

# SERIALIZACIÓN

- García Morales Rebeca  
- Ramírez Vega Gerardo



## ¿QUÉ ES LA SERIALIZACIÓN

Podemos entender la serialización como “un proceso en el cual podemos convertir un objeto en una secuencia de bytes para almacenar el objeto o transmitirlo en la memoria. Ya sea también para una base de datos o un archivo, su propósito principal es guardar el estado de un objeto para volver a crearlo cuando sea necesario.”

Según Microsoft en su sitio web oficial para desarrolladores MSDN, JavaScript Object Notation (JSON) “es un formato de codificación eficaz que permite intercambios rápidos de cantidades pequeñas de datos entre los exploradores de cliente, tales como Internet Explorer, Google Chrome y servicios web, por lo cual se considera un gran método de desarrollo web para la serialización en la creación de aplicaciones.” [2]



## ¿PARA QUÉ SIRVE?

“La serialización permite al desarrollador guardar el estado de un objeto y volver a crearlo según sea necesario, gracias a que proporciona almacenamiento de los objetos e intercambio de datos. A través de la serialización, un desarrollador puede realizar acciones como enviar un objeto a una aplicación remota por medio de un servicio web, pasar un objeto de un dominio a otro, pasar un objeto a través de un firewall como una cadena XML o mantener la seguridad o información específica del usuario entre aplicaciones.”

[2]





## SERIALIZACIÓN PARA APLICACIONES JAVA

Para serializar datos puede utilizar la serialización eXtreme Data format (XDF), Java, el plug-in ObjectTransformer o los plug-ins DataSerializer. Para optimizar la serialización con cualquiera de estas opciones, puede utilizar la modalidad COPY\_TO\_BYTES para mejorar el rendimiento hasta un 70 por ciento.

Las siguientes figuras nos ayudan a determinar qué tipo de método de serialización es más apropiado para las necesidades de desarrollo.”

# TRANSMISIÓN DE DATOS CON JSON

X

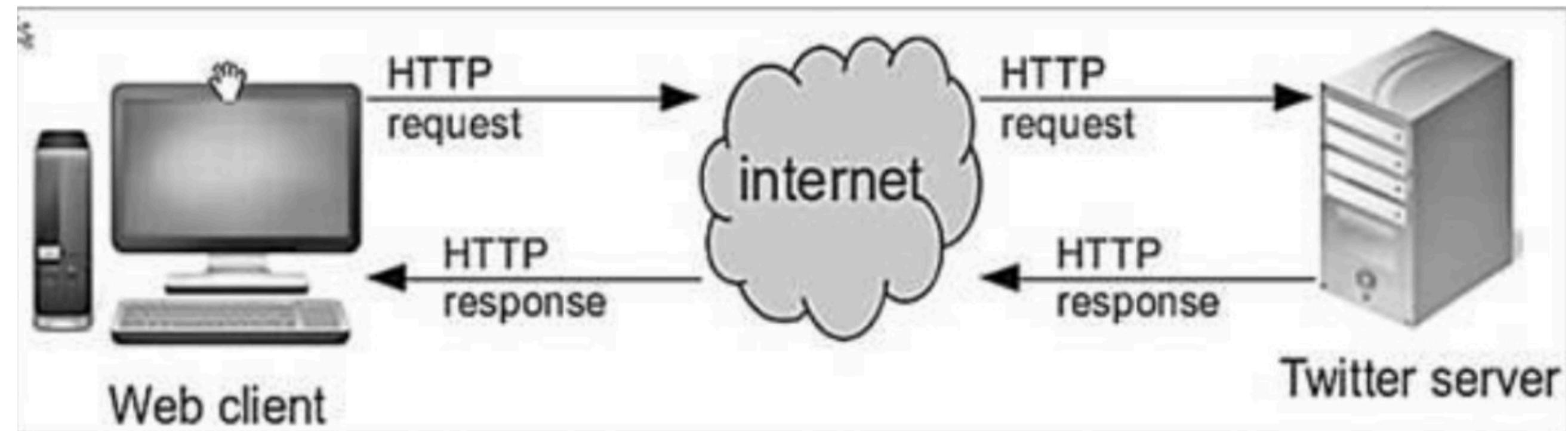


FIGURA 7 ARQUITECTURA DE LAS PRUEBAS APLICADAS POR LOS INVESTIGADORES [3].

# TRANSMISIÓN DE DATOS CON JSON

BIG DATA



x

```
1 {
2 "firstname":"John",
3 "lastname":"Smith"
4 }

5
6 message Person{
7     required string name = 1;
8     required int32 id = 2;
9     optional string email = 3;
10    enum PhoneType {
11        MOBILE = 0;
12        HOME = 1;
13        WORK = 2;
14    }
15    message PhoneNumber{
16        required string number =
17        optional PhoneType type =
18        repeated PhoneNumber
19    }
20    phone = 4;
21
22 [DataContract]
23 class Person
24 {
25     [DataMember]
26     internal string name;
27
28     [DataMember]
29     internal int age;
30 }
31
32 Person p = new Person();
33 //Set up Person object
34 MemoryStream stream1 = new MemoryStream();
35 DataContractJsonSerializer ser = new
DataContractJsonSerializer(typeof(Person))
;
36 ser.WriteObject(stream1, p);
37
38 {"age":42,"name":"John"}
39
40 Person p2 =
(Person)ser.ReadObject(stream1);
41
42
```

J codificacion de nombre y apellido de un objeto de una persona

Objeto personal serializado por medio de JSON

clase de objeto que se va a serializar

Proceso de serializacion

Resultado de ejecutar el código

Proceso de deserializacion del objeto



## CONCLUSIONES

Podemos observar la importancia del almacenamiento de datos de modo persistente en los programas JAVA. También puede usarse para obtener entrada del usuario usando la consola, aunque en la actualidad es mucho más común emplear interfaces gráficas para interactuar con el usuario.

También podemos añadir que hoy en día la serialización no tiene mucha importancia en distintas aplicaciones así que podemos concluir que hay que mejorar la capa de serializacion porque es vital ya que si se mejora en el rendimiento de serializar o deserializar, con un menor volumen de datos a transmitir, la ganancia final de velocidad puede ser la diferencia entre que la aplicación sea viable o no.

# REFERENCIAS

1

IBM Documentation. (2022, junio 27). Ibm.com.  
<https://www.ibm.com/docs/es/wxs/8.6.0?topic=overview-serialization>

2

KathleenDollard. (s/f).  
Serialización - Visual Basic. Microsoft.com.  
Recuperado el 13 de marzo de 2024, de  
<https://learn.microsoft.com/es-es/dotnet/visual-basic/programming-guide/concepts/serialization/>

3

Mora-Castillo, J. A. (2016).  
Serialización/deserialización de objetos y transmisión de datos con JSON: una revisión de la literatura. Revista Tecnología en Marcha, 29(1), 118.  
<https://doi.org/10.18845/tm.v29i1.2544>

