Texto 1

Um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos existentes entre eles. Os diagramas de classes também mostram as propriedades e as operações de uma classe e as restrições que se aplicam à maneira como os objetos estão conectados. A UML utiliza a palavra característica como um termo geral que cobre as propriedades e operações de uma classe.

FOWLER, Martin. UML Essencial. Grupo A, 2011.

Texto 2

O modelo relacional é muito simples e elegante: um banco de dados é uma coleção de uma ou mais relações, em que cada relação é uma tabela com linhas e colunas. Essa representação tabular simples permite que até usuários iniciantes entendam o conteúdo de um banco de dados e possibilita o uso de linguagens de alto nível simples para consultar os dados. As principais vantagens do modelo relacional em relação aos modelos de dados mais antigos são sua representação de dados simples e a facilidade com que mesmo consultas complexas podem ser expressas.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de Gerenciamento de Bancos de Dados. Grupo A, 2008.

Texto 3

O principal construtor para representar dados no modelo relacional é a relação. Uma relação consiste em um esquema de relação e em uma instância de relação. A instância de relação é uma tabela, e o esquema de relação descreve os cabeçalhos de coluna da tabela. Primeiro, descreveremos o esquema de relação e depois a instância de relação. O esquema especifica o nome da relação, o nome de cada campo (ou coluna ou atributo) e o domínio de cada campo. Um domínio é descrito em um esquema de relação pelo nome de domínio e tem um conjunto de valores associados.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de Gerenciamento de Bancos de Dados. Grupo A, 2008.

Considerando os textos acima e os recursos de modelagem estudados nas unidades, defina as formas de modelar dados e conceitos em classes e bancos de dados, descrevendo os objetivos de cada modelo e discuta as relações entre eles para a abstração adequada do software a ser implementado.

Resposta:

No desenvolvimento de software, duas abordagens são essenciais para modelar dados: a modelagem de classes e a modelagem relacional. Cada uma tem um papel importante na construção de sistemas eficientes e organizados.

## Modelagem de Classes

A modelagem de classes descreve a estrutura estática do sistema, definindo as classes, seus atributos (propriedades) e métodos (operações). Muito comum em sistemas orientados a objetos, ela é representada por diagramas na UML, que ajudam a visualizar as relações entre as entidades, como associações, herança, composição e agregação.

Exemplo:
Em um sistema de biblioteca, podemos ter a classe Livro, com atributos como título, autor e anoPublicacao, além de métodos como emprestar() e devolver(). Essa classe pode se relacionar com a classe Usuário, indicando quem está com o livro no momento.

## Modelagem Relacional

Já a modelagem relacional organiza os dados que serão armazenados no banco de dados, utilizando tabelas (relações). Cada tabela representa uma entidade, e suas linhas correspondem às instâncias dessa entidade. Um dos grandes benefícios desse modelo é a facilidade de consulta e manipulação de dados por meio de SQL.

Exemplo:
No mesmo sistema de biblioteca, podemos ter uma tabela Livros com colunas como id_livro, título, autor e ano_publicacao. A tabela Emprestimos pode relacionar os livros aos usuários por meio de chaves estrangeiras, garantindo a rastreabilidade dos empréstimos.

## Relação entre os Modelos

Embora tenham propósitos distintos, os dois modelos se complementam. A modelagem de classes organiza a estrutura e o comportamento dos objetos no sistema, enquanto a modelagem relacional define como esses dados serão armazenados de forma eficiente. Para integrar esses dois modelos, muitas vezes usamos ferramentas de mapeamento objeto-relacional (ORM), que facilitam a conversão entre objetos no código e registros no banco de dados. Essa integração é fundamental para garantir que o sistema seja não apenas funcional, mas também eficiente e fácil de manter.