

# Lenguajes de Programación 2

**Archivos, Flujos y Persistencia de Objetos**



**PUCP**

Esquema clásico de organización de datos:

Conj. bytes operados como unidad = campo

Conj. campos con significado = registro

Conj. registros almacenados = archivo

Conj. archivos = base de datos



**PUCP**

## Registro

Extremadamente ligado a la estructura de un archivo.

Algunos lenguajes fuerzan esta relación: Pascal

Otros dan mayor libertad: C, C++, Java, C#



Datos según el medio de almacenamiento

Temporales: Memoria volátil: RAM

Persistentes: Memoria no-volátil: Disco duro



**PUCP**

## Flujo

Secuencia de bytes que son accedidos en el mismo orden en que fueron creados.

Opera de manera similar a una cola.

El acceso es secuencial.

Cualquier memoria puede trabajarse como flujo.



## Objeto Persistente

El objeto reemplaza al registro.

Un objeto que es almacenado en memoria persistente  
se le conoce como *objeto persistente*.

Una forma de crearlo: *Serialización*.



## Serialización

Se dice que un objeto es *serializado* cuando es escrito hacia un *flujo* de una forma tal que puede ser leído luego y reconstruido.

El proceso inverso se llama *deserialización*.



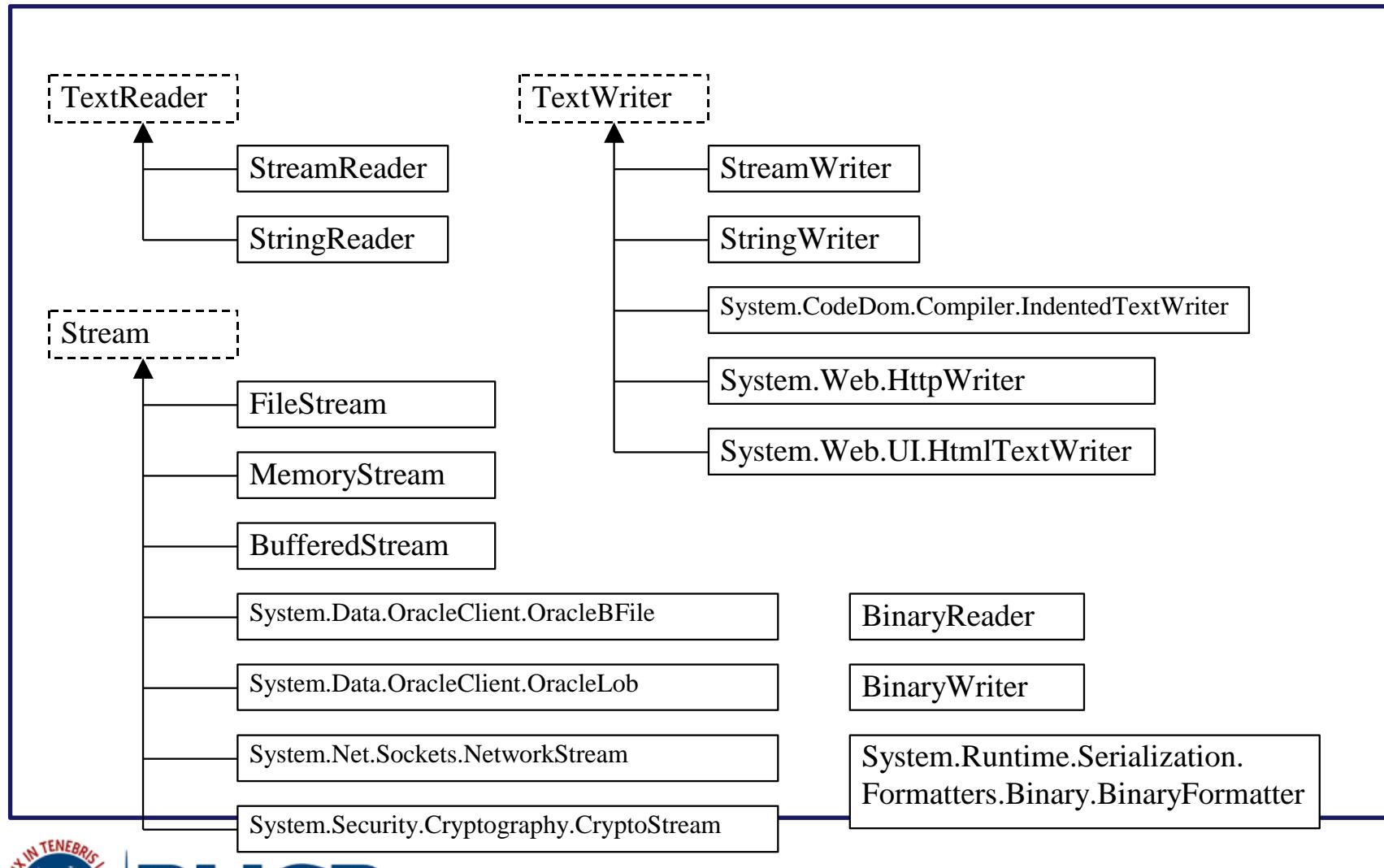
## Serialización: Problemas

Determinar quién es responsable de serializar y deserealizar un objeto: El objeto mismo o un mecanismo externo de serialización.

Como manejar los diagramas de clases.



# Manejo desde C#



## Manejo de Consola: Ejemplo I

```
Console.WriteLine( "Ingrese el texto a Leer." );
Console.WriteLine( " Finalizar con CTRL+Z." );
while(true) {
    int Entero = Console.In.Read();
    if( Entero == -1 )
        break;
    char Caracter = (char)Entero;
    Console.WriteLine( "Leido Entero=" + Entero );
    Console.WriteLine( ", caracter=" + Caracter );
}
Console.WriteLine( "Fin del ingreso" );
```



## Manejo de Consola: Ejemplo 2

```
Console.WriteLine( "Ingrese enteros." );
Console.WriteLine( "Finalice con '*'." );
int Contador = 0;
int Suma = 0;
while(true) {
    string Linea = Console.In.ReadLine();
    if( Linea == null ) break;
    if( Linea[0] == '*' ) break;
    Suma += Int32.Parse(Linea);
    Contador++;
}
Console.WriteLine( Contador + " enteros." );
Console.WriteLine( "Promedio " + (Suma / Contador) );
```



## Manejo de Consola: Ejemplo 3

```
Console.WriteLine( "Ingrese palabras." );
string Linea = Console.In.ReadLine();
char[] separadores = { ' ', ',', '.', ':' };
string[] Tokens = Linea.Split( separadores );
for(int i = 0; i < Tokens.Length; i++)
    Console.WriteLine("Token " + (i + 1) + " = " +
        Tokens[i]);
```



## Manejo de Archivo de Texto: Escritura

```
FileStream Archivo;
Archivo = new FileStream( args[0], FileMode.Create,
    FileAccess.Write );
StreamWriter Escriptor = new StreamWriter( Archivo );

Console.WriteLine( "Ingrese el texto a almacenar: " );
while( true ) {
    string Linea = Console.ReadLine();
    if( Linea == null ) break;
    Escriptor.WriteLine( Linea );
}
Escriptor.Close();
Archivo.Close();
```



## Manejo de Archivo de Texto: Lectura

```
FileStream Archivo;
Archivo = new FileStream( args[0], FileMode.Open,
 FileAccess.Read );
StreamReader Lector = new StreamReader( Archivo );
while( true ) {
    string Linea = Lector.ReadLine();
    if( Linea == null ) break;
    Console.WriteLine( Linea );
}
Lector.Close();
Archivo.Close();
```



## Manejo de Consola: Ejemplo I

```
...
while(true) {
    int Entero = System.in.read();
    if( Entero == -1 ) break;
    char Caracter = (char)Entero;
    System.out.println( "leido: numero=" + Entero +
        ", correspondiente al caracter=" + Caracter );
}
```



## Manejo de Consola: Ejemplo 2

```
...
BufferedReader Entrada = new BufferedReader(
new InputStreamReader( System.in ) );
int Contador = 0;
int Suma = 0;
while(true) {
    String Linea = Entrada.readLine();
    if( Linea == null )break;
    if( Linea.charAt(0) == '*' )break;
    Suma += Integer.parseInt( Linea );
    Contador++;
}
```



## Manejo de Consola: Ejemplo 3

```
BufferedReader Entrada = new BufferedReader(  
new InputStreamReader( System.in ) );  
String Linea = Entrada.readLine();  
StringTokenizer Tokens = new StringTokenizer( Linea );  
int Contador = 0;  
while ( Tokens.hasMoreTokens() )  
    System.out.println( "Token " + (++Contador) + " = " +  
        Tokens.nextToken() );
```



## Manejo de Texto: Escritura

```
FileWriter Escriptor = new FileWriter( NombreArchivo ) ;  
BufferedReader Entrada = new BufferedReader(   
    new InputStreamReader( System.in ) ) ;  
while(true) {  
    String Linea = Entrada.readLine() ;  
    if( Linea == null ) break;  
    Escriptor.write( Linea + "\n" ) ;  
}  
Escriptor.close();
```



## Manejo de Texto: Lectura

```
FileReader Lector = new FileReader( NombreArchivo );  
while( Lector.ready() )  
    System.out.print( (char)Lector.read() );  
Lector.close();
```

