

Projeto de Programação em Ambiente Web

Licenciatura em Segurança Informática em Redes de Computadores

2023/2024

Realizado por:

Alexandre Rocha – 8210649

David Santos – 8220651

Nuno Gomes – 8220652

Índice

| | |
|-------------------------|-----------|
| Índice | 1 |
| Introdução | 2 |
| Contextualização | 2 |
| Backoffice | 3 |
| Instruções | 3 |
| Mockups | 3 |
| Diagramas de Classe | 4 |
| Estrutura | 5 |
| Demonstração | 7 |
| API | 10 |
| Controladores e Rotas | 10 |
| Swagger | 11 |
| Angular | 12 |
| Mockups | 12 |
| Serviços | 15 |
| Dados | 16 |
| Demonstração | 16 |
| Conclusão | 22 |

Introdução

Este relatório visa apresentar e analisar o trabalho desenvolvido no âmbito da disciplina de Programação em Ambiente Web (PAW). O foco principal do projeto foi a implementação de um sistema web utilizando tecnologias modernas e práticas recomendadas no desenvolvimento de aplicações web. Durante o processo, foram explorados diversos aspectos do desenvolvimento web, incluindo front-end, back-end, integração com APIs e segurança da informação. A escolha das tecnologias e abordagens foi fundamentada em critérios de eficiência, escalabilidade e manutenção.

Contextualização

A disciplina de Programação em Ambiente Web tem como objetivo proporcionar aos alunos uma compreensão abrangente das tecnologias e práticas utilizadas no desenvolvimento de aplicações web. No contexto deste trabalho, foram abordados tópicos essenciais como HTML, CSS, JavaScript, frameworks front-end (neste caso Angular), desenvolvimento de APIs RESTful com Node.js.

A importância de desenvolver aplicações web robustas e seguras é cada vez mais evidente no cenário atual, onde a presença online é crucial para o sucesso de negócios e iniciativas. Este projeto permitiu aos alunos aplicar teorias aprendidas em sala de aula num ambiente prático, simulando desafios reais encontrados na indústria.

Backoffice

Instruções

Para garantir a acessibilidade ao projeto desenvolvido pelo nosso grupo, é crucial criar instruções abrangentes para os desenvolvedores que venham a trabalhar nele no futuro. Essas instruções devem ser claras e abrangentes, abordando todos os aspectos essenciais do projeto. Abaixo estão as diretrizes a serem seguidas:

1. Linha de comandos:
 - a. No *Visual Studio Code (VSC)*, usar o atalho “*ctrl + c*” (“*ctrl + ^*” em *Apple OS*) para abrir um terminal;
 - b. Inserir “*cd backend*” seguido de “*npm i*” para instalar as dependências necessárias para o projeto;
 - c. Para ligar o servidor, inserir “*npm start*”;
2. No browser:
 - a. Na barra de navegação inserir “*localhost:3000*”;
 - b. Para teste como utilizador com privilégios de administrador, usar as seguintes credenciais: email - aa@aa.aa | pass - aa ;
 - c. Existem vários outros utilizadores já criados, e as suas credenciais seguem o esquema do anterior por ordem alfabética.

Mockups

O mockup apresenta o layout de um dashboard do website, que inclui várias funcionalidades e informações essenciais para a gestão de doações. Na barra lateral esquerda, encontramos um menu com as seguintes opções: Dashboard, Doações, Entidades, Funcionários, e Sistema de Pontos.

Na área principal do dashboard, há uma visão geral da semana que exibe:

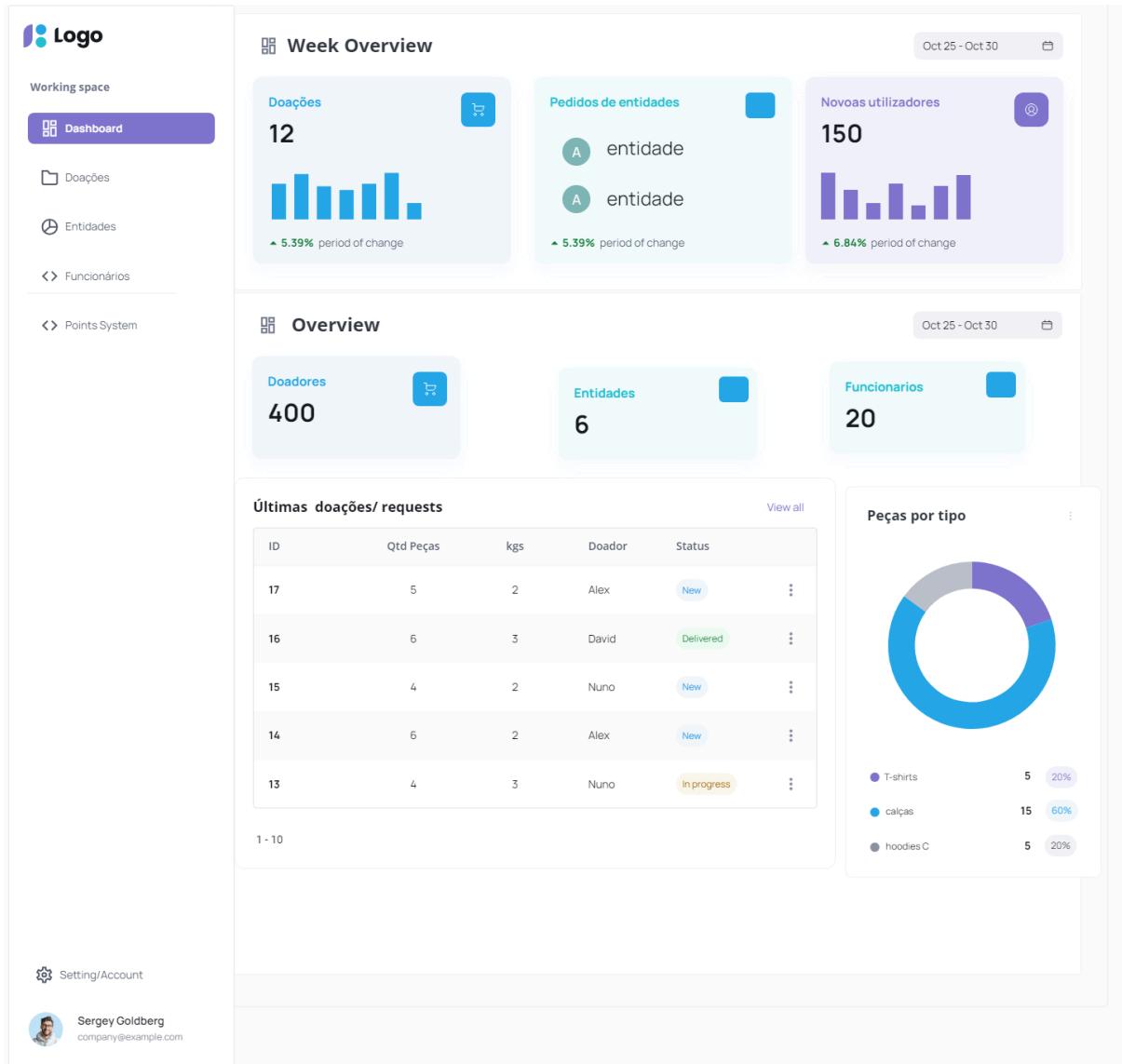
- Doações realizadas, com um total de 12 doações, e um gráfico de barras.
- Pedidos de entidades, listando duas entidades.
- Novos utilizadores, com um total de 150, e um gráfico de barras.

Abaixo, há uma visão geral adicional que mostra:

- Doadores, com um total de 400.
- Entidades, totalizando 6.
- Funcionários, com um total de 20.

Além disso, uma tabela exibe as últimas doações/requests, detalhando a quantidade de peças, o peso em kgs, o doador, e o status da doação. Também há um gráfico circular que mostra a distribuição das peças por tipo (T-shirts, calças, hoodies).

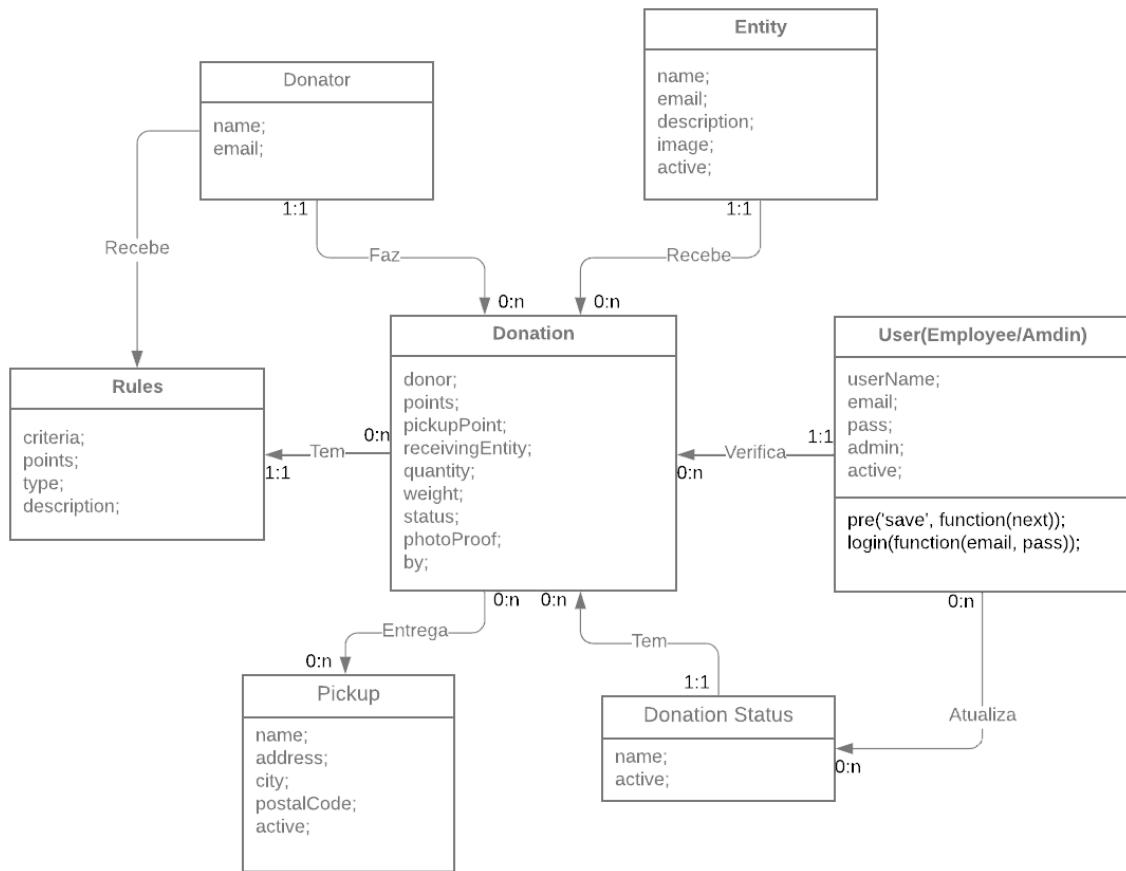
Com base neste mockup, nosso grupo de trabalho concluiu que ele já fornece uma ideia clara de como o website deve ficar, tornando desnecessária a criação de mais mockups.



Diagramas de Classe

Os diagramas de classe são fundamentais no desenvolvimento de software orientado a objetos, oferecendo uma representação visual da estrutura do sistema. No nosso projeto de Programação em Ambiente Web, utilizamos estes diagramas para organizar e definir claramente as entidades e suas inter-relações, facilitando a compreensão e o planeamento da arquitetura do sistema.

Estes diagramas ilustram as classes, e os relacionamentos entre elas, proporcionando uma visão detalhada das interconexões e colaborações dentro do sistema. Ao servir como um guia para a codificação e uma ferramenta de comunicação eficaz entre os membros da equipe, os diagramas de classe garantiram que todos compartilhassem uma compreensão comum da estrutura do software.



Uma doação segue um conjunto de regras específicas que determinam a atribuição de pontos para cada contribuição efetuada. O doador realiza a doação e, conforme as diretrizes estabelecidas, recebe pontos correspondentes.

A doação é então verificada e analisada por um administrador ou funcionário, que realiza o *upload* de uma imagem para atualizar o estado da doação. Após a conclusão bem-sucedida do processo de doação e a atualização do estado, a entidade beneficiária recebe a doação no ponto de coleta designado (*Pickup Point*).

Estrutura

A estrutura do sistema *web* delineada é organizada em várias rotas baseadas em funcionalidades específicas. Cada rota principal possui sub-rotas que atendem a diferentes ações e necessidades dentro do sistema. A seguir, detalhamos a estrutura das rotas principais e suas respectivas sub-rotas:

1. Rota Principal "/"

- / - Página inicial com redirecionamento automático para /home.
- /home - Página principal.

2. Autenticação "/auth"

- **/login** - Página de login, suportando métodos GET e POST.
- **/logout** - Página de logout com redirecionamento no final, método GET.
- **/register** - Página de registro de novos usuários, suportando métodos GET e POST.

3. Dashboard "/dashboard"

- **/** - Página principal do dashboard, método GET.

4. Utilizadores "/user"

- **/** - Redireciona para o dashboard.
- **/entities** - Gestão de entidades.
 - **/create** - Página de criação de novas entidades, suportando métodos GET e POST.
 - **/:id** - Exibição de uma única entidade, método GET.
 - **/edit** - Edição de uma entidade específica, suportando métodos GET e POST.
- **/employees** - Gestão de funcionários.
 - **/u/:id** - Atualização do status ou função de um funcionário, método POST.
 - **/:id** - Edição de um funcionário, suportando métodos GET e POST.
- **/donators** - Gestão de doadores.
 - **/create** - Página de criação de novos doadores, suportando métodos GET e POST.
 - **/u/:id** - Atualização do status ou função de um doador, método POST.
 - **/:id** - Exibição de um único doador, método GET.
 - **/edit** - Edição de um doador específico, suportando métodos GET e POST.

5. Pontos "/points"

- **/** - Exibição de todas as regras de pontos, método GET.
- **/rule/create** - Página de criação de novas regras de pontos, método GET.
- **/rule/save** - Guardar uma nova regra na base de dados, método POST.
- **/rule/:id** - Exibição de uma regra específica, método GET.
- **/rule/edit/:id** - Página de edição de uma regra específica, método GET.
- **/rule/update/:id** - Guardar as alterações de uma regra, método POST.
- **/rule/:id** - Exclusão de uma regra, método DELETE.

6. Coletas "/pickups"

- **/** - Exibição de todas as coletas, método GET.
- **/create** - Página de criação de novas coletas, método GET.
- **/save** - Guardar de uma nova coleta na base de dados, método POST.
- **/:id** - Exibição de uma coleta específica, método GET.
- **/edit/:id** - Página de edição de uma coleta específica, método GET.
- **/update/:id** - Guardar as alterações de uma coleta, método POST.
- **/:id** - Exclusão de uma coleta, método DELETE.
- **/:id** - Atualização do status de uma coleta, método PUT.

- **/:id/donations** - Exibição de todas as doações associadas a uma coleta específica, método GET.

7. Doações "/donations"

- **/** - Exibição de todas as doações, método GET.
- **/create** - Página de criação de novas doações, método GET.
- **/save** - Guardar de uma nova doação na base de dados, método POST.
- **/:id** - Exibição de uma doação específica, método GET.
- **/update/:id** - Atualização do status de uma doação, método POST.
- **/:id/upload-proof** - Upload de imagem de prova para uma doação, método POST.

Essa estrutura modular permite uma clara separação de responsabilidades e facilita a manutenção e a escalabilidade do sistema. Cada segmento funcional do site é bem definido com rotas específicas para criação, visualização, edição e exclusão de recursos, garantindo uma navegação intuitiva e eficiente para os utilizadores.

Demonstração



Copyright © Grupo 6
Powered by blood, sweat, tears and ChatGPT

Copyright © Grupo 6
Powered by blood, sweat, tears and ChatGPT

| |
|-----------|
| EMPLOYEES |
| ENTITIES |
| DONATORS |
| DONATIONS |
| POINTS |
| PICKUPS |

Weekly Report

| | | |
|----------------------|----------------------|----------------------|
| New Donations 3 | New Donors 7 | New Users 0 |
| View | View | View |

Overview

| | | | |
|----------------------|----------------------|----------------------|-------------------------|
| Donations 15 | Entities 3 | Pickup Points 1 | Point System Rules 3 |
| View | View | View | View |

Recent Donations

| ID | Donor | Status | Actions |
|--------------------------|------------------|----------------|----------------------|
| 666a1c1c6f1fb7a52159c062 | david@dd.dd | waiting_pickup | View |
| 6669f50f6f1fb7a52159bc3a | david@dd.dd | waiting_pickup | View |
| 6667a1f2b75509100e0c6798 | david@dd.dd | verified | View |
| 664762fd1d438b4bf193f4eb | mmm@google.pt | completed | View |
| 664761e31d438b4bf193f4a9 | minnie@google.pt | verified | View |

Average Data



| |
|-----------|
| EMPLOYEES |
| ENTITIES |
| DONATORS |
| DONATIONS |
| POINTS |
| PICKUPS |

Employee List

| Name | Email | Role | Edit | Remove | Make Admin |
|------------|-------------------------|----------|----------------------|------------------------|----------------------------|
| aa | aa@aa.aa | Admin | | | |
| bb | bb@bb.bb | Employee | Edit | Remove | Make Admin |
| cc | cc@cc.cc | Employee | Edit | Remove | Make Admin |
| dd | dd@dd.dd | Admin | | | |
| ee | ee@ee.ee | Employee | Edit | Remove | Make Admin |
| Nuno Gomes | nunogomes64@hotmail.com | Employee | Edit | Remove | Make Admin |
| b1 | b1@gmail.com | Employee | Edit | Remove | Make Admin |
| bb | bb@gmail.com | Employee | Edit | Remove | Make Admin |
| ola | ola@ola.pt | Employee | Edit | Remove | Make Admin |

| |
|-----------|
| EMPLOYEES |
| ENTITIES |
| DONATORS |
| DONATIONS |
| POINTS |
| PICKUPS |

Donator List

| |
|---------------|
| João Silva |
| David Santos |
| Donor Teste |
| donor teste 2 |

[Adicionar](#)

| |
|-----------|
| EMPLOYEES |
| ENTITIES |
| DONATORS |
| DONATIONS |
| POINTS |
| PICKUPS |

Status List

Donations List

Create Donation

| Id | Donor | Status | Actions |
|--------------------------|--------------------|----------------|----------------------|
| 666a1c1c6f1fb7a52159c062 | david@dd.dd | waiting_pickup | View |
| 6669f58f6f1fb7a52159bc3a | david@dd.dd | waiting_pickup | View |
| 6667a1f2b75509100e0c6798 | david@dd.dd | verified | View |
| 664762fd1d438b4bf193f4eb | mmm@google.pt | completed | View |
| 664761e31d438b4bf193f4a9 | minnie@google.pt | verified | View |
| 66475610af4f8a617addbca4 | alexvr25@gmail.com | verified | View |
| 663573011506977e120195b2 | david@donor.pt | send_to_entity | View |
| 66356aac64e14d462ec8270e | alexvr25@gmail.com | verified | View |
| 6634fb010cde1be7d8ae8d27 | joao@example.com | verified | View |
| 6634f1611562504d60e27d49 | joao@example.com | verified | View |
| 6634efe5e52b7f7713070a7 | joao@example.com | verified | View |
| 6634e4b63a4612ac783123ab | joao@example.com | completed | View |
| 6634e305d4f3dc620d66c4fd | joao@example.com | verified | View |
| 6634de2a1cd7a7862a641e73 | joao@example.com | verified | View |
| 66342e3c81351753d64ab77b | joao@example.com | verified | View |

| |
|-----------|
| EMPLOYEES |
| ENTITIES |
| DONATORS |
| DONATIONS |
| POINTS |
| PICKUPS |

Point System Rules List

Create Rule

| Rule For Points | Rule Type | Description | Actions |
|------------------------|------------------|--|----------------------|
| kg | 10 | pointsAssignment For each kg in a donation, the donor receives 10 points | View |
| item | 4 | pointsAssignment for each item, the donor receives 4 points | View |
| convert | 200 | pointsConversion for each 200 points equals to 1€ | View |

API

Controladores e Rotas

Na construção da API REST, foram desenvolvidos controladores dedicados para cada serviço. Para manter a organização e facilitar a compreensão, as rotas de cada controlador iniciam com o nome do serviço em sua forma pluralizada.

Donation (Doação):

- `/` - GET: Recupera todas as doações
- `/` - POST: Adiciona uma nova doação
- `/{donationID}` - GET: Recupera uma doação específica

Donator (Doador):

- `/` - GET: Recupera todos os doadores
- `/` - POST: Adiciona um novo doador
- `/{donatorID}` - GET: Recupera um doador específico
- `/{donatorID}` - PUT: Atualiza um doador
- `/{donatorID}` - DELETE: Remove um doador
- `/{donatorID}/donations` - GET: Recupera doações feitas por um doador
- `/{donatorID}/convertPoints/{qtd}` - POST: Converte pontos em créditos

Entity (Entidade):

- `/` - GET: Recupera todas as entidades
- `/` - POST: Adiciona uma nova entidade
- `/{entityID}` - GET: Recupera uma entidade específica
- `/{entityID}` - PUT: Atualiza uma entidade
- `/{entityID}` - DELETE: Remove uma entidade
- `/{entityID}/donations` - GET: Recupera doações associadas a uma entidade

Pickups (Entregas):

- `/` - GET: Recupera todas as entregas
- `/{id}` - GET: Recupera uma entrega específica

Rules (Regras):

- `/` - GET: Recupera todas as regras
- `/{id}` - GET: Recupera uma regra específica

Auth (Autenticação):

- `/login/donator` - POST: Login para doadores

- `/register/donator` - POST: Registro para doadores
- `/logout/donator` - POST: Logout para doadores
- `/login/entity` - POST: Login para entidades
- `/register/entity` - POST: Registro para entidades
- `/logout/entity` - POST: Logout para entidades

É importante destacar que, embora todos esses métodos tenham sido projetados, apenas aqueles considerados úteis pelo nosso grupo foram efetivamente implementados.

Swagger

Donation API 1.0.0 OAS 2.0
[Base URL: /api]

API documentation for Donation, Donator, Entity, Pickup, and Rules services

Donation

| | | |
|-------------------|---|---|
| <code>GET</code> | <code>/donations</code> Get all donations | ▼ |
| <code>POST</code> | <code>/donations</code> Create a new donation | ▼ |
| <code>GET</code> | <code>/donations/{donationId}</code> Get donation by ID | ▼ |

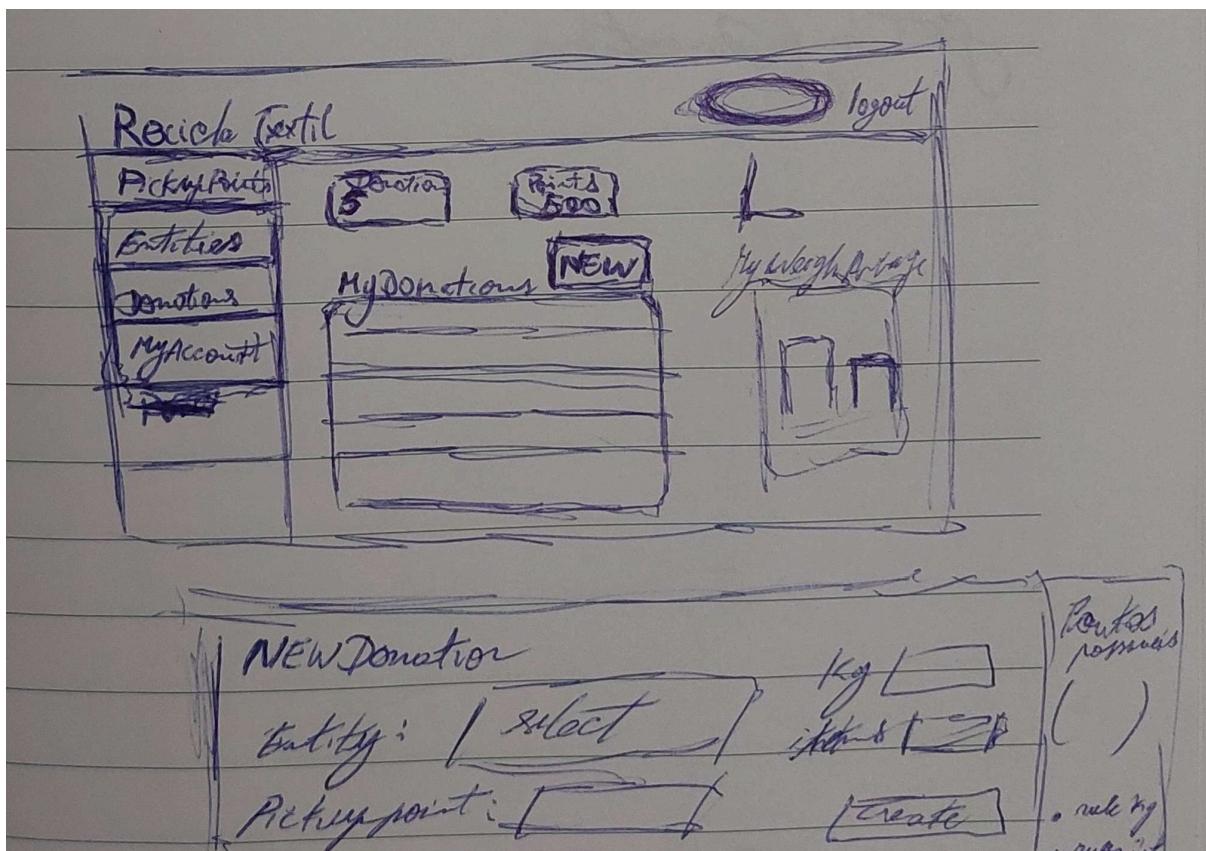
Donator

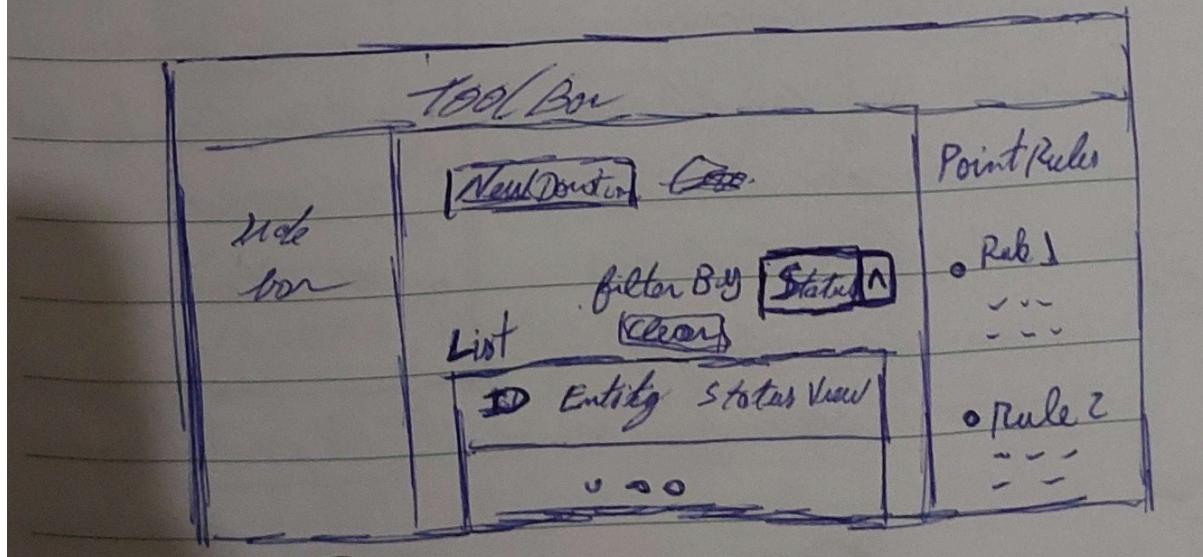
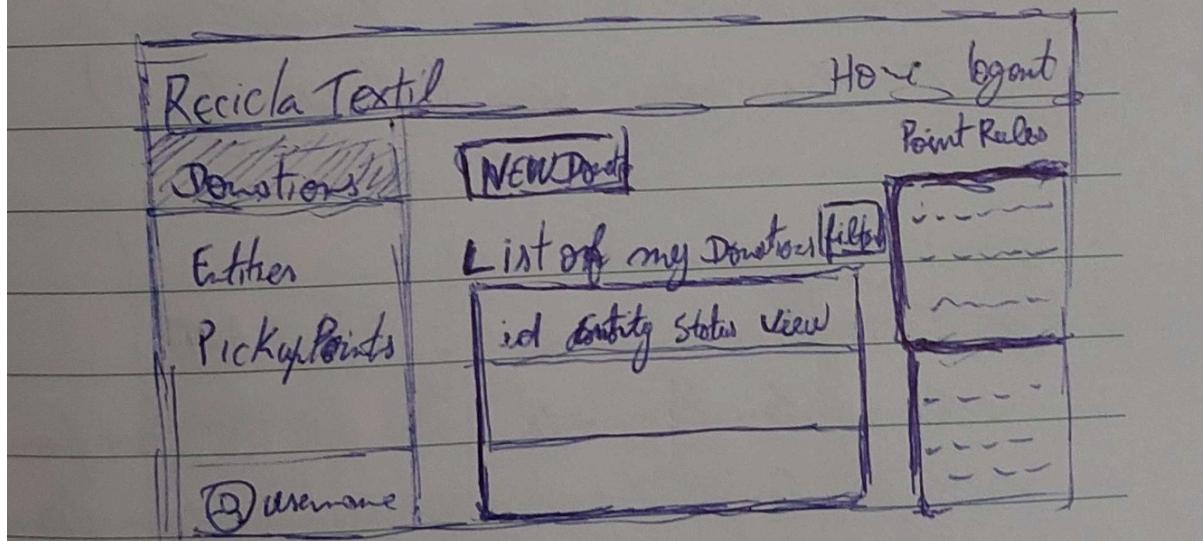
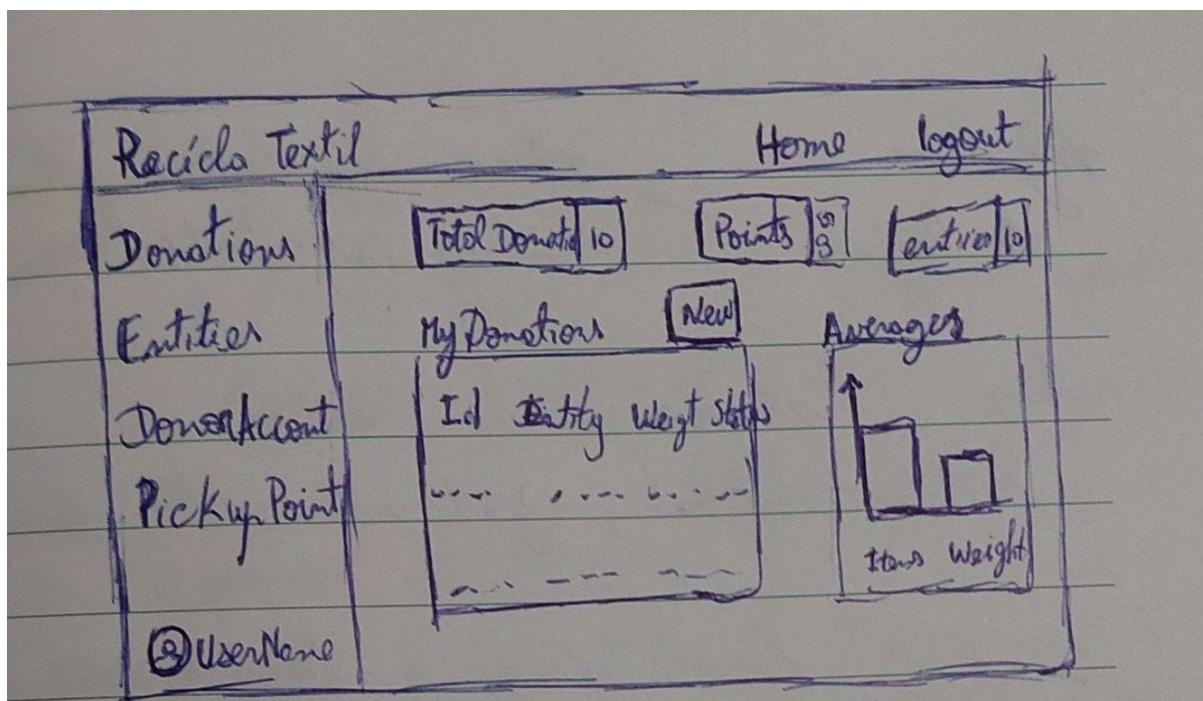
| | | |
|---------------------|---|---|
| <code>GET</code> | <code>/donators</code> Get all donators | ▼ |
| <code>POST</code> | <code>/donators</code> Create a new donator | ▼ |
| <code>GET</code> | <code>/donators/{donatorId}</code> Get donator by ID | ▼ |
| <code>PUT</code> | <code>/donators/{donatorId}</code> Update an existing donator | ▼ |
| <code>DELETE</code> | <code>/donators/{donatorId}</code> Delete an existing donator | ▼ |
| <code>GET</code> | <code>/donators/{donatorId}/donations</code> Get donations made by a specific donator | ▼ |
| <code>GET</code> | <code>/donators/{donatorId}/convertPoints/{qtd}</code> Convert donator's points to credit | ▼ |

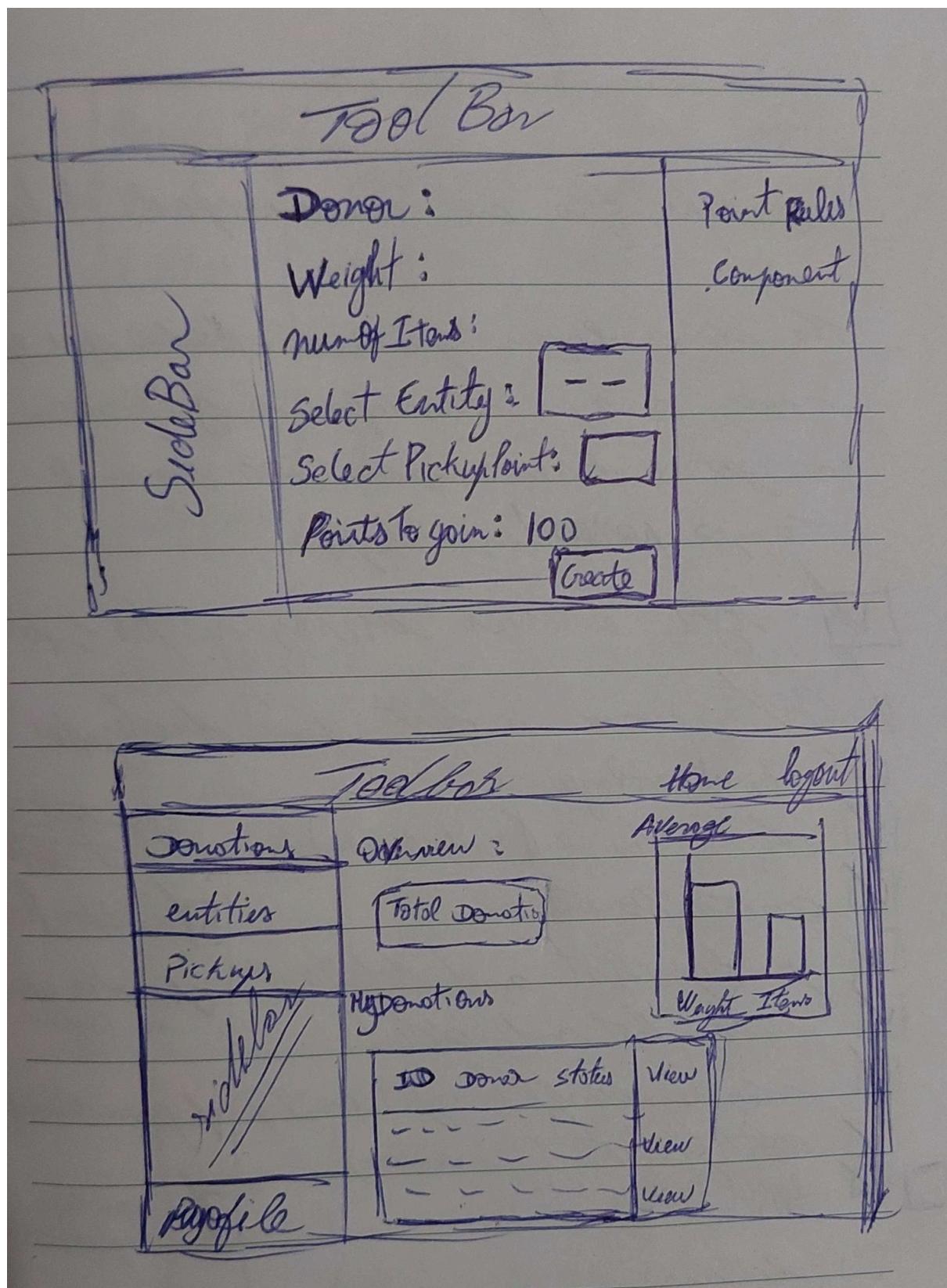
Angular

Mockups

Para a aplicação em Angular, foram desenvolvidos vários *mockups* em papel. Estes modelos preliminares servem como esboços detalhados, permitindo visualizar e ajustar a interface do utilizador e a experiência geral antes da implementação digital definitiva. A criação destes *mockups* em papel é uma etapa crucial no processo de *design*, pois facilita a identificação de melhorias e ajustes necessários, assegurando que o produto final atenda às expectativas e necessidades dos utilizadores.







Serviços

Durante o desenvolvimento da aplicação *web*, deparamo-nos com diversas situações e desafios que exigem a adição de certos serviços essenciais para complementar e melhorar a funcionalidade da aplicação. Esses serviços são cruciais para garantir uma experiência de utilizador eficiente, segura e fluida. Abaixo, descrevemos detalhadamente os principais serviços que foram integrados ao longo do desenvolvimento, explicando suas funções e a importância de cada um.

Serviços Adicionados

1. Serviço REST

O serviço REST (Representational State Transfer) é uma parte fundamental no desenvolvimento de aplicações web modernas. Este serviço permite a comunicação eficiente entre o *frontend* da aplicação Angular e o *backend*, geralmente através de APIs. Implementar um serviço REST oferece várias vantagens, tais como:

- **Facilidade de Comunicação:** Através de requisições HTTP (GET, POST, PUT, DELETE), podemos facilmente trocar dados entre a aplicação cliente e o servidor.
- **Desempenho:** Serviços REST são geralmente leves e eficientes, possibilitando uma rápida recuperação e envio de dados.
- **Escalabilidade:** Devido ao seu design baseado em recursos, é fácil adicionar novos endpoints conforme a aplicação cresce.
- **Manutenção e Extensibilidade:** APIs REST são bem documentadas e padronizadas, facilitando a manutenção e adição de novas funcionalidades.

Na nossa aplicação, a implementação do serviço REST foi fundamental para realizar operações de CRUD (Create, Read, Update, Delete) de maneira eficiente, permitindo uma interação dinâmica e responsiva com a base de dados.

2. Serviço de Autenticação

A autenticação é um aspecto crucial em qualquer aplicação que requer controlo de acesso, garantindo que apenas utilizadores autorizados podem aceder a determinadas funcionalidades e dados. No desenvolvimento da nossa aplicação Angular, implementamos um robusto serviço de autenticação com os seguintes componentes:

- **Login e Logout:** Implementação de formulários de login e logout para gerenciar sessões de usuário.
- **Tokens JWT:** Utilização de JSON Web Tokens (JWT) para assegurar que as comunicações entre cliente e servidor sejam seguras e que as identidades dos usuários sejam verificadas.

A implementação deste serviço não apenas protegeu os dados sensíveis, mas também melhorou a confiança dos utilizadores na aplicação, oferecendo uma experiência de uso segura e eficiente.

3. Proteção de Rotas

A proteção de rotas é um mecanismo essencial para garantir que apenas utilizadores autorizados podem aceder a certas partes da aplicação. No Angular, utilizamos Guards (Guardiões) para implementar essa funcionalidade. Os principais aspectos da proteção de rotas incluem:

- **Route Guards:** Implementação do tipo de guardas CanActivate para verificar permissões antes de carregar rotas específicas.
- **Redirecionamento:** Redirecionamento de usuários não autenticados para a página de login ou de acesso negado, conforme apropriado.
- **Controle de Acesso por Função:** Implementação de lógica baseada em papéis de utilizador, permitindo ou negando o acesso a rotas com base no papel (doador ou entidade).

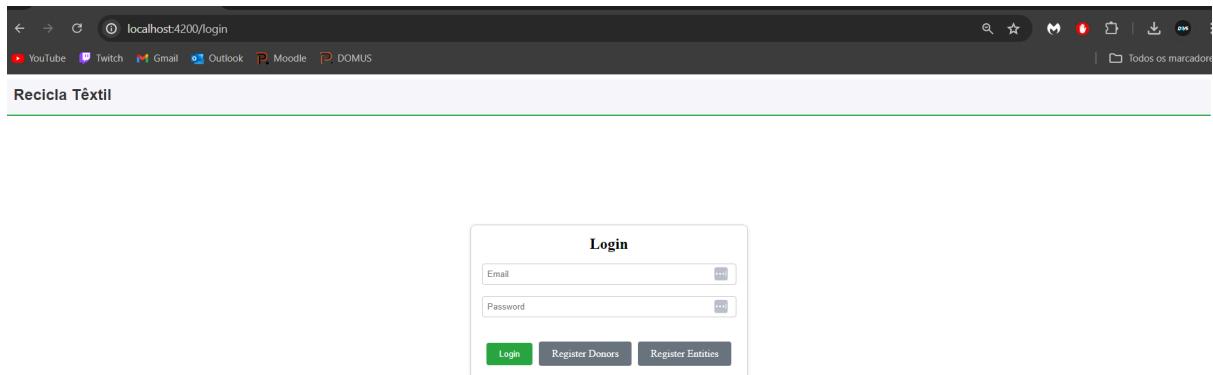
Essa camada adicional de segurança assegura que os recursos da aplicação sejam acedidos de maneira controlada, protegendo dados sensíveis e funcionalidades críticas contra acessos não autorizados.

Dados

Aqui estão as credenciais para o teste da aplicação:

1. Doador: email - teste@donor.pt password - teste
2. Entity: email - felgueiras@centrosaude.com password - saude

Demonstração



localhost:4200/home

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcadores

Recicla Têxtil

HOME **LOG-OUT**

Donations

Entities

Pickup Points

My Total Donations

0

View

My Points

0

View

Entities

3

View

No Donations Yet

You haven't made any donations yet. Start making a difference today!

New Donation

My Profile

localhost:4200/home

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcadores

Recicla Têxtil

HOME **LOG-OUT**

Donations

Entities

Pickup Points

My Total Donations

3

View

My Points

0

View

Entities

3

View

Recent Donations

| Id | Entity | Status | Actions |
|--------------------------|----------------------------|----------------|----------------------|
| 666ad6915298026cbb6f9ce7 | Cinema | waiting_pickup | View |
| 666ad6855298026cbb6f9cd5 | Hospital São Lucas | waiting_pickup | View |
| 666ad606406eae8130d35ec | Centro de Saúde Felgueiras | waiting_pickup | View |

New Donation

Average Data

Average Data

| Category | Value |
|----------------|-------|
| Average Weight | ~3 |
| Average Items | ~7 |

Highcharts.com

My Profile

localhost:4200/donations/new

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcadores

Recicla Têxtil

[HOME](#) [LOG-OUT](#)

[Donations](#)

[Entities](#)

[Pickup Points](#)

[My Profile](#)

Create New Donation

Weight:

Number of Items:

Select Entity:

Select Pickup Point:

Points to Gain: 0

[Create](#)

Point Rules

kg: For each kg in a donation, the donor receives 10 points (10 points)

item: for each item, the donor receives 4 points (4 points)

convert: for each 200 points equals to 1€ (200 points)

localhost:4200/entities

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcadores

Recicla Têxtil

[HOME](#) [LOG-OUT](#)

[Donations](#)

[Entities](#)

[Pickup Points](#)

[My Profile](#)

Entities Receiving Donations

Help make a difference by supporting verified entities that rely on donations.

[Donate Now](#)

ID: 6633f53f381f621071e1b7b4

Name: Hospital São Lucas

Email: contato@saolucas.com

[View](#)

ID: 6634b8f0178b312feafa59c8

Name: Cinema

Email: alexvr25@gmail.com

[View](#)

ID: 6668ed3afbe69689d098aa85

Name: Centro de Saúde Felgueiras

Email: felgueiras@centrosaude.com

[View](#)

Recicla Têxtil

HOME LOG-OUT

Donations

Entities

Pickup Points

My Profile

Pickup Points

Explore and support local pickup points that facilitate donation pickups across various cities.

Donate Now

Filter by City: All Cities

ID: 6633d8cab0d498485e0e6e4f
Name: Drop-off Felgueiras
Address: Rua 1, Felgueiras - 4610-105
View

localhost:4200/profile

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcados

Recicla Têxtil

HOME LOG-OUT

Donations

Entities

Pickup Points

My Profile

Welcome, Donor Teste!

Personal Information
Name: Donor Teste
Email: teste@donor.pt

Account Details
Points: 0
Credit: 0

Back

Points to Convert
0

Conversion Rule
for each 200 points equals to 1€
Convert Points

localhost:4200/home

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcadores

Recicla Têxtil

HOME **LOG-OUT**

Donations

Entities

Pickup Points

My Profile

My Total Donations

2

View

Recent Donations

| ID | Entity | Status | Actions |
|--------------------------|----------------------------|----------------|----------------------|
| 666ad606406eae8130d35eec | Centro de Saúde Felgueiras | waiting_pickup | View |
| 666a1c1c6fb7a52159c062 | Centro de Saúde Felgueiras | waiting_pickup | View |

Average Data

Average Data

Values

Average Weight

Average Items

Highcharts.com

localhost:4200/donations

YouTube Twitch Gmail Outlook Moodle DOMUS Todos os marcadores

Recicla Têxtil

HOME **LOG-OUT**

Donations

Entities

Pickup Points

My Profile

My Donations

Make a difference with your donation today!

Donate Now

Filter by Status: waiting_pickup

| ID | Donor | Status | Actions |
|--------------------------|----------------|----------------|----------------------|
| 666ad606406eae8130d35eec | teste@donor.pt | waiting_pickup | View |
| 666a1c1c6fb7a52159c062 | david@dd.dd | waiting_pickup | View |

Point Rules

The screenshot shows a web browser window with the URL `localhost:4200/profile`. The page title is "Entity Profile: Centro de Saúde Felgueiras". The left sidebar has links for "Donations", "Entities", "Pickup Points", and "My Profile". The main content area displays "Entity Information" with fields for Name (Centro de Saúde Felgueiras), Email (felgueiras@centrosaude.com), and Description (Centro de Saúde Felgueiras contacta: felgueiras@centrosaude.com). The top right of the page includes standard browser controls like search, refresh, and download, along with a "HOME" and "LOG-OUT" button.

Conclusão

O desenvolvimento deste projeto em Programação em Ambiente Web proporcionou uma oportunidade valiosa para a aplicação prática dos conceitos teóricos estudados ao longo do curso. Os alunos puderam experimentar a criação de uma aplicação web completa, enfrentando e superando desafios técnicos e de design. Através da integração de tecnologias e práticas de desenvolvimento moderno, o projeto não só reforçou o conhecimento adquirido, mas também preparou os alunos para futuras empreitadas no campo do desenvolvimento web.