INSTALACIÓN DE REACT CON VITE

Material sacado del video de youtube de Miguel Durán (@midudev):

<https://www.youtube.com/watch?v=XYpadB4VadY>

1. Procedemos a la instalación con el siguiente comando desde el terminal o git bash o el terminal que utilices:

$ npm create vite@latest

? **Project name:** » vite-projectfrontend

√ **Project name:** ... frontend

? **Select a framework:** » - Use arrow-keys. Return to submit.

> Vanilla

Vue

React

Preact

Lit

Svelte

Others

√ **Select a framework:** » Vanilla

? **Select a variant:** » - Use arrow-keys. Return to submit.

> JavaScript

TypeScript

√ **Select a variant:** » JavaScript

Scaffolding project in D:\Programación\PROYECTOS\xpaccount\frontend...

Done. Now run:

cd frontend

npm install

npm run dev

Elegimos las opciones “Vanilla” (EcmaScript 6 puro) y JavaScript para poder instalar React desde cero.

1. Instalamos el plugin de react con la versión exacta (-E)

annwho@DESKTOP-NSHOUU3 MINGW64 /d/Programación/PROYECTOS/xpaccount (main)

$ cd frontend/

annwho@DESKTOP-NSHOUU3 MINGW64 /d/Programación/PROYECTOS/xpaccount/frontend (main)

$ npm install @vitejs/plugin-react -E

added 73 packages, and audited 74 packages in 13s

8 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

Ahora si vamos al package.json veremos que no existe ninguna dependencia instalada de React todavía:

{

  "name": "frontend",

  "private": true,

  "version": "0.0.0",

  "type": "module",

  "scripts": {

    "dev": "vite",

    "build": "vite build",

    "preview": "vite preview"

  },

  "devDependencies": {

    "vite": "^4.2.0"

  },

  "dependencies": {

    "@vitejs/plugin-react": "3.1.0"

  }

}

1. Por lo tanto procedemos a la instalación de las dependencias necesarias para que funcione react con la versión exacta (-E): react (contiene la biblioteca de React hooks y useX) y react-dom( sirve de vínculo con el navegador):

$ npm install react react-dom -E

added 4 packages, and audited 78 packages in 2s

8 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

1. Montamos nuestra configuración de Vite creando el fichero vite.config.js (siempre es igual):

*import* { defineConfig } *from* "vite";

*import* react *from* "@vitejs/plugin-react";

*export* *default* *defineConfig*({

  plugins: [*react*()],

});

Ahora, como vemos en nuestro index.html, tenemos un script que carga desde el inicio nuestro javascript main.js por lo que en él, borraremos todo lo que trae por defecto e importaremos el createRoot del react-dom. Luego, con createRoot, nos traemos el Id “app” de nuestro index.html y lo guardamos en la variable “root”. Finalmente renderizamos la variable root:

*import* {createRoot} *from* "react-dom/client";

const *root* = *createRoot*(*document.getElementById*("app"));

*root.render*(<h1>Hola, Mundo!"</h1>)

Esto al ejecutarlo con el comando “npm run dev”, nos dará el siguiente error:

X [ERROR] The JSX syntax extension is not currently enabled

main.js:4:12:

4 │ root.render(<h1>Hola, Mundo!"</h1>)

╵ ^

The esbuild loader for this file is currently set to "js" but it must be set to "jsx" to be able to parse JSX syntax. You can use "loader: { '.js': 'jsx' }" to do that.

Esto significa que al instalar de manera manual React, éste no sabe distinguir entre los archivos de Javascript .js y los de React .jsx, por lo que habrá que modificar la extensión de nuestro archivo main.js a main.jsx y modificarlo también dentro de index.html

D:\Programación\PROYECTOS\xpaccount\frontend> npm run dev

> frontend@0.0.0 dev

> vite

VITE v4.2.0 ready in 845 ms

➜ Local: http://localhost:5173/

➜ Network: use --host to expose

➜ press h to show help

22:13:18 [vite] page reload main.jsx

Ahora ya vemos que la web muestra nuestro “Hola Mundo”.

1. Una vez lo tenemos funcionando, instalamos las dependencias para desarrolladores que nos ayudarán al proceso de “lintig”, es decir, que el código quede más “limpio” tanto en el servidor como en el navegador. Nos informa de los errores y warnings de nuestro código, variables que no se usan, etc. En este caso usaremos la dependencia llamada “**standard**”:

PS D:\Programación\PROYECTOS\xpaccount\frontend> npm install standard -D

added 208 packages, and audited 286 packages in 15s

92 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

1. Creamos la carpeta “src” junto con todos sus archivos:
   1. El primero de todos, App.jsx. Creamos la función App y renderizamos una etiqueta h1 (por ejemplo) para ver que se muestra en nuestra web. Es importante ir comprobando que a cada paso que demos, vamos avanzando, ya que si existe un problema, será más fácil de encontrar. Luego vamos a main.jsx y sustituimos la etiqueta h1 dentro de root.render(<h1>Hola mundo</h1>) por nuestro componente: root.render(<App/>).

Prueba técnica para Juniors y Trainees de React en Live Coding

APIs:

* Facts Random: <https://catfact.ninja/fact>
* Imagen Random: <https://cataas.com/cat/says/hello>

Recupera un hecho aleatorio de gatos de la primera API y muestra una imagen de un gato con la primera palabra del hecho recuperado usando la segunda API.