

Arquitectura web



Arquitectura Web

- Definición de Arquitectura
- Cliente - Servidor - HTTP
- API REST
- Consumo de Servicios mediante AJAX

Terminología


- Una **interfaz** es una capa de abstracción para que dos sistemas se comuniquen.
- Una **arquitectura de Sw** es como está diseñado un sistema, funciones de sus componentes y como se comunican.
- Un **servicio WEB** es un sistema que permite la comunicación entre equipos que están en internet y usan protocolo HTTP, y es la base de las API remotas.



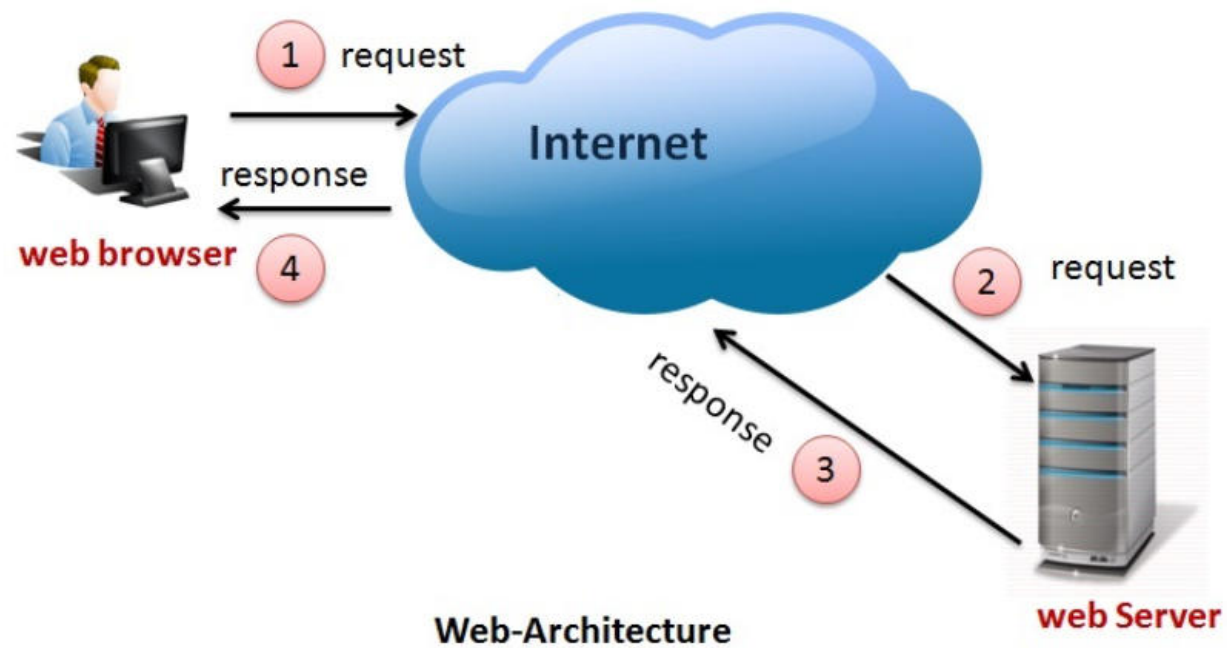
¿Qué es la Arquitectura de Software?

La arquitectura de software representa la **estructura o estructuras** del sistema que consiste en **componentes de software**, las propiedades **externas** visibles de esos componentes y las **relaciones** entre ellos.

Un **componente** es un **pieza de software** que puede ser código fuente, código binario, un ejecutable, o una biblioteca con una **interfaz definida**.



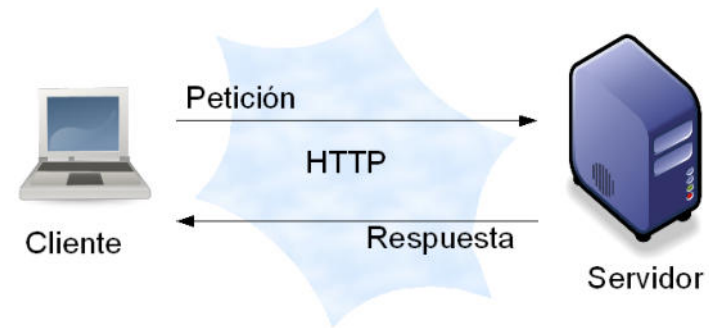
Arquitectura Web - Resumen



Patrón Cliente - Servidor

Participan 2 **componentes**:

- Un **servidor** que provee uno o más servicios a través de una **interfaz**.
- Un **cliente** que usa esos servicios como parte de su operación.
en el acceso al servidor.



Protocolo HTTP V1.1

- Protocolo de Transferencia de Hipertexto
- Capa de Aplicación (Capa 7 - Modelo OSI)
- Sincrónico – Sin Estado
- Cliente / Servidor
 - Solicitud / Respuesta
- Puerto 80 / 443 (HTTPS)
- Métodos Principales
 - GET
 - POST
 - PUT
 - DELETE
- RFC-216 (<https://tools.ietf.org/html/rfc2616>)



¿Qué es una API?

- Una **interfaz de programación de aplicaciones** (API) es un conjunto de **herramientas, definiciones y protocolos** que se usan para diseñar e **integrar aplicaciones**.
- Permite que un producto o servicio se comuniquen con otros productos y servicios, sin la necesidad de saber cómo se **implementan internamente**.
- Se ejecutan siempre en el servidor y algunos lenguajes de programación utilizados son JAVA, Ruby, PHP, etc

REST – Tipos de APIs

- Pueden ser:
 - LOCALES (privadas)
 - REMOTAS (privada/públicas)
 - Gratuitas
 - Autenticadas
 - Pagas
- Las REMOTAS utilizan servicios web para lo cual pueden usar arquitecturas
 - SOAP
 - REST - RESTFUL

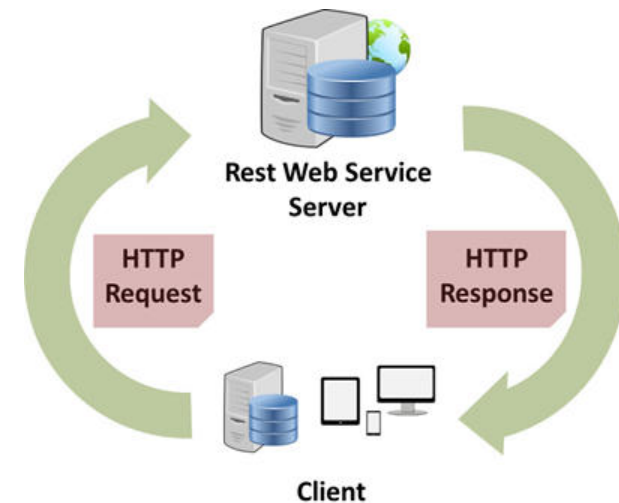
¿Qué es REST?

REST o Representational State Transfer

Es una arquitectura de representación de transferencia de estados en la **comunicación entre cliente y servidor.**

Permite :

- ✓ Guardar datos en cache
- ✓ Guardar el estado en cada petición y no entre peticiones.
- ✓ Definir que datos puede otra aplicación acceder y manipular mediante la definición permisos



REST – formatos de intercambio

- Se **apoya** sobre el protocolo **HTTP**.
- Todos **los objetos se manipulan mediante URI** (identificador uniforme/Único de recursos). Permite identificar directamente un recurso.
- Utiliza como formato de intercambio **JSON** / XML
 - JSON es un ***formato*** ligero de intercambio de datos.
 - Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo.




API REST - Ejemplo

Ingresar a <https://reqres.in/>

GET	LIST USERS
GET	SINGLE USER
GET	SINGLE USER NOT FOUND
GET	LIST <RESOURCE>
GET	SINGLE <RESOURCE>
GET	SINGLE <RESOURCE> NOT FOUND
POST	CREATE
PUT	UPDATE
PATCH	UPDATE
DELETE	DELETE

En resumen...

- La comunicación se inicia desde el **cliente al servidor**.
 - Los clientes pueden establecer **sesiones** en el servidor que mantendrá el **estado** de los clientes conectados.
 - Se deben proveer **mecanismos** a los clientes para **ubicar** a los servidores, **gestionar errores** y proveer **seguridad** en el acceso al servidor.
- 

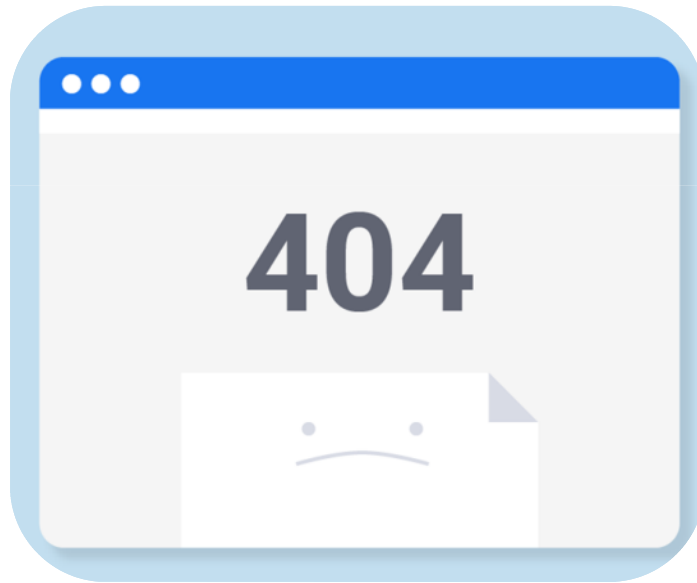
RESPUESTAS DEL SERVIDOR A UNA PETICIÓN

REST - Peticiones

Crea una solicitud HTTP que contiene toda la información necesaria, es decir, **un REQUEST a un servidor tiene toda la información necesaria y solo espera una RESPONSE**, ósea una respuesta en concreto.



Protocolo HTTP - Códigos de Estado



¿Lo Conocen?

Protocolo HTTP - Códigos de Estado

1xx: Respuestas informativas. Indica que la petición ha sido recibida y se está procesando.

2xx: Respuestas correctas. Indica que la petición ha sido procesada correctamente.

3xx: Respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.

4xx: Errores causados por el cliente. Indica que ha habido un error en el procesamiento de la petición a causa de que el cliente ha hecho algo mal.


5xx: Errores causados por el servidor. Indica que ha habido un error en el procesamiento de la petición a causa de un fallo en el servidor.



REST - Métodos

Los métodos nos permiten interactuar con las API's

Los métodos básicos de HTTP, como son:

- **Post:** Para crear recursos nuevos. (ej. enviar nueva información)
 - **Get:** Para obtener un listado o un recurso en concreto.
 - **Put:** Para modificar un recurso (ej. Actualizar información).
 - **Patch:** Para modificar un recurso (en forma parcial).
 - **Delete:** Para borrar un recurso.
- 

¿Qué es AJAX?

AJAX = Asynchronous JavaScript And XML.

- **Permite** que los **sitios web** se **actualicen** de forma **asincrónica** mediante el **intercambio** de **datos** con un **servidor web detrás** de **escena**.
- Es posible **actualizar partes** de un **sitio web**, **sin recargar** todo el **sitio**.

jQuery - AJAX

El método ajax realiza un pedido http y recupera la información del servidor.

\$.ajax() puede ser usado para enviar requests GET, POST, PUT, DELETE, etc.

La sintaxis básica para realizar un ajax en jQuery es:

```
$.ajax(url, [options])  
$.ajax({settings})
```

```
$.ajax({  
  method: "POST",  
  url: "some.php",  
  data: { name: "John", location: "Boston" }  
})  
.done(function( msg ) {  
  alert( "Data Saved: " + msg );  
});
```

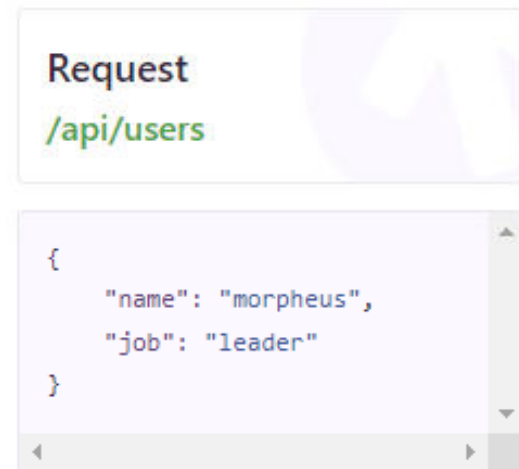
jQuery - Ajax

Crear un documento html que permite ingresar dos campos: nombre y trabajo.

Crear un botón el cual ejecute el AJAX que envíe los datos mediante un POST.

- a) Recuperar la respuesta y mostrarla por consola.
- b) Recuperar la respuesta y mostrarla en elementos html.

Utilizar: <https://reqres.in/>



Referencias

- [Modelo en Capas - Martin Fowler](#)
- [AJAX - JQuery](#)
- [API RESTful](#)

¿Preguntas?



Gracias!

