

# CRUD

- **CRUD (Create, Read, Update e Delete)** são operações em Desenvolvimento de Software de aplicativos que interagem com Banco de Dados. Elas consistem em cruzar o aplicativo de ponta a ponta, desde o Front End até o Back End, enviando, recebendo e manipulando os dados.
- O **CRUD** é essencial para o Desenvolvimento de Software e para o Desenvolvimento Web, ainda mais complexo, pois essa comunicação de ponta a ponta também envolve segurança de dados e métodos que evitam que dados sensíveis sejam expostos. como é o caso do **JWT (JSON Web Token)** que manipula ações administrativas através de tokens para maior segurança de uma aplicação.

## Exemplos práticos do dia a dia.

### Create

- **Create** é uma operação de criação de um registro em uma aplicação, em uma loja online, por exemplo, ao criar uma conta na loja para fazer compras, o método **Create** é usado após o preenchimento dos dados de cadastro de um formulário, seu envio é pela operação **Create**.

### Read

- **Read** é uma operação de leitura de um registro em uma aplicação, em uma loja online por exemplo, ao acessar a página de produtos, é solicitado a busca dos produtos que são listados no lado do cliente através do **Read**.

### Update

- **Update** é uma operação de edição/alteração de um registro em uma aplicação, um exemplo é quando, nessa mesma loja online, um produto entra ou sai de uma promoção. Seu valor é alterado pelo administrador do ecommerce, e o produto em específico é alterado no sistema, logo, o usuário verá que o produto entrou ou saiu da promoção.

## Delete

- **Delete** é uma operação de deleção de algum registro em uma aplicação, quando um coturno por exemplo, fica fora de estoque, e a loja decide não trabalhar mais com ele pois as vendas não foram boas ou teve muita reclamação, o administrador do sistema então executa a operação **CRUD** delete e remove esse item do sistema.

## Exemplos de operações CRUD.

- Utilizarei aqui exemplos de uma **lista de tarefas** para ilustrar o aplicativo feito em Javascript e simplificar e facilitar o entendimento. Os exemplos são em **Javascript (Node)**.

### Create:

- **Método POST:** Adicionar uma tarefa
- Adicionei a Task "Fazer uma xícara de café"

```
// Cria nova tarefa
route.post('/tasks', (req, res) => {
  const newTask = req.body
  tasks.push(newTask)
  res.json({ message: 'Tarefa criada com sucesso!' })
})
```

## Read:

- **Método GET:** Busca as tarefas existentes
- Ao entrar no aplicativo, foi me exibido todas as tarefas existentes

```
// Busca as tarefas existentes
route.get('/tasks', (req, res) => {
  res.json(tasks)
})
```

## Update:

- **Método PUT:** Atualiza um tarefa existente (recebe um parâmetro "id" do item que iremos atualizar)
- Estava muito calor, editei a tarefa "Fazer uma xícara de café" para "Fazer um suco de laranja"

```
// Edita uma tarefa existente
route.put('/tasks/:id', (req, res) => {
  const taskId = req.params.id
  const updatedTask = req.body

  tasks = tasks.map(task => {
    if (task.id === taskId) {
      return updatedTask
    }

    return task
  })

  res.json({ message: 'Tarefa atualizada com sucesso!' })
})
```

## Delete:

- **Método DELETE:** Deleta uma tarefa existente (recebe também um parâmetro (id) da tarefa a ser deletada)
- Deletei a Task "Fazer um suco de laranja" pois decidi comprar um refrigerante.

```
// Deleta uma tarefa existente
route.delete('/tasks/:id', (req, res) => {
  const taskId = req.params.id

  tasks = tasks.filter(task => task.id !== taskId)

  res.json({ message: 'Tarefa excluída com sucesso!' })
})
```