**tipos de dados/campos do MongoDB**

[**PRÓXIMA ATIVIDADE**](https://cursos.alura.com.br/course/nodejs-api-rest-express-mongodb/task/103638/next)

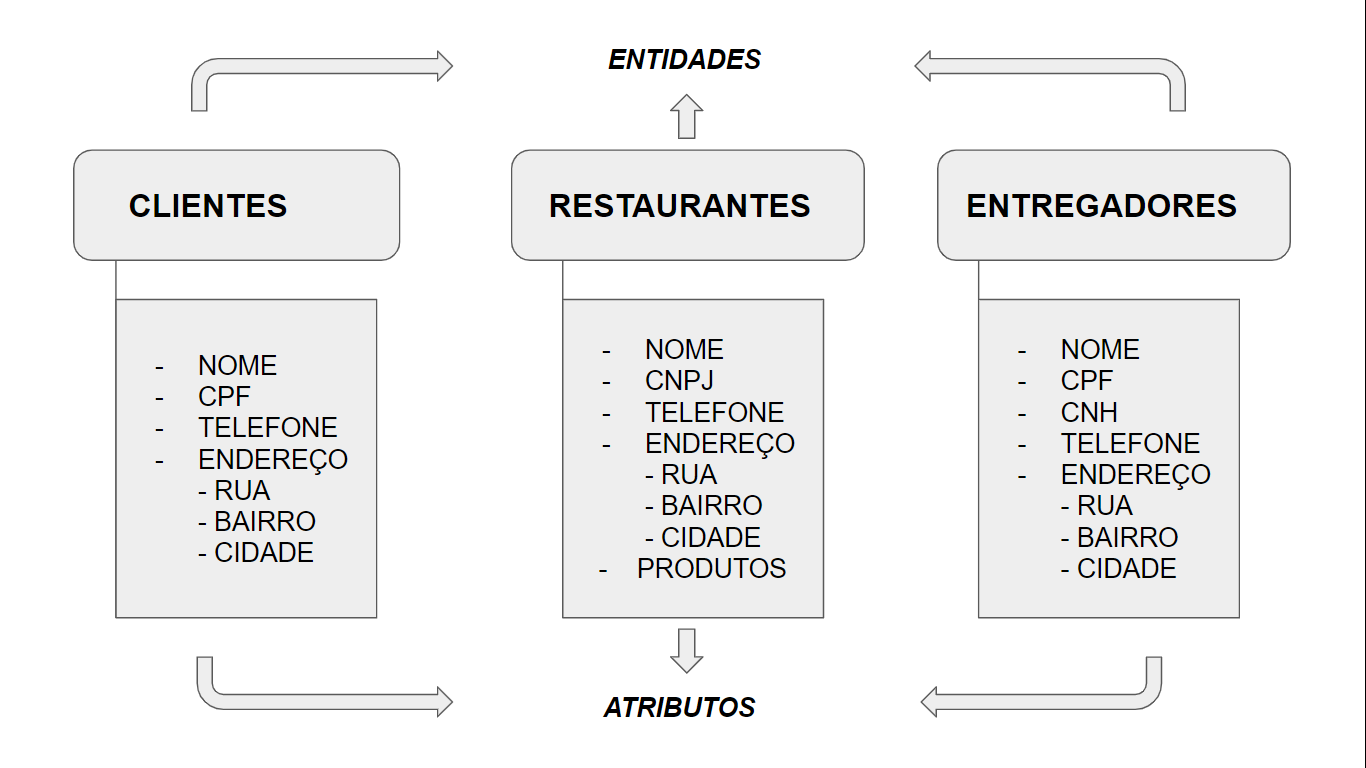
O MongoDB é um banco de dados orientado a documentos, isso significa que seus dados são armazenados dentro de documentos semelhantes a JSON, na prática usamos uma versão binária chamado BSON(Binary Json).

Na construção do JSON ou BSON, precisamos entender alguns conceitos importantes sobre sua estrutura de dados. O primeiro conceito é a definição de **Entidades e Atributos**.

Imagina uma plataforma de entrega de refeições, ela usa um aplicativo para conectar clientes com restaurantes e entregadores, cada público desses serão usuários que precisarão ter um cadastro no banco de dados da plataforma. Podemos dizer que os clientes, restaurantes e entregadores, são as **Entidades**.

Cada entidade terá um cadastro a ser preenchido com dados importantes. Mas quais dados são importantes para solicitarmos? Esses dados requeridos, chamamos de **Atributos**.

Vamos para um exemplo prático:



A vantagem de usarmos dados orientados a objetos, é que conseguimos criar subdocumentos com identificações individualizadas, sem que outros dados sejam alterados.

**Tipos de dados**

O MongoDB suporta uma variada lista de tipos de dados para os documentos no formato binário BSON, você pode ter acesso a mais tipos clicando no [link da documentação oficial BSON Types](https://docs.mongodb.com/manual/reference/bson-types/).

Vamos conhecer alguns:

* **String:** É o tipo que contém uma sequência de texto;
* **Integer (32 bit e 64-bit):** É o tipo para armazenar um valor numérico;
* **Boolean:** Usamos para armazenar valores True ou False;
* **Double:** Usamos para armazenar valores de ponto flutuantes;
* **Min/Max Keys:** É o tipo usado para comparar um valor contra o menor e maior elementos do BSON mais elevados respectivamente;
* **Array:** É usado para armazenar matrizes ou lista de valores múltiplos em uma única chave;
* **TimeStamp:** Este tipo é utilizado quando armazenamos uma data e hora;
* **Object:** Usamos para armazenar documentos incorporados;
* **Null:** É tipo para valores nulos;
* **Symbol:** É o tipo para valor string, que utiliza algum caractere ou símbolo especial, geralmente de outros idiomas;
* **Date:** Usamos para armazenar a data ou a hora atual;
* **ObjectID:** É o tipo que armazena o identificador do documento (ID);
* **Binary Data:** Usamos para armazenar um conjunto de dados binários;
* **Regular Expression:** Utilizado para expressões regulares, representadas por caracteres específicos previsto em ordem alfabética;
* **Code:** Utilizado para armazenar códigos em JavaScript dentro do documento.