
ARQUITECTURA WEB

Autor: Luciano Straccia

Versión 2023.01



Agenda

- Cliente-Servidor
- Front End
- Back End
- Patrones de interacción (MVC y otros)
- Cliente liviano y cliente pesado
- CDN (Content Delivery Network)

Cliente - Servidor

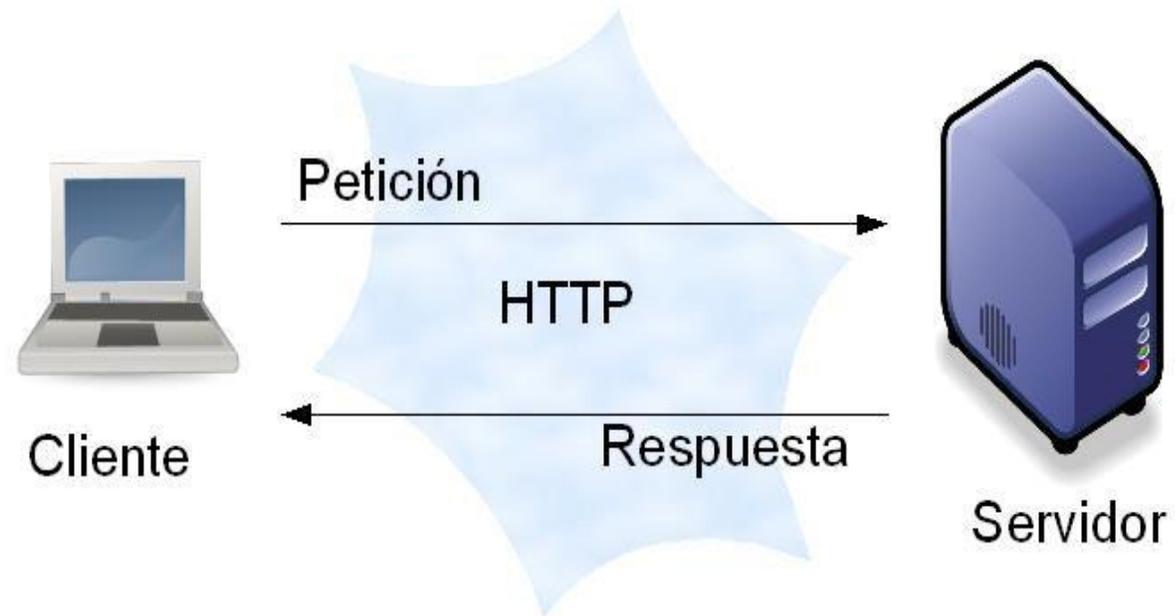


Arquitectura Cliente-Servidor

- Separación del cliente (consumidor) y el servidor (prestador del servicio)
- Componentes:
 - Servidores: ofrecen servicios a otros subsistemas
 - Clientes: puede haber varias instancias ejecutándose concurrentemente
 - Red: para acceder servicios
- Características:
 - Clientes conocen servidores y servicios disponibles
 - Servidores no necesitan conocer identidad de clientes
 - Clientes acceden a servicios a través de llamadas



Arquitectura Cliente-Servidor





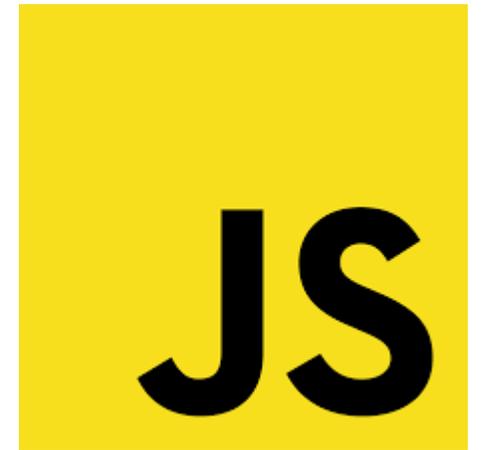
Front End y Back End

- El front-end es aquello que ven los usuarios e incluye elementos visuales, como botones, casillas de verificación, gráficos y mensajes de texto. Permite a los usuarios interactuar con la aplicación.
- El back-end son los datos y la infraestructura que permiten que la aplicación funcione. Almacena y procesa los datos de las aplicaciones para los usuarios.

Front End



Front End





Front End

- Lenguaje de marcado (HTML – XML)
- XSLT (Extensible Stylesheet Language Transformations): XML a HTML
- CSS (Cascading Style Sheets)
- DOM (Documento Object Model)
- Bootstrap: front-end toolkit (CSS + JS)



W3C

- W3C: World Wide Web Consortium
- Consorcio internacional con sede en el MIT
- Creado por Tim Berners-Lee en 1994
- Misión:
 - W3C is leading the Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web.
 - Llevar la Web a su máximo potencial mediante el desarrollo de protocolos y directrices que garanticen el crecimiento a largo plazo de la Web.



DOM (Document Object Model)

- El **Modelo de Objetos del Documento (DOM)**, por sus siglas en Inglés) es una interfaz de programación para los documentos HTML y XML.
- El modelo de objeto de documento (DOM) proporciona otras formas de presentar, guardar y manipular este mismo documento.
- El DOM es una representación completamente orientada al objeto de la página web y puede ser modificado con un lenguaje de script (por ejemplo, JavaScript).
- W3C DOM es un estándar para su utilización:
<https://dom.spec.whatwg.org/>

Back End



Back End

- Lado servidor
- Lenguajes de programación
- Persistencia
- Servicios



Aspectos a considerar

- **Concurrencia:** en el front end cada usuario tiene su propia copia de una aplicación, por lo que no tiene que gestionar los problemas de concurrencia. El back end utiliza varias estrategias para gestionar miles de solicitudes de usuarios al mismo tiempo.
- **Seguridad:** en el front end especialmente orientada a la validación de datos y la transferencia hacia el back end. En el back end la seguridad es más compleja.

Patrones de Interacción



Patrones de Interacción

- Los patrones de interacción pueden ser considerados un subgrupo de los patrones de arquitectura
- Buscan determinar cuáles componentes son totalmente internos del software y cuáles poseen interacción con los usuarios
- Nos cuestionan cómo realizar la interacción entre front end y back end



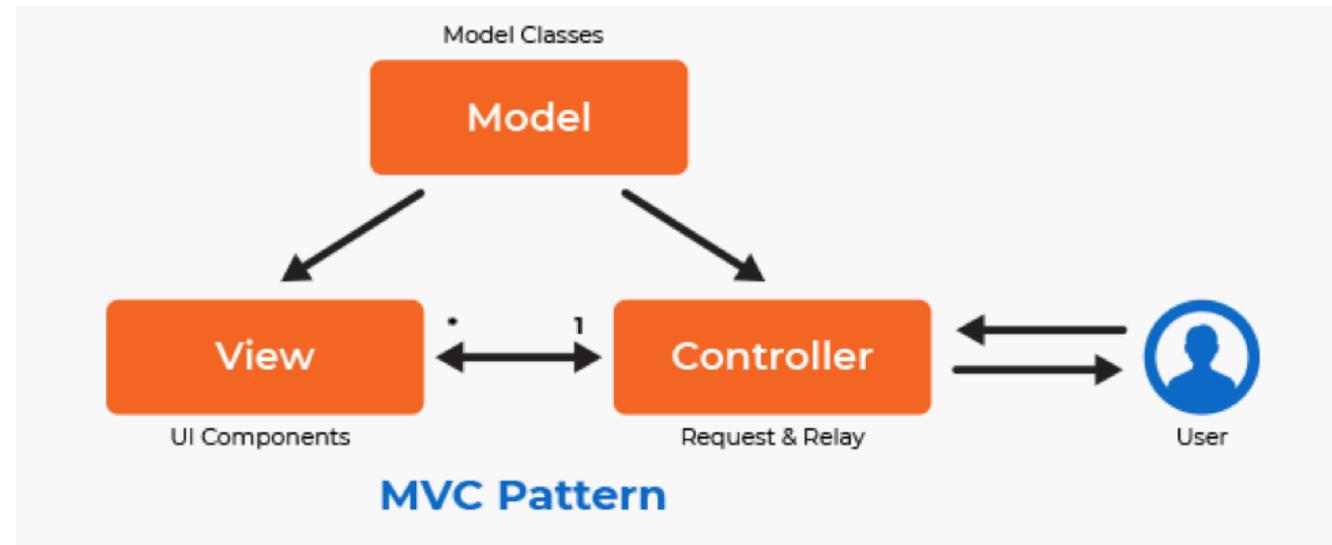
Patrones de Interacción

- Basados en arquitecturas de capas
- Deben responder a:
 - ¿Cómo se relacionan las capas definidas?
 - ¿Todas pueden comunicarse entre sí o hay orden y limitaciones?
 - ¿Se relacionan entre sí directamente o requieren “mediadores”?



Patrón de Interacción: MVC

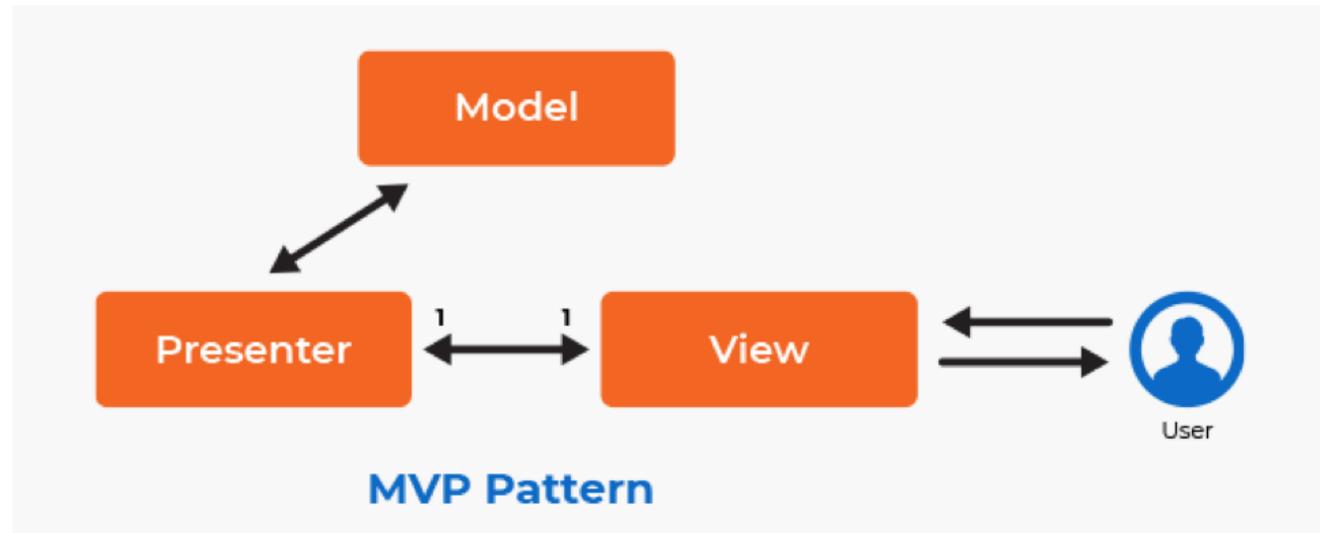
- Resuelve la interacción con el usuario
- No resuelve la interacción interna (separación de lógica y persistencia)





Patrón de Interacción: VMP / MVP

- Reduce la carga sobre la Vista



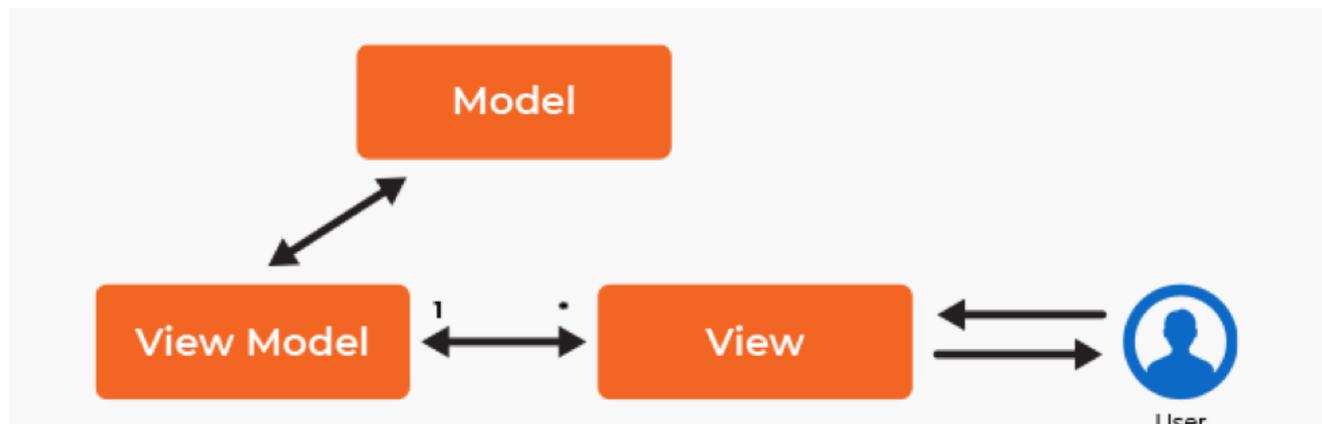


Patrón de Interacción: VMP / MVP

- El Patrón Modelo-Vista-Presentador (MVP) surge para ayudar a realizar pruebas automáticas de la interfaz gráfica, para ello la idea es codificar la interfaz de usuario lo más simple posible, teniendo el menor código posible, de forma que no merezca la pena probarla. En su lugar, toda la lógica de la interfaz de usuario, se hace en una clase separada (que se conoce como Presentador), que no dependa en absoluto de los componentes de la interfaz gráfica y que, por tanto, es más fácil de realizar pruebas



Patrón de Interacción: MVVM





Patrón de Interacción: MVVM

- MVVM significa Modelo Vista VistaModelo, porque en este patrón de diseño se separan los datos de la aplicación, la interfaz de usuario pero en vez de controlar manualmente los cambios en la vista o en los datos, estos se actualizan directamente cuando sucede un cambio en ellos, por ejemplo si la vista actualiza un dato que está presentando se actualiza el modelo automáticamente y viceversa.
- Las principales diferencias entre MVC y MVVM son que en MVVM el “Controller” cambia a “ViewModel” y hay un “binder” que sincroniza la información en vez de hacerlo un controlador “Controller” como sucede en MVC.
- El MVVM puede ser entendido como una correlación con el patrón de diseño Observer

Cliente liviano y cliente pesado



Cliente pesado

- Thick client o Fat client
- El procesamiento se realiza localmente en el cliente
- Menor exigencia al servidor y a la transferencia
- Requiere clientes con mayor capacidad de procesamiento y recursos locales



Cliente pesado

- El cliente recibe del servidor los datos
- El cliente renderiza y genera el HTML para exponerlo al usuario en el navegador web



Cliente pesado





Cliente liviano

- Thin client
- El procesamiento se realiza principalmente en el servidor
- Mayor exigencia al servidor y a la transferencia
- Permite clientes con menos capacidad de procesamiento



Cliente liviano

- El servidor procesa los datos
- El servidor renderiza para generar el HTML y servirlo ya generado el cliente
- El cliente expone el HTML en el navegador web



Cliente liviano

handlebars



CDN (Content Delivery Network)



CDN (Content Delivery Network)

- Red de entrega de contenido
- Grupo de servidores distribuidos geográficamente
- Buscan reducir el tráfico y mejorar el tiempo de respuesta
- Utilizan almacenamiento en caché y son útiles para contenido estático
- Proveedores CDN: por ejemplo, Cloudflare

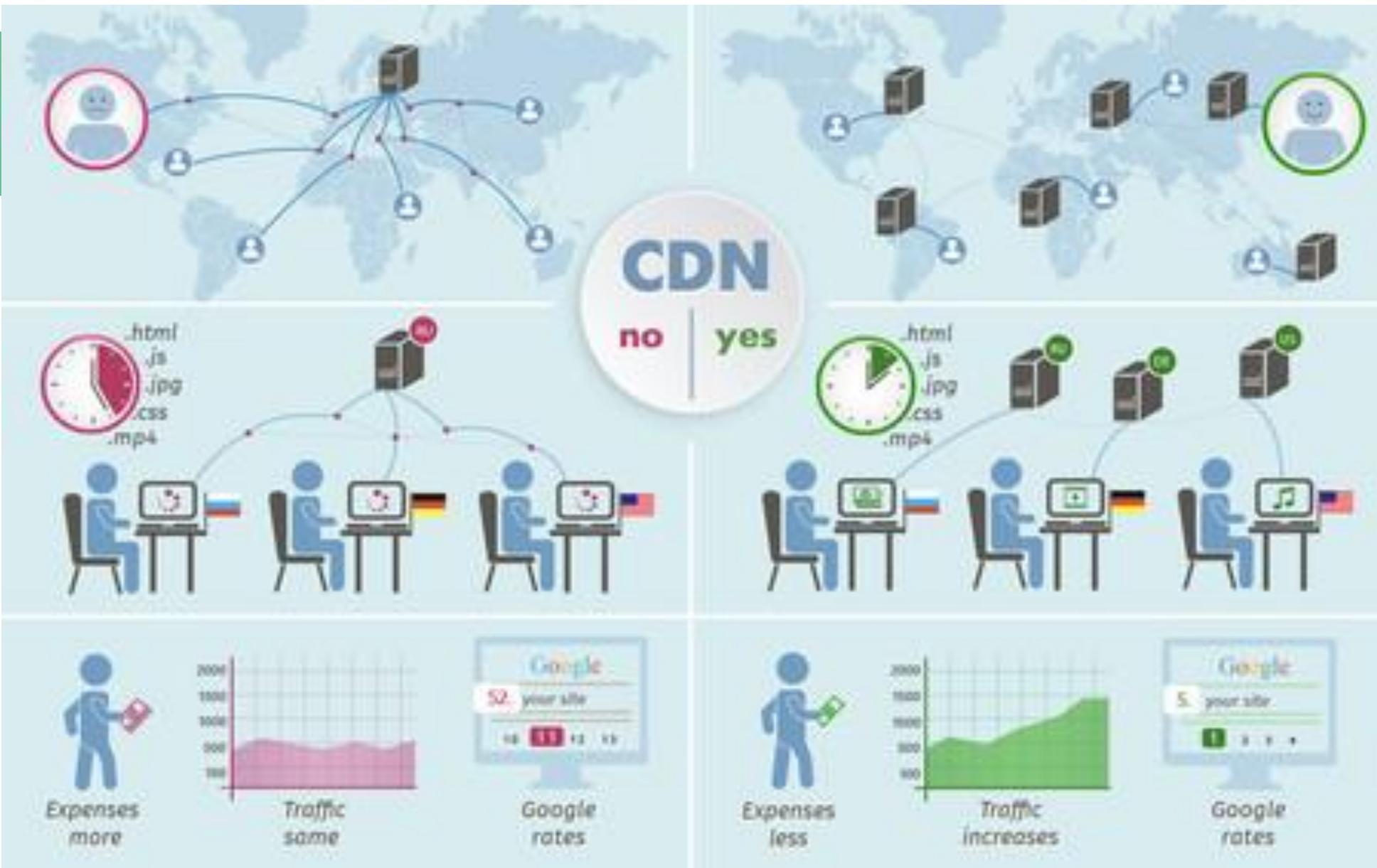


Imagen tomada del sitio de Cloudflare



Variantes Cliente - Servidor

- Cliente liviano (Thin Client) o Cliente pesado (Fat Client)
- Stateless (sin estado) o Stateful (con estado)

ARQUITECTURA WEB

Autor: Luciano Straccia
