



UNIVERSIDADE
LUSÓFONA

Parque de estacionamento – Front End

Trabalho Final de curso

Relatório Final

Nome do Aluno: Manuel Sousa

Nome do Aluno: Daniel Rocha

Nome do Orientador: João Carvalho

Trabalho Final de Curso | LEI ou LIG | 22/07/2022

www.lusofona.pt

Direitos de cópia

Parque de estacionamento – Front-End, Copyright de Manuel Sousa e Daniel Rocha, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Índice

Índice.....	iii
Resumo.....	iv
Abstract	v
Lista de Figuras	vi
1 Identificação do Problema.....	1
2 Viabilidade e Pertinência.....	2
3 Levantamento e análise dos Requisitos	3
3.1 Requisitos funcionais - Implementados.....	3
3.1.1 Aplicação móvel.....	3
3.1.2 Website	3
3.2 Requisitos não funcionais - Implementados.....	3
3.3 Requisitos funcionais – Não Implementados.....	4
3.3.1 Aplicação móvel.....	4
3.3.2 Website	4
4 Benchmarking.....	5
5 Solução Desenvolvida.....	6
5.1 Arquitetura de Solução	7
5.2 Tecnologia.....	8
5.3 Tecnologia mobile	14
5.4. Workflow de entrada.....	21
6 Método e planeamento.....	23
7 Resultados	24
7.1. Processo de Login.....	24
7.2. Processo de Verificação.....	26
7.3. Processo de Adicionar ou Editar Utilizador	27
8 Conclusão e trabalhos futuros	29
Bibliografia	30
Glossário.....	31

Resumo

No âmbito da cadeira de *trabalho final de curso* foi desenvolvido um projeto que visa melhorar futuramente o *parque de estacionamento* da *universidade Lusófona*. Atualmente, o sistema que gere este parque ainda é um pouco antiquado, consistindo num guarda que vê se a pessoa que pretende estacionar no parque está presente na lista dos utilizadores com acesso ao mesmo.

Utilizando câmaras para detetar a matrícula do veículo, uma base de dados e uma aplicação o nosso objetivo é conseguir facilitar o trabalho do guarda e da pessoa que pretende estacionar o seu carro de modo a tornar todo este processo mais rápido.

Este trabalho está focado na parte *frontend* deste projeto onde o nosso objetivo é tornar a experiência do utilizador o mais agradável e simples possível criando uma *aplicação mobile* onde é possível abrir a cancela apenas com um toque. Na parte administrativa e de segurança vamos desenvolver um *website* onde os administradores vão conseguir modificar os utilizadores com permissão a aceder ao parque e onde os guardas, numa parte inicial, decidem se a cancela pode ou não ser aberta.

Palavras-chave: *parque de estacionamento, universidade Lusófona, frontend, aplicação mobile, website.*

Abstract

As part of the course *final work subject*, we are developing a project that aims to improve the future of *Lusófona university's parking lot*. Currently, the system that manages the parking at this university is still outdated, consisting of a guard who checks if the person who wants to enter the park is present in the list of users with access to it. Using cameras to detect the vehicle registration number, a database and an application, our objective is to make the work easier for the guard and the person who wants to park their car to make this whole process faster.

This work is focused on the Front-End of this project where our goal is to make the user experience as pleasant and simple as possible by creating a *mobile application* where it is possible to open the gate with just one touch. In the administrative and guard part, we will develop a "website" where administrators will be able to modify users with permission to access the park and where guards, initially, decide whether the gate can be opened or not.

Keywords: *parking lot, Lusófona university, frontend, mobile application, website.*

Lista de Figuras

Figura 1 - Serviços que fornecem software de parquímetro com detecção de matrícula.....	5
Figura 2 – Arquitetura de solução	7
Figura 3 - Mapa do Website	8
Figura 4 - Página Login.....	9
Figura 5 - Código da Encriptação.....	9
Figura 6 - Página Inicial	10
Figura 7 - Tabela de utilizadores.....	11
Figura 8 - Página Não Autorizada	11
Figura 9 - Página Adicionar	12
Figura 10 - Página Detalhes.....	12
Figura 11 - Página Editar	13
Figura 12 - Aplicação Mobile.....	13
Figura 13 - Página Loading Mobile	14
Figura 14 - Página Login Mobile	15
Figura 15 - Página Inicial	15
Figura 16 - Página Mudar Senha Mobile	16
Figura 17 - Função salvar utilizador	17
Figura 18 - Botão Login Mobile	17
Figura 19 - Botão Entrar Mobile	18
Figura 20 - Botão Sair Mobile.....	18
Figura 21 - Função Encriptar Mobile.....	18
Figura 22 - Botão Mudar Password	19
Figura 23 - Axure: Ferramenta usada para o desenvolvimento do mockup	20
Figura 24 – ASP.NET: Framework usada para o desenvolvimento do website	20
Figura 25 - Flutter: Ferramenta usada para desenvolvimento da app mobile	20
Figura 26 - Protótipo da aplicação mobile	20
Figura 27 - A Processar pedido Aplicação Mobile	21
Figura 28 – Pedido processado Aplicação Mobile	21
Figura 29 - Pedido entrada Aplicação Website	22
Figura 30 - Consola Detecção de Matrículas.....	22
Figura 31 - Diagrama de Gantt	23
Figura 32 - Página Login.....	24
Figura 33 - Servidor a validar utilizador	24
Figura 34 - Página Inicial	25
Figura 35 - Página Login Mobile	25
Figura 36 - Servidor a validar utilizador - Mobile	25
Figura 37 - Página Inicial Mobile.....	25
Figura 38 - Validação da Matrícula	26
Figura 39 – Processar Pedido	26
Figura 40 - Outro pedido em processamento	26
Figura 41 - Adicionar utilizador	27
Figura 42 - Utilizador Adicionado	27
Figura 43 - Editar Utilizador	28

1 Identificação do Problema

Considerada, pelo Scimago Institutions Ranking, a Universidade Lusófona de Humanidades e Tecnologias a melhor universidade privada de Portugal é esperado que a mesma acompanhe as tendências e que forneça aos seus alunos e docentes a melhor experiência possível, porém, ao ver como funciona o processo de aceder ao parque de estacionamento desta instituição, deparamo-nos com um sistema demorado e pouco eficaz. Quando alguém quer deixar a sua viatura no parque tem de falar com o guarda que se encontra presente na cancela, informando-o das suas credenciais, tendo depois de ficar à espera que o guarda introduza essas credenciais no sistema, verifique se estão presentes, e autorize a sua entrada.

O projeto desenvolvido neste TFC, apesar de não ter sido testado fisicamente no parque da Universidade, implementa um sistema que tem a capacidade de fornecer aos utilizadores e ao guarda do parque de estacionamento da Lusófona uma experiência de utilização mais rápida através do uso de uma aplicação mobile onde o utilizador é capaz de abrir a cancela do parque ao pressionar apenas um botão e através de um website que apresenta ao guarda a informação de autorização de entrada ou saída do utilizador.

2 Viabilidade e Pertinência

Em termos de viabilidade do projeto este não se fica por um mero trabalho académico pois o seu propósito foi criar um sistema que gere um parque de estacionamento que futuramente poderá ser implementado fisicamente na Universidade Lusófona ou noutros lugares que contenham um parque de estacionamento.

Em termos de pertinência, a nossa aplicação aumenta a segurança da entrada do parque pois introduzimos a verificação da matrícula do veículo e verificamos se esta está associada ao utilizador que pretende entrar, aumenta também a rapidez com que uma pessoa demora a aceder ao parque, como já foi referido anteriormente, e informa aos utilizadores quantos lugares estão disponíveis.

3 Levantamento e análise dos Requisitos

Inicialmente foram definidos alguns requisitos a serem implementados no frontend, tanto funcionais como não funcionais, pensando na melhor maneira de responder às necessidades dos nossos utilizadores de uma maneira simples. No fim do desenvolvimento deste projeto alguns requisitos foram realizados e outros não, acrescentando também alguns que não foram introduzidos inicialmente.

3.1 Requisitos funcionais - Implementados

3.1.1 Aplicação móvel

- ❖ O utilizador deverá conseguir fazer login na aplicação móvel com as suas credenciais.
- ❖ Caso o utilizador tenha autorização, este consegue fazer um pedido para abrir a cancela de modo a ser capaz de entrar ou sair do parque.
- ❖ Implementação de encriptação AES nas palavras de passe. A razão do acrescentamento deste requisito foi para aumentar a segurança da aplicação.
- ❖ Caso o utilizador saia da aplicação sem fazer o logout, se voltar a abrir não precisa de voltar a fazer o login, aparecendo automaticamente no ecrã principal. A razão do acrescentamento deste requisito foi pelo fato de no telemóvel ser maçante estar sempre a digitar as credenciais sempre que abrir a aplicação.

3.1.2 Website

- ❖ O guarda e o administrador conseguem fazer login no website com uma conta pré-criada.
- ❖ Na página inicial é possível verificar a matrícula do utilizador que fez um pedido de abertura da cancela e confirma a abertura ou não da mesma caso a matrícula corresponda à matrícula associada ao utilizador na base de dados.
- ❖ Deverá ser possível de ver uma tabela com informações dos utilizadores com acesso ao parque, onde apenas os administradores têm acesso.
- ❖ O administrador é capaz de introduzir novos utilizadores com acesso ao parque, e atualizar dados de utilizadores já existentes.
- ❖ Implementação de encriptação AES nas palavras-passe. A razão do acrescentamento deste requisito foi para aumentar a segurança do website.
- ❖ O guarda consegue saber dados do utilizador ao digitar a matrícula num campo de procura caso haja algum imprevisto.

3.2 Requisitos não funcionais - Implementados

- ❖ Tendo sido desenvolvida em Flutter, um software para criação de aplicações híbridas, a aplicação móvel suporta tanto Android como o IOS.

- ❖ O utilizador deve estar conectado a uma rede local do parquímetro para a aplicação ser capaz de comunicar com o servidor backend.
- ❖ Não tendo sido testado por utilizadores, acreditamos que este projeto irá ter um impacto positivo nos utilizadores melhorando a sua experiência de acesso ao parque, tanto no aspeto de segurança como na rapidez de acesso.
- ❖ Tanto a aplicação como o website são capazes de enviar e receber informações processadas pelo backend.
- ❖ As comunicações com o backend devem ser feitas através de gRPC.

3.3 Requisitos funcionais – Não Implementados

3.3.1 Aplicação móvel

- ❖ Inicialmente o objetivo era desenvolver este projeto e implementar na Universidade, porém o mesmo não se sucedeu, logo o utilizador é capaz de fazer login, porém não usa as credenciais da lusófona como foi pensado no início.
- ❖ É possível ver uma estimativa da quantidade de lugares vagos no parque de estacionamento.

3.3.2 Website

- ❖ Sendo um dos objetivos a capacidade de o guarda abrir e fechar cancela através do website este requisito não foi implementado pois o projeto não foi posto em prática fisicamente.

4 Benchmarking

No caso de concorrência, após fazermos uma pesquisa sobre o tema do nosso trabalho, conseguimos encontrar uma quantidade considerável de projetos de empresas que vendem câmeras de detecção para matrículas e com software embutido, como por exemplo a [Vicrodel\[1\]](#), [Prevalta \[2\]](#), etc.



Figura 1 - Serviços que fornecem software de parquímetro com detecção de matrícula.

A diferença entre o nosso software é pela aplicação que irá ser desenvolvida para a nossa faculdade onde, apenas os alunos e docentes registados que pertencem há universidade conseguem pedir para abrir a cancela e saber quantos lugares estão disponíveis. A vantagem do software feito por essas empresas é o facto de que a matrícula do carro é lida quando ele chega sem o utilizador ter de pressionar qualquer abrindo imediatamente a cancela no caso de a matrícula estar registada.

5 Solução Desenvolvida

Para a realização deste TFC foi desenvolvido um website, para uso dos guardas e administradores do parque de estacionamento, e uma aplicação mobile compatível com IOS e android para os utilizadores que desejam estacionar o seu automóvel no parque.

O website é destinado aos administradores e guardas do parque do modo a fazerem a gestão do mesmo. Através da sua utilização, estes serão capazes de gerir funcionalidades tais como, gerir os utilizadores, como editar informações e adicionar novos, autorizar a entrada e saída dos mesmos no parque.

A aplicação é destinada aos utilizadores do parque que após feito o login, poderão fazer um pedido de abertura da cancela do parque, que, caso os seus dados sejam válidos, será aceite pelo guarda.

Os outros grupos que também estão a trabalhar neste projeto desenvolveram um backend que constitui num servidor ligado a uma base de dados que contem todos os utilizadores com acesso ao parque, com as respetivas matrículas registadas. E um sistema de deteção de matrículas que irá usar duas cameras para a recolha de dados.

Na prática, o utilizador, ao aceder ao parque, terá de clicar no botão de entrada através da aplicação mobile, sendo o pedido enviado ao backend e, com o uso das câmeras será tirada uma fotografia à parte frontal do veículo do utilizador que pretende abrir a cancela. Da fotografia tirada é extraída a informação da matrícula do veículo, onde é verificado se a mesma se encontra presente na base de dados dos utilizadores com o devido acesso ao parque de estacionamento. Finalizando, o backend envia uma resposta para o website que o guarda está a supervisionar, onde mostra se o utilizador que está a tentar aceder ao parque é elegível. Sendo ele aceite, o guarda abre a cancela para a permissão de passagem ao utilizador.

5.1 Arquitetura de Solução

Acerca da arquitetura do projeto, como podemos observar na figura abaixo, todas as comunicações feitas entre o Frontend e o Backend foram feitas através do software gRPC, que é um sistema de chamada de procedimento remoto desenvolvido pela Google.

A comunicação entre o backend e a deteção de matrículas também é feita usando gRPC. No momento em que a aplicação mobile envia o pedido de entrada/saída ao backend, o backend envia o pedido á deteção de matrículas, que retorna como resposta a fotografia e a matrícula que foi detetada. Esta informação é enviada posteriormente para o website.

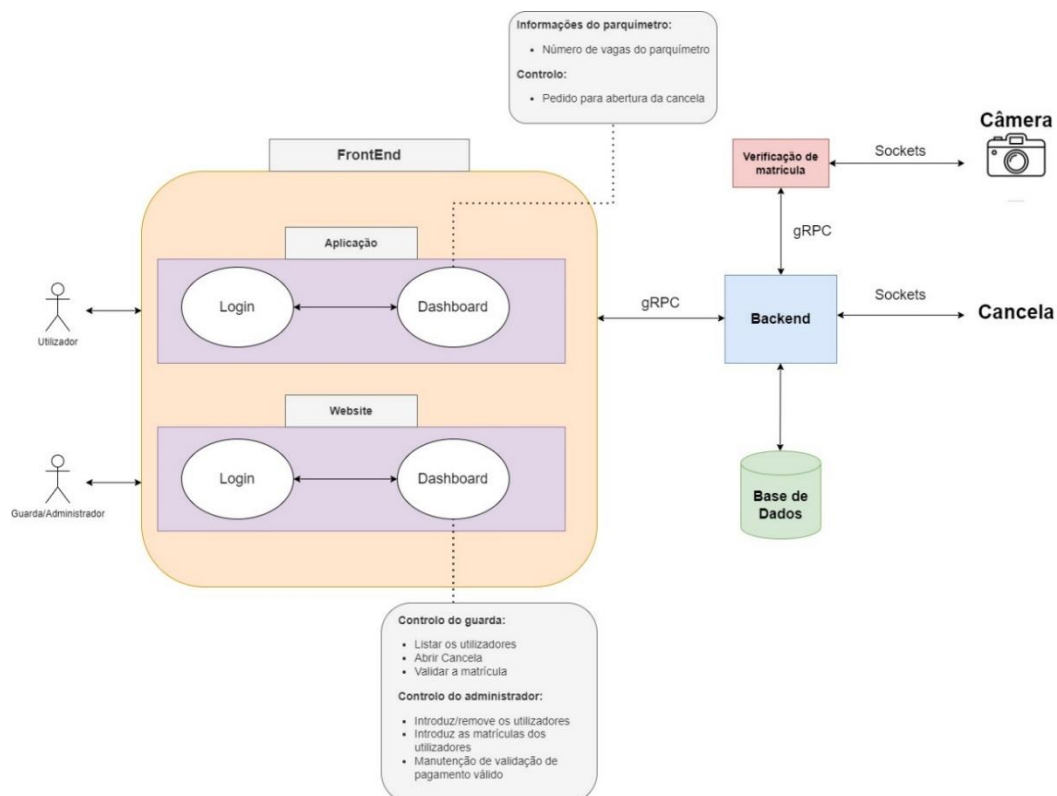


Figura 2 – Arquitetura de solução

Website

Mapa do Website

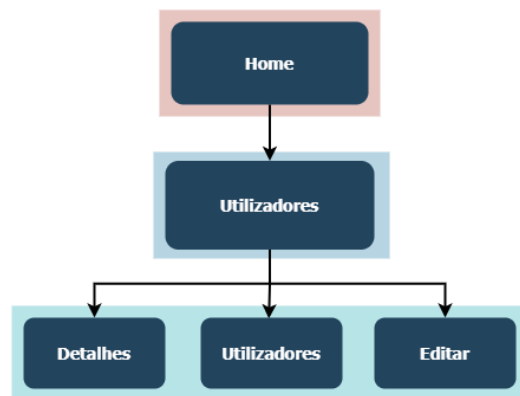


Figura 3 - Mapa do Website

5.2 Tecnologia

O website, começou por ser desenvolvido em Angular, mas após dificuldades em comunicarmo-nos através do gRPC com o servidor backend optamos por desenvolvê-lo usando a framework ASP.NET, criada pela Microsoft, pois era mais fácil de implementar uma comunicação entre o servidor com a utilização de gRPC.

Com este website, os guardas têm a opção de validar a entrada de utilizadores no parque e os administradores são capazes de ver e gerir informações acerca dos seus utilizadores.

Página Login

Esta é a primeira página que o guarda ou administrador tem acesso ao entrar no nosso website, onde através das suas credencias podem fazer login.

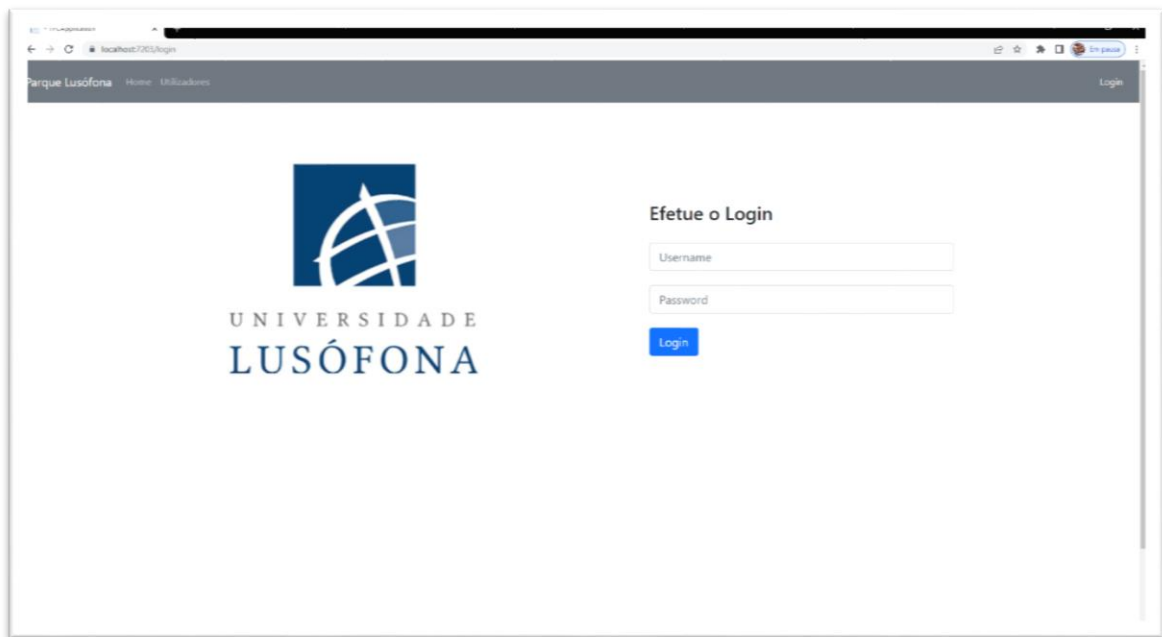


Figura 4 - Página Login

Quando o utilizador faz Login no website a sua palavra-passe vai ser encriptada com o padrão de criptografia avançada AES de modo a ter uma maior segurança contra-ataques.

```
1. referência
public static string Encrypt(string password)
{
    byte[] iv = new byte[16];
    byte[] array;
    using (Aes aes = Aes.Create())
    {
        aes.Key = Encoding.UTF8.GetBytes(key);
        aes.IV = iv;
        Console.WriteLine(iv.ToString());
        ICryptoTransform encryptor = aes.CreateEncryptor(aes.Key, aes.IV);
        using (MemoryStream ms = new MemoryStream())
        {
            using (CryptoStream cryptoStream = new CryptoStream((Stream)ms, encryptor, CryptoStreamMode.Write))
            {
                using (StreamWriter streamWriter = new StreamWriter((Stream)cryptoStream))
                {
                    streamWriter.Write(password);
                }
                array = ms.ToArray();
            }
        }
    }
    return Convert.ToBase64String(array);
}
```

Figura 5 - Código da Encriptação

Página Home

Sendo esta a página principal, o seu objetivo é apresentar uma imagem do veículo que pretende entrar ou sair do parque e se o veículo em questão tem acesso válido ou inválido para a cancela ser aberta.

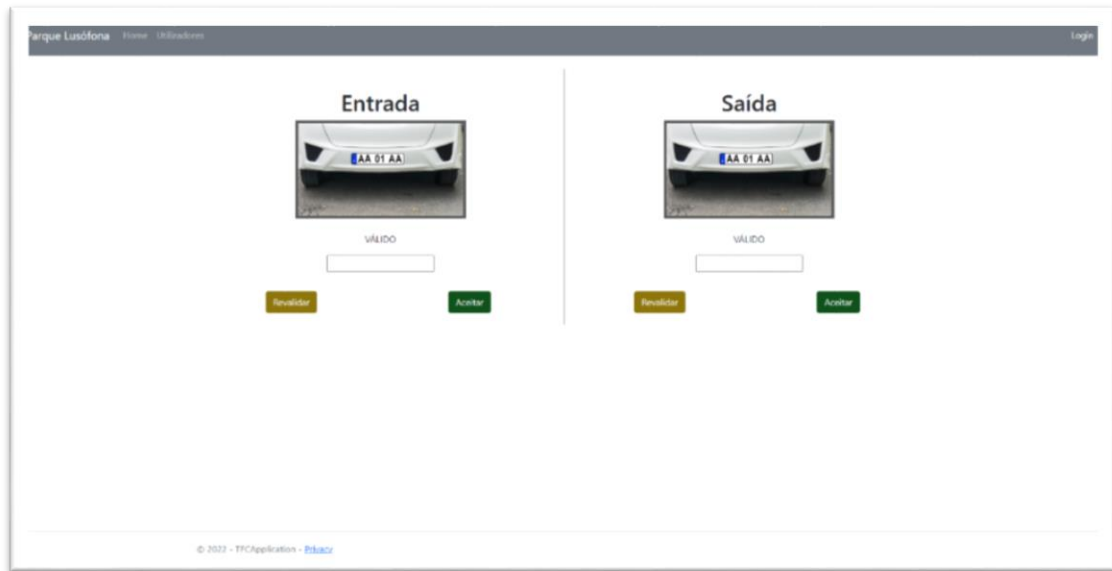


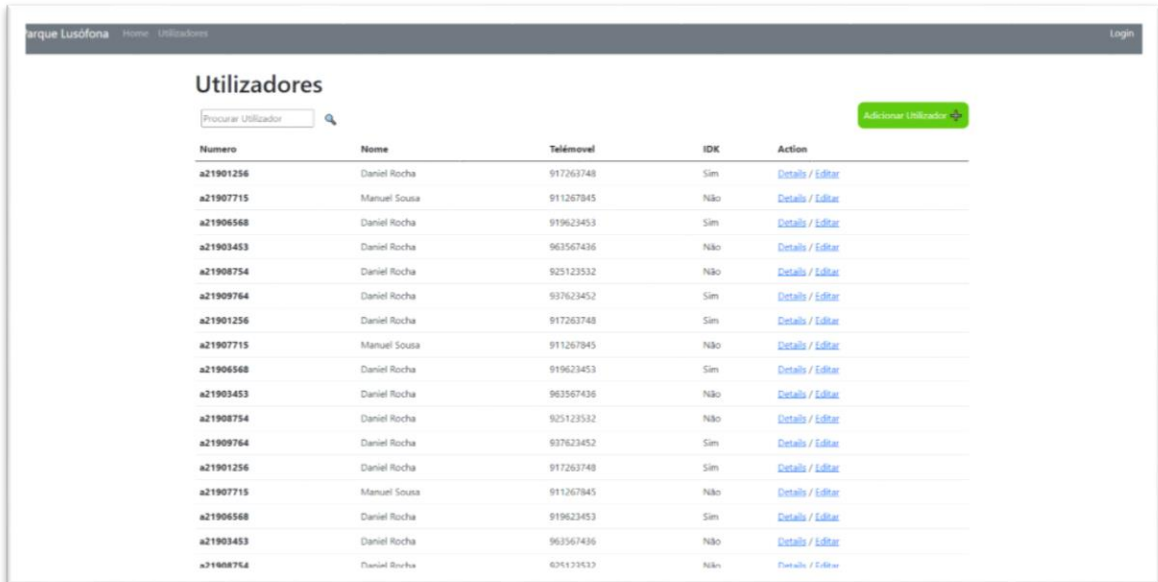
Figura 6 - Página Inicial

Caso a matrícula da imagem não corresponda há matrícula presente na caixa de texto, o guarda tem a capacidade de editar manualmente a matrícula e revalidar a mesma ao clicar no botão laranja “revalidar”, caso a matrícula esteja certa com a imagem o guarda após ver se o utilizador está ou não apto para entrar no parque clica no botão verde “Aceitar” para ser capaz de receber novos pedidos.

Também existe o botão “Procurar Utilizador” caso exista algum problema com um carro e o guarda precise de saber quem é o dono do mesmo, este botão irá fornecer o número e nome do utilizador.

Página Utilizadores

Apenas acessível pelo administrador esta página contém uma tabela com todos os utilizadores que já tiveram acesso ao parque, sendo possível pesquisar um utilizador em específico através do seu número, ver os detalhes e editar a informação dos mesmos.



Numero	Nome	Telémovel	IDK	Action
a21901256	Daniel Rocha	917263748	Sim	Detalho / Editar
a21907715	Manuel Sousa	911267945	Não	Detalho / Editar
a21906568	Daniel Rocha	919623453	Sim	Detalho / Editar
a21903453	Daniel Rocha	963567436	Não	Detalho / Editar
a21908754	Daniel Rocha	925123532	Não	Detalho / Editar
a21909764	Daniel Rocha	937623452	Sim	Detalho / Editar
a21901256	Daniel Rocha	917263748	Sim	Detalho / Editar
a21907715	Manuel Sousa	911267945	Não	Detalho / Editar
a21906568	Daniel Rocha	919623453	Sim	Detalho / Editar
a21903453	Daniel Rocha	963567436	Não	Detalho / Editar
a21908754	Daniel Rocha	925123532	Não	Detalho / Editar
a21909764	Daniel Rocha	937623452	Sim	Detalho / Editar
a21901256	Daniel Rocha	917263748	Sim	Detalho / Editar
a21907715	Manuel Sousa	911267945	Não	Detalho / Editar
a21906568	Daniel Rocha	919623453	Sim	Detalho / Editar
a21903453	Daniel Rocha	963567436	Não	Detalho / Editar
a21908754	Daniel Rocha	925123532	Não	Detalho / Editar

Figura 7 - Tabela de utilizadores

Ao clicar no botão verde com a etiqueta “Adicionar Utilizador” o administrador é capaz de adicionar um utilizador novo a ter acesso ao parque.

Caso o guarda tente aceder a esta página, visto que não tem permissões é redirecionado para uma página de erro



Figura 8 - Página Não Autorizada

Página Adicionar

Esta página permite ao administrador adicionar um novo utilizador ao parque.

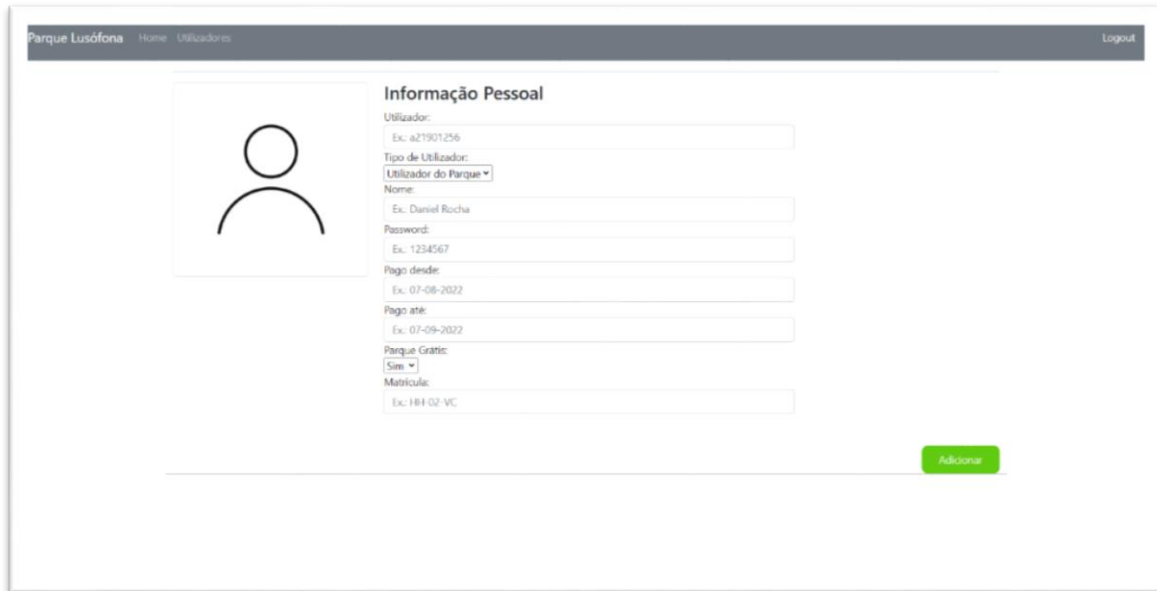


Figura 9 - Página Adicionar

Página Detalhes

Ao clicar no link “details” há frente de cada utilizador na página utilizadores temos acesso a esta página que mostra todos os detalhes sobre um utilizador em específico.

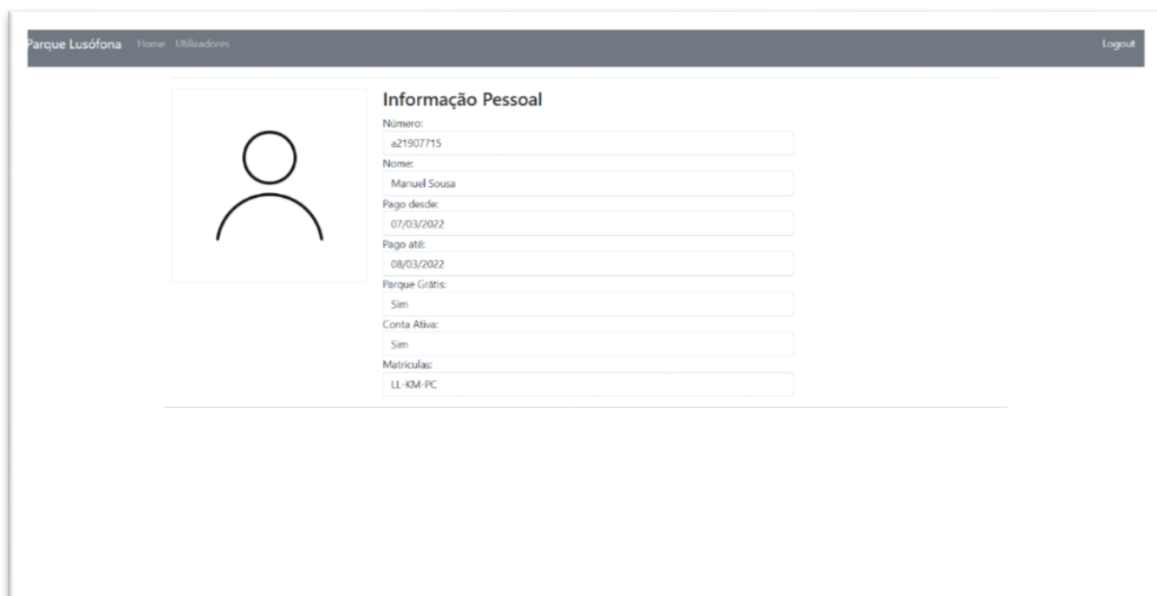


Figura 10 - Página Detalhes

Página Editar

Parecida há página detalhes esta é acessada através do clique do link “edit” na página utilizadores, onde poderão ser editadas informações acerca de um utilizador específico.

The screenshot shows a web interface for editing user information. At the top, there is a navigation bar with 'Parque Lusófona', 'Home', 'Utilizadores', and 'Logout'. The main content area is titled 'Informação Pessoal' and contains a form with the following fields: 'Utilizador' (a21907715), 'Nome' (Manuel Sousa), 'Pago desde' (07/03/2022), 'Pago até' (08/03/2022), 'Parque Grátis' (Sim), 'Conta Ativa' (Sim), and 'Matrículas' (LL-KM-PC). There is an 'Adicionar Matrícula' button next to the license field and a green 'Editar' button at the bottom right.

Figura 11 - Página Editar

Aplicação mobile

Mapa do mobile

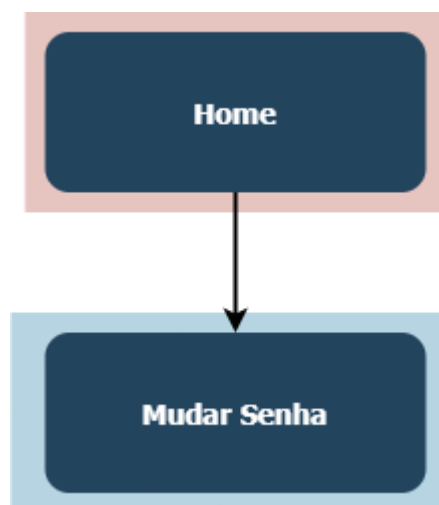


Figura 12 - Aplicação Mobile

5.3 Tecnologia mobile

Para a criação da aplicação mobile, tínhamos inicialmente a ideia de desenvolver utilizando a framework Ionic pela recomendação do orientador, mas depois, pelo facto de termos aprendido e desenvolvido uma aplicação com a framework Flutter na cadeira de computação móvel e, com alguma pesquisa, verificámos que seria mais fácil comunicar com o backend através de gRPC com essa framework, decidimos então que a aplicação iria ser desenvolvida nessa framework. Após alguns problemas de tentativa de comunicação com o servidor, decidimos mudar para a framework Xamarin que tem a mesma linguagem com o servidor. Após voltarmos a tentar comunicar com o servidor voltamos a ter dificuldades então decidimos mais uma vez tentar voltar ao Flutter e depois de algumas tentativas finalmente conseguimos estabelecer a ligação.

Com esta aplicação mobile, os utilizadores têm a opção de abrir as cancelas do parque para entrar ou sair, caso esteja pago e mudar a sua senha caso achem necessário.

Página Loading

Esta é a primeira página da aplicação mobile que serve para ler as informações num ficheiro de texto para no caso que o utilizador sair da aplicação, quando ele voltar ele não precise de voltar a fazer login a menos que faça logout da sua conta.

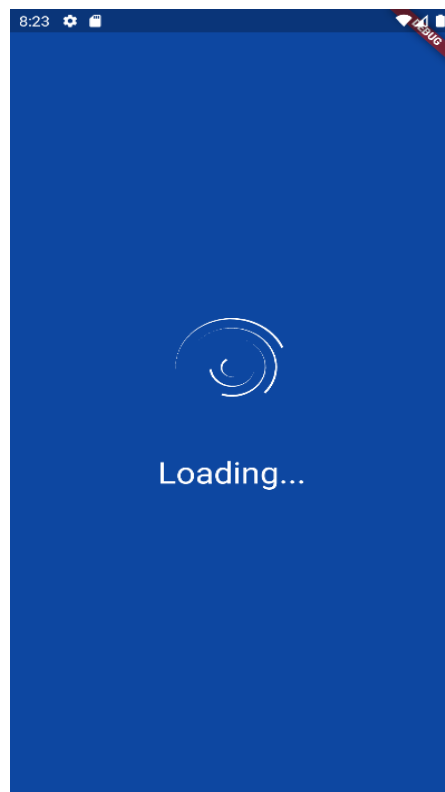


Figura 13 - Página Loading Mobile

Página Login

Esta é a página onde o utilizador escreve as suas credenciais para poder usar a aplicação de mobile.



Figura 14 - Página Login Mobile

Página Home

A página "Home" é a página principal onde o utilizador pode pedir para entrar ou sair do parque de estacionamento, ver a estimativa dos lugares vagos e aceder á página de mudar a sua senha.

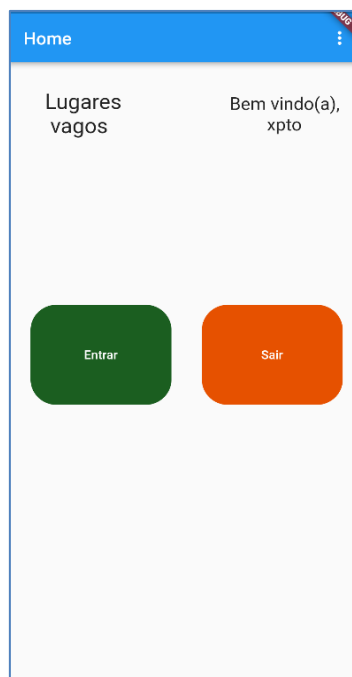


Figura 15 - Página Inicial

Página Mudar Senha

Esta página como o nome já indica serve para mudar a senha do utilizador já autenticado caso queira trocar de senha.



Figura 16 - Página Mudar Senha Mobile

Excertos de código do mobile

Neste excerto de código que está implementado na página “Loading” faz com que a aplicação guarde o nome do utilizador num ficheiro, então quando o utilizador sair da aplicação e voltar a entrar, se este não tiver saído da conta, em vez de voltar para a página “Login”, vai diretamente para a página inicial com a sua conta iniciada.

```
void _read() async {
  try {
    final Directory directory = await getApplicationDocumentsDirectory();
    final File file = File('${directory.path}/username.txt');
    bool fileExists = await File('${directory.path}/username.txt').exists();
    if (fileExists) {
      globals.text = await file.readAsString();
    } else {
      _write("");
    }
    await Future.delayed(const Duration(seconds: 2), (){});
    if(globals.text != "") {
      Navigator.pushNamed(context, '/home');
    } else {
      Navigator.pushNamed(context, '/login');
    }
  } catch (e) {
    print("Couldn't read file");
  }
}
```

Figura 17 - Função salvar utilizador

Este excerto de código seguinte foi implementado na página do “Login” está feito dentro do botão que faz o utilizador se autenticar com o seu nome de utilizador e com a palavra de passe, e faz comunicação com o servidor na porta 7024, enviando as informações com a palavra de passe encriptada. O servidor em seguida compara com a sua base de dados se as informações enviadas são equivalentes e, caso sejam manda uma validação para o utilizador entrar na página inicial da aplicação.

```

RoundedButton(
  text: "LOGIN",
  press: () async {
    final channel = ClientChannel(
      '10.0.2.2',
      port: 7024,
      options: const ChannelOptions(
        credentials: ChannelCredentials.insecure(),
      ), // ChannelOptions
    ); // ClientChannel
    final stub = LoginClient(channel);

    var login = await stub.loginRequest(LoginData()
      ..login = username
      ..password = _encrypt(pass)
    );
    if (login.loginEfetuado == true) {
      _write(username);
      Navigator.pushNamed(context, '/home');
    } else {
      setState(() {
        error = "Dados inválidos";
      });
    }
    await channel.shutdown();
  },
) // RoundedButton

```

Figura 18 - Botão Login Mobile

No botão “Entrar” e “Sair”, estando estes implementados na página inicial servem para quando o utilizador quiser entrar ou sair do parque, estes mandam um pedido ao servidor com duas variáveis, uma com o nome do utilizador para o servidor saber quem está a enviar pedido, e outra com um verdadeiro ou falso para saber se o pedido é de entrada ou saída. Após o servidor receber o pedido ele devolve uma mensagem para o utilizador saber qual o ponto de situação, sendo estes por exemplo se podem entrar, se não está pago, se alguém mandou um pedido primeiro, etc.

```
ButtonEnter(  
  text: "Entrar",  
  press: () async {  
    final channel = ClientChannel(  
      '10.0.2.2',  
      port: 7024,  
      options: const ChannelOptions(  
        credentials: ChannelCredentials.insecure(),  
      ), // ChannelOptions  
    ); // ClientChannel  
    final stub = MobileClient(channel);  
  
    try {  
      globals.mensagem = "A Processar...";  
      setState(() {  
      });  
      var response = await stub  
        .abrirCancela(Pedido()  
          ..cancelaEntrada = true  
          ..login = globals.text  
        );  
  
      globals.mensagem = response.toString();  
  
      setState(() {  
      });  
      print(globals.mensagem);  
    } catch (e) {  
      print('Caught error: $e');  
    }  
    await channel.shutdown();  
  }  
), // ButtonEnter
```

Figura 19 - Botão Entrar Mobile

```
ButtonExit(  
  text: "Sair",  
  press: () async {  
    final channel = ClientChannel(  
      '10.0.2.2',  
      port: 7024,  
      options: const ChannelOptions(  
        credentials: ChannelCredentials.insecure(),  
      ), // ChannelOptions  
    ); // ClientChannel  
    final stub = MobileClient(channel);  
  
    try {  
      globals.mensagem = "A Processar...";  
      setState(() {  
      });  
  
      var response = await stub  
        .abrirCancela(Pedido()  
          ..cancelaEntrada = false  
          ..login = globals.text  
        );  
  
      globals.mensagem = response.toString();  
      setState(() {  
      });  
    } catch (e) {  
      print('Caught error: $e');  
    }  
    await channel.shutdown();  
  }  
), // ButtonExit
```

Figura 20 - Botão Sair Mobile

Este excerto de código serve para encriptar a palavra de passe em AES, onde é encriptado da mesma forma que o website feito por este grupo, onde foi utilizado uma linguagem diferente.

```
_encrypt(String pass) {  
  
  final key = encrypt.Key.fromUtf8("tfc_2022_frontend_daniel_manuel1");  
  final iv = encrypt.IV.fromLength(16);  
  final encrypter = encrypt.Encrypter(encrypt.AES(key, mode: encrypt.AESMode.cbc));  
  
  final encrypted = encrypter.encrypt(pass, iv: iv);  
  return encrypted.base64;  
}
```

Figura 21 - Função Encriptar Mobile

Esta parte do código implementado na página de mudar a senha, estabelece conexão com o servidor e envia o nome de utilizador, senha antiga e a senha nova encriptadas, depois de ter sido feita a verificação entre os campos da “senha nova” e “confirmar senha nova” serem idênticos. Depois o servidor verifica na sua base de dados se a senha antiga coincide com a senha guardada e envia um positivo ou negativo consoante o resultado. Caso seja positivo aparece uma mensagem a dizer que a senha foi alterada com sucesso e depois vai para a página inicial, ou caso seja negativo aparece a dizer que a senha não foi alterada por dados incorretos.

```
- RoundedButton(  
  text: "MUDAR PASSWORD",  
  press: () async {  
    final channel = ClientChannel(  
      '10.0.2.2',  
      port: 7024,  
      options: const ChannelOptions(  
        credentials: ChannelCredentials.insecure(),  
      ), // ChannelOptions  
    ); // ClientChannel  
    final stub = LoginClient(channel);  
  
    if (newpass == newpass2) {  
      var login = await stub.changePassword(  
        NewLoginData()  
          ..login = username  
          ..oldPass = _encrypt(oldpass)  
          ..newPass = _encrypt(newpass)  
      );  
  
      if(login.passAlterada == true) {  
        setState(() {  
          mensagemPass = "Senha alterada com sucesso";  
        });  
  
        await Future.delayed(const Duration(seconds: 1), (){});  
        Navigator.pushNamed(context, '/home');  
      } else {  
        setState(() {  
          mensagemPass = "Senha não alterada por dados incorretos";  
        });  
      }  
    } else {  
      setState(() {  
        mensagemPass = "Senha de confirmação não coincide";  
      });  
    }  
    await channel.shutdown();  
  },  
), // RoundedButton
```

Figura 22 - Botão Mudar Password

Para a entrada do utilizador no parque optámos por desenvolver uma aplicação mobile, pois ao analisar muitas opções, assim como tecnologia NFC ou Contactless, percebemos que pelo lado financeiro fica mais barato desenvolver a aplicação mobile, visto que toda a gente usa um smartphone atualmente e no caso de cartões NFC ou Contactless tínhamos de gastar com a criação dos mesmos e com um sensor que os lesse, por outro lado como essas tecnologias implicariam que a pessoa tivesse de sair do carro ou baixar a janela para se esticar e encostar o cartão ao sensor, o que não seria bom em dias de chuva, a tecnologia mobile pareceu-nos a melhor opção.

Para a criação da aplicação mobile, tínhamos inicialmente a ideia de a desenvolver utilizando a framework Ionic, mas pelo facto de que na cadeira de computação móvel termos desenvolvido uma aplicação com a framework Flutter e, com alguma pesquisa, verificamos que seria mais fácil comunicar com o backend através de gRPC com essa framework decidimos que a aplicação iria ser desenvolvida usando essa framework. A aplicação vai consistir em dois botões, um para entrar e outro para sair do parque de estacionamento e vai mostrar a quantidade lugares que ainda estão vagos.

Da parte do website, apesar de termos começado a desenvolver usando a framework angular, estávamos com dificuldades em usar o gRPC para nos comunicarmos com o backend, decidimos então desenvolvê-lo usando a framework ASP.NET, criada pela Microsoft, pois foi mais fácil de estabelecer comunicação com o servidor backend. Ainda no website, os guardas vão ter a opção de listar todos os utilizadores com acesso ao parque assim como permitir a sua entrada; caso haja algum erro com a leitura da matrícula estes podem inseri-la manualmente. Todas as trocas de comunicação entre front-end o backend irão ser realizadas através de gRPC.

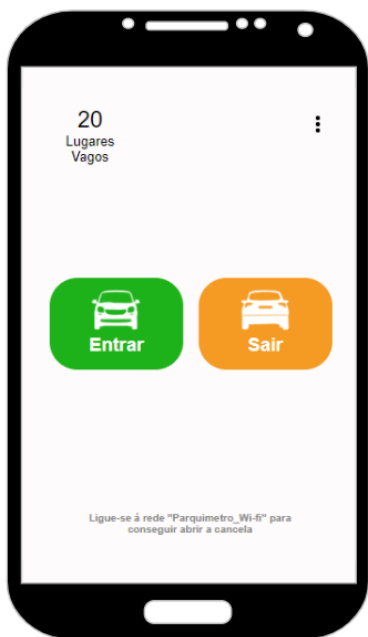


Figura 26 - Protótipo da aplicação mobile



Figura 23 - Axure: Ferramenta usada para o desenvolvimento do mockup



Figura 24 – ASP.NET: Framework usada para o desenvolvimento do website



Figura 25 - Flutter: Ferramenta usada para desenvolvimento da app mobile

5.4. Workflow de entrada

Ao chegar à cancela do parque de estacionamento da Universidade Lusófona, o utilizador que tem a intenção de entrar no parque abre a aplicação mobile desenvolvida pela nossa parte e, estando autenticado, autenticação essa feita através de comunicação entre a aplicação com o servidor backend, pressiona no botão de entrada do parque.

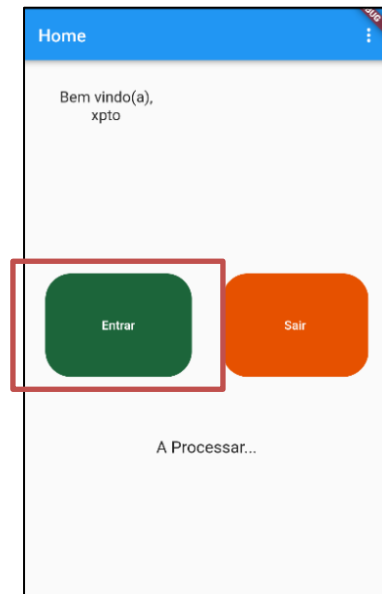


Figura 27 - A Processar pedido
Aplicação Mobile

Ao pressionar esse botão é enviado um pedido de entrada no parque ao servidor backend, onde o mesmo, ao receber este pedido, irá verificar se é de entrada ou de saída, verifica também, no caso de ser de entrada, se o utilizador não se encontra atualmente no parque e se pagou o respetivo lugar.



Figura 28 – Pedido processado
Aplicação Mobile

Após esta verificação, o backend faz um pedido ao website frontend para saber se não existe nenhuma entrada a ser verificada no momento. No caso de não existir o backend envia um pedido de matrícula para a deteção de matrículas, esta tira foto da matrícula, usa o algoritmo para detetar e ler a mesma e envia para o backend os dados da foto e da matrícula. O backend, posteriormente guarda a foto no servidor, verifica se a matrícula existe no sistema, se corresponde ao utilizador que pretende entrar e por fim envia os dados que recebeu da deteção de matrículas e a validação para o website de modo a serem apresentados ao segurança e este possa concluir a verificação.

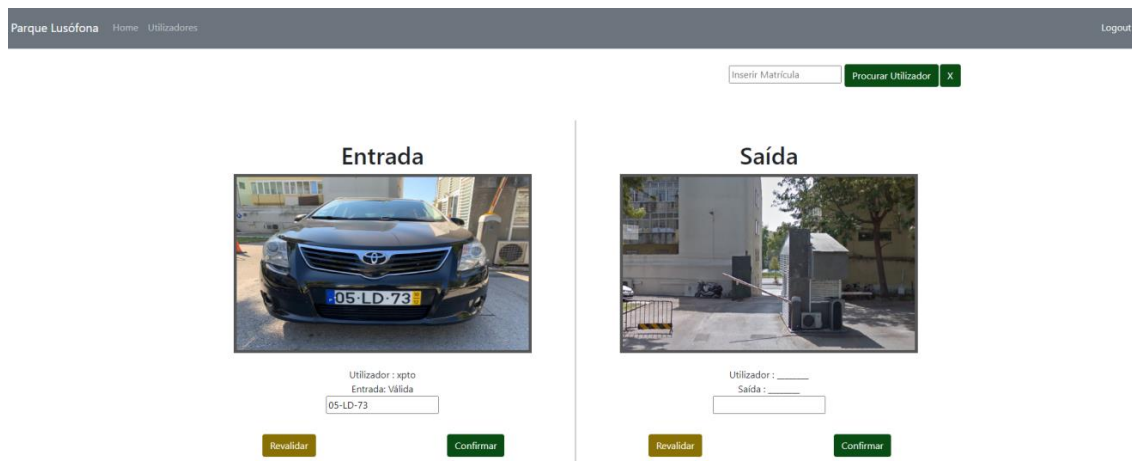


Figura 29 - Pedido entrada Aplicação Website

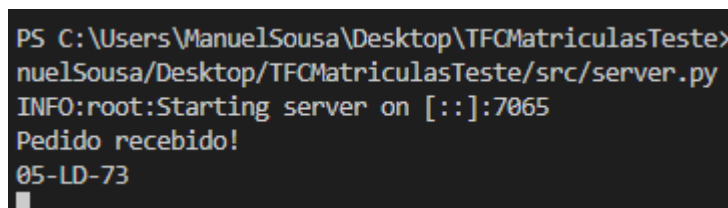


Figura 30 - Consola Deteção de Matrículas

No caso de a matrícula apresentada ao segurança corresponder com a matrícula da foto e a entrada ser validada pelo backend o segurança pode confirmar a entrada do utilizador no parque, caso contrário caso a matrícula apresentada esteja em desacordo com a matrícula da foto, o segurança tem a opção de inseri-la manualmente e revalidar o pedido de entrada. Após isto o backend envia de novo a validação de entrada e no caso de ser válido pode ser aberta a cancela permitindo a entrada ao utilizador.

Para o momento da saída do parque o processo irá ser semelhante.

6 Método e planeamento

Para a fase de desenvolvimento, foi realizado um diagrama de Gantt feito em PowerPoint da nossa estimativa sobre a quantidade de tempo que iríamos gastar para a conclusão de cada fase do projeto. As fases que sentimos mais dificuldade foram as fases de desenvolvimento do website e da app mobile, em específico a ligação com gRPC com o servidor, que nos levou bastante tempo para conseguirmos estabelecer uma ligação, até ao ponto que tivemos de mudar linguagens/frameworks onde estivemos a desenvolver aplicações, para ter uma melhor compatibilidade com a framework gRPC. Uma vez que conseguimos estabelecer a ligação, tornou-se bastante fácil começar a trocar dados com o servidor, mas surgiu um novo problema, que foi tentar adaptar-nos à nova framework do website, pois como já referido anteriormente no relatório, onde inicialmente começamos por trabalhar em angular, já tínhamos feito bastante em angular e aprendido sobre o mesmo, onde tivemos de descartar o que já tínhamos feito e aprender outra do início.

Nesta fase do projeto estamos em desenvolvimento do website onde já existe comunicação com o servidor backend. Da parte da aplicação mobile só vai ser começado a ser feito o desenvolvimento da mesma após a entrega deste relatório.

Como tivemos problemas com o desenvolvimento do website em Angular e decidimos mudar a framework para o desenvolver o mesmo ainda não está completo sendo que neste momento já é possível, através de comunicação com o backend, a autenticação no mesmo e detetar se existe algum dado a ser verificado pelo segurança.

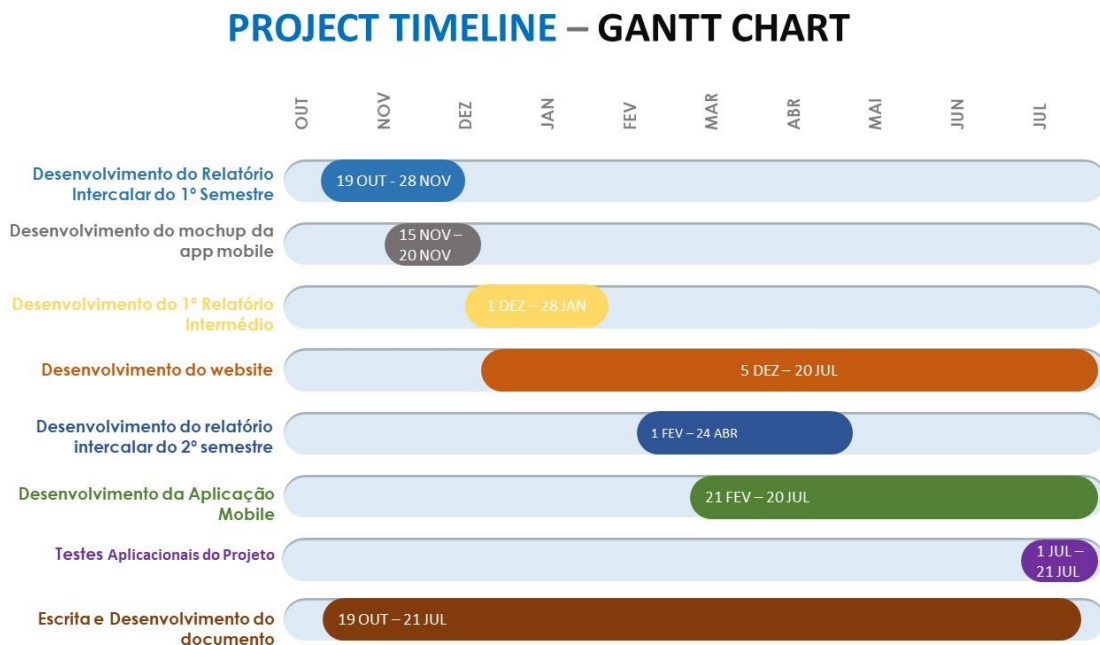


Figura 31 - Diagrama de Gantt

7 Resultados

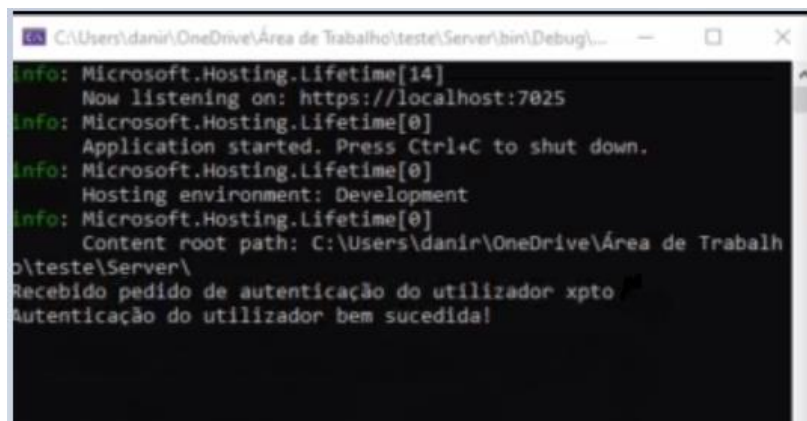
7.1. Processo de Login

O segurança introduz as credencias definidas pelo servidor backend onde o nome de utilizador é admin e a password é admin.



Figura 32 - Página Login

É feito um pedido de autenticação ao servidor que retorna ao website a validação.



```
C:\Users\danir\OneDrive\Área de Trabalho\teste\Server\bin\Debug\...  
Info: Microsoft.Hosting.Lifetime[14]  
Now listening on: https://localhost:7025  
Info: Microsoft.Hosting.Lifetime[0]  
Application started. Press Ctrl+C to shut down.  
Info: Microsoft.Hosting.Lifetime[0]  
Hosting environment: Development  
Info: Microsoft.Hosting.Lifetime[0]  
Content root path: C:\Users\danir\OneDrive\Área de Trabalho\teste\Server\  
Recebido pedido de autenticação do utilizador xpto  
Autenticação do utilizador bem sucedida!
```

Figura 33 - Servidor a validar utilizador

Caso a autenticação seja válida o segurança é redirecionado para a página de validação de entradas e saídas de utilizadores no parque.

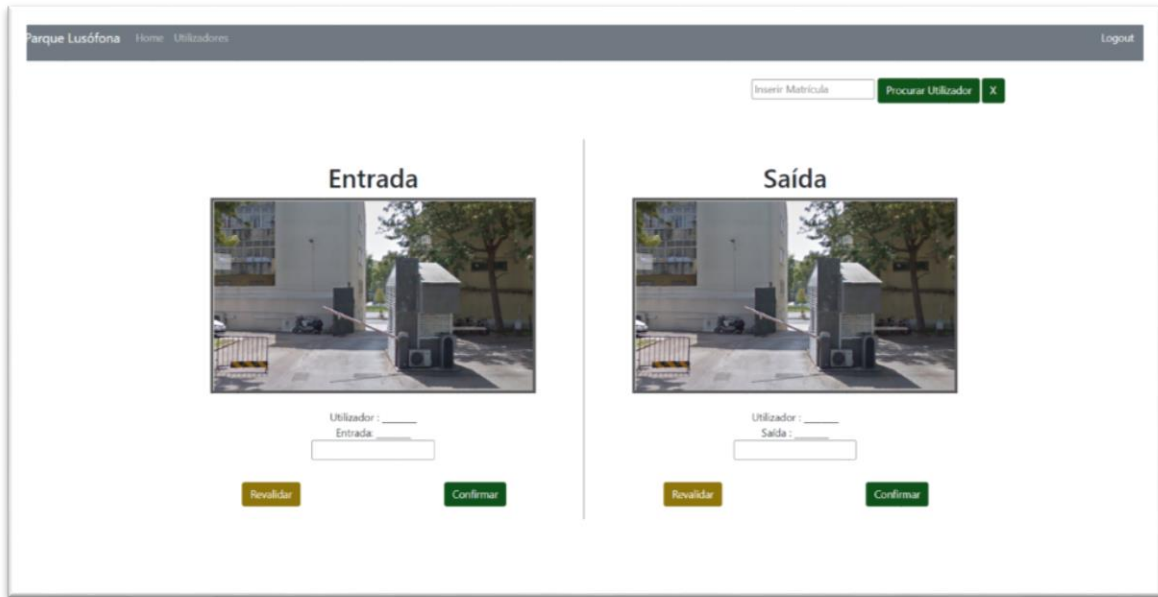


Figura 34 - Página Inicial

No caso do mobile o utilizador introduz as credenciais da sua conta, caso estejam corretas é feito um pedido de autenticação ao servidor que retorna à aplicação a validação. Caso a autenticação seja válida o utilizador é redirecionado para página de inicial.



Figura 35 - Página Login Mobile



Figura 37 - Página Inicial Mobile

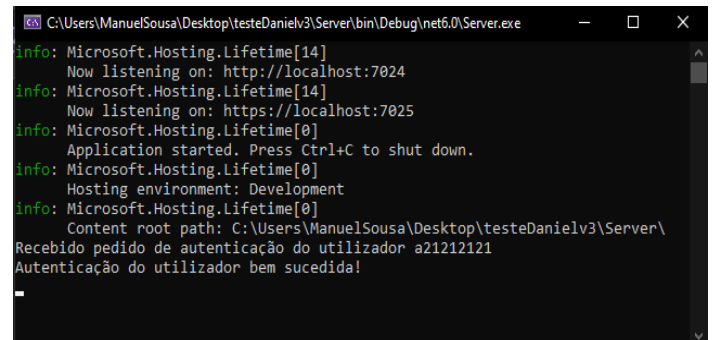


Figura 36 - Servidor a validar utilizador - Mobile

7.2. Processo de Verificação

Após o utilizador ter feito login na aplicação, ao clicar “Entrar” vai ser enviada informação ao backend, que através de um pedido de entrada á deteção de matrículas, envia ao website a informação acerca do carro que pretende entrar.

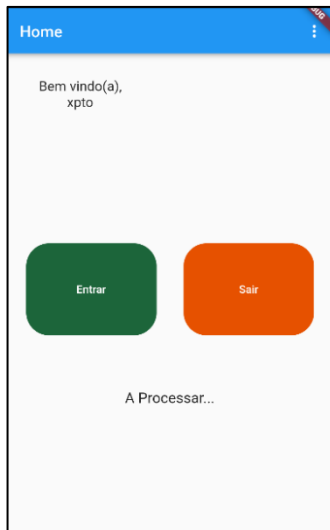


Figura 39 – Processar Pedido



Figura 38 - Validação da Matricula

Após receber o pedido o guarda poderá revalidar ou confirmar o mesmo, durante este processo caso algum utilizador esteja a tentar fazer um pedido de entrada no parque, aparece uma mensagem que diz:

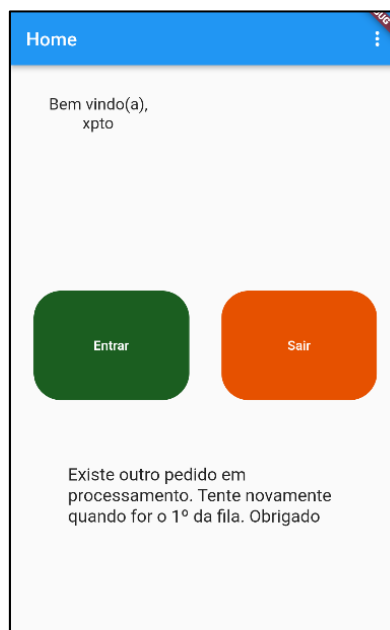
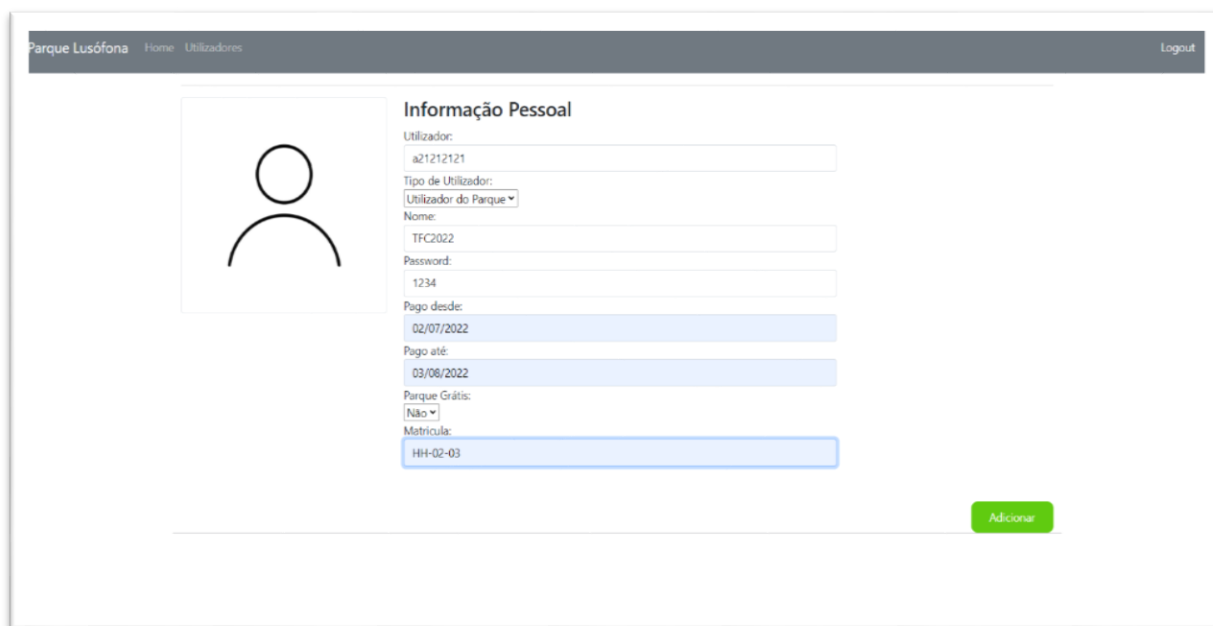


Figura 40 - Outro pedido em processamento

7.3. Processo de Adicionar ou Editar Utilizador

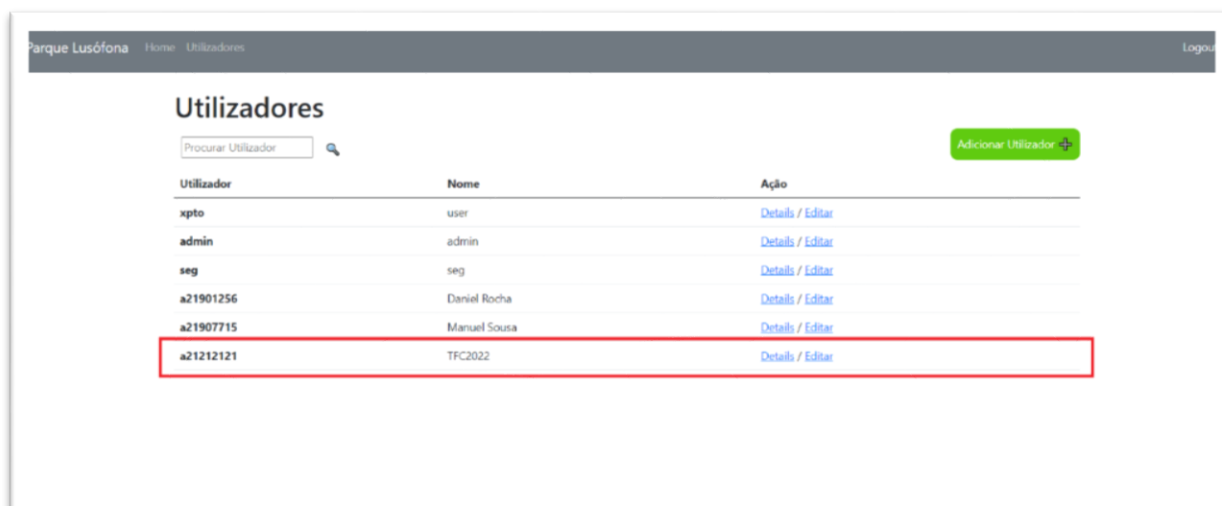
Neste teste o administrador está a adicionar um utilizador novo. Começa por entrar na página de adicionar utilizador e introduz a informação acerca do novo utilizador.



The screenshot shows a web interface for adding a user. The header includes 'Parque Lusófona', 'Home', 'Utilizadores', and 'Logout'. The main content area is titled 'Informação Pessoal' and contains several input fields: 'Utilizador:' (a21212121), 'Tipo de Utilizador:' (Utilizador do Parque), 'Nome:' (TFC2022), 'Password:' (1234), 'Pago desde:' (02/07/2022), 'Pago até:' (03/08/2022), 'Parque Grátis:' (Não), and 'Matricula:' (HH-02-03). A green 'Adicione' button is located at the bottom right of the form.

Figura 41 - Adicionar utilizador

Depois de introduzir todos os campos e clicar no botão adicionar podemos confirmar que o novo utilizador já se encontra na lista de utilizadores.



The screenshot shows a list of users. The header includes 'Parque Lusófona', 'Home', 'Utilizadores', and 'Logout'. The main content area is titled 'Utilizadores' and contains a search bar 'Procurar Utilizador' and a green 'Adicionar Utilizador' button. The table below lists the users:

Utilizador	Nome	Ação
xpto	user	Details / Editar
admin	admin	Details / Editar
seg	seg	Details / Editar
a21901256	Daniel Rocha	Details / Editar
a21907715	Manuel Sousa	Details / Editar
a21212121	TFC2022	Details / Editar

Figura 42 - Utilizador Adicionado

Caso o administrador queira editar um utilizador clica no botão editar, faz as devidas alterações e confirma as alterações no botão editar.



The screenshot shows a web form titled "Informação Pessoal" for editing a user. On the left is a placeholder for a profile picture. The form fields are as follows:

Field Label	Value
Utilizador:	a21212121
Nome:	TFC2022
Pago desde:	02/01/2022
Pago até:	03/01/2022
Parque Grátis:	Sim
Conta Ativa:	Sim
Matriculas:	05-LD-73

At the bottom right of the form is a green "Editar" button. Below the "Matriculas" field is a small "Adicionar Matricula" button.

Figura 43 - Editar Utilizador

8 Conclusão e trabalhos futuros

Esta unidade curricular permitiu-nos ter uma melhor noção do que é realizar um projeto de uma grande complexidade com um nível mais próximo do empresarial, comparando com os outros trabalhos que desenvolvemos ao longo desta licenciatura. Através das experiências positivas e negativas que tivemos no decorrer deste ano letivo a fazer este projeto conseguimos ver que competências, como ter uma boa organização e boa comunicação entre todos os elementos, são essenciais para o bom funcionamento do projeto. Procurar várias formas e tecnologias para resolver certos problemas, avaliar essas e escolher a que nos parecia melhor no nosso projeto. Também nos ajudou a melhorar o nosso conhecimento acerca do mundo tecnológico e a melhorar a nossa autonomia o que será uma mais valia para o futuro quando entrarmos no mercado de trabalho.

Apesar de concluirmos este projeto existem alguns aspetos que poderão ser melhorados ou acrescentados futuramente, assim como incorporar os dados dos utilizadores da Lusófona na base de dados, melhorar o design da aplicação mobile e da aplicação web e também implementar novas funcionalidades que poderão necessárias para o bom funcionamento do projeto num ambiente real.

Bibliografia

- [DEISI21] DEISI, Regulamento de Trabalho Final de Curso, Set. 2021.
- [TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.
- [ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt,
acedido em Out. 2021.
- [Vicrodel] Vicrodel, [Vicrodel | Sistemas Electrónicos Lda](#), acessido em Nov. 25 2021.
- [Prevalta] Prevalta, [CAMARAS DE MATRICULAS ABERT. AUTOMATICA | Prevalta](#), acessido em
Nov. 25 2021.
- [gRPC] © 2022 gRPC Authors, [gRPC](#)

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso