Insper

Plataformas, Microsserviços, DevOps e APIs

Avaliação Intermediária – AI . 2026.2

25/09/2026

Prezado(a) Aluno(a),

Você terá 120 minutos a partir do início oficial da prova para concluir esta avaliação, administre bem o seu tempo. Leia atentamente as instruções a seguir e as questões da prova antes de começar a resolvê-la.

1. Identifique-se com letra legível na última folha da prova.
2. Esta avaliação é composta de 10 questões de múltipla escolha (0,8 ponto cada) e 1 questão dissertativa (2,0 pontos), totalizando 10,0 pontos.
3. Verifique se a prova está completa e/ou se há problemas de impressão e comunique o aplicador antes de iniciar a prova. Comunicação posterior não será considerada. A prova tem 8 páginas.
4. **A única folha a ser devolvida ao aplicador é a última página (página 8)**, com o formulário de respostas e sua identificação – não erre o lançamento das respostas e não esqueça a identificação.
5. A resolução da prova poderá ser feita a lápis ou a caneta. Avaliações feitas a lápis, no entanto, não serão revisadas pelo professor.
6. Em caso de dúvida sobre alguma questão desta avaliação, redija um texto na folha de respostas ou seu verso explicitando-a para que o professor avalie a pertinência durante a correção.
7. Portar qualquer material estranho à folha de questões – celular (ligado ou desligado), tablet, notebook, anotações e livro – constituirá violações ao Código de Ética e de Conduta e acarretará sanções nele previstas.
8. Consulta a colegas e a qualquer material estranho à folha de questões constituirão violações ao Código de Ética e de Conduta e acarretarão sanções nele previstas. Faça o seu trabalho de maneira ética!
9. Você somente poderá sair da sala depois de entregar a prova. Caso necessite sair durante a realização da avaliação, peça autorização antecipadamente ao aplicador.

Boa Prova!

Para uso exclusivo do Professor

Orientações durante a avaliação:

Permitido utilizar folha para rascunho/anotações?
( ) SIM (X) NÃO

Permitido uso de calculadora?
( ) SIM (X) NÃO

Permitido realizar consulta de material?
( ) SIM (X) NÃO

Após a correção da avaliação:

Avaliação passível de revisão?
( ) SIM ( ) NÃO

Q1. *(Sistemas distribuídos / Teorema CAP)*

O saldo das contas de uma fintech é replicado entre dois data centers. Durante uma falha, o enlace entre eles caiu por 40 minutos, e cada data center continuou atendendo os clientes da sua região. A equipe decidiu que, em uma próxima falha desse tipo, o data center que perder contato com o outro passará a recusar operações de débito até o enlace voltar.

À luz do teorema CAP, o que essa decisão significa?

( ) **A)** O sistema passa a ser CA: como a partição foi resolvida pela equipe, consistência e disponibilidade voltam a ser garantidas ao mesmo tempo em todos os data centers.

( ) **B)** O sistema passa a ser AP: o data center isolado continua disponível e apenas adia a sincronização dos saldos para quando o enlace entre os dois for restabelecido.

( ) **C)** O sistema passa a ser CP: durante a partição ele abre mão da disponibilidade de parte das operações para não permitir saldos divergentes entre os data centers.

( ) **D)** O teorema CAP deixa de se aplicar, porque recusar operações é uma decisão de regra de negócio e não uma propriedade do sistema distribuído que armazena os saldos.

( ) **E)** O sistema passa a garantir as três propriedades, já que a tolerância a partição é obtida justamente ao recusar as operações no data center que ficou isolado.

Q2. *(Escalabilidade / Escalonamento horizontal)*

Um teste de carga mostrou que cada réplica do gateway atende até 400 requisições por segundo antes de a latência degradar. O pico de tráfego previsto para a Black Friday é de 2.600 requisições por segundo, e a equipe exige que o pico continue sendo atendido mesmo que uma réplica qualquer caia.

Qual é o menor número de réplicas que atende a essa exigência?

( ) **A)** 6 réplicas.

( ) **B)** 7 réplicas.

( ) **C)** 8 réplicas.

( ) **D)** 9 réplicas.

( ) **E)** 10 réplicas.

Q3. *(Containerização / Imagem e container)*

Um aluno criou dois containers a partir da mesma imagem `store/account:1.0`. Dentro do primeiro, gravou o arquivo `/tmp/relatorio.csv`; ao listar `/tmp` no segundo, o arquivo não estava lá. Depois ele removeu o primeiro container com `docker rm`, criou outro a partir da mesma imagem, e o arquivo também não apareceu.

Qual explicação está correta?

( ) **A)** A imagem é somente-leitura: cada container tem a própria camada gravável, que some com ele; para o dado sobreviver, é preciso um volume.

( ) **B)** O arquivo foi gravado na imagem, mas só aparece nos novos containers depois de um `docker pull`, que atualiza a versão local da imagem.

( ) **C)** O diretório `/tmp` é apagado sempre que o container reinicia; em qualquer outro diretório o arquivo seria visto pelos dois containers.

( ) **D)** Os dois containers compartilham o disco, mas o Docker isola os arquivos pelo usuário que os criou, e o segundo container roda com outro usuário.

( ) **E)** O `docker rm` apagou também a imagem; o terceiro container foi criado a partir de uma cópia nova, baixada novamente do registry.

Q4. *(Internet / IP privado e NAT)*

Em casa, um aluno sobe o serviço no notebook, que recebeu do roteador Wi-Fi o endereço 192.168.0.23. No próprio notebook, `http://192.168.0.23:8080/health` responde. Um colega, na casa dele, tenta abrir o mesmo endereço e não obtém resposta.

Por que o colega não consegue acessar?

( ) **A)** Porque a porta 8080 é bloqueada por padrão por todas as operadoras, e o serviço teria de escutar na porta 80 para ser acessível pela internet.

( ) **B)** Porque o endereço é privado: só tem sentido dentro da rede da casa do aluno, e a internet vê apenas o IP público do roteador, que faz NAT.

( ) **C)** Porque falta cadastrar um nome de domínio: sem DNS, um endereço IP não pode ser acessado a partir de outra rede, apenas da rede local.

( ) **D)** Porque o colega deveria usar `127.0.0.1:8080`, que é o endereço pelo qual qualquer máquina chega ao serviço publicado no notebook.

( ) **E)** Porque endereços começados em 192 são IPv6, e a rede da casa do colega trabalha apenas com endereços IPv4 fornecidos pela operadora.

Q5. *(Docker Compose / Rede entre serviços)*

No `compose.yaml` abaixo, o `account` não consegue abrir conexão com o banco (*Connection refused*), embora o container `db` esteja no ar e saudável.

```
services:
  db:
    image: postgres:17
  account:
    build: ./account-service
    environment:
      DATABASE_HOST: localhost
      DATABASE_PORT: 5432
```

Qual é a causa e a correção?

( ) **A)** Falta publicar a porta do banco com `ports: 5432:5432`; sem isso, nenhum outro container da mesma rede consegue alcançar o Postgres.

( ) **B)** Falta a chave `depends_on`; com ela, o Compose passa a traduzir `localhost` para o endereço do serviço do qual o `account` depende.

( ) **C)** Os dois serviços estão em redes diferentes; é preciso declarar uma rede e ligar ambos a ela, pois o Compose não cria nenhuma rede sozinho.

( ) **D)** A porta está errada: dentro da rede do Compose, o Postgres escuta na porta 5433, reservada para o acesso entre containers.

( ) **E)** Dentro do container, `localhost` é o próprio `account`; o host deve ser `db`, nome que o DNS do Compose resolve.

Q6. *(Microsserviços / Isolamento)*

Uma loja dividiu o monólito em três microsserviços — `account`, `order` e `product` —, cada um com o próprio banco de dados e implantado no próprio container. Uma nova versão do `product` entrou com um erro que derruba o serviço sempre que alguém faz uma busca no catálogo.

Se a arquitetura segue os princípios de microsserviços, o que se espera?

( ) **A)** O catálogo fica fora do ar, mas `account` e `order` seguem atendendo o que não depende dele; só o `product` é corrigido e reimplantado.

( ) **B)** A aplicação inteira cai, porque os três serviços rodam no mesmo processo e uma exceção não tratada em um deles encerra também o processo dos outros dois.

( ) **C)** Os três serviços precisam ser reimplantados juntos para voltar à versão anterior, porque formam uma única versão da aplicação e não podem ter versões diferentes.

( ) **D)** O banco do `account` e o do `order` também ficam inconsistentes, porque os três bancos são réplicas uns dos outros e propagam o erro entre si.

( ) **E)** O `order` passa a responder também pelas buscas do catálogo até a correção, porque os serviços assumem automaticamente as funções de quem ficou fora do ar.

Q7. *(ORM / Mapeamento objeto-relacional)*

O `account-service` usa JPA com a classe abaixo e salva um cliente novo:

```
@Entity
@Table(name = "accounts")
public class AccountModel {
    @Id
    private String id;
    private String name;
    private String email;
    // construtores, getters e setters omitidos
}

accountRepository.save(new AccountModel("a1", "Ana", "ana@insper.edu.br"));
```

Qual afirmação descreve corretamente o papel do ORM nesse código?

( ) **A)** Ele exige que o desenvolvedor escreva o comando `INSERT` dentro do método `save`, pois o ORM só gera consultas de leitura sobre as tabelas.

( ) **B)** Ele cria uma nova tabela a cada objeto salvo, nomeada pelo `id` do objeto, e por isso cada cliente acaba ocupando uma tabela própria no banco.

( ) **C)** Ele envia o objeto Java serializado ao banco, que o armazena como um bloco binário na tabela, já que o PostgreSQL entende objetos Java de forma nativa.

( ) **D)** A classe equivale à tabela `accounts`, cada atributo a uma coluna e cada objeto a uma linha; o ORM traduz o `save` em um `INSERT`.

( ) **E)** Ele guarda o objeto em memória e só grava no banco quando o serviço é reiniciado, momento em que todos os objetos pendentes são enviados de uma só vez.

Q8. *(API Gateway / Roteamento)*

O gateway (Spring Cloud Gateway) tem as rotas abaixo. No `compose.yaml`, só o `gateway` publica porta no host (`8080:8080`); `auth` e `account` ficam apenas na rede interna.

```
routes:
  - id: auth
    uri: http://auth:8080
    predicates:
      - Path=/auth/**
  - id: accounts
    uri: http://account:8080
    predicates:
      - Path=/accounts/**
```

Um cliente faz `GET http://localhost:8080/accounts/42`. O que acontece?

( ) **A)** O gateway responde com um redirecionamento (302) para `account:8080`, e o cliente refaz a chamada diretamente ao serviço de contas, sem o gateway.

( ) **B)** O gateway casa o caminho com `/accounts/**` e repassa a requisição a `account:8080` pela rede interna; o cliente só conhece o gateway.

( ) **C)** A requisição falha, porque o `account` não publica porta no host e, por isso, não pode ser alcançado nem mesmo pelo gateway da plataforma.

( ) **D)** O gateway envia a requisição às duas rotas ao mesmo tempo e devolve ao cliente a primeira resposta que chegar, descartando a outra.

( ) **E)** O gateway consulta diretamente a tabela de contas no banco e monta a resposta, sem precisar chamar o serviço `account`.

Q9. *(Load balancer / Proxy reverso)*

No hands-on, o Nginx passou a ser o único componente com porta publicada no host, com três réplicas do gateway atrás dele, acessíveis apenas pela rede interna do Compose.

Qual afirmação descreve corretamente o papel do Nginx nessa arquitetura?

( ) **A)** Ele é um proxy direto (*forward proxy*): fica na frente dos clientes e esconde o endereço deles dos servidores que eles acessam na internet.

( ) **B)** Ele substitui o gateway: passa a aplicar as rotas de cada microsserviço, e as três réplicas do gateway ficam redundantes e podem ser removidas.

( ) **C)** Ele guarda as sessões dos usuários e as replica entre as réplicas do gateway, que só assim conseguem responder a qualquer requisição de qualquer cliente.

( ) **D)** Ele divide cada requisição em três partes, processadas em paralelo pelas réplicas, o que reduz a um terço o tempo de resposta de cada chamada.

( ) **E)** Ele é um proxy reverso: recebe as requisições no lugar dos servidores, que o cliente não enxerga, e como balanceador as reparte entre as réplicas.

Q10. *(Docker Compose / Volumes)*

O serviço `db` está declarado assim:

```
services:
  db:
    image: postgres:17
    volumes:
      - db-data:/var/lib/postgresql/data
volumes:
  db-data:
```

A equipe roda `docker compose down` e, em seguida, `docker compose up -d`. Na semana seguinte, roda `docker compose down -v` e, de novo, `up -d`. O que acontece com os dados do banco?

( ) **A)** Somem já no primeiro `down`, porque o container do banco é removido e os dados estavam dentro dele.

( ) **B)** Sobrevivem aos dois `down`, porque um volume nomeado só é removido de forma explícita, com o comando `docker volume rm`.

( ) **C)** Somem no primeiro `down` e reaparecem no `up` seguinte, que refaz o banco a partir da imagem `postgres:17`.

( ) **D)** Sobrevivem ao primeiro `down`, que preserva o volume nomeado, e somem no segundo, porque o `-v` remove os volumes.

( ) **E)** Sobrevivem aos dois `down`, porque o `-v` apenas mostra na tela os volumes que seriam removidos, sem apagá-los.

Q11. *(Dissertativa – Comunicação entre serviços – 2,0 pontos)*

Ao receber `POST /orders`, o `order-service` faz, em sequência, três chamadas HTTP síncronas (via OpenFeign) e só então responde ao cliente. A resposta mostra na tela o pedido confirmado e o seu valor já convertido para a moeda do cliente; o e-mail de confirmação, por sua vez, pode chegar alguns minutos depois sem prejuízo nenhum.

serviço chamado	para quê	tempo típico
`account`	confirmar que o cliente existe e está ativo	40 ms
`exchange`	converter o valor do pedido para a moeda do cliente	120 ms
`notification`	enviar o e-mail de confirmação do pedido	900 ms

Numa sexta-feira, o provedor de e-mail usado pelo `notification` passou a responder em 30 s. O checkout parou de funcionar e, minutos depois, até rotas que nada têm a ver com pedidos ficaram lentas no gateway, embora `account` e `exchange` estivessem saudáveis.

(a) *(0,8 ponto)* Explique por que a lentidão de um único serviço derrubou o checkout e chegou a afetar outras rotas.

(b) *(1,2 ponto)* Proponha um redesenho: para cada uma das três chamadas, diga se deve continuar síncrona ou passar a assíncrona, por meio de um broker de mensagens. Na justificativa, responda: *o cliente precisa do resultado dessa chamada para receber a resposta do pedido?*

NOME: _____

	A	B	C	D	E
Q1					
Q2					
Q3					
Q4					
Q5					

	A	B	C	D	E
Q6					
Q7					
Q8					
Q9					
Q10					

Q11 (a) por que o checkout parou e outras rotas foram afetadas:

Q11 (b) marque o tipo de cada chamada; justifique dizendo se o cliente precisa do resultado dela para receber a resposta:

	síncrona	**assíncrona**	**justificativa**
`account`	☐	☐	_____
`exchange`	☐	☐	_____
`notification`	☐	☐	_____

Devolva APENAS esta página a(o) aplicador(a)

Para uso exclusivo do Professor: **Avaliação passível de revisão?** ( ) SIM ( ) NÃO