

*Universidade de Brasília*  
*Faculdade Gama*  
*Engenharia de Software*



**UnB**

Disciplina: **203291 – MED / Medição e Análise**

***CFP – Cosmic Function Points***

**Elaine Venson**

*elainevenson@unb.br*

# COSMIC

---

## O Método COSMIC de Tamanho Funcional

- Método padronizado para a medição do tamanho funcional de software
- Aplicável a sistemas de negócios e a software em tempo real
- Unidade de medida PFC – **P**ontos de **F**unção **C**OSMIC
- A definição do método consta do manual de medição COSMIC, versão atual = 4.01
- COSMIC - **C**ommon **S**oftware **M**easurement **I**nternational **C**onsortium
  - Grupo voluntário internacional de especialistas em medição de software
- O método COSMIC foi aceito pelo ISO/IEC JTC1 SC7 em dezembro de 2002 como o Padrão Internacional ISO/IEC 19761 'Engenharia de Software – COSMIC-FFP – Um método para a medição funcional de tamanho'

<http://www.cosmicon.com>

# COSMIC

---

- **Domínios de aplicação**

- Software aplicativo de negócio (apoio à administração do negócio: bancos, contabilidade, compras, distribuição, manufatura)
- Software de tempo-real (acompanhamento e controle de eventos no mundo real: softwares de telefonia, softwares embarcados, softwares contidos nos sistemas operacionais)
- Híbridos dos tipos anteriores (ex: sistemas em tempo real para reservas de companhias aéreas)

- **Não aplicável a**

- Software com uso de algoritmos matemáticos complexos como: sistemas especialistas, software de simulação, processamento contínuo de variáveis (ex: jogos, edição musical, etc)

- **Limitações de fatores que influem no tamanho funcional**

- Todo método de medição funcional funciona sobre a hipótese de um modelo simplificado de representação funcional que é reconhecido como razoável na média para o seu domínio de aplicabilidade
- Cuidados devem ser tomados quando é necessário medir e comparar pedaços muito pequenos de software, ou pequenas mudanças em um pedaço de software, onde a suposição de razoabilidade é quebrada. No caso do COSMIC, pequenos pedaços de software compreendem poucos movimentos de dados.

# COSMIC

---

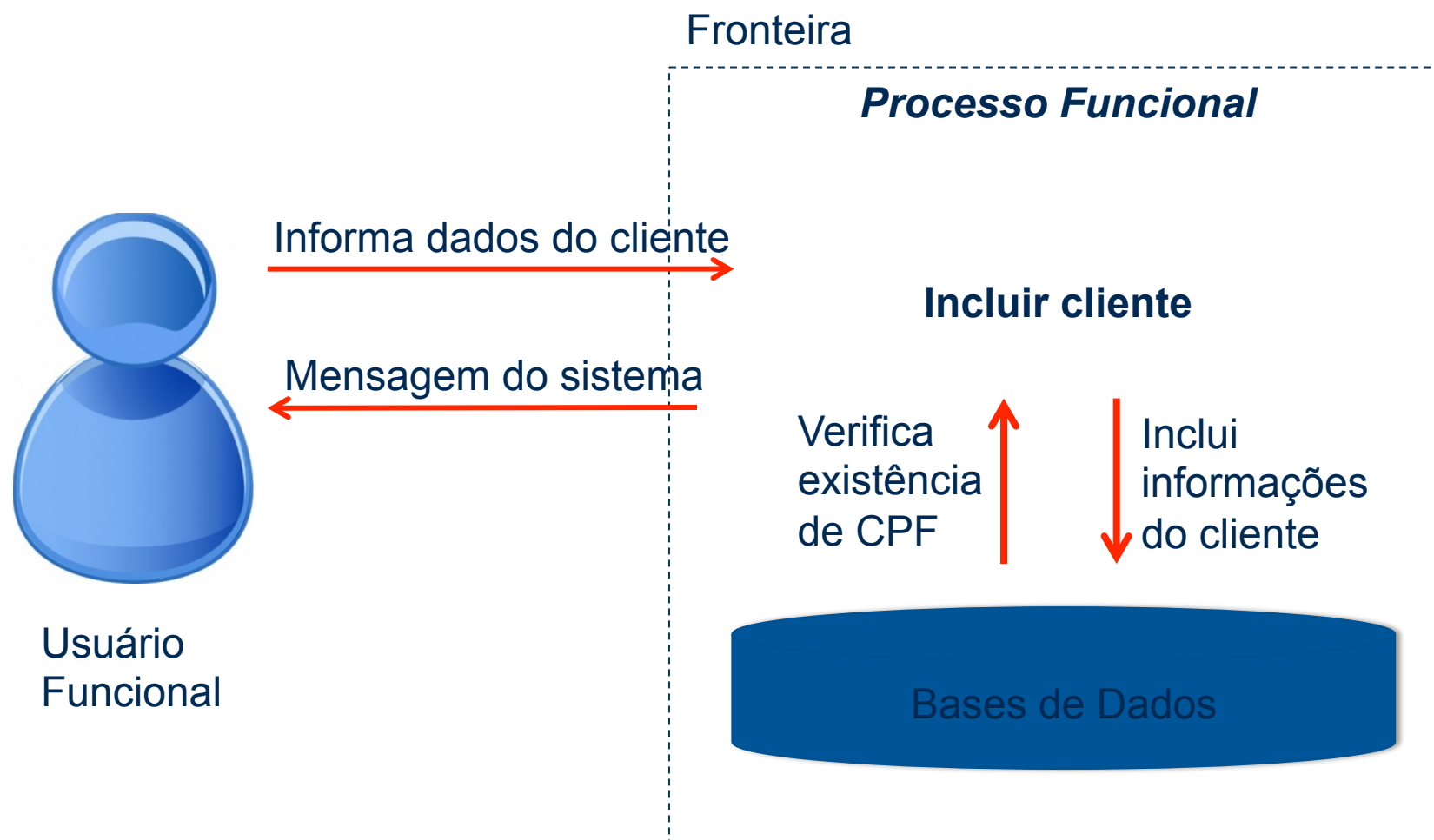
- **Método COSMIC de Tamanho Funcional**
  - Envolve a aplicação de um conjunto de **modelos**, **princípios**, **regras** e **processos** aos Requisitos Funcionais do Usuário (ou RFU) de um dado **pedaço de software**.
  - O resultado é um número, o valor de uma quantidade representando o tamanho funcional do pedaço de software de acordo com o método COSMIC em unidades de **Pontos de Função COSMIC** (ou PFC)
  - Foi projetado para ser **independente** de quaisquer decisões de implementação contidas nos artefatos operacionais do software a ser medido

# COSMIC

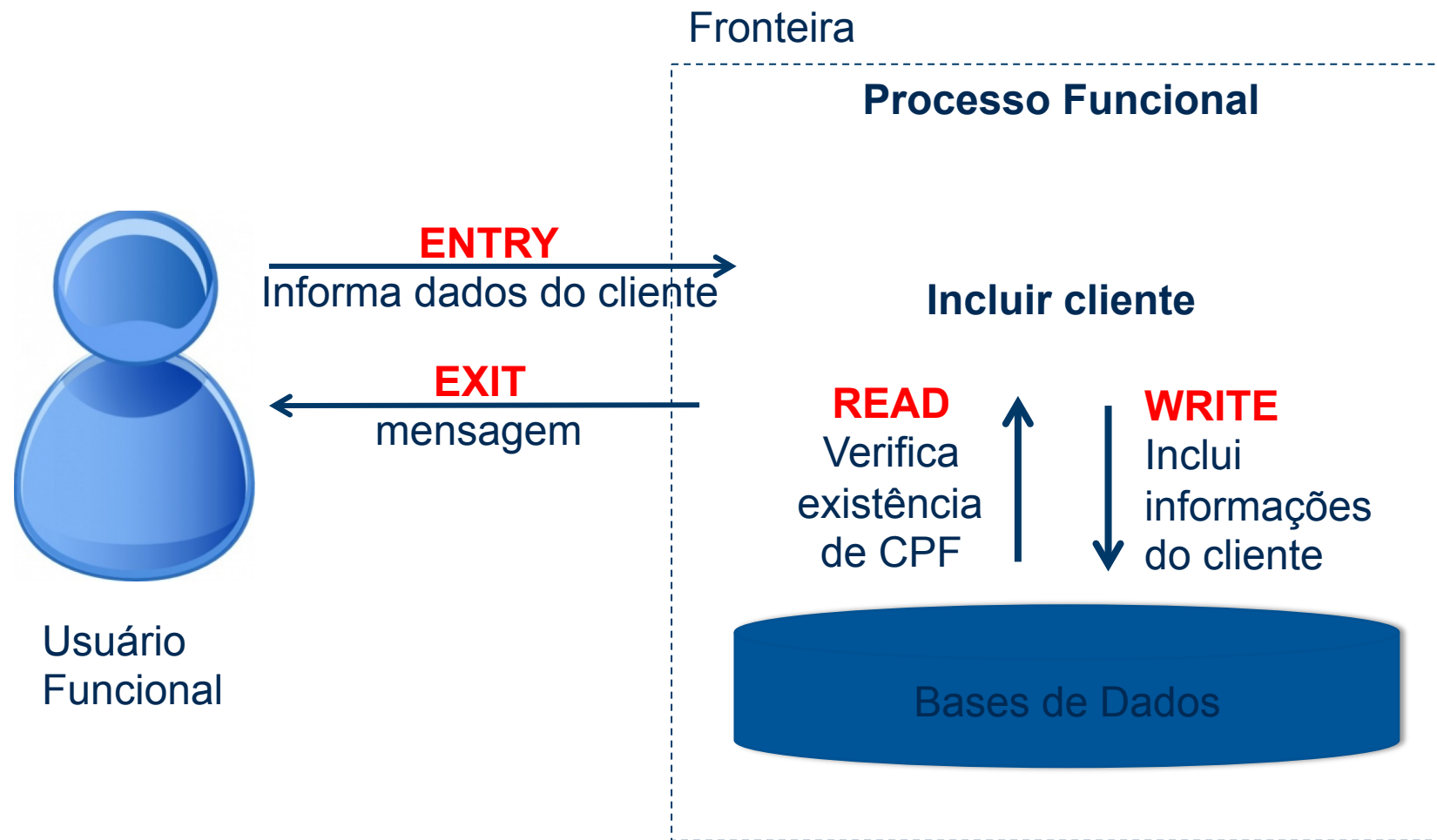
---

- O tamanho funcional produzido pelo método COSMIC é a soma das **movimentações de dados** dos **processos funcionais** do pedaço de software
- Cada uma das movimentações de dados é contada como 1 ponto de função COSMIC (PFC)

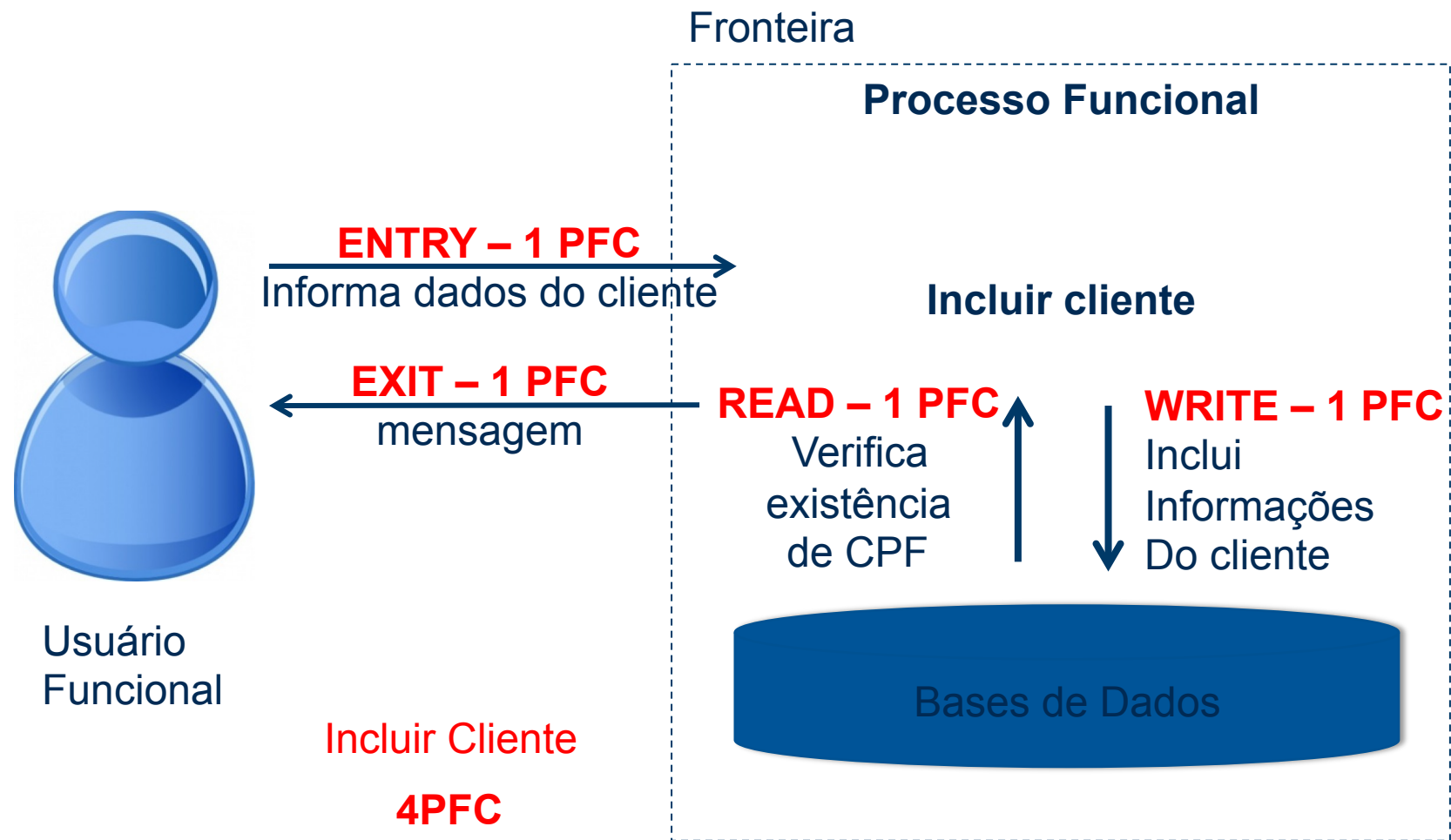
# Movimentações de Dados



# Tipos de Movimentos de Dados

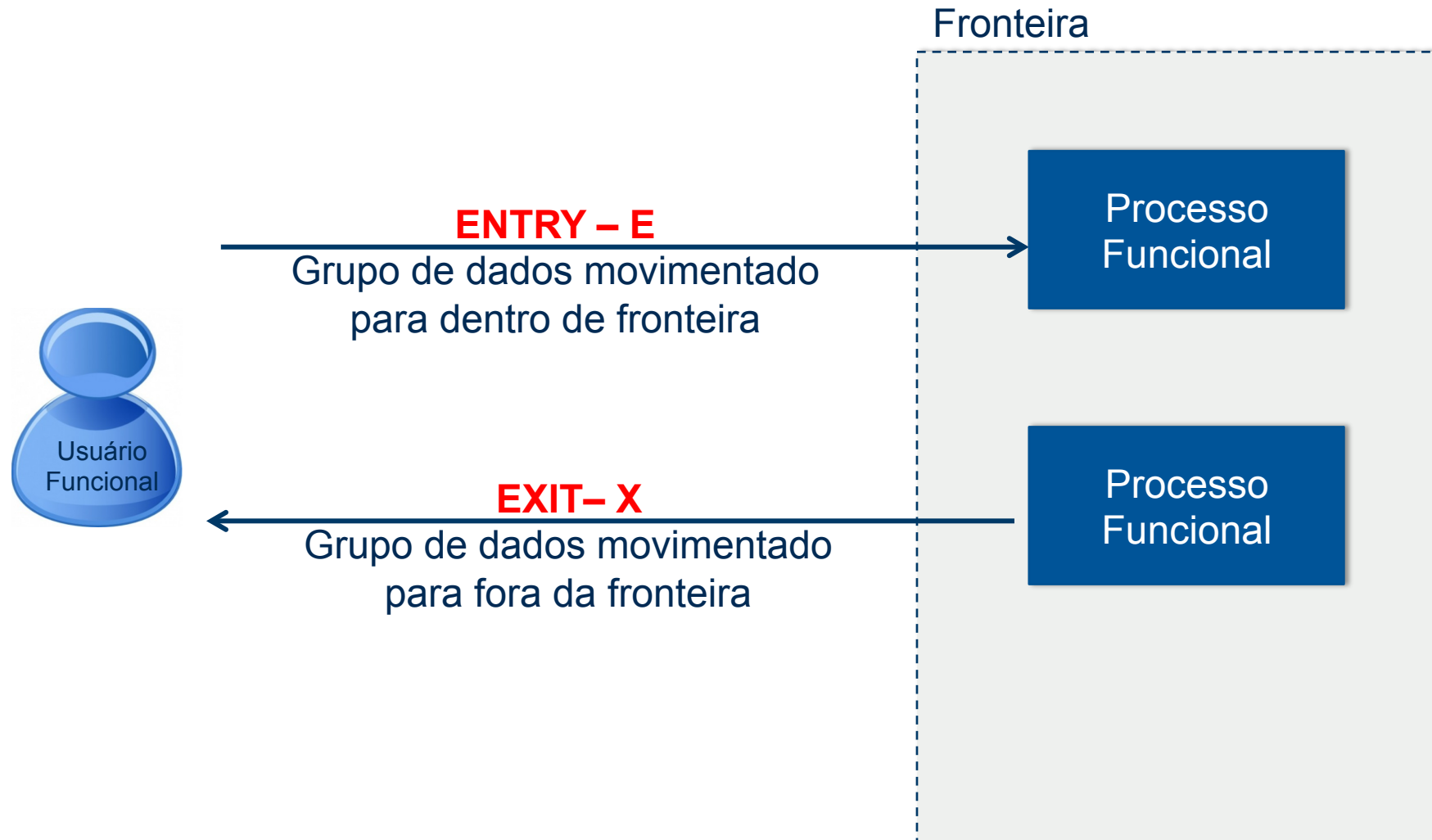


# Tipos de Movimentos de Dados



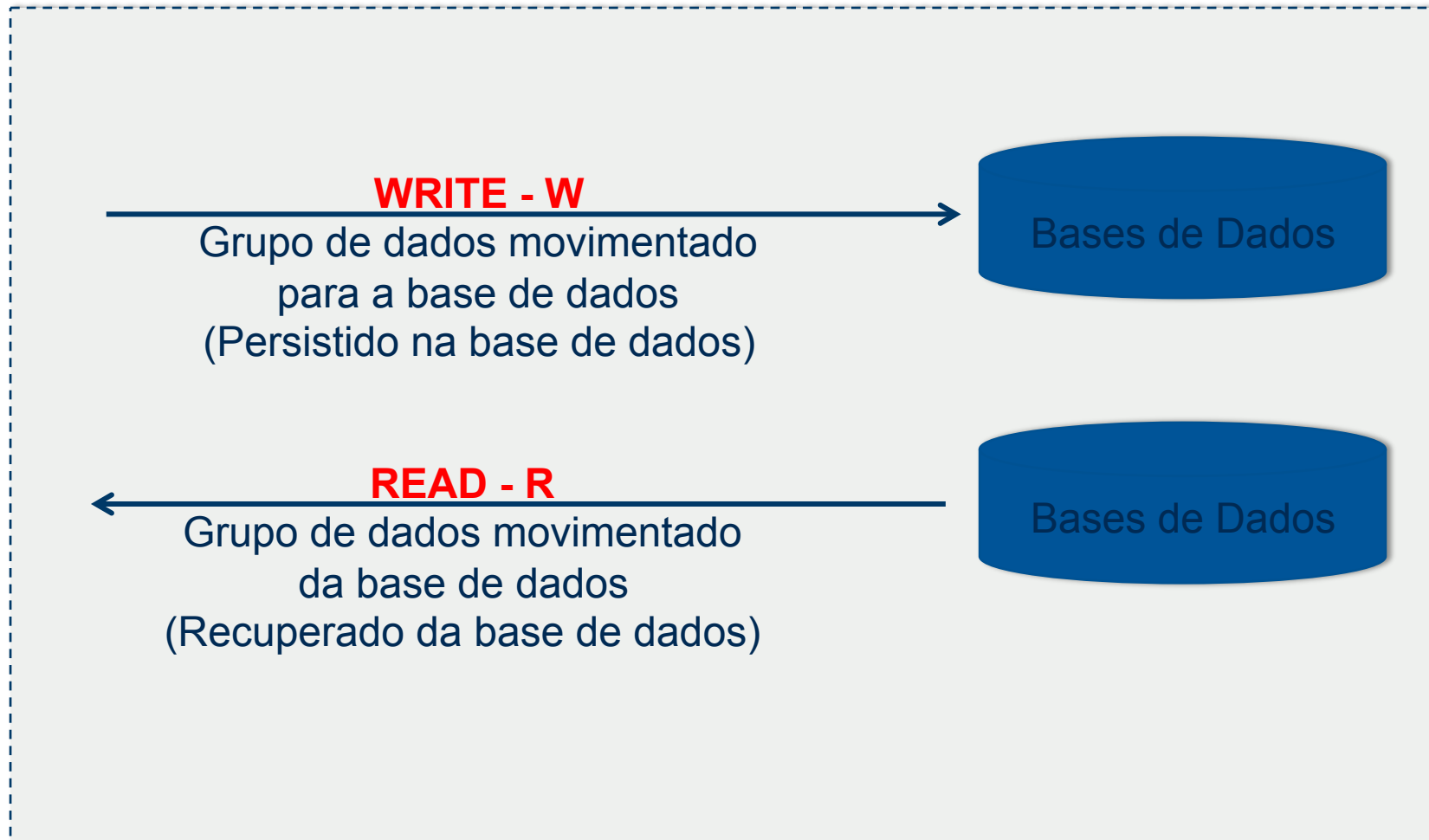


# Tipo de Movimento de Dados



# Tipo de Movimento de Dados

Fronteira



# Tipos de Usuário



# Usuários Funcionais



- **Definição:**
  - Um usuário funcional é uma **origem** e/ou um **destinatário** pretendido para os dados nos **Requisitos Funcionais do Usuário** de um pedaço de software
- **Regras:**
  - Os usuários funcionais de um pedaço de software a ser medido devem ser derivados do **propósito da medição**
  - Quando o propósito da medição de um pedaço de software estiver relacionado ao esforço para desenvolver ou modificar um **pedaço de software**, então os usuários funcionais deverão ser aqueles para quem a funcionalidade nova ou modificada deverá ser fornecida

# Fronteira



A **fronteira** é definida como uma interface conceitual entre o software sendo medido e os seus usuários funcionais.

# Fronteira

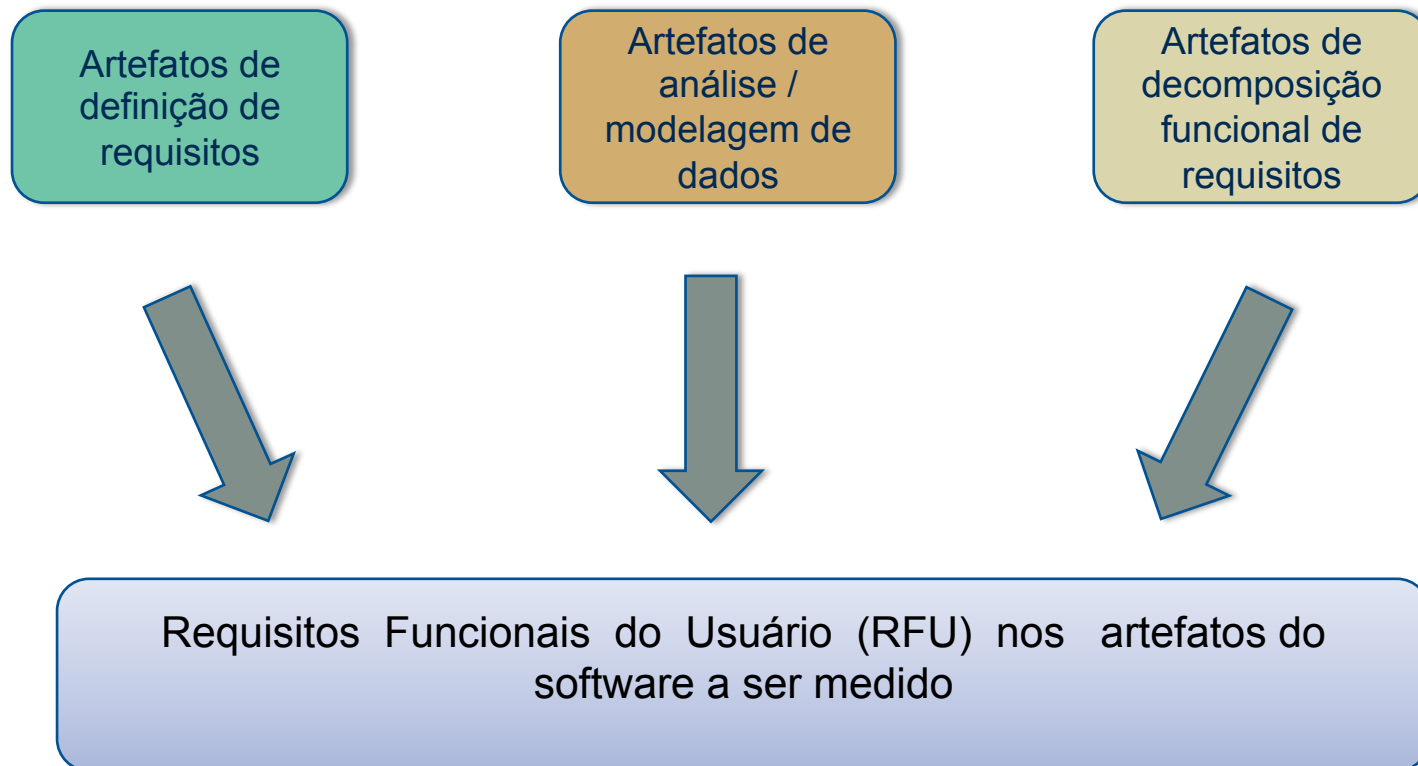


- A fronteira de um pedaço de software é o **limite conceitual** entre o referido pedaço e o ambiente no qual o mesmo opera, conforme percebido externamente segundo a perspectiva de seus usuários funcionais.
- A fronteira permite que o medidor distinga, sem ambiguidade, o que está incluído dentro do software medido daquilo que é parte do ambiente operacional do referido software.

# Requisitos Funcionais do Usuário - RFU

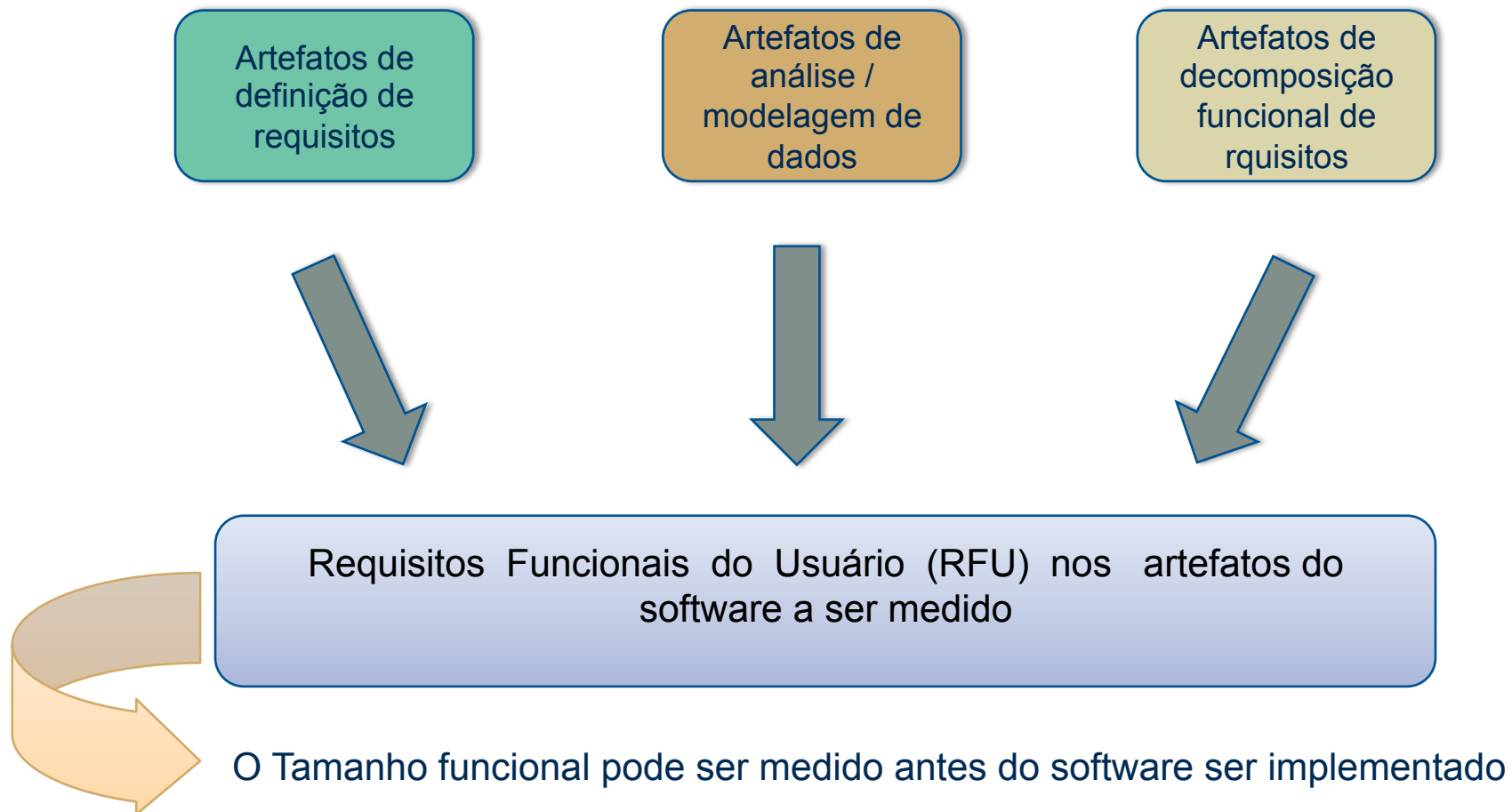
- O método de medição COSMIC consiste em aplicar um conjunto de modelos, princípios e processos aos **requisitos funcionais do usuário** (FUR) de um dado pedaço de software.
- O resultado é um valor quantitativo que representa o tamanho funcional do pedaço de software sendo medido de acordo com o método COSMIC.
- O método foi idealizado para ser independente de qualquer decisão de implementação incluída nos artefatos do software a ser mensurado.
- Funcionalidade deve ser entendida como sendo a informação processada que o software deve prover para seus usuários.
- Mais precisamente, uma expressão dos RFU descrevem **o que o software deve fazer para o usuário funcional** que, por sua vez, são a fonte e o destino pretendido para os dados tratados pelo software

# Requisitos Funcionais do Usuário - RFU

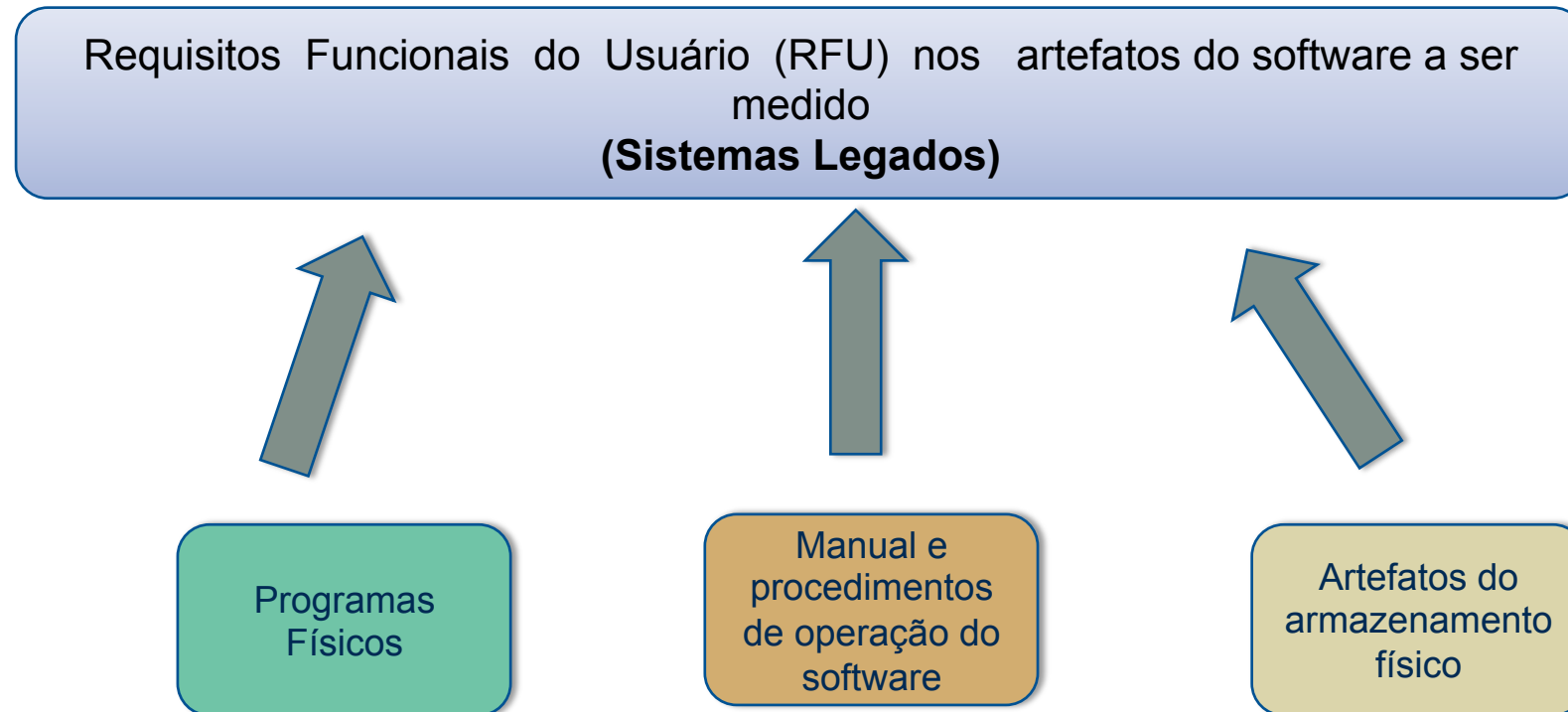




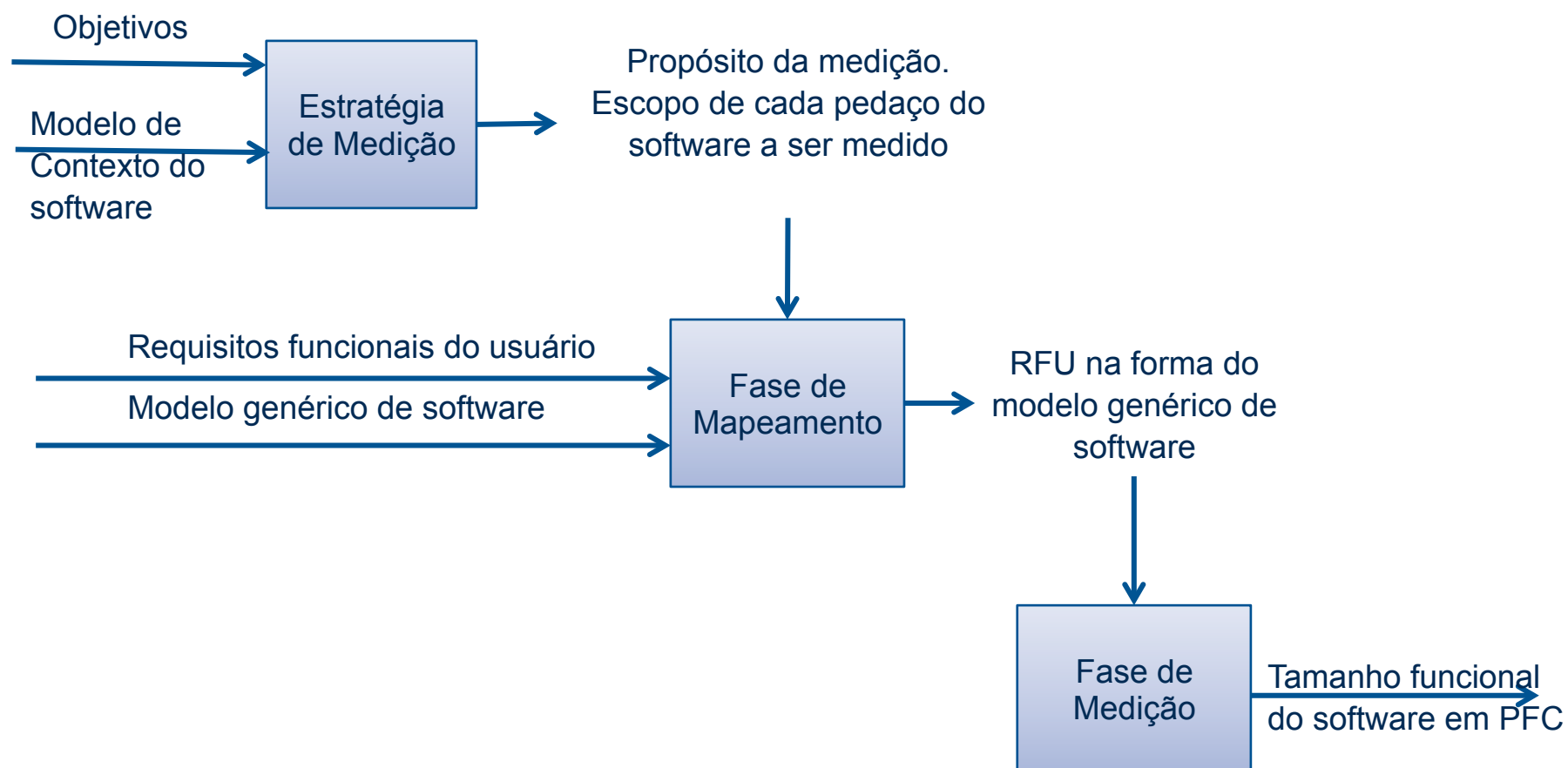
# Requisitos Funcionais do Usuário - RFU



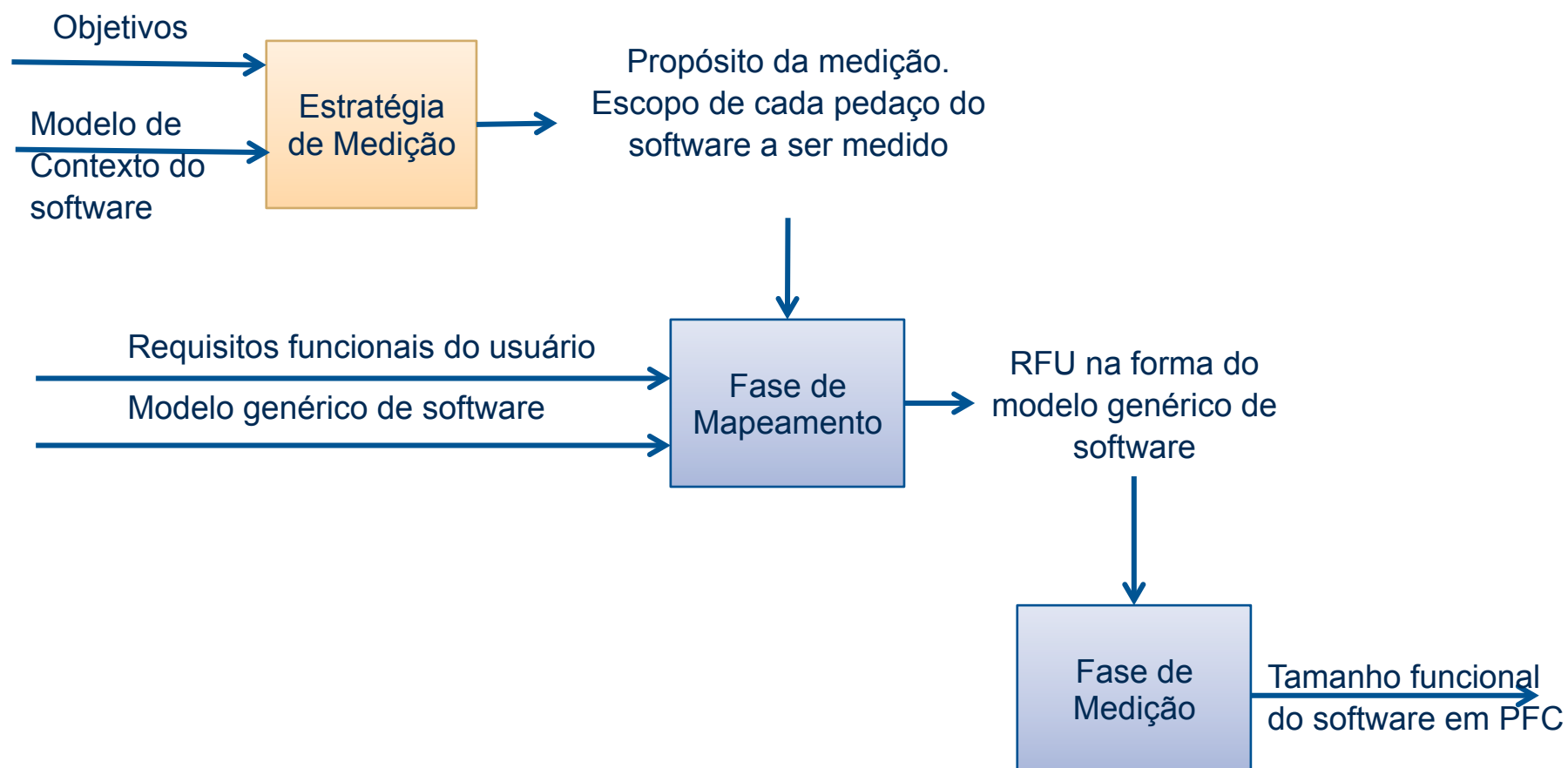
# Requisitos Funcionais do Usuário - RFU



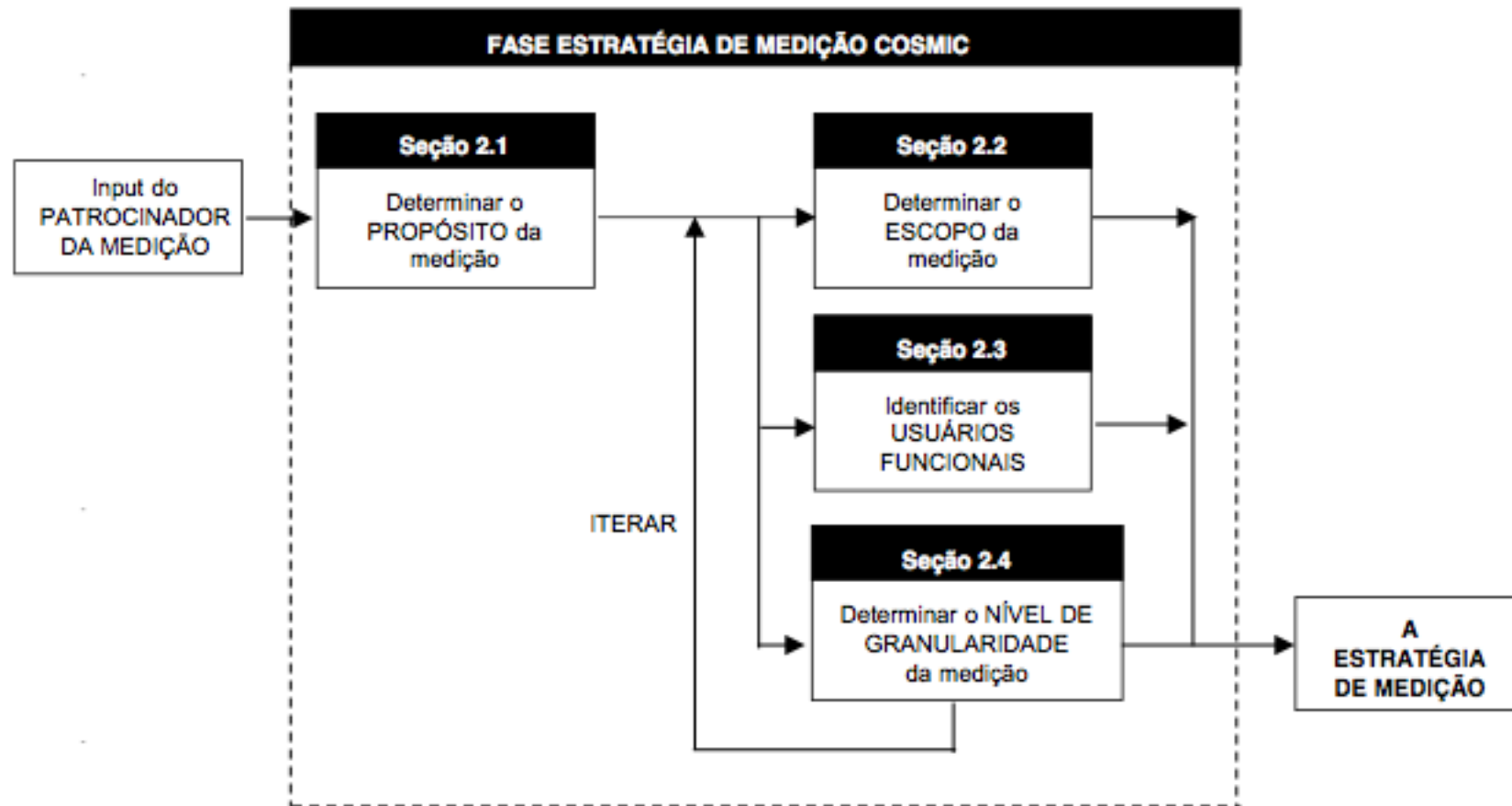
# Fases do processo de medição COSMIC



# Fases do processo de medição COSMIC



# Estabelecendo uma estratégia de medição



# Modelo de Contexto de Software COSMIC

- Um **pedaço de software** a ser medido com o método COSMIC deve ser cuidadosamente **definido** (no escopo da medição)
- Tal definição deve levar em conta, em seu contexto, qualquer outro **software** e/ou **hardware** com o qual o mesmo **interaja**
- Este *Modelo de Contexto de Software* apresenta os **princípios** e **conceitos** necessários a esta definição

# Modelo de Contexto de Software COSMIC

## PRINCÍPIOS:

- a) O Software é delimitado pelo hardware
- b) O software é normalmente estruturado em **camadas**
- c) Uma camada pode conter um ou mais pares de pedaços de software distintos e qualquer pedaço de software pode ser composto de **componentes pares** distintos
- d) Qualquer pedaço de software a ser medido deverá ser definido por seu escopo de medição, o qual deve estar integralmente contido em uma única camada
- e) O escopo de um pedaço de software a ser medido depende do **propósito** da medição

# O modelo de contexto de software para o COSMIC

## PRINCÍPIOS:

- f) Os usuários funcionais de um pedaço de software devem ser identificados a partir dos **requisitos funcionais do usuário** do pedaço de software a ser medido, como fontes e/ou destinos pretendidos para os dados
- g) Um pedaço de software interage com os seus usuários funcionais por meio de **movimentações de dados** através de uma **fronteira**, e o pedaço de software pode mover dados de e para o **armazenamento** persistente dentro da fronteira
- h) Os RFU do software podem ser expressos em diferentes **níveis de granularidade**
- i) O nível de granularidade no qual as medições devem ser normalmente efetuadas é o dos **processos funcionais**



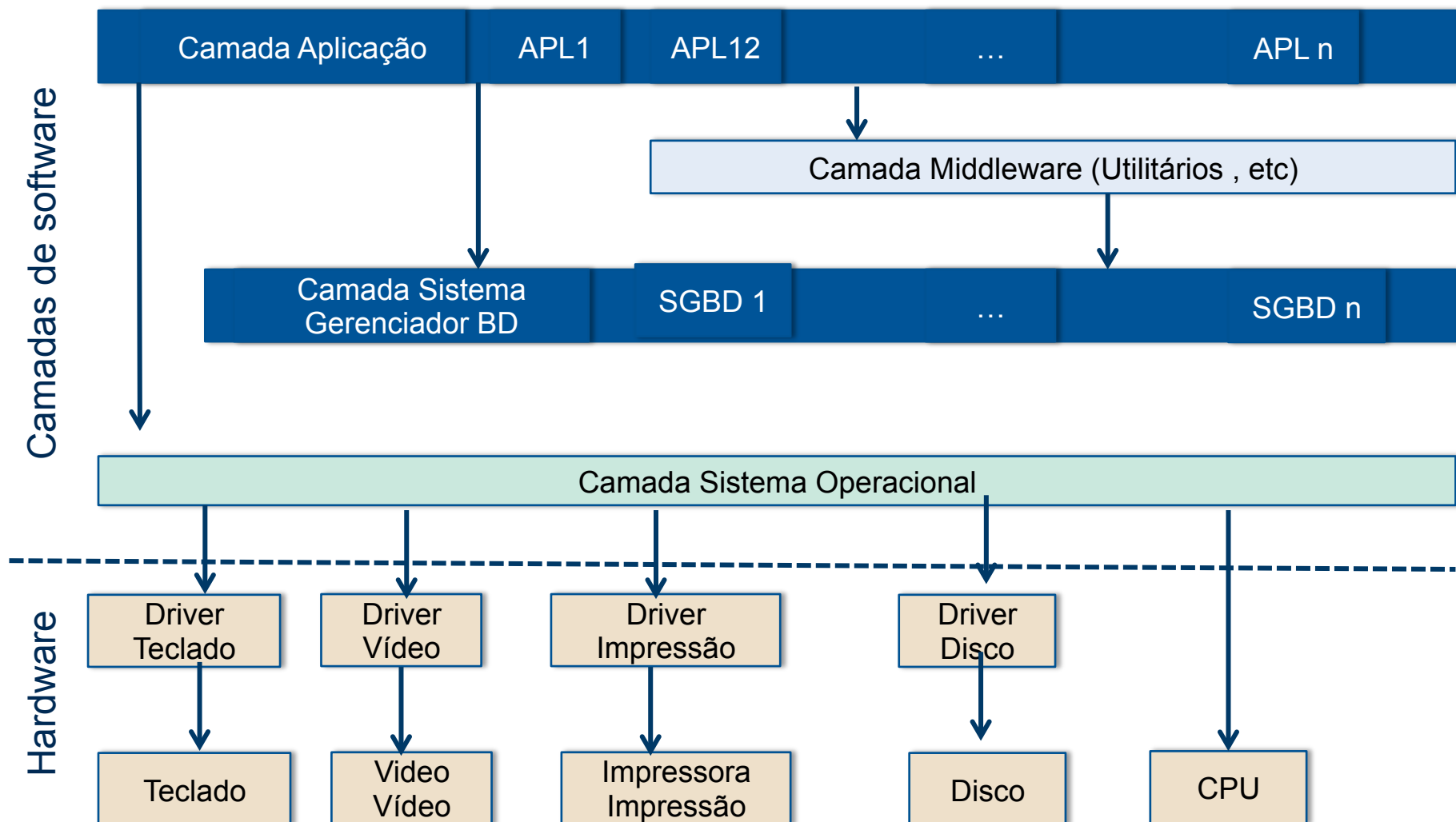
# O modelo de contexto de software para o COSMIC

---

## PRINCÍPIOS:

j) Se não for possível medir no nível de granularidade dos processos funcionais, nesse caso os RFU do software devem ser medidos através de uma abordagem de **aproximação** e **escalonados** para o nível de granularidade dos processos funcionais

# Estratégia de Medição – Modelo de Contexto de Software

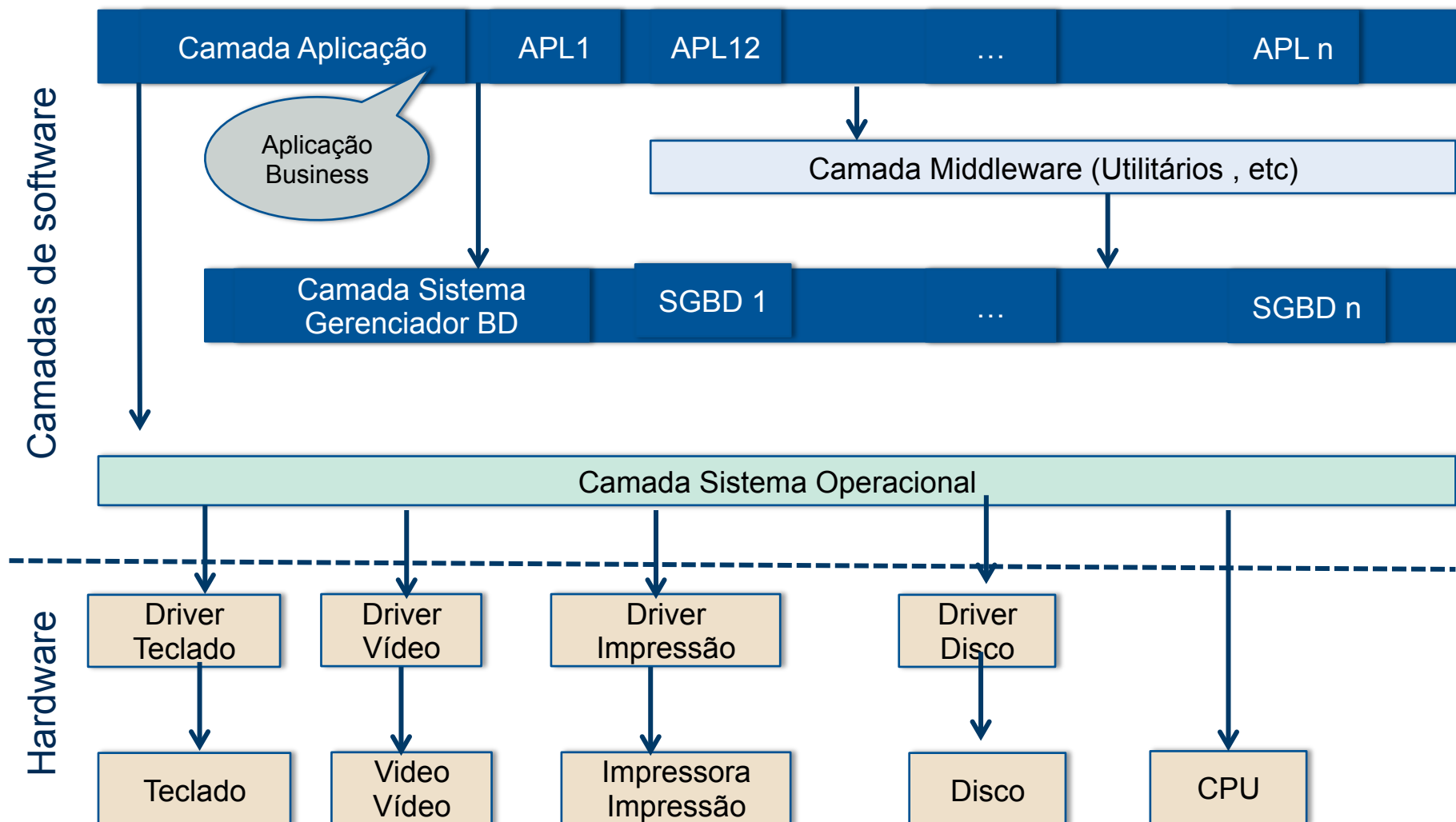


# Camadas

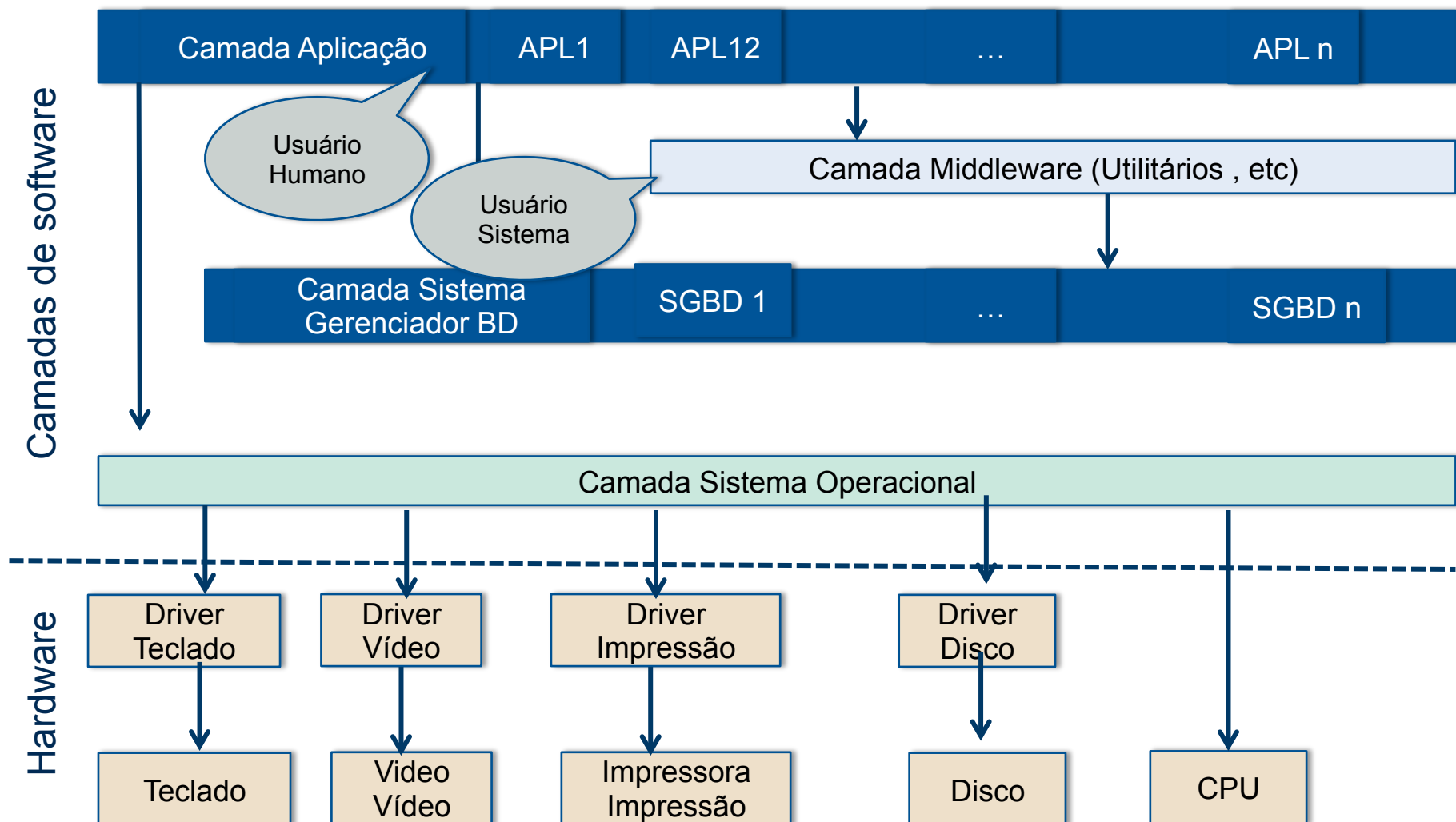
---

- Uma camada é uma partição resultante da divisão funcional de um sistema de software que, juntamente com o hardware, forma um sistema computacional completo onde:
  - As camadas são organizadas segundo uma hierarquia
  - Há apenas uma camada em cada nível de hierarquia
  - Há uma dependência hierárquica do tipo superior/subordinado entre os serviços funcionais providos pelo software em quaisquer duas camadas da arquitetura de software que troquem dados diretamente
  - Os software em quaisquer duas camadas da arquitetura de software que troquem dados interpretam apenas parte daqueles dados identicamente
- O software em uma camada troca dados com o software em outra camada através dos seus **respectivos processos funcionais**

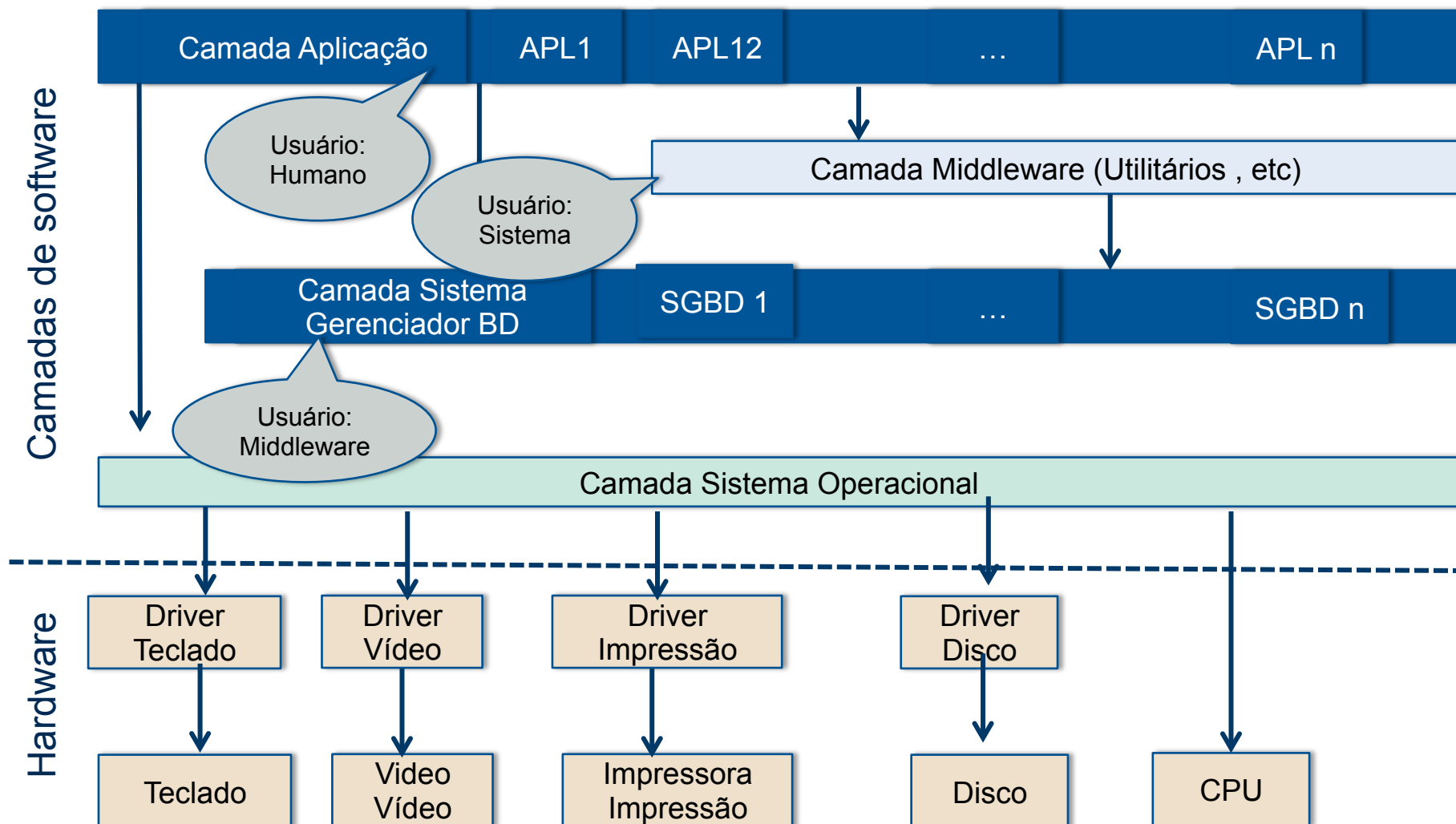
# Estratégia de Medição – Modelo de Contexto de Software



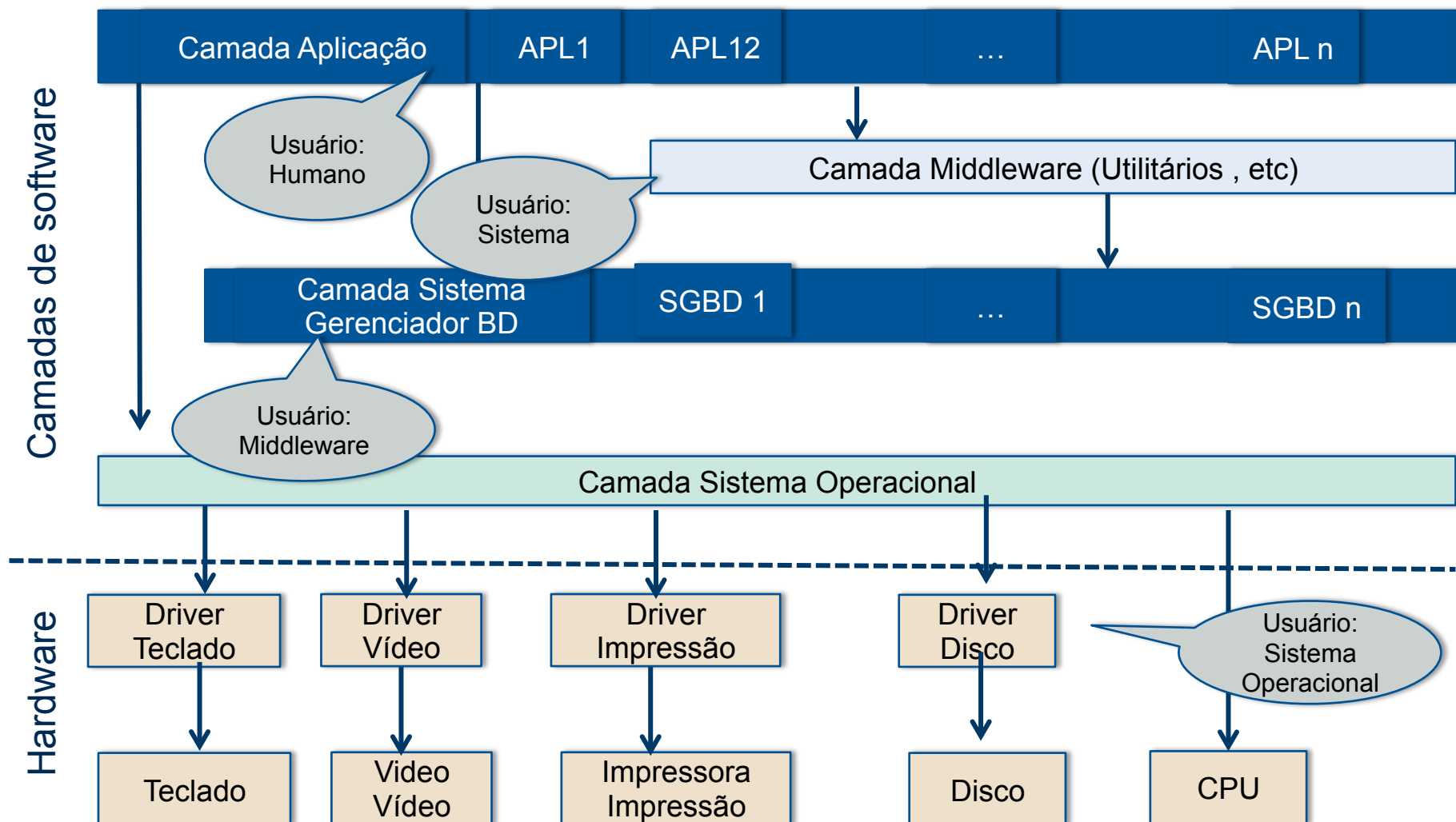
# Estratégia de Medição – Modelo de Contexto de Software



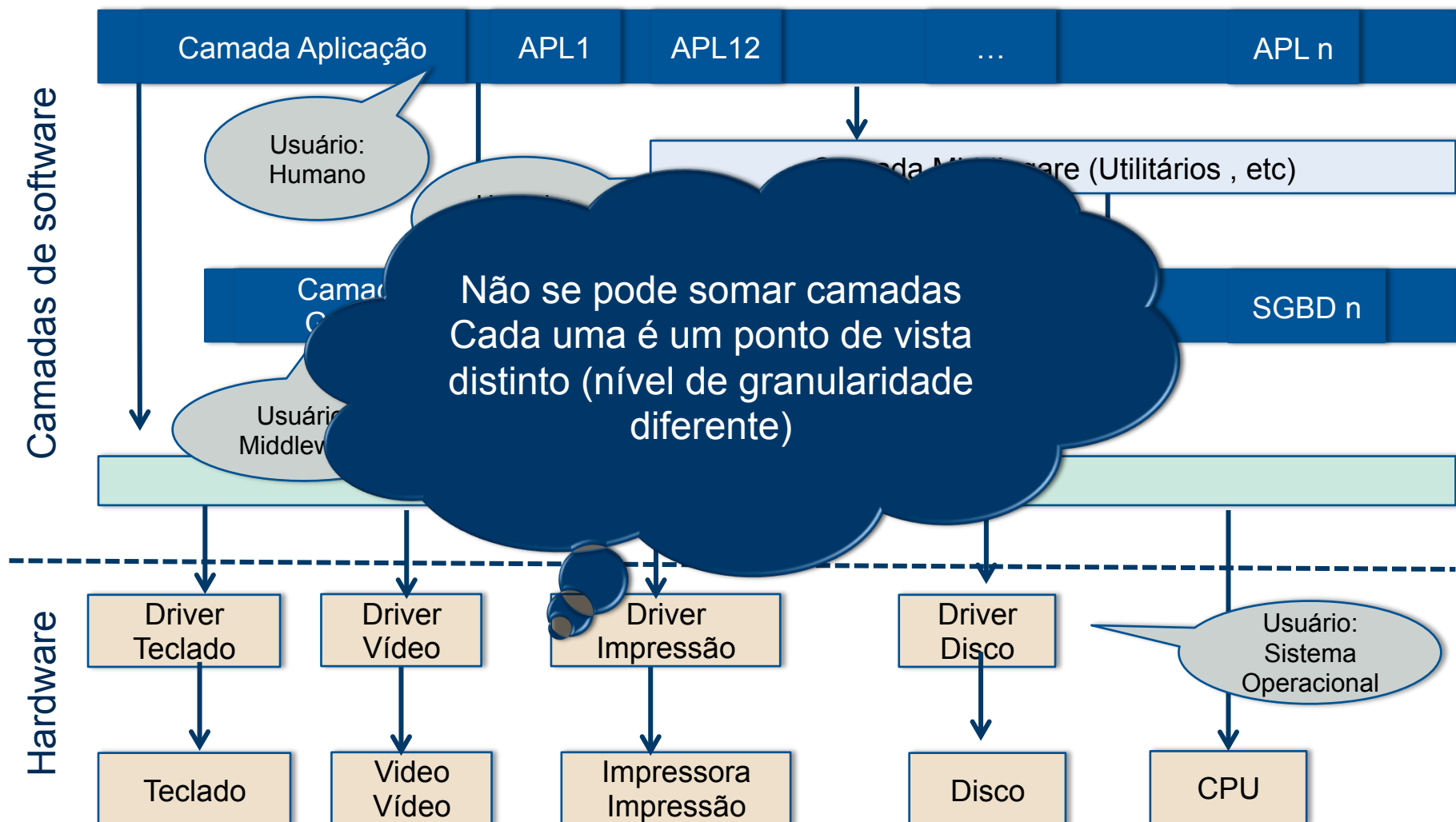
# Estratégia de Medição – Modelo de Contexto de Software



# Estratégia de Medição – Modelo de Contexto de Software



# Estratégia de Medição – Modelo de Contexto de Software





# Nível de Granularidade

---

- Qualquer nível de **expansão da descrição** de um pedaço de software tal que a cada nível de expansão adicional a descrição da funcionalidade do pedaço de software está em um **nível de detalhe maior e uniforme**.
- Os medidores devem estar cientes de que, quando os requisitos evoluem cedo na vida de um projeto de software, a qualquer momento partes da funcionalidade requerida pelo mesmo terão sido documentadas em **diferentes níveis de granularidade**

# Camada e Componentes Pares

- Uma camada é uma partição resultante da divisão funcional de um sistema de software que, juntamente com o hardware, forma um sistema computacional
- Dois pedaços de software são pares quando residirem na mesma camada

# Propósito da Medição

---

- Define a necessidade e a forma de utilização da medição
- Ajuda o medidor a determinar
  - O escopo a ser medido e os artefatos que serão necessários para a medição
  - Os usuários funcionais
  - O momento no ciclo de vida do projeto onde acontecerá a medição
  - A exatidão requerida para a medição
  - O nível de granularidade no qual os RFU serão medidos

# Exemplos

---

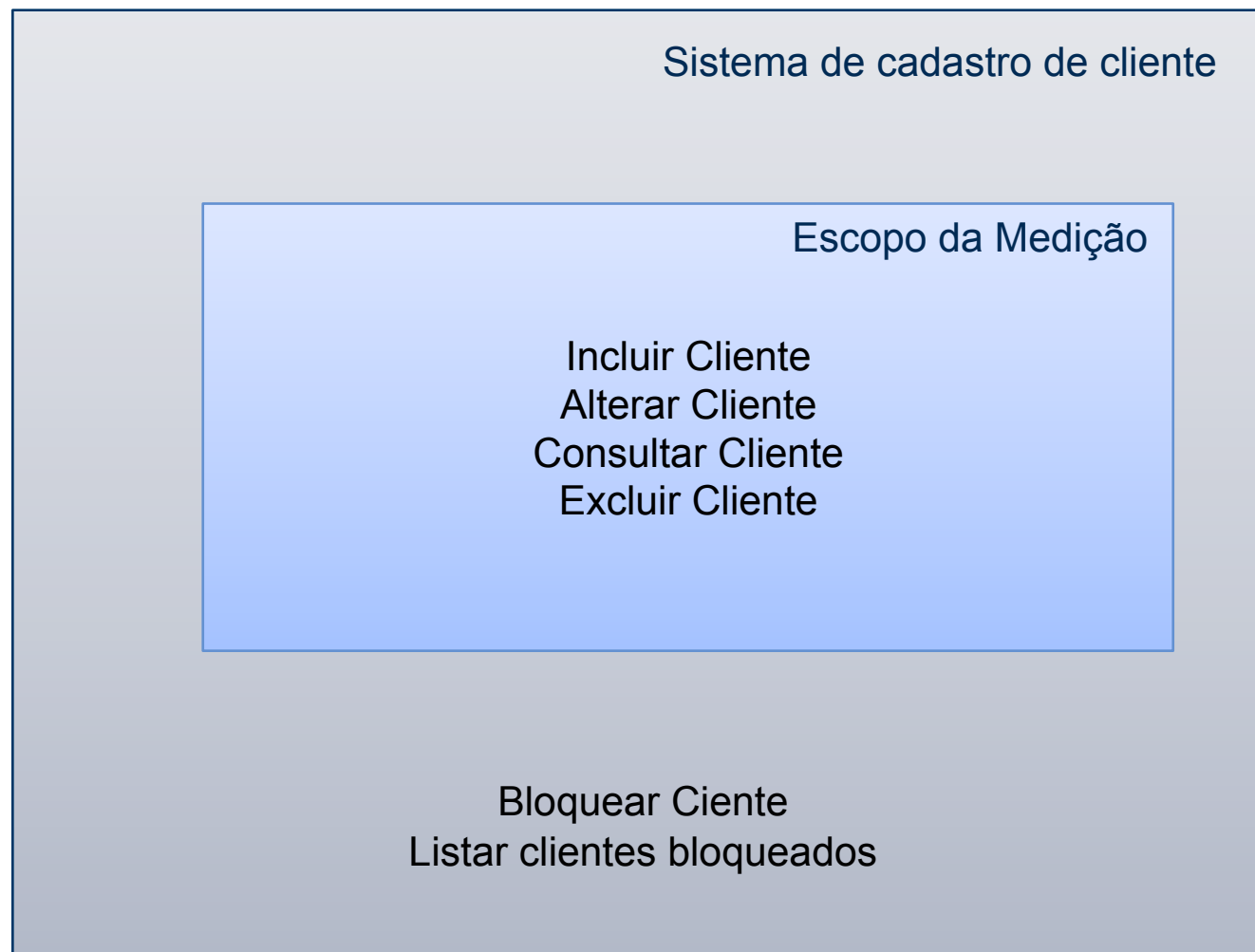
- Estimar o esforço do desenvolvimento
- Gerenciar a variação não controlada no escopo do projeto (*scope creep*)
- Utilizar como entrada no cálculo da produtividade no projeto

# Escopo da Medição

---

- Conjunto de requisitos funcionais do usuário (RFU) a serem incluídos na medição de tamanho funcional
- O escopo deve ser definido antes de ser iniciada a medição
- O escopo de uma medição de tamanho funcional deve estar alinhado ao propósito de medição, pode ser derivado dele
- O escopo de uma medição qualquer não deve contemplar mais do que uma camada do software a ser medido

# Exemplo

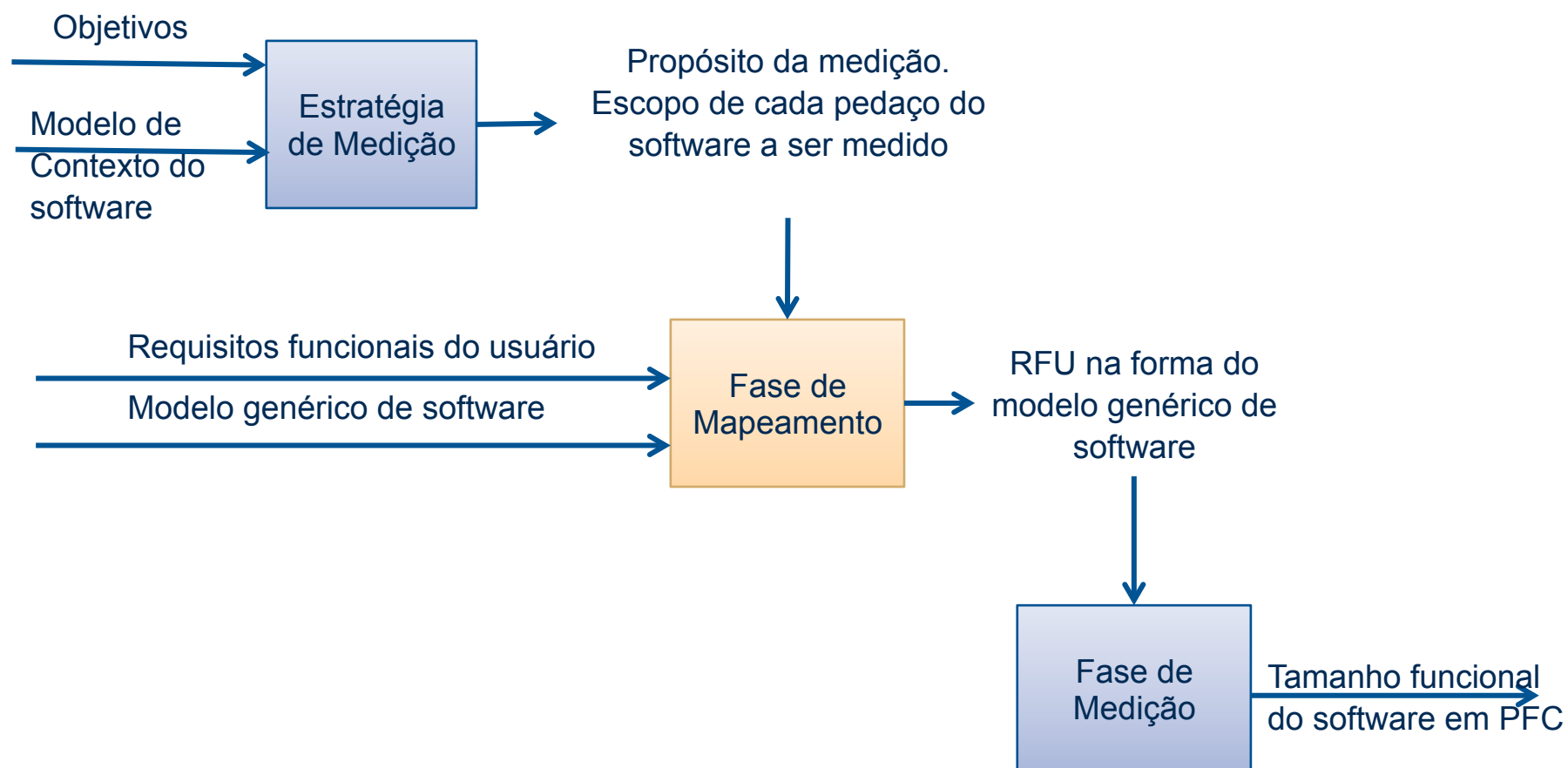


# Pontos Críticos

---

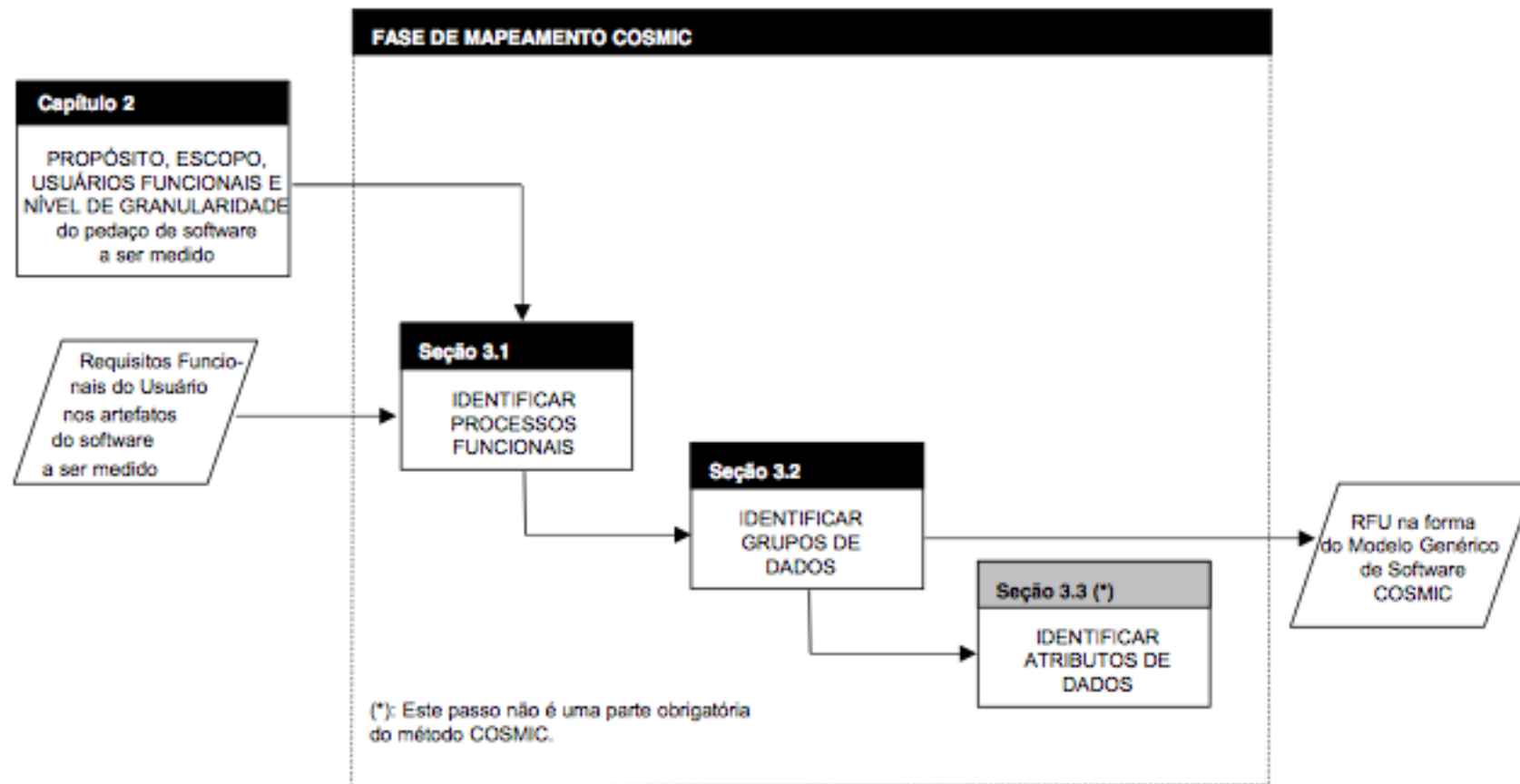
- Estabelecer o propósito da medição
- Definir o escopo total do pedaço de software a ser medido e o escopo dos pedaços a serem medidos separadamente, considerando as camadas e componentes pares da arquitetura do software
- Estabelecer os usuários funcionais do pedaço de software a ser medido
- Estabelecer o nível de granularidade dos artefatos de software a serem medidos

# Fases do processo de medição COSMIC





# Fase de Mapeamento

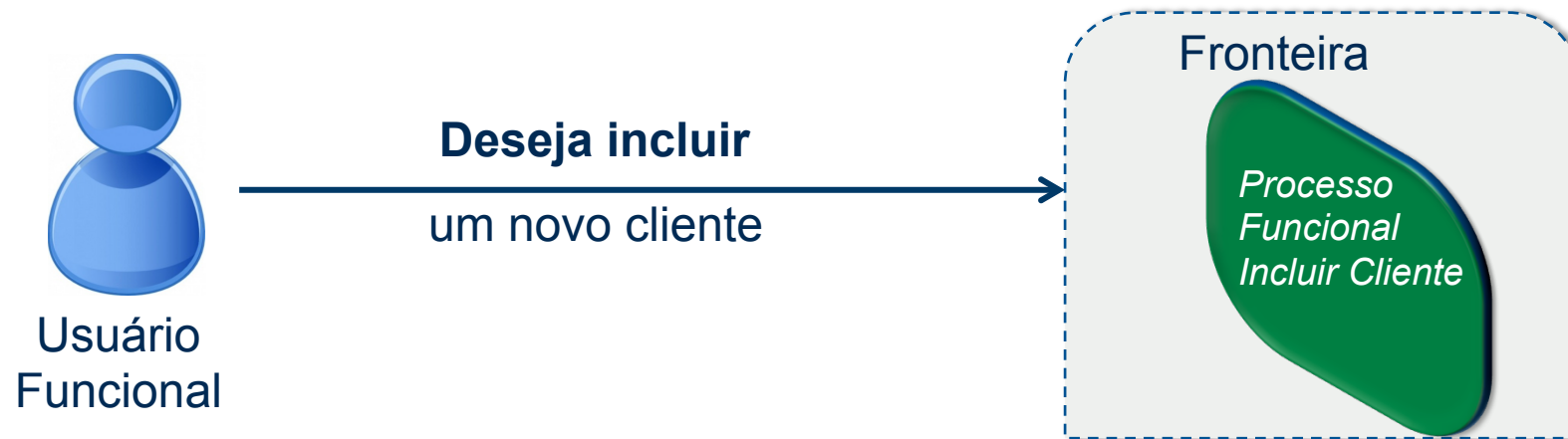


# Aplicando o modelo genérico de software COSMIC

---

- Modelo genérico de software COSMIC deve ser aplicado aos requisitos funcionais do usuário de cada pedaço de software distinto para o qual um escopo de medição, também distinto, tenha sido definido.
- Aplicar esse modelo diz respeito a identificar
  - o conjunto de eventos disparadores percebidos por cada um dos (tipos de) usuários funcionais identificados nos RFU
  - seus processos funcionais associados,
  - objetos de interesse,
  - grupos de dados e
  - movimentações de dados que devem ser providos como respostas a tais eventos.

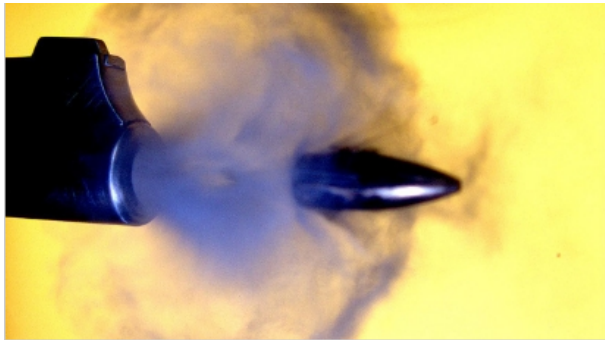
# Processo Funcional



- É um **componente elementar** de um conjunto de requisitos do usuário
- Compreende um **conjunto de movimentações** único, coeso e independentemente executável
- Inicia quando um usuário funcional percebe um **evento** e aciona o processo
- Termina quando tiver executado tudo que é **requerido** em resposta ao evento disparador

# Evento Disparador

---



- Um evento (alguma coisa que acontece) que faz com que um **usuário funcional** inicie (dispare) um ou mais **processos funcionais**.
- Eventos de relógio e oriundos do decorrer do tempo podem ser eventos disparadores.

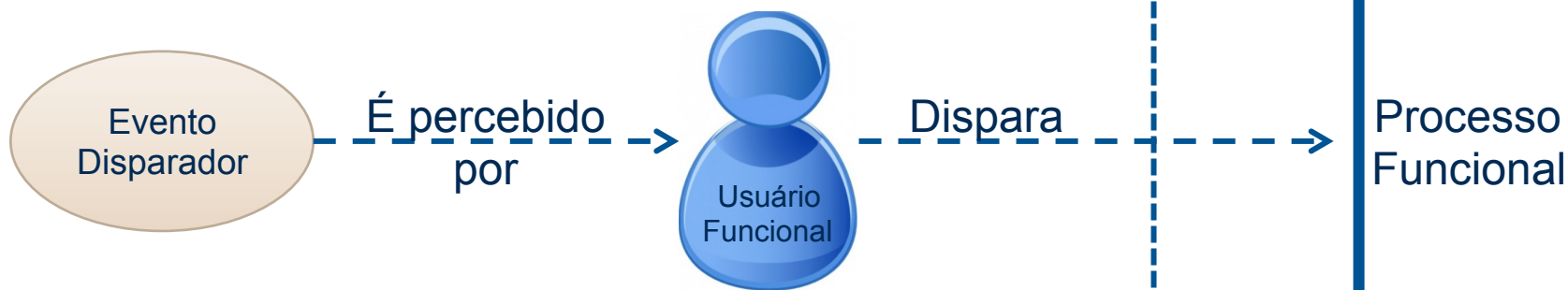
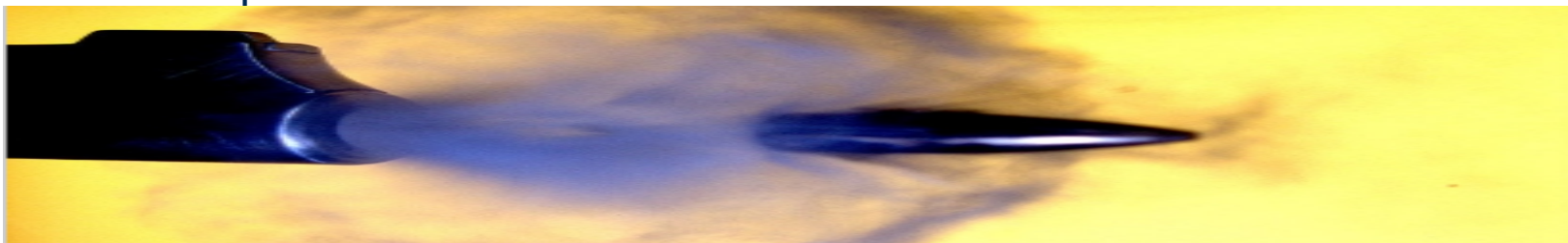
# Evento Disparador



**Deseja Incluir**  
Um novo Cliente



Evento disparador

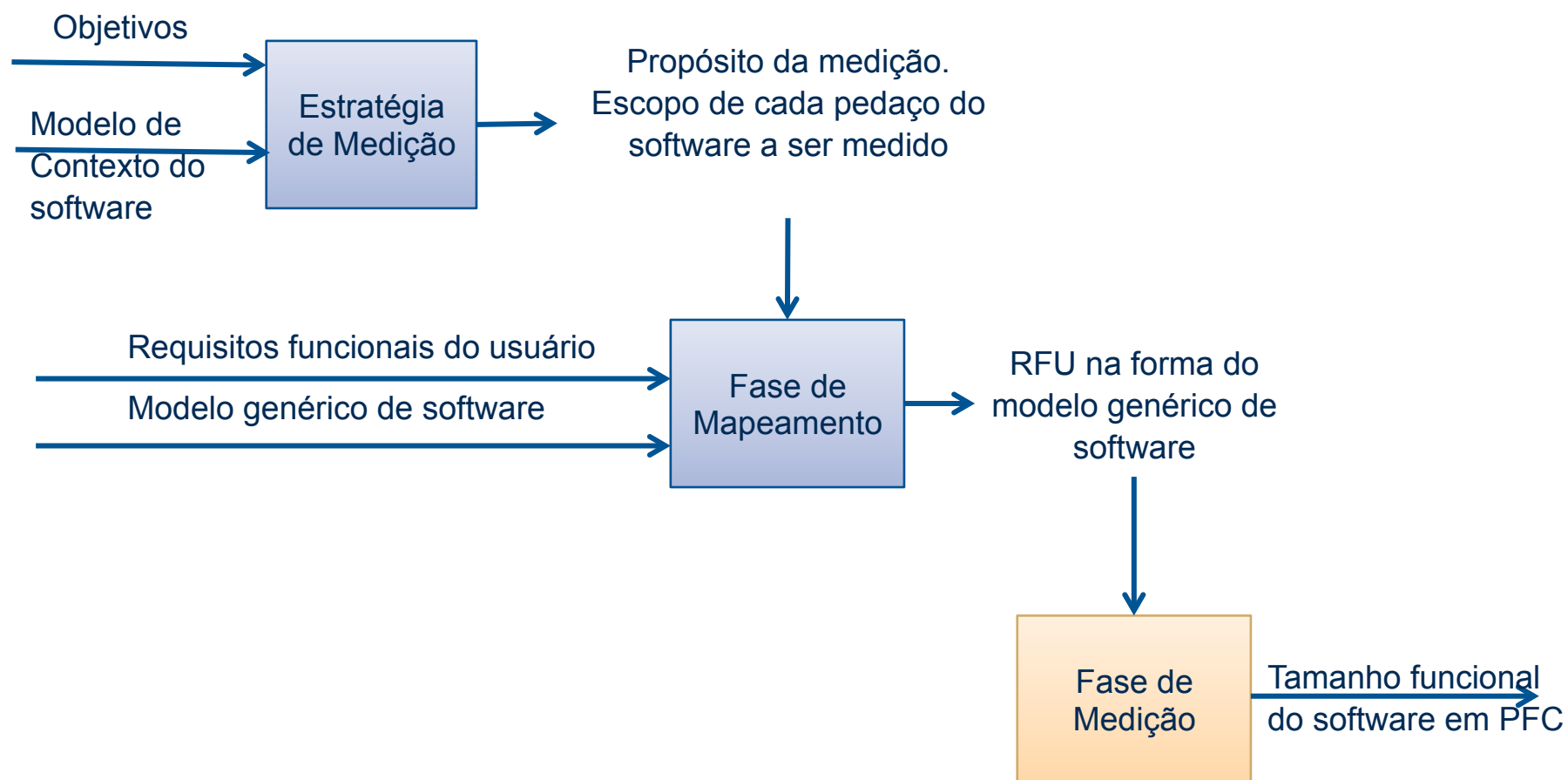


# Outros Conceitos

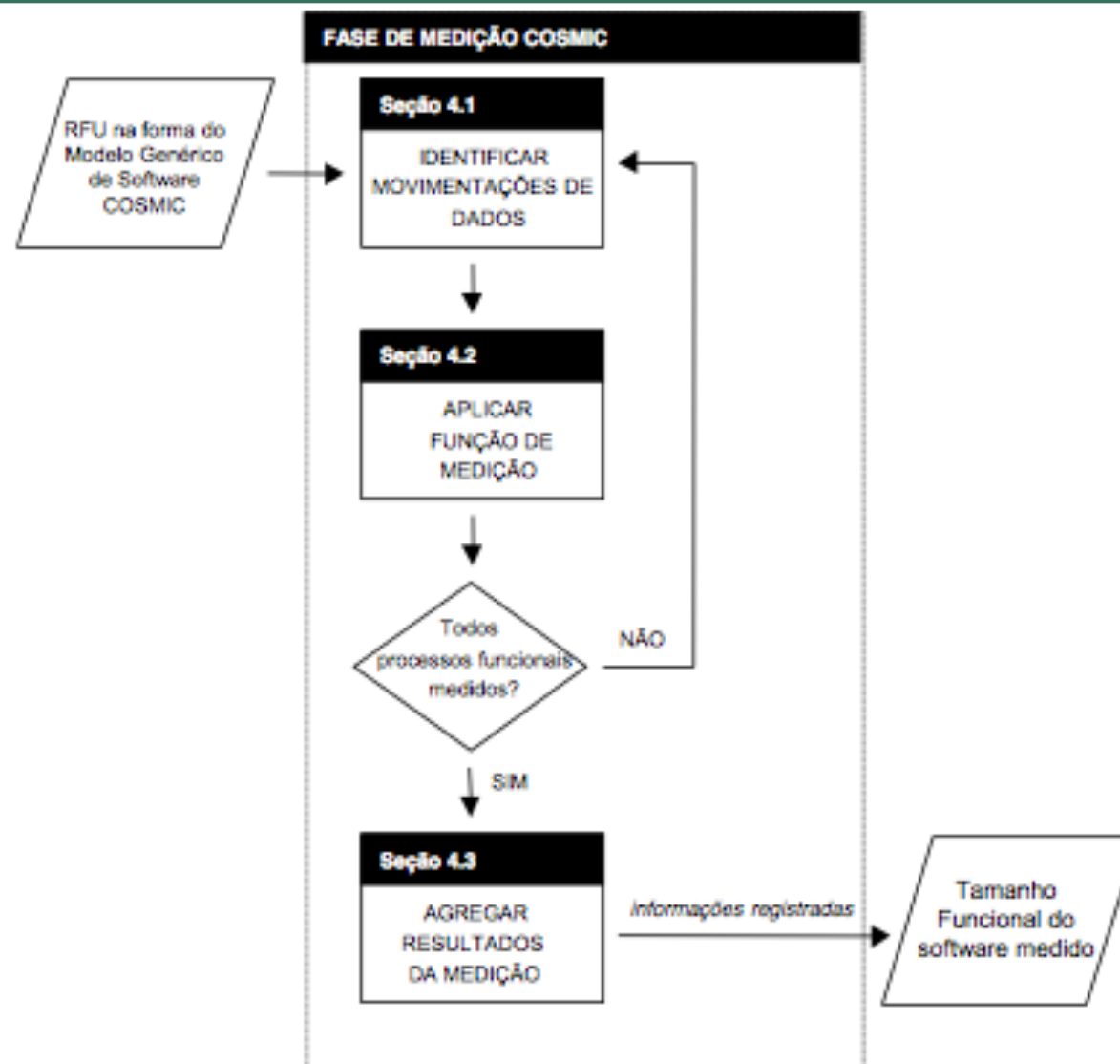
---

- Um **Objeto de Interesse** é qualquer coisa que é identificada do ponto de vista dos requisitos do usuário.
  - Pode ser qualquer coisa física, assim como qualquer objeto conceitual ou parte de um objeto conceitual no mundo do usuário funcional. O objeto de interesse é um conceito do qual o software deve processar e/ou armazenar dados.
- Um **grupo de dados** é um conjunto de atributos de dados distintos, não vazio, não ordenado e não redundante, onde cada atributo de dado descreve um aspecto complementar do mesmo objeto de interesse
- Um **atributo de dado** é a menor parcela de informação, dentro de um grupo de dados identificado, que possui significado segundo a perspectiva dos requisitos funcionais do usuário do software

# Fases do processo de medição COSMIC



# Fase de Medição





# Movimentação de Dados

---

- Um componente funcional básico que **movimenta um único grupo de dados** (tipo)
- Há quatro subtipos de tipos de movimentação de dados: Entry (E), Exit (X), Read (R) e Write (W)
- Para fins de medição, considera-se que cada subtipo de movimentação de dados inclui certas **manipulações de dados associadas**
- De uma forma mais precisa, trata-se de uma ocorrência de uma movimentação de dados, e não um tipo de movimentação de dados, que realmente movimenta as ocorrências do grupo de dados

# Entry (E)

---

- Uma **movimentação de dados** (tipo) que movimenta um **grupo de dados** de um **usuário funcional** através da **fronteira** para dentro do **processo funcional** no qual o mesmo é requerido.
- Considera-se que uma entry (tipo) inclui certas manipulações de dados associadas

# Exit (X)

---

- Uma **movimentação de dados** que movimenta um **grupo de dados** de um **processo funcional** através da **fronteira** na direção do **usuário funcional** que o requisita.
- Considera-se que uma Exit (tipo) inclui certas manipulações de dados associadas

# Read (R)

---

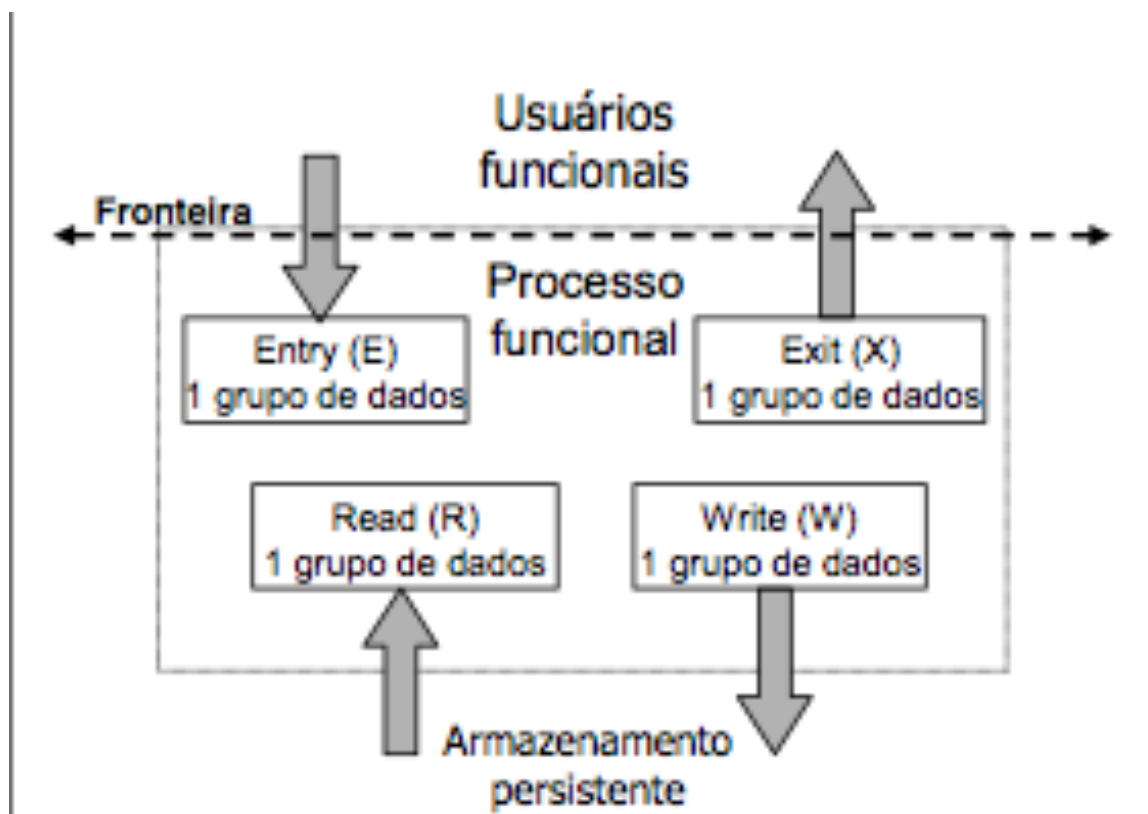
- Uma **movimentação de dados** que movimenta um **grupo de dados** do **armazenamento persistente** para dentro do alcance do **processo funcional** que dele necessita.
- Considera-se que um Read (tipo) inclui certas manipulações de dados associadas

# Write (W)

---

- Uma **movimentação de dados** que movimenta um **grupo de dados** do interior de um **processo funcional** para o **armazenamento persistente**.
- Considera-se que um Write (tipo) inclui certas manipulações de dados associadas

# Definições



# Reportando a medição

---

- Um resultado de medição COSMIC deve ser designado como “**x** CFP (v.**y**)”, onde:
  - “**x**” representa o valor numérico do tamanho funcional,
  - “v.**y**” representa a identificação da versão do padrão COSMIC utilizado para obter o valor numérico do tamanho funcional “x”.

# Exercício de fixação

## APLICANDO O MÉTODO COSMIC

### SISTEMA DE CADASTRO DE CLIENTES

---

- Foi solicitado um sistema de cadastro de clientes. Inicialmente o sistema deverá permitir a inclusão, a consulta, a alteração e a exclusão de clientes
- Requisitos da Inclusão:
  - A identificação do cliente será pelo CPF
  - Na inclusão o sistema deverá validar as informações fornecidas e verificar se o CPF informado já existe
  - Caso exista o CPF informado pelo usuário, o sistema deverá retornar uma mensagem de erro
  - Caso não exista, o sistema deverá incluir as informações do cliente e retornar a mensagem de inclusão realizada com sucesso.
- Requisitos de Consulta:
  - O usuário poderá consultar todas as informações do usuário por nome completo ou CPF
  - Quando o usuário informar o nome, o sistema apresentará uma lista de clientes com nome semelhante ao fornecido pelo usuário
  - O usuário escolhe o cliente e o sistema apresenta todas as informações do cliente selecionado na lista.



# Exercício de fixação

## APLICANDO O MÉTODO COSMIC

### SISTEMA DE CADASTRO DE CLIENTES

---

- Requisitos de Alteração:
  - Na alteração o usuário informa o CPF
  - O sistema apresenta os dados do cliente
  - Caso o sistema não encontre o CPF informado, deverá ser apresentada uma mensagem de erro
  - O usuário poderá alterar qualquer informação, exceto o CPF
  - O sistema apresenta mensagem de sucesso
- Requisitos de Exclusão:
  - Na exclusão o usuário informa o CPF
  - O sistema deverá verificar se o cliente existe
  - O sistema exclui o cliente e apresenta mensagem de sucesso
  - Caso o sistema não encontre o CPF informado, deverá ser apresentada uma mensagem de erro.