UdeA
Analytics &
DataScience

DATA STREAMING Y SERVICIOS EN LA NUBE

Procesamiento de Datos

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Magister - Efraín Alberto Oviedo
alberto.oviedo@udea.edu.co

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN ANALÍTICA Y CIENCIA DE DATOS

AGENDA

UdeA
Analytics &
DataScience

1. **Serialización de Datos**
   - XML
   - Json
   - Yaml
2. Protocol Buffers
3. Apache Thrift

Serialización de Datos

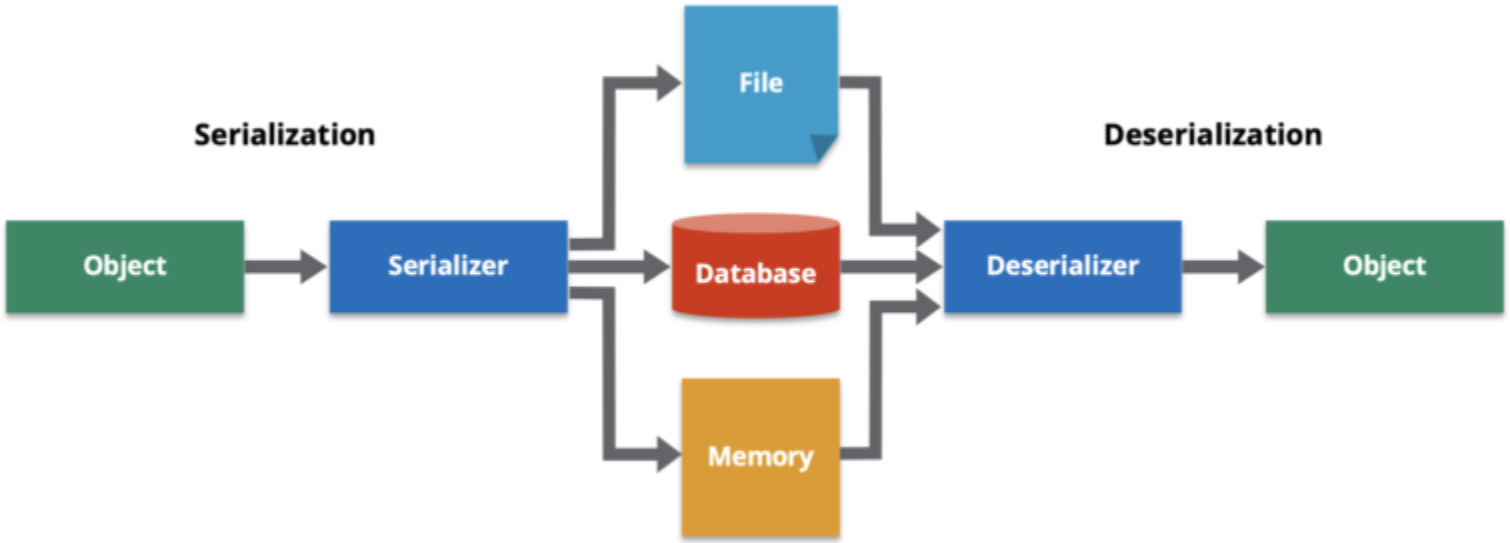
- Proceso de codificación de un objeto en un flujo de bytes con el fin de almacenarlo o transmitirlo. El objeto recibido/leído, es considerado como idéntico al original

- También suele ser utilizado como método de persistencia de objetos en forma de archivos, en memoria o en base de datos

- Otra de sus aplicaciones, permite detectar cambios en las variables en función del tiempo

Serialización de Datos

Proceso de Serialización



https://hazelcast.com/glossary/serialization/

Lenguajes de Serialización

https://community.cisco.com/t5/developer-general-knowledge-base/xml-vs-json-vs-yaml/ta-p/4729758

AGENDA

1. **Serialización de Datos**
   - **XML**
   - Json
   - Yaml
2. Protocol Buffers
3. Apache Thrift

XML

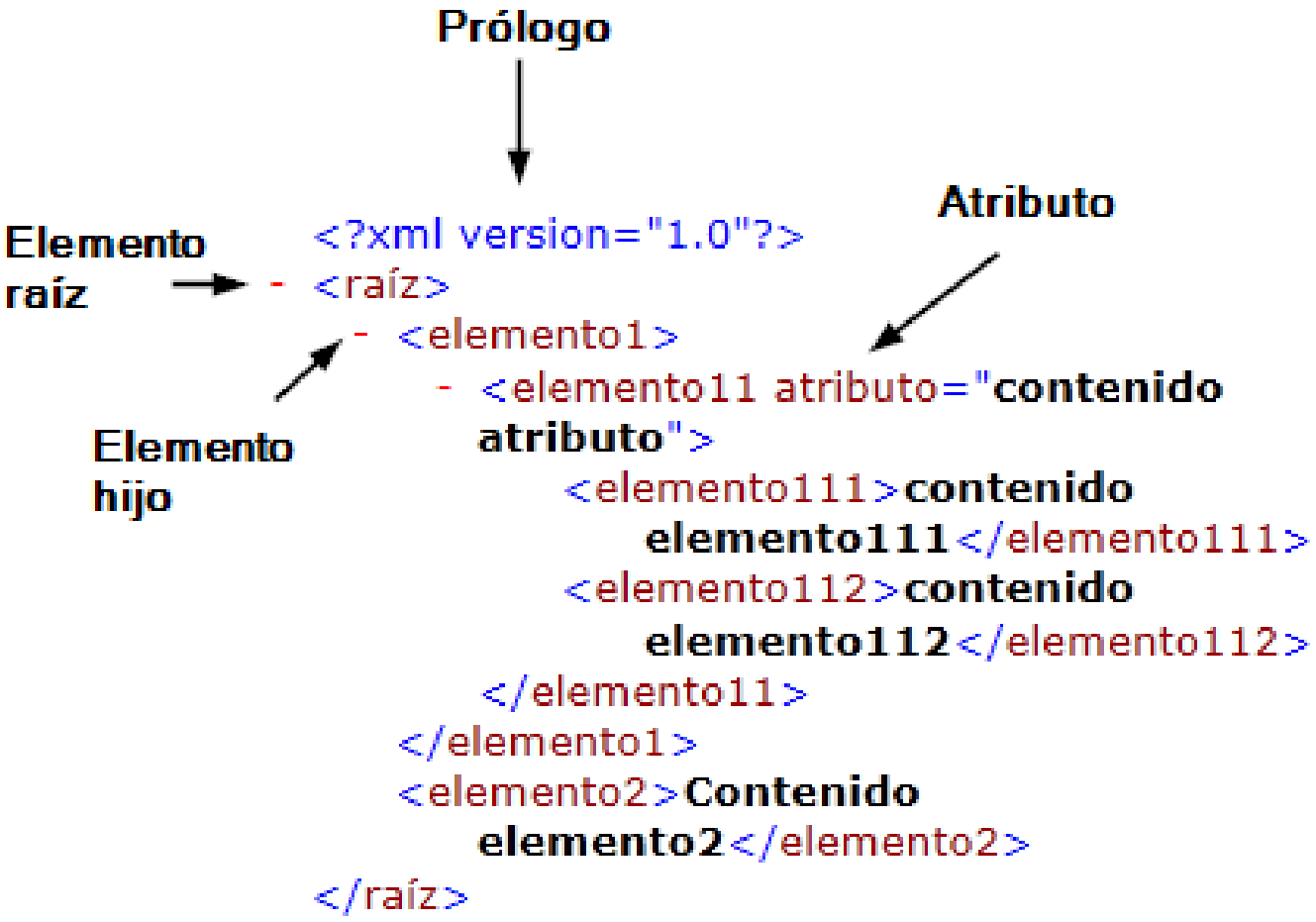
XML (Lenguaje extensible de marcas)

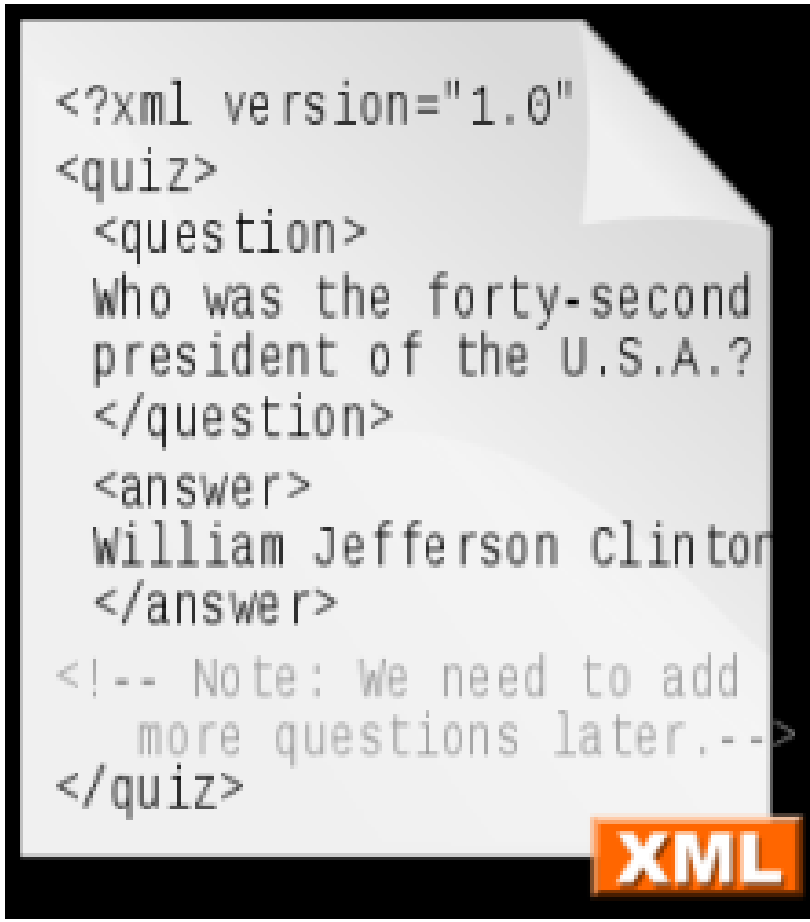
- Estándar internacionalmente conocido

- Define etiquetas personalizadas para describir y organizar datos

- Se utiliza para transferir información de productos, transacciones, inventario, etc.

XML





https://www.filmac.com/xml-el-lenguaje-sin-vocabulario-de-uso-universal/

XML

ID	Nombre	Edad	Sexo	Profesión	Salario
1	Juan	33	M	Ingeniero	4.500.000
2	Ana	38	F	Arquitecta	6.200.000

```
<?xml version="1.0" encoding="UTF-8" ?>
<empleados>
        <Id>1</Id>
        <Nombre>Juan</Nombre>
        <Edad>33</Edad>
        <Sexo>M</Sexo>
        <Profesión>Ingeniero</Profesión>
        <Salario>4500000</Salario>
</empleados>
<empleados>
        <Id>2</Id>
        <Nombre>Ana</Nombre>
        <Edad>38</Edad>
        <Sexo>F</Sexo>
        <Profesión>Arquitecta</Profesión>
        <Salario>6200000</Salario>
</empleados>
```

AGENDA

1. **Serialización de Datos**
   - XML
   - **Json**
   - Yaml
2. Protocol Buffers
3. Apache Thrift

JSON

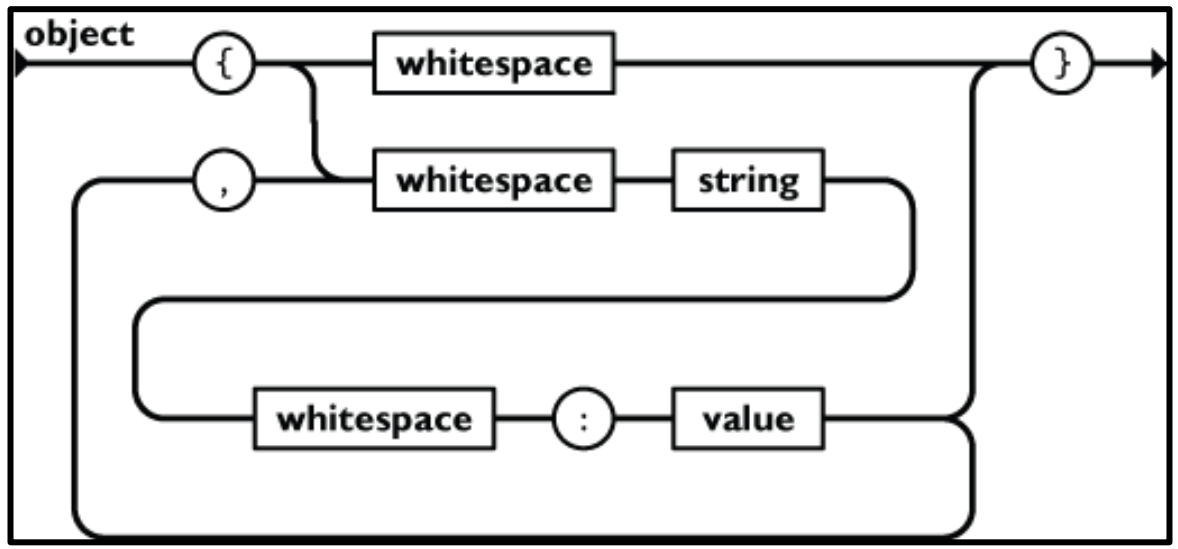
JSON (JavaScript Object Notation)

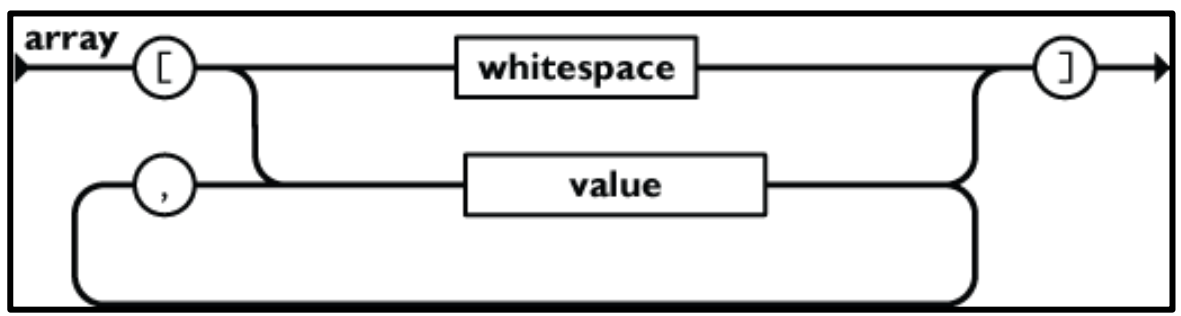
- Representa datos estructurados en la sintaxis de JavaScript

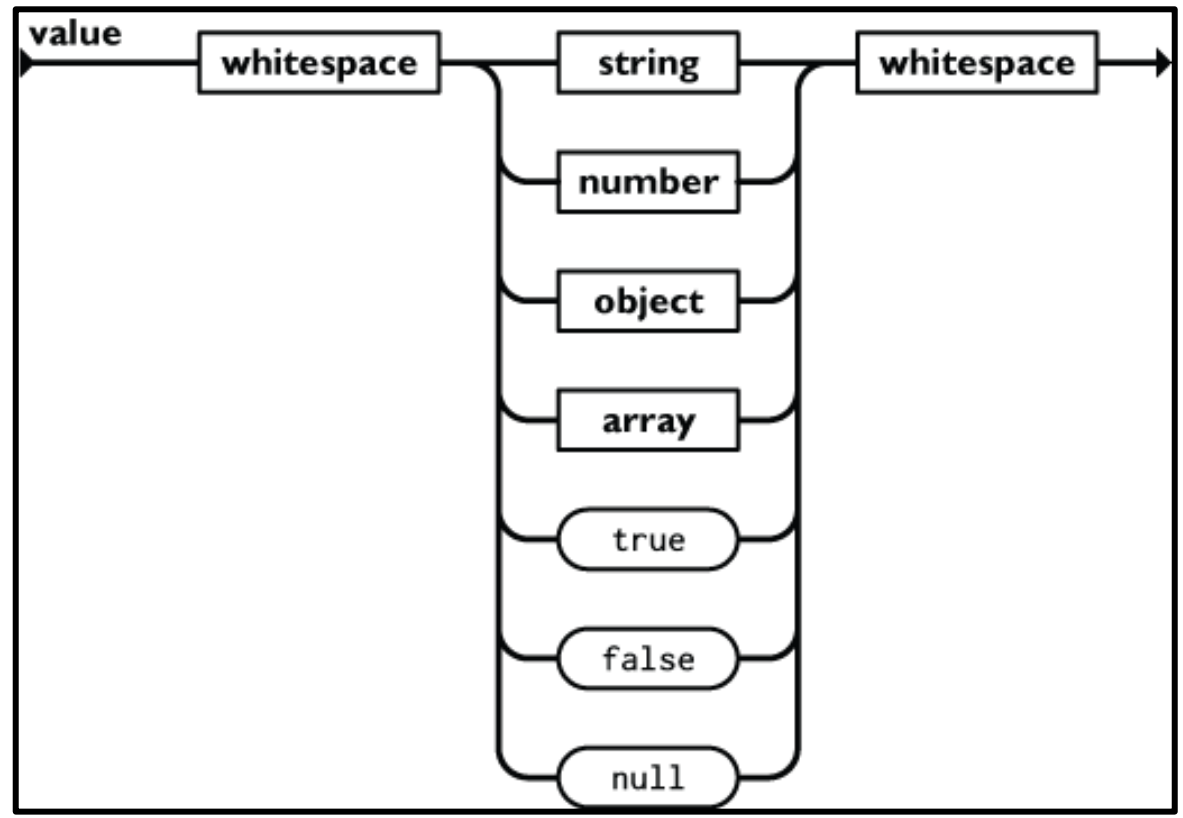
- Permite transmitir datos en aplicaciones Web

- Es un archivo de texto con extensión .json

JSON







https://www.json.org/

JSON

ID	Nombre	Edad	Sexo	Profesión	Salario
1	Juan	33	M	Ingeniero	4.500.000
2	Ana	38	F	Arquitecta	6.200.000


```
{
  "empleados": [
    {
      "Id": 1,
      "Nombre": "Juan",
      "Edad": 33,
      "Sexo": "M",
      "Profesión": "Ingeniero",
      "Salario": 4500000
    },
    {
      "Id": 2,
      "Nombre": "Ana",
      "Edad": 38,
      "Sexo": "F",
      "Profesión": "Arquitecta",
      "Salario": 6200000
    }
  ]
}
```

AGENDA

1. **Serialización de Datos**
   - XML
   - Json
   - **Yaml**
2. Protocol Buffers
3. Apache Thrift

YAML

YAML (YAML Ain't Markup Language)

- Lenguaje de serialización de datos utilizado frecuentemente en el diseño de archivos de configuración.

- Fue diseñado para ser útil y amigable para las personas que trabajan con datos

- La sangría se usa para la estructura, los dos puntos separan los pares clave valor y los guiones crean listas con viñetas

- Los archivos generados tienen extensión .yml o .yaml

```
YAML: YAML Ain't Markup Language™

What It Is:
  YAML is a human-friendly data serialization
  language for all programming languages.

YAML Resources:
  YAML Specifications:
  - YAML 1.2:
    - Revision 1.2.2       # Oct 1, 2021 *New*
    - Revision 1.2.1       # Oct 1, 2009
    - Revision 1.2.0       # Jul 21, 2009
  - YAML 1.1
  - YAML 1.0

  YAML Matrix Chat:  '#chat:yaml.io'     # Our New Group Chat Room!
  YAML IRC Channel:  libera.chat#yaml    # The old chat
  YAML News:         twitter.com/yamlnews
  YAML Mailing List: yaml-core           # Obsolete, but historical

  YAML on GitHub:                        # github.com/yaml/
    YAML Specs:         yaml-spec/
    YAML 1.2 Grammar:   yaml-grammar/
    YAML Test Suite:    yaml-test-suite/
    YAML Issues:        issues/

  YAML Reference Parsers:
  - Generated Reference Parsers
  - YPaste Interactive Parser
```

YAML

https://yaml.org/

YAML

ID	Nombre	Edad	Sexo	Profesión	Salario
1	Juan	33	M	Ingeniero	4.500.000
2	Ana	38	F	Arquitecta	6.200.000

```yaml
empleados:
  - Id: 1
    Nombre: Juan
    Edad: 33
    Sexo: M
    Profesion: Ingeniero
    Salario: 4500000
  - Id: 2
    Nombre: Ana
    Edad: 38
    Sexo: F
    Profesion: Arquitecta
    Salario: 6200000
```

Ejercicio – Representación de Datos

Represente en formato XML, JSON y YAML la información relacionada con las ciudades que ha visitado (País, departamento, fecha de visita, duración, motivo, etc)

Valide el ejercicio con un editor online

- https://jsoneditoronline.org/
- https://jsonformatter.org/
- https://jsonformatter.org/yaml-formatter

Ejercicio - Serialización de Datos

Notebook: 01_Serialización_de_Datos

Ejercicio:

Serialice en formato XML, JSON y YAML la información relacionada con las ciudades que ha visitado (País, departamento, fecha de visita, duración, motivo, etc)

AGENDA

UdeA
Analytics &
DataScience

1. Serialización de Datos
   - XML
   - Json
   - Yaml
2. **Protocol Buffers**
3. Apache Thrift

Protocol Buffers

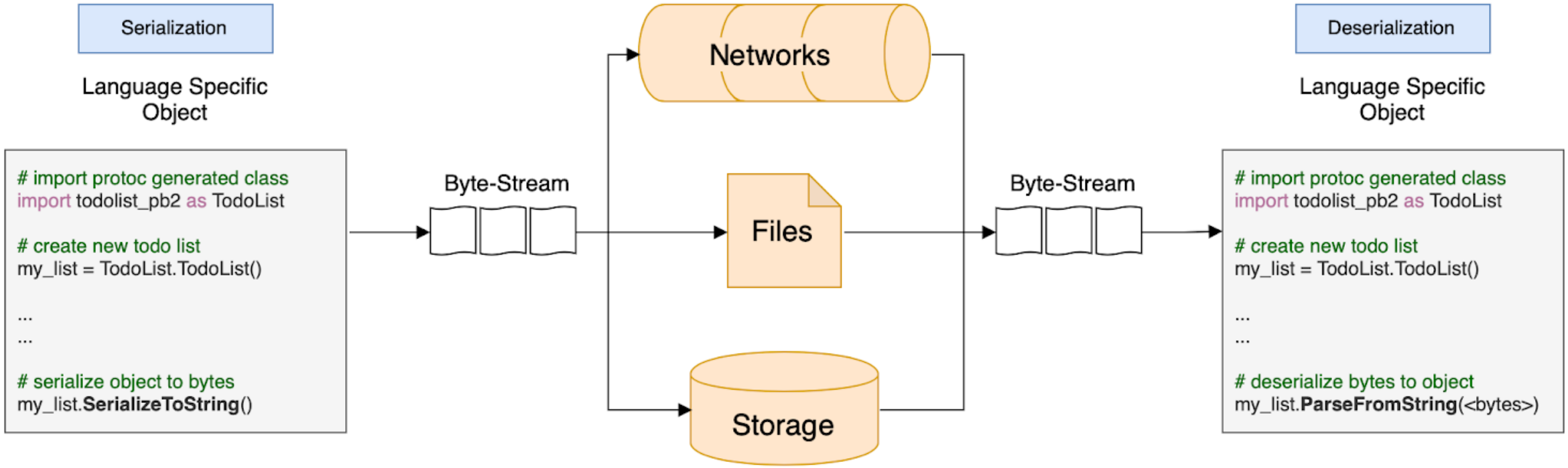
- Formato de serialización de datos que facilita el intercambio de datos entre aplicaciones, aún si están desarrolladas en distintos lenguajes.

- Desarrollado por Google y liberado en 2008 como proyecto de código abierto. Pensado para reemplazar el formato xml con el objetivo de ser más simple y rápido (entre 3 y 10 veces más pequeña, y entre 20 y 100 veces más rápida)

- La comunicación se realiza en formato cliente – servidor de forma local o remota

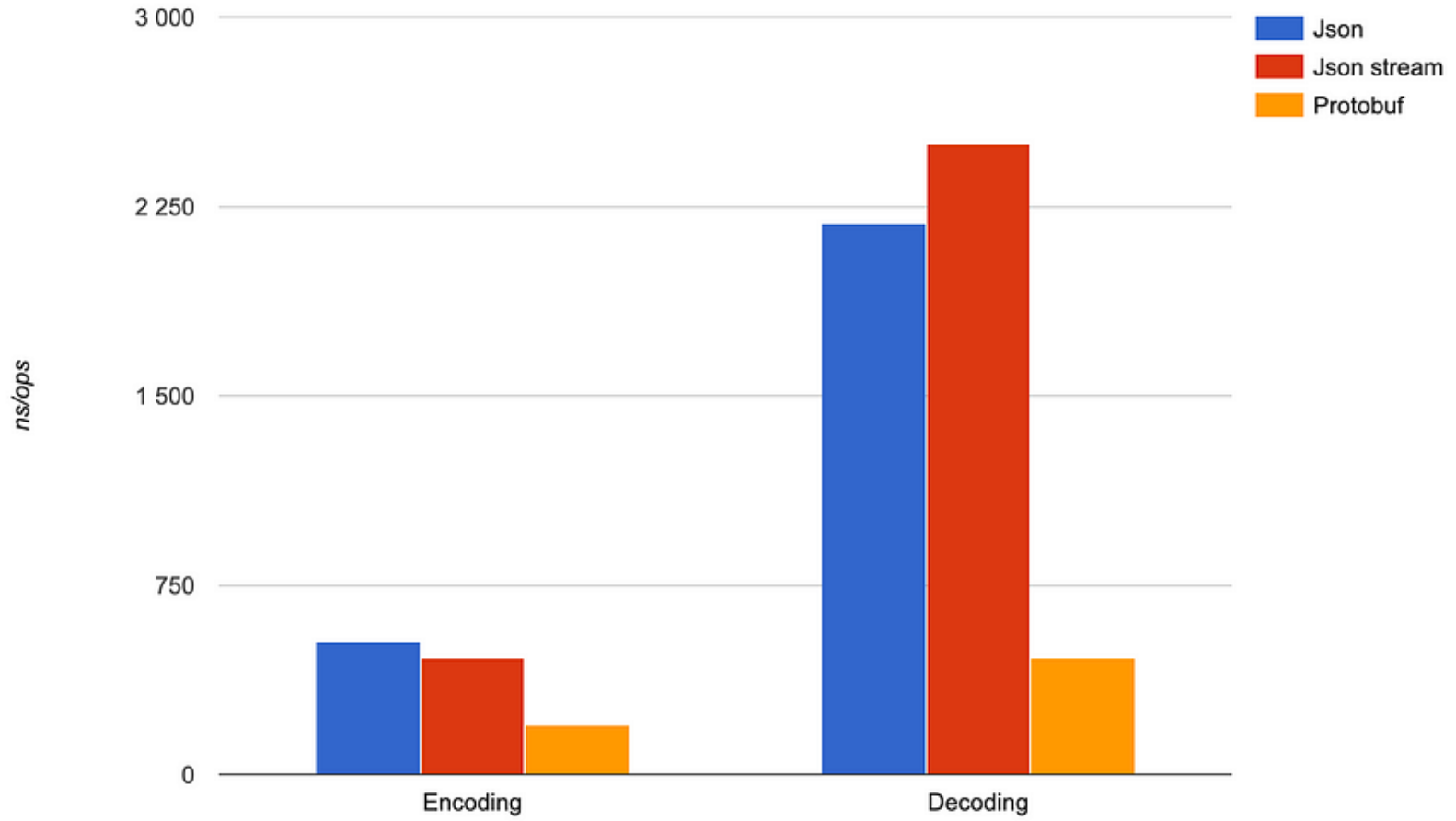
- Asigna etiquetas a los datos para utilizar un formato binario

Protobuf - Proceso



https://www.freecodecamp.org/news/googles-protocol-buffers-in-python/

Protobuf - Rendimiento



https://mnwa.medium.com/what-the-hell-is-protobuf-4aff084c5db4

Protobuf – Empleados

empleado.proto

```
syntax = "proto3";
message Empleado {
int32 id = 1;
string nombre = 2;
int32 edad = 3;
string sexo = 4;
string profesion = 5;
int32 salario = 6;
}
```

```
emp1.id = 1
emp1.nombre = 'Juan'
emp1.edad = 33
emp1.sexo = 'M'
emp1.profesion = 'Ingeniero'
emp1.salario = 4500000
```

```
b'\x08\x01\x12\x04Juan\x18!"\x01M*\tIngeniero0\xa0\xd4\x92\x02'
```

```
syntax = "proto2";

package tutorial;

message Person {
  optional string name = 1;
  optional int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    optional string number = 1;
    optional PhoneType type = 2 [default = HOME];
  }

  repeated PhoneNumber phones = 4;
}

message AddressBook {
  repeated Person people = 1;
}
```

Protobuf

Tutorial
https://protobuf.dev/getting-started/pythontutorial/

Guía de programación
https://protobuf.dev/programming-guides/proto3/

Notebook:
01_Serialización_de_Datos

AGENDA

UdeA
Analytics &
DataScience

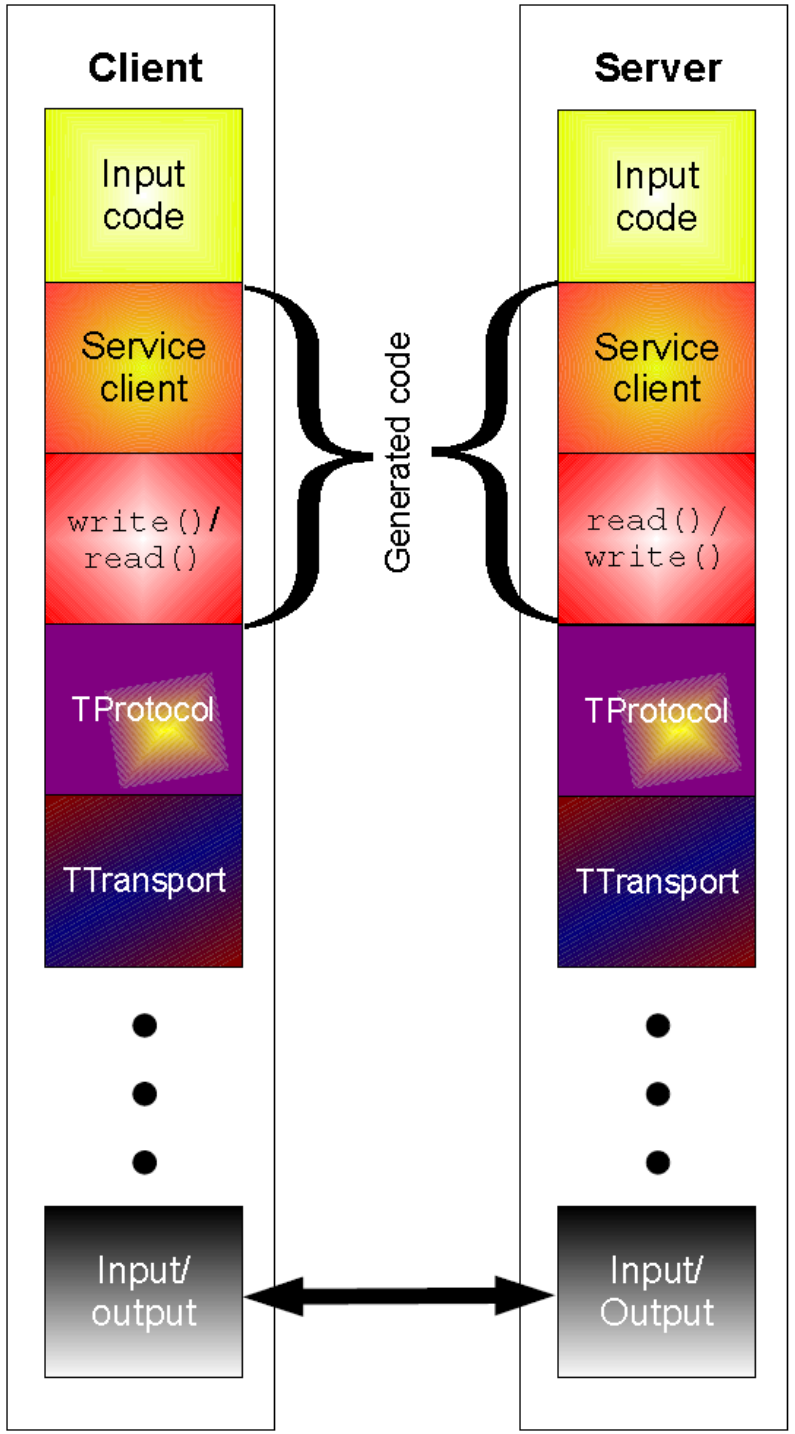
1. Serialización de Datos
   - XML
   - Json
   - Yaml
2. Protocol Buffers
3. **Apache Thrift**

Apache Thrift

- Protocolo para la gestión de los macrodatos, que a demás de permitir **serialización** de objetos en formato binario, incluye el desarrollo de interfaces de **servicios** utilizando archivos de extensión .thrift

- Se enfoca principalmente en la capa de comunicación entre el cliente y el servidor RPC (Llamado a procedimientos remotos)

- Fue desarrollado por **Facebook** y posteriormente liberado, es de código abierto y se utiliza generalmente en C++ pero se puede adaptar a otros lenguajes



Apache Thrift

A partir del archivo .thrift se genera código tanto para el cliente como para el servidor que permite el intercambio de datos.

Protocolos (Binario, Json, etc)

Transporte (Socket, Memoria, etc)

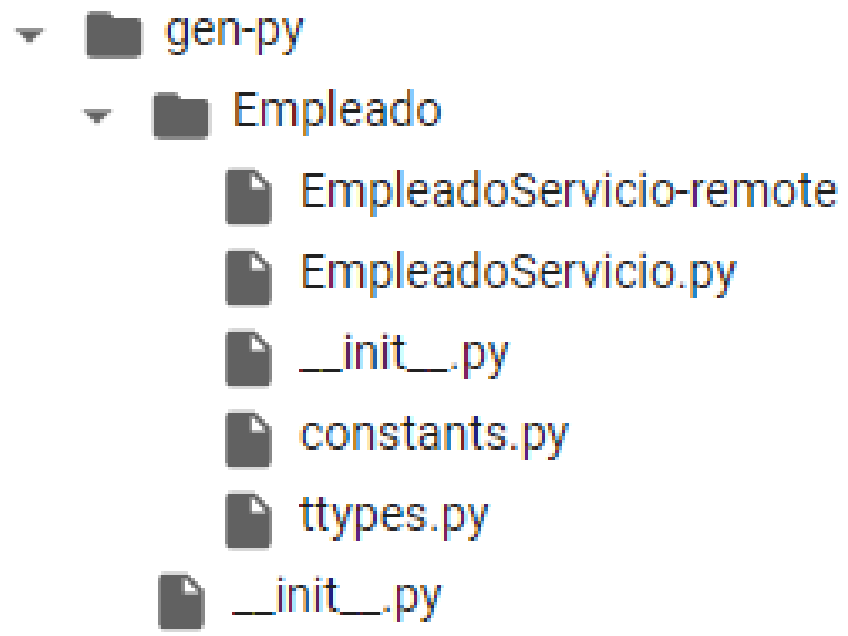
Apache Thrift

empleado.thrift

```
struct Empleado {
  1: i32 id,
  2: string nombre,
  3: i32 edad,
  4: string sexo,
  5: string profesion,
  6: i32 salario,
}

service EmpleadoServicio {
  string msg(),
  string send_Emp(1: Empleado
new_emp),
  bool mayor40(1: Empleado emp)
}
```

!thrift -r --gen py local/data/empleado.thrift



https://dataschool.com/data-modeling-101/running-jupyter-notebook-on-an-ec2-server/