



MINISTÉRIO DA EDUCAÇÃO

INSTITUTO FEDERAL DO ESPÍRITO SANTO

REITORIA

Avenida Rio Branco, 50 – Santa Lúcia – 29056-255 – Vitória – ES

27 3357-7500

CONCURSO PÚBLICO EDITAL Nº 03 / 2014

Professor do Magistério do Ensino Básico, Técnico e Tecnológico

ÍNDICE DE INSCRIÇÃO	310
CAMPUS	Serra
ÁREA/SUBÁREA/ESPECIALIDADE	Ciência da Computação/Metodologia e Técnicas de Computação/Sistemas de Informação

PROVA DE CONHECIMENTOS ESPECÍFICOS | DISCURSIVA MATRIZ DE CORREÇÃO

QUESTÃO 01

1. Apresentar e explicar a fórmula da Segunda lei da termodinâmica indicando a irreversibilidade do processo entrópico.
2. Associar o conceito de entropia da termodinâmica com o conceito de entropia de sistemas e exemplificar. Apresentar a fórmula da quantidade de informação e indicar que a mesma é formalmente equivalente à da segunda lei. Apresentar o fato de que o processo entrópico é a causa fundamental da obsolescência dos sistemas de informação que leva a necessidade de voltar ao estágio inicial de concepção do sistema formando de ciclo de vida de sistemas de informação.
3. Apresentar e explicar os princípios do Holismo:
 - a. O todo é maior que a soma das partes evidenciando que as relações entre as partes levam o todo a um funcionamento com resultados maiores que a soma das contribuições individuais de cada parte.
 - b. O comportamento das partes pode ser explicado pelo comportamento do todo evidenciando que as regras gerais de um sistema influenciam o comportamento de seus elementos individuais.
 - c. Interdependência e complementaridade entre as partes de um sistema, indicando que as relações entre os elementos são complementares no sentido de que a são as ações coletivas dos elementos é que possuem uma função e finalidade definida.
 - d. Exemplificar o princípio holístico no contexto de Sistemas de Informação apresentando por exemplo os elementos Hardware e software que são complementares, interdependentes produzem um resultado maior que se vistos isoladamente e só

possuem finalidade quando vistos em conjunto.

4. Explicar o conceitos:
 - a. Equilíbrio Dinâmico como uma tendência de muitos sistemas naturais e artificiais de manterem dentro de um limite os valores de um dado conjunto de variáveis, mesmo se as condições externas forcingem à sua alteração.
 - b. Importância do Feedback para o equilíbrio dinâmico, indicando que o feedback é o principal mecanismo para o sistema perceber que deve reagir para manter o equilíbrio dinâmico.
5. Exemplificar o conceito de homeostase em sistemas de Informação. Um exemplo é o conceito de *Throughput* e a necessidade de mantê-lo, por exemplo em um servidor Web, mesmo diante de picos de acesso. Outros exemplos podem ser utilizados.
6. Apresentar e explicar o modelo de sistema com os componentes de entrada-saída-processamento- feedback, evidenciando o enfoque comportamental e exteriorizante desse modelo. Apresentar um exemplo no contexto de sistemas de informação utilizando este modelo, por exemplo, num sistema Web onde as entradas são requisições e as saídas são as páginas Web geradas pelo Servidor. Outros exemplos poderão ser utilizados.
7. Explicar
 - a. O princípio de organização espontânea dos sistemas, evidenciando que na natureza diversas formas de ordem parecem não possuir um motivo específico, o que evidencia que a organização pode surgir de fatores indiretamente associados a mesma.
 - b. Descrever pré-condições e mecanismos para se atingir a auto-organização: Exemplo: abertura e dinamismo.
 - c. Exemplificar os fatores que propiciam a auto-organização em sistemas de informação: Por exemplo: aplicação do conceito de Sistemas aberto, que permite identificar fatores de desempenho positivos e negativos sob o ponto de vista do ambiente, dinamismo por meio de processos de melhoria contínua.

QUESTÃO 02

1. Explicar o que é Teoria da Decisão e fornecer um exemplo de sua aplicação apresentando esta como um ferramental a ser utilizado para auxiliar na escolha de uma entre mais de uma opção na presença de incerteza.
2. Explicar o que é Teoria dos Jogos e fornecer um exemplo de sua aplicação, apresentando esta como um ferramental que auxilia em um processo de tomada de decisão que envolve competição propiciando a escolha de ações que maximizam a vantagem do competido frente a incerteza sobre a decisão futura do outro competidor.
3. A resposta também deverá relacionar expor as diferenças entre Teoria da Decisão e Teoria dos Jogos.
4. Expor o significado de cada elemento do problema de decisão e apresentar um exemplo e uma matriz de decisão contendo as ações, os estados da natureza, a utilidade de cada para (ação, estado da natureza).
5. Explicar os critérios:
 - a. Laplace onde considera-se que todos os estados da natureza têm uma probabilidade de ocorrência igual e escolhe-se a opção que maximiza o valor médio da relação de utilidade.
 - b. Maxmin onde considera-se o cenário pessimista onde sempre ocorrerá o pior estado da natureza seja qual for ao opção escolhida.
 - c. Savage onde se constrói uma matriz de pesares e se escolhe a opção que produziria o arrependimento mínimo
 - d. Hurwicz onde se utiliza o índice de otimismo associado a decisão a ser tomada.
 - e. Valor Médio Esperado onde se considera as probabilidades de ocorrência de cada estado da natureza e se escolhe a opção onde a média da utilidade é maior.
 - f. Perda da Oportunidade Esperada onde se constrói uma matriz de pesares e considera-se as probabilidades de ocorrência de cada estado da natureza.

6. Utilizar um exemplo para explicar os critérios:
 - a. Laplace, Maxmin, Savage e Hurwicz.
 - b. Valor Médio Esperado e Perda da Oportunidade Esperada. No exemplo deverá ser apresentada uma matriz de decisão.

QUESTÃO 03

a) Serviços

A orientação a serviços é um paradigma de projeto (*design*) que abrange um conjunto específico de princípios de design. A aplicação desses princípios ao design da lógica resulta em uma lógica orientada a serviços. A unidade mais fundamental desta lógica é o serviço. O princípio fundamental é que as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços.

SOA é uma abordagem de arquitetura para a criação de sistemas criados a partir de serviços autônomos. Com a SOA, a integração é uma previsão, e não algo em que não se pensa: a solução final provavelmente será composta de serviços desenvolvidos em diferentes linguagens de programação, hospedadas em plataformas diferentes, com uma variedade de modelos de segurança e de processos comerciais.

SOA possibilita a comunicação entre sistemas desenvolvidos em linguagens de programação distintas e sendo executados em sistemas operacionais diferentes. Esse cenário é muito comum em empresas de médio e grande porte que possuem uma infraestrutura de TI complexa.

SOA é focada em processos de negócios. Esses processos podem ser executados em diferentes etapas e em diferentes sistemas. O principal objetivo de um serviço é representar, de forma natural, uma etapa de uma funcionalidade de negócio. Basicamente tudo que tenha uma interface definida e faça parte de um processo de negócio pode ser um serviço. Um serviço pode ser definido como uma interface que recebe mensagens, executa ações e/ou retorna alguma informação.

Um exemplo básico de serviço é o de Diretório, que armazena todos os componentes e informações chaves sobre o sistema. É usado para automatizar a ação de busca de elementos da integração, também pode ser usado para gerenciar mensagens ou interfaces assim como metadados. Outro exemplo importante é o serviço de notificação que assim que detecta um evento determinado, informa aos componentes interessados que ele ocorreu.

Outras possíveis respostas para conceituar serviço:

a.1) Um serviço, do ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizado para outro sistema. Ele é uma unidade autocontida de software que realiza uma determinada tarefa. Ele tem 3 componentes: uma interface, um contrato e uma implementação. A interface define como um provedor de serviço irá realizar/executar as requisições provenientes de um consumidor de serviço. O contrato define como o provedor de serviço e o consumidor de serviço deveriam interagir. A implementação é o próprio código do serviço.

a.2) Outra opção de resposta, de uma maneira mais formal, poderia definir um serviço como "uma função independente, sem estado (*stateless*) que aceita uma ou mais requisições e devolve uma ou mais respostas através de uma interface padronizada e bem definida". Serviços podem também realizar partes discretas de um processo tal como editar ou processar uma transação. Serviços não devem depender do estado de outras funções ou processos. A tecnologia utilizada para prover o serviço, tal como uma linguagem de programação, não pode fazer parte da definição do serviço. Uma vez que a interface de serviço é separada de sua implementação, um provedor de serviço pode

executar (responder a) uma requisição sem o consumidor do serviço saber como ele faz isso.

a.3) Um serviço, do ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizado para outro sistema. Um serviço deve funcionar de forma independente do estado de outros serviços, exceto nos casos de serviços compostos (*composite services*), e deve possuir uma interface bem definida. Normalmente, a comunicação entre o sistema cliente e aquele que disponibiliza o serviço é realizada através de Web Services.

b) Interfaces autodescritas

Para ser efetivamente útil um serviço precisa separar entre interface e implementação. A interface é exposta, onde os usuários, aplicativos ou outros serviços precisam conhecê-la para consumi-la; a implementação é privada, ou “escondida”. Desta forma, diferentes implementações podem oferecer a mesma interface, e modificações em uma implementação não causará interferência nos usuários, aplicativos e serviços que se utilizam daquela interface. Os serviços estão disponíveis para consumo através de interfaces e através de certo meio de comunicação – web-service, arquivos, etc. Além do meio de comunicação, que pode variar, a interface também define a estrutura da mensagem que vai ser recebida e enviada, o que pode ser algo específico do negócio.

Os serviços e suas interfaces associadas devem permanecer estáveis, permitindo que sejam reconfigurados ou reagregados para atender às sempre novas necessidades das empresas. Os serviços permanecem estáveis, contando com interfaces baseadas em padrões e em mensagens bem definidas. Um exemplo prático disso é o uso de esquemas SOAP e XML para definição de mensagens.

c) Troca de mensagens

Em SOA, a comunicação com Serviços é feita através de mensagens com estruturas bem definidas. Estas mensagens possuem o formato definido pela interface do Serviço. Um exemplo simples é de um programa que importa um arquivo com dados em determinado formato. Neste contexto, o programa é o Serviço, e o arquivo é a Mensagem.

Quando as mensagens são enviadas, os serviços perdem o controle do que acontece depois. Essas mensagens podem ser equipadas com inteligência suficiente para autogovernar as partes lógicas do processamento. Esta arquitetura é similar ao passado da arquitetura distribuída que suporta mensagens e separação de interface de processamento lógico. O que distingue é como esses três componentes fundamentais (serviço, descrição e mensagem) são projetados. É neste ponto que entra a orientação de serviços. Alguns exemplos de mensagens que podem ser trocadas através de Serviços num processo de negócios:

- Endereço – Leitura de endereços cadastrados em certo software;
- Documento de Despacho – Enviado ao cliente quando uma venda foi despachada;
- Contatos – Leitura de contatos cadastrados em certo software;
- Cientes – Leitura de clientes cadastrados em certo software;
- Descontos – Leitura de uma lista de descontos disponíveis para um usuário específico;
- Estoque – Leitura de produtos e sua quantidade em estoque atual;
- Formas de Pagamento – Leitura das formas de pagamento utilizadas pelo negócio;
- Lista de Preço – Leitura de uma lista de preços específica para o cliente;
- Grupos de Produtos – Leitura de grupos de produtos utilizados pelo negócio;
- Nota Fiscal – O Serviço cria uma nota fiscal no aplicativo;
- Pedido de Orçamento – Cria um pedido de orçamento;
- Confirmação de venda – Leitura de confirmações de vendas com identificador;

d) Comunicação síncrona e assíncrona

Existem dois tipos básicos para um modelo de comunicação: síncrono e assíncrono.

Uma comunicação síncrona estabelece basicamente que quando o remetente envia uma requisição, ele

espera que o receptor envie uma resposta. Ou seja, é necessário que o remetente espere por uma resposta da requisição para continuar com o seu processamento. Se existir uma falha por alguma razão, o remetente é responsável pelo reenvio da mensagem.

* A comunicação síncrona requer que exista uma coordenação entre a troca de mensagens e o processamento nos remetentes e receptores. Esse sincronismo entre mensagem e processamento gera um forte acoplamento, no caso da comunicação entre dois sistemas isso que um sistema está dependente do outro, e terá que esperar uma resposta para suas mensagens antes de continuar com o processamento. Esse tipo de comunicação é usado principalmente em sistemas onde o remetente necessita de uma informação ou notificação que processada por outros sistemas e com resposta imediata. A comunicação síncrona requer uma infraestrutura confiável, pois um sistema pode ficar esperando por uma resposta que foi perdida, assim ele pode ficar enviando novas requisições e esperando a resposta, isso pode gerar uma queda de desempenho. Um exemplo deste tipo de comunicação é uma consulta online de um produto na qual o usuário espera até que o resultado seja apresentado na tela.

O modelo síncrono de comunicação entre as aplicações representa uma sequência bem definida de operações, com as saídas e entradas de cada etapa sucessiva orquestradas de uma forma ordenada. Existem algumas regras para a coordenação das mensagens, e elas irão depender do tipo de comunicação síncrona usada. Porém existem três tipos que são os mais populares: (i) Pedido-Resposta; (ii) Sentido único; (iii) Verificação.

* Uma comunicação assíncrona estabelece basicamente que o remetente envia uma requisição e ele não espera que o receptor envie uma resposta. O receptor de uma mensagem não é obrigado a manipular a mensagem ou responder ao remetente. O remetente continua a executar seus processos sem considerar como o receptor irá manipular seu pedido.

Se ocorrer uma falha por alguma razão, o middleware (camada de software responsável pela integração/comunicação entre receptores e emissores) é responsável pelo reenvio da mensagem. Se necessário, o sistema de recepção pode enviar uma resposta de retorno para o remetente através de uma chamada/mensagem assíncrona.

Existem três tipos populares de comunicação assíncrona: (i) passagem de mensagem; (ii) Publicar/subscrever (*publish/subscriber*); (iii) Transmissão.

A comunicação assíncrona é usada principalmente em contextos de transferência de informações. Aplicações onde uma alteração tem que ser informada a outras aplicações que possuem cópia dos dados alterados e esses dados precisaram ser atualizados. Outro exemplo é quando um evento dispara notificações para outras aplicações.

e) Baixo acoplamento

SOA é uma arquitetura que está associada a uma maneira de projetar software com baixo acoplamento (sem dependência complexa), considerando as necessidades/regras do negócio.

O acoplamento refere-se a uma conexão ou relacionamento entre dois elementos. Uma medida de acoplamento se compara com um nível de dependência. Esse princípio estabelece a criação de um tipo específico de relacionamento dentro e fora dos limites do serviço, com ênfase em reduzir as dependências entre o contrato de serviço, sua implementação e os consumidores de serviço.

O princípio do baixo acoplamento de serviço permite que o design e a lógica de um serviço possam evoluir independentemente de sua implementação, ao mesmo tempo em que garante a interoperabilidade básica com consumidores que se utilizam das capacidades do serviço. Existem vários tipos de acoplamento envolvidos no design de um serviço,

cada um deles podendo exercer influência no conteúdo e na granularidade de seu contrato. O objetivo do baixo acoplamento é minimizar a dependência entre sistemas, assim modificações e falhas em um sistema irão gerar poucas consequências em outros sistemas.

f) Ligação dinâmica (*Dynamic Binding*)

A relação entre os serviços do provedor e do consumidor deve ser idealmente dinâmica; ela é estabelecida em tempo de execução através de um mecanismo de binding. SOA suporta o conceito de descoberta dinâmica de serviços. Numa situação real, o consumidor de serviços busca no registro de serviços por um determinado serviço e o serviço de registros retorna uma lista de provedores de serviços. O consumidor seleciona o provedor de serviço "com melhor custo-benefício" a partir da lista e é feito um *binding* (interligação dinâmica) com o provedor usando um ponteiro fornecido pela entrada do serviço de registros.

A única dependência entre provedor e consumidor é o contrato, o qual é fornecido por um serviço terceiro, que é o serviço de registros. A dependência é *run time* (dinâmica). Todas as informações que o consumidor precisa sobre o serviço é obtida e usada em run time. As interfaces de serviços são descobertas dinamicamente e as mensagens são construídas dinamicamente. O serviço consumidor não sabe o formato da mensagem de requisição ou a mensagem de resposta ou mesmo a localização do serviço até que o serviço seja efetivamente necessário.

No caso de Web Services, os serviços são tratados dinamicamente por meio de URIs, permitindo que seus locais subjacentes e topologias de implantação mudem ou evoluam com o tempo, com pouco impacto em relação ao próprio serviço (isso também é válido em relação aos canais de comunicação de um serviço).

g) Registro de serviços

A Arquitetura Orientada a Serviços pode ser bem representada a partir do seguinte processo, chamado de "*find-bind-execute paradigm*" ou "paradigma de procura-consolida-executa". Este processo assume que os provedores de serviços registrem informações em um registro central, com suas características, indicadores, e aspectos relevantes às tomadas de decisões. O registro é utilizado pelo cliente para determinar as características dos serviços necessários, e se o mesmo estiver disponível no registro central, como por exemplo por um catálogo de serviços, o cliente poderá utilizá-lo, sendo este oficializado através de um contrato que enderece este serviço.

Os serviços são projetados para serem descritos independente da sua parte interna, para que possam ser encontrados e avaliados através de mecanismos de descobertas disponíveis. Um registro SOA é um recurso que especifica os direitos de acesso a dados e permite a provedores de serviço descobrirem e se comunicarem com consumidores eficientemente, criando um elo entre provedores de serviço e consumidores. O registro é um catálogo de informações que é constantemente atualizado com informações sobre diferentes serviços num projeto SOA. Esses serviços não são necessariamente Web e nem mesmo relacionados a sistemas computacionais.

Um exemplo de registro de serviços é o UDDI. O serviço de registro UDDI é um Web Service que gerencia informação sobre provedores, implementações e metadados de serviços. Provedores de serviços podem utilizar UDDI para publicar os serviços que eles oferecem. Usuários de serviços podem usar UDDI para descobrir serviços que lhes interessem e obter os metadados necessários para utilizar esses serviços podem ter três partes:

"páginas brancas" descrevem a companhia: nome, endereço, contatos, etc.

"páginas amarelas" incluem as categorias, baseada em taxonomias padrões.

"páginas verdes" descrevem a interface para o serviço, em nível de detalhe suficiente para se escrever uma aplicação que use o Web service.

h) Composição de serviços em processos de negócios

SOA é focada em processos de negócios. Esses processos podem ser executados em diferentes etapas e em diferentes sistemas. O principal objetivo de um serviço é representar, de forma natural, uma etapa de uma funcionalidade de negócio.

Basicamente tudo que tenha uma interface definida e faça parte de um processo de negócio pode ser

um serviço. Neste contexto, um serviço pode ser visto através de uma interface que recebe mensagens, executa ações e/ou retorna alguma informação. Uma composição de serviços é composta de serviços montados a fim de fornecer as funcionalidades para automatizar uma tarefa ou um processo específico de negócios.

As implementações SOA dependem de uma rede de serviços. Cada serviço irá implementar uma ação, por exemplo, consultar a lista de professores de um departamento ou a lista de alunos matriculados em um curso. Se um serviço apresenta uma interface simples que abstrai a complexidade, os usuários podem acessar os serviços, sem conhecimento do serviço desenvolvido. Essa característica da arquitetura orientada a serviço é conhecida como interoperabilidade.

Resposta esperada com relação ao EXEMPLO de uso de SOA:

Espera-se que a resposta também faça uma ligação de SOA com a implementação de WEB SERVICES:

SOA não é uma arquitetura concreta, assim existem diversas maneira de implementá-la. Devido a sua fácil implementação e utilização, e o fato de serem independentes de linguagem de programação, os Web Services (WS) tornaram-se a principal tecnologia para implementar da SOA.

Web Services pode ser visto como um sistema de software que segue um conjunto de padrões abertos de interoperabilidade. Estes padrões permitem a interoperação na Web independentemente da plataforma de hardware, sistema operacional, infraestrutura de rede ou linguagem de programação. A grande utilização dos WS é baseada no fato de que ele usa protocolos abertos de comunicação na Internet e XML para transacionar o seu negócio. Um Web Service é, portanto, um sistema de software que pode agir a pedido de qualquer computador conectado à rede no mundo, que se comunica usando padrões XML.

WS são uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. O único requisito é que os dois sistemas conheçam o formato em que as mensagens serão trocadas e que tenham suporte para manipulação de mensagens no formato XML.

De forma básica, um WS é um sistema de software identificado por uma URI, cujas interfaces públicas e mapeamentos são definidos e descritos usando-se a linguagem XML. Além do XML, os WS são baseados nos seguintes padrões:

- Protocolo de comunicação: SOAP (Simple Object Access Protocol)
- Linguagem de descrição: WSDL (Web Services Description Language)
- Registro de serviços: UDDI (Universal Description, Discovery and Integration)
- Linguagem de composição: WS-BPEL

A interação com o serviço Web deve se basear somente nas diretrizes, no esquema e nos comportamentos baseados no contrato de um serviço. O contrato de um serviço geralmente é definido usando-se o WSDL, e contratos para agregações de serviços podem ser definidos usando-se o BPEL (que, por sua vez, usa o WSDL para cada serviço agregado).

QUESTÃO 04

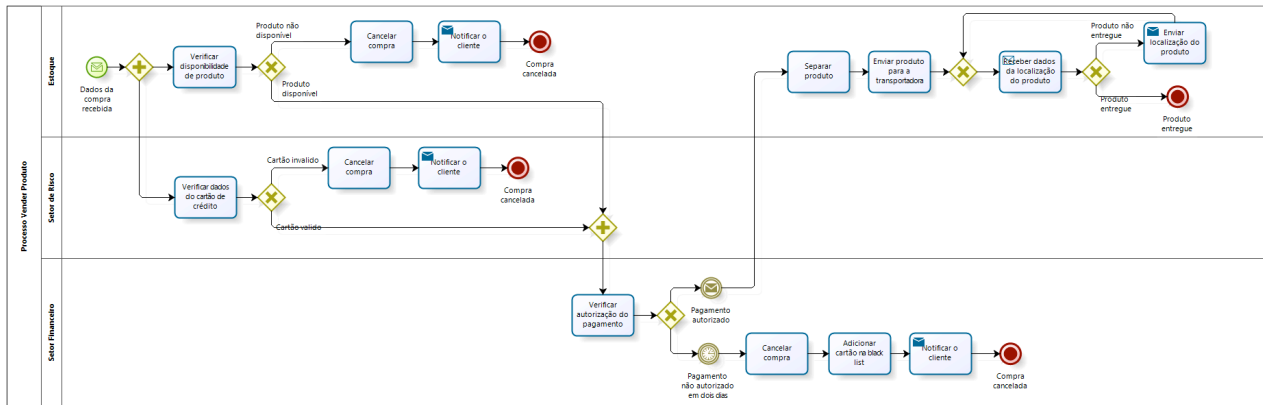
O processo deve conter os seguintes fluxo:

- A. Processo contendo o caminho de sucesso:
 1. O Estoque e o Setor de Risco recebem a mensagem de uma nova venda;
 2. O Estoque verifica se o produto existe;
 3. O Setor de Risco verifica se o cartão é válido;
 4. As atividades do item b e item c devem ser executadas em paralelo.
 5. O Estoque identifica que o produto existe;
 6. O Setor de Risco valida o cartão de crédito;

7. O Financeiro solicita processamento da compra à Administradora de cartão de crédito, através de uma mensagem;
 8. A Administradora de cartão de crédito processa o pagamento;
 9. O Estoque separa o produto;
 10. O Estoque envia uma mensagem à Transportadora solicitando o transporte do produto;
 11. A Transportadora envia mensagens para o Estoque com a localização do produto e o Estoque passa essa informação para o cliente;
 12. A Transportadora informa que ao Estoque que o produto foi entregue;
 13. O Estoque informa o cliente que o produto foi entregue;
 14. O Estoque finaliza o processo com o produto entregue;
- B. Atividades em paralelo:
1. O Estoque e o Setor de Risco recebem a mensagem de uma nova venda;
 2. O Estoque verifica se o produto existe;
 3. O Setor de Risco verifica se o cartão é válido;
 4. As atividades do item b e item c devem ser executadas em paralelo.
- C. Finalizar processo devido a não existência do produto no Estoque:
1. O Estoque identifica que o produto não existe;
 2. O Estoque envia uma mensagem ao cliente informando o ocorrido;
 3. O Estoque finaliza a instancia do processo.
- D. Finalizar o processo devido a não validade do cartão de crédito pelo Setor de Risco:
1. O Setor de Risco identifica que o cartão não é válido;
 2. O Setor de Risco adiciona o cartão inválido no lista de cartões bloqueados;
 3. O Setor de Risco cancela a compra.
 4. O Setor de Risco informa o cliente sobre o cancelamento da compra;
 5. O Setor de Risco finaliza a instancia do processo.
- E. Finalizar o processo devido a não validade do cartão de crédito pelo Setor de Financeiro:
1. O Setor Financeiro envia uma mensagem à Administradora de Cartão de Crédito solicitando o processamento do pagamento;
 2. A Administradora de Cartão de Crédito envia uma mensagem com o resultado do processamento;
 3. O Setor Financeiro recebe a mensagem e conclui que o cartão é inválido;
 4. O Setor Financeiro cancela a compra;
 5. O Setor Financeiro adiciona o cartão inválido no lista de cartões bloqueados;
 6. O Setor de Financeiro informa o cliente sobre o cancelamento da compra;
 7. O Setor de Financeiro finaliza a instancia do processo.
- F. Finalizar o processo devido a demora no processamento do pagamento:
1. O Setor Financeiro aguarda dois dias pela mensagem de processamento de pagamento da Administradora de Cartão de Crédito;
 2. Após o prazo de dois dias o Setor Financeiro finaliza a instancia do processo, caso não tenha recebido a mensagem.
- G. Finalizar o processo devido a demora no processamento do pagamento:
1. O Setor Financeiro aguarda dois dias pela mensagem de processamento de pagamento da Administradora de Cartão de Crédito;
 2. Após o prazo de dois dias o Setor Financeiro notifica o cliente sobre o cancelamento da compra.
 3. O Setor Financeiro finaliza a instancia do processo, caso não tenha recebido a mensagem.
- H. Envio de mensagem de localização do produto:
1. A Transportadora envia mensagens sobre a localização do produto;
 2. O Estoque recebe as mensagem da Transportadora e verifica que não é uma mensagem de entrega;
 3. O Estoque envia uma mensagem para o cliente informando a localização do produto.

- I. Envio de mensagem de entrega do produto:
 1. A Transportadora envia uma mensagem de entrega do produto;
 2. O Estoque recebe as mensagem da Transportadora e o Estoque verifica que é uma mensagem de entrega;
 3. O Estoque envia uma mensagem para o cliente informando que o produto foi entregue;
 4. O Estoque finaliza o processo com sucesso.

Segue o modelo sugerido como resposta da questão.



QUESTÃO 05

A resposta do candidato deve conter as seguintes informações:

- a. Um sistema de informação sensível ao processo é um sistema computacional que gerencia e executa processos operacionais envolvendo: pessoas, aplicativos computacionais e fontes de informação. Esses sistemas são construído baseados nos modelos de processos da organização. Dessa forma, o projetista do sistema pode focar na execução do fluxo como um todo e não somente nos componentes necessários para a execução de uma atividade.
- b. Um usuário de um sistema de informação sensível ao processo executa as atividades em uma ordem pré-estabelecida pelo processo. Em outras palavras, o sistema é responsável por delimitar qual a tarefa do usuário e quanto esse deve executar a tarefa.
- c. Em um sistema orientado a dados a modelagem de dados é ponto de partida para o desenvolvimento de sistemas. Além disso, a lógica do processo de negócio está espalhada dentro do banco de dados ou em código do sistema. Por outro lado, em um sistema orientado a processo, o ponto de partida no desenvolvimento são os modelos de processos da organização.
- d. Antes da década de 90, os processos de negócio eram estruturados para se adequarem as limitações computacionais e, assim, tornando o processo de negócio ineficiente.
- e. A infraestrutura dos sistemas de informação sensível ao processo permitem um isolamento entre o processo e os componentes que executam funcionalidades do sistema. Por exemplo, é possível que a sequência de atividades seja controlada por um sistema de workflow e executar um serviço web presente em arquitetura orientada a serviço.