



**Universidade
Federal do Ceará
Campus Quixadá**



Projeto de Interfaces WEB
Atividade 7
MongoDB

Michel de Melo Vieira
389959
Prof. Victor Farias

1. Popule a base de dados

A) Adicione 4 usuários à sua coleção **usuários** com nome “pedro”, “alice”, “maria” e “joao”. Os outros atributos são de sua escolha

```
use usuarios
db.usuarios.insert({nome: "Pedro", email: "pedro@mail.com", senha: "lua123", idade: 20})
db.usuarios.insert({nome: "Alice", email: "alice@mail.com", senha: "sol123", idade: 21})
db.usuarios.insert({nome: "Maria", email: "maria@mail.com", senha: "escorpiao5", idade: 22})
db.usuarios.insert({nome: "Joao", email: "joao@mail.com", senha: "poker449", idade: 20})
```

```
var idpedro = db.usuarios.findOne({nome: "Pedro"});
var idalice = db.usuarios.findOne({nome: "Alice"});
var idmaria = db.usuarios.findOne({nome: "Maria"});
var idjoao = db.usuarios.findOne({nome: "Joao"});
```

B) Adicione 3 posts para Pedro e 2 posts para Maria. Atributos texto e likes são de sua escolha e cid começa vazio ([])

```
> var p1p = {texto: "Bom Dia Minha gente!", likes: 8, uid: idpedro._id, cid: []};
> var p2p = {texto: "Iniciando mais uma nova etapa #NewJob", likes: 13, uid: idpedro._id, cid: []};
> var p3p = {texto: "Confiança é como uma borracha... Fica menor a cada erro cometido...", likes: 20, uid: idpedro._id, cid: []};
> db.post.insert(p1p);
> db.post.insert(p2p);
> db.post.insert(p3p);
> var p1m = {texto: "Talvez por serem tão curtas as férias são intensas, mas deixam também um sabor amargo.", likes: 30, uid: idmaria._id, cid: []};
> var p2m = {texto: "Pendura a alma no varal e deixe que as coisas ruins evaporem.", likes: 25, uid: idmaria._id, cid: []};
> db.post.insert(p1m);
> db.post.insert(p2m);
```

C) Adicione 2 comentários para um post de Pedro e mais 1 comentário para um post de Maria.

```
> var pp1 = db.post.findOne({texto: p2p.texto});  
  
//pp1 representa o segundo post de pedro já que p2p não possui o id da mesma  
  
> var c1p = {texto: "Parabéns meu amigo você merece", pid: pp1._id, uid: idalice._id};  
> var c2p = {texto: "Parabéns Irmão.", pid: pp1._id, uid: idjoao._id};  
> db.comentario.insert(c1p);  
> db.comentario.insert(c2p);  
  
//pm1 representa o primeiro post de Maria já que p1m não possui o id do mesmo  
  
> var pm1 = db.post.findOne({texto: p1m.texto});  
> var c1m = {texto: "Os pequenos detalhes são sempre os mais importantes.", pid:pm1._id, uid: idalice._id};  
> db.comentario.insert(c1m);
```

2. Elabore as seguintes consultas

usando o mínimo possível de chamadas a find()

```
//Atualizar tabela Post inserindo os ids dos comentarios  
  
//Comentario 1 e 2 do post 1 de Pedro  
  
//Atualizando UID do post de Pedro  
  
> var c1pp = db.comentario.findOne({uid: idalice._id});  
> var c2pp = db.comentario.findOne({uid: idjoao._id});  
> pp1.cid = [c1pp._id,c2pp._id];  
> var criterio = {_id: pp1._id};  
> db.post.update(criterio, pp1);  
  
//Atualizando UID do post bde Maria  
  
> var c1pm = db.comentario.findOne({uid: idalice._id});  
> pm1.cid = [c1pm._id];  
> var criterio = {_id: pm1._id};  
> db.post.update(criterio,pm1);
```

A) Obtenha todos os comentários de um dado post

```
> db.comentario.find({pid: pp1._id});
```

B) Obtenha todos os posts que tenham mais de 10 likes

```
> db.post.find({likes: {"$gt": 10}});
```

C) Obtenha todos os posts que não tem comentários

```
> db.post.find({cid: {"$eq": []}});
```

D). Obtenha todos os posts que pelo menos 2 comentários

```
> db.post.find({cid: {$size: 2}});
```

E) Obtenha todos os posts de Pedro

```
> db.post.find({uid: idpedro._id});
```

F) Obtenha todos os posts de Pedro que não tenha comentários

```
> db.post.find({"$and": [{uid: idpedro._id}, {cid: {$size: 0}}]});
```

G) Obtenha todos usuários que tenham posts que tenham comentários

//não consegui