

# **Escola Superior de Tecnologia e Gestão**

## **Licenciatura em Engenharia Informática**

### **Programação por Internet 1**

**Ano Letivo 2023/22**

### **Trabalho Prático 2**

**Elaborado em: 2024/05/31**

**Nome e número do(s) Aluno(s):**

**Kinga Brynczka 2020155790**

## Índice

<b>List of Figures .....</b>	<b>ii</b>
<b>1. Introdução .....</b>	<b>1</b>
<b>2. Metodologia .....</b>	<b>2</b>
<b>3. Requisitos Implementados .....</b>	<b>3</b>
3.1. Requisitos Funcionais:.....	3
3.1.1. Visualização dos unidades curriculares existentes.....	3
3.1.2. Adição de unidades curriculares.....	4
3.1.3. Remoção de uma Unidade Curricular.....	5
3.1.4. Adição e remoção de avaliações associadas a uma unidade curricular.....	5
3.1.5. Adição e remoção de registos de tempo.....	6
3.2. Requisitos Não Funcionais: .....	9
3.2.1. Persistência dos dados utilizando localStorage.....	9
3.2.2. Visualização dos dados selecionados no topo da página. ....	9
3.2.3. Interface de usuário agradável e lógica, implementada com o uso de Bootstrap e ficheiro do estilo escrito em css. ....	10
<b>4. Código JavaScript .....</b>	<b>11</b>
4.1. DOM – Document Object Model .....	11
4.2. Funções JavaScript .....	11
4.2.1. Variáveis Globais.....	11
4.2.2. Inicialização do DOM .....	11
4.2.3. Exibição das Unidades Curriculares .....	12
4.2.4. Adição de Avaliação.....	13
4.2.5. Exibição das Avaliacoes .....	14
<b>5. Resumo dos requisitos implementados e não implementados .....</b>	<b>15</b>
<b>6. Conclusão .....</b>	<b>16</b>
<b>7. Referencias .....</b>	<b>17</b>

## **Lista das Figuras**

FIGURA 1 - PÁGINA PRINCIPAL SEM UNIDADES CURRICULARES REGISTRADOS.....	3
FIGURA 2 - PÁGINA PRINCIPAL COM REGISTOS DAS UNIDADES CURRICULARES .....	3
FIGURA 3 - PREENCHIMENTO DOS DADOS DA UNIDADE CURRICULAR.....	4
FIGURA 4 - AVISO PARA PREENCHER TODOS OS CAMPOS DISPONÍVEIS .....	4
FIGURA 5 - ADIÇÃO DA UNIDADE CURRICULAR COM SUCESSO .....	4
FIGURA 6 - CAIXA DE CONFIRMAÇÃO ANTES DE REMOÇÃO DE UMA UNIDADE CURRICULAR. ....	5
FIGURA 7 - ADIÇÃO DE UMA AVALIAÇÃO AO UNIDADE CURRICULAR.....	5
FIGURA 8 - INSERÇÃO DO NOME DA AVALIAÇÃO.....	5
FIGURA 9 - REMOÇÃO DE UMA AVALIAÇÃO .....	6
FIGURA 10 - REMOÇÃO DE UMA AVALIAÇÃO.....	6
FIGURA 11 - ADICIONAR NOVO REGISTO DE TEMPO GASTO PARA ESTUDAR .....	6
FIGURA 12 - TESTAR FUNCIONALIDADE DE ADIÇÃO DO REGISTO COM CAMPOS VAZIOS .....	7
FIGURA 13 - CAIXA DE AVISO PARA PREENCHER TODOS OS CAMPOS PARA CONSEGUIR ADICIONAR REGISTO COM SUCESSO .....	7
FIGURA 14 - ADICIONAR REGISTO COM SUCESSO.....	7
FIGURA 15 - CAIXA DE INFORMAÇÃO QUE O REGISTO FOI ADICIONADO COM SUCESSO.....	8
FIGURA 16 - REGISTO AFICIONADO COM SUCESSO.....	8
FIGURA 17 - CAIXA DE CONFIRMAÇÃO PARA REMOVER O REGISTO .....	8
FIGURA 18 - CONVERSÃO DE UM OBJETO EM UMA STRING PARA GUARDAR LO NO LOCALSTORAGE UTILIZANDO MÉTODO JSON.STRINGIFY "SETÍTEM" .....	9
FIGURA 19 - CONVERSÃO DO STRING NUM OBJETO PARA APRESENTAR UTILIZANDO MÉTODO JSON.STRINGIFY "GETÍTEM" .....	9
FIGURA 20 - VISUALIZAÇÃO DOS DADOS SELECIONADOS NO TOPO DA PAGINA.....	9
FIGURA 21 - VARIÁVEIS GLOBAIS .....	11
FIGURA 22 - INICIALIZAÇÃO DO DOM .....	11
FIGURA 23 - EXIBIÇÃO DAS UNIDADES CURRICULARES.....	12
FIGURA 24 - ADIÇÃO DA AVALIACAO .....	13
FIGURA 25 - EXIBIÇÃO DAS AVALIACOES .....	14

## **1. Introdução**

Este projeto tem como objetivo desenvolver um sistema utilizando ferramentas de web para gerenciar o tempo gasto em unidades curriculares universitárias. Como os estudantes podem ter dificuldades de gerenciar múltiplas disciplinas e avaliações, a ferramenta permitirá aos estudantes registrar e acompanhar o tempo dedicado a cada unidade e suas respectivas avaliações.

Utilizando tecnologias web como HTML, CSS, JavaScript e Bootstrap, o sistema oferece funcionalidades para adicionar, remover e visualizar unidades curriculares, avaliações e registos de tempo gasto a estudar para cada uma delas. Os dados são guardados no localStorage que garante persistência dos dados após do browser ser fechado ou atualizado.

O sistema visa proporcionar uma maneira eficiente e prática de organizar o tempo de estudo, ajudando os estudantes a melhorar sua gestão de tempo e, consequentemente, seu desempenho acadêmico. Isto foi feito numa página web que oferece uma maneira simples e intuitiva de utilizar todas as funcionalidades implementadas.

## **2. Metodologia**

Para o desenvolvimento deste sistema de gestão de tempo utilizei linguagens de programação HTML, CSS, e JavaScript. Também utilizei o Bootstrap para estilização mais rápida da página web. Em seguida, implementaram-se funcionalidades principais: adição, visualização e remoção de unidades curriculares, avaliações e registos de tempo, utilizando o `localStorage` para persistência de dados.

Durante o desenvolvimento do programa de gestão de estudo fiz vários testes para garantir o funcionamento correto de cada componente. A usabilidade foi avaliada e ajustada conforme feedback, assegurando uma interface intuitiva.

Toda a documentação necessária está explicada neste relatório detalhado descrevendo as decisões tomadas e uma tabela de requisitos implementados. Essa metodologia assegurou um desenvolvimento eficiente, resultando em uma aplicação robusta e funcional.

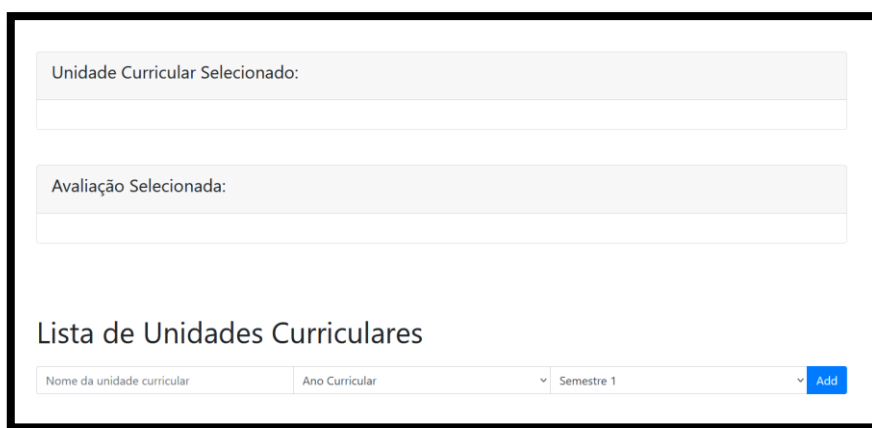
### 3. Requisitos Implementados

O sistema desenvolvido atende a diversos requisitos funcionais e não funcionais detalhadamente descritos nesta secção.

#### 3.1. Requisitos Funcionais:

##### 3.1.1. Visualização dos unidades curriculares existentes

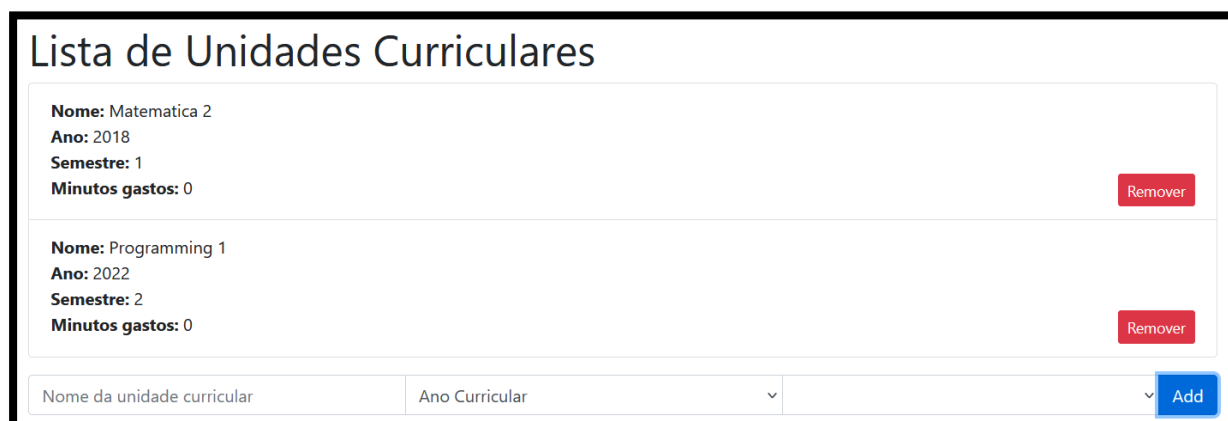
A pagina principal mostra os unidades curriculares que já foram adicionadas, caso ainda nenhuma unidade curricular encontra se no local Storage, o sistema mostra opção para adicionar unidade curricular. Também no topo da pagina são mostradas duas “caixas” que mostram informação selecionada sobre unidade curricular e também avaliação. Isto serve para quando há unidades curriculares e avaliações registadas no sistema e o utilizador esta carregando nos mesmos para verificar informação registada sobre o tempo gasto de estudo, estes aparecem no topo da pagina para facilitar utilização do sistema.



Nome da unidade curricular	Ano Curricular	Semestre 1	Add
----------------------------	----------------	------------	-----

Figura 1 - Página Principal sem unidades curriculares registados

Quando existem disciplinas guardadas na localStorage, eles aparecem em cima da opção da adição das novas unidades curriculares.



Nome: Matematica 2 Ano: 2018 Semestre: 1 Minutos gastos: 0	Remover
Nome: Programming 1 Ano: 2022 Semestre: 2 Minutos gastos: 0	Remover

Nome da unidade curricular	Ano Curricular	Semestre 1	Add
----------------------------	----------------	------------	-----

Figura 2 - Página Principal com registos das Unidades Curriculares

### 3.1.2. Adição de unidades curriculares.

Para adicionar uma unidade curricular, necessário preencher todos os campos. Em caso de utilizador não introduzir informação num ou mais campos, o sistema mostra aviso e pede para preencher todos os campos existentes.



A horizontal form with three input fields: 'Matematica 2', 'Ano Curricular' (with a dropdown arrow), and 'Semestre 2' (with a dropdown arrow). To the right of the last field is a blue button labeled 'Add'.

Figura 3 - Preenchimento dos dados da Unidade Curricular

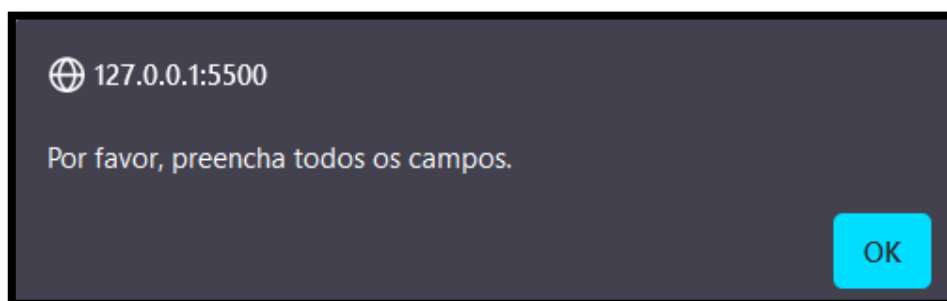


Figura 4 - Aviso para preencher todos os campos disponíveis

O sistema não permite adicionar unidade curricular enquanto a informação esta faltar. Apos de preencher todos os campos, a unidade curricular foi adicionada com sucesso. O tempo gasto para avaliação adicionada, esta automaticamente atribuído como 0 e vai ser atualizado só no momento de adicionar registos de uma avaliação que vai ser associada a esta unidade curricular.



A box titled 'Lista de Unidades Curriculares'. Inside, there is a list of details for a unit: 'Nome: Matematica 2', 'Ano: 2018', 'Semestre: 2', and 'Minutos gastos: 0'. A red 'Remove' button is located to the right of these details.

Figura 5 - Adição da Unidade Curricular com sucesso

### 3.1.3. Remoção de uma Unidade Curricular.

Ao clicando o botão “Remover” ao lado da uma Unidade Curricular especifica, sistema mostra aviso para confirmar se utilizador tem certeza que quer remover a mesma da memoria do programa, ou seja local Storage.

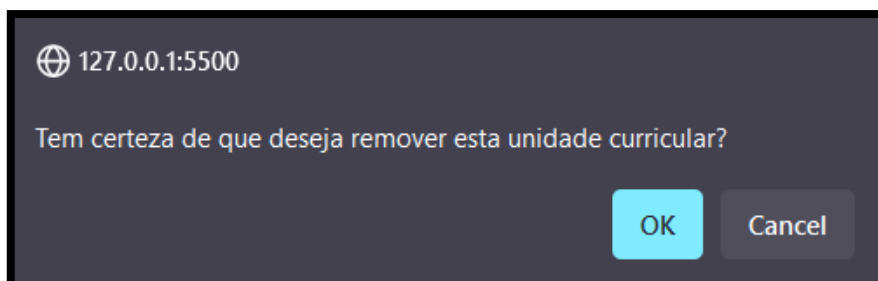


Figura 6 - Caixa de confirmação antes de remoção de uma unidade curricular.

### 3.1.4. Adição e remoção de avaliações associadas a uma unidade curricular.

Ao clicando no nome de uma unidade curricular, o nome torna-se verde com uma caixa ao volta dele que mostra que esta palavra faz parte funcional do sistema. Ao clicando na palavra nome o sistema permite adicionar avaliações associadas a esta unidade curricular.



Figura 7 - Adição de uma avaliação ao unidade curricular

Como ainda não havia nenhuma avaliação associada a esta unidade curricular, o sistema informa utilizador que ainda não existem avaliações associadas a esta unidade curricular e permite adicionar uma.

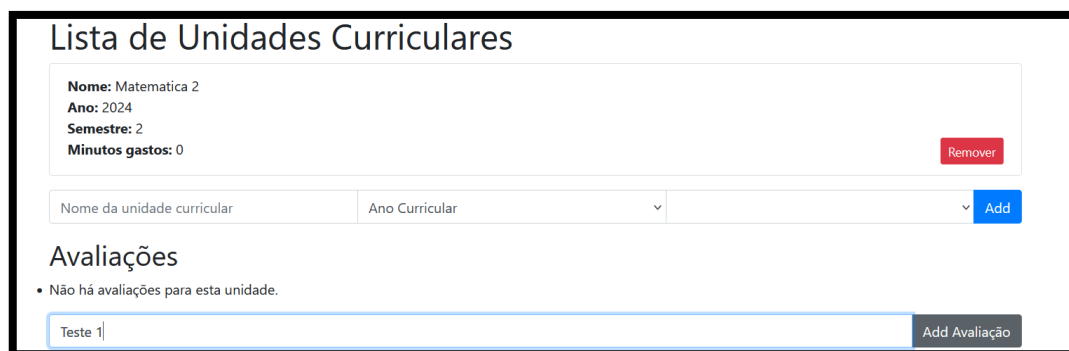


Figura 8 - Inserção do nome da avaliação



Ao clicar botão “Remover” ao lado da uma avaliação específica, o sistema permite remover avaliação se utilizador confirmar na caixa de aviso mostrada na figura 10.

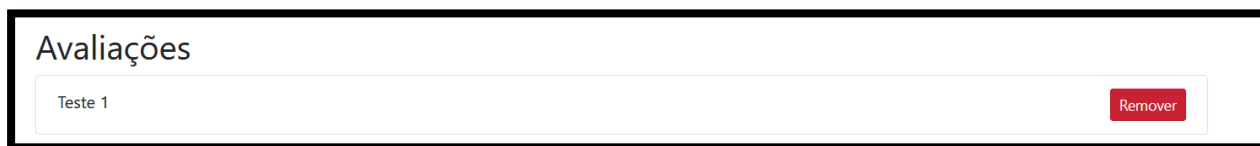


Figura 9 - Remoção de uma avaliação

Apos de confirmar na caixa de aviso que o utilizador realmente quer apagar informação da avaliação selecionada, o sistema apaga avaliação de local Storage.

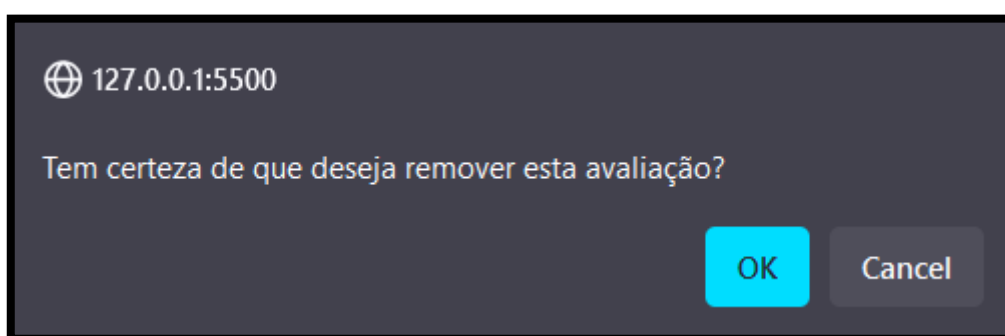


Figura 10 - Remoção de uma avaliação

### 3.1.5. Adição e remoção de registos de tempo.

Ao clicando no nome de uma avaliação, o sistema mostra parte de adicionar novo registo do tempo gasto ao estudar para esta avaliação específica.

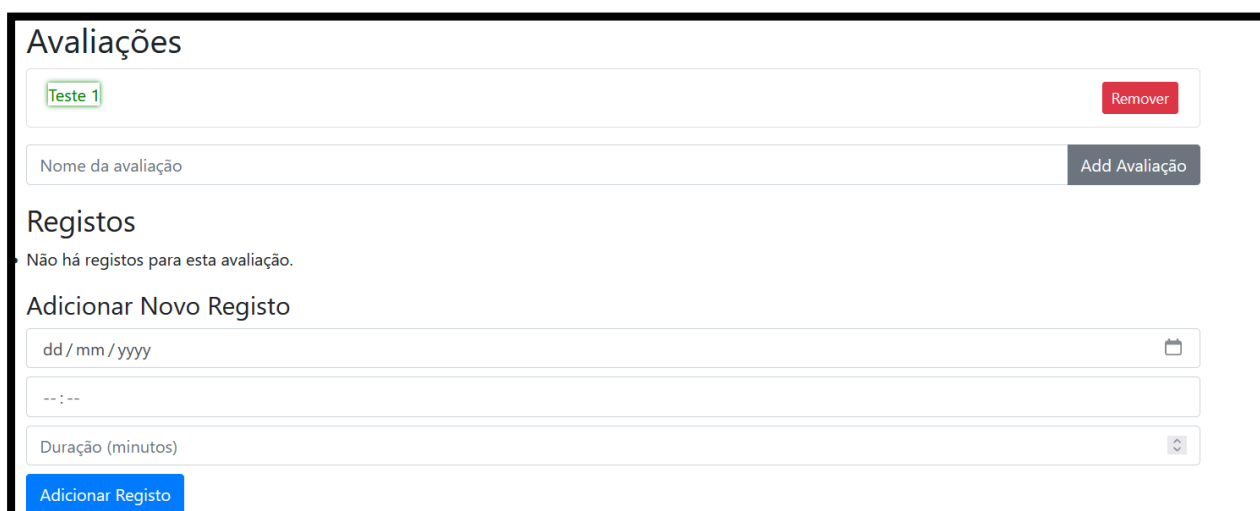


Figura 11 - Adicionar novo registo de tempo gasto para estudar

Para testar funcionalidade que o sistema não permite adicionar registo sem ter todos os campos preenchidos, não coloquei nada na parte de minutos gastos ao estudar neste registo.



Figura 12 - Testar funcionalidade de adição do registo com campos vazios

A caixa de aviso esta ser apresentada no seguinte formato:

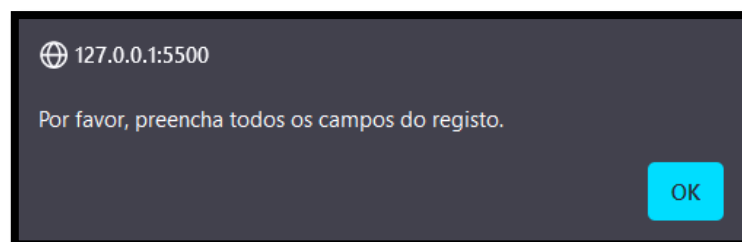


Figura 13 - Caixa de aviso para preencher todos os campos para conseguir adicionar registo com sucesso

Logo para testar funcionalidade de adicionar sucessivamente um registo, coloque minutos no ultimo campo do registo.

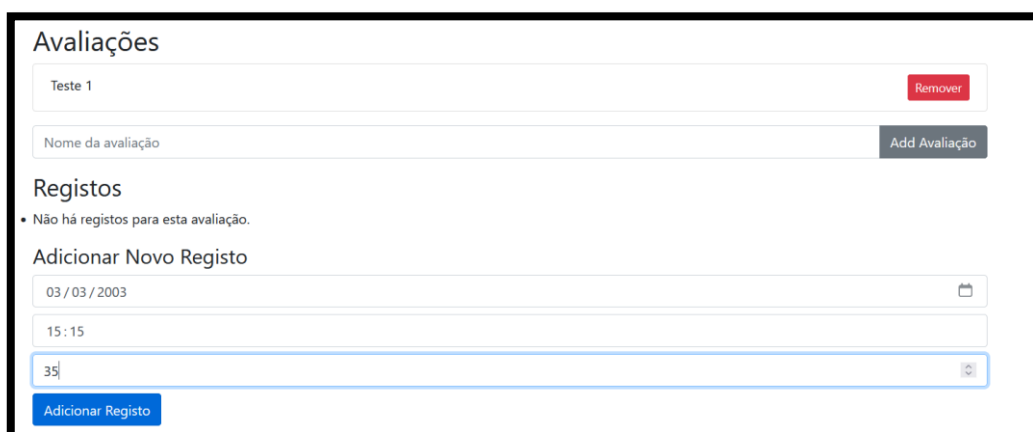


Figura 14 - Adicionar Registo com sucesso

Para confirmar que o registo foi adicionado com sucesso, a caixa com esta informação esta apresentada ao utilizador.

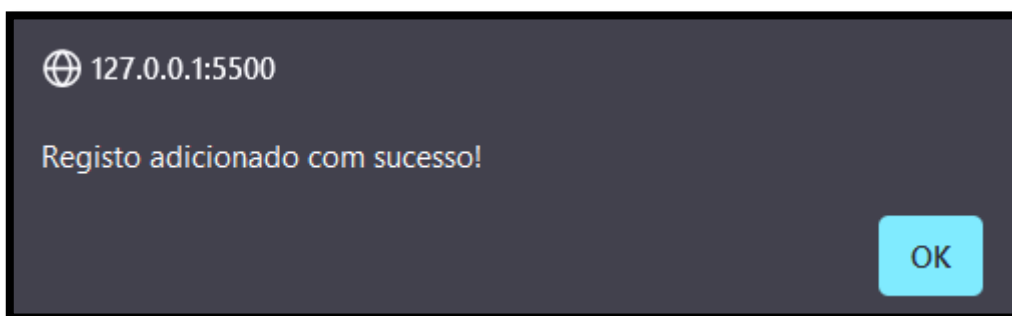


Figura 15 - Caixa de informação que o registo foi adicionado com sucesso

Apresentação de uma unidade curricular com avaliação e registo associados.

A interface de usuário é dividida em três seções principais. A primeira seção, intitulada "Lista de Unidades Curriculares", contém um formulário com campos para "Nome" (preenchido com "Matematica 2"), "Ano" (2024), "Semestre" (2) e "Minutos gastos" (35), seguido por um botão vermelho "Remove". Abaixo, há uma barra de filtros com campos para "Nome da unidade curricular", "Ano Curricular" (com uma seta para baixo) e "Semestre 1" (com uma seta para baixo), e um botão azul "Add". A segunda seção, intitulada "Avaliações", contém um formulário com o campo "Nome da avaliação" (preenchido com "Teste 1") e um botão vermelho "Remove", além de um botão cinza "Add Avaliação". A terceira seção, intitulada "Registos", contém um formulário com o campo "Registo" (preenchido com "2003-03-03 - 15:15 (35 minutos)") e um botão vermelho "Remove".

Figura 16 - Registo adicionado com sucesso

Para remover este registo, basta apertar botão “Remove” ao lado do registo que deseja apagar da memoria local Storage para aparecer seguinte caixa de confirmação:

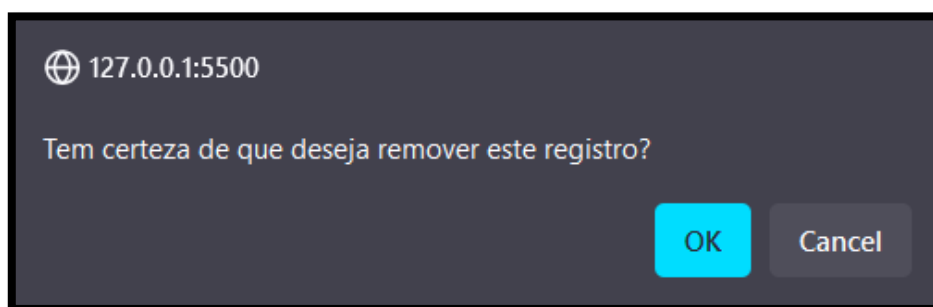


Figura 17 - Caixa de confirmação para remover o registo

## 3.2. Requisitos Não Funcionais:

### 3.2.1. Persistência dos dados utilizando localStorage.

Todos os dados registados neste sistema são guardados utilizando local Storage. Isso significa que o programa tal como o browser podem ser fechados, o atualizados ou ate o próprio computador pode ser reinicializado e os dados permanecem e podem ser visualizados apos de reabrir o programa num browser. O local Storage permite guardar os dados em forma persistente no dispositivo de utilizador. Para apagar os dados do local storage, eu utilizo tecla F12 e na parte de Storage -> local Storage aperto botao direito do rato para apagar toda informação armazenada.

Para guardar um objeto no localStorage, primeiro precisa de converter lo numa string que um passo necessário pois o localStorage não permite guardar os objetos, mas permite guardar os strings que podem ser convertidos facilmente através do JSON.stringify como mostrado na figura 18.

```
localStorage.setItem('unidadesCurriculares', JSON.stringify(unidadesCurriculares));
```

Figura 18 - Conversão de um objeto em uma string para guardar lo no localStorage utilizando método JSON.stringify "setItem"

Quando seja necessário fazer recuperação dos dados do local Storage, so pode se recuperar os strings e logo converter los de novo num objeto como mostrado na figura 19.

```
const unidadesCurriculares = JSON.parse(localStorage.getItem('unidadesCurriculares')) || [];
```

Figura 19 - Conversão do string num objeto para apresentar utilizando método JSON.stringify "getItem"

### 3.2.2. Visualização dos dados selecionados no topo da página.

Quando seleccionar uma unidade curricular e avaliação, os dados aparecem no topo da página.

The screenshot displays the top section of a web application. It features two main selection areas: 'Unidade Curricular Seleccionado:' with a dropdown menu showing 'Matematica 2', and 'Avaliação Seleccionada:' with a dropdown menu showing 'Teste 1'. Below these, there is a section titled 'Lista de Unidades Curriculares' which contains a card for 'Matematica 2' with details: 'Ano: 2024', 'Semestre: 2', and 'Minutos gastos: 35'. A red 'Remove' button is next to this card. At the bottom of this list, there are input fields for 'Nome da unidade curricular', 'Ano Curricular' (with a dropdown), and 'Semestre 1' (with a dropdown), followed by a blue 'Add' button. Below the list, there is a section titled 'Avaliações' which contains a card for 'Teste 1' with a red 'Remove' button.

Figura 20 - Visualização dos dados selecionados no topo da pagina

### **3.2.3. Interface de usuário agradável e lógica, implementada com o uso de Bootstrap e ficheiro do estilo escrito em css.**

Navegação nesta pagina foi feita em forma fácil e muito intuitiva, ou seja quando há palavras “clicáveis” o utilizador pode facilmente saber isso através de mudança do cor das palavras clicáveis que significam funcionalidades. Quando utilizador não preenche todos os campos, o sistema informe através de caixa de aviso e não permite adicionar objetos que não tem informação completa. As datas, horas e semestres só podem ser introduzidos se tiverem formato correto definido nas funções responsáveis por adição dos unidades curricular, avaliações ou registos do tempo. Quando o objeto esta adicionado com sucesso, ele aparece logo na pagina abaixo da informação nesta secção.

## **4. Código JavaScript**

### **4.1. DOM – Document Object Model**

O DOM é uma interface de programação que permite aos desenvolvedores aceder e manipular documentos HTML como uma árvore de objetos. Cada elemento do documento é representado como um objeto no DOM, o que permite a modificação dinâmica da estrutura, estilo e conteúdo do documento.

No contexto deste projeto, o DOM é manipulado para exibir, adicionar e remover unidades curriculares, avaliações e registos de tempo. Neste capítulo vou mostrar algumas funções que demonstram como o HTML pode ser manipulado através do código JavaScript.

### **4.2. Funções JavaScript**

#### **4.2.1. Variáveis Globais**

As variáveis globais são utilizadas para conseguir concluir o requisito de rastrear unidade curricular e avaliação atualmente selecionadas e logo exibir logo no topo da pagina.

```
let currentUnidadeIndex = null;  
let currentAvaliacaoIndex = null;
```

Figura 21 - Variáveis Globais

#### **4.2.2. Inicialização do DOM**

Inicialização do DOM garante que o código JavaScript vai ser executado só quando o DOM vai ser carregado totalmente.

```
document.addEventListener('DOMContentLoaded', function() {
```

Figura 22 - Inicialização do DOM

#### 4.2.3. Exibição das Unidades Curriculares

Esta função exibe as unidades curriculares armazenadas no `localStorage`. Cada unidade é exibida que possa ser clicável e são apresentadas em forma de uma lista, que permite ao utilizador intuitivamente perceber que após de clicar do nome possa fazer outra funcionalidade, como adicionar avaliação para esta unidade curricular ou exibir de suas avaliações.

```
function displayUnidadesCurriculares() {  
  const listElement = document.getElementById('unit-list');  
  listElement.innerHTML = '';  
  document.getElementById('selected-unit').classList.remove('d-none');  
  
  const unidadesCurriculares = JSON.parse(localStorage.getItem('unidadesCurriculares')) || [];  
  
  if (unidadesCurriculares.length === 0) {  
    const message = document.createElement('li');  
    message.textContent = 'Não há unidades curriculares disponíveis.';  
    listElement.appendChild(message);  
  } else {  
    unidadesCurriculares.forEach((unidade, index) => {  
      const listItem = document.createElement('li');  
      listItem.className = 'list-group-item';  
      listItem.innerHTML = `  
        <strong class="unit-name" data-index="${index}">Nome:</strong> ${unidade.nome}<br>  
        <strong>Ano:</strong> ${unidade.ano}<br>  
        <strong>Semestre:</strong> ${unidade.semestre}<br>  
        <strong>Minutos gastos:</strong> ${unidade.minutos}<br>  
        <button class="btn btn-danger btn-sm float-right" onclick="removeUnidadeCurricular(${index})">Remover</button>  
      `;  
      listElement.appendChild(listItem);  
  
      listItem.querySelector('.unit-name').addEventListener('click', function() {  
        currentUnidadeIndex = this.getAttribute('data-index');  
        // Update the selected unit name at the top of the page  
        const selectedUnitName = document.getElementById('selected-unit-name');  
        selectedUnitName.textContent = unidadesCurriculares[currentUnidadeIndex].nome;  
  
        // Display evaluations for the selected unit  
        displayAvaliacoes(currentUnidadeIndex);  
      });  
      // Set cursor style to pointer  
      listItem.querySelector('.unit-name').style.cursor = 'pointer';  
    });  
  }  
}
```

Figura 23 - Exibição das unidades curriculares

#### 4.2.4. Adição de Avaliação

Esta função adiciona uma nova avaliação à unidade curricular selecionada. Ela verifica se o nome da avaliação foi fornecido, atualiza a unidade curricular correspondente no `localStorage`, e exibe a lista atualizada de avaliações.

```
function addAvaliacao() {  
  const nomeInput = document.getElementById('avaliacao-nome');  
  const nomeAvaliacao = nomeInput.value.trim();  
  
  if (!nomeAvaliacao) {  
    alert('Por favor, insira o nome da avaliação.');    return;  
  }  
  
  const unidadesCurriculares = JSON.parse(localStorage.getItem('unidadesCurriculares')) || [];  
  if (currentUnidadeIndex !== null && unidadesCurriculares[currentUnidadeIndex]) {  
    const avaliacao = {  
      nome: nomeAvaliacao,  
      registros: []  
    };  
    unidadesCurriculares[currentUnidadeIndex].avaliacoes.push(avaliacao);  
    localStorage.setItem('unidadesCurriculares', JSON.stringify(unidadesCurriculares));  
    nomeInput.value = '';  
  
    console.log('Updated unidadesCurriculares:', unidadesCurriculares);  
  
    displayAvaliacoes(currentUnidadeIndex);  
  }  
}
```

Figura 24 - Adição da Avaliacao



#### 4.2.5. Exibição das Avaliacoes

Esta função exibe as avaliações associadas a uma unidade curricular selecionada. Ela também permite a remoção de avaliações e a exibição de registos de tempo para cada avaliação.

```
function displayAvaliacoes(index) {
  const unidadesCurriculares = JSON.parse(localStorage.getItem('unidadesCurriculares')) || [];
  const selectedUnit = unidadesCurriculares[index];

  const avaliacoesList = document.getElementById('avaliacoes-list');
  avaliacoesList.innerHTML = ''; // Clear the current list

  if (selectedUnit && selectedUnit.avaliacoes && selectedUnit.avaliacoes.length > 0) {
    selectedUnit.avaliacoes.forEach((avaliacao, idx) => {
      const listItem = document.createElement('li');
      listItem.className = 'list-group-item';
      listItem.innerHTML = `
        <span class="avaliacao-name" data-avaliacao-index="${idx}">${avaliacao.nome}</span>
        <button class="btn btn-danger btn-sm float-right" onclick="removeAvaliacao(${index}, ${idx})">Remover</button>
      `;
      avaliacoesList.appendChild(listItem);

      listItem.querySelector('.avaliacao-name').addEventListener('click', function() {
        currentAvaliacaoIndex = this.getAttribute('data-avaliacao-index');
        displayRegistos(index, currentAvaliacaoIndex);

        // Update the selected evaluation name at the top of the page
        const selectedAvaliacaoName = document.getElementById('selected-avaliacao-name');
        selectedAvaliacaoName.textContent = avaliacao.nome; // Set the selected evaluation name
      });
      listItem.querySelector('.avaliacao-name').style.cursor = 'pointer';
    });
  } else {
    const message = document.createElement('li');
    message.textContent = 'Não há avaliações para esta unidade.';
    avaliacoesList.appendChild(message);
  }

  const avaliacoesSection = document.getElementById('avaliacoes-section');
  avaliacoesSection.classList.remove('d-none');
}
```

Figura 25 - Exibição das Avaliacoes

## 5. Resumo dos requisitos implementados e não implementados

Nesta secção coloquei uma tabela com requisitos resumidos. O único requisito que não foi implementado “Contador do tempo gasto atualizado para cada avaliação”, acredito que todos os requisitos menos este foram implementados com sucesso.

<b>Requisito</b>	<b>Implementado (Sim/Não)</b>
Página inicial com lista de unidades curriculares	Sim
Adição da nova unidade curricular	Sim
Remoção da unidade curricular com confirmação	Sim
Listagem das avaliações de uma unidade curricular	Sim
Adição de nova avaliação	Sim
Remoção da avaliação com confirmação	Sim
Listagem de registos de tempo para uma avaliação	Sim
Adição de novo registo de tempo	Sim
Remoção de registo de tempo com confirmação	Sim
Manipulação e armazenamento dos dados em forma dos objetos	Sim
Armazenamento dos dados permanente utilizando localStorage	Sim
Utilização do Bootstrap e forma do estilos css	Sim
Preenchimento obrigatório dos todos os campos	Sim
Contador do tempo gasto atualizado para cada unidade curricular	Sim
Contador do tempo gasto atualizado para cada avaliação	Não
Unidade curricular selecionada apresentada no topo da página	Sim
Avaliação selecionada apresentada no topo da página	Sim

## **6. Conclusão**

O sistema desenvolvido oferece uma solução eficiente para gerenciar o tempo dedicado a unidades curriculares e avaliações associadas aos mesmos. Utilizando tecnologias web como HTML, CSS, JavaScript e Bootstrap, foi criada uma plataforma intuitiva e funcional com os dados guardados no localStorage.

A metodologia incluiu testes para garantir a eficiência e robustez do sistema. Os requisitos funcionais e não funcionais foram implementados com sucesso menos um único requisito que foi mencionado no capítulo 4 que não implementei, proporcionando uma experiência de utilizador agradável e lógica.

Embora um requisito específico não tenha sido implementado, todos os outros foram atendidos com sucesso, demonstrando a utilidade do sistema como uma ferramenta de apoio ao gerenciamento de tempo dos estudantes.

## **7. Referencias**

Aulas de Professor Nuno Gil Fonseca sobre desenvolvimento para web incluindo os exemplos do código JavaScript.