

React Nivel Avanzado

Sesión sincrónica 3

Módulos nativos



Bienvenida y actividad de bienestar

React nivel avanzado

Duración: 10 minutos.

Nombre de la práctica	Fomentando la atención plena.
Descripción de la práctica	Se realizarán en tres ocasiones, ejercicios de meditación para fortalecer la atención plena en las actividades diarias.
Palabras clave	Atención plena, fortalezas de carácter, autorregulación.
Instrucciones para el aprendedor	<p>La meditación es una herramienta que ayuda a mejorar el desempeño de las personas que la practican a menudo, ya que promueve el desarrollo de la atención plena en una sola actividad. Para fomentarla y lograr, cada vez más, estar en una zona de concentración mientras se realizan las actividades cotidianas, se pueden llevar a cabo los siguientes ejercicios:</p> <p>Designar en algún momento del día cinco minutos para realizar la meditación, sentarse en un lugar cómodo, donde no haya distracciones:</p> <ol style="list-style-type: none">1. Hacer tres respiraciones profundas: inhalar y exhalar por la nariz.2. Comienza a hacer un repaso del día, de lo que más se recuerde, por ejemplo: al levantarse ¿qué hiciste?, ¿desayunaste?, ¿te bañaste?, ¿diste los buenos días?, etcétera. Si almorzaste ¿qué fue lo que comiste?, ¿te gustó?, ¿tomaste tu alimento despacio o de prisa? Si estabas apurado(a) ¿cuál fue la causa?3. Seguir meditando, poniendo atención en esos detalles: ¿te molestase con alguien?, ¿por qué?, ¿qué fue lo que pasó?, ¿crees que hubiera sido posible reaccionar de otra forma, con más tranquilidad? <p>Con este ejercicio se toma consciencia de que las personas actúan o hacen sus actividades de forma automática. A veces, si es posible estar conscientes y presentes, modificar la actitud para enfrentar situaciones que no afecten demasiado.</p>
Fuente	Eby, D. (s.f.). <i>Creativity and Flow Psychology</i> . Recuperado de http://talentdevelop.com/articles/Page8.html

Actividad guiada

Parte 1

Duración: 75 minutos.

Ejercicio 1:

A continuación, se debe crear una lista de compras de productos del súper mercado, en esta lista se mostrarán los nombres de los productos y la cantidad de productos necesarios.

Crear un proyecto vacío usando el comando ***npx react-native init listaSuper***

1. Abrir la carpeta **listaSuper** con Visual Studio Code.
2. Renombrar el archivo App.tsx por App.jsx
3. Reemplazar el contenido del archivo App.jsx por el siguiente:

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

function App() {
  return (
    <SafeAreaView>
      <Text>Lista de súper</Text>
      <ScrollView>
        <Text>Tomate</Text>
        <Text>Cebolla</Text>
        <Text>Pollo</Text>
        <Text>Café</Text>
      </ScrollView>
    </SafeAreaView>
  )
}
export default App;
```

Actividad guiada

Parte 1

4. Probar que funcione el código ejecutando el comando: **npm run start**
5. Se abrirá el menú de Metro, seleccionar la plataforma con la que se hará la prueba (deberá estar previamente configurado, como se explica en el Prework 2).

```
Welcome to Metro v0.73.9
Fast - Scalable - Integrated

r - reload the app
d - open developer menu
i - run on iOS
a - run on Android
```

6. Presionar la letra correspondiente a la plataforma que se tenga configurada y con la que se realizará la prueba, por ejemplo, para probar la app en iOS, presionar la letra **i**. Para probar la aplicación en Android, presionar la letra **A**.
7. Aparecerá el siguiente mensaje:

```
info Opening the app on iOS...
info Found Xcode workspace "listaSuper.xcworkspace"
info No booted devices or simulators found. Launching first available
info Launching iPhone SE (3rd generation) (iOS 16.4)
info Building (using "xcodebuild -workspace listaSuper.xcworkspace
destination id=795F88B2-25D4-4D0F-BD08-96C83F4EF8AC")
□
```

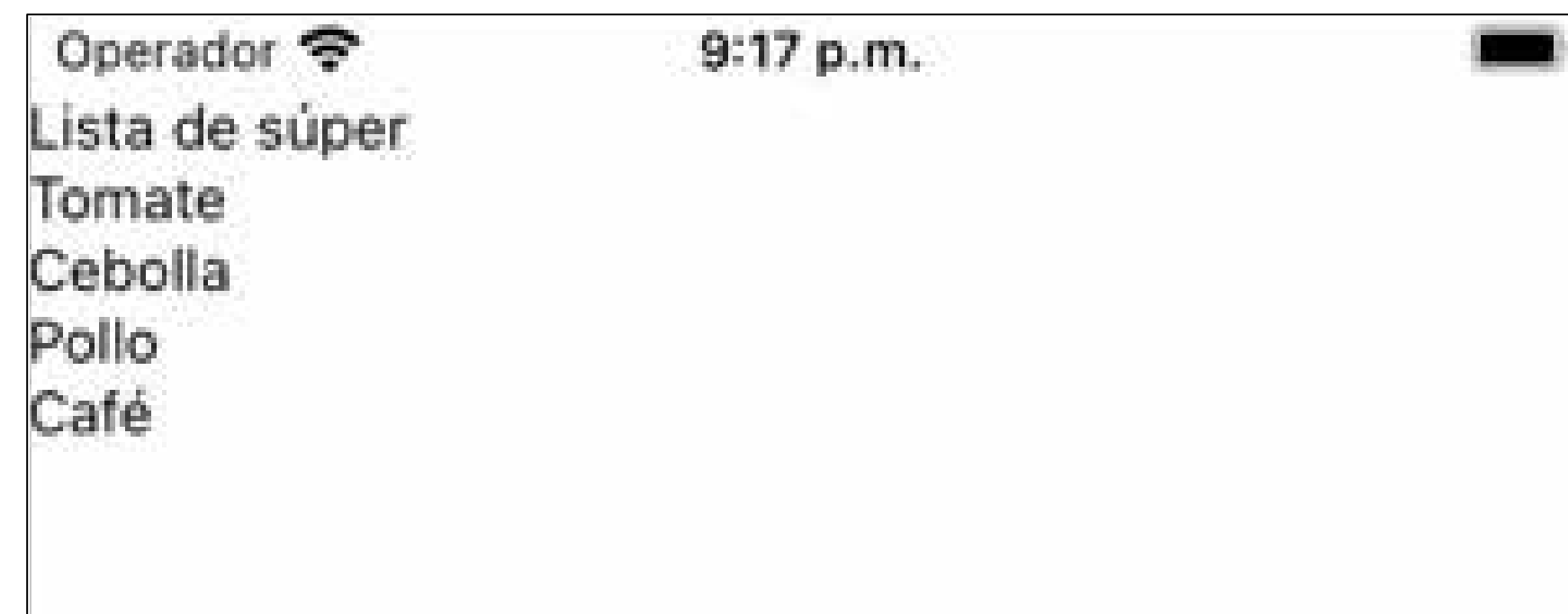
Actividad guiada

Parte 1

8. Después de un tiempo, aparecerá el siguiente mensaje, lo cual indica que ya se puede probar la aplicación en el dispositivo configurado para realizar las pruebas:

```
success Successfully launched the app on the simulator
BUNDLE ./index.js
Running "listaSuper" with {"rootTag":1,"initialProps":{}}
```

9. Una vez que la app se haya compilado y ejecutado correctamente, se verá algo como la siguiente imagen:



Actividad guiada

Parte 1

10. Agregar estilos a la pantalla, de la siguiente manera:

```
const styles = StyleSheet.create({
  title: {
    marginTop: 20,
    textAlign: 'center',
    fontSize: 24,
    fontWeight: '600',
  },
  itemsContainer: {
    marginTop: 10,
    paddingHorizontal: 24,
  },
  item: {
    fontSize: 18,
    marginVertical: 4
  },
});
```

Actividad guiada

Parte 1

11. Asignar los estilos al código existente, el archivo debe quedar de la siguiente manera:

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

function App() {
  return (
    <SafeAreaView>
      <Text style={styles.title}>Lista de súper</Text>
      <ScrollView style={styles.itemsContainer}>
        <Text style={styles.item}>Tomate</Text>
        <Text style={styles.item}>Cebolla</Text>
        <Text style={styles.item}>Pollo</Text>
        <Text style={styles.item}>Café</Text>
      </ScrollView>
    </SafeAreaView>
  )
}

const styles = StyleSheet.create({
  title: {
    marginTop: 20,
    textAlign: 'center',
    fontSize: 24,
    fontWeight: '600',
  },
  itemsContainer: {
    marginTop: 10,
    paddingHorizontal: 24,
  },
  item: {
    fontSize: 18,
    marginVertical: 4
  },
});

export default App;
```

Actividad guiada

Parte 1

12. Este es el resultado:

Lista de súper

Tomate
Cebolla
Pollo
Café

Consideraciones:

Debe:

1. Haber instalado React Native.
2. Tener 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.

Ya se tiene la aplicación con cuatro artículos agregados, pero ¿Qué sucede si se pretende agregar más artículos a la lista?

A continuación, se agregarán dos elementos más a la lista.

1. En el archivo App.jsx, ir a la línea 12 (donde se encuentra el texto “Café”).
2. Agregar un artículo con el texto “**Galletas**” debajo de “**Café**”.
3. Debajo de “**Galletas**”, agregar un artículo con el texto “**Pan**”.
4. El código debería verse como el siguiente:

```
function App() {  
  return (  
    <SafeAreaView>  
      <Text style={styles.title}>Lista de súper</Text>  
      <ScrollView style={styles.itemsContainer}>  
        <Text style={styles.item}>Tomate</Text>  
        <Text style={styles.item}>Cebolla</Text>  
        <Text style={styles.item}>Pollo</Text>  
        <Text style={styles.item}>Café</Text>  
        <Text style={styles.item}>Galletas</Text>  
        <Text style={styles.item}>Pan</Text>  
      </ScrollView>  
    </SafeAreaView>  
  )  
}
```

Actividad guiada

Parte 2

Al hacer el trabajo de esta forma, se notará como al ir agregando más elementos a la lista, más códigos será necesario escribir. Una forma de solucionar esto es crear un componente donde se incluyan los estilos y reciba el texto del artículo.

Esto se haría de la siguiente manera:

1. Dentro del proyecto, crear una carpeta con el nombre Components.
2. Dentro de la carpeta Components, crear el archivo **Item.jsx**
3. Agregar el siguiente código en **Item.jsx**

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

function Item({ name }) {
  const itemStyles = StyleSheet.create({
    fontSize: 18,
    marginVertical: 4
  });

  return (
    <Text style={itemStyles}>{name}</Text>
  )
}

export default Item;
```

Actividad guiada

Parte 2

4. En el archivo App.jsx, importar el componente Ítem de la siguiente manera:

```
import Item from './Components/Item';
```

5. Reemplazar el código creado anteriormente.

```
<Text style={styles.item}>Tomate</Text>
```

6. Por el código que llama al nuestro componente.

```
<Item name="Tomate" />
```

7. Realizar esto no debería afectar, visualmente, el proyecto hasta ahora.

Actividad guiada

Parte 2

Aún es posible reducir el código de la siguiente manera:

1. En la línea 6 del archivo **App.jsx** declarar una variable con el nombre **shoppingItems**, esta variable será la lista de los productos.

```
const shoppingItems = [  
  "Tomate",  
  "Cebolla",  
  "Pollo",  
  "Café",  
  "Galletas",  
  "Pan",  
];
```

2. En el return, reemplazar los elementos individuales y recorrer el array de elementos, deberá quedar de la siguiente manera:

```
return (  
  <SafeAreaView>  
    <Text style={styles.title}>Lista de súper</Text>  
    <ScrollView style={styles.itemsContainer}>  
      {shoppingItems.map(  
        (item, key) => <Item key={key} name={item} />  
      )}  
    </ScrollView>  
  </SafeAreaView>  
)
```


Actividad guiada

Parte 2

Así, solo habrá que agregar elementos en el array, cada vez que se quiera aumentar la cantidad de elementos mostrados.

El resultado final (se agregan más artículos) sería algo como esto:

Lista de súper
Tomate
Cebolla
Pollo
Café
Galletas
Pan
Ajo
Huevo
Jugo
Chocolates
Detergente

Consideraciones:

Debe:

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.

El uso de componentes permite reutilizar el código desarrollado, lo que ayuda a ser más productivos, y, si se requieren modificaciones en un componente, se pueden adaptar por cada específico o actualizar varios elementos de la interfaz, esto es práctico cuando se trabajan con interfaces de usuario o bloques de construcción.

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.