

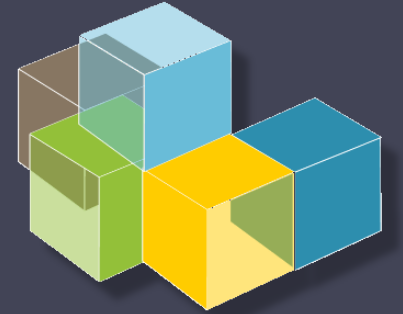


Universidad de Oviedo



Escuela de
Ingeniería
Informática

Es
Español



ARQUITECTURA
DEL SOFTWARE

Arquitectura del software

Lab. 03

React

Solid

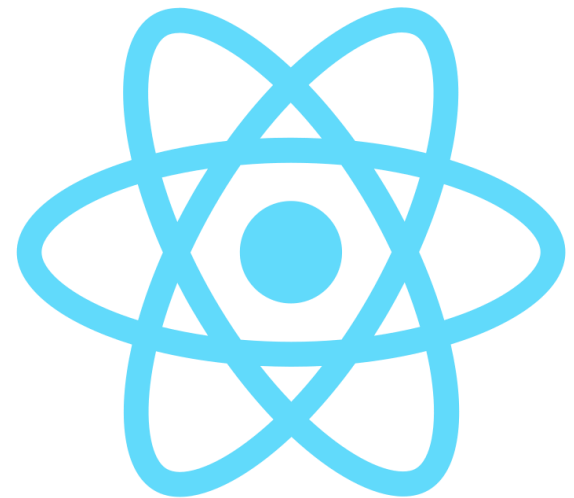
2022-23

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Cristian Augusto Alonso

Qué es React.js?

React es una librería Javascript para construir interfaces de usuario para la web así como aplicaciones móviles

- Código abierto
- Inicialmente por Facebook (Meta)
- Basada en componentes



Porqué React?

Varias razones:

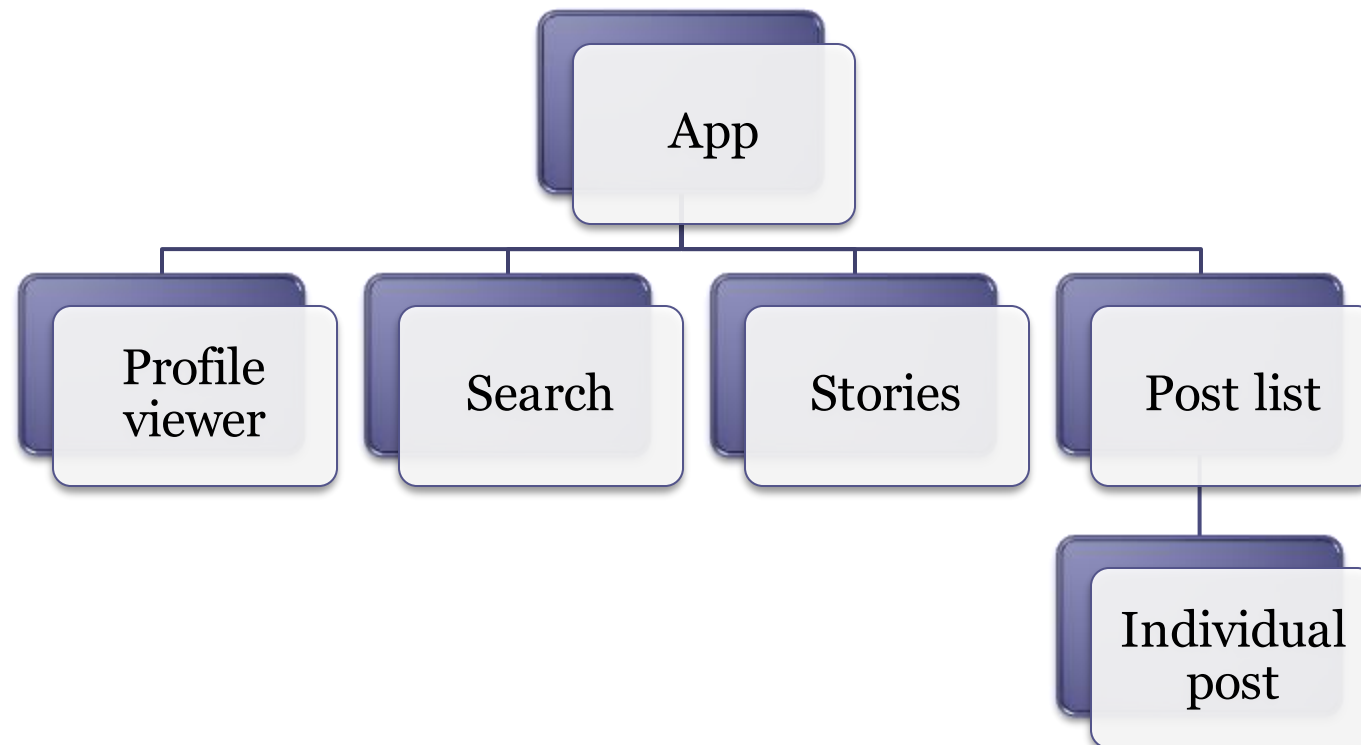
- Simplicidad y fácil de aprender
- Componentes reutilizables
- También tiene posibilidad Native (React Native)
- Ampliamente utilizada y muchas herramientas
- Alta testabilidad

Componentes

Las páginas son modeladas usando componentes

Un componente es una parte del interfaz de usuario

Ejemplo: Instagram



Componentes

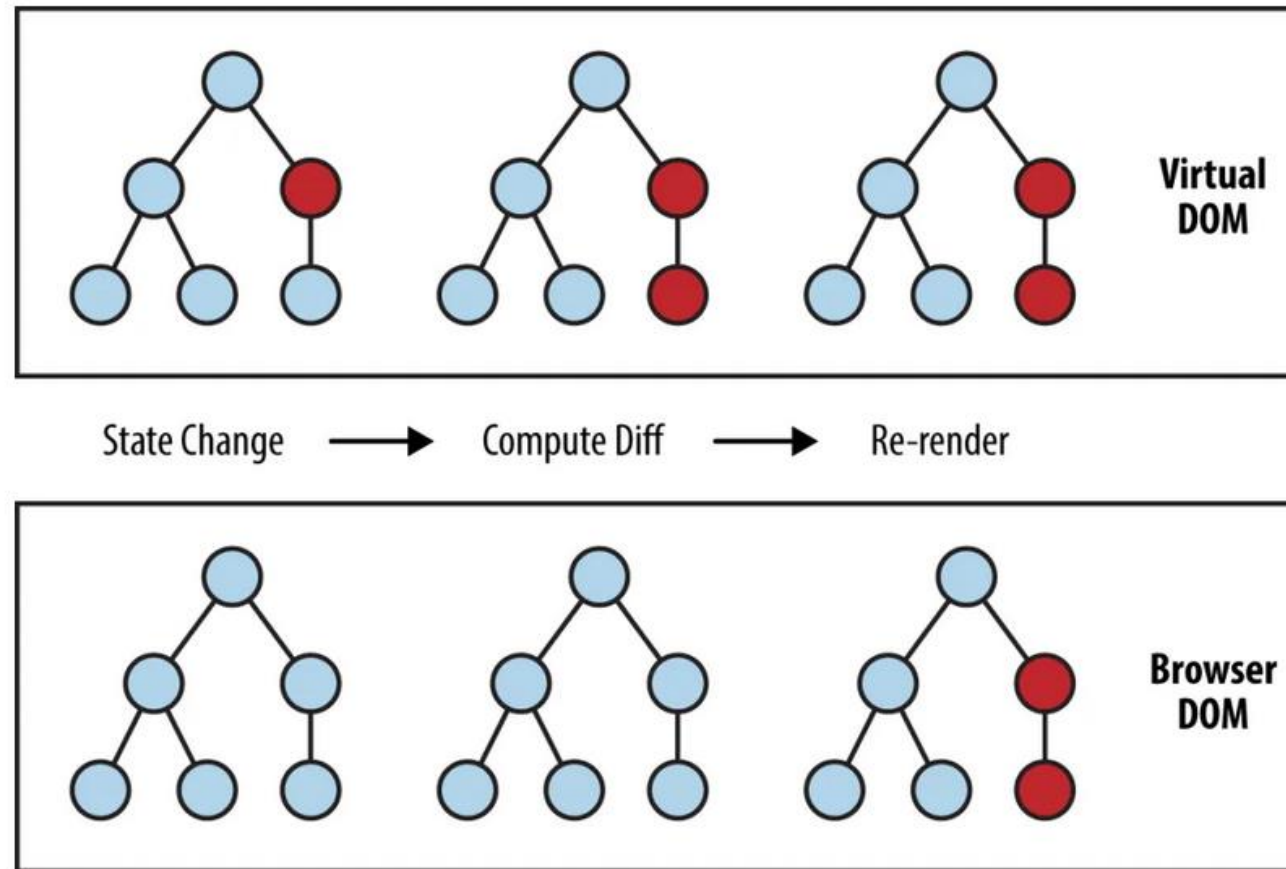
Un componente puede ser implementado como una clase o función (Hook) Javascript

- Tiene un estado
- Y un método render que control lo que se muestra en el interfaz de usuario
- Cuando cambia el estado, react actualiza el elemento y sus hijos en memoria
- La representación de este elemento en memoria se llama Virtual Dom

```
class ProfileViewer{  
  state = {}  
  render(){  
  
  }  
}
```

React **reacciona**
a cambios

Virtual DOM



También se puede usar Hooks!

Substituye clases por funciones

- En el siguiente ejemplo, usamos una *hook* `useState` para gestionar los cambios de nombre en la app
- Una vez el botón es pulsado, el estado se cambia, el DOM virtual es actualizado, y la página se refresca automáticamente

```
const App = () => {  
  const [name, setName] = useState('World');  
  return (  
    <div className="App">  
      <h1>Hello, {name}!</h1>  
      <button onClick={() => setName('James')}>  
        Click me to change the name  
      </button>  
    </div>  
  );  
}
```

SOLID

Tecnología nueva para organizar apps, información e identidades de forma descentralizada

- Desarrollada inicialmente en MIT
- Liderado por Tim Berners-Lee

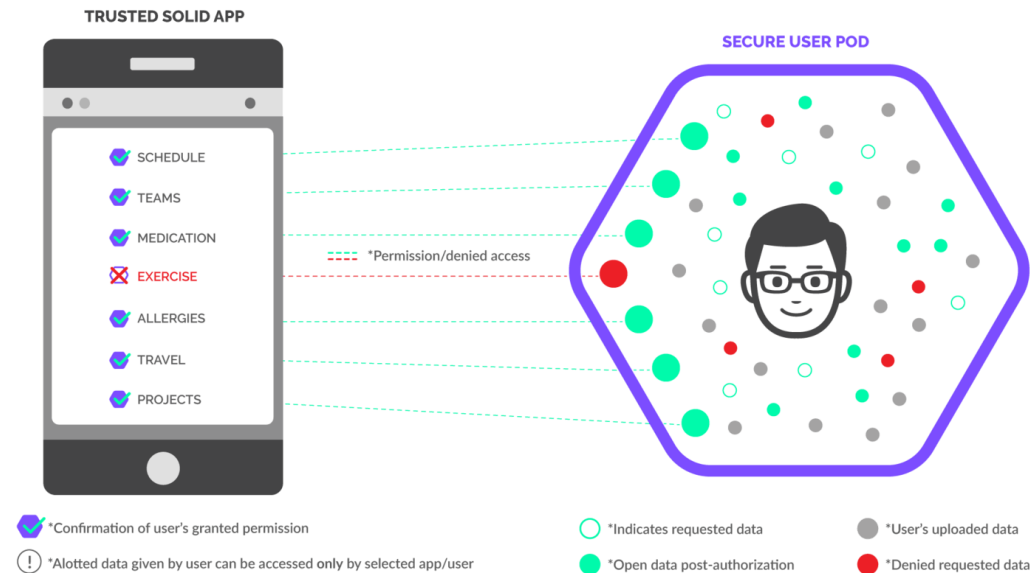


Creación de un POD



Para crear un POD, se necesita un proveedor de PODs. Se puede utilizar uno externo o alojarlo en nuestro propio servidor de pods

- <https://inrupt.net/>



Combinar Solid y React?

- Componentes React

<https://github.com/inrupt/solid-ui-react>

- Proyecto de Ejemplo (javascript)

<https://github.com/pglez82/solid-react-example>

- Proyecto de Ejemplo (typescript)

<https://github.com/Arquisoft/solid-react-example/>

- Documentación Solid

<https://docs.inrupt.com/>

- Awesome Solid

<https://github.com/pdsinterop/awesome-solid>

- Videos introducing Solid by Jackson Morgan

https://www.youtube.com/playlist?list=PLtNrK03_EIXBGf5fmrqqkYew9Z3L0YifN

Documentación adicional

Ejercicios Estado React

- I. [Ej1](#) Creamos un contador
- II. [Ej2](#) Trabajamos con estados complejos (objetos)
- III. [Ej3](#) Ejercicio con diferentes handlers()
- IV. [Ej4](#) Añadimos elementos a un array
- V. [Ej5](#) Cambiamos el comportamiento de un componente (color de fondo)

Documentación adicional

Ejercicios Renderización arrays en React

- I. [Ej1](#) Renderizar Array
- II. [Ej2](#) Refactorización
- III. [Ej3](#) Añadir elementos al array
- IV. [Ej4](#) Añadimos elementos desde un formulario

Documentación adicional

Ejercicios Programación asíncrona

- I. [Ej1](#) Fetch() -> Hacer una petición a una API
- II. [Ej2](#) useEffect()
- III. [Ej3](#) Renderización condicional
- IV. [Ej4](#) Una refactorización
- V. [Ej5](#) Peticiones utilizando librería axios()

Documentación adicional

Ejercicios Typescript + React

- I. [Ej1](#) Contador con typescript
- II. [Ej2](#) Segundo ejercicio
- III. [Ej3](#) Ejemplo de interface

Documentación adicional

Enlaces de interes

- I. Página de curso [Bootcamp Fullstack](#)
- II. [Primera conferencia de Node.js](#) de Ryan Dahl