



---

**UX LBL05 CC 1A 2025 - Aula  
03**

---

**Visualização de Grafos 3:**  
Animação e Interatividade

---

# AGENDA DO MÓDULO

**Visualização De Grafos 1:**  
Aplicações, Biblioteca D3.Js  
E Exercícios

01

s03

**Visualização De Grafos 2:**  
Representações De Dados

02

s06

Exercício 1: Representação de Dados 1 ponto

**Visualização De Grafos 3:**  
Animação E Interatividade

03

s08

Exercício 2: Animação e Interatividade 1 ponto

**Testes De Usabilidade**

04

s09

Registro dos Testes 1 ponto

# ARTEFATOS

Orientação UX/Design



## **ART.3** Entendimento da Experiência do Usuário

1. Personas
2. User Stories

Orientação UX/Design



## **ART.8** Protótipo do front-end

1. Protótipo interativo
2. Wireframes ou Rascunhos
3. Fluxo de Usabilidade

Orientação UX/Design



## **ART.10** Front-end da aplicação

Primeira Entrega Parcial

Orientação Computação

UX/Design

## **ART.14** Aplicação integrada: front-end + execução dos algoritmos

Entrega do front-end integrado aos algoritmos (pelo menos 2) que processam o grafo e resolvem o problema do projeto.

Orientação Computação

UX/Design

## **ART.18** Refinamento da Aplicação

# ARTEFATO

## Dúvidas dos feedbacks de front-end?

## □ Atendimentos durante o DEV

### <> Desenvolvimento projeto (Semana 08)



Aplicação integrada (front-end e execução dos algoritmos)

👤 Orient. Tomaz Mikio

🕒 Ponderada: 3 pontos

📋 Obrigatória

#### 📖 Descrição

Crie as pastas necessárias para a implementação da sua aplicação debaixo da pasta /apps

Crie um arquivo chamado aplicacao.md dentro da pasta /artefatos e descreva a aplicação implementada, quais as pastas envolvidas e qual é o procedimento para executá-la.

##### # Descrição

Espera-se que a equipe faça a **entrega do front-end integrado** aos algoritmos (pelo menos 2) que processam o grafo e resolvem o problema do projeto. .

##### # Critérios de avaliação

- . A aplicação permite executar dois (ou mais) algoritmos ensinados nas instruções pelos professores de Matemática e Programação (peso 4).
- . A aplicação permite comparar os resultados obtidos com os diferentes algoritmos implementados (peso 4).
- . Há evidências de que a aplicação passou por um conjunto de testes para verificar o seu funcionamento correto (peso 2).

Sugestão de atribuição de nota:

Não deu a largada (0 - 4); No caminho (4,1 - 7); Chegou lá (7,1 - 9); Foi mais longe (9,1 - 10)

## PROJETO DO MÓDULO

### Como foi a Review na sexta?

Como a interface **ajuda o gestor a entender** quais chamados ainda não foram atendidos ou estão próximos do tempo-limite?

Durante a navegação, é possível manter uma **visão geral do sistema** sem perder o contexto?

A visualização atual permite **identificar zonas** sem cobertura adequada de gasistas?



# REVISITANDO OS AUTOESTUDOS



## D3 Transitions

D3 transitions let you smoothly **animate** between different **chart states**. This article shows how to add transitions to selection updates, how to set transition duration, how to create staggered transitions, how to change the easing function, how to chain transitions and how to create custom tween functions.

For example, let's join an array of 5 numbers to SVG circles. When the 'Update data' button is clicked, the data is randomised and the circles jump to new positions:



Update data



Edit in CodePen

If we add a transition to the circles the circles move smoothly to their new locations:



Update data

Edit in CodePen

<https://www.d3indepth.com/transitions/>



You can then add a `.transition()` call before the `.attr` and `.style` methods that you wish to transition:

```
function update() {  
  select('svg')  
    .selectAll('circle')  
    .data(data)  
    .join('circle')  
    .attr('cy', 50)  
    .attr('r', 40)  
    .transition()  
    .attr('cx', function(d) {  
      return d;  
    });  
}
```

In this example, the `.transition` call is followed by a single `.attr` call. This results in the `cx` attribute of the circles transitioning into new positions:



Update data



## Picking, Dragging and Brushing with D3

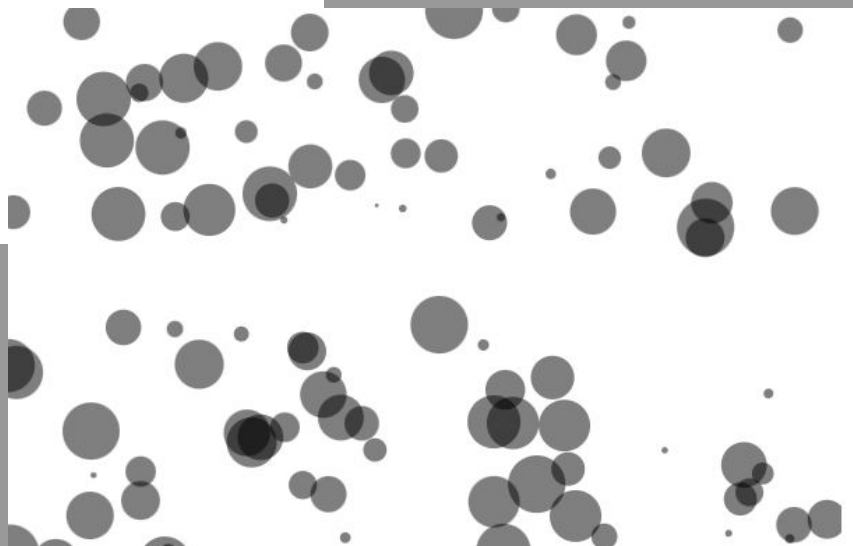
D3 provides a number of modules that help you add interactivity such as zooming, panning and brushing.

In this chapter we cover:

- **quadtree** (to make picking small items easier)
- **dragging** (for moving elements)
- **brushing** (for selecting regions)

Zoom and pan are covered in a [separate section](#).

<https://www.d3indepth.com/interaction/>



```
9 <style>
10
11   body { margin: 0; padding: 0; }
12   svg { background: ■ #f0f0f0; width: 100vw; height: 100vh; }
13   circle { fill: ■ steelblue; cursor: grab; }
14
15 </style>
16 </head>
17
18 <body>
19   <svg></svg>
20
21   <script>
22     const svg = d3.select("svg");
23
24     const drag = d3.drag()
25       .on("start", function (event) {
26         d3.select(this).raise().attr("stroke", "black");
27       })
28       .on("drag", function (event) {
29         d3.select(this)
30           .attr("cx", event.x)
31           .attr("cy", event.y);
32       })
33       .on("end", function () {
34         d3.select(this).attr("stroke", null);
35       });
36
37     svg.append("circle")
38       .attr("cx", 150)
39       .attr("cy", 150)
40       .attr("r", 40)
41       .call(drag);
42
43   </script>
```



## D3 Zoom and Pan

D3 provides a module `d3-zoom` that adds **zoom** and **pan** behaviour to an HTML or SVG element. This page shows how to create zoom behaviours, how to add zoom and pan constraints and how to zoom and pan programmatically.

In the following example, click and move the mouse to pan and use the mouse wheel to zoom. Touch gestures should also work:



**Autoestudo** (Semana 08)

Zoom e Pan em D3.js



# ATIVIDADE PONDERADA 2

VISUALIZAÇÕES INTERATIVAS

## ATIVIDADE PONDERADA 2

### VISUALIZAÇÕES INTERATIVAS COM D3.JS

## OBJETIVO

Criar uma **visualização de dados interativa** utilizando D3.js que explore os conceitos trabalhados em aula.

# ATIVIDADE PONDERADA 2

## VISUALIZAÇÕES INTERATIVAS COM D3.JS

### REQUISITOS

Estar conectada a um **conjunto de dados** em arquivo externo `.csv` ou `.json`;

Incluir pelo menos uma **animação** com `.transition()`;

Ter **interação** com o usuário, utilizando `quadtree()`, `drag()` ou `brush()`;

Implementar **zoom** ou **pan** com `d3.zoom()`.

# ATIVIDADE PONDERADA 2

## VISUALIZAÇÕES INTERATIVAS COM D3.JS

### ENTREGA

A entrega será feita em **duplas**,  
via repositório no GitHub.

O repositório deve conter uma  
**página HTML** funcional, que:

Gere a visualização completa;

Exiba uma legenda explicativa  
sobre o que está sendo  
representado.

## ATIVIDADE PONDERADA 2

VISUALIZAÇÕES  
INTERATIVAS  
COM D3.JS

## TEMA DA VISUALIZAÇÃO

Os alunos devem  
escolher um dos temas  
a seguir (A ou B).



## ATIVIDADE PONDERADA 2

### VISUALIZAÇÕES INTERATIVAS COM D3.JS

## TEMA A VISUALIZAÇÕES DO PROJETO

Dando **continuidade** à atividade anterior, as duplas devem implementar os requisitos de interação e animação na visualização relacionada ao projeto (**ponderada 01**).

A visualização deve:

- ❑ **Ter um objetivo claro** (ex: comunicar algo ao usuário, ilustrar parte da documentação ou demonstrar algum funcionamento);
- ❑ **Ter o contexto e o objetivo explicitado** na legenda ou incorporado à própria visualização.

## ATIVIDADE PONDERADA 2

### VISUALIZAÇÕES INTERATIVAS COM D3.JS

## TEMA A VISUALIZAÇÕES DO PROJETO

Algumas ideias...

Mapa interativo com animação de alocação dos gasistas e chamados.

Mapa interativo que filtra um gráfico de barras de atendimentos.

Grafo de alocação com destaque animado.

Gráfico de linha com zoom por intervalo de tempo que exibe chamados por hora

Visualização de deslocamento

Mapa com linha do tempo interativa que exiba os chamados

## ATIVIDADE PONDERADA 2

### VISUALIZAÇÕES INTERATIVAS COM D3.JS

## TEMA B

### DESENHOS INTERATIVOS

Para quem quiser explorar **visualizações mais artísticas** ou experimentar com interações lúdicas, a **Galeria de CC** está aberta! Os mesmos requisitos técnicos se aplicam, mas:

- ❑ A visualização deve ser esteticamente bem finalizada ou propor uma experiência de interação particularmente interessante para o usuário (**beleza x criatividade**).
- ❑ O texto de apoio deve conter o nome da obra e uma instrução de interação.

# ATIVIDADE PONDERADA 2

## VISUALIZAÇÕES INTERATIVAS COM D3.JS

## TEMA B DESENHOS INTERATIVOS

1. Usar o Figma para desenhar
2. Exportar SVG
3. Inserir o SVG no HTML
4. Utilizar D3.js para manipular os elementos SVG
5. Incorporar animações, interações e zoom

## ATIVIDADE PONDERADA 2

VISUALIZAÇÕES  
INTERATIVAS  
COM D3.JS

## PRAZO DE ENTREGA

1º **commit** no final da aula  
(participação em aula)

**Entrega final:**  
A combinar!



**AVALIAR ESSE  
ENCONTRO NA ADALOVE**



## PROJETO DO MÓDULO

DURANTE O DESENVOLVIMENTO!

10min # combinar horários



Acompanhamento  
de Projetos



Dúvidas?  
Sugestões?  
Melhorias?

