

	<b>UNIVERSIDAD DON BOSCO</b> <b>FACULTAD DE INGENIERÍA</b> <b>ESCUELA DE COMPUTACIÓN</b>	
<b>CICLO: 02</b>	<b>GUIA DE LABORATORIO #08</b>	
	<b>Nombre de la Práctica:</b>	<b>Aplicaciones usando API -GET</b>
	<b>MATERIA:</b>	<b>Diseño y Programación de Software Multiplataforma</b>

### I. RESULTADOS DE APRENDIZAJE

Que el estudiante:

- Adquiera el dominio para aplicaciones de lecturas de API
- Cree proyectos react native utilizando API para la extracción de datos

### II. INTRODUCCIÓN

React Navigation es una biblioteca popular para realizar enrutamiento y navegación en una aplicación React Native.

React Navigation se compone de algunas utilidades principales y los navegadores las utilizan para crear la estructura de navegación en su aplicación.

### III. PROCEDIMIENTO DE GUÍA

#### PARTE I:

1. Cree un proyecto react-native con nombre Países
2. Ahora vamos una carpeta src, que contenga una carpeta llamada components
3. Ahora procederemos a crear nuestro formulario para hacer las peticiones este archivo se guardará dentro de la carpeta components, con el nombre de Formulario.js el cual tendrá el siguiente código:

```
import * as React from 'react';
import { Text, TextInput, View, StyleSheet, TouchableWithoutFeedback }
from 'react-native';
import {Picker} from '@react-native-community/picker';
const Formulario=()=>{
  return(
    <>
    <View>
      <View>
        <Text style ={styles.input}>País</Text>
      </View>
    </View>
  )
}
```

```

        <Picker style={styles.itempais}>
        <Picker.Item label="--seleccione un pais--" value=""/>
        <Picker.Item label="El Salvador" value="sv"/>
        <Picker.Item label="Guatemala" value="gt"/>
        <Picker.Item label="Honduras" value="hn"/>
        <Picker.Item label="Nicaragua" value="ni"/>
        <Picker.Item label="Panama" value="pa"/>
        <Picker.Item label="Costa Rica" value="cr"/>
        <Picker.Item label="Mexico" value="mx"/>
        <Picker.Item label="Argentina" value="ar"/>
        <Picker.Item label="Estados Unidos" value="us"/>
        <Picker.Item label="Colombia" value="co"/>
        <Picker.Item label="España" value="es"/>
        <Picker.Item label="Peru" value="pe"/>
        </Picker>
      </View>
      <TouchableWithoutFeedback >
      <View style={styles.btnBuscar}>
      <Text style={styles.textoBuscar}>Buscar País</Text>
      </View>
    </TouchableWithoutFeedback>
  </View>
</>
);
}

```

```

const styles= StyleSheet.create({
  input:{padding:10,
  height:50,
  fontSize:20,
  marginBottom:20,
  textAlign:'center',
  color : '#000'
  },
  itempais:{
    height:60,
    backgroundColor: '#fff',
  },
  btnBuscar:{
    marginTop:50,
    height:50,
    backgroundColor: '#000',
    fontSize:20,
    marginBottom:20,
    textAlign: 'center'
  },
  textoBuscar:{
    color: '#fff',
    fontWeight: 'bold',
    textTransform: 'uppercase',
    textAlign: "center",
    fontSize:18
  }

```

```

    })
  })

export default Formulario;

```



4. Ahora procederemos a crear funciones para agregar animación a nuestro botón buscar país, modificaremos nuestro archivo formulario:

- ◆ agregar las siguientes librerías.

```

import React, {useState} from 'react';
import { Text, TextInput, View, StyleSheet, TouchableWithoutFeedback, Animated } from
'react-native';
import {Picker} from '@react-native-community/picker';

```

- ◆ declarar la siguiente variable

```

const Formulario=()=>>{

  const [animacionboton]=useState(new Animated.Value(1));

  return(

```

- ◆ ahora crearemos las siguientes funciones

```

const Formulario=()=>>{

  const [animacionboton]=useState(new Animated.Value(1));

  const animacionEntrada=()=>{
    Animated.spring(animacionboton,{
      toValue:.9
    }).start();
  }

  const animacionSalida=()=>>{
    Animated.spring(animacionboton,{
      toValue:1
    }).start();
  }
}

```

- ◆ agregar el siguiente estilo

```
const animacionSalida={()=>{
  console.log('Salida');
}}

const estiloAnimacion={
  transform:[{
    scale:animacionboton
  }]
}
```

- ◆ modificar nuestro view del Touchablewithoutfeedback

```
</View>
<TouchableWithoutFeedback
  onPressIn={()=> animacionEntrada()}
  onPressOut={()=> animacionSalida()}
>
  <Animated.View style={[styles.btnBuscar, estiloAnimacion]}>
    <Text style={styles.textoBuscar}>Buscar País</Text>
  </Animated.View>
</TouchableWithoutFeedback>
</View>
```

5. Pulse el botón buscar país y verifique que ya posee animación.

6. Leer los valores del formulario

- modificar el archivo app.js, agregar la siguiente librerías y constantes

```
import React, {useState} from 'react';
import { Text, View, StyleSheet } from 'react-native';
import Formulario from './components/Formulario';

export default function App() {
  const [busqueda, guardarbusqueda]=useState({
    pais:''
  })

  return (
    <View style={styles.app}>
      <View style={styles.contenido}>
        <Formulario
          busqueda={busqueda}
          guardarbusqueda={guardarbusqueda}
        />
      </View>
    </View>
  );
}
```

- modificar nuestro archivo formulario.js creando sus estados

```
import React,{useState} from 'react';
import { Text, TextInput, View, StyleSheet, TouchableWithoutFeedback } from 'react-native';
import { Picker } from '@react-native-community/picker';
const Formulario=({busqueda,guardarbusqueda})=>{
  const {pais}=busqueda;

  const [animacionboton]=useState(new Animated.Value(1));
```

- modificar nuestro archivo formulario.js para leer los datos seleccionados

```
<View>
  <Picker
    selectedValue={pais}
    onChange={pais=>guardarbusqueda({...busqueda,pais})}
    style={styles.itempais}>
    <Picker.Item label="--seleccione un pais--" value="" />
    <Picker.Item label="El Salvador" value="sv"/>
    <Picker.Item label="Guatemala" value="gt"/>
    <Picker.Item label="Honduras" value="hn"/>
```

## 7. agregar validación a nuestro botón

- Agregar las siguientes librerías y funciones

```
import React,{useState} from 'react';
import { Text, TextInput, View, StyleSheet, TouchableWithoutFeedback, Animated, Alert } from 'react-native';
import { Picker } from '@react-native-community/picker';
const Formulario=({busqueda,guardarbusqueda})=>{
  const {pais}=busqueda;

  const [animacionboton]=useState(new Animated.Value(1));

  const consultarPais=()=>{
    if(pais.trim()==''){
      mostrarAlerta();return;
    }
  }

  const mostrarAlerta=()=>{
    Alert.alert('Error','Debe seleccionar un país'),[{Text:'Entendido'}]
  }

  const animacionEntrada=()=>{
```

- modificar nuestro evento del botón

```
<Picker
  selectedValue={pais}
  onChange={pais=>guardarbusqueda({...busqueda,pais})}
  onPress={()=>consultarPais()}
  style={styles.itempais}>
```

## 8. Ahora vamos a crear los resultados que deberá ver el usuario si pasa la validación que creamos anteriormente:

- ◆ vamos a crear un nuevo componente de nombre pais.js que debe llevar el siguiente código:

```

import React from 'react';
import { StyleSheet, View, Text } from 'react-native';
import { Card } from 'react-native-elements';
const Pais = ({ resultado }) => {
  const { capital, name, subregion, region } = resultado;

  return (
    <Card>
      <Card.Title>{name}</Card.Title>
      <Card.Divider />
      <View style={{ justifyContent: 'center' }}>
        <Text>Capital:{capital}</Text>
        <Text>Region:{region}</Text>
        <Text>Subregion:{subregion}</Text>
      </View>
    </Card>
  );
};

export default Pais;

```

- ◆ vamos a modificar nuestro app.js agregando las siguientes librerías

```

import React, { useState, useEffect } from 'react';
import { Text, View, StyleSheet, Alert } from 'react-native';
import Formulario from './components/Formulario';
import Pais from './components/Pais';
export default function App() {

```

- ◆ vamos a modificar nuestro archivo app.js de la siguiente manera para crear las funciones y variables necesarias para poder realizar nuestra petición a nuestra API de lectura

```

export default function App() {
  const [busqueda, guardarbusqueda] = useState({
    pais: '',
  });

  const [consultar, guardarconsultar] = useState(false);
  const [resultado, guardarresultado] = useState({});

  useEffect(() => {
    const { pais } = busqueda;
    const consultarPais = async () => {
      if (consultar) {
        const url = `https://restcountries.eu/rest/v2/alpha/${pais}`;
        try {
          const respuesta = await fetch(url);
          const resultado = await respuesta.json();

          guardarresultado(resultado);
          guardarconsultar(false);
        } catch (error) {
          mostrarAlerta();
        }
      }
    };
    consultarPais();
  }, [consultar]);

  const mostrarAlerta = () => {
    Alert.alert('Error', 'No hay resultado intenta con otra ciudad o país',
      [{ Text: 'Ok' }]);
  };

  return (
    <View style={styles.app}>
      <View style={styles.contenido}>
        <Formulario
          busqueda={busqueda}
          guardarbusqueda={guardarbusqueda}
          guardarconsultar={guardarconsultar}
        />
        <Pais resultado={resultado} />
      </View>
    </View>
  );
}

```

◆ ahora deberemos modificar también nuestro archivo formulario.js

```

const [animacionboton] = useState(new Animated.Value(1));

const consultarPais = () => {
  if (pais.trim() === '') {
    mostrarAlerta();
    return;
  }
  //consultar API
  guardarconsultar(true);
};

const mostrarAlerta = () => {
  Alert.alert('Error', 'Debe seleccionar un país', [{ Text: 'Entendido' }]);
};

```

9. Realice las respectivas pruebas de la aplicación donde deberá mostrar, el nombre, la capital, región y subregión del país seleccionado.

#### IV. EJERCICIO COMPLEMENTARIO

Tomando como base el ejercicio anterior deberá modificarlo de tal forma que los países que se presentan en el picker venga de nuestra uri: <https://restcountries.eu/rest/v2>