

Explicación de los ejercicios 8 y 9

Nombre : Juan Carlos Jara

Curso: 3 nivel

Materia: Aplicaciones Móviles

EJERCICIO 8

1. javascript

```
import React, { useState } from 'react';
```

- Esta línea importa React y la función `useState` desde la librería 'react'. React se utiliza para construir interfaces de usuario (UI) interactivas en aplicaciones web, y `useState` es un hook que permite añadir estado a componentes funcionales en React.

2. javascript

```
const planets: { [key: string]: number } = { 'mercury': 0.2408467, 'venus': 0.61519726, 'earth': 1.0, 'mars': 1.8808158, 'jupiter': 11.862615, 'saturn': 29.447498, 'uranus': 84.016846, 'neptune': 164.79132 }
```

- Aquí se define un objeto llamado `planets` que mapea el nombre de los planetas a su relación con respecto a la Tierra en términos de años. Por ejemplo, en `planets['mercury']` se guarda la relación de Mercurio con respecto a la Tierra, que es 0.2408467.

3. javascript

```
function age(planet: string, seconds: number): number { let age: number; let earthYearInSeconds = 31557600; age = seconds / (earthYearInSeconds * planets[planet]); return Number(age.toFixed(2)); }
```

- Esta es una función llamada `age` que toma dos parámetros: el nombre del planeta como una cadena y los segundos vividos como un número. Calcula la edad en años en base al planeta y los segundos dados. La fórmula utilizada es dividir los segundos por la duración de un año en segundos multiplicado por la relación del planeta con respecto a la Tierra. El resultado se redondea a 2 decimales y se devuelve como un número.

4. javascript

```
const AgeCalculator: React.FC = () => { const [planet, setPlanet] = useState<string>('earth'); const [seconds, setSeconds] = useState<number>(0); const [ageResult, setAgeResult] = useState<number | null>(null);
```

- Se define un componente de función llamado `AgeCalculator` utilizando el tipo `React.FC` (Functional Component). Se utilizan tres estados (`useState`) en este componente: `planet` para el planeta seleccionado, `seconds` para los segundos vividos ingresados y `ageResult` para el resultado del cálculo de edad.

5. javascript

```
const calculateAge = () => { setAgeResult(age(planet, seconds)); }
```

- `calculateAge` es una función que se llama cuando se hace clic en el botón "Calculate Age". Utiliza la función `age` para calcular la edad en el planeta seleccionado utilizando los segundos ingresados y luego actualiza el estado `ageResult` con el resultado.

6. javascript

```
return ( <div> <h2>Age Calculator</h2> <div> <label htmlFor="planet">Select Planet:</label> <select id="planet" value={planet} onChange={(e) => setPlanet(e.target.value)}> {Object.keys(planets).map(planet => ( <option key={planet} value={planet}>{planet}</option> ))} </select> </div> <div> <label htmlFor="seconds">Enter seconds lived:</label> <input type="number" id="seconds" value={seconds} onChange={(e) => setSeconds(parseInt(e.target.value))} /> </div> <button onClick={calculateAge}>Calculate Age</button> {ageResult !== null && <p>Your age on {planet} is {ageResult} years.</p>} </div> );
```

- Esta es la parte del JSX que compone la interfaz de usuario del componente `AgeCalculator`. Incluye un título, un selector para elegir un planeta, un campo de entrada para los segundos vividos, un botón para calcular la edad y un mensaje que muestra el resultado de la edad en el planeta seleccionado.

Ejercicio 9

```
import React from 'react';
```

- `import React from 'react';`: Esta línea importa la biblioteca React, que necesitamos para definir y trabajar con componentes de React.

```
function DnDCharacter() {
```

- `function DnDCharacter() {`: Esto define una función llamada `DnDCharacter`, que es nuestro componente de React. Es una función de componente funcional de React.

```
const strength = generateAbilityScore(); const dexterity = generateAbilityScore(); const constitution = generateAbilityScore(); const intelligence = generateAbilityScore(); const wisdom = generateAbilityScore();
```

```
const charisma = generateAbilityScore(); const hitpoints = 10 +
getModifierFor(constitution);
```

- Aquí estamos inicializando las propiedades del personaje de D&D. Cada propiedad se calcula llamando a la función correspondiente (`generateAbilityScore` o `getModifierFor`) para generar un valor aleatorio.

```
function rollDice(): number { return Math.floor(Math.random() * (6 - 1) + 1);
}
```

- `rollDice` : Esta función genera un número aleatorio entre 1 y 6, simulando el lanzamiento de un dado de seis caras.

```
function findLowest(numbers: number[]): number { return
numbers.reduce((final, current) => final >= current ? current : final ); }
```

- `findLowest` : Esta función encuentra el número más bajo en un array de números utilizando el método `reduce` .

```
function generateAbilityScore(): number { let dicesRolls: number[] = []; for
(let i = 0; i < 4; i++) { dicesRolls.push(rollDice()); } let lowest =
findLowest(dicesRolls); dicesRolls.splice(dicesRolls.indexOf(lowest), 1);
return dicesRolls.reduce((final, current) => final + current); }
```

- `generateAbilityScore` : Esta función genera una puntuación de habilidad aleatoria sumando el resultado de tirar cuatro dados de seis caras y eliminando el resultado más bajo.

```
function getModifierFor(abilityValue: number): number { return
Math.floor((abilityValue - 10) / 2); }
```

- `getModifierFor` : Esta función calcula el modificador de habilidad para una puntuación de habilidad dada, según la fórmula de D&D.

```
return ( <div> <h2>DnD Character</h2> <p>Strength: {strength}</p>
<p>Dexterity: {dexterity}</p> <p>Constitution: {constitution}</p>
<p>Intelligence: {intelligence}</p> <p>Wisdom: {wisdom}</p> <p>Charisma:
{charisma}</p> <p>Hitpoints: {hitpoints}</p> </div> );
```

- Este es el retorno del componente. Renderiza un conjunto de elementos HTML que muestran las propiedades del personaje de D&D (fuerza, destreza, etc.), así como sus puntos de golpe.

```
export default DnDCharacter;
```

- `export default DnDCharacter;` : Exporta el componente `DnDCharacter` , lo que permite que sea utilizado en otros archivos de la aplicación.

Enlace del video

<https://youtu.be/fjiXcGbWFTc>