Tolerância a Falhas e Segurança

Sistemas Distribuídos e Mobile
Prof. Gustavo Torres Custodio
gustavo.custodio@anhembi.br

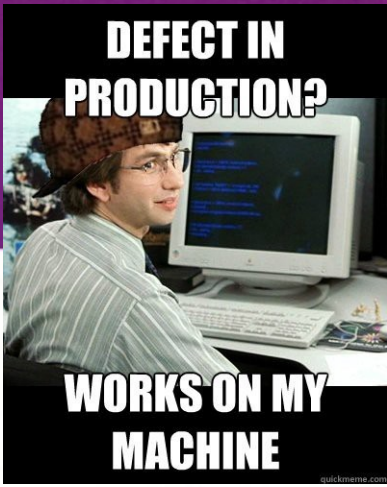
Agenda

- Tolerância a Falhas
  - Terminologia
  - Falhas
  - Dependabilidade
  - Modelo de Falhas
  - Redundância
  - Detecção de Falhas
- Segurança

Terminologia

- Falha (Fail)
  - Um componente não está de acordo com suas especificações
  - Exemplo: Programa quebrou
- Erro (Error)
  - Parte de um componente que pode levar a uma falha
  - Exemplo: Erro de programação
- Defeito (Fault)
  - Causa de um erro Programa travado
  - Exemplo: Programador sem atenção.

Lidando com Defeitos



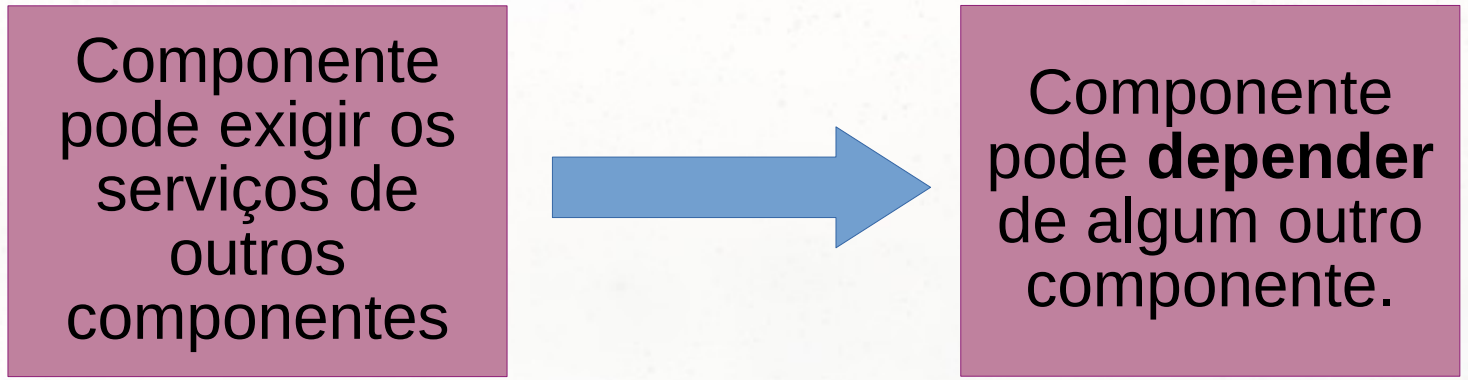
- Prevenção de Defeitos:
  - Prevenir que falhas ocorram
  - Exemplo: Não contrate programadores desatentos
- Tolerância a Defeitos:
  - Construir um componente que possa “esconder” o acontecimento de uma falha
  - Exemplo: Construir cada componente por equipes diferentes.
- Remoção das Defeitos
  - Reduzir a presença ou o número ou a seriedade do defeito
  - Exemplo: Livre-se dos programadores desatentos
- Previsão de Defeitos
  - Estimar a presença de atuais defeitos, futuros defeitos e consequências dos defeitos

ăFalhas

- Falha total em sistemas não distribuídos x falha parcial em sistemas distribuídos
  - Deve-se projetar o SD de modo tal que ele possa se recuperar automaticamente de falhas parciais sem afetar seriamente o desempenho global
  - Em casos de falhas, o sistema distribuído deve continuar a funcionar de maneira aceitável enquanto estiver sendo recuperado
  - Resumindo, o SD deve tolerar falhas e continuar a funcionar, mesmo na presença de falhas em seus componentes

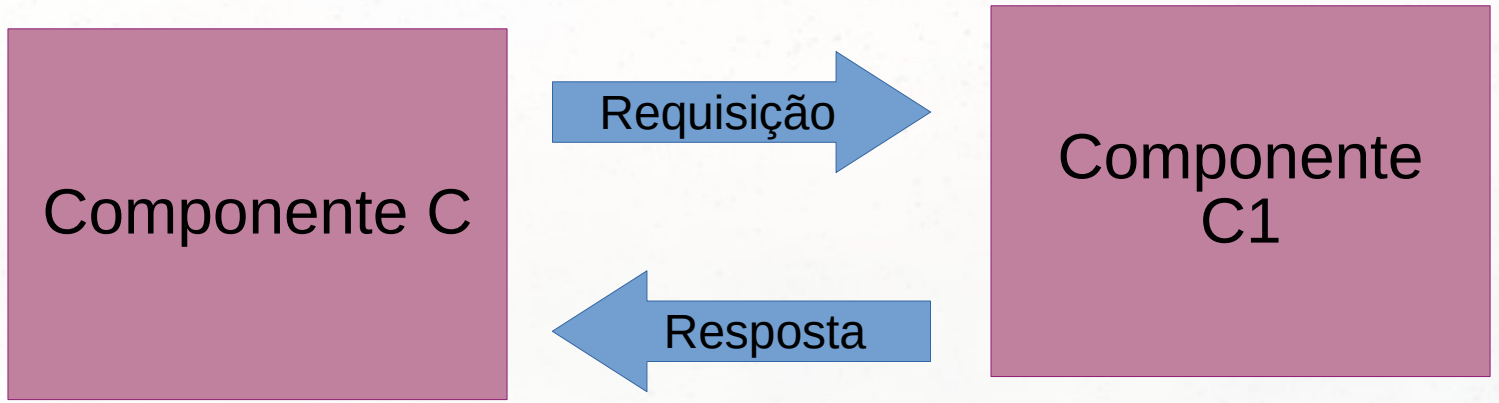
ăDependabilidade

- Um componente fornece serviços aos clientes.
- Para fornecer serviços:



Dependabilidade

- Um componente C depende de C1:
  - se a correção (correctness) do comportamento de C depende da correção (correctness) do comportamento de C1.



Dependabilidade

- Disponibilidade (Availability)
  - Prontidão para uso
- Confiabilidade (Reliability)
  - Continuidade da prestação de serviço
- Segurança (Safety)
  - Probabilidade muito baixa de catástrofes
- Manutenibilidade (Maintainability)
  - Quão fácil um sistema com falha pode ser reparado

Disponibilidade (Availability)

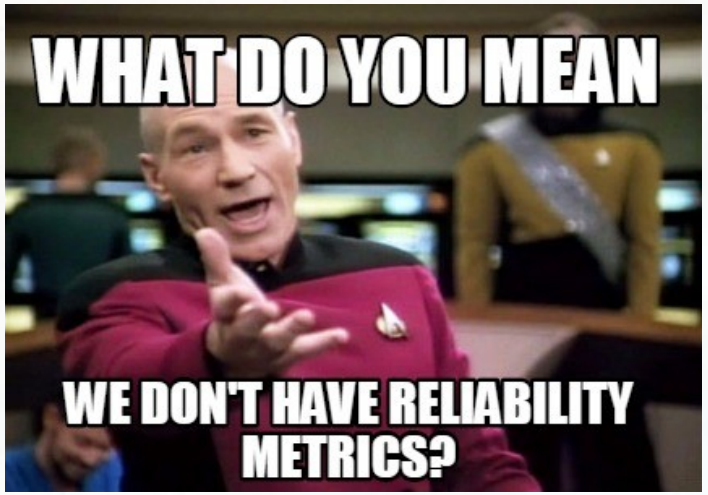
- Fração média de tempo em que C esteve disponível no intervalo [0, t).

- Existe algum sistema que precise de disponibilidade de 100%? Qual?

Confiabilidade (Reliability)

- Probabilidade condicional de que C tenha funcionado corretamente durante [0, t)
  - dado que C estava funcionando corretamente no tempo T = 0.

- Exemplo??



Segurança (Safety)

- Se um sistema deixar de funcionar corretamente durante um certo tempo,
  - nada de catastrófico acontecerá ...?

- Exemplo??

ăManutenibilidade (Maintainability)

- Facilidade com que um sistema que falhou possa ser consertado

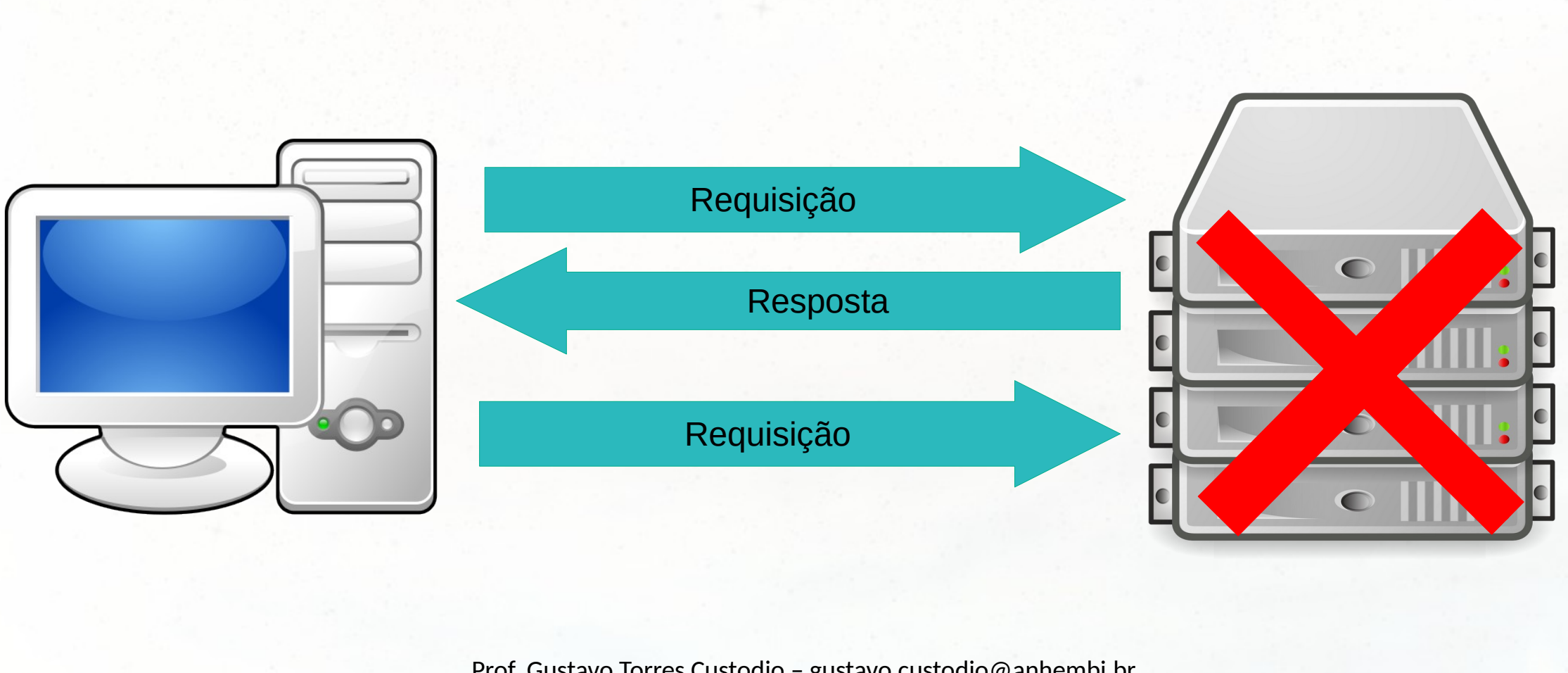
- Exemplo??

Modelo de Falhas

- Falha por crash
- Falha por omissão
  - Omissão de recebimento
  - Omissão de envio
- Falha de temporização
- Falha na resposta
  - Falha de valor
  - Falha de transição de estado
- Falha arbitrária

Falha por Crash

- O servidor para de funcionar, mas estava funcionando até parar.

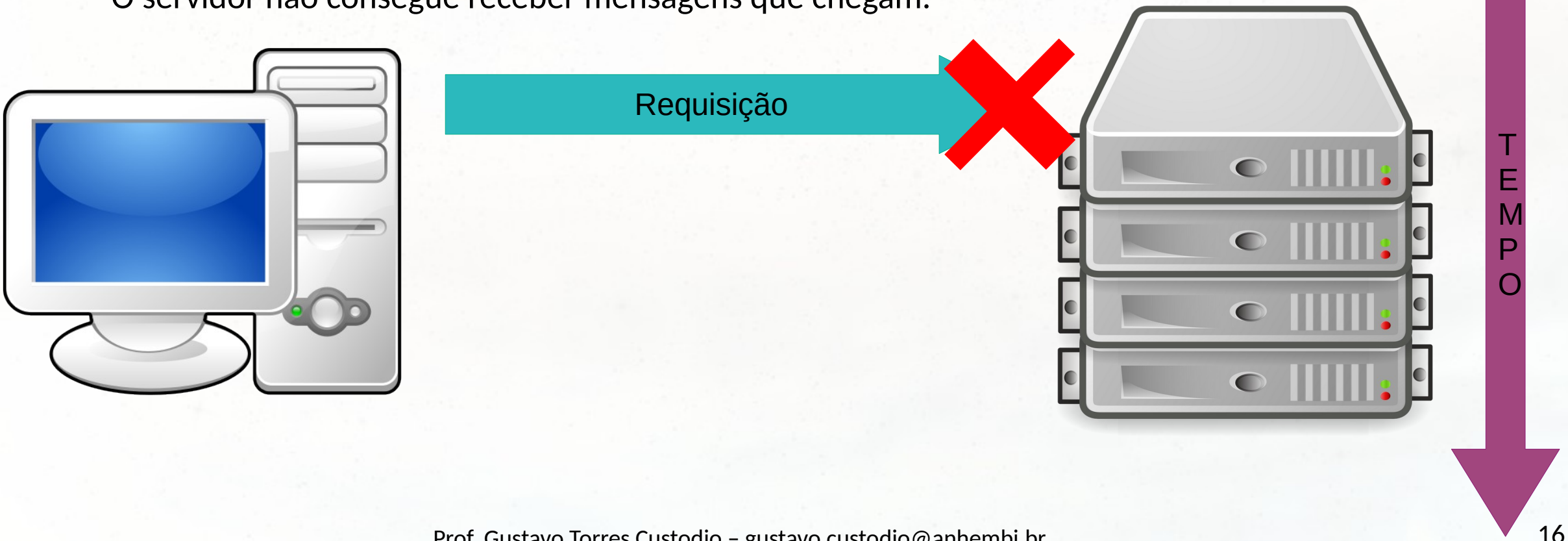


Falha por Crash

- Cliente não acha o servidor:
  - O servidor está fora do ar;
  - Soluções:
    - Retornos de variáveis “inválidas”: e.g. -1
    - Criação de exceções
      - Perda da transparência

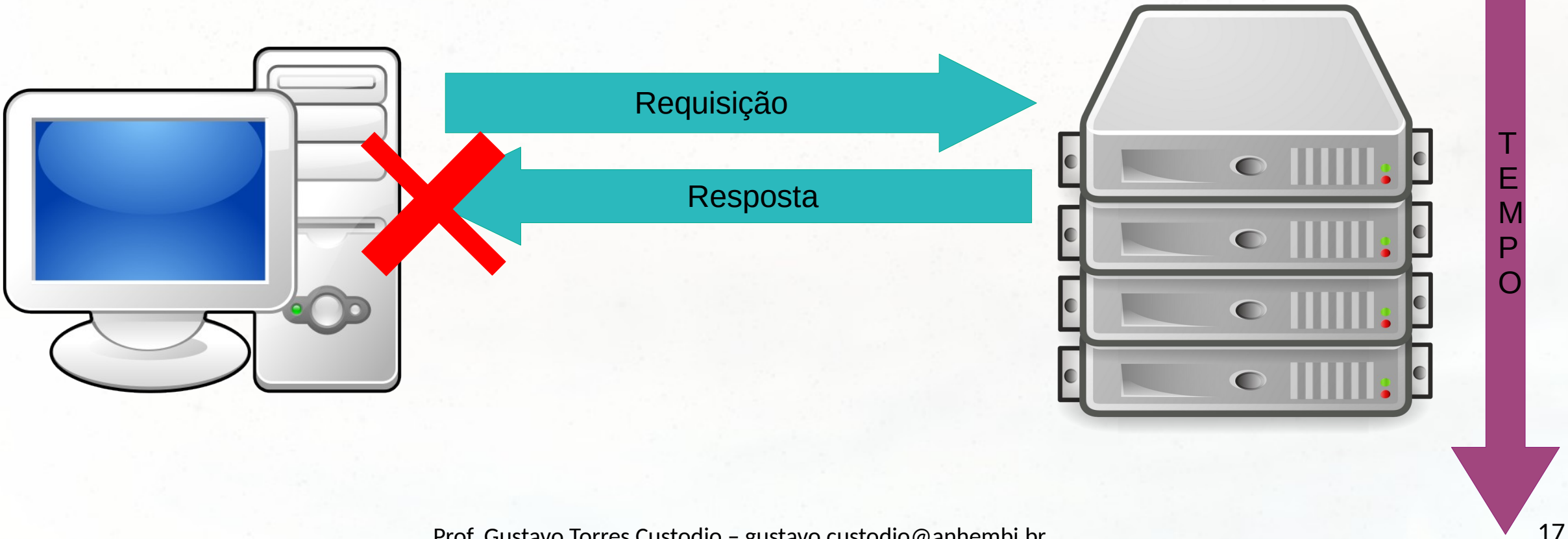
Falha por Omissão

- O servidor não consegue responder a requisições que chegam
- Omissão de Recebimento
  - O servidor não consegue receber mensagens que chegam.



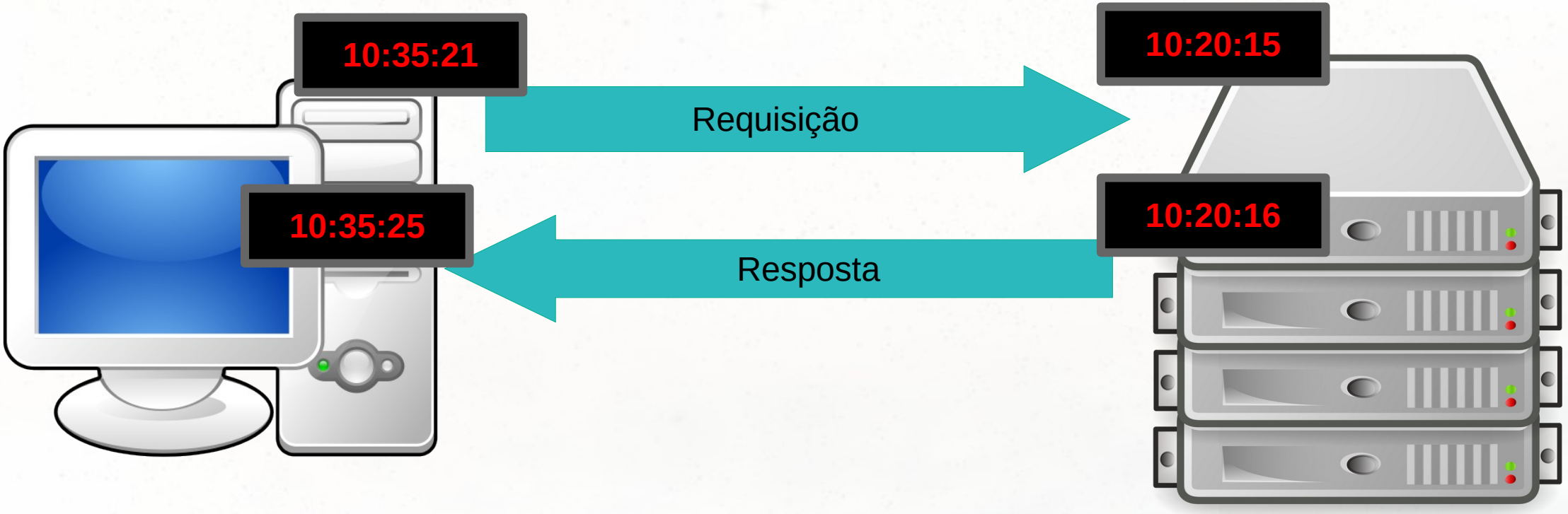
Falha por Omissão

- Omissão de envio
  - O servidor não consegue enviar mensagens



Falha de Temporização

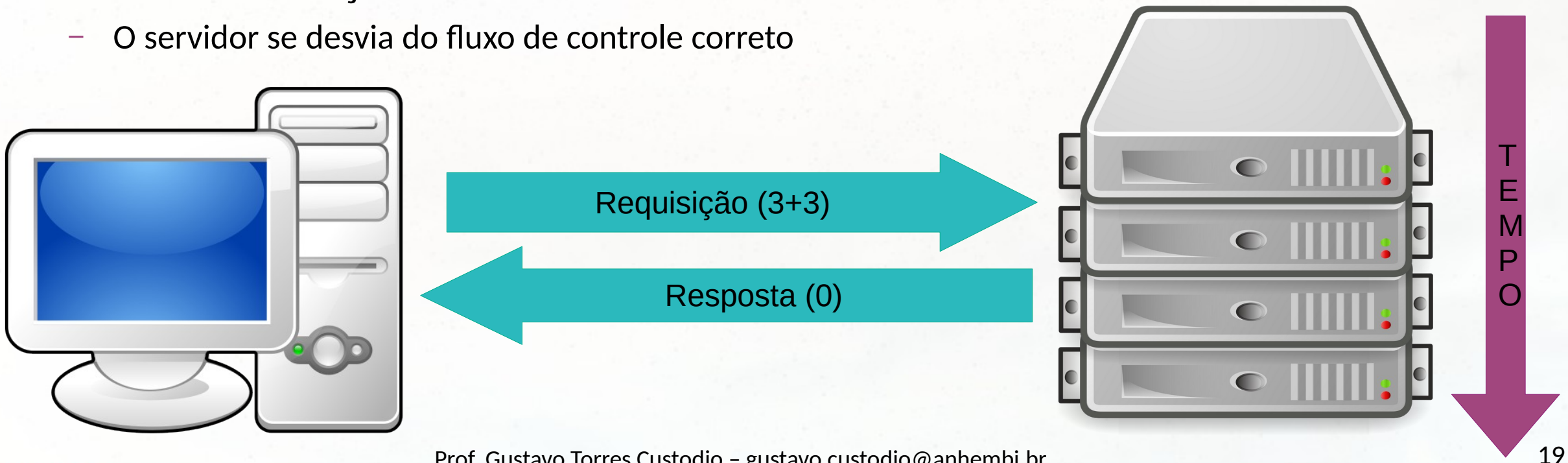
- A resposta do servidor se encontra fora do intervalo de tempo



- Veremos isso com mais detalhes na aula de sincronização

Falha na resposta

- A resposta do servidor está incorreta
- Falha de valor:
  - O valor da resposta está errado
- Falha de transição de estado
  - O servidor se desvia do fluxo de controle correto



ăFalha Arbitrária

- Geralmente são classificadas como maliciosas
  - Omissão de falhas:
    - Um componente falha em uma ação que deveria agir
  - Encarregado de falhas
    - Um componente toma uma ação que não deveria tomar
- Ambos os tipos de falhas são intencionais:
  - São tipicamente problemas de segurança (security).

Redundância

- Duplicação de componentes (hardware ou software)
- Técnica para mascarar falhas
  - Redundância de informação
    - Bits extras são adicionados para permitir recuperação de bits deteriorados.
  - Redundância de tempo
    - Uma ação é realizada e, então, se for preciso, ela é executada novamente. Ex: Transações podem ser repetidas, caso tenham sido abortadas
  - Redundância física
    - Componentes físicos duplicados podem ser usados.

ăDetecção de Falhas

- Como podemos detectar de forma **confiável** que um processo realmente travou?

- Modo Genérico:
  - Cada processo é equipado com um módulo de detecção de falhas
  - Um processo P sonda outro processo Q para uma reação
  - Se Q reage:
    - Q é considerado vivo (por P)
  - Se Q não reagir com t unidades de tempo:
    - suspeita-se que Q tenha travado

Detecção de Falhas

- Implementação
  - Se P não recebeu “pulsação” de Q dentro do tempo t: P suspeita de Q.
  - Se Q posteriormente enviar uma mensagem (que é recebida por P):
    - P deixa de suspeitar Q
    - P aumenta o valor de timeout t
  - Nota: se Q falhou, P continuará suspeitando de Q.

Falhas Genéricas

- A mensagem do cliente para o servidor foi perdida;
- A mensagem do servidor para o cliente foi perdida;
- O servidor sai do ar após receber uma solicitação;
- O cliente sai do ar após ter enviado uma solicitação.

Falhas Genéricas

- A mensagem do cliente para o servidor foi perdida:
  - Limite de tempo de espera (timeout);
  - Reenvio em kernel;
  - Retorno de erro, após diversas tentativas.

Falhas Genéricas

- O servidor sai do ar após receber uma solicitação:
  - Espera e reenvia / ache novo servidor;
  - Desiste e comunique falha.
- As falhas até o momento não são distinguíveis para um Cliente.

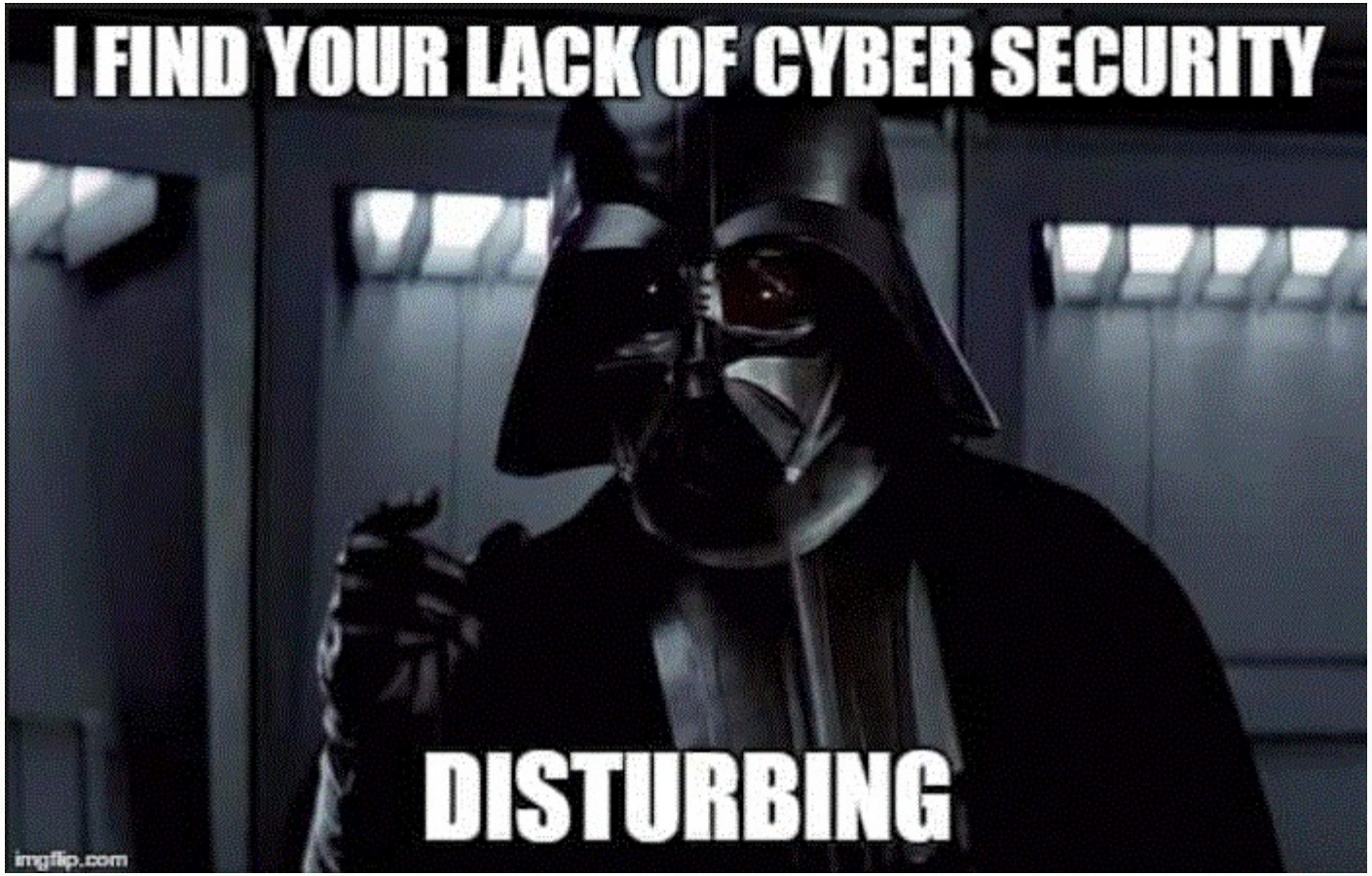
Falhas Genéricas

- O cliente sai do ar após ter enviado uma solicitação:
  - Processamento órfão;
  - Gasto de tempo do servidor;
  - Soluções:
    - Reencarnação;
    - Extermínio;
    - Expiração (quantum T).

Falhas Genéricas

- Extermínio:
  - Eliminar todos os órfãos
  - Problema: encadeamento de falhas (servidor pode ter sido cliente em uma RPC)
- Reencarnação:
  - Dividir o tempo em “épocas”
  - Nova chamada → nova época
  - Falha na rede → impossível encontrar o órfão → facilmente detectado depois (necessita nova época)
- Expiração:
  - Tempo máximo para servidor executar serviço
  - Problema: mensurar o quantum

Segurança



Segurança em Sistemas Distribuído

- A segurança em sistemas distribuídos é um aspecto importante no projeto e implementação deste tipo de sistema.
- Ocorre que os vários componentes de hardware e software precisam se comunicar e colaborar entre si sendo que existem diversos tipos de ameaça a segurança destes sistemas.
- Devemos considerar que a segurança sempre envolve o fator humano e o fator tecnológico que inclui hardware e software.
  - Como sempre, o fator humano é o elo fraco da corrente que pode ser quebrado mais facilmente.

Segurança em Sistemas Distribuído

- A segurança em um sistema, de uma maneira geral, está relacionada a:
  - Confidencialidade
  - Integridade
  - Autenticidade
  - Disponibilidade
  - Não repúdio

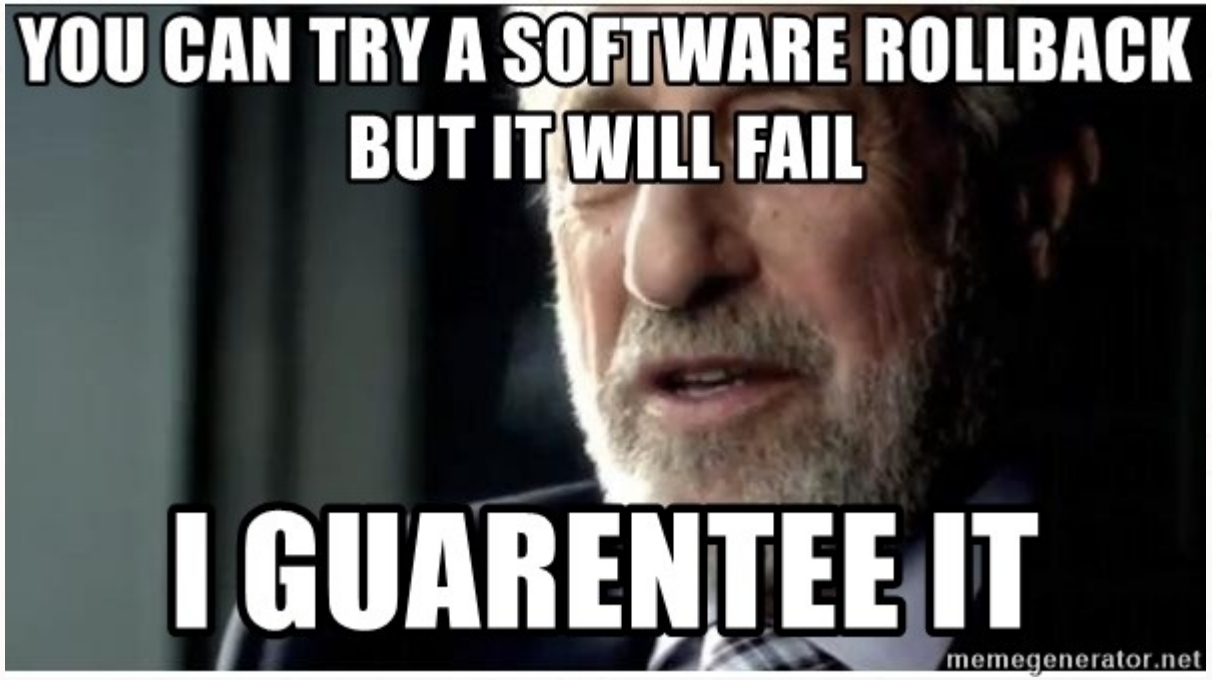
Principais ameaças à Segurança

- Interceptação
  - Captura de mensagens trocadas entre usuários e serviços em um sistema distribuído
- Interrupção
  - Tornar um serviço indisponível temporariamente ou permanentemente
- Modificação
  - Alteração de informações ou configurações em um componente de um sistema distribuído
- Fabricação
  - Uso de identidade, ticket ou certificado digital falso para acesso ou realização de alguma atividade não autorizada

Mecanismos de Segurança

- Os principais mecanismos de segurança que suportam sistemas distribuídos são:
  - Encriptação
    - Uso de criptografia para troca de mensagens e armazenamento de informações sensíveis
  - Autenticação
    - Para acesso a um recurso
  - Autorização
    - Para executar ações em um recurso
  - Auditoria
    - Registro de atividades realizadas em logs.

Exercícios


YOU CAN TRY A SOFTWARE ROLLBACK
BUT IT WILL FAIL
I GUARENTEE IT
memegenerator.net

Exercício

- Projete um sistema distribuído sobre:
  - Sistema de aluguel de carros
  - Sistema de venda de passagens
- E indique onde estaria a tolerância a falhas e a segurança do sistema.

Agradecimentos

- Prof. Bruno Iizuka pelos slides.

Obrigado!
Bom Dia!
Boa Tarde!
Boa Noite!


ecossistema
ânima