

Engenharia de Sistemas de Software Contemporâneos

Guilherme Horta Travassos



Programa de Engenharia de Sistemas e Computação

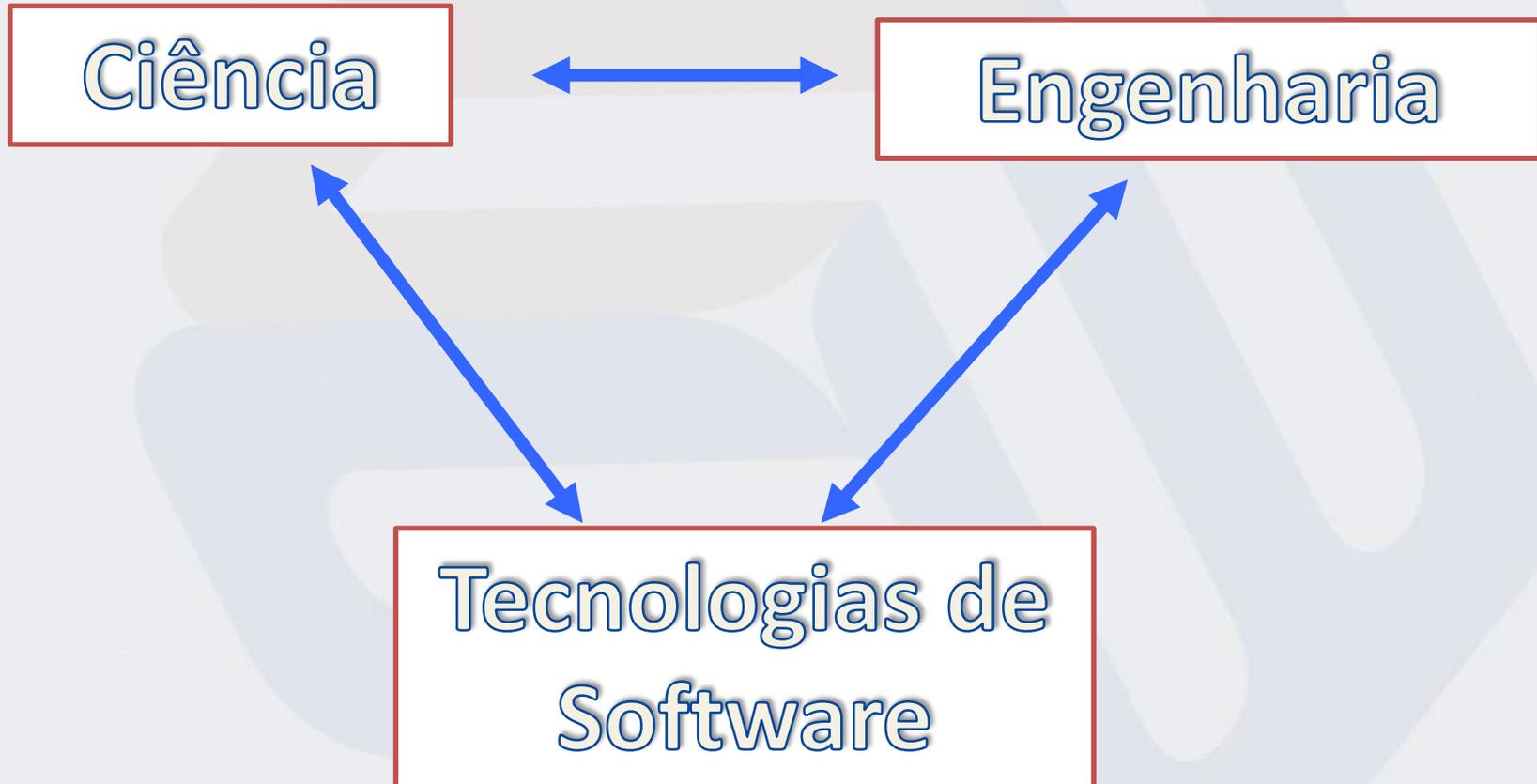
COPPE/UFRJ

Pesquisador **CNPq**, Membro da **ISERN**

ght@cos.ufrj.br

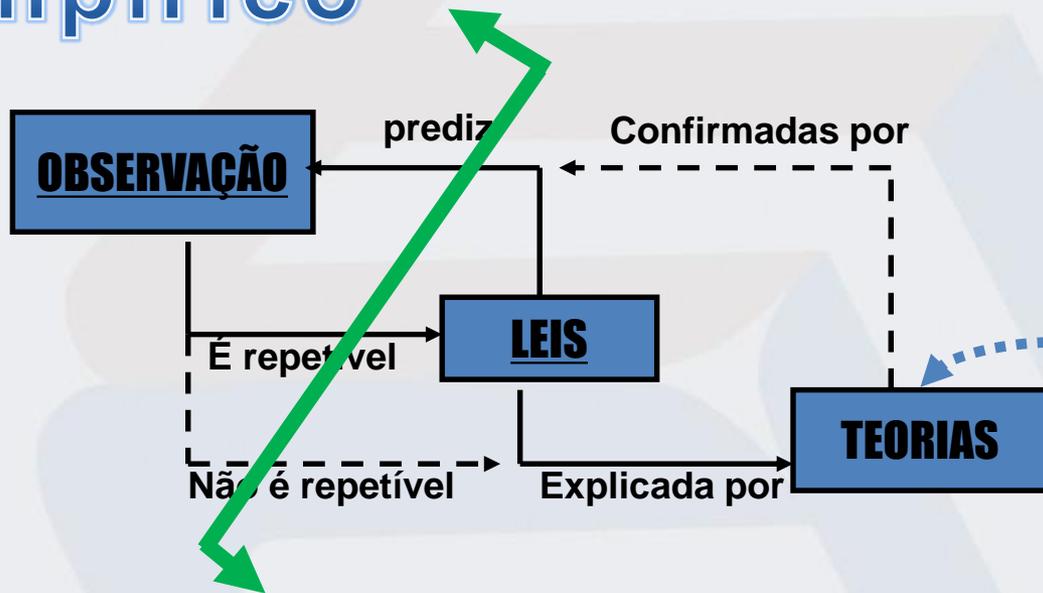
<http://www.cos.ufrj.br/~ght>

Ciência, Engenharia e Tecnologias



Ciência, Engenharia e Tecnologias

Empírico



Experimental



Ciência, Engenharia e Tecnologias

- Engenheiro

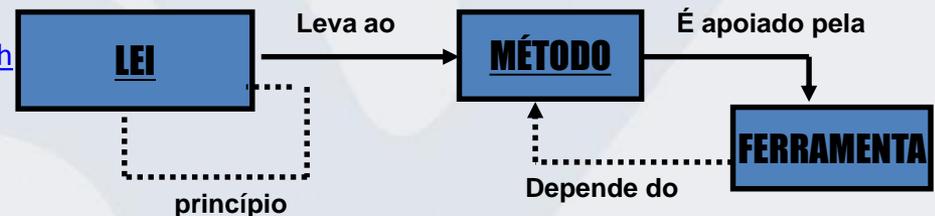
- Pessoa cujo trabalho se relaciona com projetar e construir aparatos, máquinas, motores ou coisas como estradas, pontes, edifícios dentre outros, porém sempre utilizando os princípios científicos

- Um engenheiro civil/eletricista
- Um engenheiro mecânico/ de estruturas
- Um engenheiro de software

- **Engenharia**

- O trabalho de um engenheiro, ou o estudo deste trabalho

<http://dictionary.cambridge.org/dictionary/british>



Breve História da Engenharia

- **Revolução Pré-científica**
 - Tentativa e erro. Imaginação... Primeiros engenheiros começaram a sistematicamente questionar o que funciona e porque.
- **Primeira Revolução Industrial**
 - Publicação de Galileu iniciou a análise estrutural e adoção de abordagem científica. Engenheiros desenvolvem a análise matemática e experimentos controlados. Disseminação de conhecimento através de sociedades profissionais.
- **Segunda Revolução Industrial**
 - Eletricidade e produção em massa. Currículos e escolas aparecem. Industrias químicas, elétricas, telecomunicações, marinha, aeronáutica e controle.
- **Terceira Revolução Industrial: A Era da Informação**
 - Pesquisa e Desenvolvimento. Simulação. Ciência em larga Escala. Engenharia baseada em Evidencia. **Software...**

Engenharia de Software

Conjunto de princípios, métodos e técnicas que tratam os sistemas de software como um produto de engenharia. Os quais requerem planejamento, gerenciamento, projeto, construção e manutenção.

Engenharia de Software



Desenvolvimento de Software

SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

Sistemas de Software

Amplamente utilizados por usuários diferentes
desenvolvedores

Usuários com diferentes perfis, desde usuários finais até desenvolvedores, precisam oferecer uma interface com o usuário a ser desenvolvido.

Normalmente devem ser testados antes de disponibilizados para uso. Falhas podem causar danos!

Portabilidade e manutenibilidade são características fundamentais para garantir sua longevidade e continuidade de uso.

Software está em todo lugar...

Algumas Características de Sistemas de Software

Software NÃO pode ser manufaturado (no sentido clássico)



X



Os custos do software se concentram em sua engenharia

Algumas Características de Sistemas de Software

Software não envelhece, mas deteriora...



Hardware

X



Software

Algumas Características de Sistemas de Software

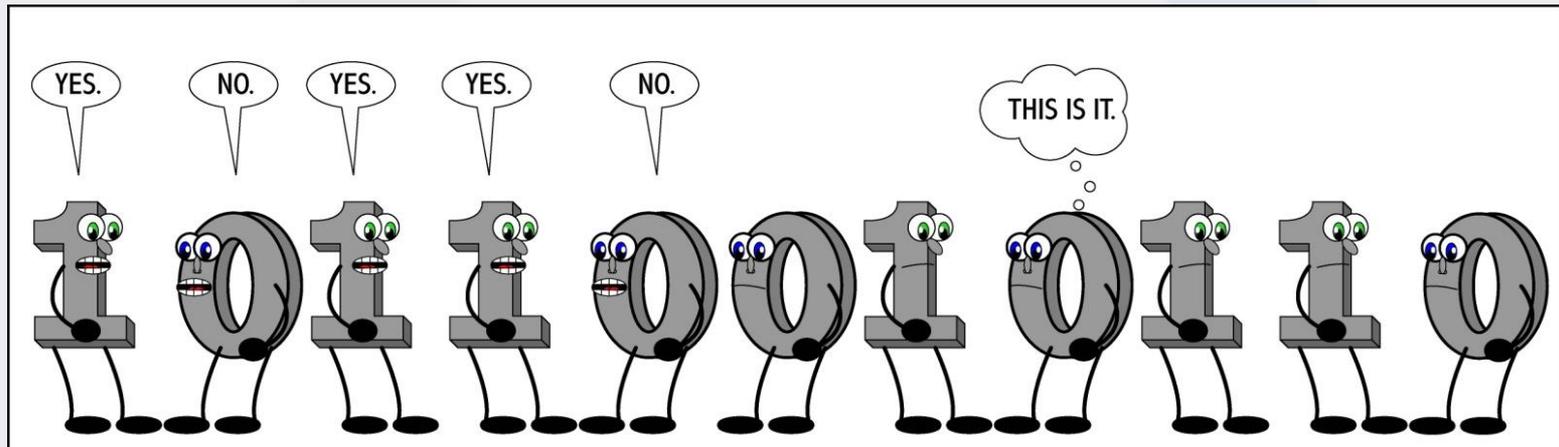
Usualmente construídos de forma customizada ao invés de montados com partes (alta qualidade) pré-construídas.



X



Algumas Características de Sistemas de Software



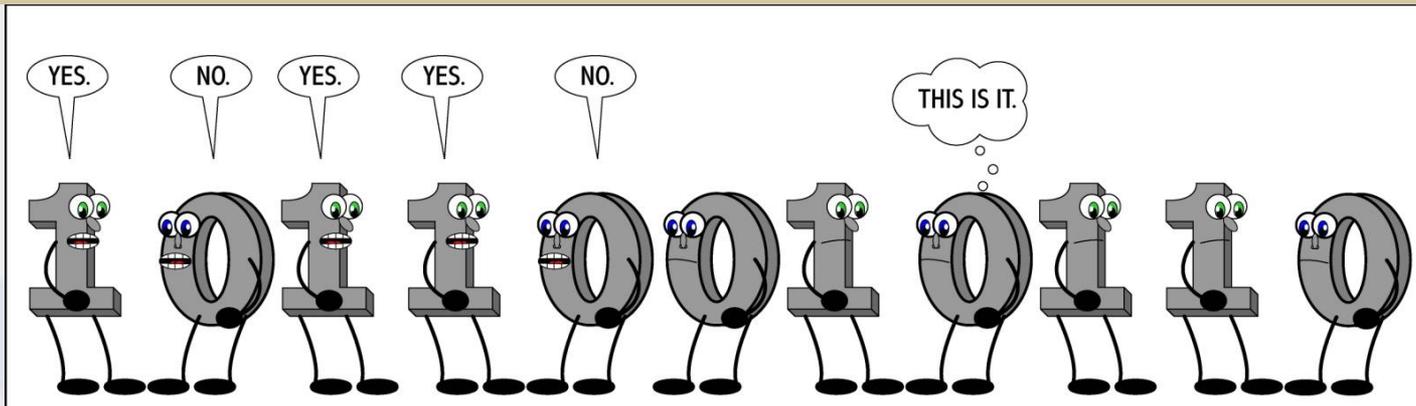
WWW.ALLNERDITY.COM

© 2013 CLAUDE WALRATH



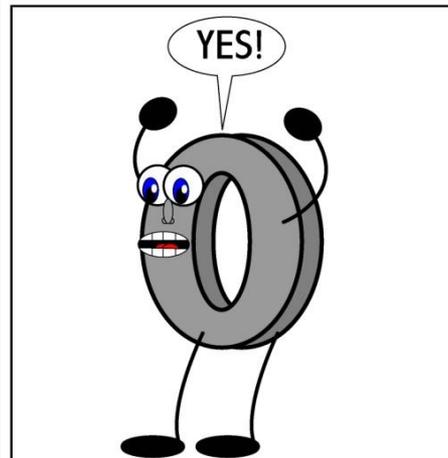
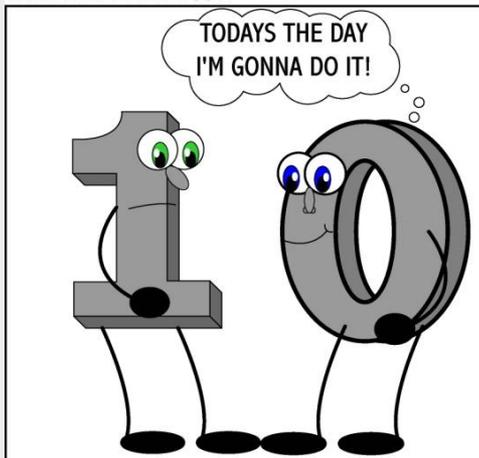
Algumas Características de Sistemas de Software

Todos os sistemas de software falham...



WWW.ALLNERDITY.COM

© 2013 CLAUDE WALRATH

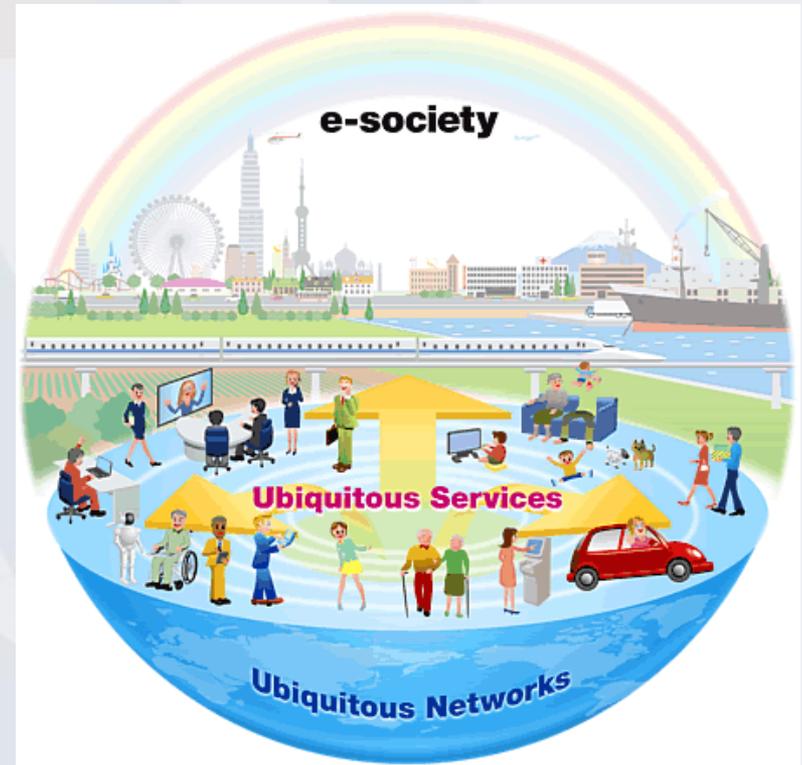


WARNING!
CATASTOPHIC
SYSTEM
FAILURE

Algumas Características de Sistemas de Software

A sociedade atual (e futura!) é altamente dependente de sistemas de software.

Enormes danos econômicos e sofrimento para a vida na Terra podem ocorrer quando os sistemas de software falham!



Os avanços no hardware continuam a superar nossa capacidade de construir software que utilize todo o potencial disponível

Algumas Características de Sistemas de Software

- Algumas Falhas de Software em 2017

(COMPUTERWORLDUK):

<http://www.computerworlduk.com/galleries/infrastructure/top-10-software-failures-2014-3599618/>

- Médicos perderam acesso ao prontuário de seus pacientes no Reino Unido
- Falha em software libertou prisioneiros nos EUA
- Casais Ingleses não conseguiram viajar para o Reino Unido
- Bloomberg cancelou depósito de US\$1 bilhão
- 600.000 pagamentos de cartão de crédito perderam
- Airbag de carro de emergência acionando erroneamente
- Falha no software da Starbucks
- F-35 detectou alvo incorretamente
- Clientes foram cobrados duas vezes por compra de alimentos
- Loja de produtos a 1
- Voo aéreo Britânico fechado
- Recall do Toyota Prius por falha em software
- Companhia Nacional de Gás Americana perdeu US\$1 bilhão
- Números de Emergência ficaram fora do ar por seis horas
- Apple foi forçada a atualizar o iOS8

por quê?

- Acesse <http://catless.ncl.ac.uk/Risks/> para ver mais problemas

Realidade dos Engenheiros de Software...

A engenharia de sistemas de software não segue uma trajetória suave...



Engenharia de Sistemas de Software

Em geral, segue um Processo de Desenvolvimento de Software que determina:

O ciclo de vida e o paradigma de desenvolvimento
As tecnologias de software (métodos, ferramentas) utilizadas durante o desenvolvimento
Quem participa (papéis) e quando
Os planos de gerenciamento, qualidade, verificação, validação e testes

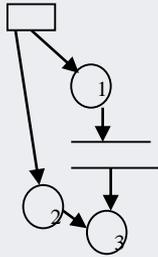


O processo define como os múltiplos desenvolvedores devem se comunicar e cooperar durante o projeto

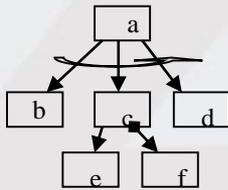
Engenharia de Sistemas de Software

Alguns paradigmas de desenvolvimento...

Funcional



Especificação



Projeto

```

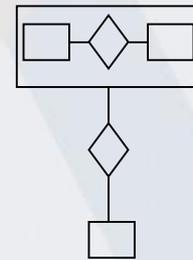
Program a
do while X
call b;
call c;
call d
end while;
  
```

Implementação

```

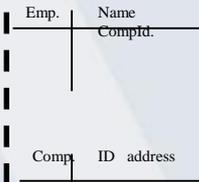
Subroutine c
call e;
if y then call f
end;
  
```

Orientado a Dados

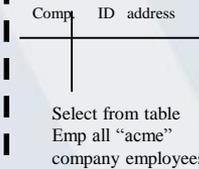


list all employees of a company

Especificação



Projeto



Create table employee(Name, Compid)

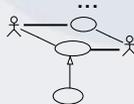
Create table Comp (Id, address)

Select from emp e, comp c where e.compid = c.id and c.id = "acme"

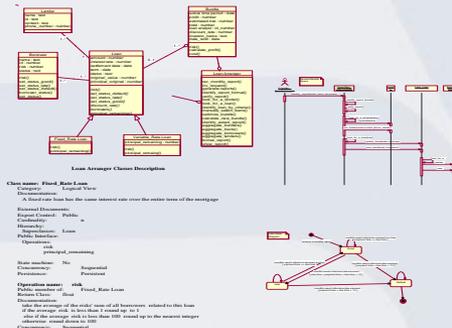
Implementação

Orientado a Objetos

3 – The gas station owner can use the system to control inventory. The system will either warn of low inventory or automatically order new parts and gas.



Especificação



Projeto

```

class parts
inherit from Stock_items;
attributes ...
services .....
relationships ...
  
```

Implementação

Realidade dos Engenheiros de Software...

A engenharia dos sistemas de software requer comunicação e colaboração entre os desenvolvedores e interessados no produto...

Não é fácil garantir a comunicação e a colaboração nos projetos...

Perspectivas de Comunicação nos Projetos de Software

REQUISITOS

Loan-Arranger Requirements Specification – Jan. 8, 1999

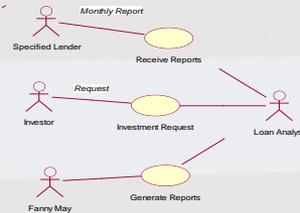
Background

Banks generate income in many ways, often by borrowing money from their depositors at a low interest rate, and then lending that same money at a higher interest rate in the form of bank loans. However, property loans, such as mortgages, typically have terms of 15, 25 or even 30 years. For example, suppose that you purchase a \$150,000 house with a \$50,000 down payment and borrow a \$100,000 mortgage from National Bank for thirty years at 5% interest. That means that National Bank gives you \$100,000 to pay the balance on your house, and you pay National Bank back at a rate of 5% per year over a period of thirty years. You must pay back both principal and interest. That is, the initial principal, \$100,000, is paid back in 360 installments (once a month for 30 years), with interest on the unpaid balance. In this case the monthly payment is \$536.82. Although the income from interest on these loans is lucrative, the loans tie up money for a long time, preventing the banks from using their money for other transactions. Consequently, the banks often sell their loans to consolidating organizations such as Fannie Mae and Freddie Mac, taking less long-term profit in exchange for freeing the capital for use in other ways.

FORMAL

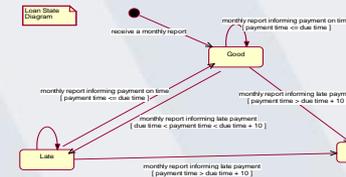
Triângulo Escaleno:

$$\{ \langle x, y, z \rangle : (x \neq y) \wedge (x \neq z) \wedge (y \neq z) \}$$

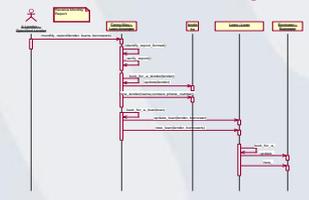
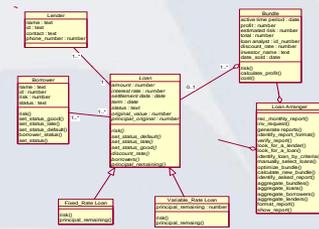


Domínio da Solução

TEST CASES			
CLASS	X	Y	Z
Scalene	3	4	5
Isosceles	5	5	8
Isosceles	3	4	3
Isosceles	4	7	7
Equilátral	2	2	2
No-triangle	1	2	3
No-triangle	5	1	4
No-triangle	3	5	2



Domínio Computacional



Requisitos Tácitos

AD-HOC

Domínio do Problema



Realidade dos Engenheiros de Software...

A engenharia dos sistemas de software requer tecnologias de software (métodos, técnicas, ferramentas e ambientes) para sua construção, entretanto...

Não temos evidencia suficiente sobre a maioria das tecnologias disponíveis...

Insuficiência de evidência sobre tecnologias...

Em sua palestra no ICGSE em 2012, Forrest Shull comentou:

Elicitação de Requisitos: 30 estudos cobrindo 43 técnicas diferentes ao longo de 20 anos de pesquisa. Dieste, O., Juristo, N., and Shull, F. "Understanding the Customer: What Do We Know about Requirements Elicitation?" IEEE Software, vol. 25, no. 2, pp. 11-13, March/April 2008.

Modelos de Maturidade/Capacidade em Processos de Software: 61 estudos; 52 modelos de processo. von Wangenheim, C., Hauck, J., Zoucas, A., Salviano, C., McCaffery, F., and Shull, F. "Creating Software Process Capability / Maturity Models," IEEE Software, vol. 27, no. 4, pp. 92-94, July / August 2010.

Desenvolvimento Distribuído de Software: "Poucos dos modelos existentes foram avaliados..." Prikladnicki, R., Audy, J. L. N., and Shull, F. "Patterns in Effective Distributed Software Development," IEEE Software, vol. 27, no. 2, pp. 12-15, March / April 2010.

Técnicas de Testes para Linha de Produto de Software: 60% dos artigos descrevem "soluções ou propostas conceituais," muito pouco relatam experiências em ambientes reais de desenvolvimento. da Mota Silveira Neto, P.A.; Runeson, P.; do Carmo Machado, I.; de Almeida, E.S.; de Lemos Meira, S.R.; Engstrom, E.; "Testing Software Product Lines," Software, IEEE, vol.28, no.5, pp.16-20, Sept.-Oct. 2011.

Ou por Yvan Labiche (Keynote no CBSOFT, 2017)

Técnicas de Projeto de Teste: "Infelizmente, ainda existe, depois de décadas de pesquisa no campo, raramente se encontram dados sólidos nos quais o engenheiro possa confiar para tomar uma decisão adequada" Labiche, Y. How much do we know about the testing techniques we use?. Keynote at SAST 2017.

Insuficiência de evidência sobre tecnologias...

E também observado em algumas de nossas pesquisas:

Modelos de Estimativa de Custo: 11 estudos (incluindo 2 repetições) usando diferentes conjuntos de dados. Não há evidência sobre a viabilidade dos modelos existentes nem possibilidade de agregação dos estudos. Kitchenham, B. ; [Mendes, E.](#) ; Travassos, G. H. (2007). Cross versus within-company cost estimation studies: A systematic review. IEEE Transactions on Software Engineering, v. 33, p. 316-329, 2007. <http://dx.doi.org/10.1109/TSE.2007.1001>

Testes Baseado em Modelos: de 85 artigos selecionados (representando 71 abordagens), 27% são especulativos, 45% apresentam apenas exemplos simples, 15% mostram provas de conceito, 5% relata alguma experiência e apenas 8% foram experimentadas. [Dias Neto, A. C.](#) ; Subramanyan, R. ; Vieira, M. E. R. ; Travassos, G. H. ; Shull, F. (2008) Improving evidence about software technologies: A look at model-based testing. IEEE Software, v. 25, p. 10-13, 2008. <http://dx.doi.org/10.1109/MS.2008.64>

Criterio de Parada de Testes: 74 critérios (3 repetidos) resultando em 108 variações. A maioria utiliza o conceito de confiabilidade. Outros são específicos. Apenas 27% foram avaliados, sem evidência sobre sua viabilidade (sem indicação de contexto). Ribeiro, V. V. ; Travassos, G. H. . Critérios para Apoiar a Decisão sobre o Momento de Parada dos Testes de Software. In: WAMPS 2013 - IX Workshop Anual do MPS, 2013, Curitiba. Anais do IX Wamps 2013. Curitiba: SOFTEX, 2013. v. 1. p. 108-119.

Insuficiência de evidência sobre tecnologias...

E também observado em algumas de nossas pesquisas:

Características de Agilidade e Práticas Ágeis: As características mais relevantes para introduzir agilidade em processos de software se relacionam com comunicação, compreensão e adaptação (não tem nada a ver com métodos ágeis!) . As praticas ágeis *Cliente Presente e Planing Poker* não são relevantes. Entretanto, *Integração Continua e Backlog* são relevantes. De Mello, R.M.; Silva, P.C.; Travassos, G.H. (2014). Agility in Software Processes: Evidence on Agility Characteristics and Agile Practices. SBQS 2014.

Estimativa de Esforço do Teste de Software: não existe consenso sobre o que é teste de software e o que deve ser considerado esforço. Por isso, os modelos disponíveis e seus fatores não são genericamente adequados e usar um ou outro é arriscado. Souza, T.S.; Ribeiro, V. V.; Travassos, G.H. (2014). Software Testing Estimation Effort: Models, Factors and Uncertainties. CACIC 2014

Teste de Sistemas Sensíveis ao Contexto: Nenhuma tecnologia disponível permite testar uma sistema deste tipo levando em consideração a variação do contexto durante a execução do teste. Matalonga, S., Rodrigues, F., Travassos, G.H. (2017). Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review. Journal of Systems and Software 131: 1-21

Realidade dos Engenheiros de Software...

Sem evidencia, o uso de tecnologias pode ser difícil e arriscado, ainda mais considerando a baixa interação entre a academia e a indústria e a pressão do Mercado por soluções mágicas e rápidas (“balas de prata”)

Alta demanda por soluções rápidas

+

Necessidade de Justificar Investimentos

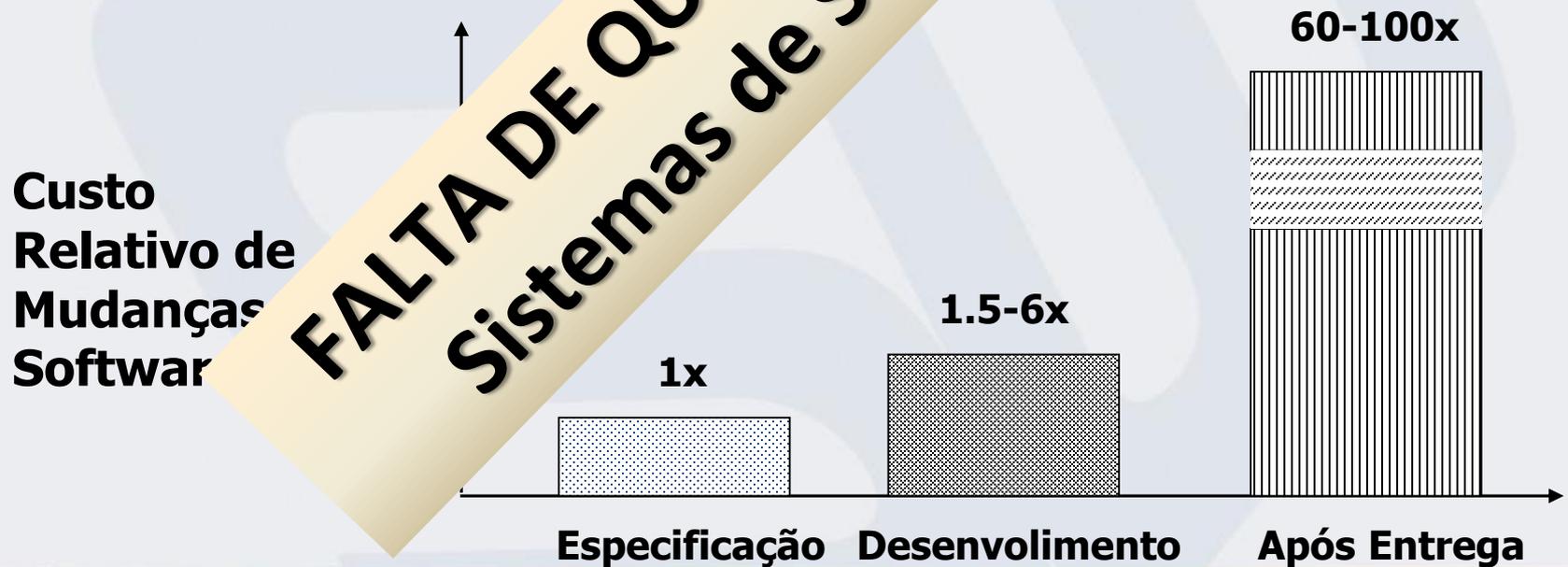
=

Tecnologia Imatura transferida para a Indústria

Realidade dos Engenheiros de Software...

Precisamos entregar sistemas de software dignos, com alta confiabilidade e qualidade

Entretanto, nossa habilidade de dar suporte aos software existentes é reduzida devido projetos mal feitos e falta de recursos



Realidade dos Engenheiros de Software...

Falta de qualidade dos sistemas de software, devido a ...

DEFEITOS

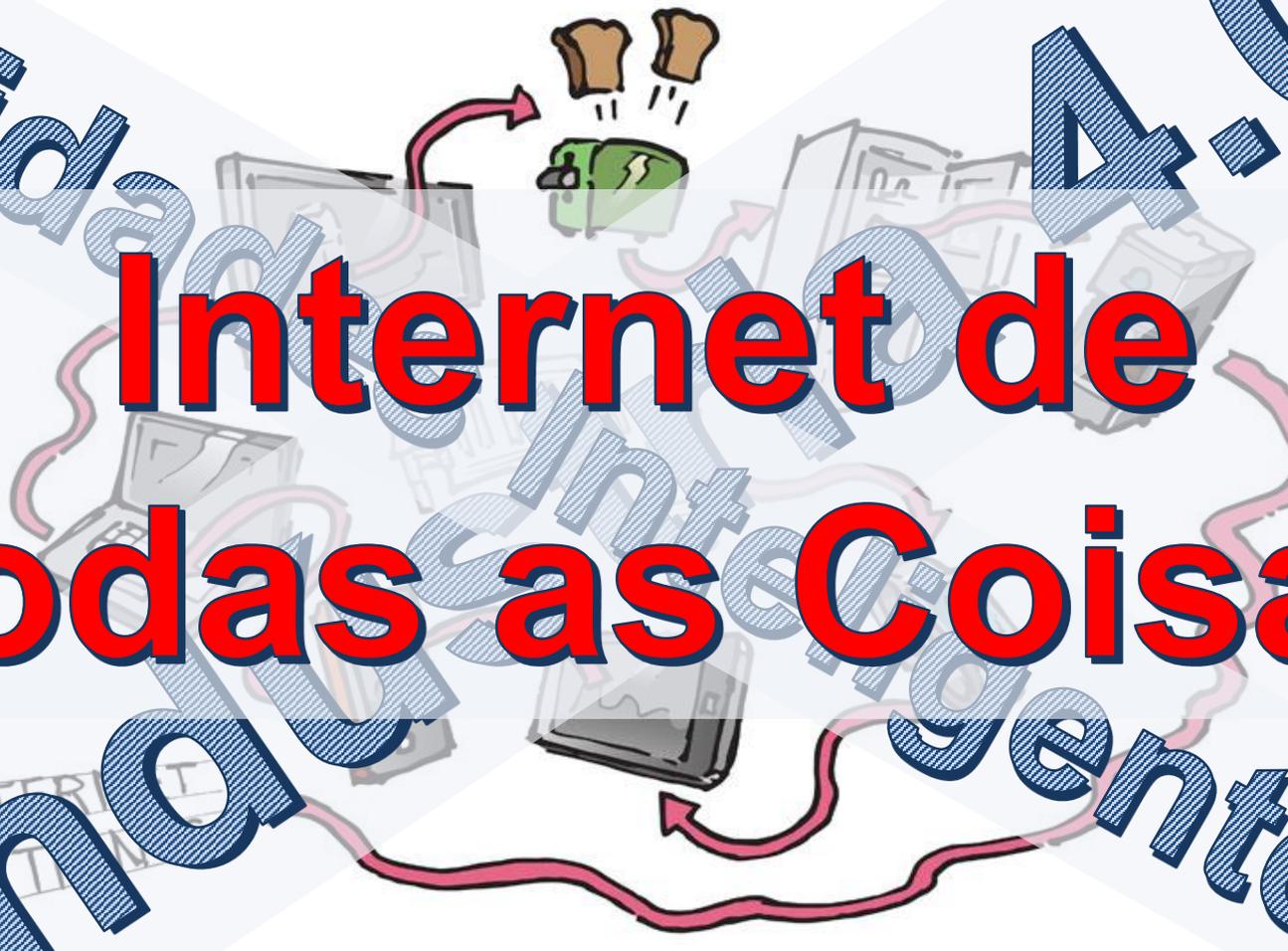
Qualidade Externa



**DÍVIDA
TÉCNICA**

Qualidade Interna

Sistemas de Software



Cidade Inteligente 4.0

Internet de Todas as Coisas

Indústria Inteligentes

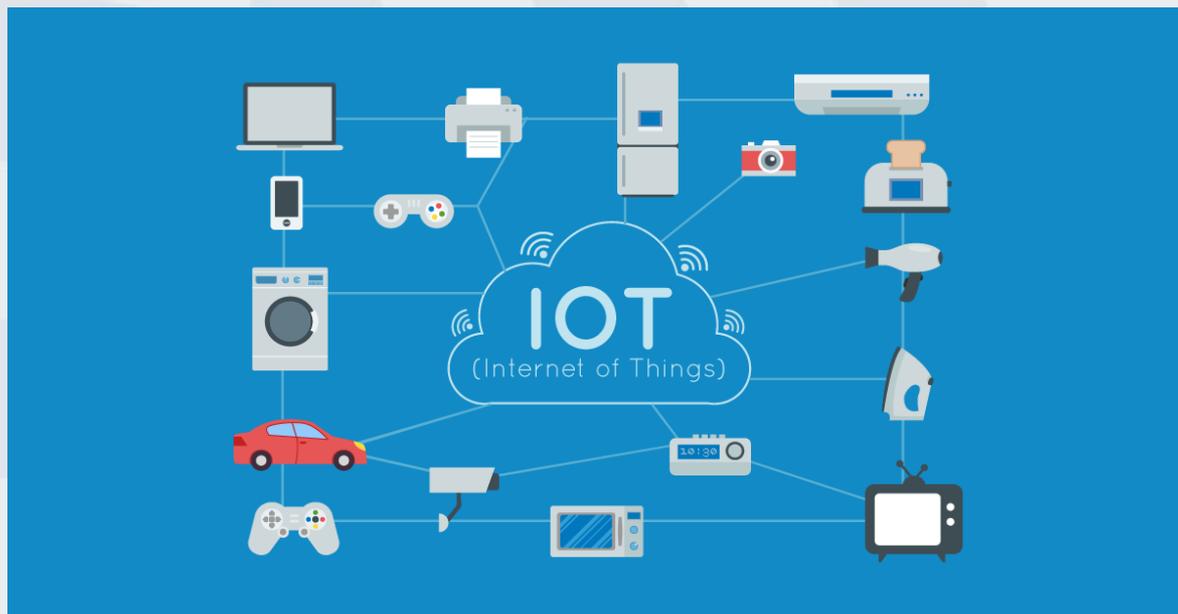
Sistemas Ubíquos

Internet das Coisas (IoT)

Paradigma que permite compor sistemas a partir de objetos (*things*) com endereços únicos, equipados com comportamentos de identificação, sensoriamento ou atuação, e recursos de processamento, podendo se comunicar e cooperar para atingir um objetivo.



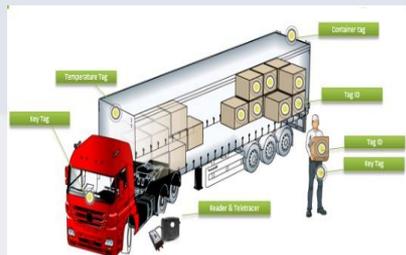
https://pbs.twimg.com/profile_images/37880000790349866/127857d65c47795d45f832f5dab323cb_400x400.jpeg



Internet das Coisas (IoT)

[Em IoT] “coisas” existem no meio físico, como sensores, atuadores e também qualquer coisa que esteja equipada com capacidades de **Identificação (leitura de tags)**, **Sensoriamento** ou **Atuação**, o que exclui entidades no domínio da Internet (hosts, terminais, roteadores, entre outros). As coisas também devem ter funcionalidades de comunicação, rede e processamento variando de acordo com os requisitos do sistema.

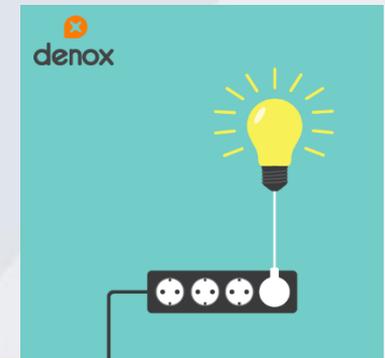
Identificação



Sensoriamento



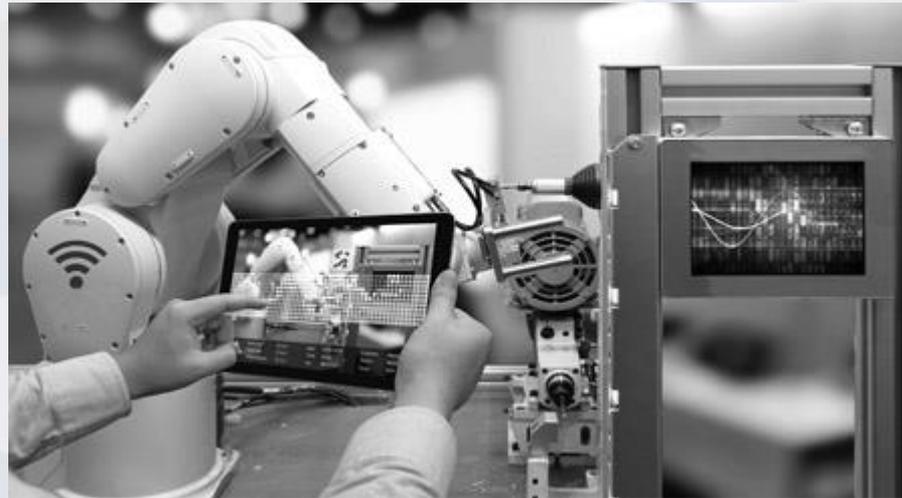
Atuação



Indústria 4.0

Um desafio para a Indústria Brasileira

Envolve sistemas ciber-físicos, a aplicação de IoT e processos de manufatura descentralizados



<https://www.bndes.gov.br/wps/portal/site/home/conhecimento/noticias/noticia/manufatura-avancada>

<http://www.industria40.gov.br/>

A Engenharia de Sistemas de Software Contemporâneos



Não existe bala de prata!!

Não existe pedra filosofal!!



A construção de Sistemas de Software Contemporâneos depende de ENGENHARIA!

Software Engineering is big science!

Alguns Desafios da Engenharia de Sistemas de Software Contemporâneos

- Como lidar com o alto grau de distribuição e heterogeneidade dos sistemas de software modernos?
- Como escalar os sistemas de software para suportar todas as coisas (usuários, objetos, infra, etc.)?
- Quem é o cliente? Como garantir requisitos?
- Como permitir um alto grau de privacidade nos sistemas?
- Como construir software compostos por todas as coisas ?
- Como garantir que equipes de pessoas desenvolvam diferentes sistemas que precisam interoperar de maneira complexa?
- Como garantir a qualidade destes sistemas?

Precisamos estar preparados para superar estes desafios!



ese.cos.ufrj.br

- **ESE (Engenharia de Software Experimental)**

- uma das áreas de pesquisa da linha de Engenharia de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ.
- Seu objetivo é evoluir o conhecimento em Engenharia de Software a partir da aplicação de abordagem científica (experimentação) na construção de novos métodos e técnicas para apoio ao desenvolvimento de software.
 - Além disso, preocupa-se também com o avanço da área, estudando e pesquisando novos modelos para o planejamento, execução e empacotamento de estudos relacionados à Engenharia de Software.
- Estas atividades são fundamentais para que a Engenharia de Software incorpore cada vez mais os princípios da Engenharia.

Interesses de Pesquisa e Desenvolvimento do Grupo ESE

- **Engenharia de Software Experimental**
 - Apoio Metodológico para Condução de Estudos em Engenharia de Software e Engenharia
 - Ambientes para Engenharia de Software e Experimentação
 - **Experimentação aplicada à Engenharia de Software**
 - Engenharia de Software Baseada em Buscas
 - Verificação, Validação e Testes de Software
 - Qualidade de Software e Experimentação Contínua
 - **Tecnologias de Software para Sistemas de Software Contemporâneos**
 - **Processos de Software para Inovação**
-
- ```
graph TD; S[Science] <--> E[Engineering]; ESE[Experimentação aplicada à Engenharia de Software] --> S; ESE --> E; ESE --> T[Software];
```

# DELFOFOS – Observatório da Engenharia do Software Contemporâneo



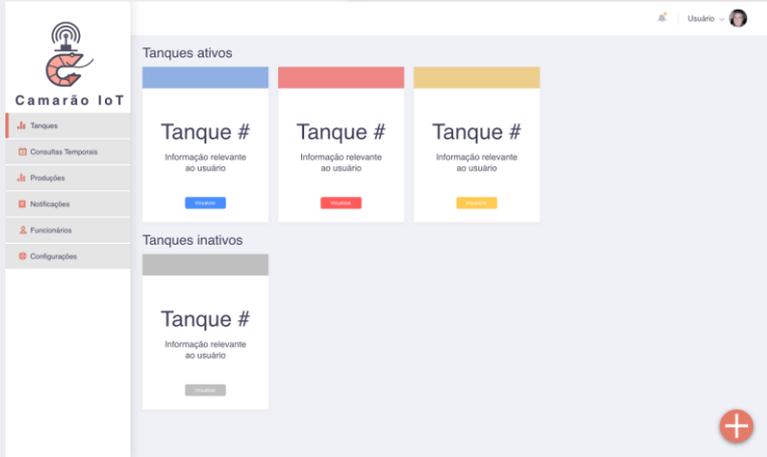
[www.delfos.cos.ufrj.br](http://www.delfos.cos.ufrj.br)

Espaço voltado à pesquisa, ao desenvolvimento e à inovação na construção de sistemas de software com qualidade e que atendam às demandas contemporâneas no paradigma da Internet de Todas as Coisas (Internet das Coisas, Indústria 4.0, Cidades Inteligentes, Sistemas Sensíveis ao Contexto, Sistemas de Sistemas, dentre outros).

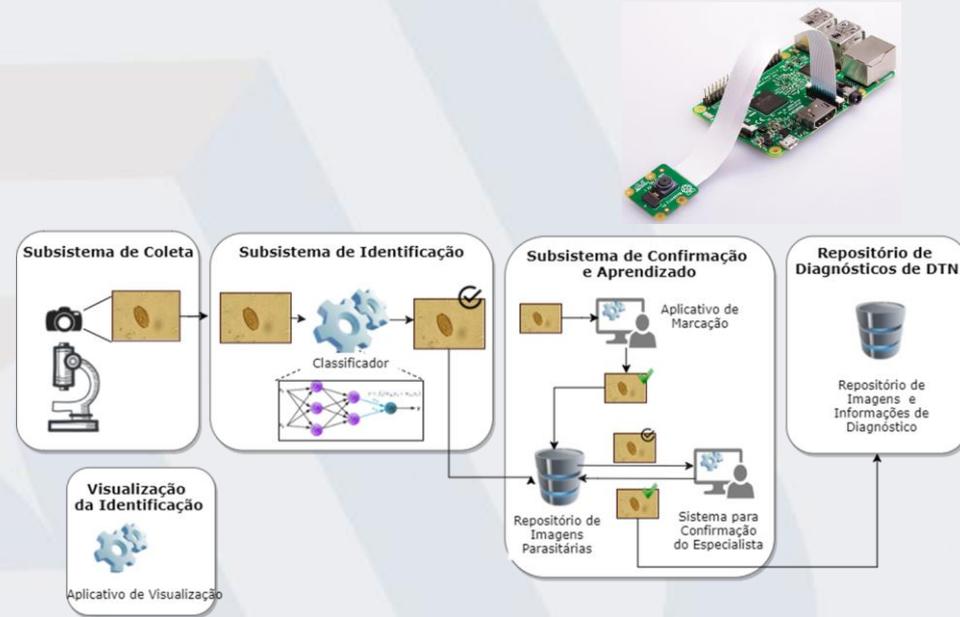
Para alavancar as oportunidades criadas em suas diferentes frentes de atuação, um ambiente propício à inovação e alavancagem de [pré-startups](#) complementa a família **Delfos**.

# DELFO – Observatório da Engenharia do Software Contemporâneo

Alguns dos projetos DELFOS:



Camarão lotizado



Parasite Watch



Qualidade de Produto de Software

Modelo de Referência para Avaliação de Produtos de Software

# Engenharia de Sistemas de Software Contemporâneos

Obrigado por sua atenção.

**Guilherme Horta Travassos**

Universidade Federal do Rio de Janeiro



COPPE/PESC

Pesquisador **CNPq**, Membro da **ISERN**

[ght@cos.ufrj.br](mailto:ght@cos.ufrj.br)

[www.cos.ufrj.br/~ght](http://www.cos.ufrj.br/~ght)