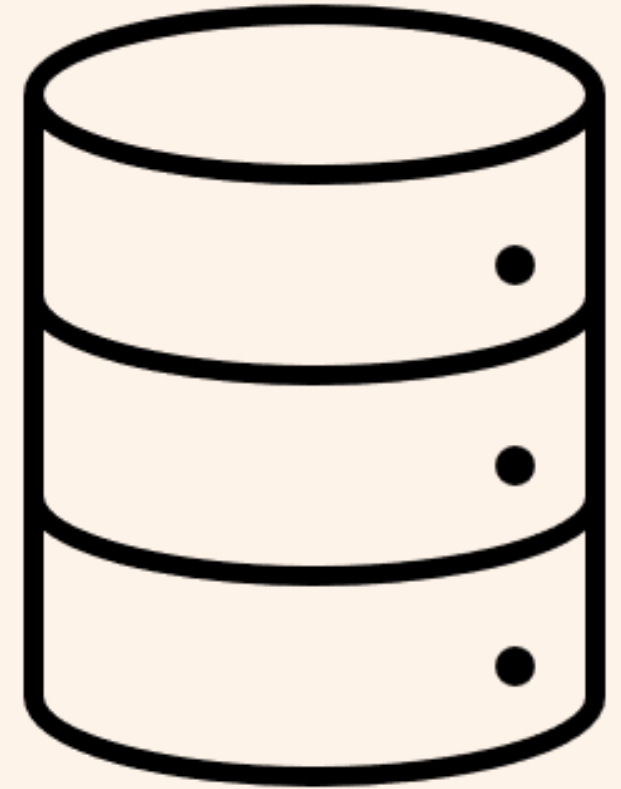


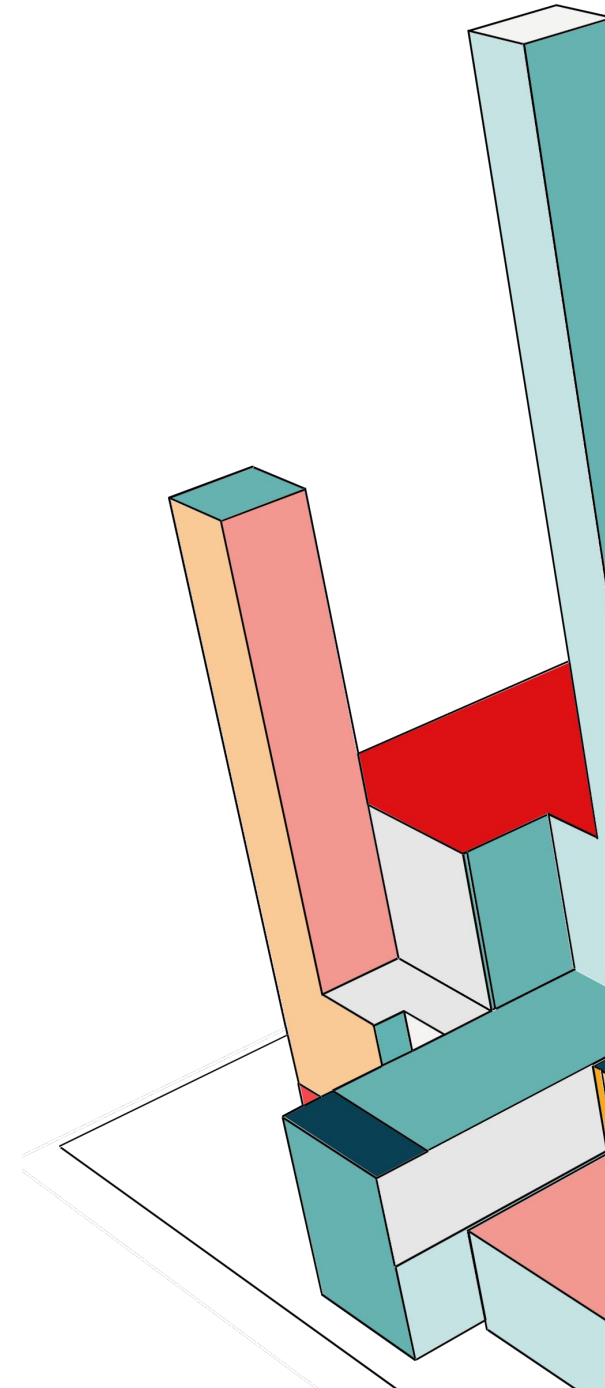
INTRODUÇÃO AOS BANCOS DE DADOS

PROFESSOR: JORGE BALDEZ



AGENDA

- O que é banco de dados?
- Componentes de um banco de dados
- Sistema Gerenciador de Banco de Dados (SGBD)
- A Importância da Linguagem SQL na Manipulação de Dados
- Banco de dados NoSQL
- Outros tipos de banco de dados
- Papel do Administrador do Banco de Dados

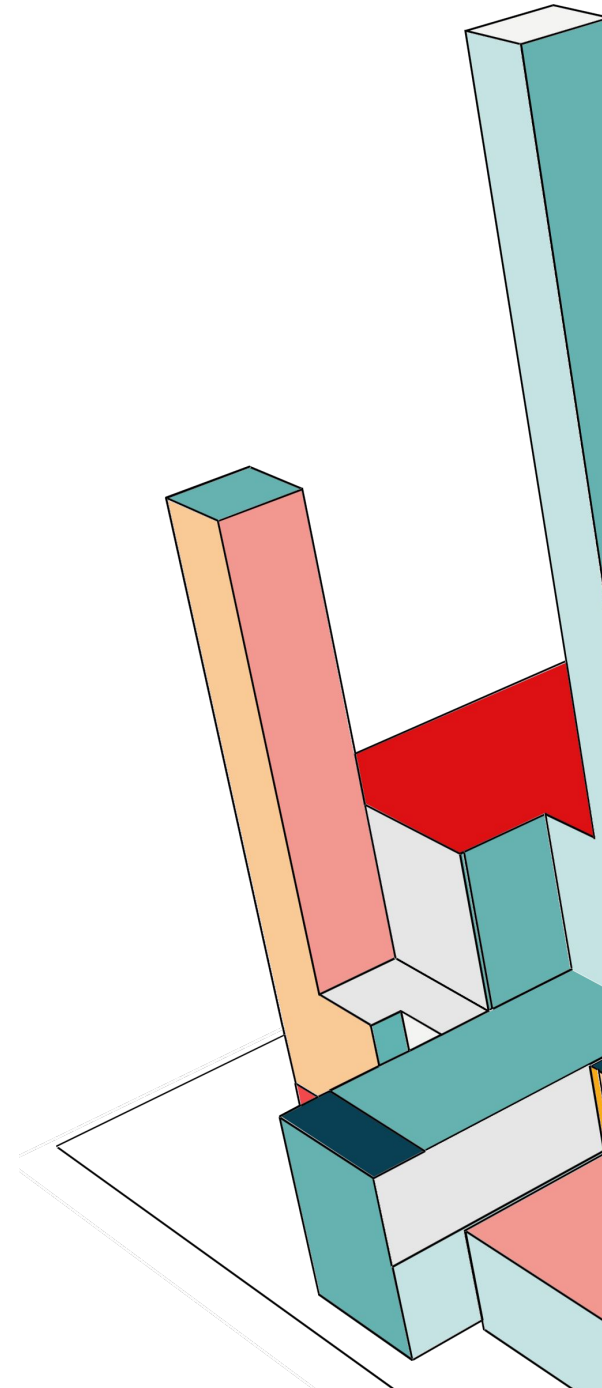


O QUE É UM BANCO DE DADOS?



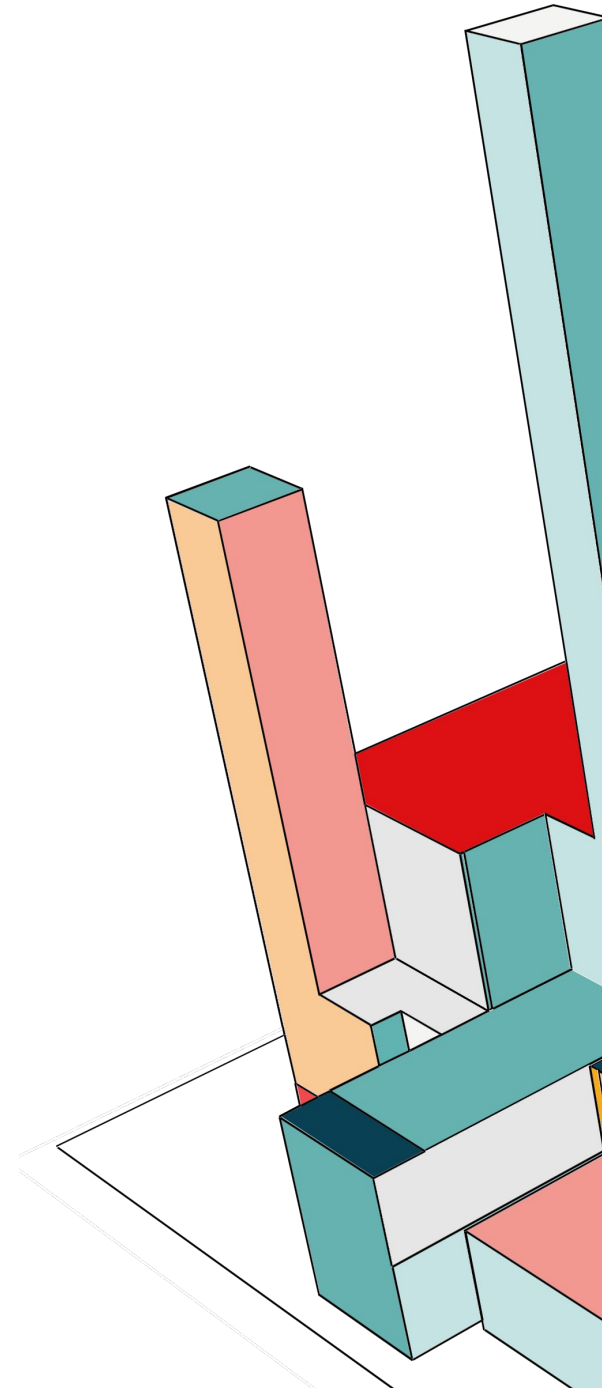
UMA GRANDE COLEÇÃO DE DADOS

- **Imagine uma biblioteca.** Em uma biblioteca, os livros são organizados em prateleiras, por assunto, autor ou título, para que você possa encontrar facilmente o livro que procura.
- **Um banco de dados é como uma biblioteca gigante para dados digitais.** Em vez de livros, ele armazena informações sobre pessoas, produtos, eventos, e qualquer outra coisa que você queira registrar. Essas informações são organizadas de forma estruturada, ou seja, seguem um padrão específico, para que seja fácil encontrá-las quando precisar.



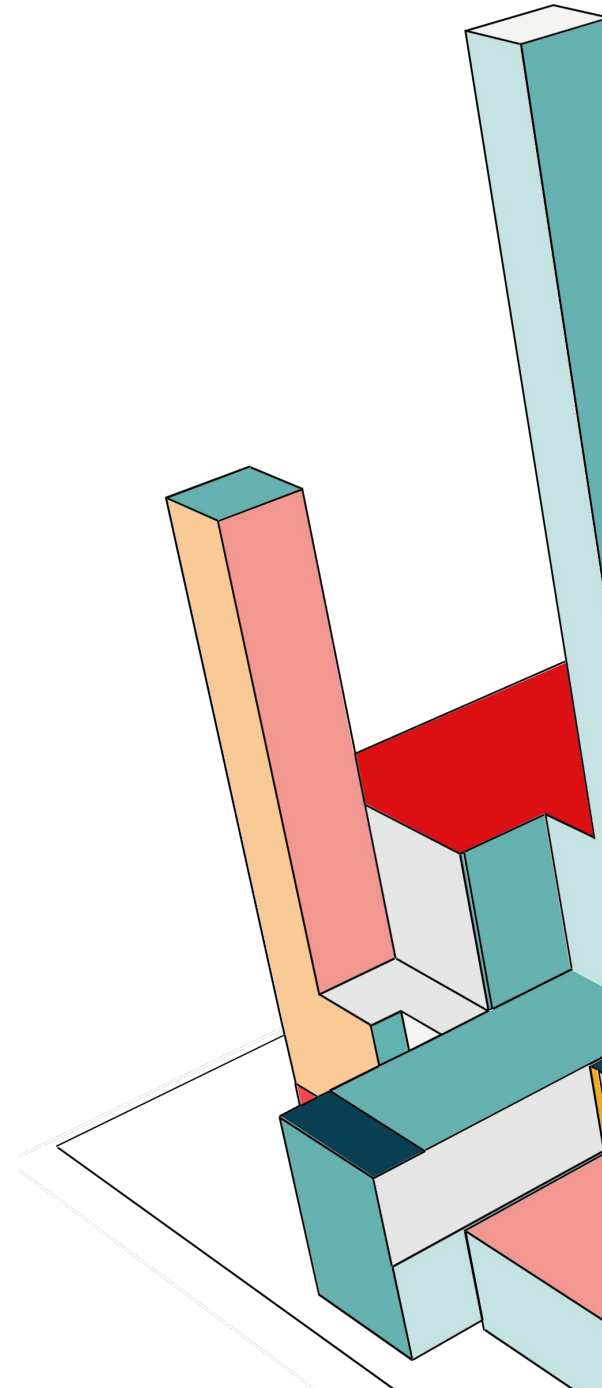
POR QUE ORGANIZAR OS DADOS?

- **Facilidade de busca:** Assim como você busca um livro por título em uma biblioteca, você pode buscar um dado específico em um banco de dados.
- **Gerenciamento eficiente:** É mais fácil adicionar, atualizar ou remover informações quando elas estão organizadas.
- **Integridade dos dados:** A organização garante que os dados sejam precisos e consistentes.
- **Compartilhamento de informações:** Várias pessoas podem acessar e utilizar os mesmos dados de forma segura.



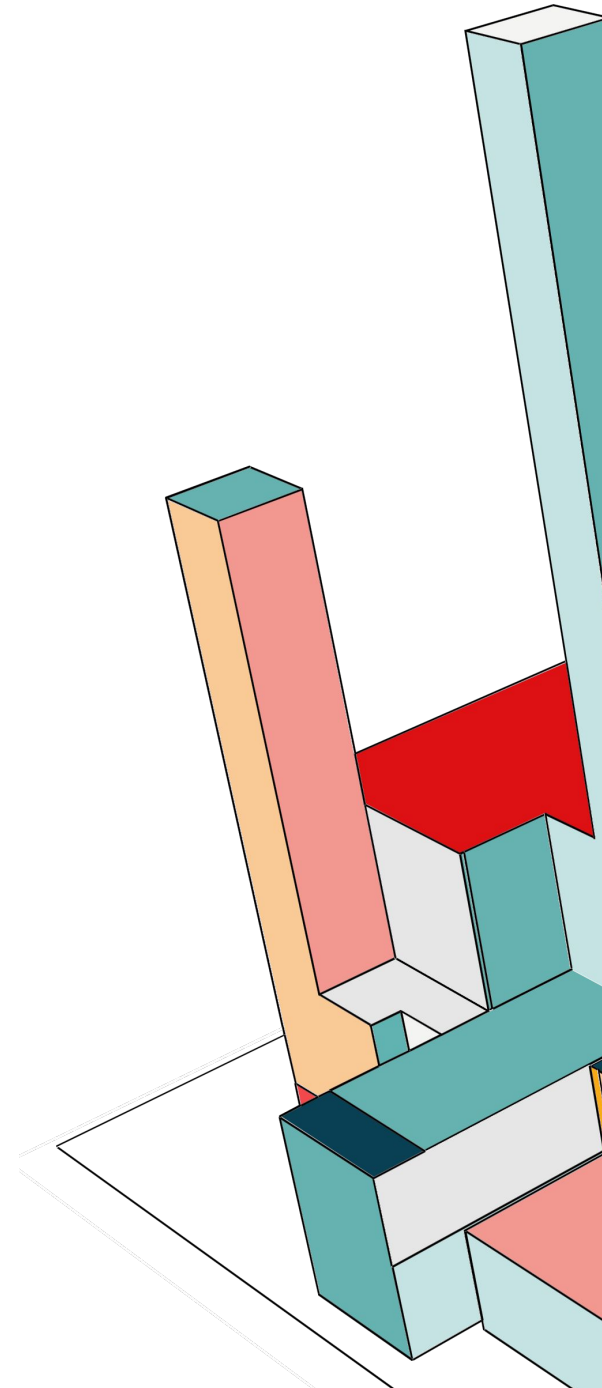
POR QUE ORGANIZAR OS DADOS?

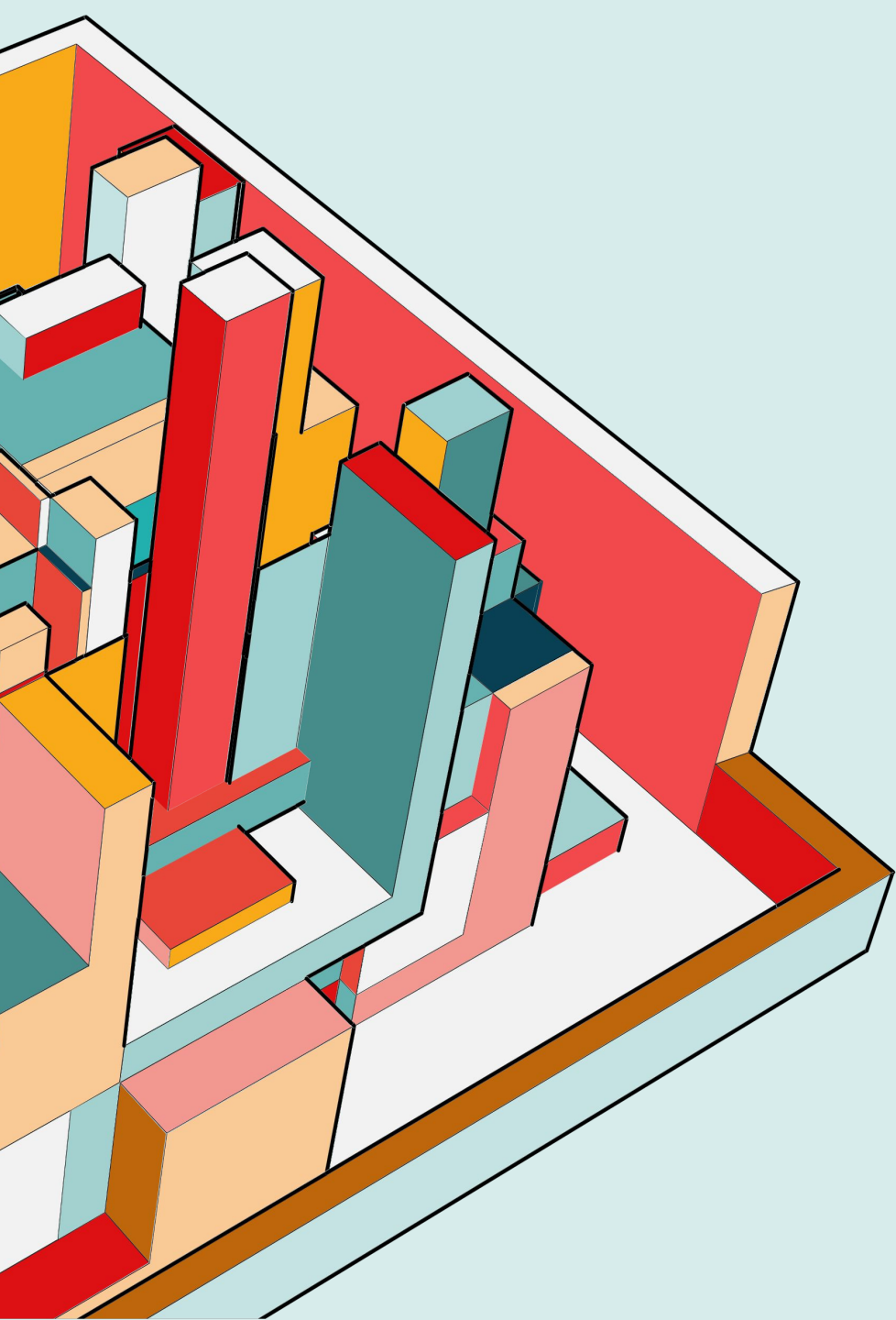
- Um banco de dados é um conjunto organizado de informações, estruturado para:
- **Armazenar:** Guardar grandes volumes de dados de forma segura.
- **Gerenciar:** Adicionar, atualizar, excluir e modificar os dados.
- **Recuperar:** Encontrar rapidamente as informações desejadas.



UM EXEMPLO PRÁTICO

- Um banco de dados de uma loja online armazena informações sobre:
 - **Cientes:** nome, endereço, histórico de compras
 - **Produtos:** descrição, preço, estoque
 - **Pedidos:** data, itens comprados, forma de pagamento
- **Para que serve um banco de dados?**
 - Os bancos de dados são essenciais para o funcionamento de diversas áreas, como:
 - **Empresas:** gerenciamento de clientes, produtos, finanças
 - **Governo:** registros de cidadãos, serviços públicos
 - **Ciência:** armazenamento de dados de pesquisas
 - **Redes sociais:** informações sobre usuários e suas interações

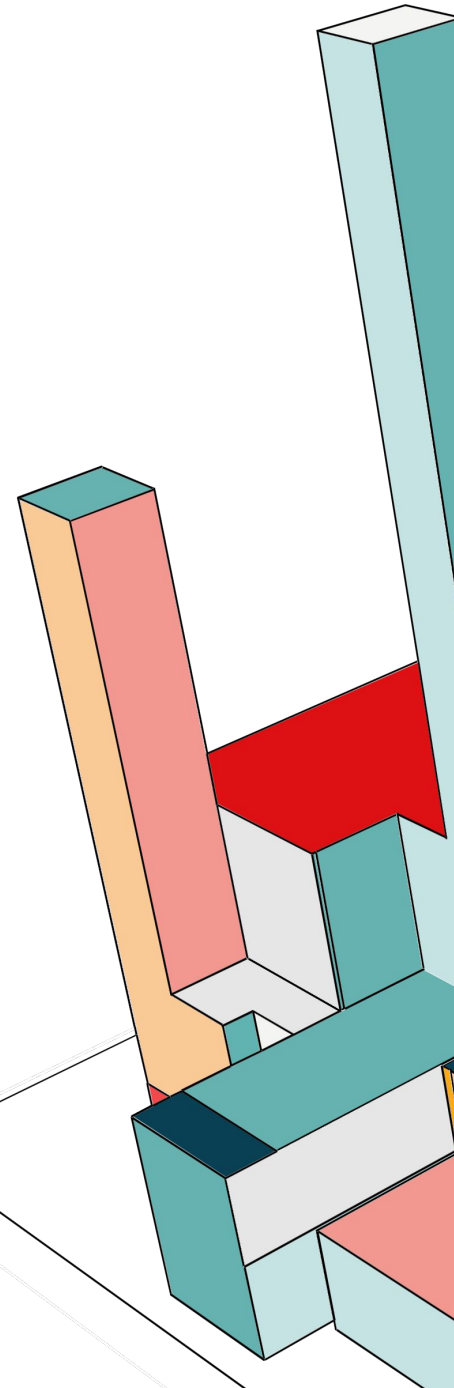




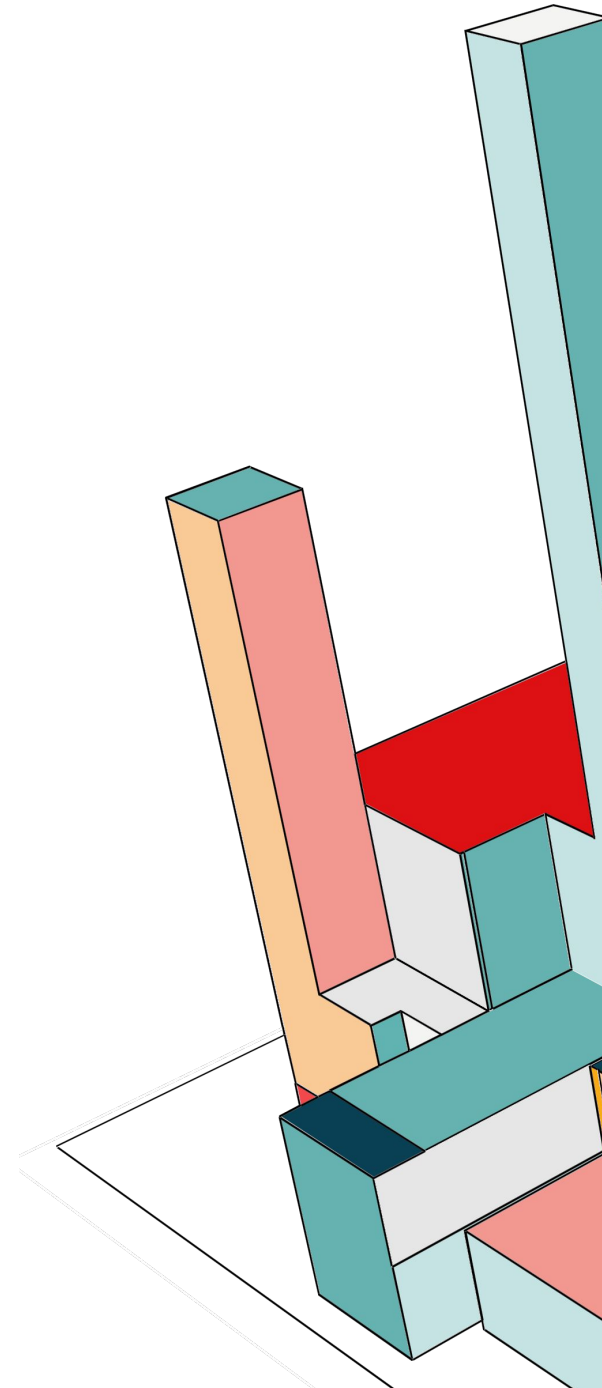
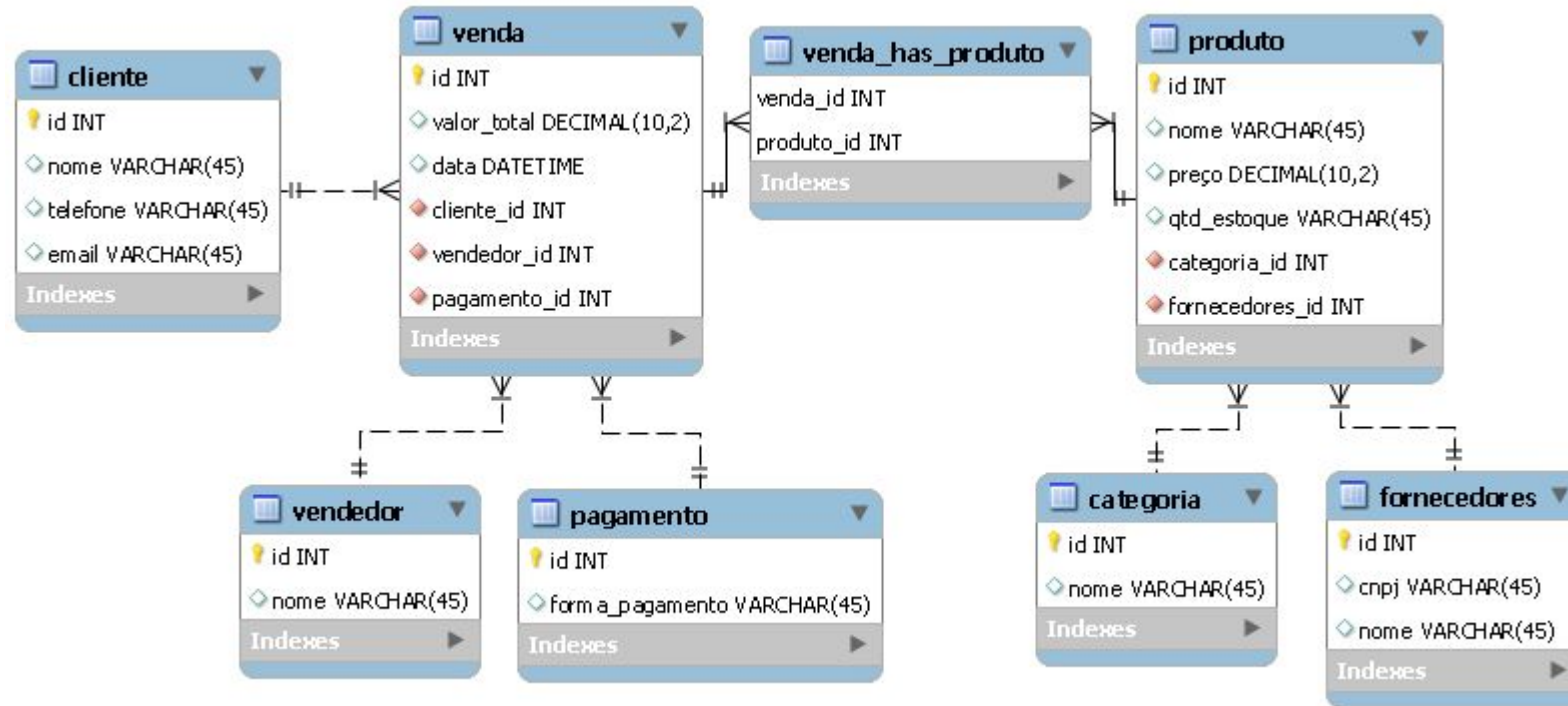
COMPONENTES DE UM BANCO DE DADOS

OS INGREDIENTES DE UM BANCO DE DADOS

- **Tabelas:** São como as caixas organizadoras de um armário. Cada tabela armazena um conjunto específico de informações, como clientes, produtos ou pedidos.
- **Campos:** São as gavetas dentro das caixas. Cada gaveta armazena um tipo específico de informação, como o nome de um cliente, o preço de um produto ou a data de um pedido.
- **Registros:** São os itens guardados dentro das gavetas. Cada registro representa uma instância única de um dado, como um cliente específico ou um pedido particular.
- **Chaves:** São as etiquetas que identificam de forma única cada registro. Elas são como os códigos de barras dos produtos, permitindo encontrar rapidamente a informação desejada.
- **Relações:** São os laços que conectam as diferentes tabelas. Elas mostram como as informações estão relacionadas entre si, por exemplo, um cliente pode fazer vários pedidos.



OS INGREDIENTES DE UM BANCO DE DADOS



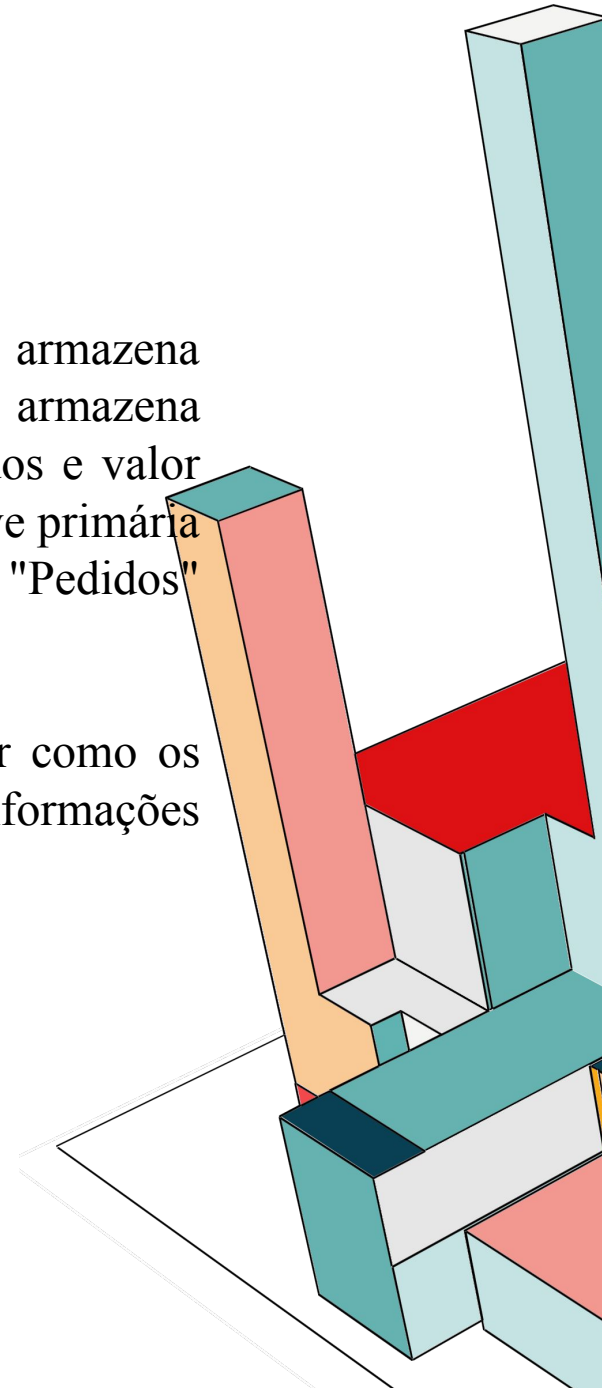
OS INGREDIENTES DE UM BANCO DE DADOS

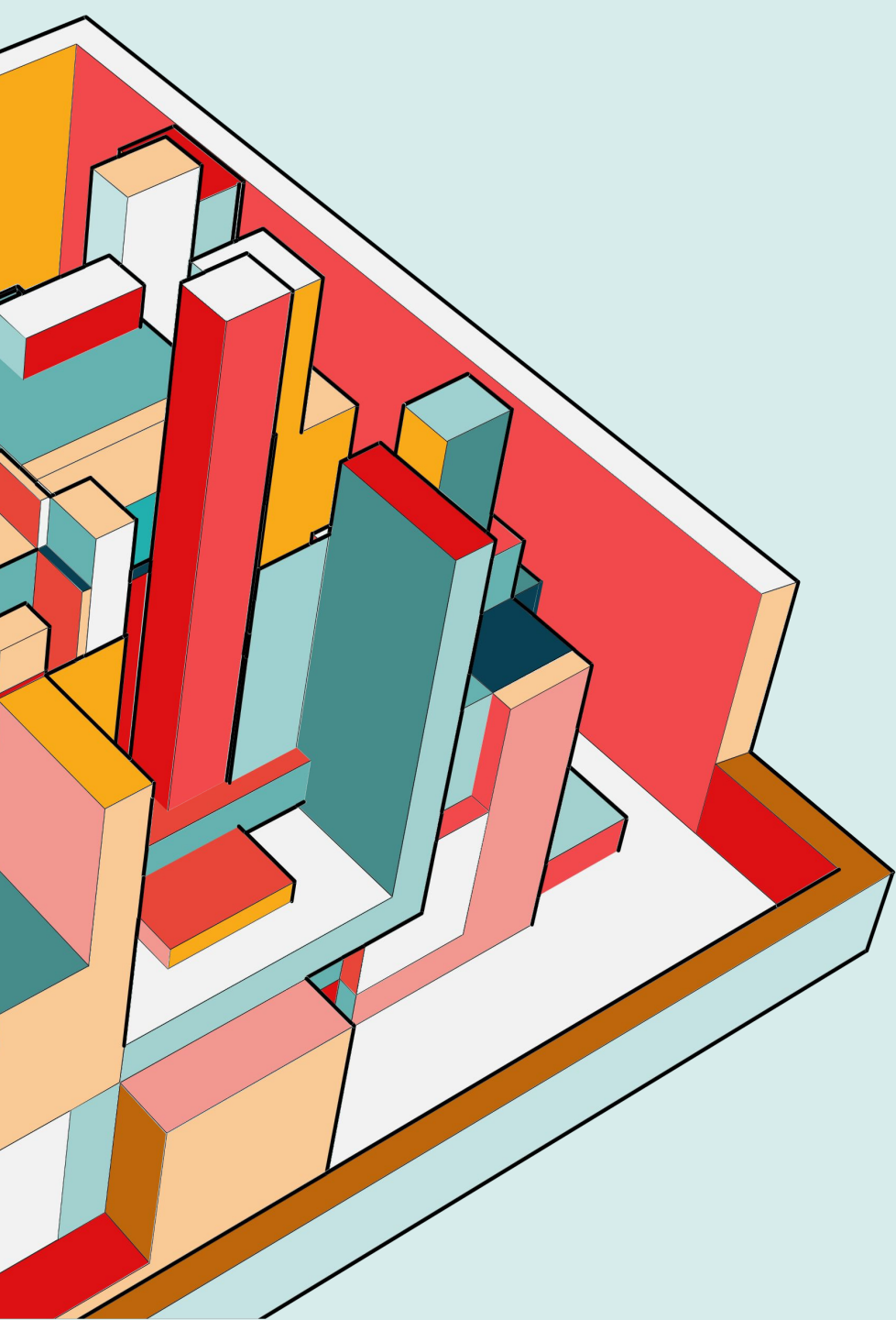
Para entender melhor, vamos usar o exemplo de uma loja online. A tabela "Clientes" armazena informações sobre os clientes, como nome, endereço e e-mail. A tabela "Pedidos" armazena informações sobre os pedidos feitos pelos clientes, como data do pedido, itens comprados e valor total. A relação entre essas duas tabelas é que um cliente pode fazer vários pedidos. A chave primária da tabela "Clientes" poderia ser o CPF do cliente, enquanto a chave primária da tabela "Pedidos" poderia ser o número do pedido.

Ao entender esses componentes, você estará dando o primeiro passo para compreender como os bancos de dados funcionam e como eles podem ser utilizados para organizar e gerenciar informações de forma eficiente.

Em resumo:

- **Tabelas:** Contêineres para os dados.
- **Campos:** Atributos dentro das tabelas.
- **Registros:** Linhas individuais dentro das tabelas.
- **Chaves:** Identificadores únicos para os registros.
- **Relações:** Conexões entre as tabelas.





SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD)

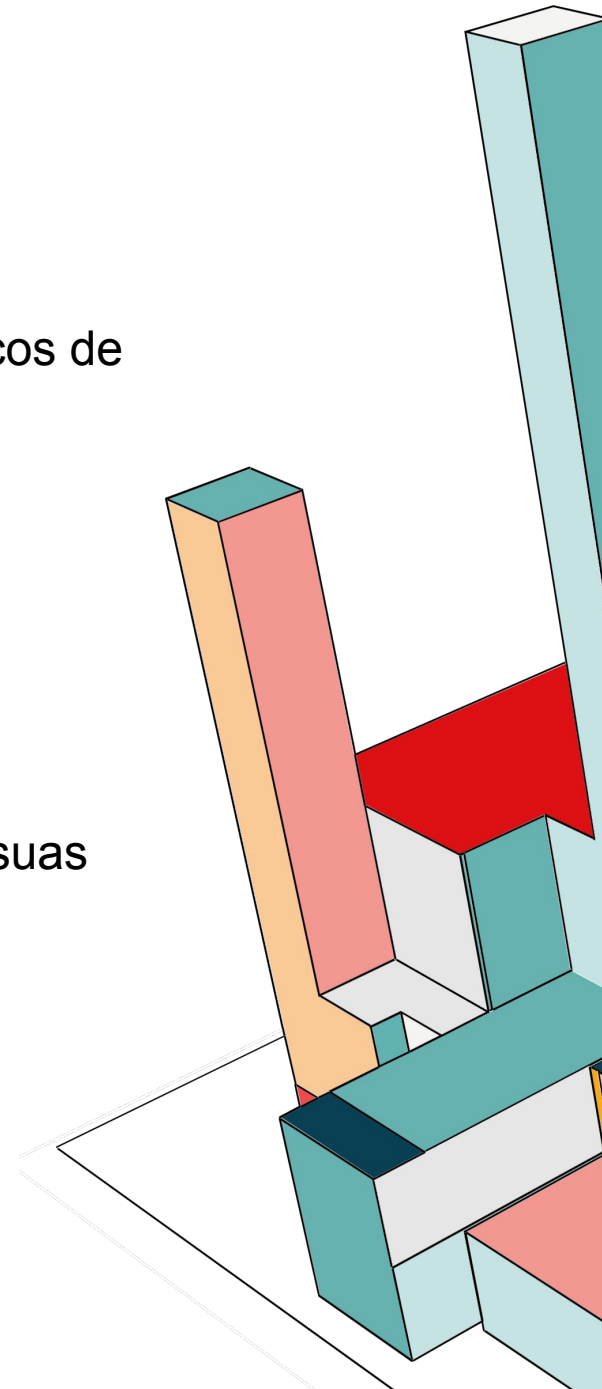
OS INGREDIENTES DE UM BANCO DE DADOS

Definição de SGBD: É um software que permite criar, manipular e gerenciar bancos de dados. Pense nele como um construtor que cria e mantém um prédio de dados.

Funcionalidades:

- **Criação:** Construir as estruturas do banco de dados (tabelas, índices, etc.).
- **Manipulação:** Inserir, atualizar e excluir dados dentro dessas estruturas.
- **Consulta:** Pesquisar e recuperar informações específicas do banco de dados.

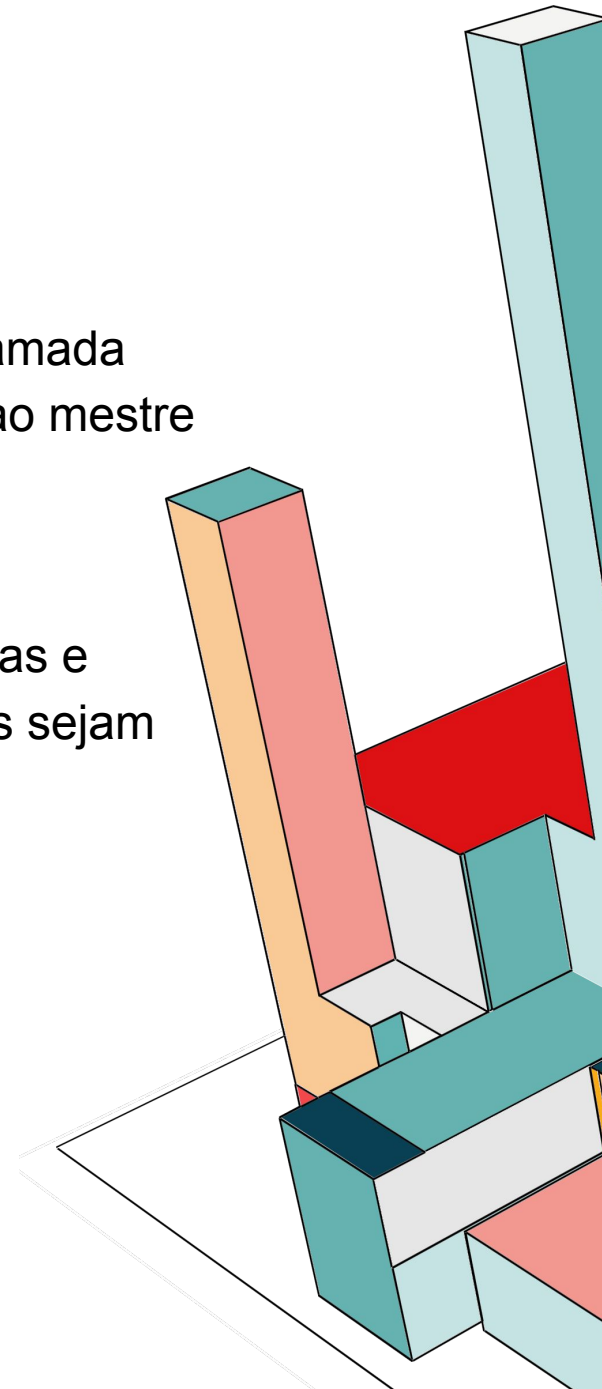
Exemplos de SGBDs: MySQL, PostgreSQL, Oracle, SQL Server. Cada um com suas características e focos, como um marceneiro, pedreiro ou eletricista, cada um especializado em uma parte da construção.



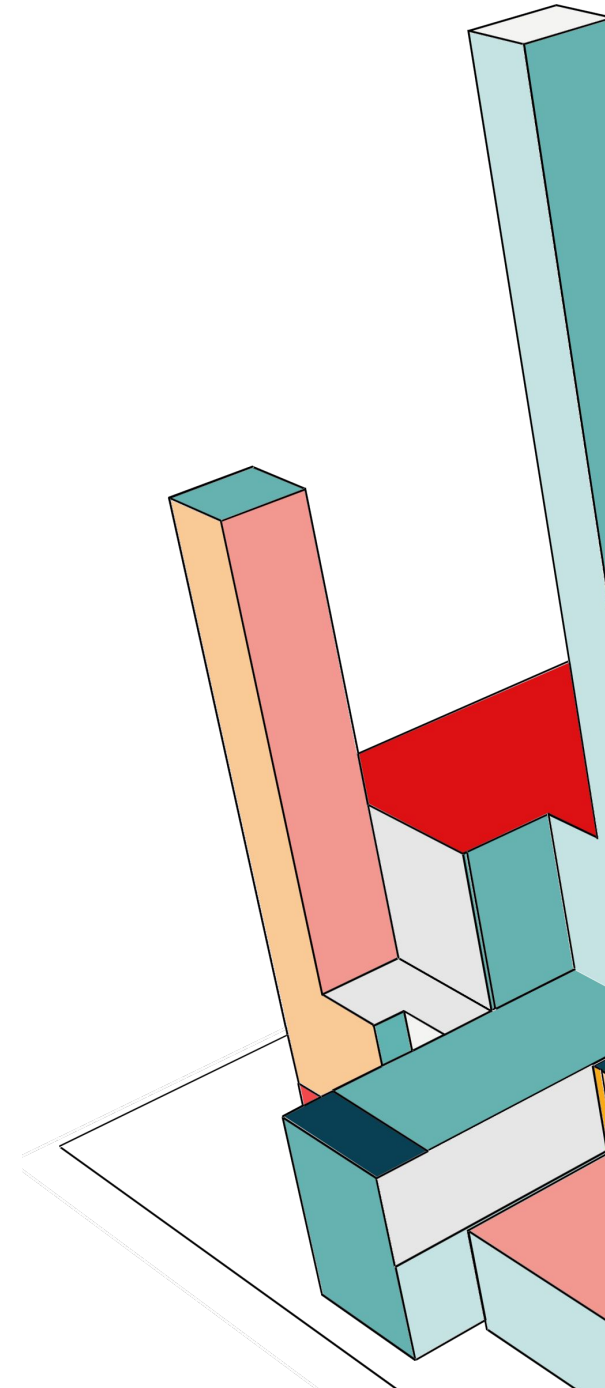
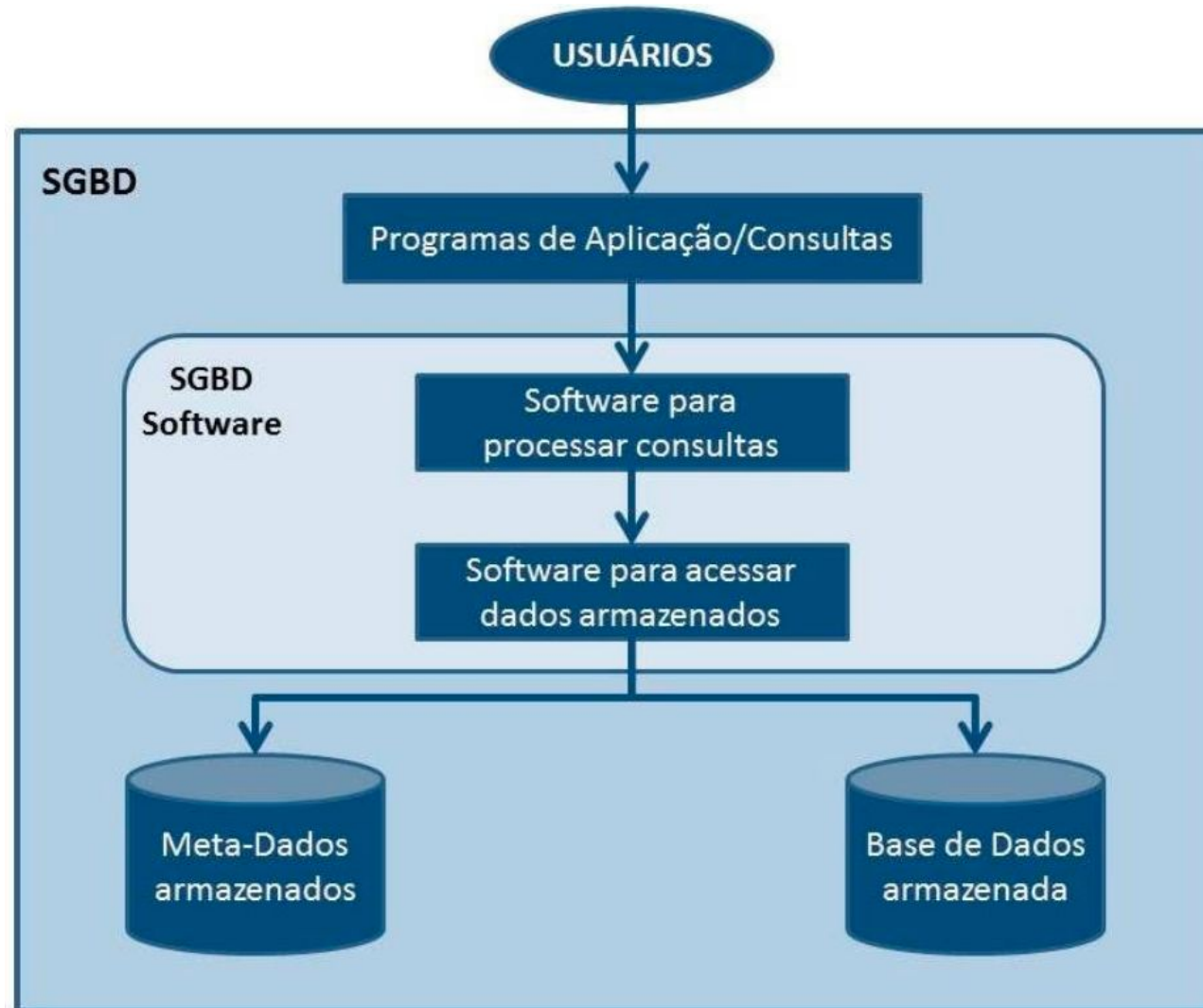
COMO O SGBD INTERAGE COM O USUÁRIO E OS DADOS

Usuário: O usuário interage com o SGBD através de uma linguagem especial chamada SQL (Structured Query Language). É como se o usuário estivesse dando ordens ao mestre de obras, dizendo o que ele quer construir ou modificar.

Dados: O SGBD armazena os dados em um formato otimizado para buscas rápidas e eficientes. Ele também garante a integridade dos dados, evitando que informações sejam perdidas ou corrompidas.



O MESTRE DE OBRAS DOS DADOS: O SGBD



POR QUE USAR UM SGBD?

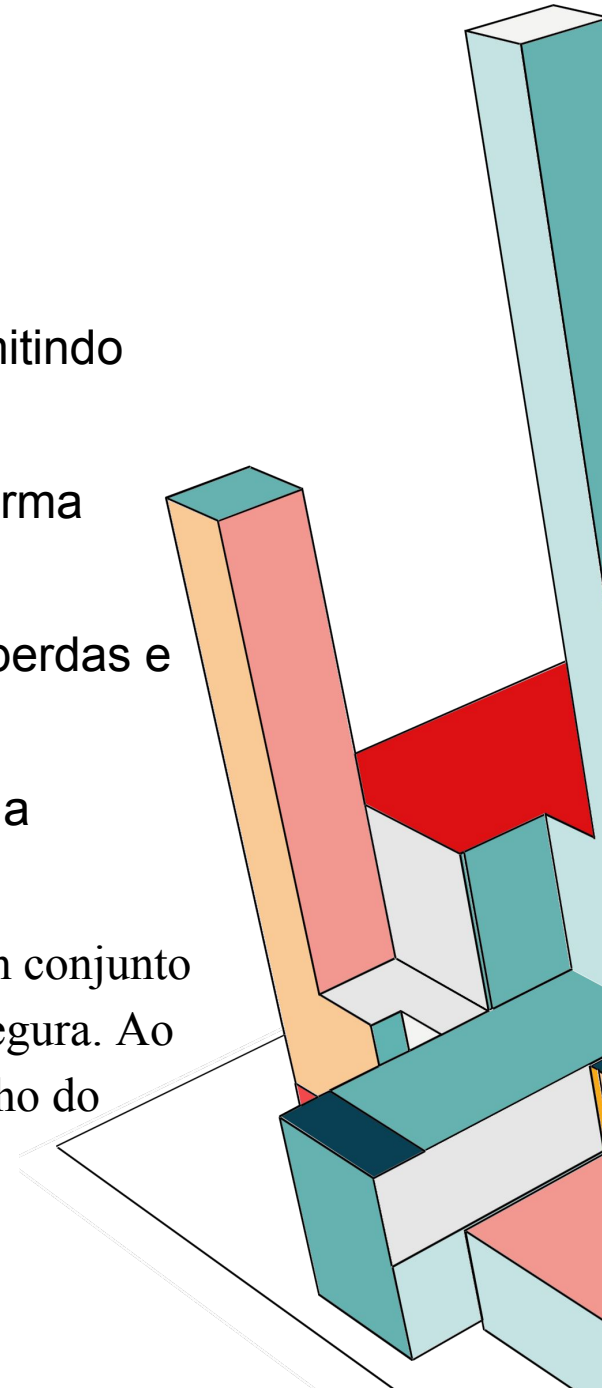
Facilidade de uso: O SGBD esconde a complexidade da gestão dos dados, permitindo que o usuário se concentre na informação em si.

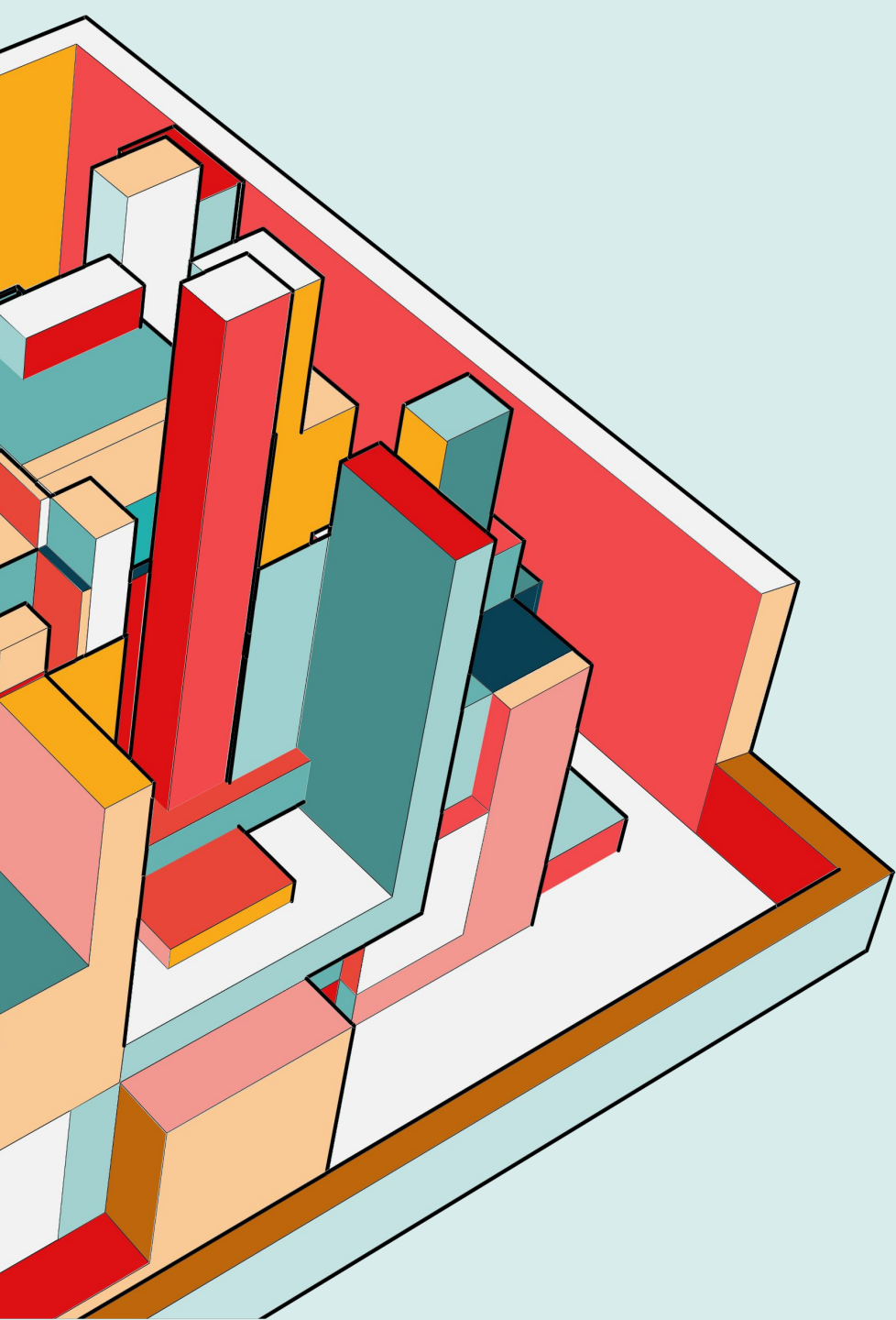
Performance: Os SGBDs são otimizados para realizar consultas complexas de forma rápida e eficiente.

Confiabilidade: Eles garantem a integridade e a segurança dos dados, evitando perdas e inconsistências.

Flexibilidade: Permitem a criação de bancos de dados complexos e a adaptação a diferentes necessidades.

O SGBD é a ferramenta essencial para quem trabalha com bancos de dados. Ele oferece um conjunto completo de funcionalidades para criar, manipular e consultar dados de forma eficiente e segura. Ao escolher um SGBD, é importante considerar as necessidades do seu projeto, como o tamanho do banco de dados, a complexidade das consultas e o nível de segurança requerido.





MODELOS DE DADOS

MODELOS DE DADOS: UMA VISÃO GERAL

Modelo Relacional:

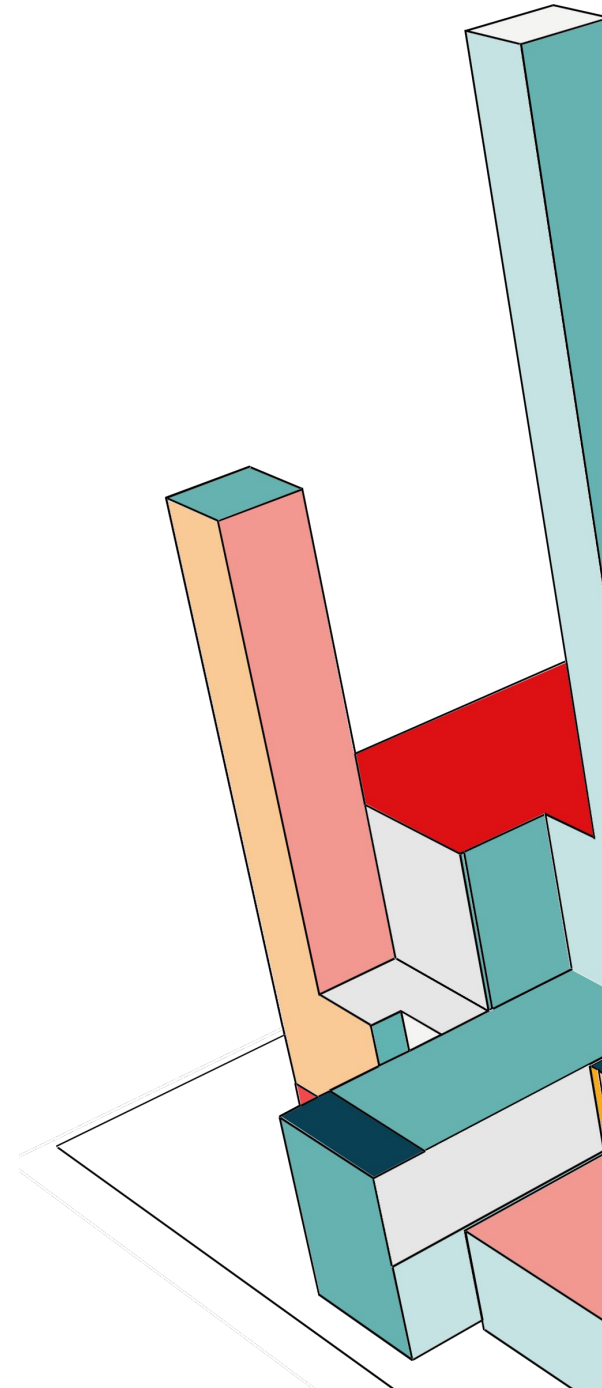
- Baseado em tabelas com linhas e colunas.
- Relacionamentos entre tabelas definidos por chaves.
- Mais utilizado devido à sua estrutura clara e padronizada.

Modelo Hierárquico:

- Estrutura em árvore, com um nó raiz e nós filhos.
- Relacionamentos hierárquicos entre os dados.
- Menos flexível que o modelo relacional.

Modelo em Rede:

- Extensão do modelo hierárquico, permitindo múltiplos pais para um nó.
- Complexidade maior e menos utilizado na prática.



MODELOS DE DADOS: UMA VISÃO GERAL

Modelo Documental:

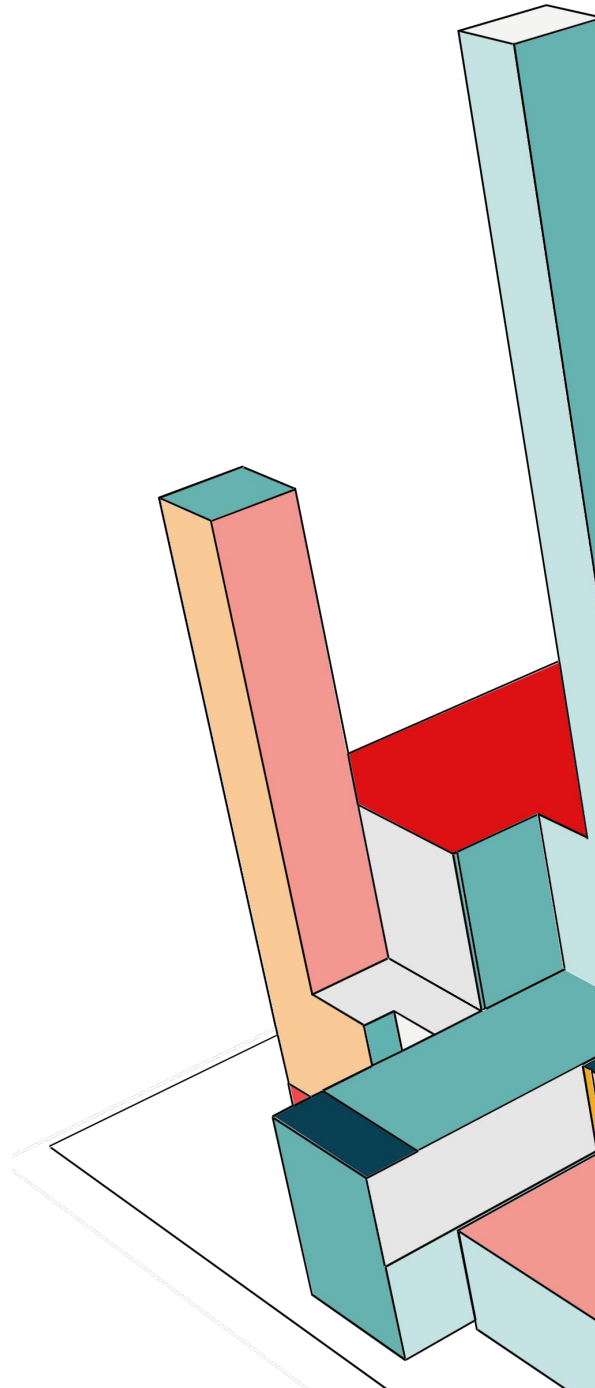
- Dados armazenados em documentos JSON ou BSON.
- Flexível e escalável, ideal para dados semiestruturados.

Modelo Chave-Valor:

- Dados armazenados como pares chave-valor.
- Simples e rápido, mas com menos estrutura que outros modelos.

Modelo Gráfico:

- Representa dados como grafos, com nós e arestas.
- Ideal para dados com relações complexas e não lineares.



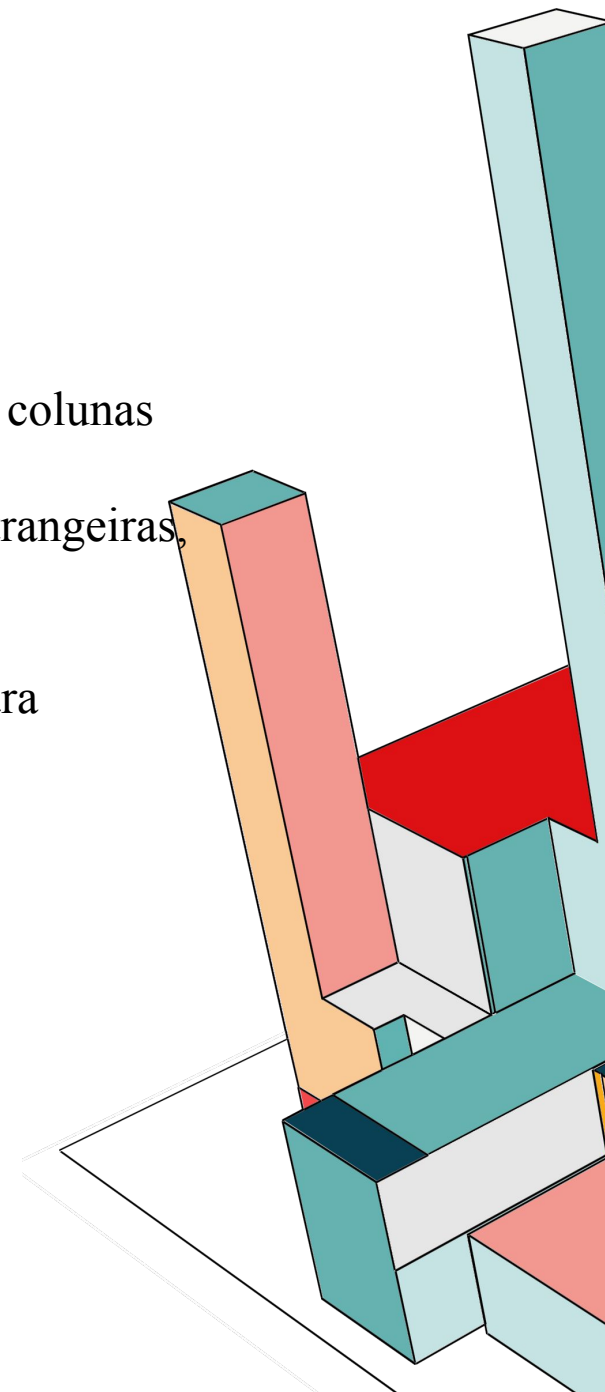
BANCOS DE DADOS RELACIONAIS: A BASE DA ESTRUTURAÇÃO DE DADOS

- **Conceito de Tabela e Relacionamento:**

- **Tabelas:** Unidades básicas de armazenamento, organizadas em linhas (registros) e colunas (campos).
- **Relacionamentos:** Ligações entre tabelas, estabelecidas por chaves primárias e estrangeiras, garantindo a integridade dos dados.

- **Linguagem SQL:**

- **Padrão para bancos relacionais:** Linguagem de consulta estruturada, utilizada para manipular, consultar e modificar dados.
- **Operações:** Seleção, inserção, atualização e exclusão de dados.



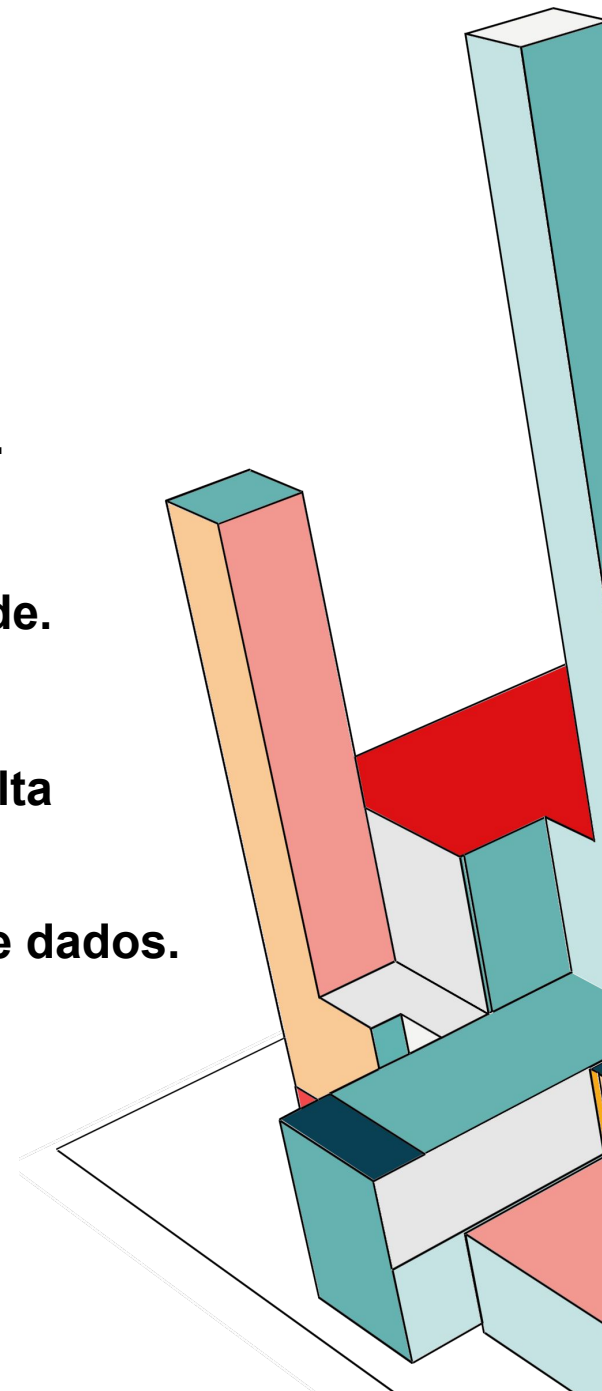
BANCOS DE DADOS RELACIONAIS: A BASE DA ESTRUTURAÇÃO DE DADOS

Vantagens:

- **Estrutura organizada:** Facilita a compreensão e a manutenção dos dados.
- **Integridade dos dados:** Garantida por relacionamentos e restrições.
- **Padronização:** SQL é amplamente utilizado, facilitando a interoperabilidade.

Desvantagens:

- **Escalabilidade:** Pode ser desafiadora para grandes volumes de dados e alta concorrência.
- **Rigidez:** Esquemas fixos podem limitar a flexibilidade para novos tipos de dados.

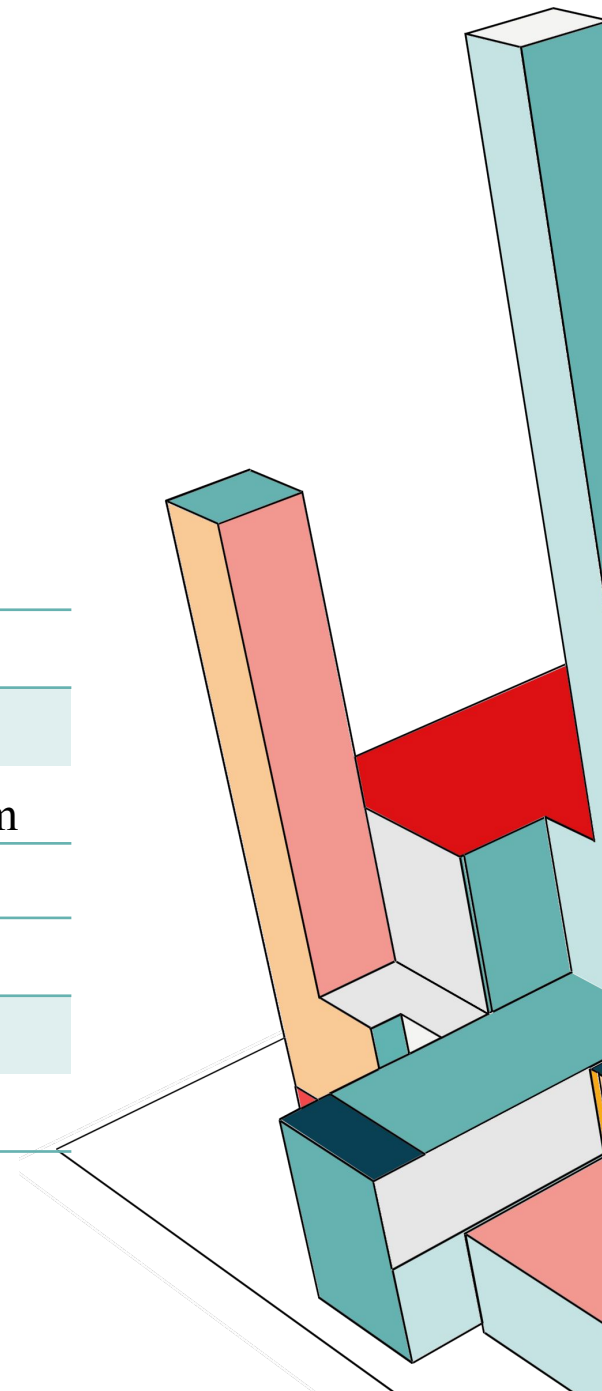


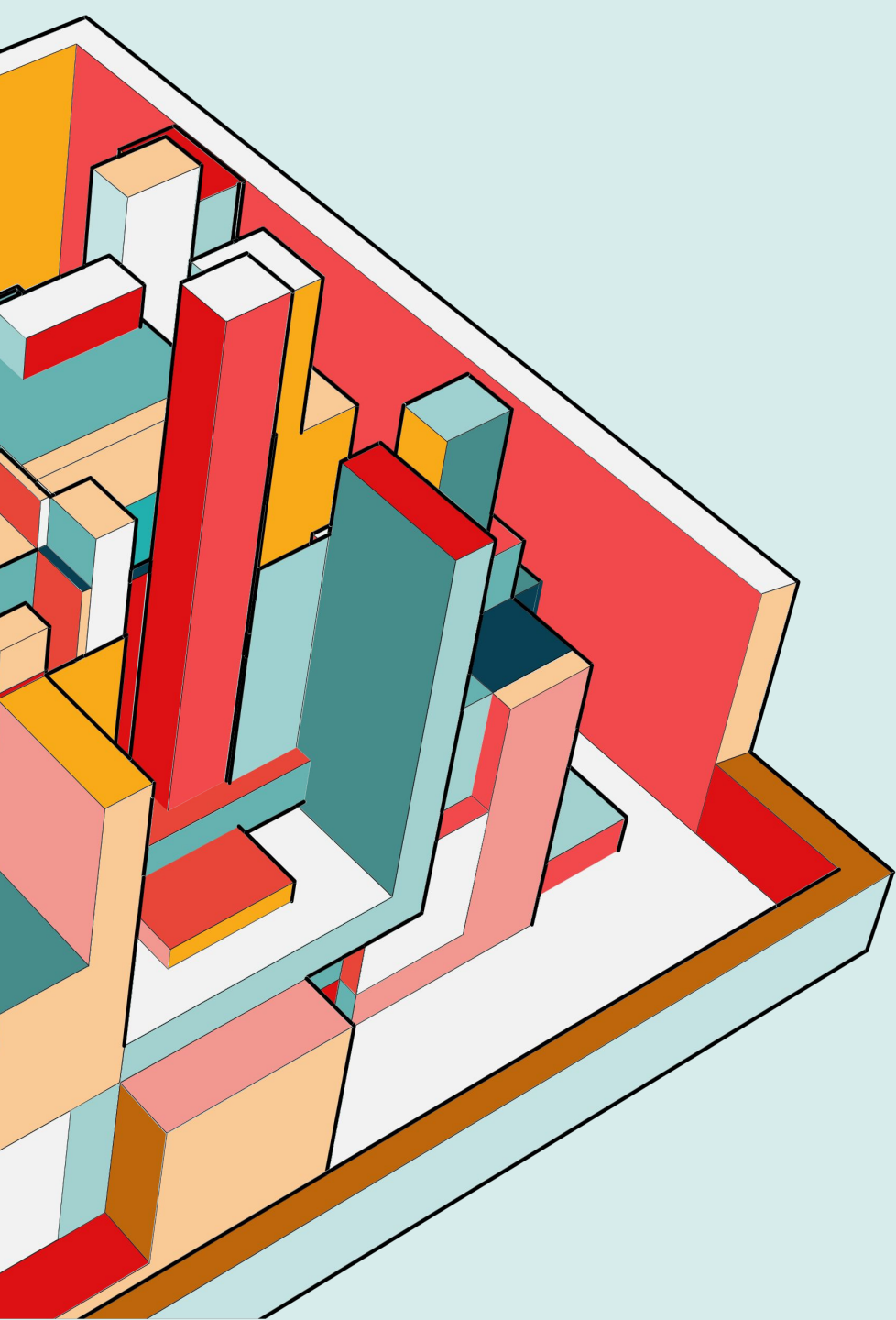
BANCOS DE DADOS RELACIONAIS: A BASE DA ESTRUTURAÇÃO DE DADOS

Exemplo de relacionamento: A tabela "Pedidos" poderia ter uma coluna "ID_Cliente" como chave estrangeira, referenciando a tabela "Clientes" e estabelecendo uma relação entre os pedidos e seus respectivos clientes.

ID_Cliente	Nome	Email
1	João Silva	Joao.silva@sistema.com
2	Maria Souza	Maria.Souza@sistema.com

ID_Cliente	ID_pedidos	Produto
1	1	camisa
2	2	Bermuda





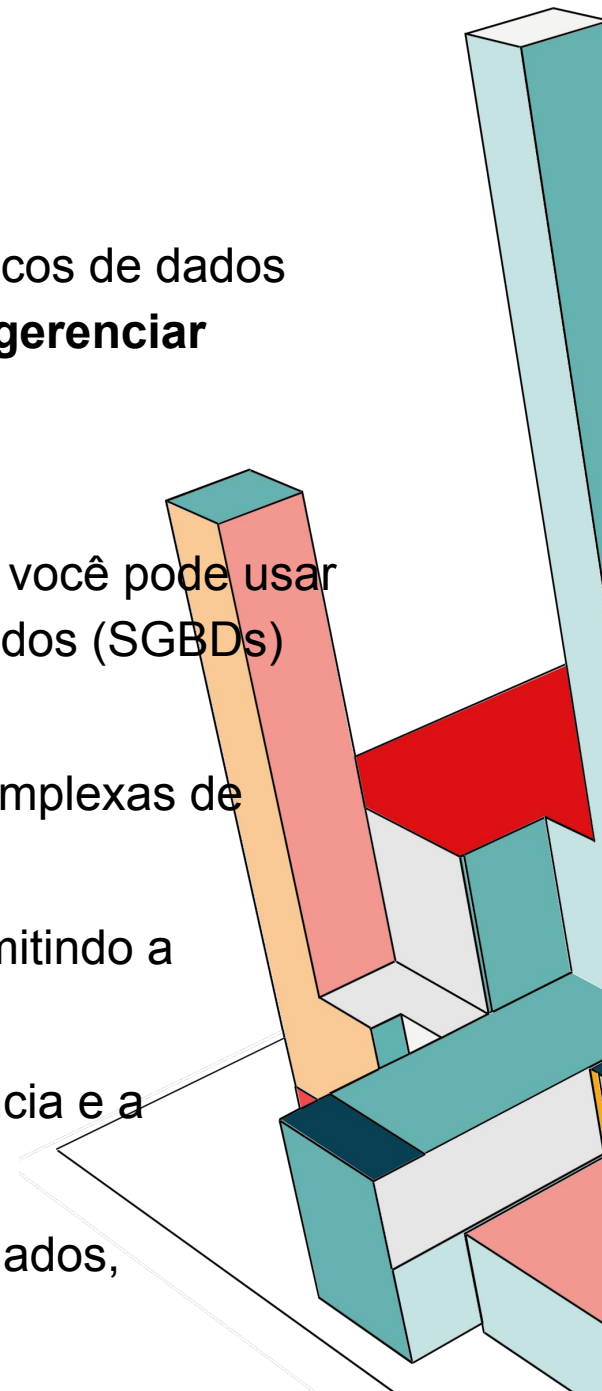
A IMPORTÂNCIA DA LINGUAGEM SQL NA MANIPULAÇÃO DE DADOS

BANCOS DE DADOS RELACIONAIS: A BASE DA ESTRUTURAÇÃO DE DADOS

SQL (Structured Query Language) é a linguagem padrão para interagir com bancos de dados relacionais. Sua relevância reside na sua capacidade de **manipular, consultar e gerenciar dados de forma eficiente e segura.**

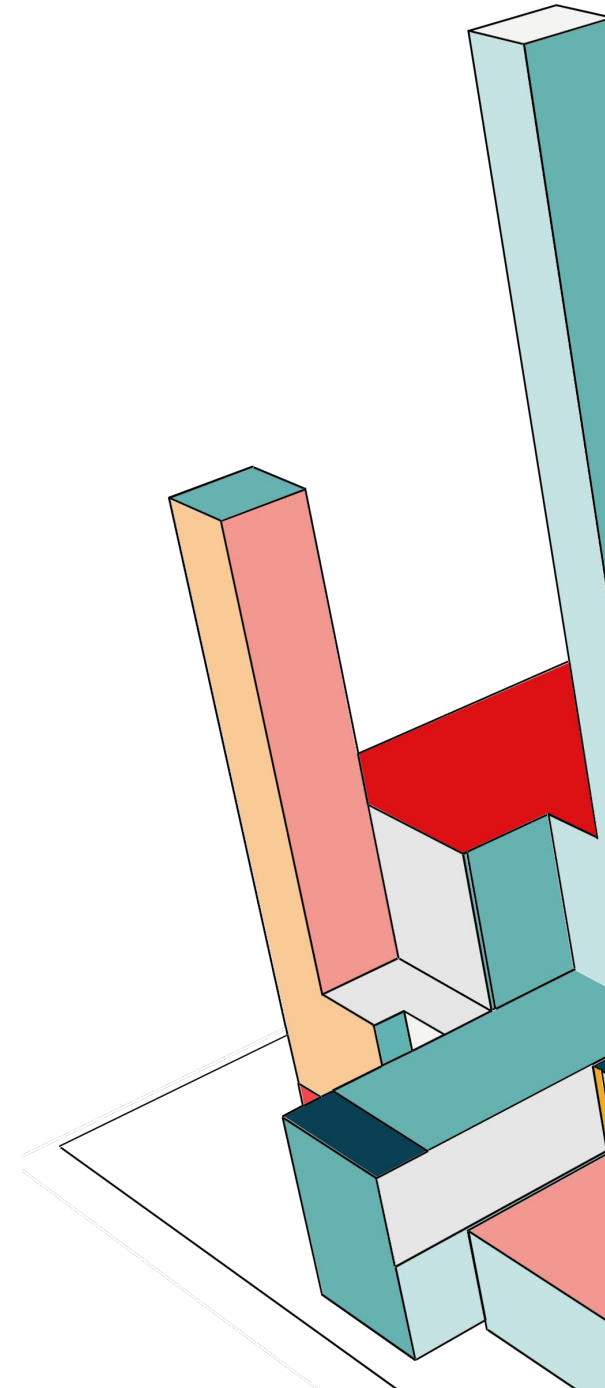
Por que a SQL é tão importante?

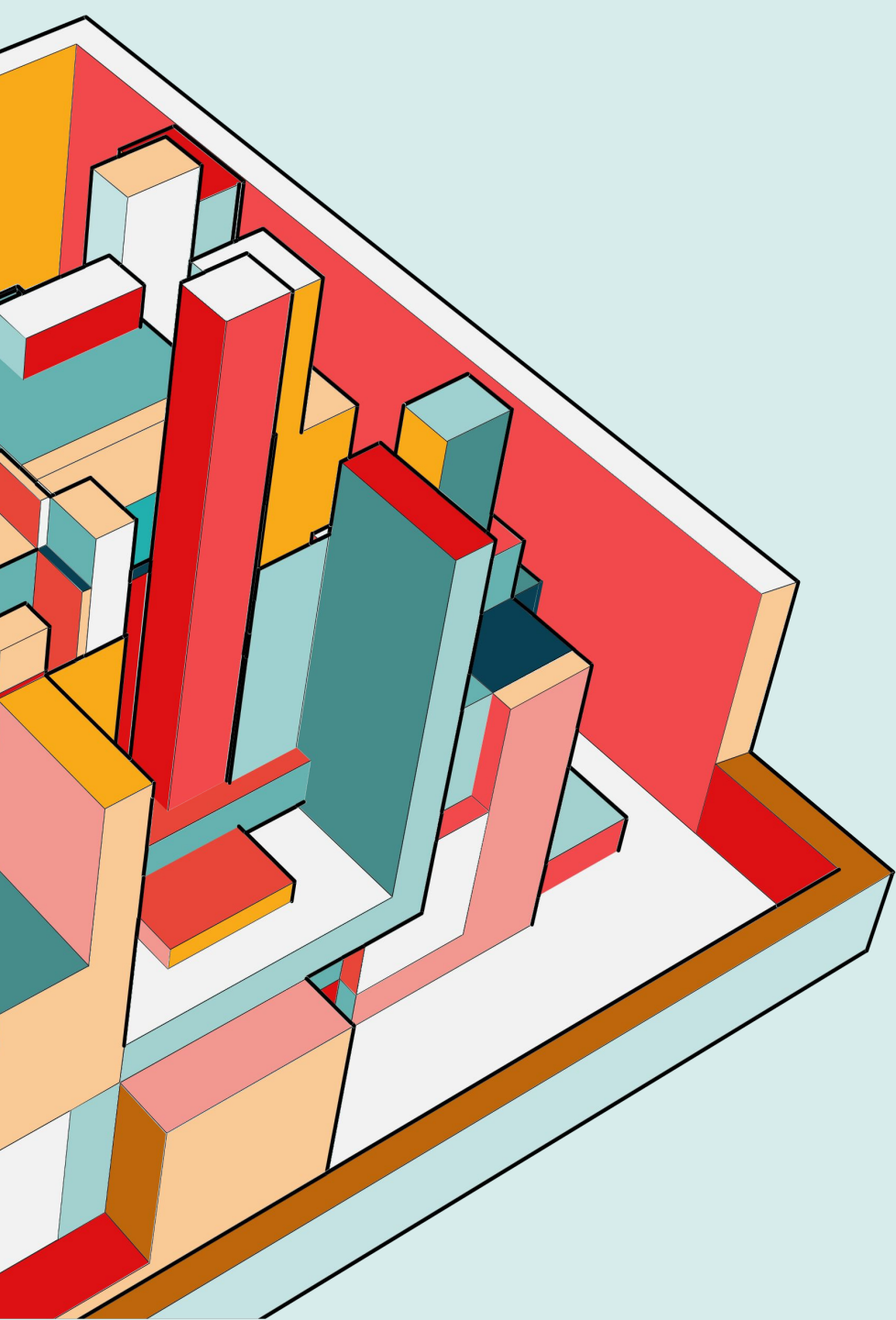
- **Padronização:** A SQL é um padrão universalmente aceito, o que significa que você pode usar seus conhecimentos em diversos sistemas de gerenciamento de bancos de dados (SGBDs) como MySQL, PostgreSQL, Oracle, SQL Server, etc.
- **Flexibilidade:** A SQL permite realizar desde consultas simples até análises complexas de dados, adaptando-se a diferentes necessidades e níveis de complexidade.
- **Eficiência:** A SQL é otimizada para lidar com grandes volumes de dados, permitindo a recuperação e o processamento de informações de forma rápida e eficiente.
- **Integridade dos dados:** A SQL oferece mecanismos para garantir a consistência e a integridade dos dados, evitando erros e inconsistências.
- **Versatilidade:** A SQL pode ser utilizada em diversas áreas, como análise de dados, desenvolvimento de aplicações web, inteligência de negócios e muito mais.



QUAIS SÃO AS PRINCIPAIS OPERAÇÕES REALIZADAS COM SQL?

- SELECT:** Recupera dados de uma ou mais tabelas.
- INSERT:** Insere novos registros em uma tabela.
- UPDATE:** Atualiza os valores de registros existentes.
- DELETE:** Remove registros de uma tabela.
- CREATE TABLE:** Cria uma nova tabela.
- ALTER TABLE:** Modifica a estrutura de uma tabela.
- DROP TABLE:** Exclui uma tabela.





BANCOS DE DADOS NOSQL

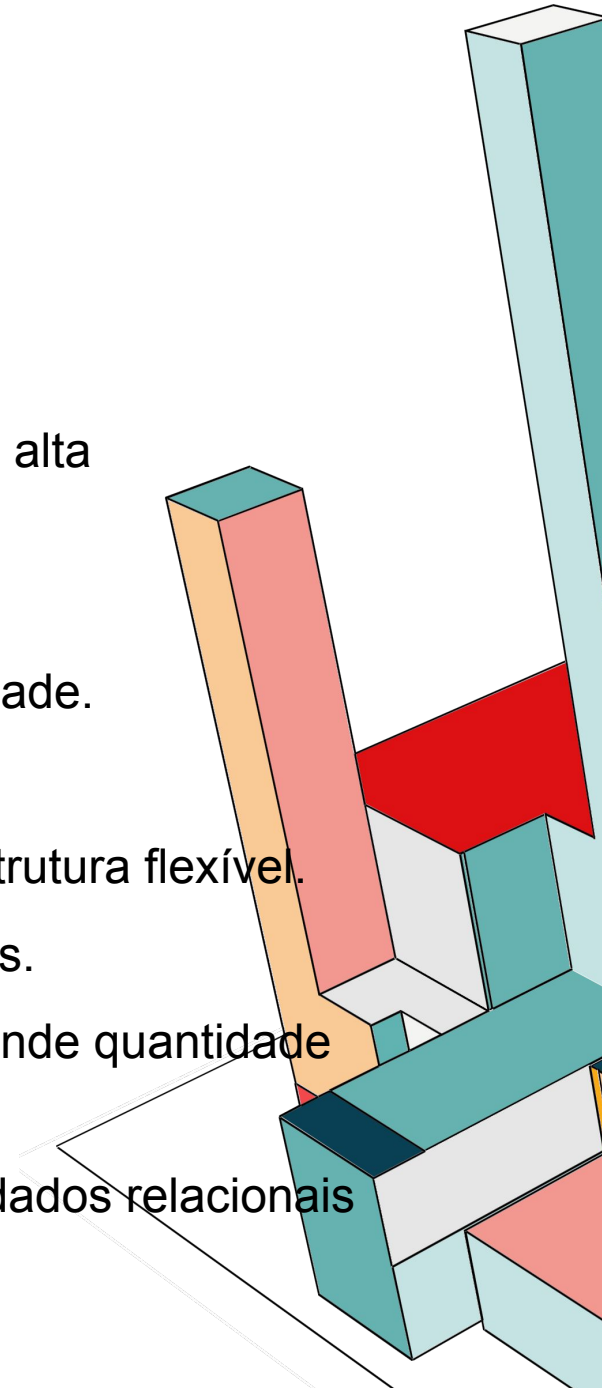
BANCOS DE DADOS NOSQL: UMA NOVA ABORDAGEM PARA GRANDES VOLUMES DE DADOS

Surgimento e Motivação:

- Limitações dos bancos relacionais para lidar com grandes volumes de dados e alta concorrência.
- Necessidade de modelos de dados mais flexíveis para aplicações modernas.
- Surgimento do NoSQL para atender às demandas de escalabilidade e flexibilidade.

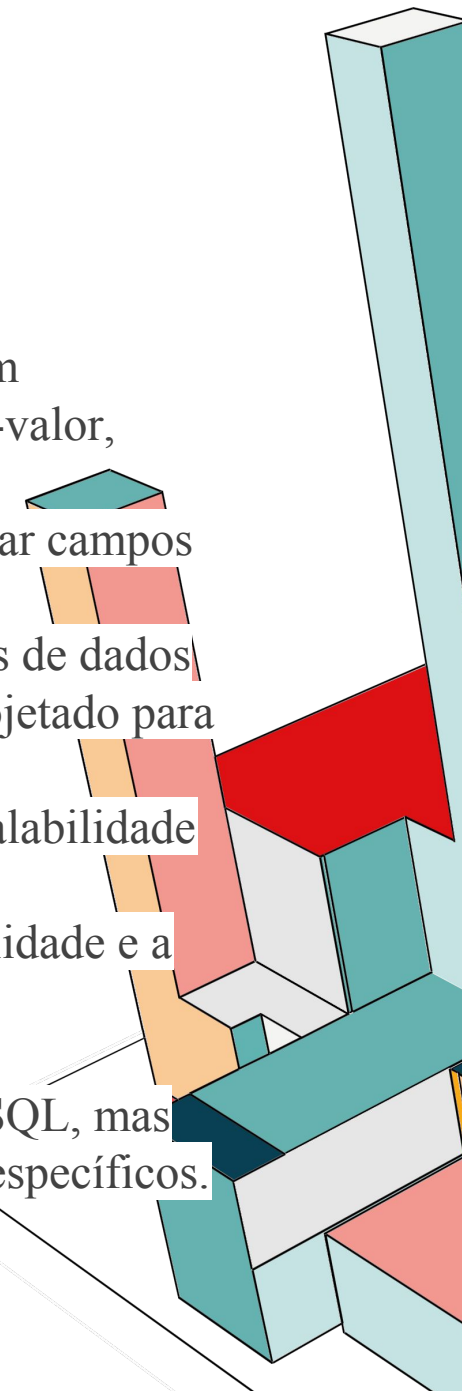
Tipos de NoSQL:

- **Document:** Armazenam dados em documentos semelhantes a JSON, com estrutura flexível.
- **Chave-Valor:** Armazenam dados em pares de chave-valor, simples e eficientes.
- **Coluna:** Organizam dados em colunas, ideais para dados analíticos e com grande quantidade de colunas esparsas.
- **Gráfico:** Representam dados como grafos, com nós e arestas, perfeitos para dados relacionais complexos.



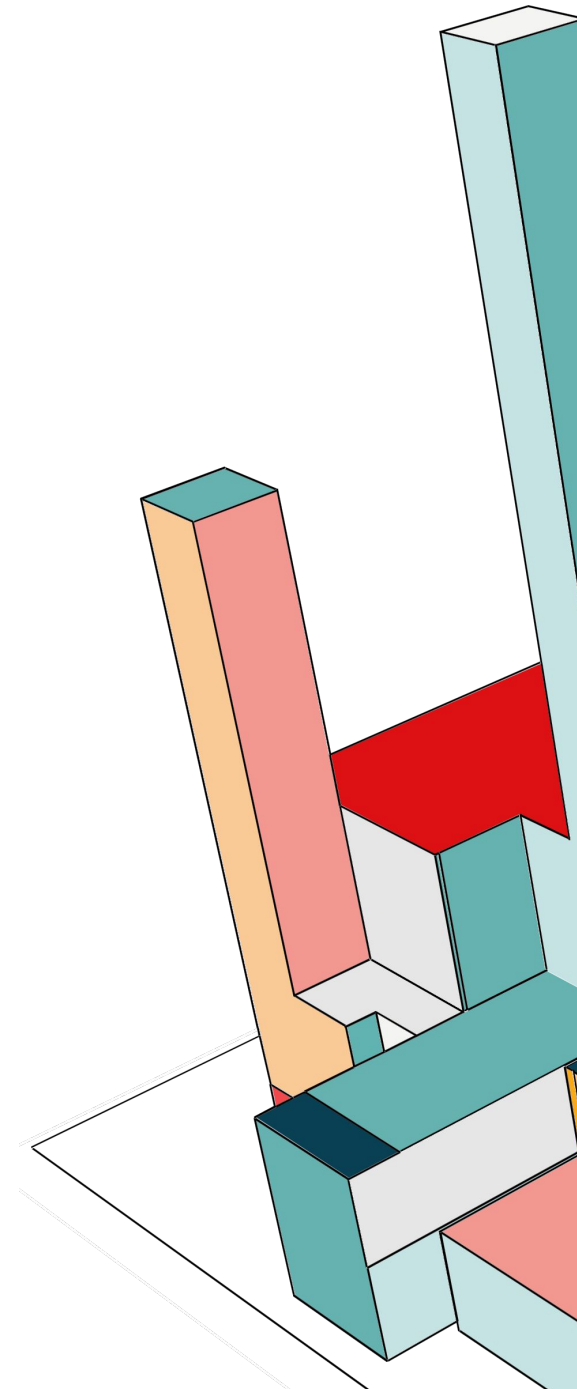
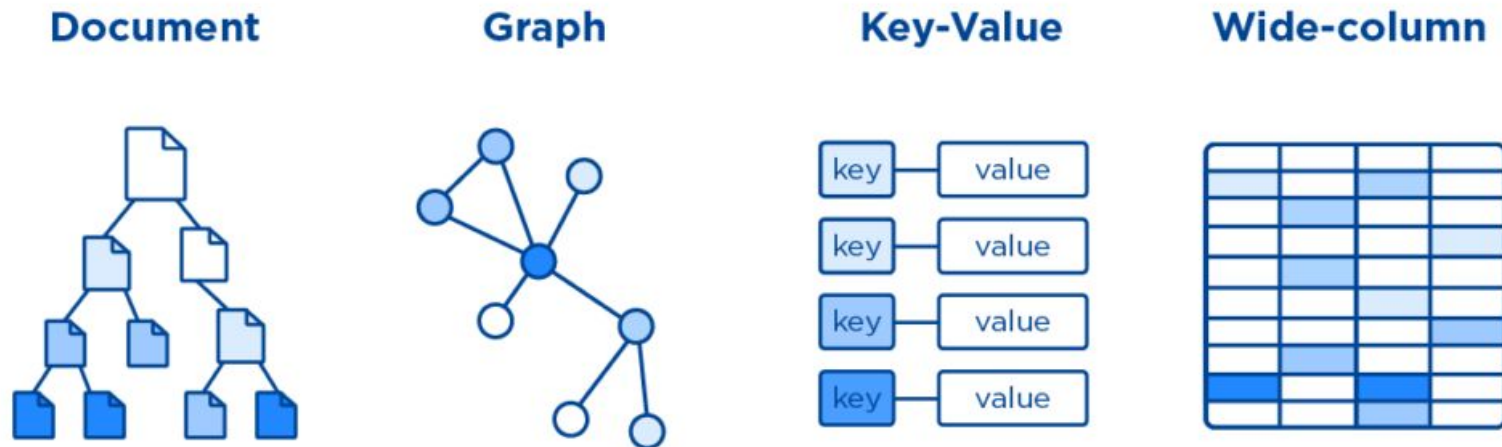
BANCOS DE DADOS NOSQL: UMA NOVA ABORDAGEM PARA GRANDES VOLUMES DE DADOS

- 1.Estrutura flexível:** Os bancos de dados NoSQL não possuem uma estrutura tabular rígida e podem armazenar dados de forma flexível, muitas vezes usando formatos como documentos, pares chave-valor, grafos ou colunas.
- 2.Eschema dinâmico:** Eles permitem um esquema dinâmico, o que significa que você pode adicionar campos aos seus documentos (ou equivalentes) sem a necessidade de alterar um esquema central.
- 3.Diversidade de modelos:** Existem várias categorias de bancos de dados NoSQL, incluindo bancos de dados de documentos, bancos de dados de colunas, bancos de dados de grafos e muito mais, cada um projetado para tipos específicos de dados e casos de uso.
- 4.Escalabilidade:** Os bancos de dados NoSQL são frequentemente usados para cenários de alta escalabilidade e distribuição, pois podem ser facilmente dimensionados horizontalmente.
- 5.Consistência flexível:** Em geral, eles não seguem o modelo ACID estrito, priorizando a disponibilidade e a partição (teorema CAP), o que pode resultar em diferentes níveis de consistência, dependendo da configuração.
- 6.Consultas variadas:** A capacidade de consulta varia de acordo com o tipo de banco de dados NoSQL, mas muitos deles oferecem recursos de consulta flexíveis que podem ser adequados para casos de uso específicos.



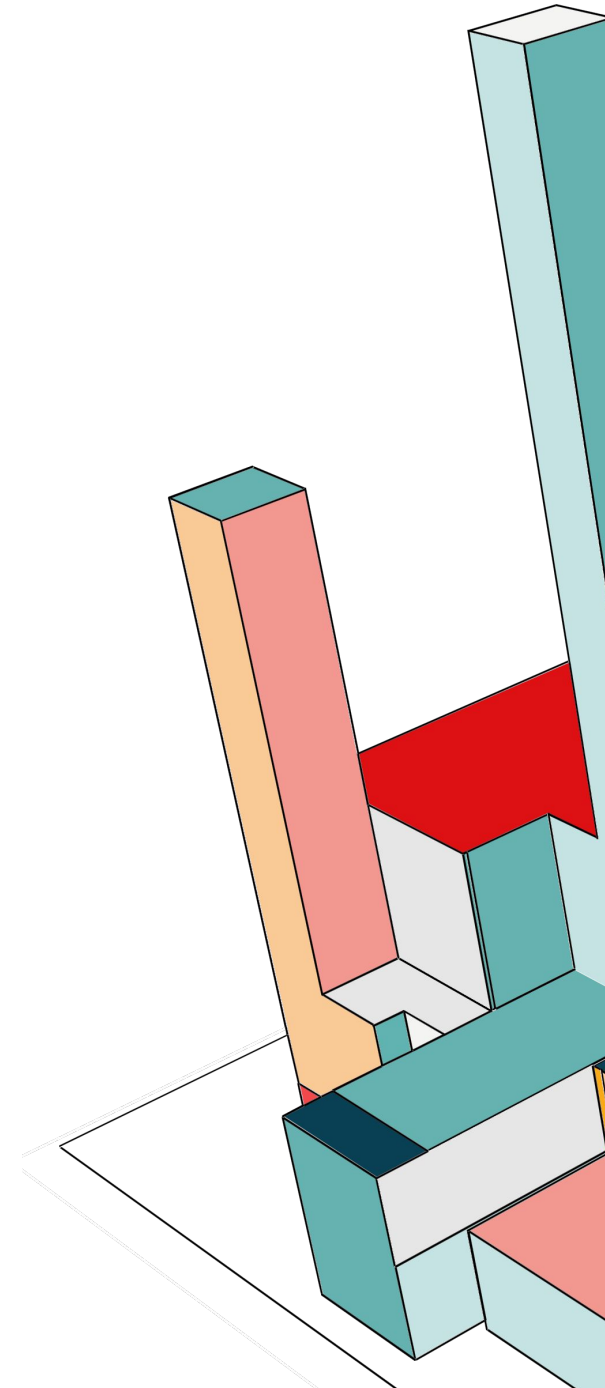
BANCOS DE DADOS NOSQL: UMA NOVA ABORDAGEM PARA GRANDES VOLUMES DE DADOS

- **Casos de Uso: Internet das Coisas (IoT):** Grandes volumes de dados gerados por sensores.
- **Análise de redes sociais:** Dados complexos e interconectados.
- **Gerenciamento de conteúdo:** Conteúdo rico e semiestruturado.
- **Aplicações móveis:** Escalabilidade e alta disponibilidade.



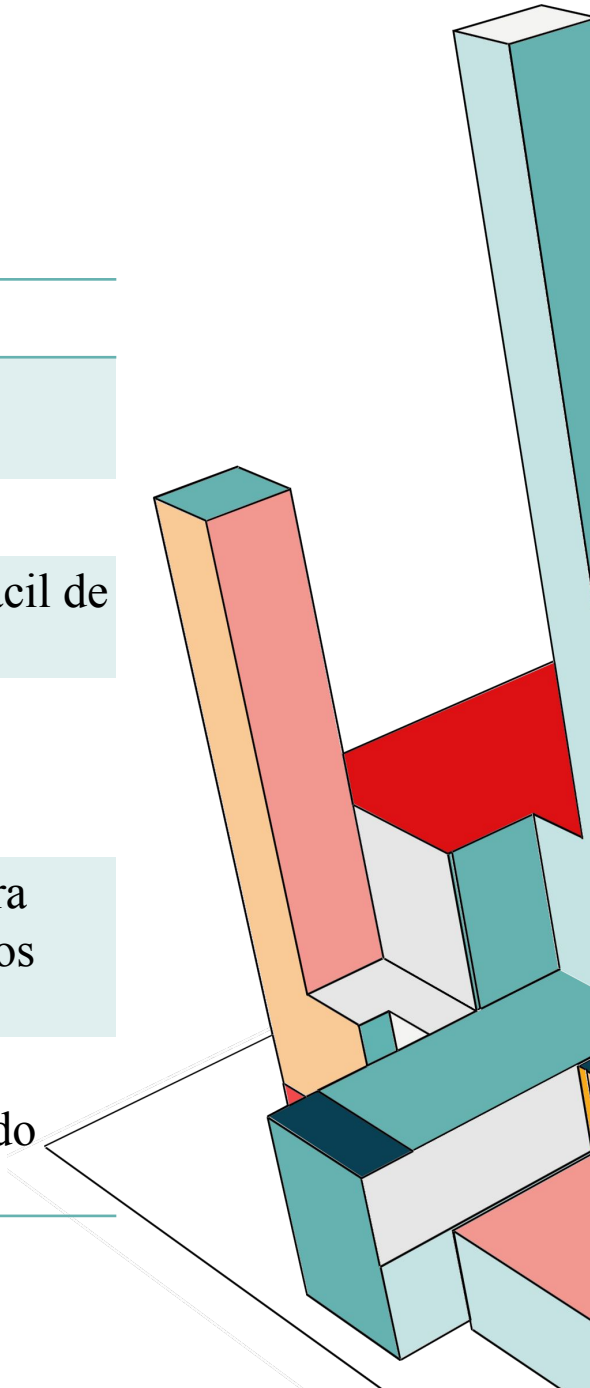
AO ESCOLHER UM BANCO DE DADOS NOSQL, É IMPORTANTE CONSIDERAR OS SEGUINTE FATORES

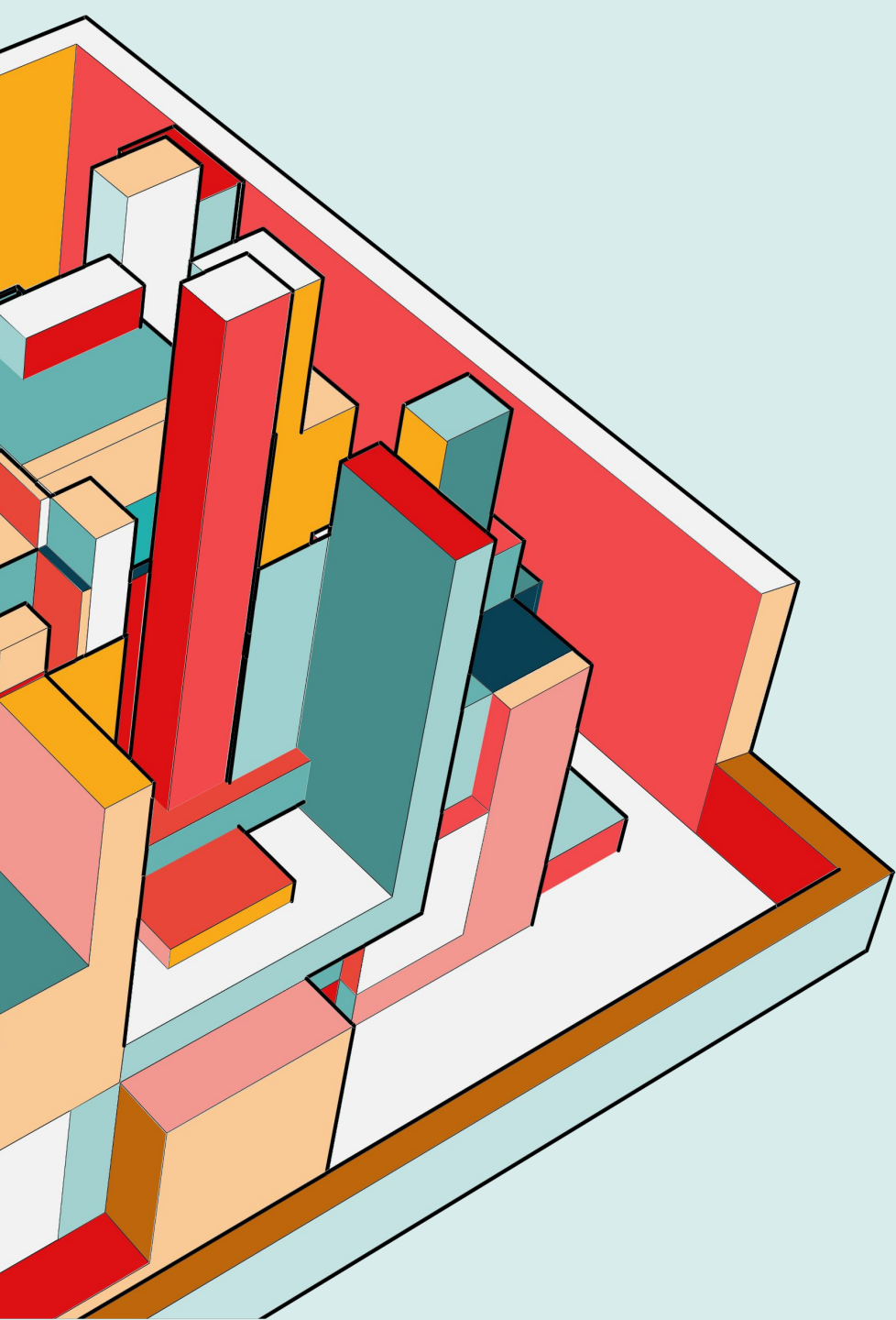
- **Tipo de dados:** Qual é a natureza dos dados que você precisa armazenar?
- **Padrões de acesso:** Como os dados serão acessados?
- **Requisitos de escalabilidade:** Quanto os dados vão crescer?
- **Tolerância a falhas:** Qual é o nível de disponibilidade necessário?



COMPARAÇÃO ENTRE BANCOS DE DADOS RELACIONAIS E NOSQL

Característica	Bancos de Dados Relacionais	Bancos de Dados NoSQL
Modelo de Dados	Tabelas com relacionamentos	Documentos, chave-valor, coluna, grafo
Esquema	Estruturado e rígido	Flexível e dinâmico
Escalabilidade	Vertical e horizontal (com limitações)	Horizontal, distribuída e fácil de escalar
Transações	ACID (Atomicidade, Consistência, Isolamento, Durabilidade)	Eventualmente consistente (muitos NoSQL)
Consultas	SQL, poderosa e expressiva	Linguagens específicas para cada tipo, geralmente menos expressivas que SQL
Casos de uso	Aplicações empresariais, transacionais, OLTP	Big data, análise, IoT, aplicações móveis, conteúdo não estruturado





OUTROS TIPOS DE BANCOS DE DADOS

ALÉM DOS RELACIONAIS E NOSQL: OUTROS MODELOS DE DADOS

Bancos de Dados em Memória:

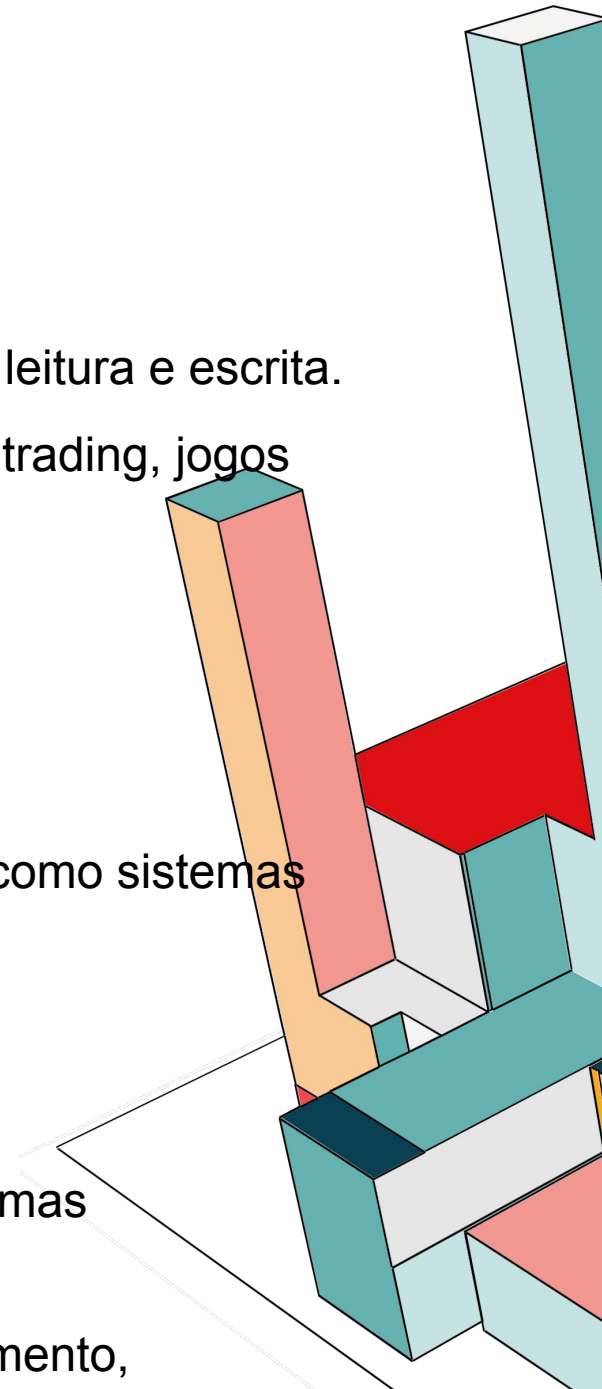
- Armazenam todos os dados na memória RAM do servidor.
- Oferecem latência extremamente baixa e alto desempenho para operações de leitura e escrita.
- Ideal para aplicações que exigem respostas em tempo real, como sistemas de trading, jogos online e análise de dados em tempo real.

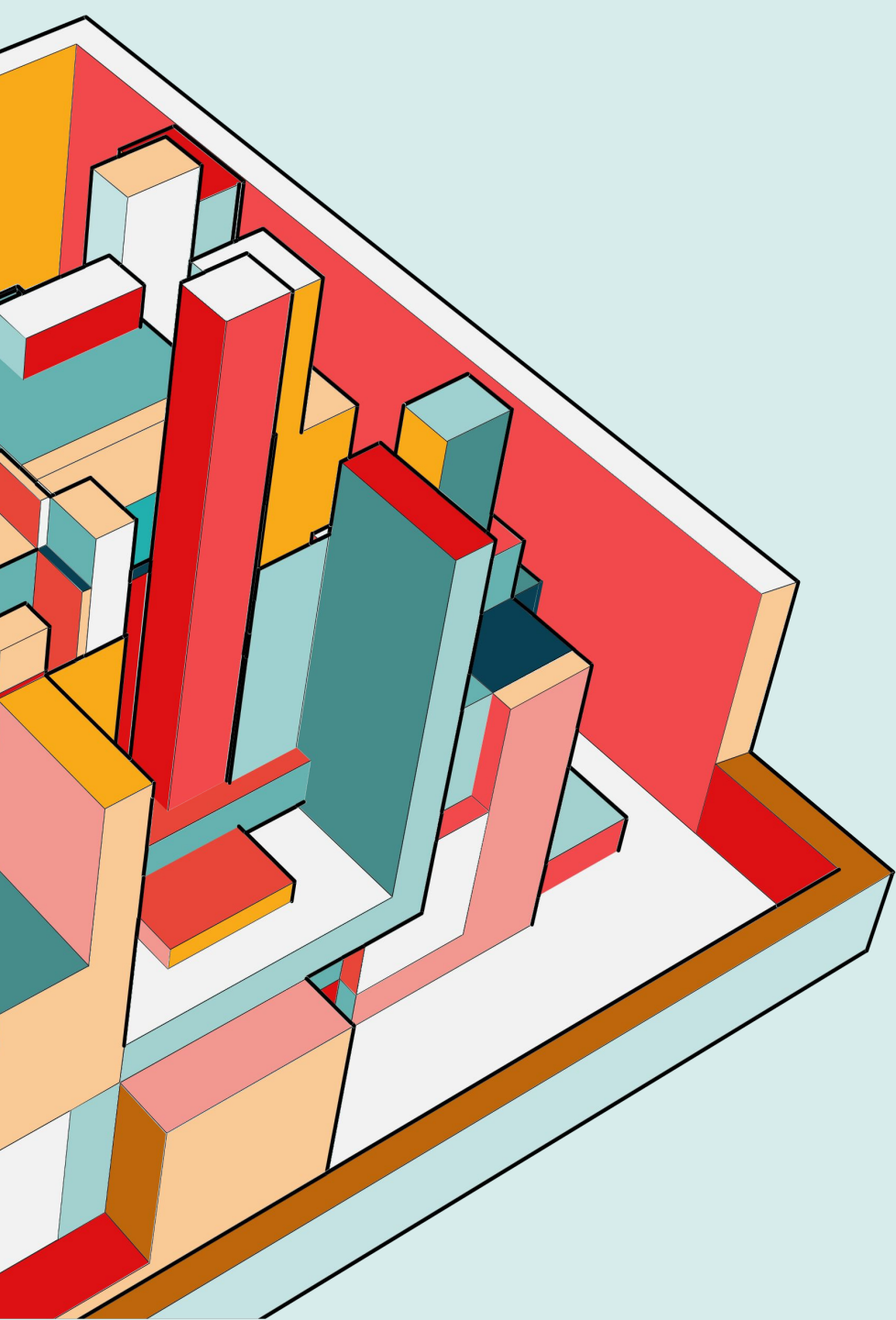
Bancos de Dados Orientados a Objetos:

- Armazenam dados como objetos, com atributos e métodos.
- Permitem modelar dados de forma mais natural e próxima ao mundo real.
- Ideal para aplicações que exigem um alto nível de abstração e complexidade, como sistemas de CAD e simulações.

Bancos de Dados Espaciais:

- Especializados em armazenar e consultar dados geográficos.
- Suportam tipos de dados espaciais como pontos, linhas, polígonos e outras formas geométricas.
- Ideal para aplicações de GIS (Geographic Information Systems), como mapeamento,





O PAPEL DO ADMINISTRADOR DE BANCO DE DADOS

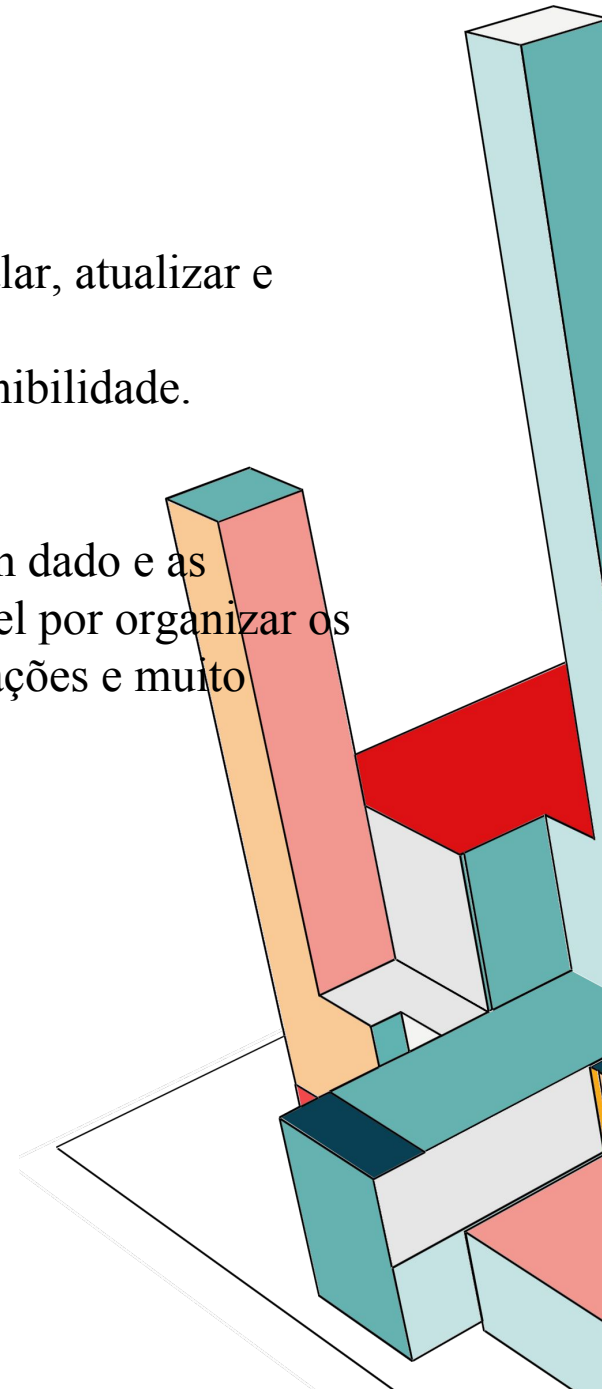
O ARQUITETO DOS DADOS: CONHEÇA O DBA

DBA (Database Administrator) é o profissional responsável por gerenciar, configurar, instalar, atualizar e monitorar bancos de dados.

- É o guardião dos dados de uma organização, garantindo sua integridade, segurança e disponibilidade.

Quem é o DBA?

Imagine um banco de dados como uma biblioteca gigantesca, onde cada livro representa um dado e as prateleiras são as tabelas. O **DBA** é o bibliotecário dessa biblioteca digital. Ele é responsável por organizar os livros, garantir que ninguém os rasgue, criar um sistema eficiente para encontrar as informações e muito mais.

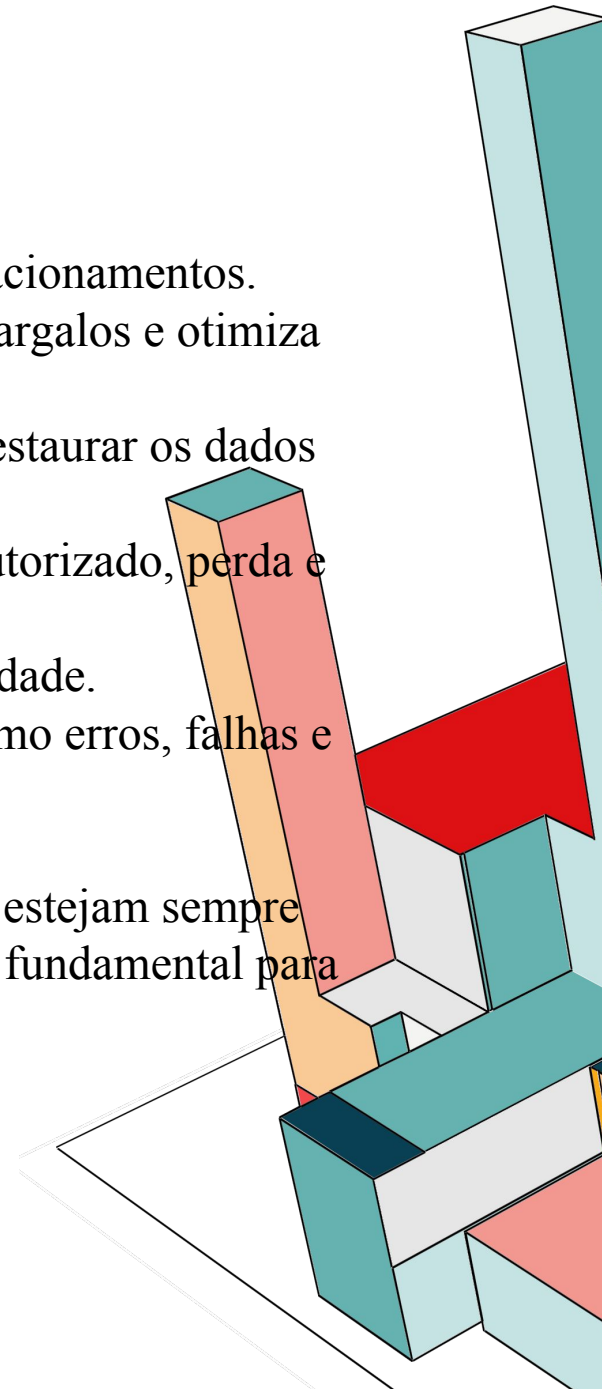


QUAIS SÃO AS RESPONSABILIDADES DE UM DBA?

- Projeto e implementação:** Cria e configura bancos de dados, definindo sua estrutura e relacionamentos.
- Gerenciamento do desempenho:** Monitora o desempenho do banco de dados, identifica gargalos e otimiza consultas para garantir que as aplicações funcionem de forma rápida e eficiente.
- Backup e recuperação:** Realiza backups regulares dos dados e garante que seja possível restaurar os dados em caso de falhas ou desastres.
- Segurança:** Implementa medidas de segurança para proteger os dados contra acesso não autorizado, perda e corrupção.
- Otimização:** Ajusta e otimiza o banco de dados para melhorar o desempenho e a escalabilidade.
- Solução de problemas:** Identifica e resolve problemas relacionados ao banco de dados, como erros, falhas e problemas de desempenho.

Por que o DBA é importante?

Os dados são um ativo valioso para qualquer organização. O DBA garante que esses dados estejam sempre disponíveis, seguros e acessíveis quando necessário. Um banco de dados bem gerenciado é fundamental para o sucesso de qualquer negócio.

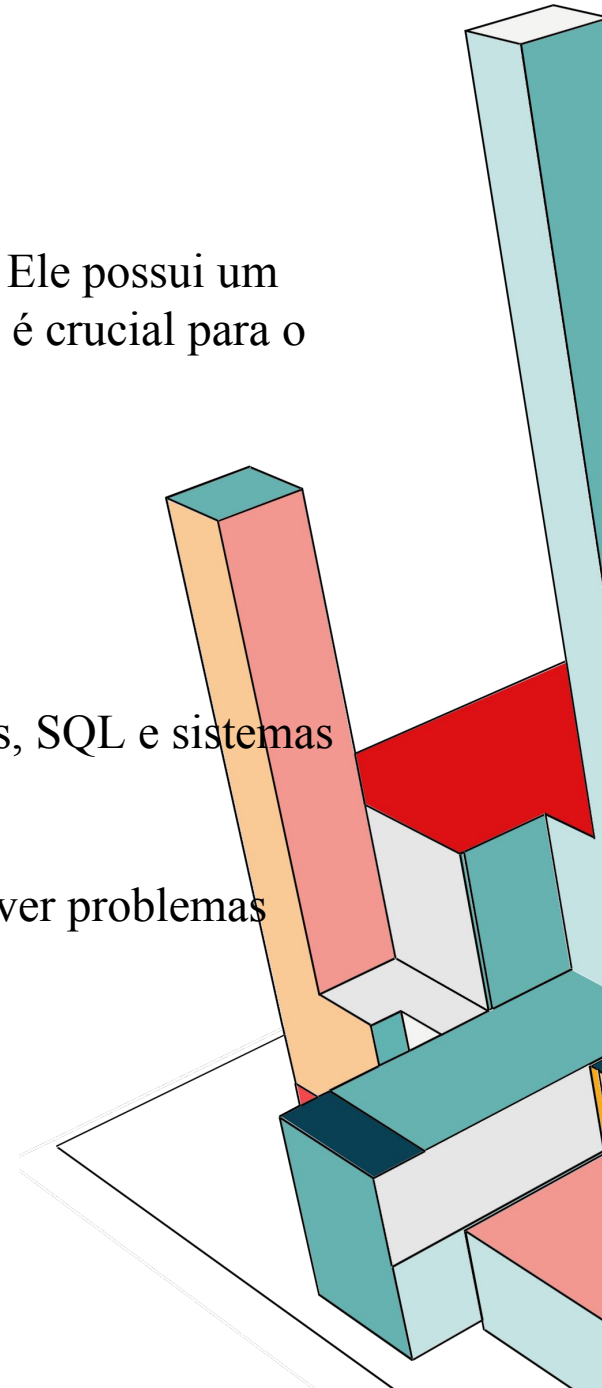


QUAIS SÃO AS RESPONSABILIDADES DE UM DBA?

O DBA é um profissional essencial para qualquer organização que utiliza bancos de dados. Ele possui um papel fundamental na garantia da integridade, segurança e disponibilidade dos dados, o que é crucial para o sucesso de qualquer negócio.

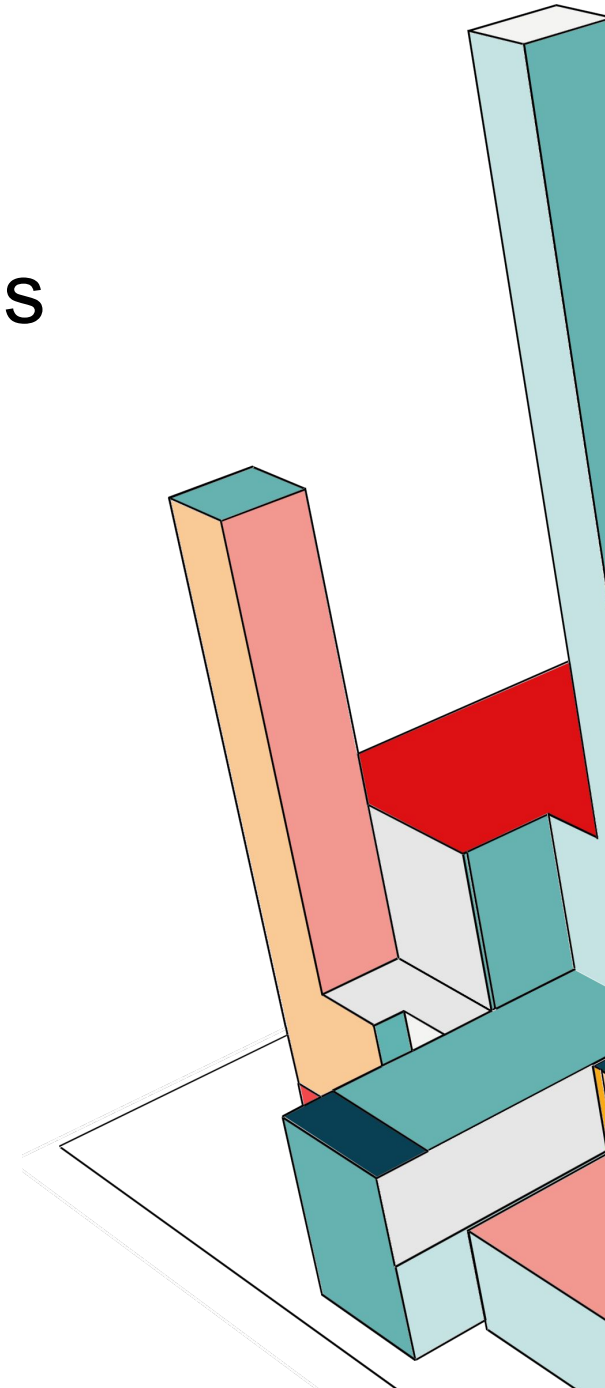
Pontos-chave a destacar:

- **Guardião dos dados:** O DBA é o responsável por proteger os dados da organização.
- **Conhecimento técnico:** O DBA precisa ter um profundo conhecimento de bancos de dados, SQL e sistemas operacionais.
- **Habilidade para resolução de problemas:** O DBA precisa ser capaz de identificar e resolver problemas complexos relacionados a bancos de dados.



RESPONSABILIDADES DO DBA

- Projeto e implementação de bancos de dados
- Garantia da performance
- Segurança dos dados
- Backup e recuperação
- Tuning de consultas



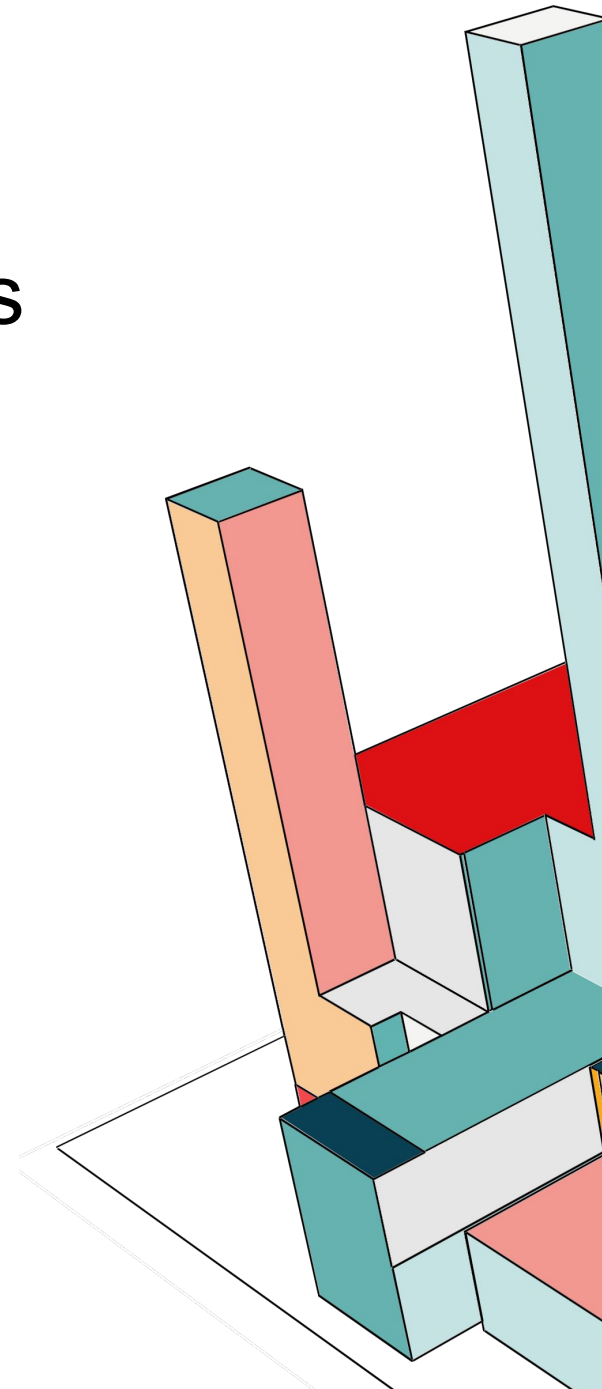
HABILIDADES DE UM DBA

Conhecimento técnico: SQL, SGBDs, sistemas operacionais

Habilidades de resolução de problemas

Comunicação

Trabalho em equipe



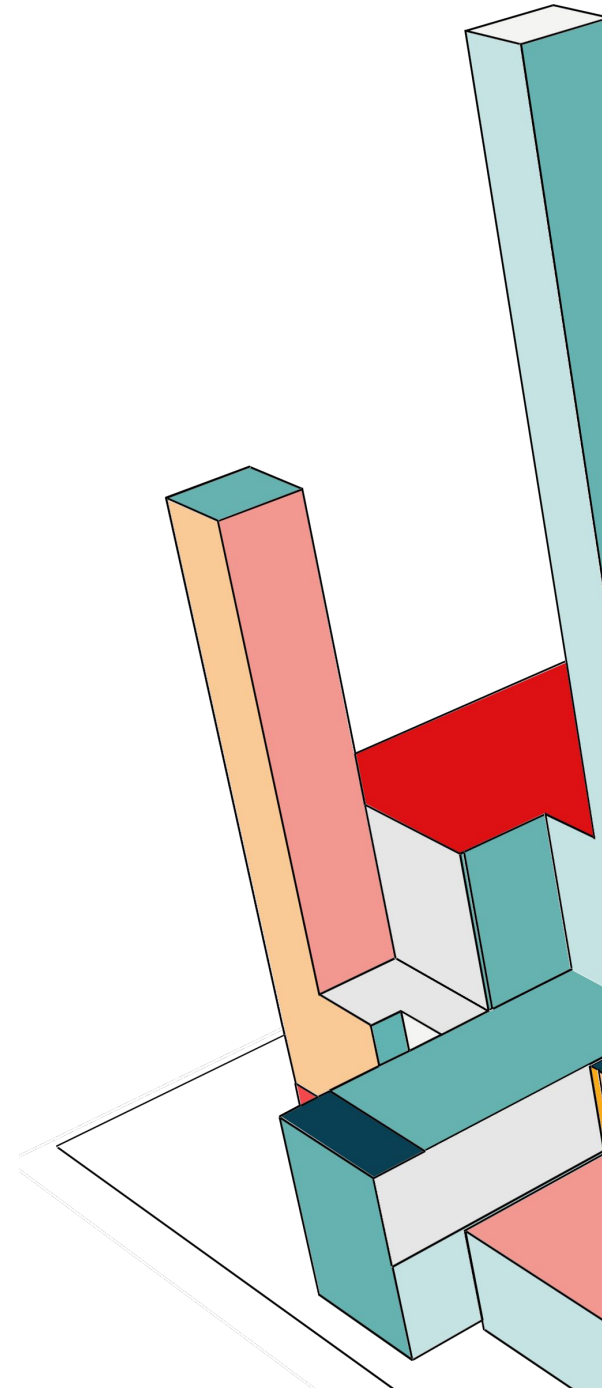
DESAFIOS DO DBA

Crescimento exponencial de dados

Segurança cibernética

Conformidade com regulamentações

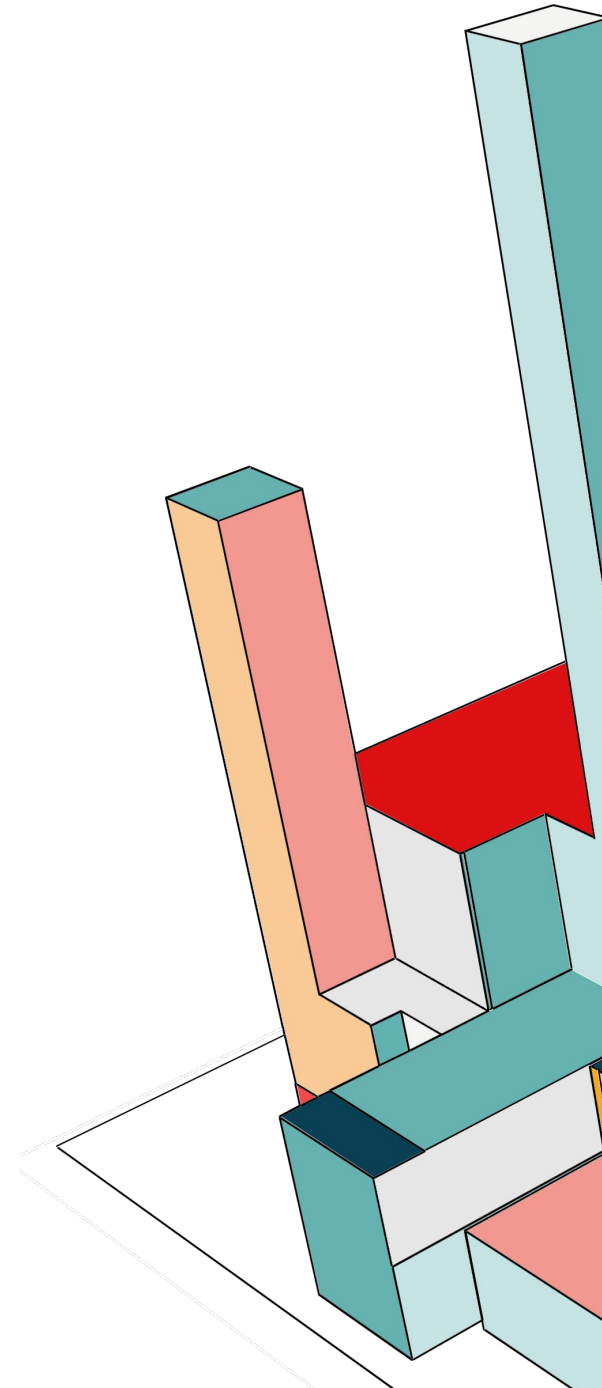
Complexidade dos sistemas



O FUTURO DO DBA

Tendências: cloud computing, big data, inteligência artificial

Novas habilidades: machine learning, DevOps



OBRIGADO

