



## I.E.S. Inca Garcilaso

### DESARROLLO DE APLICACIONES WEB EN ENTORNO CLIENTE

Ciclo formativo: Desarrollo de Aplicaciones Web

Curso: 2º

#### PRÁCTICA 4 TEMA 10: "Ejercicios básicos con el hook useState"

##### Descripción:

Realiza los siguientes ejercicios básicos e independientes entres sí

##### Actividad 1: Contador simple

**Objetivo:** Aprender a usar useState y el evento onClick.

##### **Enunciado:**

Crea un componente Contador que muestre un número inicial en 0 y un botón. Cada vez que se pulse el botón, el número debe aumentar en 1.

##### **Pistas:**

- Usa useState(0)
- Usa un botón con onClick

##### Actividad 2: Cambiar texto con un botón

**Objetivo:** Modificar el estado con un evento.

##### **Enunciado:**

Crea un componente que muestre un texto "Apagado" y un botón "Cambiar estado". Al pulsar el botón, el texto debe cambiar a "Encendido" y volver a "Apagado" si se pulsa otra vez.

##### **Pistas:**

- Usa un estado de tipo string
- Puedes comprobar el valor actual antes de cambiarlo

##### Actividad 3: Input controlado

**Objetivo:** Usar useState con eventos onChange.

##### **Enunciado:**

Crea un componente con un campo de texto (input) y un párrafo. El texto que escriba el usuario en el input debe mostrarse en tiempo real en el párrafo.

##### **Pistas:**

- Usa onChange
- Accede a event.target.value

#### **Actividad 4: Mostrar u ocultar un mensaje**

**Objetivo:** Manejar estados booleanos.

**Enunciado:**

Crea un componente con un botón "Mostrar / Ocultar".

Al pulsarlo, debe mostrarse u ocultarse un mensaje como:

"Este mensaje es secreto 🙊"

**Pistas:**

- Usa un estado booleano (true / false)
- Renderiza el mensaje de forma condicional

#### **Actividad 5: Contador con incremento personalizado**

**Objetivo:** Usar varios estados y eventos.

**Enunciado:**

Crea un contador que tenga:

- Un input donde el usuario indique cuánto quiere sumar
- Un botón "Sumar"
- Un texto que muestre el valor total

Cada vez que se pulse el botón, el contador debe aumentar según el número introducido.

**Pistas:**

- Necesitarás **dos estados**
- Convierte el valor del input a número

#### **Actividad 6: Contador con onMouseEnter y onMouseLeave**

**Objetivo:** Usar eventos de ratón.

**Enunciado:**

Crea un componente que muestre un número inicial en 0.

Cuando el usuario pase el ratón por encima de un div, el número debe aumentar en 1.

Cuando el ratón salga del div, debe disminuir en 1.

**Etiquetas HTML:** div, h2

**Eventos:** onMouseEnter, onMouseLeave

### **Actividad 7: Cambio de color con un select**

**Objetivo:** Usar useState con listas desplegables.

**Enunciado:**

Crea un componente con un select que permita elegir un color (rojo, verde, azul). Debajo, muestra un p cuyo color de texto cambie según la opción seleccionada.

**Etiquetas HTML:** select, option, p

**Eventos:** onChange

### **Actividad 8: Contador de caracteres en un textarea**

**Objetivo:** Manejar texto largo y eventos de escritura.

**Enunciado:**

Crea un componente con un textarea.

Muestra debajo un texto que indique cuántos caracteres ha escrito el usuario.

**Etiquetas HTML:** textarea, span

**Eventos:** onChange

### **Actividad 9: Formulario con onSubmit**

**Objetivo:** Introducir eventos de formulario.

**Enunciado:**

Crea un formulario con:

- Un input para el nombre
- Un botón "Enviar"

Al enviar el formulario, debe mostrarse un mensaje:

"Hola, [nombre]"

Evita que la página se recargue.

**Etiquetas HTML:** form, input, button, p

**Eventos:** onSubmit

### **Actividad 10: Checkbox para aceptar condiciones**

**Objetivo:** Usar estados booleanos con inputs.

**Enunciado:**

Crea un componente con:

- Un input type="checkbox"
- Un botón "Continuar"

El botón solo debe estar habilitado si el checkbox está marcado.

**Etiquetas HTML:** input, button

**Eventos:** onChange

### **Actividad 11: Botón que cambia de color al hacer clic**

**Objetivo:** Modificar estilos inline con useState.

**Enunciado:**

Crea un botón que tenga color azul inicialmente.

Al hacer clic, debe cambiar a color verde y volver a azul si se vuelve a pulsar.

**Etiquetas HTML:** button

**Eventos:** onClick

**Estilos:** backgroundColor

### **Actividad 12: Texto que aumenta y disminuye de tamaño**

**Objetivo:** Cambiar estilos numéricos.

**Enunciado:**

Muestra un texto con tamaño de fuente inicial de 16px.

Añade dos botones: "Aumentar" y "Disminuir", que modifiquen el tamaño del texto.

**Etiquetas HTML:** p, button

**Eventos:** onClick

**Estilos:** fontSize

### **Actividad 13: Resaltar elemento al pasar el ratón**

**Objetivo:** Cambiar estilos con eventos de ratón.

**Enunciado:**

Crea un div con un texto dentro.

Cuando el ratón esté encima, el fondo debe ponerse gris claro; cuando salga, volver al color original.

**Etiquetas HTML:** div

**Eventos:** onMouseEnter, onMouseLeave

**Estilos:** backgroundColor, padding

#### **Actividad 14: Modo claro / modo oscuro**

**Objetivo:** Cambiar varios estilos a la vez.

**Enunciado:**

Crea un componente con un botón "Cambiar tema".

Al pulsarlo, la página debe alternar entre:

- **Modo claro:** fondo blanco y texto negro
- **Modo oscuro:** fondo negro y texto blanco

**Etiquetas HTML:** div, button, h1

**Eventos:** onClick

**Estilos:** backgroundColor, color

#### **Actividad 15: Borde dinámico en un input**

**Objetivo:** Cambiar estilos según el foco.

**Enunciado:**

Crea un input de texto.

Cuando el usuario haga foco (focus), el borde debe ponerse azul.

Cuando pierda el foco (blur), el borde debe volver a gris.

**Etiquetas HTML:** input

**Eventos:** onFocus, onBlur

**Estilos:** border

#### **Actividad 16: Mostrar mensaje según contador**

**Objetivo:** Renderizar contenido según el estado.

**Enunciado:**

Crea un contador con un botón "Sumar".

- Si el contador es menor que 5, muestra: "Valor bajo"
- Si es 5 o mayor, muestra: "Valor alto"

**Pistas:**

- Usa una condición (if o ternario)
  - Renderiza textos distintos
-

### **Actividad 17: Login / Logout simulado**

**Objetivo:** Renderizado condicional de componentes.

**Enunciado:**

Crea un componente con un botón que inicialmente muestre "Iniciar sesión".  
Al pulsarlo:

- Cambia el estado a "logueado"
- Muestra el texto "Bienvenido, usuario"
- Cambia el botón a "Cerrar sesión"

**Etiquetas HTML:** button, p

**Eventos:** onClick

### **Actividad 18: Mostrar detalles de un producto**

**Objetivo:** Mostrar u ocultar bloques de información.

**Enunciado:**

Muestra el nombre de un producto y un botón "Ver detalles".  
Al pulsar el botón, se deben mostrar:

- Descripción
- Precio

Si se vuelve a pulsar, los detalles se ocultan.

**Etiquetas HTML:** div, h3, p, button

**Renderizado:** condicional por bloque

### **Actividad 19: Alerta según texto introducido**

**Objetivo:** Renderizado condicional según la entrada del usuario.

**Enunciado:**

Crea un input de texto.

Si el usuario escribe más de 10 caracteres, muestra un mensaje de alerta:

"Has escrito demasiado texto"

Si no, no se muestra nada.

**Etiquetas HTML:** input, p

**Eventos:** onChange

### **Actividad 20: Lista vacía o con elementos**

**Objetivo:** Renderizado condicional con arrays.

**Enunciado:**

Crea un componente con un botón "Añadir elemento".

- Si la lista está vacía, muestra: "No hay elementos"
- Si hay elementos, muestra una lista (ul) con los elementos añadidos.

**Etiquetas HTML:** button, ul, li, p

**Renderizado:** condicional con arrays

### **Actividad 21: Lista de tareas simple**

**Objetivo:** Manejar arrays en el estado.

**Enunciado:**

Crea un componente con:

- Un input de texto
- Un botón "Añadir tarea"

Cada vez que se pulse el botón, la tarea se añade a una lista que se muestra en pantalla.

**Pistas:**

- Usa el operador spread (...)
- Renderiza la lista con map

### **Actividad 22: Eliminar elementos de una lista**

**Objetivo:** Modificar arrays existentes.

**Enunciado:**

Partiendo del ejercicio anterior, añade un botón "Eliminar" junto a cada tarea. Al pulsarlo, la tarea correspondiente debe eliminarse de la lista.

**Pistas:**

- Usa filter
- Identifica cada elemento por su índice o id

### **Actividad 23: Lista con elementos seleccionables**

**Objetivo:** Trabajar eventos sobre elementos de una lista.

**Enunciado:**

Crea una lista de nombres.

Al hacer clic sobre un nombre:

- Se debe marcar como seleccionado (por ejemplo, cambiando el estilo)
- Solo puede haber **un elemento seleccionado a la vez**

**Pistas:**

- Guarda en el estado el índice o id del elemento seleccionado
- Usa renderizado condicional para el estilo

### **Actividad 24: Contador de elementos**

**Objetivo:** Derivar información a partir del array.

**Enunciado:**

Crea un componente que permita añadir elementos a una lista.

Muestra en todo momento un texto que indique:

“Número de elementos: X”

El valor debe actualizarse automáticamente.

**Pistas:**

- Usa array.length
- No necesitas un estado extra

### **Actividad 25: Lista de elementos con mostrar / ocultar**

**Objetivo:** Combinar arrays y renderizado condicional.

**Enunciado:**

Crea una lista de elementos y un botón "Mostrar / Ocultar lista".

- Si la lista está oculta, no se renderiza el ul
- Si está visible, se muestran los elementos

**Pistas:**

- Combina un estado booleano con un estado array
- Renderiza condicionalmente el bloque de la lista