

# Modelagem de Sistemas

Guilherme Henrique Pasqualin Algeri

[guilherme.algeri@sistemafiep.org.br](mailto:guilherme.algeri@sistemafiep.org.br)

# Portabilidade

# Portabilidade

Portabilidade pode ser definida como a facilidade na qual o software pode ser transferido de um sistema computacional ou ambiente para outro

# Portabilidade

Em outras palavras, o software é dito portátil se ele pode ser executado em ambientes distintos

# Portabilidade

Note que o termo ambiente pode referir-se tanto à plataforma de hardware quanto a um ambiente de software como, por exemplo, um sistema operacional específico

# Portabilidade

De um modo geral, a portabilidade refere-se à habilidade de executar um sistema em diferentes plataformas

# Portabilidade

É importante observar que à medida que aumenta a razão de custos entre software e hardware, a portabilidade torna-se cada vez mais importante

# Portabilidade

Adicionalmente, podemos ter a portabilidade de componentes e a portabilidade de sistemas. Esta última situação pode ser vista como caso especial de reusabilidade



# Portabilidade

O reuso de software ocorre quando todo o sistema de software é reutilizado, implementando-o em diferentes sistemas computacionais

# Portabilidade

A portabilidade de um componente ou sistema de software é proporcional à quantidade de esforço necessário para que funcione num novo ambiente

# Portabilidade

Se uma quantidade menor de esforço é exigida quando comparada ao trabalho de desenvolvimento, então o sistema é dito portátil

# Portabilidade

Dois aspectos relevantes na portabilidade de programas são a transferência e adaptação

# Portabilidade

**A transferência** é o movimento de componente (código de um programa e dados associados) de um ambiente para outro

# Portabilidade

**A adaptação** engloba as modificações exigidas para fazer com que o programa seja executado em um novo ambiente

# Portabilidade

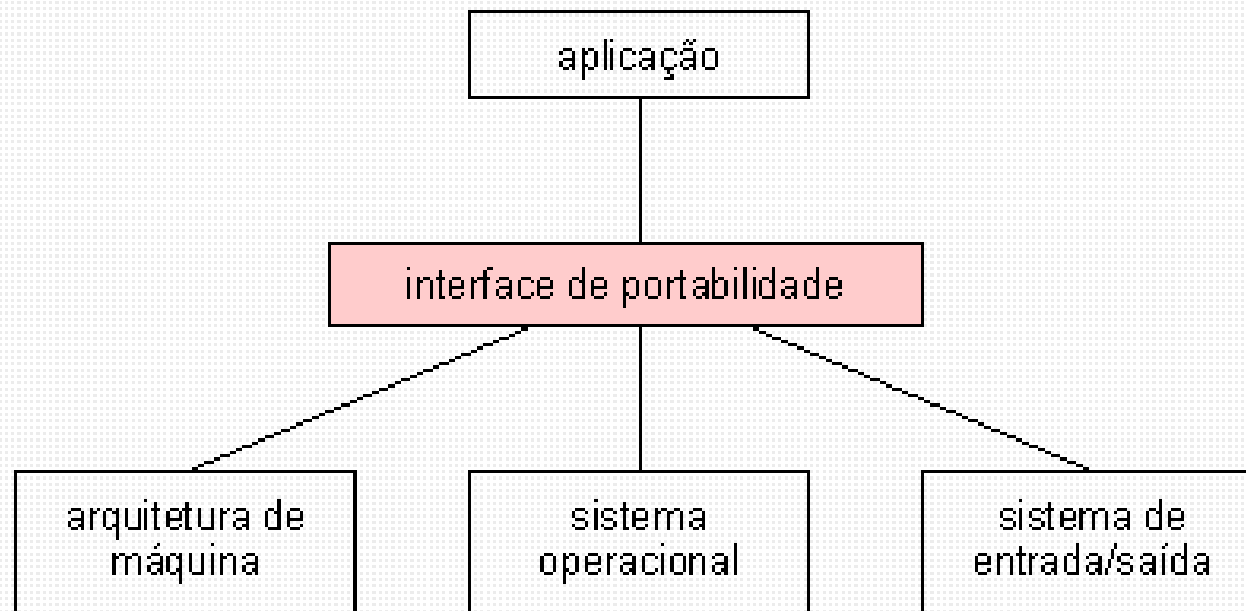
Cabe salientar que o ambiente no qual um sistema de software opera é normalmente composto de hardware, sistema operacional, sistema de entrada e saída (E/S), bem como a aplicação

# Portabilidade

Uma abordagem geral que poderia ser adotada para obter um sistema portátil tentaria separar as partes do sistema que dependem do ambiente externo numa camada ou interface de portabilidade, conforme ilustrado na **Figura 5**



# Portabilidade



**Figura 5.** Separação de partes de um sistema computacional.

# Portabilidade

A interface de portabilidade mostrada na figura acima poderia ser vislumbrada e projetada como um conjunto de tipos de dados abstratos ou objetos, os quais iriam encapsular os elementos não portáveis procurando esconder características do software da aplicação

# Portabilidade

Assim, quando o sistema de software muda de hardware ou sistema operacional, apenas a interface de portabilidade precisaria ser alterada

# Portabilidade

Assim, quando o sistema de software muda de hardware ou sistema operacional, apenas a interface de portabilidade precisaria ser alterada

# Reusabilidade

# Reusabilidade

Uma característica das engenharias é fazer uso de projetos existentes a fim de reutilizar componentes já desenvolvidos, objetivando minimizar o esforço em novos projetos

# Reusabilidade

Dessa forma, componentes que já tenham sido desenvolvidos e testados podem ser reutilizados. Considere os elevados níveis de reusabilidade que encontramos tanto na indústria de automóveis quanto de aparelhos eletrônicos

# Reusabilidade

Na indústria de automóveis, por exemplo, um motor é geralmente reutilizado de um modelo de carro para outro



# Reusabilidade

Em Engenharia de Software, à medida que aumenta a pressão para reduzir custos de desenvolvimento e manutenção de sistemas de software, bem como pela obtenção de sistemas com qualidade elevada,

# Reusabilidade

torna-se necessário considerar a reusabilidade como requisito não funcional no desenvolvimento de novos sistemas

# Reusabilidade

O reuso pode ser visto sob diferentes perspectivas. Ele pode ser orientado a componentes, orientado a processos ou orientado ao conhecimento específico de um domínio

# Reusabilidade

Note que ainda poderíamos considerar o reuso de requisitos. Entretanto, aqui, iremos concentrar nossa atenção no reuso orientado a componentes. Exemplos desse tipo de reuso são:

# Reusabilidade

- **Aplicação:** Toda a aplicação poderia ser reutilizada;
- **Subsistemas:** Os principais subsistemas de uma aplicação poderiam ser reutilizados;

# Reusabilidade

- **Objetos ou módulos:** Componentes de um sistema, englobando um conjunto de funções, podem ser reutilizados;
- **Funções:** Componentes de software que implementam uma única função (como uma função matemática) podem ser reutilizados.

# Reusabilidade

Dois tipos de reuso que estamos mais interessados são o reuso de subsistemas e objetos ou componentes, que chamaremos, simplesmente, de reuso de componentes

# Reusabilidade

Este tipo de reuso não envolve apenas o código, mas também engloba a arquitetura e projeto associados



# Reusabilidade

É importante observar que podemos obter ganhos se reutilizarmos tanto projetos quanto arquiteturas. Isto minimiza esforços de desenvolvimento e requer menos alterações ou adaptações

# Reusabilidade

Na realidade, quando se tem em mente que é necessário prover suporte à fácil modificação, indiretamente, obtemos componentes reutilizáveis

# Reusabilidade

Assim, o requisito reusabilidade pode envolver a arquitetura de um sistema de software ou componentes deste

# Reusabilidade

O que determinará quão fácil será conseguir componentes reutilizáveis é a interdependência ou acoplamento entre os componentes

# Reusabilidade

Por exemplo, uma biblioteca de componentes que oferece um conjunto de funcionalidades já implementadas e testadas oferece ao usuário (programador) um recurso valioso,

# Reusabilidade

já que basta incorporar esses componentes ao seu código e acessar suas funcionalidades através de sua interface

# Segurança

# Segurança

Em um sistema de software, este requisito não funcional caracteriza a segurança de que acessos não autorizados ao sistema e dados associados não serão permitidos



# Segurança

Portanto, é assegurada a integridade do sistema quanto a ataques intencionais ou acidentes. Dessa forma, a segurança é vista como a probabilidade de que a ameaça de algum tipo será repelida

# Segurança

Adicionalmente, à medida que os sistemas de software tornam-se distribuídos e conectados a redes externas, os requisitos de segurança vão se tornando cada vez mais importantes.

Exemplos de requisitos de segurança são:

# Segurança

- Apenas pessoas que tenham sido autenticadas por um componente de controle acesso e autenticação poderão visualizar informações dado que a confidencialidade permite esse tipo de acesso apenas às pessoas autorizadas

# Segurança

- As permissões de acesso ao sistema podem ser alteradas apenas pelo administrador de sistemas

# Segurança

- Deve ser feito cópias (backup) de todos os dados do sistema a cada 24 horas e estas cópias devem ser guardadas em um local seguro, sendo preferencialmente num local diferente de onde se encontra o sistema

# Segurança

- Todas as comunicações externas entre o servidor de dados do sistema e clientes devem ser criptografadas

# Segurança

Requisitos de segurança são essenciais em sistemas críticos, tais como sistemas de controle de voo de aeronaves, uma vez que é impossível confiar num sistema deste tipo se ele não for seguro

# Segurança

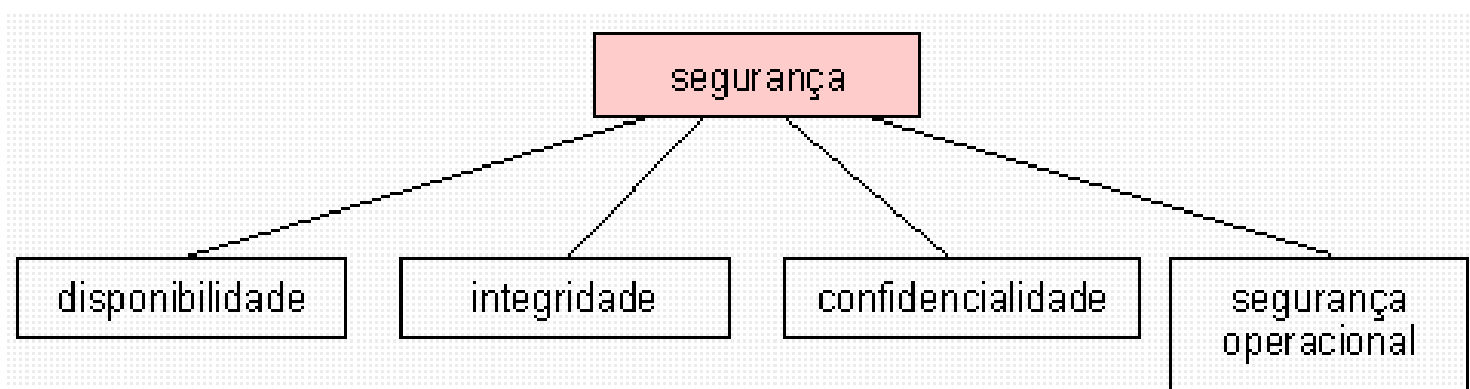
Assim, pode-se dizer que todos os sistemas críticos possuem associados a eles requisitos de segurança



# Segurança

Os requisitos não funcionais de segurança envolvem diferentes aspectos. A **Figura 6** mostra tipos de segurança que podemos encontrar

# Segurança



**Figura 6.** Tipos de segurança.

# Segurança

Também é importante considerar as diferentes ênfases encontradas para segurança:

# Segurança

- **Disponibilidade:** Refere-se a assegurar o sistema contra qualquer interrupção de serviço

# Segurança

- **Integridade:** O foco na integridade ocorre principalmente em sistemas comerciais, onde se busca assegurar que acesso ou atualizações não autorizadas ocorram

# Segurança

- **Confidencialidade:** A ênfase aqui é a de não permitir a revelação não autorizada de informações

# Segurança

- **Segurança operacional:** Refere-se à fase considerada para o sistema em uso

# Segurança

Note que para satisfazer ao requisito de qualidade não funcional de segurança em um sistema de software, alguns métodos podem ser empregados



# Segurança

Esses métodos podem ser vistos como um refinamento da meta de prover segurança a um sistema de software. Como exemplos desses métodos, podemos considerar:

# Segurança

## Identificação

Identifica o nome do usuário, informando ao sistema quem está utilizando-o

# Segurança

## Autenticação

Visa assegurar que os usuários são, de fato, quem afirmam ser. Para tanto, fazem um teste de identidade. Este método envolve alguns aspectos, tais como:

# Segurança

**Tipo de protocolo usado:** isto requer operação de senha

# Segurança

**Quantidade de autenticações:** pode requerer uma única senha ou múltiplas senhas ou procedimentos. Por exemplo, alguns bancos já fazem uso de múltiplas senhas durante operação de autenticação

# Segurança

**Partes envolvidas:** isto pode envolver a autenticação de uma parte envolvida (cliente) ou de ambas as partes envolvidas (cliente e sistema)

# Segurança

## Tempo de acesso

Busca limitar o tempo de acesso ao sistema a fim de reduzir qualquer tipo de ameaça

# Segurança

## Auditoria de segurança

Objetiva habilitar pessoal autorizado a monitorar o sistema e, seletivamente, rastrear eventos importantes



# Segurança

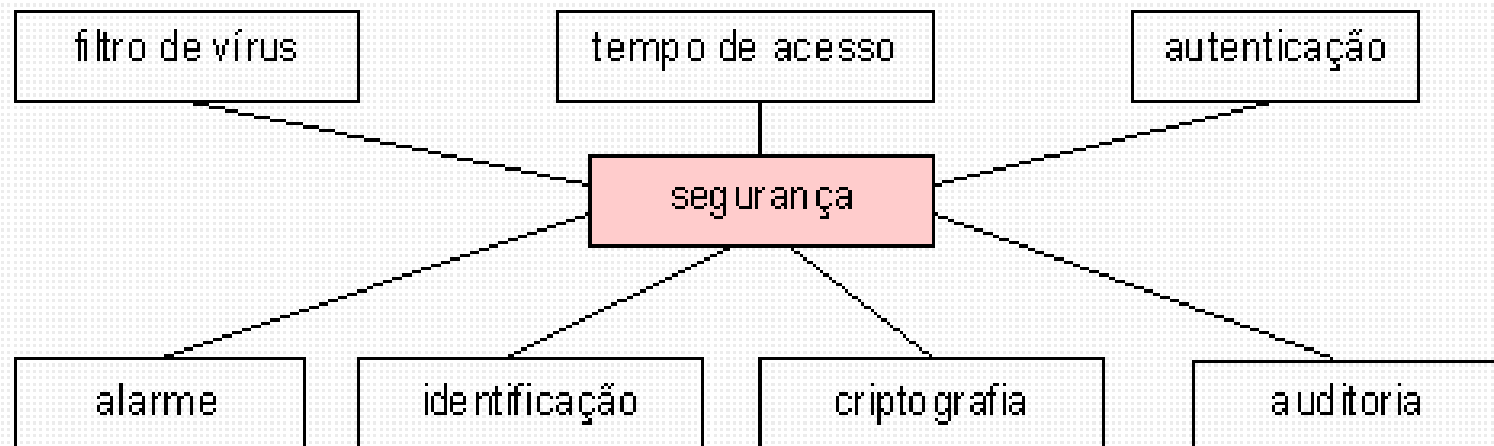
## Alarme

Esta operação visa prevenir acessos, potencialmente suspeitos às informações vitais ou dados do sistema, notificando esses acessos à supervisão de segurança do sistema ou devidas autoridades

# Segurança

**A Figura 7** apresenta alguns métodos que podem ser empregados em requisitos de segurança

# Segurança



**Figura 7.** Métodos usados para prover segurança

# Segurança

É importante observar que a arquitetura de um sistema de software deve levar em consideração esses aspectos a fim de atender aos requisitos de segurança

# Segurança

Entretanto, o tipo de sistema determinará quais fatores precisarão ser levados em conta

# Segurança

Assim, poderá haver a inserção de componentes específicos de segurança bem como a conexão destes com outros componentes funcionais

# Conclusão

# Conclusão

**A elicitação de requisitos funcionais e não funcionais é etapa fundamental no desenvolvimento de sistemas de software**



# Conclusão

Elementos de entrada desse processo de elicitação compreendem os principais influenciadores do sistema, envolvendo projetista, arquiteto e usuários

# Conclusão

Aliado a isso, a experiência do arquiteto de software é de grande importância

# Conclusão

Como saída deste processo, tem-se um conjunto de requisitos funcionais oferecendo suporte às funcionalidades do sistema e uma lista de requisitos não funcionais oferecendo suporte à arquitetura de software

# Conclusão

Cabe destacar que, quanto da análise de arquiteturas candidatas para um sistema de software, um arquiteto ou engenheiro de software **considera os requisitos não funcionais como um dos principais critérios para sua análise**

# Conclusão

Por fim, é importante notar que uma cobertura completa de todos os **possíveis requisitos não funcionais** está fora do objetivo desta aula

# Conclusão

Para tanto, você pode consultar o livro intitulado **NonFunctional Requirements in Software Engineering** de L. Chung, E. Yu, and J. Mylopoulos

# Conclusão

Todavia, um subconjunto dos mais **proeminentes requisitos não funcionais** foi apresentado

# Conclusão

Esses requisitos servem, via de regra, como critério para análise arquitetural objetivando a definição da arquitetura de software de um sistema



# Referências

# Conclusão

**Engenharia de Software / Ian Sommerville ;**  
tradução Ivan Bosnic e Kalinka G. de O.  
Gonçalves ; revisão técnica Kechi Hirama. — 9.  
ed. — São Paulo : Pearson Prentice Hall, 2011

# Obrigado!

Guilherme Henrique Pasqualin Algeri  
[guilherme.algeri@sistemapiep.org.br](mailto:guilherme.algeri@sistemapiep.org.br)  
(42) 9 9148-8117