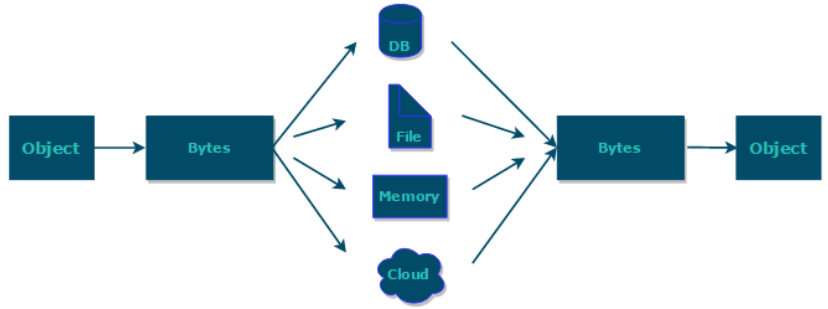
Serialização

Serialização

- A serialização é o processo de converter um objeto em um fluxo de bytes para armazenar o objeto ou transmiti-lo para a memória, um banco de dados ou um arquivo.
- Sua finalidade principal é salvar o estado de um objeto para recriá-lo quando necessário.
- O processo inverso é chamado desserialização.

Serialização

- O objeto é serializado para um fluxo que carrega os dados.
- O fluxo também pode ter informações sobre o tipo do objeto, como sua versão, cultura e nome do assembly.
- Desse fluxo, o objeto pode ser armazenado em um banco de dados, um arquivo ou memória.



Serialização

- O processo de serialização é independente da aplicação, um dado serializado em uma plataforma deve poder ser deserializada por qualquer outra.

- Visando garantir a comunicação entre aplicações.

Serialização

- Inicialmente, quando a capacidade de armazenamento das máquinas e transferência de dados na rede era baixo, a serialização era feita convertendo os objetos/estruturas de dados para *stream* de *bytes*, *byte-stream-based encoding*.

Serialização

- Com o avanço tecnológico, permitiu-se o uso de um padrão não tão compacto, quanto um simples *streams* de bytes, surgindo a proposta de se padronizar o uso para XML, um *text-based encoding*, que tem a vantagem de ser compreensível a humanos, pode ser aberto em leitor de texto simples e cumprir o propósito de ser entendível pelo programas, independente de sua linguagem.

- Depois sugiram alternativas mais leves e de leitura mais amigáveis para humanos como JSON, que hoje é o mais utilizado, e o YAML.

Serialização - Binária

- Serialização:

```
string serializar = "Oi gente";
FileStream fs = new FileStream(@"C:\Teste\Ricardo.data", FileMode.Create);
BinaryFormatter bf = new BinaryFormatter();
bf.Serialize(fs, serializar);
fs.Close();
```

- Desserialização

```
FileStream fs = new FileStream(@"C:\Teste\Ricardo.data", FileMode.Open);
BinaryFormatter bf = new BinaryFormatter();
string dadosDesserializados = (string)bf.Deserialize(fs);
fs.Close();
```

Serialização - XML

- Serialização:

```
FileStream stream = new FileStream(@"C:\Teste\pessoa1.xml", FileMode.Create);
XmlSerializer xml = new XmlSerializer(typeof(Pessoa));

xml.Serialize(stream, p);
stream.Close();
```

- Desserialização

```
XmlSerializer serializer = new XmlSerializer(typeof(Pessoa));
StreamReader reader = new StreamReader(@"C:\Teste\pessoa1.xml");
Pessoa deserialized = (Pessoa)serializer.Deserialize(reader.BaseStream);
reader.Close();
```

Serialização - JSON

- Serialização:

```
JsonSerializer serializer = new JsonSerializer();
StreamWriter sw = new StreamWriter(@"C:\Teste\pessoa.json");
JsonWriter writer = new JsonTextWriter(sw);
serializer.Serialize(writer, p);
sw.Close();
writer.Close();
```

- Desserialização

```
string json = File.ReadAllText(@"C:\Teste\pessoa.json");

Pessoa p... = JsonConvert.DeserializeObject<Pessoa>(json);
```

Serialização - SOAP

- Serialização:

```
FileStream stream = new FileStream(@"C:\Teste\pessoa1.data", FileMode.Create);
SoapFormatter soap = new SoapFormatter();

soap.Serialize(stream, p);
stream.Close();
```

- Desserialização

```
FileStream fs = new FileStream(@"C:\Teste\pessoa1.data", FileMode.Open);
Pessoa p = new Pessoa();
SoapFormatter soap = new SoapFormatter();
p = (Pessoa)soap.Deserialize(fs);

fs.Close();
```