 MAUÁ	ECM516 – Prova 1 ^o semestre	
Disc.: ECM516 - ARQUITETURA DE SISTEMAS COMPUTACIONAIS		
Curso: ENGENHARIA DE COMPUTAÇÃO		Série: 4a.

Aluno(a): GABARITO	RA:
Prof. Antonio Sergio Ferreira Bonato	
São Caetano do Sul, 25 de junho de 2022.	
Assinatura:	

Instruções da Prova

1. Avaliação **individual**.
2. Avaliação totalmente **sem consulta**.
3. Tempo total de avaliação **90 minutos**. Atenção ao tempo!
4. Utilize os **espaços indicados para cada resposta** de cada questão.
5. Caso precise de **rascunho utilize a última folha**.
6. A **prova** pode ser feita totalmente a **lápiz**.
7. Boa Prova!

1) (2,0 pontos) Quanto ao protocolo HTTP, explique:

a) A afirmação "O HTTP é *cliente-servidor*." e quais as consequências disso.

O protocolo HTTP é utilizado para comunicação entre cliente (navegador) e servidor Web. É baseado no modelo cliente/servidor ou requisição/resposta (request/response).

Uma das consequências é que um servidor nunca consegue "chamar" um cliente. Toda comunicação se baseia no cliente enviando uma requisição e o servidor enviando uma resposta. Deste modo um cliente nunca pode, por exemplo, ser notificado pelo servidor de uma mudança no seu estado, como ocorre no design pattern Observer.

b) A afirmação "O HTTP é *stateless*." e quais as consequências disso.

O HTTP é stateless, isto é, a princípio, nenhuma requisição é mantida no servidor. A consequência é que não é possível armazenar informações de uma transação em uma aplicação sem o uso de subterfúgios como cookies, por exemplo.

Se não fosse pelo uso de cookies e da memória do servidor de aplicação nunca seria possível manter um carrinho de compras, por exemplo.

2. (2,0 pontos) Defina os conceitos de arquitetura abaixo:

a) Arquitetura de Software

- Foco nos requisitos não funcionais.
- Define como o software será dividido em módulos e como estes módulos serão integrados.
- Estabelece o como será o controle do software, isto é, como os diversos módulos serão coordenados nas suas interações.
- Deve ser dividida em arquitetura de apresentação, arquitetura de aplicação e arquitetura de dados.
- Atualmente segue padrões como o MVC (Model-View-Controller) e os padrões de projeto do GoF (Gang of Four).

b) Arquitetura de Sistemas

- Foco nos requisitos não funcionais.
- Nível mais alto do que a Arquitetura de Software.
- Indica como o Sistema de Informação será disposto na infraestrutura de hardware e de software disponível.
- Define como ele se integra com outros sistemas de informação.
- Tipos comuns de arquitetura: Monolítica, Cliente/Servidor, Arquitetura em Camadas, Arquitetura Orientada a Serviços, Microserviços

3. (2,0 pontos) Relacione os design patterns da coluna da esquerda com seus objetivos na coluna da direita.

A	Abstract Factory	(H)	lidar com uma requisição de modo dinâmico
B	Factory Method	(B)	encapsular a escolha da classe concreta a ser utilizada na criação de objetos de um determinado tipo.
C	Singleton	(G)	notificar objetos sobre mudanças em outros objetos sem saber quem eles são.
D	Facade	(E)	combinar classes não relacionadas que possuam interfaces incompatíveis
E	Adapter	(A)	encapsular a escolha das classes concretas a serem utilizadas na criação dos objetos de diversas famílias.
F	Strategy	(D)	fornecer uma interface simplificada
G	Observer	(J)	possibilitar a criação de novos objetos a partir da cópia de objetos existentes.
H	Chain of responsibility	(I)	definir como um algoritmo deve ser implementado.
I	Template Method	(C)	permitir a criação de uma única instância de uma classe e fornecer um modo para recuperá-la.
J	Prototype	(F)	utilizar diferentes variações de um algoritmo

4. (2,0 pontos) Em uma determinada API, o **microserviço Performance** depende de dados existentes na base de dados pertencente a outros microserviços, o **microserviço Vendas** e o **microserviço Entregas**. Como essa funcionalidade poderia ser implementada sem o acesso direto a essas bases? Para resolver esse tipo de problema, os microserviços precisam se comunicar de alguma forma. Há dois tipos essenciais de comunicação entre microserviços: o síncrono e o assíncrono.

a) Explique como se dá a comunicação assíncrona entre serviços

A comunicação assíncrona tem algumas formas de implementação. Uma delas é baseada em eventos. Seu funcionamento se dá em função de um barramento de eventos. Neste modelo de comunicação assíncrona, cada microserviço se conecta ao barramento de eventos. Uma vez conectado ao barramento, um microserviço pode:

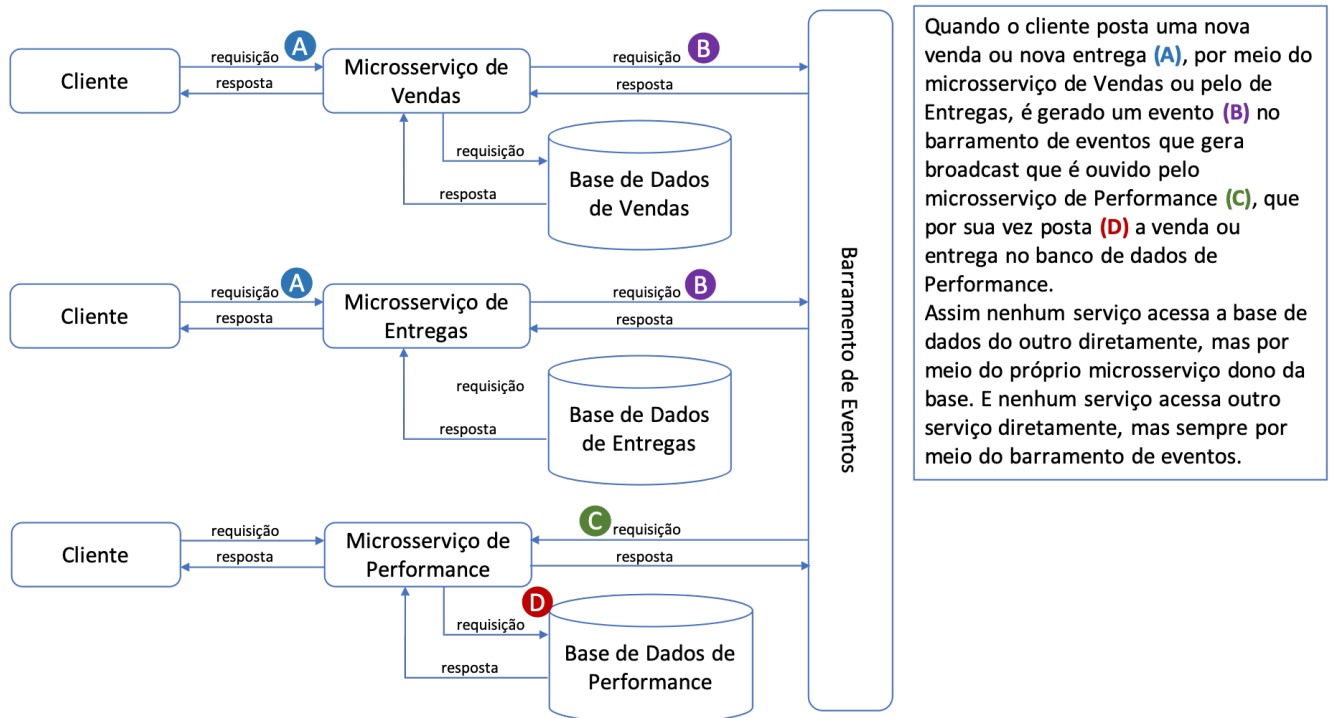
- emitir eventos, o que significa que eles são enviados ao barramento.
- receber eventos do barramento que são gerados por outros microserviços.

Há também a dependência entre serviços. Mas pelo fato de a comunicação ser assíncrona, o microserviço de origem pode se ocupar com outras tarefas enquanto aguarda uma eventual resposta do barramento de eventos.

Uma outra forma de comunicação assíncrona se baseia na construção de uma base de dados em função de outras. Ela pode ser uma espécie de view contendo somente as partes de interesse.

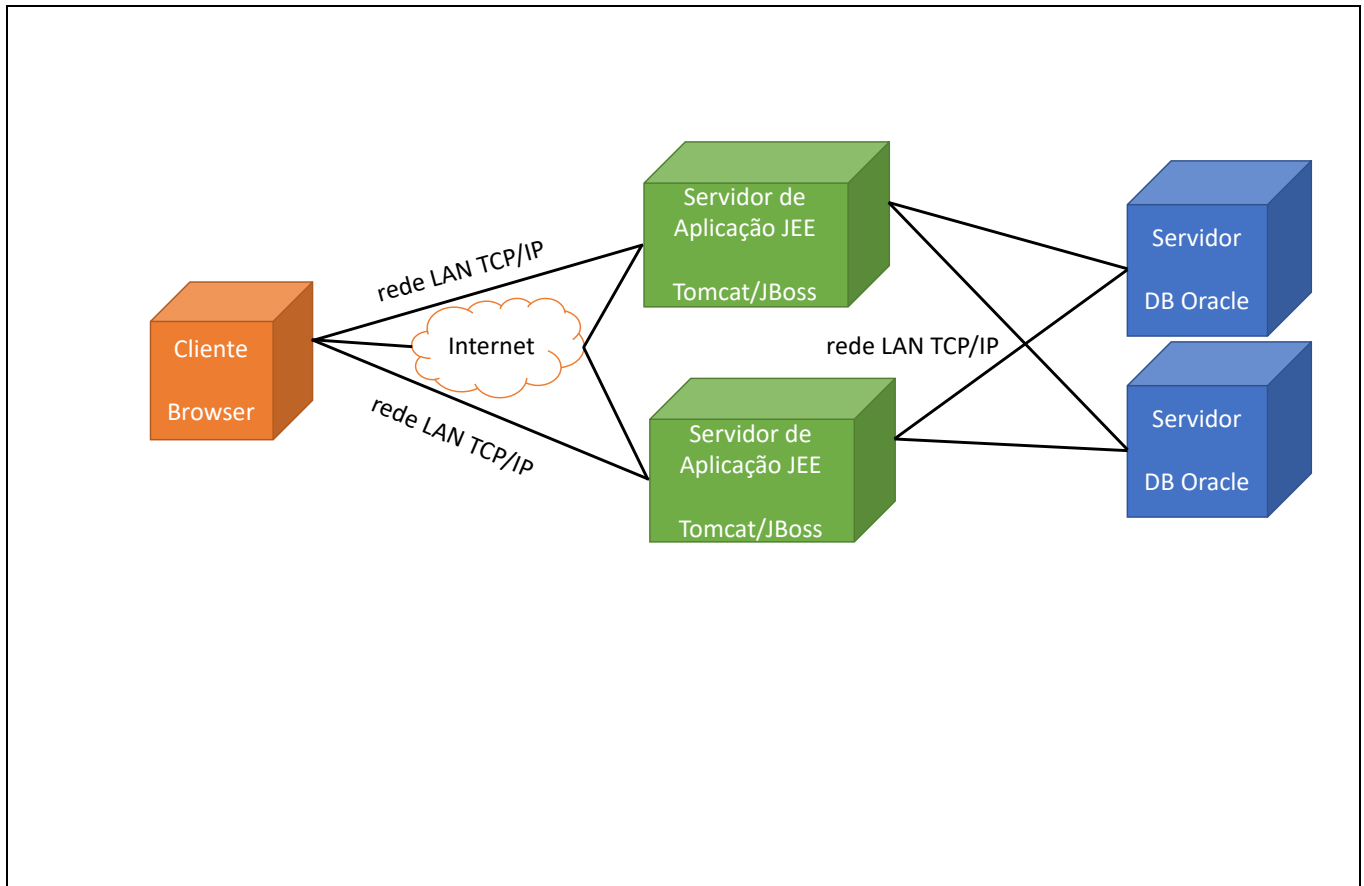
Quando o microserviço de Vendas recebe uma nova requisição, ele armazena os dados de um novo pagamento em sua base, garantindo o seu funcionamento. Além disso, ele emite um evento que é direcionado a um barramento de eventos. O barramento de eventos se encarrega de fazer uma espécie de envio broadcast para que microserviços interessados naquele evento sejam notificados. Neste exemplo, o microserviço de Performance recebe o evento e atualiza a sua base. Note que a requisição é assíncrona e, caso o microserviço de Performance esteja temporariamente indisponível, o microserviço de Vendas não deixa de funcionar. E o oposto também é verdadeiro: caso o microserviço de Vendas esteja indisponível, o microserviço de Performance pode operar utilizando a base de dados que possuir no momento. A mesma ideia se aplica ao microserviço de Entregas.

b) Desenhe esquematicamente como se dá a comunicação assíncrona entre os microserviços Performance, Vendas e Entregas.



5) (2,0 pontos) Um sistema de vendas legado foi desenvolvido na arquitetura cliente servidor, utilizando Delphi e banco de dados Oracle. Agora precisa ser refeito na arquitetura em camadas, para ser acessível pela web. A empresa irá desenvolver o sistema em JavaEE, com servidores Tomcat/JBoss. E irá continuar usando banco Oracle. Ela deseja também ter alta disponibilidade tanto no banco quanto nos servidores de aplicação, além de escalabilidade. Faça:

a) O diagrama esquemático de arquitetura do novo sistema.



b) A descrição da infraestrutura de software necessária para atender a nova arquitetura com a disponibilidade e escalabilidade desejadas.

Do lado cliente apenas um browser. Do lado backend servidores de aplicação JEE, com Tomcat e JBoss e servidores de banco de dados Oracle.

Para ter alta disponibilidade e escalabilidade é necessário ter software de arquitetura em cluster (Windows Server, RedHat, Veritas Cluster, etc.) e software de balanceamento de carga (Big-IP).

RASCUNHO