Esta clase va a ser

- **grabada**

Certificados oficialmente por

CODERHOUSE

Semana 5. REACT JS

Routing, navegación y eventos

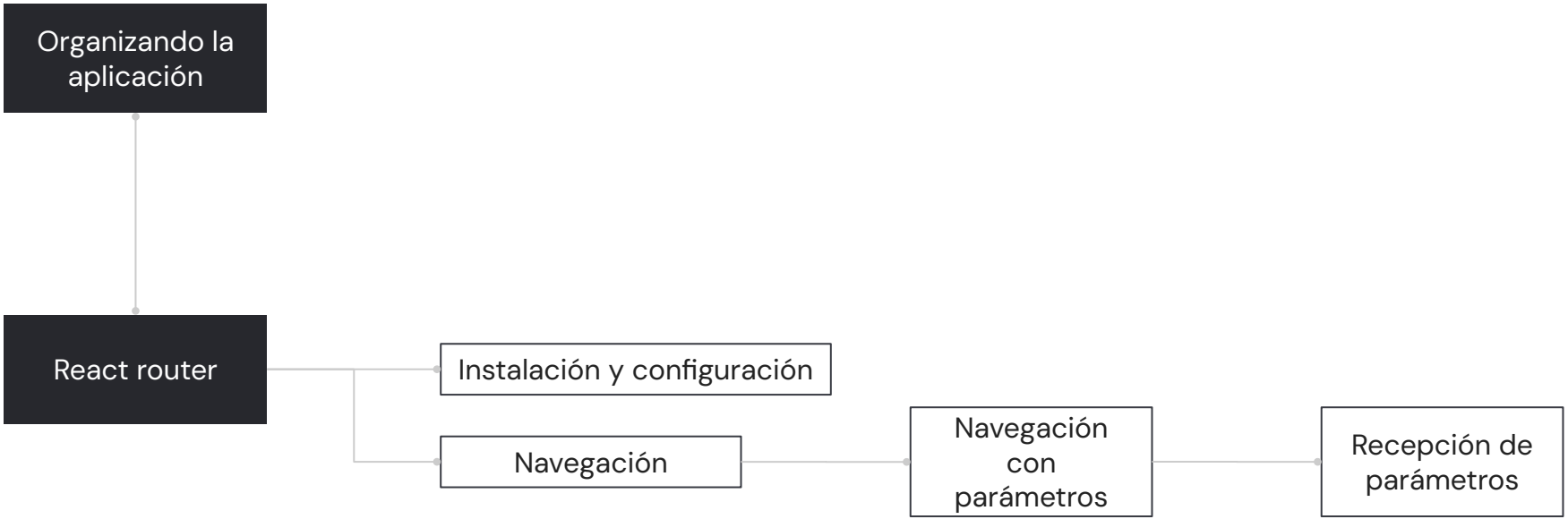
Certificados oficialmente por

CODERHOUSE

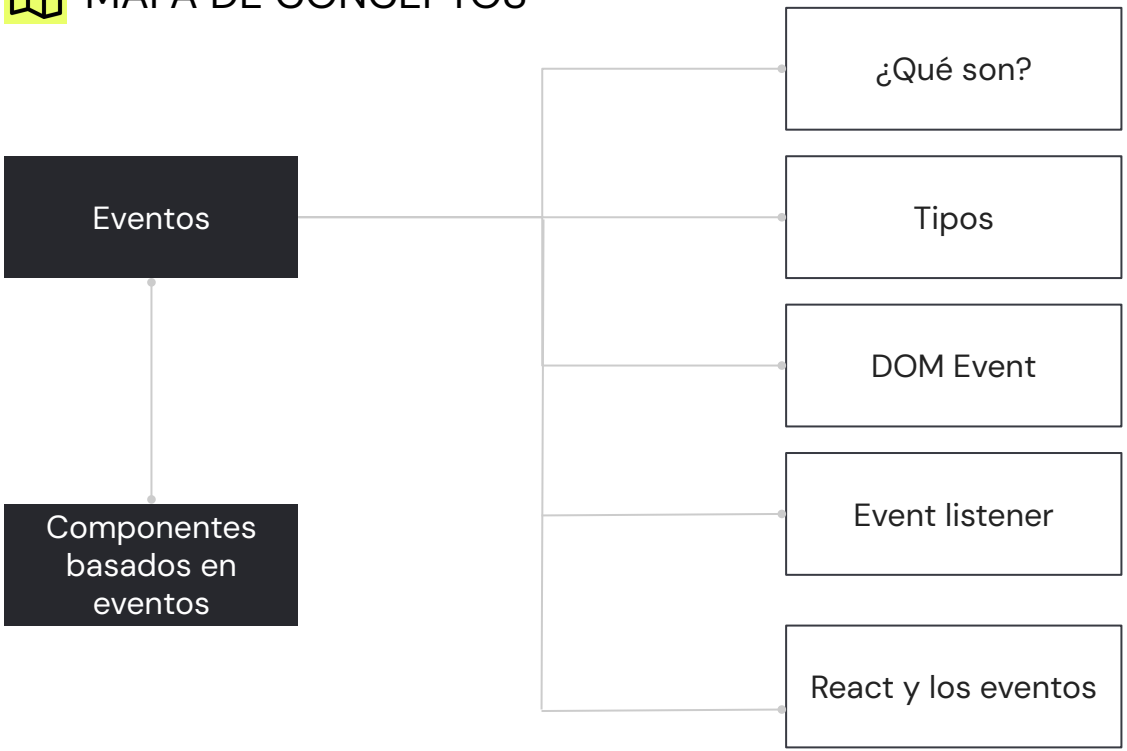
Objetivos de la clase

- **Organizar** nuestra aplicación.
- **Configurar** la navegabilidad entre componentes.
- **Entender** el sistema de eventos de React y su implementación.
- **Diseñar** componentes orientados a eventos.

MAPA DE CONCEPTOS



MAPA DE CONCEPTOS



REPASO

Semana 5

En eL contenido pregrabado de esta semana aprendiste:

- ✓ Cómo implementar navegación en una SPA de React.
- ✓ Cómo crear rutas estáticas y dinámicas utilizando parámetros URL.
- ✓ Incorporar React Router a tu proyecto.
- ✓ Utilizar eventos sintéticos creando event listeners inline en elementos JSX.
- ✓ Crear componentes que respondan a callbacks recibidos mediante props.

VIDEO N° 5.1 - Routing

Recuperamos el tema visto

Para crear una buena **navegación** en React, necesitamos un Enrutador/Router que nos permita definir diferentes rutas o “paths” en nuestra aplicación e indicar qué componente queremos mostrar en cada una de ellas.

La facilidad con la que nuestra aplicación permite **agregar funcionalidades y navegarlas** es un factor clave en términos de experiencia y escalabilidad.

VIDEO N° 5.2 - Navegar una ruta

Recuperamos el tema visto

React Router es una librería standard para crear rutas en una SPA de React. Una vez instalada en nuestro proyecto, podemos utilizar los componentes **BrowserRouter, Routes, y Route** para configurar las distintas rutas que queremos crear en nuestra app.
También utilizamos el componente **Link** para remplazar enlaces internos de nuestra web.
React Router también nos brinda una extensa lista de hooks a utilizar, como **useParams** que nos permite obtener el parámetro URL en el que se encuentra el usuario.

VIDEO N° 5.3 - Eventos

Recuperamos el tema visto

Eventos - Revemos el patrón de **EventListeners** en JavaScript y sus particularidades en React.
Los **eventos sintéticos** tienen algunas ventajas en lo relativo a compatibilidad entre navegadores y performance. Para utilizarlos solo debemos crear eventos “inline” en los elementos JSX.

VIDEO N° 5.4 - Componentes basados en eventos

Recuperamos el tema visto

Componentes orientados a Eventos.
Al crear un componente, puedes dejar algunas tareas a cargo del componente padre que lo renderice.
Este patrón, lo podemos implementar enviando una función a través de Props, y luego en el componente hijo asignarla como handler a un evento de cualquier elemento JSX.

Puesta en común microdesafío

¡Vamos a recuperar lo trabajado durante la semana!

Duración: **10 minutos**.

CODERHOUSE

PUESTA EN COMÚN - MICRODESAFÍO

Homepage

Consigna

Utiliza React Router para crear una ruta de bienvenida a nuestra app y una ruta donde podamos ver el listado de tareas. No olvides agregar navegación para desplazarte de un componente a otro.

[Código inicial en stackblitz]

Tips

- ✓ Recuerda que al utilizar react-router no queremos que la página "*recargue*", utiliza el componente Link para generar enlaces.
- ✓ react-router-dom ya está instalado en la aplicación de stackblitz: no olvides importar los componentes necesarios.

Recuerda:
Los microdesafíos si bien no requieren entrega, en cada clase en vivo se realizará una puesta en común en torno a ellos. Te recomendamos realizarlo para poner en práctica los contenidos pregrabados.

Paso a paso

- Creamos el componente **Home.**
- Utilizando el componente **Route** creamos dos rutas, una para Home y otra para TodoList.
- Creamos el wrapping de las rutas utilizando **Routes y BrowserRouter.**
- Creamos un menú para navegar a ambas rutas, utilizando componente **Link.**

¡Buen trabajo! 😎

Agregar un router a tu app

Vamos a practicar lo visto

Duración: **15 minutos**

CODERHOUSE

ACTIVIDAD

Agregar un router a tu app

Descripción de la actividad.

En tu aplicación, instala react-router-dom, agrégala al root de tu app, y configura tus rutas apuntando a tu Home. Si tienes otro nombre, o la organizaste distinta, no hay problema.
Si aún no tienes creados todos los componentes de tu aplicación puedes practicar con la siguiente app:

Código

Paso a paso

- Instalamos react router con npm:
  npm react-router-dom.
- Utilizamos **Route** para crear ruta al directorio raíz “/” (ItemListContainer) y al detalle “/item”.
- Modificamos los enlaces de NavBar para utilizar el componente **Link o NavLink.**

Puesta en común

Duración: 10 minutos

CODERHOUSE

Ejemplo en vivo

Crearemos rutas dinámicas en nuestra app para mostrar diferentes productos y categorías. Utilizaremos useParams() para leer los parámetros url y mostrar los elementos correspondientes.

Duración: **15 minutos**

código ejemplo

CODERHOUSE

Paso a paso

- Modificamos la ruta de ItemDetail para tomar un valor dinámico: **/item/:id**
- Creamos un Link en el componente Item para navegar a la ruta de su detalle **/item/{id}**
- Creamos una nueva ruta para mostrar el listado de items de una única categoría: **/category/:id**
- Utilizamos **useParams** para leer el parámetro actual de la URL y solicitar los productos.

Break

¡En 5/10 minutos volvemos!

Ejemplo en vivo

Creamos un input y le asignamos un evento sintético “onChange”, para guardar su valor en el estado del componente.

Duración: **10 minutos**

CODERHOUSE

Paso a paso

- Creamos el **Input** con el atributo **onChange.**
- Creamos la función **handleOnChange** y la asignamos como callback del evento onChange.
- Recibimos el objeto **event** como parámetro y vemos sus propiedades.
- Creamos un **estado** y le asignamos el valor del input cada vez que éste cambia.

Crear una máscara de Input

Vamos a practicar lo visto

Duración: **15 minutos**

CODERHOUSE

ACTIVIDAD

Crear una máscara de Input

Descripción de la actividad.

En stackblitz crea un input de texto que no permita el ingreso de vocales, cancelando su evento **onKeyDown** en los keys adecuados.
Pista: el synthetic event de keydown tiene varias propiedades, encuentra cuál te puede dar la información de la tecla ;)

Código inicial

Paso a paso

- Crea una función **handleKeyDown** y asígnala como callback del evento **onKeyDown** del elemento indicado.
- Recibe el **objeto de evento** como parámetro y accede al valor de la tecla presionada.
- Realiza alguna validación que compruebe si la tecla presionada es una vocal.
- En caso de que sea vocal, puedes **cancelar la acción predeterminada del evento** y lanzar una alerta o mediante consola un mensaje de error

Puesta en común

Duración: 10 minutos

CODERHOUSE

Ejemplo en vivo

Creamos un componente que responda a un evento, llamando a un callback recibido a través de props. Mediante el callback enviamos información desde el componente hijo hacia el componente donde se creó el callback.

Duración: **10 minutos**

Paso a paso

- Creamos el componente padre con el **event handler** a utilizar.
- Invocamos al componente hijo, enviando como **prop** el event handler.
- Asignamos un evento inline a un elemento dentro del componente hijo, por ejemplo: onClick a un botón.
- Como respuesta al evento onClick, asignamos el event handler recibido mediante props y enviamos algún parámetro a la función.

Preentrega 2

Navega las rutas

CODERHOUSE

PREENTREGA N° 2

Navega las rutas

Para tu segunda preentrega, debes implementar una herramienta de routing, la cual permitirá navegar a través de las diferentes vistas para tu tienda: catálogo principal de productos, catálogo de productos filtrados por categorías, y vista en detalle de un producto.

Debes entrar al link para acceder a la **consigna completa**.

Objetivos

- ✓ Implementar la funcionalidad de navegación entre las diferentes vistas utilizando enlaces y rutas.
- ✓ Desarrollar la navegabilidad básica de la aplicación, permitiendo navegar desde el catálogo al detalle de cada item.

Formato

- ✓ Link a último commit de git. Debe tener el nombre “NavegaLasRutas+Apellido”, por ejemplo “NavegaLasRutas+Fernandez”

¿Preguntas?

Te invitamos a dejar tu pregunta a través del chat

#Coderalert

Ingresa a la **Guía de actividades hacia el Proyecto Final** y realiza la actividad “Sincronizar Counter”.

Ten en cuenta que el desarrollo de la misma será importante para la resolución del Proyecto Final.

Te compartimos los siguientes CoderTips

CODERTIPS

Te recomendamos

- ✓ Hay **diferentes formas de crear “routers”** con React Router; te animo a que investigues en la documentación.
- ✓ Crea una ruta con path “*” para generar una página de “**Not found**”. Te ayudará a no confundir errores en tus componentes con errores de la navegación.
- ✓ El uso de **addEventListener** puede ser más complejo que los eventos “inline” ya que debemos utilizar useEffect para comprobar el montaje/desmontaje.
- ✓ Con el patrón de componentes orientados a eventos, quizá ya estés pensando en crear almacenar en algún componente el **estado del carrito:** ¡espera! La semana que viene veremos una mejor forma de trabajarlo.

Resumen
de la clase hoy

- ✓ Organización de navegabilidad
- ✓ Routing estático
- ✓ react-router-dom
- ✓ DOM y Synthetic events.
- ✓ Diseño de eventos en componentes.

La próxima semana

Los próximos temas que vamos a ver

Contenido Pregrabado

- ✓ Video 6.1 - Contexto
- ✓ Video 6.2 - Nodo Proveedor y Custom Provider
- ✓ Infografía 6.3 - Recapitulando
- ✓ Video 6.4 - Otras técnicas para control condicional - Parte I
- ✓ Video 6.5 - Otras técnicas para control condicional - Parte II
- ✓ Video 6.6 - Render optimization
- ✓ Microdesafío - User Context

Clase en vivo (2h)

- ✓ Context y técnicas de rendering.

CODERHOUSE

La próxima semana

Recuerda que, a partir de ahora, tienes disponible el contenido pregrabado en la plataforma y que es necesario que lo veas en forma previa a la próxima clase.

CODERHOUSE

Opina y valora esta clase

Muchas gracias.

#DemocratizandoLaEducación

¡Gracias por estudiar con nosotros! ✨