

React Nivel Avanzado

Sesión sincrónica 8

# APIs





# Bienvenida y actividad de bienestar

Duración: 10 minutos.

Nombre de la práctica	Concentrarse en lo positivo.
Descripción de la práctica	Analizarás sucesos que te hayan ocurrido recientemente, buscando orientar el análisis hacia las consecuencias positivas.
Palabras clave	Resiliencia y esperanza.
Instrucciones para el aprendedor	<p>¿Qué es lo primero que piensas cuando recibes una noticia inesperada?, o bien, ¿qué te imaginas cuando un acontecimiento complejo se presenta ante ti?</p> <p>La mayoría de las personas automáticamente se concentra en el peor de los escenarios, independientemente del tipo de noticia que reciban. Martin Seligman (2011), sugiere hacer un breve ejercicio para fomentar la resiliencia y la esperanza con base en la premisa antes señalada:</p> <ol style="list-style-type: none"><li>1. Piensa en una noticia reciente de la que te hayas enterado y que creas que es negativa para ti.</li><li>2. Luego de analizarla, haz una tabla con tres columnas. En la primera, señala cuál sería el peor de los escenarios posibles que pudieran resultar; en la segunda, señala cuál sería el mejor y en la última, cuál es el que realmente tiene mayor probabilidad de ocurrir.</li><li>3. Reflexiona sobre los tres escenarios, ¿cómo enfrentarías cada uno de ellos?</li></ol> <p>Procura repetir este ejercicio cada vez que te enfrentes a una situación complicada. Hacerlo te dará perspectiva y te ayudará a cultivar tu resiliencia.</p>

# Bienvenida y actividad de bienestar

Fuente	Seligman, M. (2011). <i>Building Resilience</i> . Recuperado de <a href="https://hbr.org/2011/04/building-resilience">https://hbr.org/2011/04/building-resilience</a>
--------	---

# Actividad guiada

## Parte 1

**Duración: 75 minutos.**

Implementa las siguientes APIs en tu aplicación:

1. Muestra un indicador circular que indique que la aplicación está cargando:

```
import React from "react";
import { ActivityIndicator, StyleSheet, Text, View } from "react-native";

const App = () => (
  <View style={[styles.container, styles.horizontal]}>
    <ActivityIndicator />
    <ActivityIndicator size="large" />
    <ActivityIndicator size="small" color="#0000ff" />
    <ActivityIndicator size="large" color="#00ff00" />
  </View>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center"
  },
  horizontal: {
    flexDirection: "row",
    justifyContent: "space-around",
    padding: 10
  }
});

export default App;
```

# Actividad guiada

## Parte 1

2. Crea un switch en la aplicación:

```
import React, { useState } from "react";
import { View, Switch, StyleSheet } from "react-native";

const App = () => {
  const [isEnabled, setIsEnabled] = useState(false);
  const toggleSwitch = () => setIsEnabled(previousState => !previousState);

  return (
    <View style={styles.container}>
      <Switch
        trackColor={{ false: "#767577", true: "#81b0ff" }}
        thumbColor={isEnabled ? "#f5dd4b" : "#f4f3f4"}
        ios_backgroundColor="#3e3e3e"
        onValueChange={toggleSwitch}
        value={isEnabled}
      />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center"
  }
});

export default App;
```

# Actividad guiada

## Parte 1

3. Ajusta automáticamente la altura, posición o relleno inferior en función de la altura del teclado

```
import React from 'react';
import { View, KeyboardAvoidingView, TextInput, StyleSheet, Text, Platform,
TouchableWithoutFeedback, Button, Keyboard } from 'react-native';

const KeyboardAvoidingComponent = () => {
  return (
    <KeyboardAvoidingView
      behavior={Platform.OS === "ios" ? "padding" : "height"}
      style={styles.container}
    >
      <TouchableWithoutFeedback onPress={Keyboard.dismiss}>
        <View style={styles.inner}>
          <Text style={styles.header}>Header</Text>
          <TextInput placeholder="Username" style={styles.textInput} />
          <View style={styles.btnContainer}>
            <Button title="Submit" onPress={() => null} />
          </View>
        </View>
      </TouchableWithoutFeedback>
    </KeyboardAvoidingView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1
  },
  inner: {
    padding: 24,
    flex: 1,
    justifyContent: "space-around"
  }
});
```

# Actividad guiada

## Parte 1

```
    },
    header: {
      fontSize: 36,
      marginBottom: 48
    },
    textInput: {
      height: 40,
      borderColor: "#000000",
      borderBottomWidth: 1,
      marginBottom: 36
    },
    btnContainer: {
      backgroundColor: "white",
      marginTop: 12
    }
  }
});

export default KeyboardAvoidingComponent;
```

# Actividad guiada

## Parte 1

### Consideraciones:

Debes:

1. Tener instalado React Native.
2. Haber instalado Android Studio.
3. Saber programar en JavaScript.
4. Realizar los ejercicios en orden.



# Receso

Duración: 10 minutos.

React Nivel Avanzado



# Actividad guiada

## Parte 2

**Duración: 75 minutos.**

1. Coloca un fondo de pantalla:

```
import React from "react";
import { ImageBackground, StyleSheet, Text, View } from "react-native";

const image = { uri: "https://reactjs.org/logo-og.png" };

const App = () => (
  <View style={styles.container}>
    <ImageBackground source={image} resizeMode="cover" style={styles.image}>
      <Text style={styles.text}>Inside</Text>
    </ImageBackground>
  </View>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  image: {
    flex: 1,
    justifyContent: "center"
  },
  text: {
    color: "white",
    fontSize: 42,
    lineHeight: 84,
    fontWeight: "bold",
    textAlign: "center",
    backgroundColor: "#000000c0"
  }
});
```

# Actividad guiada

## Parte 2

```
  }  
});  
  
export default App;
```

2. Muestra una barra de estado:

```
import React, { useState } from 'react';  
import { Button, Platform, SafeAreaView, StatusBar, StyleSheet, Text, View } from  
'react-native';  
  
const STYLES = ['default', 'dark-content', 'light-content'];  
const TRANSITIONS = ['fade', 'slide', 'none'];  
  
const App = () => {  
  const [hidden, setHidden] = useState(false);  
  const [statusBarStyle, setStatusBarStyle] = useState(STYLES[0]);  
  const [statusBarTransition, setStatusBarTransition] = useState(TRANSITIONS[0]);  
  
  const changeStatusBarVisibility = () => setHidden(!hidden);  
  
  const changeStatusBarStyle = () => {  
    const styleId = STYLES.indexOf(statusBarStyle) + 1;  
    if (styleId === STYLES.length) {  
      setStatusBarStyle(STYLES[0]);  
    } else {  
      setStatusBarStyle(STYLES[styleId]);  
    }  
  }  
};
```

## Actividad guiada Parte 2

```
const changeStatusBarTransition = () => {
  const transition = TRANSITIONS.indexOf(statusBarTransition) + 1;
  if (transition === TRANSITIONS.length) {
    setStatusBarTransition(TRANSITIONS[0]);
  } else {
    setStatusBarTransition(TRANSITIONS[transition]);
  }
};

return (
  <SafeAreaView style={styles.container}>
    <StatusBar
      animated={true}
      backgroundColor="#61dafb"
      barStyle={statusBarStyle}
      showHideTransition={statusBarTransition}
      hidden={hidden} />
    <Text style={styles.textStyle}>
      StatusBar Visibility:{'\n'}
      {hidden ? 'Hidden' : 'Visible'}
    </Text>
    <Text style={styles.textStyle}>
      StatusBar Style:{'\n'}
      {statusBarStyle}
    </Text>
    {Platform.OS === 'ios' ? (
      <Text style={styles.textStyle}>
        StatusBar Transition:{'\n'}
        {statusBarTransition}
      </Text>
    ) : null}
    <View style={styles.buttonsContainer}>
      <Button
        title="Toggle StatusBar"
        onPress={changeStatusBarVisibility} />
    </View>
  </SafeAreaView>
);
```



## Actividad guiada Parte 2

```
        <Button
          title="Change StatusBar Style"
          onPress={changeStatusBarStyle} />
        {Platform.OS === 'ios' ? (
          <Button
            title="Change StatusBar Transition"
            onPress={changeStatusBarTransition} />
          ) : null}
      </View>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    backgroundColor: '#ECF0F1'
  },
  buttonsContainer: {
    padding: 10
  },
  textStyle: {
    textAlign: 'center',
    marginBottom: 8
  }
});

export default App;
```

## Actividad guiada Parte 2

3. Agrega la funcionalidad de extracción para actualizar. Cuando ScrollView está en scrollY: 0, se debe deslizar hacia abajo y se activa un onRefresh:

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

const wait = (timeout) => {
  return new Promise(resolve => setTimeout(resolve, timeout));
}

const App = () => {
  const [refreshing, setRefreshing] = React.useState(false);

  const onRefresh = React.useCallback(() => {
    setRefreshing(true);
    wait(2000).then(() => setRefreshing(false));
  }, []);

  return (
    <SafeAreaView style={styles.container}>
      <ScrollView
        contentContainerStyle={styles.scrollView}
        refreshControl={
          <RefreshControl
            refreshing={refreshing}
            onRefresh={onRefresh}
          />
        >
        <Text>Pull down to see RefreshControl indicator</Text>
      </ScrollView>
    </SafeAreaView>
  );
}
```

## Actividad guiada Parte 2

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  scrollView: {
    flex: 1,
    backgroundColor: 'pink',
    alignItems: 'center',
    justifyContent: 'center',
  },
});

export default App;
```

## Actividad guiada Parte 2

### Consideraciones:

Debes:

1. Tener instalado React Native.
2. Haber instalado Android Studio.
3. Saber programar en JavaScript.
4. Realizar los ejercicios en orden.



# Cierre

React Nivel Avanzado

**Duración: 10 minutos.**

Las API son utilizadas para que una aplicación se integre con otras o con sistemas que ya existen, de tal forma que el programador no tenga la necesidad de codificarlos, lo cual, te ayudará a aumentar la velocidad en tu desarrollo, ya que estarás reutilizando código existente y no tendrás que hacerlo desde cero.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.