

Taller de Sistemas Empresariales

Aplicaciones Web Modernas



Instituto de
Computación

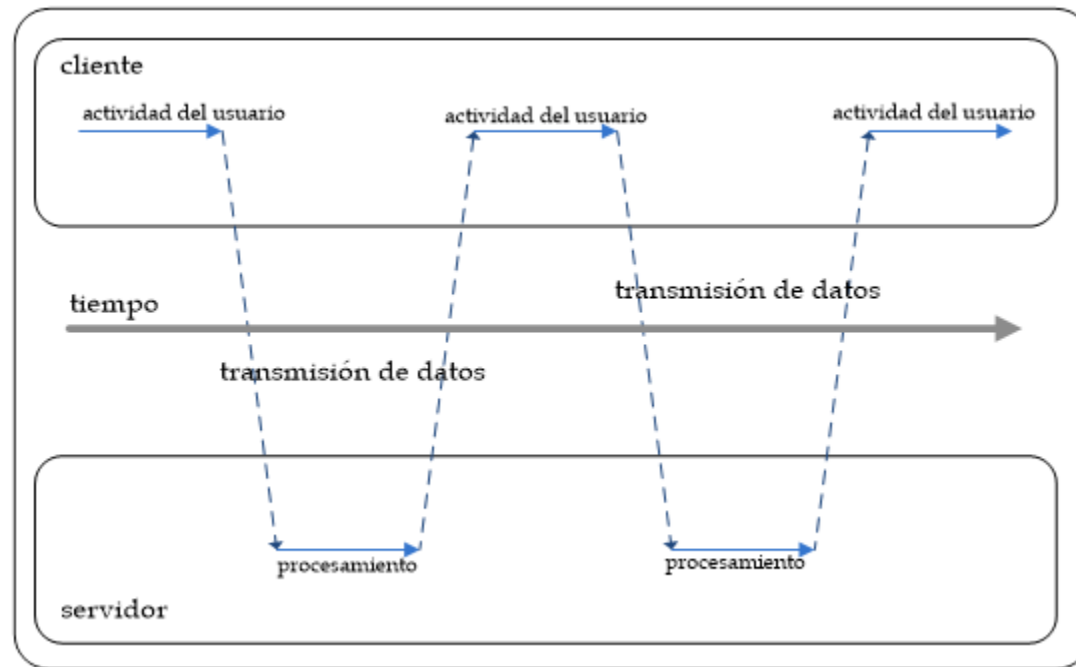


Facultad de
Ingeniería

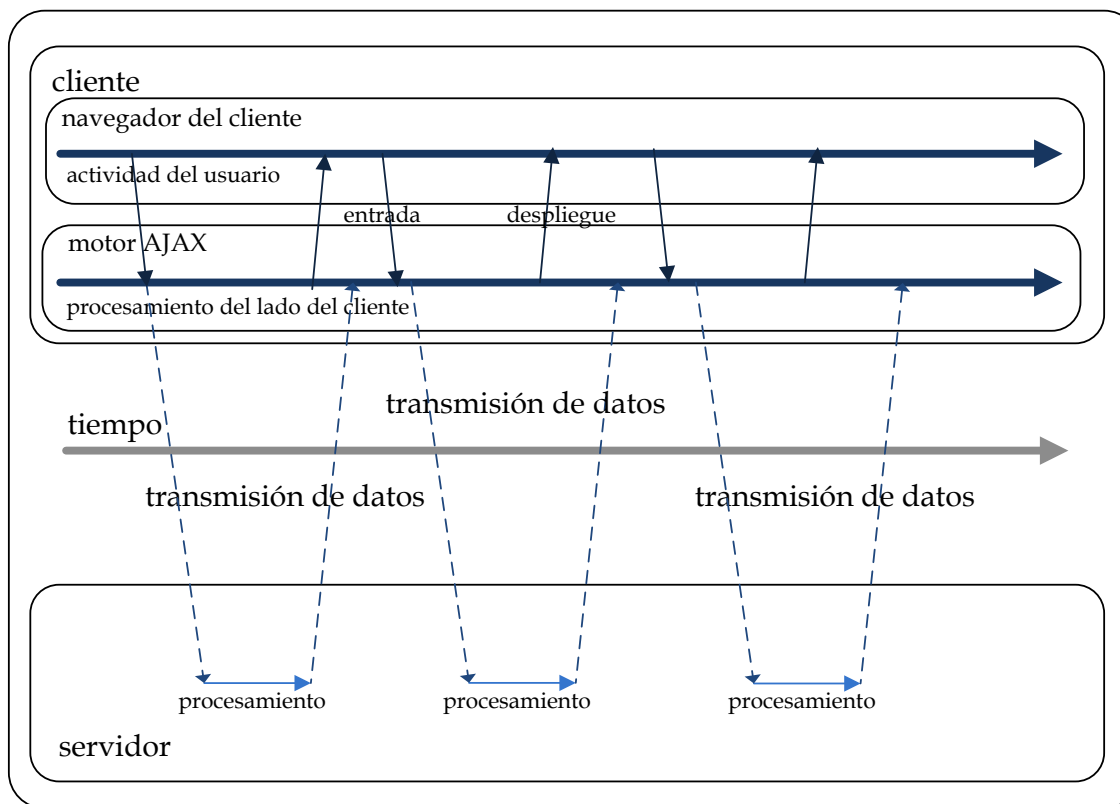


Universidad de la
República de Uruguay

- ❑ Asynchronous Javascript And XML
- ❑ Enfoque para el desarrollo web que utilizan scripts del lado del cliente para intercambiar datos con el servidor
- ❑ Pedidos asincrónicos
 - ❑ Interfaz fluída
- ❑ Más que un patrón, es una combinación de tecnologías utilizadas de una manera creativa

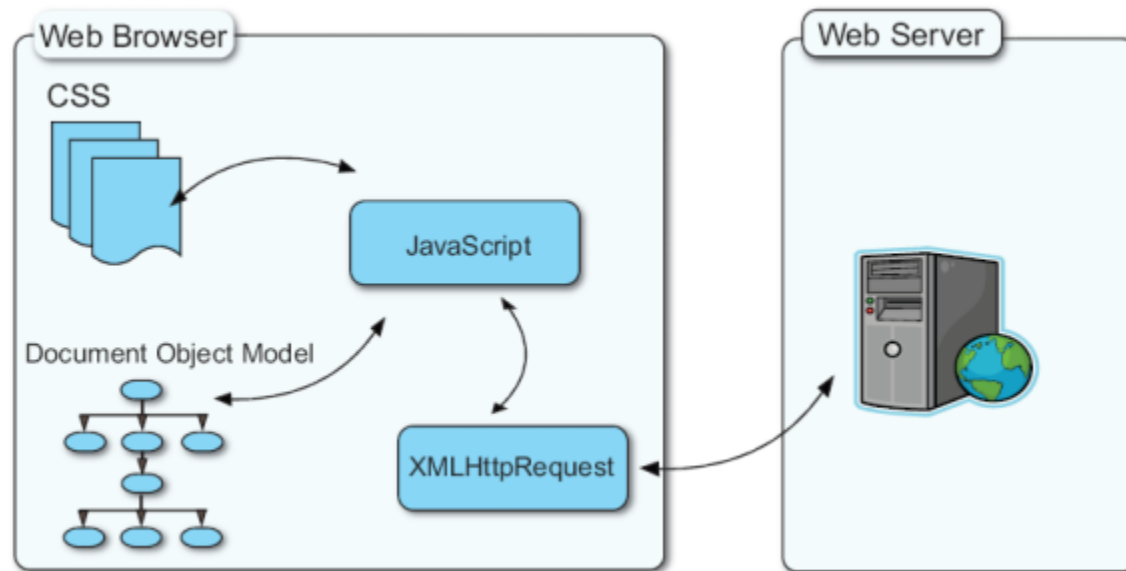


Aplicación web tradicional



Aplicación web AJAX

- ❑ Interacción entre 4 grandes tecnologías
 - ❑ JavaScript
 - ❑ CSS (Cascading Style Sheets)
 - ❑ DOM (Document Object Model)
 - ❑ Objeto XMLHttpRequest

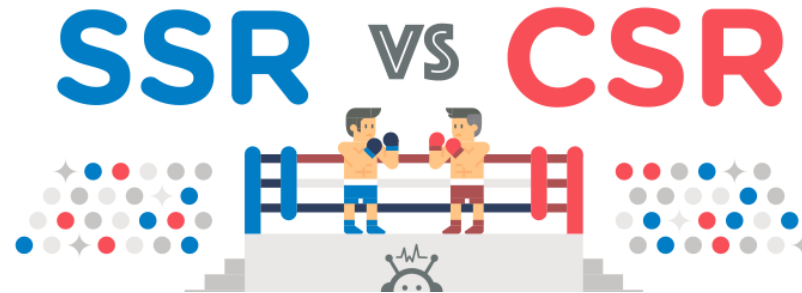


Interacción entre las 4 grandes tecnologías involucradas

- ❑ Representación más sencilla para los datos que XML
- ❑ Utilizada principalmente para el intercambio de datos entre el cliente y el servidor
- ❑ Representado como una pareja nombre:valor
- ❑ Al ser notación JavaScript, se maneja nativamente en el cliente

```
{ "NombrePropiedad1" : valor1, "NombrePropiedad2" : valor2 }
```

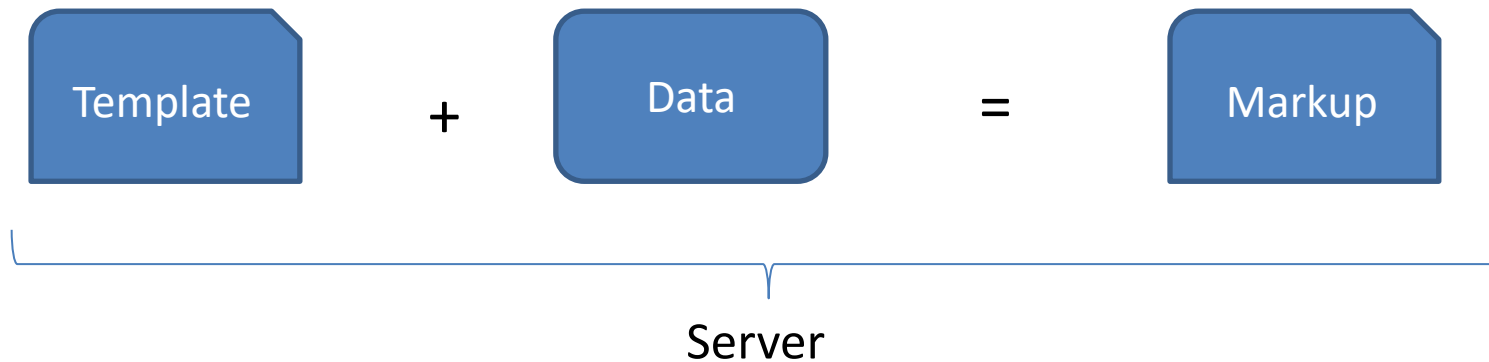
Rendering – 2 enfoques distintos



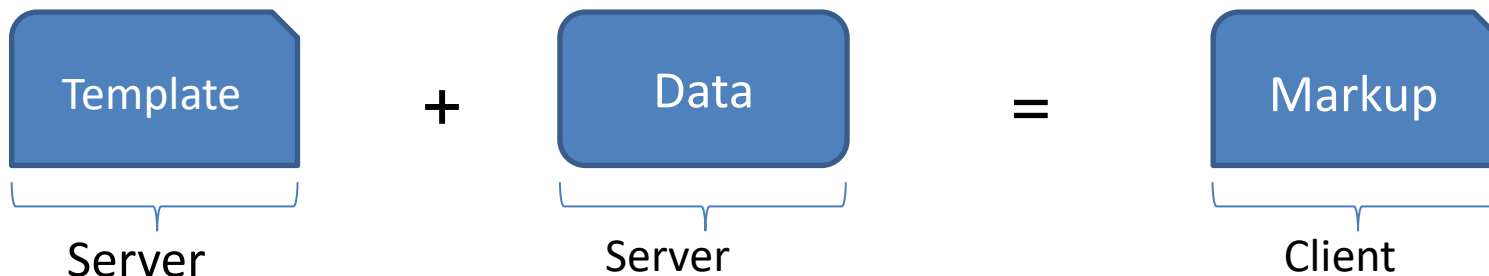
- ❑ Server-side rendering
 - El contenido final a mostrarse en el cliente (navegador) es generado en el servidor
 - Se pueden tener actualizaciones “parciales” manejando estado de la vista en el servidor (view)
- ❑ Client-side rendering
 - El contenido obtenido del servidor es parcial y por medio de la ejecución de javascript en el cliente y llamadas AJAX se genera el contenido final

Rendering – 2 enfoques distintos

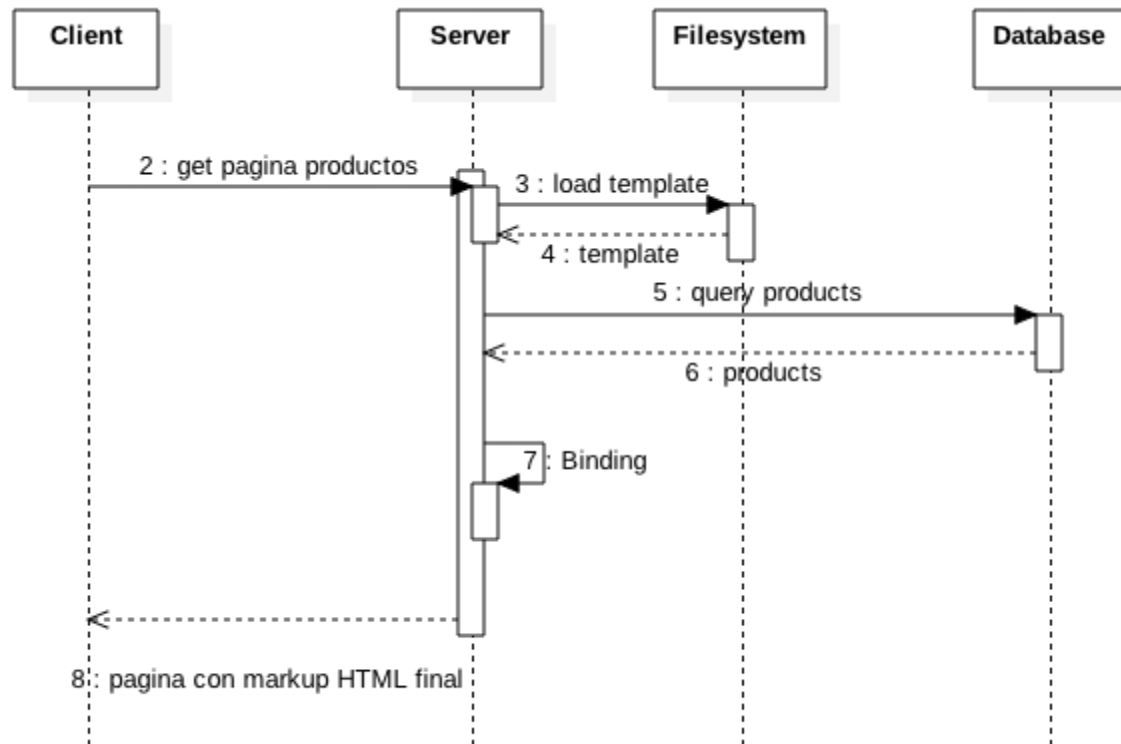
❑ Server Side Rendering



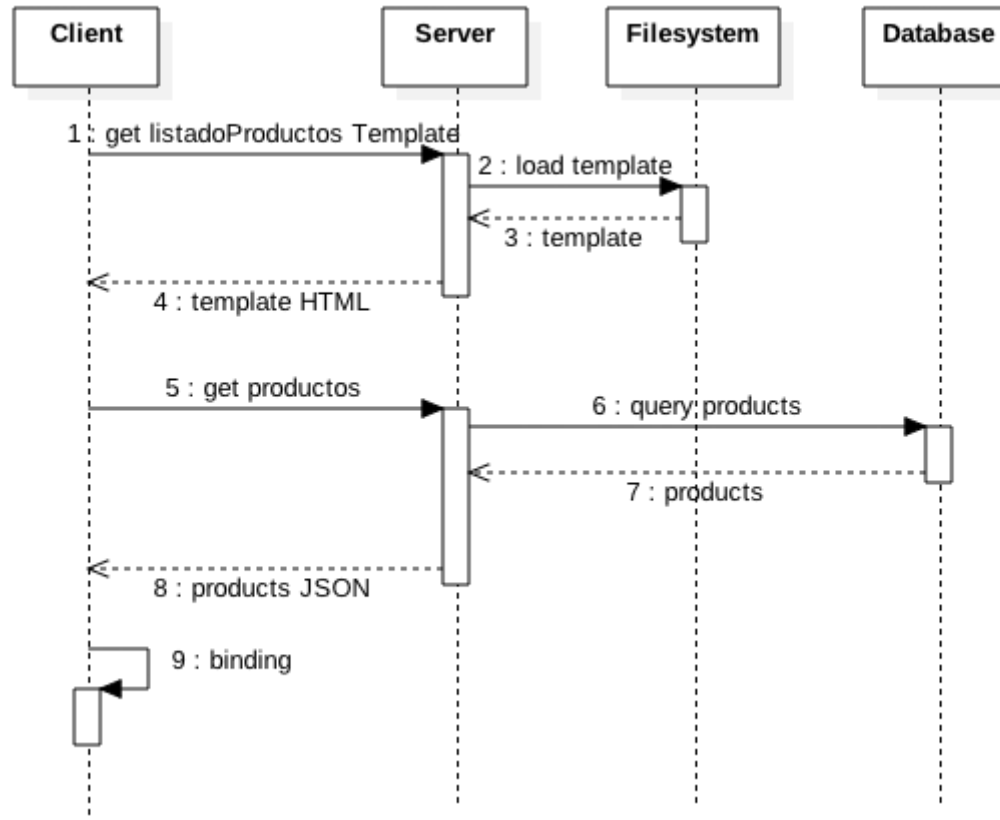
❑ Client Side Rendering



SSR – Listado de Productos



CSR – Listado de Productos



Client Side Rendering

- ❑ Requiere menos procesamiento del lado del servidor, distribuyendo la carga a los clientes
- ❑ Contenido con mayor capacidad de “caching”
- ❑ Ejemplos: Angular.js, EmberJS, ReactJS, BackboneJS

Server Side Rendering

- ❑ Al servir el contenido final, mejora la compatibilidad con indexadores de buscadores (SEO)
- ❑ Para actualizaciones parciales se requiere de manejo de estado de la vista en el servidor
- ❑ Ejemplos: JSP, JSF

Single Page Applications (SPA)

- ❑ Aplicación web de una sola página
 - Simula una aplicación de escritorio en la web
 - Mejora experiencia de usuario y usabilidad
 - Basadas fuertemente en html, css, javascript y ajax
 - Carga de todos los recursos una sola vez
 - Carga dinámica de recursos según necesidades (lazy loading)
 - Frameworks y herramientas facilitan el desarrollo
 - Surgen de una evolución natural...

Single Page Applications (SPA)

- ❑ Páginas HTML estáticas
- ❑ Programas CGI (contenido dinámico!)
- ❑ ASP clásico, PHP (frameworks lado servidor!)
- ❑ ASP.NET Web Forms (viewstate)
- ❑ ASP.NET MVC
- ❑ AJAX (XMLHttpRequest, comunicación asíncrona!)
- ❑ Aparecen frameworks y librerías JS (jQuery, Mootools, Prototype, etc.)

Single Page Applications (SPA)

❑ Primeros frameworks JS

- Objetivos de simplificar
 - Manejo del DOM (rico en selectores)
 - Uso de AJAX
 - Abstracción del navegador
- Algunos problemas
 - Spaghetti Code!
 - Difícil de mantener
 - Difícil de testear
 - Cambios en UI afectan comportamiento

Single Page Applications (SPA)

□ Frameworks JS modernos

- Algunas características
 - Arquitecturas definidas y mantenibles
 - Basadas en MVC, fácil adopción
 - Simplificación del testing (Inyección de Dependencias)
 - Estrategias de minificación y bundling
 - Estrategias de loading (Mejoran performance)
 - Abstraen la manipulación del DOM
- Angular, Ember, Vue.JS

- ❑ Framework Javascript Open Source
- ❑ Desarrollado por Google
- ❑ Comenzó como un framework tipo SPA
- ❑ Independiente de la tecnología de backend
- ❑ Modelo declarativo, extiende la sintaxis HTML
- ❑ Arquitectura de componentes.
- ❑ Fácilmente testeable (Implementa DI)
- ❑ Sincronización entre la vista y el modelo (bindings y observables)

- ❑ Superset de Javascript (Javascript tipado)
- ❑ Open Source, desarrollado por Microsoft
- ❑ Permite detectar errores en tiempo de compilación
- ❑ “Mejora” la “performance” (el javascript resultado sigue las mejores prácticas)
- ❑ En sus nuevas versiones angular incentiva su uso

- ❑ Biblioteca Javascript Open Source
- ❑ Desarrollada por Facebook
- ❑ Maneja exclusivamente las vistas de la aplicación
- ❑ Gran variedad de librerías disponibles en el ecosistema
- ❑ React Native para el desarrollo de aplicaciones nativas en plataformas móviles
- ❑ Next.JS como framework para SSR

❑ Props

- Son datos inmutables
- Pasan de padre a hijo hasta el final de la cadena

❑ State

- Son datos que se modifican en algún punto
- El estado pasa del padre al hijo
 - Componentes Stateless
 - Componentes Stateful

❑ Virtual DOM

- Cuanto más grande es el DOM, más costoso es actualizarlo
- El Virtual DOM se mantiene en memoria, y mediante un algoritmo optimizado, se cambian los nodos que efectivamente se actualizaron
- React es el encargado de cambiar el DOM original una vez se termina de actualizar el Virtual DOM

❑ JSX

- JavaScript XML
- Extensión de JavaScript, creada por React

```
ReactDOM.render(  
  <h1>Hola, mundo!</h1>,  
  document.getElementById('root')  
) ;
```

Notificaciones en tiempo real

- ❑ Necesidad de notificar eventos de interés en tiempo real
- ❑ Modelo publish/suscribe
- ❑ Permite comunicación bidireccional
- ❑ Evita requests periódicos al servidor
- ❑ Mejora experiencia de usuario
- ❑ Utilizado para mensajería instantánea web (chat)

- ❑ HTTP Long polling / Ajax Push / Reverse Ajax
 - Basados en Ajax
 - Peticiones prolongadas
 - Emulan push real
- ❑ HTTP Server push
 - Comunicación iniciada por el Servidor
 - Diferentes tecnologías
 - Basados en plugins, obsoletos (flash, silverlight, java applets, etc)
 - HTML5 Web Sockets (bidireccional)
 - HTML5 Server-Sent Events (unidireccional)
- ❑ Hidden iframe / forever frame (streaming)

Diseño responsive



Fuente: <http://bradfrost.com/blog/post/this-is-the-web/>

Diseño responsive



Fuente: <http://bradfrost.com/blog/post/this-is-the-web/>

Diseño responsive



Fuente: <http://bradfrost.com/blog/post/this-is-the-web/>

□ Distintos enfoques

- Aplicación diferente (ej. subdominio mobile)
- Misma aplicación, diferentes vistas (según user-agent)
- Misma aplicación y misma vista adaptable
 - Diferentes css
 - Mismo css (CSS3 media queries)

- ❑ Reduce tiempo desarrollo
- ❑ Evita contenidos duplicados
- ❑ Facilita mantenibilidad y cambios
- ❑ Frameworks y herramientas facilitan el desarrollo
 - **Bootstrap, Material**
 - Preprocesadores de CSS: Less - Saas

- ❑ Framework para crear sitios web responsivos
- ❑ Filosofía *Mobile First* a partir de versión 3
- ❑ Compatibilidad cross-browser
- ❑ Componentes de UI comunes
 - Paginación
 - Botones
 - Modales
- ❑ Utiliza containers y un sistema de grillas