

# DOCUMENTAÇÃO DE *SOFTWARE* PMR 2550

DEPARTAMENTO DE ENGENHARIA MECATRÔNICA

Arturo Forner-Cordero

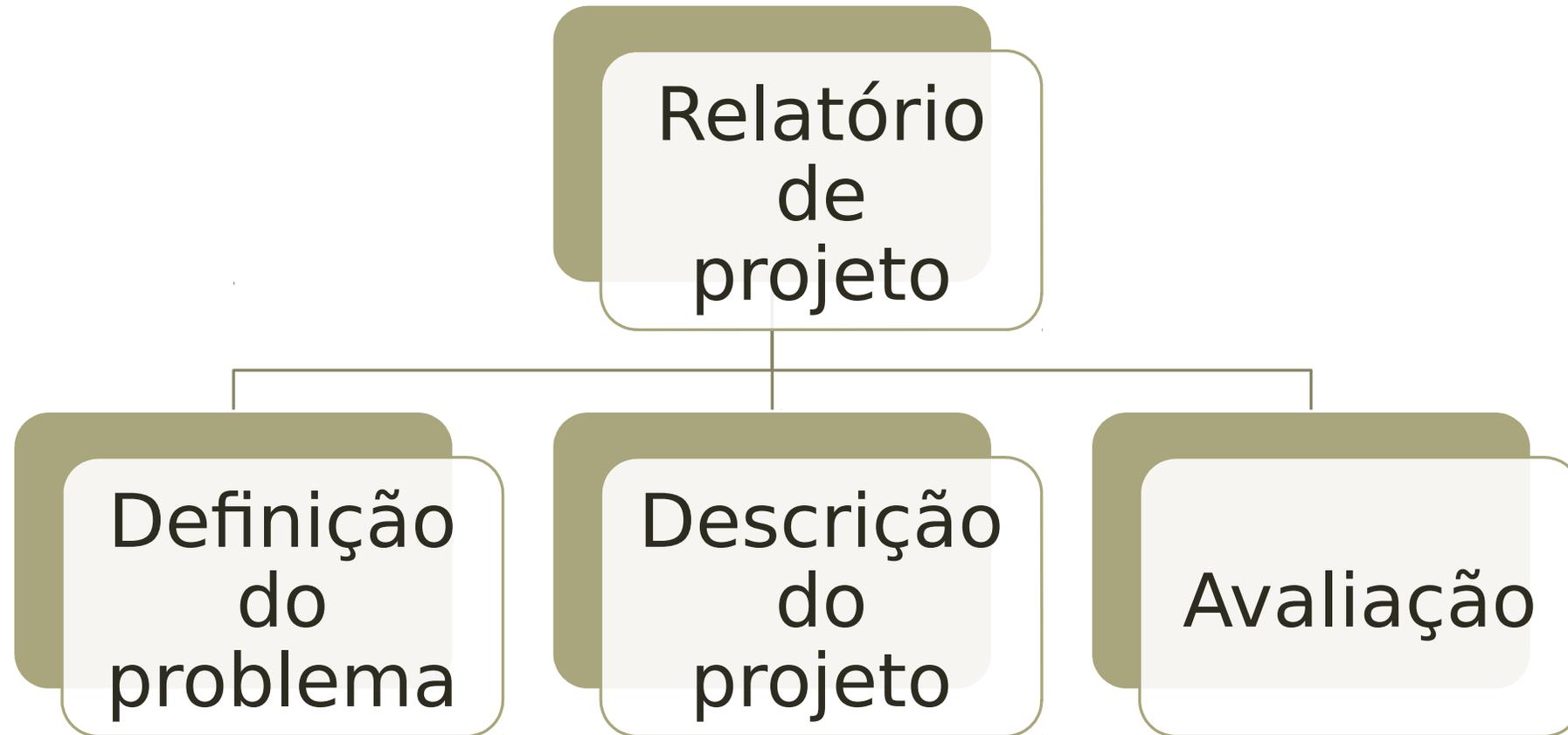
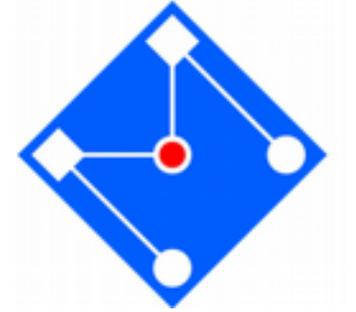
[aforner@usp.br]

Larissa Driemeier

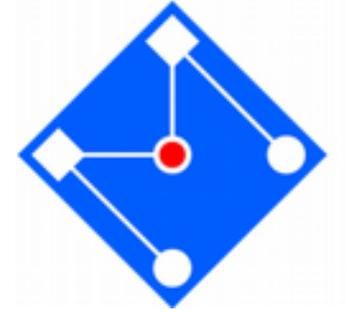
[driemeie@usp.br]

Thiago Martins [thiago@usp.br]

# DOCUMENTAÇÃO

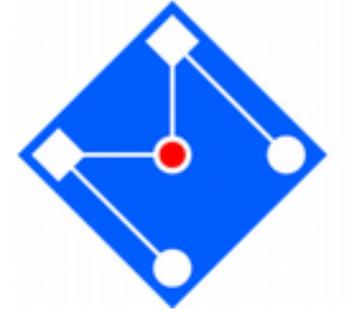


# DOCUMENTAÇÃO DE SOFTWARE



1. Documentação de Especificações (!)
2. Documentação de Arquitetura
3. Documentação Técnica (?)
4. Documentação para Usuário Final (?)
5. Documentação de Testes

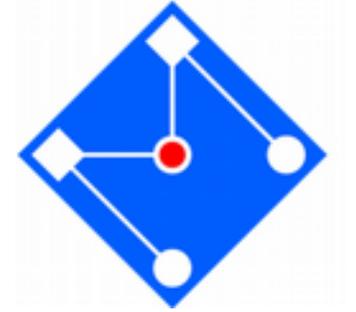
# DOCUMENTAÇÃO DE SOFTWARE



A documentação de software está fortemente ligada a *narrativas*.

- Casos de Uso
- Máquinas de estado
- Sequências.

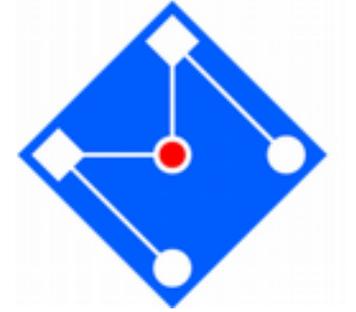
# DOCUMENTAÇÃO DE SOFTWARE



A documentação de software está fortemente ligada a *predicados lógicos* (invariantes).

- Pré-condições e pós-condições
- Entidades e relações
- Leis de recorrência

# DOCUMENTAÇÃO DE SOFTWARE



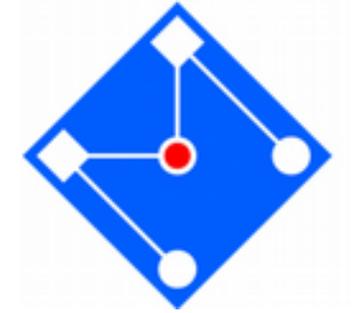
O escopo do software (e naturalmente da sua documentação) é *vasto*.

Apesar disso, há padrões (sim, no plural...) da indústria de formato de documentação.

- É importante ter consciência das capacidades e limitações destes padrões diante da abrangência do escopo de software.

“UML” → Unificada? CERTAMENTE NÃO!

# ESPECIFICAÇÕES

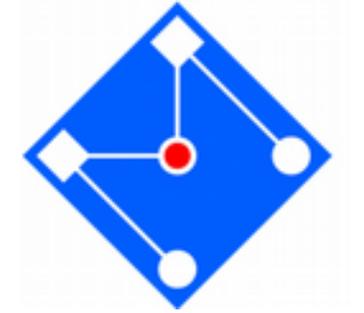


Requisitos:

- Definir *partes interessadas* (“*stakeholders*”).
- Requisitos Funcionais
- Requisitos Não-Funcionais
- Requisitos de negócio
- etc.

*Comumente* expressos em linguagem natural. Há no entanto métodos *formais* de análise de requisitos.

# ESPECIFICAÇÕES



## Requisitos\*:

- Corretos
- Inambíguos
- Completos
- Consistentes
- Hierarquizados
- Verificáveis
- Modificáveis
- Rastreáveis

\*(IEEE Recommended Practice for Software Requirements Specifications - 1998)

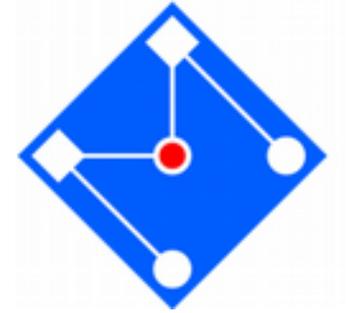
# ESPECIFICAÇÕES

## Casos de Uso

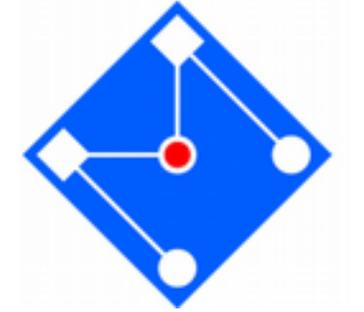
Relata interações entre *atores* e o sistema.

Forma tradicional:

- Atores
- Pré-Condições
- Pós-Condições
- “Gatilho”
- Fluxo principal



# ESPECIFICAÇÕES



## Casos de Uso

“Histórias de usuário” ← *Agile development*

Como um *comprador*, eu quero *saber quais os produtos disponíveis*.

Como um *comprador*, eu quero *pedir um produto*.

Como um *vendedor*, eu quero *saber quais os meus pedidos*.

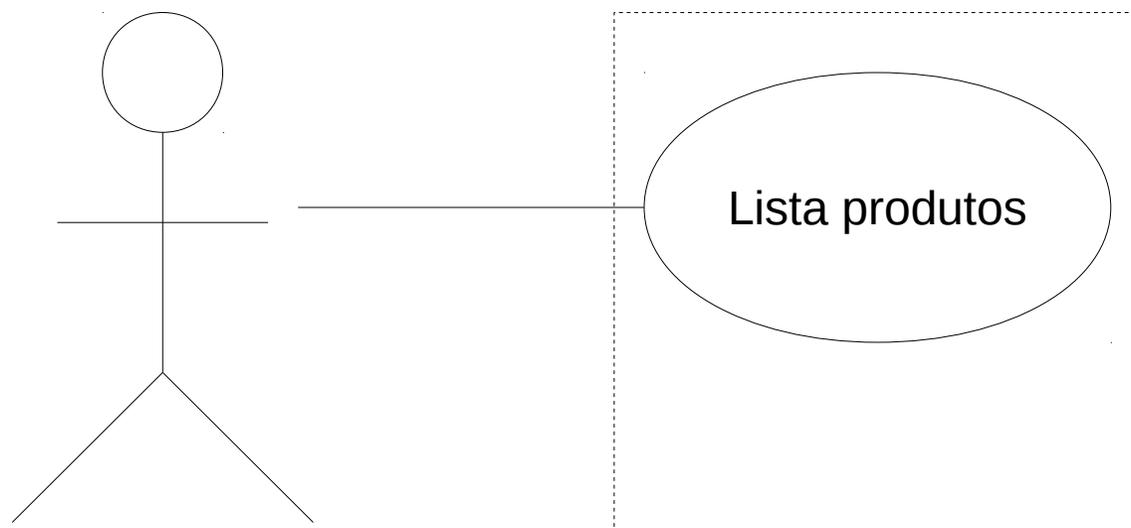


# ESPECIFICAÇÕES

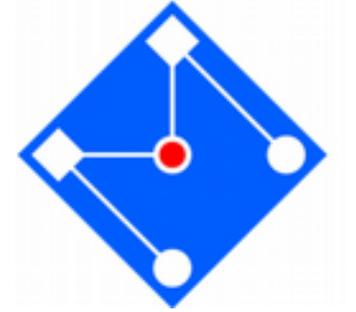
Casos de Uso

UML ← Diagramas de caso de uso

Historicamente, superutilizados por alunos de graduação



# ARQUITETURA

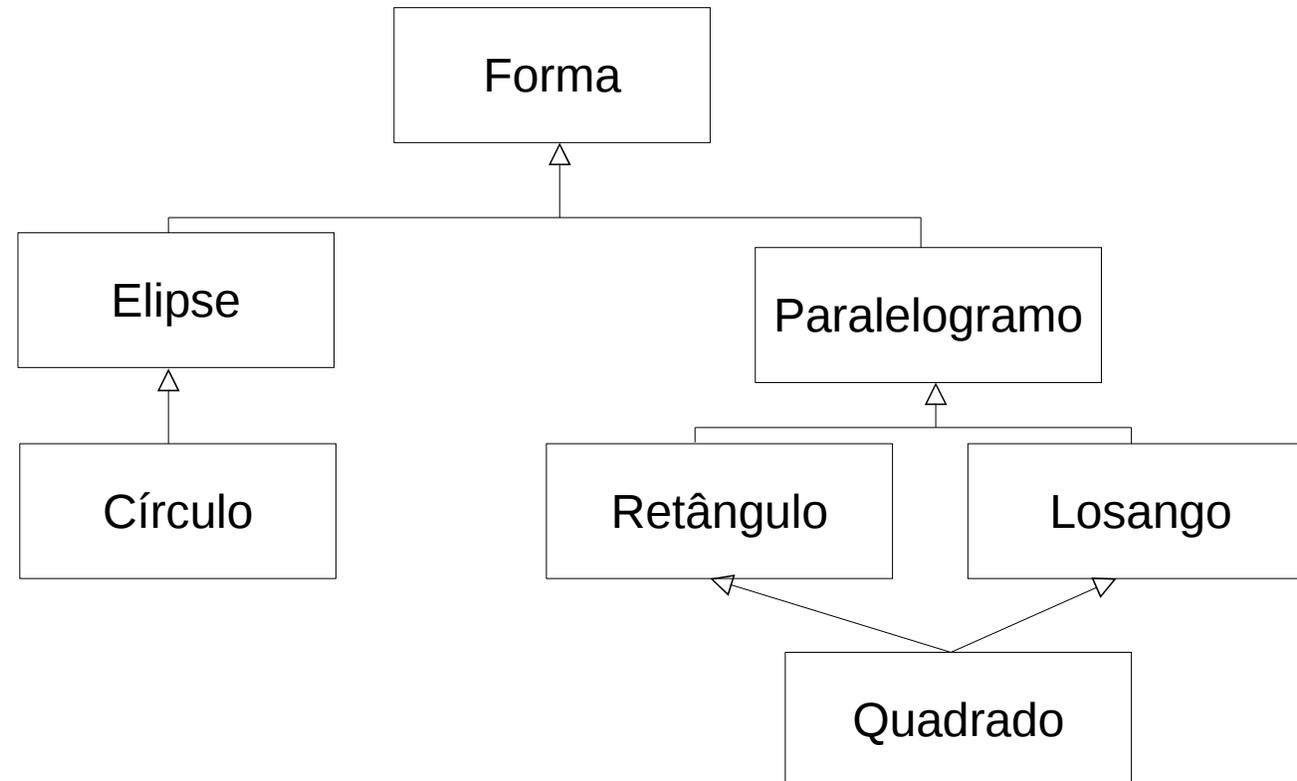
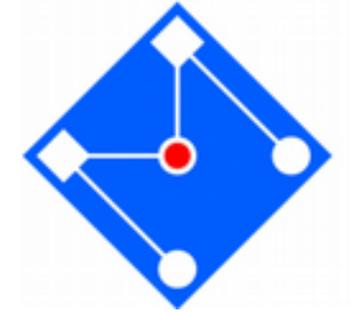


- Estrutura
- Comportamento
- Interação



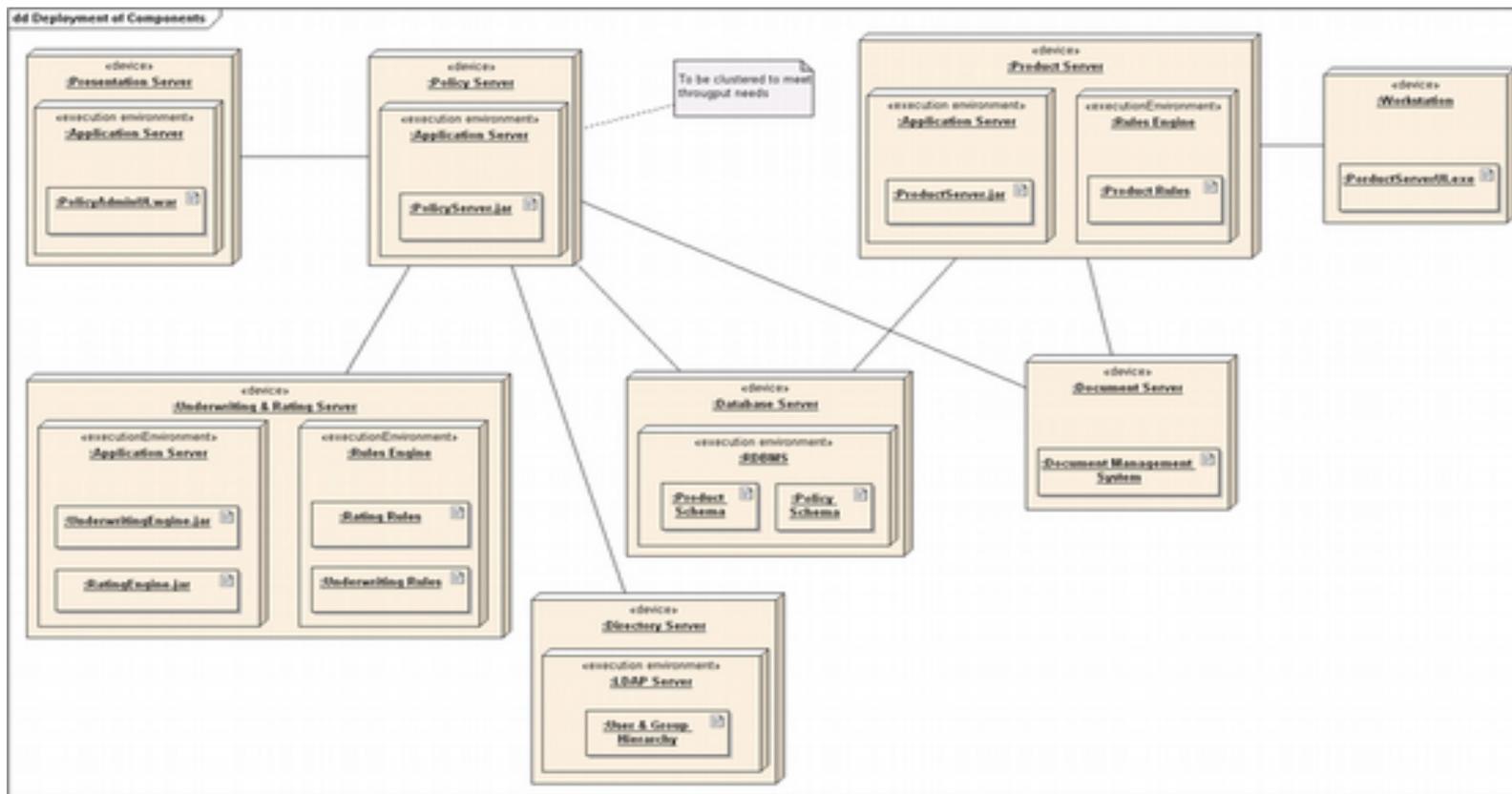
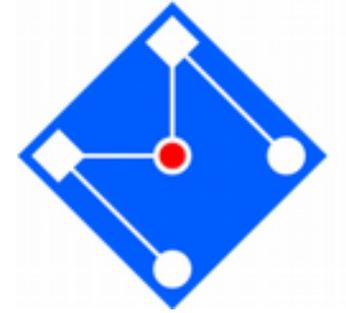
# ARQUITETURA

Estrutura: Diagrama de Classes (OO)

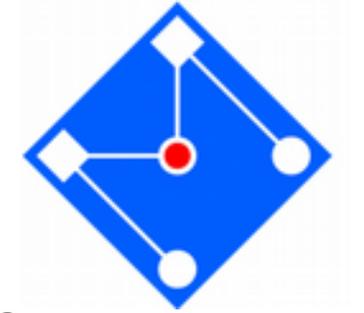


# ARQUITETURA

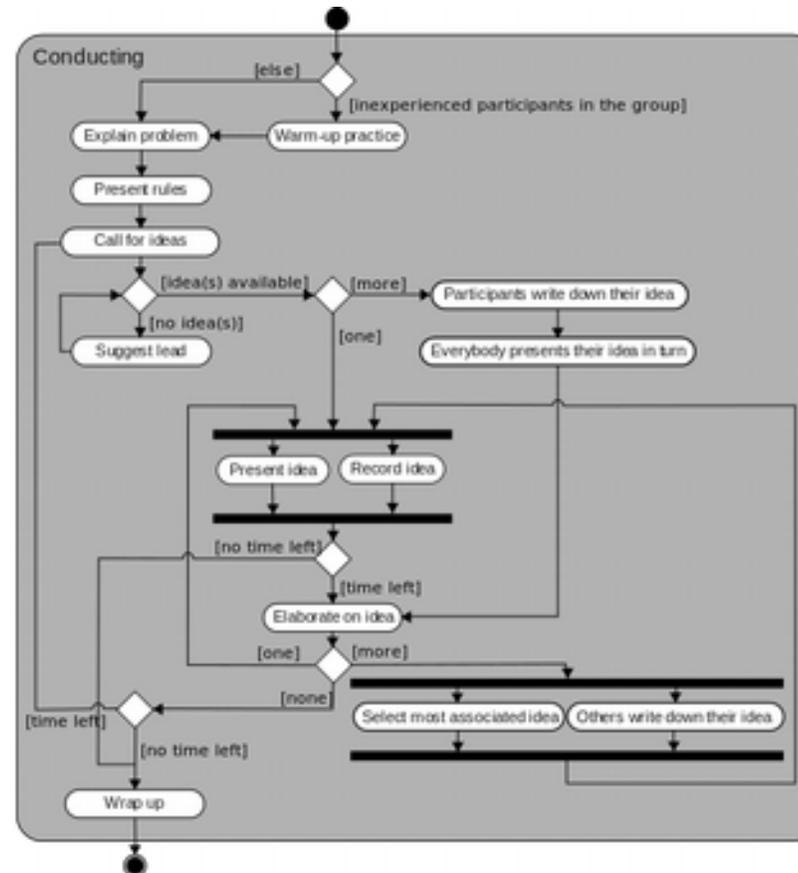
## Estrutura: Distribuição



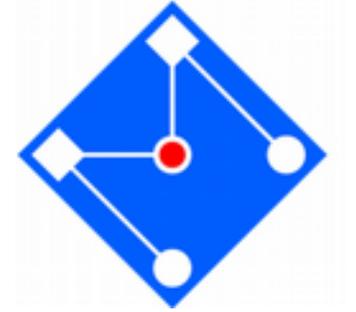
# ARQUITETURA



Comportamento: Fluxogramas, diagramas de atividade



# ARQUITETURA



Comportamento: Algoritmos – Leis de recorrência

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \cdot \mathbf{x}_0$$

$$\mathbf{p}_0 = \mathbf{r}_0$$

$$\gamma_0 = \mathbf{r}_0^T \cdot \mathbf{r}_0 / \mathbf{r}_0^T \cdot \mathbf{A} \cdot \mathbf{r}_0$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \gamma_{k-1} \mathbf{p}_{k-1}$$

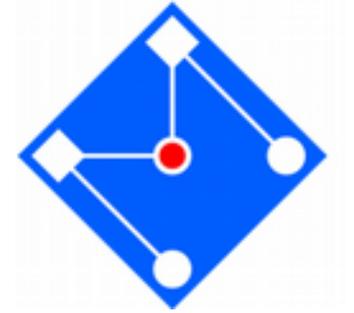
$$\mathbf{r}_k = \mathbf{r}_{k-1} - \gamma_{k-1} \mathbf{A} \cdot \mathbf{p}_{k-1}$$

$$\beta_k = \mathbf{r}_k^T \cdot \mathbf{r}_k / \mathbf{r}_{k-1}^T \cdot \mathbf{r}_{k-1}$$

$$\mathbf{p}_k = \mathbf{r}_k + \beta_k \mathbf{p}_{k-1}$$

$$\gamma_k = \mathbf{r}_k^T \cdot \mathbf{r}_k / \mathbf{p}_k^T \cdot \mathbf{A} \cdot \mathbf{p}_k$$

# ARQUITETURA



## Comportamento: Algoritmos – Pseudo código

---

**Algorithm 1** Solve  $\mathbf{A} \cdot \mathbf{x} - \mathbf{b} = 0$

---

**Input:**  $\mathbf{A}$  a  $n \times n$  symmetric positive definite matrix,  $\mathbf{b}$  and  $\mathbf{x}_0$

**Output:**  $\mathbf{x}$  such that  $\mathbf{A} \cdot \mathbf{x} - \mathbf{b} = 0$

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \cdot \mathbf{x}_0$$

$$\mathbf{p}_0 = \mathbf{r}_0$$

$$\gamma_0 = \mathbf{r}_0^T \cdot \mathbf{r}_0 / \mathbf{r}_0^T \cdot \mathbf{A} \cdot \mathbf{r}_0$$

**for**  $i = 1$  to  $n$  **do**

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \gamma_{i-1} \mathbf{p}_{i-1}$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} - \gamma_{i-1} \mathbf{A} \cdot \mathbf{p}_{i-1}$$

$$\beta_i = \mathbf{r}_i^T \cdot \mathbf{r}_i / \mathbf{r}_{i-1}^T \cdot \mathbf{r}_{i-1}$$

$$\mathbf{p}_i = \mathbf{r}_i + \beta_i \mathbf{p}_{i-1}$$

$$\gamma_i = \mathbf{r}_i^T \cdot \mathbf{r}_i / \mathbf{p}_i^T \cdot \mathbf{A} \cdot \mathbf{p}_i$$

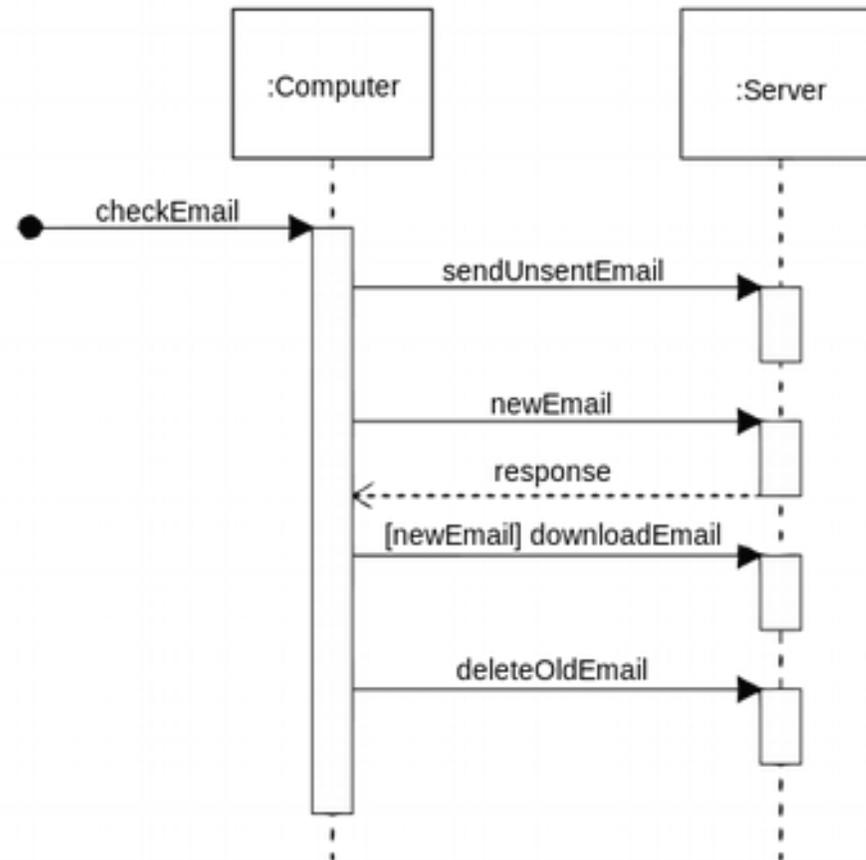
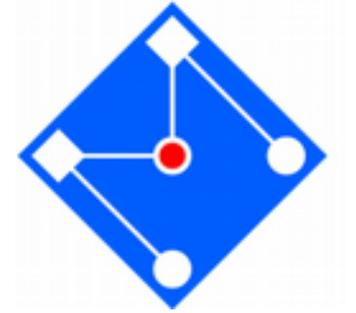
**end for**

**output**  $\mathbf{x}_n$

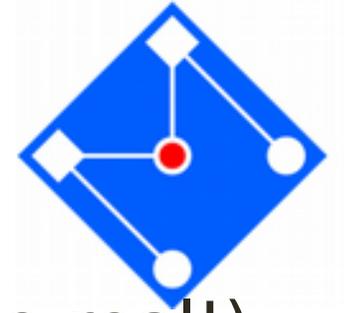
---

# ARQUITETURA

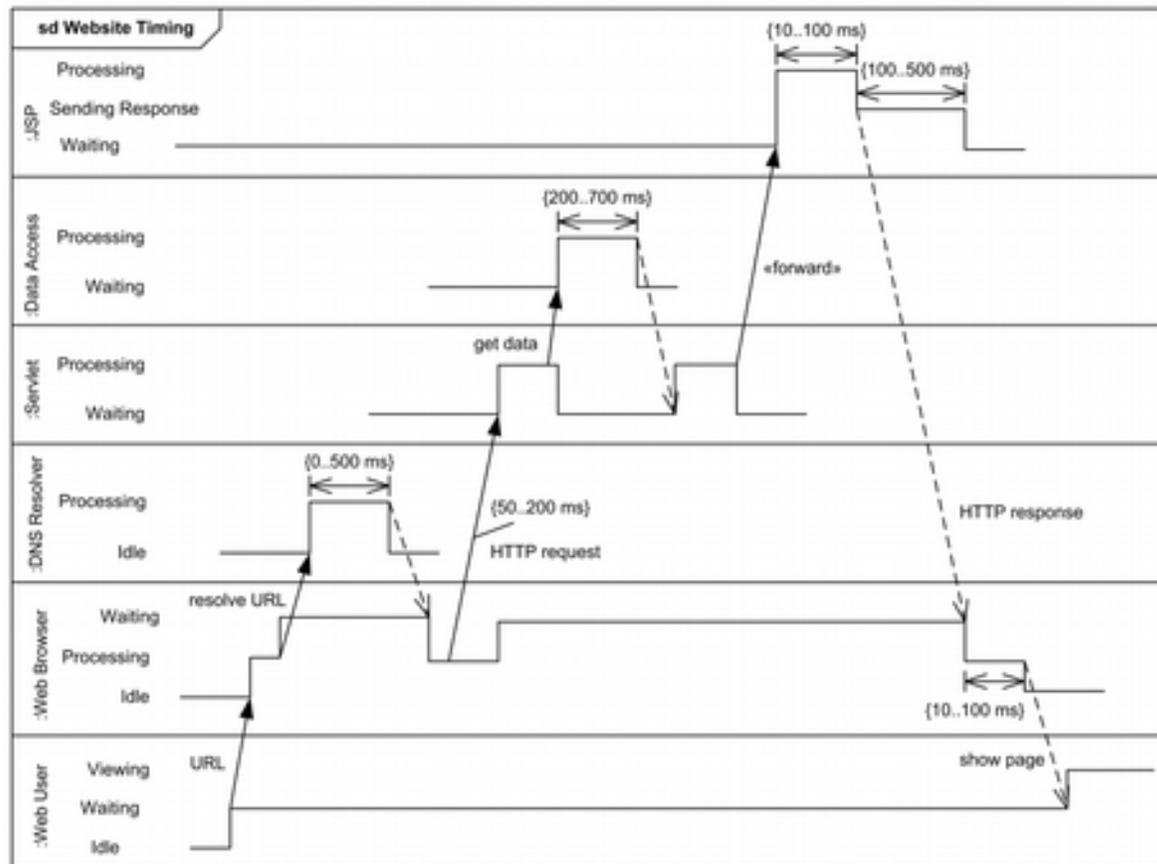
## Comportamento: Interação



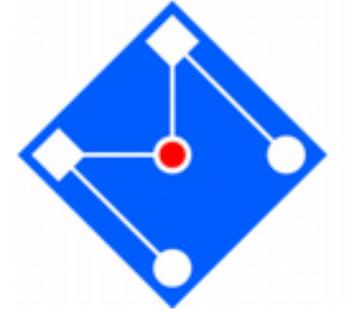
# ARQUITETURA



Comportamento: Temporização (*excelentes para tempo real!*)

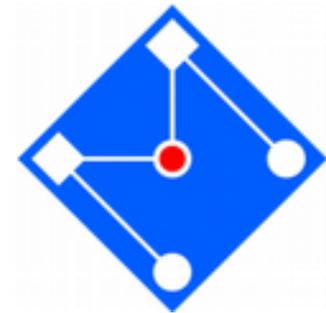


# DOCUMENTAÇÃO TÉCNICA



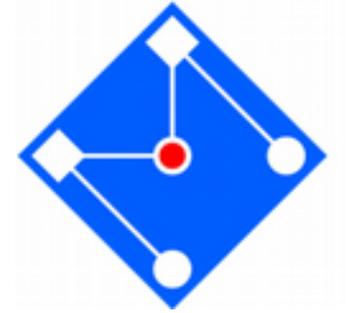
- Manual de código-fonte (cada vez mais raro)
  - Detalha distribuição de componentes pelo código-fonte
  - Auxilia na *manutenção* do código escrito

# DOCUMENTAÇÃO TÉCNICA



- Manual de *Application Programming Interface* (API)
  - Detalha interfaces dos componentes e como usá-las
  - Auxilia na geração de código novo que interage com o existente.

# DOCUMENTAÇÃO TÉCNICA



- Manual de Código fonte e API
  - Pode ser produzido a partir do código fonte e comentários!
  - doxygen e afins.