

npm run dev

<https://plucky-crabapple-e82.notion.site/Recursos-Curso-Profesional-de-React-cc84a4ca3be841dfbacab0c71467e2be>

Libreria de Formularios

react hook form

<https://react-hook-form.com/>

<https://codigofacilito.com/videos/introduccion-a-hooks>

Clase 30

Hook

Un hook es una funcion que retorna funciones , valores ,objetos o tipo de dato dentro de componentes .
Y siguen reglas .

Clase 31

Reglas de los Hooks

- Los hooks se tiene que llamar en nivel superior
- No usar hooks dentro de funcion , solo en Componentes de React
- No se puede llamar hooks de dentro de condicionales (if(){}else{}) .

Clase 32

Como funciona useEffect

- no debe usar async porque es un anti-patron
- se recomienda tener una funcion async dentro de useEffect
- setup : callback
- [] : dependencias de variables
- usado para llamado a la api , conexion a librerias de terceros .
- Limpiar dependencia cuando se destruye un component.
- Conexion y desconexion de servidor.

Clase 33

Creando Hook reusable

- Podemos crear hooks personalizados
- Implementar y logica de crear un hook
- Aumentar la reusabilidad del codigo
- Encapsulamiento y abstraído del codigo
- un hook si puede llamar un hook

Clase 35

Que son las ref

- es un hook
- sirve para referenciar algo
- es para utilizado para referencia valores funciones objetos
- devuelve un objeto
- actualiza el valor sin la necesidad de que sea el valor reactivo
- se actualiza el valor pero no el component
- en useState si se cambia de valor , el component se actualiza(render la pagina).
- no hace re-rendeo de la pagina
- son mutables , se puede modificar
- no se debe usar ref.current en HTML-JSX

Clase 38

implementado forwarded

- guarda elementos para cambiar medida , atributos html
- manipulado para el padre del componente para utilizar la referencia.

Clase 39

implementacion de useImperativeHandle

- usado para crear component reusables
- El componente padre puede acceder a valores del componente hijo
- se implementa en el hijo y se llamar o usa en el padre .

Clase 40 - 41

Introduccion a Estilo

Clase 42

Implementando Modulo de CSS

- Modulos de css , esto archivos crean clases unicas con un Hash unica que no se repite. para evitar mezcla con otro clase.
- archivoCSS.module.css
- se recomienda usar camelCase
- No usar "-" nombrar la clase css

Modulo 07

Clase 43

Como funciona el Routing

- uso de react router
<https://reactrouter.com/en/main>
- no se usa en rutas camelCase sino "-"
- Cambio de rutas
- Apilar rutas
- Arreglo de Push - Pop

Clase 44

Qué es el Client Side Routing

- componetes para cambio de rutas.
- permite escalar varias rutas.
- el router crear un arreglo de rutas.

Clase 44

Configuracion routing en nuestro proyecto

Clase 48

definiendo parametros en las rutas

- rutas con parametros
- el parametro si usa camelCase
- path: "/detail/:eventId",
- le enviamos un argumento en el navegado

Clase 49

rutas anidadas

- se tiene especificar nuevas rutas dentro de la ruta principal
- Decediente directo
- /profile/my-account
- uso de Outlet en component padre

Clase 50

Obteniendo valores de los parametros : useParams

- leer ID desde url

Clase 51 - 52

- Creacion de usuario de ticket y API
- Uso de postman

Clase 53

useFetch

- Usado para llamar a una api
- Libreria similar axios
- Objecto que esta en windwos(navegador)
- recibe un url de la api
- la respuesta es una promesa del servidor
- la promesa se convierte en formato JSON
- tiene muchos parametros
- puede enviar para guarda en una base de datos .

Clase 54

- Incrementar la reusabilidad

Clase 55

Agregando Paginacion

-

Clase 57

- Conversion de fechas
- date fns

<https://date-fns.org/>

Modulo 08

Clase 59

Qué es un State Management

- Soluciona la problema para solicitar la misma informacion
- ReactJS dio la solucion llamado Store(como una base de datos)
- Como una fuente de la verdad
- Metodologia de centralizar todo en solo lugar
- Librerias de manejadores de estados.
 - Redux
 - Zustand
 - Mobx
 - Recoil

Clase 60

Qué es Zustand y Cómo Funciona

- tiene multiples store para actualizar los componentes.

Clase 61

Instalando Zustand

-

Clase 63

Agregando icono de like y unlike a los eventos

-

Modulo 11

Clase 70

Que es memoization

- Tecnica de optimizacion
- evita calculos innecesarios
- evita doble comput
- Ejemplo : codigo fibonacci
- es recomendado usarlo cuando una lista crece (10000 elementos)
- Si no se aplica se renderiza toda la lista (10000 elementos) provocando lentitud
- es un metodo que estan react
- evita re-render innecesarios

Clase 71

Que es memo

- es una funcion
- Se usa en componente a memoriza (evitar el re-renderizar)
- Sus propiedad no ha cambiado
- Aplicable a lista de peso
- Ayuda al performace
- Tiene un costo de memoria

Clase 72

Recreacion de funciones (useCallback)

- cuando react se actualiza se recrea
- Se usa en funciones
- Funciona como un cache de funciones
- Esta funciona cuando se crea por primera vez , se queda memorizada
- Es decir , en memoria queda guarado y cuando se renderiza la pagina
- ya no se crea la funcion sino se consulta en cache la funcion.
- cuando hay un parametro como en una url , se usa [dependencias]

- para que vuelva a ser un componente reactivo solo si tiene parametros
- es decir , si tiene parametros solo se vuelve a crear
- recomendado para llamadas a la API

Clase 73

Que es useMemo()

- Es usado para calculos pequeños
- Guarda en cache el resultado de una operacion
- usado en variables y funciones

Clase 74

React dev tools

Clase 75

Dependencias useEffect

Clase 76

Strict Mode

- Solo pasa en modo estricto en desarrollo
- se puede aplicar para cierto componenten tambien
- ayuda a encontrar errores

Clase 77

Error Boundary

-

Clase 78

suspense

- suspende la interfaz cuando esta cargando una pagina
- es para mostrar un spinner
- es un componente ,
- Ayuda a ya no utiliza useEffect