

Temática 1 - Fundamentos de Sistemas Distribuídos

Qual das seguintes afirmações sobre a arquitetura cliente-servidor na web é falsa?

- | | |
|---|--|
| A | HTTP é um protocolo sem estado. |
| B | URLs identificam recursos exclusivamente na <i>web</i> . |
| C | HTML é processado apenas no servidor. ✓ |
| D | A escalabilidade é limitada pelo estado mantido no servidor. |
| E | O cliente inicia sempre a comunicação. |
| F | TCP/IP é opcional para HTTP/3. |
| G | Os <i>cookies</i> podem simular estado em HTTP. |
| H | A camada de apresentação é responsável pela “renderização” do HTML. |

Preencha o espaço vazio com o conceito mais correto.



1

Multicasting



é uma alternativa ao

broadcasting em redes Ethernet, reduzindo o tráfego desnecessário ao enviar mensagens apenas para grupos específicos.

▪ Unicast

▪ Token Passing

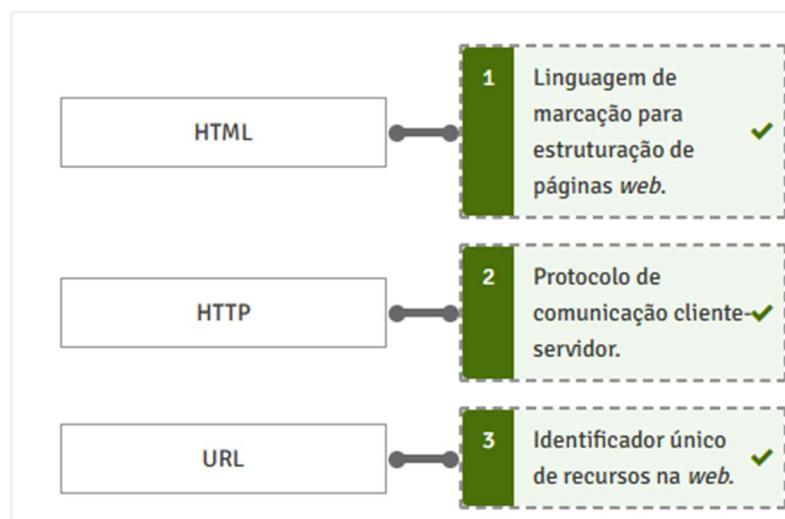
▪ Flooding

▪ Ethernet Frame

▪ ARP (Address Resolution Protocol)

▪ STP (Spanning Tree Protocol)

Relacione cada conceito com a sua descrição:



Temática 2 - Comunicação entre Processos

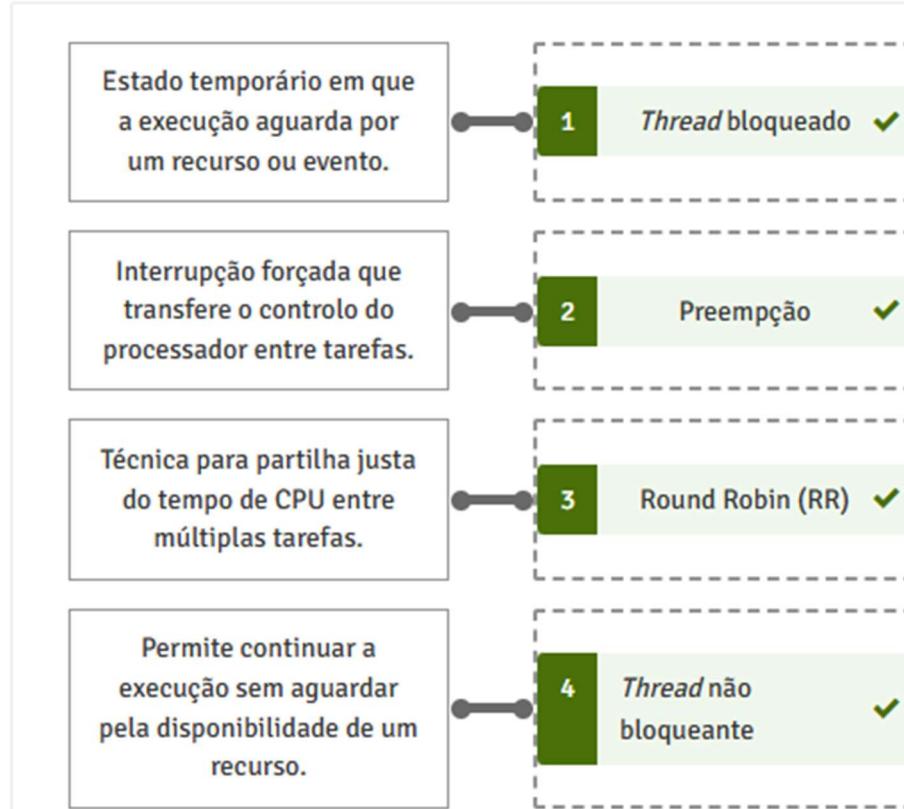
Preencha os espaços vazios com o conceito mais correto.

O protocolo **1** **HTTP** permite respostas sem armazenamento de estado no servidor, enquanto que **2** **TCP/IP** requer confirmação adicional.

Correct answers:

2 **RRA**

Relacione cada conceito com a sua descrição:



Temática 3 - Middleware

Compare SOAP e REST em termos de interoperabilidade e segurança para integração de serviços heterogéneos.

2/2,2

A escolha entre SOAP e REST para integração de serviços heterogéneos envolve, para mim, os trade-offs significativos em termos de interoperabilidade e também a segurança.

No que diz respeito à interoperabilidade:

- SOAP - Alta interoperabilidade via contratos formais e tipagem forte. Penso que é mais complexo (XML). Independente do transporte;
- REST - Alta interoperabilidade via padrões web - http, json; mais simples e leve, mas forte dependência de HTTP;

No que diz respeito à Segurança:

- SOAP - Segurança robusta ao nível da mensagem (WS-Security), permitindo encriptação de partes específicas do XML. Penso que é de certa forma complexo a implementar;
- REST - Segurança primária ao nível de transporte (HTTPS/TLS). Usa mecanismos web padrão - OAuth, JWT. Para mim é mais simples de implementar.

O que pretendia dizer é que a interoperabilidade em SOAP é alcançada pela formalidade dos contratos enquanto em REST é pela simplicidade dos padrões web. A segurança em SOAP é mais a nível da mensagem e complexa, enquanto em REST é mais focada na camada de transporte e mecanismos web padrão. Para mim a decisão final dependerá das prioridades específicas do projeto e do ambiente em que os serviços serão integrados.

Como forma de conclusão para os Serviços Heterogêneos:

- o SOAP é preferível para ambientes corporativos regulamentados e complexos, onde a formalidade do contrato e segurança são questões extremamente críticas, mesmo tendo uma maior sobrecarga; é melhor para ambientes empresariais legacy, altamente regulamentados que exigem conformidade e segurança in-message;
- o REST é preferível para APIs web, microserviços e mobile, onde a agilidade e performance e facilidade de uso são prioridades; é melhor para APIs web modernas, microserviços, mobile - onde a agilidade e performance são alvos prioritários.

Relacione cada tecnologia ou conceito com a sua função principal:



Temática 4 - Partilha de Dados

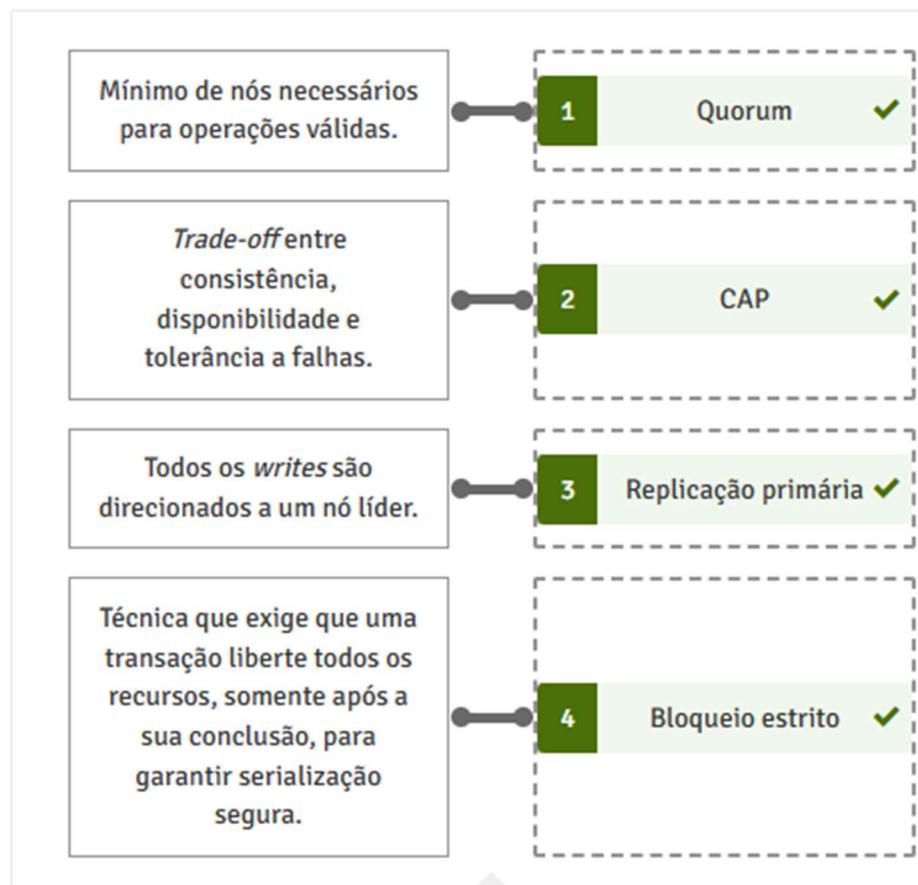
Qual é a estratégia que não mantém coerência em *caches* distribuídos?

A	Invalidação
B	Atualização
C	<i>Timeout</i>
D	<i>Broadcast</i>
E	Quorum
F	Replicação primária
G	Serialização estrita
H	Ignorar conflitos

Preencha o espaço vazio com o conceito mais correto.

O teorema **1** CAP  estabelece que os Sistemas Distribuídos não podem garantir simultaneamente consistência, disponibilidade e tolerância a partícões."

Relacione cada termo com a sua definição:



Temática 5 - Serviços em Ambientes Distribuídos

Explique por que a transparência é um requisito fundamental em sistemas de arquivos distribuídos e descreva os 5 principais tipos de transparência que devem ser assegurados.

2/2,2

A transparencia é um requisito fundamental no meu entender em sistemas de arquivos distribuídos porque o seu objetivo primordial é fazer com que um conjunto de recursos de armazenamento e computacao pareca como um unico sistema de arquivos local e unificado (para os utilizadores e aplicacoes). Sem isto, para nos utilizadores e programdores, teriamos a todo o tempo de lidar com a complexidade da distribuicao como por exemplo: a localizacao dos arquivos, a replicacao de dados... - o que trasformaria o sistema dificil de usar e mesmo de desenvolver.

Os 5 principais tipos de transparencia que devem ser assegurados sao:

1. **A transparencia de acesso** - Oculta o fato de que os dados estao armazenados remotamente, ou seja : nós como utilizadores e as aplicacoes acedem a ficheiros remotos exatamente da mesma forma que acedem aos ficheiros locais onde a interface é unificada, independentemente da localizacao fisica do ficheiro;
2. **Transparencia de Localizacao** - Oculta onde o ficheiro esta fisicamente armazenado na rede ou seja, um ficheiro /usr/local/bin/prog pode estar em qualquer servidor da rede sem que o utilizador precise de saber qual é e assim permite que os ficheiros sejam movidos sem alterar os caminhos de acesso;
3. **Transparencia de Concorrencia** - Oculta o acesso simultaneo de multiplos utilizadores ao mesmo ficheiro garantindo assim que as operacoes concorrentes sejam executadas de forma consistente - isto é frequentemente alcançado atraves de mecanismos de bloqueios ou controlo de concorrencia.
4. **Transparencia de Replicacao** - Oculta o fato de que os ficheiros possam ter multiplas copias espalhadas por diferentes servidores, para nos utilizadores aparecer como uma unica entidade logica;
5. **Transparencia de Falha** - Oculta falhas de componentes e recuperacoes permitindo assim que o sistema continue a operar ou seja, se um servidor falhar, o sistema deve ser capaz de continuar a trabalhar - por exemplo, usando uma "replica" - ou recuperar sem que nos utilizadores percebamos a interrupcao.

Em suma, para mim, a transparencia é um requisito fundamental pois consegue: - simplificar o uso, simplificar o desenvolvimento de aplicacoes, melhorar a escalabilidade e promover a portabilidade.

Preencha os espaços vazios com o conceito mais correto.



A **1** consistência em sistemas de arquivos distribuídos garante que vários clientes possam aceder e modificar o mesmo ficheiro sem causar conflitos ou perda de dados.

Associe cada tecnologia ou conceito com a sua respetiva descrição:

