Para capturar el texto necesitamos capturar el estado para lo cual creamos una constante que se iguale al useState en la App:

import React,{useState} from "react";

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

import styles from "./Style";

import RenderItem from "./RenderItem";

const tasks=[

  {

    title:"Alimentar perro",

    done:false,

    date:new Date()

  },

  {

    title:"Salir a correr",

    done:true,

    date:new Date()

  }

]

export interface Task {

  title:string,

  done:boolean,

  date:Date

}

export default function App(){

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

  const markDone=()=>{console.log("HOLA")}

  const deleteFunction=()=>{console.log("HOLA2")}

  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}/>

        <TouchableOpacity 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>

  )

}

Pero de esta forma no puedo es posible escribir debido a que el useState esta definido como vacio

Para lo cual necesitamos utilizar un listerner del textinput donde t es la variable tipo string que almacena mi entrada

import React,{useState} from "react";

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

import styles from "./Style";

import RenderItem from "./RenderItem";

const tasks=[

  {

    title:"Alimentar perro",

    done:false,

    date:new Date()

  },

  {

    title:"Salir a correr",

    done:true,

    date:new Date()

  }

]

export interface Task {

  title:string,

  done:boolean,

  date:Date

}

export default function App(){

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

  const markDone=()=>{console.log("HOLA")}

  const deleteFunction=()=>{console.log("HOLA2")}

  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 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>

  )

}

Ahora comenzaremos a enviar las tareas a la data por lo cual eliminamos la data

En este caso el useState se vuelve un arreglo de Task asea se configura con la interface lo cual necesita Renderitem para funcionar

import React,{useState} 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 newTaks={

      title:text,

      done:false,

      date:new Date()

    }

  }

  const markDone=()=>{console.log("HOLA")}

  const deleteFunction=()=>{console.log("HOLA2")}

  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>

  )

}

Ahora por buenas practicas necesitamos crear una variable temporal que almacene los datos

import React,{useState} 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)

  }

  const markDone=()=>{console.log("HOLA")}

  const deleteFunction=()=>{console.log("HOLA2")}

  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 problema será que no borra el texto del input

import React,{useState} 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=()=>{console.log("HOLA")}

  const deleteFunction=()=>{console.log("HOLA2")}

  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>

  )

}

NOTA: En este momento nuestra aplicación si se reinicia pierde las tareas por que aun no están quedando almacenadas