

## Exercício 2

1. Crie uma conta gratuita no MongoDB (<https://www.mongodb.com/online>) e implemente o Modelo Não Relacional do Mercado Livre (EX1) no MongoDB
  - a.  Crie um database,  Crie as coleções,  Insira um conjunto de documentos

<p><b>Vendedor:</b></p> <pre><code>_id: ObjectId("62288eecd622f375197120") nome: "Diogo Branquinho" cpf: 12345678912 email: "diogo@gmail.com" endereco: "FATEC Profº Jessen Vidal"</code></pre>	<p><b>Produto:</b></p> <pre><code>_id: ObjectId("6228900cd622f375197123") nome: "Panela de pressão" descricao: "Panela que exerce pressão sobre a comida" preco: 123.45 vendedor: Object   _id: ObjectId("62288eecd622f375197120")   nome: "Diogo Branquinho"</code></pre>
<p><b>Usuário:</b></p> <pre><code>_id: ObjectId("6228912cd622f375197127") nome: "Maby" cpf: 12345678912 email: "maby@gmail.com" endereco: "FATEC Profº Jessen Vidal" carrinho: Array   0: Object     id: ObjectId("6228900cd622f375197123")     nome: "Panela de pressão"     descricao: "Panela que exerce pressão sobre a comida"     preco: 123.45     vendedor: Object       _id: ObjectId("62288eecd622f375197120")       nome: "Diogo Branquinho"   1: Object     id: ObjectId("6228911dd622f375197126")     nome: "Frigideira antiaderente Polishop"     descricao: "Panela que não deixa grudar"     preco: 678.9     vendedor: Object       _id: ObjectId("62288eecd622f375197120")       nome: "Diogo Branquinho"</code></pre>	<p><b>Compra:</b></p> <pre><code>_id: ObjectId("6228922fd622f375197128") usuario: Object   _id: ObjectId("6228912cd622f375197127")   nome: "Maby"   endereco: "FATEC Profº Jessen Vidal" produto: Object   _id: ObjectId("6228900cd622f375197123")   nome: "Panela de pressão"   descricao: "Panela que exerce pressão sobre a comida"   preco: 123.45 vendedor: Object   _id: ObjectId("62288eecd622f375197120")   nome: "Diogo Branquinho"   endereco: "FATEC Profº Jessen Vidal" total: 123.45 pagamento: "À vista"</code></pre>

- b. Pontos fortes e fracos de cada tipo de banco para esse problema

**Resposta:** Para o Mercado Livre, uma solução híbrida ou até 100% não relacional faz muito sentido se levarmos em conta sua quantidade de dados, de acessos e operações que devem ser eficientes e rápidas, agilizando todo o processo de compra e entrega como um todo, impactando positivamente na experiência do usuário fornecida pelo Mercado Livre. Dito isso, pontos fortes da escolha do NoSQL para este problema é que possibilita rápidas ações, mais flexibilidade quanto aos dados e um cadastro de histórico de compras incrível, enquanto que a solução com banco relacional se torna algo muito robusto, porém com mais rigidez e complexidade devido aos relacionamentos e dependências entre os dados correlacionados, o que não é interessante caso seja necessário deletar produtos ou vendedores que estão registrados em milhares de vendas (seria uma tarefa difícil, visto que as *constraints* do banco bloqueariam essa ação).

## 2. Implemente o Modelo Não Relacional da Biblioteca (EX1) no MongoDB

- a.  Crie um database,  Crie as coleções,  Insira um conjunto de documentos

<p><b>Autor:</b></p> <pre> _id: ObjectId("6229dde30b8d13ad5b9d19f7") nome: "Kiera Cass" data_nascimento: 1981-05-19T14:24:00.000+00:00 pais: "EUA" estado: "Carolina do Sul" </pre>	<p><b>Livro:</b></p> <pre> _id: ObjectId("6229d5620b8d13ad5b9d19f0") titulo: "A Seleção" sinopse: "Para trinta e cinco garotas, a Seleção é a chance de uma vida. É a opo..." autor: Object   id: ObjectId("6229dde30b8d13ad5b9d19f7")   nome: "Kiera Cass" data_publicacao: 2012-09-17T03:00:00.000+00:00 editora: "Seguinte" edicoes: Array   0: Object     edicao: "padrão"     qtd: 15 usuarios: Array   0: Object     id: ObjectId("6229e0dc0b8d13ad5b9d19fb")     nome: "Maby Reis" </pre>
<p><b>Usuário:</b></p> <pre> _id: ObjectId("6229e0dc0b8d13ad5b9d19fb") nome: "Maby" cpf: 12345678912 email: "maby@gmail.com" telefone: 12345678901 endereco: Array   0: Object     estado: "sp"     cidade: "São José dos Campos"     rua: "Rua Caparao"     cep: 12345678     numero: 123     principal: true   1: Object     estado: "RJ"     cidade: "Rio de Janeiro"     rua: "Av. Pasteur"     cep: 12345678     numero: 321     complemento: "Segundo andar, sala 4" emprestimos: Array   0: Object     livro: "A Seleção"     data_retirada: "12/10/2021" </pre>	<p><b>Empréstimos ativos:</b></p> <pre> _id: ObjectId("623317766d145aefee33f5df") emprestimos: Array   0: Object     livro: Object       _id: ObjectId("6229d5620b8d13ad5b9d19f0")       titulo: "A Seleção"       autor: "Kiera Cass"     usuario: Object       _id: ObjectId("6229e0dc0b8d13ad5b9d19fb")       nome: "Maby Reis"       data_retirada: "12/10/2021"   1: Object     livro: Object       _id: ObjectId("6229e2c40b8d13ad5b9d19fd")       titulo: "A Elite"       autor: "Kiera Cass"     usuario: Object       _id: ObjectId("6229e0dc0b8d13ad5b9d19fb")       nome: "Maby Reis"       data_retirada: "15/10/2021" </pre>

- b. Pontos fortes e fracos de cada tipo de banco para esse problema

**Resposta:** Para um sistema de uma biblioteca simples, que não necessariamente requer extrema agilidade em seus processos, muito menos muitos acessos que precisam de resposta imediata, uma solução 100% relacional traz o conforto da organização dos dados, evitando repetições, e tendo alto controle de todas as regras do negócio, além de encaixar muito bem no problema. Uma solução com NoSQL também poderia ser aplicada, com seus pontos fortes para cadastros, criação de históricos e uma busca que traria bastante velocidade para as ações no banco, porém, ao meu ver, seria um leve exagero, já que com uma solução relativamente simples e bem consumada/difundida é possível trazer ótimos resultados, além de que qualquer alteração em funcionalidades ou adição de novas, é só questão de incrementar no banco, não é preciso repensar nada desde o começo ou ter muitas preocupações quanto a oferta dos dados (um benefício dos relacionamentos).