Tema 4

Módulo 1

**Ciclo de Vida de Desenvolvimento de Software**



Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

**Diagramas de interação**

As comunicações são explicitadas através da representação da interação durante as comunicações. Para tal, temos na UML os diagramas de interação, usados para mostrar o comportamento interativo de um sistema. Os diagramas de interação descrevem o fluxo de mensagens e fornecem contexto para uma ou mais linhas da vida de objetos dentro de um sistema. Além disso, os diagramas de interação podem ser usados para representar as sequências organizadas dentro de um sistema e servir como um meio para visualizar dados em tempo real.

Os diagramas de interação podem ser implementados em diversos cenários para fornecer um conjunto exclusivo de informações. Eles podem ser usados para modelar um sistema, como uma sequência de eventos organizada por tempo; fazer a [**engenharia reversa**](javascript:void(0)) ou avançada de um sistema ou processo; organizar a estrutura de vários eventos interativos; mostrar, de forma simples, o comportamento de mensagens e linhas da vida dentro de um sistema e até identificar possíveis conexões entre elementos de linhas da vida. Seus principais objetivos são:

Texto

O conteúdo gerado por IA pode estar incorreto.

Tipos de diagrama de interação:  
Diagrama

O conteúdo gerado por IA pode estar incorreto.

\***Os diagramas de interação servem para refinar o diagrama de classes na etapa de projeto, com o detalhamento dos métodos.**

**Diagrama de sequência**

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

\*Não é necessário construir diagramas de sequência para todos os casos de uso do sistema. Apenas os mais complexos, relacionados com o negócio da aplicação, são explorados por esse tipo de diagrama. Ele pode ser bastante trabalhoso e criá-lo para todas as partes do sistema pode não ser produtivo e nem ter benefícios compensatórios.\*

- Pode-se ter um diagrama principal nesse caso que segue o fluxo normal e diagramas complementarem para sequências alternativas e tratamento de erros assim como o de caso de uso.

**Elementos do diagrama de sequência**

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Tabela

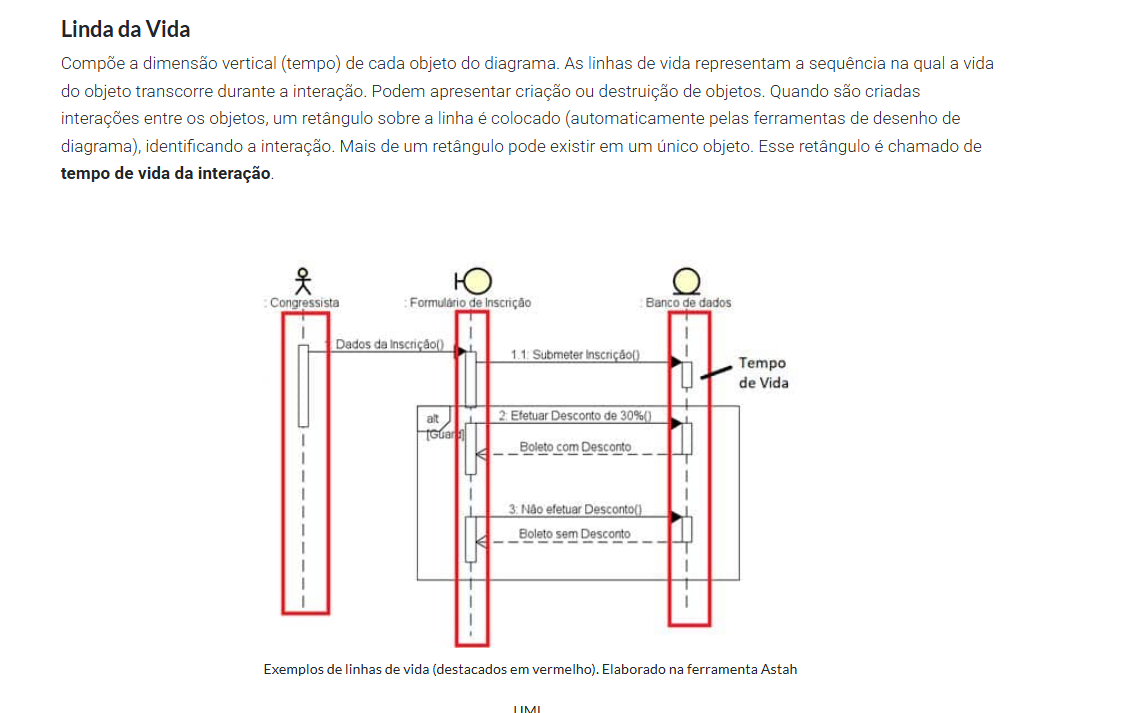
O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.



Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama, Esquemático

O conteúdo gerado por IA pode estar incorreto.

**Diagrama de comunicação**

Diagramas de comunicação servem para mostrar como os objetos interagem para executar o comportamento de um caso de uso ou de parte dele, para determinar interfaces e responsabilidades de classes e definir os papéis dos objetos que executam determinado fluxo de eventos. Mostram a comunicação, ou seja, o fluxo de mensagens que ocorre entre os objetos de um sistema de informação, assim como o diagrama de sequência. Os dois expressam informações semelhantes, mas numa forma de exibição diferente.

\*Quando usar o Diagrama de Sequência e quando usar o Diagrama de Comunicação:

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Como construir um diagrama de interação**

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Linha do tempo

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

**Em que momento construir o modelo de interações**

Uma imagem contendo Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Site

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Módulo 2

**Diagrama de classes de projeto**

Vamos tratar agora do diagrama de classes de projeto. Esse diagrama é um refinamento do diagrama de classes construído na fase de análise. Nesse refinamento, algumas classes sofrem alterações de suas propriedades com o objetivo de transformar o modelo de classes de análise no modelo de classes de projeto, assim como suas notações adicionais, transformações sobre atributos, operações e associações.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Especificação de atributos**





Tem-se também a default que é quando nenhuma visibilidade for especificada, esse é assumida.

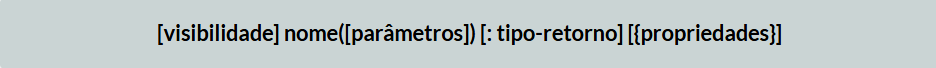
Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

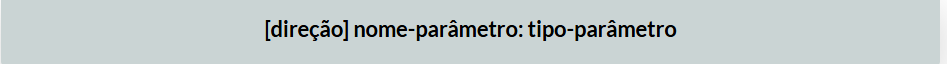
Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

**Especificação de operações/métodos**



No caso de parâmetros



Site

O conteúdo gerado por IA pode estar incorreto.

**Especificação de associações**

No modelo de classes de análise, relacionamentos entre objetos são normalmente definidos apenas com o uso da associação (ou como um de seus casos especiais, a generalização, a agregação ou a composição). As associações são os mecanismos que permitem aos objetos se comunicarem. Elas descrevem a conexão entre diferentes classes. Podem ter uma regra que especifica o propósito da associação e podem ser unidirecionais ou bidirecionais. Cada ponta da associação também possui um valor de multiplicidade que indica como os objetos de um lado se relacionam com os do outro lado. Existem diversos tipos de associação, que são modelados desde a etapa de análise:

Uma imagem contendo Linha do tempo

O conteúdo gerado por IA pode estar incorreto.

Na etapa de projeto, uma espécie de associação relevante é a dependência, que representa relacionamentos entre classes, onde uma é dependente da outra. Qualquer modificação na classe independente afetará diretamente objetos da classe dependente.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Uma dependência entre classes indica que uma classe depende dos serviços fornecidos pela outra. No modelo de análise, é suficiente para o modelador identificar a existência de associações entre classes, que é uma forma de dependência. Mas, na fase de especificação do modelo de classes, essa dependência precisa ser mais bem definida pelo projetista, uma vez que ela tem influência na forma utilizada para implementar as classes envolvidas.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

\*A notação da UML para representar uma dependência no diagrama de classes é de uma seta tracejada ligando as classes envolvidas. A seta sai da classe dependente e chega na classe da qual depende.\*

Tipos de dependências com rótulos:



**Transformação de associações em dependências**

No modelo de classes de análise, o modelador especifica uma ou mais associações entre uma classe e as demais apresentadas. Na passagem do modelo de classes de análise para o de projeto, o modelador deve estudar cada associação para identificar se ela pode ser transformada em dependências. A razão para essa transformação é definir melhor o encapsulamento. **Quanto menos dependências estruturais houver no modelo de classes, maior será a qualidade do projeto (do ponto de vista do encapsulamento e do acoplamento das classes constituintes).** De um modo geral, as associações que ligam classes de entidade permanecem como associações no modelo de projeto.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

**Classes de interface**

Uma classe de interface é uma coleção de operações com um nome e é utilizada para especificar um tipo de serviço sem especificar como será sua implementação. Essas classes não têm métodos concretos, apenas o declaram para que outra classe possa implementá-lo. Elas possibilitam que objetos externos ao sistema possam colaborar com uma ou mais classes do sistema.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Navegabilidade de associações**

As associações podem ser classificadas em bidirecionais e unidirecionais. Uma associação bidirecional indica que há um conhecimento mútuo entre os objetos associados. Ou seja, se um diagrama de classes exibe uma associação entre duas classes C1 e C2, então as duas assertivas a seguir são verdadeiras:

Texto

O conteúdo gerado por IA pode estar incorreto.

Graficamente, uma associação unidirecional é representada adicionando-se um sentido à associação. A classe para a qual o sentido aponta é aquela cujos objetos não possuem a visibilidade dos objetos da outra classe.

**Durante a construção do modelo de classes de análise, associações são normalmente consideradas “navegáveis” em ambos os sentidos, ou seja, as associações são bidirecionais. No modelo de classes de projeto, o modelador deve refinar a navegabilidade de todas as associações.**

**Texto

O conteúdo gerado por IA pode estar incorreto.**

**Tabela

O conteúdo gerado por IA pode estar incorreto.**

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Módulo 3

**Transição entre estados**  
Objetos de um sistema de software orientado a objetos se encontram em um estado particular a cada momento. Um objeto muda de estado quando acontece algum evento interno ou externo ao sistema. Quando um objeto muda de um estado para outro, diz-se que ele realizou uma **transição entre estados**.

\*\***O diagrama da UML utilizado para realizar essa análise é o diagrama de estados, também conhecido como diagrama de máquina de estado ou de transição de estado (DTE).**

Texto

O conteúdo gerado por IA pode estar incorreto.

**Diagrama de transição de estados**

Um estado é uma situação na vida de um objeto durante a qual ele satisfaz alguma condição ou realiza alguma atividade. Cada estado de um objeto é normalmente determinado pelos valores dos seus atributos e (ou) pelas suas ligações com outros objetos.

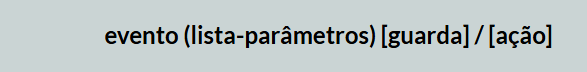
*A UML utiliza as seguintes notações para representar os estados:*

Uma imagem contendo Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

**Transições**

Os estados estão associados a outros pelas transições. Uma transição é mostrada como uma linha conectando estados, com uma seta apontando do estado origem para o estado destino. Quando ocorre uma transição entre estados, diz-se que a transição foi disparada. Note que, em uma transição, o estado subsequente pode ser igual ao estado original. Uma transição pode ser rotulada com uma expressão:



Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.



Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email, Site

O conteúdo gerado por IA pode estar incorreto.

**Identificação dos elementos de um diagrama de estados**

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

\*\*Outra fonte para identificação de eventos relevantes é analisar as regras de negócio definidas para o sistema. Normalmente, essas regras possuem informações sobre condições limites que permitem identificar estados e transições.

**Construção de diagramas de transições de estados**

Para sistemas bastante simples, a definição dos estados de todos os objetos não é tão trabalhosa. No entanto, a quantidade de estados possíveis de todos os objetos de um sistema complexo é grande.

**Para resolver o problema da explosão exponencial de estados, os diagramas de estados são desenhados por classe.**

Geralmente, cada classe é simples o suficiente para que o diagrama de estados correspondente seja compreensível. Note, contudo, que essa solução de dividir a modelagem de estados por classes do sistema tem a desvantagem de dificultar a visualização do estado do sistema como um todo. Essa desvantagem é parcialmente compensada pela construção de diagramas de interação.

Nem todas as classes de um sistema precisam de um DTE. Um diagrama de estados é desenhado somente para classes que exibem um comportamento dinâmico relevante. A relevância do comportamento de uma classe depende muito de cada situação em particular. No entanto, objetos cujo histórico precisa ser rastreado pelo sistema são típicos para se construir um diagrama de estados.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Exemplo de um diagrama de estados:

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Diagrama de atividades**

Um diagrama de atividade pode ser considerado um tipo especial de diagrama de estados, em que são representados os estados de uma atividade em vez dos estados de um objeto. **Ao contrário dos diagramas de estados, que são orientados a eventos, diagramas de atividade são orientados a fluxos de controle.**

\*\*O diagrama de atividade possui notação para representar ações concorrentes (paralelas), juntamente com a sua sincronização.

Exemplo de um diagrama de atividades:  
Diagrama

O conteúdo gerado por IA pode estar incorreto.

Ícone

O conteúdo gerado por IA pode estar incorreto.

**Fluxo de controle sequencial**

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Site

O conteúdo gerado por IA pode estar incorreto.

**Fluxo de controle paralelo**

Um diagrama de atividade pode conter fluxos de controle paralelos. Isso significa que é possível haver dois ou mais fluxos de controle sendo executados simultaneamente em um diagrama de atividades. Para sincronizar dois ou mais fluxos paralelos, as barras de sincronização são utilizadas. Há dois tipos de barra de sincronização:

Gráfico de caixa estreita

O conteúdo gerado por IA pode estar incorreto.

**Raias ou Partições (raias de natação)**

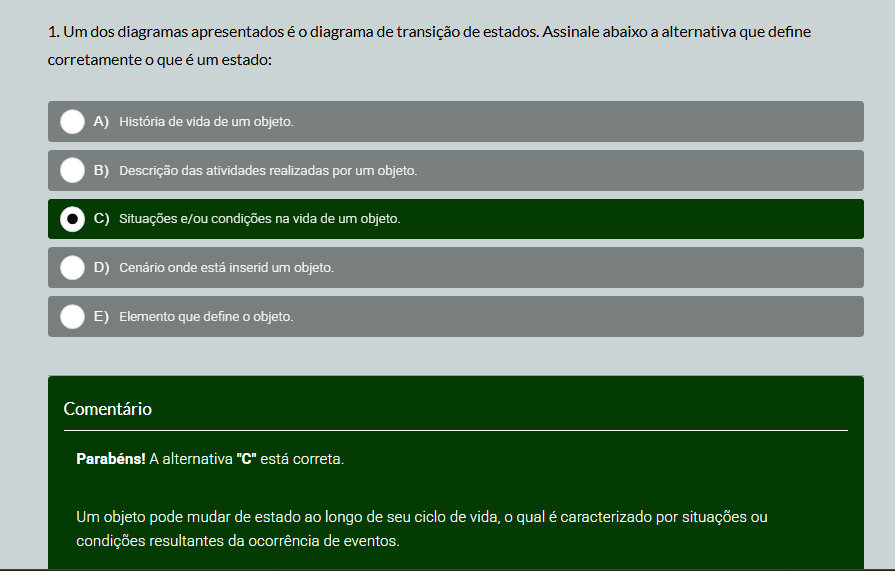
Existe para dividir a execução de processos. Isso normalmente ocorre em processos de negócio de uma organização, onde as tarefas são executadas por pessoas ou departamentos diversos.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Modelagem da lógica de uma operação complexa**

Quando um sistema de software é adequadamente decomposto em seus objetos constituintes e as responsabilidades de cada objeto estão bem definidas, a maioria das operações é bastante simples. Essas operações não necessitam de modelagem gráfica para serem entendidas. No entanto, em alguns casos, notadamente quando uma operação de uma classe de controle implementa uma regra de negócio, pode haver a necessidade de se descrever a lógica dessa operação ou da própria regra de negócio. Diagramas de atividade também podem ser utilizados com esse objetivo, nisso se assemelhando a um fluxograma.



Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Módulo 4

**Diagrama de componentes**

O diagrama de componentes apresenta o relacionamento entre os diferentes componentes de um sistema de software. Na UML, o termo "componente" refere-se a um módulo que representa sistemas ou subsistemas com capacidade de interagir. Para isso, existe uma abordagem de desenvolvimento em torno de componentes.

\*\***Nesse diagrama, são representados os diferentes componentes de software necessários para que o sistema funcione corretamente.**

Texto

O conteúdo gerado por IA pode estar incorreto.

**Componente**

Um componente nada mais é do que uma caixa preta (artefato do sistema) que possui vida autônoma e que, somente por meio de suas interfaces, oferece ou requer serviços de outras caixas pretas, sem que para isso tenha que conhecer seus conteúdos. Um componente é modelado ao longo do ciclo de vida de desenvolvimento, incluindo a concepção da funcionalidade do sistema por meio de casos de uso, definição das classes, entre outros.

Texto

O conteúdo gerado por IA pode estar incorreto.

Exemplo de um diagrama de componentes:

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Quando mais detalhes são necessários para a perfeita especificação de um componente, podemos definir uma estrutura interna de partes e conectores, usando o que podemos chamar de visão interna ou “caixa branca”. Nessa estrutura, geralmente estão contidos outros componentes. Esses componentes podem ser implementados por diferentes tipos de arquivo. (Conhecidos como OMG -> Object Management Group)

**Interfaces**

Interfaces são coleções de operações que especificam serviços de um componente. É por meio delas que os componentes se comunicam com o mundo externo, seja para oferecer ou receber serviços. Interfaces descrevem o comportamento visível externamente e não especificam qualquer estrutura (não podem incluir atributos), nem qualquer implementação.

Texto

O conteúdo gerado por IA pode estar incorreto.

**Conectores**

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

**Camadas**

A arquitetura de sistemas em camadas permite isolar a camada de apresentação das de regras de negócio e de armazenamento de dados. Isso permite aumentar o desempenho do sistema, além de facilitar a manutenção, atualização e substituição de componentes.

Texto

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Diagrama de implantação**

O diagrama de implantação especifica um conjunto de artefatos de infraestrutura que podem ser utilizados para definir a arquitetura física de execução de sistemas de informação. Ele foca a organização da arquitetura sobre a qual o software irá ser implantado e executado em termos de hardware, software básico e redes de comunicação, ou seja, quais máquinas (computadores, servidores, switches etc.), quais programas de software básico (sistema operacional, sistema gerenciador de banco de dados, browser) serão necessários para suportar o sistema, bem como definir como essas máquinas serão conectadas, com qual velocidade de conexão e quais protocolos de comunicação são utilizados.

\*\*Um diagrama de implantação representa a estrutura física do sistema em si, o que, assim como o diagrama de componentes, pode ser feito de várias formas.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Nós do diagrama de implantação**

O **diagrama de implantação** representa dispositivos de hardware e ambientes de execução de software de forma aninhada. Seus principais elementos são os **nós**, que podem representar servidores, computadores, sistemas operacionais, bancos de dados e navegadores. Os nós podem conter outros nós e são conectados por **caminhos de comunicação** para demonstrar topologias de rede.

Os principais estereótipos usados são:

* **<<device>>**: Representa dispositivos computacionais com capacidade de processamento.
* **<<artifact>>**: Representa artefatos físicos do sistema, como programas executáveis, bancos de dados e documentos.
* **<<execution environment>>**: Representa ambientes de execução, como sistemas operacionais e gerenciadores de banco de dados.

O diagrama de implantação é útil para comunicação entre equipes de desenvolvimento e infraestrutura, ajudando na configuração e instalação do sistema. Entretanto, ele é mais relevante para sistemas distribuídos e complexos, não sendo necessário para sistemas que rodam em um único computador.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Exercícios:

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.