**Introdução aos SDs**

1. “Um Sistema Distribuído (SD) é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente.” (Tanembaum, 2007). Um dos requisitos básicos para a implementação e o funcionamento de um sistema distribuído é a escalabilidade. O que define este conceito?

|  |
| --- |
| A escalabilidade é um requisito fundamental para sistemas distribuídos, e se refere à capacidade do sistema de lidar com um aumento no número de usuários, processamento de dados e carga de trabalho. De forma geral, a escalabilidade é a capacidade do sistema de crescer e lidar com mais demandas, sem que isso afete o desempenho e a qualidade do serviço prestado. |

1. Em um SD as Metas são perspectivas definidas como pontos cruciais, principalmente a viabilidade, custo, esforço e demais aspectos relacionados com a regra de negócio. Uma das metas de um SD é a “abertura do sistema”. Qual a sua relação com os chamados Middlewares?

|  |
| --- |
| A “abertura de um sistema” em um SD pode ser facilitada pelo uso de middlewares, que podem fornecer recursos como tradução de protocolos, segurança, gerenciamento de transações e serviços de mensagens. Dessa forma, os middlewares podem ajudar a tornar a integração e interoperabilidade de sistemas distribuídos mais fácil e eficiente. |

1. Ainda sobre as metas, o que define a meta de “transparência na distribuição”?

|  |
| --- |
| Um sistema transparente é aquele que é identificado como único para o usuário final, sem que o mesmo perceba localização, acesso, concorrência e o modo de funcionamento interno.  A meta de "transparência na distribuição" pode ser definida como um objetivo de tornar o processo de distribuição mais transparente para todas as partes interessadas envolvidas, como clientes, fornecedores e parceiros de negócios. Isso pode incluir medidas para garantir que as informações sobre a distribuição de produtos ou serviços sejam comunicadas de forma clara e precisa, de modo que todas as partes envolvidas compreendam as etapas do processo e os critérios para a distribuição. |

1. Qual a diferença existente entre Sistema de Processamento de Transações e Sistemas de Integração de Aplicações Empresariais?

|  |
| --- |
| Um Sistema de Processamento de Transações (SPT) é um sistema utilizado para gerenciar transações comerciais rotineiras, como processamento de vendas, pagamentos de contas, gerenciamento de estoques, entre outras atividades. Esses sistemas são caracterizados por alta disponibilidade, confiabilidade e segurança, e são projetados para executar operações em tempo real. Realizam operações sobre objetos ou sistemas que satisfazem as propriedades ACID (Atomicity, Consistency, Isolation, Durability), geralmente trabalham com transações aninhadas e utilizam de monitores TP (Transaction Processing) para gerenciamento de transações aninhadas;  Por outro lado, os Sistemas de Integração de Aplicações Empresariais (EAI) são projetados para integrar diferentes aplicativos e sistemas de informação utilizados por uma empresa. Eles permitem que diferentes sistemas "falem" uns com os outros, compartilhando informações e dados em tempo real. Eles são frequentemente usados para conectar diferentes sistemas legados e aplicativos de software para formar um ambiente de TI integrado e automatizado. Utilizam de RPC's (Remote Procedure Calls) e RMI's (Remote Method Invocations) para prover comunicação entre diversas aplicações diferentes, através da chamada de métodos e objetos remotos em uma rede. Também trabalham com Middlewares Orientados a Mensagens (MOM) para prover uma comunicação assíncrona;  Em resumo, enquanto os Sistemas de Processamento de Transações lidam principalmente com o processamento de transações individuais, os Sistemas de Integração de Aplicações Empresariais se concentram em conectar diferentes sistemas e aplicativos para criar uma infraestrutura de TI integrada e coesa. |

1. Um SD implementado com uma arquitetura centrada a dados tem por objetivo prover a integração de dados, onde os processos se comunicam por meio de um repositório comum. Esquematize abaixo um exemplo prático de um SD seguindo este estilo arquitetônico.

|  |
| --- |
| Arquitetura Centrada em Dados:  - A meta deste estilo arquitetônico é prover a integração de dados;  - Processos se comunicam por meio de um repositório comum;  - Novos componentes podem ser adicionados à arquitetura sem interação com outros clientes (Cada cliente executa seus acessos de forma independente);  EX: |

1. O que são os middlewares? Dê um exemplo de sistema que utilize este tipo de tecnologia.

|  |
| --- |
| Middlewares são softwares que atuam como intermediários entre diferentes sistemas, aplicativos ou componentes de software. Eles visam facilitar a interoperabilidade e integração de sistemas distribuídos e fornecem uma camada de abstração entre os diferentes sistemas e aplicativos, permitindo que eles se comuniquem de forma transparente e eficiente (mesmo que eles não tenham sido projetados originalmente para trabalhar juntos).  Um exemplo de sistema que utiliza middlewares é a arquitetura de microservices, que é cada vez mais comum em aplicações web modernas. Nesse tipo de arquitetura, os diferentes serviços que compõem uma aplicação são projetados como componentes independentes, cada um com sua própria interface de programação (API). Os middlewares atuam como intermediários que permitem que esses serviços se comuniquem e se integrem uns com os outros. |

1. Com a utilização de middlewares surgiu o conceito de software adaptativo. Como funciona essa abordagem?

|  |
| --- |
| Muitos profissionais adotam essa estratégia de middleware reconfigurável implementando API's, bibliotecas em variadas plataformas. E, esse conceito é conhecido como software adaptativo.  A abordagem de software adaptativo baseia-se na ideia de que os sistemas devem ser capazes de se adaptar a diferentes condições e requisitos, de forma dinâmica e automática. Isso significa que um software adaptativo pode ajustar seu comportamento e recursos em tempo real, de acordo com as mudanças no ambiente em que está sendo executado.  Este é um conceito interessante pois as aplicações distribuídas mudam continuamente. Mudanças incluem mobilidade, variação na Qualidade de Serviço (QoS), falha de hardware entre muitas outras. |

1. Para a criação de um software adaptativo é importante a utilização de três técnicas: separação de interesses, reflexão computacional e desenvolvimento de componentes. Explique cada uma delas.

|  |
| --- |
| Separação de Interesses: é uma técnica de modularizar os sistemas, separando as partes que implementam funcionalidades essenciais daquelas que implementam funcionalidades secundárias, visando separar as diferentes preocupações ou interesses que um sistema pode ter. Isso é feito para melhorar a modularidade do software e simplificar seu desenvolvimento e manutenção. As funcionalidades extras ou secundárias são confiabilidade, desempenho, segurança, etc...  Reflexão Computacional: é uma técnica que permite a um sistema monitorar e modificar seu próprio comportamento em tempo de execução. Isso é possível porque o sistema tem uma representação interna de si mesmo, que pode ser inspecionada e modificada durante a execução. A reflexão é usada para auto-otimização e auto-modificação de um programa.  Desenvolvimento de Componentes: é uma técnica de engenharia de software que visa criar sistemas a partir de módulos independentes, chamados componentes. Esses componentes podem ser desenvolvidos de forma separada e, em seguida, integrados para formar um sistema completo.  - Uma funcionalidade pode ser adicionada de forma estática (tempo de compilação) ou dinâmica (tempo de execução);  - Módulos podem ser carregados e retirados do sistema à vontade; |

1. Como são classificados os diversos tipos de Sistemas Distribuídos e como são subdivididos? Cite exemplos.

|  |
| --- |
| Os principais tipos de sistemas são:  **Sistemas de Computação Distribuído (SCD):**  Usado para realizar tarefas que exigem alta performance. São subdivididos em dois subgrupos:  - Computação em Cluster (Cluster Computing)   * Exemplo: Cluster de Beowulf   - Computação em Grade (Grid Computing)   * Exemplo: World Community Grid   **Sistemas de Informação Distribuídos (SID):**  Estão relacionados ao sistema de processamento de transações e integrações de vários sistemas de cooperação. São subdivididos em dois subgrupos:  - Sistema de Processamento de Transações e Integração de Aplicações Empresariais.  **Sistemas Distribuídos Pervasivos:**  Fornecem condições operacionais de funcionamento para dispositivos móveis. Os mais comuns são os sistemas domésticos (TV's, eletrônicos, smart phones), mas também envolve sistemas para controle de saúde (formação de redes Body Area Networks – BAN’s), redes de sensores, etc...   * + eHealth e mHealth |