



Recuerda marcar tu asistencia



Asistencia en LMS



- equipos
- {
- 1, “millonarios, “futbol”
- },
- {
- 2, cali, “futbol”,
- 3: “ millonarios”, “bal”
- }

Eventos

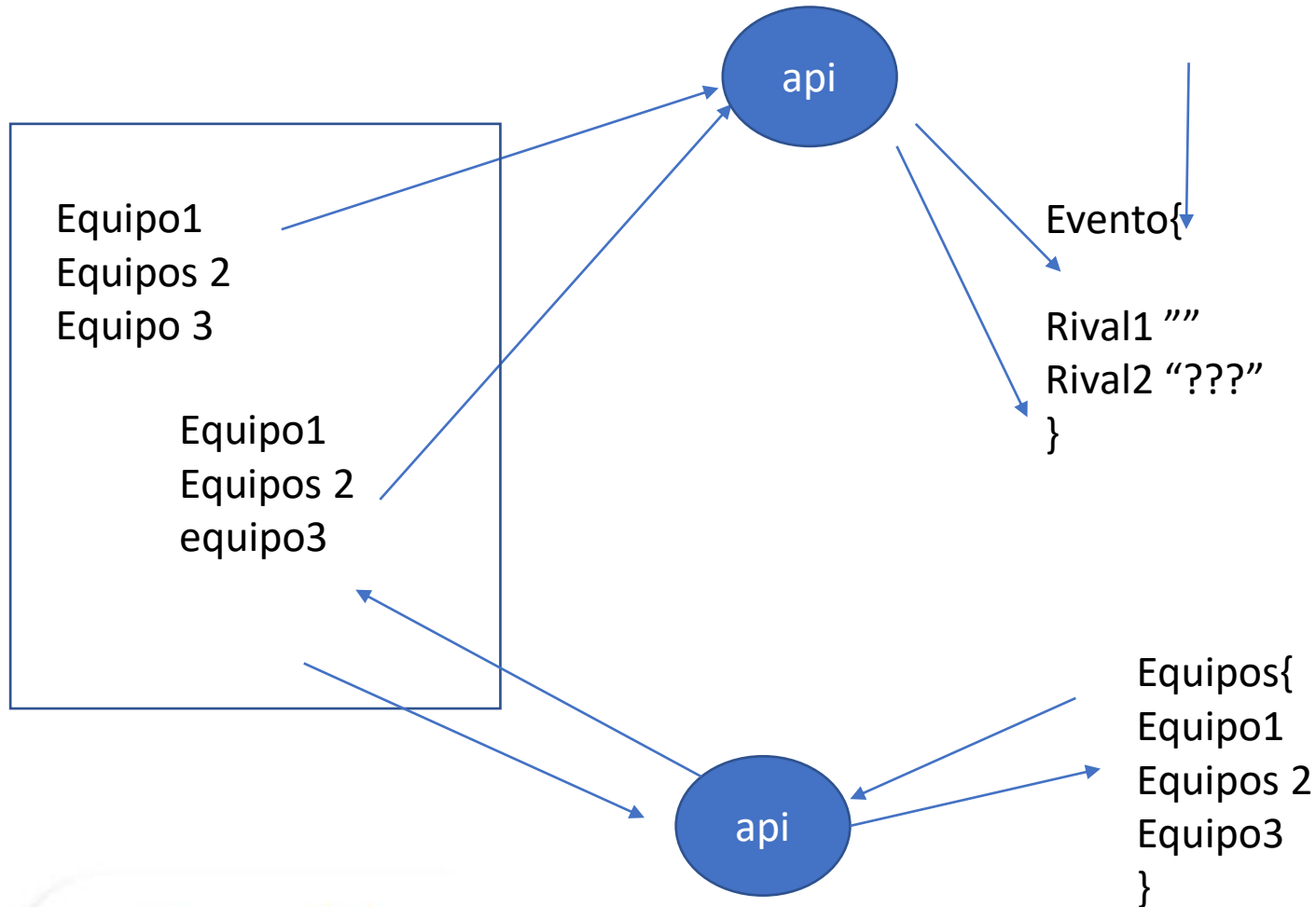
```
{
  rival1[
    name: “millonarios”
  ],
  Rival2: “manchester”
  Rival2[
    {
      name: “manchester”
    }
  ]
  Categoria: “futbol ”
]
[ mil
Cat: “Bal”
]
}
```

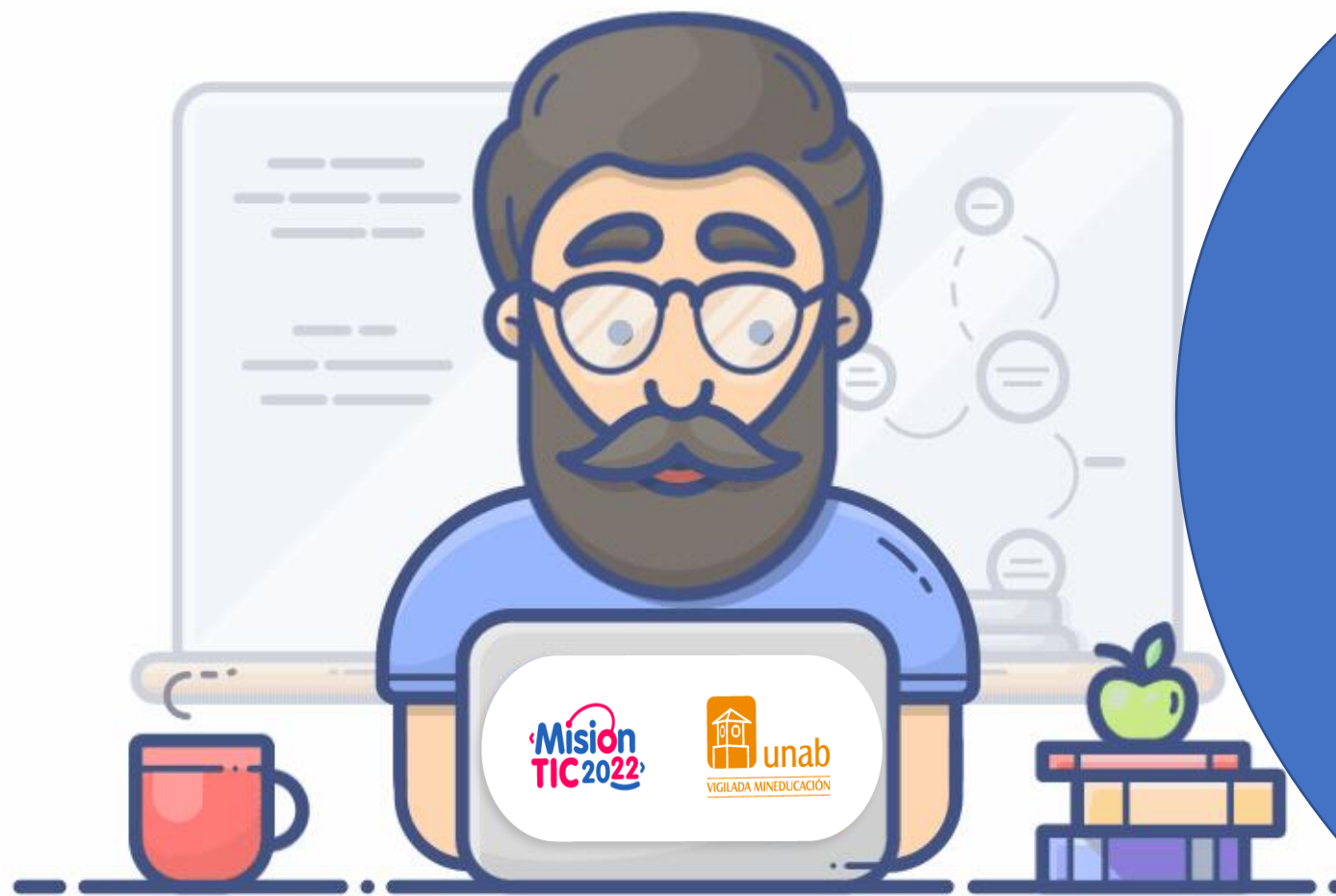
??

- Listado ev donde participó millonarios (“futbol”)
- Actualizar ev

Categorias

```
{
  1, “futbol”
  2, “baloncesto”
}
```





Semana 4 Sesión 01

SPA

Ing Edwin Garcia



El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN

Misión
TIC 2022

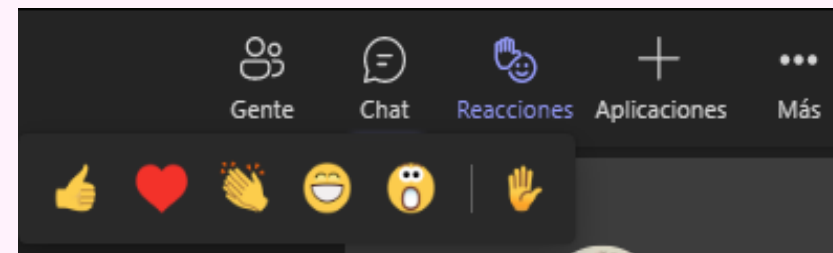
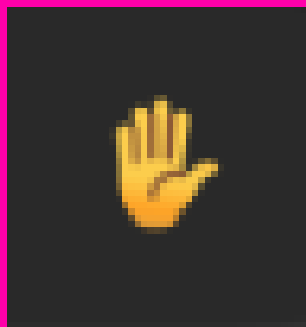
unab
VIGILADA MINEDUCACIÓN



Recuerden marcar su asistencia

Asistencia en LMS

**Pedir la
palabra para
realizar una
pregunta o
una
intervención**





Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

Mision
TIC 2022



Hechos
QUE CONECTAN ✓



SPA

Single Page Application



Una web SPA (Single Page Application) es una página web la cual está todo el contenido en una sola página, es decir, carga tan solo un archivo HTML y todo se produce dentro de este único archivo. De esta manera se puede ofrecer una experiencia más fluida, más rápida.

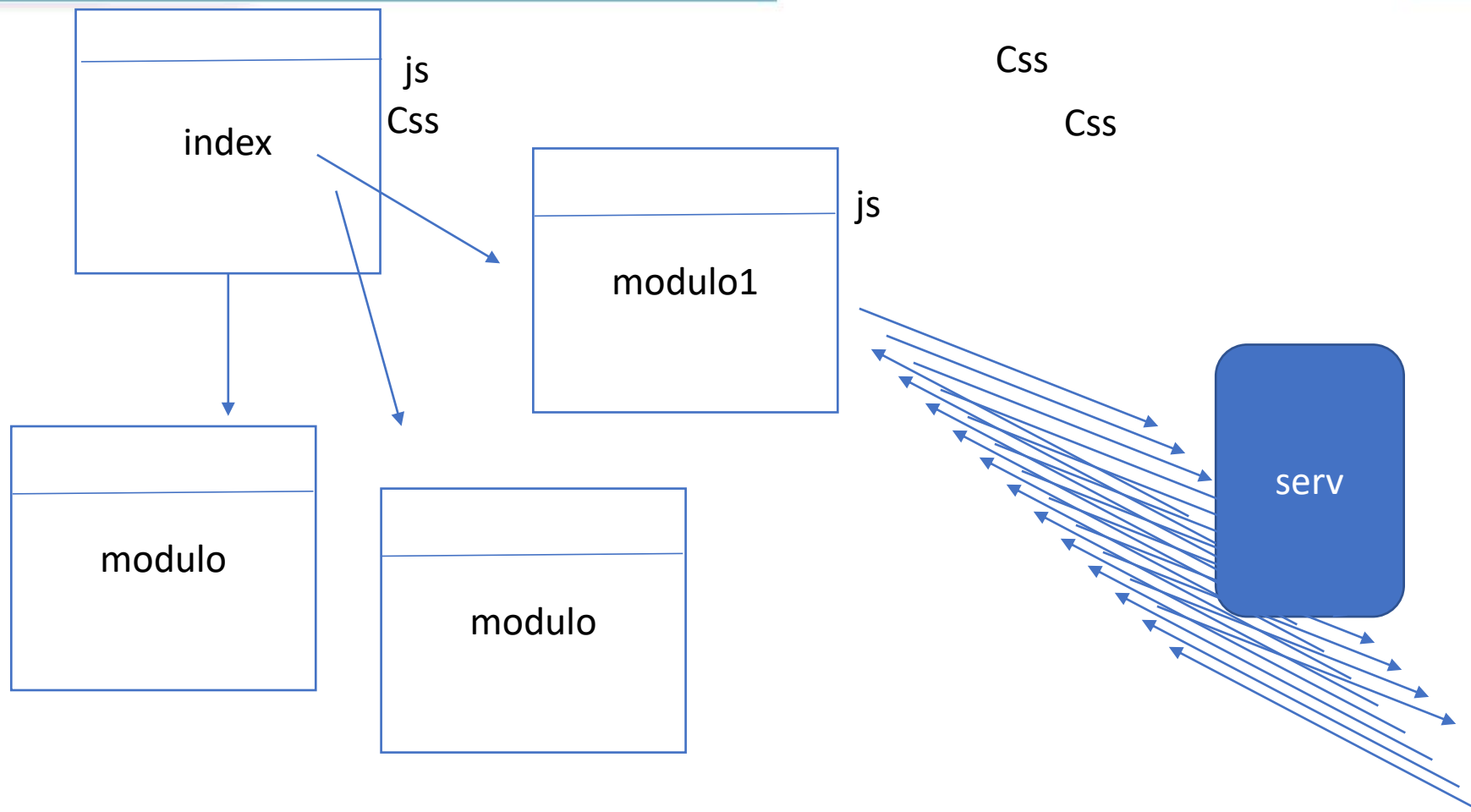


Es decir, todo el código HTML, JavaScript y CSS se carga una sola vez, puedes navegar entre los diferentes apartados de la página y el contenido ya estará precargado de antemano, es por esto que es tan rápido. En algunos casos también lo que se hace es cargar estos contenidos dinámicamente si se requieren, pero no tiene que cargar toda la página de nuevo, tan solo los nuevos contenidos.



En una SPA tenemos varias vistas, no varias páginas

Aunque solo tenemos una página, sí que podemos tener varias vistas, es decir los diferentes apartados que podría tener nuestra web. A ojos del usuario es como si tuviese varias páginas que cargan muy rápido, pero en realidad son vistas en la misma página.



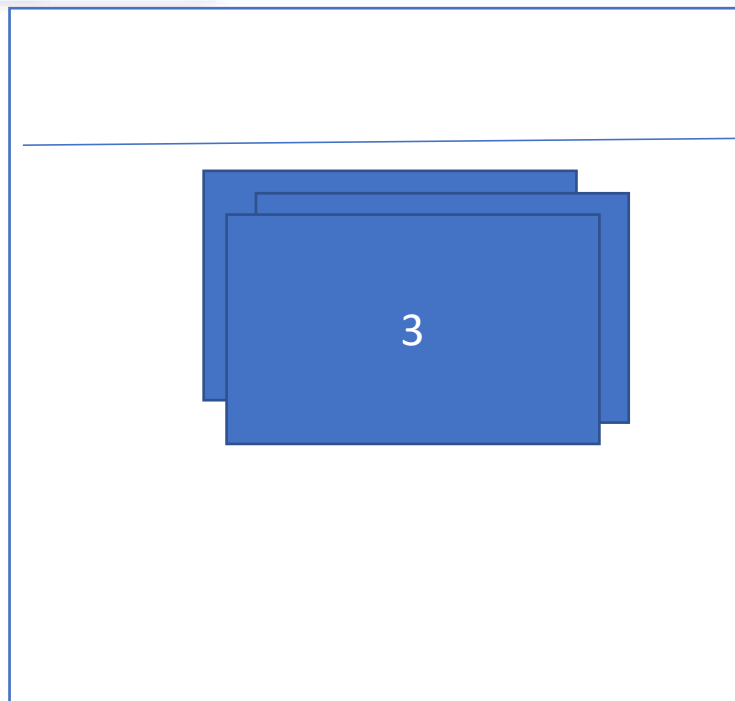
Jsp
Php
C#



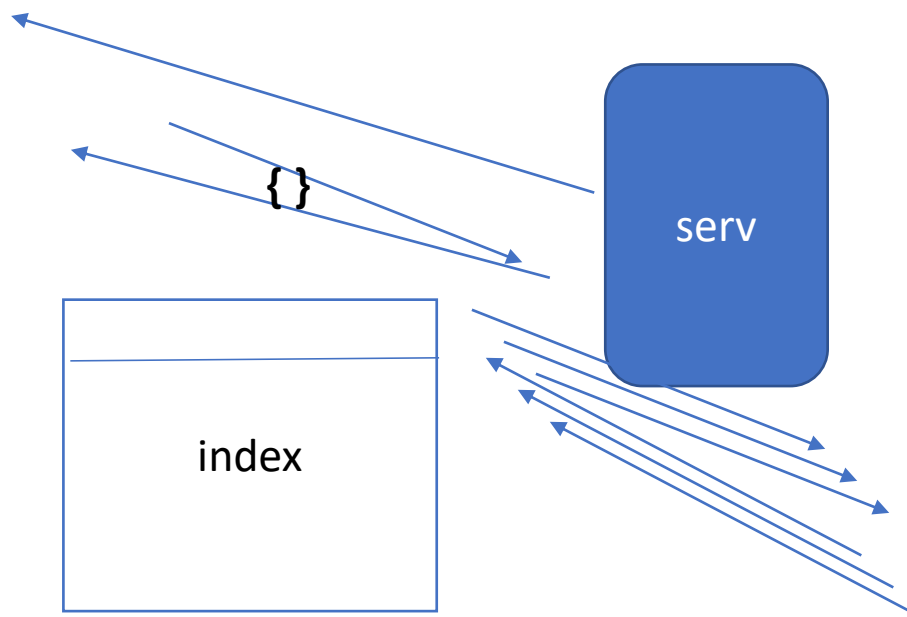
js

Js (toda la logica)
Css

Jsp
Php
C#



SPA





Entonces, si todo está en la misma página, ¿quiere decir esto que no cambia la URL del navegador?

No, no quiere decir eso, es más, lo más normal es que vaya cambiando según nos movemos entre vistas. Pero la clave aquí, es que en realidad no estás cargando una página, tan solo sustituyendo el contenido de la misma.



Podrías no cambiarla si, pero es una buena práctica ya que el navegador va guardando el historial de las URL y le permite al usuario navegar con el botón de atrás y adelante del navegador. Además de que el usuario podría acceder a un contenido concreto directamente si usa la URL en vez de navegar por la web hasta llegar a él.



SPA no quiere decir que no cambie la dirección de la barra de direcciones, es decir, la URL a la que accedes con el navegador. De hecho, es normal que al interactuar con una SPA la URL que se muestra en la barra de direcciones del navegador vaya cambiando también. La clave es que, aunque cambie esta URL, la página no se recarga nunca. El hecho de cambiar esa URL es algo importante, ya que el propio navegador mantiene un historial de pantallas entre las que el usuario se podría mover, pulsando el botón de "atrás" en el navegador o "adelante".



Con todo ello, se consigue que el usuario pueda usar el historial como lo hace en una página normal, siendo que en una SPA la navegación entre la secuencia de páginas del historial se realiza realmente entre vistas de la aplicación.



Por qué una SPA ofrece una experiencia de usuario tan agradable?

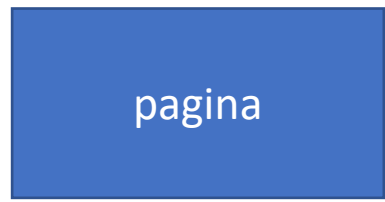
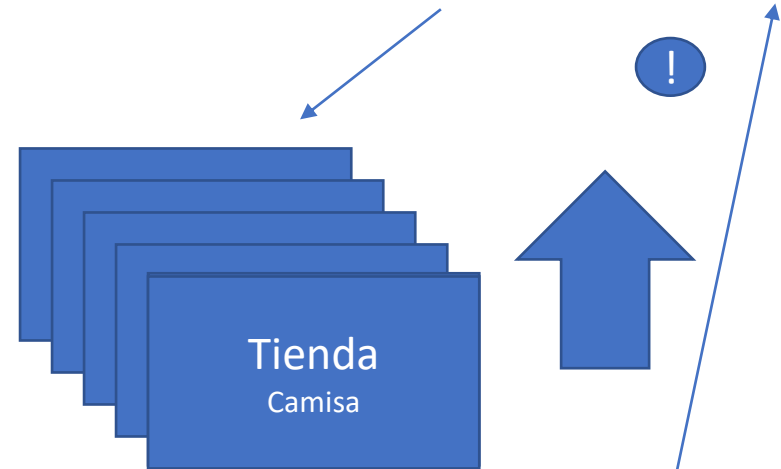
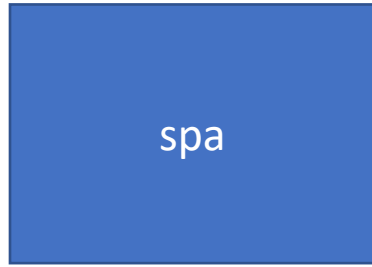
Al pesar muy poco los datos, mucho menos que si estuvieran mezclados dentro de un complejo código HTML y CSS para definir su presentación, las transmisiones son muy rápidas y las comunicaciones entre cliente y servidor se realizan muy fluidas. Nuevamente ayuda a que las páginas respondan muy velozmente al visitante, creando una experiencia de usuario muy agradable.



Las páginas de gestión, o administración de cualquier tipo de servicio, paneles de control y cosas así son muy adecuadas para las SPA. El resultado es una aplicación web se comporta muy parecido a una aplicación de escritorio.



Administración Datos,
Pag SEO





Lenguajes y tecnologías para producir una SPA

Una SPA se puede crear en **Javascript**. Ya que, al ser una aplicación web ejecutada del lado del cliente. A esto habrá que sumarle, lógicamente, el **HTML y CSS**.



Dentro de Javascript, contaremos con multitud de **librerías y frameworks** que nos facilitarán el desarrollo de las SPA. Algunas de las más usadas son:

- Angular
- VUE
- React



Este listado son las librerías y frameworks más comunes, aunque hay muchos otros. Escoger una librería y un framework dependerá, básicamente, de tus gustos, la experiencia que tengas con los mismos, el tamaño de la aplicación a desarrollar...



De todos modos, esto sólo aplica al front-end. Si queremos ampliar la pregunta al lado del **back-end**, es decir, del servidor, nos valdría **cualquier lenguaje de back-end para producir la parte del servidor**. Lo que tendríamos que hacer es crear un API REST que devuelva el JSON necesario para alimentar de datos a la SPA.



Es decir, la SPA nos dara igual cómo esté desarrollada del lado del back-end, es totalmente independiente a este.



Ventajas de las páginas web SPA

Desarrollar una página web SPA tiene muchas ventajas frente al MPA. Algunas de ellas son:



- Se tratan de aplicaciones **fáciles de desarrollar, desplegar y depurar**. Al contar con infinitos frameworks y librerías, será muy sencillo de desarrollar.
- Nos muestran el **contenido de forma sencilla** y elegante al cargar todo el contenido en una única página.



- Además, ejecutando la lógica del lado del navegador, hace que las **interacciones del usuario se ejecuten más rápido**.
- La **analítica** de la web es mucho mas **sencilla** puesto que solo analiza una única página.
- También **facilitará el SEM** puesto que todas las llamadas y posibilidades de interacción se realizarán en una única página.



- Muy útiles para la creación de páginas de aterrizaje.



Desventajas de las páginas web SPA

El SEO no es tan fácil como en una MPA, aunque esto es cada vez más sencillo de optimizar y no supone tanto problema como hace un tiempo.



En sitios muy grandes, el **mantenimiento** del código puede ser algo **complicado** y hay que tener desde el principio muy clara la estructura de la página.

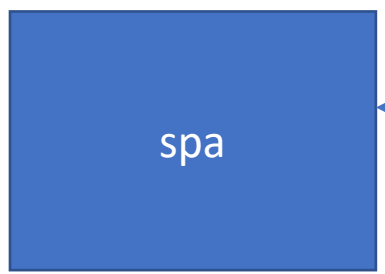
Al tener el **código fuente** directamente en el navegador, si no seguimos buenas prácticas, la **seguridad** del sitio se podría ver expuesta.



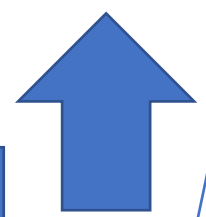
Aunque las SPA son muy rápidas, al **cargar la página por primera vez de forma completa, esto hace que esta carga sea algo lenta**, lo que puede ahuyentar posibles clientes. De todos modos, actualmente, hay métodos para mejorar esta velocidad de carga, que veremos en otro post.



Administración Datos,
Pag SEO



Codigo{
todos
}



{
}

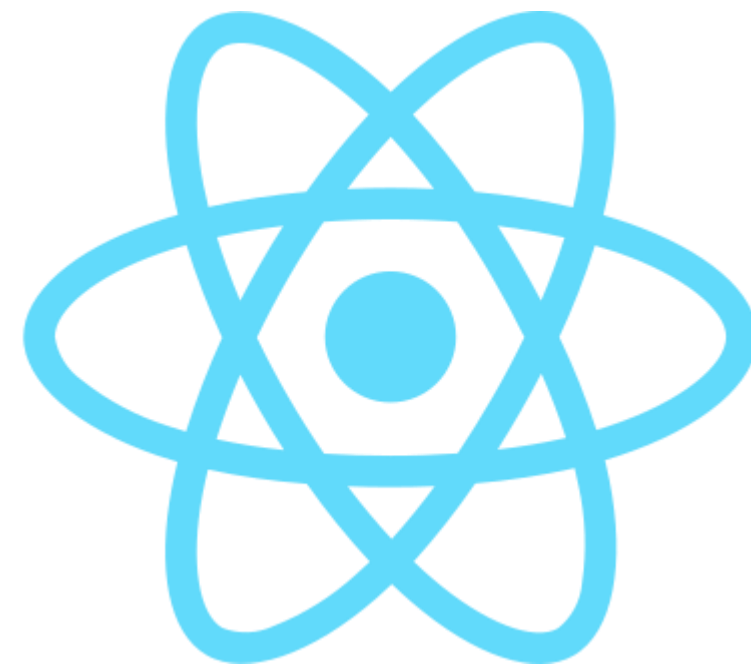




React



- React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.







React fue creado por Jordan Walke, un ingeniero de software de Facebook, quien liberó un primer prototipo de React llamado "FaxJS"

Este fue influenciado por XHP de HTML una librería de componentes para PHP. Este fue usado por primera vez en el Feed de Noticias de Facebook en 2011 y después en Instagram en 2012.

jordwalke/**FaxJs**

Fax Javascript Ui Framework



2

Contributors

1

Issue

320

Stars

19

Forks



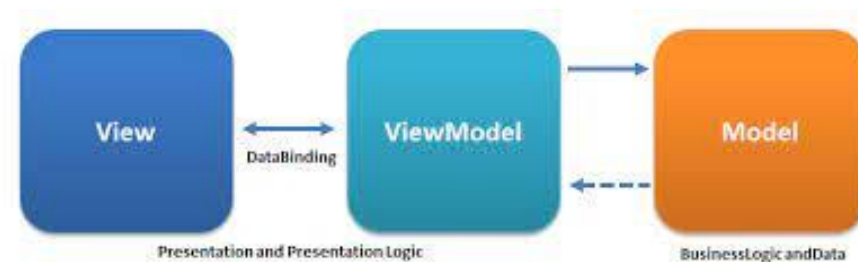
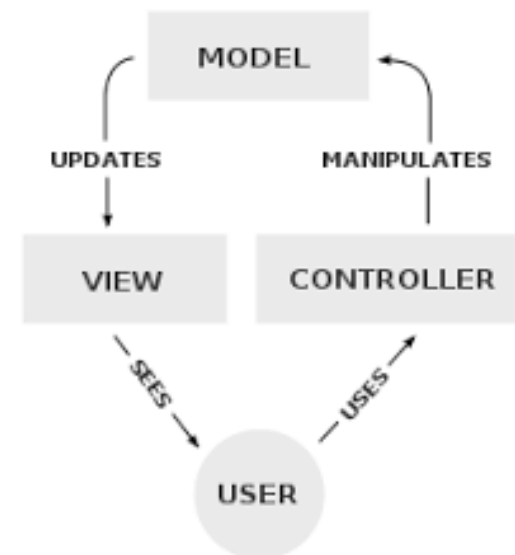


React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar.



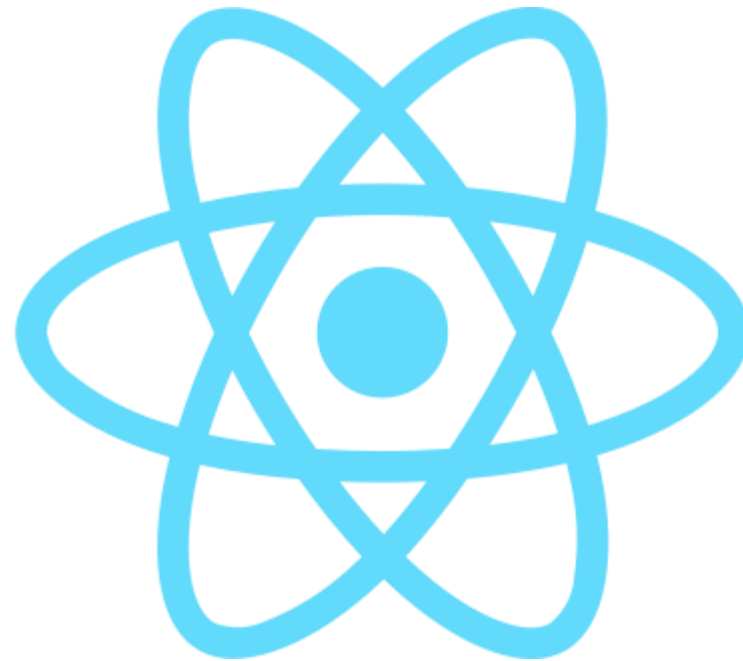
React sólo maneja la interfaz de usuario en una aplicación;

React es la Vista en un contexto en el que se use el patrón MVC (Modelo-Vista-Controlador) o MVVM (Modelo-vista-modelo de vista).





Características





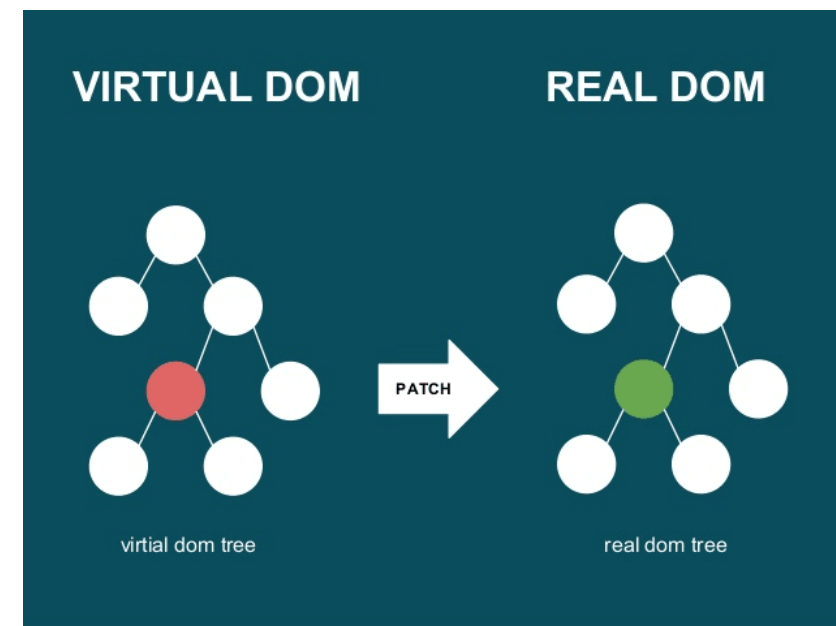
Es una librería ya que con ella puedes crear interfaces web dinámicas. La diferencia con un framework radica en que esta, no te brinda herramientas de sugerencia, pero en nuestro editor de código podemos agregar plugins que nos permite hacerla funcionar tal cual como un framework. (Vue, NG).



Virtual DOM

React mantiene un virtual DOM propio, en lugar de confiar solamente en el DOM del navegador. Esto deja a la biblioteca determinar qué partes del DOM han cambiado comparando contenidos entre la versión nueva y la almacenada en el virtual DOM, y utilizando el resultado para determinar cómo actualizar eficientemente el DOM del navegador.¹⁵¹⁶

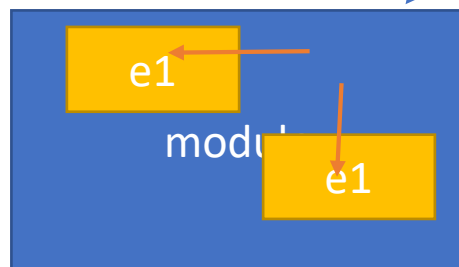
Propiedades (props) de react.





Las propiedades

Las propiedades (también conocidas como 'props') pueden definirse como los atributos de configuración para dicho componente. Éstas son recibidas desde un nivel superior, normalmente al realizar la instancia del componente y por definición son inmutables.

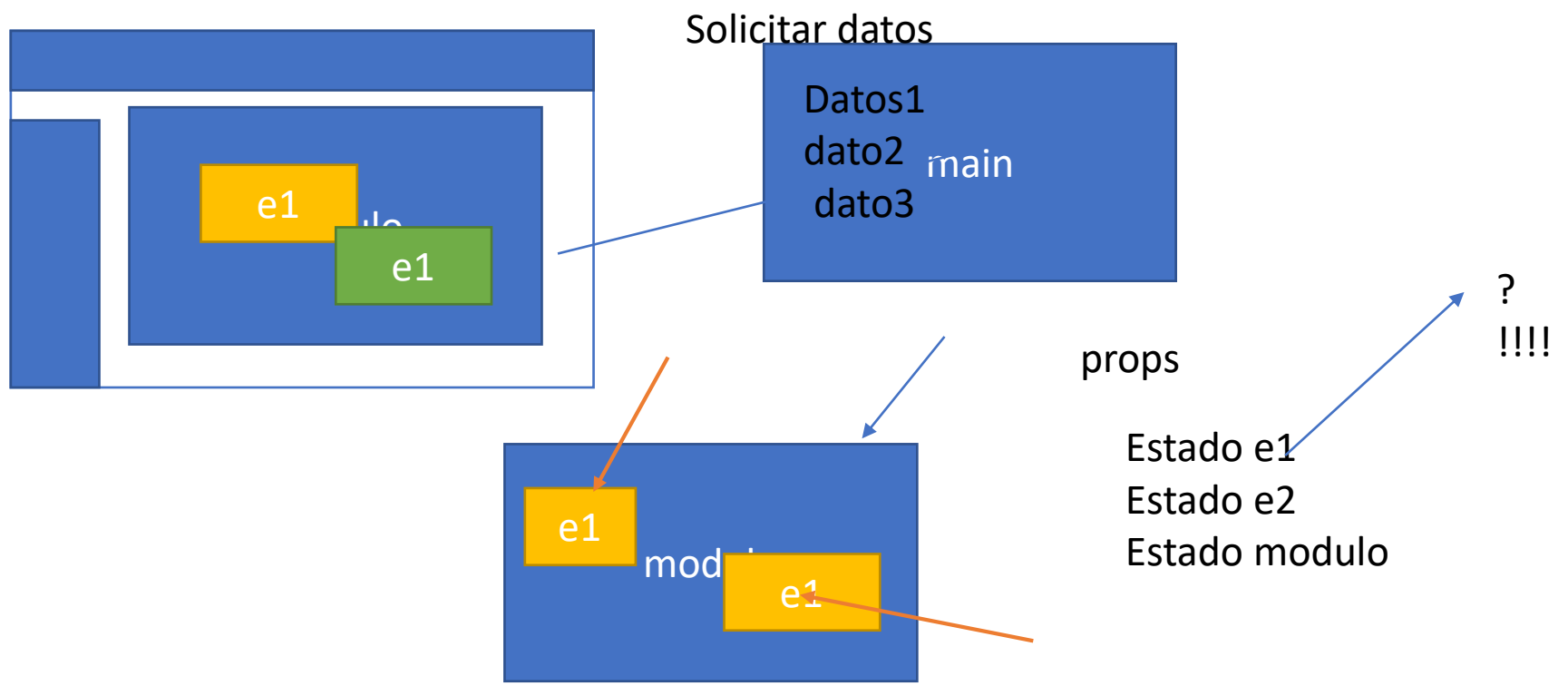


props



El Estado

El estado de un componente se define como una representación del mismo en un momento concreto, es decir, una instantánea del propio componente. Existen dos tipos de componentes con y sin estado, denominados statefull y stateless.





Ciclos de vida

El ciclo de vida es una serie de estados por los cuales pasan los componentes statefull a lo largo de su existencia. Se pueden clasificar en tres etapas de montaje o inicialización, actualización y destrucción. Dichas etapas tienen correspondencia en diversos métodos.



`shouldComponentUpdate` permite al desarrollador prevenir el re-renderizado innecesario de un componente, devolviendo falso si no es necesario. `componentDidMount` es llamado una vez que el componente es "montado" (el componente ha sido creado en la interfaz de usuario, usualmente asociándolo con el nodo del DOM). Esto es comúnmente usado para provocar la carga de datos desde una fuente remota a través de una API.



- `componentWillUnmount` es llamado inmediatamente antes de que el componente es "desmontado". Es comúnmente usado para limpiar la demanda de dependencias del componente que no será simplemente removido con el desmontaje del componente.



render es el método más importante de los ciclos de vida y el único requerido en cualquier componente. Es usualmente llamado cada vez que el estado del componente es actualizado, reflejando los cambios en la interfaz de usuario.



JSX

React utiliza una sintaxis parecida a HTML, llamada JSX. No es necesaria para utilizar React, sin embargo, hace el código más legible, y escribirlo es una experiencia similar a HTML.



Con JSX

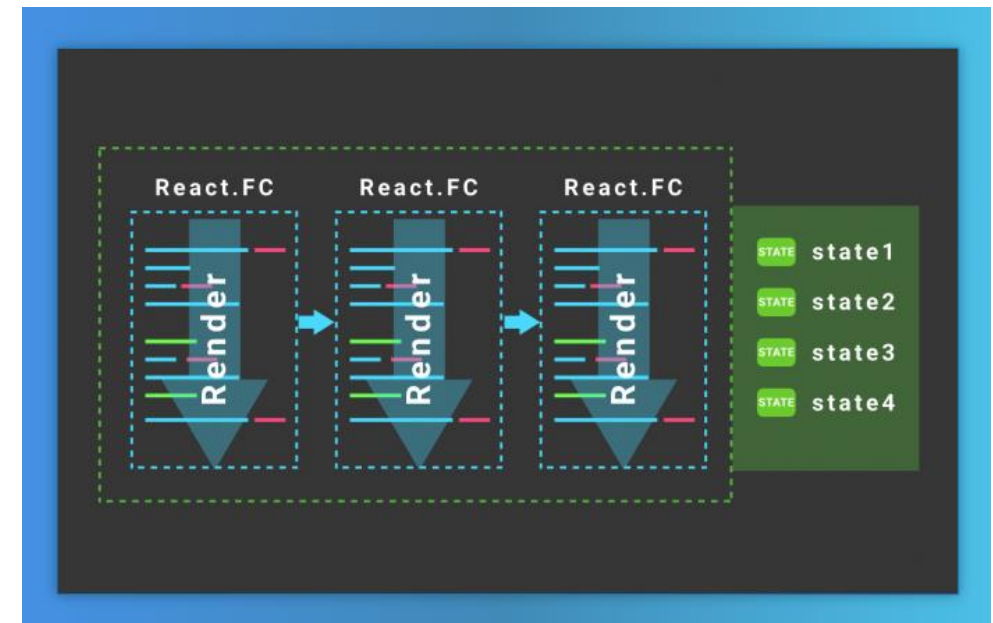
```
class Hello extends React.Component {  
  render() {  
    return <div>Hello {this.props.toWhat}</div>;  
  }  
}  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<Hello toWhat="World" />);
```

Sin JSX

```
class Hello extends React.Component {  
  render() {  
    return React.createElement('div', null, `Hello ${this.props.toWhat}`);  
  }  
}  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(React.createElement(Hello, {toWhat: 'World'}, null));
```



Los *Hooks* son una API que nos permite usar estado y otras características en un **componente funcional**.



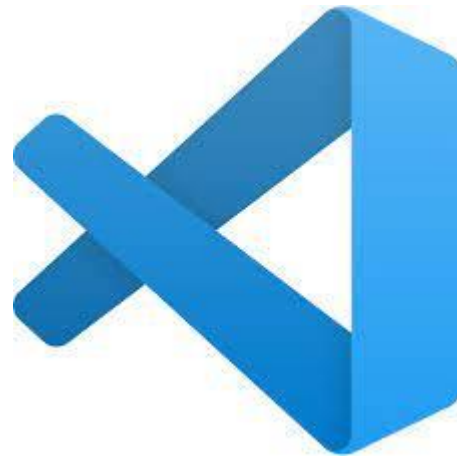


- Estructura de directorios

```
> public
  > src
    > componentes
      JS ContentHeader.js
      JS Footer.js
      JS Menu.js
      JS Navbar.js
      JS SidebarContainer.js
    > paginas
      > auth
      > proyectos
      JS Home.js
      > utils
      # App.css
      JS App.js
      JS config.js
      # index.css
      JS index.js
```



- Que necesitamos para empezar con React.





Creando nuestro primer proyecto

```
npx create-react-app my-app  
cd my-app  
npm start
```



- Npm start...

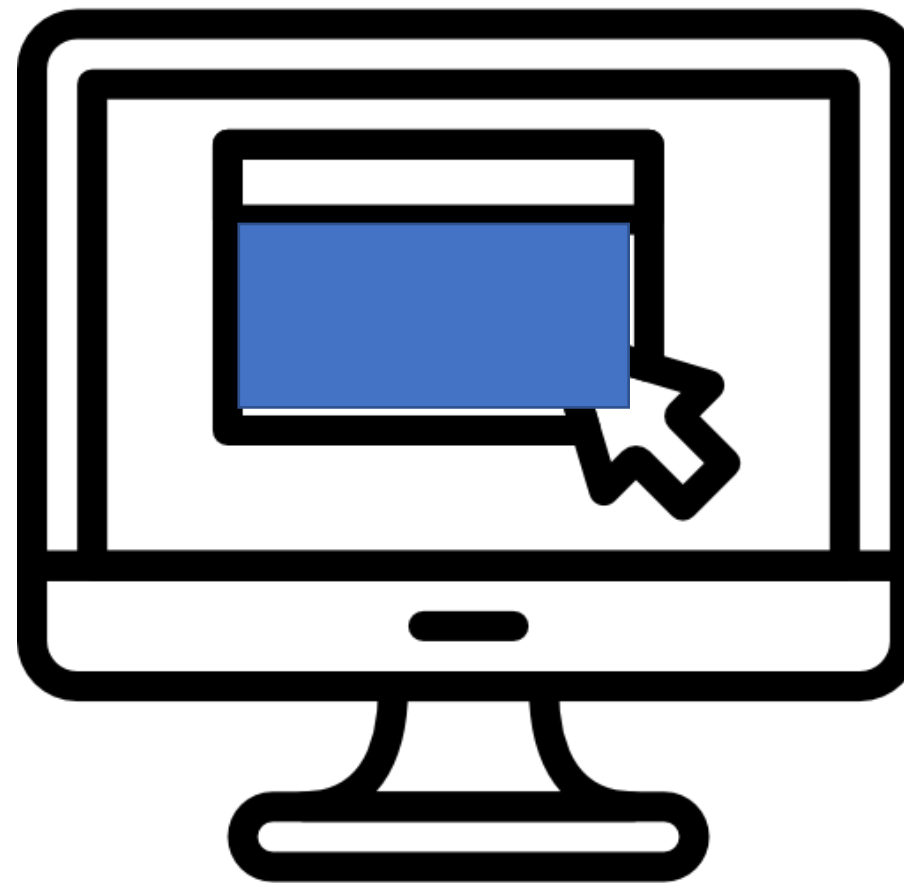


Edit src/App.js and save to reload.

[Learn React](#)



```
<title>React App</title>
</head>
<body>
  <noscript>You need to enable Ja
  <div id="root"></div>
  <!--
```





- Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'))
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```



```
1  {
2    "name": "react-course",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.16.4",
7      "@testing-library/react": "^13.3.0",
8      "@testing-library/user-event": "^13.5.0",
9      "react": "^18.2.0",
10     "react-dom": "^18.2.0",
11     "react-scripts": "5.0.1",
12     "web-vitals": "^2.1.4"
13   },
14   "scripts": {
15     "start": "react-scripts start",
16     "build": "react-scripts build",
17     "test": "react-scripts test",
18     "eject": "react-scripts eject"
19   }
20 }
```



- Hooks
- Estructura directorios
- Instalar react



- <https://www.incentro.com/es-ES/blog/que-es-web-simple-page-applications>
- <https://desarrolloweb.com/articulos/que-es-una-spa.html>
- <https://www.kikopalomares.com/blog/que-es-una-web-spa-single-page-application>
- <https://es.wikipedia.org/wiki/React>
- <https://blog.clicko.es/patron-diseno-mvvm-usando-wpf-parte-1/>
- <https://es.reactjs.org/docs/react-without-jsx.html>

Bibliografía

Practicar los temas tratados



Nos vemos en la siguiente sesión



**El futuro digital
es de todos**

MinTIC

**Muchas
Gracias**



www.mintic.gov.co

**Misión
TIC2022**

unab
VIGILADA MINEDUCACIÓN