



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

# NoSQL

## Trabalho Prático nº3

Tecnologia de Bases de Dados  
MIEIC - 2017/2018

Elementos do Grupo:

Andreia Rodrigues - up201404691 - [up201404691@fe.up.pt](mailto:up201404691@fe.up.pt)

Eduardo Leite - gei12068 - [gei12068@fe.up.pt](mailto:gei12068@fe.up.pt)

# Index

<b>1. Objetivo do Trabalho</b>	<b>4</b>
<b>2. Modelo do Documento Mongo</b>	<b>4</b>
2.1 Coleção UCS	4
2.2 Coleção Docentes	5
2.3 Coleção Ocorrências	5
<b>3. Migração da Base de Dados</b>	<b>6</b>
3.1 Coleção UCS (Tabelas XUCS, XTIPOSAULA e XDSD)	6
3.1.1 PL/SQL - Geração do Documento XML	6
3.1.2 XSL - Código de Geração do Documento JSON	7
3.2 Coleção Docentes (Tabelas XDOCENTES, XTIPOSAULA e XDSD)	9
3.2.1 PL/SQL - Geração do Documento XML	9
3.2.2 XSL - Código de Geração do Documento JSON	10
3.3 Coleção Ocorrências (Tabela XOCORRENCIAS)	12
3.3.1 PL/SQL - Geração do Documento XML	12
3.3.2 XSL - Código de Geração do Documento JSON	13
<b>4. Queries à Base de Dados</b>	<b>14</b>
4.1 Query 1	14
4.1.1 Pergunta	14
4.1.2 Resposta	14
4.1.2.1 Base de Dados Relacional	14
4.1.2.2 Base de Dados Mongo	15
4.1.3 Resultados	15
4.2 Query 2	15
4.2.1 Pergunta	15
4.2.2 Resposta	16
4.2.2.1 Base de Dados Relacional	16
4.2.2.2 Base de Dados Mongo	16
4.2.3 Resultados	16
4.3 Query 3	18
4.3.1 Pergunta	18
4.3.2 Resposta	18
4.3.2.1 Base de Dados Relacional	18
4.3.2.2 Base de Dados Mongo	19
4.3.3 Resultados	19
4.4 Query 4	20
4.4.1 Pergunta	20
4.4.2 Resposta	21

4.4.2.1 Base de Dados Relacional	21
4.4.2.2 Base de Dados Mongo	21
4.4.3 Resultados	21
4.5 Query 5	23
4.5.1 Pergunta	23
4.5.2 Resposta	23
4.5.2.1 Base de Dados Relacional	23
4.5.2.2 Base de Dados Mongo	24
4.5.3 Resultados	24
4.6 Query 6	25
4.6.1 Pergunta	25
4.6.2 Resposta	25
4.6.2.1 Base de Dados Relacional	25
4.6.2.2 Base de Dados Mongo	26
4.6.3 Resultados	26
4.6.4 Considerações	27
<b>5. Comparação Mongo - Oracle</b>	<b>27</b>
<b>6. Anexos</b>	<b>28</b>
6.1 Função print_clob()	28
6.2 Servidor NodeJS Para Inserção dos Documentos JSON na Base de Dados Mongo	28

# 1. Objetivo do Trabalho

O objetivo deste trabalho consiste na construção de uma base de dados usando uma abordagem NoSQL e comparação dos resultados com os resultados obtidos pela abordagem relacional.

## 2. Modelo do Documento Mongo

Abaixo são apresentadas as estruturas dos documentos a serem inseridos nas diferentes coleções.

Foram criadas 3 coleções diferentes:

- UCS - contém a informação relativa às unidades curriculares. Em cada unidade curricular está um subdocumento com todos os tipos de aula da UC e para cada tipo de aula existe um subdocumento correspondente às distribuições de docentes.
- Docentes- contém a informação relativa aos docentes. Em cada docente existe um subdocumento com informação sobre os serviços prestados/a prestar num determinado tipo de aula de uma UC, no ano letivo e período correspondentes.
- Ocorrências - contém a informação relativa a uma ocorrência de uma unidade curricular num determinado ano letivo e período.

Comparativamente ao modelo de dados presente na base de dados relacional foram omitidos 2 campos na tabela de Ocorrências: "objetivos" e "conteúdo". A decisão de omitir estes campos foi tomada devido ao grande aumento de tamanho dos documentos JSON quando os campos eram incluídos e problemas posteriores ao inserir estes ficheiros grandes na base de dados Mongo.

### 2.1 Coleção UCS

```
{
  "codigo": {type: String, required: true, unique: true},
  "designacao" : {type: String, required: true},
  "sigla_uc": {type: String, required: true, unique: true},
  "curso": {type: String, required: true},
  "tipos_aula": [
    {
      "id": {type: String, required: true},
      "tipo": {type: String, required: true, maxlength: 2},
      "ano_letivo": {type: String, required: true, maxlength: 9},
      "periodo": {type: String, required: true, maxlength: 2},
```

```

    "codigo": {type: String, required: true},
    "turnos": {type: Number, required: true},
    "n_aulas": {type: Number, required: true},
    "horas_turno": {type: Number, required: true},
    "distribuicoes": [
      {
        "nr": {type: String, required: true},
        "horas": {type: Number, required: true},
        "fator": {type: Number, required: true},
        "ordem": {type: Number, required: true}
      }
    ]
  }
]
}

```

## 2.2 Coleção Docentes

```

{
  "nr": {type: String, required: true, unique: true},
  "nome": {type: String, required: true},
  "sigla": {type: String, required: true},
  "categoria": {type: Number},
  "proprio": {type: String, required: true},
  "apelido": {type: String, required: true},
  "estado": {type: String, required: true, maxlength: 2},
  "distribuicoes": [
    {
      "horas": {type: Number, required: true},
      "fator": {type: Number, required: true},
      "ordem": {type: Number, required: true},
      "tipo_aula": {type: String, required: true, maxlength: 2},
      "ano_letivo": {type: String, required: true, maxlength: 9},
      "periodo": {type: String, required: true, maxlength: 2},
      "codigo": {type: String, required: true},
      "turnos": {type: Number, required: true},
      "n_aulas": {type: Number, required: true},
      "horas_turno": {type: Number, required: true}
    }
  ]
}

```

## 2.3 Coleção Ocorrências

```

{
  "codigo": {type: String, required: true, unique: true},

```

```

"ano_letivo": {type: String, required: true, maxlength: 9},
"periodo": {type: String, required: true, maxlength: 2},
"inscritos": {type: Number, required: true},
"com_frequencia": {type: Number, required: true},
"aprovados": {type: Number, required: true},
"objetivos": {type: String},
"conteudo": {type: String},
"departamento": {type: String, required: true}
}

```

### 3. Migração da Base de Dados

A migração dos dados da base de dados relacional para a base de dados NoSQL passa por várias etapas: em primeiro lugar é executado o procedimento em PL/SQL para geração do documento XML referente às tabelas que pretendemos. O XML é imprimido na consola DBMS Output através da função `print_clob()`. Depois é aplicada uma transformação do documento obtido através de XSL para obtenção do documento JSON correspondente.

Por limitações encontradas em ambos os IDE's Robo 3T e MongoCompass em relação à inserção de documentos grandes na base de dados, foi criado um servidor em NodeJS para proceder à inserção dos documentos JSON nas respetivas coleções da base de dados Mongo.

#### 3.1 Coleção UCS (Tabelas XUCS, XTIPOSAULA e XDSD)

##### 3.1.1 PL/SQL - Geração do Documento XML

```

create or replace procedure export_ucs IS
v_sqlselect VARCHAR2(4000);
v_queryctx DBMS_XMLQuery.ctxType;
v_clob_par CLOB;
BEGIN
v_sqlselect :=
'
select u.codigo, u.designacao, u.sigla_uc, u.curso, CURSOR
(
    select t.id, t.tipo, t.ano_letivo, t.periodo, t.codigo, t.turnos, t.n_aulas,
t.horas_turno, CURSOR
(
    select dsd.nr, dsd.horas, dsd.fator, dsd.ordem
from gtd10.xdsd dsd
where dsd.id = t.id
) as distribuicoes
from gtd10.xtiposaula t
where u.codigo = t.codigo) as tipos_aula
from gtd10.xucs u
';
v_queryctx := DBMS_XMLQuery.newContext(v_sqlselect);

```

```

DBMS_XMLQuery.setEncodingTag(v_queryctx, 'ISO-8859-1');
DBMS_XMLQuery.setRowSetTag(v_queryctx, UPPER('UCS'));
DBMS_XMLQuery.setRowTag(v_queryctx, UPPER('UC'));
v_clob_par := DBMS_XMLQuery.getXML(v_queryctx);
DBMS_XMLQuery.closeContext(v_queryctx);
print_clob(v_clob_par);
END;

```

### 3.1.2 XSL - Código de Geração do Documento JSON

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">

  <xsl:template match="/">
    <xsl:text>[</xsl:text>
    <xsl:apply-templates select="//UC"/>
    <xsl:text>]</xsl:text>
  </xsl:template>

  <xsl:template match='UC'>
    <xsl:variable name="total"><xsl:value-of
select="count(//UC)"/></xsl:variable>
    <xsl:text>{"codigo": "</xsl:text><xsl:value-of
select="translate(normalize-space(CODIGO), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:text>"designacao": "</xsl:text><xsl:value-of
select="normalize-space(DESIGNACAO)"/><xsl:text>",</xsl:text>
    <xsl:text>"sigla_uc": "</xsl:text><xsl:value-of
select="translate(normalize-space(SIGLA_UC), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:text>"curso": "</xsl:text><xsl:value-of
select="translate(normalize-space(CURSO), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:text>"tipos_aula": [</xsl:text><xsl:apply-templates
select='TIPOS_AULA' />
    <xsl:text>]]</xsl:text>
    <xsl:if test="position() < xs:integer($total)">
      <xsl:text>,</xsl:text>
    </xsl:if>
  </xsl:template>

  <xsl:template match='TIPOS_AULA'>
    <xsl:apply-templates select='TIPOS_AULA_ROW' />
  </xsl:template>

  <xsl:template match='TIPOS_AULA_ROW'>
    <xsl:variable name="max"><xsl:value-of
select="max(..//TIPOS_AULA_ROW/@num)"/></xsl:variable>
    <xsl:text>{"id": "</xsl:text><xsl:value-of
select="translate(normalize-space(ID), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:text>"tipo": "</xsl:text><xsl:value-of
select="translate(normalize-space(TIPO), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:text>"ano_letivo": "</xsl:text><xsl:value-of

```

```

select="translate(normalize-space(ANO_LETIVO), ' ', '')"/><xsl:text>"/></xsl:text>
    <xsl:text>"periodo": "</xsl:text><xsl:value-of
select="translate(normalize-space(PERIODO), ' ', '')"/><xsl:text>"/></xsl:text>
    <xsl:text>"codigo": "</xsl:text><xsl:value-of
select="translate(normalize-space(CODIGO), ' ', '')"/><xsl:text>"/></xsl:text>
    <xsl:text>"turnos": </xsl:text>
    <xsl:choose>
        <xsl:when test="TURNOS > 0">
            <xsl:value-of select="TURNOS"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>,</xsl:text>
    <xsl:text>"n_aulas": </xsl:text>
    <xsl:choose>
        <xsl:when test="N_AULAS > 0">
            <xsl:value-of select="N_AULAS"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>,</xsl:text>
    <xsl:text>"horas_turno": </xsl:text>
    <xsl:choose>
        <xsl:when test="HORAS_TURNO > 0">
            <xsl:value-of select="HORAS_TURNO"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>,</xsl:text>
    <xsl:text>"distribuicoes": [</xsl:text><xsl:apply-templates
select='DISTRIBUICOES' />
    <xsl:text>]] </xsl:text>
    <xsl:if test="@num < xs:integer($max)">
        <xsl:text>,</xsl:text>
    </xsl:if>
</xsl:template>

<xsl:template match='DISTRIBUICOES'>
    <xsl:apply-templates select='DISTRIBUICOES_ROW' />
</xsl:template>

<xsl:template match='DISTRIBUICOES_ROW'>
    <xsl:variable name="max"><xsl:value-of
select="max(..//DISTRIBUICOES_ROW/@num)"/></xsl:variable>
    <xsl:text>{"nr": "</xsl:text><xsl:value-of
select="translate(normalize-space(NR), ' ', '')"/><xsl:text>"/></xsl:text>
    <xsl:text>"horas": </xsl:text>
    <xsl:choose>

```



```

        <xsl:when test="HORAS > 0">
            <xsl:value-of select="HORAS"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>,</xsl:text>
    <xsl:text>"fator": </xsl:text>
    <xsl:choose>
        <xsl:when test="FATOR > 0">
            <xsl:value-of select="FATOR"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>,</xsl:text>
    <xsl:text>"ordem": </xsl:text>
    <xsl:choose>
        <xsl:when test="ORDEM > 0">
            <xsl:value-of select="ORDEM"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>}</xsl:text>
    <xsl:if test="@num < xs:integer($max)">
        <xsl:text>,</xsl:text>
    </xsl:if>
</xsl:template>
</xsl:stylesheet>

```

## 3.2 Coleção Docentes (Tabelas XDOCENTES, XTIPOSAULA e XDSD)

### 3.2.1 PL/SQL - Geração do Documento XML

```

create or replace procedure export_docentes IS
v_sqlselect VARCHAR2(4000);
v_queryctx DBMS_XMLQuery.ctxType;
v_clob_par CLOB;
BEGIN
v_sqlselect :=
'
select d.nr, d.nome, d.sigla, d.categoria, d.proprio, d.apellido, d.estado, CURSOR(
    select dsd.horas, dsd.fator, dsd.ordem, CURSOR(
        select t.tipo, t.ano_letivo, t.periodo, t.codigo, t.turnos, t.n_aulas,
t.horas_turno

```

```

        from gtdl0.xtiposaula t
        where dsd.id = t.id and d.nr = dsd.nr
    ) as tipo_aula
    from gtdl0.xdsd dsd
    where dsd.nr = d.nr
) as distribuicao
from gtdl0.xdocentes d
';
v_queryctx := DBMS_XMLQuery.newContext(v_sqlselect);
DBMS_XMLQuery.setEncodingTag(v_queryctx, 'ISO-8859-1');
DBMS_XMLQuery.setRowSetTag(v_queryctx, UPPER('DOCENTES'));
DBMS_XMLQuery.setRowTag(v_queryctx, UPPER('DOCENTE'));
v_clob_par := DBMS_XMLQuery.getXML(v_queryctx);
DBMS_XMLQuery.closeContext(v_queryctx);
print_clob(v_clob_par);
END;

```

### 3.2.2 XSL - Código de Geração do Documento JSON

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">

  <xsl:template match="/">
    <xsl:text>[</xsl:text>
    <xsl:apply-templates select="//DOCENTE"/>
    <xsl:text>]</xsl:text>
  </xsl:template>

  <xsl:template match='DOCENTE'>
    <xsl:variable name="total"><xsl:value-of
select="count(//DOCENTE)"/></xsl:variable>
    <xsl:text>{"nr": "</xsl:text><xsl:value-of
select="translate(normalize-space(NR), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:text>"nome": "</xsl:text><xsl:value-of
select="normalize-space(NOME)"/><xsl:text>",</xsl:text>
    <xsl:text>"sigla": "</xsl:text><xsl:value-of
select="translate(normalize-space(SIGLA), ' ', '')"/><xsl:text>",</xsl:text>
    <xsl:choose>
      <xsl:when test="CATEGORIA > 0">
        <xsl:text>"categoria": </xsl:text>
        <xsl:value-of select="CATEGORIA"/>
        <xsl:text>,</xsl:text>
      </xsl:when>
    </xsl:choose>
    <xsl:text>"proprio": "</xsl:text><xsl:value-of
select="normalize-space(PROPRIO)"/><xsl:text>",</xsl:text>
    <xsl:text>"apelido": "</xsl:text><xsl:value-of
select="normalize-space(APELIDO)"/><xsl:text>",</xsl:text>
    <xsl:text>"estado": "</xsl:text><xsl:value-of
select="translate(normalize-space(ESTADO), ' ', '')"/><xsl:text>",</xsl:text>

```

```

        <xsl:text>"distribuicoes": [</xsl:text><xsl:apply-templates
select='DISTRIBUICOES' />
        <xsl:text>]]</xsl:text>
        <xsl:if test="position() < xs:integer($total)">
            <xsl:text>, </xsl:text>
        </xsl:if>
    </xsl:template>

    <xsl:template match='DISTRIBUICOES'>
        <xsl:apply-templates select='DISTRIBUICOES_ROW' />
    </xsl:template>

    <xsl:template match='DISTRIBUICOES_ROW'>
        <xsl:variable name="max"><xsl:value-of
select="max(.. / DISTRIBUICOES_ROW / @num)" /></xsl:variable>
        <xsl:text>{"horas": </xsl:text>
        <xsl:choose>
            <xsl:when test="HORAS > 0">
                <xsl:value-of select="HORAS" />
            </xsl:when>
            <xsl:otherwise>
                <xsl:text>0</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:text>, </xsl:text>
        <xsl:text>"fator": </xsl:text>
        <xsl:choose>
            <xsl:when test="number(FATOR) > 0">
                <xsl:value-of select="FATOR" />
            </xsl:when>
            <xsl:otherwise>
                <xsl:text>0</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:text>, </xsl:text>
        <xsl:text>"ordem": </xsl:text>
        <xsl:choose>
            <xsl:when test="ORDEM > 0">
                <xsl:value-of select="ORDEM" />
            </xsl:when>
            <xsl:otherwise>
                <xsl:text>0</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:text>,</xsl:text>
        <xsl:apply-templates select='TIPO_AULA' />
        <xsl:if test="@num < xs:integer($max)">
            <xsl:text>, </xsl:text>
        </xsl:if>
    </xsl:template>

    <xsl:template match='TIPO_AULA'>
        <xsl:apply-templates select='TIPO_AULA_ROW' />
    </xsl:template>

```

```

<xsl:template match='TIPO_AULA_ROW'>
  <xsl:text>"tipo_aula": "</xsl:text><xsl:value-of
select="translate(normalize-space(TIPO),' ','')" /><xsl:text>", </xsl:text>
  <xsl:text>"ano_letivo": "</xsl:text><xsl:value-of
select="translate(normalize-space(ANO_LETIVO),' ','')" /><xsl:text>", </xsl:text>
  <xsl:text>"periodo": "</xsl:text><xsl:value-of
select="translate(normalize-space(PERODO),' ','')" /><xsl:text>", </xsl:text>
  <xsl:text>"codigo": "</xsl:text><xsl:value-of
select="translate(normalize-space(CODIGO),' ','')" /><xsl:text>", </xsl:text>
  <xsl:text>"turnos": </xsl:text>
  <xsl:choose>
    <xsl:when test="TURNOS > 0">
      <xsl:value-of select="TURNOS"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>0</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>,</xsl:text>
  <xsl:text>"n_aulas": </xsl:text>
  <xsl:choose>
    <xsl:when test="N_AULAS > 0">
      <xsl:value-of select="N_AULAS"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>0</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>,</xsl:text>
  <xsl:text>"horas_turno": </xsl:text>
  <xsl:choose>
    <xsl:when test="HORAS_TURNO > 0">
      <xsl:value-of select="HORAS_TURNO"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>0</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>} </xsl:text>
</xsl:template>
</xsl:stylesheet>

```

## 3.3 Coleção Ocorrências (Tabela XOCORRENCIAS)

### 3.3.1 PL/SQL - Geração do Documento XML

```

create or replace procedure export_ocorrencias is
v_sqlselect VARCHAR2(4000);
v_queryctx DBMS_XMLQuery.ctxType;
v_clob_par CLOB;

```

```

BEGIN
v_sqlselect :=
'
select o.codigo, o.ano_letivo, o.periodo, o.inscritos, o.com_frequencia,
o.aprovados, o.departamento
from gtd10.xocorrencias o
';
v_queryctx := DBMS_XMLQuery.newContext(v_sqlselect);
DBMS_XMLQuery.setEncodingTag(v_queryctx, 'ISO-8859-1');
DBMS_XMLQuery.setRowSetTag(v_queryctx, UPPER('OCORRENCIAS'));
DBMS_XMLQuery.setRowTag(v_queryctx, UPPER('OCORRENCIA'));
v_clob_par := DBMS_XMLQuery.getXML(v_queryctx);
DBMS_XMLQuery.closeContext(v_queryctx);
print_clob(v_clob_par);
END;

```

### 3.3.2 XSL - Código de Geração do Documento JSON

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
exclude-result-prefixes="xs"
version="2.0">

  <xsl:template match='/'>
    <xsl:text>[</xsl:text>
    <xsl:apply-templates select="//OCORRENCIA"/>
    <xsl:text>]</xsl:text>
  </xsl:template>

  <xsl:template match='OCORRENCIA'>
    <xsl:variable name="total"><xsl:value-of
select="count(//OCORRENCIA)"/></xsl:variable>
    <xsl:text>{"codigo": "</xsl:text><xsl:value-of
select="translate(normalize-space(CODIGO), ' ', ' ')/"><xsl:text>",</xsl:text>
    <xsl:text>"ano_letivo": "</xsl:text><xsl:value-of
select="translate(normalize-space(ANO_LETIVO), ' ', ' ')/"><xsl:text>",</xsl:text>
    <xsl:text>"periodo": "</xsl:text><xsl:value-of
select="translate(normalize-space(PERODO), ' ', ' ')/"><xsl:text>",</xsl:text>
    <xsl:text>"inscritos": </xsl:text>
    <xsl:choose>
      <xsl:when test="INSCRITOS > 0">
        <xsl:value-of select="INSCRITOS"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:text>0</xsl:text>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>,</xsl:text>
    <xsl:text>"com_frequencia": </xsl:text>
    <xsl:choose>
      <xsl:when test="COM_FREQUENCIA > 0">
        <xsl:value-of select="COM_FREQUENCIA" />

```

```

        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>, </xsl:text>
    <xsl:text>"aprovados": </xsl:text>
    <xsl:choose>
        <xsl:when test="APROVADOS > 0">
            <xsl:value-of select="APROVADOS" />
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>0</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>, </xsl:text>
    <xsl:text>"departamento": "</xsl:text><xsl:value-of
select="translate(normalize-space(DEPARTAMENTO), ' ', '')" /><xsl:text>" } </xsl:text>
    <xsl:if test="position() < xs:integer($total)">
        <xsl:text>, </xsl:text>
    </xsl:if>
</xsl:template>
</xsl:stylesheet>

```

## 4. Queries à Base de Dados

### 4.1 Query 1

#### 4.1.1 Pergunta

*How many class hours of each type did the program 233 got in year 2004/2005?*

#### 4.1.2 Resposta

##### 4.1.2.1 Base de Dados Relacional

```

select t.tipo, sum(t.turnos * t.horas_turno) previsto
from XTIPOSAULA t
natural join XUCS u
where u.curso=233
and t.ano_letivo = '2004/2005'
group by t.tipo
order by t.tipo;

```

### 4.1.2.2 Base de Dados Mongo

```
db.ucs.aggregate
([
  {$unwind: "$tipos_aula"},
  {$match: {"curso" : "233", "tipos_aula.ano_letivo" : "2004/2005"}},
  {$group: {_id : "$tipos_aula.tipo", previsto : {$sum : {$multiply :
["$tipos_aula.turnos", "$tipos_aula.horas_turno"]}}}},
  {$sort: { _id: 1 }}
])
```

### 4.1.3 Resultados

	TIPO	PREVISTO
1	P	590,5
2	T	320
3	TP	762,5

Figura 1 - Resultados da pergunta 1 em SQL.

```
/* 1 */
{
  "_id" : "P",
  "previsto" : 590.5
}

/* 2 */
{
  "_id" : "T",
  "previsto" : 320
}

/* 3 */
{
  "_id" : "TP",
  "previsto" : 762.5
}
```

Figura 2 - Resultados da pergunta 1 em NoSQL.

## 4.2 Query 2

### 4.2.1 Pergunta

*Which courses (show the code, total class hours required, total classes assigned) have a difference between total class hours required and the service actually assigned in year 2003/2004?*

## 4.2.2 Resposta

### 4.2.2.1 Base de Dados Relacional

```
select codigo, sum(turnos * horas_turno) requeridas , sum(horas) servico
from XTIPOSAULA natural join XUCS natural join XDSD
where ano_letivo='2003/2004'
group by codigo
having sum(turnos * horas_turno) <> sum(horas)
order by codigo;
```

### 4.2.2.2 Base de Dados Mongo

```
db.ucs.aggregate
([
  {$unwind: "$tipos_aula"},
  {$unwind: "$tipos_aula.distribuicoes"},
  {$match: {"tipos_aula.ano_letivo" : "2003/2004"}},
  {$group: {_id: "$codigo", requeridas : {$sum : {$multiply :
["$tipos_aula.turnos", "$tipos_aula.horas_turno"] }}, servico : {$sum :
"$tipos_aula.distribuicoes.horas"}}}},
  {$project: {"requeridas" : 1, "servico" : 1, "notEq" :
{$ne:["$requeridas", "$servico"]}}},
  {$match: {"notEq" : true}},
  {$project: {"requeridas" : 1, "servico" : 1}},
  {$sort: { _id: 1 }}
])
```

## 4.2.3 Resultados



	❖ CODIGO	❖ REQUERIDAS	❖ SERVICO
1	CI028	6	4
2	EC1101	80	32
3	EC1103	50	28
4	EC1107	180	48
5	EC1108	404	60
6	EC1207	102	30
7	EC1209	100	28
8	EC1211	52	26
9	EC1214	40	20
10	EC2102	320	50
11	EC2103	68	24
12	EC2104	72	26
13	EC2106	52	28
14	EC2107	40	20
15	EC2108	124	34
16	EC2208	252	46
17	EC2209	126	36
18	EC2211	376	80
19	EC3101	98	22
20	EC3102	222	36
21	EC3103	88	24

(...)

Figura 3 - Resultados da pergunta 2 em SQL.

```

/* 1 */
{
  "_id" : "CI028",
  "requeridas" : 6,
  "servico" : 4
}

/* 2 */
{
  "_id" : "EC1101",
  "requeridas" : 80,
  "servico" : 32
}

/* 3 */
{
  "_id" : "EC1103",
  "requeridas" : 50,
  "servico" : 28
}

/* 4 */
{
  "_id" : "EC1107",
  "requeridas" : 180,
  "servico" : 48
}

```

(...)

Figura 4 - Resultados da pergunta 2 em NoSQL.

## 4.3 Query 3

### 4.3.1 Pergunta

*Who is the professor with more class hours for each type of class, in the academic year 2003/2004? Show the number and name of the professor, the type of class and the total of class hours times the factor.*

### 4.3.2 Resposta

#### 4.3.2.1 Base de Dados Relacional

```

SELECT t1.*
FROM

```

```

(
    select t.tipo, d.nome, sum(dsd.horas) horas
    from xdocentes d
    natural join xdsd dsd
    natural join xtiposaula t
    where ano_letivo='2003/2004'
    group by t.tipo, d.nome
) t1
INNER JOIN
(
    select tipo, max(horas) max_horas
    from
    (
        select t.tipo, d.nome, sum(dsd.horas) horas
        from xdocentes d
        natural join xdsd dsd
        natural join xtiposaula t
        where ano_letivo='2003/2004'
        group by t.tipo, d.nome
        order by d.nome
    )
    group by tipo
) t2 ON t1.tipo = t2.tipo AND t1.horas = t2.max_horas
ORDER BY t1.tipo;

```

#### 4.3.2.2 Base de Dados Mongo

```

db.docentes.aggregate
([
    {$unwind: "$distribuicoes"},
    {$match: {"distribuicoes.ano_letivo" : "2003/2004"}},
    {$group: {_id: {"tipo" : "$distribuicoes.tipo_aula", "nome" : "$nome" },
    horas : {$sum : "$distribuicoes.horas"}}},
    {$sort: { horas: -1 }},
    {$group: {_id: "$_id.tipo", "nome" : {$first : "$_id.nome"}, "horas" :
    {$first : "$horas"}}},
    {$sort: { _id: 1 }},
])

```

#### 4.3.3 Resultados

	TIPO	NOME	HORAS
1	OT	João Carlos Pascoal de Faria	3,5
2	P	Antônio Almerindo Pinheiro Vieira	30
3	T	Fernando Francisco Machado Veloso Gomes	30,67
4	TP	Cecília do Carmo Ferreira da Silva	26

Figura 4 - Resultados da pergunta 3 em SQL.

```

/* 1 */
{
  "_id" : "OT",
  "nome" : "João Carlos Pascoal de Faria",
  "horas" : 3.5
}

/* 2 */
{
  "_id" : "P",
  "nome" : "Antônio Almerindo Pinheiro Vieira",
  "horas" : 30
}

/* 3 */
{
  "_id" : "T",
  "nome" : "Fernando Francisco Machado Veloso Gomes",
  "horas" : 30.67
}

/* 4 */
{
  "_id" : "TP",
  "nome" : "Cecília do Carmo Ferreira da Silva",
  "horas" : 26
}

```

Figura 5 - Resultados da pergunta 3 em NoSQL.

## 4.4 Query 4

### 4.4.1 Pergunta

*Which is the average number of hours by professor by year in each category, in the years between 2001/2002 and 2004/2005?*

## 4.4.2 Resposta

### 4.4.2.1 Base de Dados Relacional

```
select categoria, round(avg(horas),2) avg_horas
from
(
    select d.categoria, d.nome, avg(dsd.horas) horas
    from xdocentes d
    natural join xdsd dsd
    natural join xtiposaula t
    where t.ano_letivo >= '2001/2002' and t.ano_letivo <= '2004/2005'
    group by d.categoria, d.nome
)
group by categoria
order by categoria;
```

### 4.4.2.2 Base de Dados Mongo

```
db.docentes.aggregate
([
  {$unwind: "$distribuicoes"},
  {$match: {"distribuicoes.ano_letivo" : {$gte : "2001/2002", $lte :
"2004/2005"}, "categoria" : {$ne: null}}},
  {$group: {_id: {"categoria" : "$categoria", "nome" : "$nome" }, horas :
{$avg : "$distribuicoes.horas"}}},
  {$group: { _id: "$_id.categoria", horas : {$avg : "$horas"}}},
  {$sort: { _id: 1}}
])
```

### 4.4.3 Resultados

	CATEGORIA	AVG_HORAS
1	103	2,14
2	107	2,52
3	110	2,64
4	111	4,2
5	112	3,29
6	116	3,81
7	117	3,76
8	119	6,04
9	120	4,51
10	122	4,17
11	124	4
12	125	4
13	144	1,73
14	374	4,08
15	519	3,23
16	520	6,4
17	565	1,23
18	903	2
19	10108	2
20	10119	2,67
21	10801	0,6

(...)

Figura 6 - Resultados da pergunta 4 em SQL.

```

/* 1 */
{
  "_id" : 103,
  "horas" : 2.14210526315789
}

/* 2 */
{
  "_id" : 107,
  "horas" : 2.5209097513105
}

/* 3 */
{
  "_id" : 110,
  "horas" : 2.64451708310858
}

/* 4 */
{
  "_id" : 111,
  "horas" : 4.1984126984127
}

/* 5 */
{
  "_id" : 112,
  "horas" : 3.28547008547009
}

```

(...)

Figura 7 - Resultados da pergunta 4 em NoSQL.

## 4.5 Query 5

### 4.5.1 Pergunta

*Which is the total hours per week, on each semester, that an hypothetical student enrolled in every course of a single curricular year from each program would get.*

### 4.5.2 Resposta

#### 4.5.2.1 Base de Dados Relacional

```

select t.ano_letivo, u.curso, coalesce((sum(turnos * horas_turno) / 2), 0)
horas_semestre
from xtiposaula t
natural join xucs u

```

```
group by t.ano_letivo, u.curso
order by t.ano_letivo, u.curso;
```

#### 4.5.2.2 Base de Dados Mongo

```
db.ucs.aggregate
([
  {$unwind: "$tipos_aula"}, {$group: {_id : {ano_letivo:
"$tipos_aula.ano_letivo", curso: "$curso"}, horas : {$sum : {$divide :
[{$multiply : ["$tipos_aula.turnos", "$tipos_aula.horas_turno"]}, 2 ]}}}},
  {$project: {_id : 0, "ano_letivo" : "$_id.ano_letivo", "curso" :
"$_id.curso", horas : 1}},
  {$sort: { ano_letivo: 1, curso: 1}}
])
```

#### 4.5.3 Resultados

	ANO_LETIVO	CURSO	HORAS_SEMESTRE
1	1980/1981	331	1,5
2	1992/1993	331	12
3	1992/1993	433	0
4	1993/1994	331	9
5	1994/1995	275	1,5
6	1994/1995	331	4
7	1995/1996	275	5,25
8	1995/1996	331	4
9	1996/1997	255	514
10	1996/1997	275	5,25
11	1996/1997	304	2
12	1996/1997	331	15
13	1996/1997	433	5
14	1996/1997	1600	7,5
15	1997/1998	200	1
16	1997/1998	233	13
17	1997/1998	255	699,5
18	1997/1998	275	13,25
19	1997/1998	304	5,5
20	1997/1998	331	13
21	1997/1998	400	1,5

(...)

Figura 8 - Resultados da pergunta 5 em SQL.



```

/* 1 */
{
  "horas" : 1.5,
  "ano_letivo" : "1980/1981",
  "curso" : "331"
}

/* 2 */
{
  "horas" : 12.0,
  "ano_letivo" : "1992/1993",
  "curso" : "331"
}

/* 3 */
{
  "horas" : 0.0,
  "ano_letivo" : "1992/1993",
  "curso" : "433"
}

/* 4 */
{
  "horas" : 9.0,
  "ano_letivo" : "1993/1994",
  "curso" : "331"
}

```

(...)

Figura 9 - Resultados da pergunta 5 em NoSQL.

## 4.6 Query 6

### 4.6.1 Pergunta

*How many students failed with frequency in each uc each year? Show the code, initials, year and number of students who failed with frequency.*

### 4.6.2 Resposta

#### 4.6.2.1 Base de Dados Relacional

```

select u.codigo, u.sigla_uc, o.ano_letivo, (o.com_frequencia - o.aprovados)
reprovados_com_frequencia
from xucs u

```

```

join xocorrencias o on u.codigo = o.codigo
where (o.com_frequencia - o.aprovados) > 0;

```

#### 4.6.2.2 Base de Dados Mongo

```

db.ucs.aggregate([
  {
    $lookup:
    {
      from: "ocorrencias",
      localField: "codigo",
      foreignField: "codigo",
      as: "ocorrencias"
    },
  },
  {$unwind: "$ocorrencias"},
  {$project:
  {
    _id: 0,
    codigo: 1,
    "ano_letivo" : "$ocorrencias.ano_letivo",
    "sigla_uc" : 1,
    "reprovados_com_frequencia": {$subtract:
    ["$ocorrencias.com_frequencia" , "$ocorrencias.aprovados" ] }
  }
  },
  {$match: {reprovados_com_frequencia:{$gt : 0}}}
])

```

#### 4.6.3 Resultados

	CODIGO	SIGLA_UC	ANO_LETIVO	REPROVADOS_COM_FREQUENCIA
1	EM536	CNMF	1996/1997	1
2	EM542	IM	1996/1997	5
3	EM546	SST	1996/1997	1
4	EM547	TFT	1996/1997	1
5	EEC312	IM2	1996/1997	31
6	EEC313	LAB4	1996/1997	1
7	EEC414	SAI1	1996/1997	2
8	EEC415	PS	1996/1997	6
9	MEEC2080	CA	1997/1998	1
10	EC134	AM2	1996/1997	69
11	EC133	GM1	1996/1997	76
12	EC135	GM2	1996/1997	58

(...)

Figura 10 - Resultados da pergunta 6 em SQL.

```
/* 1 */
{
  "codigo" : "EC5142",
  "sigla_uc" : "CF",
  "ano_letivo" : "1998/1999",
  "reprovados_com_frequencia" : 1
}

/* 2 */
{
  "codigo" : "EC5142",
  "sigla_uc" : "CF",
  "ano_letivo" : "2000/2001",
  "reprovados_com_frequencia" : 1
}

/* 3 */
{
  "codigo" : "EC5142",
  "sigla_uc" : "CF",
  "ano_letivo" : "2001/2002",
  "reprovados_com_frequencia" : 1
}
```

(...)

Figura 11 - Resultados da pergunta 6 em NoSQL.

#### 4.6.4 Considerações

Escolhemos elaborar a query 6 para testar as funcionalidades de join do MongoDB e para utilizar a tabela de Ocorrências para a qual ainda não tínhamos encontrado uso.

Verificamos que a performance do join é muito pior que a do SQL normal, foi o único caso em que isto aconteceu.

## 5. Comparação Mongo - Oracle

A implementação de base de dados em SQL da Oracle ou em NoSQL do Mongo são duas maneiras diferentes de representar informação que têm pouco em comum.

A maior diferença entre os dois tipos de base de dados é a maneira como tratam a informação. Enquanto que em Oracle a informação é guardada da forma tradicional, com uma estrutura bi-dimensional de linhas e colunas, em Mongo a informação é guardada em documentos sem uma estrutura pré-definida, sendo flexível no tipo de dados que pode guardar.

Relativamente ao tamanho dos dados, a base de dados Oracle permite a junção de várias tabelas facilitando o acesso a qualquer tipo de dados presente na base de dados. Em oposição, não é possível junção de coleções em Mongo. Uma alternativa que permite o acesso a dados de outras coleções é a existência de redundância de dados, guardando

alguns dados em mais que uma coleção o que resulta num aumento do espaço ocupado. Uma maneira de contrariar o aumento de espaço é não incluir nos documentos os atributos cujo valor seja nulo. Caso um atributo não exista num documento, a base de dados Mongo atribui-lhe automaticamente o valor nulo, não sendo necessário a sua inclusão nos documentos.

Em relação ao tempo de processamento, a base de dados Mongo é mais rápida do que a Oracle, especialmente em casos em que estamos a lidar com grandes quantidades de dados. No desenvolvimento deste trabalho as diferenças de tempo de processamento não foram muito notórias.

Quanto à facilidade de elaboração das perguntas, a sintaxe usada em SQL é mais simples (segue sempre a mesma estrutura) e o grupo já estava familiarizado, tornando o processo muito mais rápido. Em Mongo, por haver mais do que uma maneira de fazer perguntas à base de dados (usando "find", "aggregate", etc) e pelo facto de que a base de dados não permite junção de coleções houve maior dificuldade na elaboração das perguntas. Contudo, o tamanho das queries em mongo de uma forma geral é mais pequeno, por exemplo na pergunta 4, a utilização do operador \$first permitiu tornar o tamanho da query bastante mais pequeno, algo que não é possível no SQL.

## 6. Anexos

### 6.1 Função print\_clob()

```
create or replace procedure print_clob( p_clob in clob ) as l_offset number default
1;
begin
  loop exit when l_offset > dbms_lob.getlength(p_clob);

    dbms_output.put_line( dbms_lob.substr( p_clob, 32000, l_offset ) );
    l_offset := l_offset + 32000;

  end loop;
end;
```

### 6.2 Servidor NodeJS Para Inserção dos Documentos JSON na Base de Dados Mongo

```
const express = require("express");
const app = express();
const MongoClient = require("mongodb").MongoClient;
var fs = require('fs');

var ucs_url = "../UCS.json";
var docentes_url = "../DOCENTES.json";
```

```

var ocorrencias_url = "../OCORRENCIAS.json";

var ucs_json;
var docentes_json;
var ocorrencias_json;

app.get("/insertOcorrencias", (req, res) => {

MongoClient.connect('mongodb://andreia:andreia@vdbase.inesctec.pt:27017/andreia?authSource=admin',function(err, db){
    if(err)
        console.log(err);
    else
    {
        console.log('Connected');
        var dbo = db.db("andreia");
        dbo.collection("ocorrencias").insertMany(ocorrencias_json,
function(err, res) {
            if (err) throw err;
            console.log("Ocorrencias document inserted");
            db.close();
        });
    }
});
    return res.send();
});

app.get("/insertDocentes", (req, res) => {

MongoClient.connect('mongodb://andreia:andreia@vdbase.inesctec.pt:27017/andreia?authSource=admin',function(err, db){
    if(err)
        console.log(err);
    else
    {
        console.log('Connected');
        var dbo = db.db("andreia");
        dbo.collection("docentes").insertMany(docentes_json,
function(err, res) {
            if (err) throw err;
            console.log("Docentes document inserted");
            db.close();
        });
    }
});
    return res.send();
});

app.get("/insertUcs", (req, res) => {

MongoClient.connect('mongodb://andreia:andreia@vdbase.inesctec.pt:27017/andreia?authSource=admin',function(err, db){
    if(err)
        console.log(err);

```

```

    else
    {
        console.log('Connected');
        var dbo = db.db("andreia");
        dbo.collection("ucs").insertMany(ucs_json, function(err, res) {
            if (err) throw err;
            console.log("UCS document inserted");
            db.close();
        });
    }
});

return res.send();
});

app.get("/load", (req, res) => {
    ucs_json = JSON.parse(fs.readFileSync(ucs_url, 'utf8'));
    docentes_json = JSON.parse(fs.readFileSync(docentes_url, 'utf8'));
    ocorrencias_json = JSON.parse(fs.readFileSync(ocorrencias_url, 'utf8'));
    return res.send();
});

app.get("/", (req, res) => {
    db =
MongoClient.connect('mongodb://andreia:andreia@vdbase.inesctec.pt:27017/andreia?authSource=admin',function(err, client){
    if(err)
        console.log(err);
    else
    {
        console.log('Mongo Connected');
    }
});
    return res.send();
});

app.listen(3000, () => {
    console.log("Listening on port 3000!");
});

```