

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO**

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

**DESENVOLVIMENTO DE GRÁFICOS UTILIZANDO O
ELEMENTO CANVAS**

Aluno: Mariana da Silva Fernandes

Prontuário: 1320611

São João da Boa Vista – SP

2016

Resumo

Texto do resumo...

Sumário

1	Introdução	4
2	Desenvolvimento	7
	2.1 O elemento Canvas.....	7
	2.2 API de tela.....	7
	2.2.1 Funções de formas e cor	8
	2.3 Elemento Canvas no projeto "Indra"	Erro! Indicador não definido.
3	Conclusões e Recomendações	12
4	Referências Bibliográficas	13

1 Introdução

Em janeiro desse ano, 2016, as cidades paulistas São João da Boa Vista e Águas da Prata foram atingidas por uma forte chuva, que provocou alagamentos em várias zonas das cidades, e com isso, um grande prejuízo para parte das populações.



Figura 1: Enchente na cidade de Águas da Prata – SP.

As súbitas mudanças climáticas que provocam esse tipo de ocorrência, têm sido cada vez mais comuns, não apenas em nossa região, mas ao redor de todo o mundo. Devido ao desaviso, essas variações do clima podem causar grandes danos as populações.

Tendo isso em vista, na disciplina Prática de Desenvolvimento de Sistemas, ministrada no Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus São João da Boa Vista, aos alunos do quarto do curso Integrado em Informática, foi proposto o desenvolvimento conjunto de uma portal virtual onde poderia ser consultado os dados referentes às variações climáticas, como umidade do ar, temperatura, nível de chuva, entre outros. Esses dados seriam

coletados por PCDs (Plataformas de coleta de dados), que seriam desenvolvidas por estudantes de outros cursos do Instituto Federa.

Assim, para que o projeto de desenvolvimento desse portal, que, após a votação dos alunos, recebeu o nome fantasia de Indra, fosse organizado e funcional, foram criados cinco módulos de desenvolvimento, sendo estes: [1] Usuários, responsável pelo desenvolvimento de atividades relacionadas aos perfis de usuários e administradores, [2] PCDs, responsável pela gestão das PCDs, [3] Medições, responsável pela gestão de medições das PCDs, [4] Relatórios, responsável pela gestão dos relatórios tabulares e gráficos dos dados coletados, e [5] Alertas, responsável pela gestão de alertas críticos das PCDs.

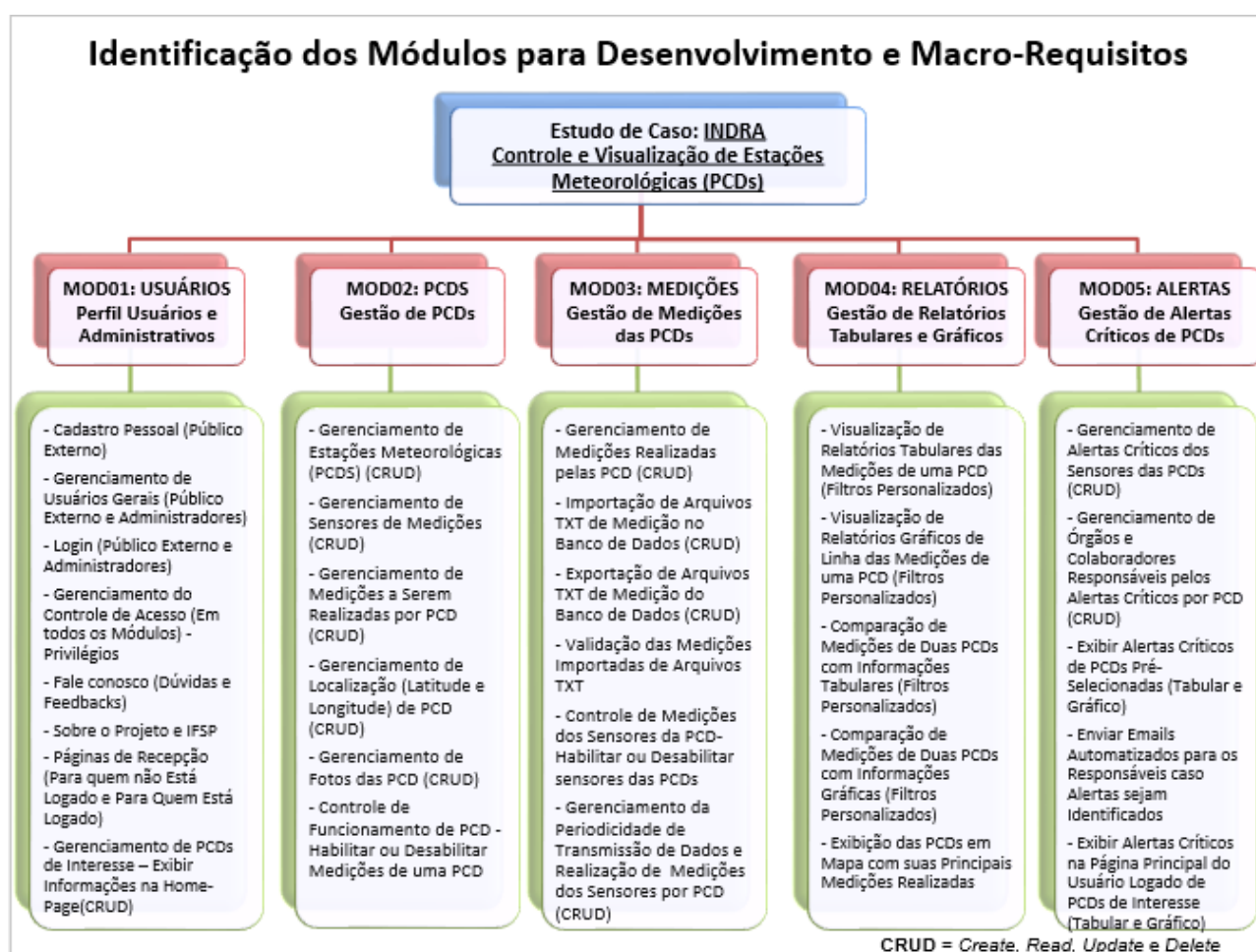


Figura 2: Divisão das funcionalidades e requisitos de cada módulo do projeto.

O quarto módulo do projeto, Relatórios, é de suma importância para o portal, uma vez que é o responsável por mostrar todos os dados provenientes das medições de cada PCD cadastrada, permitindo a visualização e comparação do histórico de todas as medições já realizadas. A visualização desses dados pode ser realizada através de tabelas e de gráficos.

Após o começo do desenvolvimento da funcionalidade que permite que um usuário possa visualizar os dados disponíveis no portal através de um gráfico, pôde-se notar que seria necessário utilizar um novo recurso de desenvolvimento gráfico para que essa função fosse implementada com sucesso, o elemento Canvas.

O Canvas é um elemento novo em HTML5, que permite que se desenhem elementos gráficos utilizando JavaScript [1], e será tratado ao longo desse relatório.

2 Desenvolvimento

2.1 O elemento Canvas

Por definição oficial do HTML5, o canvas é uma tela de bitmap dependente de resolução que pode ser usada para a renderização de elementos gráficos ou outras imagens visuais rapidamente [1], ou seja, o canvas permite desenhar elementos gráficos através da sua API de tela, utilizando a linguagem JavaScript.

Inicialmente, a tela do elemento canvas é uma área retangular em uma página HTML. Por padrão, é uma tela que não tem conteúdo ou fronteiras [2]. A medida que são utilizadas funções para desenhar elementos gráficos dentro dessa tela, essa passa a ser preenchida.

Abaixo nota-se o código necessário para a renderização de uma tela simples do elemento canvas, identificada como “meuElemento”, e a tela renderizada em si. Essa tela possui 300 pixels de largura, 200 pixels de altura, e, para possibilitar a visualização das delimitações da tela, uma borda sólida de 1 pixel.

```
<canvas id="meuElemento" width="300" height="200" style="border:1px solid #000000;"></canvas>
```

Figura 3 - Exemplo de código utilizado na renderização de uma tela simples do canvas.



Figura 4 - Tela "meuElemento" renderizada.

2.2 API de tela

A API (Application Programming Interface) de JavaScript do Canvas, que permite que se manipule elementos gráficos dentro de um elemento canvas, disponibilizando funções nativas que auxiliam o desenvolvimento dos desenhos [3].

Para se ter acesso a essas funções de desenho, é necessário que se faça referência ao contexto da tela que se está utilizando. O método utilizado para referenciar o contexto tela é o *getContext()*.

Na imagem abaixo mostra-se o código necessário para referenciar o contexto da tela “meuElemento”. Inicialmente, é usado o método *document.getElementById()*, para referenciar o elemento canvas, passando o seu ID como parâmetro. Após isso, é utilizado o método *getContext()* para referenciar o contexto da tela do elemento utilizado, passando “2d” como parâmetro, indicando o tipo de interface em que se trabalha com o elemento.

```
<canvas id="meuElemento" width="300" height="200" style="border:1px solid #000000;"></canvas>
<script>
    var elemento = document.getElementById('meuElemento');
    var contexto = elemento.getContext('2d');
</script>
```

Figura 5 - Exemplo de referência de contexto de um elemento canvas.

2.3 Funções do elemento canvas

Há um acervo amplo de funções que podem ser aplicadas na manipulação de um elemento canvas. Entre essas funções, as funções de formas e cores se destacam. Com essas funções, é possível desenhar retângulos, círculos, curvas, além de ser possível trabalhar com diversas cores no preenchimento e contorno das formas. Alguns exemplos de desenvolvimento de formas podem ser vistos abaixo.

O primeiro exemplo é o desenvolvimento de um retângulo utilizando o método *rect()* para desenhar o retângulo na tela, passando como parâmetro a posição do retângulo no eixo horizontal do elemento (5 pixels), a posição do retângulo no eixo vertical do elemento (5 pixels), sua largura (150 pixels), e sua altura (100 pixels). Além disso, também é utilizado o método *stroke()*, para desenhar o contorno do retângulo.


```

<canvas id="meuElemento"></canvas>

<script>

    var elemento = document.getElementById('meuElemento');
    var contexto = meuElemento.getContext('2d');
    contexto.rect(5, 5, 150, 100);
    contexto.stroke();

</script>

```

Figura 6 - Código necessário para desenhar um retângulo utilizando o canvas.



Figura 7 - Retângulo desenhado na tela utilizando o canvas.

No próximo exemplo é desenvolvido o desenho de um círculo na tela. Para renderizar o círculo é utilizado o método *arc()*, que recebe como parâmetros a posição do círculo no eixo horizontal do elemento (100 pixels), a posição do círculo no eixo vertical do elemento (100 pixels), o valor do raio do círculo (80 pixels), o ângulo inicial em radianos (0 pixels), e o ângulo final em radianos do círculo ($2 * \pi$). Além disso, também é utilizado o método *stroke()*, para desenhar o contorno do círculo.

```

<canvas id="meuElemento" width="1000" height="1000"></canvas>

<script>

    var elemento = document.getElementById('meuElemento');
    var contexto = meuElemento.getContext('2d');
    contexto.arc(100, 100, 80, 0, Math.PI * 2);
    contexto.stroke();

</script>

```

Figura 8 - Código necessário para desenhar um círculo utilizando o canvas.

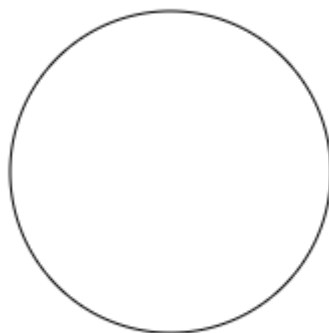


Figura 9 - Círculo desenhado na tela utilizando o canvas.

{maaaaaais}

2.4 Gráficos gerados no projeto “Indra”

Como foi dito na introdução desse relatório, o desenvolvimento do projeto “Indra” foi segmentado em diferentes módulos, cada um desses módulos sendo responsável por um aspecto distinto do projeto, e atendendo alguns requisitos fundamentais do projeto.

Assim, o módulo denominado “Relatórios” é o responsável por desenvolver todas as funcionalidades relacionadas a visualização dos dados cadastrais das PCDs, como sua localização e sensores, e relacionados aos dados coletados por essas PCDs.

Os requisitos básicos desse módulo são: exibição da localização de todas as PCDs já instaladas em um mapa; exibição de relatórios em formato tabular das medições realizadas por uma PCD; exibição de relatórios em formato tabular sobre comparações entre medições realizadas por duas ou mais PCDs; exibição de relatórios em formato gráfico das medições realizadas por uma PCD; e exibição de relatórios em formato gráfico sobre comparações entre medições realizadas por duas ou mais PCDs.

Para os requisitos relacionados a apresentação de informações gráficas, foi definido que os gráficos do tipo linear e do tipo barra seriam os mais indicados para exibir adequadamente essas informações. Abaixo nota-se ex

Dessa forma, visto que duas funcionalidades básicas a serem desenvolvidas pelo módulo “Relatórios” (exibição de relatórios em formato gráfico das medições realizadas por uma PCD e exibição de relatórios em formato gráfico sobre comparações entre medições realizadas por duas ou mais PCDs) deveriam conter elementos gráficos, fez-se necessário a pesquisa de uma forma de desenvolvimento de elementos gráficos, e a aplicação dessa no projeto.

2.4.1 Aplicação do canvas

3 Conclusões e Recomendações

Texto...

4 Referências Bibliográficas

- [1] <https://msdn.microsoft.com/pt-br/library/dn151487.aspx>
- [2] http://www.w3schools.com/html/html5_canvas.asp
- [3] <http://blog.triadworks.com.br/html5-desenhando-graficos-no-browser-com-canvas>