

2023-2024

P.PORTO

**ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO**

PROGRAMAÇÃO EM AMBIENTE WEB
MILESTONE 1
TRABALHO PRÁTICO – ÉPOCA NORMAL

Trabalho elaborado por:

Grupo 19

8220169 – César Ricardo Barbosa Castelo

8220337 – Hugo Ricardo Almeida Guimarães

8220307 – Pedro Marcelo Santos Pinho

Índice

Índice de Figuras	2
Chave de Siglas	3
1. Introdução	4
a. Contextualização	4
b. Apresentação do caso de estudo	4
1. Estrutura do Projeto	5
2. Estrutura dos dados	6
3. Funcionalidades Implementadas	8
4. Especificidades do projeto.....	9
a. Separação da API do BackOffice	9
b. Verificação do token de login	9
c. Atribuição de Pontos	9
d. Inserção de imagens.....	9
e. Alteração de doações	9
5. Rotas.....	11
a. Login	11
b. Administradores.....	11
c. Estatísticas	13
Imagens da aplicação	14
Conclusão	17

Índice de Figuras

Figura 1 - Diagrama de casos e uso para os utilizadores do BackOffice	5
Figura 2 - Estrutura dos dados de todos os utilizadores	6
Figura 3 - Estrutura de dados para as entidades	6
Figura 4 - Estrutura de dados para as doações.....	7
Figura 5 - Estrutura de dados para os itens.....	7
Figura 6 - Diagrama de Sequência do processo de alteração de uma doação	10
Figura 7 - Diagrama de atividades da verificação do token na API.....	11
Figura 8 - Diagrama de sequência para a listagem de administradores	11
Figura 9 - Diagrama de sequência para a inserção de administradores	12
Figura 10 - Diagrama de sequência para a edição de administradores	12
Figura 11 - Diagrama de sequência para a remoção de administradores.....	12
Figura 12 - Diagrama de sequência para a listagem dos países dos doadores ou entidades	13
Figura 13 - Diagrama de sequência com a quantidade de doações por ano ou por mês	13
Figura 14 - Página de login para a área de administração	14
Figura 15 - Listagem das entidades.....	14
Figura 16 - Formulário para a inserção de uma entidade	15
Figura 17 - Formulário para alteração de uma doação	15
Figura 18 - Mapa geográfico com os países dos doadores	16
Figura 19 - Gráfico de barras com os meses em que há mais doações	16

Chave de Siglas

RGPD	Regime Geral de Proteção de Dados
JSON	Java Script Object Notation
API	Application Programming Interface
CSV	Comma Separated Values
PDF	Portable Document Format

1. Introdução

a. Contextualização

Este trabalho foi desenvolvido para o âmbito da disciplina de Programação em Ambiente Web, que funcionará como integrador dos conhecimentos adquiridos no decorrer das aulas, sendo que este relatório reflete apenas o trabalho realizado para a primeira das duas entregas a fazer.

b. Apresentação do caso de estudo

A empresa de valorização social “recicla têxtil”, em parceria com o fundo ambiental, pretende reforçar a promoção da valorização e reciclagem de peças de têxteis pós-consumo. Neste contexto, é pedido o desenvolvimento de uma aplicação WEB para agilizar a recolha de peças de têxtil que os utilizadores já não usam para reaproveitamento e envio para as instituições de cariz social, instituições beneficiadoras.

A plataforma será de uso livre por parte dos utilizadores que recebem pontos convertíveis em vales de desconto em lojas de que apadrinham esta ideia. Em termos práticos, a aplicação contemplará um BackOffice de acesso restrito aos administradores/funcionários e deverá contemplar todas as funcionalidades de gestão da plataforma de doações e gestão dos pontos de entrega pelos funcionários da empresa “recicla Têxtil”. Numa futura fase existirá um FrontOffice onde os utilizadores podem registar envios via postal de peças têxtil e verificar o seu estado de pontos. O FrontOffice poderá também ser utilizado pelas instituições de cariz social para registo na plataforma e verificar os donativos recebidos.

A recolha de têxteis poderá ser realizada presencialmente num ponto de recolha ou via postal, com validação após receção via postal na empresa “recicla têxtil” por funcionários da empresa validando o donativo emitindo pontos para o utilizador doador e adicionando prova fotográfica do donativo recebido.

1. Estrutura do Projeto

O sistema do BackOffice consiste em diferentes módulos e funcionalidades que são acessíveis a diferentes tipos de utilizadores. O diagrama UML presente na Figura 1 representa os casos de uso para os administradores e os funcionários da aplicação desenvolvida para a Recicla Têxtil, destacando as funcionalidades específicas disponíveis para cada um.

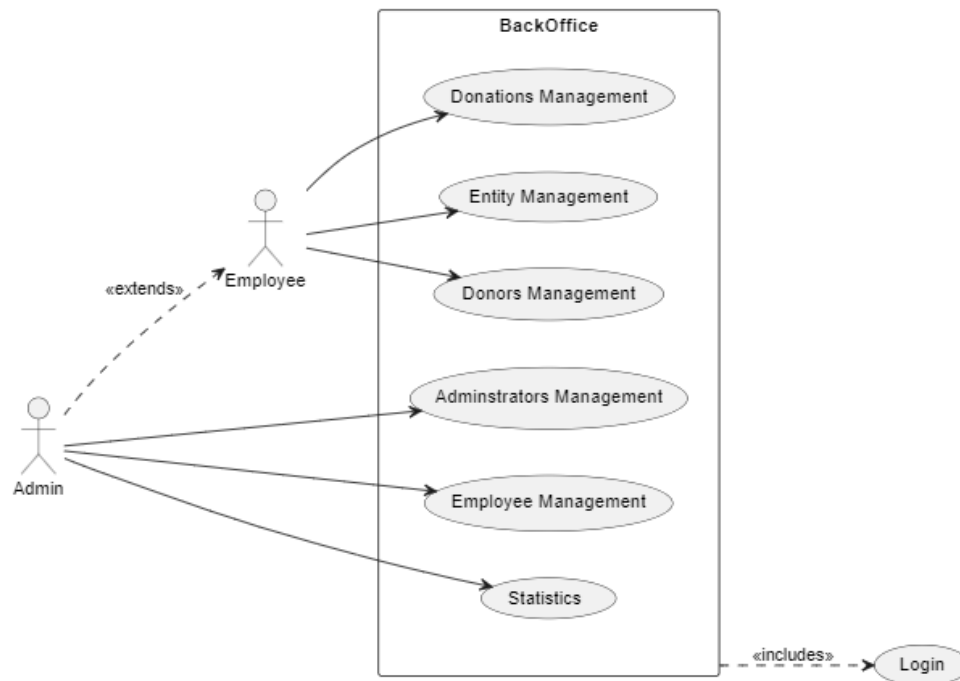


FIGURA 1 - DIAGRAMA DE CASOS E USO PARA OS UTILIZADORES DO BACKOFFICE

2. Estrutura dos dados

```
const userSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  role: { type: String, enum: ['admin', 'employee', 'donor'], required: true },
  phoneNumber: { type: String, required: false },
  address: { type: String, required: false },
  country: { type: String, required: true },
  points: { type: Number, default: 0 },
  dateOfBirthday: { type: Date, required: false },
  profileImage: { type: String, default: 'default_image.png' },
  isDeleted: { type: Boolean, default: false },
  created_at: { type: Date, default: Date.now, required: true },
});
```

FIGURA 2 - ESTRUTURA DOS DADOS DE TODOS OS UTILIZADORES

Os administradores, funcionários e doadores usam todos a mesma estrutura de dados, esta abordagem tem diversas vantagens, como por exemplo a escalabilidade, pois facilmente se cria novos utilizadores, e consegue-se facilmente adaptar o código existente de modo a encaixar o novo tipo de utilizador, no entanto, existe uma enorme desvantagem quanto aos campos que se pode colocar, pois, ao ter uma estrutura partilhada para todos os utilizadores, a estrutura de dados pode tornar-se excessivamente genérica e pode não ser otimizada para atender às necessidades específicas de cada tipo de utilizador. Isso pode resultar em campos desnecessários para determinados tipos de utilizadores ou na falta de campos importantes para outros, levando a uma sobrecarga de informações ou a uma falta de dados relevantes. Nesta estrutura pode ser visto a partir do campo “*points*”, onde implica que um administrador e funcionário também podem fazer doações, o que provavelmente não irá acontecer quando o FrontOffice for executado, havendo assim um campo inútil para os administradores e funcionários.

```
const entitySchema = new mongoose.Schema({
  name: { type: String, required: true },
  contact: {
    email: { type: String, required: true },
    phoneNumber: { type: String, required: false },
    address: { type: String, required: false }
  },
  description: { type: String, required: true },
  password: { type: String, required: true },
  country: { type: String, required: true },
  additionalInfo: { type: String, required: false },
  pointsGivenPerKg: { type: Number, require: true }, // pontos que a entidade dá por kilograma doado
  isActive: { type: Boolean, default: false }, // Indica se a entidade está ativa ou desativada
  isAccepted: { type: Boolean, default: false }, // Indica se a entidade foi aceita por um administrador ou não
  isDeleted: { type: Boolean, default: false },
  createdAt: { type: Date, default: Date.now }
});
```

FIGURA 3 - ESTRUTURA DE DADOS PARA AS ENTIDADES

Como as entidades têm campos muito diferentes, foi decidido separá-las da estrutura de dados dos utilizadores, existindo assim uma maior liberdade para adicionar campos e personalizar a estrutura de dados de acordo com as necessidades específicas de cada tipo de entidade. Isso permite uma maior flexibilidade na modelagem dos dados e garante que cada entidade possa ter os campos necessários para representar adequadamente as suas características e funcionalidades únicas.

```
const donationSchema = new mongoose.Schema({
  donorId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },
  entityId: { type: mongoose.Schema.Types.ObjectId, ref: 'Entity', required: true },
  numberOfItems: { type: Number, required: true },
  pointsGiven: { type: Number, default: 0 },
  status: { type: String, enum: ['received', 'delivered', 'lost'], default: 'received' }
});
```

FIGURA 4 - ESTRUTURA DE DADOS PARA AS DOAÇÕES

O esquema das doações define a estrutura de dados para registrar as doações dentro do sistema. Cada doação é caracterizada pelo identificador do doador, o identificador da entidade beneficiadora, o número de itens doados, os pontos atribuídos e o estado da doação. Esses campos fornecem informações cruciais sobre a doação, facilitando o acompanhamento e a gestão eficiente das doações realizadas.

```
const itemSchema = new mongoose.Schema({
  donationId: { type: mongoose.Schema.Types.ObjectId, ref: 'Donation', required: true },
  description: String,
  condition: String,
  weight: { type: Number, min: 0, required: true }
});
```

FIGURA 5 - ESTRUTURA DE DADOS PARA OS ITEMS

De modo a evitar problemas futuros, decidiu-se separar os itens das doações, onde cada item está associado à doação de onde foi originado, proporcionando uma estrutura mais organizada e facilitando a gestão individualizada de cada item dentro do contexto das doações.

Nota: Tirando as doações, onde este conceito não é tão evidenciado, todas as estruturas têm um campo *“isDeleted”* que indica se o registo foi apagado ou não, desta forma evita-se problemas de integridade referencial, onde poderiam existir doações para doadores que não existem mais. De modo a seguir normas do direito ao esquecimento do RGPD, sempre que se coloca o campo *“isDeleted”* como verdadeiro, todos os campos que poderiam de alguma forma identificar o utilizador são colocados a nulo, ou substituídos por textos genéricos, quando a nulo estes não podem ficar.

3. Funcionalidades Implementadas

Para a realização da primeira milestone, foram pedidas um conjunto de funcionalidades para a administração do site da empresa Recicla Têxtil, sendo elas:

- Autenticação
- Gestão de administradores
- Gestão de Funcionários
- Gestão de Entidades
- Gestão de Doadores
- Gestão de Doações

Para além destas funcionalidades, foram adicionados alguns extras de modo que o BackOffice ficasse mais completo, sendo elas:

- Exportação das doações para um ficheiro CSV
- Exportação das doações para um ficheiro PDF
- Gráfico geográfico com os países das entidades
- Gráfico geográfico com os países dos doadores
- Gráfico de barras com a quantidade de doações feitas em cada mês
- Gráfico de barras com a quantidade de doações feitas em cada ano

4. Especificidades do projeto

a. Separação da API do BackOffice

Para a segunda milestone, será necessário construir o FrontOffice da aplicação, que irá usar como auxílio uma API que também seria criada para o âmbito da segunda milestone, no entanto, de modo a tornar a aplicação mais modular, decidiu-se separar a API do BackOffice, onde todas as operações que necessitam de comunicar com a base de dados são realizadas na API, enquanto as operações de carregar as páginas com informação e enviá-las ao utilizador serão realizadas no BackOffice. A API está na porta 5000 enquanto o BackOffice está na porta 3000

b. Verificação do token de login

A validação do token de autenticação funciona como um middleware entre cada rota chamada, esse middleware existe tanto no BackOffice como na API, no entanto, no BackOffice o middleware apenas verifica se o token de autenticação está definido ou não, caso não esteja envia o utilizador para a página de autenticação. Já o middleware da API, verifica se o token é válido, caso seja vai verificar se o ID do utilizador guardado dentro do token existe dentro da base de dados, se existir, então a operação pedida será executada.

Para que haja uma maior segurança, decidiu-se que a validação do token do administrador estaria em um middleware diferente daquele usado para os funcionários.

c. Atribuição de Pontos

Durante o processo de criação de uma entidade, é necessário definir a quantidade de pontos que ela irá conceder em troca de cada quilograma de doações recebidas de um doador. Esses pontos desempenham um papel crucial na motivação dos doadores e na promoção da participação ativa na doação. No entanto, reconhecemos que a abordagem seguida ainda é insuficiente, logo, como trabalho futuro iremos adicionar mais detalhes para o sistema de atribuição de pontos, visando uma abordagem mais precisa e personalizada na recompensa aos doadores pelo apoio.

d. Inserção de imagens

A inserção de imagens diretamente na base de dados é algo muito pouco aconselhável, portanto para poder guardar as imagens decidiu-se que sempre que uma imagem de perfil ia ser guardada, ia-se chamar uma função presente na pasta /utils que gera um nome único para a imagem, e apenas esse nome será guardado na base de dados, já a imagem, será guardada na pasta /images/uploads. Futuramente, gostaríamos de poder guardar as imagens num servidor de ficheiros como o Dropbox ou o Google Cloud Storage.

e. Alteração de doações

Uma das principais diferenças do código das doações para as outras rotas, é que esta precisa de consultar e alterar mais do que uma coleção para poder operar, pois para poder adicionar uma doação é necessário adicionar vários items na coleção de items; alterar os pontos do doador na coleção de utilizadores; e ainda é necessário obter informação acerca da distribuição dos pontos na coleção entidades.

Para fazer a alteração de uma doação, decidiu-se apagar todos os items da doação existente e colocar os que foram alterados, o problema com essa abordagem é que se se inserir menos items do que os inseridos originalmente, ou a soma dos alterados for menor dos que existiam, a quantidade de pontos iria ser menor, mas como os pontos já forma atribuídos ao utilizador quando a doação foi criada, não seria muito correto estar a remover os pontos que o mesmo recebeu previamente, para lidar com este problema decidimos usar o algoritmo de Ostrich e ignorar esse fator.

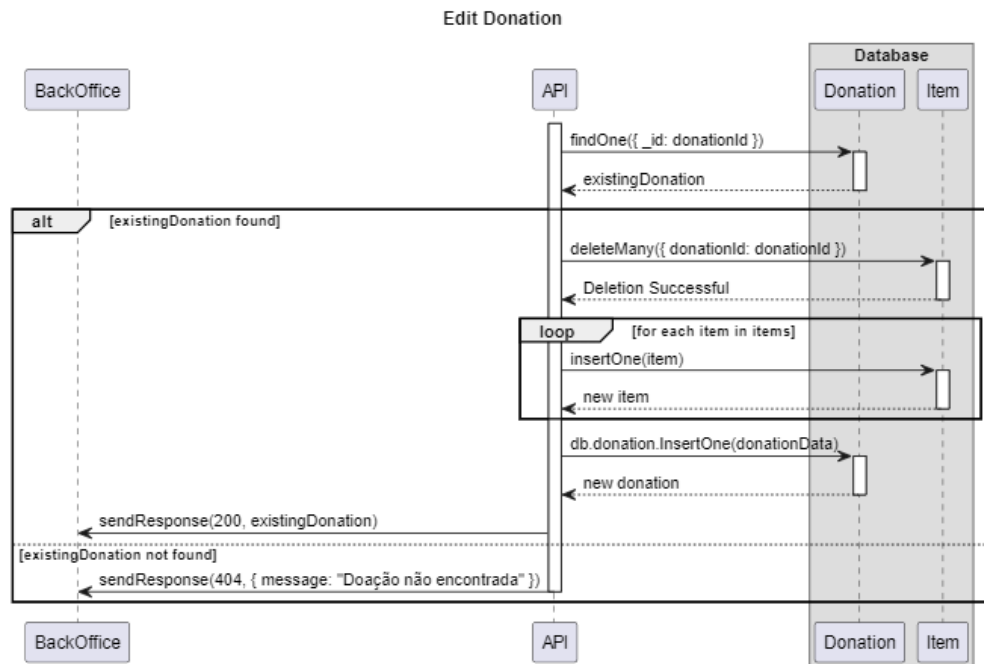


FIGURA 6 - DIAGRAMA DE SEQUÊNCIA DO PROCESSO DE ALTERAÇÃO DE UMA DOAÇÃO

5. Rotas

a. Login

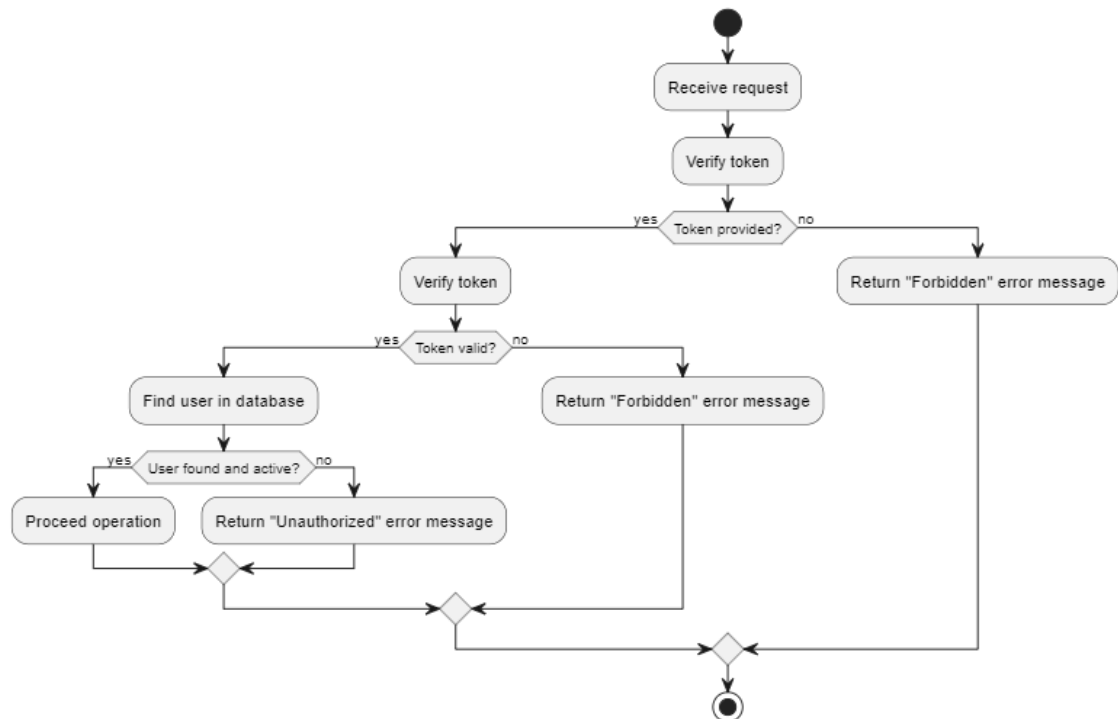


FIGURA 7 - DIAGRAMA DE ATIVIDADES DA VERIFICAÇÃO DO TOKEN NA API

b. Administradores

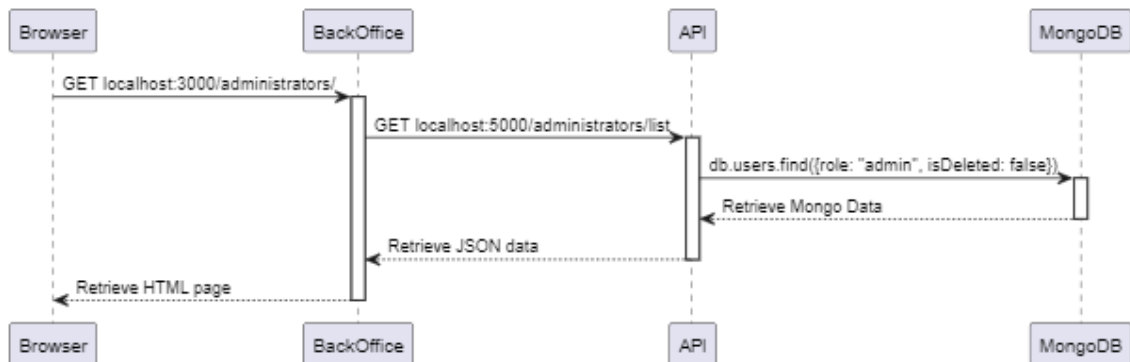


FIGURA 8 - DIAGRAMA DE SEQUÊNCIA PARA A LISTAGEM DE ADMINISTRADORES

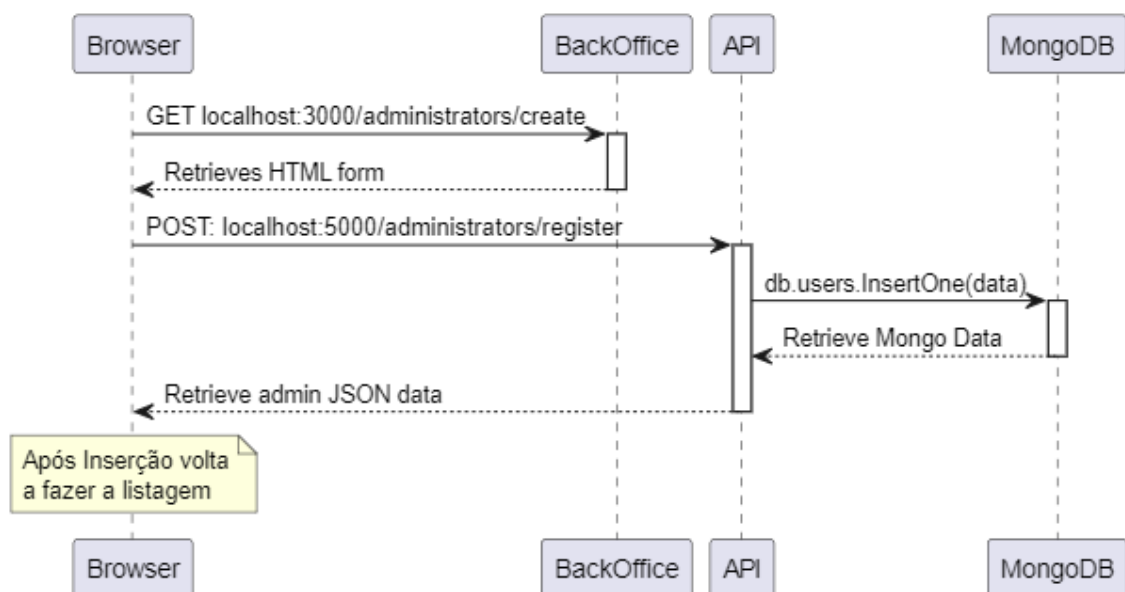


FIGURA 9 - DIAGRAMA DE SEQUÊNCIA PARA A INSERÇÃO DE ADMINISTRADORES

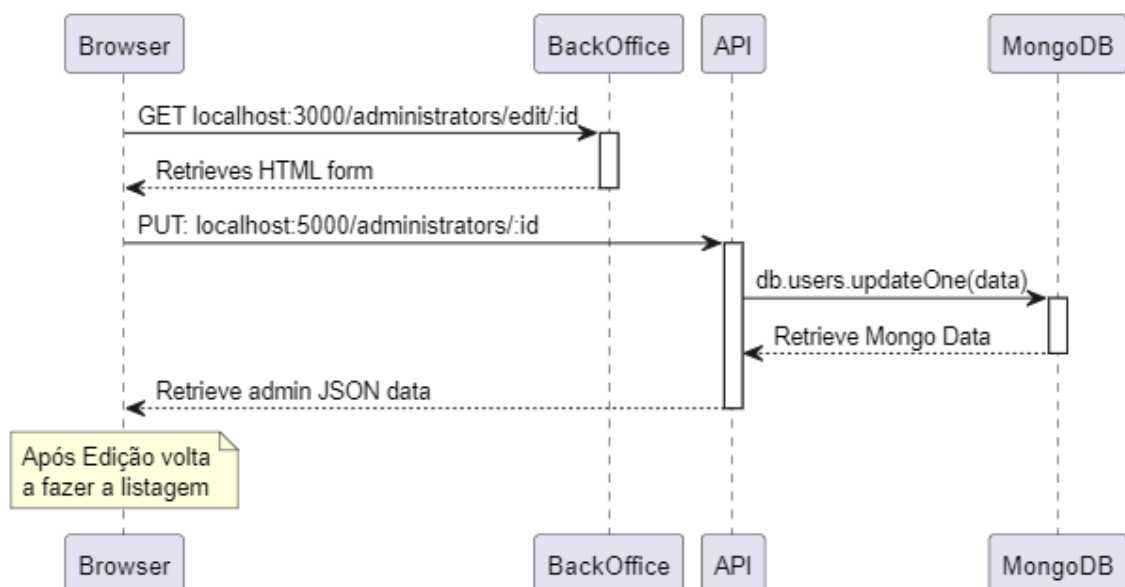


FIGURA 10 - DIAGRAMA DE SEQUÊNCIA PARA A EDIÇÃO DE ADMINISTRADORES

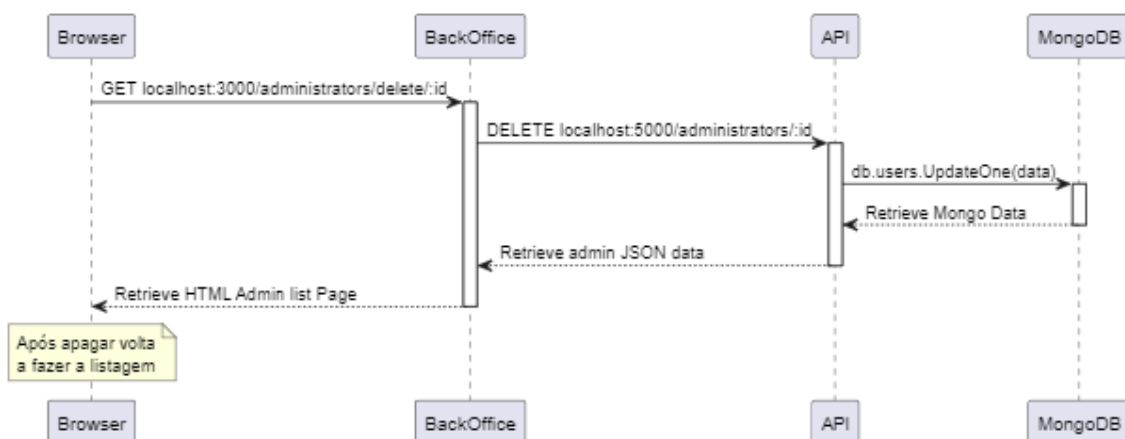


FIGURA 11 - DIAGRAMA DE SEQUÊNCIA PARA A REMOÇÃO DE ADMINISTRADORES

c. Estatísticas

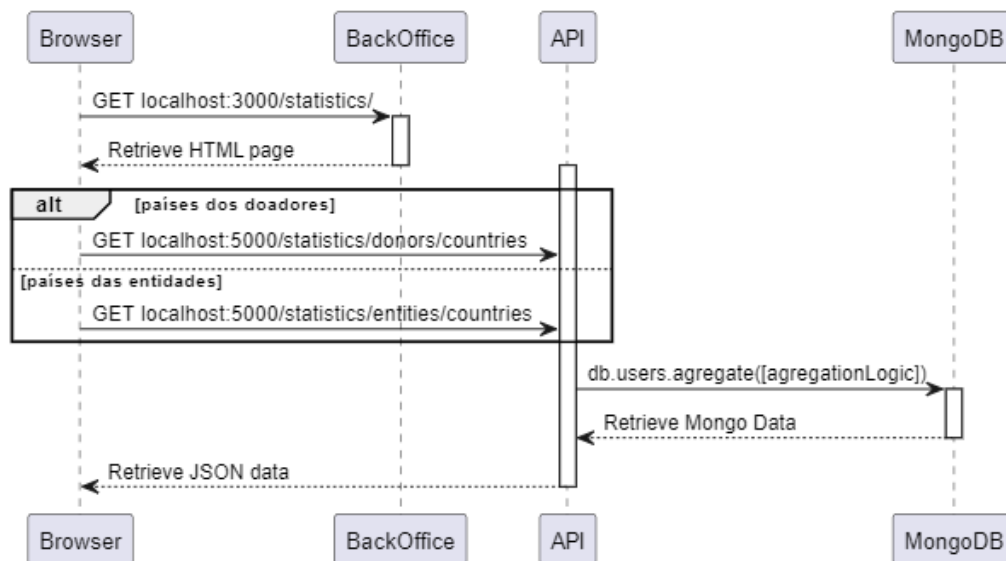


FIGURA 12 - DIAGRAMA DE SEQUÊNCIA PARA A LISTAGEM DOS PAÍSES DOS DOADORES OU ENTIDADES

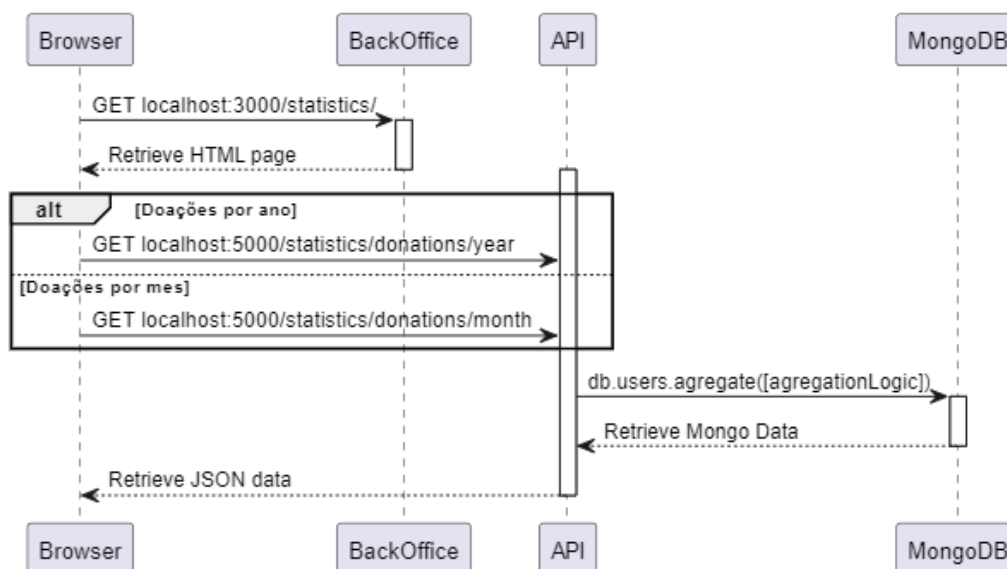


FIGURA 13 - DIAGRAMA DE SEQUÊNCIA COM A QUANTIDADE DE DOAÇÕES POR ANO OU POR MÊS

Não foram criados diagramas de sequência com o comportamento da gestão de funcionários, doadores, doações e entidades pois elas funcionam da mesma forma que a gestão de administradores, mudando apenas o nome das rotas que envés de começarem com a palavra “administrators”, começam com o respetivo nome: “employees”, “donors”, “entities” e “donations”.

Imagens da aplicação

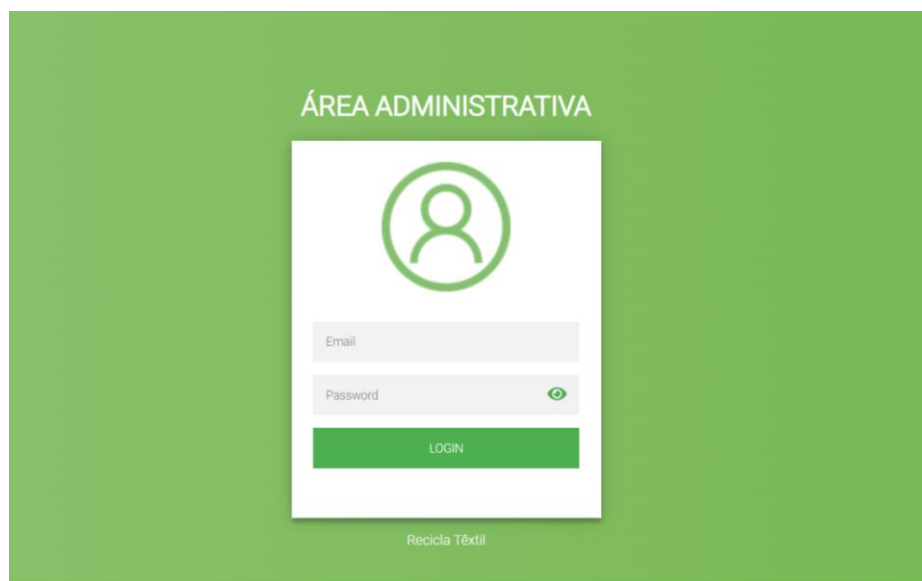


FIGURA 14 - PÁGINA DE LOGIN PARA A ÁREA DE ADMINISTRAÇÃO






<div>  Administradores Funcionários Entidades Beneficiadoras Doadores Doações Estatísticas </div>									
Aceite: <input type="text" value="-"/> <input type="button" value="Pesquisar"/>		Atividade: <input type="text" value="-"/> <input type="button" value="Pesquisar"/>		<input type="button" value="Adicionar"/>					
Nome	Email	Telemóvel	Morada	Pais					
entity	entity@entity.com	962123526	Rua das entidades	Egypt	Ativo	Aceite			
Deus ex	deusEx@entity.com	912243567	Rua do Ceu	Madagascar	Ativo	Aceite			

FIGURA 15 - LISTAGEM DAS ENTIDADES



The screenshot shows a web form titled "Adicionar Entidade" (Add Entity) on a green background. The form is white and contains several input fields: "Nome:" (Name), "Email:", "Password:", "Telefone:" (Phone), "Pontos dados por kilograma doado:" (Points given per kilogram donated), "Morada:" (Address), "País:" (Country) with a dropdown menu, "Descrição:" (Description), and "Informação adicional:" (Additional information). At the bottom left is a red "Voltar" (Back) button, and at the bottom right is a green "Adicionar Entidade" (Add Entity) button.

FIGURA 16 - FORMULÁRIO PARA A INSERÇÃO DE UMA ENTIDADE



The screenshot shows a web form titled "Editar Doação" (Edit Donation) on a light gray background. The form is white and contains several input fields: "Doador Desejado:" (Desired Donor) with a dropdown menu showing "Pedro", "Entidade Desejada:" (Desired Entity) with a dropdown menu showing "Deus ex", "Itens:" (Items) section with "Descrição:" (Description), "Condição:" (Condition) with a dropdown menu showing "-", and "Peso(KG):" (Weight in KG). Below the "Itens:" section is a blue "Adicionar item" (Add item) button. At the bottom left is a red "Voltar" (Back) button, and at the bottom right is a green "Atualizar Doação" (Update Donation) button. A status bar at the bottom shows "Nelson | Desgaste mínimo | Peso: 2 | X".

FIGURA 17 - FORMULÁRIO PARA ALTERAÇÃO DE UMA DOAÇÃO

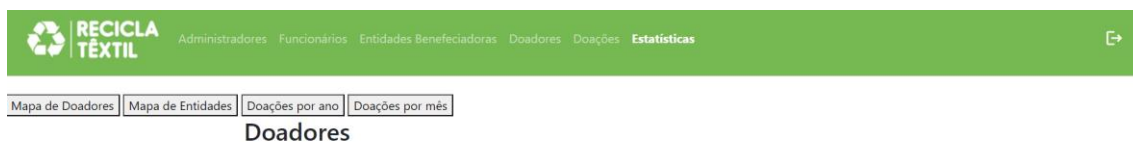


FIGURA 18 - MAPA GEOGRÁFICO COM OS PAÍSES DOS DOADORES

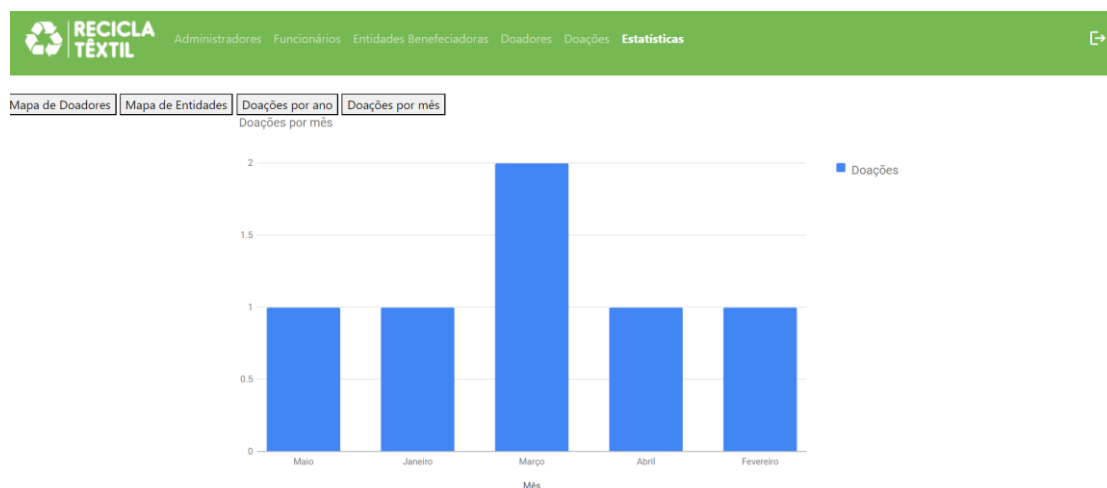


FIGURA 19 - GRÁFICO DE BARRAS COM OS MESES EM QUE HÁ MAIS DOAÇÕES

Conclusão

Embora ainda falte mais uma entrega, achamos que o trabalho até agora foi bem concebido, que conseguimos cumprir com tudo o que foi pedido, e ainda conseguimos acrescentar mais funcionalidades aquelas propostas inicialmente, no entanto, ainda gostaríamos de ter adicionado muitas mais, um exemplo de algo que gostaríamos de ter implementado é o envio de emails aos utilizadores criados com as credenciais de acesso dos mesmos.

Mesmo com um trabalho bem concebido, temos a consciência que ainda existe muitos pontos de melhoria, como por exemplo: a lógica para a atribuição de pontos aos doadores que realizam doações é insuficiente, os pontos atribuídos são apenas baseados no número de quilogramas doados; quando se edita ou remove uma doação os pontos que foram atribuídos ao utilizador não são alterados; embora a API o permita, não existe nenhuma interface no BackOffice que permita a alteração da password de um utilizador.