

UNIVERSIDAD DON BOSCO
ESCUELA DE COMPUTACION
DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA



TITULO:

Guia #06

DOCENTE:

Ing. Emerson Francisco Cartagena Candelario

PRESENTADO POR:

Alvarenga Romero, Josías Isaac #AR200778,

Antiguo Cuscatlán, 20 de septiembre de 2023

<https://github.com/isaac-rom/GuiaLabDPS/tree/master/guia06/Procedimiento/citas>

DESARROLLO

1. Cita.js

```
import { Text, StyleSheet, View, TextInput, Button, TouchableHighlight, Alert, ScrollView, TextInput } from 'react-native';
import React, { useState } from 'react';

const Cita = ({item, eliminarPaciente}) => {
  const dialogoEliminar = id => {
    console.log('eliminando ...', id);
    eliminarPaciente(id);
  }

  return(
    <View style={styles.cita}>
      <View>
        <Text style={styles.label}>Paciente:</Text>
        <Text style={styles.label}>{item.paciente}</Text>
      </View>
      <View>
        <Text style={styles.label}>Propetario:</Text>
        <Text style={styles.label}>{item.propietario}</Text>
      </View>
      <View>
        <Text style={styles.label}>Sintomas:</Text>
        <Text style={styles.label}>{item.sintomas}</Text>
      </View>
      <View>
        <TouchableHighlight onPress={() => dialogoEliminar(item.id)} style={styles.btnEliminar}>
          <Text style={styles.textoEliminar}>Eliminar &times;</Text>
        </TouchableHighlight>
      </View>
    </View>
  )
}

const styles = StyleSheet.create({
  cita: {
    backgroundColor: 'FFF',
    borderBottomColor: 'E1E1E1',
    borderStyle: 'solid',
    borderBottomWidth: 1,
    paddingVertical: 20,
    paddingHorizontal: 10
  },
  label: {
    fontWeight: 'bold',
    fontSize: 18,
    marginTop: 20
  },
  texto: {
    fontSize: 18,
  },
  btnEliminar: {
    padding: 10,
    backgroundColor: 'red',
    marginVertical: 10
  },
  textoEliminar: {
    color: '#FFF',
    fontWeight: 'bold',
    textAline: 'center'
  }
});

export default Cita;
```

```
    },
    textoEliminar: {
      color: '#FFF',
      fontWeight: 'bold',
      textAline: 'center'
    }
  })

export default Cita;
```

2. Formulario.js:

```
import React, { useState } from "react";
import { Text, StyleSheet, View, TextInput, Button, TouchableHighlight, Alert, ScrollView, TextInput } from 'react-native';
import DateTimePickerModal from "react-native-modal-datetime-picker";
import shortid from 'react-id-generator';
import color from "../utils/color";

const Formulario = ({ citas, setCitas, guardarMostrarForm, guardarCitasStorage }) => {
  const [paciente, guardarPaciente] = useState('');
  const [propietario, guardarPropietario] = useState('');
  const [telefono, guardarTelefono] = useState('');
  const [fecha, guardarFecha] = useState('');
  const [hora, guardarHora] = useState('');
  const [sintomas, guardarSintomas] = useState('');

  const [isDatePickerVisible, setDatePickerVisibility] = useState(false);
  const [isTimePickerVisible, setTimePickerVisibility] = useState(false);

  const showDatePicker = () => {
    setDatePickerVisibility(true);
  };

  const hideDatePicker = () => {
    setDatePickerVisibility(false);
  };

  const confirmarFecha = date => {
    const opciones = { year: 'numeric', month: 'long', day: '2-digit' };
    guardarFecha(date.toLocaleDateString('es-SV', opciones));
    hideDatePicker();
  };

  const showTimePicker = () => {
    setTimePickerVisibility(true);
  };

  const hideTimePicker = () => {
    setTimePickerVisibility(false);
  };

  const confirmarHora = hora => {
    const opciones = { hour: 'numeric', minute: '2-digit', hour12: false };
    guardarHora(hora.toLocaleString('es-SV', opciones));
    hideTimePicker();
  };

  const crearNuevaCita = () => {
    if (paciente.trim() === '' ||
        propietario.trim() === '' ||
        telefono.trim() === '' ||
        fecha.trim() === '' ||
        hora.trim() === '' ||
        sintomas.trim() === '') {
      mostrarAlerta();
      return;
    }

    const cita = { paciente, propietario, telefono, fecha, hora, sintomas };
    cita.id = shortid();
  };
}
```

```
const citasNuevo = [...citas, cita];
setCitas(citasNuevo);

guardarCitasStorage(JSON.stringify(citasNuevo));

guardarMostrarForm(false);

guardarSintomas('');
guardarPropietario('');
guardarPaciente('');
guardarHora('');
guardarFecha('');
guardarTelefono('');
}

const mostrarAlerta = () => {
  Alert.alert(
    'Error',
    'Todos los campos son obligatorios',
    [{
      text: 'OK'
    }]
  )
}
```

```

return(
  <>
    <ScrollView styles={styles.formulario}>
      <View>
        <Text style={styles.label}>Paciente:</Text>
        <TextInput style={styles.input}
          onChangeText={texto => guardarPaciente(texto)} />
      </View>

      <View>
        <Text style={styles.label}>Dueño:</Text>
        <TextInput style={styles.input}
          onChangeText={texto => guardarPropietario(texto)} />
      </View>

      <View>
        <Text style={styles.label}>Teléfono Contacto:</Text>
        <TextInput style={styles.input}
          onChangeText={texto => guardarTelefono(texto)}
          keyboardType="numeric" />
      </View>

      <View>
        <Text style={styles.label}>Fecha:</Text>
        <Button title="Seleccionar Fecha"
          onPress={showDatePicker}/>
        <DateTimePickerModal
          isVisible={isDatePickerVisible}
          mode="date"
          onConfirm={confirmarFecha}
          onCancel={hideDatePicker}
          local='es-SV'

```

```

          headerTextIOS="Elige la fecha"
          cancelTextIOS="Cancelar"
          confirmTextIOS="Confirmar"
        />
        <Text>{fecha}</Text>
      </View>

      <View>
        <Text style={styles.label}>Hora:</Text>
        <Button title="Seleccionar Hora"
          onPress={showTimePicker}/>
        <DateTimePickerModal
          isVisible={isTimePickerVisible}
          mode="time"
          onConfirm={confirmarHora}
          onCancel={hideTimePicker}
          local='es-SV'
          headerTextIOS="Elige la hora"
          cancelTextIOS="Cancelar"
          confirmTextIOS="Confirmar"
        />
        <Text>{hora}</Text>
      </View>

      <View>
        <Text style={styles.label}>Síntomas:</Text>
        <TextInput style={styles.input}
          onChangeText={texto => guardarSíntomas(texto)} />
      </View>

```

```

        <View>
          <Text style={styles.label}>Sintomas:</Text>
          <TextInput style={styles.input}
            onChangeText={texto => guardarSintomas(texto)}/>
        </View>

        <View>
          <TouchableHighlight onPress={() => crearNuevaCita()} style={styles.btnSubmit}>
            <Text style={styles.textoSubmit}>Crear Nueva Cita</Text>
          </TouchableHighlight>
        </View>
      </ScrollView>
    </>
  );
}

const styles = StyleSheet.create({
  formulario: {
    backgroundColor: '#FFF',
    paddingHorizontal: 20,
    paddingVertical: 10,
    flex: 1
  },
  label: {
    fontWeight: 'bold',
    fontSize: 18,
    marginTop: 20
  },

```

```

    input: {
      marginTop: 10,
      height: 50,
      borderColor: '#E1E1E1',
      borderWidth: 1,
      borderStyle: 'solid'
    },
    btnSubmit: {
      padding: 10,
      backgroundColor: color.BUTTON_COLOR,
      marginVertical: 10
    },
    textoSubmit: {
      color: '#FFF',
      fontWeight: 'bold',
      textAlign: 'center'
    }
  })

export default Formulario;

```

3. App.js

```
import { StatusBar } from 'expo-status-bar';
import React, { useState, useEffect } from 'react';
import { StyleSheet, Text, View, FlatList, TouchableHighlight, TouchableWithoutFeedback, Keyboard, Platform } from 'react-native';
import Cita from './src/components/Cita';
import Formulario from './src/components/Formulario';
import AsyncStorage from '@react-native-async-storage/async-storage';
import color from './src/utlis/color';

const App = () => {

  const [citas, setCitas] = useState([]);
  const [mostrarForm, guardarMostrarForm] = useState(false);

  useEffect(() => {
    const obtenerCitasStorage = async () => {
      try {
        const citasStorage = await AsyncStorage.getItem('citas');
        if (citasStorage) {
          setCitas(JSON.parse(citasStorage));
        }
      } catch (error) {
        console.log(error);
      }
    }
    obtenerCitasStorage();
  }, []);

  const eliminarPaciente = id => {
    const citasFiltradas = citas.filter(cita => cita.id !== id);
    setCitas(citasFiltradas);
    guardarCitasStorage(JSON.stringify(citasFiltradas));
  }
}
```

```
const mostrarFormulario = () => {
  guardarMostrarForm(!mostrarForm);
}

const cerrarTeclado = () => {
  Keyboard.dismiss();
}

const guardarCitasStorage = async (citasJSON) => {
  try {
    await AsyncStorage.setItem('citas', citasJSON);
  } catch (error) {
    console.log(error);
  }
}

return (
  <TouchableWithoutFeedback onPress={() => cerrarTeclado()}>
    <View style={styles.contenedor}>
      <Text style={styles.titulo}>Administrador de Citas</Text>
      <View>
        <TouchableHighlight onPress={() => mostrarFormulario()}
          style={styles.btnMostrarForm}>
          <Text style={styles.textoMostrarForm}>
            {mostrarForm ? 'Cancelar Crear Cita' : 'Crear Nueva Cita'}
          </Text>
        </TouchableHighlight>
      </View>
    </TouchableWithoutFeedback>
  )
```

```

<View style={styles.contenedor}>
  {mostrarForm ? (
    <>
      <Text style={styles.titulo}>Crear Nueva Cita</Text>
      <Formulario
        citas={citas}
        setCitas={setCitas}
        guardarMostrarForm={guardarMostrarForm}
        guardarCitasStorage={guardarCitasStorage}
      ></Formulario>
    </>
  ) : (
    <>
      <Text style={styles.titulo} >
        {citas.length > 0 ? 'Administra tus citas' : 'No hay citas, agregue una'}
      </Text>
      <FlatList
        style={styles.listado}
        data={citas}
        renderItem={({ item }) => <Cita item={item} eliminarPaciente={eliminarPaciente} />}
      ></FlatList>
    </>
  )}
</View>
</View>
</TouchableWithoutFeedback>
);

```

```

const styles = StyleSheet.create({
  contenedor: {
    backgroundColor: color.PRIMARY_COLOR,
    flex: 1
  },
  titulo: {
    color: '#fff',
    marginTop: Platform.OS === 'ios' ? 40 : 20,
    marginBottom: 20,
    fontSize: 24,
    fontWeight: 'bold',
    textAlign: 'center'
  },
  contenido: {
    flex: 1,
    marginHorizontal: '2.5%'
  },
  listado: {
    flex: 1
  },
  btnMostrarForm: {
    padding: 10,
    backgroundColor: color.BUTTON_COLOR,
    marginVertical: 10
  },
  textoMostrarForm: {
    color: '#fff',
    fontWeight: 'bold',
    textAlign: 'center'
  }
})

```

```
export default App;
```

4. Resultados:

Administrador de Citas

Crear Nueva Cita

No hay citas, agregue una

Administrador de Citas

Cancelar Crear Cita

Crear Nueva Cita

Paciente:

Josias

Dueño:

Juan

Teléfono Contacto:

78089291

Fecha:

SELECCIONAR FECHA

Hora:

SELECCIONAR HORA

Sintomas:

Administrador de Citas

Crear Nueva Cita

Administra tus citas

Paciente:

Josias

Propetario:

Juan

Sintomas:

Fiebre

Eliminar ×

Paciente:

Josias

Propetario:

Juan

Sintomas:

Fiebre

Eliminar ×

Paciente:

Jose

Propetario:

Miguel

Sintomas:

Vomito

Eliminar ×