MANUAL TECNICO

FIT

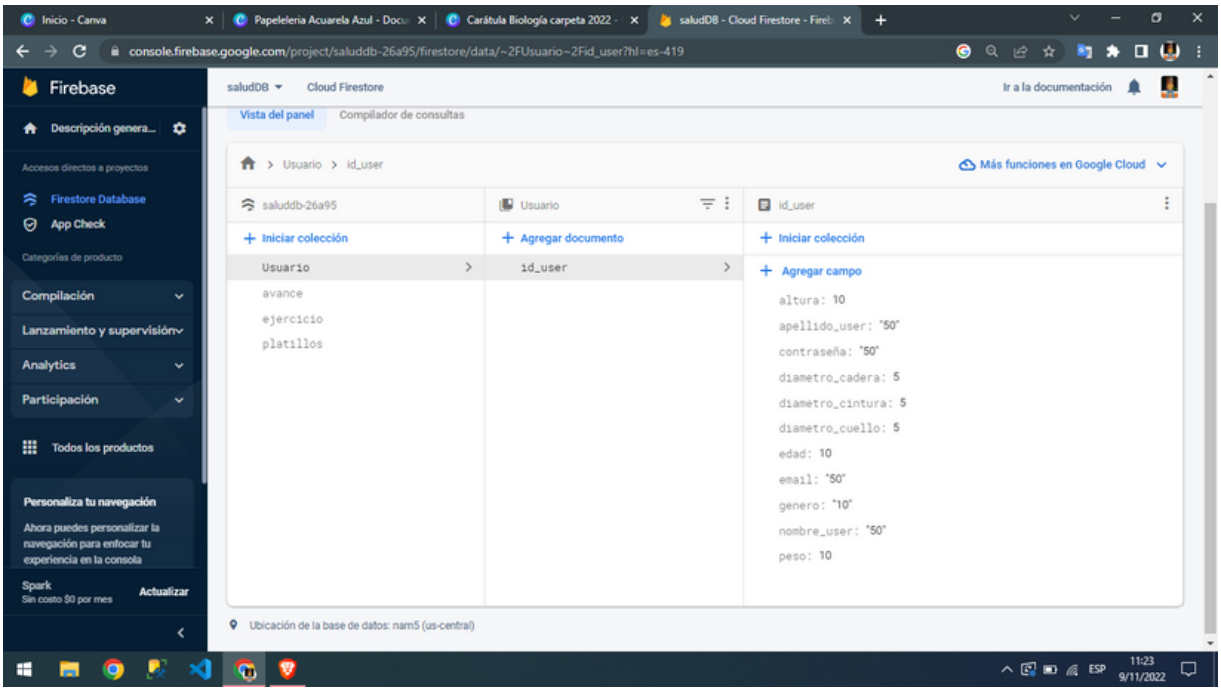
DISEÑO Y PROGRAMACIÓN DE SOFTWARE MULTIPLATAFORMA

Ing. Mario Alexander

Proyecto catedra

Alumnos	Carnet
Darío Jeffry Larreynaga Contreras	LC132111
José Ángel Rosales Portillo	RP142493
Mario Ernesto Medina Martínez	MM201254
Nelson Ernesto Muñoz Barahona	MB192012

FIt cuenta con una base de datos alojada en firebase
para poder llevar un mejor control de la informacion de los usuarios



desarrollo de codigo para el login y llamado a la base de datos para validar el usuario

```
import React from "react";
import { Text, View, TextInput, Image, StyleSheet, ScrollView, SafeAreaView, Alert } from "react-native";
import { Registrarse, Btn2 , Regresar} from "../componentes/botones";
import { useState } from "react";

const LoginScreen = ({ navigation }) => {

  const [state, setState] = useState({
    nombre:'',
    apellido:'',
    email:'',
    contraseña:'',
    edad:'',
    altura:'',
    peso:'',
  })

  const CapturarDatos = (campo, value)=>{
    setState({...state, [campo]: value});
  };

  const LoginUsuario= () =>{
    if(state.email==="" || state.contraseña==="")
    {
      Alert.alert("Campos vacios")
    }
    else{
      //await firebase.collection('usuario').add({
        // nombre: state.nombre,
        // apellido: state.apellido,
        // edad: state.edad,
        // altura: state.altura,
         //peso: state.peso,
        // email: state.email,
       //  contraseña: state.contraseña,
      // })
      Alert.alert("Inicio de sesion exitoso")

    }
  }



  return (


    <ScrollView style={styles.fondo}>


      <View style={styles.container}>
      <Image style={styles.img} source={require('../img/logo.jpeg')}/>
      </View>
      <Text></Text>
      <View style={styles.container}>
        <TextInput keyboardType="email-address" placeholder="E-mail" style={styles.txtbox} onChangeText={(value)=>CapturarDatos("email", value)}></TextInput>
        <Text></Text>
        <TextInput  placeholder="Contraseña" style={styles.txtbox} onChangeText={(value)=>CapturarDatos("contraseña", value)}></TextInput>
      </View>
      <Text></Text><Text></Text>
      <SafeAreaView style={styles.container}>
        <Registrarse text ="Iniciar Sesion" onPress={()=>AgregarUsuario()} />
        <Text></Text>
      </SafeAreaView>
      <Text></Text><Text></Text>
      <View style={styles.container}>
        <View style={{flexDirection:'row'}}>
          <Image style={styles.icon} source={require('../img/gg.png')}></Image>
          <SafeAreaView style={styles.container}>
        <Btn2 text ="Iniciar con Google" />
      </SafeAreaView>
        </View>
      </View>
      <View>
        <Text></Text>
        <Text></Text>
      </View>

    </ScrollView>
  );
}

const styles= StyleSheet.create({
  fondo:{
    backgroundColor: 'white',
  },
  container: {
    justifyContent: 'center',
    alignItems: 'center',
  },
  img:
  {
    width: 200,
    height: 175,
  },
  icon:{
    width:50,
    height:50
  },
  txtbox:{
    backgroundColor: 'white',
    borderBottomColor: '#000000',
    borderBottomWidth:1,
    width:250,
    textAlign:'center'
  },
  txtbox2:{
    backgroundColor: 'white',
    borderBottomColor: '#000000',
    borderBottomWidth:1,
    width:75,
    textAlign:'center'
  },
  btn:{
    fontSize:30,
    backgroundColor:"#00ab4c",
    borderRadius:10,
  }
});

export default LoginScreen;
```

desarrollo de codigo para la pantalla de registro donde el usuario ingresara sus datos y estos seran enviados a nuestra base de datos en firebase

```python
import React from "react";
import { Text, View, TextInput, Image, StyleSheet, ScrollView, SafeAreaView, Alert } from "react-native";
import { Registrarse, Btn2 , Regresar} from "../componentes/botones";
import { useState } from "react";

const RegisterScreen = ({ navigation }) => {

  const [state, setState] = useState({
    nombre:'',
    apellido:'',
    email:'',
    contraseña:'',
    edad:'',
    altura:'',
    peso:'',
  })

  const CapturarDatos = (campo, value)=>{
    setState({...state, [campo]: value});
  };

  const AgregarUsuario= () =>{
    if(state.nombre ===""  || state.apellido ===""  || state.email===""  || state.contraseña===""  || state.edad===""  || state.altura===""  || state.peso===""  )
    {
      Alert.alert("Campos vacios")
    }
    else{
      //await firebase.collection('usuario').add({
       // nombre: state.nombre,
       // apellido: state.apellido,
       // edad: state.edad,
       // altura: state.altura,
        //peso: state.peso,
       // email: state.email,
      //  contraseña: state.contraseña,
     // })
      Alert.alert("Los datos se han guardado")

    }
  }
```

```python
  return (


    <ScrollView style={styles.fondo}>


      <View style={styles.container}>
      <Image style={styles.img} source={require('../img/logo.jpeg')}/>
      </View>
      <Text></Text>
      <View style={styles.container}>
        <TextInput placeholder="Ingresa tu nombre" style={styles.txtbox} onChangeText={(value)=>CapturarDatos("nombre", value)}></TextInput>
        <Text></Text>
        <TextInput placeholder="Ingresa tu Apellido" style={styles.txtbox} onChangeText={(value)=>CapturarDatos("apellido", value)}></TextInput>
        <Text></Text>
        <TextInput keyboardType="email-address" placeholder="E-mail" style={styles.txtbox} onChangeText={(value)=>CapturarDatos("email", value)}></TextInput>
        <Text></Text>
        <TextInput  placeholder="Contraseña" style={styles.txtbox} onChangeText={(value)=>CapturarDatos("contraseña", value)}></TextInput>
      </View>
      <View style={styles.container}>
        <View style={{flexDirection:'row'}}>
          <TextInput keyboardType='numeric'   placeholder="Edad" style={styles.txtbox2} onChangeText={(value)=>CapturarDatos("edad", value)}></TextInput><TextInput></TextInput>
          <TextInput keyboardType="numeric" placeholder="Peso(kg)" style={styles.txtbox2} onChangeText={(value)=>CapturarDatos("peso", value)}></TextInput>
        </View>
      </View>
      <View style={styles.container}>
      <TextInput keyboardType='numeric' placeholder="Altura(m)" style={styles.txtbox2} onChangeText={(value)=>CapturarDatos("altura", value)}></TextInput>
      </View>
      <Text></Text><Text></Text>
      <SafeAreaView style={styles.container}>
        <Registrarse text ="Registrarse" onPress={()=>AgregarUsuario()} />
        <Text></Text>
        <Regresar text='Regresar'></Regresar>
      </SafeAreaView>
      <Text></Text><Text></Text>
      <View style={styles.container}>
        <View style={{flexDirection:'row'}}>
          <Image style={styles.icon} source={require('../img/gg.png')}></Image>
          <SafeAreaView style={styles.container}>
        <Btn2 text ="Continuar con Google" />
      </SafeAreaView>
        </View>
      </View>
      <View>
        <Text></Text>
        <Text></Text>
      </View>

    </ScrollView>
  );
}
```

```python
const styles= StyleSheet.create({
  fondo:{
    backgroundColor: 'white',
  },
  container: {
    justifyContent: 'center',
    alignItems: 'center',
  },
  img:
  {
    width: 200,
    height: 175,
  },
  icon:{
    width:50,
    height:50
  },
  txtbox:{
    backgroundColor: 'white',
    borderBottomColor: '#000000',
    borderBottomWidth:1,
    width:250,
    textAlign:'center'
  },
  txtbox2:{
    backgroundColor: 'white',
    borderBottomColor: '#000000',
    borderBottomWidth:1,
    width:75,
    textAlign:'center'
  },
  btn:{
    fontSize:30,
    backgroundColor:"#00ab4c",
    borderRadius:10,
  }
});

export default RegisterScreen;
```

desarrollo del splash de fit, este a sido la logica aplicada al splash de fit que se muestra cada vez que la aplicacion es ejecutada se muestra como primera pantalla

```
import React, {useRef, useState, useEffect} from 'react';
import { View, Text, StyleSheet, TouchableOpacity } from 'react-native';
import LottieView from 'lottie-react-native';

const SplashScreen = props => {
  const [authLoaded, setAuthLoaded] = useState(false);
  const [animationLoaded, setAnimationLoaded] = useState(false);

  const ref = useRef(null);

  useEffect(() => {
    setTimeout(() => {
      setAuthLoaded(true);
    }, 5000);
  }, []);

  const onAnimationFinish = () => {
    setAnimationLoaded(true);
  };

  useEffect(() => {
    if (authLoaded && animationLoaded) {
      props.navigation.replace('Registro');
    }
  }, [authLoaded, animationLoaded, props.navigation]);

  return (
    <View style={styles.root}>
      <LottieView
        ref={animation => {
          ref.current = animation;
        }}
        style={styles.lottieView}
        source={require('../assets/water-splash.json')}
        autoPlay
        loop={false}
        onAnimationFinish={onAnimationFinish}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  root: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  lottieView: {
    width: '100%',
  },
});

export default SplashScreen;
```

desarrolo y codificacion del emulador de la silueta de una persona, para que sea el guia del proceso del usuario.

```python
import React from "react";
import Svg, {
  Circle,
  Ellipse,
  G,
  TSpan,
  TextPath,
  Path,
  Polygon,
  PolyLine,
  Line,
  Rect,
  Use,
  Image,
  Symbol,
  Defs,
  LinearGradient,
  RadialGradient,
  Stop,
  ClipPath,
  Pattern,
  Mask,
  SafeAreaView,
} from "react-native-svg";
import { View, Text, StyleSheet, TouchableOpacity } from "react-native";
const ProfileScreen = ({ navigation }) => {
  var inc = 0,
    peso = 81,
    altura = 167,
    nivel = 5,
    exp = 0,
    imb = 0,
    edad = 27,
    grasa = 30,
    definicion = 0,
    IMC = peso / ((altura / 100) * (altura / 100));

  imb = (10 * peso + 6.25 * altura - 5 * edad + 5).toFixed(2);
  grasa = grasa - definicion;
  if (nivel == 0) {
    grasa = (1.2 * IMC + 0.23 * edad - 10.8 * 1 * inc * inc).toFixed(2);
    inc = (peso - peso * (grasa / 100)) * 0.0075;
  }
  if (nivel == 1) {
    grasa = (1.2 * IMC + 0.23 * edad - 10.8 * 1 * inc).toFixed(2);
    inc = (peso - (peso * grasa) / 100) * 0.015;
  }
  if (nivel == 2) {
    grasa = (1.2 * IMC + 0.23 * edad - 10.8 * 1 - 2.7).toFixed(2);
    inc = (peso - (peso * grasa) / 100) * 0.03;
  }
  if (nivel == 3) {
    grasa = (1.2 * IMC + 0.23 * edad - 10.8 * 1 - 5.4).toFixed(2);
    inc = (peso - (peso * grasa) / 100) * 0.06;
  }
  if (nivel == 4) {
    grasa = (1.2 * IMC + 0.23 * edad - 10.8 * 1 - 10.8).toFixed(2);
    inc = (peso - (peso * grasa) / 100) * 0.12;
  }
  if (nivel == 5) {
    grasa = (1.2 * IMC + 0.23 * edad - 10.8 * 1 - 21.6).toFixed(2);
    inc = (peso - (peso * grasa) / 100) * 0.24;
  }

  return (
    <View style={styles.Header}>
      <Text
        style={{
          fontSize: 15,
          color: "#FFE500",
          position: "absolute",
          top: 250,
          left: 10,
          right: 0,
          bottom: 0,
          justifyContent: "center",
          alignItems: "center",
        }}
      >
        {"Peso: " + peso + " Kg"}
      </Text>
      <Text
        style={{
          fontSize: 15,
          color: "#FFE500",
          position: "absolute",
          top: 270,
          left: 10,
          right: 0,
          bottom: 0,
          justifyContent: "center",
          alignItems: "center",
        }}
      >
        {"Altura: " + altura + " cm"}
      </Text>
```

```xml
<Text
  style={{
    fontSize: 15,
    color: "#FFE500",
    position: "absolute",
    top: 290,
    left: 10,
    right: 0,
    bottom: 0,
    justifyContent: "center",
    alignItems: "center",
  }}
>
  {"Edad: " + edad}
</Text>
<Text
  style={{
    fontSize: 15,
    color: "#FFE500",
    position: "absolute",
    top: 310,
    left: 10,
    right: 0,
    bottom: 0,
    justifyContent: "center",
    alignItems: "center",
  }}
>
  {"IMB: " + imb}
</Text>
<Text
  style={{
    fontSize: 15,
    color: "#FFE500",
    position: "absolute",
    top: 330,
    left: 10,
    right: 0,
    bottom: 0,
    justifyContent: "center",
    alignItems: "center",
  }}
>
  {"Grasa: " + grasa + "%"}
</Text>
<Svg
  style={{
    fontSize: 15,
    color: "#FFE500",
    position: "absolute",
    top: 0,
    left: 75,
    right: 0,
    bottom: 0,
    justifyContent: "center",
    alignItems: "center",
  }}
  width="250"
  height="500"
  viewBox="0 0 257 300"
  fill="none"
  id="svg8877"
>
  <Path
    d="M5.5 57
  Q 40 50 50 57
  Q 65 47 90 53  "
    //fill="#0FDA55"
    //stroke="#D9D9D9"
    id="Piel"
  />

  <Path
    d={
      "M130.5 40 Q150.5 " +
      (40 - inc / 2 - (peso * grasa) / 500) +
      " 167 50          Q190 " +
      (50 - inc - (peso * grasa) / 300) +
      " 210 55.5          Q" +
      (220 + (peso * grasa) / 500) +
      " " +
      (55.5 - inc / 2 - (peso * grasa) / 400) +
      " 253 57          V71" +
      "Q220 " +
      (70 + inc / 2 + (peso * grasa) / 300) +
      " 210 68          Q188 " +
      (70 + inc / 2 + (peso * grasa) / 300) +
      " " +
      (160 + inc * 0.2) +
      " 65          Q" +
      (159 + (peso * grasa) / (100 * nivel)) +
      " " +
      (100 + (peso * grasa) / (200 * nivel)) +
      " " +
      (159 + (peso * grasa) / (500 * nivel)) +
      " 120          Q" +
      (158 + inc + (peso * grasa) / (200 * nivel)) +
      " 160 " +
      (158 + inc / 9 + (peso * grasa) / 500) +
      " 170 Q158 190 159 200 Q" +
      (158 + inc + (peso * grasa) / 500) +
      " 230 155 257" +
      "H145" +
      "Q " +
      (150 - inc - (peso * grasa) / 400) +
      " 210 145 200      " +
      "Q" +
      (138 - inc / 1.5 - (peso * grasa) / 400) +
      " 165 129 145"
    }
    fill="#393704"
    stroke="cyan"
    id="Grasa"
  />
```

```xml
<Path
  transform="matrix(-1,0,0,1,257.75595,-0.00175479)"
  d={
    "M130.5 40 Q150.5 " +
    (40 - inc / 2 - (peso * grasa) / 500) +
    " 167 50          Q190 " +
    (50 - inc - (peso * grasa) / 300) +
    " 210 55.5          Q" +
    (220 + (peso * grasa) / 500) +
    " " +
    (55.5 - inc / 2 - (peso * grasa) / 400) +
    " 253 57          V71" +
    "Q220 " +
    (70 + inc / 2 + (peso * grasa) / 300) +
    " 210 68          Q188 " +
    (70 + inc / 2 + (peso * grasa) / 300) +
    " " +
    (160 + inc * 0.2) +
    " 65          Q" +
    (159 + (peso * grasa) / (100 * nivel)) +
    " " +
    (100 + (peso * grasa) / (200 * nivel)) +
    " " +
    (159 + (peso * grasa) / (500 * nivel)) +
    " 120          Q" +
    (158 + inc + (peso * grasa) / (200 * nivel)) +
    " 160 " +
    (158 + inc / 9 + (peso * grasa) / 500) +
    " 170 Q158 190 159 200 Q" +
    (158 + inc + (peso * grasa) / 500) +
    " 230 155 257" +
    "H145" +
    "Q " +
    (150 - inc - (peso * grasa) / 400) +
    " 210 145 200      " +
    "Q" +
    (138 - inc / 1.5 - (peso * grasa) / 400) +
    " 165 129 145"
  }
  fill="#393704"
  stroke="cyan"
  id="Grasa2"
/>
<G id="Huesos">
  <Path
    d="M106.385 206.84L105 202.28L100.231 200L111 202.28L110.077 200.55L111 253.58L107.308 257L105 254.72L106.385 206.842"
    fill="#D9D9D9"
    id="TibiaDerecha"
  />
  <Path
    d="M149.385 206.84L148 202.28L151.231 200L154 202.28L153.077 200.55L154 253.58L150.308 257L148 254.72L149.385 206.842"
    fill="#D9D9D9"
    id="TibiaIzquierda"
  />
  <Path
    d="M120 139.091L114 137L104.5 134.305L106.5 145.617L104.5 188.253L104 196.955L109 200L114 189.123V145.617L120 139.091Z"
    fill="#D9D9D9"
    id="FemurDerecho"
  />
  <Path
    d="M138 139.091L144 137L153.5 134.305L151.5 145.617L153.5 188.253L154 196.955L149 200L144 189.123V145.617L138 139.091Z"
    fill="#D9D9D9"
    id="FemurIzquierdo"
  />
  <Path
    d="M118 138H109.5L112.5 130L128 143V115H122.5L118 118Z M139 118H147.5L144.5 130L129 143V115H134.5L139 118Z"
    fill="#D9D9D9"
    id="Cinto"
  />

  <Ellipse
    cx="130"
    cy="39.5"
    rx="36"
    ry="39.5"
    fill="#D9D9D9"
    id="Craneo"
  />
  <Path d="M 135 85 H 125 V 115 H 135  z" fill="#D9D9D9" id="Columna" />
  <Path
    d="M 109 47L98 55V60.5L109 96H118L128.5 86.5V47H109Z M148.5 47L159.5 55V60.5L148.5 96H139.5L129 86.5V47H148.5Z"
    fill="#D9D9D9"
    id="Pulmones"
  />
  <Path
    d="M 165 55 L218 60 v5 L 165 60 z"
    fill="#D9D9D9"
    stroke="#D9D9D9"
    id="HuesoBrazoIzquierdo"
  />
  <Path
    d="M 95 55 L50 60 v5 L 95 60 z"
    fill="#D9D9D9"
    stroke="#D9D9D9"
    id="HuesoBrazoDerecho"
  />
  <Path
    d="M 215 60 L300 60 v5 L215 65 z"
    fill="#D9D9D9"
    stroke="#D9D9D9"
    id="HuesoAnteBrazoIzquierdo"
  />
  <Path
    d="M 45 60 L0 60 v5 L 45 65 z"
    fill="#D9D9D9"
    stroke="#D9D9D9"
    id="HuesoAnteBrazoDerecho"
  />
</G>
<G id="Musculos 1">
  <Path
    d={
      "M122 115 Q " +
      (120 - inc) +
      " 127.381 129 141 Q " +
      (136 + inc) +
      " 130 135 115 L132 118.714 H125.667Z"
    }
    fill="#C96363"
    stroke="#A12C2C"
    id="AbdomenBajo" //Path8775
  />
  <Path
    d={
      "M154.5 123 L151 127 L156.968 169.577 Q" +
      (158 + inc) +
      " 153 156. 122Z"
    }
    stroke="#A12C2C"
    fill="#C96363"
    id="MusculoPequeñoPierna"
  />
  <Path
    d={
      "M136 139.507 Q" +
      (140.214 - inc) +
      " 160.123 146.143 198 Q" +
      (153 + inc) +
      " 175.466 145.107 128Z"
    }
    fill="#C96363"
    stroke="#A12C2C"
    id="Pierna"
  />
  <Path
    d={
      "M153.7 200 Q" +
      (146 - inc) +
      " 215.957 153.7 253 Q" +
      (157 + inc) +
      " 236.473 157 200 L153.7 200 Z"
    }
    fill="#C96363"
    stroke="#A12C2C"
    id="Pantorrilla"
  />
  <Path
    d={
      "M145.828 49.5 H136 Q" +
      (157.207 + inc / 2) +
      " " +
      (46 - inc / 2) +
      " 166 56H146.828C160.414 56 158.241 54.3333 157.207 53.5L145.828 49.5Z"
    }
    fill="#C96363"
    stroke="#A12C2C"
    id="hombro izquierdo"
  />
  <Path
    d={"M137 64.5 Q138.5 " + (42 - inc / 2) + " " 148.5 43z"}
    fill="#C96363"
    stroke="#A12C2C"
    id="espalda"
  />

  <Path
    d={
      "M143 83 Q" +
      (141.105 + inc / 4) +
      " " +
      (93.3433 + inc / 4) +
      " 131 93.83L8 Q" +
      (131 - inc / 4) +
      " " +
      (84.9701 - inc / 4) +
      " 143 83"
    }
    fill="#C96363"
    stroke="#A12C2C"
    id="ab1"
  />
```

pantalla de rutinas stayle y vista.

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

const RoutineScreen = ({ navigation }) => {
  return (
    <View>
      <Text
      style={{
        fontSize: 48,
        textAlign: 'center',
        marginTop: 20,
        marginBottom: 20
      }}
      > Pantalla de Rutinas </Text>
    </View>
  );
}

export default RoutineScreen;
```

pantantalla dieta stayle y vista

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

const DietScreen = ({ navigation }) => {
  return (
    <View>
      <Text
      style={{
        fontSize: 48,
        textAlign: 'center',
        marginTop: 20,
        marginBottom: 20
      }}
      > Pantalla de dietas </Text>
    </View>
  );
}

export default DietScreen;
```

appJason

```
{
  "expo": {
    "name": "proyectodps2022",
    "slug": "proyectodps2022",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "userInterfaceStyle": "light",
    "splash": {
      "image": "./assets/splash.png",
      "resizeMode": "contain",
      "backgroundColor": "#ffffff"
    },
    "updates": {
      "fallbackToCacheTimeout": 0
    },
    "assetBundlePatterns": [
      "**/*"
    ],
    "ios": {
      "supportsTablet": true
    },
    "android": {
      "adaptiveIcon": {
        "foregroundImage": "./assets/adaptive-icon.png",
        "backgroundColor": "#FFFFFF"
      }
    },
    "web": {
      "favicon": "./assets/favicon.png"
    }
  }
}
```

README

```
# ProyectoDPS2022

Proyecto para Diseño y Programación de Software Multiplataforma DPS104 G04T 2022

## Integrantes del trabajo

- Darío Jeffry Larreynaga Contreras.
- Jorge Antonio Díaz García.
- José Ángel Rosales Portillo.
- Mario Ernesto Medina Martínez.
- Nelson Ernesto Muñoz Barahona.

## Gestion de proyecto
Para este apartado usaremos [Trello](https://trello.com/b/CfIOlifs/proyecto-dps).

## Diseños Mock Ups
Los diseños de los Mock-Ups alojados [Aqui](https://drive.google.com/file/d/1leOMwqB03nOqT7kUHYrEkt2AaT7-wtv/view?usp=sharing).

## Licencias Creative Commons implementada
El tipo de licencia utilizada aqui es una de las opciones que nos permite github la llamada, Creative commons zero v1.0 universal license, que dice "Puede copiar, modificar, distribuir y realizar el trabajo, incluso con fines comerciales, todo sin pedir permiso" para mas informacion consulte [Aqui](https://creativecommons.org/publicdomain/zero/1.0/).
```

Fit es una aplicacion nativa, desarrollada por un grupo de estudiantes en ingeniera en ciencias de la computacion, para el desarrollo de la app se hizo uso de React-Native, poniendo en practica todas las habilidades que se fueron aprendiendo durante el ciclo 2 del 2022, se uso de la herramienta de github, trello entre otras herramientas para poder llevar acaba el trabajo en equipo.

se presenta el manual tecnico para dar la informacion abierta sobre el codigo de desarollo de la app para que en el futuro otro grupos de programadores tengan una guia de como esta formada cada una de las pantallas que se le muestran al usuario y otro grupo de programador o el mismo tengan la guia de cada una de las herramientas que se han utilizado asi facilitar el seguimiento y las actualizaciones futuras de FIT..