

Ejemplo de CP M2

Que paquete deberías instalar para agregar enrutamiento a tu aplicación React.-

A. react-router-dom (El `react-router-dom` paquete contiene enlaces para usar React Router en aplicaciones web.)

¿Qué paquete deberías instalar para agregar enrutamiento a tu aplicación React?

- ☐ A. react-routing
- ☐ B. react-router-dom
- ☐ C. react-routes
- ☐ D. router-react
- ☐ E. routing-react-dom

Borrar la selección

Cual es la forma correcta de enlazar un ruta a un componente en React Router v6

B. `<Route path='/contact' element = {<Contact />} />`

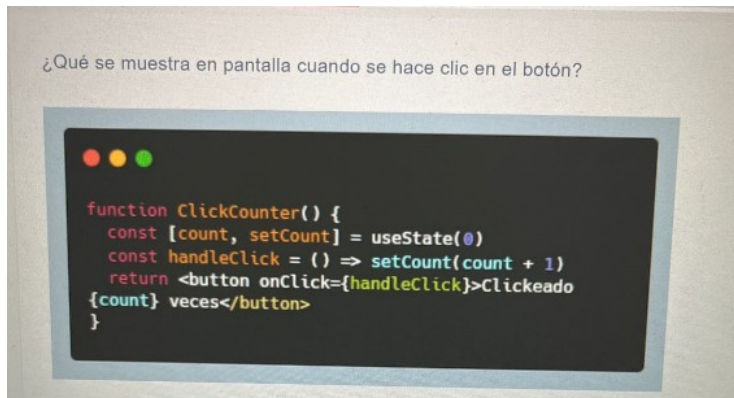
¿Cuál es la forma correcta de enlazar una ruta a un componente en React Router v6?

- ☐ A. `<Router path='/contact' component={Contact} />`
- ☐ B. `<Route path='/contact' element={<Contact />} />`
- ☐ C. `<Route link='/contact' component={<Contact />} />`
- ☐ D. `<Link route='/contact' render={<Contact />} />`
- ☐ E. `<Route to='/contact' output={<Contact />} />`

Borrar la selección

Que se muestra en pantalla cuando se hace clic en el botón..

D. Clickeado 1 veces



☐ A. Clickeado

☐ B. Clickeado -1 veces

☐ C. Clickeado 0 veces

☐ D. Clickeado 1 veces

☒ E. Error

Borrar la selección

Si quieres aplicar estilos específicos para dispositivos móviles, que tipo de Media Query podrías utilizar.

D. (max-width: 768px)

Si quieres aplicar estilos específicos para dispositivos móviles, ¿qué tipo de Media Query podrías utilizar?

☐ A. (audio: yes)

☐ B. (device-height: 480px)

☐ C. (color: black)

☐ D. (max-width: 768px)

☐ E. (min-resolution: 300dpi)

Borrar la selección

Cual es una ventaja de usar framework como Bootstrap.?

E. Facilita la creación de interfaces coherentes y responsivas

¿Cuál es una ventaja de usar un framework como Bootstrap?

- ☐ A. Hace innecesario el uso de CSS
- ☐ B. Mejora automáticamente el rendimiento web
- ☐ C. Permite escribir menos HTML
- ☐ D. Reemplaza completamente a JavaScript
- ☒ E. Facilita la creación de interfaces coherentes y responsivas

Borrar la selección

Como cambiarías el color a rojo si el botón esta deshabilitado..?

D. `const Button = styled.button `color: blue; & : disabled {color: red}`;`

Dado el siguiente código:

```
const Button = styled.button`color: blue;
```

¿Cómo cambiarías el color a rojo si el botón está deshabilitado?

- ☐ A. `Button:disabled { color: yellow; }`
- ☐ B. `const Button = styled.button` color: ${props => props.disabled ? 'green' : 'blue'}; `;`
- ☐ C. `Button { &:disabled { color: green; } }`
- ☒ D. `const Button = styled.button` color: blue; &:disabled { color: red; } `;`
- ☐ E. `Button.disabled { color: purple; }`

Borrar la selección

Cual es la salida del componente 'MessageDisplay'..?
C. Hello, World

¿Cuál es la salida inicial del componente 'MessageDisplay'?

```
function MessageDisplay() {  
  const [message, setMessage] = useState(['Hello',  
    'World'])  
  return <p>{message.join(', ')}</p>  
}
```

☐ A. ["Hello", "World"]

☐ B. Hello World

☐ C. Hello, World

☐ D. HelloWorld

☐ E. Un párrafo vacío

Borrar la selección

El hook 'useReducer' es similar a 'useState' pero es mas adecuado para...?
B. Manejar estados mas complejos que tienen lógicas y transiciones predecibles

El hook 'useReducer' es similar a 'useState' pero es más adecuado para...

☐ A. Validar propiedades en tiempo de ejecución

☐ B. Manejar estados más complejos que tienen lógicas y transiciones predecibles

☐ C. Sincronizar el estado entre múltiples componentes

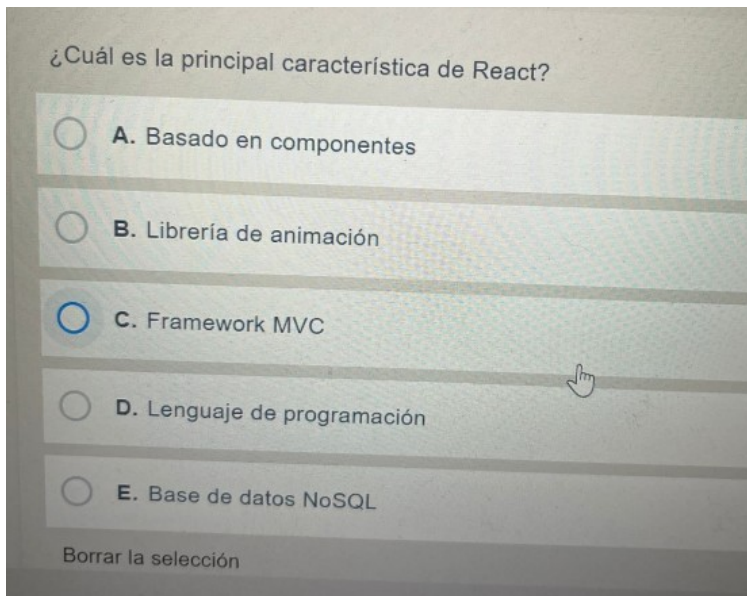
☒ D. Memorizar componentes para evitar re-renderizados

☐ E. Optimizar la renderización de listas grandes

Borrar la selección

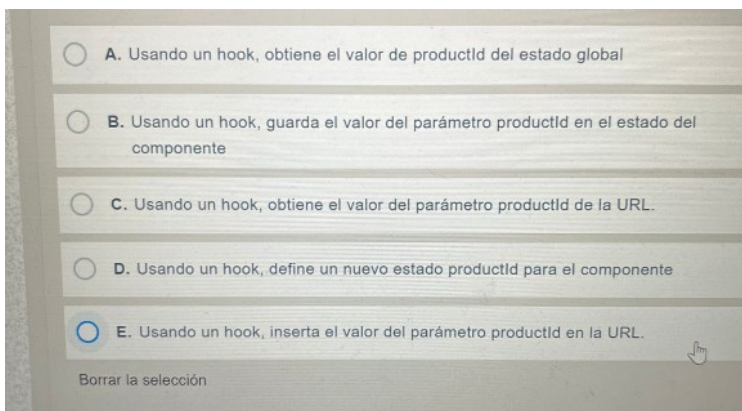
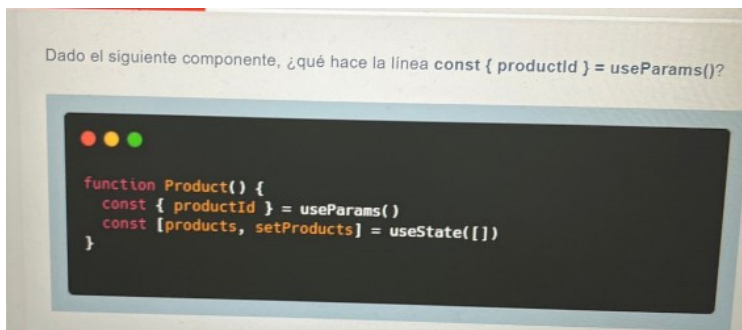
Cual es la principal característica de React..?

A. Basado en componentes.



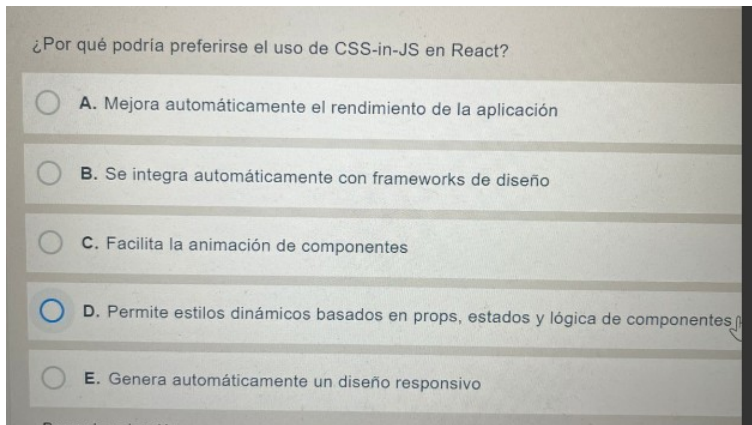
Dado el siguiente componente, que hace la línea `const {productId} = useParams()`..?

C. Usando un hook, obtiene el valor del parámetro productId de la URL



Por que podría preferirse el uso de CSS-in-JS en React..?

D. Permite estilos dinámicos basados en props, estados y lógica de componentes



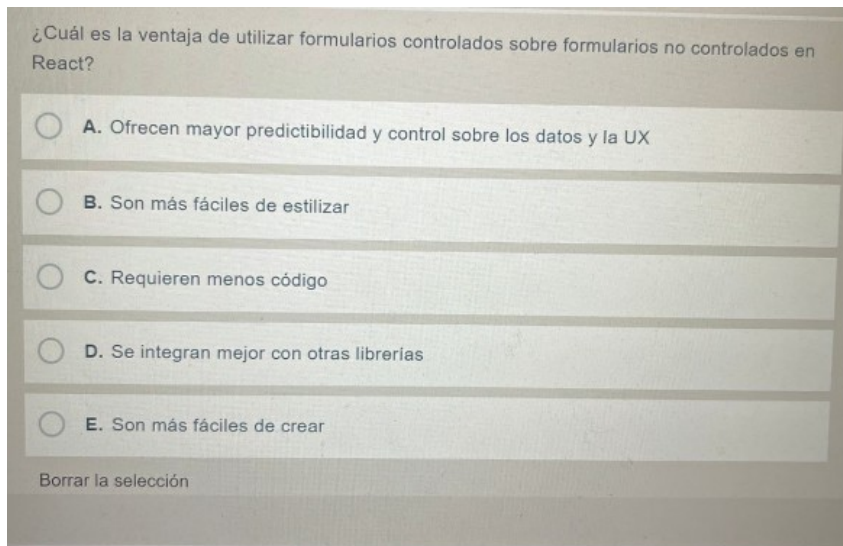
¿Por qué podría preferirse el uso de CSS-in-JS en React?

- ☐ A. Mejora automáticamente el rendimiento de la aplicación
- ☐ B. Se integra automáticamente con frameworks de diseño
- ☐ C. Facilita la animación de componentes
- ☒ D. Permite estilos dinámicos basados en props, estados y lógica de componentes
- ☐ E. Genera automáticamente un diseño responsivo

Borrar la selección

Cual es la ventaja de utilizar formularios controlados sobre formularios no controlados en React..?

A. Ofrecen mayor predictibilidad y control sobre los datos y la UX



¿Cuál es la ventaja de utilizar formularios controlados sobre formularios no controlados en React?

- ☐ A. Ofrecen mayor predictibilidad y control sobre los datos y la UX
- ☐ B. Son más fáciles de estilizar
- ☐ C. Requieren menos código
- ☐ D. Se integran mejor con otras librerías
- ☐ E. Son más fáciles de crear

Borrar la selección

Cual de la siguientes NO es una característica de Redux..?

A. Múltiples 'stores' en una aplicación

¿Cuál de las siguientes NO es una característica de Redux?

- ☐ A. Múltiples 'stores' en una aplicación
- ☐ B. Inmutabilidad del estado
- ☒ C. Estado único y centralizado
- ☐ D. Manejo de efectos secundarios con acciones
- ☐ E. Cambios predecibles

Borrar la selección

Al usar styled-components, como extenderías los estilos de un componente existente..?

C. `const NewButton = styled(Button)`background-color: red;`;`

Al usar 'styled-components', ¿cómo extenderías los estilos de un componente existente?

- ☐ A. `<Button newStyle={{ backgroundColor: 'red' }} />`
- ☐ B. `const NewButton = Button.extend`background-color: red;`;`
- ☐ C. `const NewButton = styled(Button)`background-color: red;`;`
- ☐ D. `const NewButton = styled(Button)``; NewButton.backgroundColor = 'red';`
- ☐ E. `const NewButton = styled.button(Button)`background-color: red;`;`

Borrar la selección

Dentro de un componente conectado, como se suelen despachar acciones..?

A. Usando props proporcionadas por mapDispatchToProps

Dentro de un componente conectado, ¿cómo se suelen despachar acciones?

☐ A. Usando props proporcionadas por mapDispatchToProps

☐ B. Invocando directamente al store

☐ C. Usando this.dispatch()

☐ D. Directamente llamando al action creator

☐ E. Utilizando un hook especial

Borrar la selección

Dado el siguiente código de un componente, que mostrara en pantalla.?

B. El valor de la cuenta es: 0 → Se asume que esta es la respuesta correcta.

Dado el siguiente código de un componente, ¿qué mostrará en pantalla?

```
import React from 'react'
import { useSelector } from 'react-redux'

function CounterDisplay() {
  const count = useSelector((state) => state.counter.value)
  return <div>El valor actual de la cuenta es: {count}</div>
}
```

☐ A. El valor actual de la cuenta es: state.count

☐ B. El valor actual de la cuenta es: 0

☒ C. El valor actual de la cuenta es: undefined

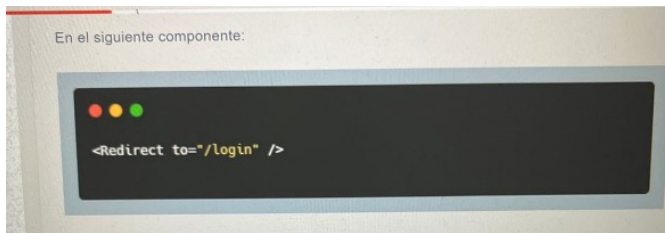
☐ D. Error, el código no compilará

☐ E. El valor actual de la cuenta es: count

Borrar la selección

A que ruta se redirigirá el usuario..?

C. /login



¿A qué ruta se redirigirá el usuario?

☐ A. /redirect

☐ B. /user

☒ C. /login

☐ D. /

☐ E. /home

Borrar la selección

Que es un componente de alto orden (HOC) en relación con los estilos..?

C. Un componente que toma otro componente y le aplica estilos adicionales o lógica relacionada con la estilización

¿Qué es un componente de alto orden (HOC) en relación con los estilos?

☐ A. Una herramienta para minificar y optimizar CSS

☐ B. Un componente que traduce automáticamente CSS a JavaScript

☒ C. Un componente que toma otro componente y le aplica estilos adicionales o lógica relacionada con la estilización

☐ D. Un preprocesador CSS incorporado en React

☐ E. Un linter especializado en estilos

Borrar la selección

E.

¿Cómo aplicarías estilos condicionales basados en props usando 'styled-components'?

- ☐ A. `const Box = styled(props => <div className={props.active ? 'active' : ''} />)`;`
- ☐ B. `const Box = styled.div` background-color: ${props => props.active || 'green'}; `;`
- ☒ C. `<div style={{ backgroundColor: active ? 'green' : 'red' }}></div>`
- ☐ D. `const Box = styled.div` backgroundColor: ${props => props.active && 'green'}; `;`
- ☐ E. `const Box = styled.div` background-color: ${props => props.active ? 'green' : 'red'}; `;`

Borrar la selección

A. → al parecer es la correcta

¿Qué acción se despacha cuando se hace clic en el botón en el siguiente componente?

```
import { useDispatch } from 'react-redux'

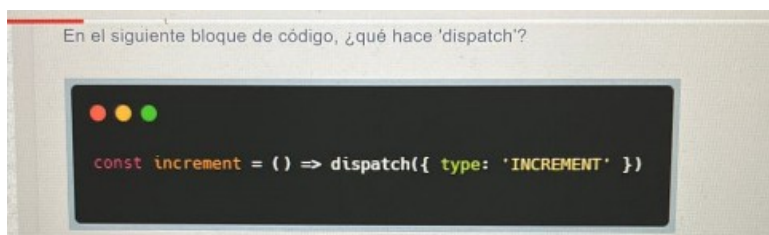
function AddTodoButton() {
  const dispatch = useDispatch()

  return (
    <button
      onClick={() => dispatch({ type: 'ADD_TODO', text: 'Aprender Redux' })}
    >
      Añadir tarea
    </button>
  )
}
```

- ☐ A. `{ type: 'ADD_TODO', text: 'Aprender Redux' }`
- ☐ B. `{ action: 'ADD_TODO', text: 'Aprender Redux' }`
- ☐ C. `{ type: 'ADD_TODO', task: 'Aprender Redux' }`
- ☐ D. `{ type: 'CREATE_TODO', text: 'Aprender Redux' }`
- ☐ E. Ninguna acción, el código tiene errores

Borrar la selección

A.



☐ A. Despacha una acción con el tipo 'INCREMENT' al store

☐ B. Llama a un action creator llamado 'INCREMENT'

☐ C. Actualiza el estado de 'INCREMENT' en el store

☐ D. Conecta el componente al reducer 'INCREMENT'

☐ E. Suscribe 'INCREMENT' a cambios en el store

Borrar la selección

A.

¿Cuál es el propósito de 'e.preventDefault()' en un formulario de React?

☐ A. Evitar que el formulario se envíe de la manera predeterminada

☐ B. Validar el formulario

☒ C. Enviar el formulario

☐ D. Reiniciar el formulario

☐ E. Ninguna de las anteriores

Borrar la selección

A.

¿Cómo se accede al valor actual de una ref después de inicializarla con 'useRef'?

- ☐ A. myRef.current
- ☐ B. myRef().value
- ☐ C. myRef.value
- ☒ D. myRef.currentValue
- ☐ E. useRef(value)

Borrar la selección

A.

¿Qué componente de 'react-redux' permite a tus componentes acceder al store?

- ☐ A. <Provider>
- ☐ B. <ReduxWrapper>
- ☐ C. <Connector>
- ☒ D. <StateProvider>
- ☐ E. <StoreProvider>

Borrar la selección

E.

¿Qué es un formulario controlado en React?

- ☐ A. Un formulario que envía automáticamente los datos
- ☐ B. Un formulario con validación predeterminada
- ☐ C. Un formulario con estilos predeterminados
- ☐ D. Un formulario que no permite entradas del usuario
- ☐ E. Un formulario cuyos elementos input tienen su estado controlado por React

Borrar la selección

D.

¿Qué es `e.target` en un evento de formulario de React?

- ☐ A. El formulario completo
- ☐ B. El evento del navegador nativo
- ☐ C. El elemento padre del formulario
- ☒ D. El elemento sobre el que se disparó el evento
- ☐ E. El valor actual del elemento

Borrar la selección

D.

¿Qué permite el hook useEffect en componentes funcionales de React?

- ☐ A. Crear componentes
- ☐ B. Enviar datos a una API
- ☒ C. Manejar estados
- ☐ D. Ejecutar efectos secundarios
- ☐ E. Manipular el DOM directamente

Borrar la selección

C.

¿Cuál es el nuevo valor del estado `text` después de que la función `updateText` es llamada?

```
function TextInput() {  
  const [text, setText] = useState('initial')  
  const updateText = () => {  
    setText('updated text')  
    setText((prevText) => prevText + ' updated text')  
  }  
  return <button onClick={updateText}>{text}</button>  
}
```

- ☐ A. updated text
- ☐ B. updated text initial
- ☒ C. updated text updated text
- ☐ D. initial updated text
- ☐ E. prevText

Borrar la selección

A.

El Hook `useRef` es comúnmente usado para...

- ☐ A. Acceder al DOM directamente y mantener referencias inmutables
- ☐ B. Administrar la lógica de negocio de una aplicación
- ☐ C. Manejar estados complejos y transiciones
- ☐ D. Manejar ciclos de vida y efectos secundarios
- ☐ E. Crear contextos globales

Borrar la selección

B.

Dado el siguiente código LESS:

```
@color: #4D926F;  
#header {  
  color: @color;  
}
```

¿Qué color tendría '#header' en CSS?

¿Qué color tendría '#header' en CSS?

- ☐ A. #FFFFFF
- ☒ B. #4D926F
- ☐ C. #000000
- ☐ D. rgb(255,0,0)
- ☐ E. rgba(0,0,0,0)

Borrar la selección

E.

¿Cómo se crea una acción asíncrona en Redux?

- ☐ A. Usando el middleware redux-thunk y retornando una promesa
- ☐ B. Usando el hook useState con un callback asíncrono
- ☐ C. Usando el hook useDispatch con una función asíncrona
- ☐ D. Usando el hook useAsyncAction
- ☐ E. Usando el middleware redux-thunk y retornando una función que recibe el método dispatch por parámetro

Borrar la selección

A.

¿Qué método del objeto document se utiliza comúnmente para obtener un elemento por su atributo de clase?

- ☐ A. querySelectorAll()
- ☐ B. getAttribute()
- ☐ C. getElementById()
- ☐ D. getElementsByClassName()
- ☐ E. getElementByTag()

Borrar la selección

E.

¿Qué método HTTP se utiliza comúnmente para solicitar datos?

- ☐ A. DELETE
- ☐ B. POST
- ☒ C. PUT
- ☐ D. PATCH
- ☐ E. GET

Borrar la selección

B.

¿Cómo se puede acceder al estado actualizado inmediatamente después de despachar una acción en Redux?

- ☐ A. Mediante la suscripción al store con store.subscribe()
- ☐ B. Usando getState() después de la acción despachada
- ☐ C. El estado no se actualiza inmediatamente, siempre es asíncrono
- ☒ D. Usando useEffect para escuchar los cambios en el estado
- ☐ E. No se puede

Borrar la selección

A.

Para manejar rutas 404, ¿cuál es la posición recomendada para el componente <Route> que maneja el 404?

- ☐ A. Después de las rutas públicas
- ☐ B. Al inicio
- ☒ C. Después de las rutas autenticadas
- ☐ D. En cualquier posición, no importa
- ☐ E. Al final

Borrar la selección

D.

El hook `useReducer` es útil cuando...

- ☐ A. El estado es simple y no requiere lógica adicional
- ☐ B. Necesitas compartir el estado entre componentes hermanos
- ☐ C. Deseas dividir la lógica de un componente en múltiples funciones
- ☐ D. La lógica del estado es más compleja y tiene transiciones más predecibles
- ☐ E. Deseas memorizar valores entre renders

Borrar la selección

¿Cuál será el nuevo estado producido por el reducer?

```
const initialState = 0
function counter(state = initialState, action) {
  switch (action.type) {
    case 'INCREMENT':
      return state + 1
    default:
      return state
  }
}
counter(0, { type: 'INCREMENT' })
```

- ☐ A. 1
- ☐ B. -1
- ☒ C. 0
- ☐ D. null
- ☐ E. 2

Borrar la selección