

# UNIVERSIDAD DON BOSCO



## FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN

Diseño y Programación de Software Multiplataforma  
(DPS) G05L

Desafío 2:  
Entrega de desafío práctico

Enlace del video:

<https://drive.google.com/file/d/1UVI4EyfuACwBa9r2BllaZlpTdVSCkSio/view?usp=sharing>

Enlace de GitHub:

[https://github.com/Josepo616/Desafio02\\_DPS\\_Lab/tree/master](https://github.com/Josepo616/Desafio02_DPS_Lab/tree/master)

Nombre:

Diego Enrique Canjura Estrada CE221950  
José Rolando Álvarez Mejía AM232553

**App.js:** Muestra el componente de React Native para crear un drawer navigator. Se importan los módulos necesarios para React, React Native y ReactNavigation, maneja funciones para el inicio y cierre de sesión. Si el usuario está autenticado, muestra el menú y el historial.

```
import React, { useState } from 'react';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { NavigationContainer } from '@react-navigation/native';
import { Button, StyleSheet, View } from 'react-native'; // Importar
StyleSheet y View

import LoginScreen from './src/screens/login';
import HomeScreen from './src/screens/HomeScreen';
import DetailsScreen from './src/screens/DetailsScreen';

const Drawer = createDrawerNavigator();

export default function App() {
  const [loggedIn, setLoggedIn] = useState(false);

  const handleLogin = () => {
    setLoggedIn(true);
  };

  const handleLogout = () => {
    setLoggedIn(false);
  };

  return (
    <NavigationContainer>
      <View style={styles.container}>
        {loggedIn ? (
          <Drawer.Navigator initialRouteName="Home">
            <Drawer.Screen name="Home" component={HomeScreen} />
            <Drawer.Screen name="Details" component={DetailsScreen} />
          </Drawer.Navigator>
        ) : (
          <LoginScreen onLogin={handleLogin} />
        )}
        {loggedIn && (
          <View style={styles.logoutButtonContainer}>
            <Button title="Cerrar Sesión" onPress={handleLogout} />
          </View>
        )}
      </View>
    </NavigationContainer>
  );
}
```

```

    })
  </View>
</NavigationContainer>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  logoutButtonContainer: {
    position: 'absolute',
    top: 20,
    right: 20,
  },
});

```

**Login.js:** Muestra el formulario de inicio de sesión y así mismo maneja su lógica. Muestra una alerta si las credenciales no son correctas.

```

import React, { useState } from 'react';
import { View, Text, TextInput, TouchableOpacity, StyleSheet } from
'react-native';

const LoginScreen = ({ onLogin }) => { // Recibe la función onLogin
como prop
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [errorMessage, setErrorMessage] = useState('');

  const handleLogin = () => {
    if (username === 'admin' && password === 'admin') {
      // Llama a la función onLogin para actualizar el estado loggedIn
y navegar a HomeScreen
      onLogin();
    } else {
      setErrorMessage('Usuario o contraseña incorrectos');
    }
  };

  return (

```

```

<View style={styles.container}>
  <Text style={styles.title}>Inicio de Sesión</Text>
  <TextInput
    style={styles.input}
    placeholder="Usuario"
    value={username}
    onChangeText={setUsername}
  />
  <TextInput
    style={styles.input}
    placeholder="Contraseña"
    secureTextEntry={true}
    value={password}
    onChangeText={setPassword}
  />
  <TouchableOpacity style={styles.button} onPress={handleLogin}>
    <Text style={styles.buttonText}>Iniciar Sesión</Text>
  </TouchableOpacity>
  {errorMessage ? <Text
style={styles.errorMessage}>{errorMessage}</Text> : null}
</View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#FFFFFF',
  },
  title: {
    fontSize: 24,
    marginBottom: 20,
    color: '#006847',
  },
  input: {
    width: '80%',
    height: 40,
    borderWidth: 1,
    borderColor: '#C1272D',
    borderRadius: 5,
    marginBottom: 10,
  },

```

```

        paddingHorizontal: 10,
      },
      button: {
        width: '80%',
        backgroundColor: '#CE1126',
        borderRadius: 5,
        paddingVertical: 10,
        alignItems: 'center',
        marginTop: 10,
      },
      buttonText: {
        color: '#FFFFFF',
        fontSize: 16,
      },
      errorMessage: {
        color: '#FF0000',
        marginTop: 10,
      },
    },
  ));

export default LoginScreen;

```

**HomeScreen.js:** Muestra todos los productos en los cards, dividiéndolos en “Alimentos” y “Bebidas”, mostrando su nombre, precio y fotografía. Cada producto se puede seleccionar y se manda al carrito, donde se muestra el total y se confirma la compra.

```

import React, { useState } from 'react';
import { View, Text, StyleSheet, TouchableOpacity, ScrollView, Image }
from 'react-native';
import AsyncStorage from '@react-native-async-storage/async-storage';

const mexicanFood = [
  { id: 1, name: 'Tacos al Pastor', price: 10, type: 'food', imageUrl:
'https://cdn7.kiwilimon.com/recetaimagen/13996/640x640/6330.jpg.webp' },
  { id: 2, name: 'Enchiladas Verdes', price: 12, type: 'food',
imageUrl: 'https://cdn7.kiwilimon.com/recetaimagen/36938/50874.jpg' },
  { id: 3, name: 'Chiles Rellenos', price: 14, type: 'food', imageUrl:
'https://assets.unileversolutions.com/recipes-v2/243044.jpg' },
  { id: 4, name: 'Tostadas de Ceviche', price: 15, type: 'food',
imageUrl:
'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhTOOPpCMiO4

```

```

e74sV0tQP0Zj2KyU17b-D6KAlrYAWaraRLkMgLmXG5Vl8ROx1b01sPdKag3ydw6gOPNmfoC
ffM4GA184xdonpaoXwXpjApJwshZgrOw81qrMmD5ZunZIRBL2F5tAa5rv0A/s1600/DSC09
151.JPG' },
  { id: 5, name: 'Mole Poblano', price: 16, type: 'food', imageUrl:
'https://cdn.recetasderechupete.com/wp-content/uploads/2023/03/Mole-pob
lano-con-arroz.jpg' },
  { id: 6, name: 'Torta de Res', price: 6, type: 'food', imageUrl:
'https://cocina-casera.com/mx/wp-content/uploads/2017/12/torta-2.jpg'
},
  { id: 7, name: 'Horchata', price: 5, type: 'drink', imageUrl:
'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRjWX492QHH9hbWDO
Q9-rJQkvfMQJunHyyz2Smv7aGzyg&s' },
  { id: 8, name: 'Jarritos', price: 4, type: 'drink', imageUrl:
'https://www.despensamexicana.es/large/Jarritos-sabor-mandarina-i548.jp
g' },
  { id: 9, name: 'Agua de Jamaica', price: 3, type: 'drink', imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/5/55/A_glass_of_h
ibiscus_tea_01.jpg/1200px-A_glass_of_hibiscus_tea_01.jpg' },
  { id: 10, name: 'Champurrado', price: 3, type: 'drink', imageUrl:
'https://images.aws.nestle.recipes/resized/2020_05_26T17_43_26_mrs_Imag
eRecipes_1438351rg_1080_850.jpg' },
  { id: 11, name: 'Michelada', price: 8, type: 'drink', imageUrl:
'https://cdn.recetasderechupete.com/wp-content/uploads/2023/05/Michelad
a-mexicana-clasica.jpg' },
  { id: 12, name: 'Chamoyada de Tamarindo', price: 8, type: 'drink',
imageUrl: 'https://cdn7.kiwilimon.com/recetaimagen/31855/36603.jpg' }
];

export default function HomeScreen({ navigation }) {
  const [order, setOrder] = useState([]);
  const [totalPrice, setTotalPrice] = useState(0);
  const [orderPlaced, setOrderPlaced] = useState(false);

  const addToOrder = (item) => {
    setOrder([...order, item]);
    setTotalPrice(prevTotalPrice => prevTotalPrice + item.price);
  };

  const placeOrder = async () => {
    setOrderPlaced(true);
    console.log("LOG Datos del pedido:", order);
    const existingOrders = await AsyncStorage.getItem('orders');

```

```

    const newOrder = { order: [...order], orderTime: new
Date().toLocaleString() };
    if (existingOrders) {
        const parsedOrders = JSON.parse(existingOrders);
        parsedOrders.push(newOrder);
        await AsyncStorage.setItem('orders',
JSON.stringify(parsedOrders));
    } else {
        await AsyncStorage.setItem('orders', JSON.stringify([newOrder]));
    }
    navigation.navigate('Details'); // Navega a la pantalla de detalles
    setOrder([]); // Reinicia el estado del pedido
    setTotalPrice(0); // Reinicia el estado del precio total
    setOrderPlaced(false); // Reinicia el estado de pedido realizado
};

// Función para dividir el array de ítems en filas de tres
const chunkArray = (arr, chunkSize) => {
    return Array.from({ length: Math.ceil(arr.length / chunkSize) },
(_, index) => {
        const start = index * chunkSize;
        return arr.slice(start, start + chunkSize);
    });
};

// Dividir los ítems en filas de tres
const foodRows = chunkArray(mexicanFood.filter(item => item.type ===
'food'), 3);
const drinkRows = chunkArray(mexicanFood.filter(item => item.type ===
'drink'), 3);

return (
    <ScrollView contentContainerStyle={styles.container}>
        <View style={styles.header}>
            <Text style={styles.bannerTitle}>Bienvenido a Nuestro
Restaurante Mexicano</Text>
            <Text style={styles.bannerText}>Selecciona los alimentos y
bebidas que deseas ordenar:</Text>
        </View>

        <View style={styles.menu}>
            <Text style={styles.menuTitle}>Alimentos:</Text>
            {foodRows.map((row, rowIndex) => (

```

```

        <View key={rowIndex} style={styles.row}>
            {row.map(item => (
                <TouchableOpacity key={item.id} style={styles.menuItem}
onPress={() => addToOrder(item)}>
                    <View style={styles.card}>
                        <Image source={{ uri: item.imageUrl }}
style={styles.cardImage} />
                        <Text style={styles.cardText}>{item.name} -
${item.price}</Text>
                    </View>
                </TouchableOpacity>
            ))}
        </View>
    ))}
    <Text style={styles.menuTitle}>Bebidas:</Text>
    {drinkRows.map((row, rowIndex) => (
        <View key={rowIndex} style={styles.row}>
            {row.map(item => (
                <TouchableOpacity key={item.id} style={styles.menuItem}
onPress={() => addToOrder(item)}>
                    <View style={styles.card}>
                        <Image source={{ uri: item.imageUrl }}
style={styles.cardImage} />
                        <Text style={styles.cardText}>{item.name} -
${item.price}</Text>
                    </View>
                </TouchableOpacity>
            ))}
        </View>
    ))}
</View>

<View style={styles.orderSummary}>
    <Text style={styles.orderSummaryTitle}>Resumen de Orden:</Text>
    {order.map(item => (
        <View key={item.id} style={styles.orderItem}>
            <Text>{item.name} - ${item.price}</Text>
        </View>
    ))}
    <Text>Total: ${totalPrice}</Text>
    {order.length > 0 && (
        <TouchableOpacity style={styles.button} onPress={placeOrder}>
            <Text style={styles.buttonText}>Realizar Pedido</Text>

```



```
        </TouchableOpacity>
      )}
    </View>

    {orderPlaced && <Text style={styles.orderPlacedMessage}>Pedido
Realizado</Text>}}
  </ScrollView>
);
}

const styles = StyleSheet.create({
  container: {
    flexGrow: 1,
    padding: 20,
    backgroundColor: 'rgba(0, 128, 0, 1)',
  },
  header: {
    marginBottom: 20,
    backgroundColor: 'rgba(204, 0, 0, 1)',
    padding: 10,
    borderRadius: 5,
  },
  bannerTitle: {
    fontSize: 24,
    fontWeight: 'bold',
    color: '#fff',
  },
  bannerText: {
    fontSize: 16,
    color: '#fff',
  },
  menu: {
    marginBottom: 20,
  },
  menuTitle: {
    fontSize: 20,
    fontWeight: 'bold',
    marginBottom: 10,
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    marginBottom: 20,
```

```
},
menuItem: {
  width: '30%',
},
card: {
  borderWidth: 1,
  borderColor: 'black',
  borderRadius: 5,
  padding: 10,
  alignItems: 'center',
  backgroundColor: '#fff',
},
cardImage: {
  width: 100,
  height: 100,
  marginBottom: 10,
  borderRadius: 5,
},
cardText: {
  textAlign: 'center',
  fontFamily: 'Arial',
  fontWeight: 'bold',
},
orderSummary: {
  borderTopWidth: 1,
  borderTopColor: '#ccc',
  paddingTop: 20,
},
orderSummaryTitle: {
  fontSize: 20,
  fontWeight: 'bold',
  marginBottom: 10,
},
orderItem: {
  marginBottom: 10,
  flexDirection: 'row',
  justifyContent: 'space-between',
  alignItems: 'center',
},
button: {
  backgroundColor: 'rgba(255, 255, 255, 1)',
  paddingVertical: 10,
  paddingHorizontal: 20,
```

```

    borderRadius: 5,
    alignItems: 'center',
  },
  buttonText: {
    color: '#000',
    fontSize: 16,
    fontWeight: 'bold',
  },
  orderPlacedMessage: {
    fontSize: 18,
    fontWeight: 'bold',
    color: 'green',
    textAlign: 'center',
    marginTop: 10,
  },
});

```

**DetailsScreen.js:** Se muestra el historial de todas las compras que se han hecho, mostrando los productos que se pidieron, con su precio y el día y la hora en que se efectuó la compra.

```

import React, { useState, useEffect } from 'react';
import { View, Text, StyleSheet, FlatList, TouchableOpacity,
ActivityIndicator } from 'react-native';
import AsyncStorage from '@react-native-async-storage/async-storage';

const DetailsScreen = ({ route, navigation }) => {
  const [orders, setOrders] = useState([]);
  const [isLoading, setIsLoading] = useState(true);

  useEffect(() => {
    const loadOrderData = async () => {
      console.log("Loading order data...");
      const savedOrders = await AsyncStorage.getItem('orders');
      console.log("Saved orders:", savedOrders);
      if (savedOrders) {
        setOrders(JSON.parse(savedOrders));
      }
      setIsLoading(false); // Una vez cargados los datos, establecer
isLoading en false
      console.log("isLoading:", isLoading); // Verificar si isLoading
cambia a false
    };
  });

```

```

    };

    loadOrderData();
  }, []);

  const handleNewOrder = () => {
    navigation.navigate('Home'); // Redirige al usuario de vuelta a la
    pantalla de inicio
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Detalles del Pedido</Text>
      {isLoading ? (
        <ActivityIndicator size="large" color="#007bff" />
      ) : (
        <FlatList
          data={orders}
          renderItem={({ item }) => (
            <View style={styles.orderItem}>
              <Text style={styles.orderTime}>Hora del Pedido:
{item.orderTime}</Text>
              <FlatList
                data={item.order}
                renderItem={({ item }) => (
                  <View style={styles.subOrderItem}>
                    <Text style={styles.itemName}>{item.name}</Text>
                    <Text style={styles.itemPrice}>Precio:
${item.price}</Text>
                  </View>
                )}
                keyExtractor={(item) => item.id.toString()}
              />
            </View>
          )}
          keyExtractor={(item, index) => index.toString()}
        />
      )}
      <TouchableOpacity style={styles.button} onPress={handleNewOrder}>
        <Text style={styles.buttonText}>Realizar otro Pedido</Text>
      </TouchableOpacity>
    </View>
  );
}

```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: '#fff',
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
  },
  orderItem: {
    marginBottom: 20,
    borderWidth: 1,
    borderColor: '#ccc',
    borderRadius: 5,
    padding: 10,
  },
  orderTime: {
    fontSize: 18,
    fontWeight: 'bold',
    marginBottom: 10,
    textAlign: 'center',
  },
  subOrderItem: {
    marginBottom: 10,
    alignItems: 'center', // Centrar los elementos dentro del recuadro
  },
  itemName: {
    fontSize: 16,
    fontWeight: 'bold',
  },
  itemPrice: {
    fontSize: 14,
  },
  button: {
    backgroundColor: '#007bff',
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 5,
    marginTop: 20,
  },
});
```

```

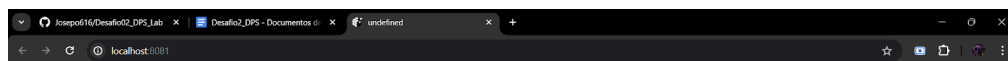
        alignSelf: 'center',
      },
      buttonText: {
        color: '#fff',
        fontSize: 16,
        fontWeight: 'bold',
      },
    },
  ));

export default DetailsScreen;

```

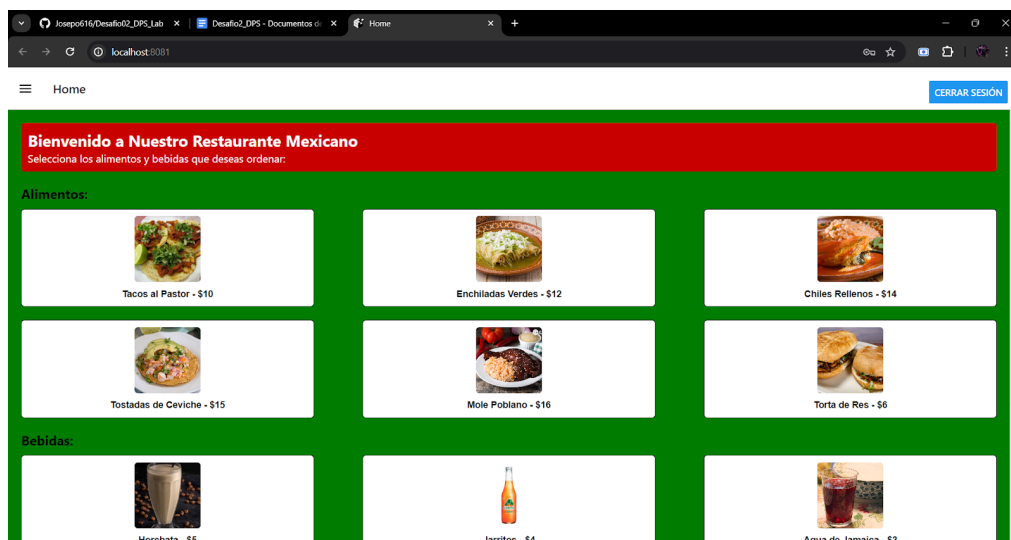
Capturas de pantalla.

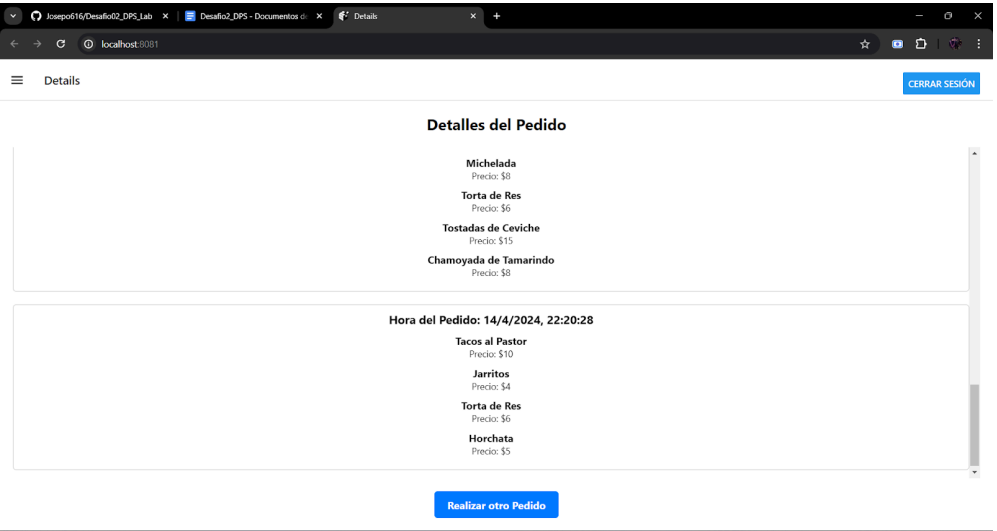
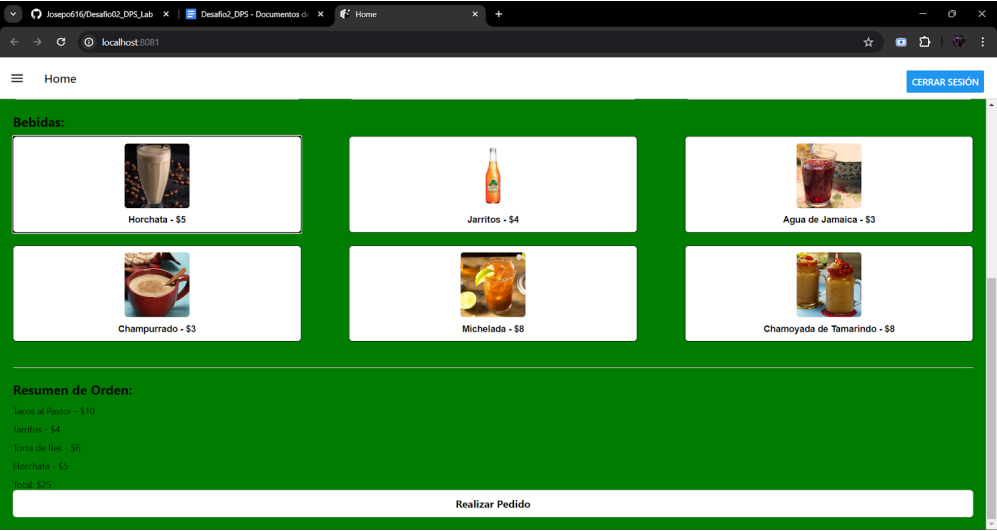
En el servidor web:



Inicio de Sesión

Iniciar Sesión





En el dispositivo móvil:

