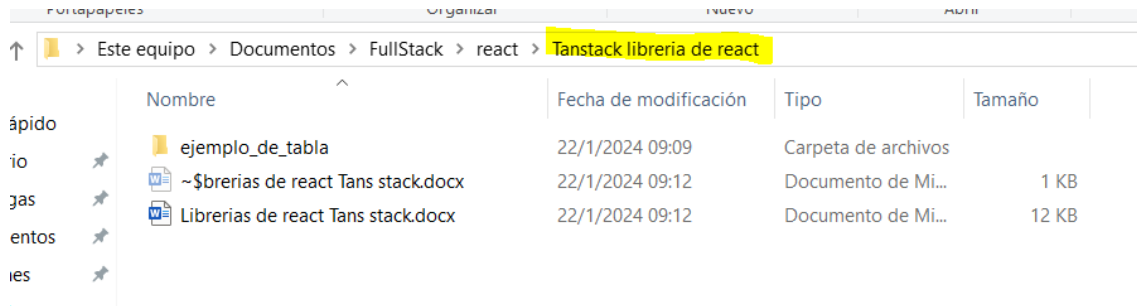
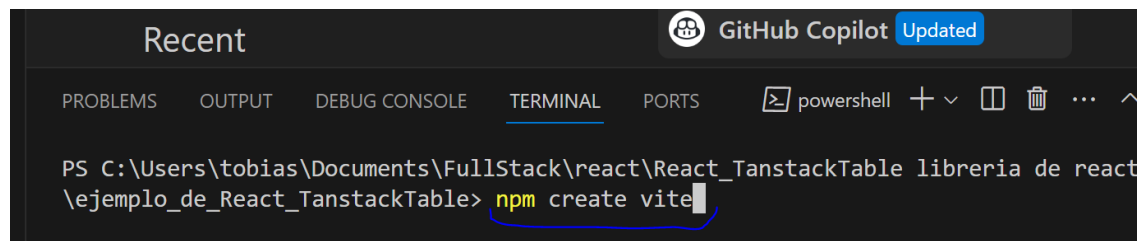


Librerías de React_TanstackTable

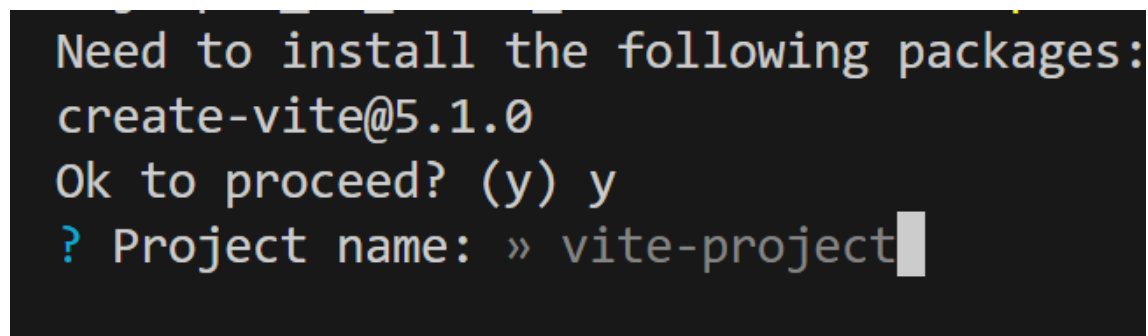
Creo una carpeta:



La abro desde el visual studio code; y en la terminal escribo:



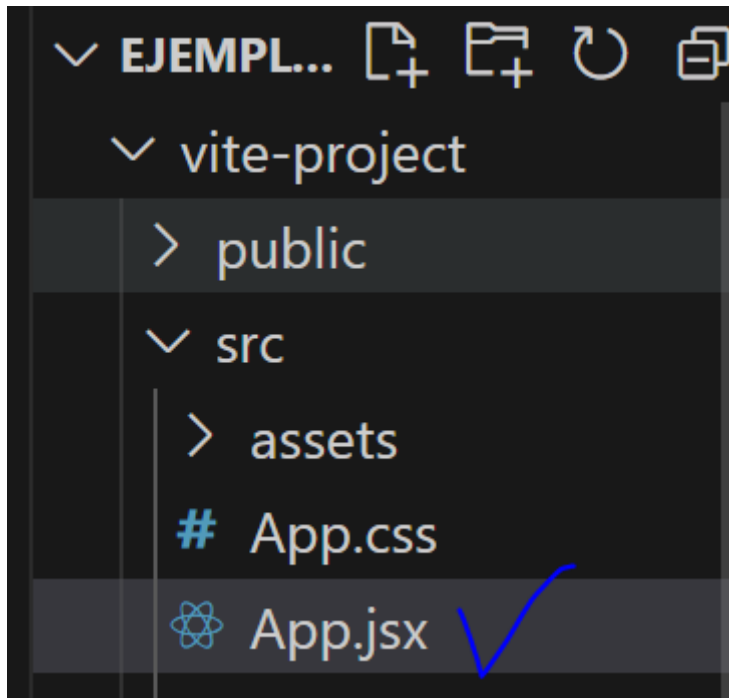
Le ponemos y



```
Vanilla
Vue
> React
Preact
Lit
Svelte
Solid
Qwik
Others
```

```
? Select a variant: »
TypeScript
TypeScript + SWC
> JavaScript
JavaScript + SWC
```

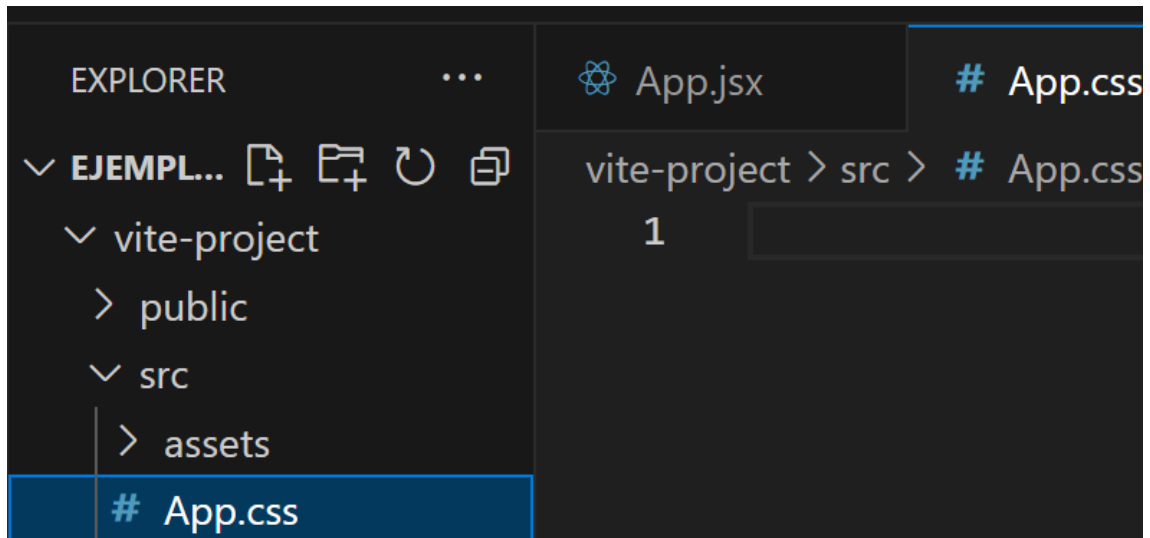
Quito todo el contenido de App.jsx



Y nos queda así:

```
App.jsx
vite-project > src > App.jsx > [default]
1  function App(){
2      return (
3          <div>
4              App
5          </div>
6      )
7  }
8  export default App
```

Al archivo App.css y a index.css también los dejo vacío:



Me paro en

```
PS C:\Users\tobias\Documents\FullStack\react\React_TanstackTable libreria de react\ejemplo_de_React_TanstackTable\<u>vite-project</u>> |
```

Npm instal o npm i

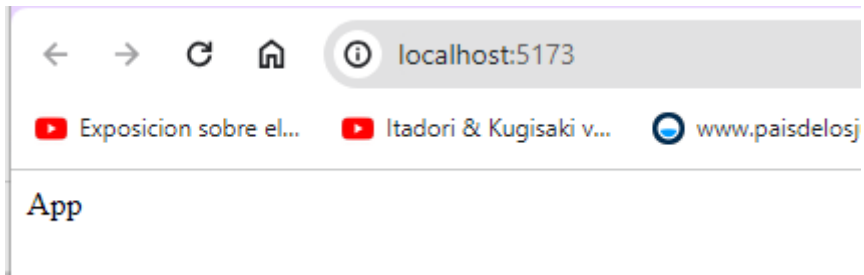
```
\_logs\2024-01-22T12_39_37_898Z-debug-0.log
ers\tobias\Documents\FullStack\react\React_TanstackTable> npm i
```

Ejecuto para ver como va:

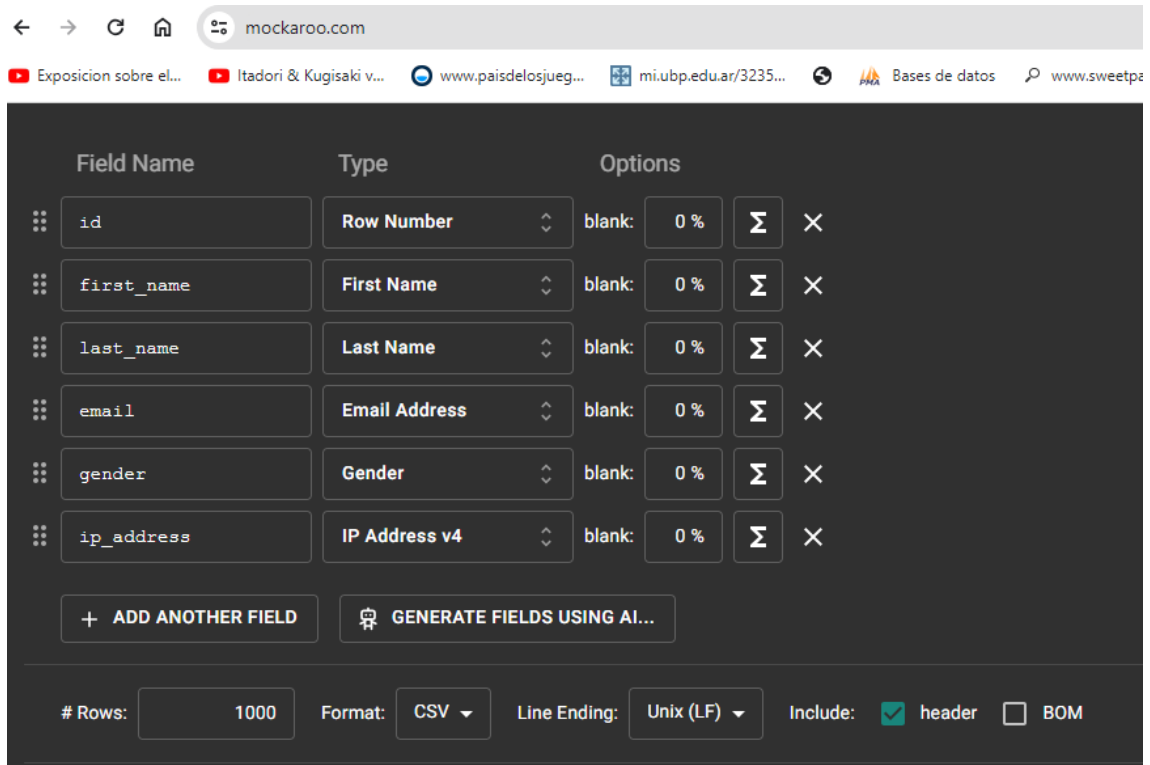
```
PS C:\Users\tobias\Documents\FullStack\react\React_TanstackTable> npm run dev
```


```
VITE v5.0.12 ready in 344 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```



Vamos a utilizar una web que se llama mockaroo <https://www.mockaroo.com/> y en la pantalla principal:



 SCHEMAS DATASETS MOCK APIS SCENARIOS PROJECTS

id	Row Number	blank: 0 %	Σ	×
name	First Name	blank: 0 %	Σ	×
lastname	Last Name	blank: 0 %	Σ	×
email	Email Address	blank: 0 %	Σ	×
country	Gender ✓	blank: 0 %	Σ	×
ip_address	IP Address v4	blank: 0 %	Σ	×

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

Rows: 1000 Format: CSV Line Ending: Unix (LF) Include: ☒ header

SCHEMAS DATASETS MOCK APIS SCENARIOS PROJECTS FUNCTIONS SIGN IN

Choose a Type

country

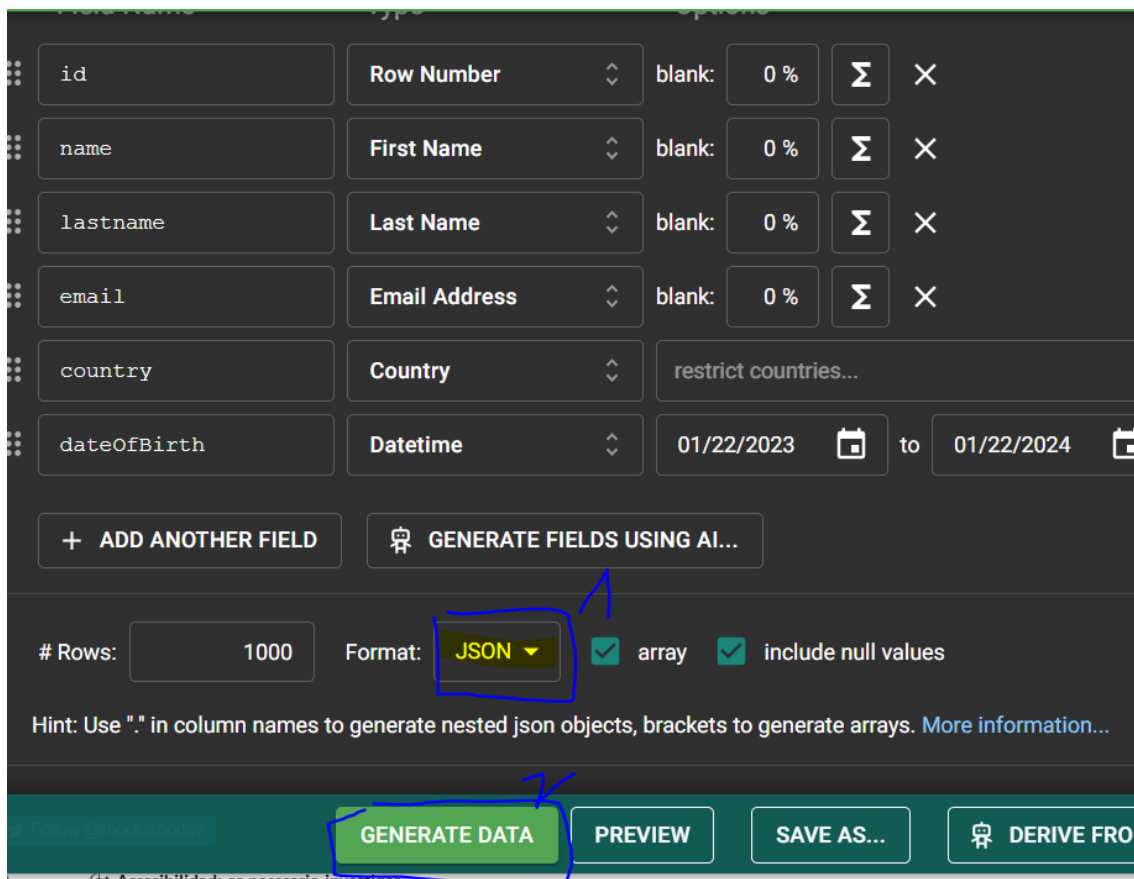
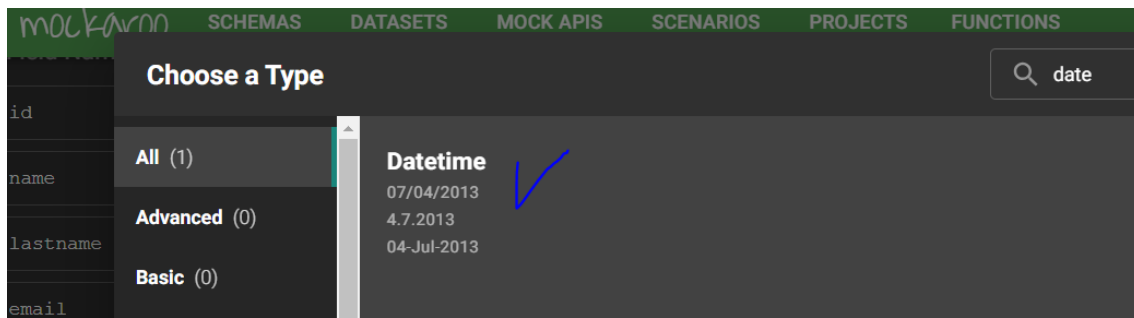
All (3)	Airport Country Code	Country ✓	Country Code
	US	Germany	ES
	CA	France	GR
	DE	Japan	FR

Advanced (0) Basic (0) Car (0) Commerce (0)

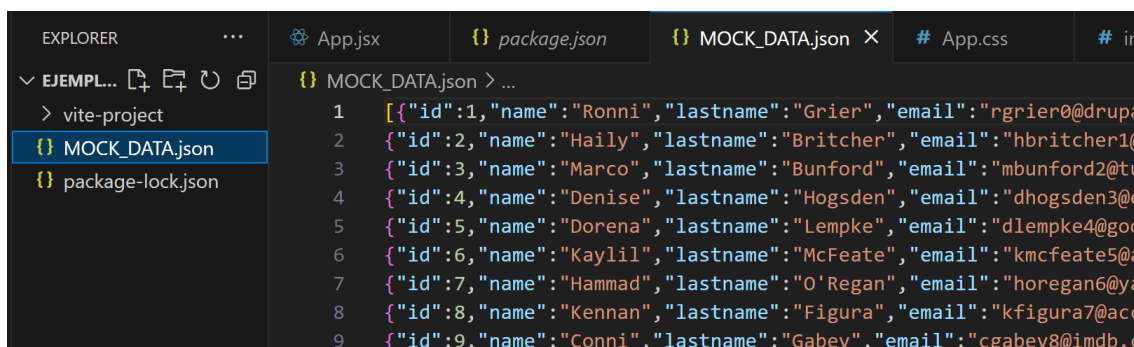
Y vuelve a la pantalla principal

id	Row Number	blank: 0 %	Σ	×
name	First Name	blank: 0 %	Σ	×
lastname	Last Name	blank: 0 %	Σ	×
email	Email Address	blank: 0 %	Σ	×
country	Country	restrict countries...		
dateOfBirth	IP Address v4 ✓	blank: 0 %	Σ	×

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...



Lo guarda en descargas, y solo lo arrastro a mi proyecto vite-project:



Click derecho para darle formato:

TanStack **Table** v8

Q Search

CTRL

K

MENU

Home

GitHub

Discord

ADAPTERS

React Table

Solid Table

Svelte Table

Vue Table

Vanilla JS/TS

React Table

The `@tanstack/react-table` adapter is a wrapper around the core table state the "react" way, providing types and the rendering implementation o

`useReactTable`

Takes an `options` object and returns a table.

```
import { useReactTable } from '@tanstack/react-table'

function App() {
  const table = useReactTable(options)

  // ...render your table
}
```

GUIDE

Introduction

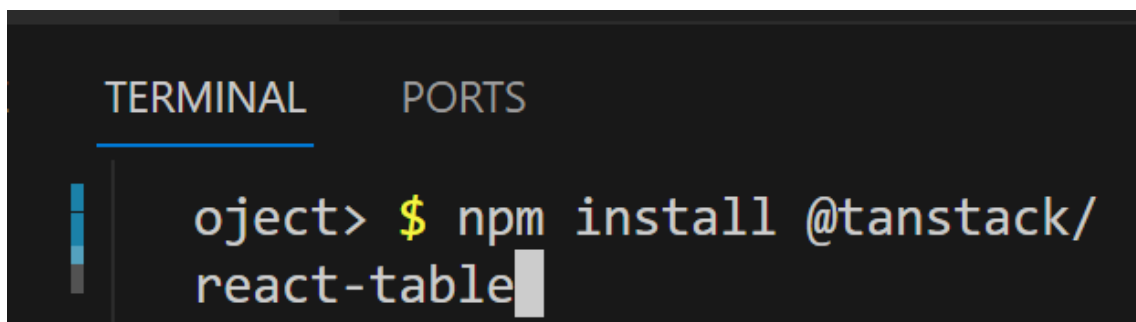
Overview

Installation ✓

Copio el comando de React Table

```
$ npm install @tanstack/react-table
```

En la terminal instalo dicha librería, antes paro el servidor:



Vuelvo a correr `npm run dev`

Creo una carpeta y archivo en el src

✓ EJEMPLO_DE_REACT_TANS...

✓ vite-project

> node_modules

> public

✓ src

> assets

✓ componentes



SimpleTable.jsx



App.jsx



SimpleTable.jsx



package.json

vite-project > src > componentes > SimpleTable.jsx

1

rf



rfc



rfc



rfce

react



rfce

react



rfcp

reactFunction

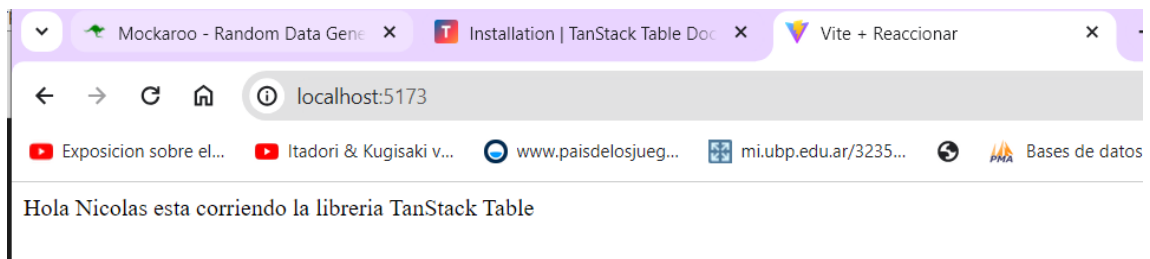
```
SimpleTable.jsx X {}  
project > src > componentes > SimpleTable  
import React from 'react'  
  
function SimpleTable() {  
  return (  
    <div>  
  
    </div>  
  )  
}  
export default SimpleTable
```

Y en App.jsx

```
App.jsx 1 X SimpleTable.jsx package.json  
vite-project > src > App.jsx > App  
1 function App(){  
2   return (  
3     <div>  
4     <Simpl> JSX element 'Simpl' has no  
5     </div> SimpleTable ✓  
6   ) SVGFEDisplacementMapElement  
7 }  
8 export default App
```

```
App.jsx × SimpleTable.jsx × {} package.json {} MOCK_DATA.js
vite-project > src > componentes > SimpleTable.jsx > SimpleTable
1 import React from 'react'
2
3 function SimpleTable() {
4   return (
5     <div>
6       ⚡ Hola Nicolas esta corriendo la libreria TanStack
7     <Table
8     </div>
9   )
10 }
11 export default SimpleTable
12
```

Lo pruebo con npm run dev



El archivo SimpleTable.jsx tiene:

```
import React from 'react'

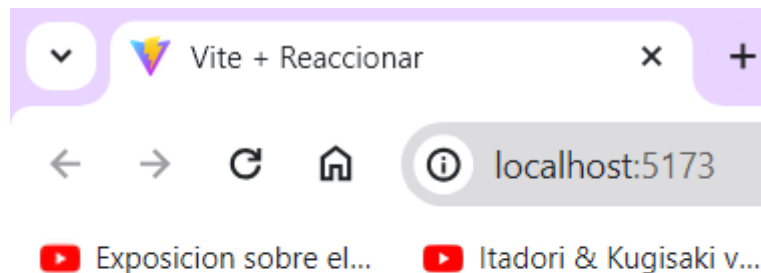
function SimpleTable() {
  return (
    <div>
      <table>
        <thead>
          <tr>
            <th>ID</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
          </tr>
        </tbody>
        <tfoot>
          <tr>
            <td>
              id
            </td>
          </tr>
        </tfoot>
      </table>
    </div>
  )
}
```

```

        </tr>
      </tfoot>
    </table>
  </div>
)
}
export default SimpleTable

```

y en el navegador se ve:



IDENTIFICACIÓN

1

identificación

23/01/2023

Para agregar y ver las cabeceras

Desde el archivo SimpleTable.jsx

```

import { useReactTable } from "@tanstack/react-table";

import data from "../../MOCK_DATA.json";

function SimpleTable() {
  const columns = [//lo que se ve en el navegador
    {
      header: "ID", //lo que se muestra en pantalla
      accessorKey: 'id' //para que lea la tabla
    },
    {
      header: "Name",
      accessorKey: 'name',
    },
    {
      header: "Lastname",
      accessorKey: 'lastname'
    },
  ],

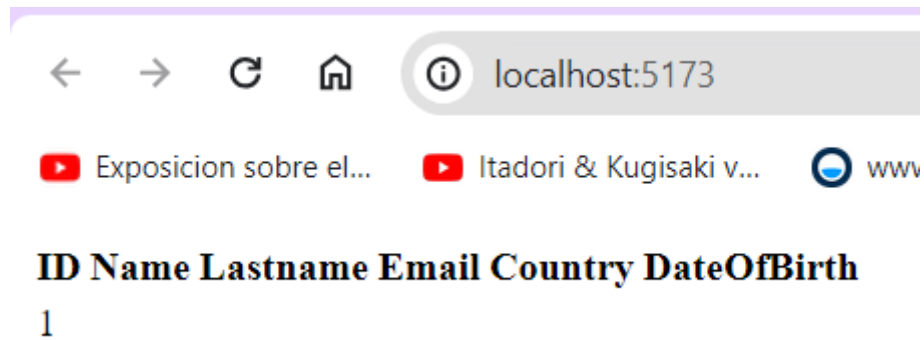
```

```

    {
      header: "Email",
      accessorKey: "email"
    },
    {
      header: "Country",
      accessorKey: "country"
    },
    {
      header: "DateOfBirth",
      accessorKey: "dateOfBirth"
    }
  ]
  const table = useReactTable({ data, columns: columns });
  return (
    <div>
      <table>
        <thead>
          {
            table.getHeaderGroups().map(headerGroup => (
              <tr key={headerGroup.id}>
                {
                  headerGroup.headers.map(header =>(
                    <th key={header.id}>
                      {header.column.columnDef.header}
                    </th>
                  ))
                }
              </tr>
            ))
          }
        </thead>
        <tbody>
          <tr>
            <td>1</td>
          </tr>
        </tbody>
        <tfoot>
          <tr>
            <td>
              id
            </td>
          </tr>
        </tfoot>
      </table>
    </div>
  )
}

```

```
export default SimpleTable
```



Ahora le agrego al mismo archivo

```
import { useReactTable, getCoreRowModel, flexRender } from
"@tanstack/react-table";

import data from "../../MOCK_DATA.json";

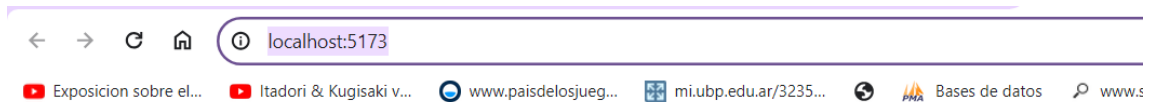
function SimpleTable() {
  const columns = [//lo que se ve en el navegador
    {
      header: "ID",//lo que se muestra en pantalla
      accessorKey: 'id'//para que lea la tabla
    },
    {
      header: "Name",
      accessorKey: 'name',
    },
    {
      header: "Lastname",
      accessorKey: 'lastname'
    },
    {
      header: "Email",
      accessorKey: "email"
    },
    {
      header: "Country",
      accessorKey: "country"
    },
    {
      header: "DateOfBirth",
      accessorKey: "dateOfBirth"
    }
  ]
}
```

```

    const table = useReactTable({ data, columns, getCoreRowModel:
getCoreRowModel() });
    return (
      <div>
        <table>
          <thead>
            {
              table.getHeaderGroups().map(headerGroup => (
                <tr key={headerGroup.id}>
                  {
                    headerGroup.headers.map(header => (
                      <th key={header.id}>
                        {header.column.columnDef.header}
                      </th>
                    ))
                  }
                </tr>
              ))
            }
          </thead>
          <tbody>
            {
              table.getRowModel().rows.map(row => (
                <tr key={row.id}>
                  {row.getVisibleCells().map((cell) => (
                    <td>{flexRender(cell.column.columnDef.cell, cell.getCont
ext())}</td>
                    )))}
                </tr>
              ))
            }
          </tbody>
          <tfoot>
            <tr>
              <td>
                id
              </td>
            </tr>
          </tfoot>
        </table>
      </div>
    )
  }
}
export default SimpleTable

```

y en el navegador



ID	Name	Lastname	Email	Country	DateOfBirth
1	Ronni	Grier	rgrier0@drupal.org	Serbia	2/19/2023
2	Haily	Britcher	hbritcher1@indiegogo.com	Hungary	6/3/2023
3	Marco	Bunford	mbunford2@tuttocitta.it	China	5/12/2023
4	Denise	Hogsden	dhogsden3@earthlink.net	Brazil	1/2/2024
5	Dorena	Lempke	dlempke4@goo.gl	Mexico	10/1/2023
6	Kaylil	McFeate	kmcfeate5@amazonaws.com	China	8/29/2023
7	Hammad	O'Regan	horegan6@yahoo.com	Czech Republic	9/28/2023
8	Kennan	Figura	kfigura7@accuweather.com	Indonesia	5/31/2023
9	Conni	Gabey	cgabey8@imdb.com	Canada	11/3/2023
10	Chrissie	Olding	colding9@topsy.com	Sweden	5/26/2023
11	Free	Menego	fmenegoa@ebay.com	Indonesia	8/30/2023
12	Jessy	Storey	jstoreyb@cloudflare.com	Nigeria	3/31/2023
13	Arv	Daintier	adaintierc@mlb.com	Kazakhstan	2/5/2023
14	Cathyleen	Dinjes	cdinjesd@miibeian.gov.cn	Portugal	1/9/2024
15	Liv	Puttan	lputtane@ehow.com	Portugal	11/5/2023

Termino el archivo y nos queda:

```
import { useReactTable, getCoreRowModel, flexRender } from
"@tanstack/react-table";

import data from "../MOCK_DATA.json";

function SimpleTable() {
  const columns = [//lo que se ve en el navegador
    {
      header: "ID",//lo que se muestra en pantalla
      accessorKey: 'id',//para que lea la tabla
      footer: '1era columna'
    },
    {
      header: "Name",
      accessorKey: 'name',
      footer: '2da columna'
    },
    {
      header: "Lastname",
      accessorKey: 'lastname',
      footer: '3era columna'
    },
    {
      header: "Email",
      accessorKey: "email",
      footer: '4ta columna'
    },
    {
      header: "Country",
      accessorKey: "country",
      footer: '5ta columna'
    }
  ];
```

```

    },
    {
      header: "DateOfBirth",
      accessorKey: "dateOfBirth",
      footer: '6ta columna'
    }
  ]
  const table = useReactTable({ data, columns, getCoreRowModel:
getCoreRowModel() });
  return (
    <div>
      <table>
        <thead>
          {
            table.getHeaderGroups().map(headerGroup => (
              <tr key={headerGroup.id}>
                {
                  headerGroup.headers.map(header => (
                    <th key={header.id}>
                      {flexRender(header.column.columnDef.header, header.g
getContext()
                      ))}
                    </th>
                  ))
                }
              </tr>
            ))
          }
        </thead>
        <tbody>
          {
            table.getRowModel().rows.map(row => (
              <tr key={row.id}>
                {row.getVisibleCells().map((cell) => (
                  <td>{flexRender(cell.column.columnDef.cell, cell.getCont
ext())}</td>
                  )))}
                </tr>
              ))
            }
          </tbody>
          <tfoot>
            {
              table.getFooterGroups().map(footerGroup =>(
                <tr key={footerGroup.id}>
                  {footerGroup.headers.map(footer=>(
                    <th key={footer.id}>
                      {flexRender(footer.column.columnDef.footer,

```

```

        footer.getContext()
    })
</th>
    })
</tr>
    })
}
</tfoot>
</table>
</div>
)
}
export default SimpleTable

```

y en el navegador :

970	Jacinta	Darwin	juarwinqx@uiuu.com	Philippines	6/31/2023
971	Tabbie	Perks	tpersqy@nih.gov	France	7/15/2023
972	Albert	Shoebottom	ashoebottomqz@ezinearticles.com	Madagascar	5/3/2023
973	Cyrill	Terrill	cterrillr0@php.net	Portugal	3/27/2023
974	Karine	Leach	kleachr1@sina.com.cn	Portugal	3/8/2023
975	Iolande	Rowlstone	irowlstoner2@omniture.com	Maldives	10/6/2023
976	Penny	Tizzard	ptizzardr3@seesaa.net	China	1/17/2024
977	Patrice	Coweuppe	pcoweupper4@reddit.com	Portugal	5/13/2023
978	Lyndsie	Gurnett	lgurnett5@usda.gov	Cyprus	12/30/2023
979	Myrtie	Lovell	mlovellr6@princeton.edu	Madagascar	2/28/2023
980	Tome	Barfield	tbarfieldr7@sogou.com	Russia	4/1/2023
981	Arlyn	Panas	apanasr8@cocolog-nifty.com	Mexico	10/2/2023
982	Rolf	Sherwyn	rsherwynr9@about.com	Chile	9/3/2023
983	Glori	Jostan	gjostanra@npr.org	Netherlands	8/1/2023
993	Virgil	Gallaher	vgallaherrk@hugedomains.com	China	10/25/2023
994	Oran	Rapley	orapleyrl@apple.com	China	10/24/2023
995	Rowe	Abeles	rabelesrm@utexas.edu	Argentina	3/20/2023
996	Verene	Melwall	vmelwallm@quantcast.com	China	12/28/2023
997	Rossy	Simonini	rsimoniniro@bandcamp.com	Indonesia	6/5/2023
998	Merrel	Alywen	malywenrp@blogtalkradio.com	Philippines	5/2/2023
999	Dominic	Trussler	dtrusslerrq@rakuten.co.jp	Indonesia	5/5/2023
1000	Valaree	Belcham	vbelchamrr@delicious.com	Sweden	11/16/2023

1era columna	2da columna	3era columna	4ta columna	5ta columna	6ta columna
--------------	-------------	--------------	-------------	-------------	-------------

Vamos al archivo index.css

```

Table.jsx  main.jsx  MOCK_DATA.json  App.css  index.css
vite-project > src > main.jsx
1  import React from 'react'
2  import ReactDOM from 'react-dom/client'
3  import App from './App.jsx'
4  import './index.css'
5
6  ReactDOM.createRoot(document.getElementById('root')).render(
7    <React.StrictMode>
8      <App />
9    </React.StrictMode>,
10 )
11

```

En el archivo index.css

```

body{
  padding: 7rem; /*espaciado entre los elementos*/
}
table{
  border-collapse: collapse;
  border-spacing: 0;
  width: 100%; /*que ocupe todo su ancho del contenedor*/
  border: 1px solid #ddd;
}
thead, tfoot{
  background: #333;
  color: white;
}

```

localhost:5173

ID	Name	Lastname	Email	Country	DateOfBirth
1	Ronni	Grier	rgrier0@drupal.org	Serbia	2/19/2023
2	Haily	Britcher	hbritcher1@indiegogo.com	Hungary	6/3/2023
3	Marco	Bunford	mbunford2@tuttocitta.it	China	5/12/2023
4	Denise	Hogsden	dhogsden3@earthlink.net	Brazil	1/2/2024
5	Dorena	Lempke	dlempke4@goo.gl	Mexico	10/1/2023
6	Kaylil	McFeate	kmcfeate5@amazonaws.com	China	8/29/2023
7	Hammad	O'Regan	horegan6@yahoo.com	Czech Republic	9/28/2023
8	Kennan	Figura	kfigura7@accuweather.com	Indonesia	5/31/2023
992	Valencia	Vanaacci	vvanaaccirj@independent.co.uk	United States	9/25/2023
993	Virgil	Gallaher	vgallaherrk@hugedomains.com	China	10/25/2023
994	Oran	Rapley	orapleyrl@apple.com	China	10/24/2023
995	Rowe	Abeles	rabelesrm@utexas.edu	Argentina	3/20/2023
996	Verene	Melwall	vmelwallm@quantcast.com	China	12/28/2023
997	Rossey	Simonini	rsimoniniro@bandcamp.com	Indonesia	6/5/2023
998	Merrel	Alywen	malywenrp@blogtalkradio.com	Philippines	5/2/2023
999	Dominic	Trussler	dtrusslerq@rakuten.co.jp	Indonesia	5/5/2023
1000	Valaree	Belcham	vbelchamrr@delicious.com	Sweden	11/16/2023

Terminamos los estilos

```

body{
  padding: 7rem; /*espaciado entre los elementos*/
}

```

```

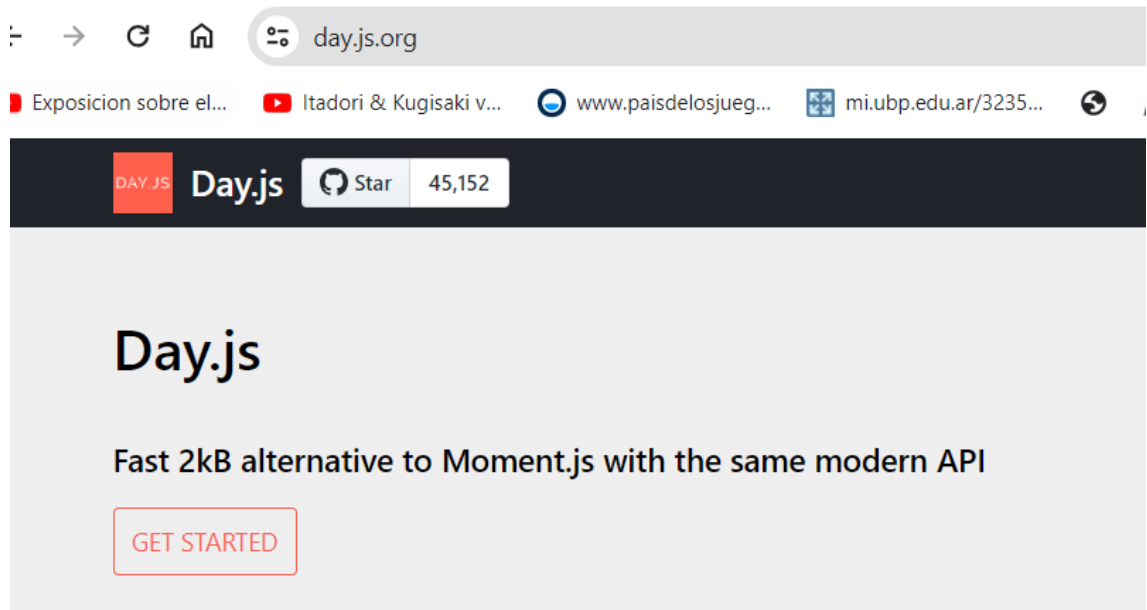
table{
  border-collapse: collapse;
  border-spacing: 0;
  width: 100%;/*que ocupe todo su ancho del contenedor*/
  border: 1px solid #ddd;
}
/*cabeceras y pie de pagina*/
thead, tfoot{
  background: #333;
  color:white;
}
/*para las filas*/
th,td{
  text-align: left;
  padding: 16px;
}
/*cuando sea una fila impar*/
tr:nth-child(even){
  background: #ccddd8;
}

```

ID	Name	Lastname	Email	Country	DateOfBirth
1	Ronni	Grier	rgrier0@drupal.org	Serbia	2/19/2023
2	Haily	Britcher	hbritcher1@indiegogo.com	Hungary	6/3/2023
3	Marco	Bunford	mbunford2@tuttocitta.it	China	5/12/2023
4	Denise	Hogsden	dhogsden3@earthlink.net	Brazil	1/2/2024
5	Dorena	Lempke	dlempke4@goo.gl	Mexico	10/1/2023

Ahora procesamos los datos

Modificamos la fecha con dia mes año para ello usamos la biblioteca dayjs



Npm i dayjs

```
SimpleTable.jsx ×  main.jsx  MOCK_DATA.js
vite-project > src > componentes > SimpleTable.jsx > Sim
1  //import React from 'react'
2  import { useReactTable, getCoreRowModel, t
3
4  import data from "../MOCK_DATA.json";
5  import dayjs from 'dayjs'
6
```

```
{
  header: "DateOfBirth",
  accessorKey: "dateOfBirth",
  footer: '6ta columna',
  cell: info => dayjs(info.getValue()).format("DD/MM/YYYY"),
},
```

En el navegador

ID	Name	Lastname	Email	Country	DateOfBirth
1	Ronni	Grier	rgrier0@drupal.org	Serbia	19/02/2023
2	Haily	Britcher	hbritcher1@indiegogo.com	Hungary	03/06/2023
3	Marco	Bunford	mbunford2@tuttocitta.it	China	12/05/2023
4	Denise	Hogsden	dhogsden3@earthlink.net	Brazil	02/01/2024

Ahora queda:

ID	Name y apellido	Email	Country	DateOfBirth
1	Ronni Grier	rgrier0@drupal.org	Serbia	19/02/2023
2	Haily Britcher	hbritcher1@indiegogo.com	Hungary	03/06/2023

Ahora implementamos la función `getPaginationRowModel`

Y luego los botones

```

SimpleTable.jsx X main.jsx X {} MOCK_DATA.js
vite-project > src > componentes > SimpleTable.jsx > SimpleTable.jsx
94 </tfoot>
95 </table>
96 <button>Primera pagina</button>
97 <button>Pagina anterior</button>
98 <button>Pagina siguiente</button>
99 <button>ultima pagina</button>
100 </div>

```

En el navegador queda:

9	Conni Gabey	cgabey8@imdb.com	Canada	03/11/2023
10	Chrissie Olding	colding9@topsy.com	Sweden	26/05/2023
1era columna		4ta columna	5ta columna	6ta columna
Primera pagina Pagina anterior Pagina siguiente ultima pagina				

Para ir a la ultima pagina:

```

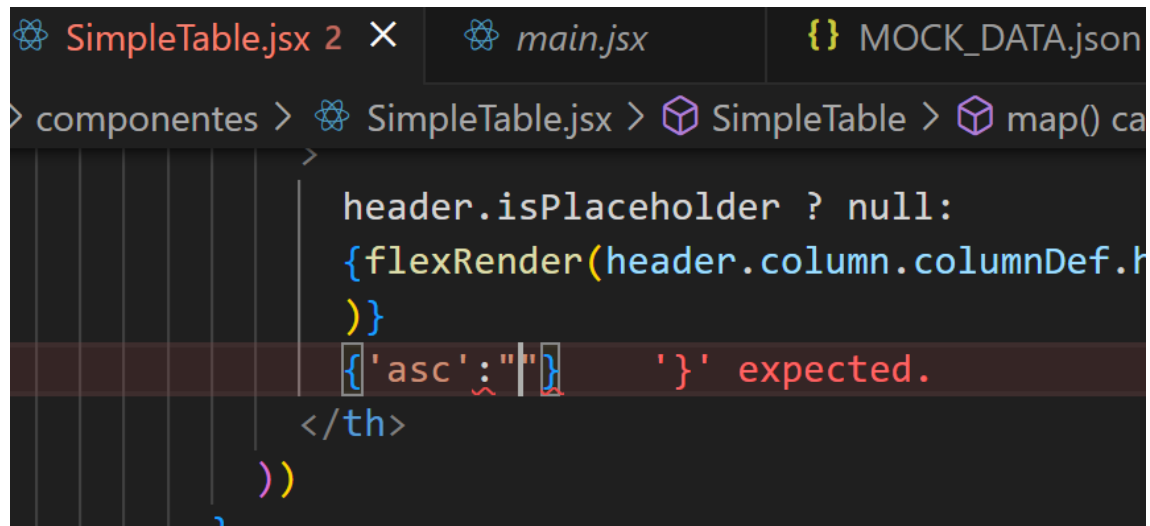
<button onClick={() => table.setPageIndex(table.getPageCount()-1)}>ultima
pagina</button>

```

Para ir a la primera pagina:

```
<button onClick={() => table.setPageIndex(0)}>Primera pagina</button>
```

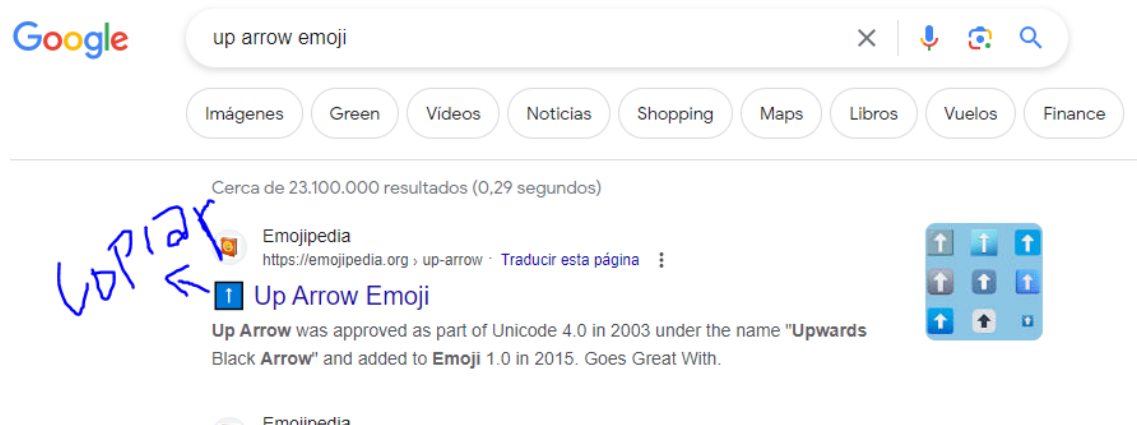
Ordenamos la tabla con la librería `getSortedRowModel`:



```
SimpleTable.jsx 2 X main.jsx {} MOCK_DATA.json
> componentes > SimpleTable.jsx > SimpleTable > map() ca
    header.isPlaceholder ? null:
      {flexRender(header.column.columnDef.h
      )}
    {'asc': ""} '}' expected.
  </th>
  ))
```

Buscamos el icono

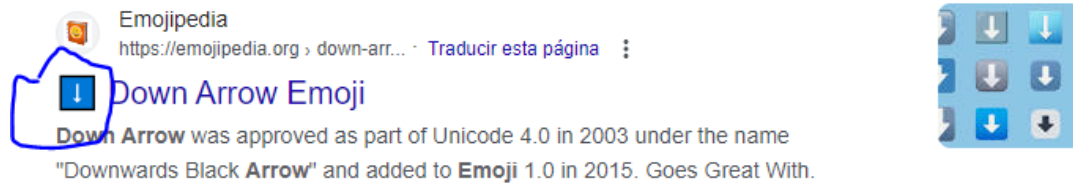
up arrow emoji



```
{'asc': "↑", 'desc': "↓"}
```

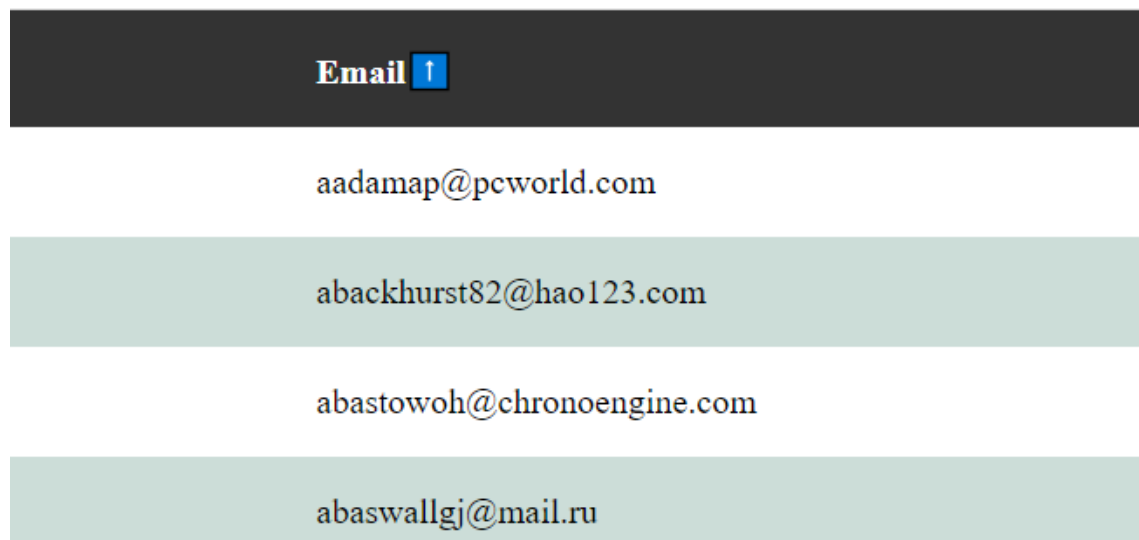

Ahora vamos con el down:

Sugerencia: Limitar esta búsqueda a resultados en idioma **inglés** . Más información para filtrar por idioma

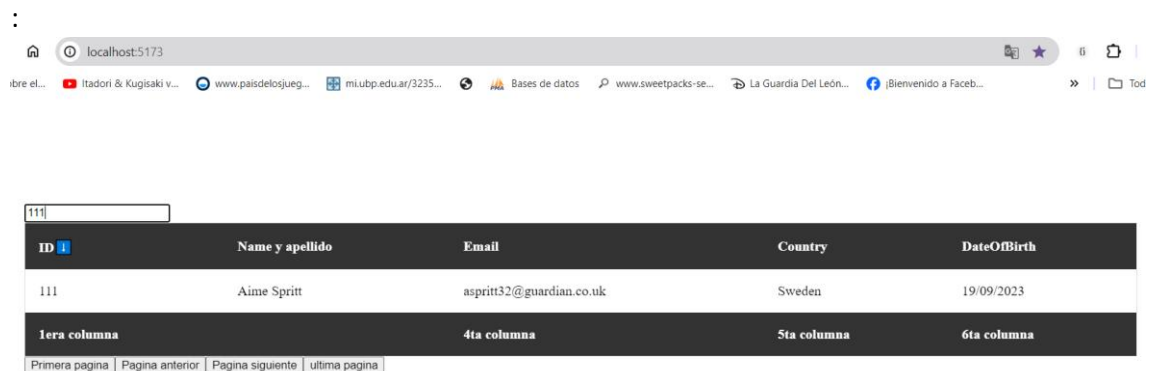


```
{asc: '' : "↑", 'desc': "↓"}
```

Y si hacemos click dentro del navegador , a algunas columnas:



Ahora implementamos la búsqueda o filtro con la herramienta `getFilteredRowModel`



Ahora corto columns del archivo SimpleTable.jsx y lo pego en App.jsx

```
const columns = [//lo que se ve en el navegador
  {
    header: "ID", //lo que se muestra en pantalla
```

```

    accessorKey: 'id',//para que lea la tabla
    footer: '1era columna',
  },
  {
    header: "Name y apellido",
    accessorFn: row => `${row.name} ${row.lastname}`
  },
  //{
  //header: "Name",
  //accessorKey: 'name',
  //footer: '2da columna',
  //},
  //{
  //header: "Lastname",
  //accessorKey: 'lastname',
  //footer: '3era columna',
  //},
  {
    header: "Email",
    accessorKey: "email",
    footer: '4ta columna',
  },
  {
    header: "Country",
    accessorKey: "country",
    footer: '5ta columna',
  },
  {
    header: "DateOfBirth",
    accessorKey: "dateOfBirth",
    footer: '6ta columna',
    cell: info => dayjs(info.getValue()).format("DD/MM/YYYY"),
  },
];

```

Corto desde el SimpleTable

```
import data from "../../MOCK_DATA.json";
```

lo llevo al App.jsx que queda:

```

import SimpleTable from "./componentes/SimpleTable";
import dayjs from 'dayjs';
import data from "../../MOCK_DATA.json";

function App(){
  const columns = [//lo que se ve en el navegador

```

```

    {
      header: "ID", //lo que se muestra en pantalla
      accessorKey: 'id', //para que lea la tabla
      footer: '1era columna',
    },
    {
      header: "Name y apellido",
      accessorFn: row => `${row.name} ${row.lastname}`
    },
    //{
    //header: "Name",
    //accessorKey: 'name',
    //footer: '2da columna',
    //},
    //{
    //header: "Lastname",
    //accessorKey: 'lastname',
    //footer: '3era columna',
    //},
    {
      header: "Email",
      accessorKey: "email",
      footer: '4ta columna',
    },
    {
      header: "Country",
      accessorKey: "country",
      footer: '5ta columna',
    },
    {
      header: "DateOfBirth",
      accessorKey: "dateOfBirth",
      footer: '6ta columna',
      cell: info => dayjs(info.getValue()).format("DD/MM/YYYY"),
    },
  ];
  return (
    <div>
      <SimpleTable data={data} columns={columns}/>
    </div>
  )
}
export default App

```

```

//archivo SimpleTable.jsx'
import { useReactTable, getCoreRowModel, flexRender,
getPaginationRowModel,

```

```

    getSortedRowModel, getFilteredRowModel } from "@tanstack/react-table";
import { useState } from 'react'

function SimpleTable({data,columns}) {

  const [sorting, setSorting] = useState([]);
  const [filtering, setFiltering]=useState("");
  const table = useReactTable({
    data, columns, getCoreRowModel: getCoreRowModel(),
    getPaginationRowModel: getPaginationRowModel(), getSortedRowModel:
getSortedRowModel(),getFilteredRowModel: getFilteredRowModel(),
    state: {
      sorting,
      globalFilter: filtering,
    },
    onSortingChange: setSorting,
    onGlobalFilterChange: setFiltering,
  });
  return (
    <div>
      <input type="text" value={filtering}
onChange={(e)=>setFiltering(e.target.value)}/>
      <table>
        <thead>
          {
            table.getHeaderGroups().map(headerGroup => (
              <tr key={headerGroup.id}>
                {
                  headerGroup.headers.map(header => (
                    <th key={header.id}
                      onClick={header.column.getToggleSortingHandler()}
                    >
                      {
                        header.isPlaceholder ? null:
flexRender(header.column.columnDef.header,
header.getContext()
                      )}
                      {
                        { asc: '↑', desc: '↓'
}[header.column.getIsSorted() ?? null]
                      }
                    </th>
                  ))
                )}
              </tr>
            ))
          }
        </thead>
        <tbody>

```

```

    {
      table.getRowModel().rows.map(row => (
        <tr key={row.id}>
          {row.getVisibleCells().map((cell) => (
            <td>{flexRender(cell.column.columnDef.cell,
cell.getContext())}</td>
          ))}
        </tr>
      ))
    }
  </tbody>
  <tfoot>
    {
      table.getFooterGroups().map(footerGroup => (
        <tr key={footerGroup.id}>
          {footerGroup.headers.map(footer => (
            <th key={footer.id}>
              {flexRender(footer.column.columnDef.footer,
                footer.getContext()
              )}
            </th>
          ))}
        </tr>
      ))
    }
  </tfoot>
</table>
<button onClick={() => table.setPageIndex(0)}>Primera
pagina</button>
<button onClick={() => table.previousPage()}>Pagina
anterior</button>
<button onClick={() => table.nextPage()}>Pagina siguiente</button>
<button onClick={() => table.setPageIndex(table.getPageCount() -
1)}>ultima pagina</button>
</div>
)
}
export default SimpleTable

```

ID	Name y apellido	Email	Country	DateOfBirth
1	Ronni Grier	rgrier0@drupal.org	Serbia	19/02/2023
2	Haily Britcher	hbritcher1@indiegogo.com	Hungary	03/06/2023
3	Marco Bunford	mbunford2@tuttocitta.it	China	12/05/2023
4	Denise Hogsden	dhogsden3@earthlink.net	Brazil	02/01/2024
5	Dorena Lempke	dlempke4@goo.gl	Mexico	01/10/2023
6	Kaylil McFeate	kmcfate5@amazonaws.com	China	29/08/2023
7	Hammad O'Regan	horegan6@yahoo.com	Czech Republic	28/09/2023
8	Kennan Figura	kfigura7@accuweather.com	Indonesia	31/05/2023
9	Conni Gabey	cgabey8@imdb.com	Canada	03/11/2023
10	Chrissie Olding	colding9@topsy.com	Sweden	26/05/2023
1era columna		4ta columna	5ta columna	6ta columna
Primera pagina	Pagina anterior	Pagina siguiente	ultima pagina	