

## Documentação do Projeto MVC: Simulador de Timer de Banheiro

Aluno	#### NOME COMPLETO ####
Aluno 1	Romeu Bernardes
Aluno 2	Guilherme Mosson
Aluno 3	Nicholas Basso

---

### Visão Geral:

Este projeto é um simulador de timer para controle de uso do banheiro em uma empresa, onde cada funcionário tem seu tempo para uso diário do banheiro, dependendo de suas condições, que estariam no cartão ID do funcionário. Só seria possível entrar no banheiro com esse cartão, que está simulado no projeto por botões. Caso o tempo dentro do banheiro se exceda, um desconto no salário é aplicado.

O projeto visa a organização do tempo dos funcionários no banheiro por parte da empresa, para solucionar possíveis problemas de produtividade por parte dos trabalhadores que passam tempo exagerado no banheiro.

---

### Tecnologias Utilizadas:

1. ASP.NET Core MVC:  
Estrutura de controle para organizar o código em camadas Model, View e Controller, cada um será explicada mais afrente;
2. Razor Pages:  
Permite misturar código C# com HTML de forma que você pode exibir informações vindas do servidor diretamente na página da web. Funciona como uma "ponte" entre a programação e o que aparece para o usuário;
3. HTML/CSS:  
HTML: A linguagem que cria a estrutura da página, como os botões e as imagens que você vê.  
CSS: O estilo que faz tudo ficar bonito, definindo cores, fontes e o layout da página.
4. JavaScript:  
Dá vida à página. É a linguagem que permite adicionar interatividade, usada em nosso projeto para definir quando o cronômetro começar a rodar, e trocar o fundo e outros objetos.
5. AJAX (jQuery):  
Permite que a página obtenha dados do servidor sem precisar ser recarregada inteira. Por exemplo, no nosso projeto quando se clica em um botão para selecionar um funcionário, a página pega os dados daquele funcionário sem precisar recarregar tudo.

## Estrutura do Código:

- **Model**

A Model é responsável por definir a estrutura dos dados dos funcionários. Cada funcionário possui atributos como nome, sexo, status de idade, se é portador de deficiência (PCD), e o tempo permitido para uso do banheiro.

Código da Model:

```
namespace timerBanheiro.Models
{
    public class FuncionariosModel
    {
        public int id{ get; set; }
        public string nome { get; set; } = string.Empty;
        public string statusIdade { get; set; } = string.Empty;
        public string sexo { get; set; } = string.Empty;
        public int descontoSalario { get; set; }
        public int pcd { get; set; }
        public int tempoSecs { get; set; }
    }
}
```

- **Descrição dos Atributos:**

id: Identificador único de cada funcionário;

nome: Nome do funcionário;

statusIdade, sexo e Pcd: seriam características específicas que influenciariam no tempo total do funcionário;

tempoSecs: Tempo permitido em segundos que o funcionário pode ficar no banheiro, definido pelas características do funcionário;

descontoSalario: Valor do desconto que será aplicado no salário.

---

- **Controller**

A Controller controla a interação entre a Model e a View. Ela gerencia as requisições feitas pelo usuário, processa os dados e retorna as informações necessárias para a interface.

- Possui uma lista estática de objetos do tipo FuncionariosModel, usada para simular uma base de dados de funcionários. Cada instância de FuncionariosModel representa um funcionário com seu nome, tempo permitido, status de idade, se é PCD (0 - falso, 1 - verdadeiro), entre outros dados.

Código exemplo de atributo da lista:

```
new FuncionariosModel {
    id = 1,
    nome = "Jorge",
    sexo = "H",
    statusIdade = "2",
    pcd = 0,
    tempoSecs = 60 + 30
}
```

- Método: **index()**

Este método é responsável por retornar a **View** principal com a lista de funcionários.

Código:

```
public IActionResult Index()
{
    return View(_funcionario);
}
```

Ao acessar a URL /Funcionarios, o método Index() retorna a lista de funcionários que pode ser visualizada ou manipulada na interface de usuário.

- Método: **GetFuncionario(int id)**

Este método é um HTTP GET que recebe o ID de um funcionário e retorna as informações desse funcionário específico no formato JSON, isto é, listado para se usar na View, se o funcionário não for encontrado, retorna um erro **404 (NotFound)**.

Utilizado nas interações AJAX (nas chamadas feitas por JavaScript na View) para buscar os dados do funcionário selecionado e enviar para ser usada na lógica de temporização do uso do banheiro, para cada funcionário ter seu próprio tempo.

Código:

```
[HttpGet]
public IActionResult GetFuncionario(int id)
{
    var func = _funcionario.FirstOrDefault(c => c.id == id);
    if (func == null)
    {
        return NotFound();
    }

    return Json(func);
}
```

Por exemplo, ao fazer uma requisição para */Funcionarios/GetFuncionario?id=1*, a controller retorna os dados do funcionário com **ID 1** (Jorge) no formato JSON.

---

- **View**

A View é responsável por exibir a interface e interagir com o usuário. Ela permite a escolha de um funcionário, exibe o cronômetro quando o funcionário entra no banheiro, e, se houver excesso do tempo no banheiro, calcula o desconto no salário do funcionário ao sair.

- *Div **foraBanheiro**:*

Para funcionamento correto da página, ela possui duas “div” principais (divisão de objetos pelo HTML), a “*foraBanheiro*”, que simula o lado de fora do banheiro, e a div “*banheiro*” que simula a parte interna do banheiro.

A div “*foraBanheiro*” contém os botões para escolha de funcionário, para simular o cartão de acesso que cada funcionário teria na empresa, e o botão de “entrar no banheiro”. Quando o usuário escolhe um funcionário, a função *getTempo()* é chamada para buscar o tempo permitido para aquele funcionário, já definido manualmente na *Controller*.

A função *entrarBanheiro()* é chamada ao clicar na porta.

- *Div banheiro:*

Além de simular a parte interna de banheiro, exibe e começa a contar o cronômetro daquele funcionário selecionado, também exibe o botão "Sair do banheiro" que chama a função `sairBanheiro()`, que, se acabar o tempo, é calculado o desconto no salário com base no tempo extra no banheiro, e volta para fora do banheiro.

- **Principais funções no script:**

- 1. Troca de Fundo (`trocarFundo()`):**

Alterna entre dois fundos de tela para simular a entrada e saída do banheiro;

- 2. Seleção de Funcionário (`getTempo()`):**

Faz uma requisição AJAX para buscar o tempo permitido no banheiro para cada funcionário;

- 3. Cronômetro (`atualizarCronometro()`):**

Controla o tempo de permanência do funcionário no banheiro e calcula o desconto salarial caso o tempo exceda o limite;

- 4. Saída do Banheiro (`sairBanheiro()`):**

Ao sair do banheiro, o cronômetro para e o desconto salarial é calculado e exibido na tela já fora do banheiro;

### Código da View:

```
@model List<timerBanheiro.Models.FuncionariosModel>
<html xmlns="http://www.w3.org/1999/xhtml">
    <link href="~/css/site.css" rel="stylesheet" />
    <link href="https://fonts.cdnfonts.com/css/seven-segment"
rel="stylesheet">
    <head>
        <title>Simulador de timer de banheiro (protótipo)</title>
    </head>
    <body>
        <div id="foraBanheiro">
            <div id="desconto" class="desconto_div"></div>
            <a onclick="entrarBanheiro()" id="botEntrar" class="image-link">
                <!--
-porta para entrar no banheiro-->
            </a>
            <div id="bots"><!--botões dos funcionarios-->
                <button onclick="jorge()">Eu sou Jorge</button>
                <button onclick="renatoAragao()">Eu sou Renato Aragão</button>
                <button onclick="amanda()">Eu sou Amanda</button>
                <button onclick="lorena()">Eu sou Lorena</button>
            </div>
        </div>
        <div id="banheiro"><!--dentro do banheiro-->
            <div id="cronometro" class="cronometro_div"></div>
            <div id="sair"><button onclick="sairBanheiro()">Sair do
banheiro</button></div>
        </div>
    </body>
</html>
```

```

<script>
    let fundoAtual = 1;
    let intervalo = null; // Variável para armazenar o estado do cronômetro
    document.body.style.backgroundImage = "url('/css/papel-de-parede-
tijolos.png')";
    function trocarFundo() {
        if (fundoAtual === 1) {
            document.body.style.backgroundImage = "url('/css/fundo-
privada.jpg')";
            fundoAtual = 2; // Altera para o segundo fundo
        } else if (fundoAtual === 2) {
            document.body.style.backgroundImage = "url('/css/papel-de-
parede-tijolos.png')";
            fundoAtual = 1; // Altera de volta para o primeiro fundo
        }
    }

    function sairBanheiro() {
        trocarFundo();
        let desconto = descontoSalario;
        document.getElementById("sair").style.display = "none";
        document.getElementById("banheiro").style.display = "none";
        document.getElementById("foraBanheiro").style.display = "flex";
        document.getElementById("botEntrar").style.display = "flex";
        if (desconto > 0) {
            desconto = desconto * 3;
            document.getElementById("desconto").innerHTML = `O desconto do seu
salário será: ${desconto} reais`;
        }
    }

    var tempo = 0;
    var descontoSalario = 0;

    function getTempo(id) {
        $.ajax({
            url: '@Url.Action("GetFuncionario", "Funcionarios")',
            type: 'GET',
            data: { id: id },
            success: function (data) {
                tempo = data.tempoSecs;
                console.log(tempo);
            }
        });
        document.getElementById("bots").style.display = "none";
    }

    function jorge() {
        getTempo(1);
        console.log(tempo);
    }

    function renatoAragao() {
        getTempo(2);
        console.log('Eu sou Renato Aragão');
    }

    function amanda() {

```

```

        getTempo(3);
        console.log('Eu sou Amanda');
    }

    function lorena() {
        getTempo(4);
        console.log('Eu sou Lorena');
    }

    function entrarBanheiro() {
        trocarFundo()
        document.getElementById("banheiro").style.display = "flex";
        document.getElementById("foraBanheiro").style.display = "none";
        document.getElementById("botEntrar").style.display = "none";
        document.getElementById("sair").style.display = "flex";

        let tempoInicial = tempo; // Usa a variável global tempo

        // Se o cronômetro já está rodando, não criar outro
        if (intervalo !== null) {
            return; // Evita que o cronômetro seja reiniciado
        }

        function atualizarCronometro() {
            if (tempoInicial <= 0) {
                document.getElementById("cronometro").style.fontSize = "3rem";
                document.getElementById("cronometro").style.color = "crimson";
                const minutos = Math.floor(tempoInicial / 60);
                const segundos = tempoInicial % 60;

                document.getElementById("cronometro").innerHTML =
                    `Tempo que será descontado do salário: ${segundos} s`;

                tempoInicial--;
                descontoSalario++;
            }

            else {
                const minutos = Math.floor(tempoInicial / 60);
                const segundos = tempoInicial % 60;

                document.getElementById("cronometro").innerHTML =
                    `${minutos}. ${segundos}`;

                tempoInicial--;
            }
        }

        // Atualize o cronômetro a cada 1 segundo
        intervalo = setInterval(atualizarCronometro, 1000);
        atualizarCronometro(); // Corrigido para não passar parâmetros
    }
}
</script>

```

---