

Engenharia de Software

11 - Segurança e Resiliência

Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

"A sabedoria protege, assim como o dinheiro protege, mas a vantagem do conhecimento é que a sabedoria dá vida a quem a possui." **Eclesiastes 7:12**

Conhecimento é uma proteção e uma vantagem duradoura. Estudar é fortalecer-se para a vida.

Objetivos

- Apresentar conceitos de **segurança**, **dependabilidade** e **resiliência** em sistemas de software, conforme tratados no capítulo do livro.

- Discutir por que esses aspectos são cada vez mais importantes no contexto de engenharia de software moderna (metodologias ágeis, sistemas distribuídos, “sistemas de sistemas”), conforme a 10ª edição enfatiza.
- Relacionar esses conceitos com exemplos práticos de falhas de segurança ou falta de resiliência em sistemas reais.

- Desenvolver no laboratório uma atividade de pequena análise de risco ou de design de mitigação de falhas para um sistema hipotético, reforçando o aprendizado ativo.

UniFAP

Introdução

Contextualização: A evolução da Engenharia de Software

Nas edições mais recentes de *Engenharia de Software*, Sommerville destaca que a disciplina **evoluiu além do desenvolvimento funcional** — ou seja, já não basta que o software “funcione corretamente”.

Hoje, espera-se que ele seja também:

- **Seguro (security):** protegido contra ataques, acesso indevido e perda de dados.
- **Confiável (dependable):** capaz de operar de forma previsível e correta.
- **Resiliente (resilient):** apto a continuar operando mesmo quando ocorrem falhas ou incidentes inesperados.

UniFAP

Introdução

Contextualização: A evolução da Engenharia de Software

Na 10ª edição, o autor reformulou o conteúdo sobre “Dependabilidade e Segurança” (capítulo 14, edições anteriores) para incluir **resiliência**, refletindo as novas demandas da engenharia moderna, marcada por sistemas distribuídos, integração contínua e operações ininterruptas.

Essas mudanças acompanham o cenário atual, em que **os sistemas estão mais interconectados, complexos e críticos** — de aplicativos financeiros a infraestruturas hospitalares e cidades inteligentes.

Uma falha de software, que antes podia ser apenas incômoda, hoje **pode comprometer segurança física, causar prejuízos econômicos e danos à reputação**.

Introdução
Definição de termos-chave

Para compreender o capítulo e o tema da aula, é importante diferenciar alguns conceitos centrais utilizados por Sommerville:

Termo	Definição (baseada no livro)	Exemplos
Dependabilidade (Dependability)	Conjunto de propriedades que refletem a confiança que podemos ter em um sistema. Inclui confiabilidade, disponibilidade, segurança (safety), integridade e manutenção.	Um sistema bancário que mantém precisão nos saldos mesmo com alto volume de acessos.
Segurança (Security)	Capacidade do sistema de **proteger dados e serviços contra acesso, modificação ou destruição não autorizada**. Engloba confidencialidade, integridade e disponibilidade (a tríade CIA).	Um sistema que usa autenticação multifatorial e criptografia de ponta a ponta.
Resiliência (Resilience)	Capacidade de um sistema de **continuar operando e se recuperar rapidamente após uma falha, ataque ou evento imprevisto**.	Um serviço de streaming que redireciona usuários automaticamente para servidores de backup quando um data center falha.

UniFAP

Introdução
Definição de termos-chave

Resumo da distinção:

- **Dependabilidade** é o *guarda-chuva conceitual* — o conjunto de qualidades que tornam o sistema confiável.

- **Segurança** protege o sistema contra ameaças intencionais.

- **Resiliência** garante que ele se mantenha disponível e se recupere diante de falhas, intencionais ou não.

UniFAP

Introdução
Por que isso importa hoje

Na visão de Sommerville, a segurança e a resiliência não são apenas características adicionais, mas requisitos fundamentais de engenharia em um cenário de sistemas abertos e interligados.

a) Sistemas conectados

- Aplicações modernas trocam dados constantemente pela internet (APIs, microserviços, integrações com terceiros).

- Um ataque ou falha em um componente pode comprometer toda a cadeia.

UniFAP

Introdução
Por que isso importa hoje

b) Computação em nuvem (Cloud)

- Serviços e dados estão hospedados em infraestrutura compartilhada.
- Exige mecanismos robustos de isolamento, redundância e recuperação automática.

c) Internet das Coisas (IoT)

- Milhares de dispositivos pequenos, com hardware limitado, conectados e potencialmente vulneráveis.
- Falhas em sensores ou gateways podem gerar riscos físicos e operacionais.

UniFAP

Introdução
Por que isso importa hoje

d) Alta disponibilidade e serviços críticos

- Sistemas hospitalares, bancários e de transporte **não podem parar**.

- Isso demanda design para falhas, redundância, balanceamento de carga e monitoramento contínuo.

A ênfase do capítulo está em como os **engenheiros de software devem planejar e projetar sistemas com essas qualidades desde o início do ciclo de vida**, e não apenas tratá-las como correções tardias ou medidas de segurança “extras”.

UniFAP

Introdução
Resumo

A introdução do capítulo mostra que **engenharia de software moderna = engenharia de confiança**.

O foco desloca-se do “funcionar corretamente” para “funcionar de forma segura, confiável e recuperável”.

Sommerville reforça que **essas propriedades são alcançadas por meio de práticas de engenharia rigorosas**, incluindo:

- Identificação e análise de riscos;
- Design orientado à tolerância a falhas;
- Adoção de padrões de segurança;
- Testes e validação contínuos;
- Planejamento de recuperação e contingência.

UniFAP

Conceitos de Dependabilidade e Segurança

A importância da dependabilidade

Segundo Sommerville, **dependabilidade** é um atributo fundamental da qualidade de um sistema.

Ela representa **a confiança que um usuário pode ter de que o software funcionará conforme o esperado**.

Na prática, um sistema dependável é aquele que:

- Cumpre sua função corretamente;
- Está disponível quando necessário;
- Protege os dados dos usuários;
- Tolera falhas sem comprometer sua operação.

UniFAP

Conceitos de Dependabilidade e Segurança

A importância da dependabilidade

> *“Sistemas críticos — como controle de tráfego aéreo, equipamentos médicos e transações bancárias — exigem níveis elevados de dependabilidade, pois falhas podem causar danos sérios.”* (Sommerville, 10ª ed.)

A dependabilidade não é um conceito isolado.
Ela é composta por **um conjunto de atributos**, que, quando equilibrados, produzem sistemas confiáveis e robustos.

UniFAP

Conceitos de Dependabilidade e Segurança

Atributos da dependabilidade

Sommerville descreve os principais atributos da dependabilidade como segue:

Atributo	Descrição	Exemplo
Confiabilidade (Reliability)	Probabilidade de o sistema realizar suas funções corretamente em um período específico.	Um servidor web que mantém 99,9% de uptime ao longo do mês.
Disponibilidade (Availability)	Grau em que o sistema está operacional e acessível quando necessário.	Plataforma de pagamentos disponível 24h, com mecanismos de failover.
Segurança (Safety)	Garantia de que o sistema não causará danos físicos ou ambientais, mesmo em caso de falha.	Sistema de controle de usina nuclear com desligamento seguro automático.

Conceitos de Dependabilidade e Segurança

Atributos da dependabilidade

Segurança da informação (Security)	Proteção contra acesso não autorizado e ataques maliciosos.	Autenticação forte e criptografia de dados sensíveis.
Integridade (Integrity)	Garantia de que dados e funções não sejam alterados de forma indevida.	Banco de dados que detecta e rejeita transações inconsistentes.
Manutenibilidade (Maintainability)	Facilidade de corrigir erros e adaptar o sistema a novas condições sem introduzir falhas.	Aplicativo modular que permite atualizações rápidas sem indisponibilidade.

UniFAP

Conceitos de Dependabilidade e Segurança

Atributos da dependabilidade

Trade-offs:

Um aumento em um atributo pode afetar outro. Por exemplo, adicionar múltiplas autenticações pode elevar a segurança, mas reduzir a disponibilidade (sistema mais lento).

A engenharia de software deve equilibrar esses fatores conforme os requisitos do projeto.

Conceitos de Dependabilidade e Segurança

Segurança (Security): um atributo essencial da dependabilidade

Na 10ª edição, Sommerville destaca que segurança é tanto um requisito funcional (o sistema deve restringir acesso) quanto um requisito de qualidade (deve proteger contra falhas externas e internas).

A segurança é vista como parte integrante da dependabilidade, mas com foco em ameaças deliberadas (ataques, invasões, sabotagens).

Conceitos de Dependabilidade e Segurança

Segurança (Security): um atributo essencial da dependabilidade

Princípio	Significado	Exemplo prático
Confidencialidade (Confidentiality)	Garantia de que apenas usuários autorizados podem acessar informações.	Dados de pacientes acessíveis apenas a médicos e enfermeiros autenticados.
Integridade (Integrity)	Garantia de que os dados não serão modificados ou corrompidos indevidamente.	Sistema de notas fiscais que impede alteração de valores emitidos.
Disponibilidade (Availability)	Garantia de que o sistema e os dados estarão acessíveis quando necessários.	Serviços bancários online com infraestrutura redundante.

Enquanto a “Safety” lida com falhas acidentais, a “Security” trata de ameaças intencionais.

UniFAP

Conceitos de Dependabilidade e Segurança

Tipos de ameaças à segurança

Sommerville classifica as **ameaças à segurança de software** em três grandes categorias:

Tipo de Ameaça	Descrição	Exemplo
Ameaças de Confidencialidade	Tentativas de acesso ou interceptação não autorizada de dados.	Vazamento de senhas ou espionagem via rede.
Ameaças de Integridade	Modificação indevida de informações ou código.	Alteração maliciosa de registros em banco de dados.
Ameaças de Disponibilidade	Impedimento do acesso legítimo ao sistema.	Ataques DDoS que sobrecarregam servidores.

UniFAP

Conceitos de Dependabilidade e Segurança

Tipos de ameaças à segurança

Essas ameaças podem se originar de **agentes externos (hackers, malware)** ou **internos (funcionários descuidados ou mal-intencionados)**.
Por isso, o projeto de sistemas deve incluir **mecanismos preventivos e corretivos** — como autenticação forte, controle de acesso, auditoria e planos de contingência.

UniFAP

Conceitos de Dependabilidade e Segurança

Engenharia de dependabilidade

1. **Identificação de riscos** – detectar possíveis falhas, vulnerabilidades e ameaças.
2. **Análise de impacto** – avaliar o efeito de cada risco no sistema e nos usuários.
3. **Definição de requisitos de dependabilidade** – estabelecer níveis mínimos de confiabilidade, disponibilidade e segurança.
4. **Design para dependabilidade** – incluir redundância, tolerância a falhas, monitoramento e recuperação.
5. **Verificação e validação** – testar não apenas funcionalidades, mas também a resistência a falhas e ataques.

A dependabilidade deve ser tratada como **requisito de engenharia desde o início**, e não adicionada como “camada de segurança” após o sistema estar pronto.

UniFAP

Quais sistemas que vocês usam diariamente demonstram alta dependabilidade?

Já usaram algum serviço que “caiu” em um momento crítico?

O que poderia ter sido feito para evitar a falha?

Vocês acham que segurança e confiabilidade devem ser tratadas pelo mesmo time de desenvolvimento?

UniFAP

Sem dependabilidade, não há credibilidade; sem segurança, não há continuidade.

E em um mundo cada vez mais digital e interconectado, essas duas propriedades definem o sucesso — ou o fracasso — de qualquer sistema de software.

Resiliência de Sistemas

O que é Resiliência em Engenharia de Software

Resiliência é uma propriedade fundamental de sistemas dependáveis modernos.

Enquanto confiabilidade e segurança buscam evitar falhas, a resiliência busca lidar com elas quando inevitavelmente acontecem.

Definição:

“Um sistema é resiliente quando consegue continuar a operar, talvez em capacidade reduzida, mesmo na presença de falhas ou ataques, e consegue se recuperar rapidamente.” (Sommerville, 10ª ed.)

Em outras palavras, a resiliência não é apenas sobreviver à falha, mas se recuperar dela com o mínimo impacto possível.

UniFAP

Resiliência de Sistemas

Relação entre dependabilidade, segurança e resiliência

Dependabilidade	Prevenir e minimizar falhas.	Falhas acidentais.	Um sistema com validação de entrada para evitar erros de execução.
Segurança (Security)	Proteger contra ações maliciosas.	Ataques e acessos não autorizados.	Firewall, autenticação e criptografia.
Resiliência	Manter a operação mesmo diante de falhas e ataques.	Falhas inevitáveis ou imprevistas.	Sistema que troca automaticamente para servidor reserva quando o principal cai.

Síntese:

A dependabilidade constrói confiança, a segurança protege, e a resiliência mantém o sistema vivo quando algo dá errado.

UniFAP

Resiliência de Sistemas

Características de um sistema resiliente

Características-chave:

Tolerância a falhas (Fault tolerance):

- Capacidade de continuar operando mesmo quando um componente falha.
- Exemplo: um cluster de servidores que replica dados e redistribui tarefas automaticamente.

Recuperação (Recovery):

- Capacidade de restaurar o estado correto do sistema após uma falha.
- Exemplo: rollback de banco de dados após uma transação corrompida.

Redundância (Redundancy):

- Duplicação de componentes críticos (hardware, serviços, dados).
- Exemplo: sistemas em nuvem com instâncias de backup em regiões diferentes.

UniFAP

Resiliência de Sistemas

Características de um sistema resiliente

Monitoramento e detecção (Monitoring):

- Supervisão contínua do comportamento do sistema para identificar anomalias antes que causem falhas.
- Exemplo: alertas automáticos de uso anormal de CPU ou tráfego de rede.

Degradação graciosa (Graceful degradation):

- O sistema reduz suas funcionalidades sem parar completamente.
- Exemplo: serviço de streaming que diminui a qualidade do vídeo em caso de sobrecarga.

Capacidade de aprendizado (Adaptive resilience):

- Sistemas que se adaptam e aprendem com falhas passadas para evitar repetições.
- Exemplo: algoritmos de IA que ajustam rotas de tráfego baseados em incidentes anteriores.

UniFAP

Resiliência de Sistemas
Estratégias de projeto para resiliência

Sommerville descreve que a resiliência deve ser projetada desde o início, e não adicionada depois.

Ela faz parte da engenharia de sistemas críticos, e pode ser implementada com as seguintes abordagens:

a) Arquitetura redundante (N-Version Design)

Desenvolver múltiplas versões independentes do mesmo componente para que uma possa substituir a outra em caso de falha.

Exemplo: sistemas de controle de voo que executam três versões do mesmo software e votam no resultado correto.

UniFAP

Resiliência de Sistemas

Estratégias de projeto para resiliência

b) Sistemas tolerantes a falhas (Fault-tolerant systems)

Usam mecanismos como checkpointing e rollback para restaurar o estado consistente após falhas.

Exemplo: sistemas de banco de dados com logs de transações.

c) Recuperação em camadas (Layered recovery)

Cada camada do sistema tem mecanismos próprios de detecção e recuperação.
Exemplo: microserviços com supervisores locais que reiniciam apenas o componente afetado.

UniFAP

Resiliência de Sistemas

Estratégias de projeto para resiliência

d) Engenharia baseada em risco (Risk-driven design)

Decisões de arquitetura e priorização de requisitos são orientadas pelos riscos mais críticos identificados.

Exemplo: priorizar redundância em módulos de pagamento de um e-commerce, mas não em relatórios administrativos.

e) Testes de resiliência e caos (Chaos engineering)

Testar deliberadamente falhas no ambiente de produção controlada para verificar a capacidade de recuperação.

Exemplo: Netflix Chaos Monkey — ferramenta que desliga instâncias aleatórias para validar robustez.

UniFAP

Resiliência de Sistemas

Exemplos práticos de resiliência

Cenário	Falha	Estratégia de Resiliência
Sistema bancário online	Servidor principal indisponível	Failover automático para instância em outra região.
Aplicação de transporte urbano	Falha de API de localização	Uso de cache local com última posição conhecida.
Plataforma de vídeo	Sobrecarga no servidor de mídia	Balanceamento de carga e degradação de qualidade.
IoT de monitoramento ambiental	Perda de conexão com sensores	Armazenamento temporário dos dados localmente e sincronização posterior.

Resiliência de Sistemas

Engenharia de resiliência no ciclo de vida do software

Fase	Ações de resiliência
Requisitos	Identificar eventos que podem comprometer o sistema e definir requisitos de recuperação.
Design	Introduzir redundância, modularidade e isolamento de falhas.
Implementação	Adotar tratamento robusto de exceções, logs detalhados e testes de falhas.
Verificação e validação	Simular falhas e ataques para avaliar o comportamento do sistema.
Operação e manutenção	Monitorar continuamente e aplicar aprendizado de incidentes anteriores.

Quais falhas poderiam ocorrer (ex.: banco de dados fora do ar, ataque, sobrecarga)?

Quais estratégias de resiliência podem ser adotadas?

Resumo

- Falhas são inevitáveis; sistemas resilientes se preparam para elas.
- Resiliência complementa segurança e dependabilidade, garantindo continuidade do serviço.
- A engenharia moderna deve integrar **resiliência como requisito não funcional** e **como prática de projeto**.

“Sistemas complexos não podem evitar todas as falhas, mas podem ser projetados para resistir e se recuperar delas.”

UniFAP

Atividade

Em seus projetos:

- Identificar riscos relativos à segurança e dependabilidade (ex: ataques, queda do serviço, perda de dados).

- Propor mitigações: medidas de segurança (autenticação, criptografia), e de resiliência (backup, fail-over, replicação, arquitetura tolerante a falhas).

- Preparar um mini-relatório ou slides curtos (5 minutos) para apresentar ao final: principais riscos + soluções propostas.

Bibliografia

SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 15 ago 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/168127



Referências

CHATGPT. Expansão conceitual sobre integração entre COBIT, ITIL e PMBOK. OpenAI, 2025. Disponível em: https://chat.openai.com/. Acesso em: 11 ago. 2025.


UniFAP


OBRIGADO
Arigato
Spasibo
Obrigado
Dankon
Maake
Multumesc
Ua Tsaug Rau Koj
Mochchakkeram
Kiitos
Merci
Raibh Maith Agat
Chokrane
Grazie
Terma Kasih
Asante
Mamana
Kia Ora
THANK YOU
Jusparxar
Welalin
Kiitos
Grazie
Obrigado
Arigato
Spasibo
Merci
Chokrane
Grazie
Kiitos
Multumesc
Terma Kasih
Maake
Mochchakkeram
Kiitos
Grazie
Obrigado
Merci
Raibh Maith Agat
Kiitos
Multumesc
Kia Ora
Grazie
Kiitos
Obrigado
Grazie
Chokrane
Merci
Mochchakkeram
Grazie
Kiitos
Obrigado
Grazie
Multumesc
Merci
Mochchakkeram
Chokrane
Grazie