React Nivel Avanzado

Reactivos

Pregunta 1

Selecciona el o los componentes que se muestran en el siguiente código:

```
import React from 'react';
import { Text, View } from 'react-native';
const HelloWorldApp = () => {
return (
<View
style={{
flex: 1,
justifyContent: "center",
alignItems: "center"
}}>
<Text>Hello, world!</Text>
</View>
)
} export default HelloWorldApp;
```

Opciones:

a. View y text
b. View
c. Text
d. Style

Pregunta 2

Establece qué versión de JavaScript permite escribir código sin tener que analizar la sintaxis de este.

Opciones:

a. React Native
b. JSX
c. React JS
d. React

Pregunta 3

Elige los elementos necesarios para poder utilizar React Native para desarrollar una aplicación.

Opciones:

a. Chocolatey,node.js
b. Chocolatey,Android Studio
c. Chocolatey,node.js,Android Studio
d. Node.js, Android Studio

Pregunta 4

Decide qué mostrará el siguiente código al momento de ejecutarse:

```
import React from 'react';
import { Text, View } from 'react-native';
const HelloWorldApp = () => {
return (
<View
style={{
flex: 1,
justifyContent: "center",
alignItems: "center"
}}>
<Text>Hello, world!</Text>
</View>
)
} export default HelloWorldApp;
```

Opciones:

a. Una aplicación mostrando con una imagen “Hola mundo”.
b. Una aplicación mostrando con un "center"
c. F9.
d. Una aplicación mostrando en Texto “Hola mundo”.

Pregunta 5

Elige la etiqueta utilizada para modificar la vista de una pantalla en la aplicación.

Opciones:

a. <Style>
b. <TextInput>
c. <View>
d. <body></body>

Pregunta 6

Selecciona cuáles son consideradas las piezas de código que se pueden reutilizar tantas veces como sea necesario, esto con el objetivo de evitar repetir código.

Opciones:

a. Variable
b. Componente
c. Función
d. Bucles

Pregunta 7

Ciclo de vida del componente que ocurre al momento de eliminar el componente.

Opciones:

a. Limpieza
b. Actualización
c. Desmontaje
d. Montaje

Pregunta 8

Elige el elemento que permite agregar funcionalidades como manejo de estados, variables globales, etc.

Opciones:

a. Hook
b. Render
c. Debug
d. Return

Pregunta 9

Decide cuál es el método que permite regresar un elemento de la interfaz de usuario.

Opciones:

a. Debug
b. Return
c. Render
d. Hook

Pregunta 10

Elige qué es lo que hará el siguiente código:

```
function Section(){
const saludo = "Buen día."
return (
<View>
<Text>Hola mundo. {saludo}</Text>
</View>
);
}
```

Opciones:

a. Se define una variable que tiene “Buen día” y se mostrará cuando se solicite.
b. Se mostrará el mensaje “Hola Mundo”.
c. Se mostrará el mensaje “Hola Mundo”. Saludo.
d. Se define una variable que tiene “Buen día” y que se mostrará junto con “Hola Mundo”.

Pregunta 11

Selecciona qué componente de React Native sustituye el componente <div> de una página web.

Opciones:

a. <ScrollView>
b. <View>
c. <image>
d. <Text>

Pregunta 12

Establece cuál es el componente para mostrar texto en una app de iOS creada con React Native.

Opciones:

a. <TextView>
b. <Text>
c. <UITextView>
d. <p>

Pregunta 13

Elige qué realizará el siguiente código:

```
const Greeting = (props) => {
return (
<View style={styles.center}>
<Text>Hello {props.name}!</Text>
}
);</View>
```

Opciones:

a. En este caso se tendrá un componente Greeting, el cual mostrará el texto del saludo y recibirá un parámetro (propiedad) llamada name, cuyo valor se imprimirá dentro del componente <Text>
b. En este caso se tendrá un componente Greeting, el cual mostrará el texto del saludo, cuyo valor se imprimirá dentro del componente <Text>
c. En este caso se tendrá un componente Greeting, el cual mostrará el texto del saludo y recibirá un parámetro (propiedad) llamada name, cuyo valor se imprimirá dentro del componente <Image>
d. En este caso se tendrá un componente Greeting, el cual mostrará el texto del saludo y recibirá un parámetro (propiedad) llamada name, cuyo valor se imprimirá dentro del componente <View>

Pregunta 14

Decide qué hace style en el siguiente código:

```
const LotsOfGreetings = () => {
return (
<View style={[styles.center, {top: 50}]}>
<Greeting name='Rexxar' />
<Greeting name='Jaina' />
<Greeting name='Valeera' />
} );</View>
export default LotsOfGreetings;
```

Opciones:

a. Define el estilo de centrado a lo que no está dentro de View.
b. Define el estilo de centrado a lo que está en saludo.
c. Define el estilo de centrado a lo que está dentro de View.
d. Define el estilo de centrado a lo que está dentro de Greeting.

Pregunta 15

Elige el componente que muestra una imagen en una app para Android usando React Native.

Opciones:

a. <img>
b. <UIImageView>
c. <im>
d. <Image>

Pregunta 16

Selecciona qué hace el estilo bigBlue en el siguiente código:

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';
const LotsOfStyles = () => {
return (
<View style={styles.container}>
<Text style={styles.bigBlue}>Best</Text>
<Text style={[styles.bigBlue, styles.red]}>App</Text>
</View>
```

```
);
};
const styles = StyleSheet.create({
container: {
marginTop: 50,
},
bigBlue: {
color: 'blue',
fontWeight: 'bold',
fontSize: 30,
},
red: {
color: 'red',
},
});
export default LotsOfStyles;
```

Opciones:

a. Pone el texto de color azul y con un tamaño de letra de 30.
b. Pone el texto en negritas y con un tamaño de letra de 30.
c. Pone el texto de color azul, en negritas y con un tamaño de letra de 30.
d. Pone el texto de color azul y un tamaño no definido.

Pregunta 17

Establece la sintaxis para cambiar el ancho y alto en la pantalla de una aplicación.

Opciones:

a. Width y height
b. Width
c. Height
d. Width y background

Pregunta 18

Selecciona qué hace el siguiente código:

```
import React from 'react';
import { View } from 'react-native';
const FixedDimensionsBasics = () => {
return (
<View>
<View style={{
width: 50, height: 50, backgroundColor: 'powderblue'
}} />
<View style={{
width: 100, height: 100, backgroundColor: 'skyblue'
}} />
<View style={{
width: 150, height: 150, backgroundColor: 'steelblue'
}} />
</View>
);
};
export default FixedDimensionsBasics;
```

Opciones:

a. Te permite elegir entre tres colores para colocar en la aplicación.
b. Coloca 3 colores en la misma aplicación.
c. Coloca 2 colores en la misma aplicación.
d. Fija un color al fondo de pantalla.

Pregunta 19

Elige:

La palabra______, dentro del código, ayudará a definir cómo van a encajar los componentes, conforme al espacio que el dispositivo tenga.

Opciones:

a. const
b. style
c. padding
d. flex

Pregunta 20

Elige cómo definir la dirección en la cual se va a mostrar un componente.

Opciones:

a. Usando la palabra flexDirection.
b. Usando la palabra fDirection.
c. Usando la palabra Direction.
d. Usando la palabra fxDirection.

Pregunta 21

Selecciona qué es lo que hace el siguiente fragmento de código:

```
function Greeting ({ name }) {
return <div> Hello {name}! </div>
}
```

Opciones:

a. Se define un componente para mostrar un saludo, este componente recibirá el nombre de la persona a quien se le saludará mediante la propiedad name. Este mismo valor se usará para imprimir dentro del return del componente.
b. Se define un componente para mostrar un saludo, este componente recibirá el nombre de la persona a quien se le saludará mediante la propiedad name.
c. Se define un componente para mostrar un saludo. Este mismo valor se usará para imprimir dentro del return del componente.
d. Se define un componente para mostrar un saludo, este componente recibirá el nombre de la persona a quien se le saludará mediante la propiedad name. Este mismo valor se usará para imprimir fuera del return del componente.

Pregunta 22

Selecciona cuál es el método que se usa para decirle al botón qué hacer cuando sea presionado.

Opciones:

a. Click.
b. OnClick.
c. onclick.
d. label.

Pregunta 23

Selecciona qué se está creando mediante el siguiente código:

```
function TextField() {
const [inputValue, setInputValue] = useState('');
const handleInputChange = ({ target }) => {
setInputValue(target.value)
}
return (<input type="text" value={inputValue} onChange={handleInputChange}
/>)
}
```

Opciones:

a. Un componente.
b. Un estado.
c. Un componente con estado.
d. Un componente sin estado.

Pregunta 24

Elige qué archivo debes eliminar cuando creas un proyecto de React Native de tal forma que sea sustituido por un archivo .jsx

Opciones:

a. App.js
b. App.jason
c. Gradle
d. App.tsx

Pregunta 25

Elige qué debes hacer para enviar información a componentes que se encuentran en niveles más abajo en el árbol virtual DOM.

Opciones:

a. Crear un contexto.
b. Eliminar los componentes hasta llegar al deseado.
c. Crear una función.
d. Crear una variable que almacene la información de los componentes.

Pregunta 26

Selecciona cuál es el protocolo que se usa para mandar una solicitud de datos a una API de REST.

Opciones:

a. Https
b. Http
c. JSON

d. XML

Pregunta 27

Selecciona qué representa el siguiente fragmento de código:

URL: https://api.coindesk.com/v1/bpi/currentprice.json
Método: GET

Opciones:

a. Uso de API Rest.
b. Conexión a una URL.
c. Uso de API Fetch.
d. Descarga de información de una URL.

Pregunta 28

Selecciona el formato preferido para leer los mensajes al momento de hacer una petición a una API.

Opciones:

a. JavaScript
b. C++
c. C#
d. JSON

Pregunta 29

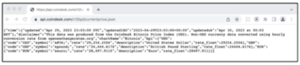
Elige cuál es un lenguaje de consulta para API que brinda una descripción completa y comprensible de los datos en una API.

Opciones:

a. GraphQL.
b. MySQL.
c. Método GET.
d. Método POST.

Pregunta 30

Elige qué es lo que muestra la siguiente imagen:


https://api.coindesk.com/v1/b
api.coindesk.com/v1/bpi/currentprice.json

```
{"time":{"updated":"Apr 29, 2023 23:03:00 UTC","updatedISO":"2023-04-29T23:03:00+00:00","updateduk":"Apr 30, 2023 at 00:03
BST"},"disclaimer":"This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly
conversion rate from openexchangerates.org","chartName":"Bitcoin","bpi":{"USD":
{"code":"USD","symbol":"&#36;","rate":"29,254.2504","description":"United States Dollar","rate_float":29254.2504},"GBP":
{"code":"GBP","symbol":"&pound;","rate":"24,444.6176","description":"British Pound Sterling","rate_float":24444.6176},"EUR":
{"code":"EUR","symbol":"&euro;","rate":"28,497.9110","description":"Euro","rate_float":28497.911}}}
```

Opciones:

a. El resultado del método POST.
b. El resultado del método GET.
c. El resultado del método OBTAIN.
d. El resultado del método CONSULT.

Pregunta 31

Selecciona cuáles son los errores más comunes en un lenguaje de programación:

Opciones:

a. Errores sintácticos.
b. Errores de lógica.
c. Errores de compilación.
d. Errores de ejecución.

Pregunta 32

Selecciona qué errores pueden provocar lo siguiente:

- Guardar datos en otra variable.
- Realizar operaciones incorrectamente a pesar de que el compilador toma como correcta la sintaxis.
- Devolver valores incorrectos a pesar de enviarse los correctos.
- Ocasionar que la aplicación falle al no haber manejo de errores para determinadas situaciones.
- Ocasionar que la aplicación se cicle o se congele al realizar demasiadas peticiones innecesarias.

Opciones:

a. Errores de compilación.
b. Errores de lógica.
c. Errores de ejecución.
d. Errores sintácticos.

Pregunta 33

Selecciona qué tipo de pruebas se utilizan para determinar si una aplicación requiere conectarse a una red o no.

Opciones:

a. Usabilidad.
b. Funcionalidad.
c. Conectividad.
d. Accesibilidad.

Pregunta 34

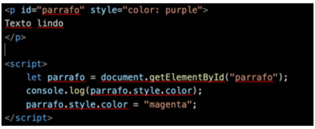
Elige las pruebas que se usan para determinar si el software cumple con lo solicitado.

Opciones:

a. Funcionalidad.
b. Conectividad.
c. Accesibilidad.
d. Usabilidad.

Pregunta 35

Elige cuáles son los errores que encuentras en configuraciones no realizadas en el entorno de desarrollo.



```
<p id="parrafo" style="color: purple">
Texto lindo
</p>

<script>
    let parrafo = document.getElementById("parrafo");
    console.log(parrafo.style.color);
    parrafo.style.color = "magenta";
</script>
```

Opciones:

a. Configuración del entorno.
b. Errores de lógica.
c. Errores de compilación.
d. Errores de ejecución.

Pregunta 36

Selecciona qué debes utilizar para detectar cuando el usuario haga clic en el botón hacia atrás en tu aplicación.

Opciones:

a. Onclick.
b. BackHandler.
c. ClickBack.
d. BHnadler.

Pregunta 37

Selecciona qué elemento no puede faltar en tu archivo App.jsx para que se muestre correctamente la aplicación.

Opciones:

a. funciones.
b. imports.
c. Export default NombreArchivo
d. componentes.

Pregunta 38

Selecciona qué mostrará el siguiente código:

Alert.alert("Hold on!", "Are you sure you want to go back?")
Opciones:

a. Un texto.
b. Regresará a la página anterior.
c. Dejará de ejecutarse el código.
d. Una alerta.

Pregunta 39

Elige el evento que permite controlar la respuesta que dará tu aplicación ante alguna acción por parte del usuario.

Opciones:

a. Contexto.
b. Listener.
c. Evento del teclado.
d. Evento del mouse.

Pregunta 40

Elige cuál es la línea de código que te permite importar bibliotecas en React Native.

Opciones:

a. import { Text, View, StyleSheet, BackHandler, Alert } from "react-native";
b. const backHandler = BackHandler.addEventListener(
"hardwareBackPress",
backAction)
c. Alert.alert("Hold on!", "Are you sure you want to go back?")
d. import React, { useEffect } from "react";