# INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ▶ FTF 2012/2 ▶ Questionários ▶ Dependabilidade: conceitos básicos ▶ Revisão da tentativa 1

# Dependabilidade: conceitos básicos

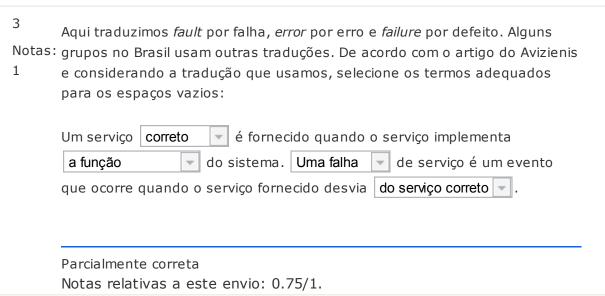
## Revisão da tentativa 1

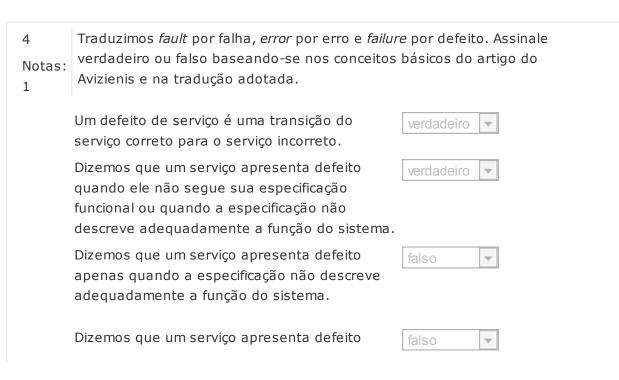
# Terminar revisão

Iniciado em	terça, 28 agosto 2012, 15:41
Completado em	quinta, 30 agosto 2012, 10:45
Tempo	1 dia 19 horas
empregado	
Notas	27.68/30
Nota	<b>92.28</b> de um máximo de 100( <b>92</b> %)

Considerando os conceitos básicos apresentados no artigo do Avizienis e 1 demais autores (A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, Notas: "Basic concepts and taxonomy of dependable and secure computing," 1 Dependable and Secure Computing, IEEE Transactions on, vol. 1, no. 1, pp. 11-33, 2004.), associe o conceito ao termo usado no artigo. Composto por um conjunto de estrutura do sistema componentes interligados, onde cada componente é outro sistema. O que é esperado que o sistema comportamento do sistema faça e é descrito na especificação funcional. Conjunto dos seguintes estado total estados: computação, comunicação, informação armazenada, interconexão e condição física. O que o sistema faz para função do sistema executar o esperado e é descrito por uma sequência de estados.

Parcialmente correta Notas relativas a este envio: 0.5/1. 2 Um serviço fornecido por um sistema, no seu papel de provedor de serviço, Notas: 1 Escolher a. o comportamento do sistema conforme percebido pelo usuário 🧹 uma resposta. 🔘 b. uma sequência de estados internos do sistema 🦹 🔘 c. a especificação funcional do sistema 🦹 🔘 d. a interface do sistema percebida pelo usuário ou outro sistema 🦹 🔘 e. uma execução correta do sistema 🦹 Correto Notas relativas a este envio: 1/1. 3 Aqui traduzimos fault por falha, error por erro e failure por defeito. Alguns Notas: grupos no Brasil usam outras traduções. De acordo com o artigo do Avizienis e considerando a tradução que usamos, selecione os termos adequados para os espaços vazios:





apenas quando ele não segue sua especificação funcional.

Correto

Notas relativas a este envio: 1/1.

Um defeito de serviço significa que ao menos um estado externo do sistema Notas: desvia do estado correto do serviço. Esse desvio é chamado de erro . A causa real ou suposta de um erro é chamada de falha .

Correto
Notas relativas a este envio: 1/1.

De acordo com o artigo de Avizienis, o período em que o sistema apresenta

Notas: um serviço incorreto é chamado de interrupção de serviço

transição de um serviço incorreto para correto é chamada de

restauração do serviço

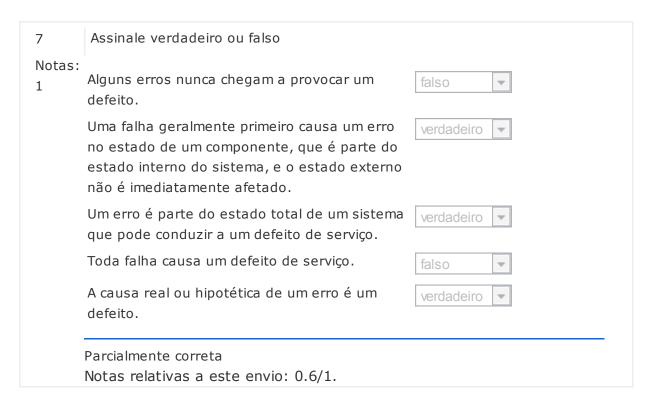
O desvio de um serviço correto pode assumir

diferentes formas, essas formas são chamadas

modos de defeito de serviço

Correto

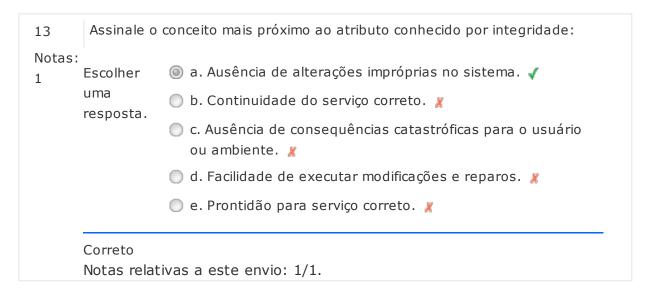
Notas relativas a este envio: 1/1.

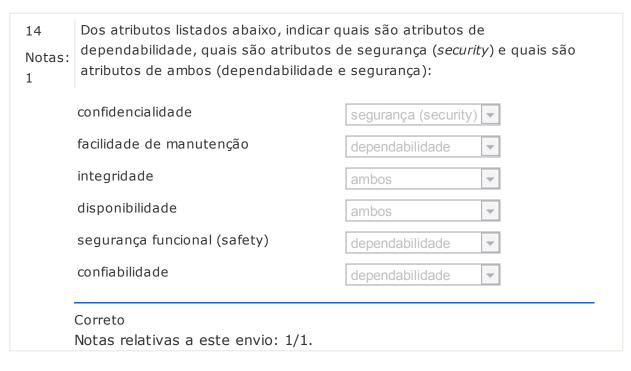


L	erro	no sistema. Uma falha externa cause um quando rro; caso contrário é dormente
	Correto Notas relat	ivas a este envio: 1/1.
9 Notas: 1	Assinale as	Avizienis apresenta duas definições para dependabilidade. s duas definições apresentadas no artigo:
1	Escolha	🗌 a. habilidade do sistema de fornecer um serviço correto 🦹
	pelo menos uma	☑ b. habilidade de evitar defeitos de serviço que sejam mais frequentes ou mais severos que o aceitável
	resposta.	$\hfill \Box$ c. habilidade de evitar erros no estado interno do sistema
		$\boxed{\hspace{-0.1cm} \ }$ d. habilidade do sistema de fornecer um serviço no qual se pode justificadamente confiar $\checkmark$
		🗌 e. habilidade de evitar falhas permanentes 🦹
	Correto Notas relat	ivas a este envio: 1/1.
10		conceito mais próximo a disponibilidade:
Notas: 1	Escolher uma	<ul> <li>a. Ausência de consequências catastróficas para o usuário ou ambiente.</li> </ul>
	resposta.	⊚ b. Prontidão para serviço correto. ∢
		🔘 c. Facilidade de executar modificações e reparos. 🦹
		🔘 d. Continuidade do serviço correto. 🦹
		🔵 e. Ausência de alterações impróprias no sistema. 🦹
	Correto Notas relat	ivas a este envio: 1/1.
11		conceito mais próximo ao atributo conhecido por confiabilidade:
Notas: 1	Escolher	⊚ a. Continuidade do serviço correto. 🗸
	uma	🔘 b. Prontidão para serviço correto. 🦹
	resposta.	🔘 c. Ausência de alterações impróprias no sistema. 🦹
		. Adsertica de dicerações improprias no sistema.
		d. Ausência de consequências catastróficas para o usuário ou ambiente.

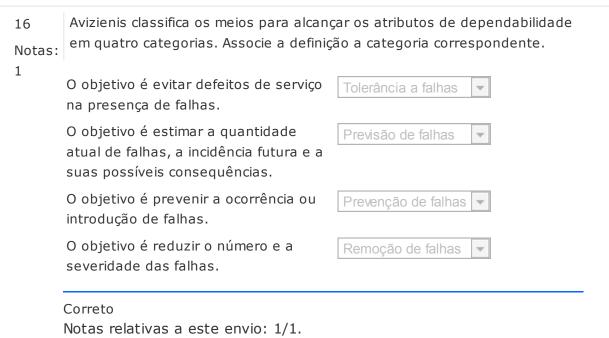
Notas relativas a este envio: 1/1.

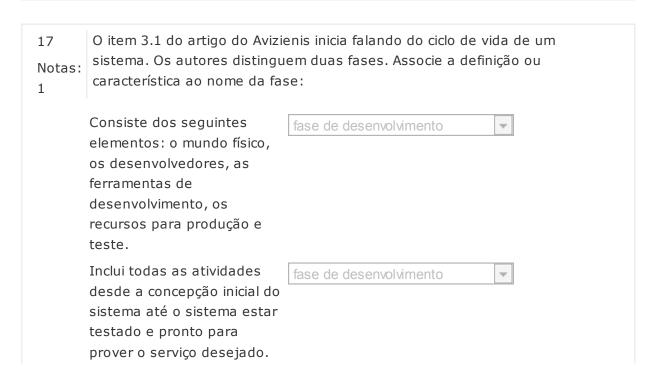
is:	funcional (	(safety):
u	Escolher	🔵 a. Prontidão para serviço correto. 🦹
	uma resposta.	🔘 b. Continuidade do serviço correto. 🦹
	resposta.	🔘 c. Ausência de alterações impróprias no sistema. 🦹
		🔵 d. Facilidade de executar modificações e reparos. 🦹
		<ul> <li>e. Ausência de consequências catastróficas para o usuário ou ambiente.</li> </ul>

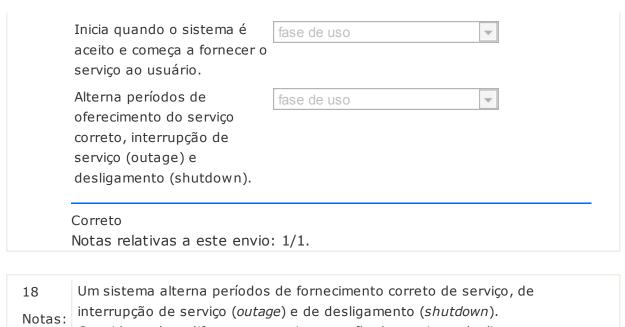


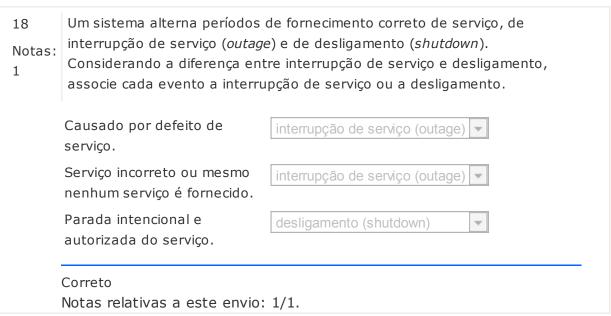


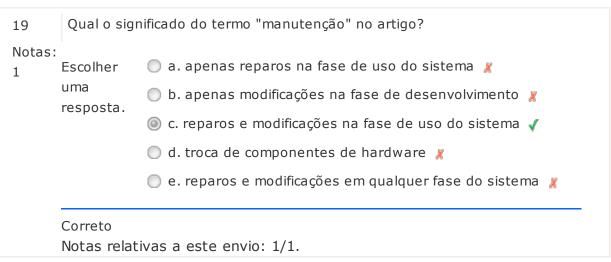
15 Vários meios foram desenvolvidos para atender os atributos de dependabilidade e segurança computacional (security). Avizienis e os Notas: demais autores do artigo classificam esses meios em quatro categorias 1 principais. Assinale entre as opções abaixo, uma opção que não corresponda a uma das quatro categorias: 🔘 a. tolerância a falhas 🦹 Escolher uma 🔘 b. prevenção de falhas 🦹 resposta. 🔘 c. remoção de falhas 🦹 ⑥ d. detecção de falhas 🔘 e. previsão de falhas 🦹 Correto Notas relativas a este envio: 1/1. 16 Avizienis classifica os meios para alcançar os atributos de dependabilidade

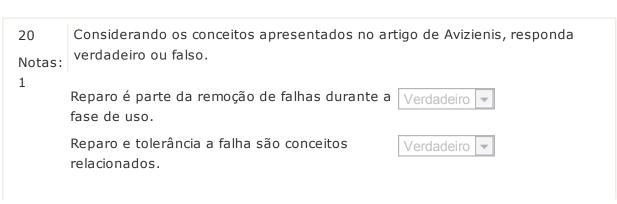












falha partic exem	é que toler tipação de ι	ância a falha ım agente e	nção e tolerânci s envolve a kterno, como po ste e reinstalaç	or	~	
Corre	to					
Notas	relativas a	a este envio	: 1/1.			

21 Responda com números decimais inteiros.
Notas:

1 No item em que Avizienis e os demais autores tratam da taxonomia de falhas (3.2.1), os autores classificam as falhas de acordo com 8 pontos de vista, também chamados de classes de falhas elementares. Se todas as classes pudessem ser combinadas, teríamos 256 diferentes classes de falhas combinadas. Entretanto, os autores identificaram apenas 31 combinações que fazem sentido, por enquanto.

Correto
Notas relativas a este envio: 1/1.

Assinale SIM para as classes de falha elementares e NÃO para os termos 22 que não correspondem as classes de falhas elementares sugeridas por Notas: Avizienis e demais autores. persistência SIM deterioração física NÃO limites SIM capacidade SIM interferência física NÃO objetivo SIM causa SIM errata NÃO fase NÃO intenção NÃO dimensão SIM permanência NÃO Parcialmente correta Notas relativas a este envio: 0.83/1.

23 Notas:	2 arupos r	combinações razoáveis de classes de falhas foram agrupadas em orincipais (com intersecção parcial). Quais são esses grupos?
1	Escolher uma	<ul> <li>a. falhas de especificação, de desenvolvimento e de operação </li> </ul>
	resposta.	🔘 b. falhas físicas, humanas e ambientais 🦹
		🔘 c. falhas internas, externas e correlacionadas 🦹
		⊚ d. falhas de desenvolvimento, físicas e de interação  √
		<ul> <li>e. falhas simples, múltiplas simultâneas e múltiplas em cascata </li> </ul>
	Correto Notas relat	tivas a este envio: 1/1.
24 Notas:	denomina	ernas e externas são agrupadas na classe elementar de falhas da:
1	Escolher	🔵 a. dimensão 🦹
	uma resposta.	□ b. intenção  X
	resposta.	🔘 c. persistência 🦹
		🔘 d. causa 🦹
		⊚ e. limites      ✓
	Correto Notas relat	tivas a este envio: 1/1.
25 Notas:	falhas den	rmanentes e transientes são agrupadas na classe elementar de nominada:
1	Escolher	a. persistência      √
	uma	🔘 b. causa 🦹
	resposta.	🔘 c. dimensão 🦹
		◯ d. fase 🔏
		○ e. limites 🗶
	Correto Notas relat	tivas a este envio: 1/1.
26 Notas:	falhac dan	hardware e de software são agrupadas na classe elementar de nominada:
1	Escolher	🔘 a. persistência 🦹
	uma resposta.	○ b. fase 🔏

		⊚ c. dimensão
		🔵 d. causa 🦹
		◯ e. limites 🦹
	Correto	tivas a este envio: 1/1.
	Notas Telat	tivas a este envio. 1/1.
27 Notas:	distinguíve	numanas, ou falhas resultantes da ação de pessoas, são eis pelo objetivo. Considerando o <b>objetivo</b> da interação humana ema, quais são as duas classes básicas de falhas humanas?
	Escolher	🔘 a. permanente e transiente 🦹
	uma	b. maliciosa e não-maliciosa   √
	resposta.	🔘 c. deliberada e não-deliberada 🦹
		🔘 d. acidental e incompetência 🦹
		○ e. desenvolvimento e operacional
	Correto Notas relat	tivas a este envio: 1/1.
28		desenvolvimento não maliciosas podem ocorrer em hardware ou
Notas:	em softwa falhas, qua	desenvolvimento não maliciosas podem ocorrer em hardware ou ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?
Notas:	em softwa falhas, qua nome espa	ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um
Notas:	em softwa falhas, qua nome espe Escolher uma	ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?
Notas:	em softwa falhas, qua nome espe Escolher	ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?
Notas:	em softwa falhas, qua nome espe Escolher uma	ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?  a. desvio da especificação  b. revisão  como se chamam?
Notas:	em softwa falhas, qua nome espe Escolher uma	re. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?  □ a. desvio da especificação  □ b. revisão  □ c. errata  ✓
Notas:	em softwa falhas, qua nome espe Escolher uma resposta.	re. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?  □ a. desvio da especificação  □ b. revisão  □ c. errata  □ d. incompetência  □ d. incompetência
Notas:	em softwa falhas, qua nome espe Escolher uma resposta.	ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?  a. desvio da especificação  b. revisão  c. errata  d. incompetência  e. update da especificação  e. update da especificação
Notas: 1	em softwal falhas, qua nome esperimental falhas, qua nome esperimental falhas f	ire. Em hardware, principalmente em microprocessadores, essas ando descobertas depois que a produção começou, recebem um ecial. De acordo com Avizienis, como se chamam?  a. desvio da especificação  b. revisão  c. errata  d. incompetência  e. update da especificação  e. update da especificação

Notas relativas a este envio: 1/1.

Uma característica comum às falhas de interação é que, para terem sucesso, elas necessitam da presença prévia de:

Escolher ② a. uma vulnerabilidade ✓ uma resposta.

□ c. uma falha externa ✗ □ d. uma falha maliciosa ✗ ○ e. uma tentativa de intrusão ✗

Correto
Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como João Luiz Grave Gross (Sair)

FTF 2012/2

# Dependabilidade: defeitos e erros

# Revisão da tentativa 1

Iniciado em	terça, 4 setembro 2012, 18:49
Completado em	domingo, 9 setembro 2012, 21:17
Tempo empregado	5 dias 2 horas
Notas	30.3/31
Nota	<b>97.74</b> de um máximo de 100( <b>98</b> %)

Question1 Notas: 1

Traduzimos *fault* por falha e *failure* por defeito. Considere a situação em que um sistema foi projetado de acordo com sua especificação e se comporta rigorosamente de acordo com essa especificação. Mesmo assim acontece um desastre inaceitável do ponto de vista do usuário. De acordo com o item 3.3 do artigo do Avizienis, assinale**verdadeiro** ou **falso** para as sentenças que seguem:

O sistema apresenta defeito porque não está fornecendo o serviço esperado mesmo seguindo a sua especificação.	Verdadeiro	_
A especificação tem uma ou mais falhas.	Verdadeiro	•

Um defeito devido a uma falha de especificação é um conceito muito vago e subjetivo e por isso não se deve levar muito a sério uma especificação.	Falso	~
O sistema não apresenta defeito porque está de acordo com a especificação.	Falso	*
O sistema apresenta defeito e a especificação não está adequadamente descrevendo a função do sistema.	Verdadeiro	•

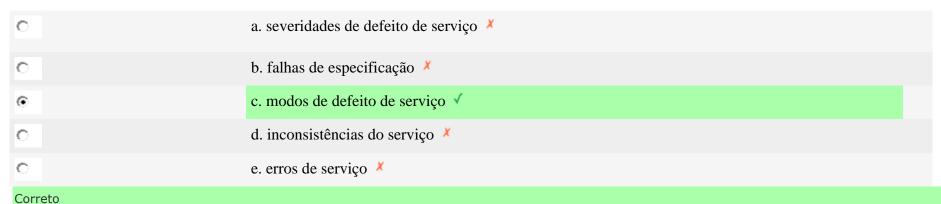
Notas relativas a este envio: 1/1.

# Question2

Notas: 1

De acordo com Avizienis, um defeito de serviço é um evento que ocorre quando o serviço fornecido desvia do serviço correto. As diferentes formas nas quais um sistema se desvia de um serviço correto são conhecidas como:

Escolher uma resposta.



Notas relativas a este envio: 1/1.

# Question3

Notas: 1

De acordo com Avizienis e demais autores, falhas de especificação podem ser de dois tipos. Assinale os dois tipos na lista a seguir:

Escolha pelo menos uma resposta.

	a. inconsequência 🗶
V	b. omissão ✓
	c. inadequação 🗶
V	d. comissão (commision) ✓
	e. incorreção 🗶

# Correto

Notas relativas a este envio: 1/1.

# Question4

Notas: 1

Considerando falhas de especificação, responda verdadeiro ou falso para as afirmações a seguir.

Falsa interpretação, suposições infundadas, inconsistências, erros tipográficos são exemplos de falhas de comissão.	verdadeiro	•
Um defeito pode ser subjetivo e questionável.	verdadeiro	┰
Um defeito é facilmente identificável e caracterizável.	falso	•

Notas relativas a este envio: 1/1.

Question5

Notas: 1

Modos de defeito de serviço caracterizam serviço incorreto de acordo com 4 pontos de vista. Assinale o ponto de vista que **não** pertence aos 4 listados no artigo de Avizienis.

Escolher uma resposta.

0	a. detectabilidade do defeito 🗶
0	b. consequência do defeito ×
0	c. domínio do defeito 🗶
0	d. consistência do defeito ×
•	e. custo do defeito ✓

### Correto

Notas relativas a este envio: 1/1.

Question6

Notas: 1

Considerando o ponto de vista do domínio do defeito, quais são os dois tipos de defeitos citados por Avizienis?

Escolher uma resposta.

a. silencioso e severo 🗡

•	b. conteúdo e temporização ✓
0	c. parada e errático 🗶
0	d. consistentes e bizantinos ×
0	e. sinalizado e não sinalizado 🗶

Notas relativas a este envio: 1/1.

Question7

Notas: 1

Considerando o domínio do defeito, quais são as duas classes de defeitos quando tanto a informação quanto a temporização são incorretas?

Escolher uma resposta.

•	a. defeitos de parada e errático ✓
0	b. adiantamento ou atraso ×
0	c. defeitos silenciosos e severos 🗶
0	d. defeitos sinalizados e não sinalizados 🗡
c	e. defeitos consistentes e bizantinos ×

# Correto

Notas relativas a este envio: 1/1.

Question8 Notas: 1 Considerando o domínio de defeitos, associe a descrição a classe de defeito:

caso especial onde nenhum serviço é fornecido na interface do sistema, ou no caso de sistemas distribuídos, nenhuma mensagem é enviada	defeito silencioso
o estado externo fica imutável, ou seja, a atividade do sistema não é mais percebida pelo usuário	defeito de parada ▼
o serviço é fornecido mas é incoerente	defeito errático
o serviço é fornecido muito cedo ou muito tarde	defeito de temporização ▼

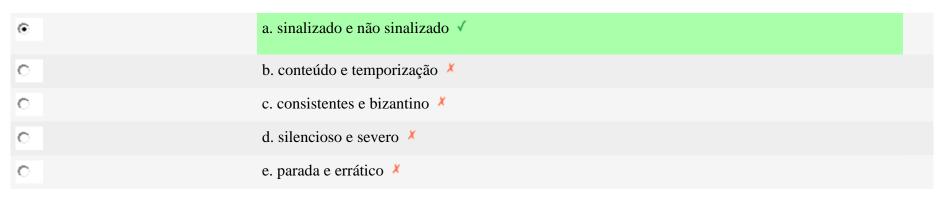
# Correto

Notas relativas a este envio: 1/1.

Question9

Notas: 1

Quais são os dois tipos de defeitos, considerando o ponto de vista da detectabilidade do defeito?



Notas relativas a este envio: 1/1.

Question10

Notas: 1

Considerando o ponto de vista da detectabilidade do defeito, os mecanismos de detecção apresentam dois modos de defeito. Associe as descrições aos termos apropriados:

Sinalização de perda de função quando nenhum defeito efetivamente ocorreu.

Nenuma indicação de perda de função quando o defeito ocorre.

Defeito não sinalizado

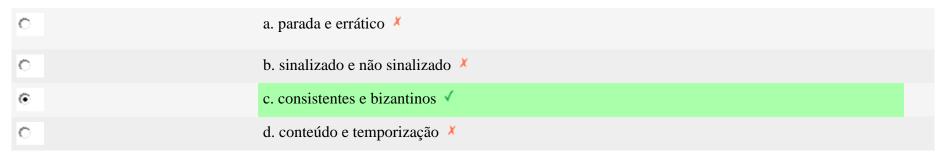
### Correto

Notas relativas a este envio: 1/1.

Question11

Notas: 1

Quais são os dois tipos de defeitos, considerando a consistência do defeito?



© e. silencioso e severo ✗

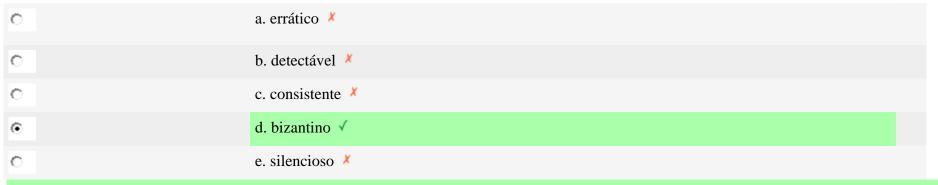
Correto

Notas relativas a este envio: 1/1.

Question12 Notas: 1

Alguns ou todos os usuários percebem o serviço incorreto de maneira diferente, alguns até podem perceber o serviço como correto. Esse tipo de defeito é chamado de:

Escolher uma resposta.



# Correto

Notas relativas a este envio: 1/1.

Question13 Notas: 1

Graduar as consequências de defeitos sobre o ambiente de um sistema permite definir a severidade do defeito. Aos seus níveis estão geralmente associadas probabilidades máximas aceitáveis de ocorrência. A quantidade, o nome e a definição dos níveis de

severidade de defeitos, assim como os níveis aceitáveis de probabilidades de ocorrência, são relacionados envolvem os atributos de dependabilidade e segurança (security) para a aplicação considerada.

à aplicação 🔻

### Correto

Notas relativas a este envio: 1/1.

Question14

Notas: 1

Considere os exemplos de critérios para determinação das classes de severidade de defeitos mencionados no artigo do Avizienis. Associe o critério ao atributo:

duração da interrupção do serviço	disponibilidade	▼
possibilidade de perdas de vidas humanas	segurança (safety)	▼
extensão da corrupção de dados e a capacidade de recuperar essa corrupção	integridade	•

### Correto

Notas relativas a este envio: 1/1.

Question15

Notas: 1

Sistemas que são projetados e implementados de tal forma que apresentam defeitos apenas em modos específicos descritos na especificação de dependabilidade e segurança (*security*) e apenas em uma extensão aceitável são sistemas com controle de defeito (*fail-controlled systems*). Seguindo a nomenclatura apresentada no artigo de Avizienis, associe a descrição ao termo apropriado:

sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos silenciosos fail-silent

sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos menores (minor failures)	fail-safe	Ţ
sistema onde os defeitos são todos (ou numa extensão aceitável) apenas defeitos da classe de parada	fail-halt ou fail-stop	-

Notas relativas a este envio: 1/1.

# Question16

Notas: 1

Falhas de desenvolvimento podem levar a defeitos parciais ou completos de desenvolvimento ou permanecer indetectáveis até a fase de uso do sistema. De acordo com Avizienis (item 3.3.2) existem dois aspectos relacionados a defeitos de desenvolvimento, são eles:

Escolher uma resposta.

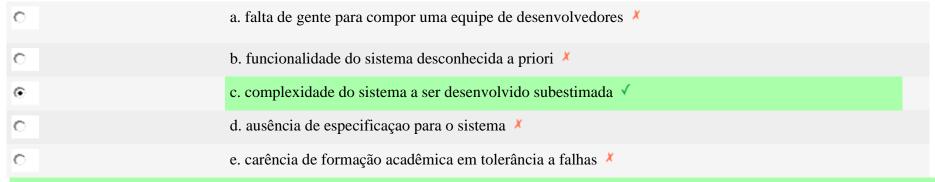
0	a. especificação com erro e alterações na especificação 🗡
0	b. tecnologia obsoleta e defeitos de desempenho 🗡
0	c. erros humanos e defeitos de produção 🗶
•	d. defeitos de orçamento e defeitos de escalonamento ✓
0	e. defeitos de ferramentas e interferências do ambiente X

# Correto

Notas relativas a este envio: 1/1.

Question17 Notas: 1 Avizienis e os demais autores citam várias causas para defeitos de desenvolvimento (item 3.3.2). Segundo eles, todas as causas citadas são geralmente devidas a:

Escolher uma resposta.



### Correto

Notas relativas a este envio: 1/1.

# Question18

Notas: 1

Assinale **verdadeiro** ou **falso** considerando o item 3.3.3 do artigo do Avizienis e demais autores.

Uma classe de falha de especificação de dependabilidade é a escolha injustificada de exagerados níveis de demanda para um ou mais atributos de dependabilidade, o que eleva os custos e pode levar a defeito de desenvolvimento.

Não é necessária uma especificação de dependabilidade, os desenvolvedores devem garantir por contrato que falhas não ocorram.

Uma especificação de dependabilidade e segurança (security) é um contrato que estabelece os objetivos de cada atributo: confiabilidade, disponibilidade, safety, confidencialidade, integridade e facilidade de manutenção.

Um defeito de dependabilidade ou segurança ocorre quando o sistema sofre defeitos de serviço mais frequentes ou mais severos do que aceitável.	Verdadeiro	•
Uma especificação de dependabilidade e segurança jamais contém falhas.	Falso	▼
Parcialmente correta		
Notas relativas a este envio: 0.8/1.		
Question19		
Notas: 2		
Localize-se no item 3.4 do artigo do Avizienis. Um erro é definido como parte do estado global do sistema que pode conduzir a ocorre quando o erro causa o desvio do serviço correto. A causa de um erro recebe o nome de	um defeito 🔻	que
Correto		
Notas relativas a este envio: 2/2.		
Question20 Notas: 1		
Escolha o termo apropriado:		
quando a presença do erro é indicada por uma mensagem de erro ou por um sinal de erro dizemos que o erro é	detectado	•
quando o erro está presente mas não é sinalizado dizemos que o erro é	latente	•
Correto		
Notas relativas a este envio: 1/1.		

Question21

### Notas: 1

Assinale verdadeiro ou falso de acordo com as opiniões expressas por Avizienis e demais autores.

### Correto

Notas relativas a este envio: 1/1.

### Question22

Notas: 1

A natureza da redundância que existe na estrutura de um sistema pode ser de dois tipos, segundo Avizienis:



C	d. de software ou de hardware 🗶		
•	e. protetora ou não intencional ✓		
Correto			
Notas relativas a este envio: 1/1			
Question23 Notas: 1			
Considerando possíveis classificações de err	ros citadas por Avizienis, assinale verdadeiro se a classificação aparece	no artigo, falso caso contrário.	
Erros podem ser classificados, considerar	ndo persistência, como erros permanentes ou erros transientes.	Verdadeiro	<b>T</b>
Erros podem ser classificados, em função	o dos danos financeiros provocados, como erros maliciosos ou não n	maliciosos. Falso	•
Erros podem ser classificados em termos	dos defeitos de serviço elementares que eles provocam.	Verdadeiro	-
Erros de conteúdo podem ser classificado afetados.	os, na área de códigos de controle de erros, em termos do padrão de	bits Falso	•
Parcialmente correta			
Notas relativas a este envio: 0.5,	/1.		
Question24 Notas: 1			
Considerando uma possível categorização d	le erros sugerida por Avizienis, associe o termo ao conceito:		
afetam apenas um componente		erro simples	•

são provocados por falhas que causam problemas simultâneos em mais do que um componente	erros múltiplos relacionados	•
Correto		
Notas relativas a este envio: 1/1.		
Question25 Notas: 1		
Localize-se no item 3.5 do artigo do Avizienis sob análise. Relacione o termo ao conceito.		
falha que produz um erro	falha ativa	_
falha que não produziu um erro (ainda)	falha dormente	<b>T</b>
Correto		
Notas relativas a este envio: 1/1.		
Question26 Notas: 1		
Relacione o termo ao conceito.		
aplicação de uma entrada a um componente que faz uma falha dormente se tornar ativa	ativação de falha	•
processo computacional que faz com que um erro seja sucessivamente transformado em outros erros	propagação de erro	•
Correto		
Notas relativas a este envio: 1/1.		

Question27

### Notas: 1

A habilidade de identificar o padrão de ativação de uma falha que causa um ou mais erros, de acordo com Avizienis, é chamada de:

Escolher uma resposta.

0	a. estímulo de ativação da falha 🗡	
•	b. reprodutibilidade da ativação da falha ✓	
0	c. probabilidade de ativar a falha 🗶	
0	d. identificação do padrão de ativação 🗡	
0	e. inevitabilidade de ativação de falha 🗡	

# Correto

Notas relativas a este envio: 1/1.

# Question28

Notas: 1

Falhas podem ser categorizadas de acordo com sua reprodutibilidade de ativação. Falhas cuja ativação é reproduzível são chamadas de:



0	d. evasivas ou soft ×
0	e. simples ×

Notas relativas a este envio: 1/1.

Question29

Notas: 1

Falhas podem ser categorizadas de acordo com sua reprodutibilidade de ativação. Falhas cuja ativação não é sistematicamente reproduzível são chamadas de:

Escolher uma resposta.

•	a. evasivas ou soft ✓
0	b. residuais ×
0	c. independentes X
0	d. intermitentes ×
0	e. sólidas ou hard 🗶

# Correto

Notas relativas a este envio: 1/1.

Question30

Notas: 1

Devido a similaridade de manifestação, dois tipos de falhas foram agrupadas com o nome de falhas intermitentes. São elas:

# Escolher uma resposta.

0	a. falhas de software e falhas de desenvolvimento de hardware 🗶	
•	b. falhas evasivas de desenvolvimento e falhas transientes de hardware 🗸	falhas permanentes e falhas transientes
0	c. falhas soft e falhas hard 🗶	
0	d. falhas naturais e falhas humanas 🗶	
0	e. falhas permanentes e falhas transientes 🗶	

Correto Notas relativas a este envio: 1/1.

# Dependabilidade: meios

# Revisão da tentativa 1

Iniciado em	segunda, 10 setembro 2012, 19:01
Completado em	segunda, 10 setembro 2012, 19:42
Tempo empregado	41 minutos 1 segundo
Notas	29.6/30
Nota	<b>98.67</b> de um máximo de 100( <b>99</b> %)

Question1 Notas: 1

Considerando o item 5 do artigo "Basic Concepts and Taxonomy of Dependable and Secure Computing", de Avizienis e demais autores, assinale **verdadeiro** para os termos que correspondem aos "meios" que foram mencionados no artigo para alcançar dependabilidade e **falso** caso contrário.

remoção de falhas	Verdadeiro	•
confidencialidade	Falso	•

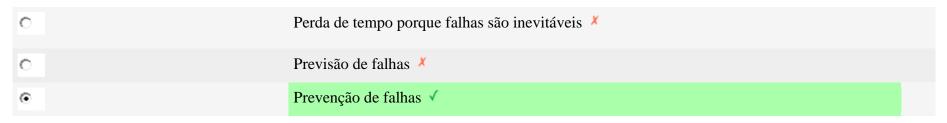
disponibilidade	Falso	<b>-</b>
prevenção de falhas	Verdadeiro	~
previsão de falhas	Verdadeiro	<b>T</b>
redundância	Falso	•
confiabilidade	Falso	•
tolerância a falhas	Verdadeiro	•

Notas relativas a este envio: 1/1.

# Question2

Notas: 1

Melhorar o processo de desenvolvimento para reduzir o número de falhas introduzidas no sistema durante sua produção é um exemplo de:



0	
0	Tolerância a falhas X
0	Remoção de falhas X
Correto	
Notas relativas a este envio: 1/1.	
Question3 Notas: 1	
Tolerância a falhas visa principalmente:	
Escolher uma resposta.	
0	aprender a conviver com defeitos ×
0	evitar falhas 🗶
•	evitar defeitos ✓
0	prevenir que falhas ocorram 🗡
0	remover falhas ×
Correto	
Notas relativas a este envio: 1/1.	

Question4

Notas: 1

De acordo com Avizienis, tolerância a falhas emprega duas estratégias básicas. São elas:

0	redundância de hardware e isolamento de falhas 🗡
0	compensação de erros e reinicialização 🗶
•	detecção de erro e recuperação do sistema ✓
0	diagnóstico de falhas e reconfiguração do sistema 🗡
0	metodologias para o desenvolvimento de software e regras para o projeto de hardware 🗡

Notas relativas a este envio: 1/1.

Question5

Notas: 1

Considerando as técnicas de tolerância a falhas, associe o conceito à técnica apropriada.

Previne nova ativação de falhas.	Tratamento de falhas	•
Identifica a presença de um erro.	Detecção de erro	•
Transforma um estado que contém um ou mais erros e falhas em um estado sem os erros detectados e sem falhas que possam ser novamente ativadas.	Recuperação	•
Elimina erros do estado do sistema.	Tratamento de erros	•

# Correto

Notas relativas a este envio: 1/1.

Question6 Notas: 1		
Assinale as técnicas para tratamento de erros:		
Escolha pelo menos uma resposta.		
<b>☑</b>	recuperação por retorno ✓	
	diagnóstico 🗡	
	reconfiguração 🗶	
▼	recuperação por avanço ✓	
	reinicialização X	
▼	compensação ✓	
	isolamento X	
Correto		
Notas relativas a este envio	p: 1/1.	
Question7 Notas: 1		
Assinale as técnicas para tratamento	o de falhas:	
Escolha pelo menos uma resposta.		
	recuperação por retorno 🗡	

<b>~</b>	diagnóstico ✓	
V	isolamento ✓	
▼	reconfiguração ✓	
	compensação 🗶	
	recuperação por avanço X	
✓	reinicialização ✓	
Correto Notas relativas a este envio:	1/1.	
Question8 Notas: 1		
Associe o conceito à técnica de tratam	nento de erros adequada.	
Leva o sistema a um estado salvo que ocorreu antes da ocorrência do erro:		
O novo estado é um estado sem erros detectados:		
O estado com erros contém redundância suficiente para permitir que erros sejam mascarados:		compensação <b>▼</b>
Correto Notas relativas a este envio:	: 1/1.	

Question9

Notas: 1

Considere a detecção de erros. Associe o conceito ao termo apropriado.

Ocorre durante a prestação normal de serviço.

detecção concorrente

Ocorre quando o serviço é suspenso e se presta a verificar a presença de erros latentes e falhas dormentes.

detecção preemptiva

### Correto

Notas relativas a este envio: 1/1.

Question10

Notas: 1

Qual dos conceitos abaixo se aplica a diagnóstico?

Escolher uma resposta.

•	Identificação e registro da causa do erro em termos de localização e tipo. ✓
0	Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. X
0	Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ⊀
0	Exclusão física ou lógica do componente com falha, ou seja, transformação da falha ativa em dormente. 🗶

# Correto

Notas relativas a este envio: 1/1.

Question11 Notas: 1 Qual dos conceitos abaixo se aplica a isolamento de falha?

Escolher uma resposta.

0	Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
С	Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. X
0	Identificação e registro da causa do erro em termos de localização e tipo. 🗡
•	Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. ✓

# Correto

Notas relativas a este envio: 1/1.

Question12

Notas: 1

Qual dos conceitos abaixo se aplica a reconfiguração?

0	Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. 🗶
С	Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✗
•	Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. ✓

0

Identificação e registro da causa do erro em termos de localização e tipo. 🗡

### Correto

Notas relativas a este envio: 1/1.

Question13

Notas: 1

Qual dos conceitos abaixo se aplica a reinicialização?

Escolher uma resposta.

0	Chaveamento para componentes extra ou redistribuição de tarefas entre os componentes livres de falhas. X
0	Identificação e registro da causa do erro em termos de localização e tipo 🗡
C	Exclusão física ou lógica do componente com falha, ou seja, transformação da falha em dormente. 🗶
•	Verificação, atualização e registro de nova configuração e atualização das tabelas e registros do sistema. ✓

# Correto

Notas relativas a este envio: 1/1.

Question14

Notas: 1

Considere as técnicas de tratamento de erros. Todas elas podem ser aplicadas por demanda, ou seja, quando ocorre um erro, mas apenas uma pode ser aplicada sistematicamente, independente da presença ou não do erro. Assinale a técnica que pode ser aplicada sistematicamente.

•	a. compensação ✓
0	b. recuperação por retorno 🗶
0	c. nenhuma das técnicas de tratamento de erro pode ser aplicada sistematicamente 🗡
0	d. recuperação por avanço 🗡

Notas relativas a este envio: 1/1.

# Question15

Notas: 1

Avizienis não lista mascaramento de falhas como uma técnica de tolerância a falhas. No texto, mascaramento de falhas é definido como:

Escolher uma resposta.

•	a. resultado do uso sistemático de compensação podendo levar progressivamente ou eventualmente à perda da redundância de proteção ✓
С	b. resultado do uso sistemático de recuperação por retorno podendo levar progressivamente ao estado inicial do sistema (estado no momento do power up) 🗶
С	c. resultado do uso sistematico de tratamento de falhas seguida imediatamente por recuperação por avanço e postergação da manutenção ⊀
C	d. resultado do uso de recuperação por retorno seguida imediatamente por recuperação por avanço e um passo posterior de tratamento de erro 🗡

Question16 Notas: 1	
Assinale <b>verdadeiro</b> ou <b>falso</b> de acordo com os conceitos do artigo do Avizienis.	
Após a detecção de um erro, sempre deve seguir-se o tratamento do erro e só então, depois disso, o tratamento da falha.	Falso
As técnicas de detecção de erros preemptiva e tratamento de erros, possivelmente acompanhadas de tratamento de falha, são comumente executadas no power up do sistema.	Verdadeiro
Um fator que distingue tolerância a falhas de manutenção é que manutenção é realizada por um agente externo.	Verdadeiro 🔻
Recuperação por avanço (rollforward) e recuperação por retorno (rollback) são mutuamente exclusivas, ou seja, só pode ser usada uma ou outra.	Falso
Falhas intermitentes não necessitam isolamento ou reconfiguração.	Verdadeiro <b>▼</b>
Correto	
Notas relativas a este envio: 1/1.	
Question17 Notas: 1	
Assinale <b>verdadeiro</b> ou <b>falso</b> de acordo com o artigo do Avizienis (item 5.2.2).	
Para falhas sólidas de projeto, canais redundantes podem apresentar projeto idêntico pois componentes de hardware sempre apresentam defeitos de forma independente uns dos outros.	Falso
As classes de falha que podem realmente ser toleradas em um sistema dependem da hipótese de falha que foi	Verdadeiro ▼

considerada no processo de desenvolvimento do sistema.		
A escolha das técnicas de tolerância a falhas a serem empregadas depende do custo do projeto e dos recursos disponíveis e não da hipótese de falha considerada.	Falso	<b>-</b>
Para falhas físicas, canais redundantes podem apresentar projeto idêntico pois podemos nos basear na hipótese que componentes de hardware apresentam defeitos independentemente uns dos outros.	Verdadeiro	<b>T</b>
Um método popular de alcançar tolerância a falhas é realizar múltiplas computações através de múltiplos canais, seja sequencialmente ou concorrentemente.	Verdadeiro	-

Notas relativas a este envio: 1/1.

Question18

Notas: 1

A existência de recursos internos ao componente, seja de hardware ou software, para que o componente, além de executar sua função, também possa realizar detecção concorrente de erros está relacionada ao conceito de:



_						
C	$\sim$	r	r	$\sim$	+	$\sim$
L	u			ᆮ	L	u

Notas relativas a este envio: 1/1.

Question19

Notas: 1

Nem todas as técnicas de tolerância a falhas são igualmente efetivas. A medida de eficiência de uma dada técnica de tolerância a falhas é chamada de (escreva o nome em português):

### Resposta:

Cobertura

### Correto

Notas relativas a este envio: 1/1.

Question20

Notas: 1

As imperfeições de tolerância a falhas, segundo Avizienis e demais autores, são devidas a dois fatores. São eles:

Escolher uma resposta.

С	problemas com confinamento de falhas e projeto diversitário mal especificado 🗡
0	ausência de projeto diversitário e de componentes auto-verificáveis X
•	deficiência na cobertura do tratamento de erros e falhas e deficiências nas hipóteses sobre falhas ✓
0	presença de falhas dependentes e ausência de isolamento de falhas 🗶

# Correto

### Question21

Notas: 1

Quando tralhamos com múltiplos componentes, uma questão importante é prevenir a propagação de erros para os componentes livres de falhas, principalmente quando um componente precisa comunicar alguma informação para outros componentes. Todos os componentes que recebem a informação devem alcançar um estado consistente entre si. O problema de alcançar um acordo entre todos os componentes sobre um valor é conhecido como:

Escolher uma resposta.

•	a. problema do consenso ✓
0	b. problema do acordo recursivo 🗡
0	c. problema da exclusão mútua 🗡
0	d. problema da consistência mútua 🗡
0	e. problema da informação de origem única 🗶

#### Correto

Notas relativas a este envio: 1/1.

# Question22

Notas: 1

Avizienis e os demais autores do artigo sob análise afirmam que tolerância a falhas é um conceito recursivo. Qual o significado desta afirmação?

Escolher uma resposta.

a. Conceito recursivo significa neste contexto que os mecanismos de tolerância a falhas não podem ser implementados nem em software nem em hardware pois sempre resta um componente

	não protegido. 🗡
С	b. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas são extremamente complexos e só podem ser implementados em software por linguagens de programação que permitem recursão. X
•	c. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas também devem ser protegidos contra as falhas que possam afetá-los. ✓
0	d. Conceito recursivo significa neste contexto que os mecanismos de tolerância a falhas não podem ser implementados em hardware, pois componentes de hardware não são recursivos. X
С	e. Conceito recursivo significa neste contexto que os mecanismos que implementam tolerância a falhas são auto-suficientes, auto-contidos e não sofrem interferência de falhas. X

Notas relativas a este envio: 1/1.

Question23

Notas: 1

Segundo Avizienis e demais autores, os termos auto-reparo, auto-cura e resiliência são:

0	relativos a técnicas de tratamento de defeitos 🗶
•	sinônimos de tolerância a falhas ✓
0	sinônimos de dependabilidade 🗶
0	referentes a técnicas complementares a tolerância a falhas 🗡

relativos a técnicas específicas de tratamento de erros 🔻

### Correto

Notas relativas a este envio: 1/1.

Question24

Notas: 1

Um dos quatro meios para atingir dependabilidade é a remoção de falhas (5.3). Segundo Avizienis, a remoção de falhas durante a fase de desenvolvimento de um sistema consiste em 3 passos. São eles:

Escolher uma resposta.

0	detecção de erro, tratamento de erro e tratamento de falha 🗶
0	verificação estática, verificação dinâmica e execuçao simbólica 🗡
0	rollback, rollforward e compensação 🗶
0	teste dinâmico, teste estático e injeção de falhas 🗡
•	verificação, diagnóstico e correção ✓

# Correto

Notas relativas a este envio: 1/1.

Question25

Notas: 1

As técnicas de verificação podem ser classificadas levando em consideração se executam ou não o sistema. De acordo com essa classificação, a verificação é chamada de:

0	a. determinística ou randômica 🗡
0	b. formal ou estatística 🗶
•	c. dinâmica ou estática ✓
0	d. regressiva ou não regressiva 🗶
C	e. numérica ou simbólica 🗡

Notas relativas a este envio: 1/1.

Question26

Notas: 1

Remoção de falhas durante a fase de uso (ou operação) de um sistema compreende duas técnicas. São elas:

Escolher uma resposta.

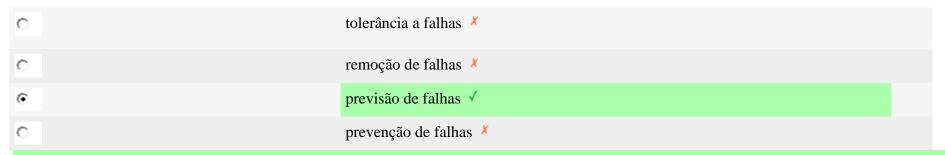
•	manutenção corretiva e manutenção preventiva ✓
0	verificação e correção 🗡
0	recuperação e compensação 🗶
0	isolamento e remoção 🗶
0	detecção e tratamento de erro 🗶

# Correto

O.,	+:-	27
Qu	esuc	n27

Realizar uma avaliação do comportamento de um sistema com respeito a ocorrência de falhas e sua ativação é uma tarefa relacionada a:

Escolher uma resposta.



#### Correto

Notas relativas a este envio: 1/1.

# Question28

Notas: 1

Considerando avaliação do comportamento do sistema com respeito a ocorrência de falhas, associe o conceito ao tipo de avaliação:

visa identificar, classificar e ordenar os modos de defeito que podem levar a defeitos do sistema

visa avaliar, em termos de probabilidades, a extensão em que certos atributos são
satisfeitos; os atributos são então vistos como "medidas"

avaliação qualitativa ou probabilística

visa avaliação quantitativa ou probabilística

### Correto

Notas relativas a este envio: 1/1.

Question29

Avizienis cita alguns métodos para avaliação quantitativa (probabilística) e qualitativa no contexto de previsão de falhas. Para os métodos listados a seguir, assinale se são adequados para avaliação qualitativa, quantitativa, ambas ou nenhuma:



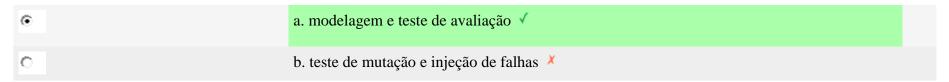
### Parcialmente correta

Notas relativas a este envio: 0.6/1.

Question30

Notas: 1

Avizienis cita duas abordagens principais e complementares para previsão de falhas probabilística. São elas:



C	c. teste funcional e teste estrutural 🗡
0	d. teste de caixa branca e teste de caixa preta 🗡
C	e. teste baseado em falhas e teste baseado em critérios 🗡

# INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ► FTF 2012/2 ► Questionários ► Tolerância a falhas na aplicação ► Revisão da tentativa 1

# Tolerância a falhas na aplicação

### Revisão da tentativa 1

# Terminar revisão

Iniciado em	sexta, 14 setembro 2012, 03:47
Completado em	segunda, 24 setembro 2012, 13:37
Tempo empregado	10 dias 9 horas
Notas	27.58/28
Nota	<b>98.49</b> de um máximo de 100( <b>98</b> %)

Notas:

1

No item 2.1 do artigo: *V. D. Florio e C. Blondia,* "A survey of linguistic structures for application-level fault tolerance", ACM Comput. Surv., vol. 40, no. 2, p. 6:1–6:37, maio 2008, quando argumentam sobre a necessidade de tolerância a falhas de software, os autores comentam sobre tolerância a falhas de hardware. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Durante muitos anos, a pesquisa na área de tolerância a falhas foi concentrada em desenvolver estruturas de hardware engenhosas e eficientes para lidar com falhas.

Durante algum tempo, tolerância a falhas de software foi considerada a única necessária para alcançar a disponibilidade e a integridade de dados demandadas por computadores modernos.

Componentes de hardware são apenas uma das fontes de não confiabilidade em sistemas computacionais, decrescendo em importância conforme a confiabilidade desses componentes aumenta.

A prevalência de falhas de hardware está aumentando continuamente.



Verdadeiro

Notas relativas a este envio: 1/1.

# Notas:

2

Ainda no item 2.1 do artigo, os autores comentam sobre a complexidade dos sistemas computacionais. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Quanto maior o nível de abstração, maior é a complexidade dos algoritmos que entram em jogo e a consequente propensão a erro das máquinas envolvidas (sejam máquinas reais ou abstratas).



A maior parte da complexidade dos modernos sistemas computacionais reside no hardware.



Dividindo a complexidade do sistema em camadas, falhas ficam confinadas a uma única camada, tornando mais fácil gerenciar a complexidade crescente dos sistemas.



Uma forma conceitual eficente e flexível de gerenciar complexidade é considerar um sistema como uma hierarquia de máquinas abstratas sofisticadas.



Mesmo que a divisão em camadas torne a complexidade transparente, a complexidade ainda é parte do sistema global sendo desenvolvido e falhas se propagando por várias camadas podem ocorrer.

Verdadeiro **▼** 

Parcialmente correta

Notas relativas a este envio: 0.8/1.

# Notas:

3

Florio e Blondia, no item 2.2, justificam prover tolerância a falhas de software no nível da aplicação baseados no fato de que a maioria dos defeitos experimentados por computadores modernos são devidos a:

Escolher uma resposta.

- a. falhas de software incluindo as falhas no nível de aplicação √
- b. falhas em device drivers e kernel de sistemas operacionais
- 🔘 c. quedas de serviço 🧣
- 🔘 d. falhas de hardware 🦹
- 🔘 e. falhas de interação com o usuário 🧣

### Correto

4

1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação é a adoção popular de componentes de software reusáveis. Muitas aplicações orientadas a objetos são construídas com componentes reusáveis. Considerando reuso de componentes, podemos afirmar, de acordo com a opinião dos autores, que:

Escolher uma resposta.

- a. o reuso de componentes aumenta os custos de desenvolvimento mas contribui para aumento de confiabilidade porque os componentes já foram exaustivamente testados
- b. o desenvolvedor da aplicação não conhece o desenvolvedor do componente reusável e por essa razão não pode confiar na qualidade do componente
- c. o reuso de componentes diminui os custos de desenvolvimento mas a confiabilidade desses componentes e o seu impacto na confiabilidade da aplicação são frequentemente desconhecidos
- d. o reuso de componentes aumenta a complexidade do projeto de sistemas e, além disso, a confiabilidade desses componentes é geralmente inadequada

Correto

Notas relativas a este envio: 1/1.

Notas:

5

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação, que eles consideram como provavelmente o mais convicente, é chamado de argumento fim-a-fim. Considerando esse argumento e a opinião dos autores, assinale verdadeiro e falso.

frequentemente funções só podem ser completamente e corretamente implementadas com o conhecimento e auxílio da aplicação que se encontra nas extremidades do sistema de suporte verdadeiro 🔻

soluções de tolerância a falhas baseadas puramente em hardware ou no sistema operacional não são capazes de prover tolerância a falhas completa e fim-a-fim na aplicação do usuário

verdadeiro 🔻

o argumento implica que todas as tarefas de tolerância a falhas podem ser realizadas no nível da aplicação

falso 🔻

como exemplo é citado que um canal de comunicação extraordinariamente confiável livra a aplicação da obrigação de prover confiabilidade



Notas relativas a este envio: 1/1.

Notas: 1

6

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Quais foram os problemas mencionados?

Escolher uma resposta.

- a. como incorporar tolerância a falhas no nível da aplicação, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas 🦹
- b. como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, quais mecanismos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas 🦹
- 🔘 c. como incorporar tolerância 🛮 como treinar a equipe de a falhas no nível da aplicação, desenvolvedores para quais recursos de tolerância aplicar tolerância a falhas no a falhas devem ser suportados, e como gerenciar mecanismos de tolerância a o código tolerante a falhas

nível da aplicação, quais falhas devem ser suportados, e como gerenciar o código tolerante a falhas

 d. como reduzir os custos de projeto redundante, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas 🦹

Correto

Notas relativas a este envio: 1/1.

7 Notas: 1

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Qual problema foi proposto primeiramente por Randell?

Escolher uma

- 🔘 a. como reduzir os custos de projeto 🦹
- D. como treinar a equipe de desenvolvedores para aplicar

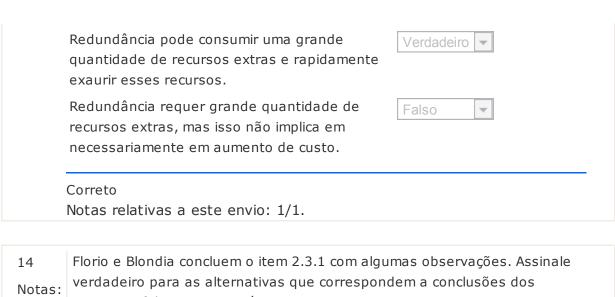
	resposta. tolerând	cia a falhas no nível da a <sub>l</sub>	olicação 🦹
	C. como	gerenciar o código tolera	ante a falhas 🦹
	🔘 d. quais suporta	mecanismos de tolerând dos 🦹	cia a falhas devem ser
	⊚ e. como ✓	incorporar tolerância a f	alhas no nível da aplicação
	Correto Notas relativas a este	envio: 1/1.	
8 Notas: 1	apropriadas para a inco Soluções inadequadas a	essário o uso de técnicas rporação de tolerância a a esse problema, segund r alto grau de intrusividad	falhas no nível da aplicação. lo o artigo de Florio e
	Correto Notas relativas a este	envio: 1/1.	
9 Notas: 1	resulta na mistura do c verdadeiro para as co	código funcional com o có nsequências ao desenvo ura, e <b>falso</b> quando a alí dos códigos:	• •
	aumentam		verdadello 🕶
	o tamanho do código re	esultante é menor	falso
	o código resultante é m	nais fácil de ser mantido	falso
	a complexidade do siste	ema diminui	falso
	as tarefas do desenvol significativamente mais		verdadeiro 🔻
	Correto Notas relativas a este	envio: 1/1.	
10 Notas:	Com relação ao proble aplicação, Florio e Blon	·	olerância a falhas no nível da
1		orporação neste nível é in vel conciliar interesses 🥻	·
			lhas devem prevalecer em no código da aplicação 🦹
		es diferentes de desenvo gadas da parte funcional	

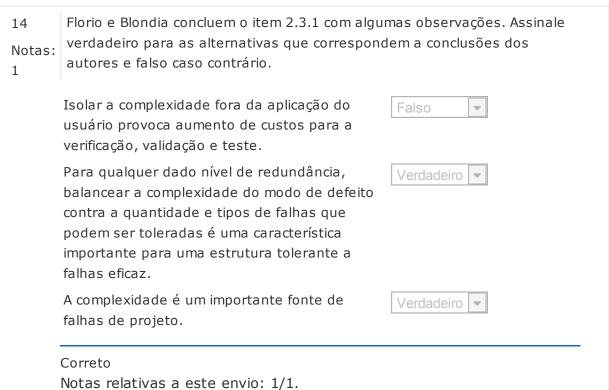
suporte a tolerância a falhas 🦹 🔘 d. uma estrutura ideal de sistema deve garantir uma adequada separação entre os interesses funcionais e os relativos a tolerância a falhas no código da aplicação 🧹 Correto Notas relativas a este envio: 1/1. 11 Segundo Florio e Blondia, uma resposta inadequada para o problema de qual recurso de tolerância a falhas deve ser suportado pelo nível da Notas: aplicação pode apresentar duas características inconvenientes: 1 🔘 a. requerer algoritmos muito sofisticados e estratégias de Escolher verificação de alto custo 🦹 uma resposta. b. requerer desenvolvedores altamente especializados e ferramentas de teste sofisticadas 🦹 🔘 c. exigir equipamento especial e linguagens de programação específicas 🦹 d. requerer alto grau de redundância e rapidamente. consumir a quantidade de redundância disponível 🧹 Correto Notas relativas a este envio: 1/1. Florio e Blondia definem 3 atributos estruturais para tolerância a falhas no 12 nível da aplicação (ALTF) chamados de SC, SA e A. Associe: Notas: 1 separação de interesses funcionais e não funcionais suporte direto a um grande conjunto de SA mecanismos de tolerância a falhas facilidade do código tolerante a falhas se adaptar ao ambiente onde se encontra no momento Correto Notas relativas a este envio: 1/1. Considerando o conceito de redundância como apresentado no item 2.3.1 13 do artigo de Florio e Blondia, é possível afirmar que: Notas:

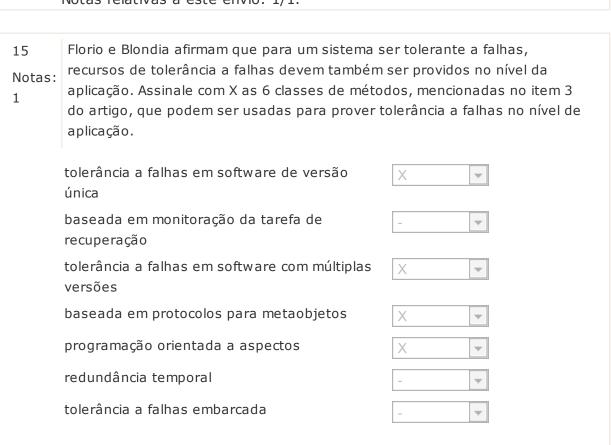
Considerando o conceito de redundância como apresentado no item 2.3.1 do artigo de Florio e Blondia, é possível afirmar que:

Tolerância a falhas é, em geral, o resultado de alguma estratégia que efetivamente explora alguma forma de redundância.

É impossível acrescentar mais confiabilidade a um canal não confiável apenas aumentando o grau de redundância de informação.







	customiza		distribuídas X	·
	redundând	cia espacial	X	₩
	Parcialmen Notas rela	te correta tivas a este envio: 0.78	/1.	
16 Notas:	Qual o requisito chave para o desenvolvimento de sistemas tolerantes a falhas (de acordo com Florio e Blondia, item 3.1)?			
1	Escolher	🔘 a. Replicação de dados 🦹		
	uma resposta.			
	resposta.	C. Recursos replicado	os em hardware	X
		🔘 d. Replicação tempo	ral 🦹	
		e. Recursos replicad	os em software 🌡	ĸ
	Correto Notas rela	tivas a este envio: 1/1.		
17		ão é usada como método n múltiplas computações.	•	-
	domínios		segundo Fiorio e	biolidia, quais os seus
1				
1	domínios í Escolher uma	?	no e codificação d	e dados 🦹
	domínios í	?	no e codificação d cia e paralelismo	e dados 🦹
1	domínios í Escolher uma	?  a. Espaço, paralelisr  b. Tempo, concorrên	mo e codificação d cia e paralelismo informação <b>√</b>	e dados 🥻
1	domínios de Escolher uma resposta.	<ul> <li>a. Espaço, paralelisr</li> <li>b. Tempo, concorrên</li> <li>c. Tempo, espaço e</li> </ul>	mo e codificação d cia e paralelismo informação <b>√</b>	e dados 🥻
18 Notas:	domínios de Escolher uma resposta.  Correto Notas reladores Considera	?  a. Espaço, paralelisr  b. Tempo, concorrên  c. Tempo, espaço e  d. Codificação de da	mo e codificação d cia e paralelismo informação √ dos, tempo e con	e dados 🔏  corrência 🦧  ar tolerância a falhas,
18 Notas:	domínios de Escolher uma resposta.  Correto Notas reladores considera segundo F	a. Espaço, paralelisr b. Tempo, concorrên c. Tempo, espaço e d. Codificação de da	mo e codificação d cia e paralelismo informação √ dos, tempo e con	e dados 🔏  corrência 🦧  ar tolerância a falhas,
1	domínios  Escolher uma resposta.  Correto Notas rela  Considera segundo F  Repetição	a. Espaço, paralelisr b. Tempo, concorrên c. Tempo, espaço e d. Codificação de da tivas a este envio: 1/1.	no e codificação d cia e paralelismo informação dos, tempo e con cação para alcanç explicação ao do	e dados 🔏  corrência 🦧  ar tolerância a falhas,

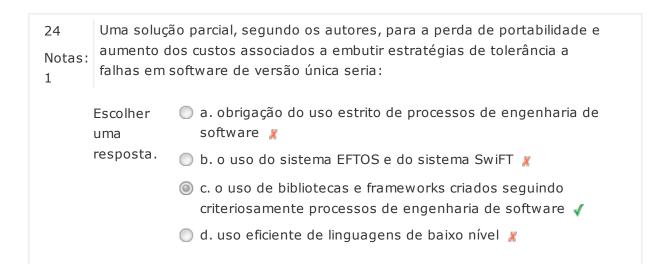
Notas:	3.1 do art		nificado aos termos na notação
	número de	e programas	p
	número de	e canais de hardware	m 🔻
	número de	e execuções	n 🔻
	Correto Notas relat	tivas a este envio: 1/1.	
20	Um sistem	na 1T/1H/1S, de acordo com Flo	orio e Blondia, é chamado de:
Notas:	Escolher	🔘 a. tolerante a falhas 🦹	
	uma resposta.	🔵 b. intolerante a falhas 🦹	
	resposta.	🔵 c. duplex 🦹	
		d. simplex   √	
		🔵 e. replicado 🦹	
	Correto Notas relat	tivas a este envio: 1/1.	
21	Tolerância	a falhas em versão única de s	software implica em:
Notas:			
1	Escolher uma	<ul> <li>a. Embutir em um sistema técnicas de detecção e re</li> </ul>	a simplex (ou seja, sem réplicas) ccuperação de erros. 🎸
	resposta.	<ul><li>b. Construir programas q replicados.</li></ul>	ue não possam ser copiados nem
		<ul> <li>c. Executar software em a processos e múltiplos usu</li> </ul>	ambientes que inibam múltiplos Jários simultâneos. 🦹
		O d. Construir programas a	plicativos sem erros. 🦹
		técnicas de ações atômic	s programas aplicativos as as, pontos de recuperação e e tratamento de exceções. 🗶
	Correto Notas relat	tivas a este envio: 1/1.	
22	_		olerância a falhas em versão única
Notas:	ue soitwa	re apresenta como desvantag	CIIS.
_	Escolher uma	a. necessidade de optar sistema SwiFT	entre o sistema EFTOS e o
	resposta.	<ul> <li>b. necessidade de desen para suprir os recursos d</li> </ul>	volver bibliotecas e frameworks e tolerância a falhas 🧣

c. aumento do tamanho da aplicação com consequente perda de transparência, da facilidade de manutençao e da portabilidade, acompanhados do aumento do tempo de desenvolvimento e dos custos 
 d. obrigação do uso estrito de processos de engenharia de software

Correto

Notas relativas a este envio: 1/1.

Considerando o conceito de SV como apresentado por Florio e Blondia, 23 responda verdadeiro ou falso: Notas: 1 A adoção do SV na camada de aplicação requer | Verdadeiro concentrar no código fonte da aplicação tanto a especificação do que fazer como a estratégia de tolerância a falhas. SV baseia-se na hipótese que todos os recursos de tolerância a falhas são providos transparentemente pelo hardware, o que gera um código leve, facilmente portável, de baixo custo e baixa necessidade de manutenção. A principal vantagem de incluir em um único Falso código tanto a funcionalidade do sistema como as estratégias de tolerância a falhas é que essa abordagem aumenta a portabilidade do código pois o mesmo carrega consigo todos os recursos necessários. Como exemplos de técnicas de detecção e Verdadeiro recuperação, que podem ser embutidas em software de versão única, os autores citam ações atômicas, pontos de recuperação e rollback e tratamento de exceções. Correto Notas relativas a este envio: 1/1.



	Correto Notas relativas a este envio: 1/1.	
25	Considerando o sistema SwiFT, assinale as se	entenças verdeiras e falsas.
Notas: 1	SwiFT inclui uma série de componentes reutilizáveis de software adequados para incluir técnicas de tolerância a falhas em software de versão única	Verdadeiro ▼
	SwiFT é o nome de um computador tolerante falhas que foi popular na década de 90	Falso 🔻
	SwiFT introduziu apenas um pequeno overhea em custo na maioria dos caso	ad Verdadeiro ▼
	SwiFT foi usado com sucesso para aumentar a tolerância a falhas em sistemas de software onde falhas residuais estavam presentes	Verdadeiro ▼
	SwiFT é o nome de um projeto que visava construir componentes de hardware para facilitar a execução de software tolerante a falhas	Falso
	Correto Notas relativas a este envio: 1/1.	
26 Notas: 1		), SA (adequação sintática) e A ersão única, SV, como os
Notas:	Notas relativas a este envio: 1/1.  Florio e Blondia definem no item 2.3 os conce interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de v	), SA (adequação sintática) e A ersão única, SV, como os
Notas:	Notas relativas a este envio: 1/1.  Florio e Blondia definem no item 2.3 os conce interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de v autores avaliam esses atributos? Indique suf	), SA (adequação sintática) e A ersão única, SV, como os iciente ou insuficiente.
Notas:	Notas relativas a este envio: 1/1.  Florio e Blondia definem no item 2.3 os conce interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de v autores avaliam esses atributos? Indique suf separação de interesses	), SA (adequação sintática) e A ersão única, SV, como os iciente ou insuficiente.
Notas:	Notas relativas a este envio: 1/1.  Florio e Blondia definem no item 2.3 os conce interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de v autores avaliam esses atributos? Indique suf separação de interesses adequação sintática	), SA (adequação sintática) e A ersão única, SV, como os iciente ou insuficiente.  insuficiente  insuficiente
Notas:	Notas relativas a este envio: 1/1.  Florio e Blondia definem no item 2.3 os conce interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de v autores avaliam esses atributos? Indique suf separação de interesses adequação sintática adaptabilidade  Correto	), SA (adequação sintática) e A ersão única, SV, como os iciente ou insuficiente.  insuficiente  insuficiente  insuficiente  após reforçarem uma série de a, alertam para uma
Notas: 1 27 Notas:	Notas relativas a este envio: 1/1.  Florio e Blondia definem no item 2.3 os conce interesses funcionais e de tolerância a falhas (adaptabilidade). Em relação a software de v autores avaliam esses atributos? Indique suf separação de interesses adequação sintática adaptabilidade  Correto  Notas relativas a este envio: 1/1.  Na conclusão do item 3.1.1, Florio e Blondia, a desvantagens da abordagem de versão única	), SA (adequação sintática) e A ersão única, SV, como os iciente ou insuficiente.  insuficiente  insuficiente  insuficiente  insuficiente  após reforçarem uma série de a, alertam para uma Qual é ela?

- ◎ c. não apresenta qualquer restrição quanto a classe de aplicação que pode se beneficiar da abordagem
- d. não apresenta impacto nos custos de desenvolvimento e manutenção

Notas relativas a este envio: 1/1.

# Notas:

28

Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens, falam das vantagens de utilizar, em software de versão única, ferramentas que permitem ao usuário tratar com átomos de tolerância a falhas sem se preocupar com sua implementação. Em relação a essas ferramentas podemos afirmar que:

Escolher uma

- a. só podem ser usadas em programas escritos em C ou
   C++
- resposta.
- b. introduzem grande overhead no desempenho para apenas um pequeno aumento nos atributos de dependabilidade
- c. não apresentam qualquer impacto no custo ou no desempenho da aplicação
- 🔘 d. prejudicam a dependabilidade do sistema 🧣
- e. permitem ao projetista reusar peças de software sofisticadas e amplamente testadas

# Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como João Luiz Grave Gross (Sair)

FTF 2012/2

# Tolerância a falhas: multiversões e protocolos de meta objetos Revisão da tentativa 1

Iniciado em	segunda, 24 setembro 2012, 13:38
Completado em	segunda, 24 setembro 2012, 14:30
Tempo empregado	52 minutos 23 segundos
Notas	28/28
Nota	<b>100</b> de um máximo de 100( <b>100</b> %)

Question1

Notas: 1

O item 3.1.2 do artigo de Florio e Blondia (Artigo: FLORIO, V.; BLONDIA. C. *A Survey of Linguistic Structures for Application-Level Fault Tolerance*, ACM Computing Surveys, Vol. 40, No. 2, April 2008.) trata de tolerância a falhas em software multi-versão (MV). Esse tipo de abordagem requer um número mínimo de versões do mesmo programa, que é igual a:

Escolher uma resposta.

0

a. qualquer número ímpar para não haver empate na votação 🗶

•	b. 2 ✓
0	c. 1 ×
0	d. sT/yH/nS ×
0	e. 3 ×

Notas relativas a este envio: 1/1.

# Question2

Notas: 1

No item 3.1.2, Florio e Blondia afirmam que a abordagem MV exige N versões de software com certa particularidade importante. Identifique essa particularidade:

Escolher uma resposta.

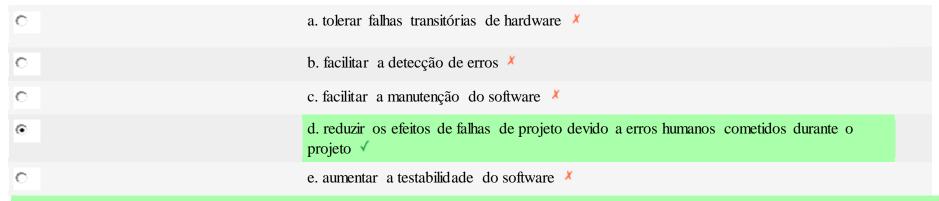
0	a. Todas as N versões devem apresentar o mesmo desempenho. 🗡
0	b. Todas as N versões devem executar sincronizadamente sobre canais idênticos de hardware. 🗡
0	c. O número N de versões deve ser ímpar para não haver empate na votação 🗡
0	d. Todas as versões devem ser idênticas. 🗶
•	e. Cada uma das N versões deve ter sido projetada e desenvolvida por um time diferente e independente. ✓

#### Correto

$\cap$		ct	in	n3
V	ue	Sι	IU	113

A abordagem MV tem por objetivo:

Escolher uma resposta.



### Correto

Notas relativas a este envio: 1/1.

# Question4

Notas: 1

A abordagem MV é subdividida, segundo Florio e Blondia, em duas abordagens principais. Assinale **sim** para as abordagens abaixo que correspondem às citadas no artigo.

tratamento de exceções	não	
bloco de recuperação	sim	_

ações atômicas distribuídas	não	<b>-</b>
programação N-versões	sim	•
programação concorrente	não	_
programação distribuída	não	_

Notas relativas a este envio: 1/1.

# Question5

Notas: 1

A abordagem conhecida com o nome de blocos de recuperação, que foi proposta por Randell em 1975, pode ser implementada da forma NT/1H/NS ou 1T/NH/NS, sendo a primeira a mais frequente. Conhecendo essas expressões podemos afirmar que:

•	a. A abordagem opera com N programas executados ou em série no mesmo hardware ou em paralelo sobre N canais de hardware. ✓
0	b. N se refere ao número de recuperações necessárias para garantir o correto funcionamento dos módulos, também chamados de blocos, do programa. 🗡
0	c. A abordagem opera com 1 programa executado N vezes em série e N vezes em paralelo. 🗡
0	d. A abordagem opera com N programas encadeados em série formando um único bloco de execução para fins de recuperação. 🗡

e. A abordagem opera com N programas executados em paralelo no mesmo hardware. 🗡

### Correto

Notas relativas a este envio: 1/1.

# Question6

Notas: 1

Na abordagem conhecida com o nome de blocos de recuperação, cada um dos N componentes funcionais, programas ou versões é chamado alternativa. Assinale a opção mais próxima do conceito da abordagem.

Escolher uma resposta.

0	a. Cada alternativa executa toda computação; os resultados são então avaliados por um teste de aceitação que calcula a média entre os resultados para compor o resultado final. 🗶
C	b. Cada alternativa colabora com as demais executando parte da computação; as diferentes partes são então combinadas após passar por um teste de aceitação. 🗡
•	c. O resultado de cada alternativa, começando pela primária, é avaliado por um teste de aceitação; caso não seja aceito, o resultado da próxima alternativa é avaliada e assim sucessivamente até que um dos resultados passe no teste de aceitação. ✓
c	d. Cada alternativa executa toda computação; os resultados são então comparados por um teste de aceitação que escolhe o melhor. 🗡
0	e. A cada execução é escolhida uma nova alternativa para que os programas não acumulem erros residuais. 🗡

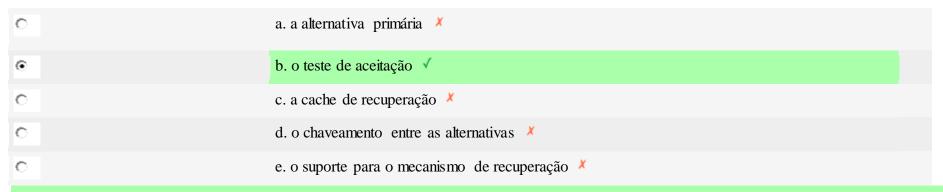
# Correto

Notas relativas a este envio: 1/1.

Question7

A eficácia da abordagem de blocos de recuperação reside na cobertura do mecanismo de detecção de erros adotado, portanto o componente mais crucial da abordagem é:

Escolher uma resposta.



### Correto

Notas relativas a este envio: 1/1.

### Question8

Notas: 1

Florio e Blondia, analisando blocos de recuperação, apontam alguns pontos positivos e negativos relacionados a abordagem. Assinale verdadeiro se a opção corresponde a uma afirmação encontrada no artigo e falso se não corresponde.

A abordagem de blocos de recuperação foi validada com sucesso através de experimentos estatísticos e modelagem matemática.

Blocos de recuperação não são uma abordagem satisfatória considerando instrusividade no código, portabilidade e facilidade de manutenção.

A abordagem de blocos de recuperação consiste em uma estratégia rígida sem possibilidade de configuração offline e com baixa adaptabilidade.

Um aspecto positivo da abordagem de blocos de recuperação é que o custo de manutenção e desenvolvimento independe do número de réplicas, apesar de depender do número de canais de hardware.

Não existem registros que a abordagem tenha sido usada com sucesso em algum campo de aplicação.

### Correto

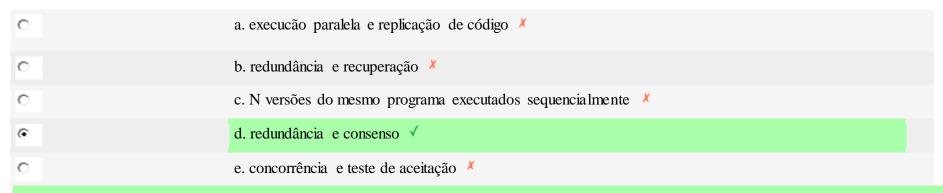
Notas relativas a este envio: 1/1.

### Question9

Notas: 1

Segundo Florio e Blondia, sistemas de programação N-versões (NVP) são baseados em:

Escolher uma resposta.



### Correto

Notas relativas a este envio: 1/1.

Question10

O autor que introduziu o conceito de programação N-versões foi:

Escolher uma resposta.



# Correto

Notas relativas a este envio: 1/1.

# Question11

Notas: 1

Na abordagem conhecida como programação N-versões, NVP, cada um dos N componentes funcionais é chamado versão. O autor da abordagem define que as versões devem ser geradas de forma independente . O número N de versões deve ser maior que 1 . As versões funcionalmente equivalentes devem ser desenvolvidas a partir

# Correto

Notas relativas a este envio: 1/1.

Question12

Notas: 1

Na abordagem conhecida como programação N-versões, cada um dos N componentes funcionais é diferente dos demais e é chamado versão. Assinale a opção mais próxima do conceito da abordagem.

Escolher uma resposta.

0	a. Cada versão colabora com as demais executando parte da computação; as diferentes partes são então combinadas após passar por consenso. X
•	b. Cada versão executa toda computação; os resultados são então processados por um algoritmo genérico de decisão e é escolhido o resultado final por consenso ou maioria. ✓
0	c. Cada versão executa toda computação; os resultados são então avaliados por um teste de aceitação que escolhe o mais correto com o resultado final. X
0	d. A cada execução é escolhida uma nova versão para que os programas não acumulem erros residuais. 🗶
0	e. O resultado de cada versão, começando pela primária, é avaliado por um teste de consenso; caso não seja aceito, o resultado da próxima versão é avaliado e assim sucessivamente até que um dos resultados passe no teste. X

# Correto

Notas relativas a este envio: 1/1.

Question13

Notas: 1

A abordagem conhecida como programação N-versões é baseada em uma conjectura fundamental, qual seja:

Escolher uma resposta.

a. as N versões diferentes executadas em paralelo replicam as falhas residuais de software que se encontram dormentes do código fonte 🗡

•	b. projetos independentes se traduzem em defeitos randômicos dos componentes, ou seja, independência estatística dos defeitos √
0	c. projetos independentes resultam em defeitos correlacionados de componentes, que rapidamente exaurem a redundância disponível <sup>⋆</sup>
С	d. usar N cópias do mesmo programa ou N versões diferentes não influencia a tolerância a falhas de um sistema 🗡

Notas relativas a este envio: 1/1.

Question14 Notas: 1

Considerando a abordagem de programação N-versões, assinale verdadeiro ou falso de acordo com o artigo de Florio e Blondia.

Avizienis afirma que a geração das versões de forma independente reduz significativamente os defeitos correlacionados.

O uso de N cópias da mesma versão evita a replicação de falhas dormentes residentes no código da versão.

Vários experimentos e estudos teóricos mostram que nem sempre é correto assumir que a geração das versões de forma independente reduza significativamente os defeitos correlacionados.

### Correto

Notas relativas a este envio: 1/1.

Question15

Notas: 1

Comparando as duas abordagens, blocos de recuperação e programação N-versões, selecione os recursos adequados.

Mecanismo usado para seleção de resposta na abordagem conhecida como programação N- versões:  Correto Notas relativas a este envio: 1/1.  Question16 Notas: 1  Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo usado para seleção de resultado, podemos afirmar que a primeira abordagem usa um mecanismo genérico independente da aplicação pendente da aplicação enquanto a segunda usa um mecanismo genérico independente da aplicação enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um mecanismo enquanto a segunda usa u	Mecanismo usado para seleção de resposta na abordagem conhecida como blocos de recuperação:	teste de aceitação
Notas relativas a este envio: 1/1.  Question16 Notas: 1  Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo usado para seleção de resultado, podemos afirmar que a primeira abordagem usa um mecanismo específico dependente da aplicação enquanto a segunda usa um mecanismo genérico independente da aplicação enquanto a segunda usa um enquanto a segunda usa um mecanismo los describidos de envio: 1/1.  Question17 Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas correta enquanto a segunda abordagem permite uma única saída correta enquanto a segunda abordagem permite enquanto a segunda abordagem permite uma única saída correta enquanto a segunda abordagem permite uma única		consenso ou algoritmo de votação
Notas relativas a este envio: 1/1.  Question16 Notas: 1  Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo usado para seleção de resultado, podemos afirmar que a primeira abordagem usa um mecanismo específico dependente da aplicação enquanto a segunda usa um mecanismo genérico independente da aplicação enquanto a segunda usa um enquanto a segunda usa um mecanismo sepecífico dependente da aplicação enquanto a segunda usa um enquant	Correto	
Notas: 1  Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo usado para seleção de resultado, podemos afirmar que a primeira abordagem usa um mecanismo genérico independente da aplicação e específico dependente da aplicação e enquanto a segunda usa um mecanismo Notas relativas a este envio: 1/1.  Question17  Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas correta enquanto a segunda abordagem permite uma única saída correta enquanto a segunda abordagem permite		
afirmar que a primeira abordagem usa um mecanismo  genérico independente da aplicação  genérico independente da aplicação  v  Correto  Notas relativas a este envio: 1/1.  Question17  Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite  diferentes saídas corretas  v enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um específico dependente da aplicação  enquanto a segunda usa um usa um específico dependente da aplicação  enquanto a segunda usa um usa um específico dependente da aplicação  enquanto a segunda usa um usa usa um usa um usa um usa um usa usa um usa um usa usa um usa		
afirmar que a primeira abordagem usa um mecanismo  genérico independente da aplicação  Correto  Notas relativas a este envio: 1/1.  Question17  Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite  diferentes saídas corretas  específico dependente da aplicação  enquanto a segunda usa um  enq	Comparando as duas abordagens, blocos de recuperação e programação N-versões, em relação ao mecanismo	usado para seleção de resultado, podemos
mecanismo  genérico independente da aplicação  Correto  Notas relativas a este envio: 1/1.  Question17  Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite  diferentes saídas corretas  enquanto a segunda abordagem permite  uma única saída correta  • enquanto a segunda abordagem permite	específico dependente da anlicacão	
Correto Notas relativas a este envio: 1/1.  Question17 Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta	animal que a primera aboradgem aba animo establica de se	gunda usa um
Notas relativas a este envio: 1/1.  Question17  Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta	mecanismo	
Notas relativas a este envio: 1/1.  Question17  Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta	Carrata	
Question17 Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta.		
Notas: 1  Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta.	Notas relativas a este envio. 1/1:	
Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeira abordagem admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta.	Question17	
admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta .	Notas: 1	
admite diferentes saídas corretas enquanto a segunda abordagem permite uma única saída correta .	Comparando blocos de recuperação e programação N-versões quanto às saídas, podemos afirmar que a primeir	ra ahordagem
admite — — enquanto a segunda abordagem permite — —.	differentes spidas corretos	
Correto	admite 🖳 ———————————————————————————————————	<u> </u>
	Correto	
Notas relativas a este envio: 1/1.		

Question18 Notas: 1

Comparando blocos de recuperação e programação N-versões quanto a estratégia de execução, podemos afirmar que a primeira al	oordagem e m sua
forma original implementa uma estratégia de execução sequencial enquanto a segunda abordagem permite execução concor	rrente .
Correto	
Notas relativas a este envio: 1/1.	
Question19	
Notas: 1	
Considerando conclusões de Florio e Blondia a respeito de programação N-versões, podemos afirmar que (assinale <b>verdadeiro</b> ou	falso):
A aplicabilidade da programação N-versões está restrita a um pequeno número de linguagens de programação como Java e C++.	Falso
O componente conhecido como EE (executivo do ambiente de programação N-versões) é simples, mas dependente da aplicação.	Falso
Programação N-versões não é adequada a sistemas de tempo-real.	Falso
Programação N-versões foi adotada com sucesso em várias áreas de aplicação incluindo aplicações de segurança crítica espaciais e aéreas.	Verdadeiro ▼

Correto Notas relativas a este envio: 1/1.

Question20

Notas: 1

Em relação a programação N-versões, assinale **verdadeiro** ou **falso** para as sentenças a seguir:

Portabilidade da programação N-versões é restrita pela portabilidade do executivo do ambiente N-versões e pela portabilidade de cada uma das versões.

A adoção de programação N-versões sempre implica em penabilidades em relação a manutenibilidade e portabilidade.

Verdadeiro

A programação N-versões nas arquiteturas 1T/NH/NS e NT/1H/NS tem uma função de custo que cresce linearmente com o número de versões.

A adoção de programação N-versões implica em um aumento substancial no custo de desenvolvimento e manutenção.

Verdadeiro

## Correto

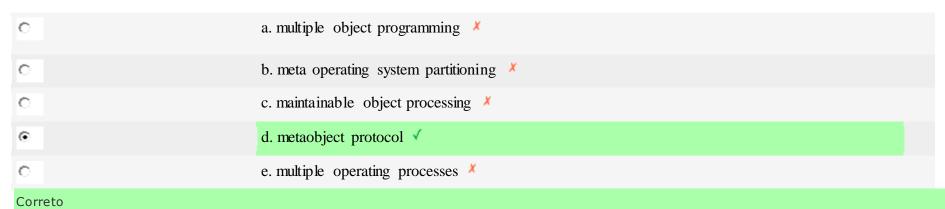
Notas relativas a este envio: 1/1.

Question21

Notas: 1

No item 3.2 do artigo de Florio e Blondia aparece a sigla MOP. MOP significa:

Escolher uma resposta.



Notas relativas a este envio: 1/1.

Question22

Notas: 1

Segundo Florio e Blondia, técnicas de estruturação genéricas como o método MOPs (item 3.2 do artigo) permite, em alguns casos, alcançar graus adequados de:

Escolher uma resposta.

0	a. facilidade na implementação de técnicas de tolerância a falhas em qualquer linguagem de programação <sup>✗</sup>
С	b. adaptabilidade e portabilidade ×
0	c. desempenho e dependabilidade 🗡
0	d. correção e desempenho
•	e. flexibilidade, transparência e separação de interesses funcionais e não funcionais ✓

# Correto

Notas relativas a este envio: 1/1.

Question23

Notas: 1

Segundo Florio e Blondia, a ideia básica do método MOPs é:

Escolher uma resposta.

a. adotar uma linguagem orientada a objetos como C++ ou Java e protocolos de comunicação certificados pelos maiores fabricantes para facilitar a interoperabilidade 🗡

0	b. abrir a implementação dos protocolos de comunicação inter-objetos de forma a inserir comandos específicos e padronizados de tolerância a falhas que os objetos são obrigados a executar X
•	c. abrir a implementação do executivo de execução (runtime executive) de uma linguagem orientada a objetos de tal forma que o programador possa adotar semânticas diferentes e customizadas, ajustando assim a linguagem para as necessidades do usuário e do ambiente ✓
C	d. abrir a implementação do compilador de uma linguagem de alto nível de tal forma que o programador possa adotar semânticas diferentes e customizadas, ajustando assim a linguagem para as necessidades do usuário e do ambiente X
0	e. abrir a implementação do sistema operacional de tal forma que o programador possa desenvolver novas chamadas do sistema (sys call), ajustando assim o sistema operacional para as necessidades do usuário e do ambiente 🗡

# Correto

Notas relativas a este envio: 1/1.

# Question24

Notas: 1

Usando MOPs o programador pode modificar o comportamento de características fundamentais como invocação de métodos, criação e destruição de objetos. Para fins de implementação de técnicas de tolerância a falhas, essa possibilidade de modificação permite:

# Escolher uma resposta.

0	a. a substituição dinâmica de programas errados do usuário por programas corretos do sistema 🗡
0	b. o aumento significativo do desempenho 🗶
0	c. a verificação automática da correção de programas 🗶
•	d. a gerência transparente de redundância espacial e temporal ✓

# Correto

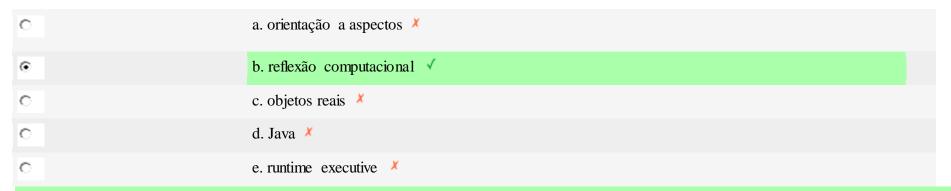
Notas relativas a este envio: 1/1.

Question25

Notas: 1

O conceito chave de suporte a MOPs (item 3.2 do artigo de Florio e Blondia) é:

Escolher uma resposta.



# Correto

Notas relativas a este envio: 1/1.

Question26

Notas: 1

MOPs oferecem ao programador de metanível uma representação de um sistema como um conjunto de elementos que representam e refletem propriedades de objetos reais. Esses elementos são chamados de:

Escolher uma resposta.

a. objetos tolerantes a falhas ×

C	b. objetos funcionais X		
0	c. objetos confiáveis 🗶		
C	d. objetos de tempo real X		
•	e. metaobjetos ✓		
Correto			
Notas relativas a este envio:	1/1.		
Question27 Notas: 1			
De acordo com Florio e Blondia, item 3	3.2, preencha as lacunas:		
O programador dos recursos de tolerância a falhas define uma quantidade de protocolos de metaobjetos e os associa invocações de métodos ou outros casos gramaticais invocações de métodos ou outros casos gramaticais entra um certo caso gramatical o/a protocolo correspondente é transparentemente executado .			
Correto			
Notas relativas a este envio:	1/1.		
Question28 Notas: 1			
Segundo Florio e Blondia, a abordagem baseada em MOPs parece constituir uma técnica promissora para a adoção efetiva, coerente e transparente de mecanismos e técnicas de TF. Assinale verdadeiro ou falso:			
Nenhuma evidência experimental ou analítica permite estimar a praticidade e generalidade da abordagem MOPs.			

Alguns estudos na década de 90 comprovam a eficiência de MOPs em alguns casos.	Verdadeiro	•
Não há dúvidas que MOPs representa uma solução prática para a efetiva integração na aplicação de usuários da maioria dos mecanismos de TF existentes.	Falso	<u> </u>

Correto Notas relativas a este envio: 1/1.

# INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ▶ FTF 2012/2 ▶ Questionários ▶ Redundância, confiabilidade e disponibilidade ▶ Revisão da tentativa 1

# Redundância, confiabilidade e disponibilidade

# Revisão da tentativa 1

# Terminar revisão

Iniciado em	domingo, 7 outubro 2012, 12:34	
Completado em	domingo, 7 outubro 2012, 16:07	
Tempo	3 horas 33 minutos	
empregado		
Notas	30.9/36	
Nota	<b>85.83</b> de um máximo de 100( <b>86</b> %)	

# Notas:

1

1

Johnson, B.W. no artigo: "Fault Tolerance", The Electrical Engineering Handbook, Ed. Richard C. Dorf Boca Raton: CRC Press LLC, 2000, apresenta inicialmente uma breve introdução aos conceitos de tolerância a falhas. Observe que a nomenclatura usada algumas vezes difere levemente da de outros autores.

De acordo com Johnson, o que significa tolerância a falhas?

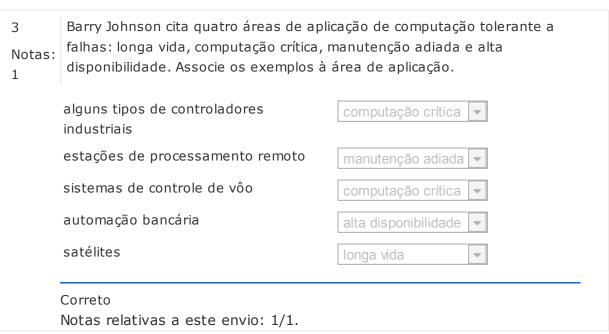
Escolher uma resposta.

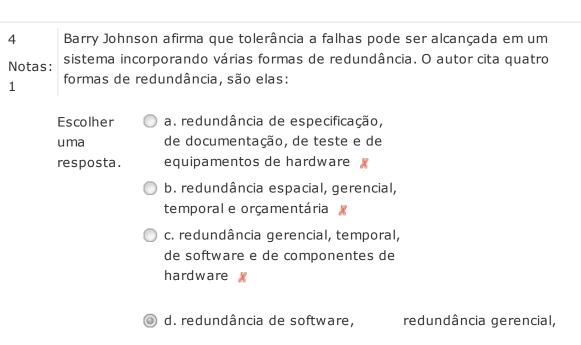
- a. a habilidade do sistema de continuar a realizar corretamente suas tarefas depois da ocorrência de falhas de hardware ou software
- b. a habilidade do sistema de prover replicação física de hardware para garantir redundância passiva
- c. garantia de correção do serviço fornecido por um sistema
- d. garantia que um sistema não vai provocar danos a pessoas ou ao meio ambiente
- e. habilidade do sistema de reconhecer e sinalizar a ocorrência de um defeito

# Correto

Notas relativas a este envio: 1/1.

2 De acordo com Barry Johnson, o que é uma falha? Notas: 🔘 a. A falha é o resultado de uma propagação de eventos Escolher 1 que provoca a queda de um dado serviço. 🦹 uma resposta. 🍥 b. A falha é simplesmente qualquer dano físico, imperfeição, ou problema que ocorre em hardware ou software. 🎸 🔘 c. A falha é um fenômeno físico provocado unicamente por desgaste de componentes de hardware. 🦹 🔘 d. Uma falha é caracterizada quando um serviço se desvia da sua especificação. 🦹 🔘 e. A falha é a consequência de um projeto mal especificado. 🦹 Correto Notas relativas a este envio: 1/1.





	de componentes de hardware		
	<ul> <li>e. redundância de equipes de desenvolvimento, documentação, procedimentos e componentes de hardware </li> </ul>		
	Correto Notas relativas a este envio: 1/1.		
5 Notas: 3	Redundância passiva é uma forma de redundância de hardware que : usa o conceito de mascaramento de falha para esconder a ocorrência de falha e prevenir que a falha se manifeste como erro.	la	
	Correto Notas relativas a este envio: 3/3.		
6			
Notas:	Segundo Barry Johnson, redundância ativa é uma forma de redundância d  hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha	ós a	
Notas:	: hardware v que usa o conceito de detecção de falha v para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundânci	ós a	
Notas: 3 7	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.	ós a	
Notas: 3 7 Notas:	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.	ós a	
Notas: 3 7 Notas:	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.	ós a	
Notas: 3 7 Notas:	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.  I localização de falha redundância ativa	ós a	
Notas: 3 7 Notas:	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.  Ilocalização de falha redundância ativa recuperação de falha redundância ativa red	ós a	
Notas: 3 7 Notas:	hardware que usa o conceito de detecção de falha para, a por a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.  localização de falha redundância ativa detecção de falha redundância ativa detecção de falha redundância ativa redund	ós a	
Notas: 3 7 Notas: 1	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.  localização de falha redundância ativa redundância passiva redundância passiva redundância passiva redundância passiva redundância ativa redundância passiva redundância passiva redundância passiva redundância passiva redundância passiva redundância ativa redundância passiva	ós a	
6 Notas: 3 7 Notas: 1 8 Notas:	hardware que usa o conceito de detecção de falha para, apor a ocorrência da falha, remover a falha do sistema. Técnicas de redundância ativa requerem que o sistema realize reconfiguração para tolerar falha Correto  Notas relativas a este envio: 3/3.  Associe a técnica à forma de redundância que a emprega.  localização de falha redundância ativa redundância passiva redundância passiva redundância passiva redundância passiva redundância ativa redundância passiva redundância passiva redundância passiva redundância passiva redundância passiva redundância ativa redundância passiva	ós a	

Avizienis quando apresenta técnicas de tolerância a falhas as divide em 9 dois grande grupos, detecção de erros e recuperação, essa última Notas: englobando tratamento do erro e tratamento da falha. A classificação de 1 Barry Johnson é diferente. Johnson fala em detecção de falhas (o que seria equivalente a detecção de erros do Avizienis) e mascaramento de falhas. Onde o "mascaramento de falhas" se encaixa na classificação de Avizienis? 🔘 a. detecção de erros concorrente 🦹 Escolher uma 🔘 b. tratamento de falha por recuperação 🦹 resposta. 🔘 c. tratamento de erro por rollforward 🦹 🍥 d. tratamento de falha por isolamento 🦹 🔘 e. tratamento de erro por compensação 🧹 Errado Notas relativas a este envio: 0/1. O que é a distância de Hamming entre duas palavras binárias? 10 Notas:

O que é a distância de Hamming entre duas palavras binárias?

Notas:

Escolher

uma

resposta.

D. quantidade de retransmissões necessárias para as palavras serem transmitidas sem erro 
C. diferença entre as paridades da palavras 

uma

c. diferença entre as paridades da palavras 

Notas relativas a este envio: 1/1.

Determine a distância de código dos seguintes códigos binários (cada linha Notas:

1

000,01	1, 101, 110	2	
00, 01,	10, 11	1	
001,010	0, 100, 111	2	
0101, 10	010	4	
Correto	elativas a este envio: 1/1.		
Notas le	dativas a este elivio. 1/1.		
13 O que é			
	Escolher		
resposta	b. números de bits ne	cessários para codificar as palavras	
	c. número máximo de corrigidos	bits errados que podem ser	
	O d. número máximo de	erros que podem ser detectados 🦹	
		de retransmissões necessárias para seja transmitida sem erro 🦹	
Correto Notas re	elativas a este envio: 1/1.		
_	de 1 , consequentemer	oar, apresenta uma distância de nte pode detectar 1 v bit(s) com	
	ente correta Elativas a este envio: 1/3.		
Notas ic	nativas a este envio. 1/5.		
15 O conce Notas: Johnson		redundância no tempo, segundo Barry	
Escolher uma resposta	comparar os resultado	ío em dois ou mais computadores e os para verificar se existe	
		áo duas ou mais vezes e comparar os car se existe discrepância ∢	
	computadores para qu	ios internos de dois ou mais ue os resultados sejam fornecidos o instante de tempo 🚜	
	d. repetir n vezes uma resultados	a computação e realizar a média dos	

resultado seja correto 🦹 Errado Notas relativas a este envio: 0/1. O problema principal associado a redundância no tempo, segundo Barry 16 Johnson, é: Notas: 1 a. sincronizar os diversos computadores que operam em Escolher paralelo 🦹 uma resposta. b. o número de repetições necessárias prejudica o desempenho do sistema 🦹 🔘 c. não ser adequada para detecção de falhas transitórias d. não ser adequada para detecção de falhas permanentes 🔏 e. garantir que o sistema tenha os mesmos dados para manipular a cada vez que ele realiza a computação redundante 🗸 Correto Notas relativas a este envio: 1/1. Responda verdadeiro ou falso segundo as afirmações de Barry Johnson. 17 Notas: Mesmo que uma falha transiente ocorra e w 1 corrompa completamente os dados de entrada, ainda assim repetir a computação no tempo é suficiente para tolerar falhas transientes. Para tolerar falhas transientes usando falso redundância temporal é necessário codificar os dados antes da transmissão da segunda computação e decodificá-los no recebimento. Um dos maiores potenciais de redundância verdadeiro temporal é a habilidade de detectar falhas permanentes usando apenas um mínimo de hardware extra. Correto Notas relativas a este envio: 1/1. 18 Barry Johnson apresenta 3 exemplos de redundância de software. São eles: Notas:

🔘 a. Programação auto-verificadora, programação ativa e

programação passiva. 🦹

1

Escolher

uma

🔘 e. repetir exaustivamente uma computação até que o

resposta. 

b. Programação modular, programação ágil e programação extrema. 

c. Programação N-versões, blocos de recuperação e programação N- auto-verificadora. 

d. Programação com estepe quente, programação com estepe frio e programação diversitária. 

e. Programação N- auto-verificadora, verificação de consistência e verificação de capacidade. 

Correto Notas relativas a este envio: 1/1.

As 3 abordagens de redundância de software mencionadas por Barry 19 Johnson têm em comum: Notas: 1 🔘 a. exigir N versões diferentes de um mesmo programa 🧹 Escolher uma 🔘 b. executar todas as versões do programa em paralelo 🦹 resposta. 🔘 c. votar sobre o resultado de N versões 🦹 🔘 d. tolerar N-1 falhas 🦹 e. exigir um teste de aceitação para cada versão do programa 🦹 Correto Notas relativas a este envio: 1/1.

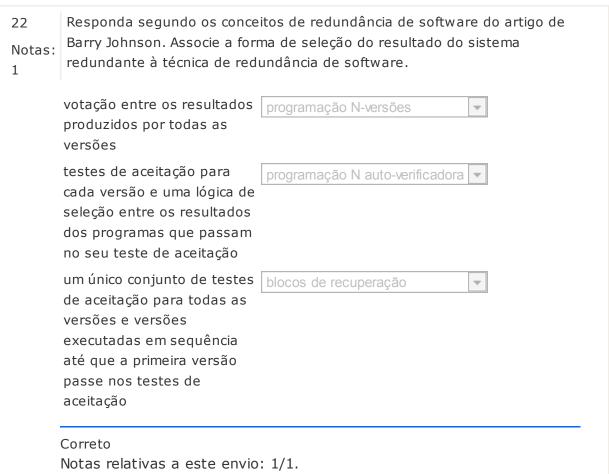
Na técnica de redundância conhecida como programação N-versões, o 20 número de falhas que podem ser toleradas com N versões de um programa Notas: é: 1 a. N / 2 Escolher uma resposta. C. N  $\bigcirc$  d. (N + 1) / 2 \*e. N - 1 Correto Notas relativas a este envio: 1/1.

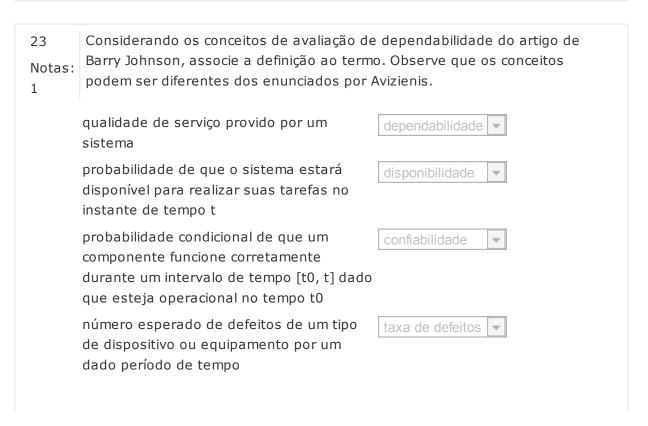
Notas:

Na técnica de redundância conhecida como programação N-autoverificadora (N self-checking), o número de falhas que podem ser toleradas com N versões de um programa é:

Escolher a. (N -1) / 2 uma b. (N + 1) / 2 resposta.



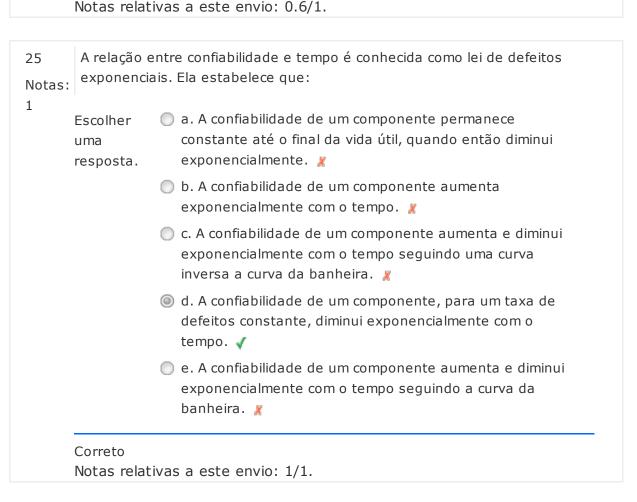


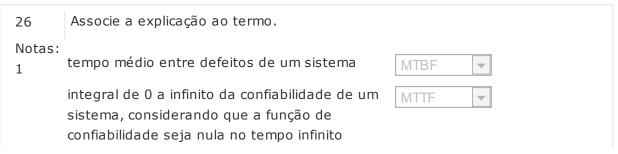


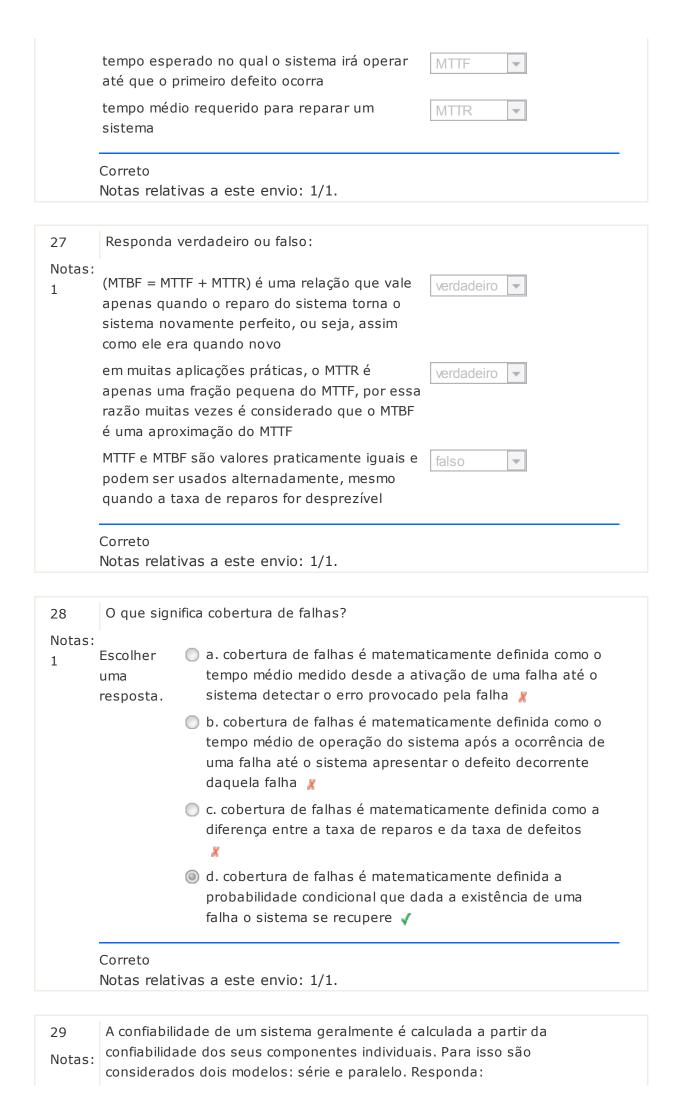
Correto

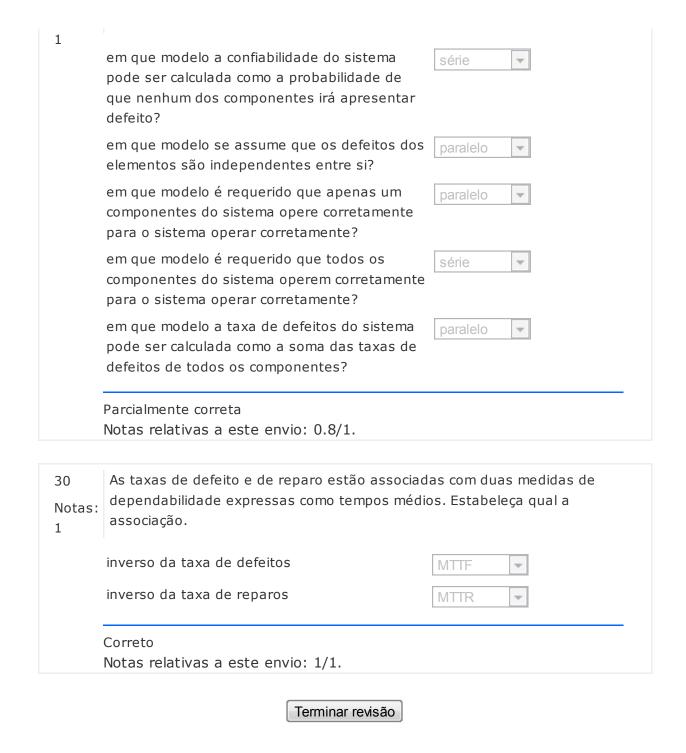
Notas relativas a este envio: 1/1.

24 Em relação a função de taxa de defeitos podemos afirmar que (responda verdadeiro ou falso): Notas: 1 é uma função monotonicamente crescente é claramente constante é claramente dependente do tempo verdadeiro pode ser expressa em termos da função de Escolher... confiabilidade inicialmente descresce, fica aproximadamente falso constante por um longo período, e depois cresce exponencialmente Parcialmente correta









Você acessou como João Luiz Grave Gross (Sair)

FTF 2012/2