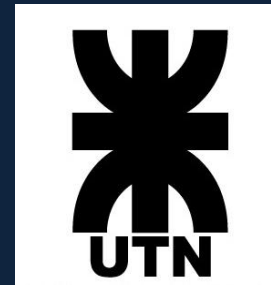


# Serialización



# Definiciones

- Es un proceso de codificación de un objeto con el fin de transmitirlo **como una serie de bytes**, y frecuentemente en un formato **humanamente legible**.
- La serie de bytes o el formato pueden ser usados para recrear un nuevo objeto que es idéntico en todo al original, incluido su estado interno.
- Es un mecanismo ampliamente usado para transportar objetos a través de una red, para hacer persistente un objeto en un archivo o base de datos, o para **distribuir objetos idénticos a varias aplicaciones o localizaciones**.

# Ventajas

- Elimina las serializaciones **ad-hoc** a las que suele recurrirse para transmisión de datos estructurados.
- El **payload** no sólo incluye los datos, sino **información de la estructura** de los mismos, dentro de la misma transmisión.
- Permite independizar tanto información como estructura de su soporte de persistencia. Puedo obtener datos de MySQL, serializarlos, transmitirlos por un canal serie, recibirlos, deserializarlos y guardarlos en una base SQLite, todo de forma más o menos transparente.

# Terminología

Serialize  
Marshalling  
"Stringify"  
Dump

Deserialize  
Unmarshalling  
Parse  
Load

# XML (eXtensible Markup Language)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

- Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.
- Proviene de SGML (desarrollado en IBM en los 70s), del que también surge el HTML (creado por Tim Berners-Lee) y es un subset de XML.
- El **payload** de un documento XML suele ser bastante mayor comparado con el de los datos que se desean transmitir.

# JSON (JavaScript Object Notation)

- Estándar de ***payload reducido*** para el intercambio de información estructurada.
- Proveniente del lenguaje de programación **JavaScript**, y fue propuesto principalmente por Douglas Crockford como formato de serialización.
- Aunque JavaScript lo soporta de forma nativa, es independiente del mismo, funciona en cualquier lenguaje.

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "age": 25,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": 10021  
  },  
  "phoneNumbers": [  
    {  
      "type": "home",  
      "number": "212 555-1234"  
    },  
    {  
      "type": "fax",  
      "number": "646 555-4567"  
    }  
  ]  
}
```

# YAML (YAML Ain't Markup Language)

- Desarrollado por Clark Evans en 2001.
- Además de ser un formato de ***payload reducido*** en comparación con XML, uno de sus principales objetivos es ser **muy fácilmente legible por el ser humano**.
- Muy utilizado en archivos de configuración.

```
---
receipt:    Oz-Ware Purchase Invoice
date:       2012-08-06
customer:
  given:    Dorothy
  family:   Gale

items:
  - part_no: A4786
    descrip: Water Bucket (Filled)
    price:   1.47
    quantity: 4

  - part_no: E1628
    descrip: High Heeled "Ruby" Slippers
    size:    8
    price:   100.27
    quantity: 1

bill-to: &id001
street: |
  123 Tornado Alley
  Suite 16
city:   East Centerville
state:  KS

ship-to: *id001

specialDelivery: >
  Follow the Yellow Brick
  Road to the Emerald City.
  Pay no attention to the
  man behind the curtain.
...
```

# Implementaciones

- Todos los lenguajes de programación de “muy” alto nivel tienen librerías desarrolladas para serializar y deserializar XML, JSON y YAML.
- **El lenguaje C** también cuenta con estas librerías, pero las características ***strongly-typed*** de sus definiciones de estructuras hacen un poco complicada su manipulación.

*Sin embargo...*

```
printf("{\“id\”: %d, \“nombre\”: \“%s\”}", (int)id, (char*)nombre);
```



# Preguntas

