



Clase 15

Serialización

Programación y Laboratorio II

Prof. Mauricio Cerizza

Serialización

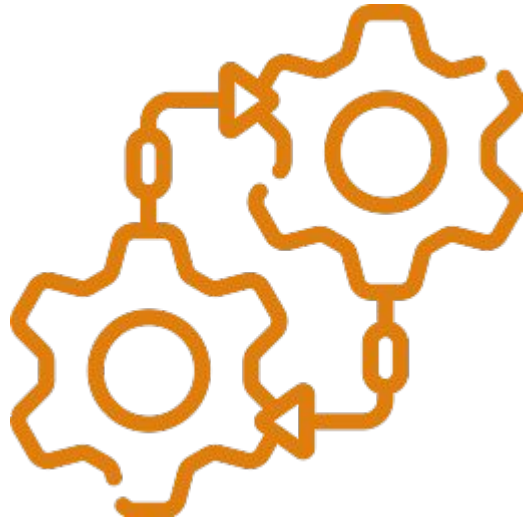
- ¿Qué es serializar?
- Usos de serializar
- Requisitos para serializar

Serialización XML

- ¿Qué es XML?
- Sintaxis de XML
- Serialización XML
- Deserilización XML

Serialización JSON

- ¿Qué es JSON?
- Sintaxis de JSON
- Serialización JSON
- Deserialización JSON



01.

Serialización

¿Qué es **SERIALIZAR?**

La **serialización** es el proceso de convertir el estado de un objeto a un formato específico o secuencia de bytes con el objetivo de transmitirlo a otro sistema o para almacenarlo en un archivo, en memoria o en una base de datos.

Su propósito principal es guardar el estado del objeto para recrearlo en algún momento en el futuro.

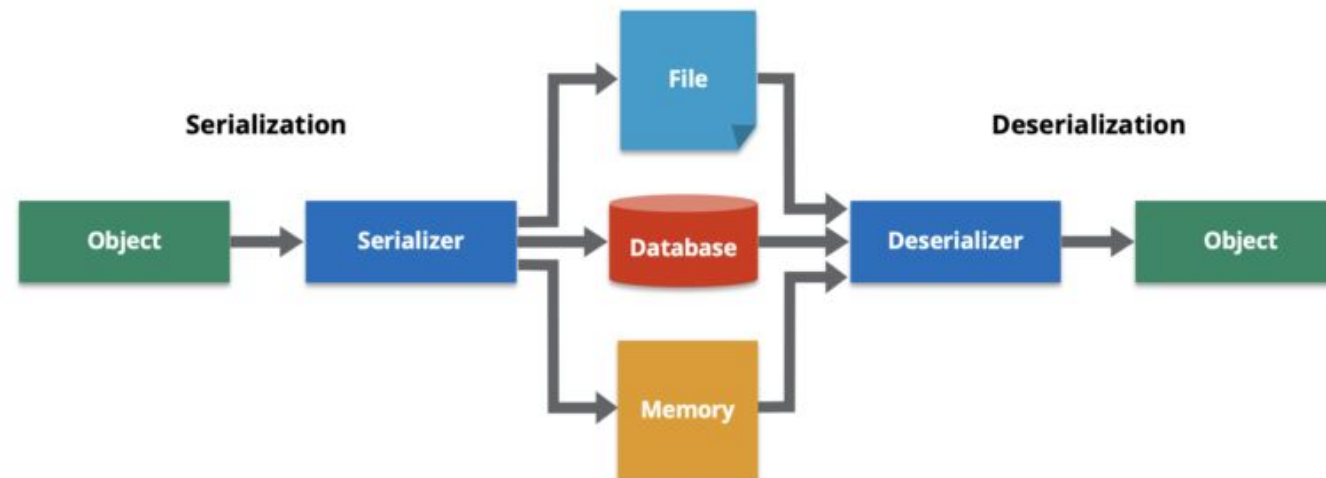
El proceso contrario, recuperar un objeto que fue serializado, se conoce como **deserialización**.



Usos de serializar

Permite al desarrollador almacenar el estado de un objeto y recrearlo a necesidad. El objetivo puede ser almacenarlo o intercambiar datos con otro sistema.

- Transferir el objeto a una aplicación o sistema externo a través de servicios web u otros sistemas de comunicación.
- Persistir el objeto en una unidad de almacenamiento.
- Persistir el objeto en una base de datos.
- Pasar el objeto a otro proceso o máquina.



Requisitos para serializar

- Los atributos a serializar deben ser públicos o estar expuestos a través de propiedades de lectura-escritura.
- Adicionalmente, para la serialización XML se debe tener un constructor público sin parámetros.



02.

Serialización XML



¿Qué es XML?

Extensible Markup Language (XML) es un lenguaje de marcado similar a HTML especificado por el *World Wide Web Consortium (W3C)*.

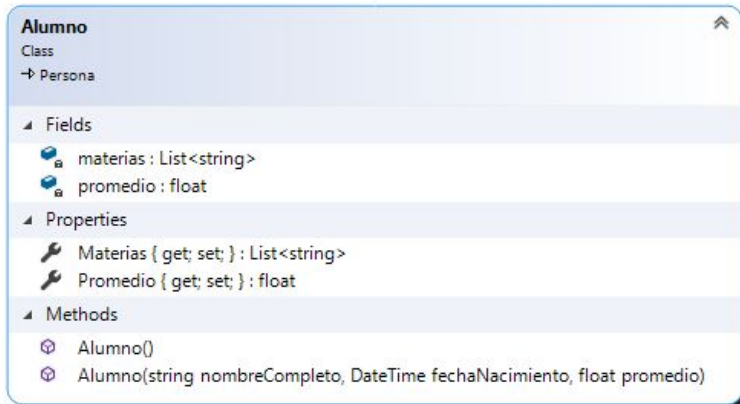
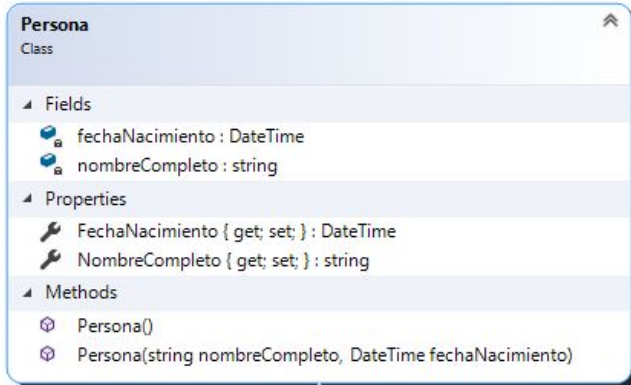
- Está diseñado para transferir datos.
- Es case-sensitive.
- Está compuesto de etiquetas `</>`.
- Permite extender el lenguaje.

W3C es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la *World Wide Web (www)* a largo plazo.

Sintaxis de XML

```
1 Alumno nuevoAlumno = new Alumno("Juan Perez",
2 new DateTime(2000, 12, 12), 7.8f);
3 nuevoAlumno.Materias.Add("Programación II");
4 nuevoAlumno.Materias.Add("Laboratorio de computación II");
5
6 Alumno nuevoAlumno2 = new Alumno("Leo Lopez",
7 new DateTime(2000, 12, 12), 7.8f);
8 nuevoAlumno2.Materias.Add("Arquitectura y sistemas operativos");
9 nuevoAlumno2.Materias.Add("Inglés II");
10 nuevoAlumno2.Materias.Add("Metodología de la investigación");
11
12 List<Alumno> nuevosAlumnos = new List<Alumno>()
13 {
14     nuevoAlumno,
15     nuevoAlumno2
16 };
```

Sintaxis de XML



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Array de objetos Alumno -->
3 <ArrayOfAlumno
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6 >
7   <Alumno> <!-- Objeto Alumno -->
8     <NombreCompleto>Juan Perez</NombreCompleto> <!-- Propiedad texto -->
9     <FechaNacimiento>2000-12-12T00:00:00</FechaNacimiento> <!-- Propiedad fecha -->
10    <Promedio>7.8</Promedio> <!-- Propiedad numérica -->
11    <Materias> <!-- Array anidado -->
12      <string>Programación II</string>
13      <string>Laboratorio de computación II</string>
14    </Materias>
15  </Alumno> <!-- Fin objeto -->
16  <Alumno> <!-- Otro Objeto Alumno -->
17    <NombreCompleto>Leo Lopez</NombreCompleto>
18    <FechaNacimiento>2000-12-12T00:00:00</FechaNacimiento>
19    <Promedio>7.8</Promedio>
20    <Materias>
21      <string>Arquitectura y sistemas operativos</string>
22      <string>Inglés II</string>
23      <string>Metodología de la investigación</string>
24    </Materias>
25  </Alumno> <!-- Fin objeto -->
26 </ArrayOfAlumno> <!-- Fin array -->
```

Serialización XML



```
1  Alumno alumno = new Alumno("Juan Perez", new DateTime(2000, 12, 12), 7.8f);  
2  
3  using (StreamWriter streamWriter = new StreamWriter("alumno.xml"))  
4  {  
5      XmlSerializer xmlSerializer = new XmlSerializer(typeof(Alumno));  
6  
7      xmlSerializer.Serialize(streamWriter, alumno);  
8  }
```

Deserialización XML

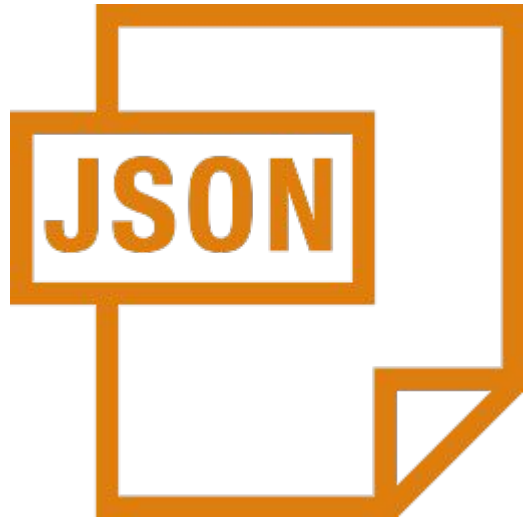
```
1  using (StreamReader streamReader = new StreamReader("alumno.xml"))
2  {
3      XmlSerializer xmlSerializer = new XmlSerializer(typeof(Alumno));
4
5      Alumno alumno = xmlSerializer.Deserialize(streamReader) as Alumno;
6
7      Console.WriteLine($"Nombre: {alumno.NombreCompleto}");
8      Console.WriteLine($"Fecha de nacimiento: {alumno.FechaNacimiento}");
9      Console.WriteLine($"Promedio: {alumno.Promedio}");
10 }
```

Ejercicios



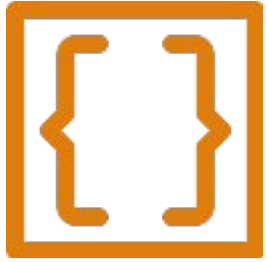
- I01 - La lista del super

https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/



03.

Serialización JSON



¿Qué es JSON?

JavaScript Object Notation (JSON) es un estándar abierto que usa texto de fácil lectura para almacenar y transferir objetos.

Se suele utilizar para transferir datos a través de la web.

Su sintaxis es de fácil lectura, ya que almacena la información de una manera organizada y fácil de acceder.

Está compuesto de pares propiedad-valor y arrays.

Sintaxis de JSON

```
1  {
2      "propiedadTexto": "valor",
3      "propiedadNumerica": 20,
4      "propiedadBooleana": true,
5      "propiedadNula": null
6  }
```

```
1  {
2      "nombreCompleto": "Juan Perez",
3      "saldo": 29638.59,
4      "edad": 30,
5      "direccion": {
6          "calle": "Av. Mitre",
7          "numero": 750,
8          "codigoPostal": "1870",
9          "piso": "2",
10         "departamento": null,
11         "localidad": "Avellaneda",
12         "provincia": "Buenos Aires",
13         "pais": "Argentina"
14     },
15     "telefonos":
16     [
17         {
18             "codigoArea": "011",
19             "numero": "42226465",
20             "interno": null,
21             "esCelular": false
22         },
23         {
24             "codigoArea": "011",
25             "numero": "42014133",
26             "interno": "116",
27             "esCelular": false
28         }
29     ]
30 }
```


Serialización JSON



```
1 Empleado empleado = new Empleado("Juan Perez", new DateTime(1990, 03, 25), 50000M);
2
3 // Genero el objeto de configuración de la serialización.
4 JsonSerializerOptions opciones = new JsonSerializerOptions();
5 opciones.WriteIndented = true;
6
7 // Serializo un objeto de tipo Empleado a formato JSON.
8 string jsonString = JsonSerializer.Serialize(empleado, opciones);
9
10 // Guardo el JSON desde un archivo.
11 File.WriteAllText("empleado.json", jsonString);
```

Serialización JSON



```
1  {  
2    "Salario": 50000,  
3    "NombreCompleto": "Juan Perez",  
4    "FechaNacimiento": "1990-03-25T00:00:00"  
5  }
```

Deserialización JSON



```
1 // Obtengo el JSON desde un archivo.
2 string jsonString = File.ReadAllText("empleado.json");
3
4 // Deserializo un objeto de tipo Empleado a partir de JSON.
5 Empleado empleado = JsonSerializer.Deserialize<Empleado>(jsonString);
6
7 Console.WriteLine($"Nombre: {empleado.NombreCompleto}");
8 Console.WriteLine($"Fecha de nacimiento: {empleado.FechaNacimiento}");
9 Console.WriteLine($"Salario: {empleado.Salario}");
```

Ejercicios



- I02 - Cartelera

https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/

Ejercicios



- Ejercicio C01 - Siempre quise tener un notepad-serializador ($\geq \nabla \leq$)

https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/