

	<p style="text-align: center;">UNIVERSIDAD DON BOSCO</p> <p style="text-align: center;">COORDINACION DE COMPUTACION</p>
<p style="text-align: center;">Ciclo I</p>	<p style="text-align: center;">DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA</p> <p style="text-align: center;">Guía de Laboratorio No. 9</p> <p style="text-align: center;">App Móvil usando API para CRUD</p>

I. RESULTADOS DE APRENDIZAJE

- Que el estudiante utilice firebase real time database como base de datos relacional
- Que conozca los métodos de acceso a nodos de firebase
- Que utilice librerías adicionales

II. INTRODUCCIÓN








En esta guía crearemos una RESTful API empleando PHP y MySQL. Sabemos que es, para qué sirve y la estructura del mismo. Ahora pondremos en práctica nuestro conocimiento de programación orientada a objetos (OOP) con PHP.

Antes de empezar debo aclarar algo, para que un API sea RESTful debe permitir acceso y consumo del servicio vía HTTP. Es decir, desde cualquier navegador o herramienta que permita métodos HTTP como GET, POST, PUT y DELETE .

III. DESARROLLO

Parte I

1. En el caso de esta aplicación se creo un sitio de hospedaje gratuito, llamado <https://000webhost.com/> que permite almacenar datos en base de datos MySQL y realizar servicios web en PHP sin costo alguno.
2. deberá crea la base de datos mysql con la siguiente estructura, dentro de nuestro hosting:
 - Imagen de la estructura de la tabla usuarios:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	id 	int(11)			No	Ninguna		AUTO_INCREMENT	 Cambiar  Eliminar ▼ Más
<input type="checkbox"/> 2	usuario	varchar(50)	utf8_unicode_ci		No	Ninguna			 Cambiar  Eliminar ▼ Más
<input type="checkbox"/> 3	contrasena	varchar(20)	utf8_unicode_ci		No	Ninguna			 Cambiar  Eliminar ▼ Más

- Imagen de la estructura de la tabla productos:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/> 2	nombre	varchar(100)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	descripcion	varchar(500)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 4	preciodecosto	decimal(10,0)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 5	preciodeventa	decimal(10,0)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 6	cantidad	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 7	fotografia	varchar(500)	utf8_unicode_ci		No	Ninguna			Cambiar Eliminar Más

3. debera crear los siguientes archivos

- para autenticarse necesita crear apiusuario.php

```
<?php
Header('Access-Control-Allow-Origin: *');
if($_GET)
{ $comando=$_GET['comando'];
  $servername = "localhost";
  $username = "id16222502_root";
  $password = "M1c0ntr@s3ñ@";
  $dbname = "id16222502_misdatos";
  // Crear la conexión
  $conn = new mysqli($servername, $username, $password,
  $dbname);
  // Check connection
  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
  }
  if($comando=='autenticar')
  { $usuario=$_GET["usuario"];
    $contrasena=$_GET["contrasena"];
    $sql = "Select * from usuarios where usuario='$usuario' and
    contrasena='$contrasena'";
    $result=$conn->query($sql);
    if($result->num_rows>0){
      echo '{"encontrado":"si"}';
    }else{
      echo '{"encontrado":"no"}';
    }
  }
  $conn->close();
}
```

- crear anuestra API REST del archivo api.php

```
<?php
Header('Access-Control-Allow-Origin: *');
```

```

if($_GET) {
    $comando=$_GET['comando'];
    $servername = "localhost";
    $username = "id16222502_root";
    $password = "M1c0ntr@s3ñ@";
    $dbname = "id16222502_misdatos";
    // Crear conexión
    $conn = new mysqli($servername, $username, $password,
$dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    if($comando=='agregar') {
        $nombre=$_GET["nombre"];
        $descripcion=$_GET["descripcion"];
        $preciodecosto=$_GET["preciodecosto"];
        $preciodeventa=$_GET["preciodeventa"];
        $cantidad=$_GET["cantidad"];
        $fotografia=$_GET["fotografia"];
        $sql = "INSERT INTO productos
(nombre,descripcion,preciodecosto,preciodeventa,cantidad,fotografia)
VALUES ('$nombre', '$descripcion',
$preciodecosto,$preciodeventa,$cantidad,$fotografia)";
        if ($conn->query($sql) === TRUE) {
            echo '{"mensaje":"Nuevo registro añadido"}';
        }
        else {
            echo '{"error: "' . $sql . '" . $conn->error.'}';
        }
    }
    if($comando=='editar') {
        $nombre=$_GET["nombre"];
        $descripcion=$_GET["descripcion"];
        $preciodecosto=$_GET["preciodecosto"];
        $preciodeventa=$_GET["preciodeventa"];
        $cantidad=$_GET["cantidad"];
        $fotografia=$_GET["fotografia"];
        $id=$_GET["id"];
        $sql = "UPDATE productos SET nombre='$nombre',
descripcion='$descripcion',preciodecosto=$preciodecosto,
preciodeventa=$preciodeventa, cantidad=$cantidad, fotografia='$fotografia'
WHERE id=$id";
        if ($conn->query($sql) === TRUE) {
            echo '{"mensaje":"Registro actualizado"}';
        }
        else { echo '{"error: "' . $sql . '" . $conn->error.'}';

```

```

    }
} if($comando=='eliminar') {
    $id=$_GET["id"];
    // sql to delete a record
    $sql = "DELETE FROM productos WHERE id=$id";
    if ($conn->query($sql) === TRUE) {
        echo '{"mensaje":"Registro eliminado"}';
    }
else {
    echo '{"error: "' . $sql . '" . $conn->error.'}';
}
} if($comando=='listar') {
    $sql = "SELECT * FROM productos";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        // obtener cada uno de los registros y almacenarlos en un vector y
        luego regresarlos en formato json
        $registros=array();
        $i=0;
        while($row = $result->fetch_assoc()) {
            //echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
            $row["lastname"]. "<br>";
            $registros[$i]=$row; $i++;
        }
        echo '{"records":'.json_encode($registros).'}';
    } else {
        echo '{"records":[]}';
    }
}
$conn->close();
}
?>

```

4. Procederemos a crear un nuevo proyecto de nombre Market y copie el siguiente código en su archivo app.js

```

import * as React from 'react';
import { SafeAreaView, Button, View, Text,
Image, TouchableOpacity, Keyboard, Alert, FlatList, StyleSheet, ScrollView } from
'react-native';
import { createAppContainer, NavigationEvents } from 'react-navigation';
import { createStackNavigator } from 'react-navigation-stack';
import { Input } from 'react-native-elements';
import Icon from 'react-native-vector-icons/FontAwesome';

class PantallaInicio extends React.Component {
    state={
        usuario:'',
        contrasena:''
    }
}

```

```

    }
    static navigationOptions = {
      header: null
    };

    Entrar()
    {
      if (!!this.state.usuario && !!this.state.contrasena)
      {
        fetch('https://guia9dps.000webhostapp.com/apiusuario.php?comando=autenticar&
        usuario='+this.state.usuario+'&contrasena='+this.state.contrasena, {
          method: 'GET'
        })
        .then((response) => response.json())
        .then((responseJson) => {
          console.log(responseJson);
          const encontrado=responseJson.encontrado;
          // Alert("Mensaje="+mensaje);
          if(encontrado=='si')
          {
            this.props.navigation.navigate('ListarProductos')
          }
          else
          {
            Alert.alert(
              'Usuario',
              'No encontrado!!',
              [
                {text: 'OK', onPress: () => console.log('OK Pressed')},
              ],
              { cancelable: false }
            )
          }
        })
        .catch((error) => {
          console.error(error);
          Alert.alert(
            'Aviso',
            'Error de Internet!!',
            [
              {text: 'OK', onPress: () => console.log('OK Pressed')},
            ],
            { cancelable: false }
          );
        });
      }
    }
    else
    {
      Alert.alert(
        'Aviso',
        'No introdujo datos',

```

```

    [
      {text: 'OK', onPress: () => console.log('OK Pressed')},
    ],
    { cancelable: false }
  );
}
}

render() {
  return (
    <View style={{ flex:1,padding:10 }}>

      <Text
style={{fontSize:34,marginTop:25,alignSelf:'center'}}>Bienvenidos</Text>
      <Image
        style={{width: 200, height: 160,alignSelf:'center',marginTop:15}}
        source={require('./imagenes/market.jpg')}
      />

      <View style={{marginLeft:10,marginRight:10}}>
        <Input
placeholder='USUARIO'
onChangeText = {(text) => this.setState({usuario: text})}
rightIcon={
          <Icon
            name='user'
            size={24}
            color='black'
          />
        }
      />
        <Input
placeholder='CONTRASEÑA'
onChangeText = {(text) => this.setState({contrasena: text})}
secureTextEntry={true}
rightIcon={
          <Icon
            name='lock'
            size={24}
            color='black'
          />
        }
      />
    </View>

    <TouchableOpacity
      style={{height: 50,backgroundColor:'red',
marginTop:15,borderRadius:5,justifyContent:
'center',marginLeft:20,marginRight:20}}
      onPress={() => {this.Entrar()}}
    >

```

```

<Text style={{color:'white',fontSize:22,textAlign:'center',
textAlignVertical:'center'}}>Entrar</Text>
</TouchableOpacity>

```

```

    </View>
  );
}
}

```

```

class listarProductos extends React.Component {
  state = {
    elementos: [],
    total:0
  }
  static navigationOptions = {
    title: 'Productos',
    headerStyle: {
      backgroundColor: '#f4511e',
    },
    headerTintColor: '#fff',
    headerTitleStyle: {
      fontWeight: 'bold',
    },
  };

  cargarRegistros()
  {
    console.log('Prueba');
    fetch('https://guia9dps.000webhostapp.com/api.php?comando=listar', {
      method: 'GET'
    })
    .then((response) => response.json())
    .then((responseJson) => {
      console.log(responseJson);
      const listado=responseJson.records;
      console.log(listado);
      this.setState({
        elementos: listado,
        total:listado.length
      })
    })
    .catch((error) => {
      console.error(error);
    });
  }

  render() {
    return (

```

```

    <View style={{flex:1}} >
    <NavigationEvents
        onWillFocus={() => {
            // Do your things here
            this.cargarRegistros();
        }}
    />

    <Text
    style={{fontSize:18,textAlign:'center',height:40,marginTop:10,backgroundColo
r:'lightgray',textAlignVertical:'center',
borderRadius:10,marginLeft:10,marginRight:10}}>{this.state.total}
productos</Text>

    <FlatList
        data={this.state.elementos}
        renderItem={({ item }) => <TouchableOpacity
            key = {item.id}
            //onPress = {() => this.alertItemName(item)}
            onPress={() =>
this.props.navigation.navigate('Detalles',item)}
        >
            <View
style={{flexDirection:'row',marginTop:15,marginLeft:2}}>
                <Image
                    style={{width: 90, height: 90}}
                    source={{uri: item.fotografia}}
                />
                <View style={{height:80,marginLeft:5}}>
                    <Text style = {{flex:1,fontSize:18}}>
                        {item.nombre}
                    </Text>
                    <Text style = {{flex:1,fontSize:16,fontWeight:
'bold',}}>
                        ${item.preciodeventa}
                    </Text>
                    <Text style = {{flex:1,fontSize:14}}>
                        Existencia {item.cantidad}
                    </Text>
                </View>
            </View>
        </TouchableOpacity>
        keyExtractor={item => item.id}
    />
    <TouchableOpacity
    style={{
        borderWidth:1,
        borderColor:'rgba(0,0,0,0.2)',
        alignItems:'center',
        justifyContent:'center',
        width:70,
        position: 'absolute',
        bottom: 10,

```



```

        right: 10,
        height: 70,
        backgroundColor: 'red',
        borderRadius: 100,
      }}
      onPress={() => this.props.navigation.navigate('Agregar')}
    >
      <Icon name="plus" size={30} color="white" />
    </TouchableOpacity>
  </View>
);
}
}

class PaginaDetalle extends React.Component {
  state={
    nombre:'',
    descripcion:'',
    cantidad:'',
    preciodecosto:'',
    preciodeventa:'',
    fotografia:'',
    id:''
  }
  static navigationOptions = {
    title: 'Editar producto',
    headerStyle: {
      backgroundColor: '#f4511e',
    },
    headerTintColor: '#fff',
    headerTitleStyle: {
      fontWeight: 'bold',
    },
  },
};

Actualizar()
{
  fetch('https://guia9dps.000webhostapp.com/api.php?comando=editar&nombre='+this.state.nombre
    + '&descripcion='+this.state.descripcion
    + '&cantidad='+this.state.cantidad
    + '&preciodecosto='+this.state.preciodecosto
    + '&preciodeventa='+this.state.preciodeventa
    + '&fotografia='+this.state.fotografia
    + '&id='+this.state.id, {
      method: 'GET'
    })
    .then((response) => response.json())
    .then((responseJson) => {
      console.log(responseJson);
    });
}

```

```

        const mensaje=responseJson.mensaje;
        console.log(mensaje);
        if(!mensaje)
            alert("Error al actualizar!");
        else
        {
            alert(mensaje);
            this.props.navigation.goBack();
        }
    })
    .catch((error) => {
        console.error(error);
        alert("Error de Internet!!");
    });
}

Eliminar()
{
    fetch('https://guia9dps.000webhostapp.com/api.php?comando=eliminar&id='+this
    .state.id, {
        method: 'GET'
    })
    .then((response) => response.json())
    .then((responseJson) => {
        console.log(responseJson);
        const mensaje=responseJson.mensaje;
        console.log(mensaje);
        if(!mensaje)
            alert("Error al eliminar!");
        else
        {
            alert(mensaje);
            this.props.navigation.goBack();
        }
    })
    .catch((error) => {
        console.error(error);
        alert("Error de Internet!!");
    });
}

render() {
    const { navigation } = this.props;
    return (
        <View style={{flex:1}}>
        <ScrollView style={{flex:1}}>
        <View
        style={{flex:1,flexDirection:'row',alignContent:'center',alignItems:'center'
        ,height:60}}>
        <TouchableOpacity

```

```

        style={{flex:1,height:
40,backgroundColor:'black',borderRadius:5,justifyContent:
'center',marginLeft:5}}
        onPress={() => {this.Actualizar()}}
      >
      <Text style={{color:'white',fontSize:22,textAlign:'center',
textAlignVertical:'center',padding:3}}>Actualizar</Text>
    </TouchableOpacity>
    <TouchableOpacity
      style={{flex:1,height:
40,backgroundColor:'black',borderRadius:5,justifyContent:
'center',marginLeft:5,marginRight:5}}
      onPress={() => {this.Eliminar()}}
    >
      <Text style={{color:'white',fontSize:22,textAlign:'center',
textAlignVertical:'center',padding:3}}>Eliminar</Text>
    </TouchableOpacity>
  </View>

  <View style={{ flex: 1,padding:20 }}>

    <NavigationEvents
      onWillFocus={() => {
        // Do your things here
        console.log("Entro aqui"+navigation.getParam('nombre'));
        this.setState({nombre:navigation.getParam('nombre'),
descripcion:navigation.getParam('descripcion'),
                          cantidad:navigation.getParam('cantidad'),
preciodecosto:navigation.getParam('preciodecosto'),
preciodeventa:navigation.getParam('preciodeventa'),
fotografia:navigation.getParam('fotografia'),
                          id:navigation.getParam('id')
                        });
      }}
    />

    <Input
      label="Nombre"
      value={this.state.nombre}
      placeholder="Nombre"
      onChangeText = {(text) => this.setState({nombre: text})}
    />
    <Input
      label="Descripción"
      value={this.state.descripcion}
      inputStyle={{marginTop:10}}
      placeholder="Descripción"
      onChangeText = {(text) => this.setState({descripcion: text})}
    />

```

```

<Input
  label="Precio de costo"
  value={this.state.preciodecosto}
  inputStyle={{marginTop:10}}
  placeholder='Precio de costo'
  onChangeText = {(text) => this.setState({preciodecosto: text})}
/>
<Input
  label="Precio de venta"
  value={this.state.preciodeventa}
  inputStyle={{marginTop:10}}
  placeholder='Precio de venta'
  onChangeText = {(text) => this.setState({preciodeventa: text})}
/>
<Input
  label="Cantidad"
  value={this.state.cantidad}
  inputStyle={{marginTop:10}}
  placeholder='Cantidad'
  onChangeText = {(text) => this.setState({cantidad: text})}
/>
<Input
  label="Fotografía"
  value={this.state.fotografia}
  inputStyle={{marginTop:10}}
  placeholder='URL de fotografía'
  onChangeText = {(text) => this.setState({fotografia: text})}
/>
<Image
  style={{width: 100, height: 100,alignSelf:'center'}}
  source={{uri: this.state.fotografia}}
/>
</View>
</ScrollView>
</View>

);
}
}

```

```

class PaginaAgregar extends React.Component {
  state={
    nombre:'',
    descripcion:'',
    preciodeventa:'',
    preciodecosto:'',
    cantidad:'',
    fotografia:'',
  }
  static navigationOptions = {
    title: 'Agregar producto',
  }
}

```

```

    headerStyle: {
      backgroundColor: '#f4511e',
    },
    headerTintColor: '#fff',
    headerTitleStyle: {
      fontWeight: 'bold',
    },
  },
};

Guardar()
{
  fetch('https://guia9dps.000webhostapp.com/api.php?comando=agregar&nombre='+this.state.nombre
    + '&descripcion='+this.state.descripcion
    + '&cantidad='+this.state.cantidad
    + '&preciodecosto='+this.state.preciodecosto
    + '&preciodeventa='+this.state.preciodeventa
    + '&fotografia='+this.state.fotografia, {
      method: 'GET'
    })
    .then((response) => response.json())
    .then((responseJson) => {
      console.log(responseJson);
      const mensaje=responseJson.mensaje;
      console.log(mensaje);
      if(!mensaje)
        alert("Error al agregar!");
      else
      {
        alert(mensaje);
        this.props.navigation.goBack();
      }
    })
    .catch((error) => {
      console.error(error);
      alert("Error de Internet!!");
    });
}

render() {
  return (
    <View style={{ flex: 1, padding:20 }}>
      <Input
        placeholder='Nombre'
        onChangeText = {(text) => this.setState({nombre: text})}
      />
      <Input
        inputStyle={{marginTop:10}}
        placeholder='Descripción'
        onChangeText = {(text) => this.setState({descripcion: text})}

```

```

    />
    <Input
      inputStyle={{marginTop:10}}
      placeholder='Precio de costo'
      onChangeText = {(text) => this.setState({preciodecosto: text})}
    />
    <Input
      inputStyle={{marginTop:10}}
      placeholder='Precio de venta'
      onChangeText = {(text) => this.setState({preciodeventa: text})}
    />
    <Input
      inputStyle={{marginTop:10}}
      placeholder='Cantidad'
      onChangeText = {(text) => this.setState({cantidad: text})}
    />
    <Input
      inputStyle={{marginTop:10}}
      placeholder='URL de fotografía'
      onChangeText = {(text) => this.setState({fotografia: text})}
    />
    <TouchableOpacity
      style={{height: 50,backgroundColor:'red',
marginTop:15,borderRadius:5,justifyContent:
'center',marginLeft:20,marginRight:20}}
      onPress={() => {this.Guardar()}}
    >
      <Text style={{color:'white',fontSize:22,textAlign:'center',
textAlignVertical:'center'}}>Guardar</Text>
    </TouchableOpacity>
  </View>

  );
}
}
const RootStack = createStackNavigator(
  {
    Inicio: PantallaInicio,
    ListarProductos: listarProductos,
    Detalles:PaginaDetalle,
    Agregar:PaginaAgregar,
  },
  {
    initialRouteName: 'Inicio',
  }
);

const AppContainer = createAppContainer(RootStack);

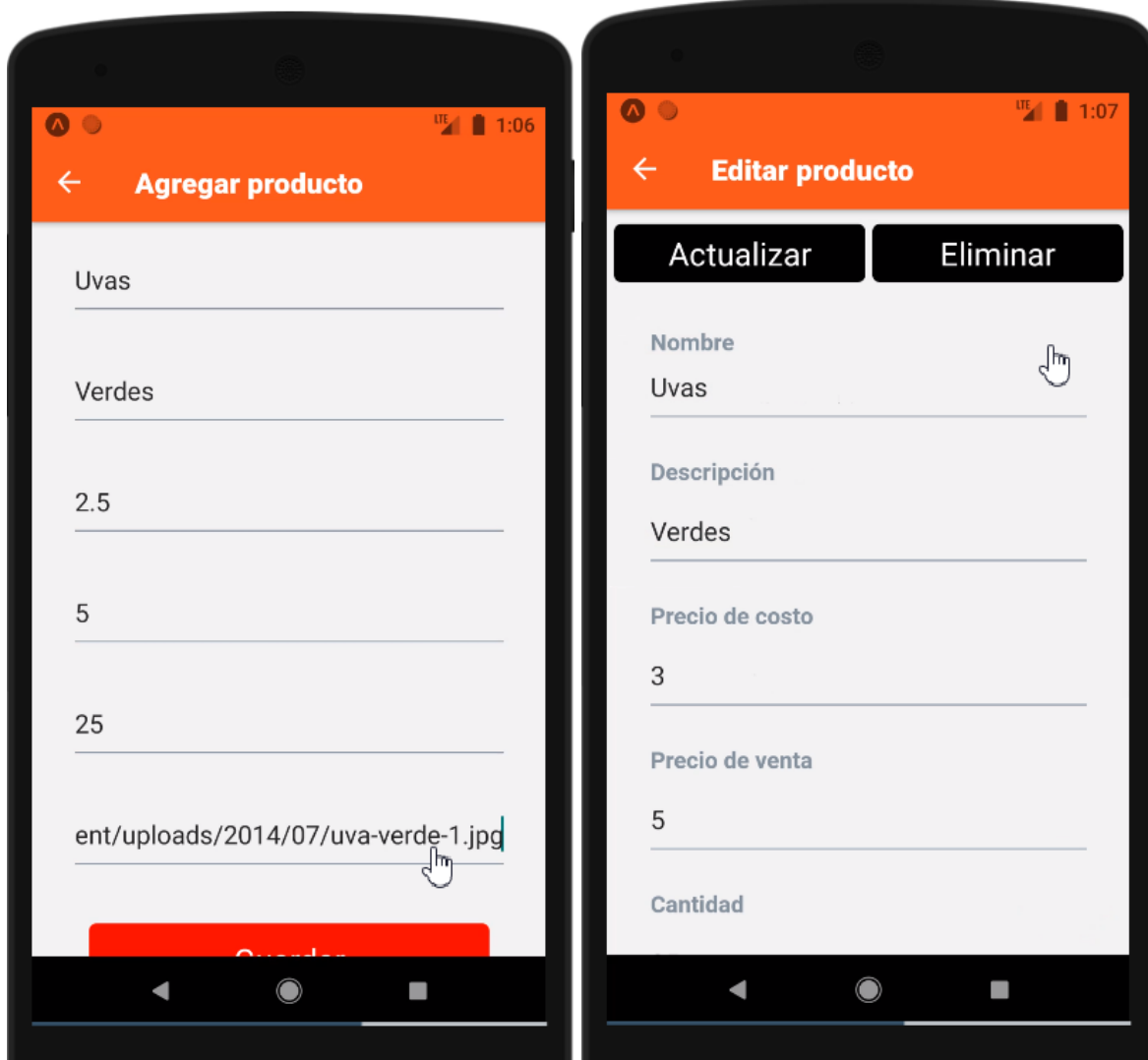
export default class App extends React.Component {
  render() {
    return <AppContainer />;
  }
}

```

```
}  
}
```

5. Crear una nueva carpeta de nombre imagenes y guardar archivo market.jpg proporcionado en los recursos.
6. Realice las respectivas pruebas en la aplicación usando el usuario:karens, contraseña: 123456, la cual debería presentarnos las siguientes pantallas:





V. EJERCICIOS COMPLEMENTARIOS

- Basándose en la aplicación creada anteriormente cree una app para gestión de cartera de clientes en donde se debe manejar la siguiente información:
 - Nombre y apellidos de los clientes reales y posibles.
 - Dirección postal y del trabajo.
 - Teléfono y correo electrónico.
 - Datos adicionales: nivel económico, posibilidades de compra, intereses, etc.