

## Capítulo 1 Exercícios

1.1 - alguns exemplos de recursos de hardware não: Processadores, onde um SD pode dividir o processamento de recursos entre vários máquinas, Memória e Armazenamento onde ocorre o mesmo coisa que os processadores, Disparitivos de Redes e Impressoras; exemplos de Software não: Programas de Páginas, Archives & Serviços Web onde empresas podem usar esses serviços ou recursos de maneira compartilhada em nuvem, codigo fonte onde usuários diferentes podem trabalhar num mesmo projeto e conhecendo ou não, como um software livre; Bibliotecas Compartilhadas onde várias aplicações podem utilizá-las de forma independente e simultânea.

1.2 - A única forma para sincronizar máquinas num SD é através de mensagens, o fator limitante desse método é que a falta de um relógio global e de uma noção única de tempo faz com que a comunicação sempre envolva muita demora na troca de mensagens, dependendo do conteúdo do código, volume de dados, capacidade de processamento, entre outras coisas.

1.3 - As vantagens de um servidor único é quando você fizer alterações dentro do jogo todos veem de maneira simultânea, porém ao centralizar o servidor ele fica muito mais suscetível a falhas e quando falha todos os jogadores notam o problema correndo o risco da jogo inteiro ficar parado ou, desvantagem era que num PDA é mais difícil de ocorrer e mais fácil de resolver, distribuindo o servidor em várias máquinas onde caso uma pequena parte falhe é provável que não seja o jogo inteiro que por conta disso falhe.

1.4 - Com base na localização da estação e na conexão à rede Wi-fi, o PDA pode automaticamente receber informações relevantes sobre a estação, como horários de trem em tempo real, restaurantes próximos, banheiros, bilhetes etc. Alguns serviços que implicam neste tipo de sistema são, desenvolvimento de um painel portal de informações para gerenciar as informações relevantes no PDA, a compatibilidade de diversas dispositivos diferentes, além da privacidade do usuário, além da manutenção periódica das informações que precisam estar sempre atualizadas pois impactam imediatamente na experiência do usuário.

1.5 - Ter um roteador em rede é bastante vantajoso pois você poderá automatizar o processamento de dados de todo serviço, não limitando seu uso só à capacidade de computação da sua máquina e distribuir essa computação em rede também faz com que o tratamento de dados seja menor, além de diminuir a possibilidade de falha de usuário com seus arquivos e máquina já que em rede o backup é mais facilitado.

1.6 - HTML, URLs e HTTP tornaram a internet acessível globalmente. Qualquer pessoa com uma conexão à internet pode salvar informações e recursos online além de serem amplamente reproduzidos e competirem com uma variedade de navegadores, sistemas operacionais e dispositivos, proporcionando uma experiência consistente para os usuários e também facilitam o uso das rotinas web como por exemplo as URLs simplificam a localização de recursos na web, permitindo que os usuários obtenham informações com facilidade com facilidade, apenas digitando um endereço.



- 1.7 - 1. Heterogeneidade de linguagem: Um mecanismo de comunicação com RPC ou Web Services para traduzir chamadas entre diferentes linguagens.
2. de Dados: Converte dados em uma representação comum, como JSON, que ambas as linguagens podem entender.
3. de Hardware: Não é geralmente um problema, desde que a rede e os protocolos de comunicação sejam compatíveis.
4. de S.O.: Enrolha protocolos e bibliotecas múltipla forma para mitigar problemas de S.O.
5. de plataforma: Uso tecnologias de middleware, como CORBA ou serviços web, para permitir que objetos em diferentes plataformas interajam.

1.8 - Heterogeneidade refere-se a lidar com diferenças em tecnologias e ambientes, como linguagens de programação diferentes ou sistemas operacionais variados. Isto é importante para garantir que sistemas diversos podem se comunicar.

Abrtura do Sistema envolve a capacidade de adicionar novos serviços ou recursos ao sistema.

sistema sem perturbar o que já existe. É importante para permitir que o sistema evolua com grandes modificações.

1. Identificação de Usuários: Certifique-se de que você sabe quem são os usuários autenticados.
2. Controle de Acesso: Refine regras para quais usuários têm permissão para usar as operações protegidas.
3. Segurança na Comunicação: Mantenha a comunicação entre cliente e servidor segura, usando protocolos.
4. Auditório e Registro: Registre todas as atividades relacionadas às operações protegidas para rastreamento.

Para proteger informações confidenciais:

1. Criptografia: garante que os dados confidenciais sejam armazenados e transmitidos de forma criptografada.
2. Gerenciamento de chaves: Crie chaves criptográficas usadas para proteger os dados.
3. Proteção contra ataques: Estude paperudo para defender as operações protegidas contra ataques.

## 1.10

1. Nomenclatura Simples: Use nomes curtos e fáceis de lembrar para recursos.
2. Estrutura Hierárquica: Organize recursos em uma estrutura de árvore para facilitar a pesquisa.
3. Índices Eficientes: Implemente índices para acelerar a recuperação de recursos.
4. Escalabilidade Horizontal: Planeje para adicionar mais servidores conforme a demanda cresce.
5. Cache Inteligente: Use cache para armazenar cláusulas frequentemente acessadas.
6. Armazenamento Eficiente: escolha um sistema de armazenamento eficiente.
7. Operações Asíncronas: Use operações assíncronas para evitar bloqueios.
8. Monitoramento Contante: Monitor o desempenho e ajuste conforme necessário.
9. Pimensionamento Automático: Configure sistemas que possam se adaptar automaticamente à carga.
10. Seguranças Escalares: Garante que a segurança para lidar com muitos usuários.

1.11 - Objeto Servidor: o objeto servidor pode falhar devido a um erro interno no código ou à falta de recursos. Para tornar o objeto servidor mais tolerante a falhas, você pode implementar estratégias de monitoramento de recursos e recuperação de erros.

Rede de Comunicação: A rede de comunicação entre o cliente e o servidor pode falhar devido a interrupções na conexão ou congestionamento da rede. Para tornar a comunicação mais tolerante a falhas, você pode implementar protocolos de rede resistentes, como o TCP, que garantem a entrega confiável de mensagens.

S.O. do servidor: O sistema operacional do servidor pode travar devido a erros de software, falta de recursos ou problemas de hardware. Para tornar o sistema operacional mais tolerante a falhas, você pode usar tecnologias de virtualização em contêineres para isolas aplicativos, de modo que as falhas em um aplicativo não afetem todo o sistema além de sistemas de backup.

1.18 - Recursos podem ser transferidos entre servidores em ambientes distribuídos para otimizar desempenho, escalabilidade e recuperação de falhas. Multicast (disponibilidade relativa) de pedidos para um grupo de servidores pode operar redundância e transparência de mobilidade, mas também gera sobrecarga de rede e gerenciamento. Soluções como平衡adores de carga não frequentemente provisórios para otimizar o desempenho e evitar problemas de sobrecarga. Funcionamento depende das necessidades específicas do sistema.

1.19 - "URL" significa "Uniform Resource Locator" é uma string que indica a localização de um recurso na web. Três exemplos de recursos da web que podem ser nomeados por URLs são páginas da web, imagens e arquivos para download.

1.20 - Um exemplo de URL HTTP é "`http://www.example.com:8080/pagina/exemplo.html?parametro=valor#anexa`". Sendo composta por esquema ("`http`"), o nome do host ("`www.example.com`"), a porta (opcional, "`8080`"), o caminho ("`/pagina/exemplo.html`"), a query

string ("? parametro = **p**.valor") o fragments ( "[# ancora](#)" ). A transparéncia de localização de um URL HTTP varia, mas neste exemplo, é alta, pois fornece informações detalhadas sobre a localização do recurso na Web.