

**ATIVIDADE 2 - ENG SOFT - MODELAGEM DE SOFTWARE - 51/2021****Período:**22/03/2021 08:00 a 07/04/2021 23:59 (Horário de Brasília)**Status:**ABERTO**Nota máxima:**1,00**Gabarito:**Gabarito será liberado no dia 08/04/2021 00:00 (Horário de Brasília)**Nota obtida:****1ª QUESTÃO**

Nos diagramas de atividade da UML, \_\_\_\_\_ indicam ações que são executadas no sistema. \_\_\_\_\_ são representadas por setas contínuas e são usadas para indicar o fluxo de trabalho entre elementos do diagrama. Uma ramificação lógica é indicada por \_\_\_\_\_ e representa desvios do fluxo de controle. Um retângulo com bordas arredondadas identifica \_\_\_\_\_ e representa marcos de processamento.

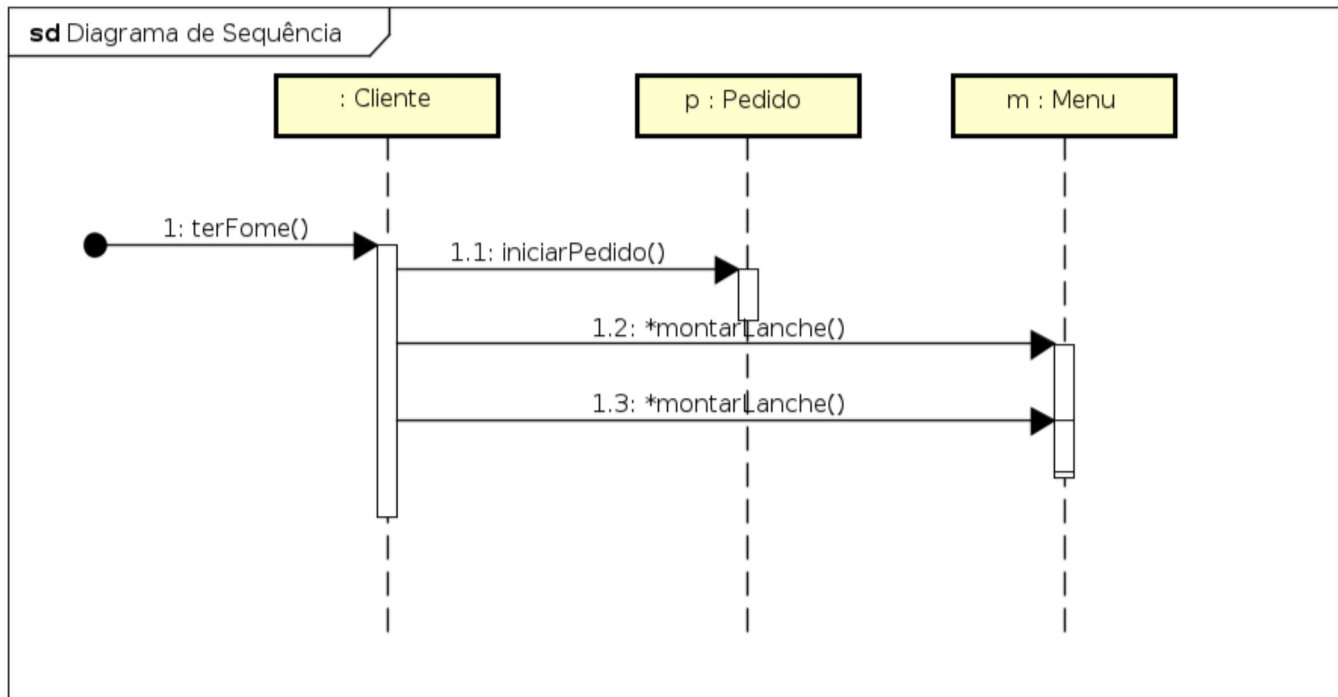
Assinale a alternativa que preenche, correta e respectivamente, as lacunas das afirmações acima.

**ALTERNATIVAS**

- ☐ eventos - Dependências - mensagens - ações
- ☐ mensagens - Transições - ações - componentes
- ☐ atividades - Dependências - mensagens - eventos
- ☐ atividades - Transições - pontos de decisão - ações
- ☐ eventos - Dependências - pontos de decisão - componentes

**2ª QUESTÃO**

Considere o seguinte diagrama de sequência UML:



Analise as afirmações a seguir:

- I - **p** é um objeto ativo da classe Pedido.
- II - A mensagem 1.1 é síncrona.
- III - A mensagem 1.2 representa uma iteração.
- IV - A mensagem 1.3 é assíncrona.

É correto o que se afirma em:

#### ALTERNATIVAS

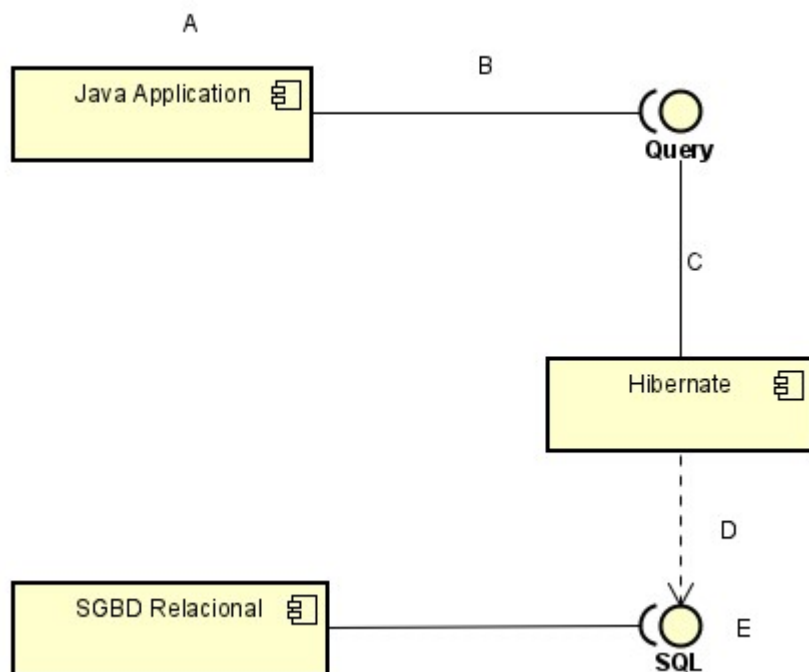
- ☐ I e II, apenas
- ☐ I e III, apenas.
- ☐ II e IV, apenas.
- ☐ I, II e III, apenas.
- ☐ I, III e IV, apenas.

#### 3ª QUESTÃO

O diagrama de componentes permite decompor o sistema em subsistemas que detalham o funcionamento interno. Os componentes representam a implementação em determinada linguagem de programação, os conceitos e as funcionalidades definidas na arquitetura lógica, que congregam as classes, os objetos e seus relacionamentos.

PERSEGUINE, V. R. **Modelagem de Software**. Maringá: Unicesumar, 2019.

Analise o diagrama de componentes a seguir:



Associe as letras indicadas aos elementos do diagrama de componentes.

(A)	( )	Interface
(B)	( )	Componentes
(C)	( )	Dependências
(D)	( )	Interface provida
(E)	( )	Interface requerida

Assinale a alternativa que completa a associação corretamente de cima para baixo.

## ALTERNATIVAS

- ☐ A, B, C, D, E.
- ☐ B, A, C, E, D.
- ☐ C, E, D, B, A.
- ☐ D, E, A, B, C.
- ☐ E, A, D, C, B.

#### 4ª QUESTÃO

Os modelos de contexto representam as perspectivas externas de onde será modelado o ambiente do sistema, os seus limites e suas relações técnicas e também as relações não técnicas.

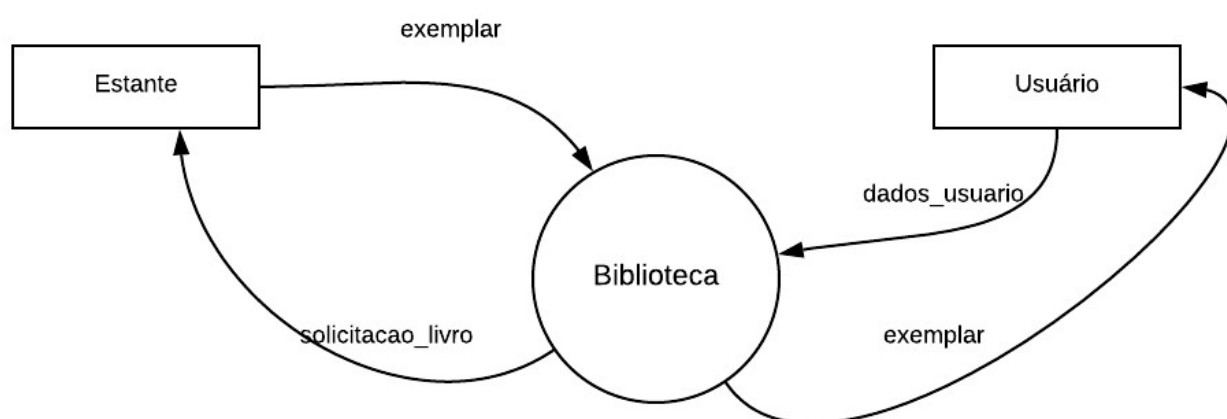
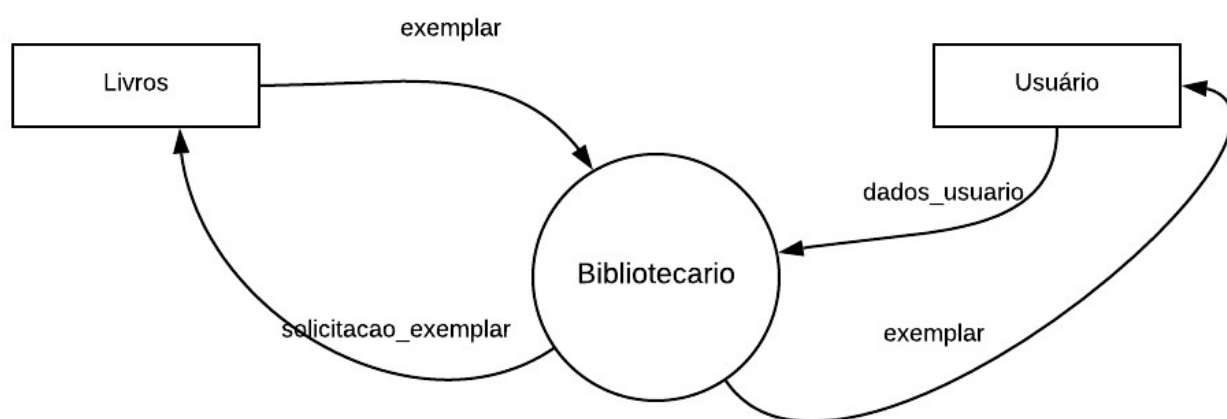
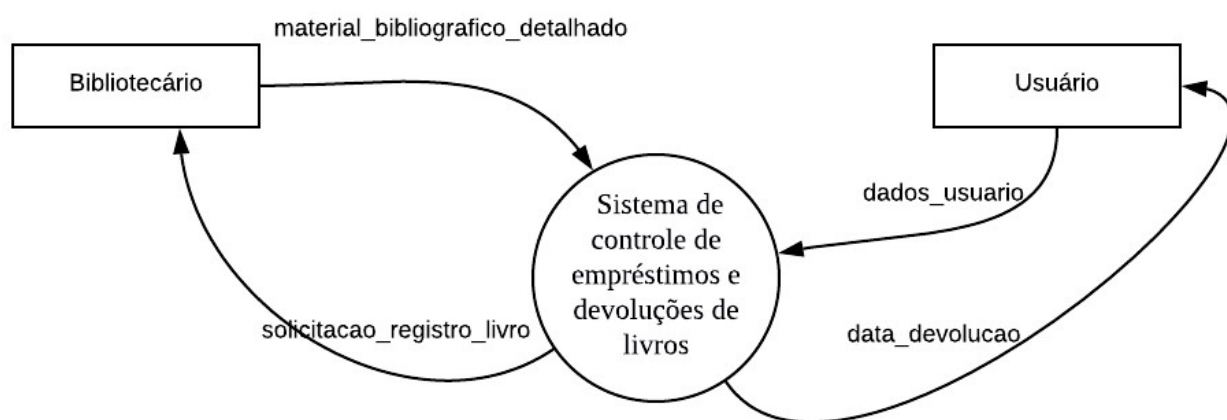
PERSEGUINE, V. R. **Modelagem de Software**. Reimpressão, Maringá-Pr.: UniCesumar, 2018.

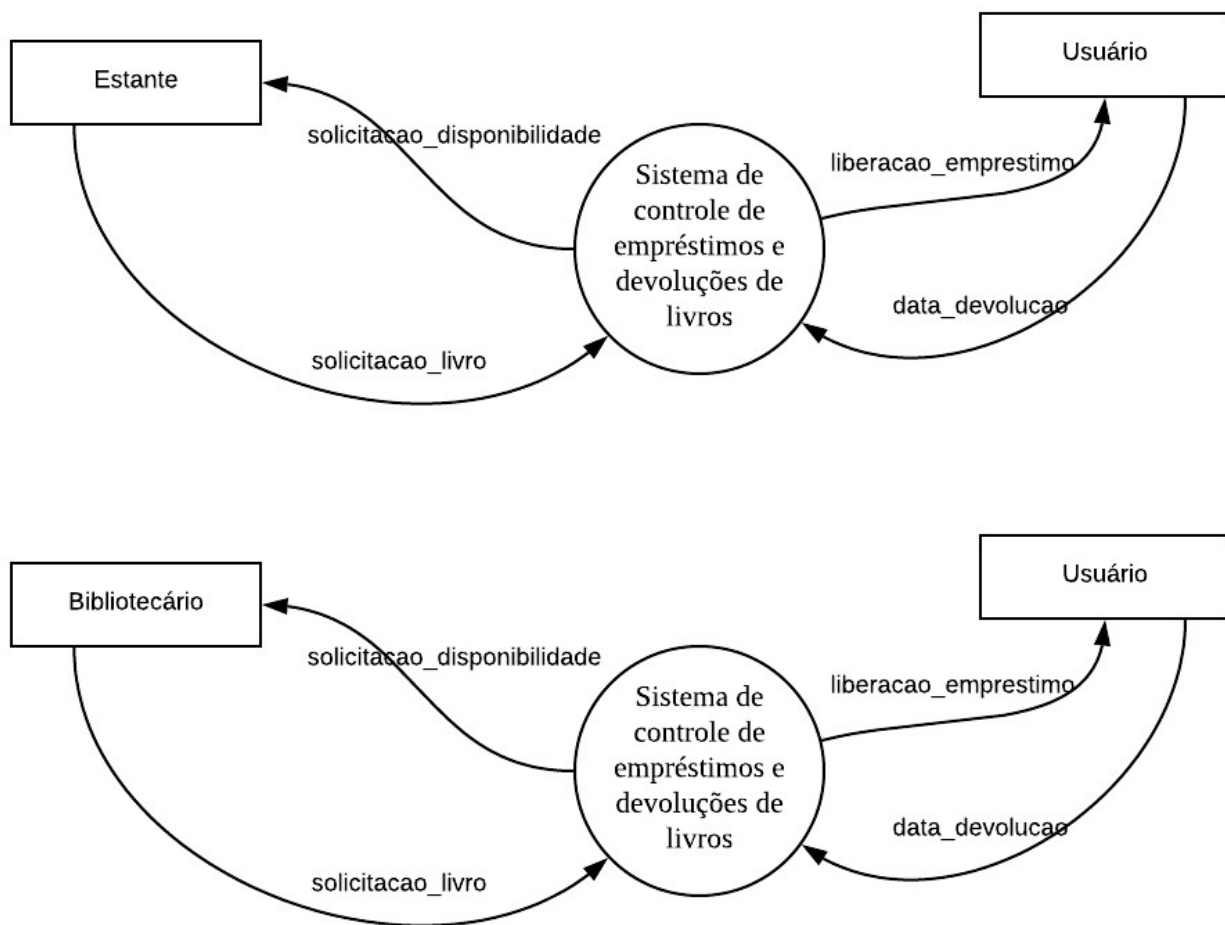
Considere a lista de eventos do sistema de controle de empréstimos e devoluções de livros de uma biblioteca.

Estímulo	Resposta	Evento
Usuário informa dados	Data de devolução	Liberar empréstimos
Sistema solicita registro dos livros	Bibliotecário informa material bibliográfico detalhado	Solicitar disponibilidade de livros

Assinale a alternativa que corresponde ao diagrama de contexto que ilustra a lista de eventos do sistema descrito.

#### ALTERNATIVAS





### 5ª QUESTÃO

O caso de uso é uma técnica de descoberta de requisitos. Em sua forma mais simples, identifica os atores envolvidos em uma interação e nomeia essa interação. O conjunto de casos de uso é registrado em um Diagrama de Casos de Uso que representa todas as possíveis interações que serão consideradas pelo documento de requisitos do sistema. Esse diagrama tem como objetivo descrever um modelo funcional de alto nível do software em análise.

PERSEGUINE, V. R. **Modelagem de Software**. Maringá: Unicesumar, 2019.

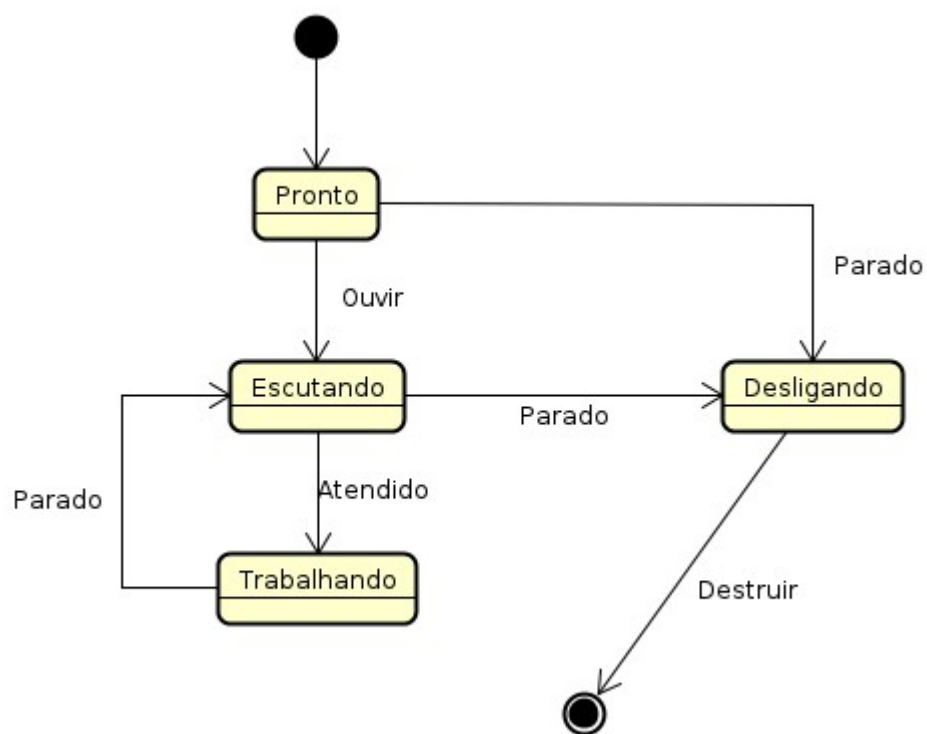
Nesse contexto, assinale a alternativa correta, quando se trata do uso de include em casos de uso.

### ALTERNATIVAS

- ☐ Expressar rotinas de exceção.
- ☐ Requerer a execução correta do caso de uso base.
- ☐ Separar um comportamento obrigatório de um opcional.
- ☐ Separar trechos que dependam da interação com um determinado ator.
- ☐ Compartilhar casos de uso, entre dois atores, sendo que um deles executa diferentes casos de uso.

### 6ª QUESTÃO

A UML é uma linguagem que disponibiliza um conjunto de símbolos com o intuito de comunicar ideias. O objetivo da UML é fornecer ferramentas para a análise, o projeto e a implementação de sistemas. Considere o diagrama a seguir:



A figura apresentada trata de diagrama da UML de \_\_\_\_\_.

Assinale a alternativa que completa corretamente a afirmação anterior.

#### ALTERNATIVAS

- ☐ Estado.
- ☐ Atividade.
- ☐ Sequência.
- ☐ Distribuição.
- ☐ Comunicação.

#### 7ª QUESTÃO

O processo de desenvolvimento de software é organizado em duas grandes categorias: Diagramas Estruturais e Diagramas Comportamentais. A especificação UML não restringe a mistura de diferentes tipos de diagramas. Portanto, não existem fronteiras entre a utilização de um ou outro modelo para atingir o seu objetivo principal, o que possibilita a utilização de acordo com o necessário para o projeto e as necessidades em questão.

PERSEGUINE, V. R. **Modelagem de Software**. Maringá: Unicesumar, 2019.

Analise a lista de diagramas a seguir:

- I - Classes.
- II - Estados.
- III - Caso de uso.
- IV - Implantação.
- V - Comunicação.
- VI - Componentes.

Agora, com base na lista apresentada, assinale a alternativa que corresponde apenas aos diagramas comportamentais:

#### ALTERNATIVAS

- ☐ I, II e III apenas.
- ☐ I, III e V apenas.
- ☐ II, III e V apenas.
- ☐ II, III e VI apenas.
- ☐ II, IV e VI apenas.

#### 8ª QUESTÃO

O paradigma de programação orientada aos objetos, fundamenta-se na utilização de objetos, que colaboram entre si, para a construção do software. A orientação aos objetos está sustentada nos seguintes pilares: abstração, encapsulamento, herança e polimorfismo.

PERSEGUINE, V. R. **Modelagem de Software**. Maringá: Unicesumar, 2019.

Com base no contexto do paradigma de programação orientada aos objetos, assinale a alternativa que implica na definição do conceito de polimorfismo.

#### ALTERNATIVAS

- ☐ Uma técnica que garante o reuso de uma classe em situações distintas.
- ☐ A possibilidade de interagir com objetos de classes diferentes ao mesmo tempo.
- ☐ Um mecanismo que permite compartilhar um comportamento previamente definido.
- ☐ Garantir, de forma centralizada, a integridade interna de várias instâncias de uma classe.
- ☐ Habilidade de uma única operação ser definida em mais de uma classe e assumir diferentes implementações em cada uma dessas classes.



### 9ª QUESTÃO

A linguagem UML é baseada nos princípios da orientação aos objetos e trata da representação gráfica parcial de um sistema na sua fase de projeto, implementação ou de sistemas existentes. O paradigma de programação orientada aos objetos, fundamenta-se na utilização de objetos, que colaboram entre si, para a construção do software. A colaboração entre os objetos é feita por meio da troca de mensagens.

PERSEGUINE, V. R. **Modelagem de Software**. Maringá: Unicesumar, 2019.

Um dos pilares da programação orientada aos objetos tem como característica omitir os membros de uma classe, além de esconder como funcionam as rotinas ou as regras de negócio.

Assinale a alternativa que represente a definição anterior:

### ALTERNATIVAS

- ☐ Herança;
- ☐ Abstração;
- ☐ Composição;
- ☐ Polimorfismo;
- ☐ Encapsulamento.

### 10ª QUESTÃO

Uma associação é um relacionamento entre dois objetos que tem por objetivo representar os motivos e as regras que conduzem os objetos ao relacionamento. Um engenheiro de software pode utilizar o recurso de navegabilidade de uma associação para mostrar como um objeto de uma classe obtém acesso a um objeto de outra classe ou, em uma associação reflexiva, para um objeto da mesma classe.

PERSEGUINE, V. R. **Modelagem de Software**. Maringá: Unicesumar, 2019.

Nesse contexto, considere que uma página Web pode ser composta de gráficos, sendo que estes também são componentes de uma pasta do site. Na situação, o relacionamento de associação do tipo todo-parte estabelecido entre página Web e gráfico é de \_\_\_\_\_.

Assinale a alternativa que completa corretamente a afirmação.

### ALTERNATIVAS

- ☐ Realização.
- ☐ Agregação.
- ☐ Composição.
- ☐ Dependência.
- ☐ Generalização.