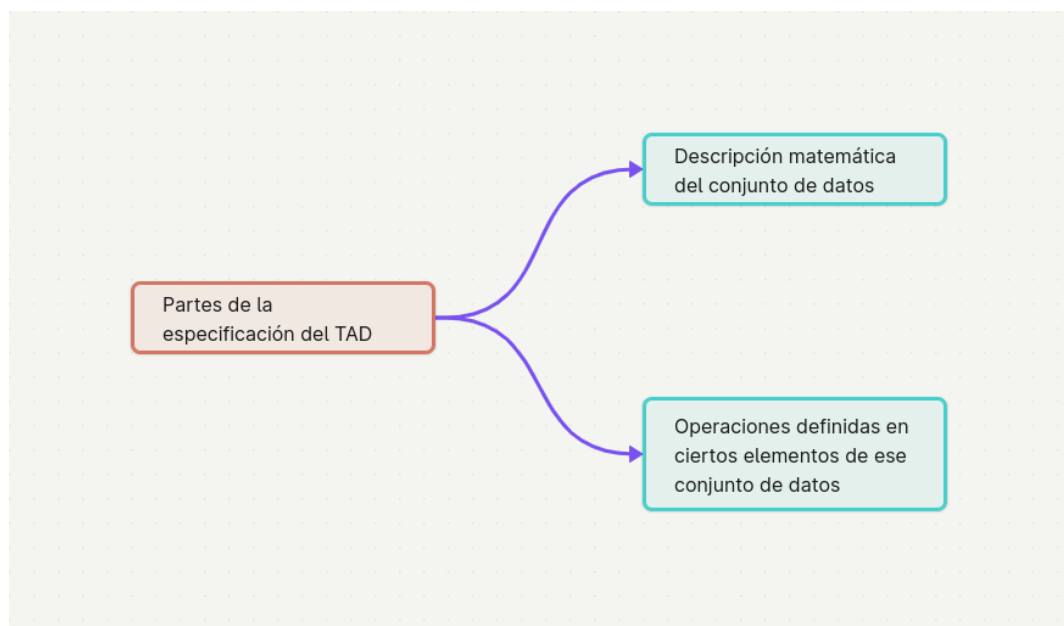
## 2.1.11 Ventajas

→ en el vídeo lo dijeron (en el PDF de la Unidad 1).

## 2.1.12 Clasificación de las operaciones

Criterio de clasificación	Tipos de operaciones	Descripción
Para crear objetos	Iniciales	Se usan para crear objetos del TAD que no requiere ningún objeto abstracto del mismo tipo
	Constructores	Usadas para crear objetos del TAD a partir de objetos del mismo tipo
Para transformar objetos	Simplificadoras	Operaciones que dan como resultado objetos descritos usando solamente operaciones iniciales y constructoras
Para analizar elementos	Analizadoras	El propósito es obtener información concerniente a cualquiera de los objetos del TAD

175 2.1.13 Especificación de los TAD



■ Figure 1

176 La especificación puede tener 3 enfoques:

177 ■ Informal (se usa lenguaje natural),

178 ■ Seminformal y

179 ■ Formal (se usan axiomas para describir las operaciones en su aspecto sintáctico y semántico).  
180

181 Solo trabajaremos con informal o semiformal  $\rightarrow$  formal (matemáticas más avanzadas).

#### 182 2.1.13.1 Especificación informal

183 1. Se establece el nombre del TAD y los datos que lo forman: TAD nombre del tipo (valores y descripción).  
184

185 2. Se especifica cada una de las operaciones con sus argumentos y una descripción funcional en el lenguaje natural.  
186

187 ► **Ejemplo 9.** No se guardo la captura :c

188 ► **Nota 10.** Se vieron 24/30 diapositivas.