

UCM

Programación de Aplicaciones Web **Electivo III**

Román Gajardo



<https://github.com/romanncodes>



roman.gajardo@gmail.com



Contenidos

- Javascript
- React
- Components
- Estados

Javascript Arrays

- https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array

Javascript - Funciones

JS



```
// Función tradicional
function (a){
    return a + 100;
}

// Desglose de la función flecha

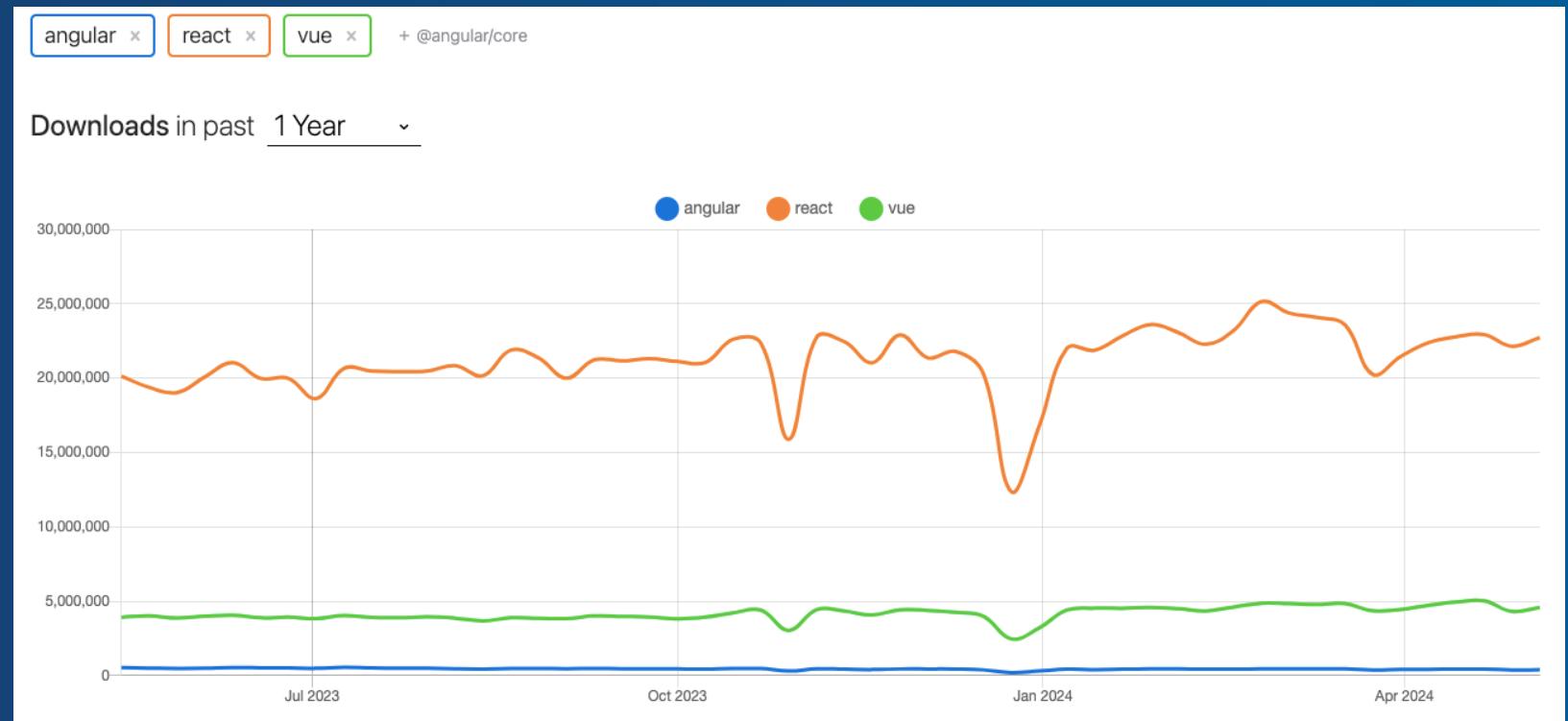
// 1. Elimina la palabra "function" y coloca la flecha entre el argumento y el
// corchete de apertura.
(a) => {
    return a + 100;
}

// 2. Quita los corchetes del cuerpo y la palabra "return" – el return está
// implícito.
(a) => a + 100;

// 3. Suprime los paréntesis de los argumentos
a => a + 100;
```

React

- React es una librería open source de JavaScript para desarrollar interfaces de usuario. Fue lanzada en 2013 y desarrollada por Facebook



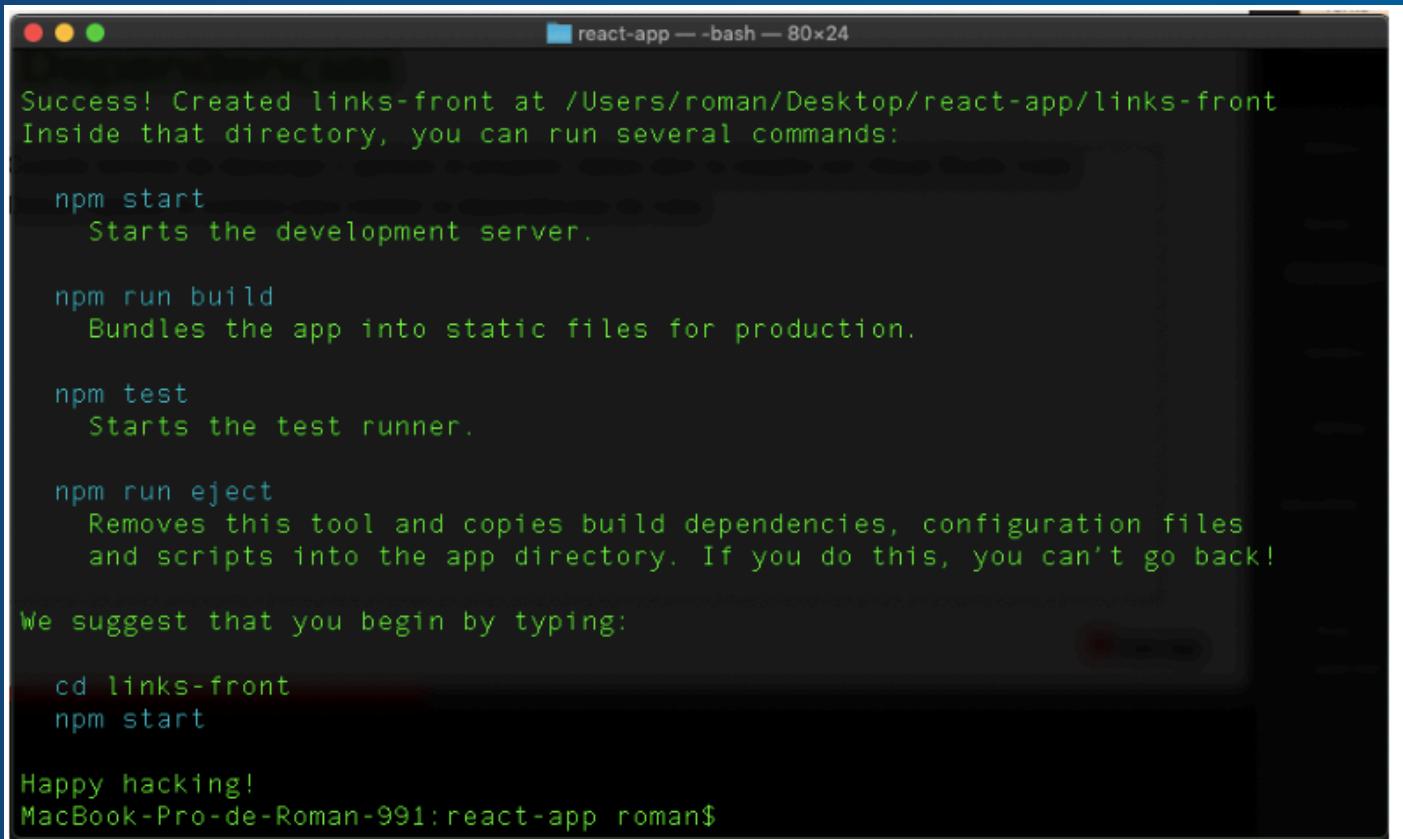
Herramientas



Visual Studio Code

Creación de un proyecto

- En la terminal escribir `npx create-react-app <nombre-proyecto>`



```
Success! Created links-front at /Users/roman/Desktop/react-app/links-front
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

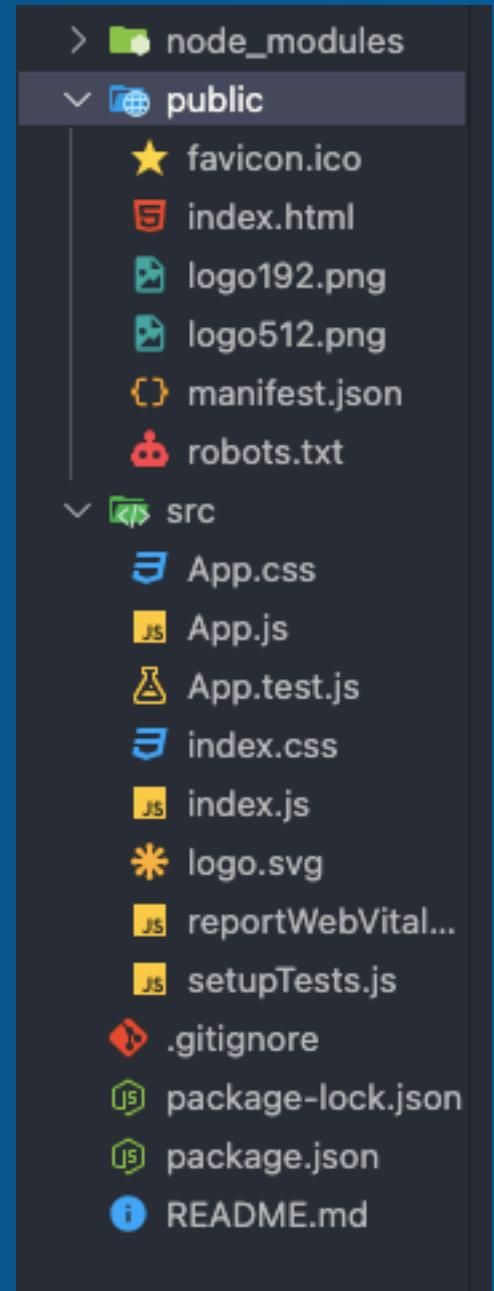
We suggest that you begin by typing:

  cd links-front
  npm start

Happy hacking!
MacBook-Pro-de-Roman-991:react-app roman$
```

Estructura de proyecto

- Node-modules: dependencias
- Public
- Src
- Package.json



Componentes

```
function Componente(){  
  return ( <div>  
    |     |   <h1>Soy Un componente</h1>  
    |     |   </div>  
  );  
  export default Componente;
```

```
import Componente from "./Componente";  
  
function Principal(){  
  return ( <div>  
    |     |   <Componente />  
    |     |   </div>);  
}
```

Componentes y properties

```
function Componente(props){  
  
    return ( <div>  
        |     |     |     <h1> {props.titulo} </h1>  
        |     |     |</div>  
    );  
}  
export default Componente;
```

```
import Componente from "./Componente";  
  
function Principal(){  
    return ( <div>  
        |     |     |     <Componente titulo="Primer Componente" />  
        |     |     |</div>);  
}
```

■ Estados

- useState

```
1  import React from "react";
2
3
4  function Componente(){
5      const [items, setItems]=React.useState(["pera", "manzana", "sandia"]);
6      const [name, setName]=React.useState('Juan');
7      return <>
8          <h1>Mi nombre es {name}</h1>
9          <h3>Listado de frutas</h3>
10         <ul>
11             {items.map((item, index) => (
12                 <li key={index}> {item} </li>
13             ))}
14         </ul>
15
16     </>
17 }
18
19 export default Componente;
```

Eventos

```
1 import React from "react";
2
3
4 function Componente(){
5     const [name, setName]=React.useState('Juan');
6
7
8     const evento = ()=>{
9         console.log("click");
10    }
11    const obtenerNombre= (event)=>{
12        let valor = event.target.value;
13        setName(valor);
14    }
15
16    return <>
17
18        <input type="text" onChange={(e)=>{setName(e.target.value)}}/>
19        <input type="text" onChange={obtenerNombre}>
20
21        <button onClick={evento}>click</button>
22
23    </>
24}
25
26 export default Componente;
```

Demo

Primer Ejemplo con React

Buscar

- R React eliminar
- J Java eliminar
- P PHP eliminar
- P Python eliminar

Primer Ejemplo con React

Buscar

- R React eliminar
- J Java eliminar
- P PHP eliminar
- P Python eliminar

componente Header

componente Search

componente Item

componente ItemList

```
graph TD; A[componente Header] --> B[componente Search]; B --> C[componente Item]; C --> D[componente ItemList]; D --> E[componente Item]; E --> F[eliminar]; D --> G[componente Item]; G --> H[eliminar]; D --> I[componente Item]; I --> J[eliminar]; D --> K[componente Item]; K --> L[eliminar];
```

Rutas

- React Router DOM es una librería que se utiliza bastante con React y que facilita el proceso para definir rutas de navegación en una aplicación web.
- Actualmente la versión es react-router-dom 6
 - <https://www.npmjs.com/package/react-router-dom>
- Instalación: npm install react-router-dom

Tipos de rutas

- Browser Router: su navegación funciona mediante / que indican en que página de la aplicación estamos.
- Hash Router: su navegación funciona por medio de /#/ utilizado para renderizar contenido utilizando los # mediante Javascript. (SPA)
- Memory Router: funciona por medio de un array de rutas, e indica en que ruta estamos renderizando ese contenido. Es utilizando para desarrollo mobile

Componentes de Rutas

- Route, Routes, HashRoute

```
1  import { HashRouter, Route, Routes } from "react-router-dom"
2
3  function Application(){
4      return <>
5          <HashRouter>
6              <Menu />
7              <Routes>
8                  <Route path="/" element={<App/>}/>
9                  <Route path="/page1" element={<Page1/>}/>
10                 <Route path="/page2" element={<Page2/>}/>
11                 <Route path="*" element={<p>404 - Not Found</p>}/>
12             </Routes>
13         </HashRouter>
14     </>
15 }
16
17 export default Application
```

Componentes de Navegación

- Link y NavLink son componentes de react-router-dom que ayudan a evitar el uso de etiquetas <a>
- Una etiqueta <a> realiza una petición al servidor, para evitar esto se debe hacer uso de #

```
1  function Menu(){
2      return <>
3          <h3>Menu</h3>
4          <nav>
5              <ul>
6                  <li>
7                      <a href="#/">Home</a>
8                  </li>
9                  <li>
10                     <a href="#/page1">Page 1</a>
11                 </li>
12                 <li>
13                     <a href="#/page2">Page 2</a>
14                 </li>
15             </ul>
16         </nav>
17     </>
18 }
19
20 export {Menu};
```

Componentes de Navegación

- Link es un componente que no hace una petición al servidor y no es necesario indicar el #

```
1 import { Link } from "react-router-dom";
2
3 function Menu(){
4     return <>
5         <h3>Menu</h3>
6         <nav>
7             <ul>
8                 <li>
9                     <Link to="/">Home</Link>
10                </li>
11                <li>
12                    <Link to="/page1">Page 1</Link>
13                </li>
14                <li>
15                    <Link to="/page2">Page 2</Link>
16                </li>
17            </ul>
18        </nav>
19    </>
20 }
21
22 export {Menu};
```

Componentes de Navegación

- NavLink tiene las mismas propiedades que Link, la diferencia es que se pueden incluir style o className para aplicar estilos ya que trae incorporado un argumento llamado isActive para definir en que link estamos ubicados

```
1 import { NavLink } from "react-router-dom";
2
3 function Menu(){
4   return <>
5     <h3>Menu</h3>
6     <nav>
7       <ul>
8         <li>
9           <NavLink
10              to="/"
11              style={({isActive}) => ({color: isActive?"green":"grey"})}>
12             Home
13           </NavLink>
14         </li>
15         <li>
16           <NavLink
17              to="/page1"
18              style={({isActive}) => ({color: isActive?"green":"grey"})}>
19             Page 1
20           </NavLink>
21         </li>
22         <li>
23           <NavLink
24              to="/page2"
25              style={({isActive}) => ({color: isActive?"green":"grey"})}>
26             Page 2
27           </NavLink>
28         </li>
29       </ul>
30     </nav>
31   </>
32 }
33 export {Menu};
```

Arreglo de rutas

- Opción que permite una mejor gestión de rutas y mejora del código

```
1 import { NavLink } from "react-router-dom";
2
3 function Menu() {
4     return <>
5         <h3>Menu</h3>
6         <nav>
7             <ul>
8                 {routes.map((item, index) => (
9                     <li key={index}>
10                        <NavLink
11                            to={item.to}
12                            style={({ isActive }) => ({ color: isActive ? "green" : "grey" })}
13                            >{item.text}</NavLink>
14                        </li>
15                    ))}
16
17                </ul>
18            </nav>
19        </>
20    }
21    const routes = [];
22    routes.push({ to: "/", text: "Home" })
23    routes.push({ to: "/page1", text: "Page 1" })
24    routes.push({ to: "/page2", text: "Page 2" })
25
26    export { Menu };
```

Cuando usar rutas dinámicas

- Cuando una aplicación tiene rutas del tipo
 - curso/react
 - curso/python
- Y el contenido de react y python poseen la misma estructura.



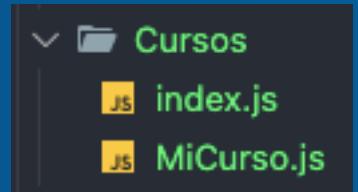
```
1 const cursos=[  
2   {  
3     id:1,  
4     name:"Curso de Java",  
5     url:"java",  
6     description:"Java con Springboot"  
7   },  
8   {  
9     id:2,  
10    name:"Curso de Python",  
11    url:"python-basic",  
12    description:"Curso de iniciacion a la programacion "  
13  },  
14  {  
15    id:3,  
16    name:"Curso de React Native",  
17    url:"react-native",  
18    description:"Desarrollo de aplicaciones mobiles"  
19  },  
20]  
21  
22  
23  
24 export {cursos}
```

Configuración de rutas dinámicas

```
1 const routes = [];
2 routes.push({ to: "/", text: "Home" })
3 routes.push({ to: "/page1", text: "Page 1" })
4 routes.push({ to: "/page2", text: "Page 2" })
5 routes.push({ to: "/cursos", text: "Cursos" })
```

```
12 function Application(){
13     return <>
14         <HashRouter>
15             <Menu />
16             <Routes>
17                 <Route path="/" element={<App/>}/>
18                 <Route path="/page1" element={<Page1/>}/>
19                 <Route path="/page2" element={<Page2/>}>/>
20                 <Route path="/cursos" element={<Cursos/>}>/>
21                 <Route path="/cursos/:url" element={<MiCurso/>}>/>
22                 <Route path="*" element={<p>404 - Not Found</p>}>/>
23             </Routes>
24         </HashRouter>
25     </>
26 }
27
28 export default Application
```

Configuración de rutas dinámicas



```
1 import { Link } from "react-router-dom"
2 import { cursos } from "../../data/cursos"
3
4 function Cursos(){
5     return <>
6         <h3>Mis cursos</h3>
7         {cursos.map((item)=>(
8             <Link key={item.id} to={item.url}>
9                 {item.name}
10            </Link>
11        )));
12    </>
13 }
14 export {Cursos}
```

```
1 import { useParams, useNavigate } from "react-router-dom"
2 import { cursos } from "../../data/cursos";
3
4 function MiCurso(){
5     const {url} = useParams();
6     const curso = cursos.find(item=>item.url==url)
7     const back = useNavigate();
8     return <>
9         <h3>{curso.name}</h3>
10        <p>{curso.description}</p>
11        <button onClick={()=>{back(-1)}}>Volver</button>
12    </>
13 }
14 export {MiCurso}
```

Configuración de rutas dinámicas

- Outlet

```
12 function Application(){
13   return <>
14     <HashRouter>
15       <Menu />
16       <Routes>
17         <Route path="/" element={<App/>}/>
18         <Route path="/page1" element={<Page1/>}/>
19         <Route path="/page2" element={<Page2/>}/>
20         <Route path="/cursos" element={<Cursos/>}>
21           <Route path=":url" element={<MiCurso/>}>/>
22         </Route>
23
24         <Route path="*" element={<p>404 - Not Found</p>}/>
25       </Routes>
26     </HashRouter>
27   </>
28 }
```

```
1 import { Link, Outlet } from "react-router-dom"
2 import { cursos } from "../../../../data/cursos"
3
4 function Cursos(){
5   return <>
6     <h3>Mis cursos</h3>
7     {cursos.map((item)=>(
8       <Link key={item.id} to={item.url}>
9         {item.name}
10        </Link>
11      )))
12     <Outlet/>
13   </>
14 }
15 export {Cursos}
```

Consumo de api con Fetch

- https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch
- Proporciona una interfaz para acceder y manipular partes del canal HTTP, tales como peticiones y respuestas

Consumo de api con Fetch

```
fetch("https://randomuser.me/api/?results=50")
  .then((res) => {
    return res.json();
})
  .then((data) => {
    console.log(data);
})
  .catch((error) => console.log("error"));
```

```
async function loadUsers() {
  try {
    const res = await fetch("https://randomuser.me/api/?results=50");
    const data = await res.json();
    console.log(data);
  } catch (error) {
    console.log(error);
  }
}
```

Ejemplo

```
export async function getCountries(){
  try{
    const res = await fetch("https://restcountries.com/v3.1/all");
    const data = await res.json();
    return data;
  }catch(error){
    console.log(error);
  }
}

export async function getCountryByName(name){
  try{
    const res = await fetch("https://restcountries.com/v3.1/name/"+name);
    const data = await res.json();
    return data;
  }catch(error){
    console.log(error);
  }
}
```

Ejemplo

```
const [countries, setCountries] = useState([]);  
  
const [name, setName]=useState('')
```

```
React.useEffect( () => {  
  async function fetchData(){  
    const data = await getUsers();  
    console.log(data)  
  }  
  fetchData();  
, [])
```