

Escola Secundária de Pinheiro e Rosa

Curso Profissional de Técnico de Gestão e Programação de Sistemas
Informáticos

2016/2019



Lowkey

Relatório da Prova de Aptidão Profissional

Tiago Manuel Pereira Miguel

Faro, junho de 2019

Escola Secundária de Pinheiro e Rosa

Curso Profissional de Técnico de Gestão e Programação de Sistemas
Informáticos

2016/2019

Lowkey

Relatório da Prova de Aptidão Profissional

Tiago Manuel Pereira Miguel

Faro, junho de 2019

AGRADECIMENTOS

Agradeço a todos aqueles que de alguma forma contribuíram para a conclusão deste trabalho.

Ao meu professor, Paulo Sopa, pela atenção, pela oportunidade de aprendizagem, compreensão demonstrados aquando as minhas falhas.

Obrigado por ter cooperado comigo.

Aos meus pais, que estiveram sempre ao meu lado, a incentivar-me a ser melhor, enquanto estudante, profissional e acima de tudo enquanto ser humano.

A todos os meus amigos e colegas, em especial ao Carlos Ferreira e ao Gonçalo Mateus, por todo o apoio, companheirismo e palavras de incentivo.

RESUMO

O presente relatório descreve o trabalho curricular realizado no ano do curso de TGPSI, (Técnico de Gestão e Programação de Sistemas Informáticos), na Escola Pinheiro e Rosa de Faro.

Durante o período de construção e desenvolvimento do projeto, várias atividades foram trabalhadas, das quais destaco: o aprofundamento dos conhecimentos adquiridos durante a formação em sala e de trabalhos desenvolvidos como este projeto.

Este projeto, consiste na criação de uma aplicação web de comunicação, denominada de “Lowkey”, cujo o objetivo é a troca de mensagens em tempo-real ente utilizadores de forma gratuita.

A aplicação é baseada em *node.js*, uma *framework* voltada para a programação web, que serviu como servidor base do projeto.

PALAVRAS-CHAVE

NodeJS; MySQL; Javascript; Handlebars; Aplicação; Web; Mensagens; Comunicação; Tempo-Real;

ÍNDICE GERAL

1. INTRODUÇÃO.....	1
1.1 Definição de Objetivos.....	1
1.2 Fundamentação da Escolha do Projeto	1
1.3 Enquadramento do Projeto.....	2
1.4 Estrutura do Relatório.....	2
 2. DESENVOLVIMENTO	3
2.1 Cronograma de Atividades.....	3
2.2 Fundamentação Teórica	5
2.2.1 Descrição do Sistema	5
2.2.2 Código-fonte Desenvolvido	5
2.2.3 Modelo E-R	7
2.2.4 Tabela de Base de Dados e Relacionamentos.....	8
2.2.5 Tipos de Dados Escolhidos (SQL)	9
2.2.6 Software Utilizado	14
2.3 Caracterização Final do Produto.....	15
2.4 Reflexão Crítica.....	33
 3. CONCLUSÃO	34
3.1 Dificuldades	34
3.2 Problemas e Obstáculos	34
3.3 Soluções Encontradas	34
 4. BIBLIOGRAFIA / WEBGRAFIA.....	35

ÍNDICE DE FIGURAS

1. Figura da Inicialização do Servidor	5
2. Figura da Organização das Rotas.....	6
3. Figura da Organização das Funções dos Sockets	6
4. Figura do Modelo Entidade-Relação	7
5. Figura da Página Inicial	16
6. Figura do Formulário de Registo.....	17
7. Figura da Mensagem de Confirmação da Conta	17
8. Figura do Formulário de Início de Sessão	18
9. Figura do Formulário de Recuperação da Conta	18
10. Figura do Formulário do Segundo Fator de Autenticação	19
11. Figura da Conta Verificada	19
12. Figura da Mensagem de Recuperação da Conta	20
13. Figura da Conta Recuperada.....	20
14. Figura da Conta Eliminada.....	21
15. Figura de Erro de Verificação.....	22
16. Figura da página Inicial da Aplicação.....	23
17. Figura de Secção de Amigos	23
18. Figura do Formulário para Adicionar Amigos	24
19. Figura da Informação de Adicionar um Amigo	24
20. Figura da Tela para aceitar um amigo.....	25
21. Figura da Tela de Conversas	25
22. Figura. da Informação dos Amigos	26
23. Figura da Informação de Mensagem.....	26
24. Figura da Informação de Estado	27
25. Figura do Menu de Enviar Ficheiros.....	27
26. Figura da Disposição do Ficheiro Enviado	28
27. Figura do Menu de Emojis	28
28. Figura do Menu de Gifs.....	29
29. Figura da Página de Definições da Conta	29
30. Figura da Segurança da Conta	30
31. Figura do Formulário de Ativação do Segundo Fator de Autenticação	30
32. Figura da Informação de Ativação do Segundo Fator de Autenticação	31
33. Figura das Sessões Ativas.....	32

ÍNDICE DE TABELAS

1. Tabela do cronograma de atividades	3
---	---

1. INTRODUÇÃO

O tema escolhido para a Prova de Aptidão Profissional, foi pensado devido à sua complexidade e à grande utilização de softwares. Têm como objetivo a comunicação entre diversos utilizadores. Sendo assim, para o desenvolvimento e elaboração do projeto foi escolhido o tema sobre a comunicação em tempo real, com um ou mais amigos, por mensagem, criando assim a aplicação web denominada de: “Lowkey”.

O projeto realizado, tenta reproduzir uma mistura de aspetos positivos e úteis das aplicações mais populares, tanto para dispositivos móveis, como para computadores, que visa manter a mesma ideia de comunicação, mas, com o objetivo de funcionar em qualquer dispositivo com acesso à internet, e é baseado num servidor em “Node.js”, designado para o desenvolvimento de aplicações de rede.

Na elaboração deste projeto, foram utilizadas, diversas novas tecnologias, para a construção do website e funcionamento, sendo elas: Adobe Photoshop CC, Adobe XD CC, Node.js, Postman, Visual Studio Code e o Xampp Control Panel, que serão aprofundadas na secção de desenvolvimento do projeto.

1.1 DEFINIÇÃO DE OBJETIVOS

Este projeto, tem como principal objetivo a aprendizagem de novas tecnologias e um estudo aprofundado sobre as linguagens e técnicas de programação, usadas na realização deste mesmo projeto.

1.2 FUNDAMENTAÇÃO DA ESCOLHA DO PROJETO

A PAP (Projeto de Aptidão Profissional) é um projeto pessoal, estruturante e onde é preciso demonstrar os conhecimentos adquiridos ao longo dos três anos de curso, cujo objetivo foi a realização de uma aplicação web de mensagens entre utilizadores. Este tema foi escolhido devido à sua complexidade, e à possibilidade de aprendizagem de novas tecnologias, linguagens de programação e novas vertentes da programação. Para isso, foi criada a aplicação “Lowkey”, que abrange esse mesmo tema e tem uma ligação a uma base de dados.

1.3 ENQUADRAMENTO DO PROJETO

Como referido anteriormente, o projeto baseia-se na comunicação em tempo-real entre utilizadores. Para isso o “Lowkey” engloba quatro funcionalidades essenciais:

o sistema de autenticação; o sistema de amizades; o sistema de definições da conta; sistema central de comunicação, e por último o sistema de administração da aplicação.

O sistema de autenticação, permite ao utilizador registar-se, iniciar sessão, verificar a conta, recuperar a conta e até mesmo eliminar a conta caso tenha sido um engano.

Já dentro da aplicação em si, o utilizador pode usar o sistema de amizades, para: enviar um pedido de amizade, rejeitar ou aceitar um pedido de amizade, e remover um amigo da sua lista de amigos.

O sistema de definições da conta permite que o utilizador consiga alterar informações sobre a sua conta, como: o nome, o nome de utilizador, o email e a palavra-passe.

Para além disso, o utilizador tem a possibilidade de ativar/desativar funções adicionais, como por exemplo: o segundo fator de autenticação, também conhecido por: “2fa” e eliminar a conta. O utilizador também tem informações sobre as sessões ativas com a sua conta.

O sistema central de comunicação, dá a possibilidade de o utilizador conseguir conversar com os seus amigos, podendo enviar “emojis”, links, imagens e outros ficheiros, que têm uma pré-visualização carregada no “chat”, a caixa de conversa com o amigo.

1.4 ESTRUTURA DO RELATÓRIO

O relatório está dividido em três partes:

- *Introdução*, onde é apresentado o resumo do trabalho realizado, os objetivos a alcançar, as finalidades do mesmo e o enquadramento do projeto para a área de *desenvolvimento*.
- *Desenvolvimento*, onde é apresentado o plano inicial do projeto, os objetivos a atingir, seguido da construção da base de dados e do website;

- *Conclusão*, que retêm a conclusão final do projeto, bem como as dificuldades encontradas ao longo do seu desenvolvimento, seguidas das suas resoluções e obstáculos.

2. DESENVOLVIMENTO

2.1 CRONOGRAMA DE ATIVIDADES

Para a realização do projeto de aptidão profissional, foi preciso uma divisão lógica, de fácil compreensão e que desse o tempo necessário para terminar cada parte.

Com isso, foi criado o seguinte cronograma, que serviu de orientação durante a realização do projeto:

ATIVIDADES	TAREFAS	CALENDARIZAÇÃO
Designs das estruturas das páginas web	Construção de design para todas as páginas web do projeto	Novembro
Sistema de Autenticação	Desenvolver um sistema de autenticação completo	Janeiro
Sistema de Amizades	Criar um sistema de amigos entre utilizadores	Fevereiro
Definições de Perfil	Página onde o utilizador pode alterar as suas definições de conta	Fevereiro
Sistema de Mensagens	Desenvolver a base para o processo de desenvolvimento do sistema de mensagens e de comunicação com o servidor em “tempo real”	Março

Resolução de Erros	Resolver erros aparentes e erros de visualização	Março
Atualizações e Melhorias	Atualizações e melhorias a nível do servidor	Abril
Aplicação para Computador	Desenvolvida uma aplicação em “electron” para as seguintes plataformas: Windows, Linux e Mac	Abril
Sistema de Ficheiros	Desenvolvimento do sistema de “upload” de ficheiros em tempo-real	Maio
Resolução de Problemas	Resolução de problemas de aparência do website e do sistema de ficheiros	Maio
Sincronização “Cross-browser”	Desenvolvimento de um sistema que permite que todas as contas abertas em simultâneo recebam atualizações	Maio
Limpeza de Código	Limpeza de código Código comentado	Junho

Tabela 1 – Cronograma de Atividades

2.2 FUNDAMENTAÇÃO TEÓRICA

2.2.1 DESCRIÇÃO DO SISTEMA

A aplicação desenvolvida, trata-se de uma aplicação comunicação / troca de mensagens entre utilizadores.

Para isso a aplicação trabalha com os seguintes dados:

- Contas;
- Códigos de Segurança;
- Códigos de Ativação / Recuperação;
- Mensagens / Conversas;
- Lista de Amigos;
- Ficheiros:
- Ficheiros em Conversas;
- Fotos de Perfil;

Mais abaixo cada dado é apresentado individualmente.

2.2.2 CÓDIGO-FONTE DESENVOLVIDO

Abaixo encontra-se alguns exemplos de código, desenvolvidos durante o período de desenvolvimento da aplicação web.

Inicialização de um servidor em node.js com a *framework express*, que permite a criação do servidor de forma simples e segura de continuar a programa. Para além disso ainda prepara os métodos para enviar informações para o servidor, como por exemplo: os dados de um formulário.

```
const express = require('express');
const app     = express();

const server  = require('http').createServer(app);
```

Figura 1 – Inicialização do Servidor

Organização das rotas da API do servidor. Todos os métodos POST estão organizados por o seguinte código:

```
const api    = require('./src/router/api/v1');
const admin  = require('./src/router/admin');
const message = require('./src/router/message');
const index  = require('./src/router/landing');

app.use('/static' , express.static(path.join(__dirname, './src/router/static')));
app.use('/api'    , api);
app.use('/admin'  , admin);
app.use('/mensagens', message);
app.use('/',      index);
```

Figura 2 – Organização das Rotas

Organização das funções executadas, quando um socket envia informações para o servidor, através de um caminho específico.

Exemplo: Para quando um socket é ligado ao servidor, uma função de “ligado” é disparada, executando um certo código.

```
io.on('connection', (socket) => {
  socket.on('onConnect' , data => events.onConnect(socket, data));
  socket.on('disconnect' , () => events.onDisconnect(socket));
  socket.on('latency' , () => events.onLatency(socket));

  socket.on('join' , data => events.onJoiningRoom(socket, data));
  socket.on('dm' , data => events.onDirectMessage(socket, data));
  socket.on('typing' , data => events.onTyping(socket, data));
  socket.on('status' , data => events.onStatusChange(socket, data));

  socket.on('friends-pending', data => events.onFriendPending(socket, data));
  socket.on('friend-remove' , data => events.onFriendRemoved(socket, data));
  socket.on('friends-accept' , data => events.onFriendAccepted(socket, data));
});
```

Figura 3 – Organização das funções dos sockets

2.2.3 MODELO E-R

O modelo Entidade-Relação da base de dados corresponde visualmente ao seguinte:

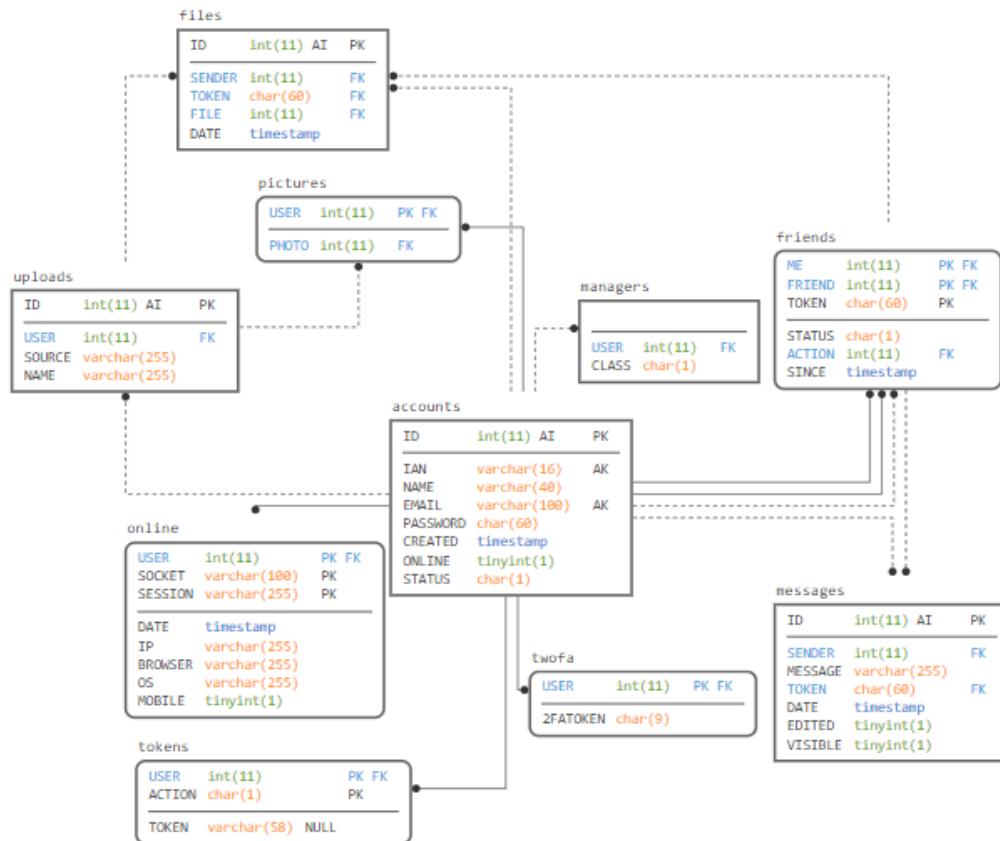


Figura 4 – Modelo Entidade-Relação

2.2.4 TABELA DE BASE DE DADOS E RELACIONAMENTOS

Para o desenvolvimento da aplicação, e conseguir chegar ao objetivo de construção de um sistema de mensagens em tempo-real, foi preciso a construção de uma base de dados bem estruturada. Para isso, foram criadas as seguintes tabelas:

- accounts: (contas)
- files: (ficheiros nas conversas)
- friends: (amigos);
- managers (administradores);
- messages: (mensagens);
- online: (sessões abertas);
- pictures: (foto de perfil);
- tokens: (códigos de ativação);
- twofa: (códigos de segurança);
- uploads: (ficheiros);

As tabelas relacionam-se da seguinte maneira:

- Friends – Accounts
 - Os campos *ME*, *FRIEND* e *ACTION*, dependem da tabela de contas.
- Managers – Accounts
 - O campo *USER* depende do campo *ID* da tabela de contas.
- Online – Accounts
 - O campo *USER* depende do campo *ID* da tabela de contas.
- Messages – Accounts
 - O campo *SENDER* depende do campo *ID* da tabela de contas.
- Messages – Friends
 - O campo *TOKEN* depende do campo *TOKEN* da tabela de amigos.
- Tokens – Accounts
 - O campo *USER* depende do campo *ID* da tabela de contas
- Twofa – Accounts
 - O campo *USER* depende do campo *ID* da tabela de contas
- Uploads – Accounts
 - O campo *USER* depende do campo *ID* da tabela de contas.

- Pictures – Accounts
 - O campo *USER* depende do campo *ID* da tabela de contas
- Pictures – Uploads
 - O campo *PHOTO* depende do campo *ID* da tabela de ficheiros.
- Files – Accounts
 - O campo *SENDER* depende do campo *ID* da tabela de contas.
- Files – Friends
 - O campo *TOKEN* depende do campo *TOKEN* da tabela de amigos.
- Files – Uploads
 - O campo *FILE* depende do campo *ID* da tabela de ficheiros.

2.2.5 TIPOS DE DADOS ESCOLHIDOS (SQL)

Tipo de dados da tabela de contas (accounts)

- Número da Conta (ID)
 - Número único que identifica cada conta individualmente, e que incrementa automaticamente.
(INT AUTO_INCREMENT PRIMARY KEY)
- Nome de Utilizador (IAN)
 - IAN ou *In App Name*, é um nome único que identifica cada utilizador, com um máximo de 16 caracteres.
(VARCHAR(16) UNIQUE NOT NULL)
- Nome (NAME)
 - Nome pessoal do utilizador que se registou, que tem o máximo de 40 caracteres.
(VARCHAR(40) NOT NULL)
- Endereço Eletrónico (EMAIL)
 - Campo único com um máximo de 100 caracteres que identifica cada conta individualmente.
(VARCHAR(100) UNIQUE NOT NULL)
- Palavra-Passe (PASSWORD)
 - Campo com um valor fixo de 60 caracteres que abriga a palavra-passe encriptada numa *hash* de tamanho fixo de 60 caracteres.
(CHAR(60) NOT NULL)

- Data de Criação (CREATED)
 - Campo de tempo que contém a data do momento exato de registo do utilizador.
(TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP)
- Atividade do Utilizador (ONLINE)
 - Campo booleano que informa se o utilizador está ativo.
(TINY INT(1) NOT NULL DEFAULT '0')
- Estado do Utilizador (STATUS)
 - Campo com um valor fixo de 1 carácter, que identifica o estado do utilizador em todas as sessões.
(CHAR(1) NOT NULL DEFAULT '1').

Tipos de dados da tabela de códigos de segurança (twofa)

- Número da Conta (USER)
 - Campo único e primário que identifica o código de segurança de cada conta presente na tabela.
(INT PRIMARY KEY)
- Código de Segurança (2FATOKEN)
 - Campo com um tamanho fixo de 9 caracteres, que abriga um código único de 9 dígitos.
(CHAR(9) NOT NULL)

Tipo de dados da tabela de códigos de ativação (tokens)

- Número da Conta (USER)
 - Campo único e primário que identifica o código de ativação de cada conta presente na tabela.
(INT PRIMARY KEY)
- Código de ativação (TOKEN)
 - Campo fixo de 58 caracteres, que abriga uma data encriptada sempre diferente.
(CHAR(58) DEFAULT NULL)
- Tipo do código de ativação (ACTION)
 - Campo com o tamanho fixo de 1 carácter que identifica cada código de ativação.

Tipos de dados da tabela de amigos (friends)

- Número da Conta (ME)
 - Número único da chave composta que identifica a conta do utilizador (INT PRIMARY KEY)
- Número da Conta do Amigo (FRIEND)
 - Número único da chave composta que identifica a conta do amigo do utilizador. (INT PRIMARY KEY)
- Estado da Amizade (STATUS)
 - Campo com um tamanho fixo de 1 carácter que tem como objetivo guardar o estado da amizade. (CHAR(1) NOT NULL)
- Sala de Mensagens (TOKEN)
 - Campo com um tamanho fixo de 60 caracteres e que faz parte da chave composta, e que identifica cada sala de mensagens entre os utilizadores. (CHAR(60) PRIMARY KEY)
- Quem tomou a Iniciativa (ACTION)
 - Campo que abriga o número da conta, no qual começou primeiro a amizade. (INT NOT NULL)
- Data de Inicio (SINCE)
 - Campo que contem a data de início da amizade. (TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP)

Tipos de dados da tabela de mensagens (messages)

- Número da Mensagem (ID)
 - Número único que identifica cada mensagem individualmente, e que incrementa automaticamente. (INT PRIMARY KEY AUTO_INCREMENT)
- Quem enviou a Mensagem (SENDER)
 - Número referente á conta que enviou a mensagem. (INT NOT NULL)
- Mensagem (MESSAGE)
 - Campo com um total máximo de 255 caracteres, que tem como objetivo guardar a mensagem enviada. (VARCHAR(255) NOT NULL)

- Sala de Mensagens (TOKEN)
 - Campo com um tamanho fixo de 60 caracteres que identifica a sala de mensagens para o qual a mensagem será enviada.
(CHAR(60) NOT NULL)
- Data de envio (DATE)
 - Campo que guarda o momento exato que a mensagem foi enviada.
(TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP)
- Mensagem editada (EDITED)
 - Campo booleano para identificar se a mensagem foi editada.
(TINYINT(1) NOT NULL DEFAULT '0')

Tipo de dados da tabela de ficheiros (uploads)

- Número do Ficheiro (ID)
 - Número único que identifica cada ficheiro individualmente, e que incrementa automaticamente.
(INT PRIMARY KEY AUTO_INCREMENT)
- Quem Enviou (USER)
 - Campo que identifica o utilizador que enviou o ficheiro para a aplicação.
(INT NOT NULL)
- Localização do Ficheiro (SOURCE)
 - Campo que contém o tamanho máximo de 255 caracteres possíveis para o nome do ficheiro modificado para não existir ficheiros com o mesmo nome.
(VARCHAR(255) NOT NULL)
- Nome do Ficheiro (NAME)
 - Campo que abriga o nome original do ficheiro, para questões de apresentação na aplicação.
(VARCHAR(255) NOT NULL)

Tipos de dados da tabela de fotografias das contas (pictures)

- Número do Utilizador (USER)
 - Campo primário que a conta do utilizador nesta tabela.
(INT PRIMARY KEY)
- Número do Ficheiro (PHOTO)
 - Campo que identifica que imagem estará visível como foto de perfil.
(INT NOT NULL)

Tipos de dados da tabela de ficheiros por chat (files)

- Número do envio (ID)
 - Número único que identifica o envio de cada ficheiro enviado nas conversas
(INT PRIMARY KEY AUTO_INCREMENT)
- Quem enviou (SENDER)
 - Campo que identifica a conta do utilizador que fez "upload" do ficheiro.
(INT NOT NULL)
- Conversa (TOKEN)
 - Campo com um tamanho fixo de 60 caracteres que identifica a conversa para o qual o ficheiro foi enviado.
(CHAR(60) NOT NULL)
- Ficheiro (FILE)
 - Campo que identifica qual o ficheiro que foi enviado.
(INT NOT NULL)
- Data de Envio (DATE)
 - Campo que guarda o momento exato que o ficheiro foi enviado.
(TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP)

Tipos de dados da tabela de utilizadores ativos (online)

- Número do Utilizador (USER)
 - Número que identifica o utilizador.
(INT PRIMARY KEY)
- Socket Ativo (SOCKET)
 - Campo único que identifica o socket ativo na sessão.
(CHAR(100) PRIMARY KEY)
- Sessão Ativa (SESSION)
 - Campo único que identifica a sessão ativa.
(CHAR(255) PRIMARY KEY)
- Data de Início (DATE)
 - Campo que guarda a data de início da sessão
(TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP)
- Endereço IP (IP)
 - Campo que abriga o endereço de IP da sessão ativa
(VARCHAR(255) NOT NULL)

- Navegador Utilizado (BROWSER)
 - Campo que guarda a informação sobre qual navegador está a ser usado na sessão.
(VARCHAR(255) NOT NULL)
- Sistema Operativo (OS)
 - Campo que abriga a informação sobre qual sistema operativo a sessão está aberta.
(VARCHAR(255) NOT NULL)
- Telefone (MOBILE)
 - Campo booleano que identifica se a sessão ativa está num dispositivo móvel ou não.
(TINYINT(1) NOT NULL DEFAULT '0')

2.2.6 SOFTWARE UTILIZADO

Para o desenvolvimento do projeto e do relatório foram necessários os seguintes programas:

- NodeJS: o NodeJS foi utilizado na aplicação como servidor de *backend* (serve-side), pois ele permite utiliza Javascript como linguagem e, para além disso, usa um modelo orientado a eventos e módulos, ou seja, é possível instalar vários módulos com diferentes funcionalidades, para facilitar o desenvolvimento e atualização de aplicações web.
- Xampp: o Xampp é um programa essencial na aplicação, porque funciona como servidor de Mysql e apache, onde está a base de dados da aplicação.

Com a utilização do apache é possível usar o *phpmyadmin*, que permite testar certos comandos em sql antes de serem implementados na aplicação.

- Postman: o Postman é uma aplicação simples, têm como único objetivo testar serviços *RESTful* (Web Api's), por meio do envio de requisições HTTP e da análise do seu retorno. Graças a ele foi possível testar as variadas rotas da aplicação.

- Visual Studio Code: o VS Code é basicamente a ferramenta principal de trabalho e desenvolvimento do projeto. O Visual Studio Code é um IDE (editor de código), que possui um terminal integrado para o debug da aplicação, e o controle de *Git*, onde é possível guardar o trabalho e aceder ou continuar a trabalhar noutro computador de forma rápida e eficaz.
- Adobe XD CC: o Adobe XD permite a construção de designs de forma fácil, que são possíveis exportar para diversos formatos. Com isso em mente, foi possível criar e alterar algumas ilustrações, e fazer testes de estrutura da página web em diversos dispositivos de tamanhos variados.
- Adobe Photoshop CC: o Photoshop é um programa muito útil que permite fazer uma edição de imagem avançada, com uma grande variedade de ferramentas à disposição. Neste projeto específico, foi utilizada para retirar certos fundos, alterar certas cores e redimensionar algumas imagens, sem recorrer ao Adobe XD, o que possibilitou o ganho de tempo.
- Microsoft Word 2019: o Word é uma aplicação extremamente útil e poderosa no que toca a criar documentos. Com inúmeras funcionalidades, o Word foi utilizado para criar o relatório de planificação e o de base de dados. Mais importante ainda, foi utilizado na criação do relatório de aptidão profissional.

2.3 CARACTERIZAÇÃO FINAL DO PRODUTO

Como tratado até agora, o produto é uma aplicação de mensagens em tempo real.

Página Inicial:

Na página inicial do website, o visitante / utilizador pode:

- Entrar na sua conta;
- Recuperar a Conta;
- Registrar-se;
- Entrar na sua conta, usando o segundo fator de autenticação

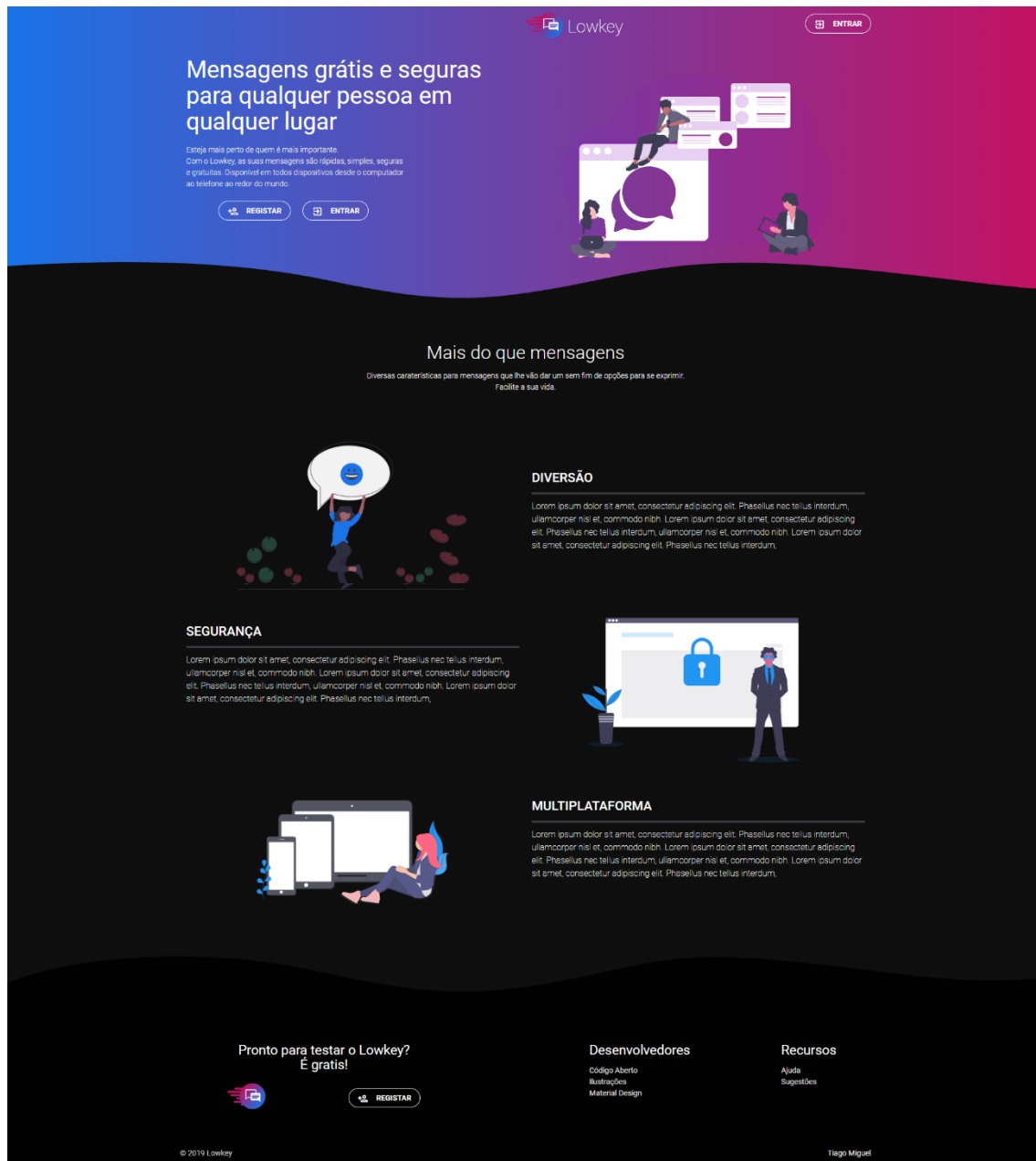


Figura 5 – Página Inicial

O formulário de registo aparece quando o utilizador clica num botão que tenha o nome de “Registar”.

The screenshot shows a dark-themed web interface for the 'Lowkey' application. At the top right, there is a logo and a button labeled 'ENTRAR'. The main content area features a central modal box titled 'Crie uma Conta' (Create an Account) with the subtitle 'Mensagens simples e seguras!'. The form contains several input fields: 'Nome' (Name), 'Nome de Utilizador' (Username), 'Endereço Eletrónico (email)' (Electronic Address (email)), 'Palavra-passe' (Password), and 'Confirmar a Palavra-passe' (Confirm Password). Below these fields are two buttons: 'LIMPAR' (Clear) and 'REGISTAR' (Register). At the bottom of the modal, there is a link that says 'Já tem conta? Inicie sessão' (Already have an account? Log in). The background of the page shows a person holding a speech bubble and some text under the heading 'DIVERSÃO' (Entertainment).

Figura 6 – Formulário de Registo

Depois que o formulário tenha sido preenchido corretamente, uma mensagem de confirmação aparecerá na tela, pedindo que o utilizador confirme a sua conta clicando no link que é enviado para o seu endereço eletrónico.

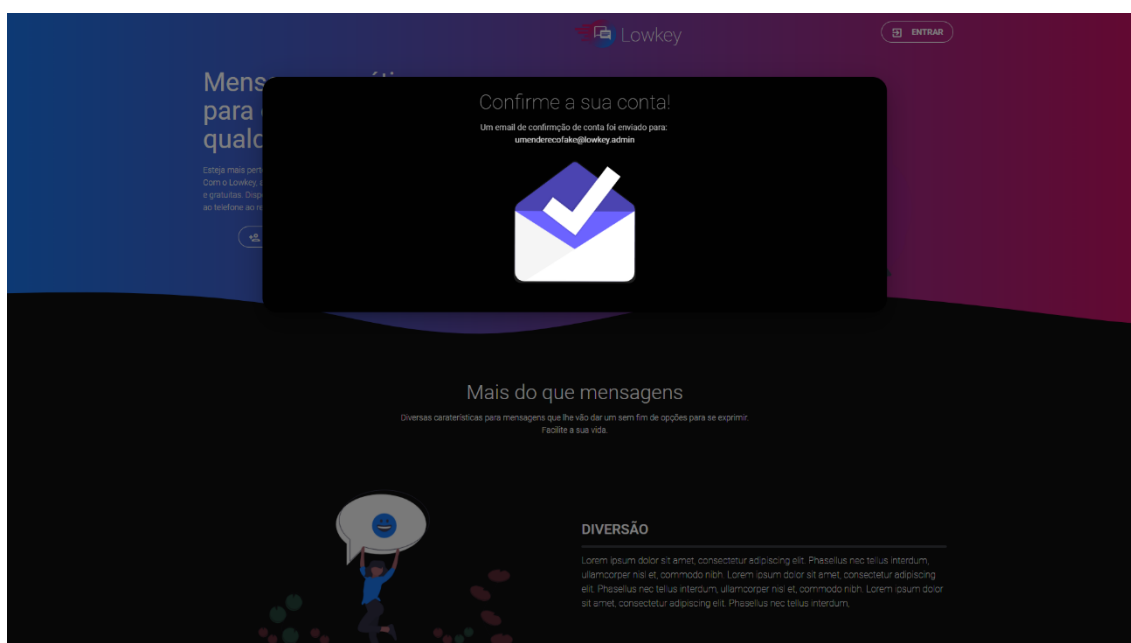


Figura 7 – Mensagem de Confirmação da Conta

Quando o utilizador clica em qualquer botão com o nome “Entrar”, o seguinte formulário aparece, e é onde o utilizador consegue iniciar sessão.

Boas-vindas!

Estamos muito animados em te ver novamente!

Nome de Utilizador ou Email

Palavra-passe

Esqueceu-se da palavra-passe?

LIMPAR ENTRAR

Ainda não tem conta? Registe-se

Facilite a sua vida.

DIVERSÃO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec tellus interdum, ullamcorper nisi et, commodo nish. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec tellus interdum, ullamcorper nisi et, commodo nish. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec tellus interdum,

Figura 8 – Formulário de Início de Sessão

Caso o utilizador tenha esquecido a sua palavra-passe, é possível recuperar, por clicar no botão / texto com o nome de: “Esqueceu-se da palavra-passe”, logo abaixo da caixa de texto da palavra-passe.

Recupere a sua conta!

Recupere a sua conta hoje

Nome de Utilizador ou Email

O Email de recuperação só é válido durante 30 minutos!

← RECUPERAR CONTA

Mais do que mensagens

Diversas características para mensagens que lhe vão dar um sem fim de opções para se exprimir. Facilite a sua vida.

DIVERSÃO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec tellus interdum, ullamcorper nisi et, commodo nish. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec tellus interdum, ullamcorper nisi et, commodo nish. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec tellus interdum,

Figura 9 – Formulário de Recuperação da Conta

Caso o utilizador tenha ativo o Segundo Fator de Autenticação na sua conta, que pode ser ativado nas definições de conta, aparece a seguinte tela, onde um código especial de 9 dígitos é enviado para o email do utilizador. Isto é uma forma de proteger a conta.

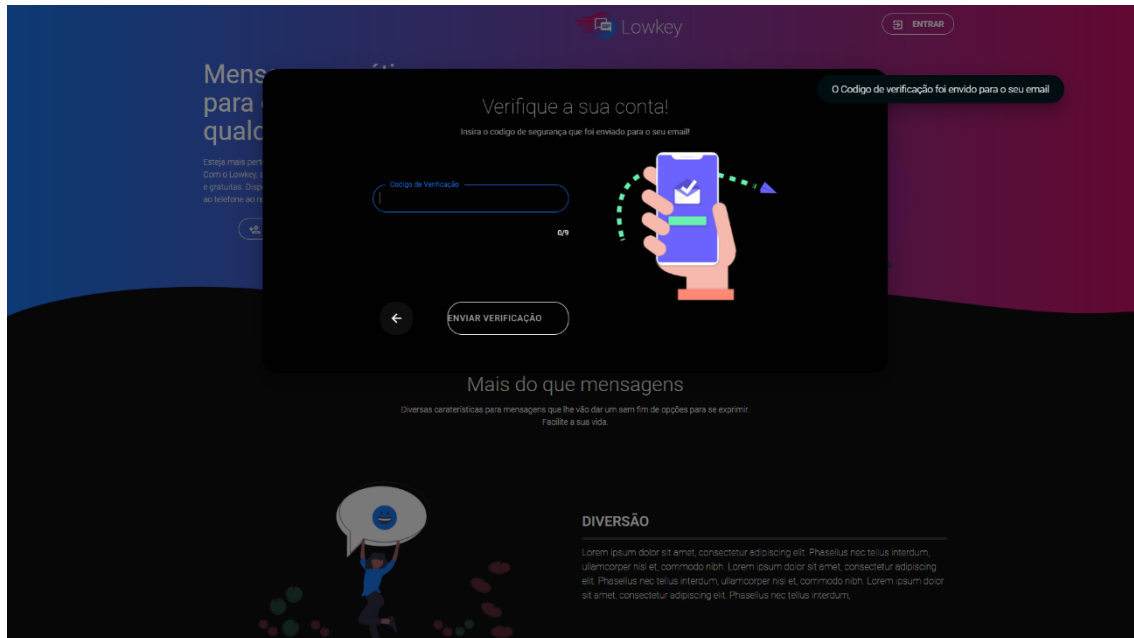


Figura 10 – Formulário do Segundo Fator de Autenticação

CONTINUAÇÃO DOS SISTEMAS EXTRA DE AUTENTICAÇÃO

Ao confirmar a conta clicando no link enviado para o endereço eletrónico do utilizador, o mesmo é deparado com a seguinte tela.

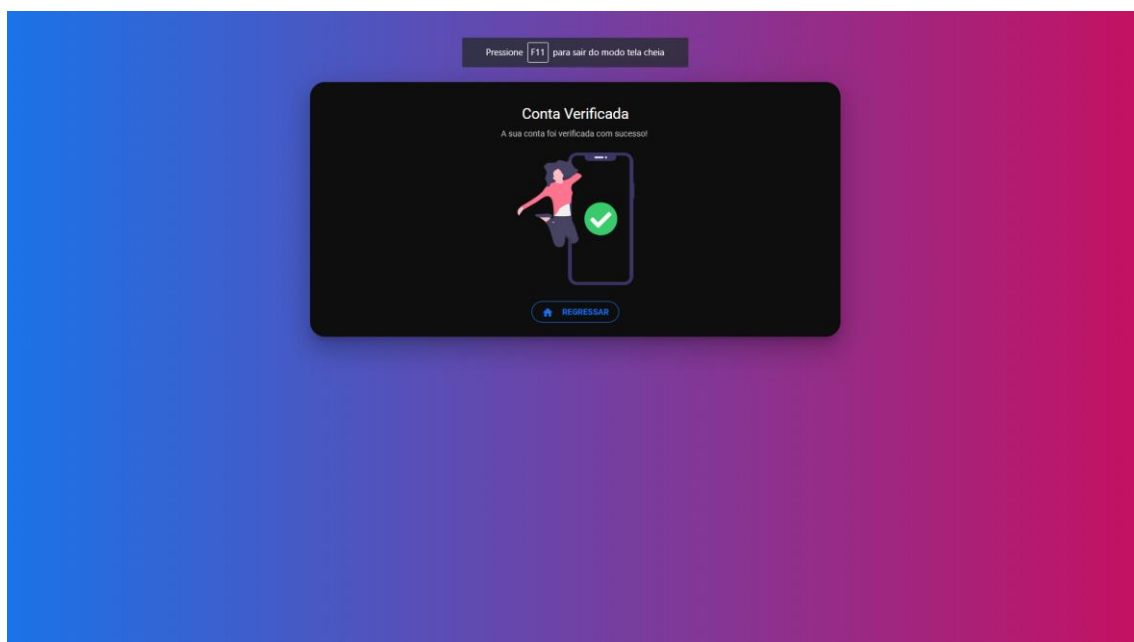


Figura 11 – Conta Verificada

ENVIO E RESPOSTA DA RECUPERAÇÃO DE CONTA

Ao inserir o email ou o nome de utilizador no formulário de recuperação de conta já acima apresentado, é apresentada a seguinte tela:

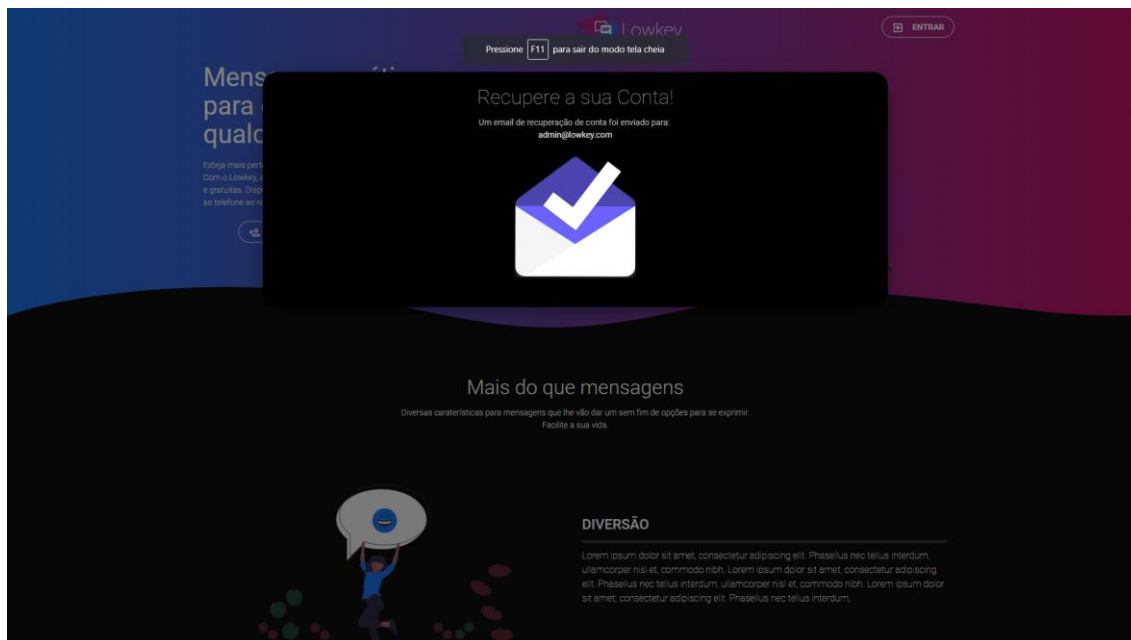


Figura 12 – Mensagem de Recuperação da Conta

Depois do utilizador ir ao endereço eletrónico e aceder ao link de recuperação, uma palavra-passe nova e segura é gerada, e a seguinte tela é apresentada:

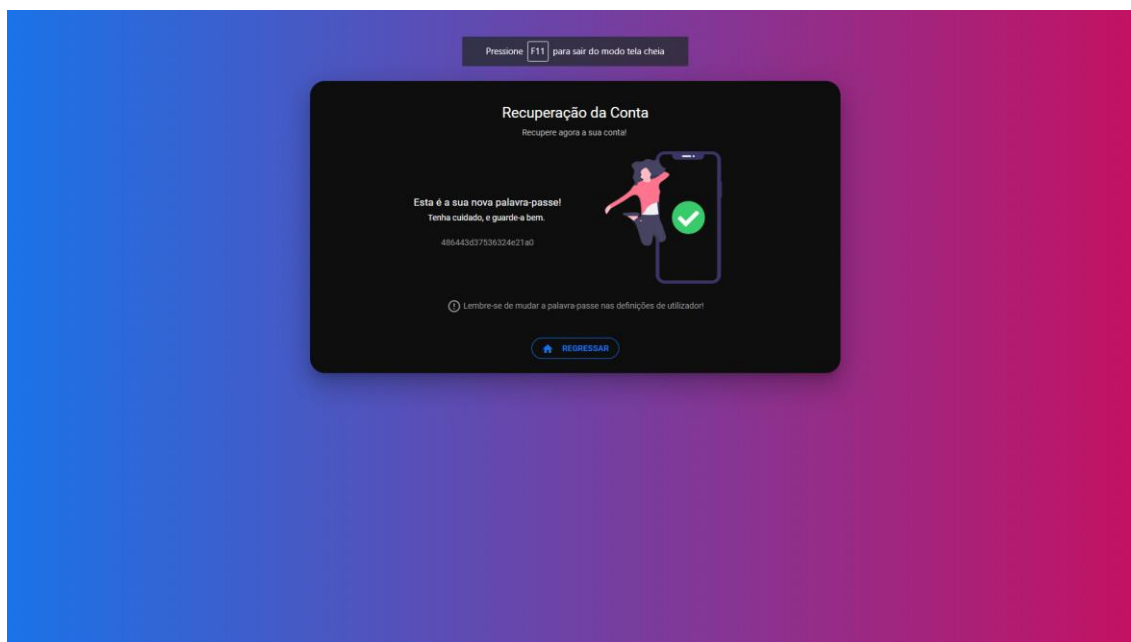


Figura 13 – Conta Recuperada

ELIMINAÇÃO DA CONTA AO REGISTRAR

Esta função de eliminar a conta ao registrar, aparece quando por exemplo alguém se regista a passar-se pelo outro utilizador, ou quando o utilizador não deseja mais continuar o registo da conta. No email enviado durante o registo ao utilizador, contem o link de confirmar a conta e o link de eliminar a conta. Ao clicar nesse mesmo link é apresentada a seguinte informação:

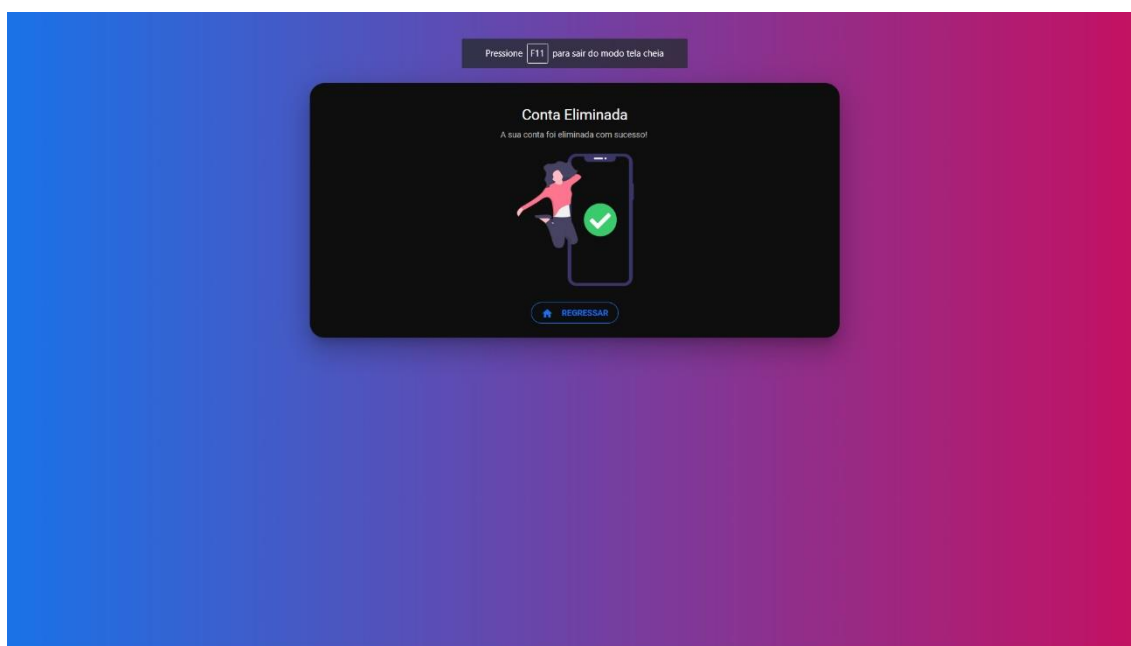


Figura 14 – Conta Eliminada

Quando a conta já esta verificada, recuperada ou eliminada, o link deixa de ser valido. Quando isso acontece a seguinte informação é mostrada ao utilizador, que por exemplo tenta confirmar a sua conta já confirmada.

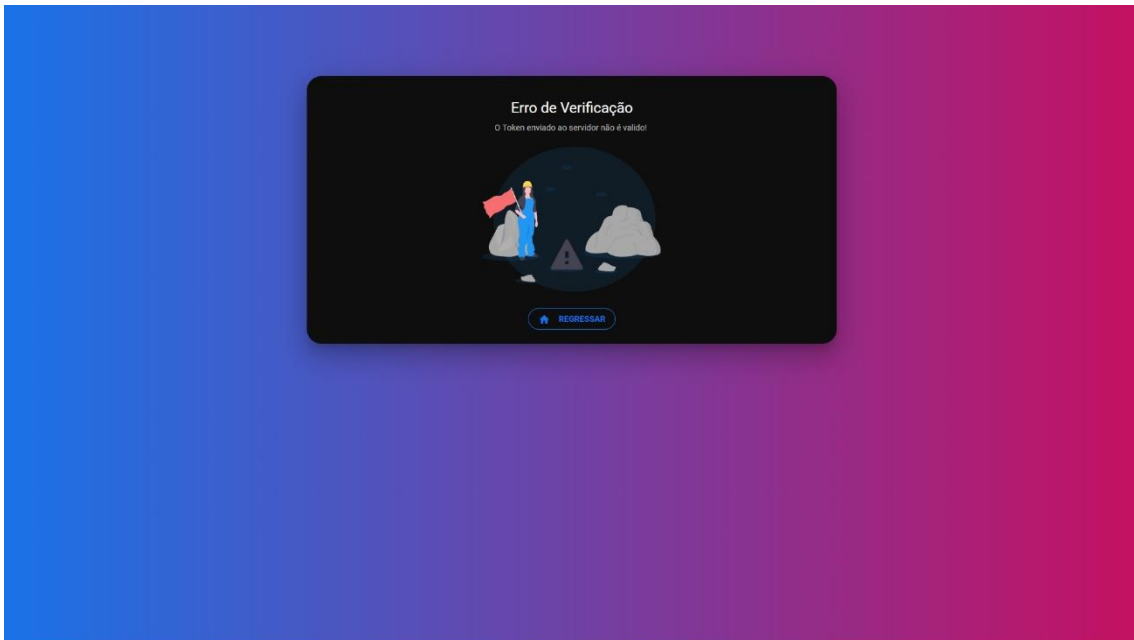


Figura 15 – Erro de Verificação

APÓS O INICIO DE SESSAO

Após o início de sessão, o utilizador entra na aplicação em si. Nesta página o utilizador pode:

- Adicionar um amigo
- Remover um amigo
- Aceitar ou rejeitar um amigo
- Conversar com um amigo
 - Enviar ficheiros
 - Enviar emojis nas conversas
 - Enviar Gifs (Tenor integrado na aplicação)
- Alterar o seu estado
- Alterar as definições de conta
- Terminar sessão
- Administrador
 - Opção de aceder à administração

Página Inicial da Aplicação:

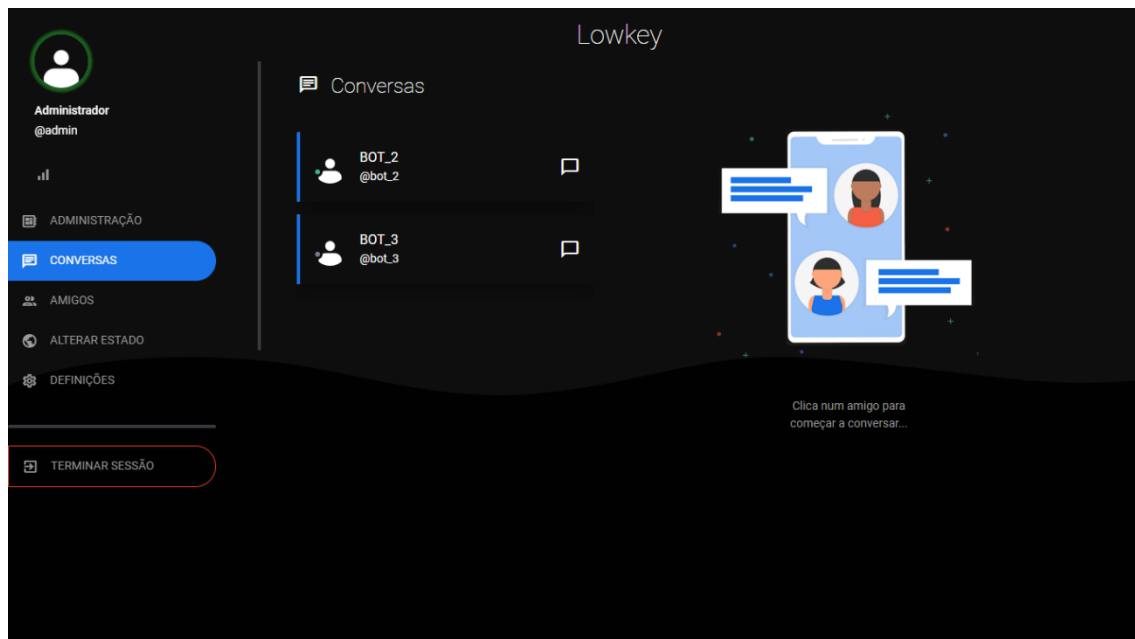


Figura 16 – Página Inicial da Aplicação

Para adicionar um amigo, o utilizador clica na aba de amigos localizada na página inicial, onde vai encontrar 2 botões que fazem surgir o formulário de adicionar um amigo.

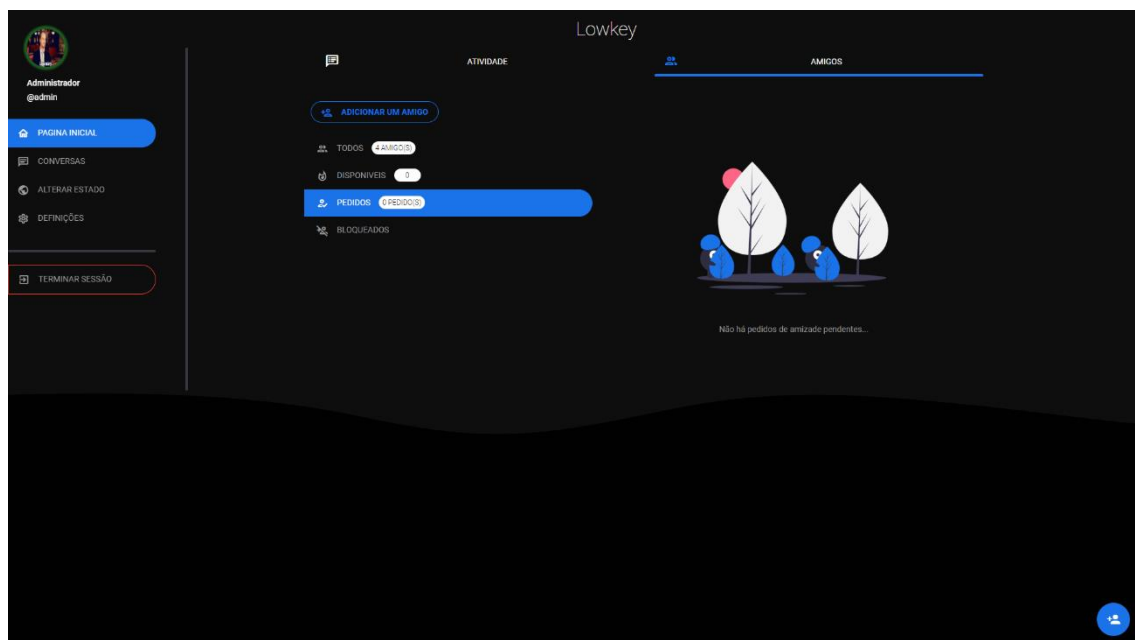


Figura 17 – Secção de Amigos

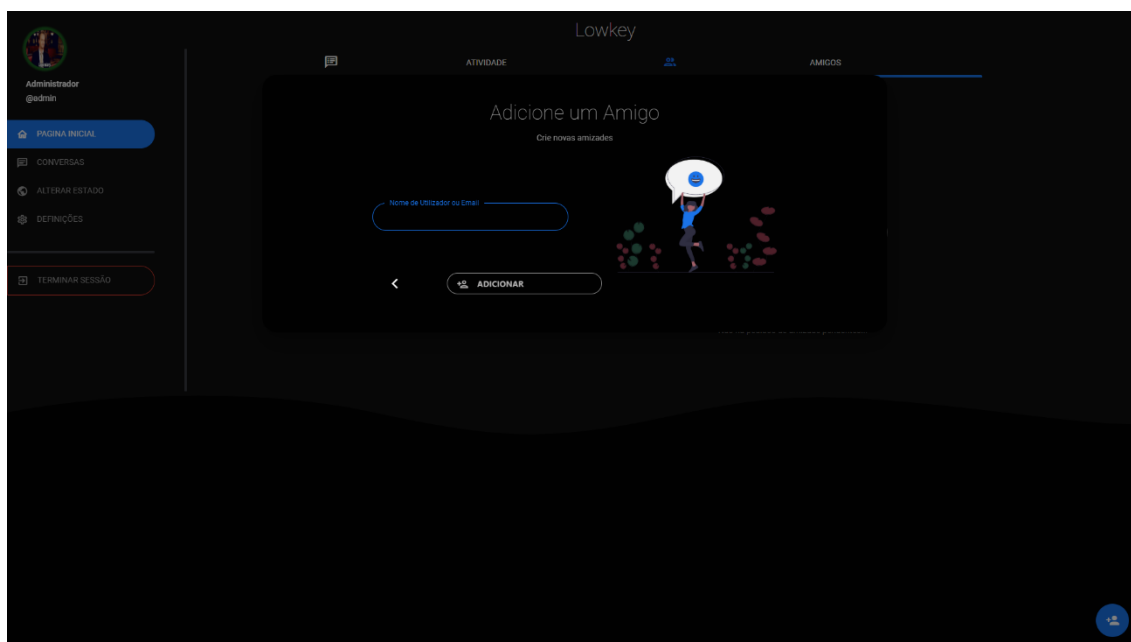


Figura 18 – Formulário para adicionar amigos

Assim que o pedido de amizade seja enviado, a seguinte mensagem de informação aparece do canto superior esquerdo do utilizador.

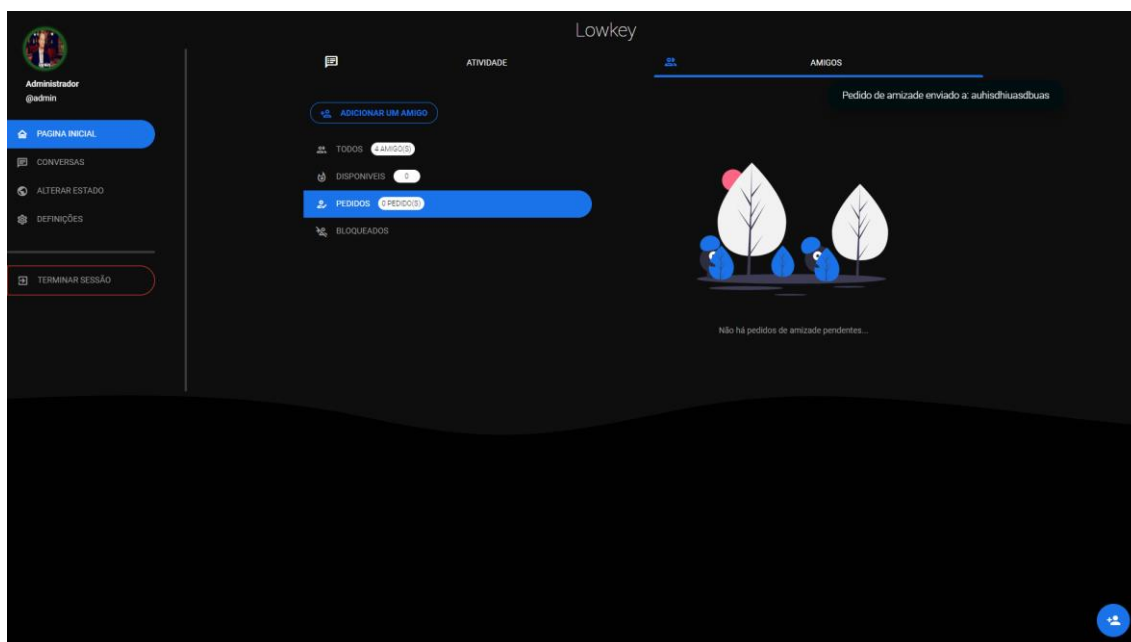


Figura 19 – Informação de Adicionar um amigo

Para o amigo que foi adicionado, ao aceder aos pedidos de amizade encontra o outro utilizador que o adicionou.

Nessa página o utilizador amigo pode aceitar só o amigo, por clicar no botão com o ícone de “certo”.

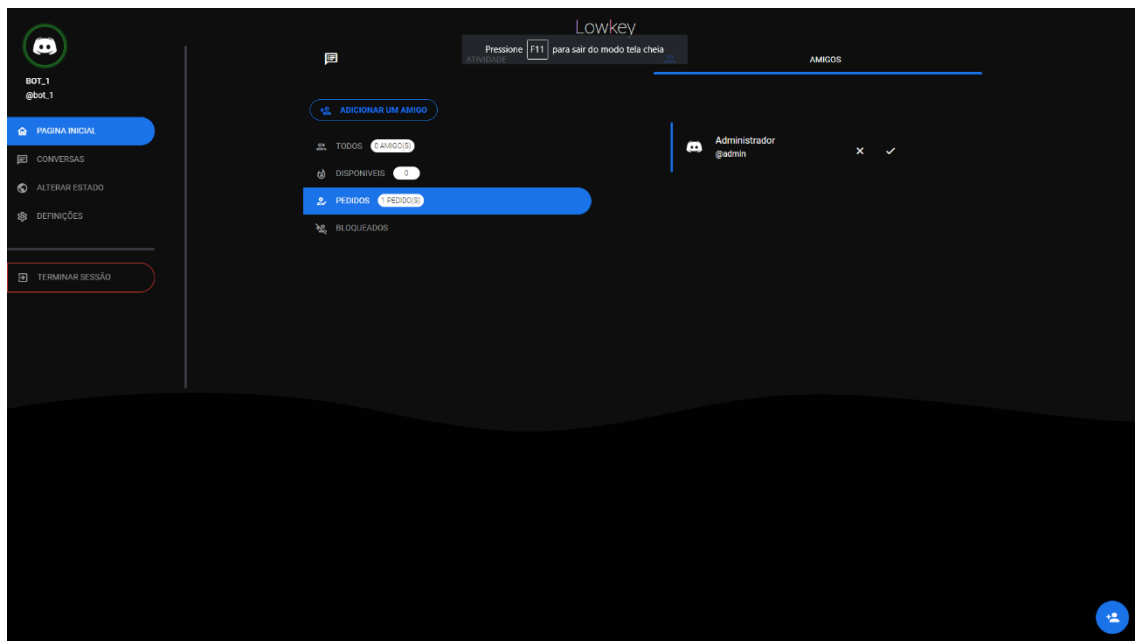


Figura 20 –Tela para aceitar um amigo

Ao entrar na aba de conversas, o utilizador pode escolher com qual amigo quer conversar. Quando clica no amigo que quer conversar, as suas conversas são carregadas, e caso ainda não tenham conversado aparece a seguinte informação.

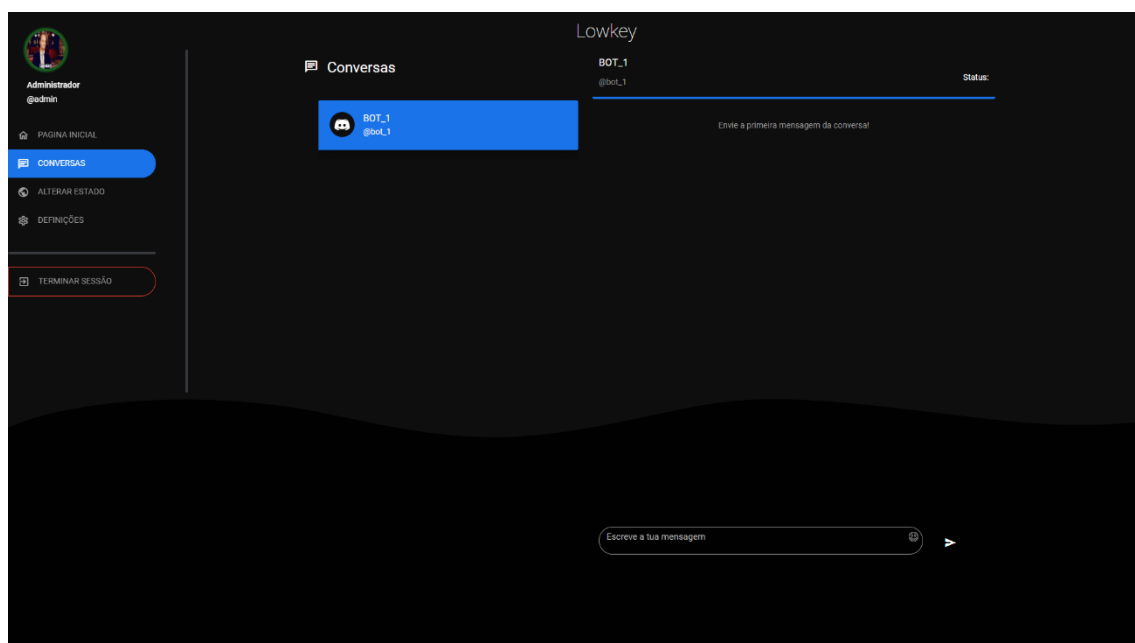


Figura 21 –Tela de conversas

Assim que um amigo já aceite na lista de amigos, entre na aplicação, uma notificação aparece no canto superior esquerdo para o utilizador, a informar que o amigo está online.

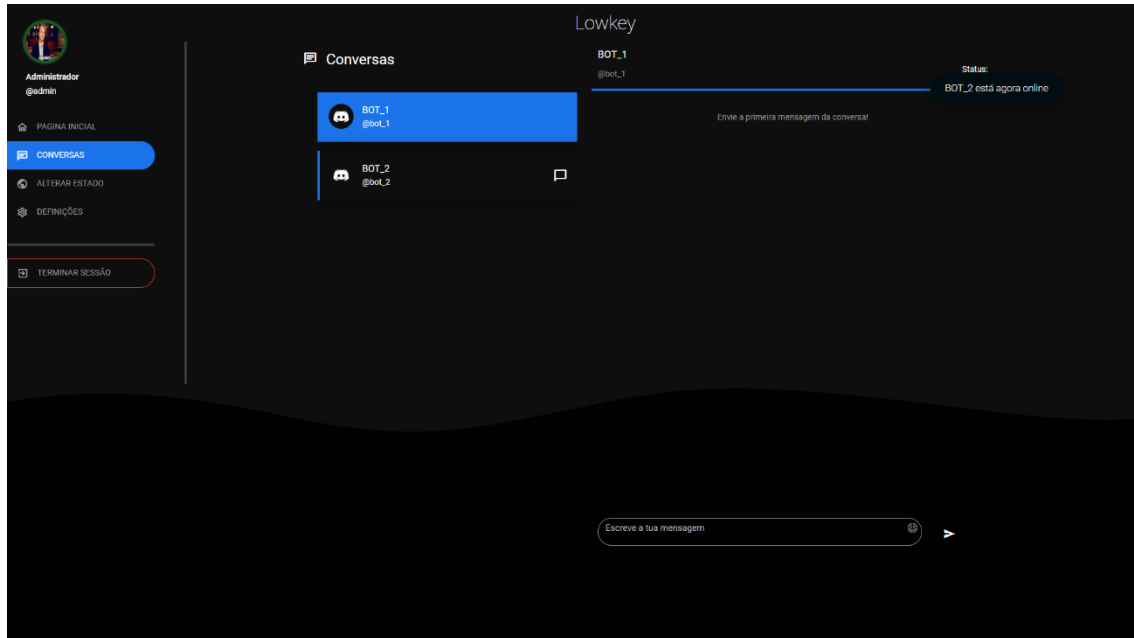


Figura 22 – Informação dos amigos

Quando o utilizador está a falar com um amigo, mas outro envia-lhe uma mensagem, a seguinte informação aparece.

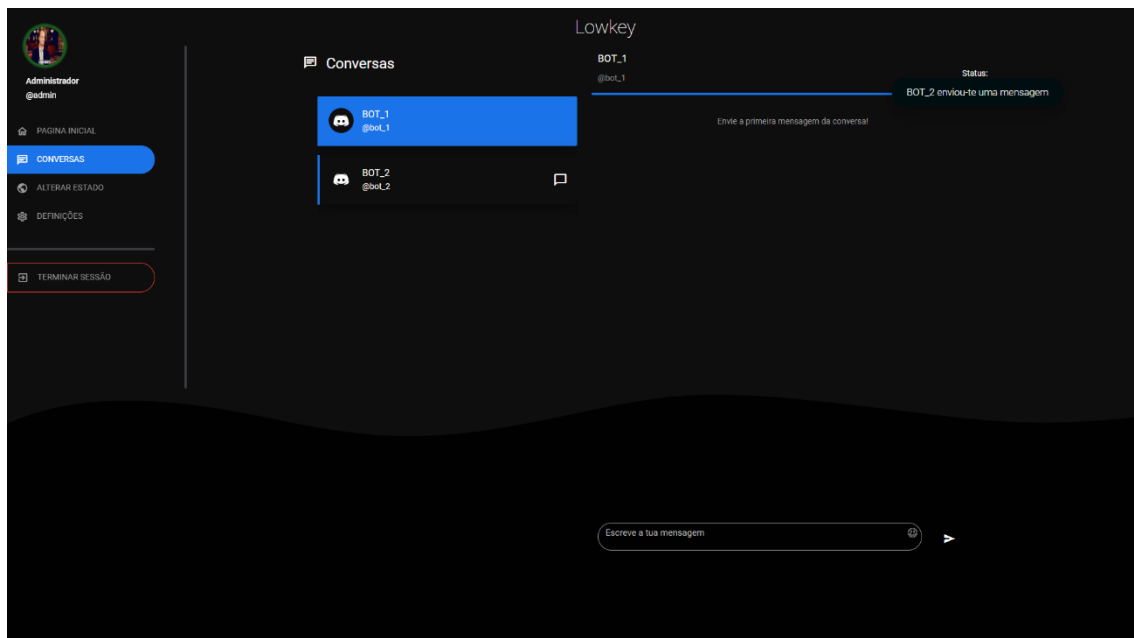


Figura 23 – Informação de Mensagem

Quando o utilizador altera o seu estado, todos os seus amigos são informados sobre o estado do utilizador, aparecendo uma informação no canto superior esquerdo, aparecendo uma informação no canto superior esquerdo.

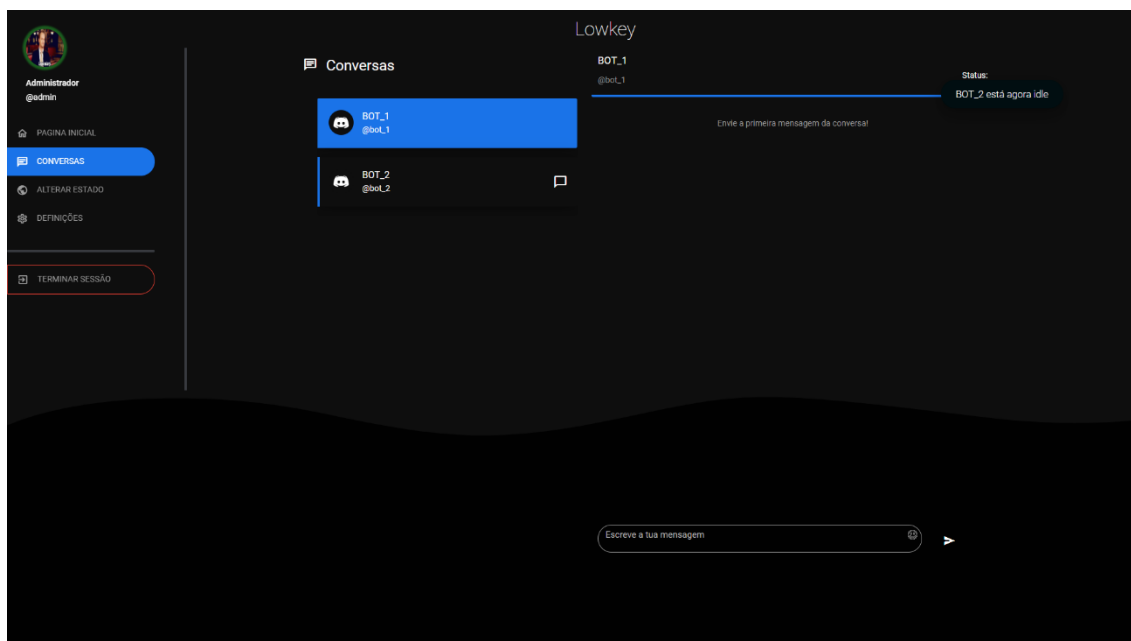


Figura 24 – Informação de Estado

O utilizador pode enviar vários ficheiros, e ainda adicionar uma mensagem adicional ao ficheiro enviado.

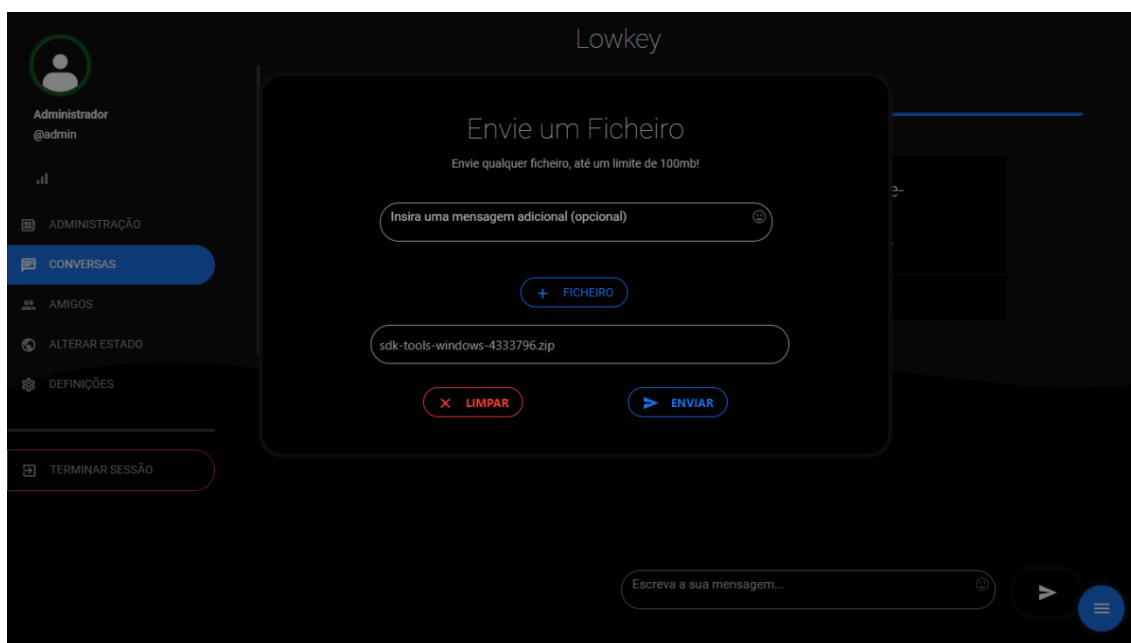


Figura 25 – Menu de Enviar Ficheiros

Quando o outro utilizador recebe o ficheiro, o ficheiro é apresentado na forma de cartão, caso seja diferente de uma imagem. O cartão dá duas opções ao utilizador: ver online ou fazer um download desse ficheiro. Caso não seja possível ao utilizador ver online, é feito automaticamente o download.

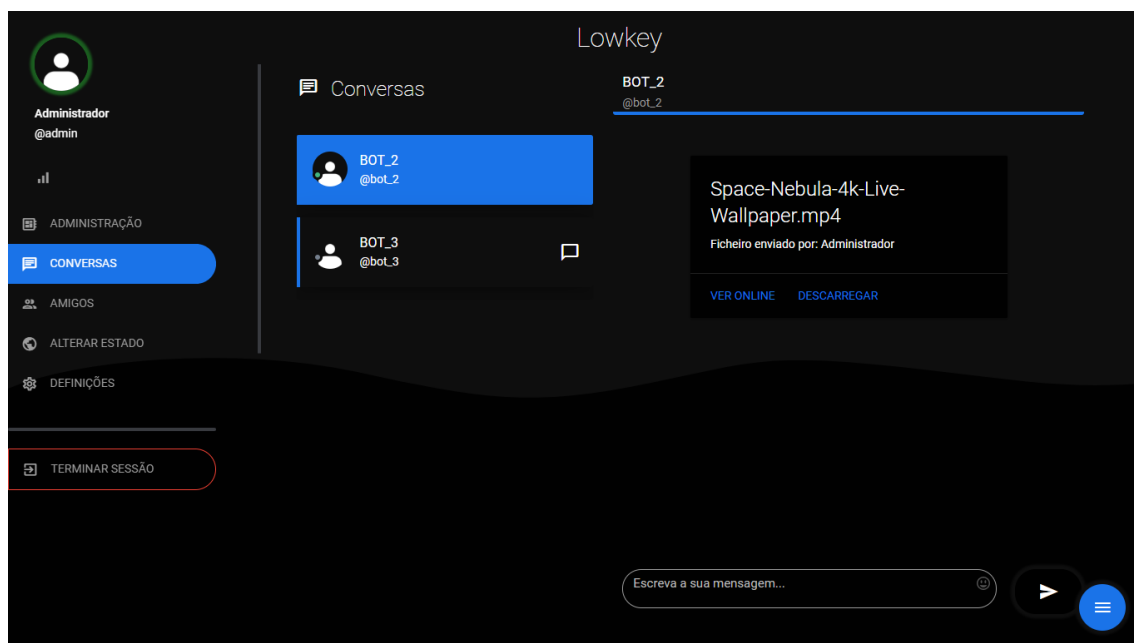


Figura 26 – Disposição do Ficheiro Enviado

No campo de texto para o utilizador escrever a sua a sua mensagem, é possível abrir um menu que contem “emojis”.

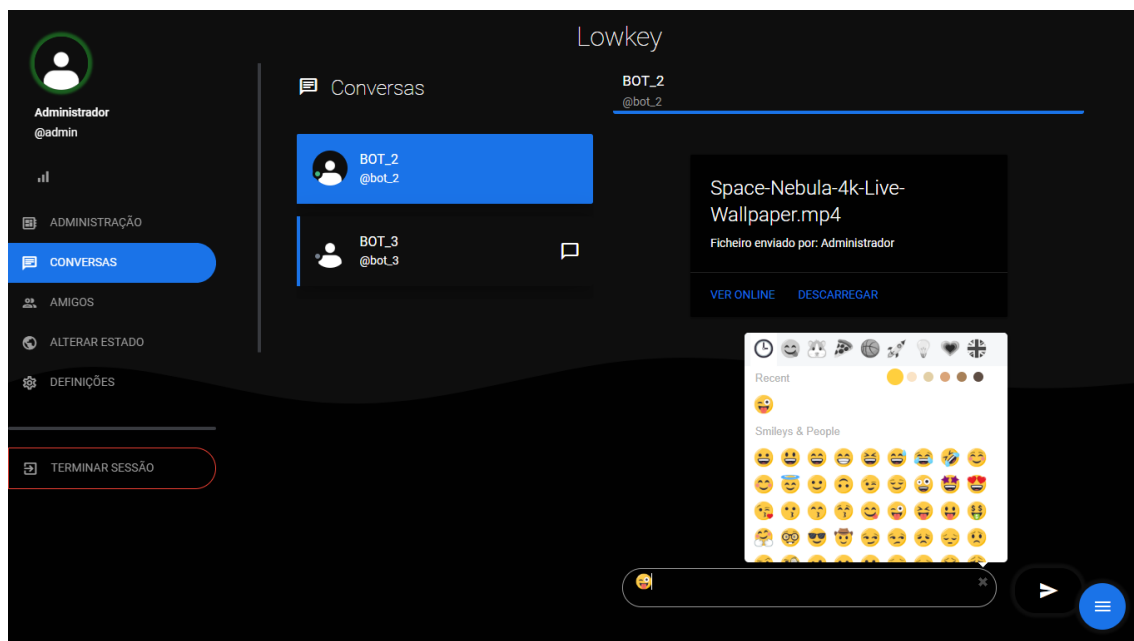


Figura 27 – Menu de Emojis

O utilizador ainda pode enviar “gifs”, visto que a aplicação tem a “API do Tenor” integrada. O tenor é um serviço de “gifs”, mundialmente utilizado, por quase todas as aplicações de mensagens, e até mesmo teclados.

Na aplicação, um menu é aberto com várias categorias de “gifs”, onde é possível procurar por “gifs” ou escolher um dos “gifs” de uma certa categoria.

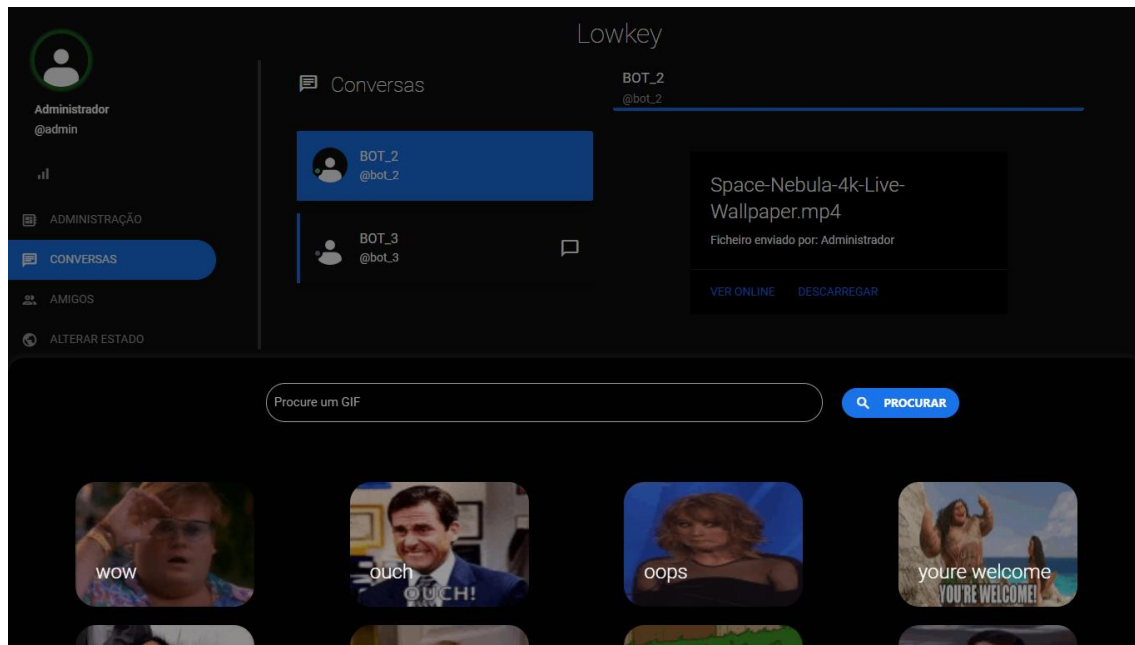


Figura 28 – Menu de Gifs

Na aba do lado esquerdo, ao selecionar as definições, a seguinte tela é mostrada ao utilizador, onde ele pode alterar as definições de conta e a segurança da conta.

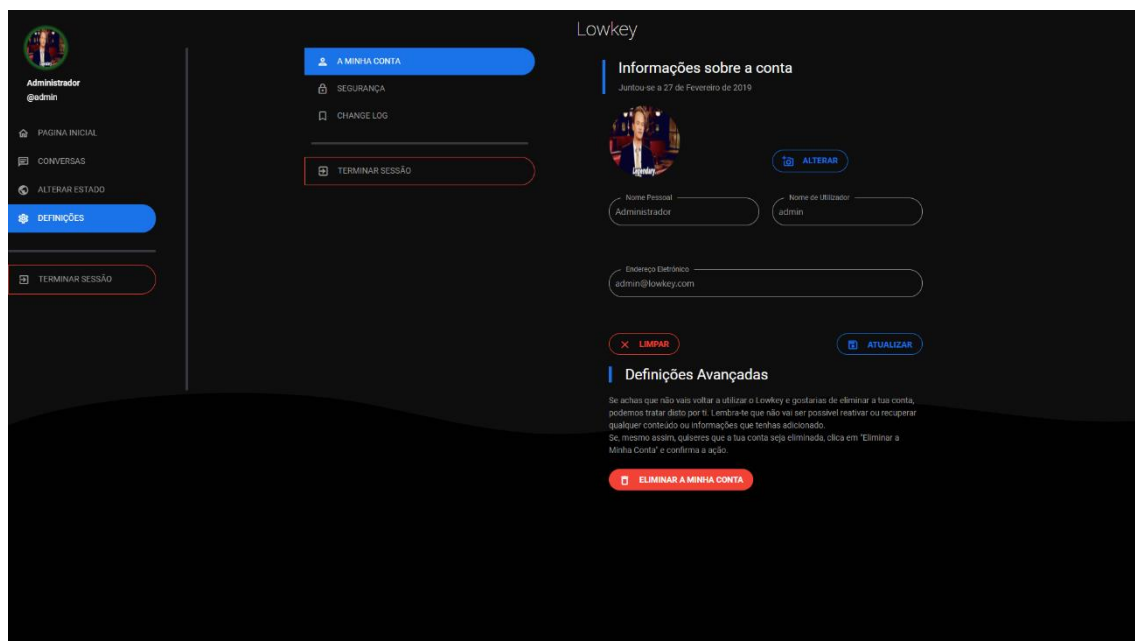


Figura 29 – Página de Definições da Conta

Na aba de segurança dentro das definições, o utilizador pode alterar a sua palavra-passe atual e ativar o segundo fator de autenticação.

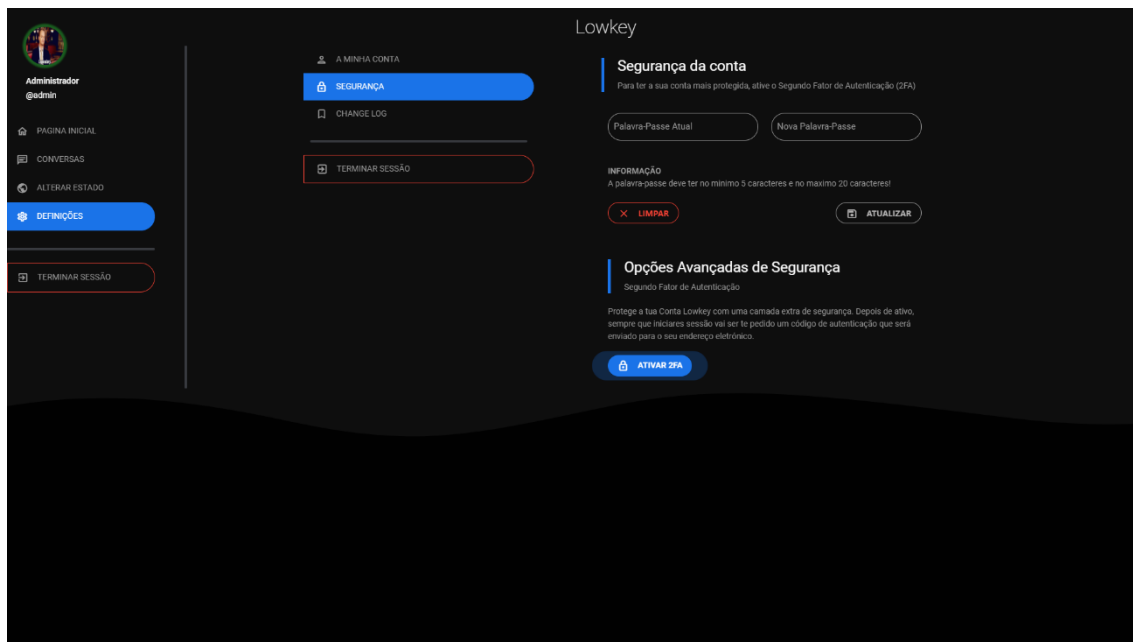


Figura 30 – Segurança da Conta

No botão “Ativar 2FA” é possível o utilizador habilitar o segundo fator de autenticação que protege mais a conta no início de sessão. Ao clicar, o seguinte formulário aparece.

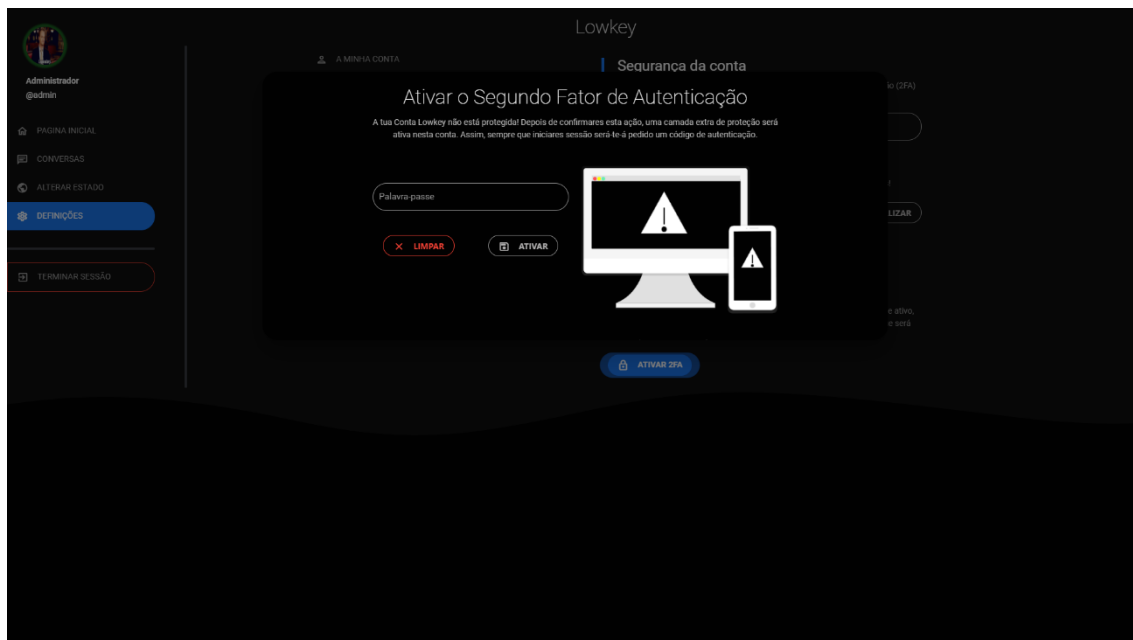


Figura 31 – Formulário de ativação do Segundo Fator de Autenticação

Assim que o utilizador ativa o segundo fator de autenticação, a seguinte informação aparece no canto superior esquerdo da tela.

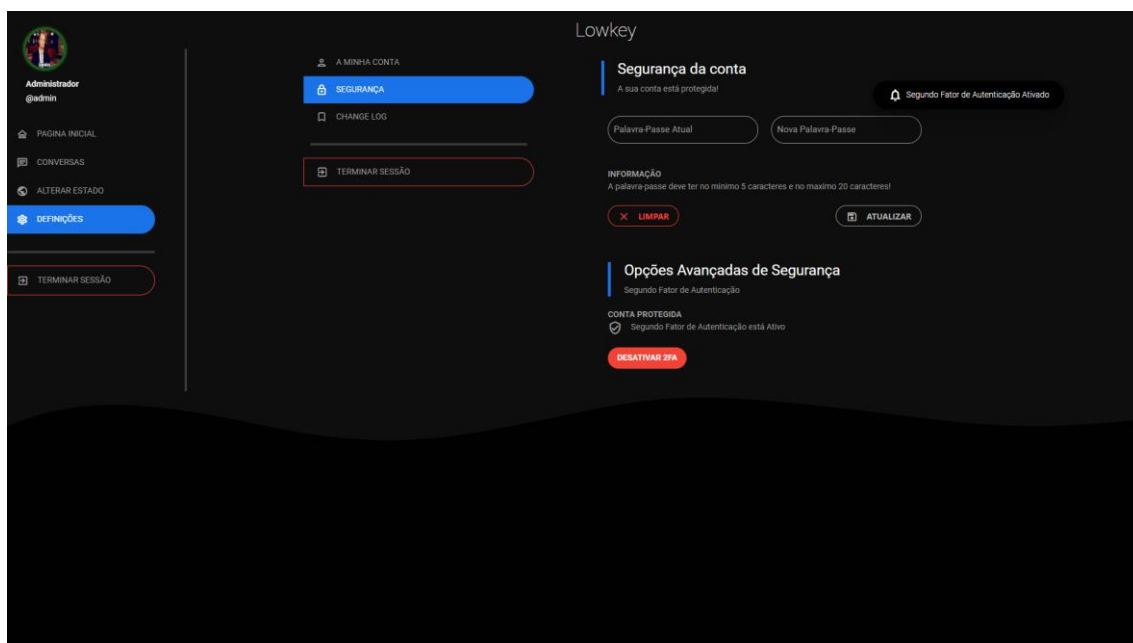


Figura 32 – Informação de Ativação do Segundo Fator de Autenticação

Caso o utilizador queira desativar o segundo fator de autenticação, aparece o formulário anterior, mas com os textos focados para a desativação, e a informação aparece da mesma forma, mas também com o texto alterado para: “Segundo Fator de Autenticação Desativado”.

Na aba de sessões do menu de definições, o utilizador pode ver as sessões que estão ativas, em que dispositivo estão ativas, a que horas que ficou ativa, a localização, o sistema operativo, o navegador e se foi num telemóvel ou não.

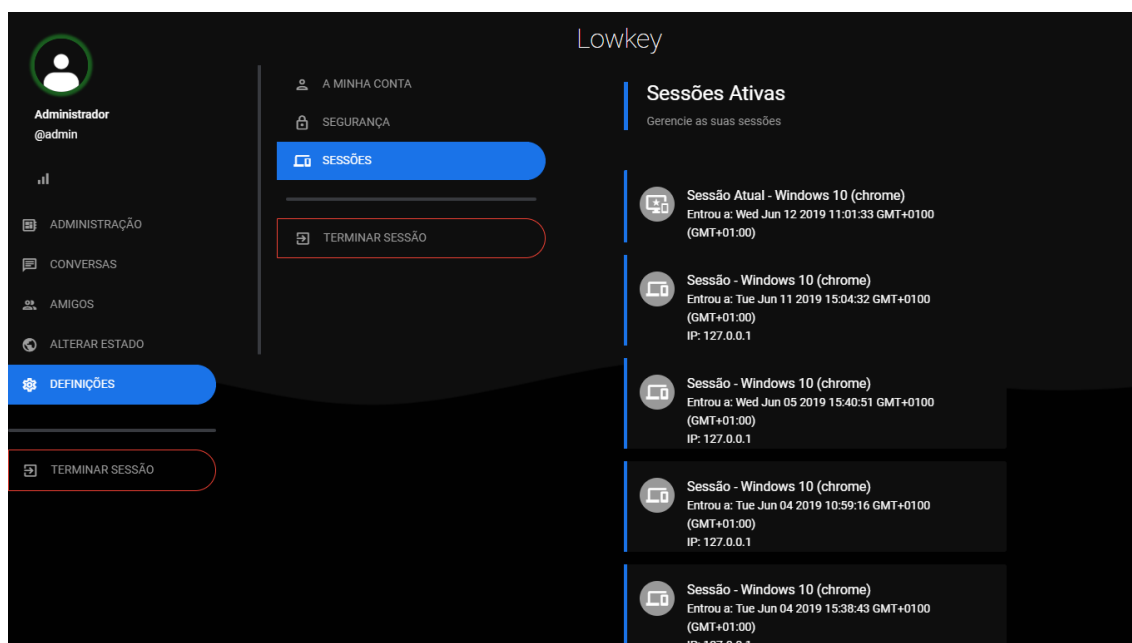


Figura 33 – Sessões Ativas

APLICAÇÃO COM SINCRONIZAÇÃO CROSS-BROWSER

Durante o desenvolvimento da aplicação, foi desenvolvido um serviço de sincronização “Cross-Browser”, que permite ao utilizador estar com duas ou mais sessões abertas ao mesmo tempo, e continuar a receber mensagens ou fazer qualquer outra tarefa na aplicação, sem se preocupar em atualizar a página para receber as últimas informações.

2.4 REFLEXÃO CRÍTICA

Esta reflexão surge no âmbito do trabalho desenvolvido para a prova de aptidão profissional, e tem como principal objetivo apresentar uma reflexão sobre pontos positivos, pontos negativos que neste exercício me proponho.

Este foi realizado, contando com o apoio do professor Paulo Sopa.

Ao realizar este trabalho, posso salientar alguns pontos positivos, entre eles:

- Estudo aprofundado de Javascript;
- Programar em Node.JS;
- Novas tecnologias;
- Segurança a nível web;
- Segurança de servidores;
- Entre outros...

Alguns pontos negativos com que me deparei, embora não fossem muitos, foi a falta de tempo para a realização completa, ou mais aprofundada do trabalho.

Alguns problemas durante a realização do projeto, devido ao facto do mesmo ser desenvolvida em Node.JS, uma tecnologia que não é ensinada no curso, mas que era necessária para a realização do mesmo.

Um dos grandes desafios que tive, foi fazer uma sincronização de todas as sessões abertas pelo utilizador.

Poderia ter corrido melhor, uma melhor divisão lógica do tempo para a realização em partes do projeto, e ter dedicado mais tempo ao mesmo.

Mas no geral, a perspetiva sobre o trabalho é mais que satisfatória, devido á sua complexidade, tempo investido e o produto final.

3. CONCLUSAO

Ao realizar este projeto, pude notar o grande desenvolvimento tecnológico e informático em todos setores. Por isso planeei desenvolver esta aplicação de comunicação e mensagens.

Este projeto deu-me uma grande aprendizagem sobre a programação web, e aperfeiçoamento de algumas técnicas e até conhecimentos novos.

O desenvolvimento do projeto exigiu um grande esforço e dedicação.

Para além das capacidades técnicas, e bases aprendidas no curso, o trabalho ainda exigiu diversas capacidades a descobrir por mim mesmo.

Foi uma experiência muito boa e penso que o balanço final é positivo.

Com isso, penso que este projeto consiga-se expandir ainda mais, e é bastante benéfico para mim a nível profissional, uma vez que no futuro irei abordar projetos parecidos ou com o mesmo nível de dificuldade.

BIBLIOGRAFIA / WEBGRAFIA

- Material Design
<https://material.io/>
- Materialize
<https://materializecss.com/>
- UnDraw
<https://undraw.co/illustrations>
- ExpressJS
<https://expressjs.com/>
- NPM (Node Package Manager)
<https://www.npmjs.com/>
- Moodle
<http://moodle.aeprosa.pt/>