

Engenharia de Software – ESWI2



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

Professor
Zady Castaneda Salazar

Agenda

- **Conceito de engenharia, software, e engenharia de software.**
- **Evolução de software.**
- **Crisis do software.**



*Para
Refletir*

O que vocês
entendem
por Engenheiro?



Engenheiro

Engenheiro (do latim **ingenium) é uma pessoa com formação técnico-científica que o torna capaz de resolver problemas tecnológicos, práticos e muitas vezes complexos, ligados à concepção, realização e implementação de produtos, sistemas ou serviços.**





*Para
Refletir*

O que vocês
entendem
por Engenharia?



Engenharia

“É a capacidade de aplicar os conhecimentos (matemáticos, técnicos e científicos) na criação, aperfeiçoamento e implementação de utilidades, tais como materiais, estruturas, máquinas, aparelhos, **sistemas ou processos**, que realizem uma determinada função ou objetivo.

Para obter tais resultados, o engenheiro estuda o problema, planeja uma solução, verifica a viabilidade econômica e técnica e por fim coordena o desenvolvimento ou produção.”

Exemplo de engenharia civil



A **Ponte de Tacoma** sempre balançava, porém neste dia o vento atingiu uma velocidade de aproximadamente 65 km por hora; com isto começou a gerar movimentos de torção, vindo a estrutura a colapsar.

https://pt.wikipedia.org/wiki/Ponte_de_Tacoma_Narrows

Exemplo de engenharia civil



Falkirk Wheel, conecta dois canais que foram divididos por 72 anos - Union Canal and Forth e Clyde Canal. Construído no 2002.

É um elevador de barcos rotativo, no trecho final do sistema de canais da Escócia, que permite que barcos pequenos possam acessar dois canais

<https://manojvictor.wordpress.com/page/2/>

<https://www.youtube.com/watch?v=n61KUGDWz2A>

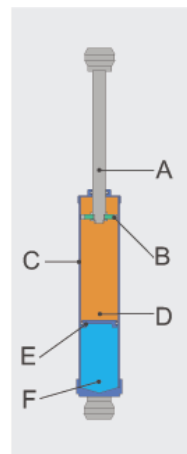
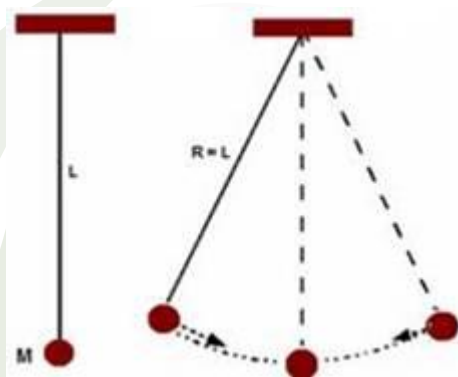


Exemplo de engenharia civil

CONHEÇA A TECNOLOGIA USADA NOS PRÉDIOS RESISTENTES A TERREMOTOS

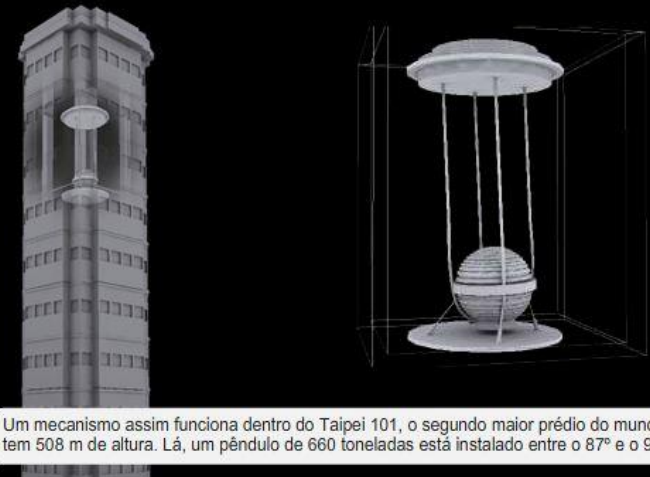


https://www.youtube.com/watch?v=_pDj7OawkYs



Contrapeso

Um meio de compensar as oscilações causadas pelo vento ou por tremores é instalar um grande peso no topo de prédios altos e movê-lo no sentido contrário ao do movimento do edifício.



Um mecanismo assim funciona dentro do Taipei 101, o segundo maior prédio do mundo, que tem 508 m de altura. Lá, um pêndulo de 660 toneladas está instalado entre o 87º e o 92º andar.



*Para
Refletir*

O que vocês
entendem
por Software?



Software

“Programas de computador e documentação associada. Os produtos de software podem ser desenvolvidos para um cliente específico ou para um mercado geral”.
(SOMMERVILLE,2007)



Software

- ❑ INSTRUÇÕES: que quando executadas produzem a função e o desempenho desejados
- ❑ ESTRUTURAS DE DADOS: que possibilitam que os programas manipulem adequadamente a informação
- ❑ DOCUMENTOS: que descrevem a operação e o uso dos programas



Característica do Software

- Desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico
- Não se desgasta mas se deteriora
- A maioria é feita sob medida em vez de ser “montada” a partir de componentes existentes



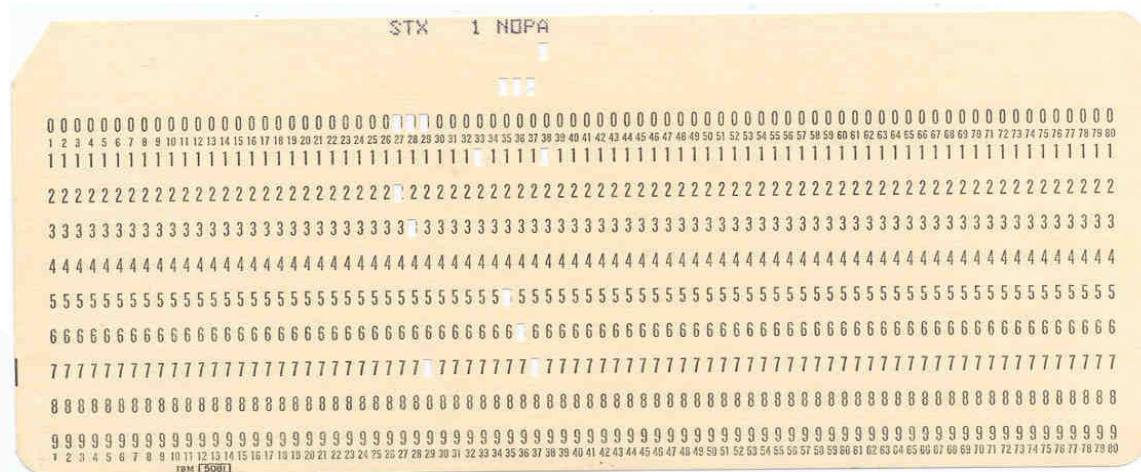
Evolução do Software

- Nos anos 40, quando se iniciou a evolução dos sistemas computadorizados, grande parte dos esforços, e consequentes custos, era concentrada no desenvolvimento do hardware, em razão, principalmente das limitações e dificuldades encontradas na época.
- Os sistemas foram desenvolvidos e financiados por governos para fins militares, como o ENIAC.



Evolução do Software

➤ À medida que a tecnologia de hardware foi sendo dominada, as preocupações se voltaram, no início dos anos 50, para o desenvolvimento dos sistemas operacionais, assim como das chamadas linguagens de programação de alto nível, como **FORTRAN** e **COBOL**, e dos respectivos compiladores.



Evolução do Software

(1950 - 1965)

- O hardware sofreu contínuas mudanças
- O software era uma arte "secundária" para a qual havia poucos métodos Sistemáticos
- O hardware era de propósito geral
- O software era específico para cada aplicação
- Não havia documentação

Evolução do Software

➤ Já no início dos anos 60, com o surgimento dos **sistemas operacionais** com características de multiprogramação, a eficiência e utilidade dos sistemas computacionais tiveram um considerável crescimento, para o que contribuíram também, de forma bastante significativa, as constantes quedas de preço do hardware.

Evolução do Software

(1965 - 1975)

- Multiprogramação e sistemas multiusuários
- Sistemas de tempo real
- 1ª geração de SGBD's
- Produto de software - *software houses*
- Bibliotecas de Software
- Manutenção quase impossível

Evolução do Software

➤ Uma consequência deste crescimento foi a necessidade, cada vez maior, de **desenvolver grandes sistemas de software** em substituição aos pequenos programas aplicativos utilizados até então.

Evolução do Software

➤ Desta necessidade, **surgiu um problema** nada trivial devido à falta de experiência e à não adequação dos métodos de desenvolvimento existentes para pequenos programas, o que foi caracterizado, ainda na década de 60 como a "**crise do software**", mas que, por outro lado, permitiu o nascimento do termo "**Engenharia de Software**".

Engenharia de Software

“É um ramo da engenharia cujo foco é o desenvolvimento dentro de custos adequados de sistemas de software de alta qualidade.” (SOMMERVILLE,2007)

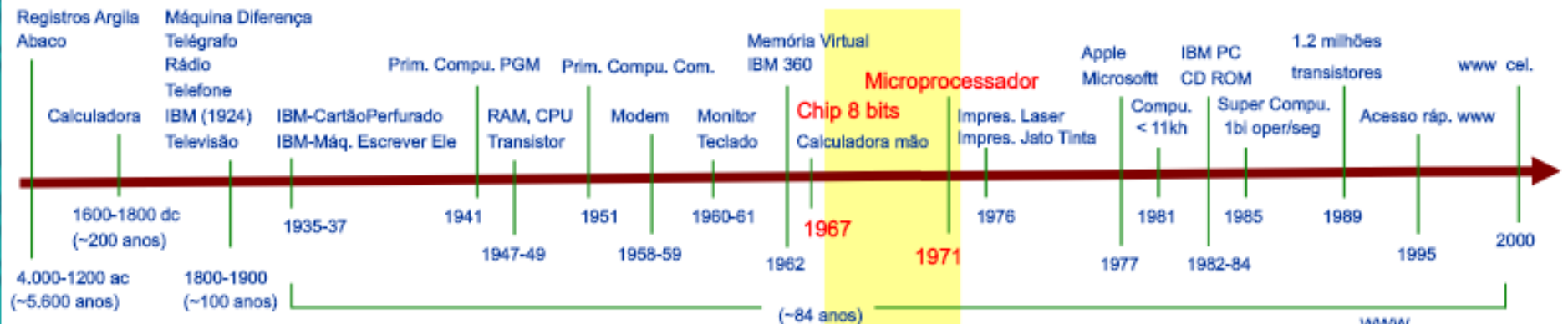


Evolução do Software

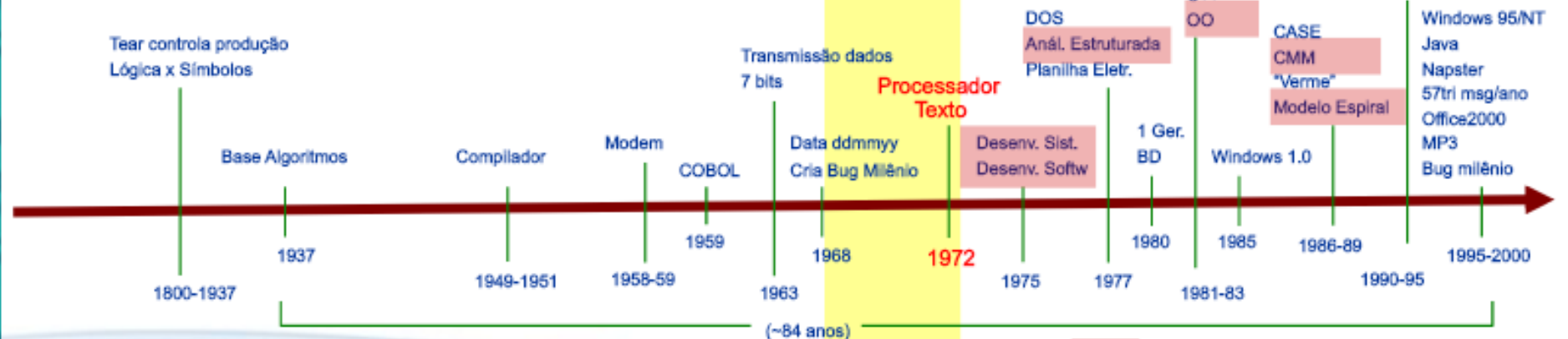
- A experiência inicial na construção de aplicações de computador mostrou que o desenvolvimento informal de software não era suficiente. Projetos importantes apresentavam, algumas vezes atraso.
- O software, cujo custo superava as previsões, não era confiável, era difícil de manter e seu desempenho era insatisfatório.
- O desenvolvimento de software estava em **CRISE**.
- Os **custos de hardware estavam caindo**, enquanto os **custos de software aumentavam rapidamente**.

Evolução do Software e Hardware

Evolução do Hardware



Evolução do Software



Fonte: IEEE Computer Society

Serão estudados em Engenharia de Software

Crise do Software

Evolução do Software

- A **crise do software** foi um termo utilizado nos anos 1970, quando a engenharia de software era praticamente inexistente.
- Expressava as dificuldades do desenvolvimento de software frente ao rápido crescimento da demanda por software, da complexidade dos problemas a serem resolvidos e da inexistência de técnicas estabelecidas para o desenvolvimento de sistemas que funcionassem adequadamente ou pudessem ser validados.

Evolução do Software

As causas da crise do software estão ligadas a complexidade do processo de software e a relativa imaturidade da engenharia de software como profissão.

A crise se manifesta de varias formas:

1. Projetos estourando o **orçamento**;
2. Projetos estourando o **prazo**;
3. Software de baixa **qualidade**;
4. Software muitas vezes não atingiam os **requisitos**;
5. Projetos ingereciáveis e o código difícil de **manter**.

Evolução do Software

➤ Atualmente, apesar da constante queda dos preços dos equipamentos, o custo de desenvolvimento de software não obedece a esta mesma tendência. Pelo contrário, corresponde a uma percentagem cada vez maior no custo global de um sistema informatizado.

Evolução do Software

➤ A principal razão para isto é que a tecnologia de desenvolvimento de software implica, ainda, em grande carga de trabalho, os projetos de grandes sistemas de software envolvendo em regra geral um grande número de pessoas num prazo relativamente longo de desenvolvimento.

Evolução do Software

- Novas técnicas e métodos são necessários para controlar a complexidade inerente aos grandes sistemas de software.
- Essas técnicas tornaram-se parte da **ENGENHARIA DE SOFTWARE** e são amplamente usadas hoje em dia.

Evolução do Software

- Porem, assim como aumentou a habilidade de produzir software, cresceu também a necessidade por sistemas de software mais complexos.
- Novas tecnologias resultantes da convergência de computadores e sistemas de comunicação, e as complexas interfaces com o usuário, impuseram novos desafios aos engenheiros de software.

Evolução do Software

➤ Como muitas empresas ainda não aplicam as técnicas de **ENGENHARIA DE SOFTWARE** de forma efetiva, muitos produzem software de baixa confiabilidade, com atraso e com custo além do orçamento.

Bibliografía:

- Capítulo 1 (Software and Software Engineering) do livro Software Engineering — A Practioner's Approach (de Roger Pressman)
- Capítulo 1 (Introduction) do livro Software Engineering (de Ian Sommerville)