



# Embarcadero conference

embarcadero®



# MongoDB com Delphi

Um oceano de possibilidades

[Thulio Bittencourt]

# [MongoDB com Delphi - Um oceano de possibilidades]

- Porque surgiu o NoSQL
- Os dados nunca dorme
- “NO” não é não!
- Como surgiu o MongoDB
- Quem utiliza?
- Não pense relacional
- Simples e rápido como tem que ser
- O Programador tem a força
- Com grandes poderes vem grandes responsabilidades
- Todo programar é um pouco “Mongo”
- Quando usar MongoDB
- Estudo de Caso ([Golan Telecon](#))
- Recursos Animais Delphi + MongoDB
- Full Text Search
- Geospatial
- AutoComplete
- POS
- Firedac + MongoDB Usando SQL
- Recursos Especiais

# [Porque surgiu o NoSQL]

- As primeiras noções de SQL foram descritas em um paper de 1970 (<http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>)
- O paradigma de programação com orientação a objetos só começou a ser divulgado em 1980.
- O surgimento do C++ "Evolução da linguagem C" com orientação a objetos só ocorrido em 1983
- A internet nasceu (ainda que bem rústica nessa época) em meados dos anos 80.
- SQL e banco de dados relacional é um conceito bem antigo, "*pré orientação a objetos*" e "*pré internet*".
- NoSQL são sistemas de armazenamento de dados que visam resolver problemas atuais como escalabilidade, velocidade, adaptabilidade e consistência.

[Os dados nunca dormem]

2012 - 2016 - Crescimento de 1,1 bilhão de usuários na internet

Hoje estima-se em 3,2 bilhões de pessoas conectadas à internet

<https://www.domo.com>

## Por Minuto



1.736.111 fotos



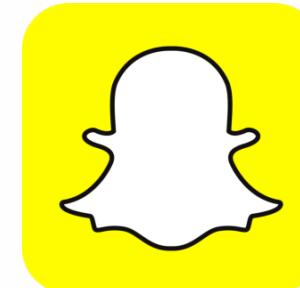
4.166.667 Likes



694 corridas



300 horas



284.722 snaps

[“NO”, não é não!!!]



Not  
only SQL

# [Como Surgiu o MongoDB]

UM DOS CASE DE SUCESSO MAIS METEÓRICO DA HISTÓRIA DOS BANCOS DE DADOS  
COM O PIOR NOME DE BANCO DE DADOS DO MERCADO.

- 2007 início dos projetos
- 2008 lançado a primeira versão
- Atualmente na Versão 3.2

## humongous

Enorme, grande, gigante.



Dwight Merriman



Eliot Horowitz



[Quem utiliza?]

Com certeza você já utilizou  
algum serviço Mongo

terra



globo.com

foursquare™



EASY TAXI



ingresso.com

EA  
GAMES™



mercado  
livre



[Não pense relacional]

Porque é tão difícil mudar nossa forma de pensar?

木林森森

ÁRVORE

BOSQUE

FLORESTA





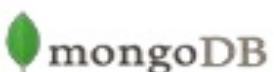
[Não pense relacional]

## Relational



## Document Data Structure

```
{   customer_id : 1,  
    first_name : "Mark",  
    last_name : "Smith",  
    city : "San Francisco",  
    location : [40.74, -73.97],  
    image : <binary>,  
    phones: [ {  
        number : "1-212-777-1212",  
        dnc : true,  
        type : "home"  
    },  
    {  
        number : "1-212-777-1213",  
        type : "cell"  
    } ] }
```





[Não pense relacional]

- Mude sua forma de pensar
- Deixe os paradigmas de lado
- Escalabilidade + Escalabilidade
- RAD ao Extremo
- **BOP** (Banco Orientado a Problemas) X **BOR** (Banco Orientado a Respostas)

Se ficar difícil, **PARE!!!**

Você está sendo um **mono**

[Não pense relacional]



Qual a melhor ferramenta?



**[Simples e rápido como tem que ser]**

- Uma das menores curvas de aprendizado entre os bancos NoSQL
- Utiliza a linguagem/notação javascript
- Facilidade de instalação e utilização do banco
- Sem dependências para utilização, não precisa de instalador
- Suporte completo do Delphi via Firedac, TCP, Rest

## VAMOS INSTALAR O MONGODB?

```
mongod --dbpath=C:\data\db --logpath=C:\data\db\log.txt --install
```

[O Programador tem a força]

## VAMOS CONFIGURAR NOSSA CONEXÃO



## OPERAÇÃO DE INSERT

```
INSERT INTO PRODUTOS (ITEM, QTD) VALUES ('ABACAXI', 15);
```

```
db.produtos.insert( { item: "Abacaxi", qtd: 15 } );
```

**Não existe create table, nem alter table  
Basta inserir um registro e tudo passa a existir**



[O Programador tem a força]

## OPERAÇÃO DE UPDATE

```
UPDATE PRODUTOS SET QTD = 20 WHERE ITEM = 'ABACAXI';
```

```
db.produtos.update( { item: "Abacaxi" } , { $set : { qtd : 20 } } );
```



[O Programador tem a força]

## OPERAÇÃO DE BUSCA

```
SELECT * FROM PRODUTOS WHERE ITEM = 'ABACAXI';
```

```
db.produtos.find({ item: "Abacaxi" });
```



[O Programador tem a força]

## OPERAÇÃO DE DELETE

```
DELETE FROM PRODUTOS WHERE ITEM = 'ABACAXI';
```

```
db.produtos.remove({ item: "Abacaxi" });
```

# [O Programador tem a força]



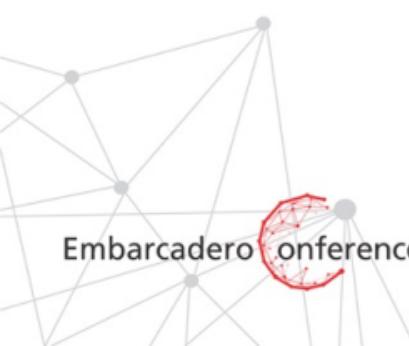
- Não existem mais create ou alter tables
- E disse o programador: “haja banco, e houve banco”
- O fluxo vai da aplicação para o banco
- Adeus DBA's (sic)
- Pensado para ser RAD
- Somente uma linguagem RAD como o Delphi para acompanhar essa velocidade

VAMOS CRIAR O NOSSO  
PRIMEIRO EXEMPLO?



[O Programador tem a força]

SERÁ QUE REALMENTE É MAIS FÁCIL?



Embarcadero conference

# SQL Implementation

```
$base_query = "SELECT imsi FROM moc WHERE callEventStartTimeStamp >= ". $start  
    . " UNION SELECT imsi FROM mtc WHERE callEventStartTimeStamp >= ". $start  
$base_query = "SELECT imsi FROM (" . $base_query . ") AS qry ";  
  
if (isset($args['imsi']))  
    $base_query .= "WHERE imsi = " . $this->_connection->real_escape_string($args['imsi']) . "";  
$base_query .= "GROUP BY imsi ";  
  
$mte_join_query = "SELECT 'mte' AS type, imsi, SUM(callEventDuration) AS duration, SUM(CEILING(callEventDuration/60)*60) AS duration_round "  
    . ", SUM(chargeAmount) charge "  
    . ", SUM(IF(SUBSTRING(callingNumber, 1, 3)='972', callEventDuration, IF(CHAR_LENGTH(callingNumber)<=10, callEventDuration, 0))) AS israel_duration "  
    . ", SUM(IF(SUBSTRING(callingNumber, 1, 3)='972', CEILING(callEventDuration/60)*60, IF(CHAR_LENGTH(callingNumber)<=10, CEILING  
(callEventDuration/60)*60, 0))) AS israel_duration_round "  
    . ", SUM(IF(SUBSTRING(callingNumber, 1, 3)!='972', IF(CHAR_LENGTH(callingNumber)>10, callEventDuration, 0), 0)) AS non_israel_duration "  
    . ", SUM(IF(SUBSTRING(callingNumber, 1, 3)!='972', IF(CHAR_LENGTH(callingNumber)>10, CEILING(callEventDuration/60)*60, 0), 0)) AS  
non_israel_duration_round "  
    . ", SUM(IF(callEventDuration = 0, 1, 0)) AS sms_count "  
    . "FROM mtc "  
    . "WHERE callEventStartTimeStamp >= ". $start . "  
    . "GROUP BY type, imsi";
```

# Implementation cont.

```
$moc_join_query = "SELECT 'moc' AS type, imsi, SUM(callEventDuration) AS duration, SUM(CEILING(callEventDuration/60)*60) AS duration_round "
    .", SUM(chargeAmount) charge "
    .", SUM(IF(SUBSTRING(connectedNumber, 1, 3)='972', callEventDuration, IF(CHAR_LENGTH(connectedNumber)<=10, callEventDuration, 0))) AS
israel_duration "
    .", SUM(IF(SUBSTRING(connectedNumber, 1, 3)='972', CEILING(callEventDuration/60)*60, IF(CHAR_LENGTH(connectedNumber)<=10, CEILING
(callEventDuration/60)*60, 0))) AS israel_duration_round "
    .", SUM(IF(SUBSTRING(connectedNumber, 1, 3)!='972', IF(CHAR_LENGTH(connectedNumber)> 10, callEventDuration, 0), 0)) AS non_israel_duration "
    .", SUM(IF(SUBSTRING(connectedNumber, 1, 3)!='972', IF(CHAR_LENGTH(connectedNumber)> 10, CEILING(callEventDuration/60)*60, 0), 0)) AS
non_israel_duration_round "
    .", SUM(IF(callEventDuration = 0, 1, 0)) AS sms_count "
    ."FROM moc "
    ."WHERE callEventStartTimeStamp >=". $start . " "
    ."GROUP BY type, imsi";
// $gprs_join_query = $this->build_query($args, "gprs");
```

# It still continues

```
$group_query = "SELECT base.imsi, moc.duration AS moc_duration, moc.charge AS moc_charge,"  
    ."mte.duration AS mtc_duration, mtc.charge AS mtc_charge,"  
    ."mte.duration_round AS mtc_duration_round, moc.duration_round AS moc_duration_round,"  
    ."moc.israel_duration AS moc_israel_duration, moc.non_israel_duration AS moc_non_israel_duration,"  
    ."moc.israel_duration_round AS moc_israel_duration_round, moc.non_israel_duration_round AS moc_non_israel_duration_round,"  
    ."mte.israel_duration AS mtc_israel_duration, mtc.non_israel_duration AS mtc_non_israel_duration,"  
    ."mte.israel_duration_round AS mtc_israel_duration_round, mtc.non_israel_duration_round AS mtc_non_israel_duration_round,"  
    ."mte.sms_count AS mtc_sms_count, moc.sms_count AS moc_sms_count"  
    ."FROM "  
    ."( ".$base_query . " ) AS base "  
    ." LEFT JOIN (" . $mte_join_query . " ) AS mte ON base.imsi = mte.imsi "  
    ." LEFT JOIN (" . $moc_join_query . " ) AS moc ON base.imsi = moc.imsi ";
```

```
if (isset($args['limit'])) {  
    $limit = (int) $args['limit'];  
}  
else {  
    $limit = 100000;  
}
```

# How it works with Mongo?

```
$base_match = array(
    '$match' => array(
        'source' => 'nrtrde',
        'unified_record_time' => array('$gte' => new MongoDB\BSON\DateTime($charge_time)),
    )
);
$where = array(
    '$match' => array(
        'record_type' => 'MOC',
        'connectedNumber' => array('$regex' => '^972'),
        'event_stamp' => array('$exists' => false),
        'deposit_stamp' => array('$exists' => false),
        'callEventDurationRound' => array('$gt' => 0), // not sms
    ),
);
$group = array(
    '$group' => array(
        '_id' => '$imsi',
        "moc_israel" => array('$sum' => '$callEventDurationRound'),
        'lines_stamps' => array('$addToSet' => '$stamp'),
    ),
);
$project = array(
    '$project' => array(
        'imsi' => '$_id',
        '_id' => 0,
        'moc_israel' => 1,
        'lines_stamps' => 1,
    ),
);
```

# How it works with Mongo?

```
$having = array(
    '$match' => array(
        'moc_israel' => array('$gte' => Billrun_Factory::config()->getConfigValue('nrtrde.thresholds.moc.israel'))
    ),
);

$moc_israel = $lines->aggregate($base_match, $where, $group, $project, $having);

//sms out to all numbers
$where['$match']['record_type'] = 'MOC';
$where['$match']['callEventDurationRound'] = 0;
$group['$group']['sms_out'] = $group['$group']['mte_all'];
unset($group['$group']['mte_all']);
unset($having['$match']['mte_all']);
$group['$group']['sms_out'] = array('$sum' => 1);
$having['$match']['sms_out'] = array('$gte' => Billrun_Factory::config()->getConfigValue('nrtrde.thresholds.smsout'));
$project['$project']['sms_out'] = 1;
unset($project['$project']['mte_all']);

$sms_out = $lines->aggregate($base_match, $where, $group, $project, $having);
```

[Com grandes poderes vem grandes responsabilidades]

- Atenção na tipificação dos dados
- O Mongo não deixa você fazer besteira
- Esqueça os relacionamentos e vamos embedar
- Projetar, Modelar, Programar, Testar, Debugar  
Pela União dos seus poderes...
- Ser RAD só depende de você



HORA DO NOSSOS ULTRA  
MEGA POWER RAD CRUD?

[Todo programador é um pouco MONGO]

# A MALDIÇÃO DA TABELA VÍNCULO

```
CREATE TABLE VINCULO (
    ID          INTEGER NOT NULL,
    PRODUTO     INTEGER,
    TIPO        INTEGER,
    BASE        VARCHAR(512),
    RAZAO       VARCHAR(512),
    SUBTIPO    INTEGER,
    CODIGO      VARCHAR(14)
);
```

**2.147.483,647**

Tipo de Produtos

Existe de tudo nessa tabela, até a cura do câncer e da AIDS, é só procurar com atenção



- Grande volume de dados, Escrita (**Sharding**)
- Bases exclusivas para consultas (**ReplicaSet**)
- Operações Atômicas e Temporais (**Documentos**)
- Armazenamento de Arquivos (**GridFS**)
- Cluster no mundo dos mortais é possível sim!
- Dados baseados em Geolocalização
- Alterações contínuas de estrutura (**schema**)
- Você não tem um DBA entre seus funcionários
- Ou simplesmente está “cansado” dos DBA’s



- 500 Milhões de Registros de Chamadas por mês
- 10 tipos de CDR's com mais de 1 TB por Collection
- 500M / 28 Dias / 24hs / 60min / 60sec = 232 CDR Sec / 15.360 CDR Min
- 500M / 5 Dias / 24hs / 60min / 60sec = 1152 CDR Sec / 69.120 CDR Min
- 1,32 Teras = 1.436.122,538 (um bilhão quatrocentos e trinta e seis milhões cento e vinte e dois mil quinhentos e trinta e oito)

[Recursos Animais do Delphi + MongoDB]



A famosa busca fonética agora é tão simples quanto criar um index, ou melhor é só criar um index mesmo!!!

## Criar o Index

```
db.musica.createIndex( { letra: "text", $language: "pt" } )
```

## Realizar a Busca

```
db.musica.find( { $text: { $search: "Caminho" } } )
```

## Realizar a Busca Vários Termos

```
db.musica.find( { $text: { $search: "Caminho Vento Documento" } } )
```

## Partes de Frases

```
db.musica.find( { $text: { $search: "\"Caminhando Con\"" } } )
```

## Excluindo Termos da Busca

```
db.musica.find( { $text: { $search: "Caminhar -vento" } } )
```

Eu não sei o que vocês fizeram no verão passado, mas o MongoDB pode me dizer pelo menos onde você esteve

### Criar o Index

```
db.places.createIndex( { loc : "2dsphere" } )
```

```
db.places.find(  
  {  
    location: {  
      $nearSphere: {  
        $geometry: {  
          type : "Point",  
          coordinates : [ -73.9667, 40.78 ]  
        },  
        $minDistance: 1000,  
        $maxDistance: 5000  
      }  
    }  
  }  
)
```

Vamos procurar uma agulha no palheiro?

+ 4 Milhões de Registros em Segundos

Sim, é possível usar em nosso Mundo

# Ponto de Vendas com documentos Embedados [ Ações atômicas e temporais ]

Situações Extremas requerem soluções extremas

MONGODB + SQL

[ Sim é Possível!!! Graças a Firedac ]

- Facilidade para atualização das bases com replicaset
- MMAPv1 X WiredTiger (Storage Engine)
- Upserts
- Aggregation
- MapReduce
- Importação de Arquivos CSV
- \$Isolated e Two Phase Transactions

@ bittencourtthulio@gmail.com

 thuliobitten

 /bittencourtthulio

 /bittencourtthulio

[www.zabuu.com.br/blog](http://www.zabuu.com.br/blog)

[www.zabuu.com.br/zabuulabs](http://www.zabuu.com.br/zabuulabs)

# [OBRIGADO]

Embarcadero conference





# Embarcadero conference

embarcadero®