
1-	FUNDAMENTOS DE LAS APLICACIONES WEB:	2
	CONSULTAR LOS RECURSOS QUE SE SOLICITAN PARA DESPLEGAR ESA APLICACIÓN WEB	2
	CONSULTAR AL DOM	2
2-	JAVASCRIPT:	3
	JAVASCRIPT	3
	CREAR VARIABLES	3
	STRING.....	3
	ARRAY.....	4
	OBJETOS	4
	FUNCIONES	4
	COMPARACIONES	5
3-	REACT:	6
	INSTALAR REACT	6
	CREAR UN PROYECTO	7
	INICIALIZAR EL PROYECTO	8
	EXTENSIONES	10
	EJERCICIO 1	12
	FUNCIONES y PROPS	13
	ARROW FUNCTION (SIMPLIFICAR FUNCIONES)	14
	OBJETOS	15
	ARRAYS.....	16
	CONSOLE.LOG y DEBUGGER.....	17
	ESTADO DEL COMPONENTE, RENDERIZADO CONDICIONAL Y EVENTOS	19
	EJERCICIO 2	21
	RENDERIZAR UNA COLECCIÓN DE ELEMENTOS (LISTAS) CON REACT	22
	LISTAS KEY	25
	EXPORTAR MODULOS	26
	FORMULARIOS EJERCICIO 4	27
	RECUPERAR DATOS DEL SERVIDOR – FETCHING – EJERCICIO 5.....	31
	LOGICA DE NEGOCIO (ARQUITECTURA).....	35
4-	GIT:	37
	INSTALAR GIT	37
	GITHUB.....	37

1- FUNDAMENTOS DE LAS APLICACIONES WEB:

//enlace Youtube Bootcamp

https://www.youtube.com/playlist?list=PLV8x_i1fqBw0Kn_fbIZTa3wS_VZAqddX7

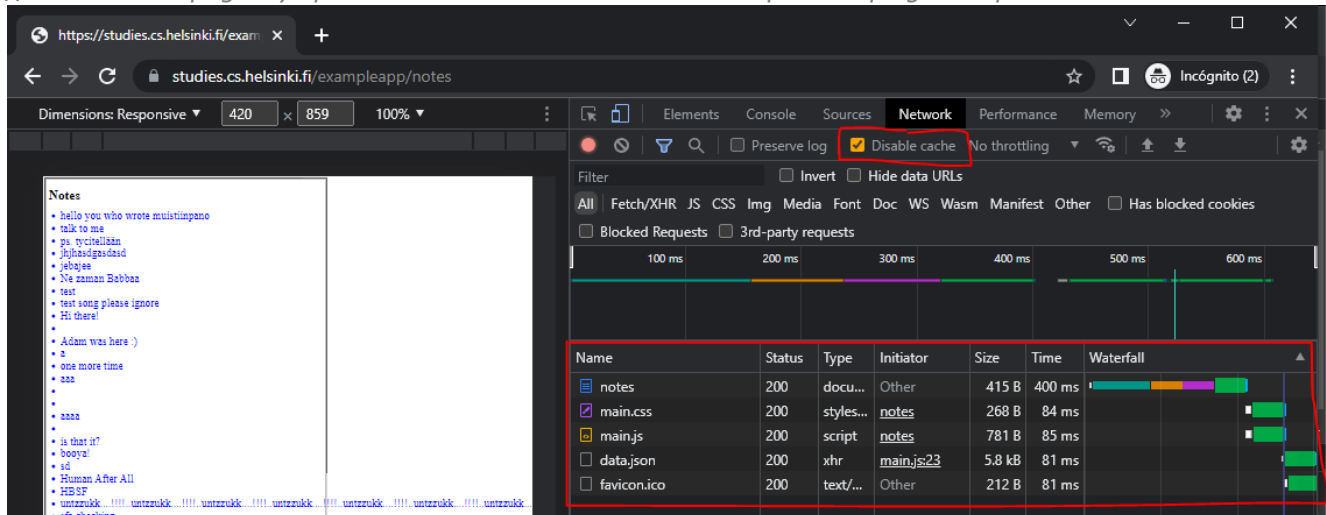
//enlace página web Bootcamp

<https://fullstackopen.com/>

CONSULTAR LOS RECURSOS QUE SE SOLICITAN PARA DESPLEGAR ESA APLICACIÓN WEB

*** pestaña Network ***

// Actualizar la página y aparecen todos los recursos solicitados para desplegar la aplicación web



*** status code ***

200 ok

201 Created

301 Cambia la página totalmente

302 Lo he encontrado y realiza redirección

400... errores (404 not found)

CONSULTAR AL DOM

*** pestaña Console ***

// Consultar cuantos links hay en la página

`document.querySelectorAll('a')`

// Consultar el primer elemento

`document.querySelector('a')`

// cambiar un elemento del DOM por otro tipo de elemento, es decir escribo HTML

`document.querySelector('a.home-article').innerHTML='Voy a cambiarlo por esto'`

// leer el contenido del texto de un elemento del DOM

`document.querySelector('a.home-article').textContent`

// cambio el texto de un elemento del DOM

`document.querySelector('a.home-article').textContent='Cambio el texto'`

// para saber cuanto mide en altura el primer elemento li (en pixeles)

`document.querySelector('li').clientHeight`

2- JAVASCRIPT:

JAVASCRIPT

```
*** imprimir por consola *** console.log()
console.log('Hola mundo')
```

CREAR VARIABLES

```
*** variables: let, const, var ***
```

```
// let: aconsejable
```

```
//const: constante
```

```
//var: NO aconsejable, metodo antiguo
```

//diferencia entre let y var, el valor de vara que se crea en un bloque sigue valiendo fuera del bloque, mientras que una variable let funciona dentro de un bloque sin seguir manteniendo su valor fuera de el

```
let firstName='Miguel'
const lastName=4
var isDeveloper=true
```

```
*** TIPADO DÉBIL Y DINAMICO ***
```

// Tipado débil: el tipo de variable puede cambiar a lo largo de la aplicación. Ejm: que inicialmente la variable sea un String y luego guardamos en ella un entero

// Dinámico: el tipo de variable puede cambiar a lo largo de la aplicación

```
let firstName='Miguel'
firstName=3
console.log(firstName)
```

```
*** tipo de variable ***
```

```
// tipos primitivos
```

```
let numero=2 //tipo numero
let string="String" //String
let boolean=true //booleano
let indefinido=undefined //indefinido
let nulo=null //nulo
let BigInt //BigInt
let Simbol//Simbol
```

STRING

```
*** toUpperCase ***
```

```
// cambiar un String a letras mayúsculas
```

// LOS String son INMUTABLES, es decir no puedes cambiar su valor, si puedes hacer copias cambiando su valor

// ejm, el siguiente código NO cambia el String a mayúsculas

```
let secondName='Miguel'
secondName.toUpperCase()
console.log(secondName)
```

```
// hay que guardarlo en otro String
```

```
let secondName='Miguel'
let secondNameMayus=secondName.toUpperCase()
console.log(secondNameMayus)
```

ARRAY

*** crear – añadir valores – acceder a valores ***

```
const list = []           // creo el array
list.push(157)            // introduzco valor al array
list[0]                   // accedo al valor del array (posición 0)
console.log(list[0])      // imprimo el valor del array
```

*** concat *** es decir, se pueden introducir diferentes tipos de variables dentro del Array

// este ejemplo no va a imprimir nada en consola (undefined) ya que lo que estamos introduciendo en realidad con concat es otra lista

```
const listMutando = []
listMutando.concat(157)
console.log(list[0])
```

// por lo tanto para imprimirlo tenemos que imprimir la lista

```
const listMutando = []
const anotherList=listMutando.concat(157)
console.log(anotherList)
```

OBJETOS

*** crear un objeto ***

```
const persona={
  name:'Miguel',
  twitter:'@midudev',
  age:18,
  isDeveloper:true,
  links: ['https://midu.tube','https://midu.live']
}
```

*** imprimir datos de un objeto ***

```
console.log(persona.name)
console.log(persona.links[0])
```

```
const field='twitter'
console.log(persona[field])
```

FUNCIONES

*** crear una función *** forma 1 (function expresion) a una constante asignarle una función

```
const sumar = (operando1, operando2) => {
  console.log(operando1)
  console.log(operando2)
  return operando1 + operando2
}

const resultado = sumar(2, 2)
console.log(resultado)
```

*** crear una función *** forma 2 declarando una función

```
function restar(a, b) {  
  return a - b  
}
```

COMPARACIONES

*** comparar sin tener en cuenta el tipo de dato *** ==

// este ejemplo dará -> true

```
2=="2"
```

*** comparar teniendo en cuenta el tipo de dato *** ===

// este ejemplo dará -> false

```
2==="2"
```

3- REACT:

INSTALAR REACT

*** primero es instalar nodejs ***

// descargar de la web

<https://nodejs.org/en/> // la LTS (Long Time Support)

// instalar el ejecutable

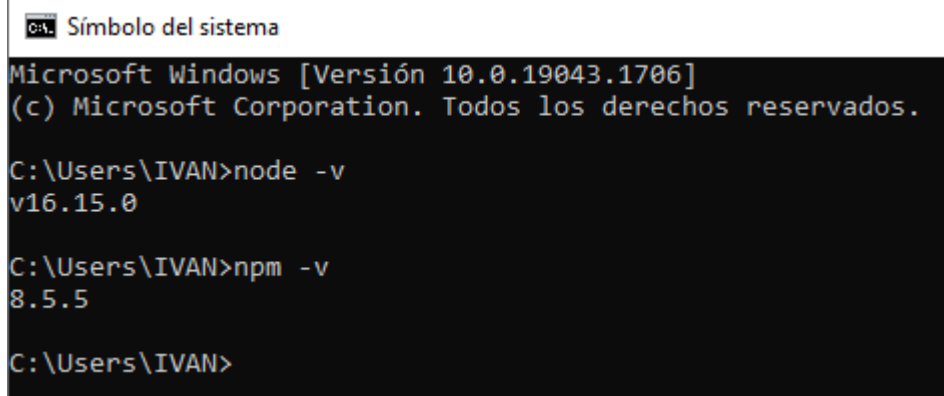
node-v16.15.0-x64.msi

// en la terminal de Windows (cmd)

cmd

node -v //comprobamos versión de node instalada

npm -v //comprobamos que npm también se ha instalado



```
Microsoft Windows [Versión 10.0.19043.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\IVAN>node -v
v16.15.0

C:\Users\IVAN>npm -v
8.5.5

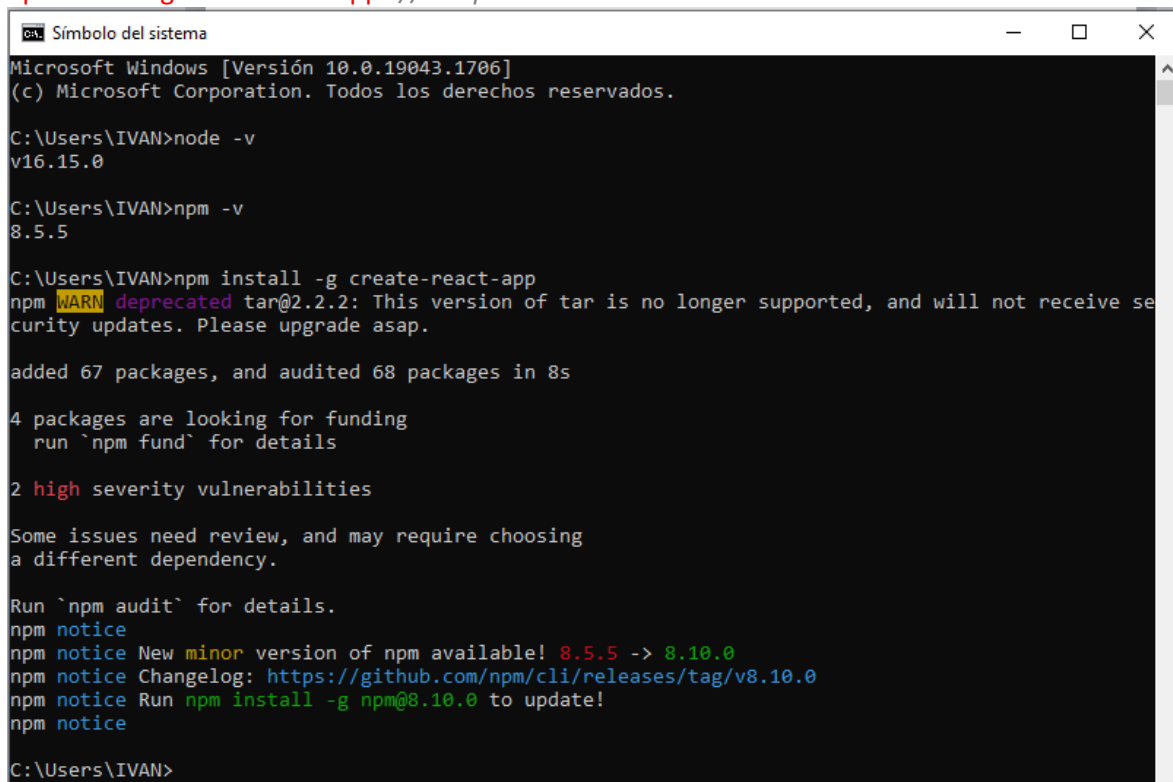
C:\Users\IVAN>
```

*** instalar react ***

// en la terminal de Windows (cmd)

cmd

npm install -g create-react-app //comprobamos versión de node instalada



```
Microsoft Windows [Versión 10.0.19043.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\IVAN>node -v
v16.15.0

C:\Users\IVAN>npm -v
8.5.5

C:\Users\IVAN>npm install -g create-react-app
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

added 67 packages, and audited 68 packages in 8s

4 packages are looking for funding
  run `npm fund` for details

2 high severity vulnerabilities

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.5.5 -> 8.10.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.10.0
npm notice Run npm install -g npm@8.10.0 to update!
npm notice
C:\Users\IVAN>
```

CREAR UN PROYECTO

// en la terminal de Windows (cmd), entrar en la carpeta donde quiero realizar el proyecto

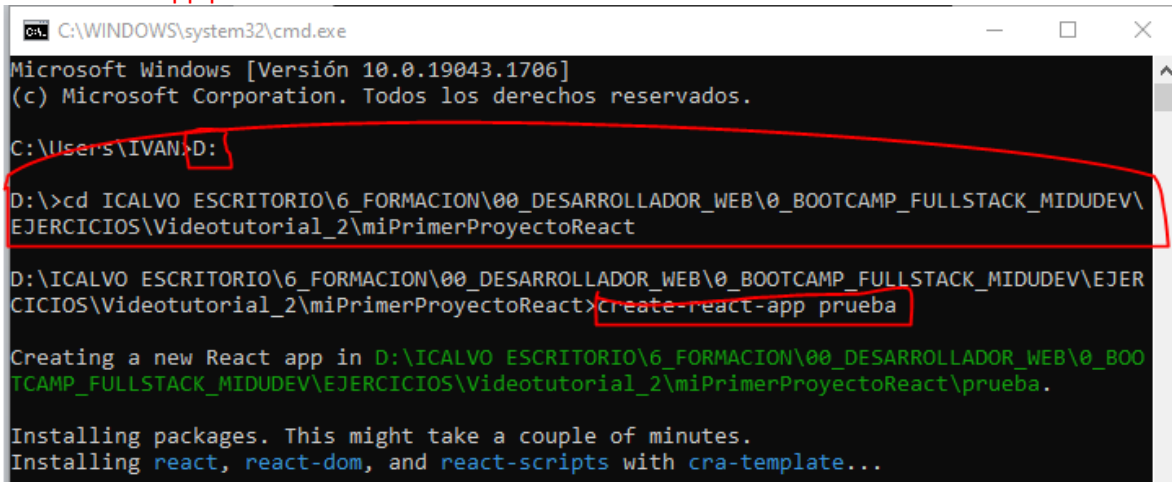
D:

cd ICALVO

ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Vi
deotutorial_2\miPrimerProyectoReact

// creo el proyecto

create-react-app prueba



A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt shows the user navigating to a directory and running the "create-react-app prueba" command. The output indicates that a new React app is being created in the specified directory and that packages (react, react-dom, react-scripts) are being installed. Red boxes highlight the directory path and the command.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

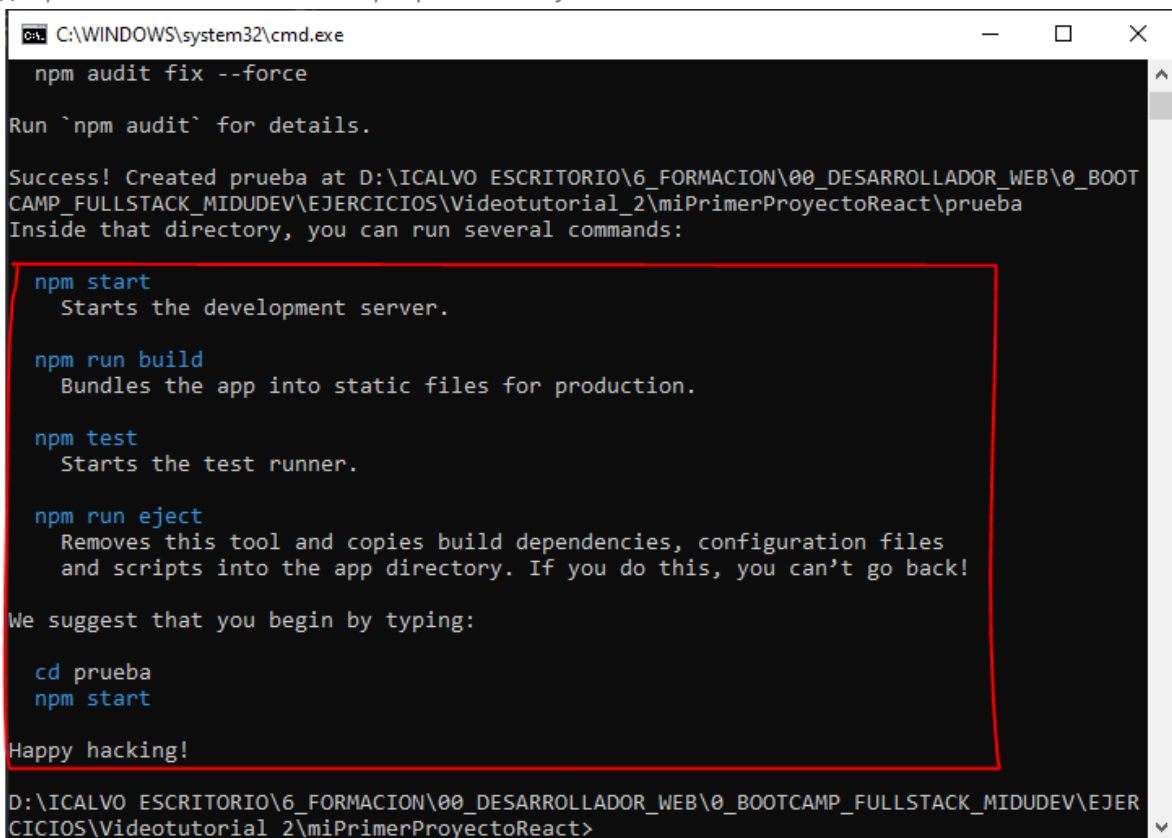
C:\Users\IVAN>D:
D:\>cd ICALVO ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Vi
deotutorial_2\miPrimerProyectoReact
D:\ICALVO ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Vi
deotutorial_2\miPrimerProyectoReact>create-react-app prueba

Creating a new React app in D:\ICALVO ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Vi
deotutorial_2\miPrimerProyectoReact\prueba.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

// (tarda un poco en crear el proyecto)

// aparecen comandos con los que poder trabajar



A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt shows the output of the "create-react-app prueba" command, including a list of commands that can be used to manage the app. A red box highlights the list of commands.

```
C:\WINDOWS\system32\cmd.exe
npm audit fix --force

Run `npm audit` for details.

Success! Created prueba at D:\ICALVO ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Vi
deotutorial_2\miPrimerProyectoReact\prueba
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 prueba
  npm start

Happy hacking!

D:\ICALVO ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Vi
deotutorial_2\miPrimerProyectoReact>
```

INICIALIZAR EL PROYECTO

// entro dentro de la carpeta donde he generado los archivos react

cd prueba

// inicializar el proyecto react

npm start

// se abre en el explorador (permitir acceso de red)

// informa del localhost:3000

```
C:\> Windows PowerShell

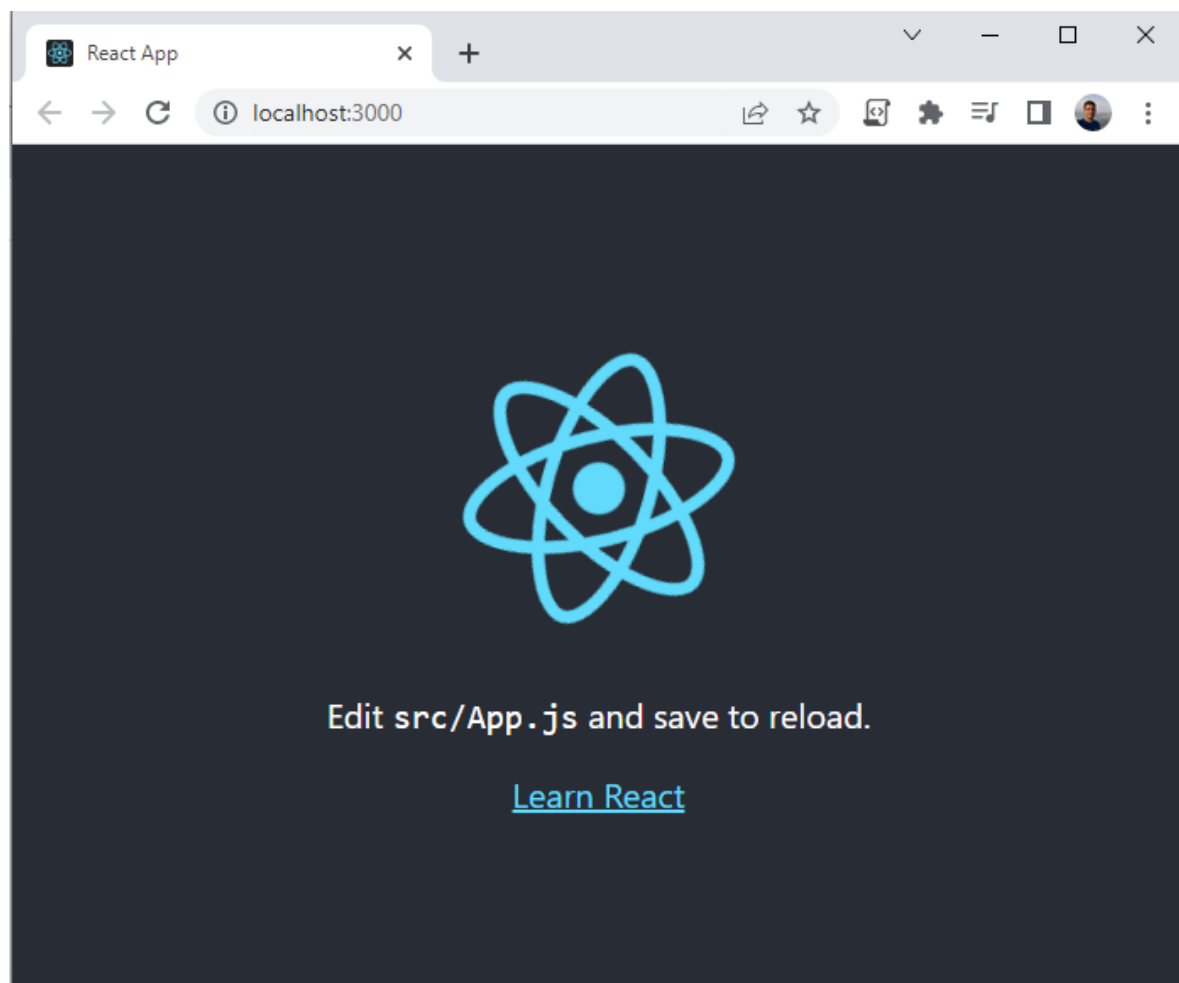
Compiled successfully!

You can now view prueba in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.56.1:3000

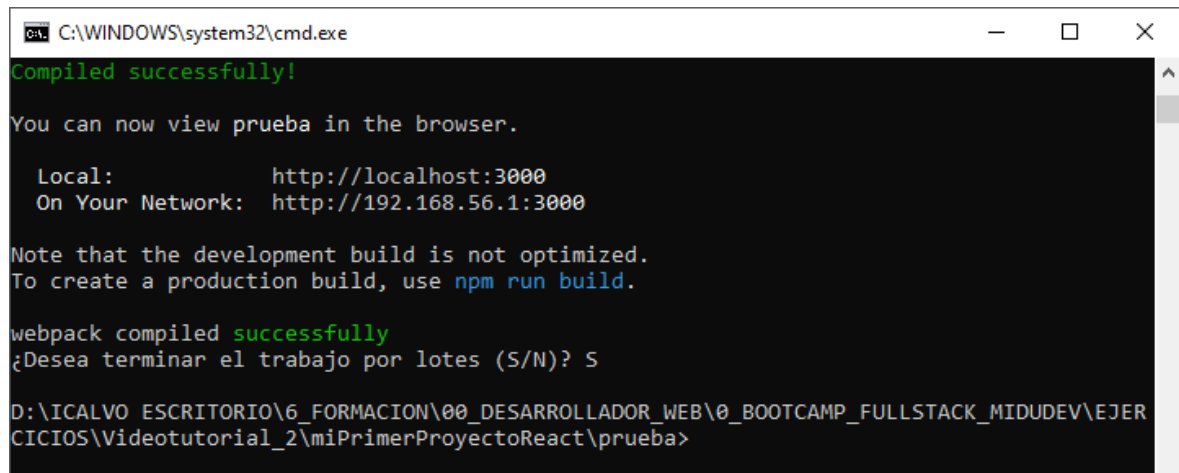
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



// para pausar el proyecto desde la terminal

Ctrl+C



```
C:\WINDOWS\system32\cmd.exe

Compiled successfully!

You can now view prueba in the browser.

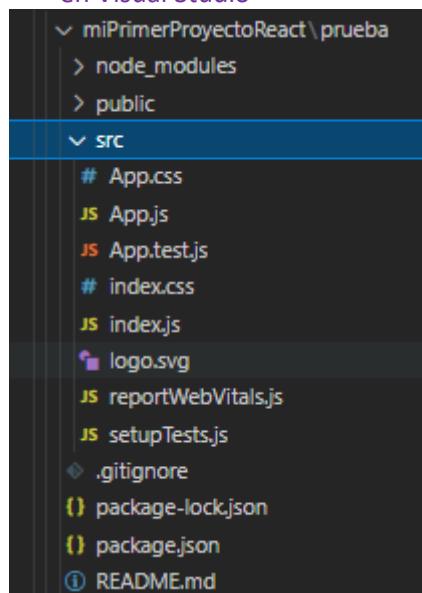
  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
¿Desea terminar el trabajo por lotes (S/N)? S

D:\ICALVO ESCRITORIO\6_FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\Videotutorial_2\miPrimerProyectoReact\prueba>
```

*** en Visual Studio ***



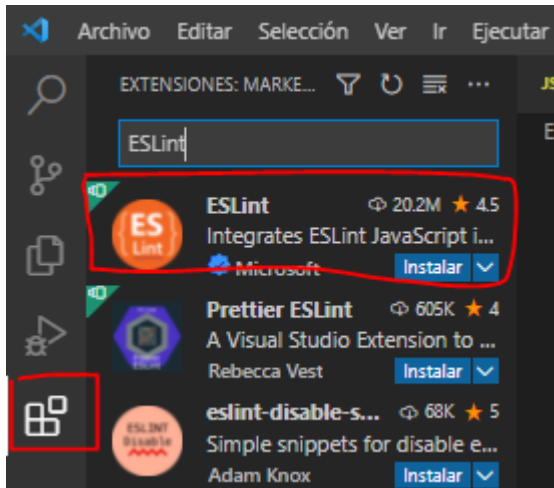
// ejecutar el proyecto desde la Terminal de Visual Studio (Ctrl+J para mostrar la terminal)

Ctrl+J

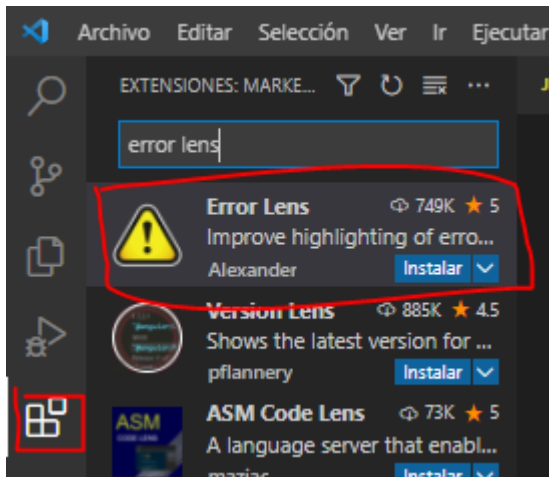
EXTENSIONES

*** EXTENSIONES VISUAL STUDIO ***

ESLint



error lens

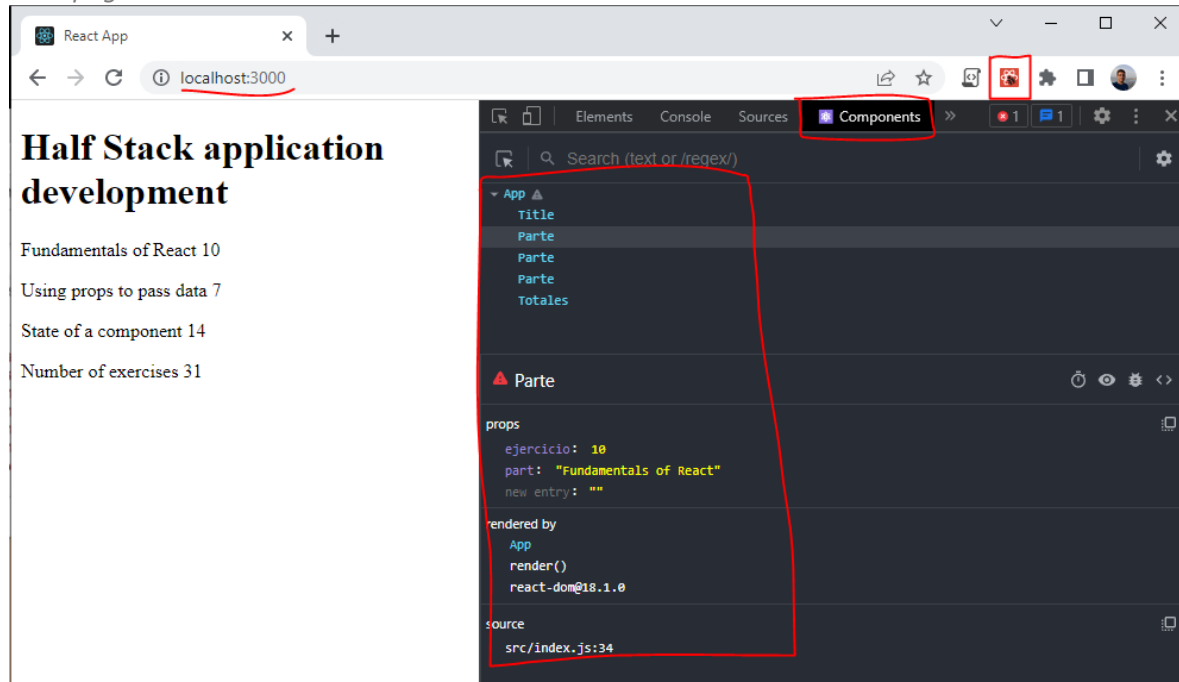


*** EXTENSIONES CHROME ***

// instalar la extensión para Chrome

React Developer Tools

// activar la extensión y estando dentro de una página creada con React puedo ver los Components y Profiles de la página



EJERCICIO 1

*** crear proyecto React e iniciarlo ***

// abrir terminal (cmd) dentro de la carpeta creo el proyecto "courseinfo"

create-react-app courseinfo

// entro dentro de la carpeta donde he generado los archivos react

cd courseinfo

// inicializar el proyecto react

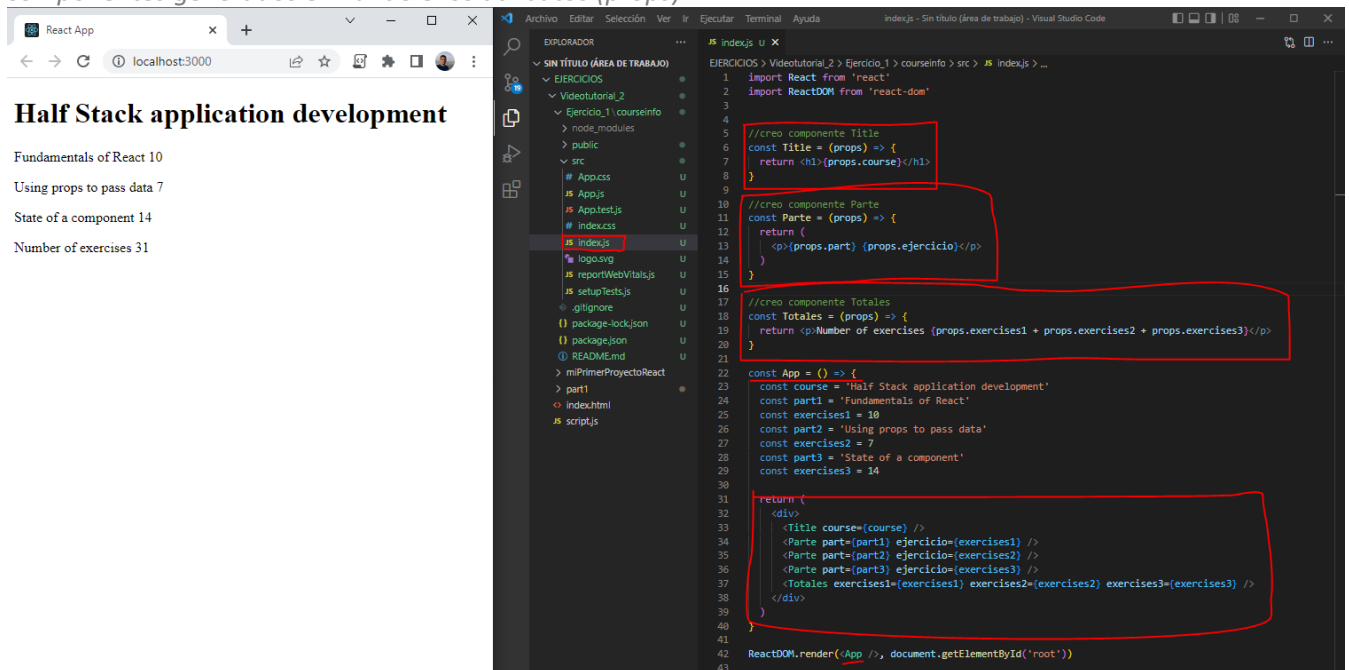
npm start

*** modificar el archivo src/index.js ***

// crear los componentes

```
const Title = (props) => {  
  return <h1>{props.course}</h1>  
}
```

// dentro del compoennte App, declaro en return lo que quiero imprimir en pantalla, en este caso los componentes generados enviandole los atributos (props)



FUNCIONES y PROPS

*** declarar una función ***

// ListOfClicks es el nombre de la función

// {clicks} es el parámetro que se le pasa a la función

```
const ListOfClicks({clicks})=>{  
  
}
```

// props son equivalentes a los atributos de una función

*** declarar props (atributos) ***

Counter // es el nombre de la función

contador // es el nombre de la props (atributo)

{contadorValue} // es el valor de la props (en este caso es una variable anteriormente declarada)

```
40 <p>El valor del contador es:</p>  
41 <Counter contador={contadorValue}/>  
42 <p>{mensajePar}</p>
```

*** recibir props (atributos) en una Función ***

Counter // declaro la función

props // indico que lleva props (atributos)

props.contador // recibo en la función el valor que tiene "contador"

```
12 const Counter = (props) =>{  
13   return <h1>{props.contador}</h1>;  
14 };
```

*** recibir props (atributos) en una Función de forma simplificada ***

// cuando se pasa un solo props (atributo) para simplificar y no tener que poner "props"

{contador} // indico el nombre de la props (atributo)

```
18 const Counter = ({contador}) =>{  
19   return <h1>{contador}</h1>;  
20 };
```

ARROW FUNCTION (SIMPLIFICAR FUNCIONES)

// Arrow functions are a new way to write anonymous function expressions

*** partiendo de una función convencional ***

// Nombrando una función convencional

```
export function getNumber() {  
  return 2  
}
```

// Pasamos a una Arrow Function

```
export const getNumber = () => 2;
```

*** simplificar una Arrow Function ***

```
{notes.map((note) => {  
  return (<div><p>{note.id}{note.date}{note.content}</p></div>)  
}}}
```

// mejorar el código y hacerlo más legible

//CUANDO SOLO HAY UN PARÁMETRO se pueden eliminar los paréntesis del parámetro

```
{notes.map(note => {  
  return (<div><p>{note.id}{note.date}{note.content}</p></div>)  
}}}
```

//CUANDO SE QUIERE DEVOLVER LO PRIMERO, se pueden eliminar las llaves y el return creando un return implícito

```
{notes.map(note =>  
  (<div><p>{note.id}{note.date}{note.content}</p></div>)  
)}
```

OBJETOS

*** crear un objeto con estado***

// genero el objeto "counters", left, right y mensaje son sus atributos

```
21 //creo un objeto "counters" y le indico un estado
22 const [counters, setCounters] = useState({
23   left: 0,
24   right: 0,
25   mensaje: 'Mensaje en el estado'
26 });
```

// genero un método que devuelve "setCounters" los atributos del objeto actualizado

...counters, // devuelve todos los atributos con el valor actual para no perder su valor

```
31 const handleClickLeft = () => {
32   setCounters({
33     ...counters,
34     left: counters.left + 1,
35   });
36 }
```

ARRAYS

*** longitud de un array *** `.length`

// "clicks" es un array

```
<p>Clicks totales:{clicks.length}</p>
```

*** unir carácter al final del array *** `.join`

// "clicks" es un array

```
{clicks.join(",")}
```

*** filtrar *** `filter`

// método de array para filtrar elementos de un array, devuelve un array con los elementos que contiene lo indicado. En este caso filtramos todos los que sean el String 'L' o 'R'

```
const left = clicks.filter(click => click === 'L');  
const right = clicks.filter(click => click === 'R');
```

// luego se puede calcular el número de elementos que contiene ese String

```
{left.length}
```


CONSOLE.LOG y DEBUGGER

*** console.log ***

// imprime por Consola el valor que le indico

```
console.log({clicks});
```

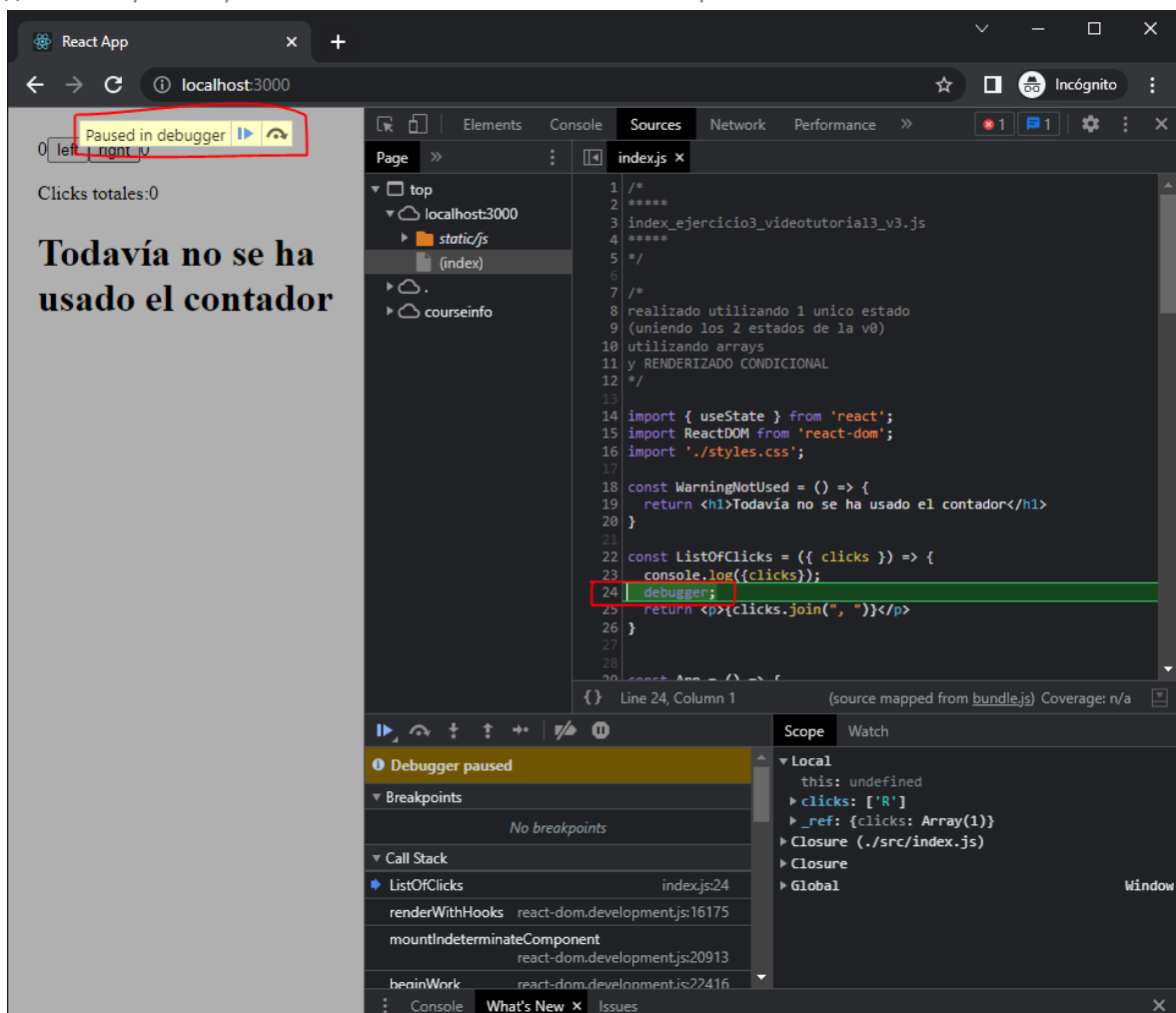
// cuando se quiere loggear las props, las podemos imprimir por consola ({props})

```
export const Note = (props) => {  
  console.log({props})  
}
```

*** debugger ***

// para la ejecución cuando llega a esa línea

// una vez parado puedo ver los valores de cada elemento posicionando el raton sobre el

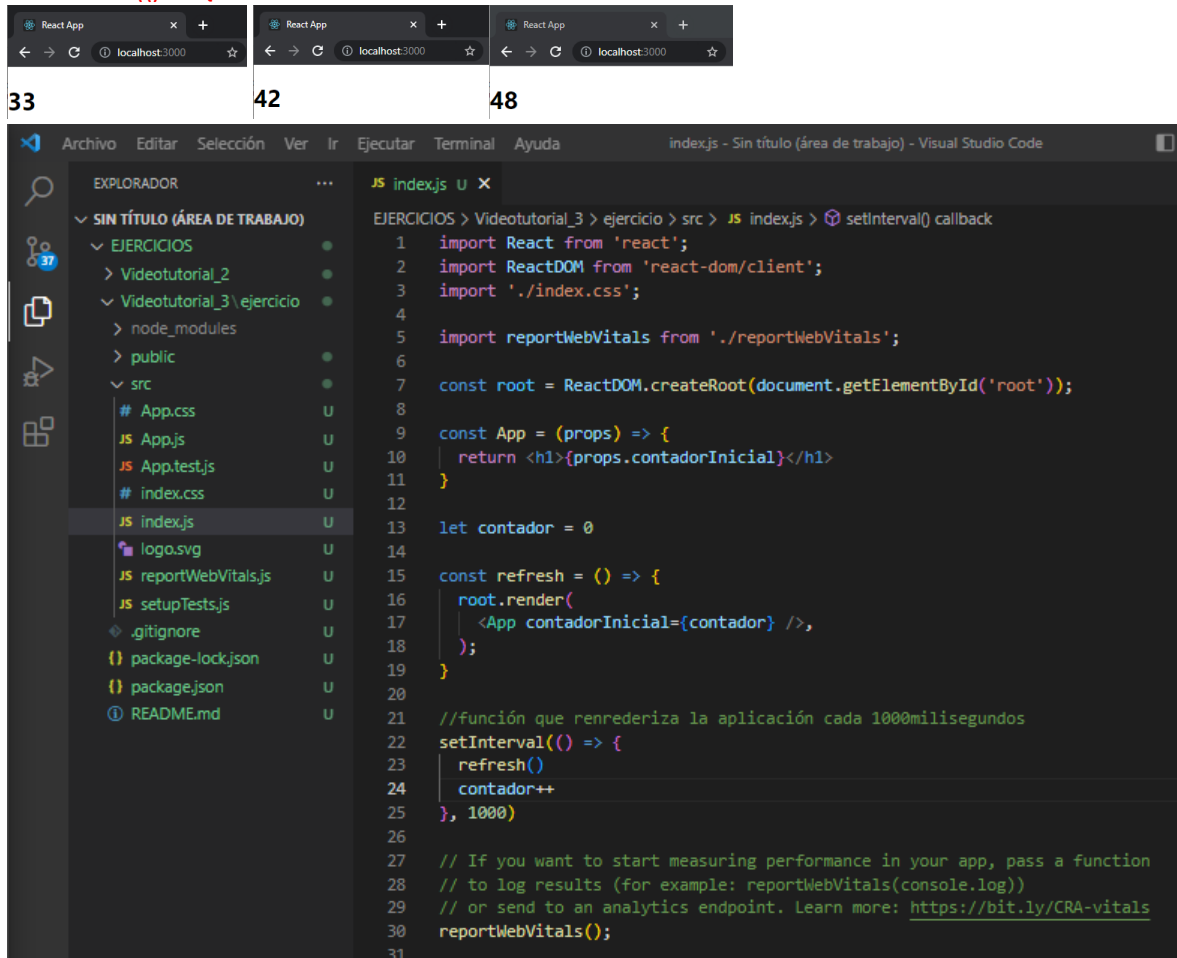


ESTADO DEL COMPONENTE, RENDERIZADO CONDICIONAL Y EVENTOS

*** Realizar una página dinámica ***

// función setInterval podemos refrescar el DOM cada ciertos milisegundos

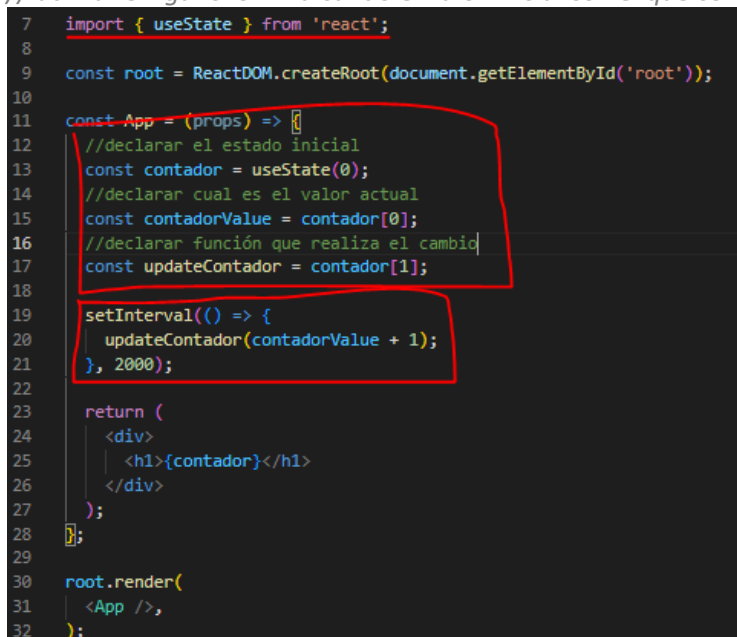
setInterval(() => {



*** Estados de un componente ***

// importar el useState y utilizarlo dentro del componente

// utilizar el "gancho" indicando el valor inicial con el que comienza, valor y metodo de actualizacion

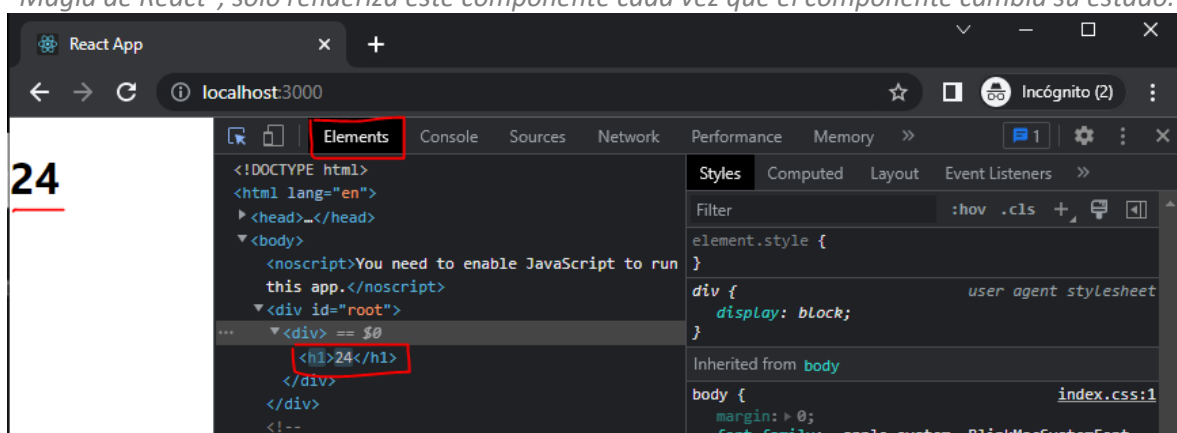


// useState devuelve un array de 2 elementos, es decir contador es un array de 2 elementos

```

11 const App = (props) => {
12
13   const [contadorValue, updateContador] = useState(0);
14
15   setInterval(() => {
16     updateContador(contadorValue + 1);
17   }, 2000);
18
19   return (
20     <div>
21       <h1>{contadorValue}</h1>
22     </div>
23   );
24 };
```

// en la aplicación se puede ver como React solo actualiza los elementos modificados y no todo el DOM, es la "Mágia de React", solo renderiza este componente cada vez que el componente cambia su estado.



*** Reseteo un elemento a su estado inicial ***

// realizo una constante "INITIAL_STATE" con los valores iniciales requeridos

// fuera de App

```

const INITIAL_COUNTER_STATE = {
  left: 0,
  right: 0,
  mensaje: 'Mensaje en el estado'
};
```

// dentro de App, creo el elemento con estado inicial la Constante

```
const [counters, setCounters] = useState(INITIAL_COUNTER_STATE);
```

// genero una función que resetea "counters" a la Constante

```

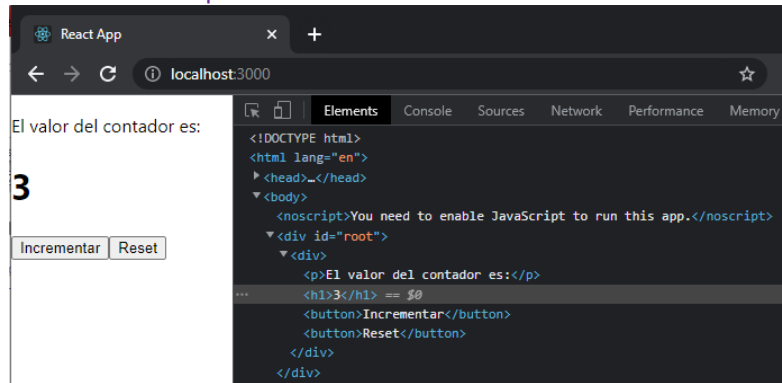
const handleReset = () => {
  setCounters(INITIAL_COUNTER_STATE);
};
```

// genero un botón que al pulsarlo ejecuta la función de reseteo

```
<p><button onClick={handleReset}>Reset</button></p>
```

EJERCICIO 2

Crear un boton que actualiza un contador



```
12 const App = (props) => {
13   //utilizo el estado
14   const [contadorValue, updateContador] = useState(0);
15
16   const handleClick = () => {
17     updateContador(contadorValue + 1);
18   }
19
20   const handleClickReset = () => {
21     updateContador(0)
22   }
23
24   return (
25     <div>
26       <p>El valor del contador es:</p>
27       <h1>{contadorValue}</h1>
28       <button onClick={handleClick}>Incrementar</button>
29       <button onClick={handleClickReset}>Reset</button>
30     </div>
31   );
32 };
```

RENDERIZAR UNA COLECCIÓN DE ELEMENTOS (LISTAS) CON REACT

// Renderizar una lista de elementos

// teniendo una lista de objetos:

```
const notes = [
  {
    id: 1,
    content: "HTML is easy",
    date: "2019-05-30T17:30:31.0982",
    important: true
  },
  {
    id: 2,
    content: "Browser can execute only JavaScript",
    date: "2019-05-30T18:39:34.091Z",
    important: false
  },
  {
    id: 3,
    content: "GET and POST are the most important methods of",
    date: "2019-05-30T19:20:14.298Z",
    important: true
  }
];
```

// forEach() se itera una lista pero no devuelve nada, no hace return

// por tanto utilizamos un "map" en cada iteración se puede cambiar el valor y además está devolviendo cada uno de los elementos transformados, por tanto

notes.map((note) // hago el map a la lista "notes" a cada elemento (objeto) de la lista lo llamo "note"

=> // genero una función que devuelve un atributo del objeto "id"

```
export default function App() {
  return (
    <div>
      {notes.map((note) => {
        return <p><strong>{note.id}</strong></p>
      })}
    </div>
  );
}
```

*** modificar la info de un array o listas *** prevNotes

// dos formas de añadir nuevos elementos a un array o List pero tienen una pequeña diferencia importante

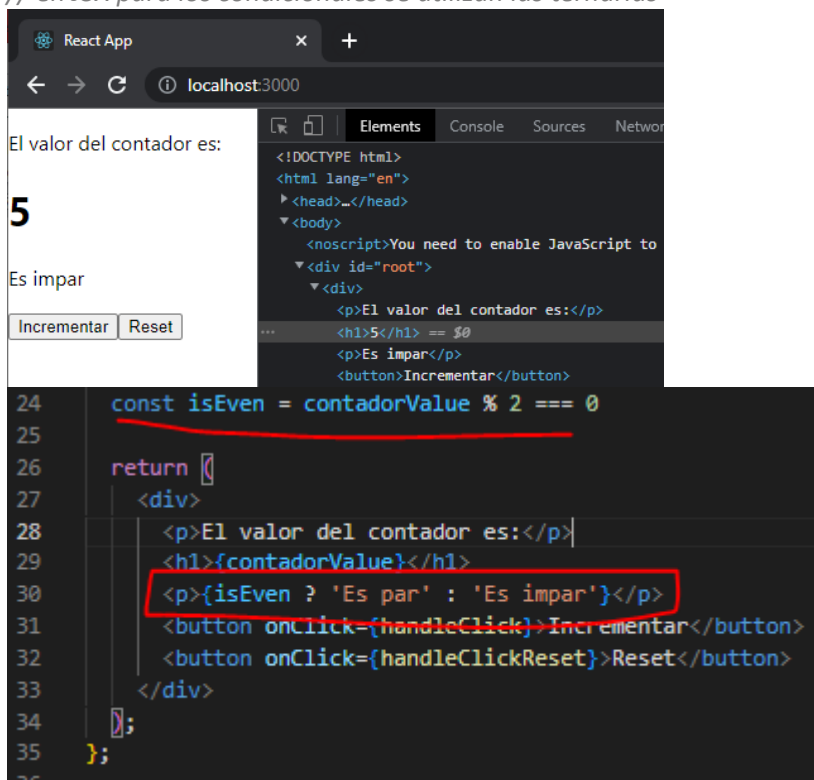
setNotes(notes.concat(newNote)); // en este caso "notes" vale lo que tenga guardado la variable

setNotes((prevNotes) => prevNotes.concat(newNote)); // en este caso ejecuta teniendo en cuenta los valores que tiene "notes" aunque no se haya modificado el estado de notes, por lo que es mejor utilizarlo con "prev"

```
setNotes(notes.concat(newNote));
setNotes((prevNotes) => prevNotes.concat(newNote));
```

*** RENDERIZADO CONDICIONAL ***

// dependiendo de una condición queremos renderizar una cosa u otra
 // en JSX para los condicionales se utilizan las ternarias



// ejemplo Renderizado Condicional

// creo dos funciones (código fuera de App) "WarningNotUsed" y "ListOfClicks"

```

const WarningNotUsed = () => {
  return <h1>Todavía no se ha usado el contador</h1>
}

const ListOfClicks = ({ clicks }) => {
  return <p>{clicks.join(", ")}</p>
}
```

// creo la condición ternaria, dentro de App en return

// si a longitud del array "clicks es igual a 0" ejecuta la función "WarningNotUsed" sino "ListOfClicks"

```
{clicks.length === 0 ? <WarningNotUsed /> : <ListOfClicks clicks={clicks} />}
```

*** IMPORTANTE PARA RENDERIZAR COLECCIÓN DE ELEMENTOS (LISTAS) EN REACT v18 ***

```
✖ ▶Warning: ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createroot
```

```
import React from 'react';
import * as ReactDOMClient from 'react-dom/client';
import './styles.css';
```

```
export default function App() {

}
```

```
const rootElement = document.getElementById('root');
const root = ReactDOMClient.createRoot(rootElement);

root.render( // deprecated starting React 18
  <React.StrictMode>
    <App />
  </React.StrictMode>,
);
```

*** Renderizar una colección con condicion ***

// evalua si la lista "notes" no es "undefined" y si contiene elementos o no

```
export default function App() {
  if (typeof notes === "undefined" || notes.length === 0) {
    return "No tenemos notas que mostrar";
  }

  return (
    <div>
      {notes.map((note) => {
        return
        <div><p><strong>{note.id}</strong><small><time>{note.date}</time></small>{note.content}</p></div>
      })}
    </div>
  );
}
```


LISTAS KEY

// Al renderizar una lista si no se utilizan las Key nos sale una advertencia:

```
✖ Warning: Each child in a react-jsx-dev-runtime.development.js:97
  list should have a unique "key" prop.

  Check the render method of `App`. See https://reactjs.org/link/warning-keys
  for more information.
    at div
    at App
```

// Las keys se utiliza para dar una pista a React para que cada vez que se renderiza App tenga que realizar el mínimo trabajo posible y saber donde tiene que conservar ese componente al hacer la iteración. Por lo tanto la Key tiene que ser único para cada elemento

// en el primer elemento que se renderiza se le identifica la key, en este caso puedo pasar la "id" ya que va a ser única

`<div key={note.id}...`

```
//creo un array de objetos
const notes = [
  {
    id: 1,
    content: "HTML is easy",
    date: "2019-05-30T17:30:31.0982",
    important: true
  },
  {
    id: 2,
    content: "Browser can execute only JavaScript",
    date: "2019-05-30T18:39:34.091Z",
    important: false
  },
  {
    id: 3,
    content: "GET and POST are the most important methods of",
    date: "2019-05-30T19:20:14.298Z",
    important: true
  }
];

export default function App() {
  return (
    <div>
      {notes.map(note =>
        (<div key={note.id}><p>{note.id}{note.date}{note.content}</p></div>)
      )}
    </div>
  );
}
```

// eliminamos el "Warning"

EXPORTAR MODULOS

// Consiste en extraer a un fichero los módulos para tenerlos ordenados en diferentes archivos

// Crear un nuevo fichero

Note.js

```
JS Note.js
```

// ahí creo el componente y lo exporto

*** Exportación por defecto ***

export default Note // Exporta por defecto el componente Note

```
JS Note.js U ●
ejercicios_React > src > JS Note.js > [⌘] default
1  const Note=({id,content,date})=>{
2    //console.log(note);
3    return (<li><p><small><time>{date}</time></small></p><p>{content}</p></li>)
4  }
5
6  // con default lo exportamos pero sin nombre
7  //donde lo importamos podemos darle el nombre que queramos
8  export default Note;
```

import Note from './Note.js' // Importo poniendole el nombre que queramos indicándole de donde tiene que traer el módulo (se le llama importar MODULO)

```
//importamos Note desde Note.js
```

```
import Pepito from './Note.js'
```

*** Exportación nombrada ***

export // lo exportamos con su nombre

```
JS Note.js U ●
ejercicios_React > src > JS Note.js > [⌘] Note
1  //con export exportamos el elemento de forma "nombrada"
2  //es decir lo exportamos con su nombre
3  export const Note=({id,content,date})=>{
4    //console.log(note);
5    return (<li><p><small><time>{date}</time></small></p><p>{content}</p></li>)
6  }
```

import {Note} from './Note.js' // Importo poniendole el nombre entre llaves

```
//importamos Note desde Note.js
```

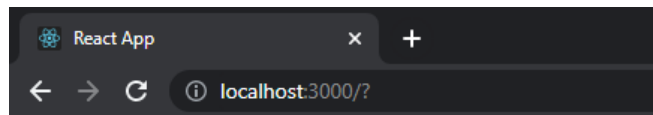
```
import {Note} from './Note.js'
```

import {Note as Pepito2} from './Note.js' // de esta forma también se le puede cambiar el nombre

```
//importamos Note desde Note.js
```

```
import {Note as Pepito2} from './Note.js'
```

FORMULARIOS EJERCICIO 4



Notes

Show all

1. 2019-05-30 17:30:31

1 - HTML is easy

2. 2019-05-30 18:39:34

2 - Browser can execute only JavaScript

3. 2019-05-30 19:20:14

3 - GET and POST are the most important methods of

Crear nota

*** index.js ***

```
JS index.js
import App from './App.js'; // importar la Aplicación
const notes = [ //declaro un array de objetos
<App notes={notes}/> //envío mediante props las notas generadas

import React from 'react';
import * as ReactDOMClient from 'react-dom/client';
import './styles.css';
import App from './App.js';

//creo un array de objetos
const notes = [
  {
    id: 1,
    content: "HTML is easy",
    date: "2019-05-30 17:30:31",
    important: true
  },
  {
    id: 2,
    content: "Browser can execute only JavaScript",
    date: "2019-05-30 18:39:34",
    important: false
  },
  {
    id: 3,
    content: "GET and POST are the most important methods of",
    date: "2019-05-30 19:20:14",
    important: true
  }
];
```

```
    }  
  ];  
  
  const rootElement = document.getElementById('root');  
  const root = ReactDOMClient.createRoot(rootElement);  
  
  root.render( // deprecated starting React 18  
    <React.StrictMode>  
      <App notes={notes}/>  
    </React.StrictMode>,  
  );
```

*** Note.js ***

JS Note.js

// archivo donde declaro el elemtno Note que devuelve una sentencia html con los datos pasados por parámetros

```
//con export exportamos el elemento de forma "nombrada"  
//es decir lo exportamos con su nombre  
export const Note = ({ id, content, date }) => {  
  //console.log(note);  
  return (<li><p><small><time>{date}</time></small></p><p>{id}{ " -  
"}{content}</p></li>  
)  
}  
  
// con default lo exportamos pero sin nombre  
//donde lo importamos podemos darle el nombre que queramos  
export default Note;
```

*** App.js ***

JS App.js

```
import { useState } from 'react'; // importar el Hook de estado "useState"  
import { Note } from './Note.js'; // importar el elemento Note creado en un archivo externo  
export default function App(props) { // exporto la aplicación que trabaja con las props que se envían desde  
  index.js (son las "notes")  
  const [notes, setNotes] = useState(props.notes); // creo un estado "notes" será un array ya que el estado  
  inicial que devuelve son las props.notes que es un array  
  const handleChange = (event) => // creo un evento al cambiar el texto del cuadro de texto que recoge el texto  
  introducido y mediante setNewNote cambia el estado de newNote  
  const handleSubmit = (event) => { // al enviar el formulario  
    event.preventDefault(); // quito la acción por defecto que tiene el formulario al enviar que es de actualizar la  
    página  
    const noteToAddToState // creo un nuevo objeto del tipo "notes"  
    setNotes(notes.concat(noteToAddToState)); // añado este nuevo objeto al array inicial "notes"  
  }  
  import './styles.css';  
  import { useState } from 'react';  
  import { Note } from './Note.js';  
  
  //hago que las Notas por props
```

```
export default function App(props) {
  const [notes, setNotes] = useState(props.notes);
  const [newNote, setNewNote] = useState('');
  const [showAll, setShowAll] = useState(true);

  const handleChange = (event) => {
    //console.log(event.target.value)
    setNewNote(event.target.value)
  }
  const handleSubmit = (event) => {
    /*para evitar el comportamiento por defecto que tiene el Submit
    de actualizar la página cada vez que lo ejecutamos */
    event.preventDefault();
    console.log("crear nota");
    console.log(newNote)

    const noteToAddToState = {
      id: notes.length + 1,
      content: newNote,
      date: new Date().toISOString(),
      important: Math.random() < 0.5
    };

    console.log(noteToAddToState);

    /*para modificar un array hay que crear uno nuevo*/
    /*al array le concatenamos el nuevo elemento añadir*/
    setNotes(notes.concat(noteToAddToState));
    /*como tengo el control del input*/
    /*he colocado en el input el value, puedo borrar su valor*/
    setNewNote("");
  }

  const handleShowAll = () => {
    setShowAll(() => !showAll);
  }

  if (typeof notes === "undefined" || notes.length === 0) {
    return "No tenemos notas que mostrar";
  }

  return (
    <div>
      <h1>Notes</h1>
      <button onClick={handleShowAll}>{showAll ? 'Show all' : 'Show only
important'}</button>
      <ol>
        {notes
          /* El filtro devuelve un booleano*/
          .filter((note)=>{
```

```
        if(showAll===true) return true;
        return note.important===true;
    })
    .map(note => <Note key={note.id} id={note.id} content={note.content}
date={note.date} />))
    </ol>

    <form onSubmit={handleSubmit}>
        <input type='text' onChange={handleChange} value={newNote} />
        <button>Crear nota</button>
    </form>
</div>
);
}
```

RECUPERAR DATOS DEL SERVIDOR – FETCHING – EJERCICIO 5

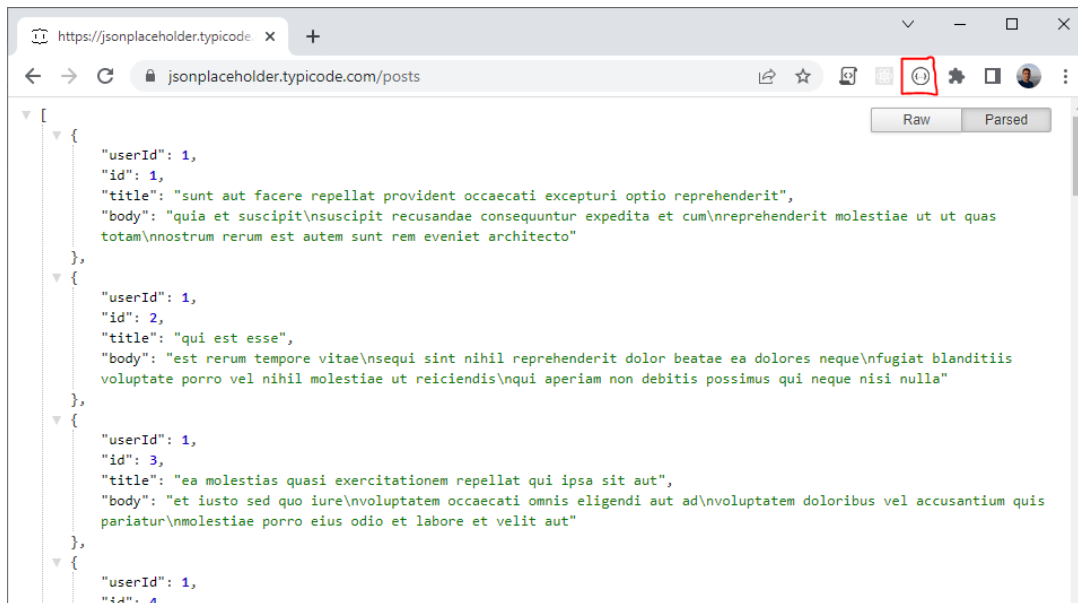
*** instalar JSON Formatter ***

// para poder ver adecuadamente página de datos JSON formateada



// página json con la que trabajar

// jsonplaceholder.typicode.com/posts

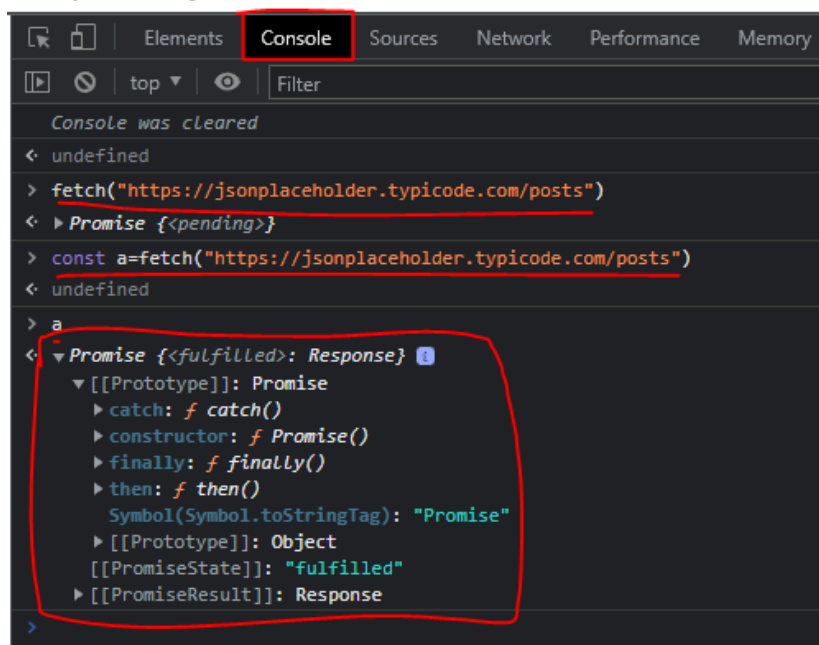


*** recuperar información de una API externa *** fetch

// utilizando un "fetch" método que permite recuperar datos a partir de una dirección

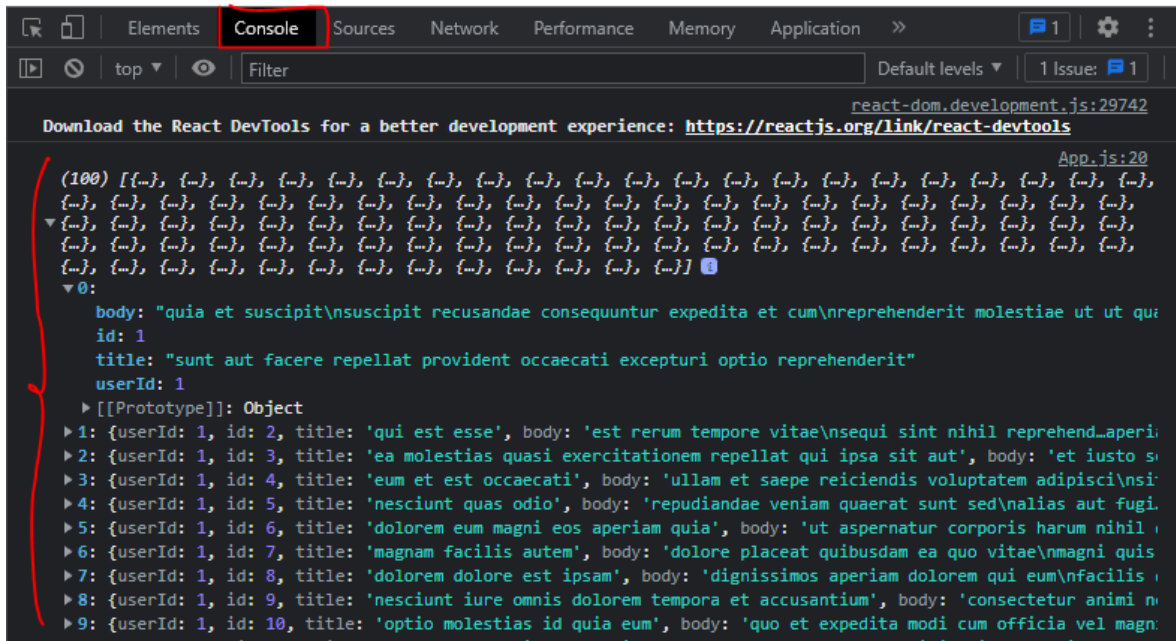
// "fetch" nos devuelve una "promesa" es un objeto que guarda un valor futuro (es decir se queda pendiente haciendo cosas y se resolverá en un futuro de forma correcta o de forma negativa)

// el "fetch" se guarada en una variable



// se hace el "fetch" y una vez tengas la respuesta ".then" entonces transformas la respuesta a un JSON

```
fetch('https://jsonplaceholder.typicode.com/posts')
  .then((response)=>response.json())
  .then((json)=>{
    console.log(json);
  });
```



*** recuperar información una sola vez, no generar un bucle infinito cada vez que se renderiza la página ***
useEffect

// "useEffect" es un "Hook" que se ejecuta cada vez que se renderiza nuestro componente, controlando cuantas veces se tiene que renderizar

```
import { useEffect, useState } from 'react';
```

// para controlar cuando se tiene que ejecutar el "useEffect" utilizamos las dependencias

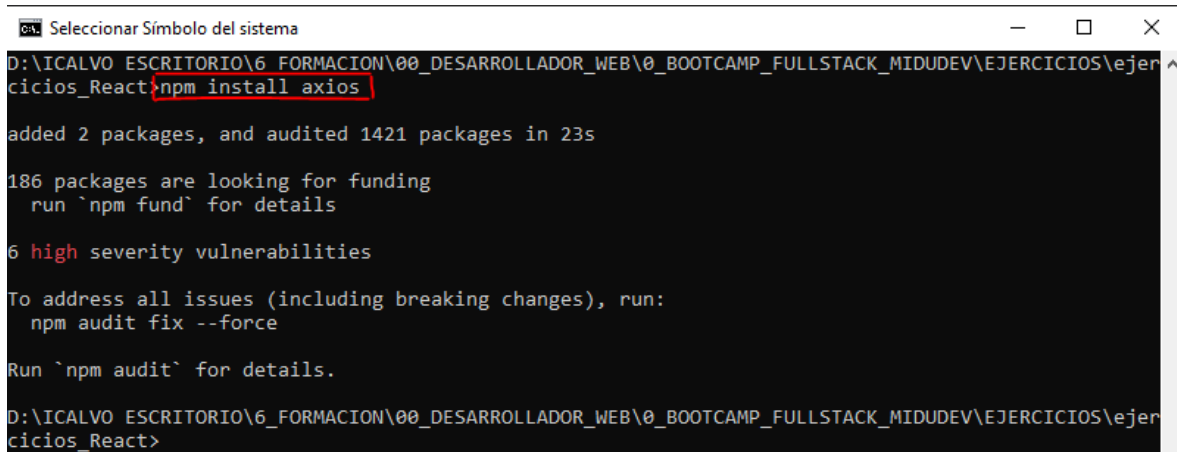
,[] // indicándole que realice el "useEffect" solo cuando se modifique la dependencia en este caso es un array vacío por lo tanto solo se ejecutará una vez

```
useEffect(() => {
  fetch('https://jsonplaceholder.typicode.com/posts')
    .then((response) => response.json())
    .then((json) => {
      setNotes(json)
    });
}, []);
```


*** Axios ***

// "axios" es un "grapper" ("pinza") de "fetch" mismas funciones que "fetch" pero con otras ventajas.

Npm install axios // instalar en cmd



```
ca. Seleccionar Símbolo del sistema
D:\ICALVO ESCRITORIO\6 FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\ejercicios_React>npm install axios
added 2 packages, and audited 1421 packages in 23s
186 packages are looking for funding
  run `npm fund` for details
6 high severity vulnerabilities
To address all issues (including breaking changes), run:
  npm audit fix --force
Run `npm audit` for details.
D:\ICALVO ESCRITORIO\6 FORMACION\00_DESARROLLADOR_WEB\0_BOOTCAMP_FULLSTACK_MIDUDEV\EJERCICIOS\ejercicios_React>
```

```
import axios from 'axios';
```

// en "axios" no hace falta hacer la transformación a .json como en "fetch", lo realiza automáticamente

// utilizando "axios" en vez de "fetch":

```
useEffect(() => {
  setLoading(true);
  setTimeout(() => {
    axios
      .get('https://jsonplaceholder.typicode.com/posts')
      .then((response) => {
        const { data } = response;
        setNotes(data);
        setLoading(false);
      });
  }, 2000);
}, []);
```

*** modificar los datos *** post

// genero un elemento objeto tipo del que vamos a introducir

```
const noteToAddToState = {
  title: newNote,
  body: newNote,
  userId: 1
};
```

// con "axios.post" introducimos el nuevo objeto

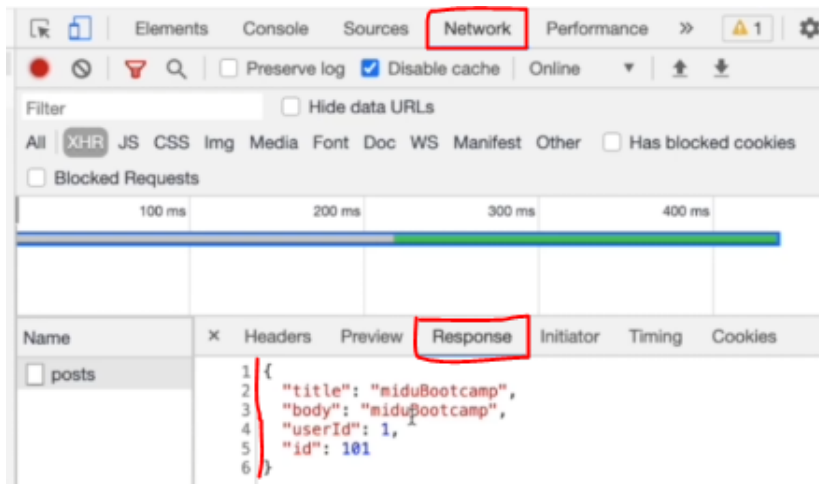
const { data } = response // con la respuesta la guardo en un elemento "data"

prevNotes => prevNotes.concat(data) // añado a los datos anteriores "prevNotes"

setNotes // actualizo el estado de Notes

```
axios
  .post('https://jsonplaceholder.typicode.com/posts', noteToAddToState)
  .then(response => {
    const { data } = response
    setNotes(prevNotes => prevNotes.concat(data))
  })
```

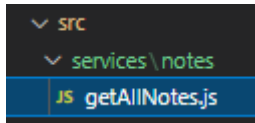
// post normalmente estas APIs te devuelven la info. que has lanzado mas algo a mayores en este caso la "id"



LOGICA DE NEGOCIO (ARQUITECTURA)

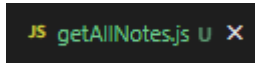
// LOGICA DE NEGOCIO: es lo que puedes hacer en tu aplicación sin que dependa de React, es decir todo lo que no dependa de React habría que extraerlo en módulos diferentes al código React.

*** genero la estructura de carpetas ***



// declaro un servicio totalmente agnóstico a React que lo que hace es recuperar datos de otra API

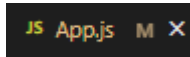
// de esta forma el día de mañana cambio de React a otra tecnología este código me seguirá funcionando



```
import axios from "axios"

export const getAllNotes = () => {
  return axios
    .get('https://jsonplaceholder.typicode.com/posts')
    .then((response) => {
      const { data } = response;
      return data
    });
}
```

// ahora en nuestra API de React



getAllNotes() // llamamos a este método que nos devuelve una promesa "notes"

.then(notes => // cuando la promesa ha sido resuelta se le indica las funciones a realizar

```
useEffect(() => {
  setLoading(true);
  setTimeout(() => {
    getAllNotes()
      .then(notes => {
        setNotes(notes);
        setLoading(false);
      });
  }, 2000);
}, []);
```

*** controlar error de las "promesas" ***

// al recibir una promesa y procesarla puede suceder que el proceso finalice correctamente y se ejecutaria el ".then" o que el proceso finalice de forma negativa produciendose un error entonces se ejecuta el ".catch"

```
createNote(noteToAddToState)
  .then(newNote => {
    setNotes((prevNotes) => prevNotes.concat(newNote));
  })
  .catch((e) => {
    console.error(e);
  });
```


4- GIT:

INSTALAR GIT

*** Descargar Git e Instalar ***

Seguir el videotutorial: "0_Descargar y Configurar Git.mp4"

GITHUB

www.github.com

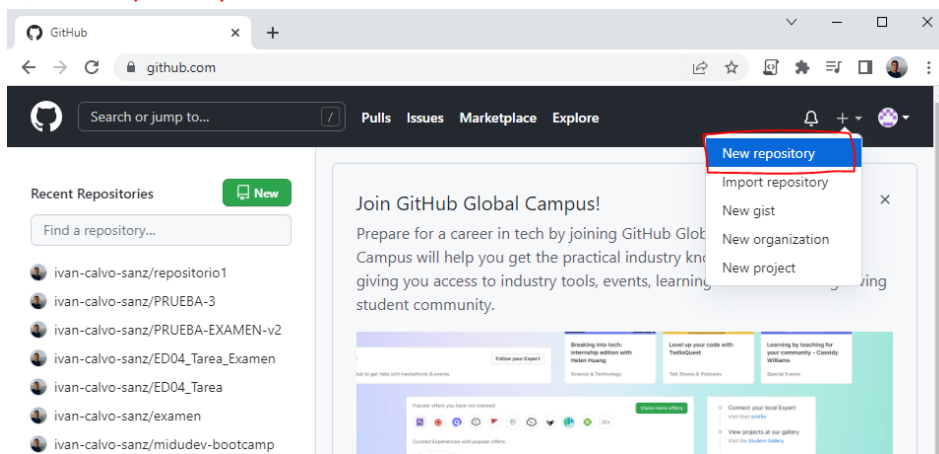
Email: ivan.calvo.sanz@gmail.com

Usuario: ivan-calvo-sanz

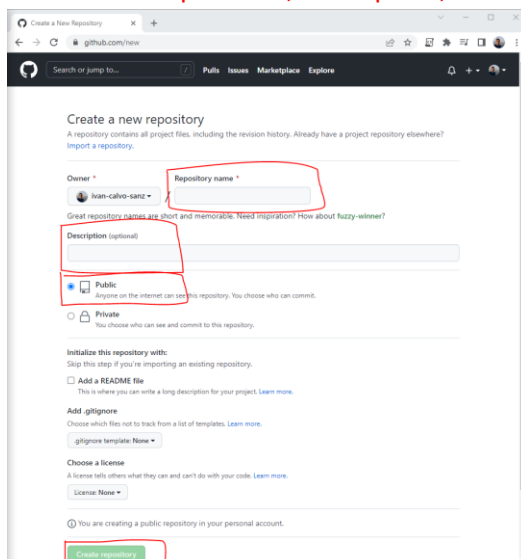
Contraseña

*** Crear nuevo Repositorio ***

+ / New repository



Nombre del repositorio, Descripción, Public, Generar repositorio



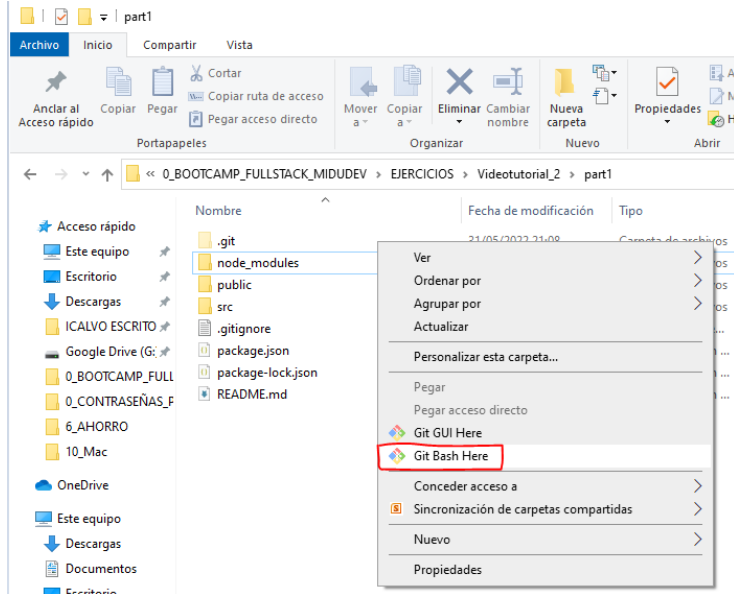
*** aparece un Paso a Paso *** subir el código

...or create a new repository on the command line

```
echo "# ClinicaVeterinariaGUI" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/chenaoh/ClinicaVeterinariaGUI.git
git push -u origin main
```

*** Abri Git Bash ***

Dentro de la carpeta del proyecto – Botón derecho – Git Bash Here



// inicializo git

git init

// verifico lo que voy a subir, en este caso indica que NO tengo commits todavía (todo lo que aparece en rojo es lo que voy a cargar en el repositorio)

git status

// agregar todos los archivos en rojo

git add .

```

MINGW64:/d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FULLSTACK_MIDUDEV/EJERCICIO...

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git init
Initialized empty Git repository in D:/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOT
CAMP_FULLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1/.git/

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        README.md
        package-lock.json
        package.json
        public/
        src/

nothing added to commit but untracked files present (use "git add" to track)

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory

```

// hacer un commit

git commit -m "Primer commit"

```
IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (master)
$ git commit -m "Primer commit"
[master (root-commit) 2b427ef] Primer commit
19 files changed, 28335 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 public/favicon.ico
```

//he realizado un commit pero los archivos están en el fichero local no en el repositorio remoto

git status

// para subirlo a nuestra rama principal

git branch -M main

// para subirlo al repositorio remoto (copio la instrucción del "Paso a Paso" que muestra Github)

git remote add origin https...

// hago push, para hacer efectiva la sincronización de la ubicación local con la remota

git push -u origin main

// el repositorio se ha subido

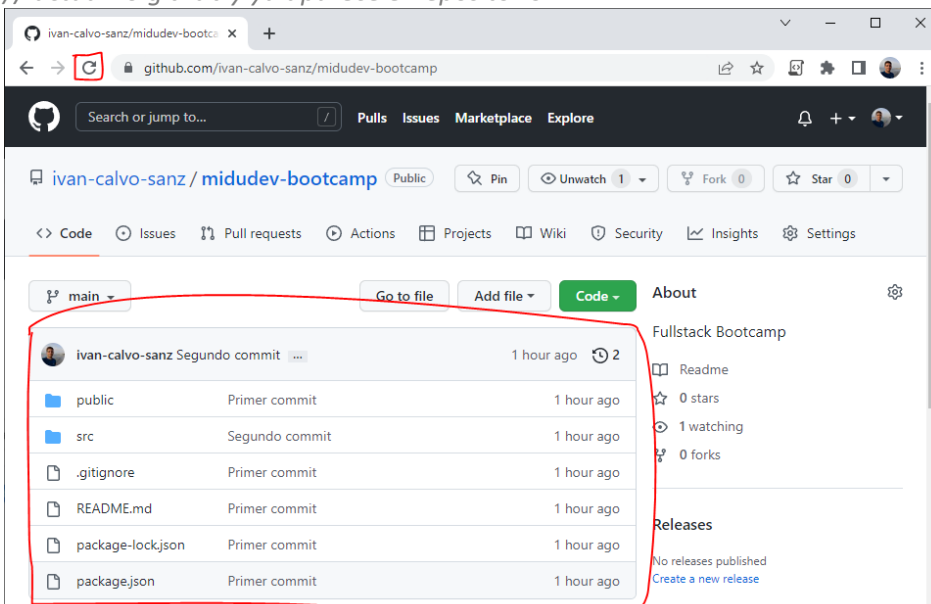
```
IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (master)
$ git status
On branch master
nothing to commit, working tree clean

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (master)
$ git branch -M main

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git remote add origin https://github.com/ivan-calvo-sanz/midudev-bootcamp.git

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git push -u origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (23/23), done.
Pushing to 'https://github.com/ivan-calvo-sanz/midudev-bootcamp.git':
Counting objects: 100% (23/23), done.
Compressing objects: 100% (23/23), done.
Writing objects: 100% (23/23), 388.13 KiB | 6.18 MiB/s | 0 min 13 s
Total 23 (delta 13), reused 0, compressed 10.00 KiB, 0 delta.
remote: Compressing objects: 100% (23/23), done.
remote: Total 23 (delta 13), reused 0, compressed 10.00 KiB, 0 delta.
Unpacking objects: 100% (23/23), done.
done.
To https://github.com/ivan-calvo-sanz/midudev-bootcamp.git
 * [new branch] main -> main
```

// actualizo github y ya aparece el repositorio



*** realizo un cambio en el proyecto y vuelvo a realizar un nuevo repositorio ***

// Añado este código

```

1  import './App.css';
2  import Mensaje from './Mensaje.js'
3
4
5  const Description = () => {
6    return <p>Esta es la app del curso fullstack bootcamp</p>
7  }
8
9  function App() {
10   return (
11     <div className="App">
12
13       <Mensaje color='red' message='Estamos trabajando' />
14       <Mensaje color='green' message='En un curso' />
15       <Mensaje color='yellow' message='De React' />
16       <Mensaje color='yellow' message='Cambio para que se vea en Git' />
17     </div>
18   );
19 }
20
21 export default App;
22
23
24

```

//veo lo que se ha modificado (en este caso el fichero App.js)

git status

```

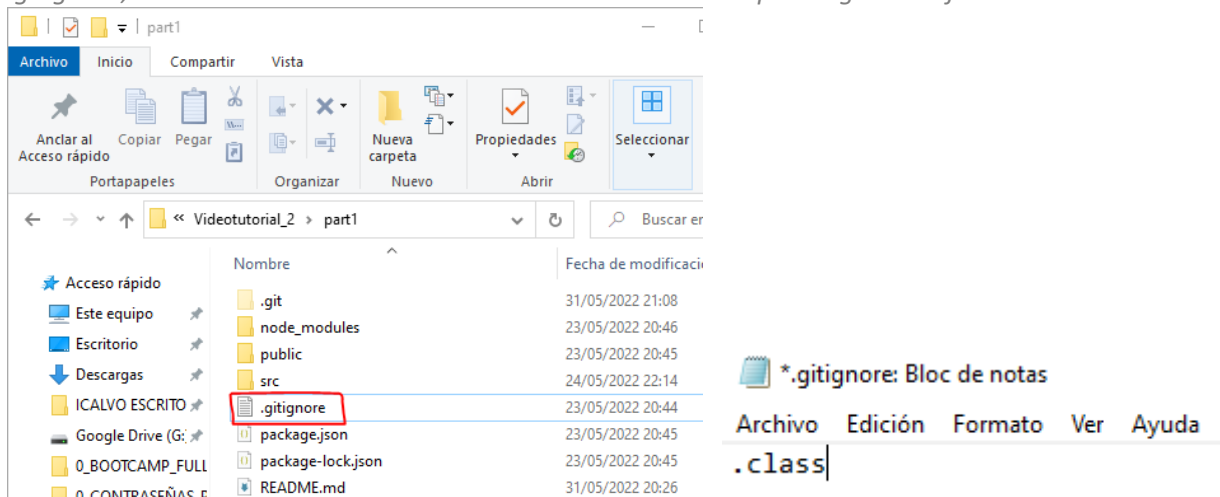
IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FULLSTACK_MTDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src/App.js

no changes added to commit (use "git add" and/or "git commit -a")

```

// para evitar que un tipo de archivos no realice commits, hay que realizar un nuevo documento de texto .gitignore, en este caso introduciendo .class todos los archivos que tengan este formato no los va commitear



// agregar todos los archivos en rojo

git add .

// hacer un commit

git commit -m "Segundo commit"

// hago push, para hacer efectiva la sincronización de la ubicación local con la remota

git push -u origin main

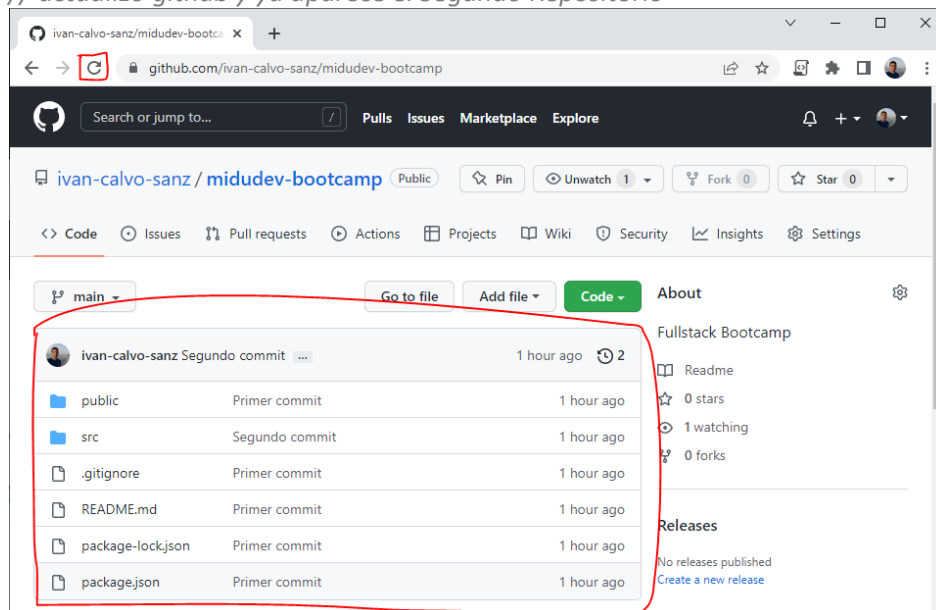
// el repositorio se ha subido

```
IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git add .
warning: LF will be replaced by CRLF in src/App.js.
The file will have its original line endings in your working directory

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git commit -m "Segundo commit"
[main da1ff58] Segundo commit
1 file changed, 1 insertion(+)

IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), done.
Total 4 (delta 3), reused 0 (delta 0), compressed 0 (delta 0)
To github.com:ivan-calvo-sanz/midudev-bootcamp
   da1ff58..d1ff58  main -> main
```

// actualizo github y ya aparece el Segundo Repositorio



// verifico cuales son los commit realizados

git log

```
IVAN@DESKTOP-RMR1GEU MINGW64 /d/ICALVO ESCRITORIO/6_FORMACION/00_DESARROLLADOR_WEB/0_BOOTCAMP_FU
LLSTACK_MIDUDEV/EJERCICIOS/Videotutorial_2/part1 (main)
$ git log
commit da1ff587216e84d1db23536bd10f446f1e14a7a0 (HEAD -> main, origin/main)
Author: ivan-calvo-sanz <ivan.calvo.sanz@gmail.com>
Date: Tue May 31 21:07:58 2022 +0200

    Segundo commit

commit 2b427ef02104a855baf0a0284797452c683e30bc
Author: ivan-calvo-sanz <ivan.calvo.sanz@gmail.com>
Date: Tue May 31 21:00:11 2022 +0200

    Primer commit
```

