Marcar elementos

Se debe recibir el item de tipo task para identificarla y marcala dentro de la lista

Se debe definir el indice por el titulo

import React,{useState,useEffect} from "react";

import { View,Text,TextInput, TouchableOpacity,FlatList, \_View} from 'react-native'

import styles from "./Style";

import RenderItem from "./RenderItem";

const tasks=[

]

export interface Task {

  title:string,

  done:boolean,

  date:Date

}

export default function App(){

const [text,setText]=useState('')

const[tasks,setTasks]=useState<Task[]>([])

  const addTask=()=>{

    const tmp=[...tasks]

    const newTaks={

      title:text,

      done:false,

      date:new Date()

    }

    tmp.push(newTaks)

    setTasks(tmp)

    setText('')

  }

  const markDone=(task:Task)=>{

    const tmp=[...tasks]

    const index=tmp.findIndex(el=>el.title===task.title)//busca el indice

    const todo=tmp[index]//se modifica un elemento del array

    todo.done=!todo.done//cambia el valor del done

    setTasks(tmp)

  }

  const deleteFunction=()=>{

  }

  return(

    <View style={styles.container}>

       <Text style={styles.title}>

          Mis tareas por hacer

       </Text>

       <View style={styles.inputconteiner}>

        <TextInput placeholder="Agregar una nueva tarea"

        style={styles.textinput}

        value={text}

        onChangeText={(t:string)=>setText(t)}

        />

        <TouchableOpacity

        onPress={addTask}

        style={styles.addButon}>

          <Text style={styles.whitetext}>

            Agregar

          </Text>

        </TouchableOpacity>

       </View>

       <View style={styles.scrollContainer}>

        <FlatList

          renderItem={({item})=>(

          <RenderItem

          item={item}

          markDone={markDone}

          deleteFuntion={deleteFunction}

          />

          )}

         data={tasks}

        />

       </View>

    </View>

  )

}

El error es debido a que debemos pasar los atributos a la función markdone en el renderItem

import React from "react";

import { View,Text,TouchableOpacity } from "react-native";

import styles from "./Style";

import { Task } from "./App";

interface ItemProps{

    item: Task;

    markDone:(task:Task)=>void;

    deleteFuntion:()=>void;

}

export default function RenderItem({item, markDone,deleteFuntion}:ItemProps){

    return (

    <View style={styles.itemContainer}>

      <TouchableOpacity onPress={()=>markDone(item)}>

        <Text style={item.done ? styles.textDone: styles.text}>{item.title}</Text>

        <Text style={styles.text}>{item.date.toDateString()}</Text>

      </TouchableOpacity>

      {

        item.done &&

      (<TouchableOpacity style={styles.removeButtom} onPress={deleteFuntion

          <Text style={styles.whitetext}>

              Eliminar

            </Text>

        </TouchableOpacity>)

      }

    </View>

    )

  }

Funcion de eliminar

Igual que la otra recibe las tareas y

import React,{useState,useEffect} from "react";

import { View,Text,TextInput, TouchableOpacity,FlatList, \_View} from 'react-native'

import styles from "./Style";

import RenderItem from "./RenderItem";

const tasks=[

]

export interface Task {

  title:string,

  done:boolean,

  date:Date

}

export default function App(){

const [text,setText]=useState('')

const[tasks,setTasks]=useState<Task[]>([])

  const addTask=()=>{

    const tmp=[...tasks]

    const newTaks={

      title:text,

      done:false,

      date:new Date()

    }

    tmp.push(newTaks)

    setTasks(tmp)

    setText('')

  }

  const markDone=(task:Task)=>{

    const tmp=[...tasks]

    const index=tmp.findIndex(el=>el.title===task.title)//busca el indice

    const todo=tmp[index]//se modifica un elemento del array

    todo.done=!todo.done//cambia el valor del done

    setTasks(tmp)

  }

  const deleteFunction=(task:Task)=>{

    const tmp=[...tasks]

    const index=tmp.findIndex(el=>el.title===task.title)

    tmp.splice(index,1)//splice permite quitar un elemento dado el index y la cantidad

    setTasks(tmp)

    storeData(tmp)

  }

  return(

    <View style={styles.container}>

       <Text style={styles.title}>

          Mis tareas por hacer

       </Text>

       <View style={styles.inputconteiner}>

        <TextInput placeholder="Agregar una nueva tarea"

        style={styles.textinput}

        value={text}

        onChangeText={(t:string)=>setText(t)}

        />

        <TouchableOpacity

        onPress={addTask}

        style={styles.addButon}>

          <Text style={styles.whitetext}>

            Agregar

          </Text>

        </TouchableOpacity>

       </View>

       <View style={styles.scrollContainer}>

        <FlatList

          renderItem={({item})=>(

          <RenderItem

          item={item}

          markDone={markDone}

          deleteFuntion={deleteFunction}

          />

          )}

         data={tasks}

        />

       </View>

    </View>

  )

}

En el renderItem tenemos el mismo problema que resolvemos de la misma forma

import React from "react";

import { View,Text,TouchableOpacity } from "react-native";

import styles from "./Style";

import { Task } from "./App";

interface ItemProps{

    item: Task;

    markDone:(task:Task)=>void;

    deleteFuntion:(task:Task)=>void;

}

export default function RenderItem({item, markDone,deleteFuntion}:ItemProps){

    return (

    <View style={styles.itemContainer}>

      <TouchableOpacity onPress={()=>markDone(item)}>

        <Text style={item.done ? styles.textDone: styles.text}>{item.title}</Text>

        <Text style={styles.text}>{item.date.toDateString()}</Text>

      </TouchableOpacity>

      {

        item.done &&

      (<TouchableOpacity style={styles.removeButtom} onPress={()=>deleteFuntion(item)}>

          <Text style={styles.whitetext}>

              Eliminar

            </Text>

        </TouchableOpacity>)

      }

    </View>

    )

  }