TUTORIAL ESTRUCTURA BASICA REACT – NODEJS – EXPRESS

INICIAR SERVIDOR Y PROYECTO CONFIGURACION BÁSICA

SERVER

1. Iniciar servidor

```
npm init -y
```

2. Instalar extensions necesarias

```
npm install express mysql cors nodemon
```

3. crear archivo de servidor

server.js

4. Modificar package del servidor

Dentro de “scripts”:

```
“start”: “nodemon server.js”
```

5. referencias o llamados dentro de server.js

```
Import express from express
Mysql
cors
```

6. package.json

```
“type”: “module”
```

7. inicializar app en severs.js

```
Conts app = express();
App.use(cors());
//test para saber si está corriendo el server
App.listen(8081, ()=> {
Console.log(“Listening”)
})
```

FRONT END

1. Crear react app

```
npx create-react-app apptest
```

// en caso de haber incompatibilities

```
npm install react@18 react-dom@18
```

2. actualizer todo

```
npm install
```

3. dirigirse a la ruta principal del proyecto

Instalar bibliotecas

```
npm install axios boostrap react-router-dom
npm install web-vitals
```

// iniciar app

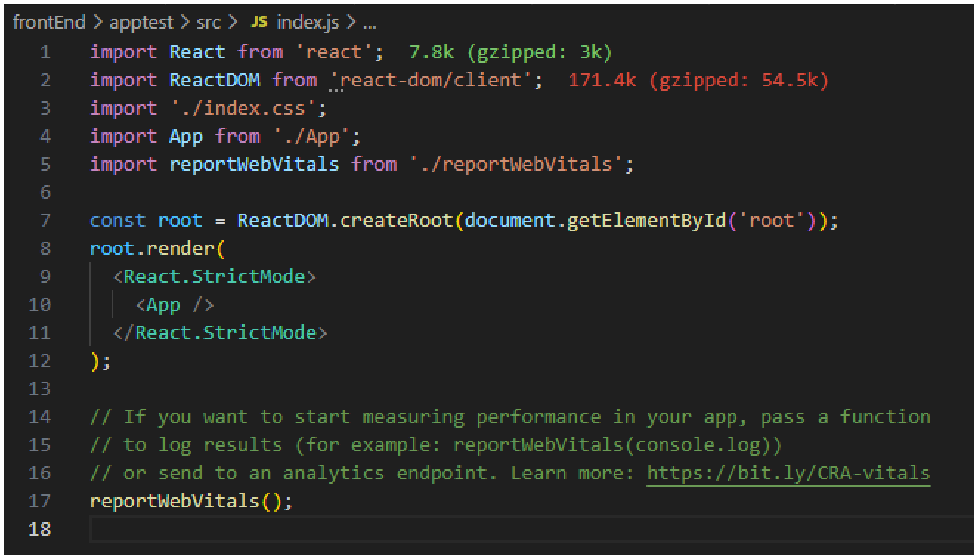
```
npm run dev
```

```
\\
npm install bootstrap
npm install react-bootstrap
npm start
```

ESTRUCTURA

Archivo principal:

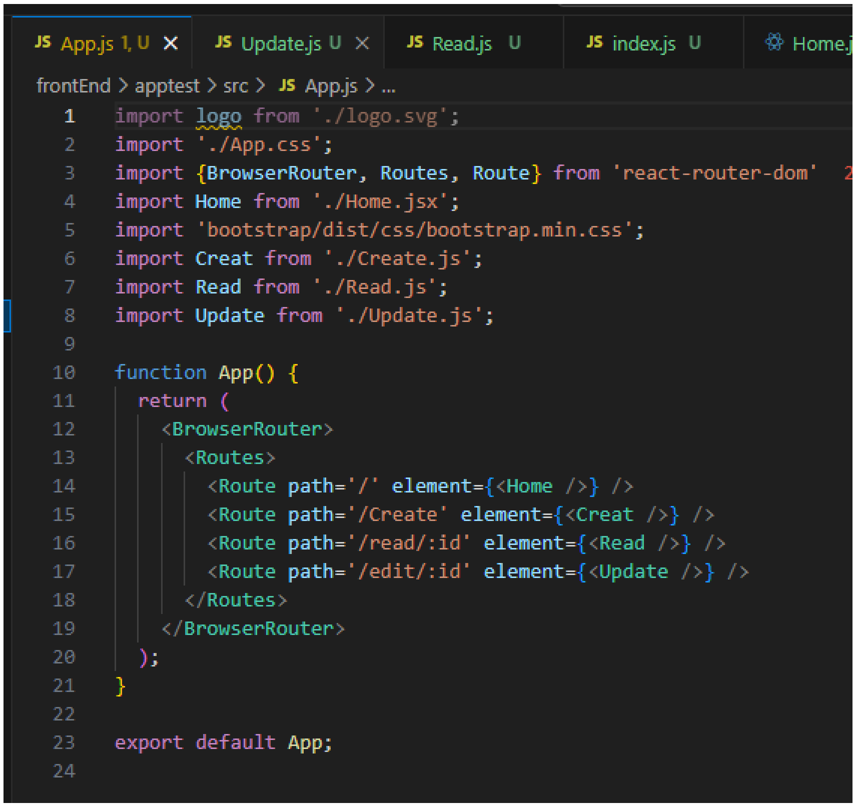
Index.js



```
import React from 'react';  7.8k (gzipped: 3k)
import ReactDOM from 'react-dom/client';  171.4k (gzipped: 54.5k)
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>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```

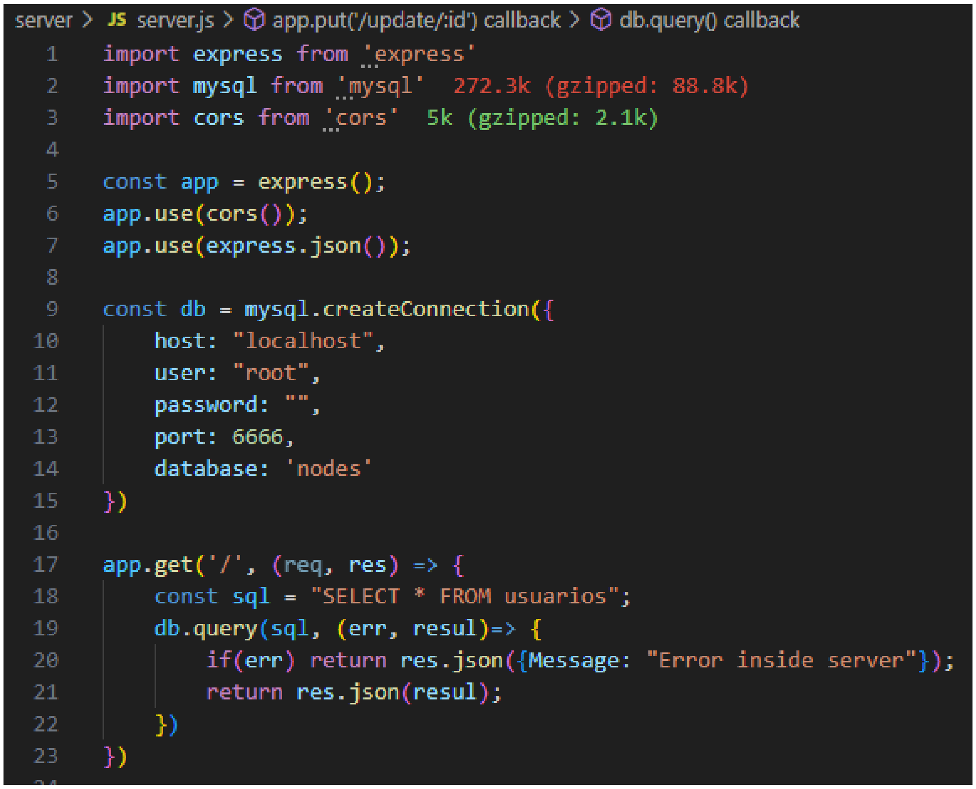
App.js (ejemplo)

Home, Crear, Read, Update: son las páginas a visitar como serían los archivos php y el path es como se accede a ella y los parámetros que recibe



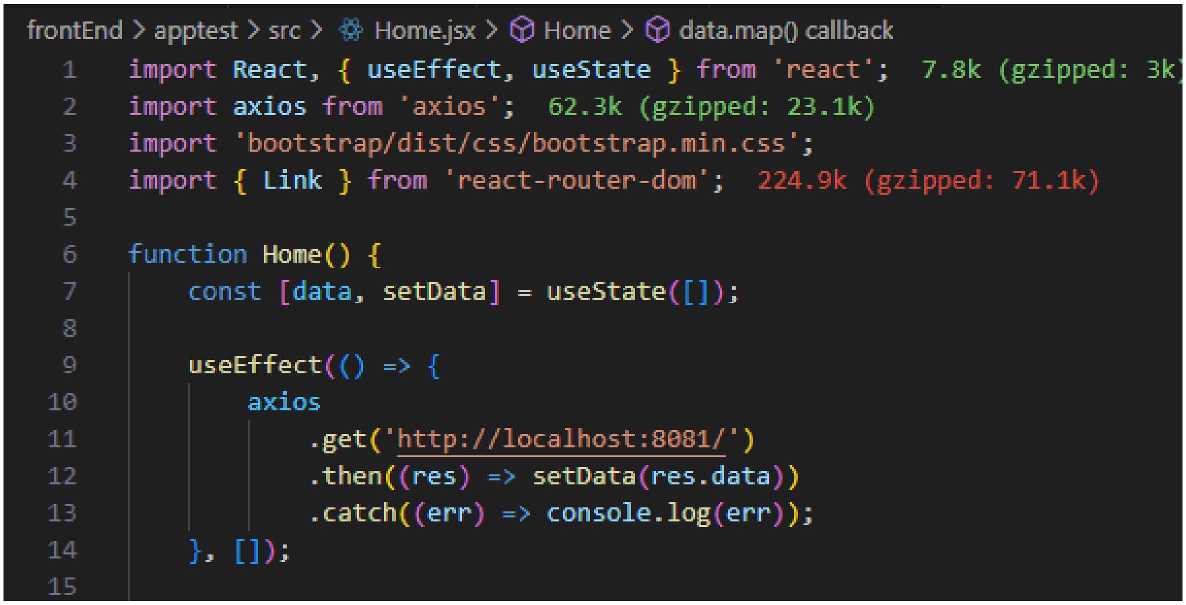
```
import logo from './logo.svg';
import './App.css';
import {BrowserRouter, Routes, Route} from 'react-router-dom'  2
import Home from './Home.jsx';
import 'bootstrap/dist/css/bootstrap.min.css';
import Creat from './Create.js';
import Read from './Read.js';
import Update from './Update.js';

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path='/' element={<Home />} />
        <Route path='/Create' element={<Creat />} />
        <Route path='/read/:id' element={<Read />} />
        <Route path='/edit/:id' element={<Update />} />
      </Routes>
    </BrowserRouter>
  );
}

export default App;

```

Configuración de consultas dentro de sever,js (donde están las consultas) conexión a la db y consulta sin parametros para página inicial:


server > JS server.js > app.put('/update/:id') callback > db.query() callback

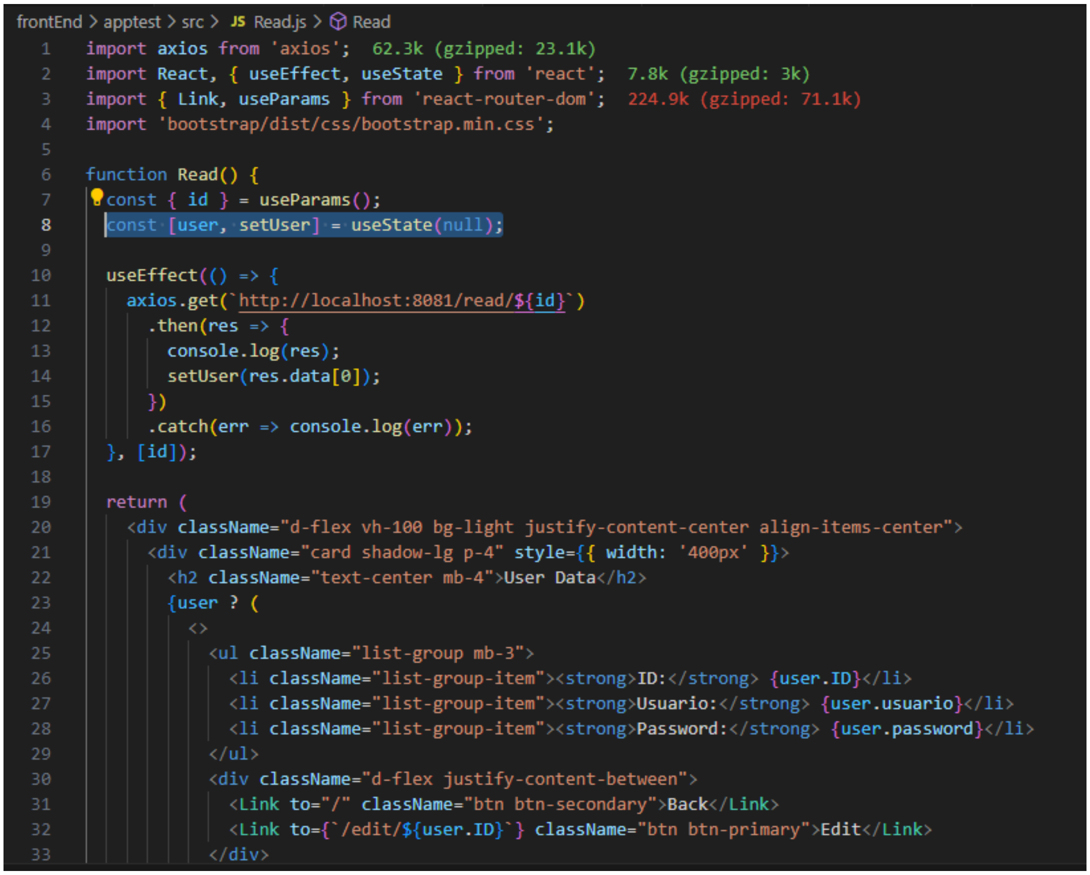
```
import express from 'express'
import mysql from 'mysql'  272.3k (gzipped: 88.8k)
import cors from 'cors'  5k (gzipped: 2.1k)

const app = express();
app.use(cors());
app.use(express.json());

const db = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "",
    port: 6666,
    database: 'nodes'
})

app.get('/', (req, res) => {
    const sql = "SELECT * FROM usuarios";
    db.query(sql, (err, resul)=> {
        if(err) return res.json({Message: "Error inside server"});
        return res.json(resul);
    })
})
```

En el archivo Home.js (menú inicial) con directorio '/' definido en App.js

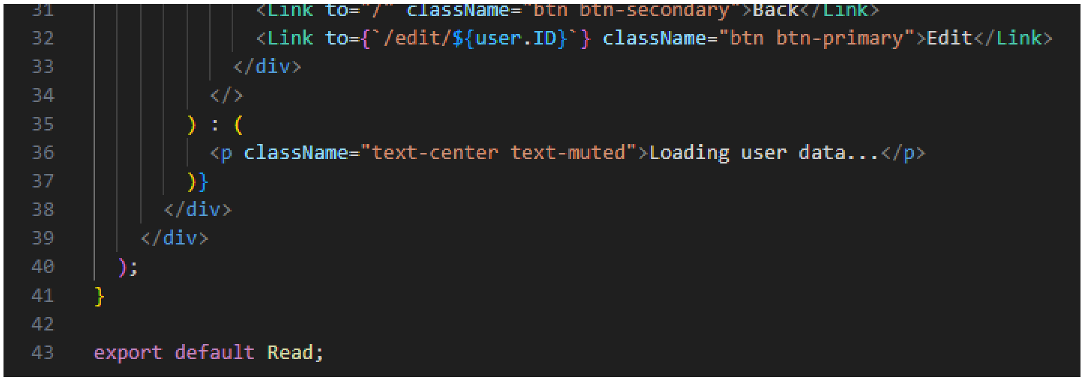

frontEnd > apptest > src > Home.jsx > Home > data.map() callback

```
import React, { useEffect, useState } from 'react';  7.8k (gzipped: 3k)
import axios from 'axios';  62.3k (gzipped: 23.1k)
import 'bootstrap/dist/css/bootstrap.min.css';
import { Link } from 'react-router-dom';  224.9k (gzipped: 71.1k)

function Home() {
    const [data, setData] = useState([]);

    useEffect(() => {
        axios
            .get('http://localhost:8081/')
            .then((res) => setData(res.data))
            .catch((err) => console.log(err));
    }, []);

```

Ejemplo archivo Read,js con parámetros (front) //CONSULTA SELECT



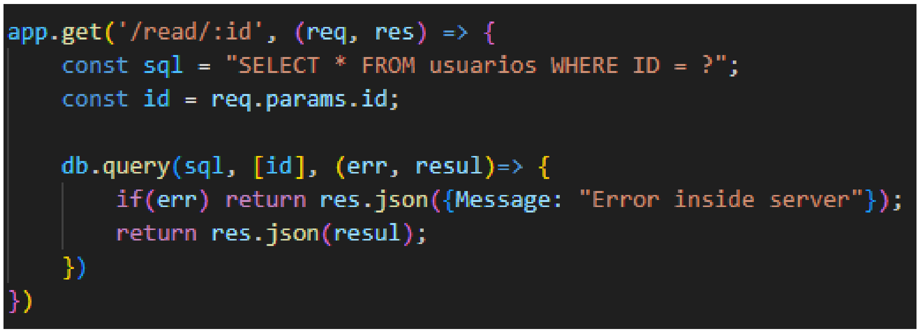
```
import axios from 'axios';  62.3k (gzipped: 23.1k)
import React, { useEffect, useState } from 'react';  7.8k (gzipped: 3k)
import { Link, useParams } from 'react-router-dom';  224.9k (gzipped: 71.1k)
import 'bootstrap/dist/css/bootstrap.min.css';

function Read() {
  const { id } = useParams();
  const [user, setUser] = useState(null);

  useEffect(() => {
    axios.get(`http://localhost:8081/read/${id}`)
      .then(res => {
        console.log(res);
        setUser(res.data[0]);
      })
      .catch(err => console.log(err));
  }, [id]);

  return (
    <div className="d-flex vh-100 bg-light justify-content-center align-items-center">
      <div className="card shadow-lg p-4" style={{ width: '400px' }}>
        <h2 className="text-center mb-4">User Data</h2>
        {user ? (
          <>
            <ul className="list-group mb-3">
              <li className="list-group-item"><strong>ID:</strong> {user.ID}</li>
              <li className="list-group-item"><strong>Usuario:</strong> {user.usuario}</li>
              <li className="list-group-item"><strong>Password:</strong> {user.password}</li>
            </ul>
            <div className="d-flex justify-content-between">
              <Link to="/" className="btn btn-secondary">Back</Link>
              <Link to={`/edit/${user.ID}`} className="btn btn-primary">Edit</Link>
            </div>
```



```
              <Link to="/" className="btn btn-secondary">Back</Link>
              <Link to={`/edit/${user.ID}`} className="btn btn-primary">Edit</Link>
            </div>
```

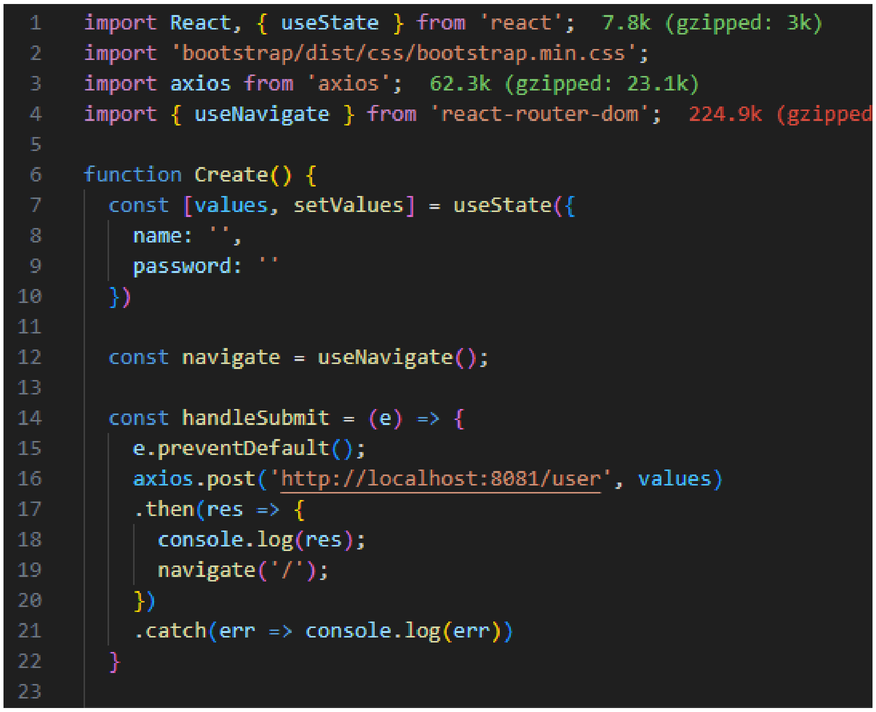
```
          </>
        ) : (
          <p className="text-center text-muted">Loading user data...</p>
        )}
      </div>
    </div>
  );
}

export default Read;
```

Para la consulta dentro del server.js (backend)

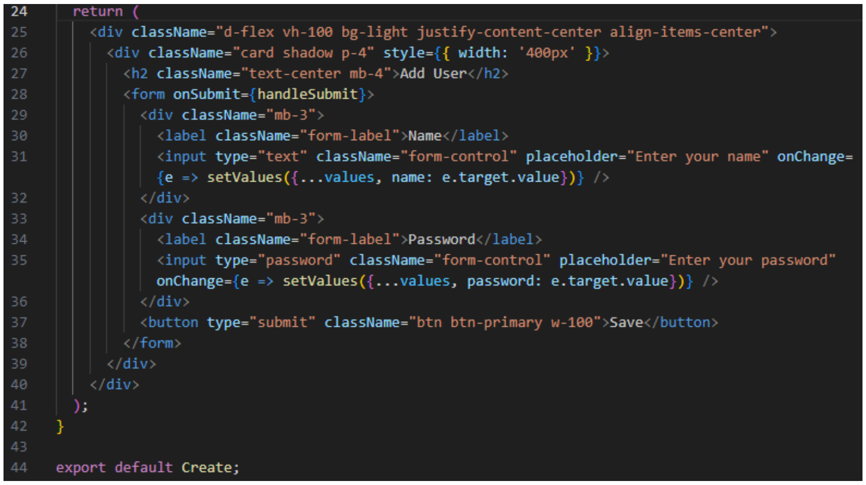


```
app.get('/read/:id', (req, res) => {
    const sql = "SELECT * FROM usuarios WHERE ID = ?";
    const id = req.params.id;

    db.query(sql, [id], (err, resul)=> {
        if(err) return res.json({Message: "Error inside server"});
        return res.json(resul);
    })
})
```

Ejemplo archivo Create,js con parámetros (front) //CONSULTA INSERT

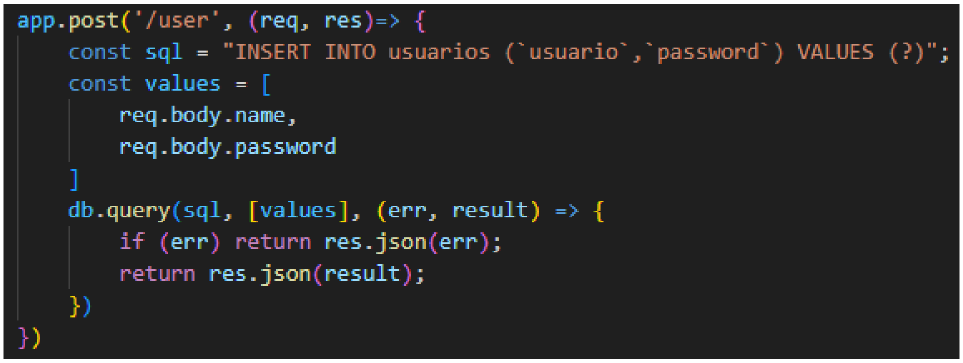


```
import React, { useState } from 'react';  7.8k (gzipped: 3k)
import 'bootstrap/dist/css/bootstrap.min.css';
import axios from 'axios';  62.3k (gzipped: 23.1k)
import { useNavigate } from 'react-router-dom';  224.9k (gzipped

function Create() {
  const [values, setValues] = useState({
    name: '',
    password: ''
  })

  const navigate = useNavigate();

  const handleSubmit = (e) => {
    e.preventDefault();
    axios.post('http://localhost:8081/user', values)
    .then(res => {
      console.log(res);
      navigate('/');
    })
    .catch(err => console.log(err))
  }

```



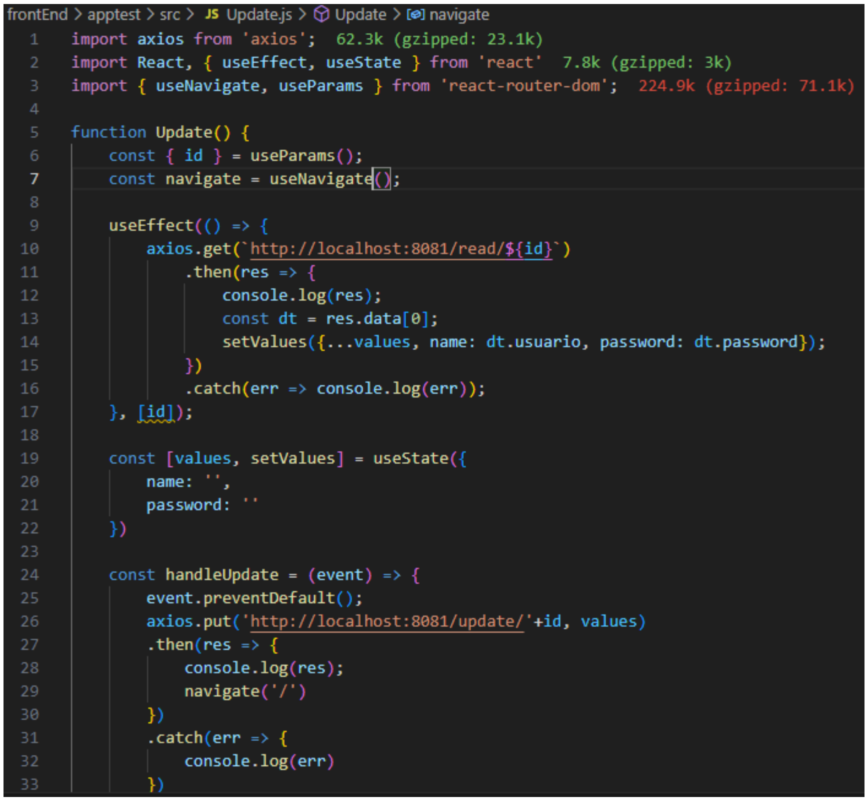
```
  return (
    <div className="d-flex vh-100 bg-light justify-content-center align-items-center">
      <div className="card shadow p-4" style={{ width: '400px' }}>
        <h2 className="text-center mb-4">Add User</h2>
        <form onSubmit={handleSubmit}>
          <div className="mb-3">
            <label className="form-label">Name</label>
            <input type="text" className="form-control" placeholder="Enter your name" onChange=
            {e => setValues({...values, name: e.target.value})} />
          </div>
          <div className="mb-3">
            <label className="form-label">Password</label>
            <input type="password" className="form-control" placeholder="Enter your password"
            onChange={e => setValues({...values, password: e.target.value})} />
          </div>
          <button type="submit" className="btn btn-primary w-100">Save</button>
        </form>
      </div>
    </div>
  );
}

export default Create;
```

Para la consulta dentro del server.js (backend)

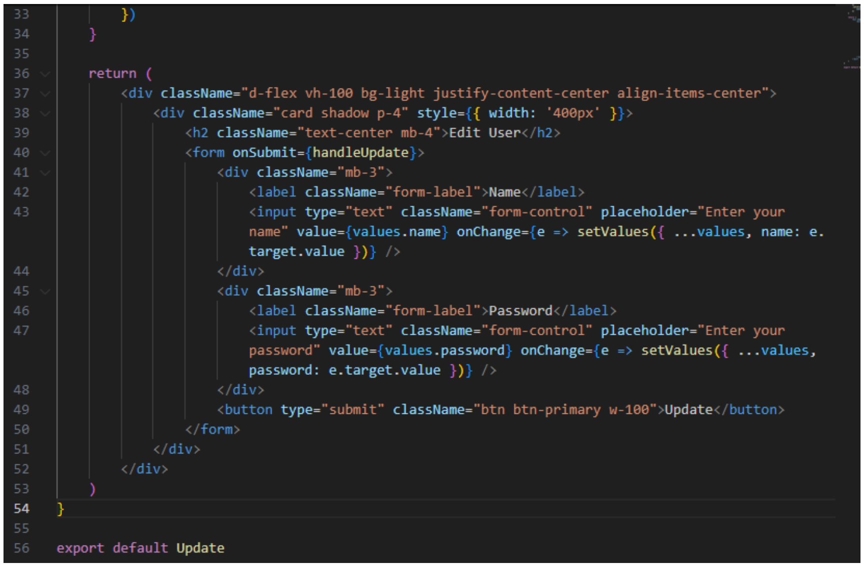


```
app.post('/user', (req, res)=> {
    const sql = "INSERT INTO usuarios (`usuario`,`password`) VALUES (?)";
    const values = [
        req.body.name,
        req.body.password
    ]
    db.query(sql, [values], (err, result) => {
        if (err) return res.json(err);
        return res.json(result);
    })
})
```

Ejemplo archivo Update,js con parámetros (front) //CONSULTA UPDATE DELETE

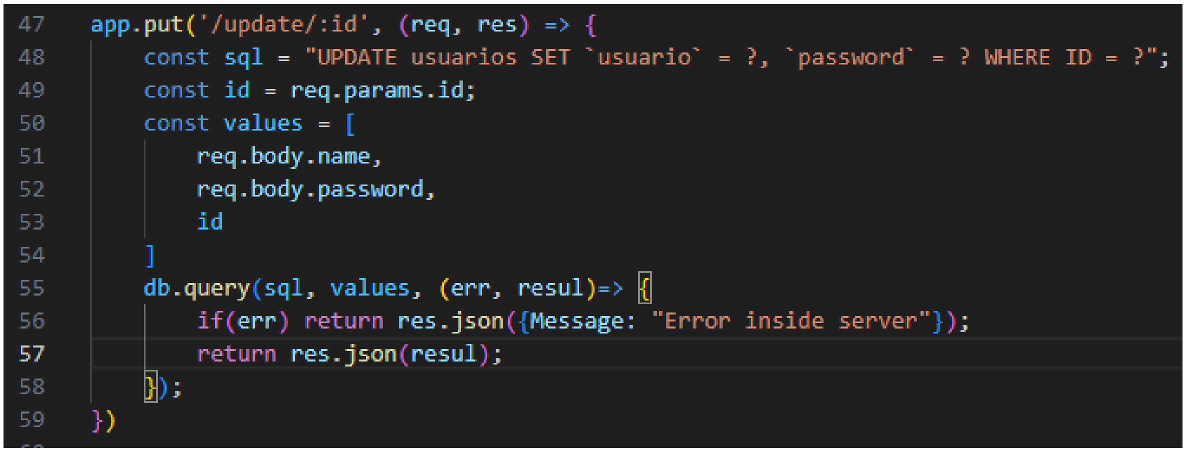


```
import axios from 'axios';  62.3k (gzipped: 23.1k)
import React, { useEffect, useState } from 'react'  7.8k (gzipped: 3k)
import { useNavigate, useParams } from 'react-router-dom';  224.9k (gzipped: 71.1k)

function Update() {
    const { id } = useParams();
    const navigate = useNavigate();

    useEffect(() => {
        axios.get(`http://localhost:8081/read/${id}`)
            .then(res => {
                console.log(res);
                const dt = res.data[0];
                setValues({...values, name: dt.usuario, password: dt.password});
            })
            .catch(err => console.log(err));
    }, [id]);

    const [values, setValues] = useState({
        name: '',
        password: ''
    })

    const handleUpdate = (event) => {
        event.preventDefault();
        axios.put('http://localhost:8081/update/'+id, values)
        .then(res => {
            console.log(res);
            navigate('/')
        })
        .catch(err => {
            console.log(err)
        })
```



```
        })
    }

    return (
        <div className="d-flex vh-100 bg-light justify-content-center align-items-center">
            <div className="card shadow p-4" style={{ width: '400px' }}>
                <h2 className="text-center mb-4">Edit User</h2>
                <form onSubmit={handleUpdate}>
                    <div className="mb-3">
                        <label className="form-label">Name</label>
                        <input type="text" className="form-control" placeholder="Enter your
                        name" value={values.name} onChange={e => setValues({ ...values, name: e.
                        target.value })} />
                    </div>
                    <div className="mb-3">
                        <label className="form-label">Password</label>
                        <input type="text" className="form-control" placeholder="Enter your
                        password" value={values.password} onChange={e => setValues({ ...values,
                        password: e.target.value })} />
                    </div>
                    <button type="submit" className="btn btn-primary w-100">Update</button>
                </form>
            </div>
        </div>
    )
}

export default Update
```

Consulta server.js (backend)



```
app.put('/update/:id', (req, res) => {
    const sql = "UPDATE usuarios SET `usuario` = ?, `password` = ? WHERE ID = ?";
    const id = req.params.id;
    const values = [
        req.body.name,
        req.body.password,
        id
    ]
    db.query(sql, values, (err, resul)=> {
        if(err) return res.json({Message: "Error inside server"});
        return res.json(resul);
    });
})
```