ESTUDO MODELAGEM - Cecília S. Almeida

**Modelar** significa criar um modelo que explique as características de funcionamento e comportamento de um [software](https://pt.wikipedia.org/wiki/Software) a partir do qual ele será criado, facilitando seu entendimento e seu projeto, através das características principais que evitarão erros de [programação](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_de_computadores), projeto e funcionamento. É uma parte importante do desenho de um [sistema de informação](https://pt.wikipedia.org/wiki/Sistema_de_informa%C3%A7%C3%A3o).

Os modelos de dados são ferramentas que permitem demonstrar como serão construídas as estruturas de dados que darão suporte aos processos de negócio, como esses dados estarão organizados e quais os relacionamentos que pretendemos estabelecer entre eles.[[1]](https://pt.wikipedia.org/wiki/Modelagem_de_dados#cite_note-1)

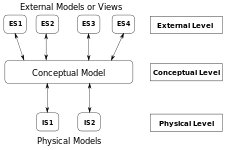
A abordagem que se dispensa ao assunto normalmente atende a três perspectivas:

* Modelagem Conceitual: é usada como representação de alto nível e considera exclusivamente o ponto de vista do usuário criador dos dados;
* Modelagem Lógica: agrega mais alguns detalhes de implementação.
* Modelagem Física: demonstra como os dados são fisicamente armazenados.

Quanto ao objetivo, podemos identificar as seguintes variações:

* modelagem de dados [entidade-relacionamento](https://pt.wikipedia.org/wiki/Modelo_de_entidade_e_relacionamento) (leitura, construção e validação dos modelos);
* modelagem de relacionamentos complexos, grupos de dados lógicos e ciclo de vida das entidades;
* modelagem de dados corporativa;
* modelagem de dados distribuídos ([cliente/servidor](https://pt.wikipedia.org/wiki/Cliente/servidor));
* modelagem e [reengenharia](https://pt.wikipedia.org/wiki/Reengenharia) de dados [legados](https://pt.wikipedia.org/wiki/Legados) e
* modelagem de dados para [Data Warehouse](https://pt.wikipedia.org/wiki/Data_Warehouse).
* O banco de dados é utilizado para guardar informações.
* O banco de dados é feito para ajudar as pessoas

## **Modelos**

****

De acordo com a abordagem que utilizam, os modelos de dados normalmente são classificados da seguinte forma:

* **Modelo Conceitual**: O modelo conceitual é um diagrama em blocos que demonstra todas as relações entre as entidades, suas especializações, seus atributos e auto-relações.
* **Modelo Lógico**: O modelo lógico mostra as ligações entre as tabelas de banco de dados, as chaves primárias, os componentes de cada uma, etc.
* **Modelo Físico**: Inclui a análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados (estrutura de armazenamento, endereçamento, acesso e alocação física), sendo uma sequência de comandos executados em [SQL](https://pt.wikipedia.org/wiki/SQL) a fim de criar as tabelas, estruturas e ligações projetadas até então e finalmente criar o banco de dados.

Exemplo de Modelo Físico (SQL ANSI 2003)

CREATE TABLE Produtos (

COD\_PROD Texto(1) PRIMARY KEY,

MODELO Texto(1),

DESCRICAO Texto(1),

COR Texto(1),

CATEGORIA Texto(1),

QUANT\_PROD Texto(1),

)

### **Modelo Lógico de Dados**

Um modelo lógico de dados para uso meramente operacional/transacional deve:

* Ser completamente [normalizado](https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados);
* Representar fielmente o NEGÓCIO, e NÃO necessariamente a base de dados desejada, a qual será construída posteriormente por ocasião do Projeto Físico;
* Conter descrição sucinta das entidades, atributos e relacionamentos;
* Conter os nomes de entidades e atributos, extensos e abreviados, atribuídos de acordo com algum padrão adotado na organização e formados por termos previamente convencionados em um [glossário](https://pt.wikipedia.org/wiki/Gloss%C3%A1rio);
* Contemplar, para cada um dos atributos, o tipo de dado, tamanho e opcionalidade.

## **Identificação de Objetos**

* **Coisas Tangíveis**: elementos que têm existência concreta, que ocupam lugar no espaço.

Ex: Meio de Transporte (avião, carro, barco, navios, etc);

* **Funções**: percepção dos objetos através da função por eles exercida (papel, atribuição, classificação, capacitação, etc).

Ex: Organização (órgãos funcionais - venda, suporte, despacho de mercadorias, etc), especialistas (médicos, engenheiros, etc), cliente (pessoa atendida), atendente (pessoa que atende), etc;

* **Eventos ou Ocorrências**: alguns objetos só conseguem ser individualizados ou percebidos enquanto uma certa ação se desenrola (identifica-se características que tornam determinado fato materializável).

Ex: vôo comercial, acidente de trânsito, jogo de futebol, etc.

* **Interações**: resultantes das associações entre objetos em função de um processo executado - cada objeto participante da interação preserva suas características não sendo impactados pela materialização da interação.

Ex: compra de um imóvel, adoção de uma criança, venda de um produto;

* **Especificações**: são elementos que definem características de outros objetos.

Ex: modelos de carro (cor, dimensões, etc), espécies animais (mamíferos, carnívoros, gatinhos. lulinhas, esponjinhas, etc.)

## **Definição**

Uma definição deve:

* ser única e especial (dentro de qualquer dicionário no qual ela aparece);
* ser estabelecida no singular;
* estabelecer o que o conceito é (não o que ele não é), o que faz, quando algum elemento passa a ser, ou deixa de ser, pertencente a esse grupo;
* ser estabelecida como uma frase ou sentença descritiva;
* ser expressa sem definições embutidas de outros termos;
* estabelecer o significado essencial do conceito;
* ser precisa e não-ambígua;
* ser concisa;
* ser significativa por si só;
* evitar raciocínio circular.

## **Atributos**

Quanto ao tipo, podem ser classificados como:

* **Descritivos**: representam as características intrínsecas dos objetos;
* **Nominativos**: além de cumprirem a função de descritivos servem como definidores de nomes ou rótulos de identificação dos objetos (nome, código, número, sigla, etc);
* **Referenciais**: representam uma citação ou ligação do objeto em questão com outro objeto, não propriamente definindo uma característica do objeto mas explicitando um relacionamento existente[[2]](https://pt.wikipedia.org/wiki/Modelagem_de_dados#cite_note-2).

Ex: Cidade de nascimento, Nome do fabricante do carro, Local de trabalho, etc.

## **Relacionamentos**

Na descrição de um relacionamento devem aparecer:

* Sua função;
* O que ele representa;
* Quais as regras de seu estabelecimento;
* Quais as exceções a seu estabelecimento;
* Quando ocorre;
* Quando pode deixar de existir.

## **Recomendações**

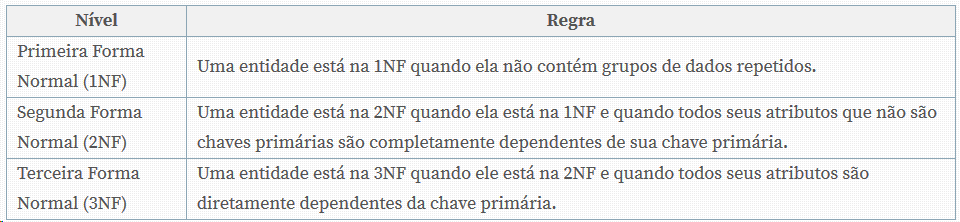
Um Modelo Lógico de Dados para uso meramente operacional/transacional **não** deve conter:

* **Replicações de atributos**: fisicamente pode ser interessante alguma redundância com o objetivo de melhorar a performance de determinado(s) processo(s). No modelo lógico isso não pode ser feito; um atributo só é representado na Entidade que o pertence.
* **Atributos derivados**: pelos mesmos motivos apontados anteriormente, a implementação das tabelas pode requerer o armazenamento de uma informação derivada de outra(s) (valor do saldo por exemplo). Tal tipo de informação não se constitui um atributo do modelo lógico.
* **Atributos repetitivos**: o uso de atributos repetidos, como Telefone-1 e Telefone-2, não é admitido. Se existe a possibilidade de uma pessoa possuir mais de um telefone, então Telefone deve ser representado como uma entidade, mantendo relacionamento Nx1 com a entidade Pessoa.

## **Normalização de Dados**

Normalização de dados é um processo no qual atributos de dados em um modelos de dados são organizados para aumentar a coesão dos tipos de entidade. Em outras palavras, o objetivo da normalização de dados é reduzir e até eliminar redundância de dados, uma questão importante para desenvolvedores, pois é incrivelmente difícil armazenar objetos em um banco de dados relacional que mantém a mesma informação em vários lugares. A tabela a seguir resume as três principais regras de normalização descrevendo como aumentar os níveis de normalização em tipos de entidade.

Com respeito à terminologia, um esquema de dados é considerado estar em um nível de normalização do seu tipo de entidade menos normalizado. Por exemplo, se todos os tipos de entidade estão na segunda forma normal (2NF) ou maior, então dizemos que o esquema de dados está na 2NF.



## **Diversificar para melhorar desempenho**

Esquemas de dados normalizados, quando colocados em produção, normalmente sofrem problemas de desempenho. Isso faz sentido – as regras de normalização focam em reduzir redundância de dados, não em melhorar desempenho do acesso aos dados. Uma parte importante da modelagem de dados é Diversificar porções do esquema de dados para melhorar tempo de acesso aos dados.

Observando que se o projeto inicial e normalização dos dados atinge o desempenho necessário para a aplicação, nada precisa ser feito. A Diversificação deve ser aplicada apenas quando os testes de desempenho mostram que temos um problema com os objetos, revelando que precisamos melhorar o tempo de acesso aos dados.

## 

## 

## 

## **Sistema de Recomendação**

Devido ao crescente aumento de volume de informação, vem se a necessidade de apresentar as melhores informações para usuários de um determinado sistema. Em um e-commerce, por exemplo, poderíamos filtrar produtos e gerar recomendações baseadas no interesse do usuário. Deste modo, não iríamos somente mostrar toda a variedade de produtos do e-commerce, mas sim o que interessa o usuário, podendo assim gerar uma conversão de compra de um determinado produto. Um outro exemplo seria em serviços de streaming, através de um sistema de recomendação poderíamos filtrar filmes e fazer com que os usuários do sistema passassem mais tempo utilizando o serviço. O objetivo dos SR é gerar recomendações válidas de itens que possam interessar aos usuários. Sugestões de livros e produtos no Amazon, filmes e séries no Netflix, amigos no Facebook, vídeos no Youtube, lugares no Foursquare e outras infinidades de recomendações.

**Filtragem Colaborativa:**

* Recomendações baseadas em todos os usuários
* Amplamente utilizados em *e-commerces* e Redes Sociais

**Filtragem Baseada em Conteúdo:**

* Recomendações baseadas no histórico do usuário.
* Utilizados, por exemplo, em anúncios patrocinados nas redes sociais.

**Similaridade de item/usuário:**

* Recomendações baseadas em itens similares aos que o usuário já adquiriu ou usuários mais similares entre si (vizinho mais próximo).
* Problema de Cold-Start.

**Método Híbrido**

**Links Úteis:**

* [**http://igti.com.br/blog/como-funcionam-os-sistemas-de-recomendacao/**](http://igti.com.br/blog/como-funcionam-os-sistemas-de-recomendacao/)
* [**https://bcc.ime.usp.br/tccs/2014/marcost/monografia\_final.pdf**](https://bcc.ime.usp.br/tccs/2014/marcost/monografia_final.pdf)
* [**https://www.udemy.com/inteligencia-artificial-sistemas-de-recomendacao-em-python/**](https://www.udemy.com/inteligencia-artificial-sistemas-de-recomendacao-em-python/)
* [**https://jaxenter.com/top-5-machine-learning-libraries-java-132091.html**](https://jaxenter.com/top-5-machine-learning-libraries-java-132091.html)

## **Estudo requisições HTTP**

HTTP é um protocolo baseado em *streams* de texto. Muito resumidamente, o cliente abre um socket para falar com o servidor e, nesse socket, envia requisições(*requests*), as quais o servidor responderá com respostas(*responses*).

O protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso. Embora esses métodos possam ser descritos como substantivos, eles também são comumente referenciados como *HTTP Verbs (Verbos HTTP)*.Dentre os mais usados estão:

**GET**

O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

**POST**

O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

**PUT**

O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

**DELETE**

O método DELETE remove um recurso específico.

Existem dois tipos de mensagens, requisições e respostas, cada uma com seu próprio formato.

### 

Exemplo de uma requisição HTTP:



As requisições consistem dos seguintes elementos:

* Um método HTTP, geralmente é um verbo como GET, POST, DELETE, PUT, etc, ou um substantivo como OPTIONS ou HEAD que define qual operação o cliente quer fazer. Tipicamente, um cliente que pegar um recurso (usando GET) ou publicar dados de um formulário HTML (usando POST), embora mais operações podem ser necessárias em outros casos.
* O caminho do recurso a ser buscado; a URL do recurso sem os elementos que são de contexto, por exemplo sem o protocolo protocol (http://), o domínio domain (aqui como developer.mozilla.org), ou a porta port TCP (aqui indicada pelo 80 que é ocultado por ser o número da porta padrão)
* A versão do protocolo HTTP.
* Cabeçalhos opcionais que contém informações adicionais para os servidores.
* Ou um corpo de dados, para alguns métodos como POST, similares aos corpos das respostas, que contém o recurso requisitado.

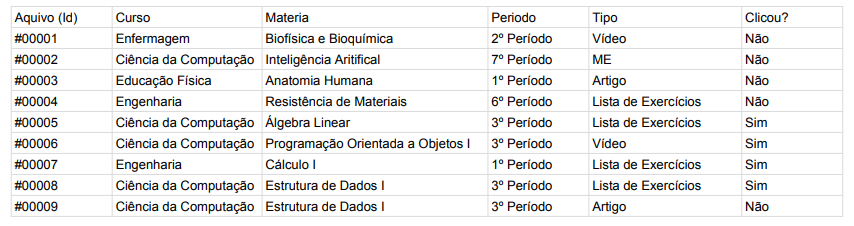
### 

### Exemplo de resposta HTTP:



Respostas consistem dos seguintes elementos:

* A versão do protocolo HTTP que elas seguem.
* Um código de status, indicando se a requisição foi bem sucedida, ou não, e por quê.
* Uma mensagem de status, uma pequena descrição informal sobre o código de status.
* Cabeçalhos HTTP, como aqueles das requisições.
* Opcionalmente, um corpo com dados do recurso requisitado

**Análise da Base e Definição do Modelo de Machine Learning**André Arthur da RochaTendo em mente o objetivo do projeto, o algoritmo de aprendizagem de máquinas que pareceu mais interessante para o sistema de recomendações foi a Árvore de Decisão. Em nossa “biblioteca”, os arquivos serão classificados por seu tipo, disciplina, curso(s) em que a disciplina é ministrada e período no qual a disciplina é normalmente alocada no curso. Numa árvore de decisão, cada aspecto de um “objeto” tem um peso diferente, uns acabam tendo mais importância que outros na decisão final, que no caso seria clicar ou não em um arquivo recomendado, com isso em mente podemos definir qual desses aspectos é mais importante para cada usuário, tendo assim uma customização individual baseada no histórico de cada usuário. Para aqueles mais interessados em listas de exercícios, mais listas de exercícios seriam recomendadas, para aqueles que clicam em mais arquivos de um determinado período, mais arquivos deste período seriam recomendados, etc.  
  
Um exemplo de como seria feita a análise e quais resultados podem ser traçados a partir deste exemplo:  
  
O usuário do exemplo teve um interesse grande em listas de exercício de matérias do terceiro período do curso de ciência da computação. Poderia inclusive ser traçado qual desses aspectos teve maior peso, e basear as recomendações seguintes nessa análise. A medida que o usuário fosse utilizando a “biblioteca”, as recomendações ficariam mais precisas, porém, seria importante não popular todas os resultados de recomendações apenas com os resultados mais precisos, pois assim o usuário poderia acabar preso numa “bolha”. Por exemplo, caso fossem exibidas nove recomendações de arquivo por vez, como no exemplo, as cinco primeiras deveriam ser recomendações precisas, enquanto as outras quatro poderiam ser recomendações mais gerais.

**Utilização de Redes Neurais de Classificação**

Podemos utilizar um modelo de rede neural utilizando o keras e tensorflow na linguem python para criar uma previsão das proximas materias e recomendar ao usuário , a partir de seu historico . Para isso podemos criar um modelo que olha os codigos das ultimas 10 materias , ou as materias mais vistas de um usuário , e criar uma previsão a partir dos perfis criados como base , Exemplo :

Exemplo Tabela de Materias :

|  |  |
| --- | --- |
| cod\_materia | nom\_materia |
| 12 | ‘Calculo 1’ |
| 13 | ‘Fisica 1’ |
| ... | ... |
| 60 | ‘Projeto 5’ |

Exemplo Historico de Usuário :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cod\_mat\_2 | cod\_mat\_2 | cod\_mat\_3 | cod\_mat\_4 | ... | **cod\_previsão** |
| 12 | 43 | 67 | 89 | ... | 112 |
| 04 | 26 | 87 | 27 | ... | 23 |
| 01 | 17 | 45 | 52 | ... | 05 |

Visualização de Exemplo de Modelo de Rede Neural:

