

# ISPGAYA

instituto superior politécnico

Licenciatura em Engenharia

Informática

Aplicações e Tecnologias Web

2023/2024

Daniel Nunes, nº2021101582

Diogo Neves, nº2021103926

Luís Gonçalves, nº2021103937

José Rodrigues, nº2021102951

## Jémis



**Relatório de estágio/projeto orientado pelo/a Professor/a Doutor/a José Monteiro e apresentada ao Instituto Superior Politécnico de Gaia**

Junho de 2023

## Resumo

Este projeto tem como objetivo criar uma aplicação web de votação em que, a partir de um questionário que avalia cada resposta do usuário, seja indicado o melhor candidato levando em consideração as suas preferências. O utilizador poderá criar uma conta inserindo o email e palavra-passe que deseja usar, sendo que após o registo ser bem-sucedido, terá acesso à sua área de trabalho.

Na área de trabalho, o utilizador poderá criar candidatos para a eleição, seguidos pelas ideias que defende, às quais poderá atribuir um peso definido, que será totalizado em 100%.

O utilizador poderá também inserir as suas próprias ideias, que serão comparadas com as ideias dos candidatos criados, sugerindo assim a votação no candidato que está mais comprometido com as ideias que o utilizador defende.

Por fim, o utilizador poderá alterar dados, apagar informações e até mesmo excluir a sua conta.

Na realização deste trabalho foram usadas as seguintes tecnologias:

- HTML;
- CSS;
- JavaScript;
- Framework Next.js;
- Node.js;
- TypeScript;
- MongoDB.

## **Abstract**

The goal of this project is to create a voting web application in which, from a questionnaire that evaluates each user's answers, the best candidate is indicated, taking into account the user's preferences. The user will be able to create an account by entering his or her email and password, and after the registration is successful, the user will have access to the work area.

In the work area, the user will be able to create candidates for the election, followed by the ideas he defends, to which he will be able to assign a defined weight, which will total 100%.

The user will also be able to insert his own ideas, which will be compared to the ideas of the created candidates, thus suggesting voting for the candidate who is most committed to the ideas that the user defends.

Finally, the user will be able to change data, delete information and even delete his account.

The following technologies were used in the realization of this work:

- HTML;
- CSS;
- JavaScript;
- Framework Next.js;
- Node.js;
- TypeScript;
- MongoDB.

## Índice

---

### Conteúdo

Resumo .....	2
Abstract .....	3
Introdução.....	5
Páginas HTML .....	6
• Página Login .....	6
• Página Registo .....	7
• Página Área de trabalho.....	9
Base de dados.....	11
• Escolha da Base de dados .....	11
• Ligação da base de dados.....	12
Funcionamento do sistema de escolha .....	14
Conclusões .....	15
Referências bibliográficas .....	16

### Índice de figuras

---

Figura 1 - Página de Login (tema escuro) .....	6
Figura 2 - Página de Login (tema escuro) .....	7
Figura 3 - Página de Registo (tema escuro) .....	7
Figura 4 - Código para a validação da palavra-passe e confirmação na página de registo .....	8
Figura 5 - Página de Registo (tema claro).....	9
Figura 6 - Área de trabalho (sem dados).....	9
Figura 7 - Área de trabalho (com dados) .....	10
Figura 8 - Formulário de criação de indagações .....	10
Figura 9 - Variáveis de conexão do MongoDB .....	12
Figura 10 - Função de conexão à base de dados .....	12
Figura 11 - Funções de "ReadDelete" e de "Insert" da base de dados .....	13
Figura 12 - Exemplo da escolha do candidato .....	14

## **Introdução**

Este projeto tem como finalidade a criação de uma aplicação web de votação com a funcionalidade de a partir de um questionário que avalia cada resposta do utilizador e com essas respostas indica o melhor candidato tendo em conta as suas preferências. O utilizador poderá criar uma conta a partir do seu email e password que após o registo poderá aceder á sua área de trabalho (*dashboard*). O utilizador na sua área de trabalho poderá criar candidatos á eleição seguido pelas suas ideias que terão um peso inserido pelo utilizador (que será 100%).

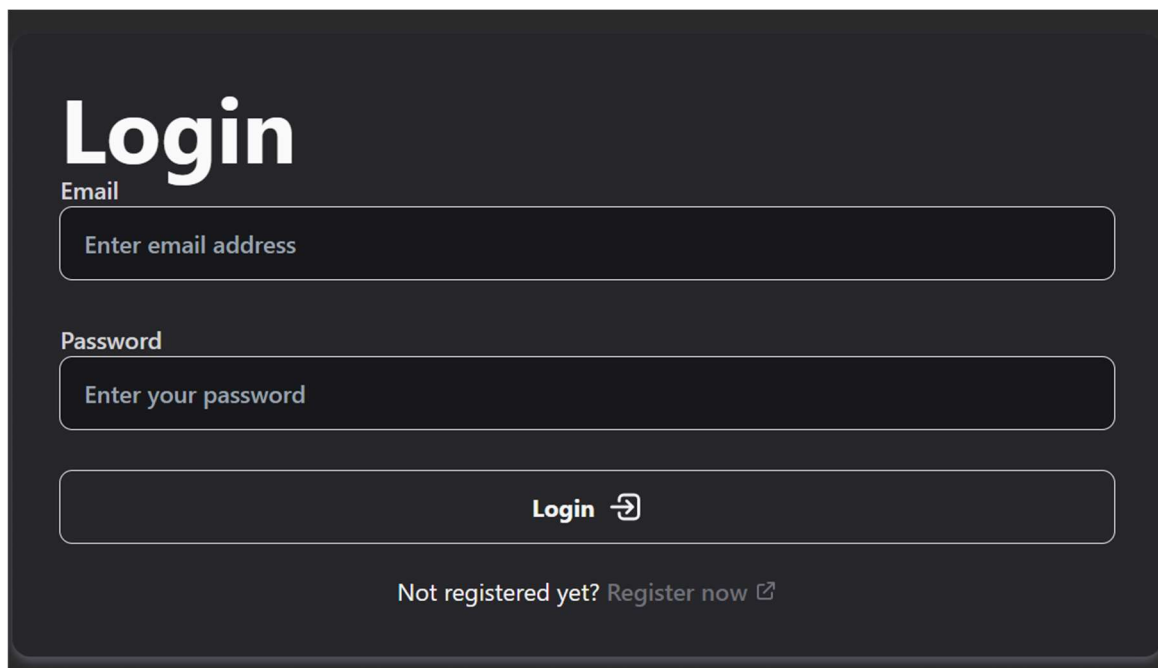
O utilizador também poderá inserir as suas ideias onde vão ser comparadas com as ideias dos candidatos criados e será sugerido o voto do candidato que melhor se adequa á sua ideia.

Por fim o utilizador poderá alterar dados, apagar dados e apagar conta.

## Páginas HTML

- **Página Login**

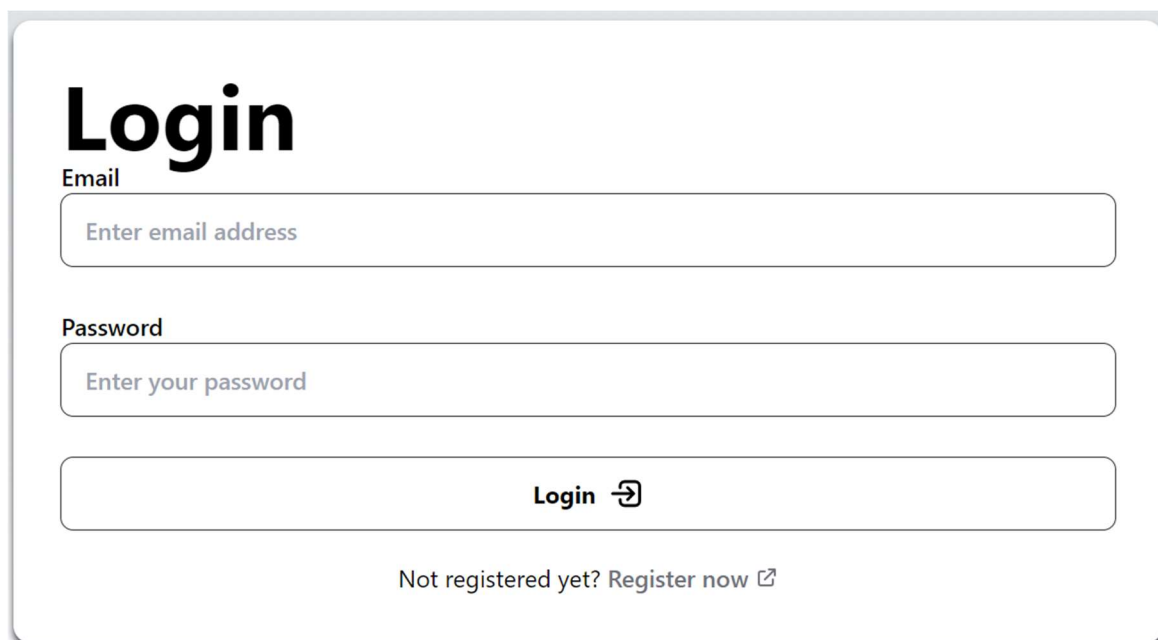
De maneira a um utilizador poder aceder á sua área de trabalho o mesmo terá de ser submetido a um formulário de login.

A login form with a dark theme. It features a large 'Login' title in white. Below the title are two input fields: 'Email' with the placeholder 'Enter email address' and 'Password' with the placeholder 'Enter your password'. A 'Login' button with a right-pointing arrow icon is positioned below the password field. At the bottom, there is a link that says 'Not registered yet? Register now' followed by an external link icon.

*Figura 1 - Página de Login (tema escuro)*

Ao preencher os campos de “Email” e “Password” e acionar o botão “Login”, irá ser iniciada uma pesquisa na base de dados por uma correspondência existente nos utilizadores registados. Caso o utilizador não possua uma conta, é possível criar uma facilmente através da indicação “Register now”, que redireciona o utilizador para a página de registo através de um *hyperlink*.

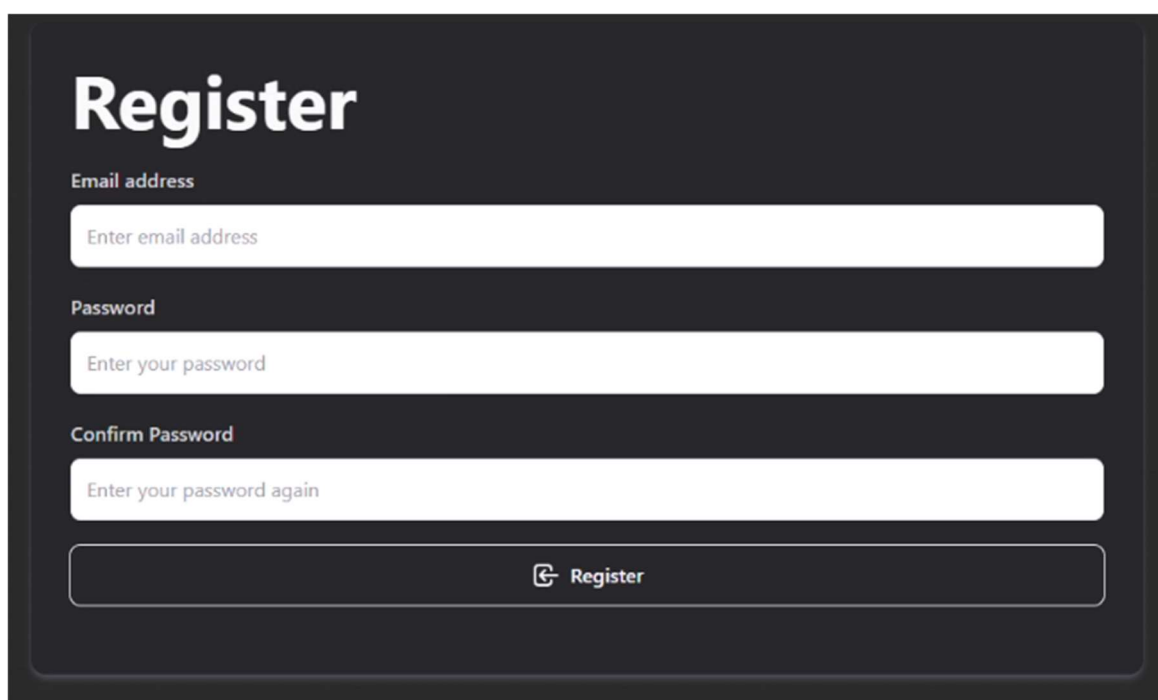
Além disso, o site oferece dois sistemas de paleta de cores, o que significa que dependendo do modo de cor definida no browser ou no sistema operativo, irá mudar para o tema escuro (imagem acima) ou o tema claro (imagem abaixo).



The image shows a login form with a dark background. At the top left, the word "Login" is written in a large, bold, white font. Below it, the label "Email" is in a smaller white font, followed by a white input field with the placeholder text "Enter email address". Below this, the label "Password" is in a smaller white font, followed by another white input field with the placeholder text "Enter your password". At the bottom, there is a white button with the text "Login" and a right-pointing arrow icon. Below the button, the text "Not registered yet? Register now" is displayed in a small white font, followed by a right-pointing arrow icon.

*Figura 2 - Página de Login (tema escuro)*

- **Página Registo**



The image shows a register form with a dark background. At the top left, the word "Register" is written in a large, bold, white font. Below it, the label "Email address" is in a smaller white font, followed by a white input field with the placeholder text "Enter email address". Below this, the label "Password" is in a smaller white font, followed by a white input field with the placeholder text "Enter your password". Below that, the label "Confirm Password" is in a smaller white font, followed by a white input field with the placeholder text "Enter your password again". At the bottom, there is a white button with the text "Register" and a right-pointing arrow icon.

*Figura 3 - Página de Registo (tema escuro)*

Com o objetivo de criar uma conta para um utilizador foi feita uma página de registo onde o mesmo pode inserir o seu email e palavra-passe, sendo que a última tem um campo de confirmação para assegurar que o utilizador inseriu realmente a palavra-passe desejada.

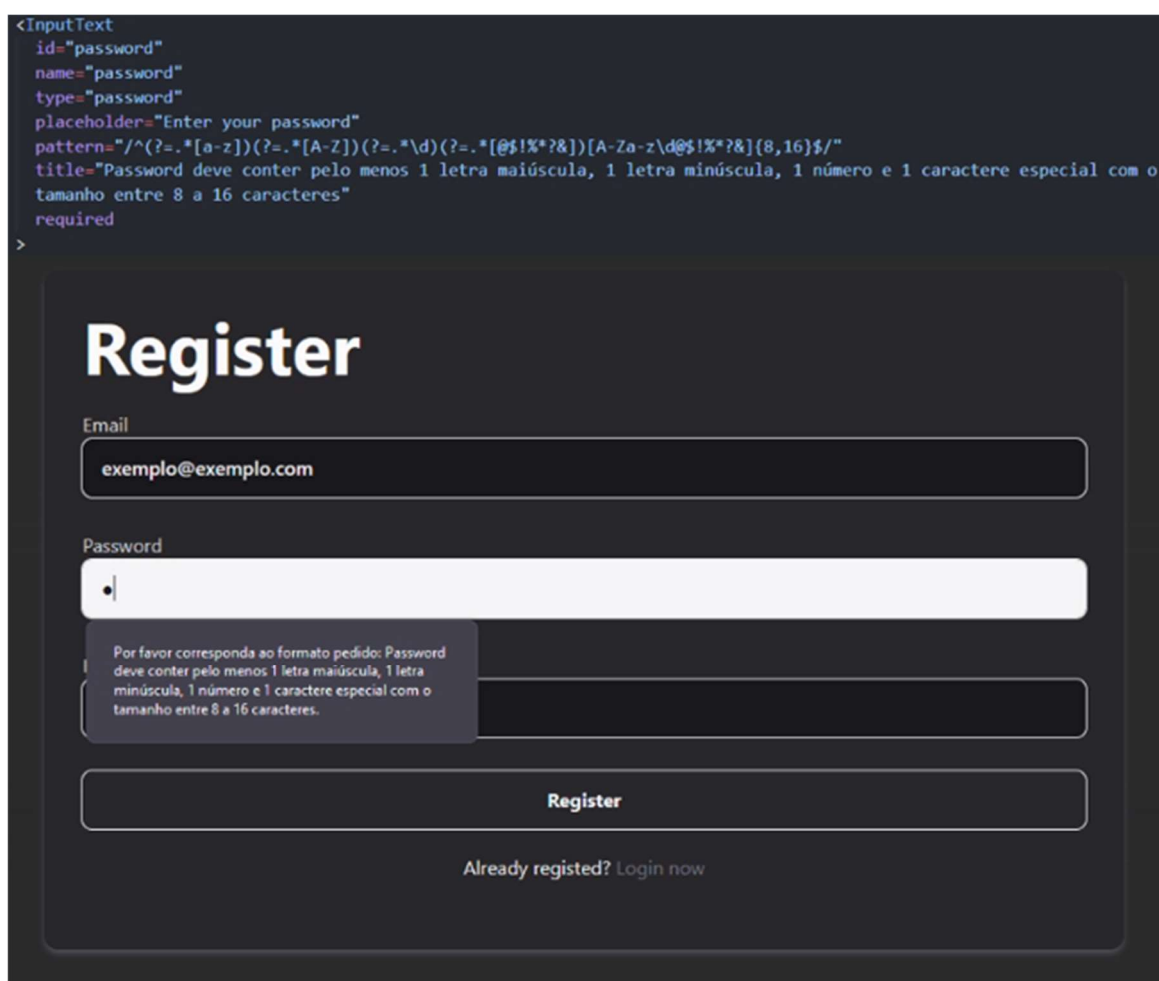
Relativamente aos critérios para a aceitação do email, será necessário um formato válido, confirmação a qual é obtida através de métodos incorporados do HTML.

Relativamente aos critérios para a aceitação da palavra-passe que o utilizador inseriu, existem vários critérios:

- O tamanho está limitado entre 8 e 16 caracteres;
- Necessita de conter pelo menos uma letra maiúscula e uma letra minúscula;
- Necessita de conter um caractere especial.

Ao clicar no botão “Register”, a informação será primeiramente avaliada relativamente ao preenchimento válido de todos os campos e posteriormente enviada para a base de dados caso esteja tudo correto.

```
<InputText
  id="password"
  name="password"
  type="password"
  placeholder="Enter your password"
  pattern="/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,16}$/"
  title="Password deve conter pelo menos 1 letra maiúscula, 1 letra minúscula, 1 número e 1 caractere especial com o tamanho entre 8 a 16 caracteres"
  required
>
```

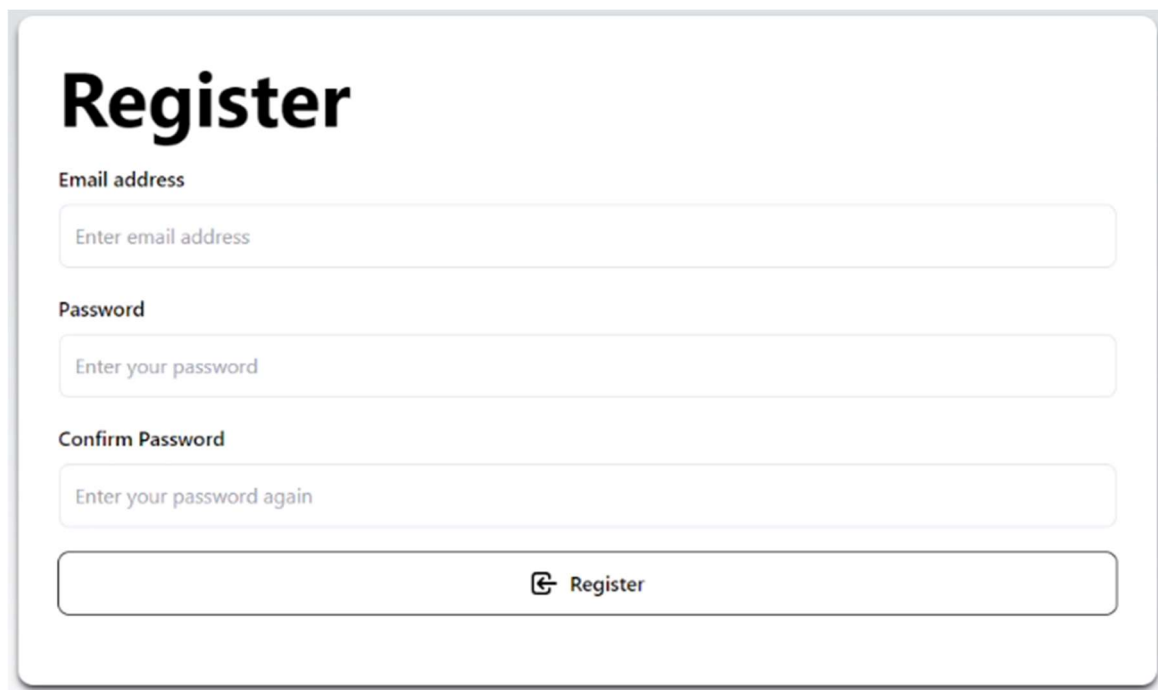


*Figura 4 - Código para a validação da palavra-passe e confirmação na página de registo*

Tal como na página de login, a página de registo poderá mudar a paleta de cores conforme a cor definida no browser ou sistema operativo.

Tal como na página de login, esta página poderá mudar a paleta de cores dependendo do modo de cor definida no browser ou no sistema operativo. O exemplo do tema escuro foi demonstrado acima, sendo que a imagem abaixo demonstra um exemplo do tema claro.






**Register**

Email address

Password

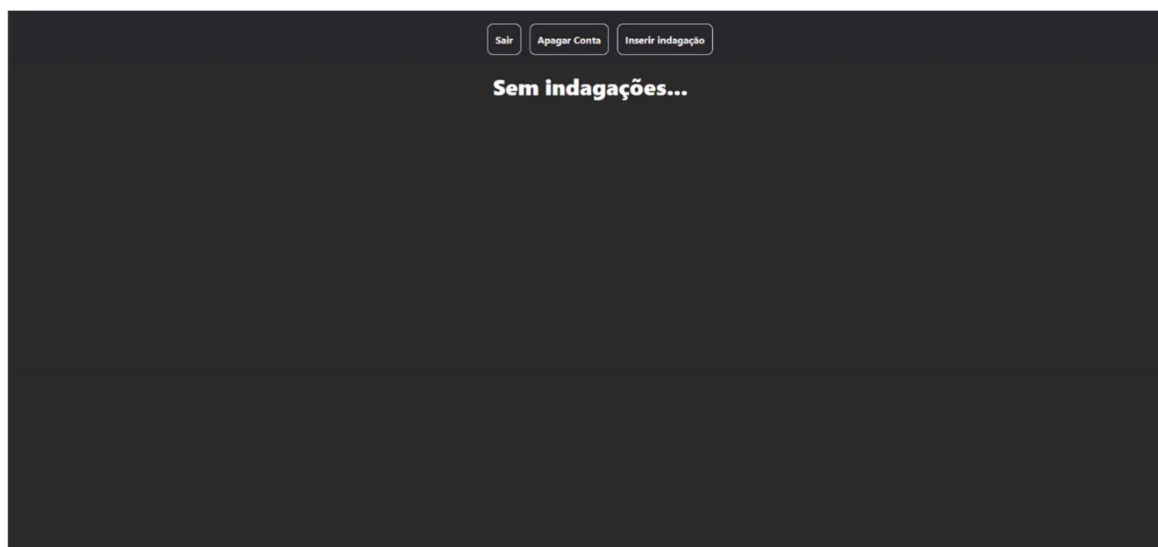
Confirm Password

 Register

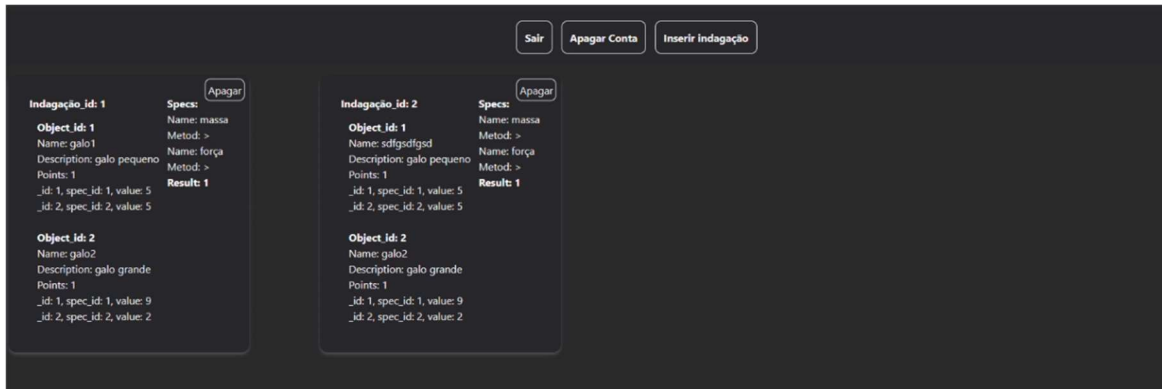
*Figura 5 - Página de Registo (tema claro)*

- **Página Área de trabalho**

Na página HTML da área de trabalho é possível observar várias tabelas com detalhes das indagações inseridas. Nas tabelas é ainda possível eliminar as indagações pretendidas selecionando o botão “Apagar”, eliminar todos os dados do utilizador, incluindo a sua conta, no botão “Apagar Conta”, e por fim sair da conta através do botão “Sair”.



*Figura 6 - Área de trabalho (sem dados)*



*Figura 7 - Área de trabalho (com dados)*

Ao selecionar o botão “Inserir indagação” será possível adicionar uma nova indagação preenchendo todos os campos pedidos e submetendo selecionando o botão “Inserir” voltando para a área de trabalho.

*Figura 8 - Formulário de criação de indagações*

## FALTA PRINT E EXPLICAÇÃO DA PARTE DE MOSTRAR O MELHOR CANDIDATO

Relativamente ao funcionamento da escolha do melhor candidato para o utilizador votar, a parte lógica está explicitada no tópico “Funcionamento do sistema de escolha”, detalhado mais à frente neste relatório.

## Base de dados

- **Escolha da Base de dados**

Como forma de armazenamento, organização e proteção dos dados foi optado o “MongoDB” que consiste num software de base de dados que utiliza “NoSQL”, ou seja, armazena dados de forma não relacional, sendo usado por muitas organizações devido à sua boa capacidade de armazenamento de grandes quantidades de dados e a velocidade de armazenamento dos mesmos.

A arquitetura da base de dados é constituída essencialmente por coleções e documentos. Os documentos funcionam associando um valor com uma chave, onde também é possível fazer as operações “CRUD”, ou seja, “Create, Read, Update and Delete”, que em português traduz-se para “Criar, Ler, Atualizar e Apagar” um valor.

Neste caso o MongoDB terá duas coleções, “Users” e “Indagações”, onde na primeira serão guardadas as credenciais dos utilizadores e na última serão guardados os dados que o utilizador submeter na página de trabalho. A partir destas duas coleções é possível, por exemplo, saber as indagações de um utilizador a partir de agregações (conforme o esquema abaixo).

Exemplo de agregação:

```
{  
  from: "Indagacoes",  
  localField: "_id",  
  foreignField: "user_id",  
  as: "Indagacoes",  
}
```

Este exemplo de agregação tem funcionamento semelhante ao “join” usado em “SQL”, ou seja, a partir de um “id” de um utilizador é possível encontrar as indagações que contêm o mesmo “id” de utilizador, retornando assim todas as indagações relacionadas ao mesmo.

- **Ligação da base de dados**

A ligação à base de dados é um processo fundamental para a recolha e armazenamento de dados. Esta consiste em estabelecer uma conexão entre a aplicação/sistema e a base de dados, permitindo a importação e exportação de dados de forma eficiente.

Para estabelecer a ligação, é necessário fornecer informações como:

- Nome do servidor onde a base de dados está hospedado;
- Nome da base de dados;
- Credenciais de autenticação (por exemplo, nome de utilizador e palavra-passe);

Antes de fazer as conexões precisamos de aceder à própria base de dados então utilizamos o seguinte código:

```
const uri = process.env.MONGODB_URI; // URI de conexão do MongoDB
const dbName = process.env.DATABASE_NAME; // Nome do seu banco de dados
```

*Figura 9 - Variáveis de conexão do MongoDB*

Estas duas variáveis armazenam a conexão com o “MongoDB” na variável “uri” e dentro do “MongoDB” o nome da base de dados, cuja variável é denominada “dbname”.

Após a inicialização das mesmas faz-se a ligação com a base de dados da seguinte forma:

```
async function Conn(collectionName: string) {
  try {
    /* declaração de variavel de conexão */
    client = new MongoClient(uri, {
      serverApi: {
        version: ServerApiVersion.v1,
        strict: true,
        useUnifiedTopology: true,
        deprecationErrors: true,
      },
    });

    /* conexão a base de dados */
    await client.connect();
    db = client.db(dbName);

    /* verificação se a coleção existe */
    const collection = await db.listCollections().toArray();
    if (
      collection.filter((c: { name: string }) => c.name === collectionName)
        .length === 1
    ) {
      let data = db.collection(collectionName);
      return data;
    } else throw new Error("Collection não encontrado");
  } catch (err: any) {
    console.log(err.name + ": " + err.message);
  }
}
```

*Figura 10 - Função de conexão à base de dados*

Esta função utiliza a variável uri estabelecida anteriormente e com essa conexão é associado o nome da base de dados dando assim a ligação á base de dados.

```
export const ReadDeleteUser = async () => {  
  
  const data = await Conn("Users");  
  let result = await data.find().toArray();  
  return result;  
};  
export const InsertUser = async () => {  
  
  return await Conn("Users");  
};  
export const ReadDeleteIndagacoes = async () => {  
  
  const data = await Conn("Indagacoes");  
  return await data.find().toArray();  
};  
  
export const InsertIndagacoes = async () => {  
  
  return await Conn("Indagacoes");  
};
```

*Figura 11 - Funções de "ReadDelete" e de "Insert" da base de dados*

As funções demonstradas na imagem acima contêm as 2 coleções que irão ser utilizadas: uma para os utilizadores ("Users") e uma para as indagações ("Indagacoes"). As funções "ReadDelete" e "Insert" serão diferentes para as coleções, já que a conexão necessária não é a mesma.

As funções "ReadDelete" são chamadas quando for necessária a leitura de dados da base de dados ou a eliminação de dados da mesma, conectando primeiro à coleção necessária e depois retornando todos os dados da base de dados necessários, convertidos em um "array".

Relativamente às funções "Insert", são chamadas quando é desejada a inserção de algum elemento na base de dados, retornando a conexão com a coleção correspondente.

## Funcionamento do sistema de escolha

```
[
  {
    "_id": 1,
    "User_id": 1,
    "specs": [
      {
        "_id": 1,
        "name": "massa",
        "metod": ">"
      },
      {
        "_id": 2,
        "name": "força",
        "metod": ">"
      }
    ],
    "objects": [
      {
        "_id": 1,
        "name": "galo1",
        "descricao": "galo pequeno",
        "valeus": [
          {
            "_id": 1,
            "spec_id": 1,
            "value": 5
          },
          {
            "_id": 2,
            "spec_id": 2,
            "value": 5
          }
        ]
      },
      {
        "_id": 2,
        "name": "galo2",
        "descricao": "galo grande",
        "valeus": [
          {
            "_id": 1,
            "spec_id": 1,
            "value": 9
          },
          {
            "_id": 2,
            "spec_id": 2,
            "value": 2
          }
        ]
      }
    ],
    "points": 1
  },
  {
    "_id": 2,
    "User_id": 1,
    "specs": [
      {
        "_id": 1,
        "name": "massa",
        "metod": ">"
      },
      {
        "_id": 2,
        "name": "força",
        "metod": ">"
      }
    ],
    "objects": [
      {
        "_id": 1,
        "name": "galo1",
        "descricao": "galo pequeno",
        "valeus": [
          {
            "_id": 1,
            "spec_id": 1,
            "value": 5
          },
          {
            "_id": 2,
            "spec_id": 2,
            "value": 5
          }
        ]
      },
      {
        "_id": 2,
        "name": "galo2",
        "descricao": "galo grande",
        "valeus": [
          {
            "_id": 1,
            "spec_id": 1,
            "value": 9
          },
          {
            "_id": 2,
            "spec_id": 2,
            "value": 2
          }
        ]
      }
    ],
    "points": 1
  }
],
  "result": -1
]
```

Figura 12 - Exemplo da escolha do candidato

De forma a navegar e escolher qual candidato é mais adequado tendo em conta as suas diferentes características, foi criado um sistema de pontos em que o candidato que obtiver mais será o recomendado ao utilizador.

Neste exemplo, o utilizador escolhe como características importantes para si a massa do galo e o tamanho do galo.

A indagação 1 apresenta mais peso na “spec1” (massa) do que a indagação 2, somando assim um ponto na sua classificação. Enquanto isso, a indagação 2 apresenta mais peso na “spec2” (força) do que a indagação 1, logo somará um ponto na sua classificação.

Como o utilizador só escolheu 2 características, as indagações tiveram o mesmo resultado, logo a recomendação terminará num empate (representado pelo valor “-1” na variável “result”). Este surgirá no output como “empate”, o que significa que ambos têm o mesmo peso de preferência.

## **Conclusões**

Em conclusão, este projeto permitiu a criação de uma aplicação web de votação que ajuda utilizadores a escolherem o melhor candidato com base nas suas preferências. O uso de tecnologias modernas e a integração com uma base de dados MongoDB possibilitam um funcionamento eficiente e escalável da aplicação. Possíveis melhorias futuras podem incluir a implementação de recursos adicionais, como a personalização da interface do utilizador e a análise de dados estatísticos das votações.

## **Referências bibliográficas**

*Docs | Next.js.* (sem data). Obtido 29 de junho de 2023, de <https://nextjs.org/docs>

*Document Database—NoSQL | MongoDB.* (sem data). Obtido 29 de junho de 2023, de <https://www.mongodb.com/document-databases>

*Documentation | Node.js.* (sem data). Obtido 29 de junho de 2023, de <https://nodejs.org/en/docs>