



Licenciatura em Ciências da Computação
UC PROJETO - 'ExploreBraga'

Afonso Cruz - a91639
Henrique Faria - a91637
Pedro Martins - a91681
Rui Gonçalves - a91652

2023-2024

Conteúdo

1	Introdução	4
2	A pilha tecnológica	5
2.0.1	MongoDB	5
2.0.2	ExpressJs	5
2.0.3	React	5
2.0.4	NodeJs	5
3	Requesitos da aplicação	6
4	Arquitetura da aplicação	7
4.1	Esquemas da base de dados (MongoDB)	7
4.1.1	Esquema da coleção de Utilizadores (Users)	7
4.1.2	Esquema da coleção dos Eventos (Events)	9
4.1.3	Esquema da coleção dos Locais (Locations)	11
5	Api de dados utilizando o (ExpressJs)	13
5.1	Caminhos dos Eventos (revent.js)	13
5.2	Caminhos dos Locais (rlocations.js)	14
5.3	Caminhos dos Utilizadores (rusers.js)	14
5.4	Caminhos dos admins (radmin.js)	15
5.5	Caminhos das avaliações (rLocReviews.js)	15
6	Middleware	16
7	APIs / Pacotes externos mais relevantes	17
7.1	Bcrypt - Utilizado no backend	17
7.2	JWT - Utilizado no backend	17
7.3	EmailJS - Utilizado no frontend	17
7.4	Meteorologia - Utilizado no frontend	17
7.5	Vista mapa - Utilizado no frontend	17
7.6	React Snackbars - Utilizado no frontend	17
8	Funcionalidades	18
8.1	Funcionalidades de um utilizador não autenticado	18
8.1.1	Meteorologia - Utilizando o OpenWeather	18
8.1.2	Envio de sugestões - Utilizando o EmailJS	18
8.1.3	Criar conta	19
8.1.4	Iniciar sessão	19
8.1.5	Visualizar locais e eventos em lista	20
8.1.6	Filtrar eventos ou locais por categoria	21
8.1.7	Filtrar eventos ou locais por nome	22
8.1.8	Ver informação de um evento ou local	23
8.1.9	Visualização da localização de um evento ou local num mapa	24
8.1.10	Ver as avaliações de um local específico	25
8.1.11	Partilhar local ou evento	26
8.1.12	Ver eventos que irão acontecer naquele local	27
8.2	Funcionalidades de um utilizador autenticado	28
8.2.1	Adicionar foto de perfil	28
8.2.2	Editar os dados do utilizador	30
8.2.3	Criar local ou evento	31
8.2.4	Os meus locais/eventos	32
8.2.5	Lista de locais favoritos / eventos com interesse	33
8.2.6	Editar local ou evento	34
8.2.7	Escrever uma avaliação num local	35
8.3	Funcionalidades de um utilizador com permissões admin	36
8.3.1	Página admin	36

8.3.2	Menu Admin	37
9	Futuros Updates	38
9.1	Configurar site para telefone	38
9.2	Adicionar um código QR para cada usuário	38
9.3	Introduzir pagamentos	38
9.4	Criar um estatuto empresa	38
10	Conclusão	39

1 Introdução

Este projeto foi feito no âmbito da UC Projeto da licenciatura em Ciências da Computação, na Universidade do Minho. e foi auto-proposto, pelo grupo, uma vez que, o tema ”desenvolvimento web” não é abordado na nossa licenciatura, ser um tema que despertou o nosso interesse e por sabermos haver procura no mercado de trabalho.

A nossa aplicação, ganhou o nome de 'Explore Braga' e consiste numa página web onde pode ser consultada informação sobre locais e eventos a acontecer na cidade de Braga. Caso o utilizador decida, pode criar uma conta e ganhar acesso a mais funcionalidades como criar eventos e locais, poder adicionar avaliações a locais, criar uma lista com os seus locais favoritos e criar uma lista para eventos, aos quais tem interesse em participar.

2 A pilha tecnológica

A pilha tecnológica escolhida é constituída por:

- MongoDB - Base de dados para armazenamento de dados.
- ExpressJs - Framework para criação de APIs e lógica de backend.
- React - Biblioteca para construção da interface de usuário.
- NodeJs - Ambiente de execução para o servidor.

Decidimos utilizar esta pilha, pela facilidade de encontrar soluções e estratégias de resolução de problemas, e porque sentimos ser algo bastante procurado por empresas no mercado de trabalho. O facto de todo o código ser escrito utilizando javascript consideramos que também ajudou.

2.0.1 MongoDB

MongoDB é uma base de dados NoSQL orientada a documentos. Nela podemos armazenar dados em formato de documentos JSON. Ao contrário das base de dados relacionais tradicionais, que armazenam dados em tabelas com linhas e colunas, o MongoDB é mais flexível, permitindo a introdução de dados sem uma estrutura fixa, o que facilita a escalabilidade horizontal e a manipulação de grandes volumes de dados não estruturados.

2.0.2 ExpressJs

ExpressJs é uma framework web para Node.js. Ele fornece um conjunto robusto de funcionalidades para criar aplicativos web e APIs.

2.0.3 React

React é uma biblioteca JavaScript desenvolvida pelo Facebook para construir interfaces de usuário (UI). Ela permite criar componentes reutilizáveis que gerenciam seu próprio estado, fazendo com que a criação de UIs interativas e dinâmicas seja mais eficiente. React utiliza um conceito chamado "virtual DOM" para otimizar a atualização de elementos na página, tornando a experiência do usuário mais rápida e fluida.

2.0.4 NodeJs

NodeJs é um ambiente de execução JavaScript baseado no motor V8 do Chrome, permitindo que JavaScript seja executado no lado do servidor. Ele é conhecido pela sua performance e capacidade de lidar com operações de I/O não bloqueantes, o que o torna ideal para aplicações em tempo real e de alta concorrência.

3 Requisitos da aplicação

Em grupo, delimitámos quais seriam as funcionalidades que gostaríamos ver representadas na nossa aplicação.

Conseguimos chegar às seguintes funcionalidades:

Funcionalidades sem necessidade de login
O utilizador deve poder visualizar a lista de locais já existentes e que se encontrem ativos no momento.
O utilizador deve poder visualizar a lista de eventos já existentes e que se encontrem ativos no momento.
O utilizador deve poder consultar toda a informação disponibilizada sobre os diversos locais e eventos que se encontrem ativos.
O utilizador pode partilhar o local ou evento.
O utilizador deve poder utilizar a lista de filtros disponíveis, para facilitar a busca por locais e eventos.
O utilizador deve ser capaz de ver a informação sobre a meteorologia e a temperatura atual num dado momento na cidade de braga.

| **Funcionalidades com necessidade de login** |

- | |
|---|
| O utilizador deve poder editar dados como 'email' e 'nome de utilizador' e a 'foto de perfil' da sua conta. |
| O utilizador deve poder criar locais e eventos novos, que ficam com o estado de 'pendente' até que um utilizador com estatuto de 'admin' os aprove. |
| O utilizador deve poder editar ou apagar locais e eventos anteriormente criados por ele. |
| O utilizador deve poder adicionar 1 avaliação por cada local. |
| O utilizador deve poder apagar avaliações feitas anteriormente por ele. |
| O utilizador deve poder associar um evento a um local criado por si anteriormente. |

Posto isto, decidimos começar a criar os esquemas da base de dados, que iriam suportar as necessidades da aplicação.

4 Arquitetura da aplicação

4.1 Esquemas da base de dados (MongoDB)

Para conseguir suportar todas as necessidades que delimitámos chegámos à decisão de ter 3 coleções (recordar que uma coleção no MongoDB é um grupo de documentos, ou seja, uma coleção chamada 'Users', será constituído por 1 ou mais users).

As 3 coleções criadas, são '**Users**' (Utilizadores), '**Events**' (Eventos) e '**Locations**' (Locais).

Conseguimos com o auxílio do mongoose (biblioteca de modelagem de dados para MongoDB e NodeJS) criar os esquemas da base de dados.

Podemos também referir que a nossa base de dados está alojada nos servidores do **Atlas**, utilizando uma conta grátis disponibilizada pelo MongoDB.

4.1.1 Esquema da coleção de Utilizadores (Users)

Decidimos então que este esquema deve ter os seguintes atributos:

Atributos	Tipo	Descrição
_id	Object_id	Identificador do utilizador
username	String	Nome do utilizador
email	String	Email do utilizador
hashedPassword	String	Palavra passe encriptada
AdminPermission	Booleano	Verifica se o utilizador é admin
EventCreator	Lista de Object.id (referência aos Eventos)	Lista com os IDs dos eventos criados pelo utilizador
EventHasInterested	Lista de Object.id (referência aos Eventos)	Lista com os IDs dos eventos que o utilizador tem interesse em participar
LocalCreator	Lista de Object.id (referência aos Locais)	Lista com os IDs dos locais criados pelo utilizador
LocalFavorites	Lista de Object.id (referência aos Locais)	Lista com os IDs dos locais favoritos do utilizador
Birthdate	Data	Data de nascimento do utilizador
ProfileImage	String	Conversão da imagem de perfil do utilizador para base64

```
● ● ●
1 // Esquema da Coleção Users
2 const userSchema = new mongoose.Schema({
3   username: {
4     type: String,
5     required: true,
6     unique: true
7   },
8   email: {
9     type: String,
10    required: true,
11    unique: true
12  },
13   HashedPassword: {
14     type: String,
15     required: true
16  },
17   birthDate: {
18     type: Date,
19     required: true
20  },
21   ProfileImage: String,
22   AdminPermission: {
23     type: Boolean,
24     default: false
25  },
26   EventCreator: [
27     type: Schema.Types.ObjectId,
28     ref: 'Events' // Referência ao modelo de evento, remova se não for necessário
29   ],
30   EventHasInterest: [
31     type: Schema.Types.ObjectId,
32     ref: 'Events' // Referência ao modelo de evento, remova se não for necessário
33   ],
34   LocalCreator: [
35     type: Schema.Types.ObjectId,
36     ref: 'Locations' // Referência ao modelo de local, remova se não for necessário
37   ],
38   LocalFavorites: [
39     type: Schema.Types.ObjectId,
40     ref: 'Locations' // Referência ao modelo de local, remova se não for necessário
41   ]
42 });


```

Figura 1: Esquema dos utilizadores

4.1.2 Esquema da coleção dos Eventos (Events)

Decidimos então que este esquema deve ter os seguintes atributos:

Atributos	Tipo	Descrição
_id	Object_id	Identificador do evento
Type	String	Categoria do evento
Name	String	Nome do evento
BegDate	Data	Data de inicio do evento
EndDate	Data	Data de fim do evento
Description	String	Descrição do evento
AgeRecomendation	Inteiro	Idade recomendada para frequentar o evento
Creator	Object_id (referência aos Utilizadores)	Identificador do ID do utilizador que criou o evento
InterestedUsers	Lista de Object_id (referência aos Utilizadores)	Lista com os IDs dos utilizadores que demonstraram interesse
Image	Lista de Strings	Lista com as imagens convertidas para base64
Adress	String	Morada
Price	Inteiro	Preço do evento
Status	Enum ['Pendente', 'Ativo']	Define se o evento está ativo ou pendente de aprovação
LocAssoc	Object_id (referência aos Locais)	Id do local associado ao evento

```
1 // Esquema do Coleção Events
2 const eventSchema = new mongoose.Schema({
3   Type: {
4     type: String,
5     required: true
6   },
7   Name: {
8     type: String,
9     required: true
10 },
11 BegDate: {
12   type: Date,
13   required: true
14 },
15 EndDate: {
16   type: Date,
17   required: true
18 },
19 Description: {
20   type: String,
21   required: true
22 },
23 AgeRecomendation: {
24   type: Number,
25   required: true
26 },
27 Creator: {
28   type: Schema.Types.ObjectId,
29   ref: 'Users'
30 },
31 InterestedUsers: [
32   type: Schema.Types.ObjectId,
33   ref: 'Users'
34 ],
35 Image: [String],
36 Address: {
37   type: String,
38   required: true
39 },
40 Price: {
41   type: Number,
42   required: true
43 },
44 Status: {
45   type: String,
46   enum: ['Pending', 'Active']
47 },
48 LocAssoc: {
49   type: Schema.Types.ObjectId,
50   ref: 'Locations'
51 }
52 });

```

Figura 2: Esquema dos Eventos

4.1.3 Esquema da coleção dos Locais (Locations)

Decidimos então que este esquema deve ter os seguintes atributos:

Atributos	Tipo	Descrição
_id	Object_id	Identificador do local
Type	String	Categoría do local
Name	String	Nome do local
Description	String	Descrição do local
AgeRecomendation	Inteiro	Idade recomendada para frequentar o local
Creator	Object_id (referência aos Utilizadores)	ID do utilizador que criou o local
Reviews	Lista de (Classification, _id, Comment, RevUserId)	Lista das avaliações do local feitas por utilizadores (divididas em 4 parâmetros)
Reviews.Classification	Inteiro	Classificação do local entre 1 e 5
Reviews._id	Object_id	Identificador da avaliação
Reviews.Comment	String	Comentário da avaliação
Reviews.RevUserId	Object_id (Referência aos utilizadores)	Identificador do utilizador que fez a avaliação
Image	Lista de Strings	Lista com as imagens convertidas para base64
Adress	String	Morada
Status	Enum ['Pendente', 'Ativo']	Define se o evento está ativo ou pendente de aprovação
LocEvents	Lista de Object_id (referência aos Eventos)	Lista cos os IDs dos eventos que foram associados a este local

```
1 // Esquema da Coleção Locations
2 const locationSchema = new mongoose.Schema({
3   Type: {
4     type: String,
5     required: true
6   },
7   Name: {
8     type: String,
9     required: true
10 },
11   Description: {
12     type: String,
13     required: true
14 },
15   AgeRecomendation: Number,
16   Creator: {
17     type: Schema.Types.ObjectId,
18     ref: 'Users',
19     required: true
20 },
21   Reviews: [
22     RevUserId: {
23       type: Schema.Types.ObjectId,
24       ref: 'Users'
25     },
26     classification: {
27       type: Number,
28       required: true,
29       min: 1,
30       max: 5
31     },
32     comment: {
33       type: String,
34       required: true
35     }
36   ],
37   Status: {
38     type: String,
39     enum: ['Pending', 'Active'] // As 2 opções válidas para o atributo
      "status"
40   },
41   Image: [String],
42   Address: String,
43   LocEvents: [
44     type: Schema.Types.ObjectId,
45     ref: 'Events'
46   ]
47 });


```

Figura 3: Esquema dos Locais

5 Api de dados utilizando o (ExpressJs)

Apartir do momento em que temos os esquemas da base de dados criados, podemos utilizar o **ExpressJS** para construir uma api de dados, que depois podem ser consultados no frontend, utilizando o **Axios**.

Esta api será **RESTful**, o que significa, que os diferentes caminhos serão divididos em 4 categorias, utilizando os seguintes métodos:

- Get - Serve para receber informação da base de dados.
- Post - Serve para enviar informação para a base de dados.
- Put - Serve para dar atualização da informação da base de dados.
- Delete - Serve para apagar informação da base de dados.

Foram então definidos as seguintes caminhos para a nossa api, separados por categorias. Sendo elas:

- Eventos
- Locais
- Utilizadores
- Admin
- Avaliações

5.1 Caminhos dos Eventos (revent.js)

Aqui serão explicitadas todos os caminhos relacionados com os eventos. Os mesmos terão ainda como prefixo '/events'.

Tipo	Nome	Descrição
GET	/displayAllEvents	Caminho para pedir a informação de todos os eventos a base de dados e ordenar pela data.
GET	/SelectEvents	Caminho para pedir a informação de todos os eventos a base de dados apartir da data atual ordenados.
GET	/:eventId	Caminho para pedir a informação de um evento específico, em que o id do evento iguala o <i>eventId</i> .
GET	/buscarLocaisEvents	Caminho para pedir informação sobre os eventos criados por um utilizador específico.
POST	/create	Caminho para criar um novo evento.
PUT	/interested/:eventId	Caminho para adicionar ou remover o interesse de um utilizador num evento.
PUT	/edit/:eventId	Caminho para editar um evento criado anteriormente.
DELETE	/eventDelet	Caminho para apagar um evento criado anteriormente.

5.2 Caminhos dos Locais (rlocations.js)

Aqui serão explicitadas todos os caminhos relacionados com os locais. Os mesmos terão ainda como prefixo '/locations'.

Tipo	Nome	Descrição
GET	/seebycategories	Caminho para pedir a informação de todos os locais.
GET	/infoToEdit/:localId	Caminho para pedir a informação do local cujo identificador é igual ao <i>localId</i> .
GET	/assocEvents/:localId	Caminho para pedir a informação de eventos associados ao local cujo identificador é igual ao <i>localId</i> .
GET	/:localId	Caminho para pedir informação sobre um local cujo identificador seja igual ao <i>localId</i> .
POST	/create	Caminho para criar um novo local.
PUT	/edit/:localId	Caminho para editar um local criado anteriormente cujo identificador seja igual ao <i>localId</i> .
PUT	/addToFavorites/:locationId	Caminho para adicionar ou remover o <i>locationId</i> à lista de locais favoritos do utilizador.
DELETE	/locationDelete	Caminho para apagar um local criado anteriormente.

5.3 Caminhos dos Utilizadores (rusers.js)

Aqui serão explicitadas todos os caminhos relacionados com os utilizadores. Os mesmos terão ainda como prefixo '/users'.

Tipo	Nome	Descrição
GET	/myaccount	Caminho para pedir os dados da conta do utilizador.
GET	/showEventsUser	Caminho para pedir a informação dos eventos criados por um utilizador.
GET	/showLocationsUser	Caminho para pedir informação dos locais criados por um utilizador.
GET	/userInterestedEvents	Caminho para pedir informação sobre um local cujo identificador seja igual ao <i>localId</i> .
GET	/userFavoritePlaces	Caminho para pedir informação sobre um local cujo identificador seja igual ao <i>localId</i> .
POST	/createNewUser	Caminho para criar uma nova conta de utilizador.
POST	/login	Caminho para dar login.
PUT	/updateAccount	Caminho para alterar o email, nome de utilizador ou a imagem de perfil.

5.4 Caminhos dos admins (radmin.js)

Aqui serão explicitadas todos os caminhos relacionados com os admins. Os mesmos terão ainda como prefixo '/admin'.

Método	Rota	Descrição
GET	/displayAllUsers	Caminho para obter informações sobre todos os utilizadores registados.
GET	/events	Caminho para obter informações sobre todos os eventos registados.
GET	/locations	Caminho para obter informações sobre todos os locais registados.
PUT	/eventAction/:eventId/:action	Caminho para realizar uma ação em um evento (aceitar, cancelar ou deixar pendente).
PUT	/locationAction/:localId/:action	Caminho para realizar uma ação em um local (aceitar, cancelar ou deixar pendente).
DELETE	/eventDelete	Caminho para excluir um evento.
DELETE	/locationDelete	Caminho para excluir um local.

5.5 Caminhos das avaliações (rLocReviews.js)

Aqui serão explicitadas todos os caminhos relacionados com as avaliações dos utilizadores aos locais. Os mesmos terão ainda como prefixo '/locations/reviews/'.

Método	Rota	Descrição
POST	/addReview/:localId	Caminho para adicionar uma avaliação para um local específico.
DELETE	/deleteReview	Caminho para excluir uma avaliação de um local.
GET	/:localid	Caminho para obter as avaliações de um local específico.

6 Middleware

O middleware **authPage** é uma parte fundamental da nossa aplicação, responsável por garantir a segurança das rotas. Quando um utilizador faz uma requisição para uma rota protegida, este middleware é acionado para verificar se o utilizador está autenticado e autorizado a acessar o recurso.

O processo inicia-se com a extração do token JWT do cabeçalho da solicitação. Esse token é então verificado e decodificado para obter as informações do utilizador, como seu ID e email. Com base nessas informações, o middleware consulta o banco de dados para obter mais detalhes sobre o utilizador, como suas permissões de administrador.

Se o utilizador for válido e autorizado, as informações pertinentes, como as permissões de administrador e o ID do utilizador, são adicionadas ao objeto req, que representa a requisição. Isso permite que as rotas subsequentes acessem essas informações, por exemplo, para determinar se um utilizador tem permissão para executar determinada ação.

Por outro lado, se o utilizador não for válido ou não estiver autorizado, o middleware define a propriedade authenticated do objeto req como false e passa o controle para o próximo middleware na cadeia. Isso impede que utilizadores não autorizados acessem recursos protegidos, garantindo a integridade e a segurança da nossa aplicação.

7 APIs / Pacotes externos mais relevantes

7.1 Bcrypt - Utilizado no backend

O **Bcrypt** é uma biblioteca que tem o propósito de esconder palavras passe criadas por utilizadores na forma de texto simples utilizando o algoritmo hash.

O funcionamento desta ferramenta utiliza parâmetros importantes: o '*salt*' e as '*saltRounds*'.

No caso do '*salt*', é uma sequência aleatória de caracteres que são acrescentados à palavra passe escolhida pelo utilizador, o que resulta numa palavra passe criptografada e aumenta a segurança contra ataques de força bruta, como o *rainbow tables*. Isto é possível porque, o hash gerado será sempre diferente, mesmo que a palavra passe seja igual.

No caso do *saltRounds* é uma propriedade que pode ser alterada, de acordo com o nível de segurança pretendido e da capacidade de processamento da máquina, onde quanto maior o número atribuído, mais lento será o processamento para calcular o hash associado à palavra passe.

7.2 JWT - Utilizado no backend

A escolha para tecnologia de autenticação colmatou no JWT (Jason Web Token) devido à sua simplicidade, segurança e flexibilidade.

Jwt é um mecanismo de autorização que permite a troca segura de informações entre duas partes.

Os JWTs consistem em 3 partes:

- Cabeçalho: Contém informações sobre o token, como o algoritmo usado para assinatura e o tipo de token.
- Payload: Carrega as reivindicações do usuário, como sua identidade, permissões e outra informações relevantes.
- Assinatura: Garante a integridade do token sendo gerado usando uma chave secreta compartilhada entre o cliente e o servidor.

7.3 EmailJS - Utilizado no frontend

Decidimos que faria sentido ter uma caixa de denúncia de erros ou envio de sugestões por parte dos utilizadores diretamente para nós. A melhor forma foi introduzir um serviço de email e acabámos por escolher o EmailJS, ficando sempre disponível no rodapé do site, uma caixa de texto em que um utilizador (não precisa de estar autenticado) pode deixar a sua mensagem.

7.4 Meteriologia - Utilizado no frontend

Considerámos que também seria interessante ter disponibilizada a condição meteriológica da cidade de Braga e a temperatura em graus Celsius. Para isso utilizámos a api do '*OpenWeather*'. A informação é atualizada a cada 60 segundos.

7.5 Vista mapa - Utilizado no frontend

Uma outra funcionalidade interessante é a disponibilização de um serviço que representa a localização de um local ou evento com vista de mapa. Neste caso, decidimos utilizar o '*Open Street Map*'.

7.6 React Snackbars - Utilizado no frontend

Para melhorar a experiência do utilizador, decidimos incorporar este serviço que demonstra que envia um pop-up personalizado (em termos de cor e mensagem) quando alguma funcionalidade é realizada de forma correta ou não.

8 Funcionalidades

O *front-end* da aplicação foi conseguido, utilizando React e CSS. Nesta secção, iremos demonstrar através de imagens e de pequenas descrições as funcionalidades da nossa aplicação.

8.1 Funcionalidades de um utilizador não autenticado

8.1.1 Meteorologia - Utilizando o OpenWeather



Figura 4: Funcionalidade da meteorologia

Como referido em cima, utilizando a API do '*Open Weather*' conseguimos expor a temperatura e a condição meteorológica na cidade de Braga.

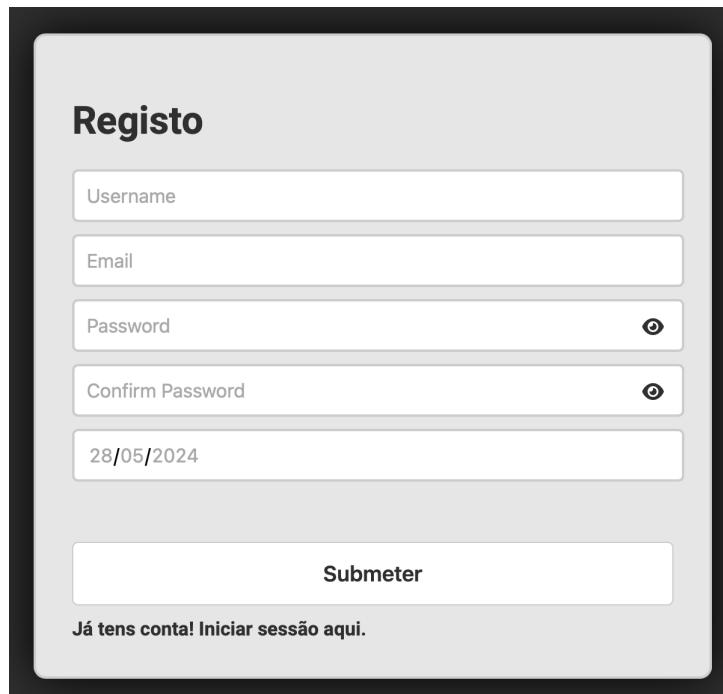
8.1.2 Envio de sugestões - Utilizando o EmailJS



Figura 5: Funcionalidade de envio de sugestões

Como referido anteriormente, utilizando a API do '*EmailJS*' conseguimos receber mensagens com sugestões ou erros encontrados dos utilizadores do website.

8.1.3 Criar conta



The image shows a registration form titled "Registo". It contains five input fields: "Username", "Email", "Password" (with a visibility icon), "Confirm Password" (with a visibility icon), and a date field showing "28/05/2024". Below the fields is a "Submeter" button. At the bottom, there is a link: "Já tens conta! Iniciar sessão aqui."

Figura 6: Criar uma nova conta

8.1.4 Iniciar sessão



The image shows a login form titled "Iniciar sessão". It contains three input fields: "Email", "Password" (with a visibility icon), and a checkbox labeled "Lembra-me". Below the fields is an "Entrar" button. At the bottom, there is a link: "Registrar conta".

Figura 7: Iniciar sessão numa conta criada anteriormente

8.1.5 Visualizar locais e eventos em lista

Sábado - Quinta 25/05 - 30/05/2024	Desporto teste3 22:12 - 23:13	braga 14,00 € Afonso10
Quinta 30/05/2024	Educacao Workshop de programação 15:29 - 16:29	Universidade do minho, campus de gualter 1,00 € user123
Terça 11/06/2024	Educacao Workshop De Programação em C 16:31 - 18:31	Universidade do Minho - Campus de Gualter Grátis jordan21pt

Figura 8: Visualizar lista de eventos ou locais

Aqui representámos uma funcionalidade para ambos os locais e os eventos, que consiste numa reprentação de todos os eventos (similar para os locais) em formato de lista.

No caso, dos eventos temos uma função extra, implementada através de um cron job, que todos os dias às 24h, verifica eventos cuja data terminou à 7 dias e elemina-os de forma a libertar espaço na base de dados. Esta rotina de manutenção automática é crucial para garantir que a plataforma BragaExplore continue a funcionar de maneira eficiente e organizada, proporcionando aos utilizadores uma experiência de uso fluida e rápida.

8.1.6 Filtrar eventos ou locais por categoria

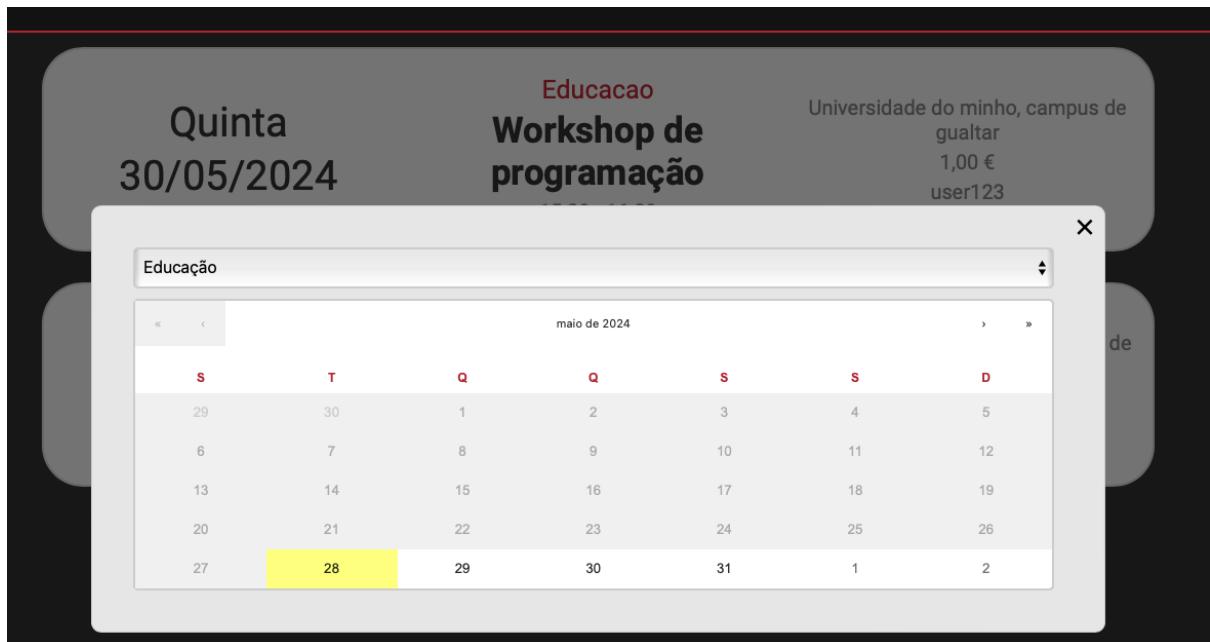


Figura 9: Filtrar lista de eventos ou locais por categorias

Uma das funcionalidades tanto dos locais como dos eventos, é filtrar por categoria. E no caso, dos eventos, ainda tem um extra, poder visualizar apenas eventos que irão acontecer após uma data especificada pelo utilizador.

8.1.7 Filtrar eventos ou locais por nome



Figura 10: Filtrar lista de eventos ou locais por nome

Outra possibilidade para filtrar tanto eventos como locais, é pelo nome.

8.1.8 Ver informação de um evento ou local

Universidade do Minho - Campus de Gualtar



Categoria: Educação

A Universidade do Minho é uma instituição pública de ensino superior fundada em 1973 na cidade de Braga, integrando-se no grupo das "Novas Universidades", iniciando as suas actividades académicas em 1975/76.

Morada: Rua da Universidade, 4710-057 Braga

Idade recomendada: 18

 jordan21pt (Criador)

Figura 11: Visualizar a informação de um local ou evento

Nesta página é possível visualizar a informação detalhada, tanto dos locais como dos eventos, com pequenas diferenças entre eles.

No caso, as diferenças mais significativas, será a adição das classificações apenas na página dos locais, que irá ser demonstrada mais abaixo.

8.1.9 Visualização da localização de um evento ou local num mapa

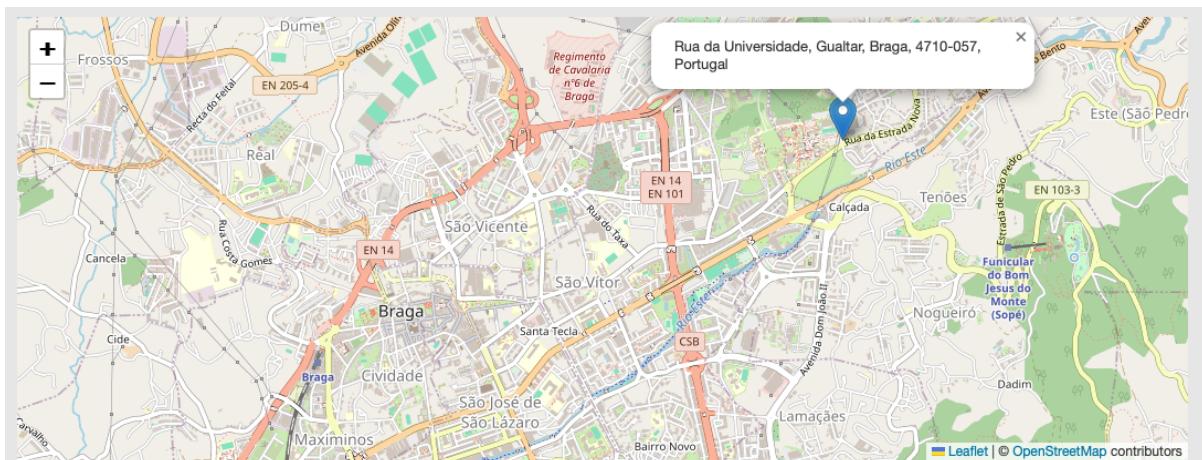


Figura 12: Visualização da localização de um evento ou local em mapa

Como referido anteriormente, utilizando o Open Street Map conseguimos disponibilizar ao utilizador, uma vista da localização tanto dos locais como dos eventos num mapa, de modo a facilitar a sua descoberta.

8.1.10 Ver as avaliações de um local específico



Figura 13: Visualização de avaliações na página de um local

Como descrito acima, na página onde são disponibilizado as informações dos locais é também disponibilizado um espaço de avaliações.

No caso, um utilizador não autenticado apenas pode consultar as avaliações deixadas por outros utilizadores.

8.1.11 Partilhar local ou evento

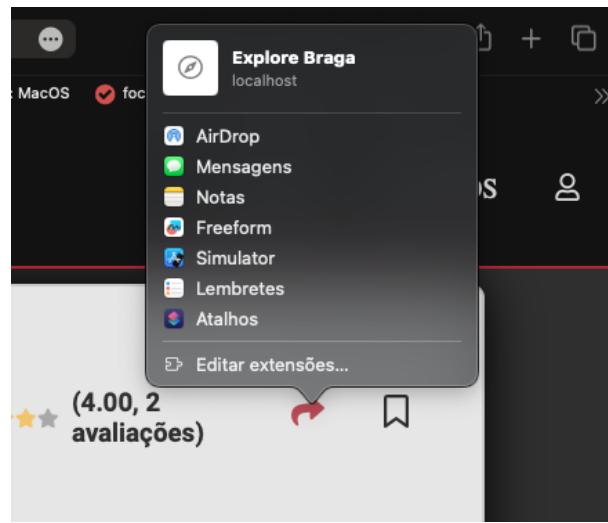


Figura 14: Partilha de um local ou evento

Qualquer visitante da aplicação tem a possibilidade de partilhar a página de um local ou evento com outras pessoas através de um click. (Notar que: neste momento, esta funcionalidade apenas existe para o *Safari*)

8.1.12 Ver eventos que irão acontecer naquele local

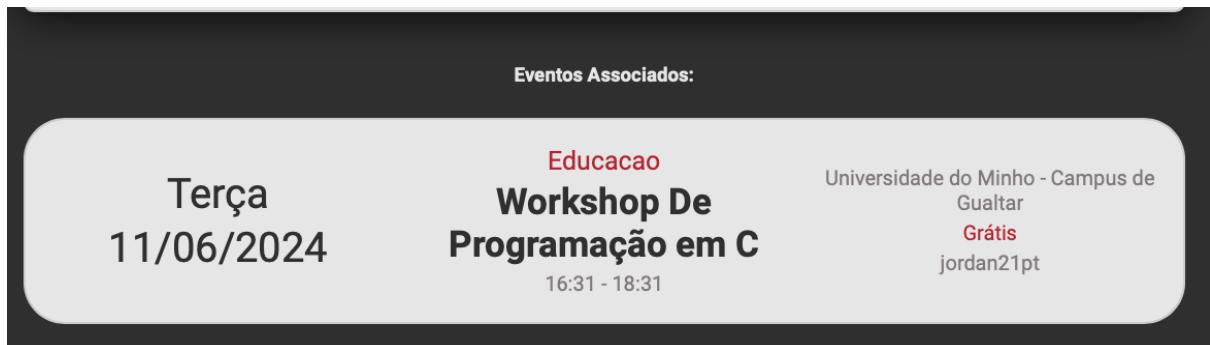


Figura 15: Visualização da eventos associados a um local

No final da página de um local, o dono desse local, ao criar um novo evento, pode e deve associar o evento ao local, ficando assim a informação disponibilizada para todos os utilizadores após o clique de um botão.

8.2 Funcionalidades de um utilizador autenticado

Aqui serão descritas as funcionalidades que um utilizador ganha por criar conta / iniciar sessão. De referir que continua a ter acesso às funcionalidades de um utilizador que não esteja autenticado.

8.2.1 Adicionar foto de perfil

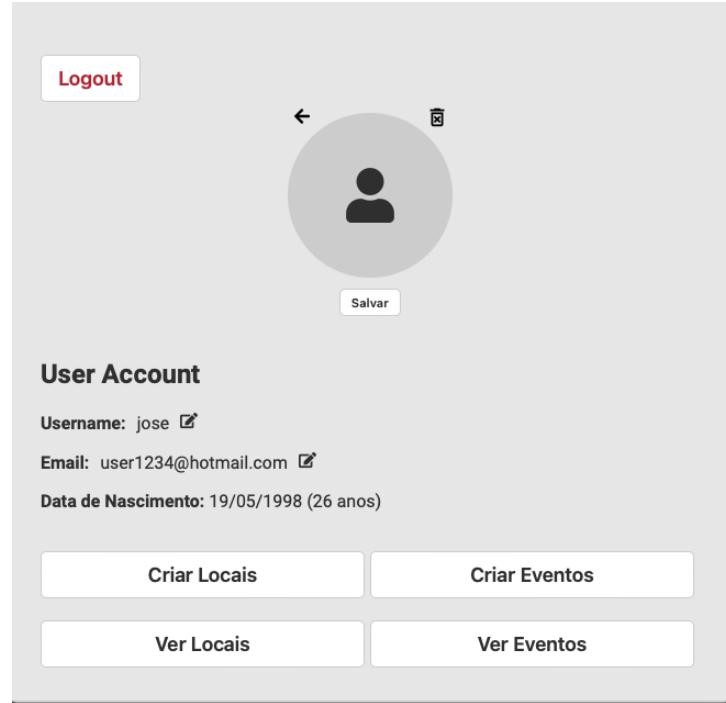


Figura 16: Adicionar foto de perfil - passo 1

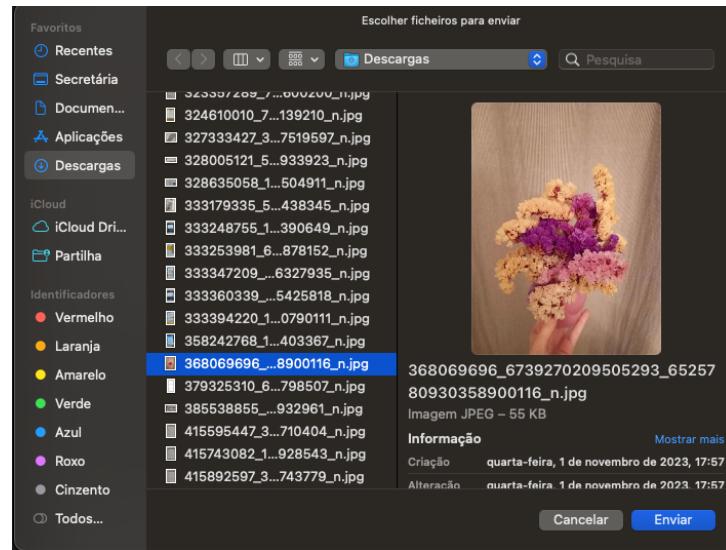


Figura 17: Adicionar foto de perfil - passo 2

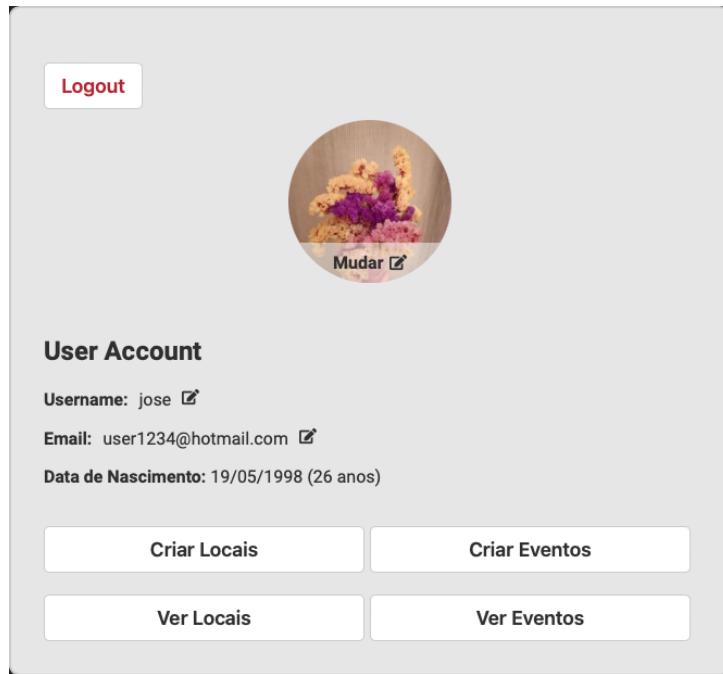


Figura 18: Adicionar foto de perfil - passo 3

Após um utilizador criar uma conta ou iniciar sessão, fica com acesso ao menu do utilizador, onde é possível fazer a adição de uma foto de perfil que depois será partilhada no website, quando é disponibilizada a informação de um local que ele seja o dono, ou quando é realizada uma avaliação.

8.2.2 Editar os dados do utilizador

The screenshot shows a user profile edit page. At the top is a placeholder profile picture with a 'Mudar' (Change) link below it. Below the picture is the section title 'User Account'. Underneath are three fields: 'Username: jose' with a pencil icon, 'Email: user1234@hotmail.com' with a pencil icon, and 'Data de Nascimento: 19/05/1998 (26 anos)'. At the bottom are four buttons: 'Criar Locais', 'Criar Eventos', 'Ver Locais', and 'Ver Eventos'.

Figura 19: Editar os dados - 1

This screenshot is identical to Figure 19, except the 'Username' field now contains 'joao' instead of 'jose'. The other fields and buttons remain the same.

Figura 20: Editar os dados - 2

This screenshot is identical to Figure 20, but includes a 'Salvar' (Save) button next to the 'Username' input field. The rest of the interface is the same.

Figura 21: Editar os dados - 3

Aqui podemos ver a funcionalidade de editar os dados do utilizador, como *username*, *email* e *foto de perfil*. Apenas fazemos demonstração para o *username*, o procedimento seria idêntico para os outros campos. Na primeira foto, o utilizador tem o *username* de 'jose' e faz a alteração para 'joao'. É ainda verificado se o novo *username* ou *email* já estão sendo utilizados.

8.2.3 Criar local ou evento

The screenshot shows a web-based form titled 'Crie o seu local ...'. The form fields include:

- A dropdown menu labeled 'Escolha uma categoria'.
- A text input field labeled 'Nome'.
- A text area labeled 'Descrição'.
- A dropdown menu labeled 'Idade recomendada'.
- A text input field labeled 'Endereço'.
- A button labeled '+Adicionar fotos'.
- A large 'Submeter' button at the bottom.

Figura 22: Criar um local novo

Um utilizador autenticado deve poder criar um local ou evento, preenchendo um formulário como o representado na foto, ficando este em estado 'pendente' enquanto não for verificado por um utilizador com permissão de admin.

No caso do evento, surge ainda a opção de associar o próprio a um local que seja da autoria do mesmo utilizador.

8.2.4 Os meus locais/eventos

Meus Locais				
Nome do Evento	Tipo	Endereço	Estado	Ações
Universidade do Minho - Campus de Gualtar	Educação	Rua da Universidade, 4710-057 Braga	Active	<button>Edit</button> <button>Delete</button>
Santuário do Sameiro	MonumentosHistóricos	Avenida de Nossa Senhora do Sameiro, 56 4715-616 Braga, Portugal	Active	<button>Edit</button> <button>Delete</button>
Nova Arcada	Compras	Avenida de Lamas 100, 4700-038 Braga	Active	<button>Edit</button> <button>Delete</button>

Figura 23: Visualizar os locais ou eventos de um utilizador

A qualquer momento, fará sentido que um utilizador possa verificar a lista dos seus eventos ou locais, para facilitar a gestão dos mesmos. Ganharia aqui, neste menu, a opção de poder editar ou mesmo apagar um evento ou local.

8.2.5 Lista de locais favoritos / eventos com interesse

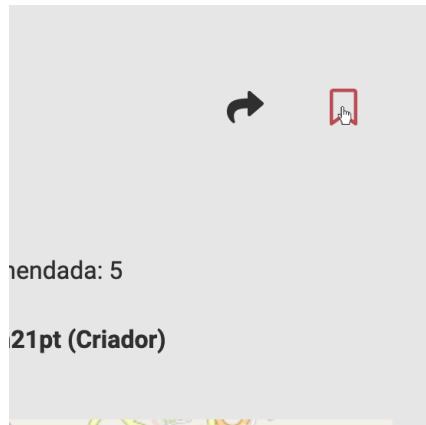


Figura 24: Adicionar local à lista de favoritos - 1

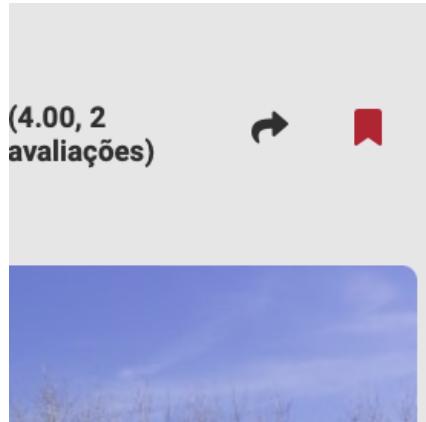


Figura 25: Adicionar local à lista de favoritos - 2



Figura 26: Adicionar local à lista de favoritos - 3

Qualquer utilizador autenticado, pode guardar um local (numa lista de favoritos) ou um evento (numa lista de interesse), que mais tarde, podem ser consultadas, como um dos filtros na página de visualização dos locais e dos eventos.

8.2.6 Editar local ou evento



Figura 27: Editar local / evento - 1

A screenshot of a "Editar local" (Edit location) form. The form has several input fields: a dropdown menu set to "Educação", a text input for the name "Universidade do Minho - Campus de Gualtar", a text area with a description of the university, a dropdown for age recommendation set to "18", a text input for the address "Rua da Universidade, 4710-057 Braga", and a button "+Adicionar fotos" (Add photos). Below the form is a section titled "Fotos do Local" (Local photos) showing three small thumbnail images of the university campus.

Figura 28: Editar local / evento - 2

Apartir do botão editar, na página "Os meus locais" ou "Os meus eventos" surge então a opção de editar um local ou evento. A informação atual é pedida à base de dados e disponibilizada ao utilizador, pré-preenchendo assim os campos, para que este possa verificar todos os dados atuais durante a atualização dos dados.

8.2.7 Escrever uma avaliação num local

The screenshot shows a user interface for leaving a review. At the top, there is a rating section with five stars, one of which is highlighted in yellow, followed by the text '(1 Estrela)'. Below this is a text input field with the placeholder 'Conta-nos a tua opinião...'. Underneath the input field is a 'Submeter' button. Below the form, the word 'Avaliações' is displayed in bold black text, followed by a horizontal line.

Figura 29: Fazer avaliação num local

Como referido anteriormente esta funcionalidade é exclusiva dos locais, sendo preciso o utilizador estar autenticado. E convém também referir que quando um utilizador escreve uma avaliação, deixa de o poder voltar a fazer no mesmo local até que apague a primeira, evitando assim o 'spam' de avaliações.

8.3 Funcionalidades de um utilizador com permissões admin

Este será o tipo de utilizador que terá o máximo de permissões na nossa aplicação. No caso, tem todas as funções dos 2 níveis descritos acima (utilizador autenticado e não autenticado) e ainda acrescenta as seguintes.

8.3.1 Página admin

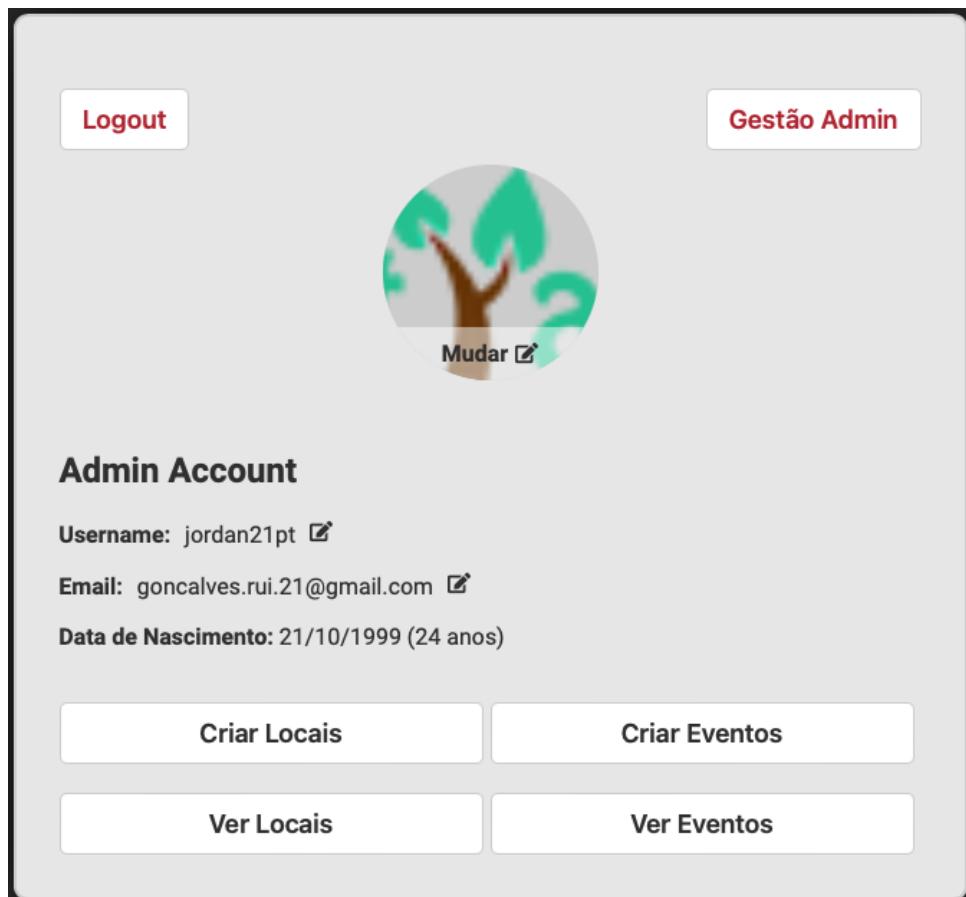


Figura 30: Pagina de admin

Um utilizador com perfil de admin ganha um novo menu dentro da aplicação através do botão de *Gestão Admin*.

8.3.2 Menu Admin

Eventos Locais Usuários						
<input type="text" value="Pesquisar por evento ..."/> <input type="text" value="Pesquisar por estado"/> Abrir Filtros						
Nome do Evento	Tipo	Data de Início/fim	Estado	Username	Ações	
teste3	Desporto	25/05/2024 - 30/05/2024	Active	Afonso10	✓ Aceitar	🕒 Pendente
Workshop de programação	Educacao	30/05/2024 - 30/05/2024	Active	user123	✓ Aceitar	🕒 Pendente
Workshop De Programação em C	Educacao	11/06/2024 - 11/06/2024	Active	jordan21pt	✓ Aceitar	🕒 Pendente
Explorando a Serenidade no Jardim de Santa Barbara	Turismo	28/06/2024 - 29/06/2024	Active	Afonso10	✓ Aceitar	🕒 Pendente
Campeonato de jogos mentais	Desporto	07/07/2024 - 07/07/2024	Active	user123	✓ Aceitar	🕒 Pendente
Torneio de Padel "Desafio nas Quadras" não eliminar	Desporto	06/12/2024 - 06/12/2024	Active	Afonso10	✓ Aceitar	🕒 Pendente
landna	Desporto	10/12/2024 - 10/12/2024	Active	jordan21pt	✓ Aceitar	🕒 Pendente
Pending event	Cultura	13/12/2024 - 14/12/2024	Active	user123	✓ Aceitar	🕒 Pendente

Figura 31: Menu de admin

Neste menu para admins, é possível visualizar os vários utilizador, eventos e locais, em qualquer estado, em formato lista, e através do botão de *Detalhes* pode ver os detalhes de um evento ou local específico.

Um admin pode ainda, decidir, aceitar um evento que tenha sido criado por outro utilizador, colocá-lo pendente (o que faz com o que evento deixe de aparecer na lista falada acima), ou mesmo apagá-lo.

9 Futuros Updates

Com o objetivo de melhorar continuamente a experiência dos nossos utilizadores e aumentar a funcionalidade do nosso site, temos planeadas várias atualizações para o futuro. Estas melhorias visam tornar a navegação mais intuitiva, facilitar a interação com o conteúdo e oferecer novas funcionalidades que respondam às necessidades dos utilizadores. Caso decidamos continuar a desenvolver o projeto, estas atualizações serão fundamentais para garantir o crescimento e a relevância da plataforma. Abaixo, detalhamos algumas das principais atualizações que estamos a planear implementar:

9.1 Configurar site para telefone

Atualmente o site fica desformatado quando é acessado por telefone, apenas a página **Home** está adaptada corretamente para visualização em dispositivos móveis. Furturamente gostaríamos de implementar um design responsivo para todas as páginas, garantindo uma experiência de usuário consistente e otimizada em todos os tipos de dispositivos

9.2 Adicionar um código QR para cada usuário

Para promover uma maior dinâmica e interação com os utilizadores, temos em mente adicionar um código QR para cada utilizador. Assim, ao comparecer a um evento publicitado no site, o dono desse evento poderá fazer scan do código como confirmação de presença, e o utilizador ganhará pontos. Ainda estamos a determinar para que servirão esses pontos, mas temos algumas ideias, como oferecer descontos em eventos pagos ou até mesmo criar um ranking entre os usuários.

9.3 Introduzir pagamentos

Para eventos pagos, seria benéfico que o site tivesse uma funcionalidade de pagamento, simplificando o processo para o utilizador.

9.4 Criar um estatuto empresa

No caso de negócios de grande dimensão, seria vantajoso permitir que vários utilizadores fossem associados ao mesmo negócio, com permissões específicas para editar a informação sobre o local. Esta funcionalidade permitiria uma gestão mais eficiente, facilitando a atualização de detalhes sobre eventos, promoções e outras informações relevantes. Além disso, poderia incluir a possibilidade de definir diferentes níveis de acesso, garantindo que apenas utilizadores autorizados possam fazer alterações críticas, enquanto outros podem apenas visualizar ou editar informações menos sensíveis.

10 Conclusão

Sentimos que este trabalho foi bastante desafiador, pois representou a nossa primeira experiência coletiva com desenvolvimento web. No entanto, conseguimos atingir um bom resultado, cumprindo na totalidade os requisitos que definimos para a aplicação BragaExplore.

Durante o desenvolvimento, enfrentamos vários obstáculos técnicos e conceptuais, que exigiram uma abordagem proativa e colaborativa. Tivemos que nos familiarizar com diversas tecnologias e ferramentas, desde a criação e gestão de bases de dados até ao desenvolvimento de funcionalidades complexas como autenticação de utilizadores e permissões administrativas.

A rotina de manutenção automática, que verifica e elimina eventos antigos, achamos que escalando para um grande número de eventos será essencial para manter a base de dados organizada e eficiente, prevenindo a acumulação de dados desnecessários e garantindo um desempenho otimizado do sistema, ficando assim já preparado para ser disponibilizado online.

Em suma, o resultado final deste projeto foi extremamente recompensador. Não só cumprimos os objetivos propostos, como também superámos as nossas próprias expectativas em termos de aprendizagem e desenvolvimento de competências. Este projeto pôs à prova as nossas capacidades de auto-aprendizagem e resolução de problemas, o que é algo fundamental para qualquer finalista da licenciatura em Ciências da Computação. A experiência adquirida ao longo deste processo será, sem dúvida, valiosa para os nossos futuros projetos e carreiras profissionais.