Arquitetura de software

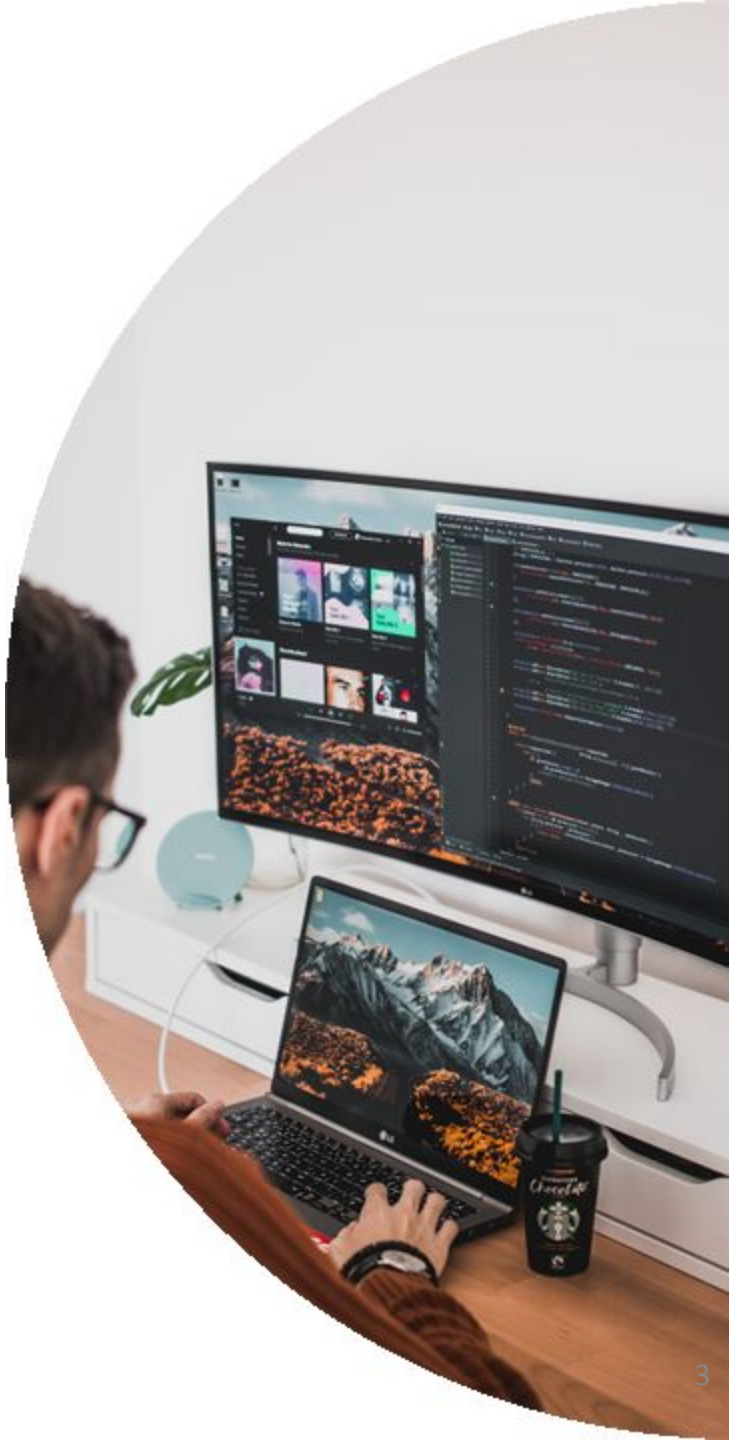
Prof. Eduardo Cizeski Meneghel

Objetivo

- **Fundamentos para sistema com boa performance;**

- **Fundamentos para sistemas seguros.**

Fundamentos para sistemas com boa performance



Introdução

1. Definições gerais;
2. Aspectos relativos à performance;
3. Obtenção de performance.

Definições gerais

O desempenho é a carga de trabalho que um sistema consegue realizar, em um determinado intervalo de tempo, com a quantidade de recursos demandada.

Definições gerais

Um sistema tem bom desempenho quando atende às expectativas relativas aos tempos para executar processamentos, sob determinadas cargas, sem saturar os recursos computacionais disponíveis.

Definições gerais

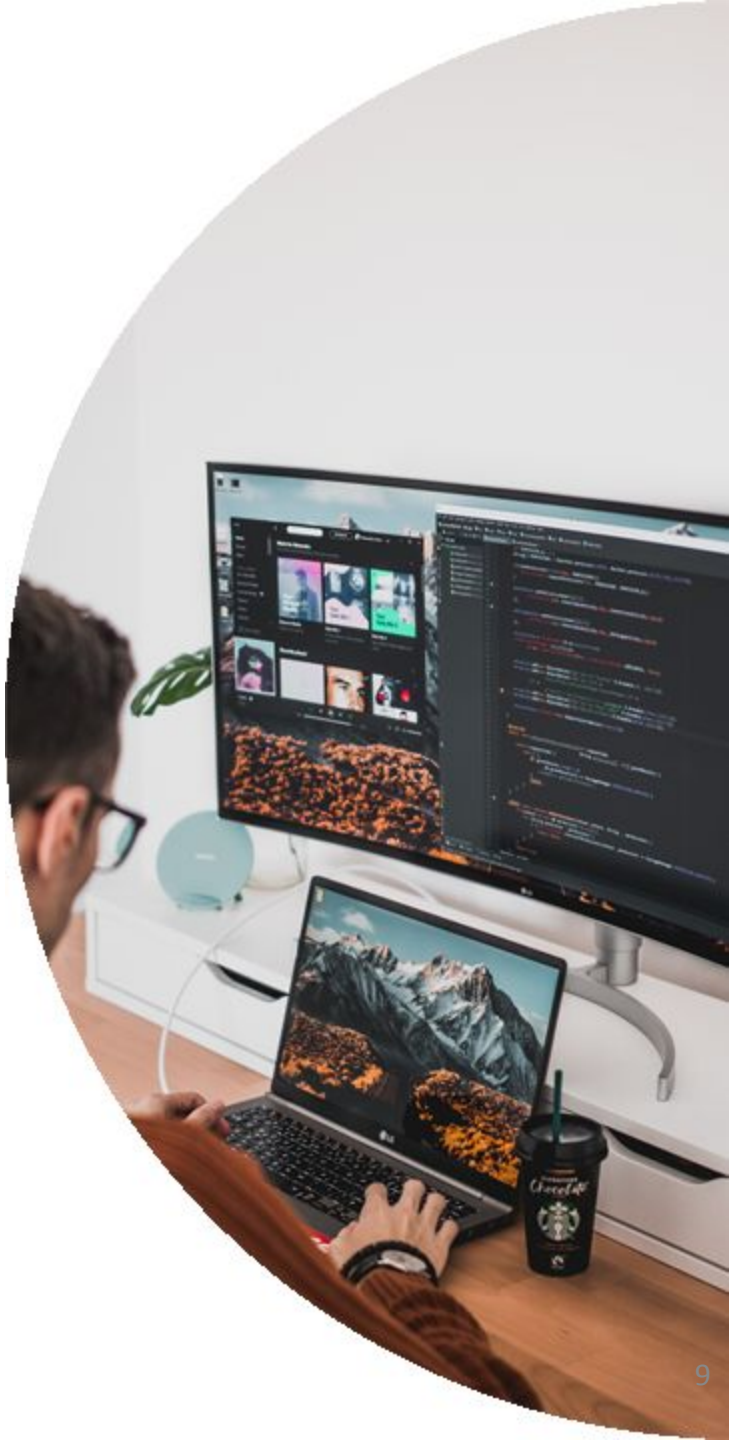
Melhorar o desempenho sempre impacta o custo, por isso, é importante determinar qual abordagem utilizar. Ex: Melhorar a infraestrutura custa menos do que melhorar o código. Outras vezes, melhorar o código compensa a economia em infraestrutura.

Definições gerais

Relação com a escalabilidade

- Performance significa o tempo necessário para atender determinada quantidade de demanda.
- Escalabilidade significa que o sistema suporta um aumento desta demanda.

Aspectos relativos à performance



Aspectos relativos à performance

Workload

Significa “carga de trabalho”.

Aspectos relativos à performance

Workload

É fundamental, para o design de sistemas com bom desempenho, conhecer qual será o tamanho do workload que deverá ser suportado.

Aspectos relativos à performance

Recursos computacionais

Conhecer as capacidades de processamento da infraestrutura.

Aspectos relativos à performance

Enfileiramento

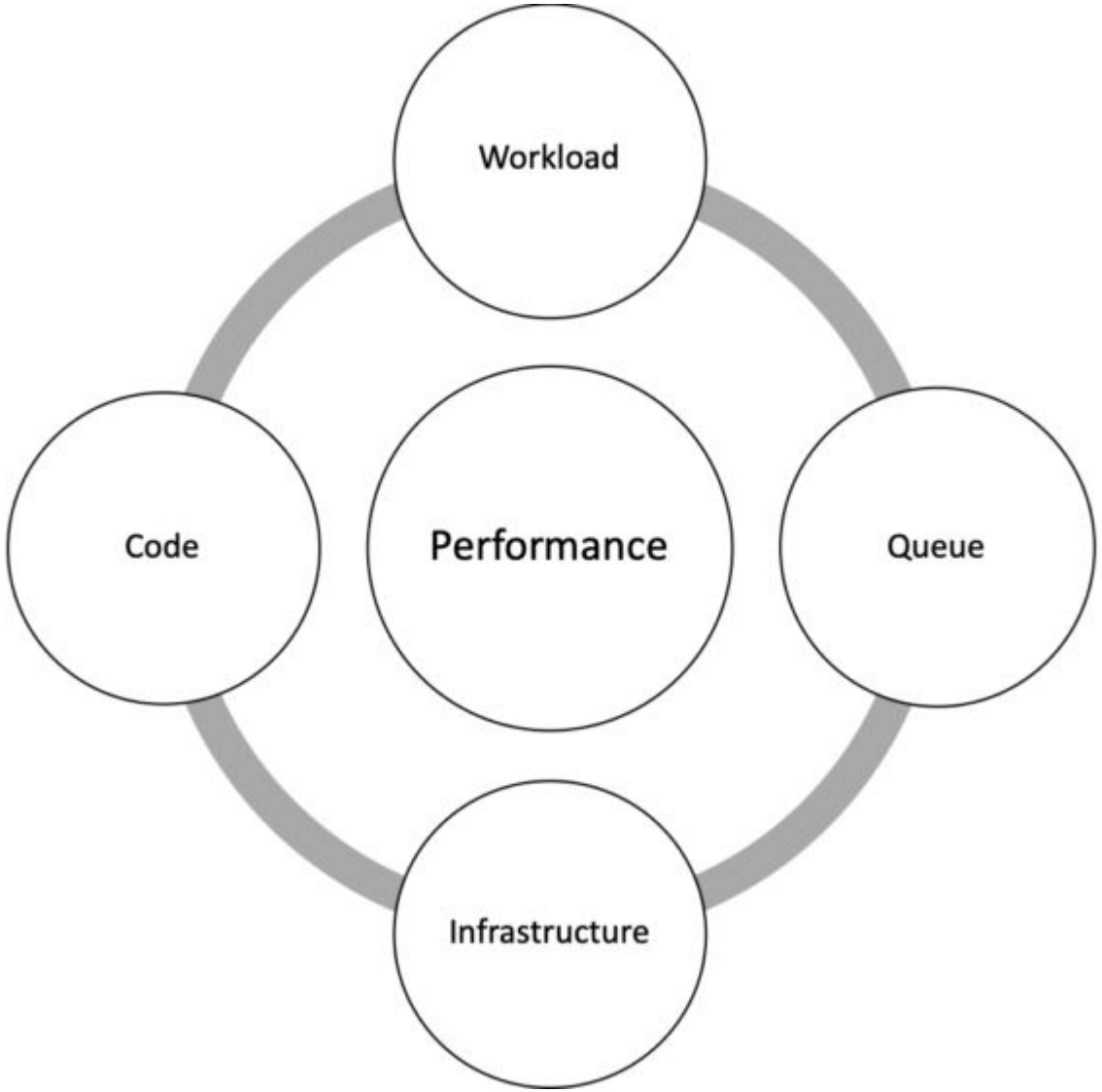
Permite tratar situações onde os recursos não estarão disponíveis em determinados momentos.

Aspectos relativos à performance

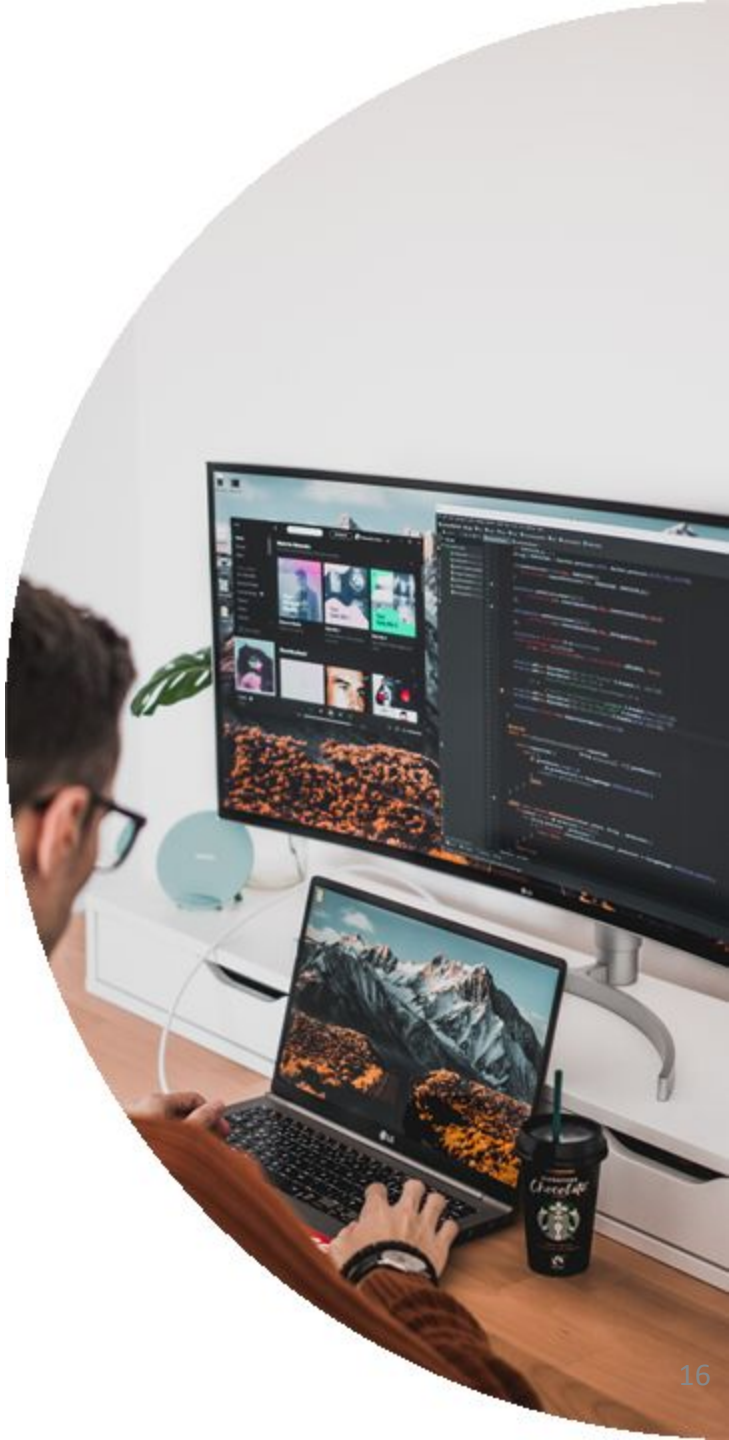
Código

O código fonte implementado deve fazer utilização correta dos recursos.

Aspectos relativos à performance



Obtenção de performance



Obtenção de performance

Priorização de requests

Existem situações onde deve-se separar os recursos computacionais para atender requisições mais importantes.

Obtenção de performance

Rate limiters

A limitação de taxa é uma estratégia para limitar o tráfego de rede. Ela limita a frequência com que alguém pode repetir uma ação dentro de um determinado período de tempo.

Obtenção de performance

Rate limiters

Conceitualmente, a estratégia consiste em limitar o throughput como forma de proteger o response time.

Obtenção de performance

Recursos computacionais

1) Fazer melhor uso dos recursos computacionais

Obtenção de performance

Recursos computacionais

2) Aumentar a quantidade de recursos computacionais disponíveis (efetivo, implica no incremento direto do custo).

Obtenção de performance

Caching

Seja para evitar consultas complexas para o banco de dados ou para armazenar o resultado de computação de alto custo, caching é uma forma simples de substituir recursos de custo elevado por outros mais baratos.

Obtenção de performance

Responsividade

Muitas vezes, é mais importante a percepção do usuário do que o próprio desempenho.

Obtenção de performance

Microotimização

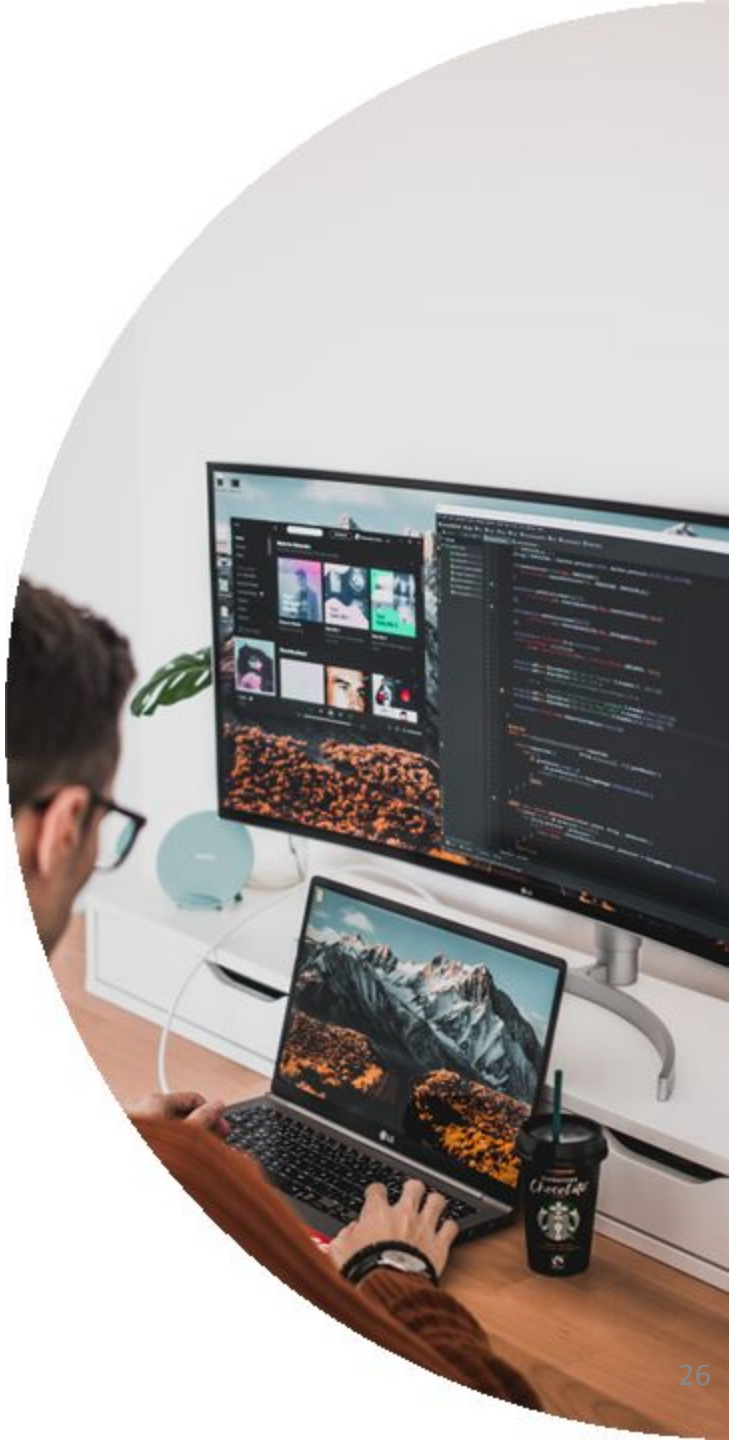
Há uma visão romântica de que a melhoria do desempenho ocorre “escovando bits“, entretanto, na prática, geralmente os ganhos mais percebidos acontecem por adaptações do design arquitetural.

Obtenção de performance

Microotimização

Ex: Adição de caching, buscando minimizar o uso de recursos com alto custo computacional, como a rede.

Fundamentos para sistemas seguros



Introdução

1. Definições gerais;
2. Detalhamento dos atributos de segurança;
3. Outros conceitos de segurança;
4. Recomendações de segurança;
5. LGPD.

Definições gerais

A segurança é, sob o ponto de vista da arquitetura de software, um atributo de qualidade.

Definições gerais

Segurança precisa ser adequadamente descrito e priorizado de acordo com os possíveis impactos para o atendimento dos objetivos de negócio.

Definições gerais

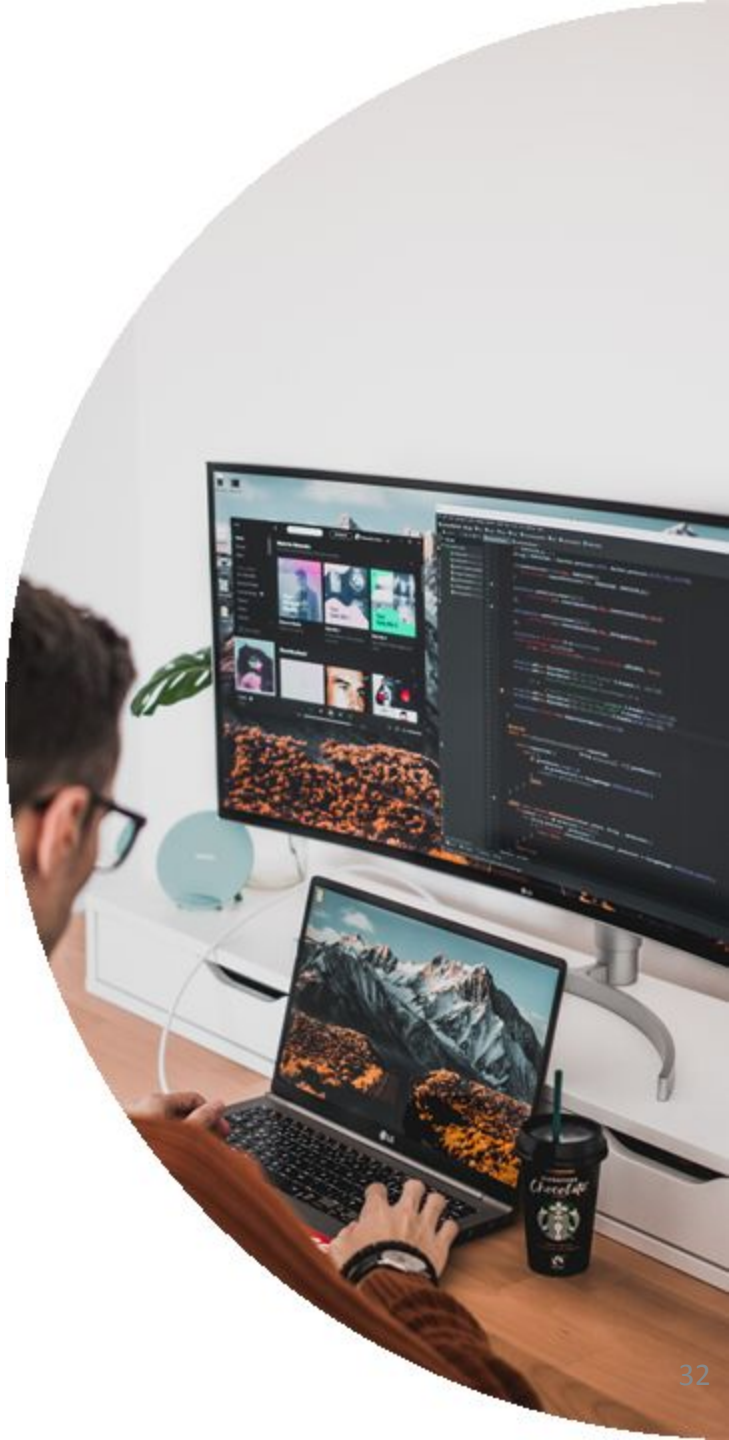
Origens dos problemas de segurança:

- Maldade;
- Inocência;
- Estupidez;
- Infortúnio humano.

Definições gerais

Segurança bem implementada desencoraja e coíbe condutas maliciosas, previne enganos e protege contra infortúnios.

Detalhamento dos atributos de segurança



Detalhamento dos atributos de segurança

Confidencialidade

Garantia de que o acesso a leitura e modificação de informações será restrito a aqueles que possuem autorização.

Detalhamento dos atributos de segurança

Integridade

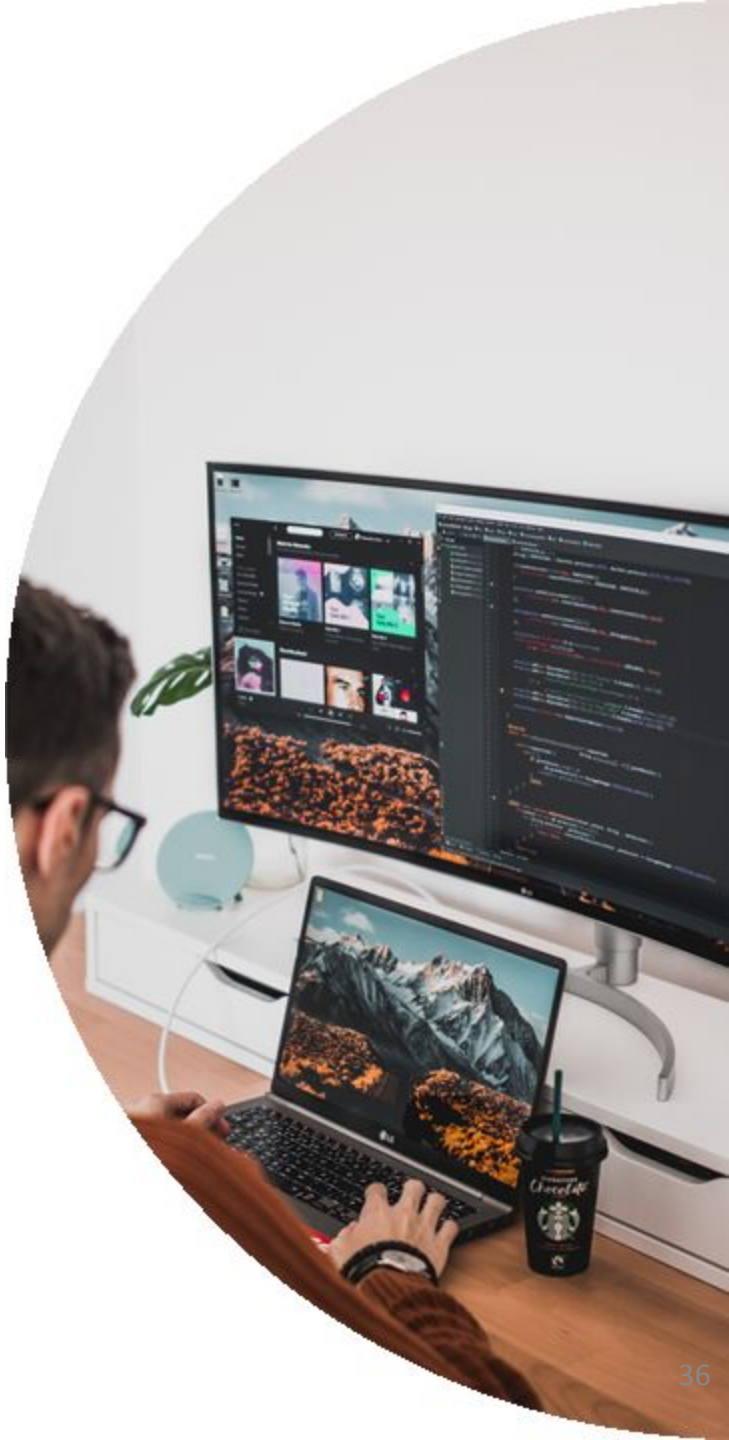
Indica que apenas dados válidos são aceitos.

Detalhamento dos atributos de segurança

Disponibilidade

Garante que dados permanecerão disponíveis quando necessários, frente a conduta maliciosa, inocência ou estúpida dos usuários.

Outros conceitos de segurança



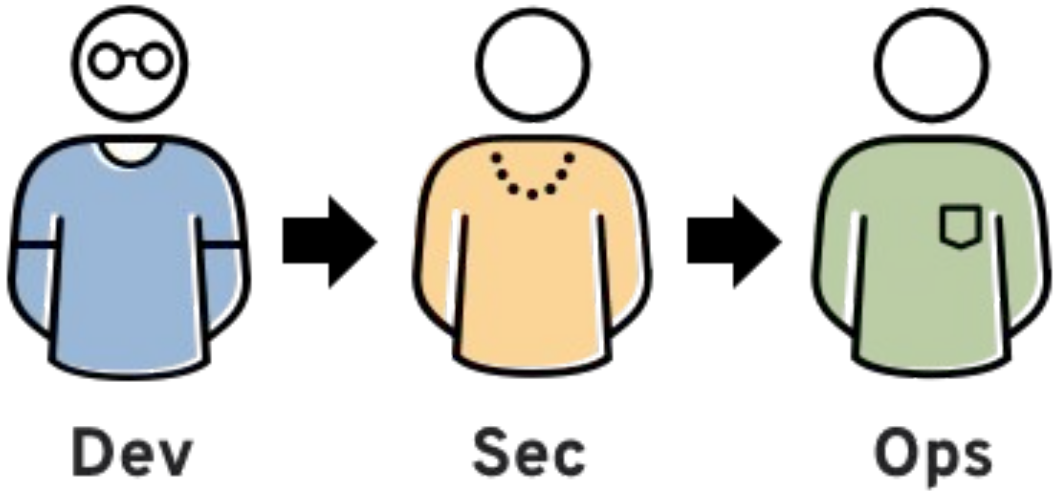
Outros conceitos de segurança

DevSecOps

DevSecOps significa desenvolvimento, segurança e operações. Trata-se de uma extensão da prática de DevOps.

Outros conceitos de segurança

DevSecOps



Outros conceitos de segurança

DevSecOps

DevSecOps é uma abordagem de segurança integrada, não apenas uma camada de proteção em torno de aplicações e dados.

Outros conceitos de segurança

SSDLC

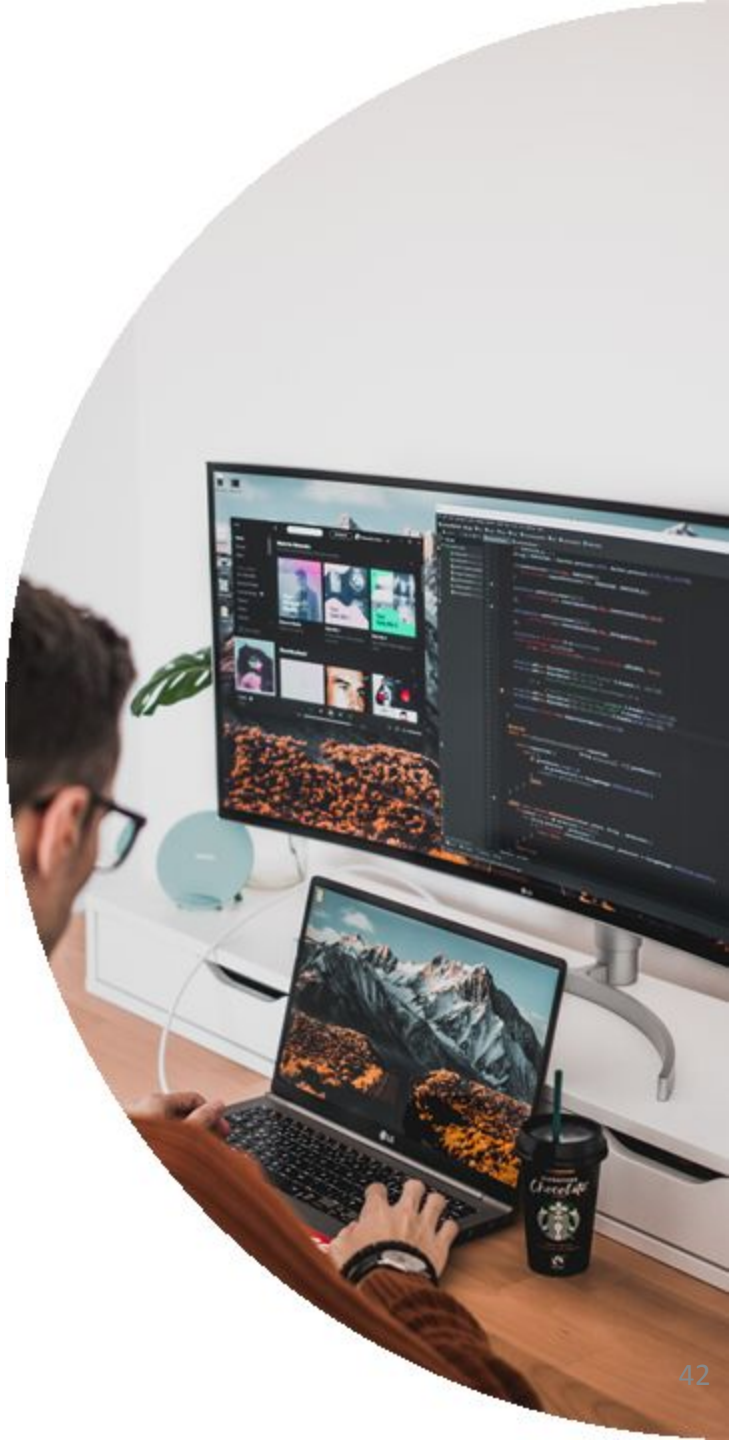
Secure Software Development Life Cycle

Outros conceitos de segurança

SSDLC - Secure Software Development Life Cycle

Artigo - SSDL

Recomendações de segurança



Recomendações de segurança

Padrões de Segurança

São diretrizes que ajudam a garantir que os microsserviços sejam desenvolvidos com as melhores práticas de segurança.

Recomendações de segurança

Redução de privilégios

Riscos de segurança são consideravelmente mitigados quando o projeto de interação do software restringe privilégios ao mínimo necessário, solicitando “elevação de autoridade” para operações com impactos mais altos.

Recomendações de segurança

Autenticação e Autorização

São fundamentais para garantir que apenas usuários autorizados tenham acesso aos recursos em um ambiente.

Recomendações de segurança

Estabelecer relações de confiança

Adotando certificados ou outros mecanismos de identificação.

Recomendações de segurança

Decompor sistemas em “contextos delimitados”

A decomposição de sistemas em componentes autônomos, além de facilitar o evolvability e a estruturação de times, é excelente estratégia para desenvolvimento de sistemas seguros.

Recomendações de segurança

Monitorar sistemas consistentemente

Todos os eventos sensíveis relacionados a segurança devem ser monitorados e logados em bases não violáveis.

Recomendações de segurança

Adotar práticas defensivas em diversos níveis de "profundidade

Componentes com criticidade para a segurança não devem "confiar" em verificações prévias, Por isso, implementam e avaliam solicitações de outros componentes seguindo alguma forma de controle de acesso.

Recomendações de segurança

Projetar segurança como se “ofensores” pudessem “ler o manual”

Não assumir que seus “segredos” estão seguros. Eventualmente, dados e estratégias se tornam públicos.

Recomendações de segurança

Segurança na Infraestrutura

A segurança na infraestrutura é uma prática que garante que os recursos de hardware e software em um ambiente estejam protegidos contra ameaças externas e internas.

Recomendações de segurança

Segurança na Infraestrutura

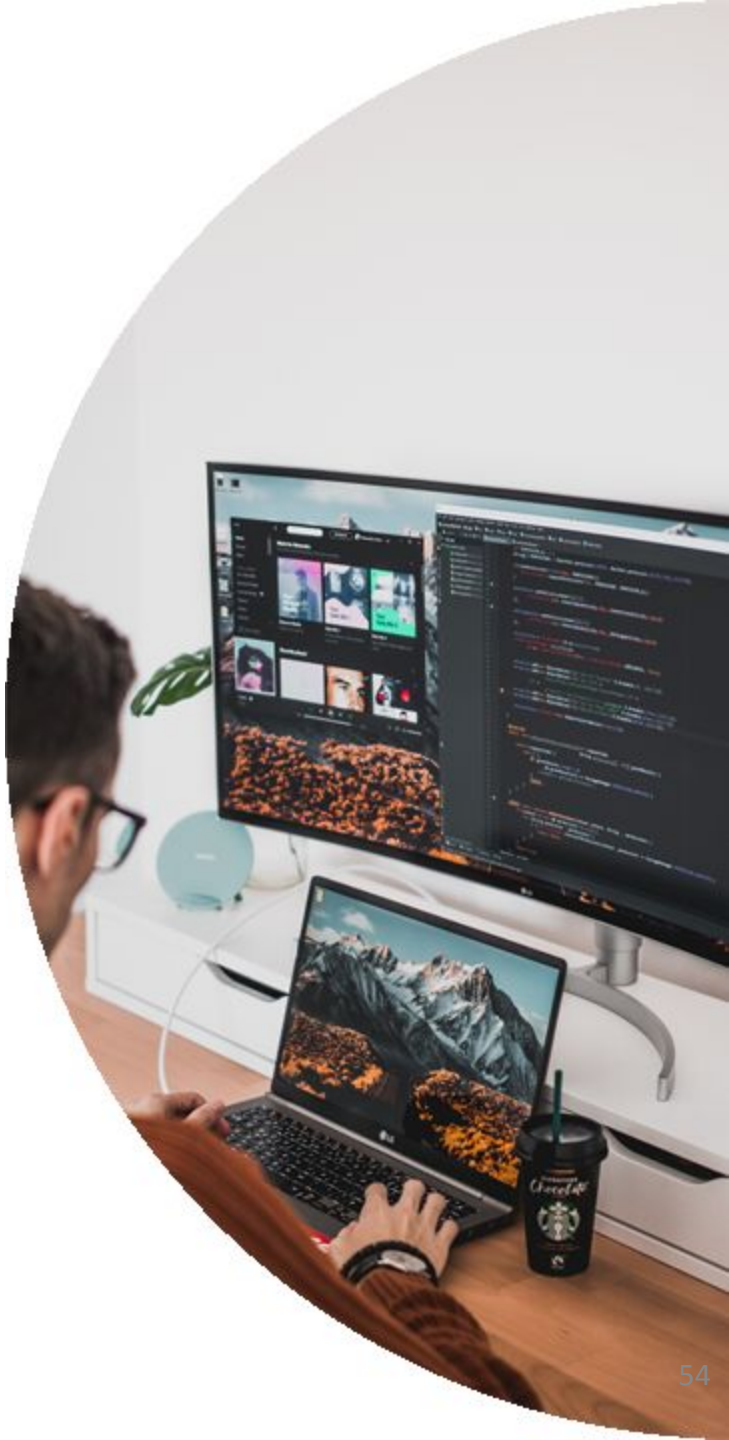
Isso inclui a adoção de políticas de segurança para a rede e a adoção de ferramentas de detecção de intrusão.

Recomendações de segurança

Testes de Segurança

Eles podem ser realizados através de ferramentas de testes de penetração, que simulam ataques para identificar vulnerabilidades e falhas de segurança.

LGPD



LGPD

Artigo: Recomendações de Boas Práticas para Implementação da LGPD em Processos de Desenvolvimento de Software

LGPD

Tópicos interessantes:

PG 14 - Lei Geral de Proteção de Dados

PG 22 - Privacy by design

PG 27 - Ciclo de desenvolvimento com privacy by design

Encerramento

- Manual do arquiteto de software, Elemar Júnior.
- Segurança em Microsserviços, Leandro Lopes