Universidade da Beira Interior

Departamento de Informática

Elaborado por:

Pedro Mourato 42293 | Eduardo Gonçalves 41714
Marcelo Pereira 41429 | Rafael Victória 41857

Orientador:

Professor Doutor Paulo Fazendeiro
Professor Doutor José Morgado

28 de dezembro de 2022

Conteúdo

Lista de Figuras

Capítulo

1

Introdução

1.1 Enquadramento

Este **Projeto Prático** foi elaborado no âmbito da unidade curricular **Programação de Dispositivos Móveis**.

1.2 Objetivos

Este projeto prático tem como principal objetivo desenhar e prototipar uma aplicação que seja utilizada por condutores na sua procura diária de um lugar de estacionamento.

1.3 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.

2. O segundo capítulo – **Engenharia de Software** – apresenta os requisitos funcionais, requisitos não funcionais e também os digramas.

3. O terceiro capítulo – **Tecnologias e Ferramentas Utilizadas** – descreve as ferramentas utilizadas no desenvolvimento da aplicação.

4. O quarto capítulo – **Desenvolvimento e Implementação** – são apresentados em detalhe os algoritmos pensados e métodos utilizados na implementação dos mesmos, bem como as escolhas face à implementação.

5. O quinto capítulo – **Problemas Encontrados** – expõe uma análise aos problemas encontrados ao longo da resolução deste Projeto.

6. O sexto e último capítulo – **Conclusões e Trabalho Futuro** – apresenta-se uma reflexão do trabalho e conhecimentos adquiridos ao longo do desenvolvimento deste Trabalho Prático bem como uma retrospetiva do percurso face à possibilidade de trabalho futuro no mesmo.

Capítulo

2

Engenharia de Software

2.1 Introdução

Neste capítulo vamos abordar os requisitos funcionais e não funcionais do nosso projeto, assim como alguns diagramas para facilitar um melhor entendimento do programa desenvolvido.

2.2 Requisitos

Requisitos Funcionais

1. A aplicação deve permitir o registo de novos utilizadores;

2. A aplicação deve suportar autenticação com as credenciais inseridas aquando do registo;

3. Após autenticação, a aplicação deve permitir a alteração de todos os dados de utilizador (exceto nome-de-utilizador)

numa interface dedicada, deve também permitir apagar completamente o registo de um utilizador;

4. A aplicação deve permitir a um utilizador registar um lugar livre guardando uma foto- grafia georeferenciada e também marcar um lugar como ocupado;

5. Toda a informação relativa aos utilizadores e lugares livres deve estar armazenada numa base de dados (deve ser elaborado um modelo de dados, implementado numa base de dados, de suporte a esta aplicação);

6. O utilizador deve poder dar feedback sobre um local de estacionamento assinalado (e implicitamente sobre quem o assinalou).

Requisitos não funcionais

1. O Sistema deve ser fácil de usar, com uma interface intuitiva e recursos claros e simples de usar;

2. O Sistema deve ter uma boa velocidade de carregamento e tempo de resposta para garantir uma experiência de utilizador satisfatória;

3. O Sistema deve ser atualizado regularmente para corrigir erros e adicionar novas funcionalidades;

4. O Sistema deve ser responsivo, adaptando-se a diferentes tamanhos de tela e dispositivos para garantir uma boa experiência de utilizador em todos os dispositivos;

5. O Sistema deve ter uma política de privacidade clara e transparente para proteger as informações pessoais dos clientes.

2.3 Diagramas

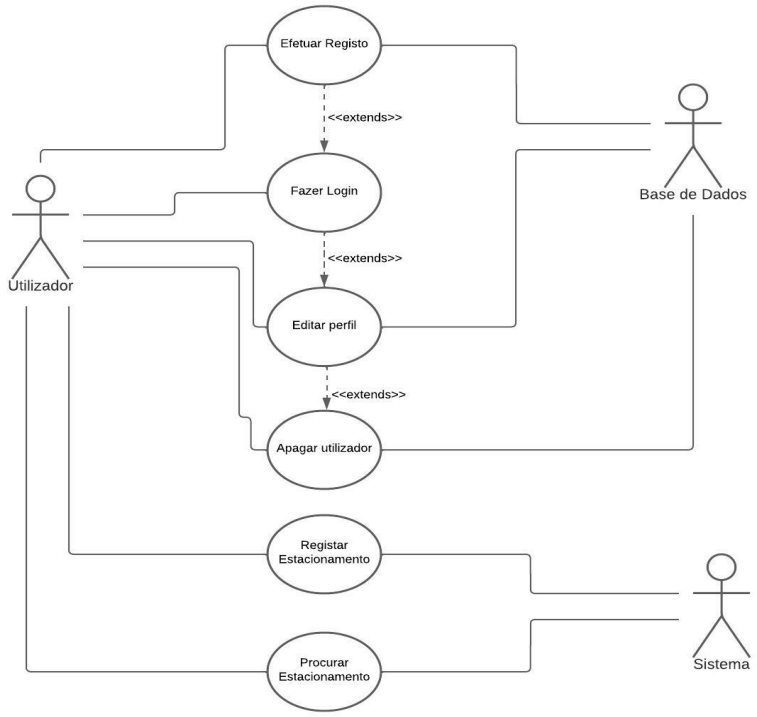

Figura 2.1: Diagrama de Casos de uso

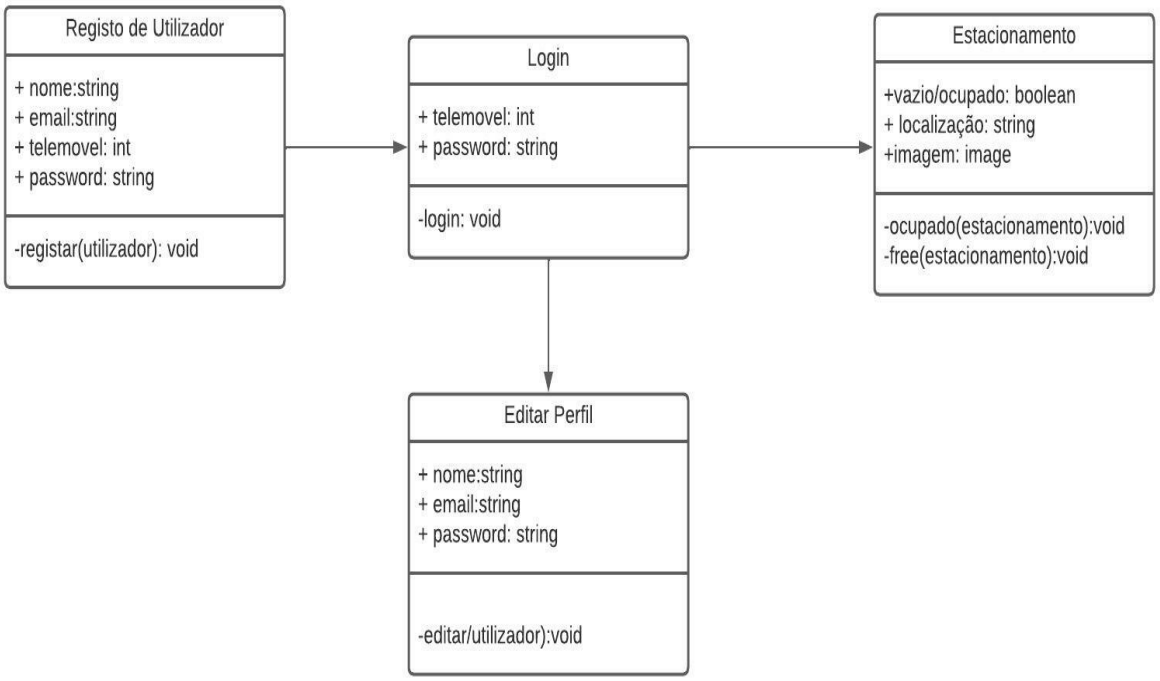

Figura 2.2: Diagrama de classes

2.4 Conclusão

A partir de Engenharia de Software foi-nos possível acelerar o processo de desenvolvimento do programa, pois já tínhamos conhecimentos das necessidades do programa bem como uma estrutura mais definida, tornando o trabalho em equipa mais fácil e mais produtivo.

Capítulo

3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Neste capítulo é apresentado o projeto, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.

3.2 Ferramentas e Tecnologias Utilizadas

É descrito em seguida, as ferramentas e tecnologias utilizadas:

1. **Java** — Java é uma linguagem de programação de alto nível, orientada a objetos e de propósito geral. Ela foi criada no início dos anos 90 pela empresa Sun Microsystems (agora pertencente a Oracle) e tornou-se uma das linguagens mais populares da atualidade.

Uma das principais características do Java é a sua portabilidade, ou seja, um programa escrito em Java pode ser executado em qualquer plataforma, desde que haja uma máquina virtual Java (JVM) instalada nela. Isso permite que o código Java seja facilmente distribuído e executado em diferentes sistemas operacionais, como Windows, Mac, Linux, entre outros.

Outra vantagem do Java é a sua simplicidade e sintaxe clara, o que a torna uma linguagem fácil de aprender para iniciantes. Além disso, o Java possui uma ampla biblioteca de classes padrão, que fornecem muitas funções úteis e facilitam o desenvolvimento de aplicações.

O Java é amplamente utilizado em muitos campos, incluindo aplicativos de desktop, aplicativos para dispositivos móveis, sistemas web, jogos, sistemas embarcados, entre outros. É uma linguagem de programação muito versátil e poderosa, que tem seu uso difundido em diversos setores da indústria e da tecnologia.

2. **Android Studio** — Android Studio é um ambiente de desenvolvimento integrado (IDE) para o sistema operacional Android. Ele foi criado pelo Google e é oferecido gratuitamente para download.

   O Android Studio é uma ferramenta completa para o desenvolvimento de aplicativos para dispositivos Android, incluindo smartphones, tablets e outros dispositivos. Ele

fornece uma série de recursos e ferramentas para auxiliar os desenvolvedores a criar aplicativos de qualidade de forma rápida e eficiente.

O Android Studio é uma ferramenta importante para os desenvolvedores de aplicativos para Android e é amplamente utilizado pelos profissionais de tecnologia ao redor do mundo.

3. **Firebase** — Firebase é uma plataforma de desenvolvimento de aplicativos móveis e web desenvolvida pelo Google. Ela fornece uma série de serviços e recursos para ajudar os desenvolvedores a criar e gerenciar aplicativos de maneira eficiente. Algumas das principais funcionalidades do Firebase incluem, Armazenamento de dados, Autenticação, Análise, Notificações push.

4. **Overleaf** - Overleaf é uma plataforma de colaboração em tempo real para escrita e edição de documentos LaTeX. Ele é projetado para facilitar a colaboração em projetos de escrita científica, permitindo que os usuários trabalhem em documentos de maneira simultânea e compartilhem facilmente as alterações. A plataforma também fornece uma série de recursos úteis, como um editor LaTeX integrado, suporte para vários idiomas e opções de formatação de documentos, além de integração com outras ferramentas de pesquisa e gerenciamento de referências. Overleaf é uma opção popular para escritores científicos e aca-

démicos que desejam usar o LaTeX para criar documentos profissionais de alta qualidade.

3.3 Conclusões

Neste capítulo foi descrito as tecnologias e ferramentas utilizadas na realização e desenvolvimento do projeto referido neste documento.

Capítulo

4

Desenvolvimento e Implementação

4.1 Introdução

Neste capítulo são apresentados em detalhe os algoritmos pensados e métodos utilizados na implementação dos mesmos, bem como as escolhas face à implementação.

4.2 Detalhes de Implementação

Nesta secção são descritos os detalhes de implementação, através da explicação dos respetivos algoritmos.

Código

```
    <uses-permission android:name="android.permission.
        ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CAMERA" />
```

```
    <uses-permission android:name="android.permission.
        ACCESS_FINE_LOCATION" />
```

Excerto de Código 4.1: Permite acesso à internet, câmera e localização.

```
DatabaseReference databaseReference = FirebaseDatabase.getInstance().
    getReferenceFromUrl("https://projpdm-8fe85-default-rtdb.firebaseio.
    com");
```

Excerto de Código 4.2: Base de Dados

```
StorageReference storageReference = FirebaseStorage.getInstance().
    getReference();
```

Excerto de Código 4.3: Storage da base de dados.

```
if (!emailTxt.isEmpty() && !passwordTxt.isEmpty() && !fullnameTxt.
    isEmpty() && !conPasswordTxt.isEmpty()) {
databaseReference.child("users").child(teste2).child("fullname").
    removeValue();
databaseReference.child("users").child(teste2).child("fullname").
    setValue(fullnameTxt);
databaseReference.child("users").child(teste2).child("email").
    removeValue();
databaseReference.child("users").child(teste2).child("email").setValue(
    emailTxt);
databaseReference.child("users").child(teste2).child("password").
    removeValue();
databaseReference.child("users").child(teste2).child("password").
    setValue(passwordTxt);
finish();
}
```

Excerto de Código 4.4: Atualizar a base de dados com novos valores.

```
private void uploadImage(){

String addressloc2 = addressloc.getText().toString();
StorageReference free = storageReference.child("Free Parking");
```

```
StorageReference ref = free.child(addressloc2);
imgCamera.setDrawingCacheEnabled(true);
imgCamera.buildDrawingCache();
Bitmap bitmap = ((BitmapDrawable) imgCamera.getDrawable()).getBitmap();
ByteArrayOutputStream baos = new ByteArrayOutputStream();
bitmap.compress(Bitmap.CompressFormat.JPEG,100,baos);
byte[] data = baos.toByteArray();
UploadTask uploadTask = ref.putBytes(data);
uploadTask.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {
        Toast.makeText(free.this, "Unsuccess", Toast.LENGTH_SHORT).show
            ();
    }
        }).addOnSuccessListener(new OnSuccessListener<UploadTask.
            TaskSnapshot>() {
             @Override
             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot)
                 {
                 Toast.makeText(free.this, "Success", Toast.LENGTH_SHORT
                     ).show();
             }
        });
    }
```

Excerto de Código 4.5: Enviar imagem para a storage da base de dados.

```
databaseReference.child("free").addListenerForSingleValueEvent(new
   ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
        String lol = feedback3.getText().toString();
        Date currentTime = Calendar.getInstance().getTime();
        int xx = currentTime.getHours();
        int zz = currentTime.getMinutes();
        int yy = currentTime.getDay();
        int aa = currentTime.getMonth();
        uploadImage();
        databaseReference.child("free").child(addressloc2).child("
            feedback").setValue(lol);
```

```
        databaseReference.child("free").child(addressloc2).child("hour"
            ).setValue(xx+":"+zz);
        atabaseReference.child("free").child(addressloc2).child("date")
            .setValue(yy+"/"+aa);

        finish();
        }
```

Excerto de Código 4.6: Envia os dados da localização para a base de dados.

```
private void getLocation(){

        try{
            locationManager=(LocationManager) getApplicationContext().
                getSystemService(LOCATION_SERVICE);
            locationManager.requestLocationUpdates(LocationManager.
                GPS_PROVIDER,5000,5,free.this);
        }catch (Exception e){

            e.printStackTrace();
        }

    }
```

Excerto de Código 4.7: Recebe a localização atual.

```
protected void onActivityResult(int requestCode, int resultCode,
    @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (resultCode == RESULT_OK && requestCode == CAMERA_REQ_CODE)
            {
            Bitmap img = (Bitmap) data.getExtras().get("data");
            imgCamera.setImageBitmap(img);
        }
    }
```

Excerto de Código 4.8: Carregar imagem na aplicação.

```
databaseReference.child("users").addListenerForSingleValueEvent(new
    ValueEventListener() {
```

```
        @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {

        if(snapshot.hasChild(phoneTxt)){
            final String getPassword= snapshot.child(phoneTxt).child("
                password").getValue(String.class);
            if(getPassword.equals(passwordTxt)){
            Toast.makeText(Login.this, "Success Logged In", Toast.
                LENGTH_SHORT).show();
            Intent i = new Intent(Login.this,MainActivity.class);
            i.putExtra("phoneTxt",phoneTxt);
            Intent phoneid = new Intent(Login.this,Edit.class);
            phoneid.putExtra("phoneTxt",phoneTxt);
            startActivity(i);
            finish();
            }
            else{
                Toast.makeText(Login.this, "Incorrect Password", Toast.
                    LENGTH_SHORT).show();
                     }

                }else{
                Toast.makeText(Login.this, "Incorrect Password", Toast.
                    LENGTH_SHORT).show();
                     }
            }
```

Excerto de Código 4.9: Efectua login se o utilizador estiver registado.

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        final EditText fullname=findViewById(R.id.fullname);
        final EditText email = findViewById(R.id.email);
        final EditText phone = findViewById(R.id.phone);
        final EditText password = findViewById(R.id.password);
        final EditText conPassword = findViewById(R.id.conPassword);

        final Button registerBtn = findViewById(R.id.registerBtn);
```

```
final TextView loginNowBtn = findViewById(R.id.loginNowBtn);
```

Excerto de Código 4.10: Dados a inserir no registo.

Aplicação

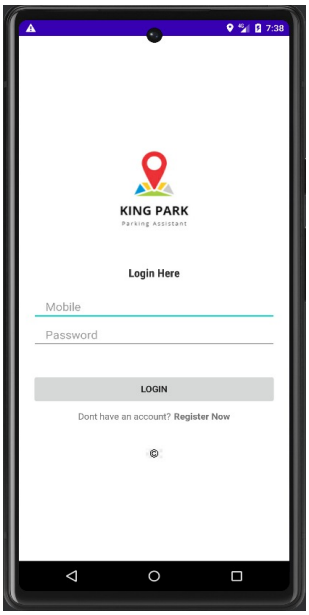

Figura 4.1: Ecrã de Login

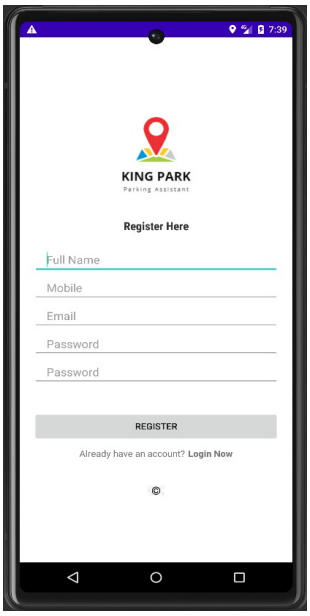

Figura 4.2: Ecrã de Registo

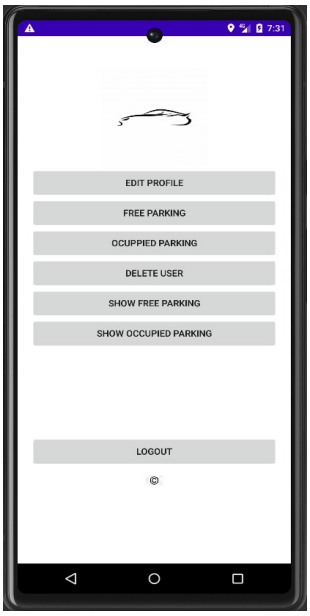

Figura 4.3: Ecrã principal

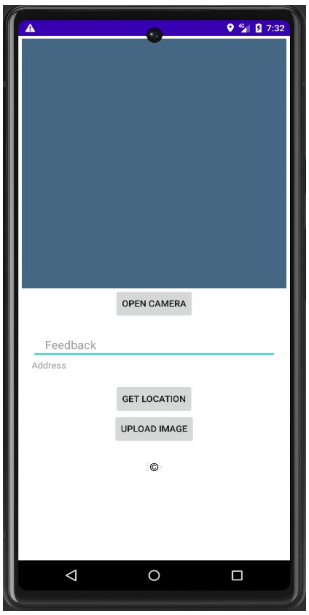

Figura 4.4: Ecrã de registo de parque livre

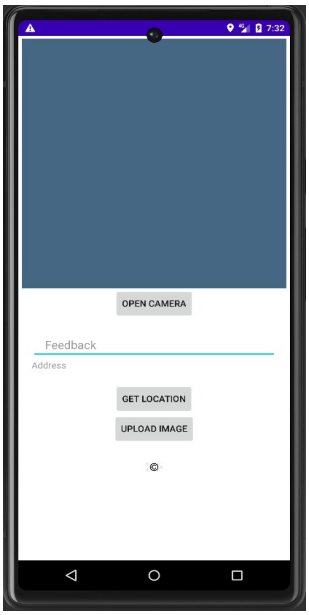

Figura 4.5: Ecrã de registo de parque ocupado

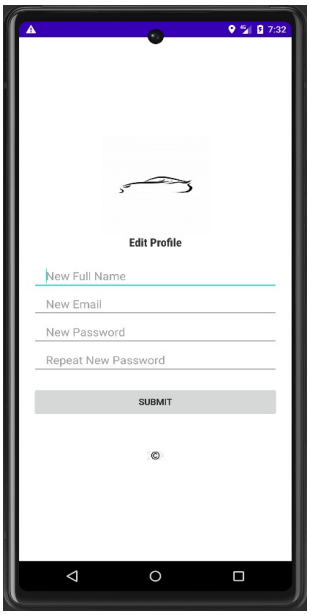

Figura 4.6: Ecrã de editar perfil

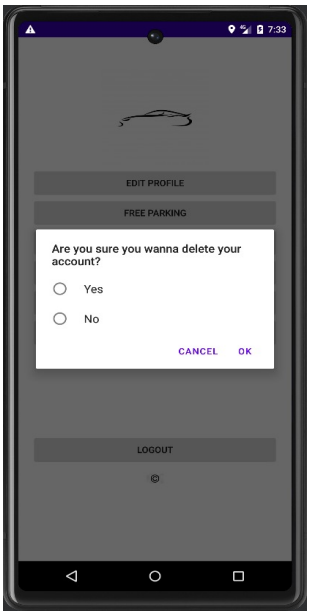

Figura 4.7: Ecrã de apagar utilizador

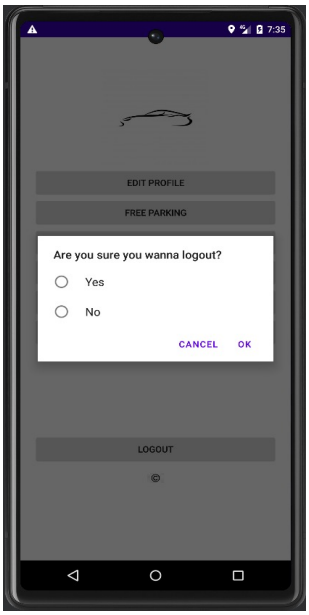

Figura 4.8: Ecrã de logout

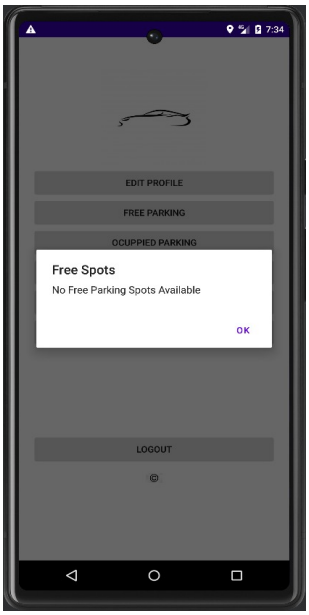

Figura 4.9: Ecrã de mostrar lugares ocupados

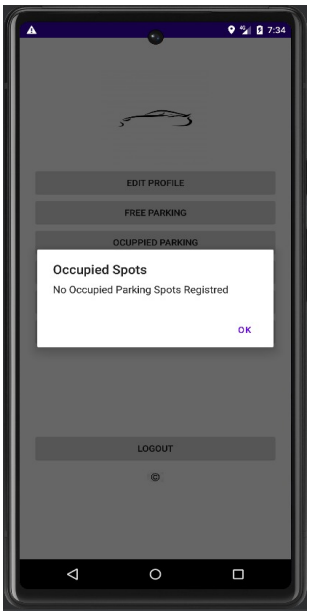

Figura 4.10: Ecrã de mostrar lugares livres

Base de dados

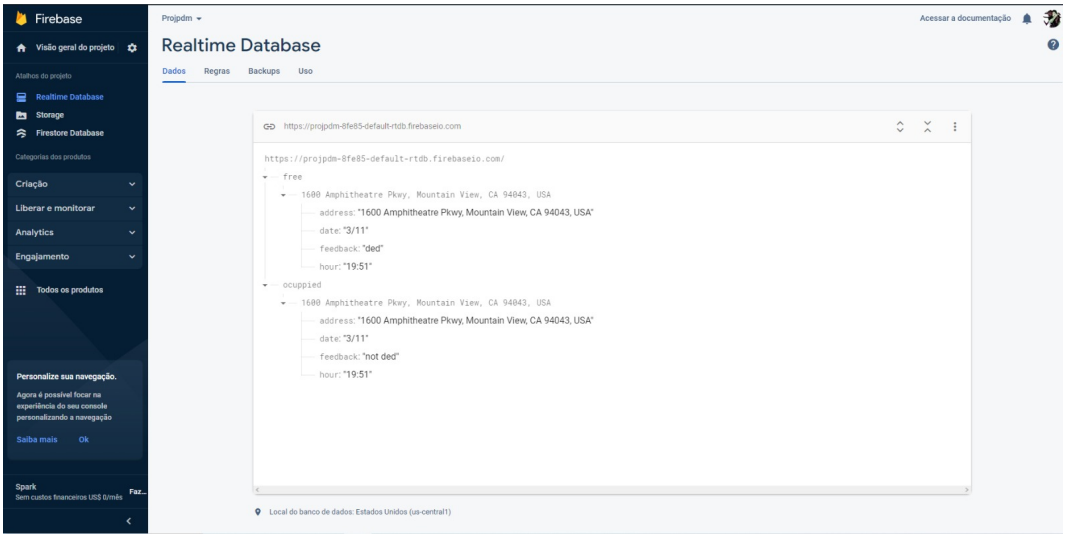

Figura 4.11: Realtime database

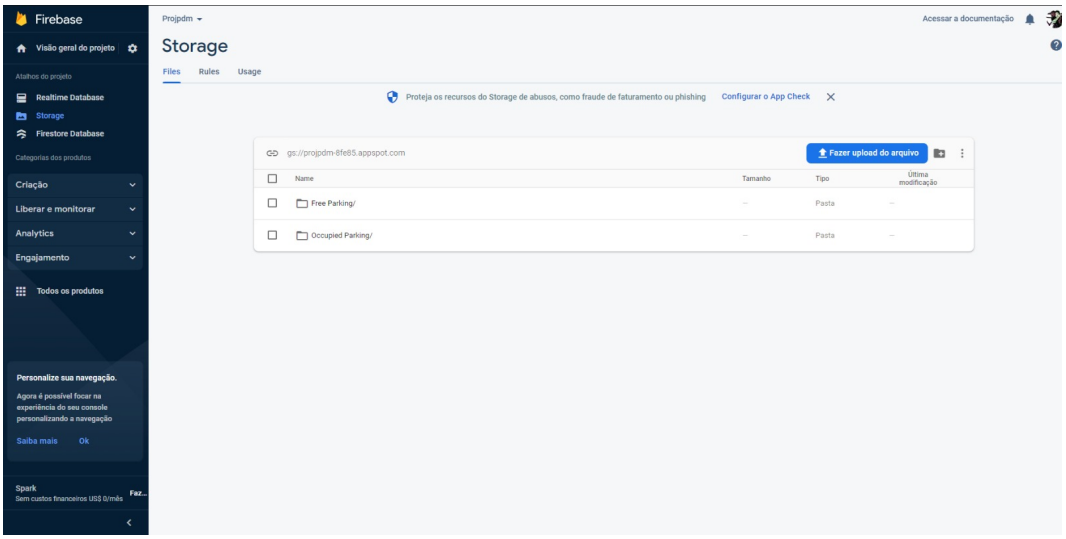

Figura 4.12: Storage

4.3 Conclusões

No presente capítulo foram apresentados os passos necessários ao desenvolvimento do Projeto Prático e do funcionamento do mesmo. Desta forma, através do conteúdo exposto neste capítulo, encontra-se a apresentação do projeto desenvolvido e o funcionamento do mesmo.

Capítulo

5

Problemas Encontrados

5.1 Introdução

Este capítulo está organizado da seguinte forma:

1. **Introdução** – Apresentação de como o capítulo está organizado;

2. **Problemas Encontrados** – Análise aos problemas encontrados durante a resolução do trabalho;

3. **Conclusão**

5.2 Problemas Encontrados

Ao desenvolver a aplicação para ajudar motoristas a encontrar lugares de estacionamento disponíveis, alguns problemas que foram encontrados incluem:

Fiabilidade da informação: é importante garantir que as informações sobre lugares de estacionamento disponíveis e ocupados estejam sempre atualizadas e sejam precisas. Isso pode ser um desafio, já que os usuários podem não atualizar as informações de forma consistente ou podem fornecer informações incorretas.

Integração com outros sistemas: pode ser necessário integrar a aplicação com outros sistemas, como o Google Maps ou outras plataformas de geolocalização, para fornecer rotas e informações precisas sobre a localização dos lugares de estacionamento.

5.3 Conclusão

Este capítulo permitiu analisar o trabalho ao longo do tempo, bem como as escolhas e pensamento crítico. Esta análise permitiu tirar conclusões acerca das estratégias utilizadas, as quais serão expostas no Capítulo seguinte.

Capítulo

6

Conclusões e Trabalho Futuro

6.1 Conclusões Principais

Este projeto permitiu adquirir um melhor conhecimento acerca da linguagem de programação Java,e também permitiu ganhar uma melhor compreensão de certos temas recorrentes na Unidade Curricular de Programação de Dispositivos Móveis.

6.2 Trabalho Futuro

Numa perspectiva futura, seria interessante a criação de uma opção de reserva de estacionamentos, permitindo que os usuários reservem um lugar de estacionamento com antecedência.