1. O dimensionamento de uma infraestrutura de Cloud Computing é extremamente difícil pois requer o balanço de várias características. Que características deve ter em conta quando dimensiona? Porque não existe uma solução única para este problema?

O dimensionamento de uma infraestrutura em cloud obriga a equacionar múltiplas características que muitas vezes se contrapõem. Em primeiro lugar, há o equilíbrio entre **escala** e **elasticidade**: é necessário prever um provisionamento de recursos que suporte picos de carga sem incorrer em subutilização permanente. À escala junta-se o compromisso entre **desempenho** e **custo**. Acresce a **disponibilidade** e a **resiliência**: garantir SLAs elevados implica redundância geográfica, replicação de dados e mecanismos de failover, tudo a aumentar o custo e a complexidade. A **segurança** e a **conformidade** regulatória também influenciam as opções de arquitetura. Finalmente, é imprescindível considerar a **manutenibilidade** e a **observabilidade**, pois as infraestruturas elásticas exigem ferramentas robustas de monitorização, automação e orquestração.

Não existe uma solução única porque cada aplicação tem requisitos distintos, orçamentos e políticas de risco diferentes, e as condições variam com a evolução do negócio e da tecnologia. Os trade-offs entre desempenho, custo, segurança e fiabilidade são, por definição, contextuais, pelo que o dimensionamento deve ser sempre adaptado ao perfil de carga, à criticidade dos serviços e às restrições orçamentais em vigor.

2. Numa infraestrutura baseada em containers, os mesmos apresentam-se por definição associados a uma rede privada, isolada da rede do anfitrião. Explique o motivo por que isto acontece e como na tecnologia Kubernetes um serviço deve ser exposto à internet.

Os containers isolam processos utilizando namespaces de rede e pontes virtuais que os ligam a uma sub-rede privada do host. Este isolamento impede a contaminação e conflitos de porta entre aplicações, reforça a segurança ao restringir o alcance de cada container e assegura que todos os pacotes passem por camadas de filtragem e controlo (firewalls, políticas de rede). No Kubernetes, cada Pod recebe um CIDR interno e comunica através do kube-proxy e de CNI plugins, permanecendo invisível fora do cluster. Para expor um serviço à Internet, recorre-se tipicamente a um **Service** do tipo LoadBalancer (em cloud providers) ou NodePort, complementado por um recurso **Ingress** que implementa regras HTTP(S), TLS e encaminhamento avançado via um controlador. Desta forma, o tráfego externo entra pelo ponto de entrada definido, atravessa regras de routing e chega de forma segura e balanceada aos Pods desejados.

3. Enumere e defina 4 princípios da Orientação a Serviços (SOA).

Um dos pilares do SOA é o **Loose Coupling**, que exige que serviços dependam o mínimo possível das implementações uns dos outros, comunicando-se apenas através de mensagens padronizadas. O **Service Contract** define formalmente a interface e as políticas de uso, garantindo entendimento unívoco entre produtores e consumidores. A **Statelessness** recomenda que cada serviço não retenha estado entre chamadas, facilitando replicação e escalabilidade horizontal. Por fim, a **Reusability** incentiva que serviços sejam concebidos para múltiplos cenários de negócio, promovendo economia de esforço e consistência funcional, já que a lógica encapsulada pode ser combinada em diferentes composições sem duplicação de código.

4. Defina o que é uma SAML Assertion, para que serve e quais as vantagens da sua utilização.

Uma **SAML Assertion** é um documento XML assinado digitalmente que contém declarações sobre um sujeito, nomeadamente a autenticação realizada, atributos de identidade ou decisões de autorização. Opera entre um **Identity Provider** (IdP), que gera e assina as assertions, e um **Service Provider** (SP), que as valida e concede acesso a recursos sem requerer nova autenticação. A adoção de SAML permite implementar Single Sign-On entre domínios heterogéneos, reduzindo a gestão de credenciais e o risco de phishing, pois as credenciais nunca saem do IdP. A utilização de padrões abertos assegura interoperabilidade entre fornecedores distintos e integração transparente em ambientes corporativos que exigem robustez e controlo centralizado de identidade.

5. Em que consiste o padrão Publish/Subscribe? Este padrão pode ser comparado a que outro padrão? Quais as suas vantagens?

O padrão **Publish/Subscribe** baseia-se no desacoplamento entre produtores de eventos e consumidores através de tópicos ou canais geridos por um broker. O publisher publica mensagens sem conhecer os subscritores, e estes recebem apenas as categorias de evento a que subscreveram. Esta abordagem é comparável ao **Observer Pattern**, mas estende-se a sistemas distribuídos, suportando milhares de participantes e garantindo entrega assíncrona e tolerância a falhas. As principais vantagens residem no baixo acoplamento temporal, na flexibilidade de escalar ambos independentemente e na capacidade de processamento em paralelo de fluxos de eventos, fator crítico em arquiteturas baseadas em micro serviços e em sistemas de streaming de dados em tempo real.