



## Ciclo 3

# Semana 4

*Arquitectura de software parte 2 y metodología de desarrollo Scrum parte 2.*

---

**Lectura 1** - Arquitectura de Software: sistemas distribuidos



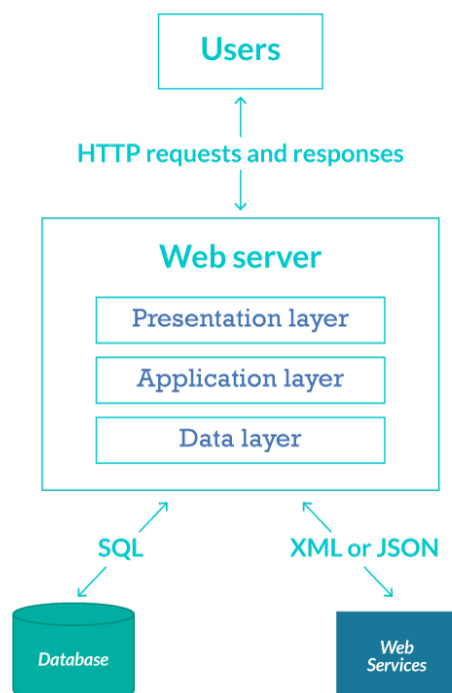
## | Arquitectura de Software: sistemas distribuidos

El concepto sistema distribuido, corresponde a una colección de equipos, o sistemas lógicos independientes, que se muestra al usuario como un sistema único y coherente, también hace referencia al procesamiento de información que se distribuye en varios ordenadores en lugar de limitarse a una sola máquina.

Para la interacción entre sistemas o unidades funcionales, se encuentran dos tipos:

- **Interacción Procedimental:** Un sistema hace un llamado a un servicio conocido ofrecido por otro sistema, y espera una respuesta.
- **Interacción basada en Mensajes:** Un sistema envía información acerca de lo que se requiere de otro sistema. No se requiere esperar una respuesta.

Para el caso nuestro, el tipo de interacción que vamos a utilizar será el basado en mensajes, el cual genera un proceso, el cual se ilustrará mediante un ejemplo: Un sistema (capa de lógica de negocio) se comunica con otro sistema (capa de datos) para recuperar la lista completa de usuarios del sistema.



Semana 4

Arquitectura de software parte 2 y metodología de desarrollo Scrum parte 2.

- Un componente (o capa) que genera un mensaje con los servicios de información requeridos hacia otro componente.
- Este componente utiliza un sistema intermediario, que, a su vez, envía al componente receptor.
- El receptor analiza el mensaje y crea un mensaje de respuesta para el componente emisor.
- No es necesario que emisor y receptor estén pendientes del otro. Solo se comunican con el middleware

Existen tres tipos de sistema de paso mensajes, como los siguientes:

**XML: (eXtensible Markup Language).** El lenguaje de marcado extensible (XML) se utiliza para describir datos, de una forma flexible para diseñar formatos de información y compartir electrónicamente datos estructurados a través de la Internet pública, así como de las redes corporativas. Es una recomendación formal del *World Wide Web Consortium* (W3C), es similar al Lenguaje de Marcado de Hipertexto (HTML), por cuanto contienen símbolos de marcado para describir el contenido de las páginas o archivos.

Este es un ejemplo de la notación:

```
<catalogo>
  <libro>
    <autor>Raúl González Duque</autor>
    <titulo>Python para todos</titulo>
    <genero>Computación</genero>
  </libro>
</catalogo>
```

**JSON: (JavaScript Object Notation).** Es un formato para intercambio de información de estándar abierto que utiliza una estructura de texto para codificar la información en forma de dupla o par, que se tipifica de la siguiente manera: “clave” : “valor”, y delimitado por corchetes – similar a Java o JavaScript. Es ampliamente usado para intercambio de información entre servicios Web y API REST. Un ejemplo es este:

Semana 4

Arquitectura de software parte 2 y metodología de desarrollo Scrum parte 2.

```
{
  libro: {
    autor: Raúl González Duque,
    titulo: Python para todos,
    genero: Computación
  }
}
```

**YAML: (Ain't Another Markup Language).** Es otro formato para el intercambio de información que tiene como objetivo facilitar el mapeo de estructuras de datos más complejas (como listas y arreglos asociativos) en un documento de texto plano legible para un ser humano. Por sus características tiene amplia utilización en la Web, junto con XML y JSON.

```
---
libro:
  autor: Raúl González Duque
  titulo: Python para todos
  genero: Computación
```

Como se ha explicado en semanas anteriores, para el desarrollo del proyecto, se utilizará el formato JSON para paso de mensajes entre capas, el cual es el que se produce como resultado de la aplicación en SpringBoot de @RestController para la generación de las APIs.