

Sesión 3: Componentes y Propiedades Básicas en React Native a través del Proyecto Carnet Estudiantil Digital

Objetivo de la sesión

- Aplicar el concepto de componentes a través del desarrollo del carnet estudiantil digital.
 - Comprender la diferencia entre componentes funcionales y de clase.
 - Personalizar componentes mediante props.
 - Desarrollar el carnet con ambas estructuras de componentes.
-

1. Introducción a los Componentes en React Native

Un **componente** en React Native es una parte reutilizable de la interfaz de usuario. En nuestro proyecto, el carnet de estudiante se representará como un **componente**, el cual contendrá datos como el nombre, identificación y foto del estudiante.

2. Diferencias entre Componentes Funcionales y de Clase

React Native permite crear componentes usando dos enfoques diferentes:

2.1. Componentes Funcionales

- Se definen como funciones de JavaScript.
- Más concisos y fáciles de leer.
- No tienen estado interno antes de los hooks, pero ahora pueden manejar estado con useState.
- Mejor rendimiento en aplicaciones modernas.

Ejemplo aplicado en el proyecto:

```

const CarnetEstudianteFuncional = ({ nombre, identificacion,
carrera, foto }) => {
  return (
    <View style={styles.card}>
      <Image source={{ uri: foto }} style={styles.imagen} />
      <Text style={styles.nombre}>{nombre}</Text>
      <Text>ID: {identificacion}</Text>
      <Text>Carrera: {carrera}</Text>
    </View>
  );
};

```

2.2. Componentes de Clase

- Se definen mediante la sintaxis de clases de ES6.
- Utilizan el método render() para devolver la interfaz de usuario.
- Se accede a las propiedades mediante this.props.
- Antes de los hooks, eran la única forma de manejar estado interno.

Ejemplo aplicado en el proyecto:

```

class CarnetEstudianteClase extends Component {
  render() {
    return (
      <View style={styles.card}>
        <Image source={{ uri: this.props.foto }}
style={styles.imagen} />
        <Text style={styles.nombre}>{this.props.nombre}</Text>
        <Text>ID: {this.props.identificacion}</Text>
        <Text>Carrera: {this.props.carrera}</Text>
      </View>
    );
  }
}

```

```

        </View>
    );
}
}

```

3. Comparación de Componentes Funcionales vs. Componentes de Clase

Característica	Componentes Funcionales	Componentes de Clase
Sintaxis	Más simple y concisa	Verbosa y estructurada
Manejo de props	{props.nombre}	this.props.nombre
Estado	Mediante hooks (useState)	Mediante this.state
Rendimiento	Mejor rendimiento	Puede ser menos eficiente
Ciclo de vida	Con useEffect	Métodos como componentDidMount
Facilidad de lectura	Alta	Media

Conclusión: Para la mayoría de los casos, los componentes funcionales son recomendados por su simplicidad y mejor rendimiento.

4. Construcción del Carnet Estudiantil Digital

4.1. Creación del Componente Funcional

1. Crea un archivo `CarnetEstudianteFuncional.js` en la carpeta `components`.
2. Escribe el siguiente código:

```
import React from 'react';
import { View, Text, Image, StyleSheet } from 'react-native';

const CarnetEstudianteFuncional = ({ nombre, identificacion,
carrera, foto }) => {
  return (
    <View style={styles.card}>
      <Image source={{ uri: foto }} style={styles.imagen} />
      <Text style={styles.nombre}>{nombre}</Text>
      <Text>ID: {identificacion}</Text>
      <Text>Carrera: {carrera}</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  card: {
    padding: 20,
    backgroundColor: '#f0f0f0',
    borderRadius: 10,
    alignItems: 'center',
    marginVertical: 10,
  },
  imagen: {
    width: 100,
    height: 100,
```

```

        borderRadius: 50,
        marginBottom: 10,
    },
    nombre: {
        fontSize: 22,
        fontWeight: 'bold',
    },
  });

export default CarnetEstudianteFuncional;

```

4.2. Creación del Componente de Clase

1. Crea un archivo CarnetEstudianteClase.js en la carpeta components.
2. Escribe el siguiente código:

```

import React, { Component } from 'react';
import { View, Text, Image, StyleSheet } from 'react-native';

class CarnetEstudianteClase extends Component {
  render() {
    return (
      <View style={styles.card}>
        <Image source={{ uri: this.props.foto }}
style={styles.imagen} />
        <Text style={styles.nombre}>{this.props.nombre}</Text>
        <Text>ID: {this.props.identificacion}</Text>
      </View>
    );
  }
}

```

```
        <Text>Carrera: {this.props.carrera}</Text>
      </View>
    );
  }
}
```

```
const styles = StyleSheet.create({
  card: {
    padding: 20,
    backgroundColor: '#f0f0f0',
    borderRadius: 10,
    alignItems: 'center',
    marginVertical: 10,
  },
  imagen: {
    width: 100,
    height: 100,
    borderRadius: 50,
    marginBottom: 10,
  },
  nombre: {
    fontSize: 22,
    fontWeight: 'bold',
  },
});
```

```
export default CarnetEstudianteClase;
```

5. Integración en la Aplicación Principal

Modifica el archivo App.js para utilizar ambos componentes:

```
import React from 'react';
import { View, StyleSheet } from 'react-native';
import CarnetEstudianteFuncional from
'./components/CarnetEstudianteFuncional';
import CarnetEstudianteClase from
'./components/CarnetEstudianteClase';

const App = () => {
  return (
    <View style={styles.container}>
      <CarnetEstudianteFuncional
        nombre="Ana López"
        identificacion="202320001"
        carrera="Ingeniería de Software"
        foto="https://randomuser.me/api/portraits/women/50.jpg"
      />

      <CarnetEstudianteClase
        nombre="Juan Pérez"
        identificacion="202320002"
        carrera="Medicina"
        foto="https://randomuser.me/api/portraits/men/50.jpg"
      />
    </View>
  )
}
```

```
    );  
  };  
  
  const styles = StyleSheet.create({  
    container: {  
      flex: 1,  
      justifyContent: 'center',  
      alignItems: 'center',  
      backgroundColor: '#e3f2fd',  
    },  
  });  
  
  export default App;
```

6. Actividades Prácticas

Cada estudiante debe:

1. Personalizar su carnet con su nombre, foto y carrera.
2. Implementar ambos tipos de componentes.
3. Agregar nuevos campos, como la universidad o correo electrónico.

7. Evaluación de la Sesión

Preguntas clave:

1. ¿Cuál es la diferencia entre un componente funcional y uno de clase?
2. ¿Cómo se pasan y utilizan las props en un componente?
3. ¿Cuál es la estructura básica de un componente funcional?

Entrega de evidencia:

- Captura de pantalla del carnet funcional y de clase.
- Código correctamente estructurado y comentado.