

# Engenharia de Software



```
attachEvent("onreadystatechange",H),e.at
boolean Number String Function Array Date
=();function F(e){var t=_le={};return
t[1])===!1&&e.stopOnFalse){r=!1;break}n=!
to=u.length;r&&(s=t,c(r))}return this},re
nction(){return u=[],this},disable:functio
ending",r={state:function(){return this,argument
romise)?e.promise().done(n.resolve).fail(n.
id(function(){n=s},t[1~e][2].disable,t[2][2
=0,n.h.call(arguments),r=n.length,i=1!=r||e
r),l=Array(r);r>t;t++)n[t]&&b.isFunction(nl
/><table></table><a href='/'>a</a><input ty
/TagName("input")[0],r.style.cssText="top:1px
test(r.getAttribute("style")),hrefNormalized:
```

# Agenda

---

- Apresentação
- Carga horária disciplina: 70h
- Horários
- Celular
- Avaliações
- Manter comunicação
- Contato ( [clenio.Fonseca@df.senac.br](mailto:clenio.Fonseca@df.senac.br) )
- Metodologia: Atividades teóricas, Exercícios , trabalhos, prática em laboratório\*, Projetos...



# O que é Engenharia de Software?

```
<!-- coverpop -->
<div id="CoverPop-cover" class="splash" ng-if="hasCover">
  <div class="CoverPop-content splash-center">

    <!-- the popup content (form, welcome message, etc.) -->

    <div style="padding-left: 15px; padding-right: 15px;">
      <a href="#" class="CoverPop-close pull-right"
        style="color:white;font-size: 18px;font-weight: 700;">X</a>
    </div>

    <a ng-href="{{ cover[0].meta | find_key:'link' }}" targets="_blank">
      
    </a>

    <!-- <a class="CoverPop-close" href="#">or skip signup</a -->

    </div>
  </div>
</div>
<script>
  window.fbAsyncInit = function () {
    FB.init({
      appId: '717776412180277',
      cookie: true,
      xfbml: true,
      version: 'v9.0'
    });

    FB.AppEvents.logPageView();

  });

  (function (d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) { return; }
    js = d.createElement(s); js.id = id;
    js.src = "https://connect.facebook.net/en_US/sdk.js";
    fjs.parentNode.insertBefore(js, fjs);
  })(document, 'script', 'facebook-jssdk');
</script>
```

```
<meta property="fb:pages" content="497792183708495" />
<meta property="fb:app_id" content="717776412180277" />
<meta property="og:title" content="{{title}}" />
<meta property="og:url" content="{{url}}" />
<meta property="og:description" content="{{description}}" />
<meta property="og:image" content="{{image}}" />
<meta property="og:image:width" content="1200" />
<meta property="og:image:height" content="630" />
<meta property="og:type" content="website" />
<meta property="og:site_name" content="tr.moot.net" />
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:site" content="{{twitterAccount}}" />
<meta name="twitter:title" content="{{title}}" />
<meta name="twitter:url" content="{{url}}" />
<meta name="twitter:description" content="{{description}}" />
```

```

--red: #dc3545;
--orange: #fd7e14;
--yellow: #ffc107;
--green: #28a745;
--teal: #20c997;
--cyan: #17a2b8;
--white: #fff;
--gray: #6c757d;
--gray-dark: #343a40;
--primary: #007bff;
--secondary: #6c757d;
--success: #28a745;
--info: #17a2b8;
--warning: #ffc107;
--danger: #dc3545;
--light: #f8d7da;
--dark: #343a40;
--breakpoint-xs: 0;
--breakpoint-sm: 576px;
--breakpoint-md: 768px;
--breakpoint-lg: 992px;
--breakpoint-xl: 1200px;
--font-family-sans-serif: -apple-system, BlinkMacSystemFont, "Segoe UI", "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Symbol", "Noto Color Emoji";
--font-family-monospace: SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;

```

# Conceitos

---

- Área da computação que se dedica ao desenvolvimento, manutenção e gerenciamento de sistemas de software de forma sistemática e controlada.
- A Engenharia de Software busca aplicar um conjunto de técnicas e processos para garantir que o desenvolvimento de software seja feito de maneira organizada e eficaz, desde a concepção até a entrega e manutenção do produto final.

# Principais aspectos da Engenharia de Software:



---

## Desenvolvimento:

- Criação de software, desde a definição dos requisitos até a implementação e testes.

## Manutenção:

- Adaptação do software a novas necessidades, correções de erros e melhorias contínuas.

## Gerenciamento:

- Planejamento, organização e controle do processo de desenvolvimento, garantindo que o software seja entregue dentro do prazo e orçamento.

## Qualidade:

- Adoção de práticas e ferramentas para assegurar que o software atenda aos requisitos do cliente e seja livre de erros.

## Processos:

- Utilização de metodologias e frameworks como Scrum, Kanban e Lean para otimizar o desenvolvimento.

## Ferramentas:

- Utilização de ambientes de desenvolvimento integrados (IDEs), ferramentas de teste automatizadas e outras tecnologias para facilitar o trabalho do engenheiro de software.



# Ementa

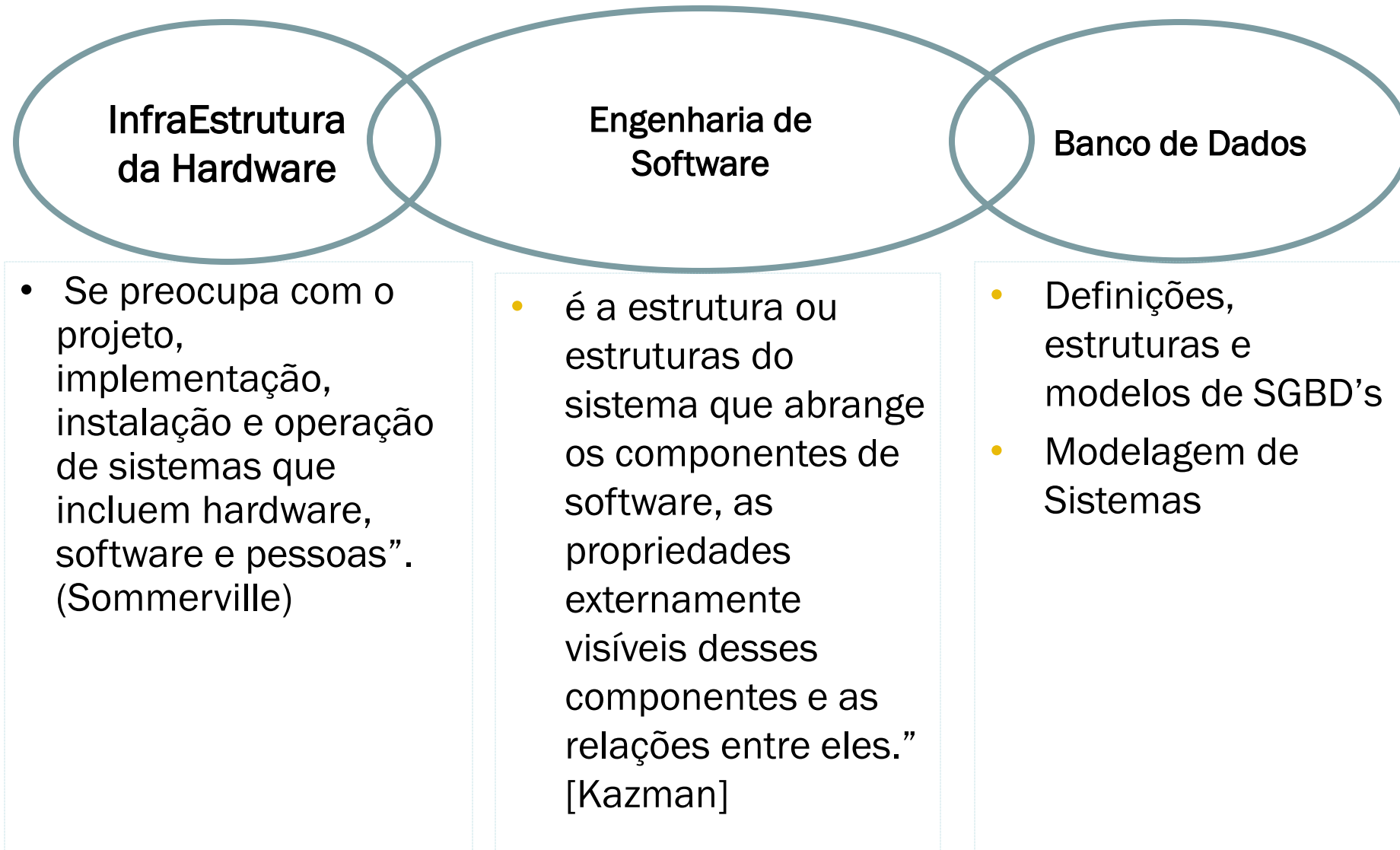
## Desafios

...especificação, implementação e aquisição de soluções de software que façam uma efetiva utilização dos recursos tecnológicos visando lidar com a crescente complexidade dos sistemas em contraste com as restrições de orçamento e cronograma.

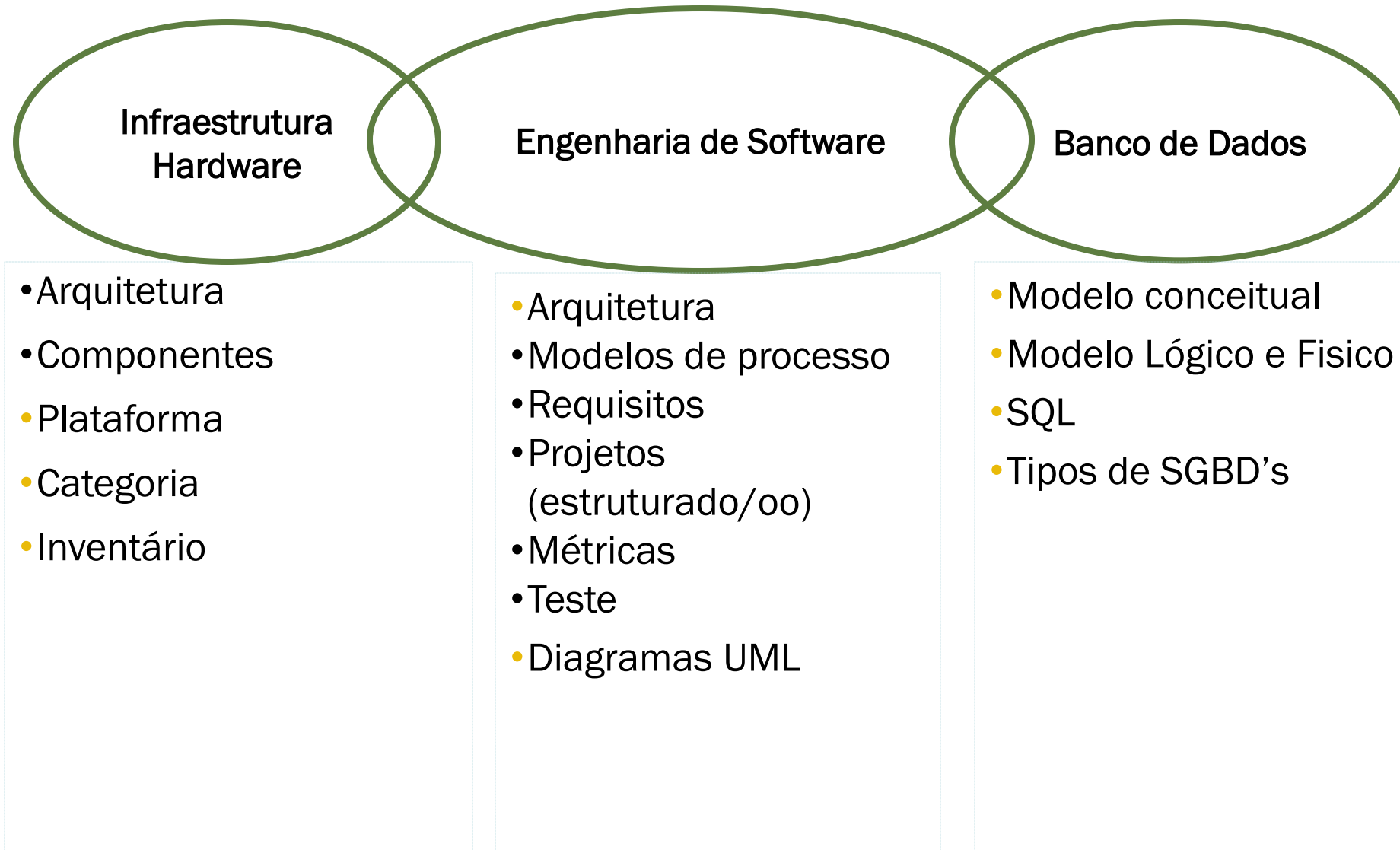
## Objetivo da Disciplina

O conhecimento adquirido na disciplina permitirá ao aluno ter uma visão geral do mercado de software e entender os componentes envolvidos no processo de desenvolvimento de sistemas complexos.

# Relacionamento

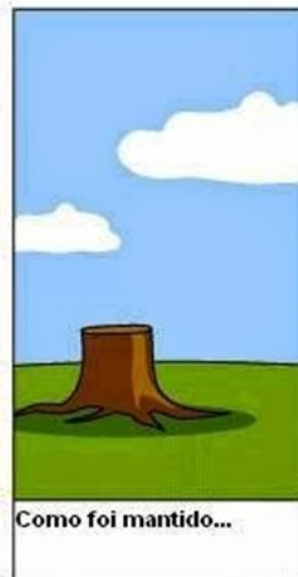
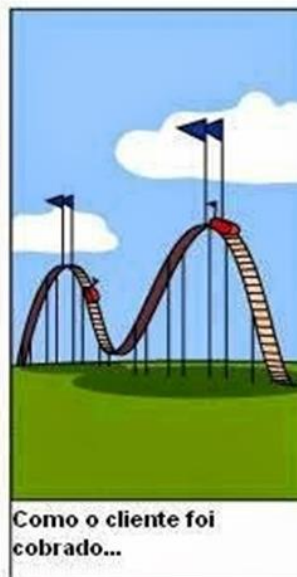
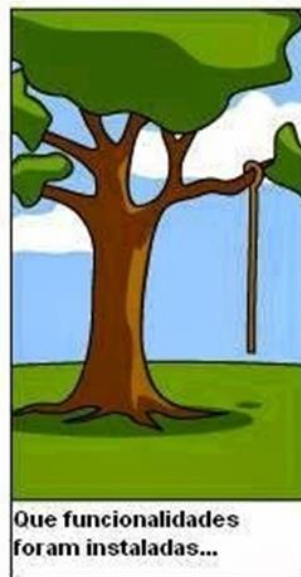
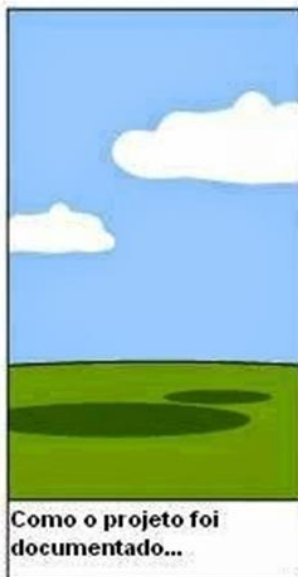


# Relacionamento





# Projeto de Software



# Definição: Arquitetura de Sistemas



- “Definição e representação de uma estrutura para a composição de um sistema em termos de seus componentes computacionais, suas propriedades e interações.” [Pressman]
- “A arquitetura de software de um programa ou sistema computacional é a estrutura ou estruturas do sistema que abrange os componentes de software, as propriedades externamente visíveis desses componentes e as relações entre eles.” [Kazman]

- O aumento do tamanho e da complexidade dos softwares

- A estrutura do software é importante, e adotar a estrutura correta pode trazer benefícios

Quem faz ?

- O Arquiteto de Software.



# Motivação: Arquitetura de Sistemas



# Quais habilidades desenvolveremos ?



- Pesquisar soluções de softwares tanto de licença livre quanto comercial
- Identificar as arquiteturas de software e suas respectivas aplicações e limitações;
- Identificar as categorias de softwares;
- Identificar as principais plataformas de execução de softwares.
- Conhecer as metodologias e projetos de Software.



# O que é software?



- **Programas** de computador
- Inclui documentação sobre operação e uso dos programas.
- Instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados
- Estruturas de dados que permitem a manipulação das informações;

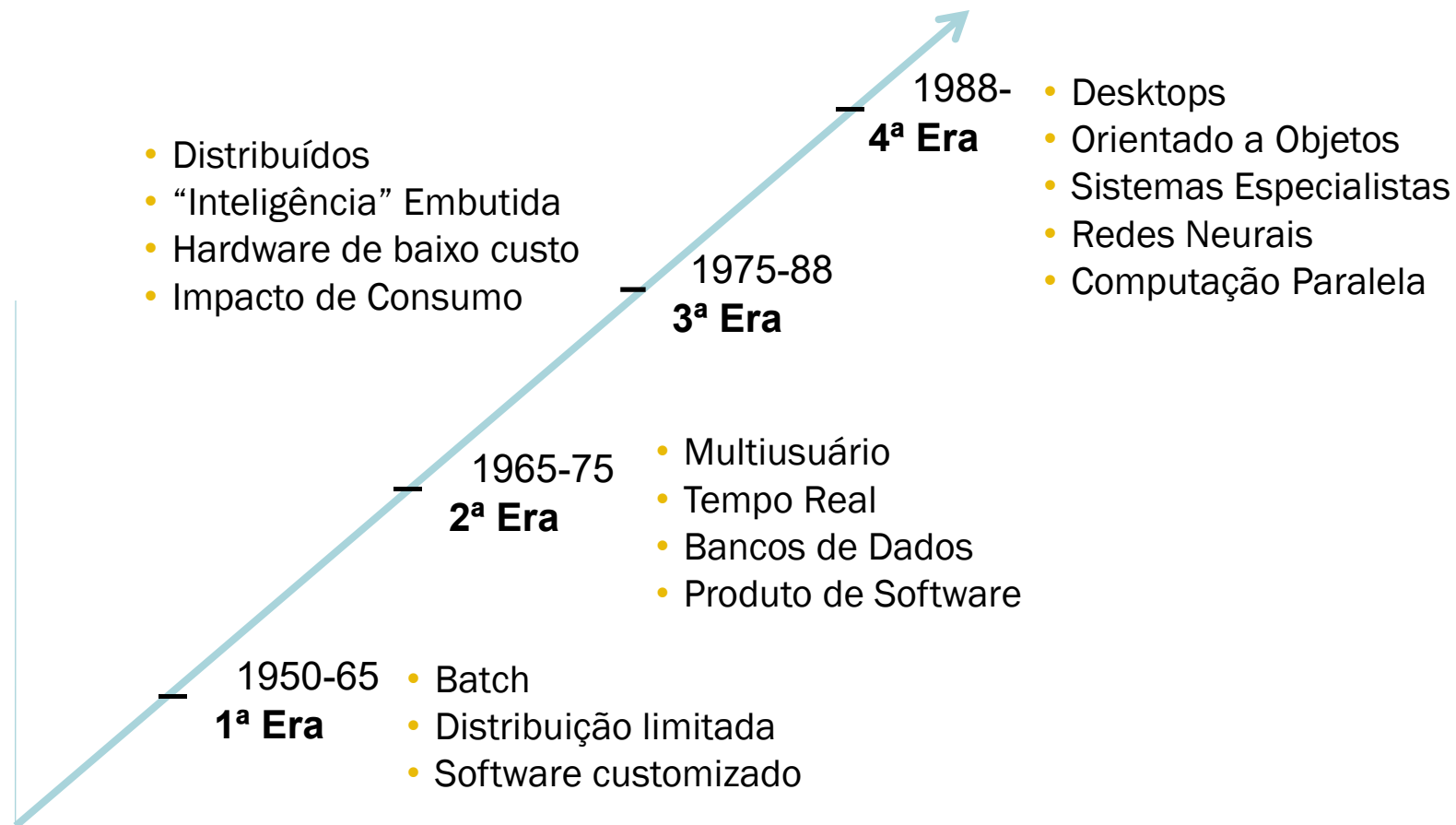
*“O Software ultrapassou o Hardware como chave para o sucesso de muitos sistemas baseados em computador” (Pressman, pg. 3, 1992)*

# A importância do Software

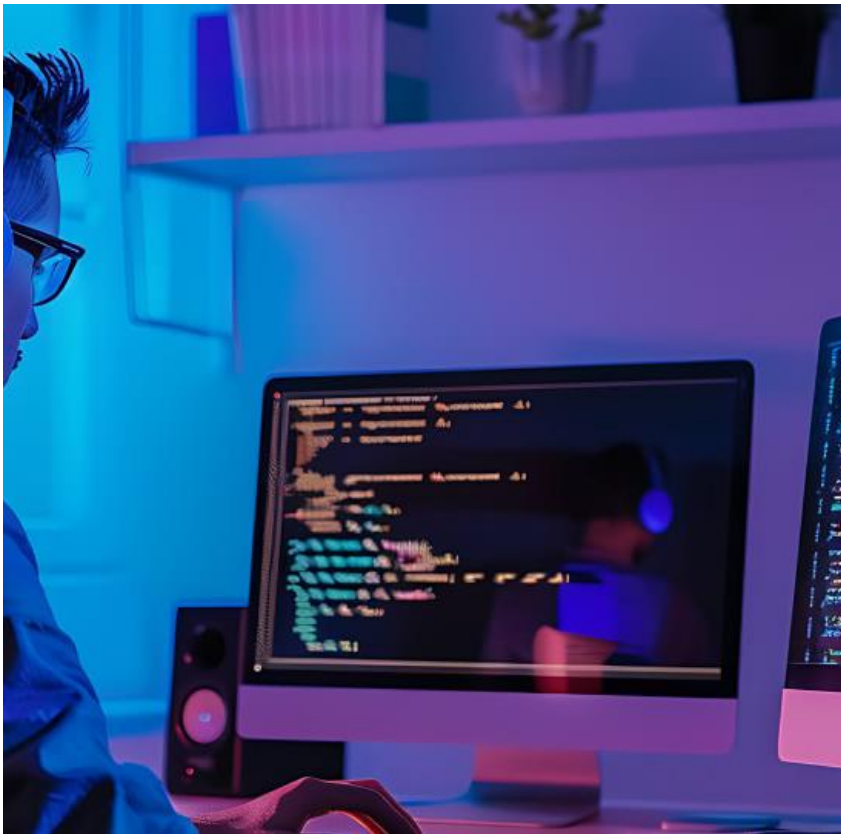
- Durante as 3 primeiras décadas da era do computador, o principal desafio era desenvolver um **HARDWARE** de baixo custo de processamento e armazenamento com alto desempenho.
- O hoje o desafio é melhorar a qualidade (e reduzir os custos) das soluções baseadas em **SOFTWARE!**



# A evolução do software



# Características do Software



- Software é um elemento de sistema lógico, não físico.
- Software não se desgasta, mas se deteriora.
- A construção da maioria dos softwares é personalizada, ou seja, não é montada a partir de componentes existentes.



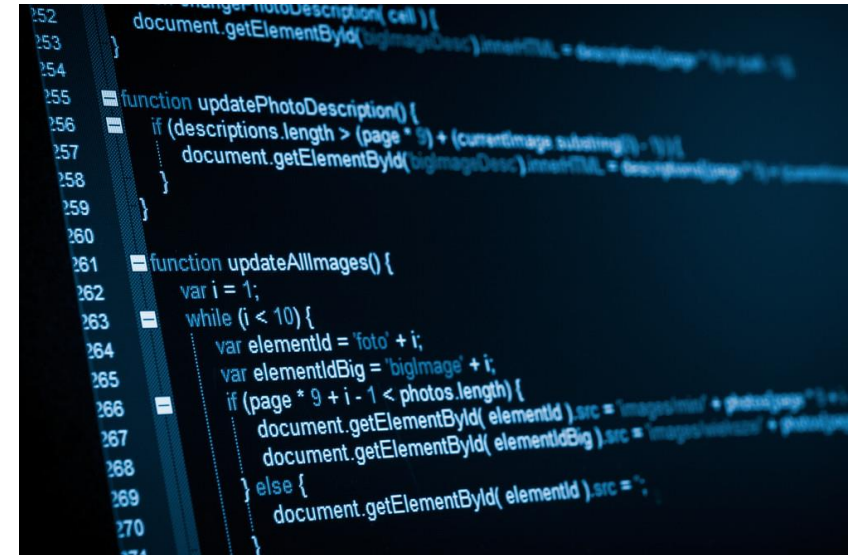
# Características do Software

- O Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico:

- Custos são concentrados no trabalho de engenharia.
- Projetos não podem ser geridos como projetos de manufatura.
- “Fábrica de Software!”

- Software não desgasta!

- Software não é sensível aos problemas ambientais que fazem com que o hardware se desgaste.
- Ver curvas de falha, páginas 14 e 15 do Pressman\*.
- Toda falha indica erro de projeto ou implementação: manutenção do SW é mais complicada que a do HW.



```
252 document.getElementById( 'bigImageDesc' ).innerHTML = description[page * 9 + i];
253 }
254
255 function updatePhotoDescription() {
256   if (descriptions.length > (page * 9) + (currentImage.substring(0, 1))) {
257     document.getElementById( 'bigImageDesc' ).innerHTML = description[page * 9 + i];
258   }
259 }
260
261 function updateAllImages() {
262   var i = 1;
263   while (i < 10) {
264     var elementId = 'foto' + i;
265     var elementIdBig = 'bigImage' + i;
266     if (page * 9 + i - 1 < photos.length) {
267       document.getElementById( elementId ).src = 'images/' + photos[page * 9 + i - 1];
268       document.getElementById( elementIdBig ).src = 'images/' + photos[page * 9 + i - 1];
269     } else {
270       document.getElementById( elementId ).src = '';
271     }
272   }
273 }
```

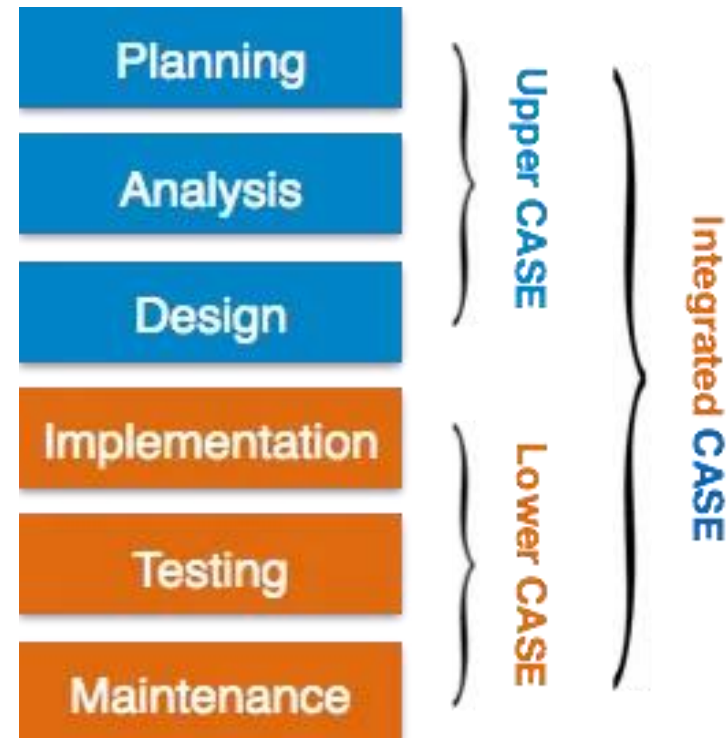
# Ferramenta CASE: Computer-Aided Software Engineering



- Provê suporte computacional a um determinado método ou linguagem
- Ambiente de desenvolvimento: conjunto de ferramentas integradas (CASE)
- Exemplos: Rational Rose, Jbuilder, ErWin, Power Designer, DBDesigner, System Architect
- *Ferramentas CASE* é uma classificação que abrange todas as ferramentas baseadas em computadores que auxiliam atividades de engenharia de software, desde análise de requisitos e modelagem até programação e testes

# CASE

- Facilita, com notação com sintaxe e semântica bem definidas
  - com representação gráfica ou textual
- Usada para descrever os artefatos gerados durante o desenvolvimento de software
- Exemplos: UML – Unified Modeling Language, Java



# UML- Unified Modeling Language



- Uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos.
- Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos.
- Por meio de seus diagramas é possível representar sistemas de softwares sob diversas perspectivas de visualização.
- Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores - por apresentar um vocabulário de fácil entendimento.

Silva, Douglas Marcos. "UML - Guia de Consulta Rápida", Novatec, 2001



# Outros tópicos da Disciplina



- APF
- Etapas do processo de Software (Levantamento de requisitos, Análise e projeto, Implementação, Testes, Implantação e manutenção)
- Metodologias Ágeis(XP, Kanban, Scrum)
- OO
- Documento de Visão
- RUP
- Requisitos
- Testes de Software



# Obrigado

---

Clenio Emidio

Clenio.fonseca@df.senac.br

[www.cecursos.com.br](http://www.cecursos.com.br)