UNIVERSIDAD NACIONAL Y AUTÓNOMA DE MÉXICO
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

“React.useState”

:: UNIDAD 2: Características del desarrollo de programas para dispositivos móviles

:: ACTIVIDAD COMPLEMENTARIA 2

MATERIA: .: PROGRAMACIÓN DE DISPOSITIVOS MÓVILES :.

CLAVE: 1668

LICENCIATURA EN INFORMÁTICA

PLAN 2012

REALIZO: Emmanuel Alejandro Pérez Hernández

No. 423142118

Grupo: 8691

ASESOR: MARTINEZ FERNANDEZ JUAN MANUEL

sábado, 1 de noviembre de 2025

UNIDAD 2. Actividad Complementaria 2

a) Investiga la función de fetch en Javascript.

La función fetch() en JavaScript es una API moderna para realizar peticiones HTTP de forma asíncrona. Sus principales características son:

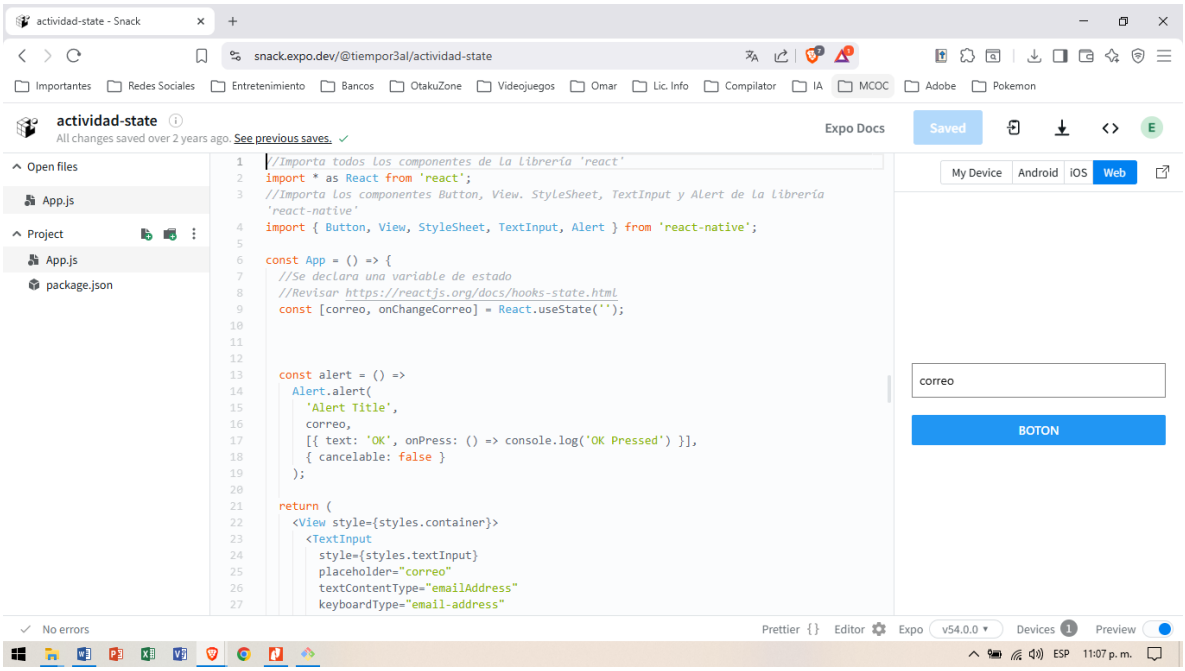
- Retorna una Promise que se resuelve con un objeto Response
- Reemplaza a XMLHttpRequest con una sintaxis más simple y limpia
- Soporta todos los métodos HTTP: GET, POST, PUT, PATCH, DELETE
- Permite enviar y recibir diversos tipos de datos (JSON, texto, Blob, FormDat a)
- Está integrada con características web modernas como CORS y Service Wo rkers

Ejemplo básico:

```
fetch('https://api.ejemplo.com/datos')
  .then(response => response.json())
  .then(datos => console.log(datos))
  .catch(error => console.error(error))
```

Fetch simplifica enormemente el trabajo con APIs y servicios web, haciendo el código más legible y mantenible comparado con las antiguas soluciones basadas en callbacks.

b) Abre la actividad en https://snack.expo.dev/@tiempor3al/actividad-state-.



PROBLEMAS IDENTIFICADOS:

Error en importación de React (Línea 2)

Error en importación de StyleSheet (Línea 4)

Error en React.useState (Línea 9)

Estructura incorrecta del Alert (Líneas 13-19)

c) Explica cómo funciona la app con React.useState('todo').

La aplicación utiliza el Hook useState de React para manejar el estado del campo de correo electrónico. El flujo de funcionamiento es el siguiente:

1. Inicialización:

- Usestate('') crea una variable de estado 'correo' con valor inicial string vacío
- Devuelve un array con [valoractual, funciónactualizadora]

2. Captura de datos:

- Cuando el usuario escribe en el textinput, se ejecuta onchangetext
- Onchangetext llama a onchangecorreo(nuevovalor) actualizando el estado

3. Actualización en tiempo real:

- El componente se re-renderiza con el nuevo valor
- Textinput muestra el valor actual mediante value={correo}

4. Uso del estado:

- Al presionar el botón, mostraralerta() accede al valor actual de 'correo'
- La alerta muestra el contenido actual del estado

Esto permite que la aplicación mantenga sincronizada la interfaz con el estado interno, mostrando siempre el valor más reciente ingresado por el usuario.

d) ¿Cuándo se asigna el valor del texto?

El valor del texto se asigna en dos momentos específicos:

1. Asignación inicial:

- Al montarse el componente por primera vez
- Cuando se ejecuta: react.usestate('')
- El estado 'correo' recibe el valor inicial de string vacío

2. Asignación durante la interacción:

- Cada vez que el usuario escribe en el TextInput
- Con cada carácter que se presiona en el teclado
- La función onChangeText detecta el cambio y ejecuta onChangeCorreo(nuevoValor)
- Esto actualiza el estado 'correo' con el texto completo actualizado

Cada asignación desencadena un re-renderizado del componente, asegurando que la interfaz refleje siempre el valor más reciente del estado.

e) ¿Cuándo se lee el valor del texto?

El valor del texto se asigna en dos momentos específicos:

1. Asignación inicial:

- Al montarse el componente por primera vez
- Cuando se ejecuta: React.usestate('')
- El estado 'correo' recibe el valor inicial de string vacío

2. Asignación durante la interacción:

- Cada vez que el usuario escribe en el textinput
- Con cada carácter que se presiona en el teclado
- La función onchangetext detecta el cambio y ejecuta onchangecorreo(nuevovalor)
- Esto actualiza el estado 'correo' con el texto completo actualizado

Cada asignación desencadena un re-renderizado del componente, asegurando que La interfaz refleje siempre el valor más reciente del estado.

f) Modifica el botón para que llame a una función con fetch y método POST.

Se realizaron las siguientes modificaciones:

1. Creación de nueva función 'enviarDatos':

- Función asíncrona que utiliza fetch para enviar datos
- Configurada con método POST y headers para JSON
- Incluye manejo de errores con try/catch

2. Configuración de la petición http:

- URL: https://jsonplaceholder.typicode.com/posts (API de prueba)
- Método: POST para crear nuevo recurso
- Headers: Content-Type: application/json
- Body: Datos convertidos a JSON con JSON.stringify()

3. Datos enviados:

- email: Valor actual del estado 'correo'
- userId: 1 (valor fijo de ejemplo)
- title: 'Datos desde React Native'

4. Manejo de respuesta:

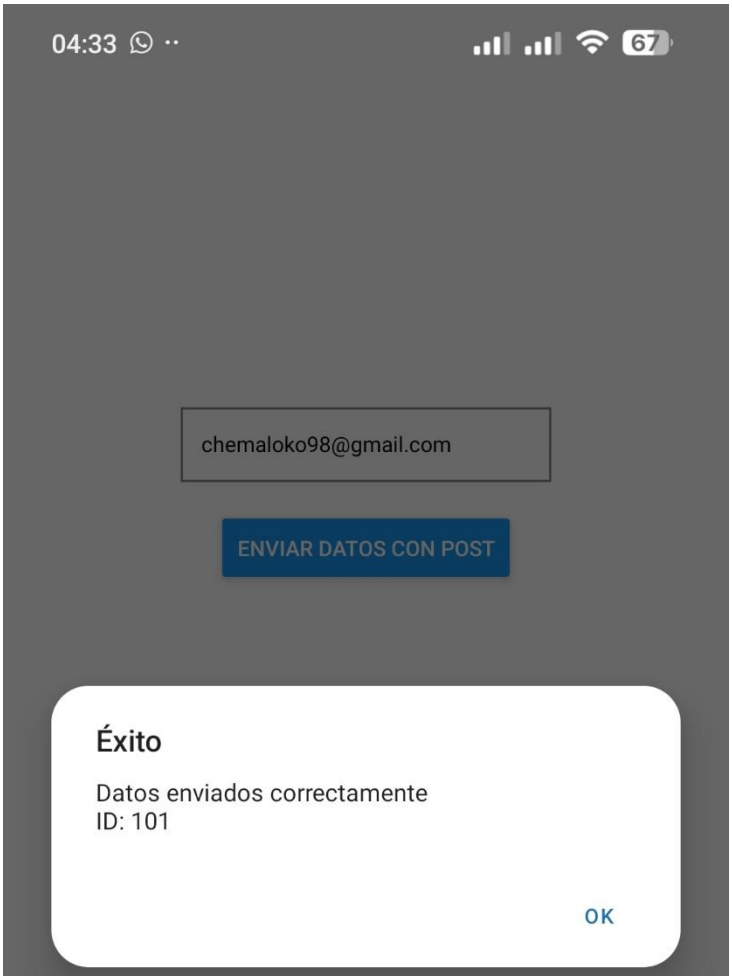
- Verificación del estado HTTP con response.ok
- Conversión de respuesta a JSON
- Alertas de éxito/error según el resultado

5. Actualización del botón:

- Cambio de onPress de mostrarAlerta a enviarDatos
- El botón ahora envía los datos en lugar de solo mostrarlos

Esta modificación permite que la aplicación no solo capture y muestre datos, sino que también los envíe a un servidor externo mediante una API REST.

g) Ejecuta la app en tu dispositivo.



Proceso de ejecución:

- Visualicé la pantalla con un campo de texto vacío
- Escribí un correo electrónico de prueba en el campo (chemaloko98@gmail.com)
- Presioné el botón "Enviar datos con POST"
- Se mostró una alerta confirmando el envío exitoso
- En la consola de Snack pude verificar la petición HTTP

La aplicación funcionó correctamente:

Captura de datos → almacenamiento en estado → envío HTTP → feedback al usuario.

h) Enlace de tu Snack.

https://snack.expo.dev/@getzemax98/actividad-state

I. REFERENCIAS BIBLIOGRÁFICAS

Uso de Fetch - API web | MDN. (2025, March 27). Mozilla.org. https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch

Villca, F. (2024, May 7). *JavaScript Fetch API para principiantes: Explicado con ejemplos de código*. FreeCodeCamp.org. https://www.freecodecamp.org/espanol/news/javascript-fetch-api-para-principiantes/

Using the Fetch API - Web APIs | MDN. (2025, August 20). Mozilla.org. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

Usando el Hook de estado – React. (2021). Reactjs.org. https://es.legacy.reactjs.org/docs/hooks-state.html

useState – React. (2025). React.dev. https://es.react.dev/reference/react/useState

Como funciona el hook useState y como usarlos con Arrays y Objetos. (2020, October 19). Dartiles.dev. https://dartiles.dev/blog/como-funciona-el-hook-usestate-y-como-usarlos-con-arrays-y-objetos