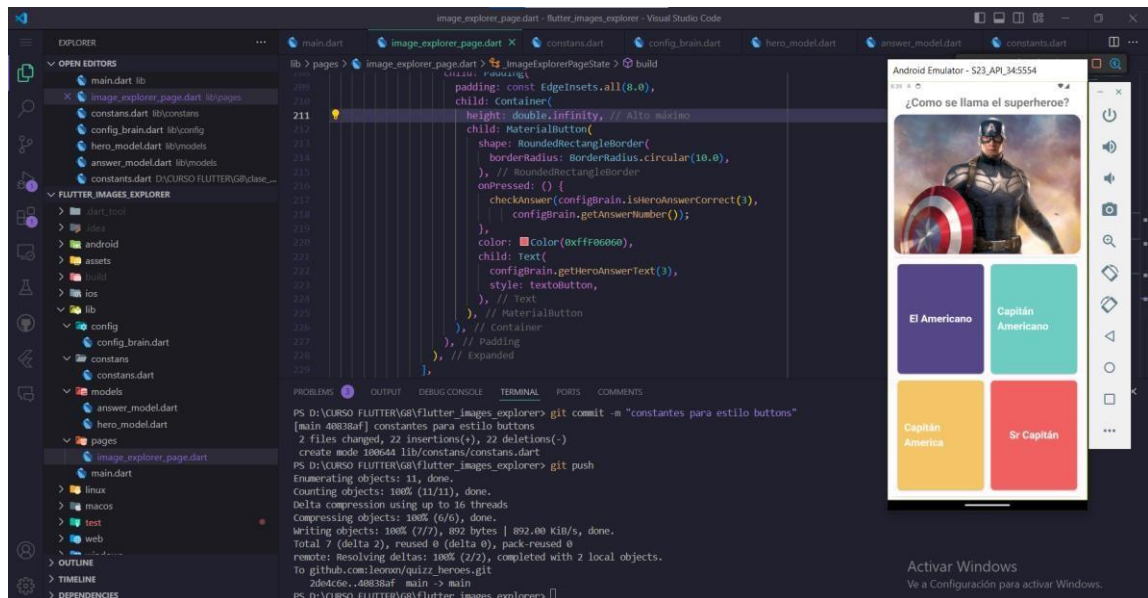


RETO QUIZZAPP SUPERHEROES

NOMBRE: CARLOS ROBERTO LEÓN VELÁSQUEZ Link github:

https://github.com/leonxn/quizz_heroes.git CAPTURAS

(CÓDIGO Y RESULTADO):



```
import 'package:flutter/material.dart';
import 'package:flutter_images_explorer/config/config_brain.dart';
// import 'package:rflutter_alert/rflutter_alert.dart';
import 'package:flutter_images_explorer/constans/constans.dart';
import 'package:quickalert/quickalert.dart';

class ImageExplorerPage extends StatefulWidget {
  @override
  State<ImageExplorerPage> createState() => _ImageExplorerPageState();
}

class _ImageExplorerPageState extends State<ImageExplorerPage> {
  ConfigBrain configBrain = ConfigBrain();
  List<Widget> score = [];
  List<Widget> buttonAnswer = [];

  void generateButtonAnswer() {
    for (int i = 0; i < 4; i++) {
      buttonAnswer.add(
        Expanded(
          child: Padding(
```

```

padding: const EdgeInsets.all(8.0),
child: Container(
  height: double.infinity,
  child: MaterialButton(
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10.0),
    ),
    onPressed: () {
      print(configBrain.getHeroAnswerText(i));
      checkAnswer(configBrain.isHeroAnswerCorrect(i),
        configBrain.getAnswerNumber());
      // buttonAnswer.clear();
      // generateButtonAnswer();
    },
    color: configBrain.getColorButtonAnswer(i),
    child: Text(
      configBrain.getHeroAnswerText(i),
      style: textoButton,
    ),
  ),
),
),
),
),
);
}
print(buttonAnswer[0]);
}

void checkAnswer(bool userAnswer, String numberQuestion) {
  if (configBrain.isFinished() == true) {
    int totalpuntaje = configBrain.puntaje;
    // Alert(
    //   context: context,
    //   type: totalpuntaje == 0 ? AlertType.error : AlertType.success,
    //   title: "QuizApp Heroes",
    //   desc:
    //     "El cuestionario ha llegado a su fin, obtuviste $totalpuntaje
puntos.",
    //   buttons: [
    //     DialogButton(
    //       color: Color(0xff564788),
    //       child: Text(
    //         "Aceptar",
    //         style: TextStyle(
    //           color: Colors.white,
    //           fontSize: 18,
    //           fontWeight: FontWeight.bold,
    //         ),
    //       ),
    //     ),
    //   ],
    // ),
  }
}

```

```

//      ),
//      onPressed: () {
//          configBrain.restarQuizz();
//          score.clear();
//          Navigator.pop(context);
//          setState(() {});
//      }},
//      ]).show();

QuickAlert.show(
  context: context,
  type: totalpuntaje == 0 ? QuickAlertType.error :
QuickAlertType.success,
  title: 'QuizApp Heroes',
  // text:
  //      'El cuestionario ha llegado a su fin, obtuviste $totalpuntaje
puntos.',
  widget: RichText(
    textAlign: TextAlign.center,
    text: TextSpan(
      children: [
        TextSpan(
          text: 'El cuestionario ha llegado a su fin,',
          style: TextStyle(
            fontSize: 20.0,
            fontWeight: FontWeight.normal,
            color: Colors.black,
          ),
        ),
        TextSpan(
          text: 'obtuviste $totalpuntaje puntos.',
          style: TextStyle(
            fontSize: 20.0,
            fontWeight: FontWeight.bold,
            color: Color(0xFF060600),
          ),
        ),
      ],
    ),
    backgroundColor: Colors.white,
    titleColor: Colors.black,
    textColor: Colors.black,
    confirmBtnText: 'Aceptar',
    onConfirmBtnTap: () {
      configBrain.restarQuizz();
      score.clear();
      buttonAnswer.clear();
    },
  ),
);

```

```

        generateButtonAnswer();
        //buttonAnswer.clear();
        Navigator.pop(context);
        setState(() {});
    },
);
} else {
    if (true == userAnswer) {
        score.add(itemScore(numberQuestion, true));
        configBrain.updatePuntaje(true);
        int totalpuntaje = configBrain.puntaje;
        print("total puntaje=$totalpuntaje");
        print("CORRECTO");
    } else {
        score.add(itemScore(numberQuestion, false));
        configBrain.updatePuntaje(false);
        int totalpuntaje = configBrain.puntaje;
        print("total puntaje=$totalpuntaje");
        print("INCORRECTO");
    }
    configBrain.nextQuestion();
    buttonAnswer.clear();
    generateButtonAnswer();
}
setState(() {});
}

Widget itemScore(String numberQuestion, bool isCorrect) {
    return Row(
        children: [
            Text(
                "$numberQuestion: ",
                style: TextStyle(
                    color: Colors.black,
                    fontSize: 20,
                ),
            ),
            Icon(
                isCorrect ? Icons.check : Icons.close,
                color: isCorrect ? Colors.greenAccent : Colors.redAccent,
            ),
        ],
    );
}

@override
void initState() {
    // TODO: implement initState
}

```

```

    super.initState();
    generateButtonAnswer();
}

//String heroName = "";
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(
        "¿Como se llama el superheroe?",
        style: TextStyle(
          fontSize: 28,
          fontWeight: FontWeight.bold,
          color: Colors.black54,
        ),
      ),
      centerTitle: true,
      backgroundColor: Colors.white,
    ),
    body: Padding(
      padding: const EdgeInsets.symmetric(horizontal: 16),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          Container(
            width: MediaQuery.of(context).size.width,
            height: MediaQuery.of(context).size.height / 3,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(20),
              color: Colors.red,
              image: DecorationImage(
                image: AssetImage(
                  configBrain.getHeroImg(),
                ),
                // NetworkImage(
                //   heroList[indexHeroList].imageUrl,
                // ),
                fit: BoxFit.cover,
              ),
            ),
            // child: Image.network(
            //   heroList[indexHeroList],
            //   fit: BoxFit.cover,
            // ),
          ),
          Divider(),

```

```

// Text(
//   heroName,
//   style: TextStyle(
//     fontSize: 25,
//     fontWeight: FontWeight.bold,
//   ),
// ),
// Divider(),
Expanded(
  flex: 1,
  child: Row(
    children: [
      buttonAnswer[0],
      buttonAnswer[1],
      // Expanded(
      //   child: Padding(
      //     padding: const EdgeInsets.all(8.0),
      //     child: Container(
      //       height: double.infinity,
      //       child: MaterialButton(
      //         shape: RoundedRectangleBorder(
      //           borderRadius: BorderRadius.circular(10.0),
      //         ),
      //         onPressed: () {
      //           checkAnswer(configBrain.isHeroAnswerCorrect(0),
      //             configBrain.getAnswerNumber());
      //         },
      //         color: configBrain.getColorButtonAnswer(0),
      //         child: Text(
      //           configBrain.getHeroAnswerText(0),
      //           style: texttoButton,
      //         ),
      //       ),
      //     ),
      //   ),
      // ),
      // Expanded(
      //   child: Padding(
      //     padding: const EdgeInsets.all(8.0),
      //     child: Container(
      //       height: double.infinity,
      //       child: MaterialButton(
      //         shape: RoundedRectangleBorder(
      //           borderRadius: BorderRadius.circular(10.0),
      //         ),
      //         onPressed: () {
      //           checkAnswer(configBrain.isHeroAnswerCorrect(1),
      //             configBrain.getAnswerNumber());

```

```

        //      },
        //      color: configBrain.getColorButtonAnswer(1),
        //      child: Text(
        //          configBrain.getHeroAnswerText(1),
        //          style: textoButton,
        //      ),
        //  ),
        //  ),
        //  ),
        //  ),
    ],
),
Expanded(
  flex: 1,
  child: Row(
    children: [
      buttonAnswer[2],
      buttonAnswer[3],
      // Expanded(
      //   child: Padding(
      //     padding: const EdgeInsets.all(8.0),
      //     child: Container(
      //       height: double.infinity,
      //       child: MaterialButton(
      //         shape: RoundedRectangleBorder(
      //           borderRadius: BorderRadius.circular(10.0),
      //         ),
      //         onPressed: () {
      //           checkAnswer(configBrain.isHeroAnswerCorrect(2),
      //             configBrain.getAnswerNumber());
      //         },
      //         color: configBrain.getColorButtonAnswer(2),
      //         child: Text(
      //           configBrain.getHeroAnswerText(2),
      //           style: textoButton,
      //         ),
      //       ),
      //     ),
      //   ),
      //   ),
      //   ),
      //   ),
      // Expanded(
      //   child: Padding(
      //     padding: const EdgeInsets.all(8.0),
      //     child: Container(
      //       height: double.infinity,
      //       child: MaterialButton(
      //         shape: RoundedRectangleBorder(

```

```

        //          borderRadius: BorderRadius.circular(10.0),
        //        ),
        //      onPressed: () {
        //        checkAnswer(configBrain.isHeroAnswerCorrect(3),
        //          configBrain.getAnswerNumber());
        //      },
        //      color: configBrain.getColorButtonAnswer(3),
        //      child: Text(
        //        configBrain.getHeroAnswerText(3),
        //        style: textoButton,
        //      ),
        //    ),
        //  ),
        // ),
    ],
  ),
),
Divider(),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: score,
)
// Row(
//   mainAxisAlignment: MainAxisAlignment.spaceAround,
//   children: [
//     ElevatedButton(
//       onPressed: () {
//         heroName = heroList[indexHeroList].name;
//         setState(() {});
//       },
//       child: Text("Mostrar superheroe"),
//     ),
//     ElevatedButton(
//       onPressed: () {
//         indexHeroList = Random().nextInt(6);
//         heroName = "";
//         setState(() {});
//       },
//       child: Text("Siguiete"),
//     ),
//   ],
// )
],
),
),
);
}

```



```
}
```

```
import 'package:flutter/material.dart';
import 'package:flutter_images_explorer/models/hero_model.dart';
import 'package:flutter_images_explorer/models/answer_model.dart';

class ConfigBrain {
  List<HeroModel> heroList = [
    HeroModel(
      name: "Capitán América",
      imageUrl: "assets/image/capitan.png",
      answer: [
        AnswerModel(textAnswer: "El Americano", isCorrect: false),
        AnswerModel(textAnswer: "Capitán Americano", isCorrect: false),
        AnswerModel(textAnswer: "Capitán America", isCorrect: true),
        AnswerModel(textAnswer: "Sr Capitán", isCorrect: false),
      ],
    ),
    HeroModel(
      name: "Iron Man",
      imageUrl: "assets/image/ironman.png",
      answer: [
        AnswerModel(textAnswer: "Iron Trooper", isCorrect: false),
        AnswerModel(textAnswer: "Hombre de Acero", isCorrect: false),
        AnswerModel(textAnswer: "Iron Man", isCorrect: true),
        AnswerModel(textAnswer: "Iron Maiden", isCorrect: false),
      ],
    ),
    HeroModel(
      name: "Hulk",
      imageUrl: "assets/image/hulk.png",
      answer: [
        AnswerModel(textAnswer: "Hulk", isCorrect: true),
        AnswerModel(textAnswer: "Hombre Verde", isCorrect: false),
        AnswerModel(textAnswer: "Super Hulk", isCorrect: false),
        AnswerModel(textAnswer: "Hulk Hogan", isCorrect: false),
      ],
    ),
    HeroModel(
      name: "Spiderman",
      imageUrl: "assets/image/spiderman.png",
      answer: [
        AnswerModel(textAnswer: "Mr Araña", isCorrect: false),
        AnswerModel(textAnswer: "El hombre tarantula", isCorrect: false),
        AnswerModel(textAnswer: "Spidermen", isCorrect: false),
        AnswerModel(textAnswer: "Spiderman", isCorrect: true),
      ],
    ),
  ],
}
```

```

    ],
),
HeroModel(
    name: "Mujer maravilla",
    imageUrl: "assets/image/mujer_maravilla.png",
    answer: [
        AnswerModel(textAnswer: "Mujer Maravilla", isCorrect: true),
        AnswerModel(textAnswer: "Srta Maravilla", isCorrect: false),
        AnswerModel(textAnswer: "Sra Maravilla", isCorrect: false),
        AnswerModel(textAnswer: "Super Maravilla", isCorrect: false),
    ],
),
HeroModel(
    name: "Antman",
    imageUrl: "assets/image/antman.png",
    answer: [
        AnswerModel(textAnswer: "Araña Grande", isCorrect: false),
        AnswerModel(textAnswer: "El Insecto", isCorrect: false),
        AnswerModel(textAnswer: "Antman", isCorrect: true),
        AnswerModel(textAnswer: "El abejorro", isCorrect: false),
    ],
),
HeroModel(
    name: "Pantera Negra",
    imageUrl: "assets/image/pantera.png",
    answer: [
        AnswerModel(textAnswer: "Gato Negro", isCorrect: false),
        AnswerModel(textAnswer: "Pantera Negra", isCorrect: true),
        AnswerModel(textAnswer: "Don Gato", isCorrect: false),
        AnswerModel(textAnswer: "Black Tiger", isCorrect: false),
    ],
)
];

int heroIndex = 0;
int puntaje = 0;
String getHeroImg() {
    return heroList[heroIndex].imageUrl;
}

// Respuestas solo Texto
String getHeroAnswerText(int index) {
    return heroList[heroIndex].answer[index].textAnswer;

    //return heroList[heroIndex].answer[0].textAnswer;
    // return heroList
    //      .map((hero) {
    //          hero.answer.map((answer) {

```

```

        //          answer.textAnswer;
        //      }).toList();
        //  })
        //      .toList();
    }

    // Respuestas validacion si es correcta

    bool isHeroAnswerCorrect(int index) {
        return heroList[heroIndex].answer[index].isCorrect;
        //return heroList[heroIndex].answer[0].isCorrect;
    }

    String getAnswerNumber() {
        return (heroIndex + 1).toString();
    }

    void nextQuestion() {
        if (heroIndex < heroList.length - 1) {
            heroIndex++;
        }
    }

    Color getColorButtonAnswer(int index) {
        switch (index) {
            case 0:
                return Color(0xff564788);
            case 1:
                return Color(0xff6CCDC3);
            case 2:
                return Color(0xffF3C566);
            case 3:
                return Color(0xffF06060);
            default:
                return Colors.grey;
        }
    }

    bool isFinished() {
        print(heroIndex);
        print(heroList.length - 1);
        return heroIndex > heroList.length - 2 ? true : false;
    }

    void restarQuizz() {
        heroIndex = 0;
        puntaje = 0;
    }

```

```

void updatePuntaje(bool isCorrect) {
  if (isCorrect) {
    puntaje++;
  }
}

import 'package:flutter_images_explorer/models/answer_model.dart';

class HeroModel {
  String name;
  String imageUrl;
  List<AnswerModel> answer;

  //CONSTRUCTOR POR DEFECTO
  // HeroModel(this.name, this.imageUrl);

  //CONSTRUCTOR CON PARAMETROS
  // HeroModel.conParametros({required this.name, required this.imageUrl});
  HeroModel({
    required this.name,
    required this.imageUrl,
    required this.answer,
  }); //PARAMETROS REQUIRED O SEA SON NECESARIOS SI O SI
}

class AnswerModel {
  String textAnswer;
  bool isCorrect;

  AnswerModel({
    required this.textAnswer,
    required this.isCorrect,
  });
}

import 'package:flutter/material.dart';

final TextStyle titulo = TextStyle(
  fontSize: 24,
  fontWeight: FontWeight.bold,
);

```