

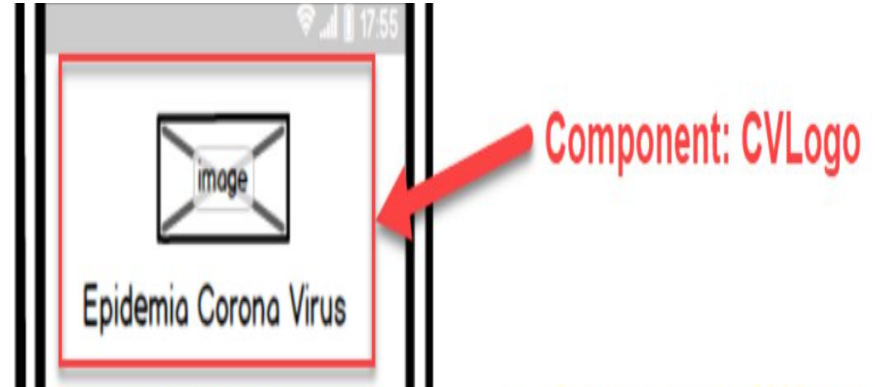
# Procesual hito 2

Estudiante: Jose Luis Rodriguez Peredo

Materia: Programación de Dispositivos Móviles.

Docente: Ing. William Barra.

1)Desarrollar el  
componente  
CVLogo y mostrar  
la vista

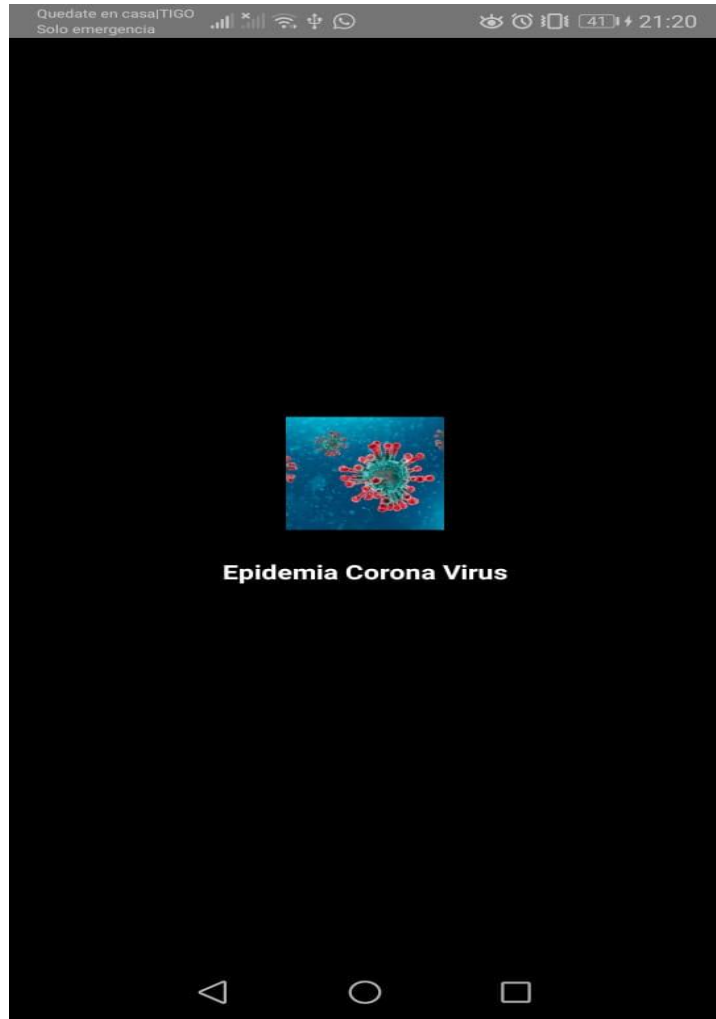


Respuesta en la siguiente diapositiva->

```

import React, {Component} from 'react';
import {View, Text, Image, StyleSheet} from 'react-native';
import imgLogo from '../images/virus.jpg';
import Constants from '../Config/Constants';
export default class CVLogo extends Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <View style={styles.container}>
        <Image source={imgLogo} style={styles.image} />
        <Text style={styles.text}>{Constants.NAME_APP}</Text>
      </View>
    );
  }
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
    backgroundColor: 'black'
  },
  image: {
    width: 80,
    height: 80,
  },
  text: {
    color: 'white',
    fontWeight: 'bold',
    backgroundColor: 'transparent',
    marginTop: 20,
  },
});

```



## 2) Desarrollar el componente CVCasos y mostrar la vista



COCHABAMBA	
Casos Confirmados	Nro de Casos
Casos Sospechosos	Nro de Casos

Respuesta en la siguiente diapositiva->

```

import React from 'react';
import {StyleSheet, View, Text, TextInput, Image} from
'react-native';
import PropTypes from 'prop-types';
import Colors from '../src/Config/Colors';
const CVCasos=(props)=>{
  const
{typeCase,secureTextEntry,placeholder,onChangeText,autoCorr
ect} = props;
  return(
    <View>
    <Text style={styles.text}>{typeCase}</Text>
    <TextInput
    style={styles.textInput}
    onChangeText={onChangeText}
    placeholder={placeholder}
    autoCorrect={autoCorrect}
    secureTextEntry={secureTextEntry}
    selectionColor={Colors.blue}
    placeholderTextColor={Colors.white}
    underlineColorAndroid="transparent">
    </TextInput>
    </View>
  );
}
CVCasos.propTypes = {
  onChangeText:PropTypes.func.isRequired,
  placeholder: PropTypes.string.isRequired,
  autoCorrect: PropTypes.bool,
  secureTextEntry: PropTypes.bool,
};

```

```

const styles = StyleSheet.create({
  textInput: {
    backgroundColor:Colors.blue,
    alignItems: 'center',height: 40,
    borderColor: Colors.silver,
    paddingLeft: 40,
    borderRadius: 10,
    borderBottomWidth: StyleSheet.hairlineWidth,
    marginBottom: 15,
    marginLeft:50,
  },
  text: {
    position: 'absolute',
    color:Colors.white,
    backgroundColor:'transparent',
    zIndex: 99,
    width: 100,
    height: 100,
    left: -40,top: 0,
  },
});
export default CVCasos;

```



### 3) Desarrollar el componente CVCiudad y mostrar la vista



The screenshot shows a web form titled "SANTA CRUZ". It contains two rows of labels and input fields. The first row has the label "Casos Confirmados" and an input field labeled "Nro de Casos". The second row has the label "Casos Sospechosos" and another input field labeled "Nro de Casos". A green arrow points from the text "Component: CVCiudad" to the first input field.

SANTA CRUZ	
Casos Confirmados	Nro de Casos
Casos Sospechosos	Nro de Casos

Component: CVCiudad

Respuesta en la siguiente diapositiva->

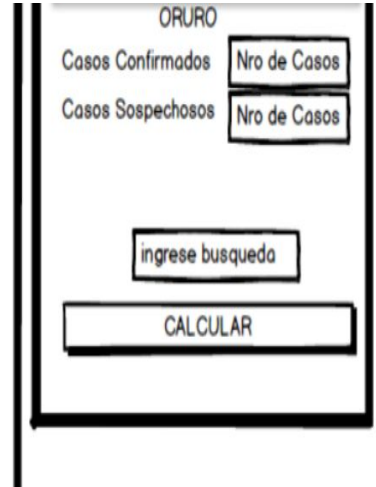
```

import React from 'react';
import {StyleSheet,View,Text}from 'react-native';
import PropTypes from 'prop-types';
import CVCasos from './CVCasos';
import Colors from '../src/Config/Colors';
import Constants from '../src/Config/Constants';
const CVCiudad = (props) =>{
  const {Ciudad, onChangeTextConf, onChangeTextSosp} = props;
  return(
    <View>
      <Text style={styles.text}>{Ciudad}</Text>
      <CVCasos
        onChangeText={onChangeTextConf}
        placeholder={Constants.NCASOS}
        autoCorrect={false}
        secureTextEntry={false}
        typeCase={Constants.CONFIRMADOS}>
      </CVCasos>
      <CVCasos
        onChangeText={onChangeTextSosp}
        placeholder={Constants.NCASOS}
        autoCorrect={false}
        secureTextEntry={false}
        typeCase={Constants.SOSPECHOSOS}>
      </CVCasos>
    </View>
  );
}
const styles = StyleSheet.create({
  text:{
    marginLeft:50,
    marginBottom:10,
    color:Colors.white,
    backgroundColor:'transparent',
  },
});
export default CVCiudad;

```



## 4) Desarrollar el componente CVScreen y mostrar la vista



The diagram shows a rectangular window titled "ORURO". Inside the window, there are two rows of labels and input fields. The first row has the label "Casos Confirmados" followed by an input field labeled "Nro de Casos". The second row has the label "Casos Sospechosos" followed by another input field labeled "Nro de Casos". Below these input fields, there is a button labeled "ingrese busqueda". At the bottom of the window, there is a wide button labeled "CALCULAR".

**Component: CVScreen**  
Este componente tiene todos los componentes creados anteriormente.

Respuesta en la siguiente diapositiva->



```

import React,{Component} from 'react';
import {StyleSheet,View,TextInput} from 'react-native';
import CVLogo from './CVLogo';
import CVCiudad from './CVCiudad';
import Colors from '../src/Config/Colors';
import Constants from '../src/Config/Constants';
import Button from './Components/login/Button';
export default class CVScreen extends Component{
  constructor(props){
    super(props);
    this.state = {
      confCB:o,
      sospCB:o,
      confOR:o,
      sospOR:o,
      confSC:o,
      sospSC:o,
      search:''
    };
    this._onChangeTextSospCb=this._onChangeTextSospCb.bind(this);
    this._onChangeTextConfCb=this._onChangeTextConfCb.bind(this);
    this._onChangeTextSospSc=this._onChangeTextSospSc.bind(this);
    this._onChangeTextConfSc=this._onChangeTextConfSc.bind(this);
    this._onChangeTextSospOr=this._onChangeTextSospOr.bind(this);
    this._onChangeTextConfOr=this._onChangeTextConfOr.bind(this);
    this._onChangeTextInputSearch=this._onChangeTextInputSearch.bind(this);
    this._onPressCalcular=this._onPressCalcular.bind(this);
  }
  _onChangeTextConfCb(confCB){
    this.setState({
      confCB:confCB
    });
  }
  _onChangeTextSospCb(sospCB){
    this.setState({
      sospCB:sospCB
    });
  }
  _onChangeTextConfSc(confSC){
    this.setState({
      confSC:confSC
    });
  }
  _onChangeTextSospSc(sospSC){
    this.setState({
      sospSC:sospSC
    });
  }
}

```

```

  _onChangeTextConfOr(confOR){
    this.setState({
      confOR:confOR
    });
  }
  _onChangeTextSospOr(sospOR){
    this.setState({
      sospOR:sospOR
    });
  }
  _onChangeTextInputSearch(search){
    this.setState({
      search:search
    });
  }
  _onPressCalcular(){
    let n1CB, n2SC, n3OR, busqueda;
    if ( this.state.search === 'Conf' ){
      n1CB= parseInt(this.state.confCB);
      n2SC= parseInt(this.state.confSC);
      n3OR= parseInt(this.state.confOR);}
    if (this.state.search === 'Sosp'){
      n1CB=parseInt(this.state.sospCB);
      n2SC= parseInt(this.state.sospSC);
      n3OR= parseInt(this.state.sospSC);}
    let mayor = Math.max(n1CB, n2SC, n3OR);
    console.log('Pricionaste el boton calcular');
    if(this.state.search==='Conf'){
      console.log('El mayor # de casos confirmados es:',mayor);}
    if(this.state.search==='Sosp'){
      console.log('El mayor # de casos sospechosos es: ',mayor)}
  }
  render(){
    return(
      <View style={styles.container}>
        <CVLogo style={styles.logo}></CVLogo>
        <View>
          <CVCiudad
            Ciudad={Constants.CB}
            onChangeTextConf={this._onChangeTextConfCb}
            onChangeTextSosp={this._onChangeTextSospCb}>
          </CVCiudad>
          <CVCiudad
            Ciudad={Constants.SC}
            onChangeTextConf={this._onChangeTextConfSc}
            onChangeTextSosp={this._onChangeTextSospSc}>
          </CVCiudad>
          <CVCiudad
            Ciudad={Constants.OR}
            onChangeTextConf={this._onChangeTextConfOr}
            onChangeTextSosp={this._onChangeTextSospOr}>
          </CVCiudad>

```

```
</View>
```

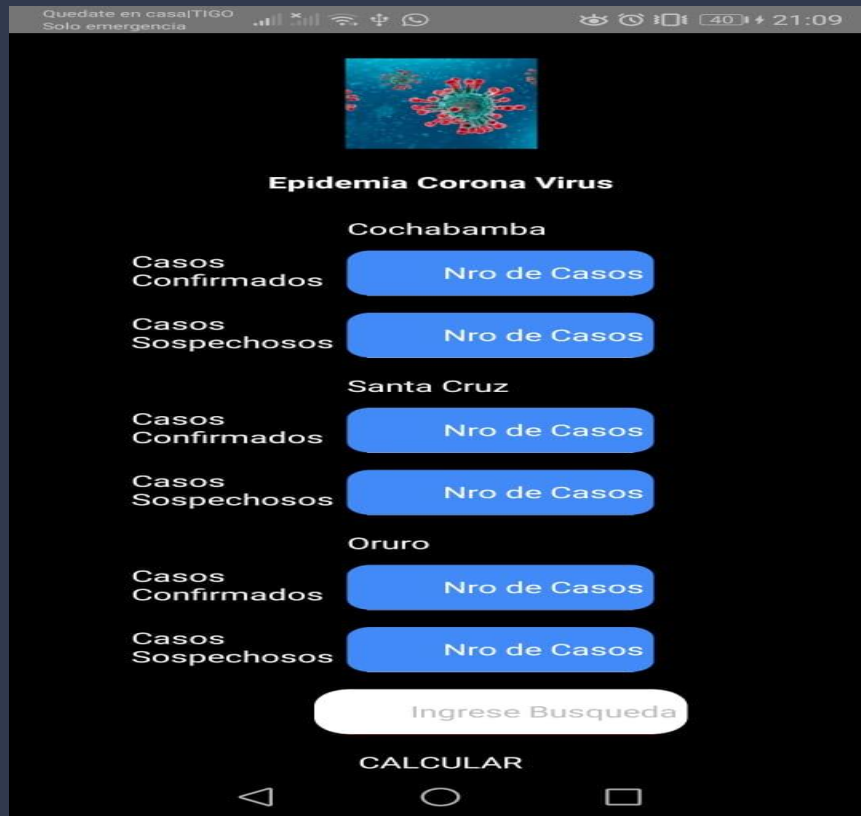
```
<TextInput
  style={styles.textInput}
  onChangeText={this._onChangeTextInputSearch}
  selectionColor={Colors.blue}
  placeholder={Constants.INGRESE}
  secureTextEntry={false}
  autoCorrect={false}
  placeholderTextColor={Colors.silver}
  underlineColorAndroid="transparent">
</TextInput>
<Button
  style={styles.boton}
  onPress={this._onPressCalcular}
  titleButton={Constants.CALCULAR}>
</Button>
</View>
);
```

```
}
}
const styles= StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: Colors.black,
    alignItems: 'center',
    justifyContent: 'space-between',
  },
  logo:{
    flex: 1,
    width: '100%',
    resizeMode: 'contain',
    alignSelf: 'center',
  },
  text:{
    color:Colors.black,
    fontSize:20,
    fontWeight:'bold',
  },
  textInput:{
    backgroundColor:Colors.blue,
    borderColor:Colors.white,
    alignItems: 'center',
    height: 40,
    paddingLeft: 40,
    borderRadius: 15,
    color:Colors.black,
    borderBottomWidth: StyleSheet.hairlineWidth,
    marginBottom: 15,
    marginLeft:50,
  },
});
```

```
boton:{
```

```
  borderBottomWidth: StyleSheet.hairlineWidth,
  backgroundColor:Colors.blue,
  color:Colors.blue,
  borderRadius:15,
  borderColor:Colors.black
},
```

```
});
```



5)Desarrollar el App.js mostrando funcionalidad del examen.

The screenshot shows a mobile application interface for tracking COVID-19 cases. The interface is organized into sections for different Bolivian cities: COCHABAMBA, SANTA CRUZ, and ORURO. Each city section contains two rows of data: 'Casos Confirmados' and 'Casos Sospechosos', each followed by an input field labeled 'Nro de Casos'. The top section features a logo and the title 'Epidemia Corona Virus'. At the bottom, there is a search bar labeled 'ingrese busqueda' and a 'CALCULAR' button.

Component: CVLogo

Component: CVCasos

Component: CVCiudad

Component: CVScreen  
Este componente tiene todos los componentes creados anteriormente.

Respuesta en la siguiente diapositiva->

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

```
import CVScreen from './src/DefensaHito2/CVScreen';
import CVCasos from './src/DefensaHito2/CVCasos';
import CVLogo from './src/DefensaHito2/CVLogo';
import CVCiudad from './src/DefensaHito2/CVCiudad';
```

```
const App: () => React$Node = () => {
  return (
    <CVLogo></CVLogo>
    <CVCasos></CVCasos>
    <CVCiudad></CVCiudad>
    <CVScreen></CVScreen>
  );
};
export default App;
```

```
[Sat Apr 11 2020 17:41:51.110] LOG Precionaste el boton calcular
[Sat Apr 11 2020 17:41:51.120] LOG {"confCB": "3", "confOR": "8", "confSC": "6", "search": "Conf", "sospCB": 0, "sospOR": 0, "sospSC": 0}
[Sat Apr 11 2020 17:42:34.442] LOG El mayor # de casos confirmados es: 18
[Sat Apr 11 2020 17:42:34.444] LOG Precionaste el boton calcular
[Sat Apr 11 2020 18:13:16.912] BUNDLE ./index.js

[Sat Apr 11 2020 18:13:19.146] LOG Running "DefensaHito2" with {"rootTag":1}
[Sat Apr 11 2020 18:13:37.299] LOG Precionaste el boton calcular
[Sat Apr 11 2020 18:13:37.349] LOG El mayor # de casos confirmados es: 11
```

