
# Computação Móvel

App de Receitas Culinárias - Relatório

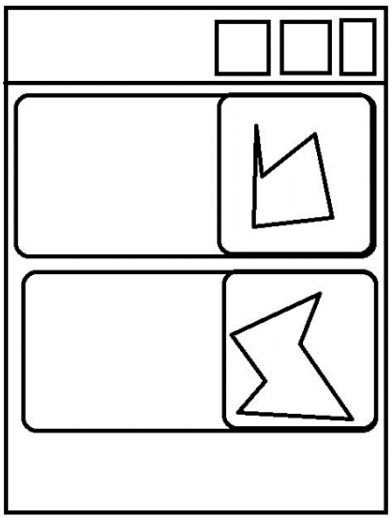
Carlos Passos 16924 | Tiago Oliveira 16931

- # Estrutura do projeto

## 1. Lista de funcionalidades:

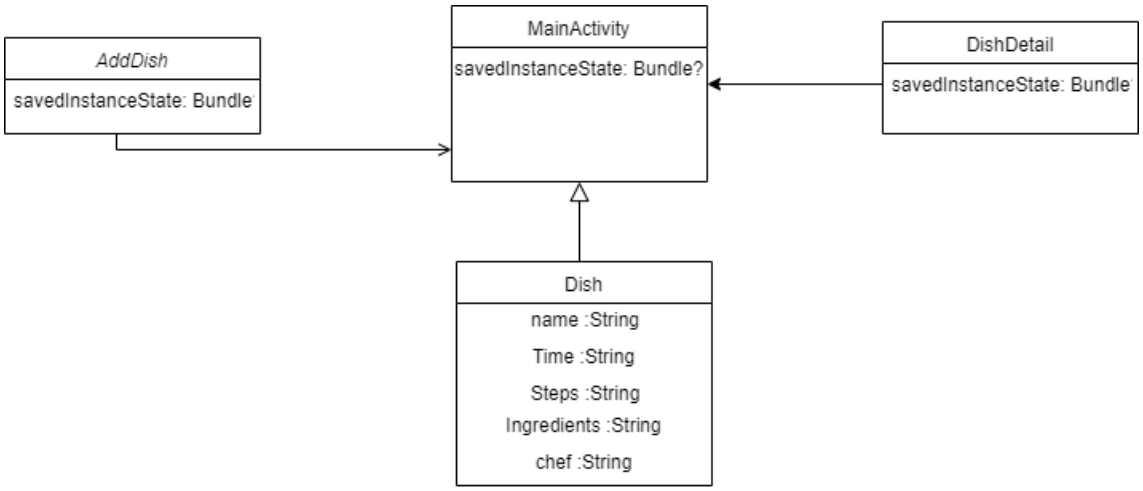
Esta aplicação tem como funcionalidades adicionar/editar/remover receitas de culinária, com o nome do chef, o nome do prato, o passo-a-passo, o tempo que demora a cozinhar e os seus ingredientes. Dá também para partilhar os pratos com outras pessoas. Podemos ainda pesquisar com mais detalhe um prato pelo nome do mesmo. Podemos também tirar fotografias e, portanto, aceder à câmara.

## 2. Desenhos e protótipos da aplicação:



Uma ideia inicial de como seria o menu.

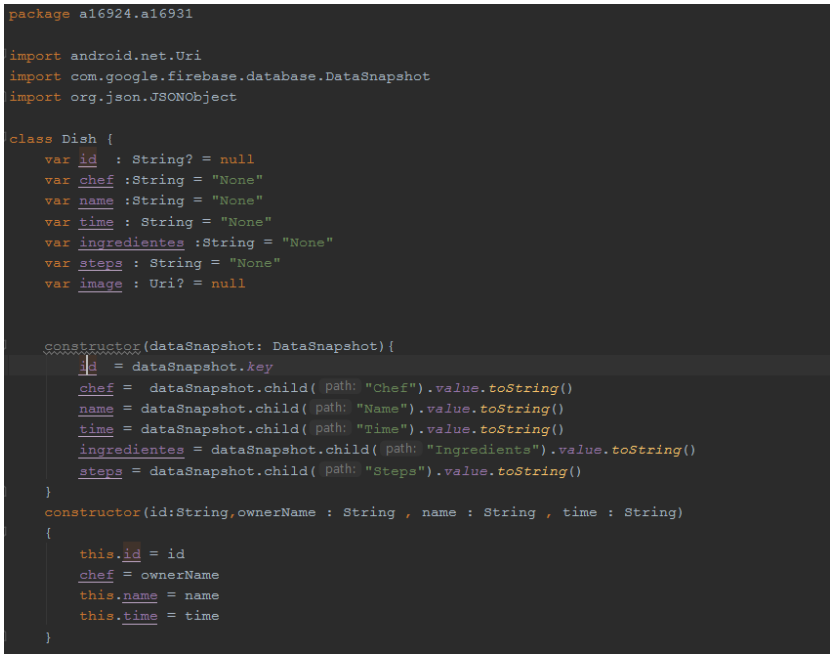
## 3. Modelo de dados:



- **Implementação do projeto**

Este projeto foi criado com base no que aprendemos nas aulas, usando o conhecimento das mesmas, desde o uso do “Firebase” para podermos armazenar os dados dos pratos (bem como removê-los), a criar butões que nos permitam partilhar as receitas com outras pessoas, fazer as transições entre as várias janelas.

Temos a classe “Dish” onde temos os dados que cada prato vai ter:



```
package a16924.a16931

import android.net.Uri
import com.google.firebase.database.DataSnapshot
import org.json.JSONObject

class Dish {
    var id  : String? = null
    var chef :String = "None"
    var name :String = "None"
    var time : String = "None"
    var ingredientes :String = "None"
    var steps : String = "None"
    var image : Uri? = null


    constructor(dataSnapshot: DataSnapshot){
        id  = dataSnapshot.key
        chef =  dataSnapshot.child( path: "Chef").value.toString()
        name = dataSnapshot.child( path: "Name").value.toString()
        time = dataSnapshot.child( path: "Time").value.toString()
        ingredientes = dataSnapshot.child( path: "Ingredients").value.toString()
        steps = dataSnapshot.child( path: "Steps").value.toString()
    }
    constructor(id:String,ownerName : String , name : String , time : String)
    {
        this.id = id
        chef = ownerName
        this.name = name
        this.time = time
    }
```

Como seria natural temos também uma classe, “AddDish”, que vai servir para adicionar cada uma delas bem como armazenar esses dados no Firebase.

Para adicionar os dados no firebase, criamos um botão. Esta função recolhe os dados postos pelo usuário e manda os para a refêrencia do Firebase e para o Json, os Dishes serão guardados consoante o número do prato dentro da base de dados. No fim de executar a função, o programa volta para a MainActivity.

```
class AddDish: AppCompatActivity() {

    val database = FirebaseDatabase.getInstance()
    val myRef = database.getReference()
    private val PERMISSION_CODE = 1000;
    private val IMAGE_CAPTURE_CODE = 1001
    var image_uri: Uri? = null
    var dish : Dish? = null
    var generator = Random()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_dish)
    }


    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.add_menu, menu)
        return true
    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.menu_Add -> {
                dish = Dish( id: "",edit_text_add_owner.text.toString(),edit_text_add_name.text.toString(),edit_text_time.
                dish?.AdicionarPassos(edit_text_add_steps.text.toString())
                dish?.AdicionarIngredientes(edit_text_Ing.text.toString())

                dish?.id = MainActivity.count.toString()
                val intent = Intent()
                intent.putExtra(DISH,dish!!.toJson().toString())
                setResult(MainActivity.ACTIVITY_RESULT_CODE,intent)

                var s : String = MainActivity.count.toString()
                myRef.child( pathString: "users").child(s).setValue(dish)
```

```
R.id.menu_Add -> {
    dish = Dish( id: "",edit_text_add_owner.text.toString(),edit_text_add_name.text.toString(),edit_text_time.text.toString())
    dish?.AdicionarPassos(edit_text_add_steps.text.toString())
    dish?.AdicionarIngredientes(edit_text_Ing.text.toString())

    dish?.id = MainActivity.count.toString()
    val intent = Intent()
    intent.putExtra(DISH,dish!!.toJson().toString())
    setResult(MainActivity.ACTIVITY_RESULT_CODE,intent)

    var s : String = MainActivity.count.toString()
    myRef.child( pathString: "users").child(s).setValue(dish)
    finish()
    return true
```

Para se tirar fotos tivemos que utilizar uma maior quantidade de código , basicamente pedimos permissão ao utilizador para aceder à camera após dada a permissão, o programa irá enviar a imagem para o layout.

```
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
            if (checkSelfPermission(Manifest.permission.CAMERA)
                == PackageManager.PERMISSION_DENIED ||
                checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE)
                == PackageManager.PERMISSION_DENIED){
                //permission was not enabled
                val permission = arrayOf(Manifest.permission.CAMERA, Manifest.permission.WRITE_EXTERNAL_STORAGE)
                //show popup to request permission
                ActivityCompat.requestPermissions( activity: this,permission, PERMISSION_CODE)

            }
            else{
                //permission already granted
                openCamera()
            }
        }
        else{
            //system os is < marshmallow
            openCamera()
        }
```

```
private fun openCamera() {
    val values = ContentValues()
    values.put(MediaStore.Images.Media.TITLE, "New Picture")
    values.put(MediaStore.Images.Media.DESCRIPTION, "From the Camera")
    //camera intent
    val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
   // cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, image_uri)
    startActivityForResult(cameraIntent,  requestCode: 0)

}
```

```
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
    //called when user presses ALLOW or DENY from Permission Request Popup
    when(requestCode){
        PERMISSION_CODE -> {
            if (grantResults.size > 0 && grantResults[0] ==
                PackageManager.PERMISSION_GRANTED){
                //permission from popup was granted
                openCamera()
            }
            else{
                //permission from popup was denied
                Toast.makeText( context: this,  text: "Permission denied", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)

    bit = data!!.extras!!.get("data") as Bitmap
    imageView2.setImageBitmap(data!!.extras!!.get("data")as Bitmap)

    if(requestCode == 1 && resultCode == Activity.RESULT_OK)
    {
        val extra : Bundle ?= data!!.getExtras()

        when(requestCode){
            PERMISSION_CODE -> {
                if (resultCode == Activity.RESULT_OK && data != null ){
                    //permission from popup was granted
                    imageView2.setImageBitmap(data.extras!!.get("data")as Bitmap)
                    openCamera()
                }
                else{
                    //permission from popup was denied
                    Toast.makeText( context: this,  text: "Permission denied", Toast.LENGTH_SHORT).show()
                }
            }
        }
    }
```

Para compartilhar as receitas temos o código simples que o permite fazer ao clicar em um simples botão.

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
        R.id.menu_share -> {

            val shareIntent: Intent
            shareIntent = Intent(Intent.ACTION_SEND)
            shareIntent.type = "text/plain"
            shareIntent.putExtra(Intent.EXTRA_SUBJECT, nameDetail)
            shareIntent.putExtra(Intent.EXTRA_TEXT,  value: "Chef:"+chefDetail+"-Steps:"+stepsDetail+"-CookTime:"+cookingTimeDetail+"Ingredients:"+ingredientsDetail)
            startActivity(Intent.createChooser(shareIntent,  title: "Share"))

            return true
        }
    }

    return super.onOptionsItemSelected(item)
}
```

- ## Tecnologias

Como tecnlogias usamos marioritariamente o Firebase em termos de armazenamento.

- ## Dificuldades

Tivemos dificuldades, a príncipio, principalmente na parte do Firebase ao tentar guardar os dados para que os armazenássemos e mais tarde quisessemos geri-los/vê-los. Após alguma insistência conseguimos fazer com que funcionasse.

Não houve particularmente outra dificuldade grande já que a maioria do trabalho já tinhamos uma noção de como fazer.

- ## Conclusões

Com este trabalho e com a disciplina em geral, aprendemos o básico de como se programa em Android Studio usando o Kotlin como linguagem, o que para ser honesto não gostamos muito da linguagem e teria sido melhor Java já que tem parecências maiores com o que nós já estamos habituados a trabalhar.

Achamos que foi uma boa opção trabalhar em Android porque nos consegue oferecer um leque diferente do que podemos fazer no futuro, e com base neste trabalho, foi possível aprender a fazer coisas básicas como ligar a uma fonte de armazenamento externa (Firebase), bem como como conseguir fazer uma aplicação simples mas que no dia a dia estamos habituados a interagir mas nunca nos perguntamos como tal era feito, e então este trabalho veio ajudar a consolidar um pouco esses conhecimentos.