Java Web API

Repaso General

Octavio Robleto

octavio.robleto@gmail.com

https://orobleto.github.io/octaviorobleto.github.io/

https://github.com/orobleto/CursoJavaWebAPI

https://www.youtube.com/user/octaviorobleto20

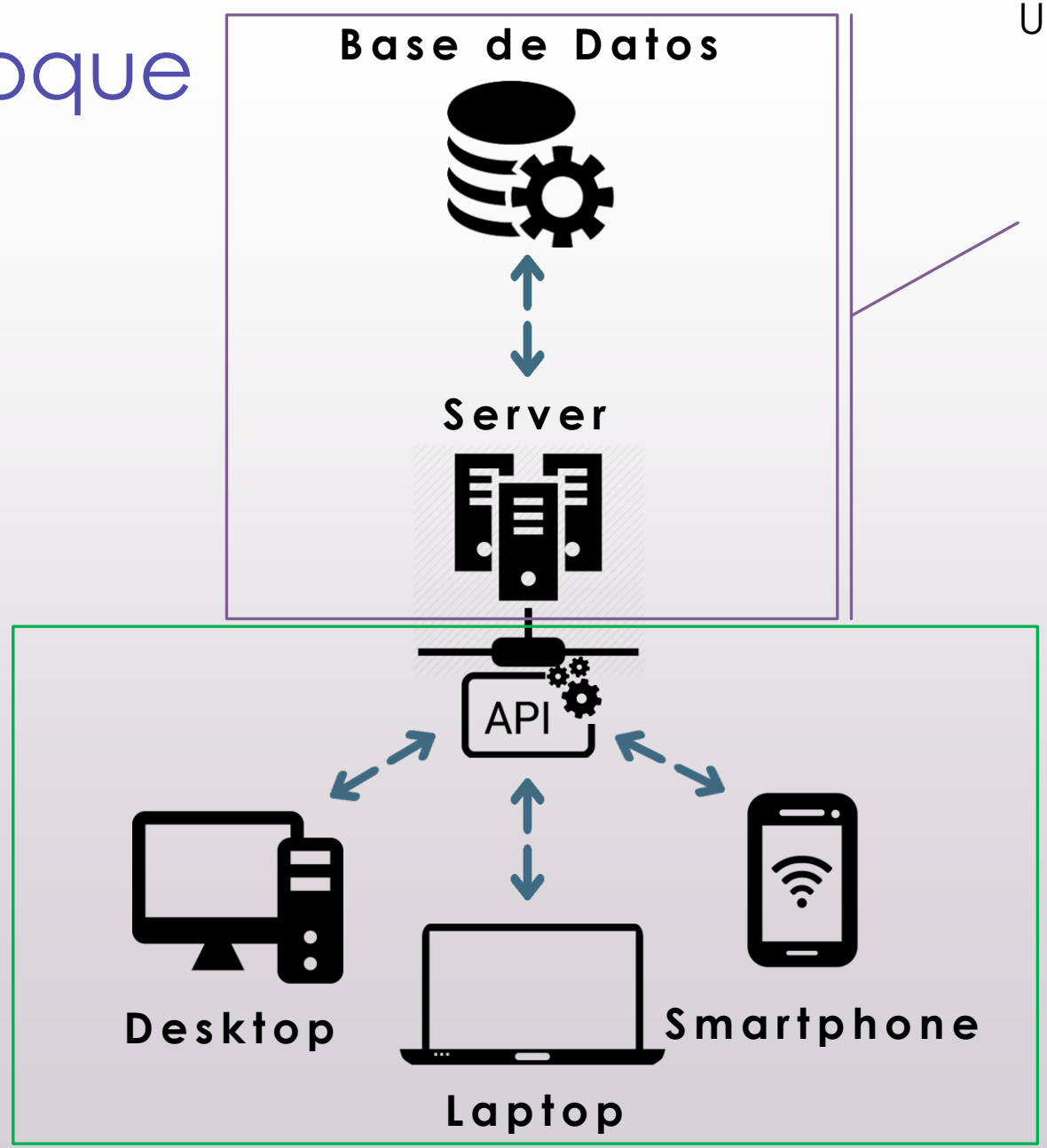
9 11 50124479

Protocolo HTTP

- Es un protocolo **cliente-servidor**, lo que significa que el cliente envía una petición al servidor y espera un mensaje de respuesta del servidor. Es un protocolo **sin estado**, lo que significa que el servidor no guarda información del cliente, cada petición es independiente de las demás.

Java
Ing. Octavio Robledo

Enfoque

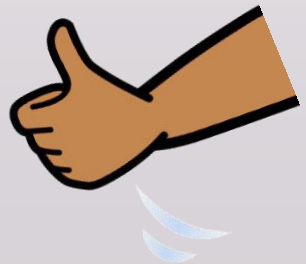


Una aplicación servidor es el elemento de la comunicación que responde a las peticiones de los clientes, proporcionando el servicio requerido

- Una aplicación cliente es el elemento de la comunicación que pide o solicita un servicio de red.

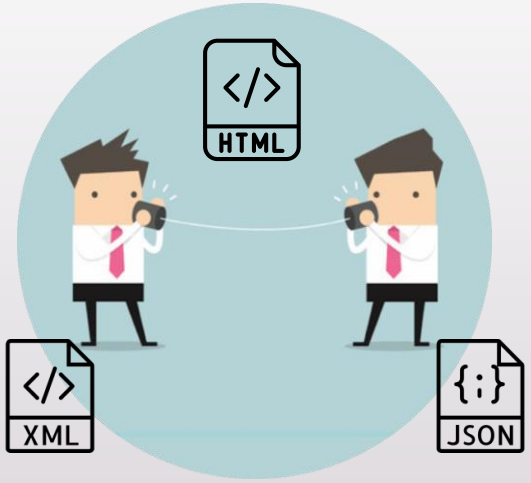
Cliente - Servidor

- **Escalabilidad :** Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden agregar nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento :** Al estar distribuidas las funciones y responsabilidades entre varias computadoras independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por el cambio. Hoy día es frecuente también tener servidores en la nube
- **Seguridad:** Permite administrar permisos de forma mas localizada.



Verbos HTTP (Request)

- Los verbos definen la acción que se quiere realizar sobre el recurso. Los verbos más comunes son:
  - **GET:** Solicitar un recurso.
  - **POST:** Publicar un recurso.
  - **PUT:** Reemplazar un recurso.
  - **DELETE:** Eliminar un recurso.



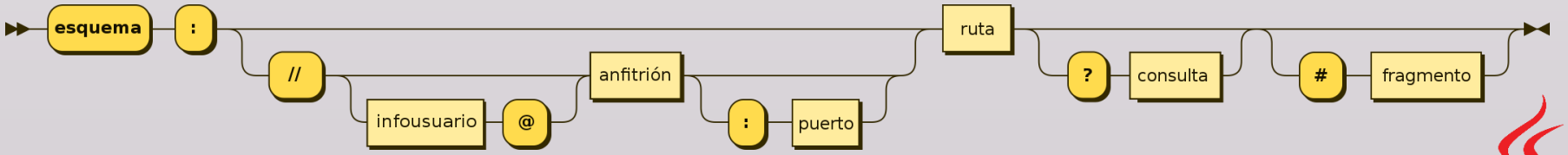
- Cuando ingresamos a una página o le damos clic a un vinculo en HTML que nos redirecciona se esta enviando por defecto el método GET.

Verbos HTTP (Response)

- Los códigos de respuesta se dividen en cinco categorías dependiendo del dígito con el que inician:
  - **1XX**: Información
  - **2XX**: Éxito
  - **3XX**: Redirección
  - **4XX**: Error en el cliente
  - **5XX**: Error en el servidor
- Los mas comunes que normalmente vemos:
  - **404**: Recurso No encontrado.
  - **500**: Error en el Servidor

URL (Uniform Resource Locator)

- **Esquema** El esquema define el protocolo a utilizar por ejemplo http:, https:, mailto:, ftp:, etc.
- **Host** La IP o el nombre del servidor (Dominio) que se quiere acceder.
- **Puerto** El puerto en el que está escuchando el servidor HTTP. Si se omite se asume que es el 80.
- **Ruta** La ruta al recurso que se quiere acceder.
- **Consulta** Contiene información adicional para el servidor en forma de propiedades (atributo=valor). Las propiedades se separan por &.
- **Fragmento** La referencia a una ubicación interna del documento.



Paginas y Sitios WEB

- Una página web es parte de un sitio web y es un único archivo con un nombre de archivo asignado, mientras que un sitio web es un conjunto de archivos llamados páginas web.
- Las páginas pueden contener textos, enlaces, contenido multimedia en general, etc.

Pueden ser:

- **Estáticas**: Básicamente informativas y están enfocadas principalmente a mostrar una información permanente, rara vez cambian.
- **Dinámicas**: Son aquellas en las que la información presentada se genera a partir de una petición del usuario.

Java
Ing. Octavio Robledo

Server-Side Processing

- Para la creación de este tipo de páginas se deben utilizar etiquetas HTML y algún lenguaje de programación que se ejecute **“del lado del servidor”.**
- Los lenguajes utilizados para la generación de este tipo de páginas son principalmente: Perl, PHP, JSP y ASP.

JSP (Java Server Pages)

- Es una tecnología creada para dar repuesta al auge de las páginas Web dinámicas que permite crear sitios Web con código Java embebido con una sintaxis en particular **<% %>.**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Login</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
    <script src='main.js'></script>
</head>
<body>

<% aqui va el codigo JAVA %>

</body>
</html>
```

Servlets

- Los servlets Java son clases Java diseñadas para responder a solicitudes HTTP en el contexto de una aplicación web.

```
public class NombreClase extends HttpServlet {

    //Metodos

}
```

Como manejar Verbos HTTP

- Los métodos en los que delega el método service las peticiones HTTP, incluyen:
  - **GET**: Solicitar un recurso → doGet.
  - **POST**: Publicar un recurso → doPost.
  - **PUT**: Reemplazar un recurso → doPut.
  - **DELETE**: Eliminar un recurso → doDelete.

Parámetros que poseen los Métodos

```
protected void verboHTTP(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

}
```

- Un objeto HttpServletRequest proporciona acceso a los datos de cabecera HTTP
- Un objeto HttpServletResponse proporciona formas de devolver datos al usuario

Client-Side Processing

- JavaScript es un lenguaje interpretado (No necesita compilación como JAVA) que se ejecuta del lado del cliente (Navegador).
- De esta forma, podemos enviar documentos a través de la Web que incorporan el código fuente de un programa, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

Que necesitamos

- https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

- https://www.eclipse.org/downloads/packages/release/2019-09/r/eclipse-ide-enterprise-java-developers

- https://tomcat.apache.org/download-90.cgi

Java EE

- Proporciona una plataforma para construir aplicaciones transaccionales, seguras, interoperables y distribuidas con los siguientes servicios
- Básicos
  - Creación de páginas web con Servlets y Java Server Pages (JSP)
  - Acceso a bases de datos relacionales con Java Database Conectivity (JDBC)
  - Sistema de mapeo objeto-relacional con Java Persistence API (JPA)
- Avanzados
  - Gestión de componentes con Enterprise Java Beans (EJB)
  - Interfaz de usuario con Java Server Faces (JSFs)
  - Gestionar transacciones con Java Transaction API (JTA)
  - Envío y recepción de mensajes con Java Message Service (JMS)

IDE

- Es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica en algunos casos.

eclipse

Java

Servidores

- Una aplicación Java EE se ejecuta dentro de un contenedor (container).

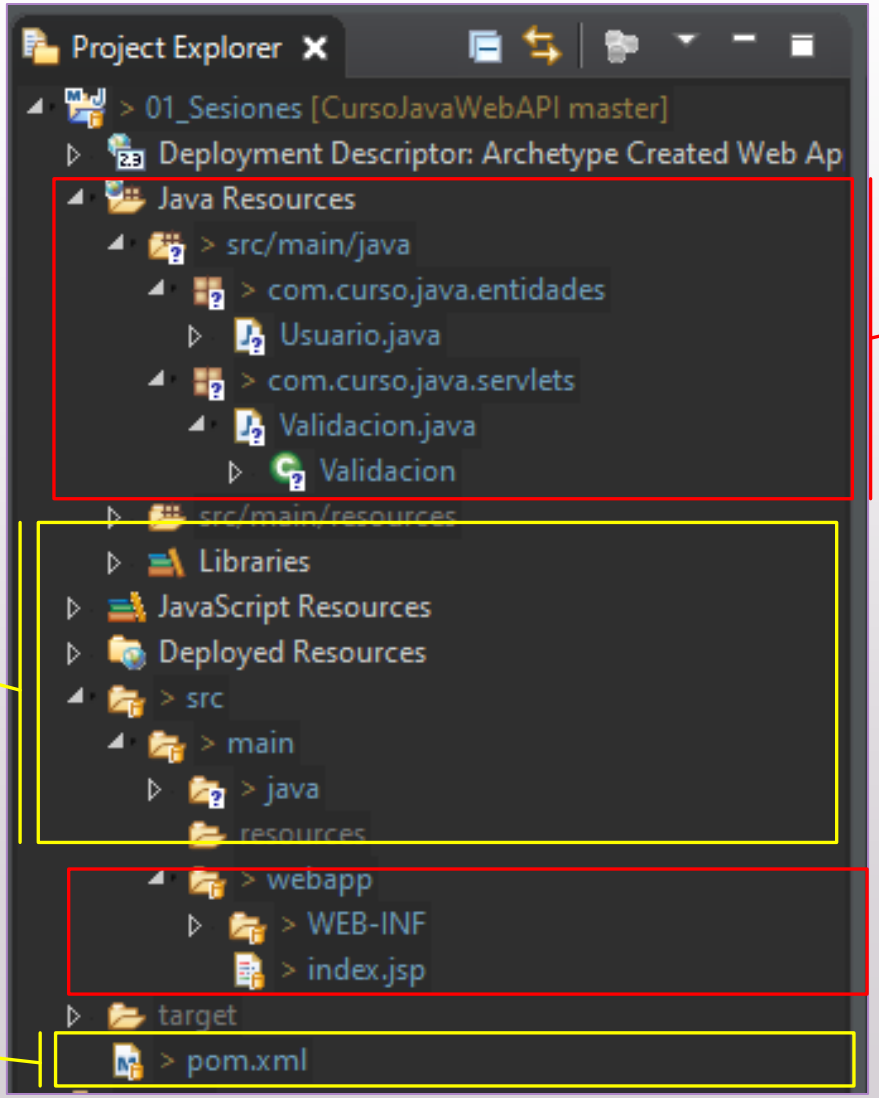
- Existen dos tipos de servidores:
  - **Servidores de aplicaciones:** Contienen todos los servicios definidos en el estándar Java EE
  - **Servidores web:** Contienen los servicios esenciales de Java EE (servlets y JSP)

Apache Tomcat

Maven

- Gestor de dependencias (librerías) y sus versiones
- Construcción de proyectos (de código a entregable .war, .ear, .zip, etc.)
- Estructura única de proyecto compatible con todos los entornos de desarrollo y sistemas de integración continua

Proyecto



Recursos JAVA

Otros Recursos

Recursos WEB

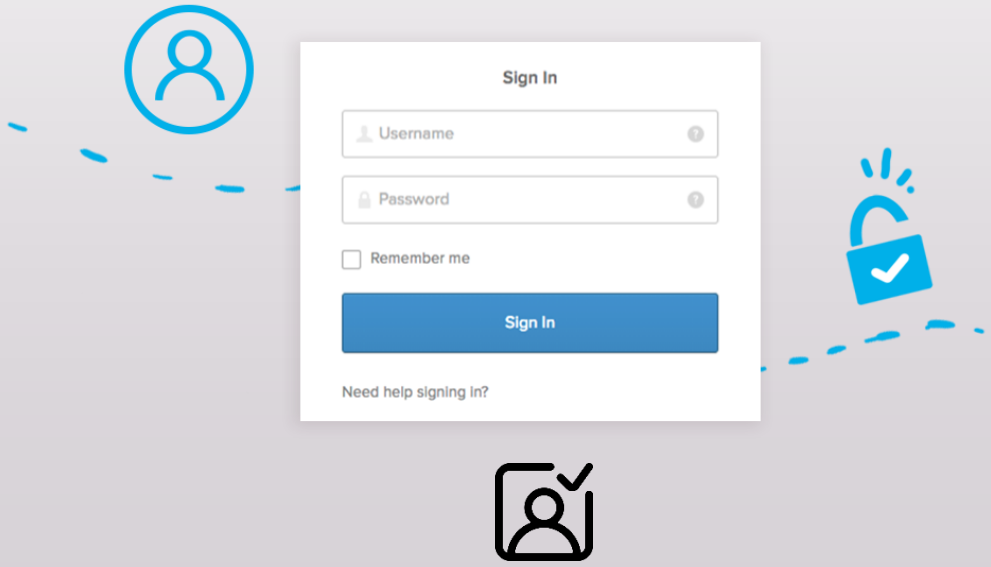
Archivo Configuración XML

Java

Ing. Otávio Roblero

Sesiones

- Una sesión es una serie de comunicaciones entre un cliente y un servidor en la que se realiza un intercambio de información. Por medio de una sesión se puede hacer un seguimiento de un usuario a través de nuestra aplicación.



Java Session

- Para manejar dicha información Java nos proporciona el objeto HttpSession, este objeto posee una estructura HashMap por cada objeto creado para guardar la información necesaria de dicha sesión.

```
protected void verboHTTP(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession session = request.getSession();
}
```