



**Universidade do Minho**  
Escola de Engenharia

Universidade do Minho  
Mestrado Integrado em Engenharia Informática

# Administração e Exploração de Base de Dados

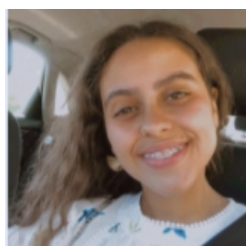
**Grupo 06**

## Trabalho Prático

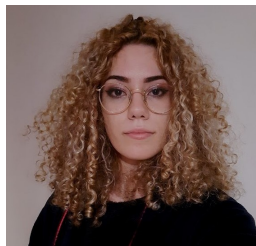
27 Janeiro 2021



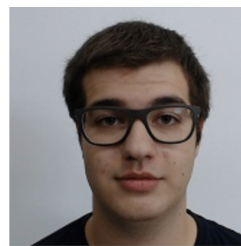
Ana Margarida  
Campos  
(A85166)



Ana Afonso  
(A85762)



Tânia Rocha  
(A85176)



Pedro Silva  
(A82522)

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Base de Dados</b>	<b>4</b>
2.1	Modelação . . . . .	5
2.1.1	Modelo Conceptual . . . . .	5
2.1.2	Modelo Lógico . . . . .	6
2.1.3	Modelo Físico . . . . .	6
2.2	Análise de atributos de cada entidade . . . . .	8
2.2.1	Database . . . . .	8
2.2.2	Sessions . . . . .	9
2.2.3	Users . . . . .	9
2.2.4	Tablespaces . . . . .	9
2.2.5	Datafiles . . . . .	10
2.2.6	Information . . . . .	10
<b>3</b>	<b>Agente de Recolha de Informação</b>	<b>11</b>
3.1	Conexão à Base de Dados . . . . .	11
3.2	Selects . . . . .	12
3.2.1	Tablespaces . . . . .	12
3.2.2	Datafiles . . . . .	12
3.2.3	Sessions . . . . .	12
3.2.4	Users . . . . .	12
3.2.5	Information . . . . .	13
3.2.6	Database . . . . .	13
3.3	Inserts . . . . .	13
<b>4</b>	<b>API REST</b>	<b>14</b>
4.1	Ligação à Base de Dados . . . . .	14
4.2	Armazenamento de Dados . . . . .	15
<b>5</b>	<b>Interface Web</b>	<b>16</b>
<b>6</b>	<b>Conclusão</b>	<b>20</b>

# 1 Introdução

O presente projeto enquadra-se na unidade curricular de Administração e Exploração da Base de Dados, na qual foi proposta a construção de um monitor de base de dados que apresente, de forma simples, os principais parâmetros de avaliação de performance de uma base de dados Oracle.

Numa primeira instância, torna-se imprescindível o estabelecimento de um raciocínio ponderado capaz de promover a elaboração de um fio condutor que se impõe como a estrutura faseada para o desenvolvimento do presente projeto.

Para solucionar da melhor forma o desafio apresentado, será fundamental a criação de uma nova PDB, respetivo schema com tablespaces permanentes e temporários e datafiles associados, a fim de armazenar os dados recolhidos. De seguida, a criação de um agente Java que através das views de administração execute a recolha da informação considerada necessária. Para a ligação à PDB será criada uma API REST que devolva os resultados no formato necessário (JSON).

A fim de garantir a demonstração, visualmente mais apelativa e intuitiva, do cumprimento de todos os requisitos propostos gerar-se-á uma interface web que apresente os dados coletados em fases efetuadas anteriormente.

## 2 Base de Dados

Para este trabalho prático é tido como objetivo a construção de um sistema de monitorização de uma base de dados *Pluggable orcl*. A construção do mesmo estrutura-se em quatro fases distintas, onde numa fase inicial considerou-se importante o planeamento a nível desta base de dados Oracle, tendo em consideração as suas características enquanto software.

Primeiramente, foi necessário compreender a Arquitetura *Multitenant* geral de um sistema Oracle, atendendo aos seus constituintes, nomeadamente a *Root*, a *Seed* e as *PDBs*.

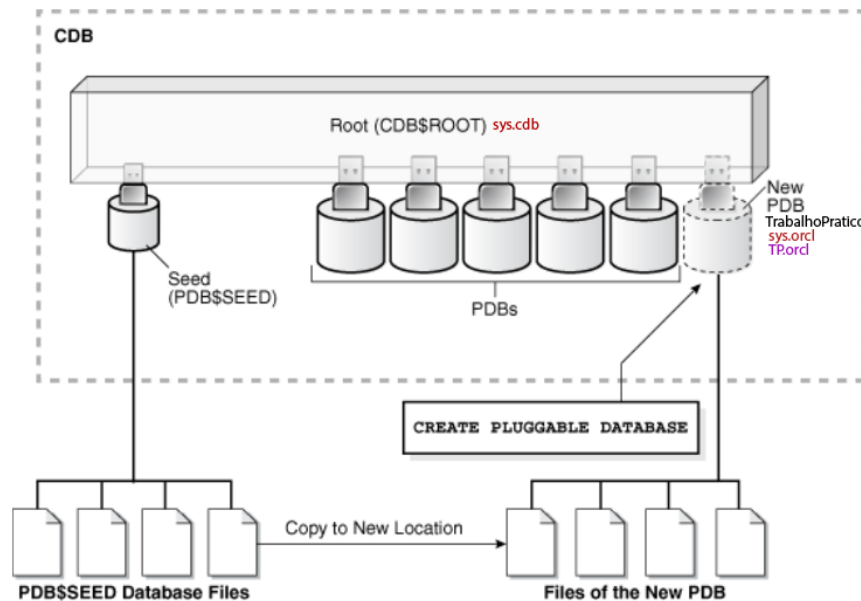


Figura 1: Arquitetura *Multitenant*

Posteriormente, procedeu-se à criação de uma *Pluggable Database* denominada **Trabalho-Pratico** e o respetivo utilizador com capacidades administrativas (**TP**), respetivo *schema* com *Tablespaces* permanentes e temporárias e ainda os *Datafiles* associados.

```
create pluggable database TrabalhoPratico admin user TP identified by tp roles = (
    DBA) FILE_NAME_CONVERT=('/u02/app/oracle/oradata/ORCL/pdbseed', '/u02/app/
    oracle/oradata/ORCL/TrabalhoPratico');

create temporary tablespace tp_temp tempfile '/u02/app/oracle/oradata/ORCL/
    tp_temp01.dbf' SIZE 100M;

create tablespace tp_tab datafile '/u02/app/oracle/oradata/ORCL/job_tab.dbf' SIZE
    100M;
```

## 2.1 Modelação

Numa primeira etapa de modelação, torna-se a imprescindível a elaboração de modelos que não só constituem uma base para todo o projeto, como também facilitam a compreensão e leitura dos dados.

Aquando do desenvolvimento da modelação, foram realizados *selects* com o objetivo de confirmar se seria possível aceder a todas as informações que constituíam o modelo deliberadamente construído.

Desta forma, desenvolveram-se os modelos conceptual e lógico que permitem uma visualização mais intuitiva de todos os elementos considerados pertinentes e, conseqüentemente, a posterior geração do modelo físico.

### 2.1.1 Modelo Conceptual

O modelo conceptual é capaz de fornecer uma descrição de alto nível de como um sistema está organizado e funciona.

Esta modelação consiste no processo de construção de um modelo que permite relacionar informação independentemente da implementação.

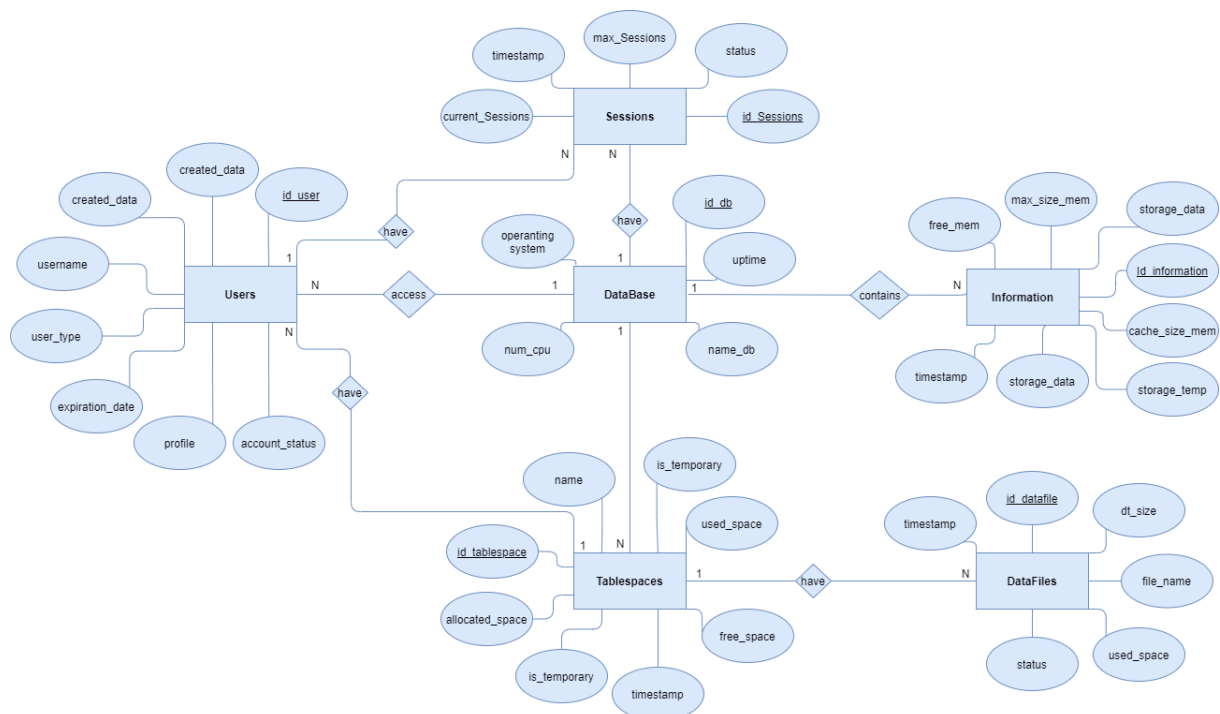


Figura 2: Modelo Conceptual

### 2.1.2 Modelo Lógico

A conceção do modelo lógico apoia-se na implementação do modelo conceptual realizada à priori.

Apesar das notáveis semelhanças, este modelo tem em consideração algumas limitações e ainda implementa recursos, nomeadamente a adequação de padrão e nomenclatura, a definição de chaves primárias e estrangeiras, a normalização, a integridade referencial, entre outras.

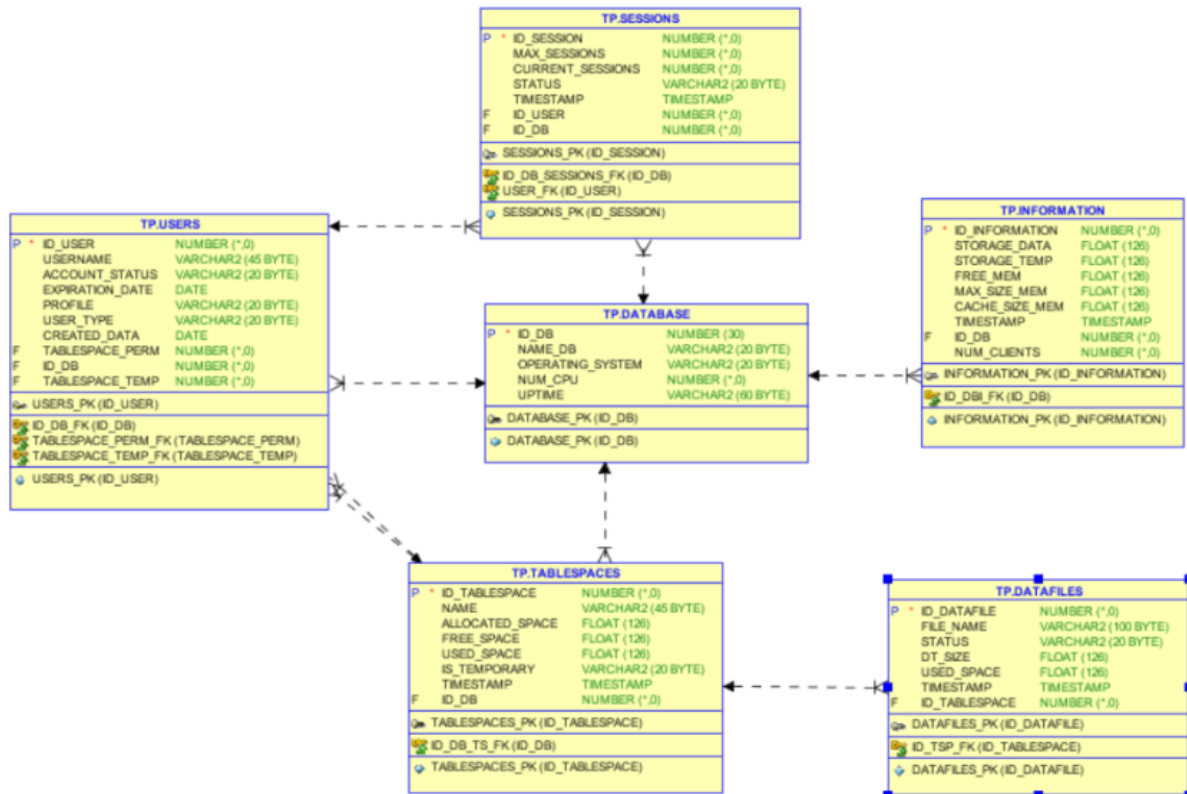


Figura 3: Modelo Lógico

### 2.1.3 Modelo Físico

No modelo físico é efetuada a modelação física do modelo da base de dados.

Neste caso, tem-se em consideração as limitações impostas pelo SGBD escolhido e deve ser criado sempre com base nos exemplos de modelação dos dados produzidos no modelo lógico.

```

CREATE TABLE database (
    id_db          NUMBER(30) NOT NULL,
    name_db        VARCHAR2(20 BYTE),
    operating_system VARCHAR2(20 BYTE),
    num_cpu        NUMBER(*,0),
    uptime         VARCHAR2(60 BYTE)
)
PCTFREE 10 PCTUSED 40 TABLESPACE tp_tables LOGGING
  STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
    2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT )
NO INMEMORY;

CREATE TABLE datafiles (
    id_datafile    NUMBER(*,0) NOT NULL,
    file_name      VARCHAR2(100 BYTE),
    status         VARCHAR2(20 BYTE),
    dt_size        FLOAT(126),
    used_space     FLOAT(126),
    timestamp      TIMESTAMP,
    id_tablespace  NUMBER(*,0)
)
PCTFREE 10 PCTUSED 40 TABLESPACE tp_tables LOGGING
  STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
    2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT )
NO INMEMORY;

CREATE TABLE information (
    id_information  NUMBER(*,0) NOT NULL,
    storage_data    FLOAT(126),
    storage_temp    FLOAT(126),
    free_mem        FLOAT(126),
    max_size_mem    FLOAT(126),
    cache_size_mem  FLOAT(126),
    timestamp       TIMESTAMP,
    id_db           NUMBER(*,0),
    num_clients     NUMBER(*,0)
)
PCTFREE 10 PCTUSED 40 TABLESPACE tp_tables LOGGING
  STORAGE ( PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1
    FREELIST GROUPS 1 BUFFER_POOL DEFAULT )
NO INMEMORY;

CREATE TABLE sessions (
    id_session      NUMBER(*,0) NOT NULL,
    max_sessions    NUMBER(*,0),
    current_sessions NUMBER(*,0),
    status          VARCHAR2(20 BYTE),
    timestamp       TIMESTAMP,
    id_user         NUMBER(*,0),
    id_db           NUMBER(*,0)
)
PCTFREE 10 PCTUSED 40 TABLESPACE tp_tables LOGGING
  STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
    2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT )
NO INMEMORY;

```

```

\lstset{style=sqlstyle}
CREATE TABLE tablespaces (
    id_tablespace    NUMBER(*,0) NOT NULL,
    name             VARCHAR2(45 BYTE),
    allocated_space   FLOAT(126),
    free_space        FLOAT(126),
    used_space        FLOAT(126),
    is_temporary      VARCHAR2(20 BYTE),
    timestamp         TIMESTAMP,
    id_db             NUMBER(*,0)
)
PCTFREE 10 PCTUSED 40 TABLESPACE tp_tables LOGGING
    STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS
    2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT )
NO INMEMORY;

CREATE TABLE users (
    id_user          NUMBER(*,0) NOT NULL,
    username         VARCHAR2(45 BYTE),
    account_status   VARCHAR2(20 BYTE),
    expiration_date   DATE,
    profile          VARCHAR2(20 BYTE),
    user_type        VARCHAR2(20 BYTE),
    created_data      DATE,
    tablespace_perm   NUMBER(*,0),
    id_db            NUMBER(*,0),
    tablespace_temp   NUMBER(*,0)
)
PCTFREE 10 PCTUSED 40 TABLESPACE tp_tables LOGGING
    STORAGE ( PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1
    FREELIST GROUPS 1 BUFFER_POOL DEFAULT )
NO INMEMORY;

```

## 2.2 Análise de atributos de cada entidade

De forma a melhorar a compreensão relativa a cada uma das entidades e respetivos atributos presentes no modelo, foram desenvolvidas tabelas que transparecem detalhadamente as informações dos mesmos.

### 2.2.1 Database

Após uma minuciosa análise a todas as informações úteis e pertinentes da base de dados *Pluggable orcl*, foram recolhidos dados considerados necessários e, posteriormente, convertidos na forma de atributos da entidade **Database** na tabela que se segue.

DATABASE	TIPO	DESCRIÇÃO
ID_DB	NUMBER(30)	Identificador único da base de dados
NAME_DB	VARCHAR2(20 BYTES)	Nome da base de dados
OPERATING_SYSTEM	VARCHAR2(20 BYTES)	Sistema operativo que contém a base de dados
NUM_CPU	NUMBER(*,0)	Número de CPUs
UPTIME	VARCHAR2(20 BYTES)	Tempo de atividade da base de dados

Figura 4: Atributos e respetivo tipo e descrição da tabela Database



### 2.2.2 Sessions

A entidade **Sessions** foi criada com o intuito de armazenar informação relevante relativa às sessões ativas/inativas de cada utilizador. Os atributos que a constituem encontram-se na tabela seguinte.

SESSIONS	TIPO	DESCRIÇÃO
ID_SESSION	NUMBER(*,0)	Identificador único da sessão
MAX_SESSION	NUMBER(*,0)	Número máximo de sessões
CURRENT_SESSIONS	NUMBER(*,0)	Número de sessões atuais
STATUS	VARCHAR2(20 BYTES)	Estado do utilizador
TIMESTAMP	TIMESTAMP	Timestamp da recolha dos dados

Figura 5: Atributos e respetivo tipo e descrição da tabela Sessions

### 2.2.3 Users

De maneira a ser possível representar os utilizadores da base de dados Oracle, foi criada a entidade **Users**. Esta contém todas as informações importantes e necessárias que permitem retratar/caracterizar cada um dos utilizadores.

USERS	TIPO	DESCRIÇÃO
ID_USER	NUMBER(*,0)	Identificador único do utilizador
USERNAME	VARCHAR2(45 BYTES)	Nome do utilizador
ACCOUNT_STATUS	VARCHAR2(20 BYTES)	Estado do utilizador
EXPIRATION_DATA	DATE	Data de expiração do utilizador
PROFILE	VARCHAR2(20 BYTES)	Tipo de perfil
USER_TYPE	VARCHAR2(20 BYTES)	Tipo de utilizador
CREATED_DATA	DATE	Data de criação do utilizador
TABLESPACE	NUMBER(*,0)	Timestamp da recolha dos dados

Figura 6: Atributos e respetivo tipo e descrição da tabela Users

### 2.2.4 Tablespaces

A criação da entidade **Tablespaces** foi imprescindível para a elaboração do trabalho, uma vez que esta permite representar cada uma das *tablespaces* associadas à *Root*. Os atributos considerados fundamentais encontram-se apresentados na seguinte tabela.

TABLESPACES	TIPO	DESCRIÇÃO
ID_TABLESPACES	NUMBER	Identificador único da tablespace
NAME	VARCHAR2(45 BYTES)	Nome da tablespace
ALLOCATED_SPACE	FLOAT(126)	Espaço alocado da tablespace
FREE_SPACE	FLOAT(126)	Espaço livre na tablespace
USED_SPACE	FLOAT(126)	Espaço utilizado da tablespace
IS_TEMPORARY	VARCHAR2(45 BYTES)	Distinção entre tipos de tablespaces (temporária, permantete e undo)
TIMESTAMP	TIMESTAMP	Timestamp da recolha dos dados

Figura 7: Atributos e respetivo tipo e descrição da tabela Tablespaces

### 2.2.5 Datafiles

A entidade **Datafile** tem como propósito a representação de cada *datafile* associado a cada uma das *tablespaces*. Na figura seguinte são visíveis todos os atributos considerados significativos que permitem ilustrar um *datafile*.

DATAFILES	TIPO	DESCRIÇÃO
ID_DATAFILE	NUMBER(*,0)	Identificador único de um datafile
FILE_NAME	VARCHAR2(100 BYTES)	Nome do datafile
STATUS	VARCHAR2(20 BYTES)	Estado do datafile
DT_SIZE	FLOAT(126)	Tamanho do datafile
USED_SPACE	FLOAT(126)	Espaço utilizado do datafile
TIMESTAMP	TIMESTAMP	Timestamp da recolha dos dados

Figura 8: Atributos e respetivo tipo e descrição da tabela Datafiles

### 2.2.6 Information

A fim de armazenar variadas informações, nomeadamente dados relativos à memória foi concebida a entidade **Information**. De seguida são apresentados em detalhe cada um dos seus atributos.

INFORMATION	TIPO	DESCRIÇÃO
ID_INFORMATION	NUMBER(*,0)	Identificador único da informação
STORAGE_DATA	FLOAT(126)	Espaço da base de dados
STORAGE_TEMP	FLOAT(126)	Espaço temporário da base de dados
FREE_MEM	FLOAT(126)	Memória livre
MAX_SIZE_MEM	FLOAT(126)	Tamanho de memória máxima permitida
CACHE_SIZE_MEM	FLOAT(126)	Tamanho da cache
TIMESTAMP	TIMESTAMP	Timestamp da recolha dos dados

Figura 9: Atributos e respetivo tipo e descrição da tabela Information

### 3 Agente de Recolha de Informação

Com o propósito de implementar o agente que, através das *views* de administração, executa a recolha da informação considerada necessária, foi desenvolvido um programa na linguagem de programação Java.

Inicialmente, e de modo a implementar o programa pretendido, foi necessário estabelecer a conexão ao servidor da base de dados. Para tal recorreu-se à ferramenta *JDBC - Java Database Connectivity*. Esta permitiu a comunicação com ambos os sistemas de base de dados, a *Pluggable Database TrabalhoPratico* e a *Root*.

#### 3.1 Conexão à Base de Dados

O excerto de código que se segue demonstra o estabelecimento da conexão à *Root* e à *Pluggable Database*.

```
public static final String DB_DRIVER = "oracle.jdbc.driver.OracleDriver";
public static final String DB_CONNECTION_ROOT = "jdbc:oracle:thin:@127.0.0.1:1521/
orclpdb1.localdomain";
public static final String DB_CONNECTION_PLUG = "jdbc:oracle:thin:@127.0.0.1:1521/
orclpdb1.localdomain";
public static final String DB_USER = "system";
public static final String DB_PASSWORD = "Oradoc.db1";
public static final String DB_USER_PDB = "TP";
public static final String DB_PASSWORD_PDB = "tp";

public static Connection getConexao(String conn, String user, String pw) {
    Connection oc = null;
    try {
        Class.forName(DB_DRIVER);
    } catch (ClassNotFoundException e) {
        System.out.println("Error driver JDBC: "+e.getMessage());
    }

    try {
        oc = DriverManager.getConnection(conn, user, pw);
        return oc;
    } catch (SQLException e) {
        System.out.println("Cannot open connection: "+e.getMessage());
    }

    return oc;
}

public static Connection getConexao_root() {
    return getConexao(DB_CONNECTION_ROOT, DB_USER, DB_PASSWORD);
}

public static Connection getConexao_plug() {
    return getConexao(DB_CONNECTION_PLUG, DB_USER_PDB, DB_PASSWORD_PDB);
}
```

## 3.2 Selects

A conexão com a *Root* permitiu a recolha de todos os dados fundamentais através do envio de comandos SQL, designadamente *queries* do tipo *select*.

De seguida são apresentados todos os *selects* desenvolvidos que possibilitam a obtenção de todas as informações consideradas pertinentes para o posterior povoamento do nosso modelo. Estes *selects* foram implementados de forma a devolverem os dados pretendidos para cada tabela.

### 3.2.1 Tablespaces

#### — Tablespaces Permanentes

```
SELECT A.tablespace_name tablespace_name , trunc(round(b.free/1024/1024)) free_mb ,
trunc(a.bytes/1024/1024) allocated_mb ,(trunc(a.bytes/1024/1024) - trunc(round(
b.free/1024/1024))) Used_Space_MB ,(SELECT contents FROM dba_tablespaces where
a.tablespace_name=tablespace_name) isTemporary FROM (SELECT tablespace_name ,
bytes , decode(autoextensible , 'YES',maxbytes,bytes) maxsize FROM dba_data_files
GROUP BY file_id , file_name , tablespace_name , autoextensible , bytes , decode(
autoextensible , 'YES',maxbytes,bytes)) a,(SELECT file_id ,tablespace_name ,sum(
bytes) free FROM dba_free_space GROUP BY file_id , tablespace_name) b WHERE A.
tablespace_name=b.tablespace_name ORDER BY A.tablespace_name ASC;
```

#### — Tablespaces Temporarias

```
SELECT TABLESPACE_NAME, trunc(ALLOCATED_SPACE/1024/1024) Allocated ,trunc(
FREE_SPACE/1024/1024) Free , trunc(TABLESPACE_SIZE/1024/1024 - FREE_SPACE
/1024/1024) Used , 'TEMPORARY' Is_temporary from DBA_TEMP_FREE_SPACE;
```

### 3.2.2 Datafiles

#### — Datafiles Permanentes

```
SELECT df.File_Name , df.File_ID , df.TablesPace_Name , Round(df.Bytes/1024/1024,2)
Size_MB , df.status , e.used_bytes/1024/1024 Used_MB from dba_data_files df, (
SELECT file_id , Sum(Decode(bytes,NULL,0,bytes)) used_bytes FROM dba_extents
GROUP by file_id) E WHERE e.file_id (+) = df.file_id;
```

#### — Datafiles Temporarios

```
SELECT b.file_name ,a.file_id ,a.tablespace_name ,b.bytes/1024/1024,b.status ,a.
bytes_used/1024/1024 FROM v$temp_extent_pool a, dba_temp_files b where a.
tablespace_name (+)=b.tablespace_name;
```

### 3.2.3 Sessions

```
SELECT a.SID isSessao , b.value MaxSessions , (SELECT count(*) from v$session a ,
v$process b where type= 'USER' and a.Creator_ADDR = b.ADDR) NumSessoesAtuais
,a.User# idUser , a.status STATUS from v$session a , v$parameter b WHERE b.name
= 'sessions'
```

### 3.2.4 Users

```
SELECT USERNAME, USER_ID, ACCOUNT.STATUS, CREATED,PROFILE,Common,
DEFAULT.TABLESPACE, TEMPORARY.TABLESPACE, EXPIRY.DATE from dba_users A order
by created DESC;
```

### 3.2.5 Information

```
SELECT count(distinct username) FROM v$session WHERE username IS NOT NULL ORDER BY
username ASC;

SELECT sum(Bytes)/1024/1024 FROM USER_TABLESPACES a, dba_data_files b where
Contents = 'PERMANENT' and a.TABLESPACE_NAME = b.TABLESPACE_NAME;

SELECT sum(A.TABLESPACE_SIZE)/1024/1024 Temporary_Storage_MB from
DBA_TEMP_FREE_SPACE A;

SELECT Name, Bytes/1024/1024 SGAMaxSize_MB from v$sgainfo where Name='Maximum SGA
Size' or Name='Buffer Cache Size' or Name='Free SGA Memory Available' order by
SGAMAXSIZE_MB asc
```

### 3.2.6 Database

```
SELECT a.DBID, a.NAME nomeBD, a.PLATFORM_NAME SO, b.CPU_CORE_COUNT_CURRENT, (
    floor(sysdate - c.startup_time) || ' days(s) ' || trunc( 24*((sysdate-c.
    startup_time) - trunc(sysdate-c.startup_time))) || ' hour(s) ' || mod(trunc
    (1440*((sysdate-c.startup_time) - trunc(sysdate-c.startup_time))), 60) || '
    minute(s) ' || mod(trunc(86400*((sysdate-c.startup_time) - trunc(sysdate-c.
    startup_time))), 60) || ' seconds') uptime FROM V$DATABASE a , V$LICENSE b ,
v$instance c;
```

## 3.3 Inserts

A inserção de todos os dados recolhidos na base de dados criada para o efeito (**TrabalhoPratico**) foi realizada a partir do desenvolvimento de *queries* do tipo *insert* e, desta forma, todos os dados foram inseridos nas entidades correspondentes.

Adicionalmente, foram criados *statements* que estabelecem conexão com a *Root* e, com os mesmos, onde foram executadas as *queries* apresentadas na secção anterior (através do comando *executeQuery*). Os resultados obtidos foram guardados num *resultSet* que é posteriormente iterado com o objetivo de construir os *inserts* a serem executados. Por fim, é estabelecida uma conexão com a *Pluggable Database* e os dados são inseridos através de um *executeUpdate*.

Devido à existência de chaves estrangeiras na base de dados a inserção teve em consideração a ordem pela qual os dados estavam a ser inseridos nas tabelas. Como tal, as primeiras tabelas a serem povoadas são aquelas que não possuem chaves estrangeiras dependentes.

## 4 API REST

A terceira etapa deste projeto consiste na criação de uma API REST, isto é, uma interface de programação de aplicações que segue conformidade com as restrições da arquitetura REST. Este género de arquitetura permite dar resposta à solicitação de uma recurso através da transmissão de uma representação do estado desse mesmo recurso, a partir de vários formatos via HTTP. Neste caso, a informação a ser transmitida abrange os resultados obtidos da ligação com a PBD, que são enviados a partir do formato JSON.

Para a implementação de todo este processo foi usada a ferramenta *NodeJS*.

### 4.1 Ligação à Base de Dados

A fim de estabelecer a conexão com uma base de dados Oracle, instalou-se o módulo **oracledb** existente na documentação **npm**. A sua utilização baseou-se na definição de *connection strings* e no uso do *package Instant Client*. Após a conexão ser efetuada a partir dos recursos anteriormente explicitados, são realizados *selects* à base de dados **TrabalhoPratico** e os dados são armazenados num ficheiro JSON estruturado para o efeito.

```
oracledb.initOracleClient({
  libDir: "C:\\\\oracle\\\\instantclient",
});
module.exports.getDb = function () {
  oracledb.getConnection(
    { user: "TP",
      password: "tp",
      connectString:
        "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(
        CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=orclpdb1.localdomain))",
    },
    function (err, connection) {
      if (err) {
        console.error(err.message);
        return;
      }
      connection
        .execute(databquery, [], {
          outFormat: oracledb.OBJECT,
        })
        .then((dados) => {
          let datab;
          fs.readFile("oracle.json", (err, data) => {
            if (err) throw err;
            datab = JSON.parse(data);
            for (var key in datab) {
              if (key === "Database") datab[key] = dados.rows;
            }
            fs.writeFileSync(
              "oracle.json",
              JSON.stringify(datab),
              function (erro) {
                if (erro) throw erro;
                console.log("complete");
              }
            );
          });
        })
        .catch((err) => {
          console.log(err, doRelease(connection));
        });
    }
  );
};
```

## 4.2 Armazenamento de Dados

Como referido anteriormente, foram armazenadas, em formato JSON, todas as informações necessárias que são acessíveis a partir da ferramenta **json-server**.

A imagem que se segue demonstra a ativação do servidor, que possibilita o acesso aos dados em formato JSON.

```
C:\Users\comta\OneDrive\Ambiente de Trabalho\4ano\TPAEBD\AEBD_PROJECT\API>json-server --watch -p 3002 oracle.json

\(^_^)/ hi!

Loading oracle.json
Done

Resources
http://localhost:3002/Tablespaces
http://localhost:3002/Datafiles
http://localhost:3002/Users
http://localhost:3002/Database
http://localhost:3002/Sessions
http://localhost:3002/Information

Home
http://localhost:3002

Type s + enter at any time to create a snapshot of the database
Watching...
```

Figura 10: Ativação do servidor

Na figura seguinte é visível um exemplo da informação apresentada no formato pretendido.

```
← → ↻ ⓘ localhost:3002/Tablespaces

[
  {
    "ID_TABLESPACE": 1,
    "NAME": "AEBD_TABLES",
    "ALLOCATED_SPACE": 500,
    "FREE_SPACE": 308,
    "USED_SPACE": 192,
    "IS_TEMPORARY": "PERMANENT",
    "TIMESTAMP": "2021-01-23T22:42:52.040Z",
    "ID_DB": 2722566360
  },
  {
    "ID_TABLESPACE": 2,
    "NAME": "AEBD_TEMP",
    "ALLOCATED_SPACE": 50,
    "FREE_SPACE": 49,
    "USED_SPACE": 1,
    "IS_TEMPORARY": "PERMANENT",
    "TIMESTAMP": "2021-01-23T22:42:52.040Z",
    "ID_DB": 2722566360
  },
  {
    "ID_TABLESPACE": 3,
    "NAME": "HR_TABLES",
    "ALLOCATED_SPACE": 500,
    "FREE_SPACE": 497,
    "USED_SPACE": 3,
    "IS_TEMPORARY": "PERMANENT",
    "TIMESTAMP": "2021-01-23T22:42:52.040Z",
    "ID_DB": 2722566360
  },
  {
    "ID_TABLESPACE": 4,
    "NAME": "SYS_AUX",
    "ALLOCATED_SPACE": 640,
    "FREE_SPACE": 36,
    "USED_SPACE": 604,
    "IS_TEMPORARY": "PERMANENT",
    "TIMESTAMP": "2021-01-23T22:42:52.040Z",
    "ID_DB": 2722566360
  },
  {
    "ID_TABLESPACE": 5,
    "NAME": "SYSTEM",
    "ALLOCATED_SPACE": 280,
    "FREE_SPACE": 4,
    "USED_SPACE": 276,
    "IS_TEMPORARY": "PERMANENT",
    "TIMESTAMP": "2021-01-23T22:42:52.040Z",
    "ID_DB": 2722566360
  },
  {
    "ID_TABLESPACE": 6,
    "NAME": "TD_TABLES",
    "ALLOCATED_SPACE": 500,
    "FREE_SPACE": 499,
    "USED_SPACE": 1,
    "IS_TEMPORARY": "PERMANENT",
    "TIMESTAMP": "2021-01-23T22:42:52.040Z",
    "ID_DB": 2722566360
  }
],
```

Figura 11: Informação em formato JSON

## 5 Interface Web

A última fase do projeto consistiu na implementação de uma interface *web*. Esta foi elaborada de modo a que fossem apresentadas todas as informações e resultados recolhidos nas fases anteriores, permitindo assim uma visualização mais apelativa e agradável de todos os dados.

De seguida é exibido o resultado final da implementação da interface *web*.

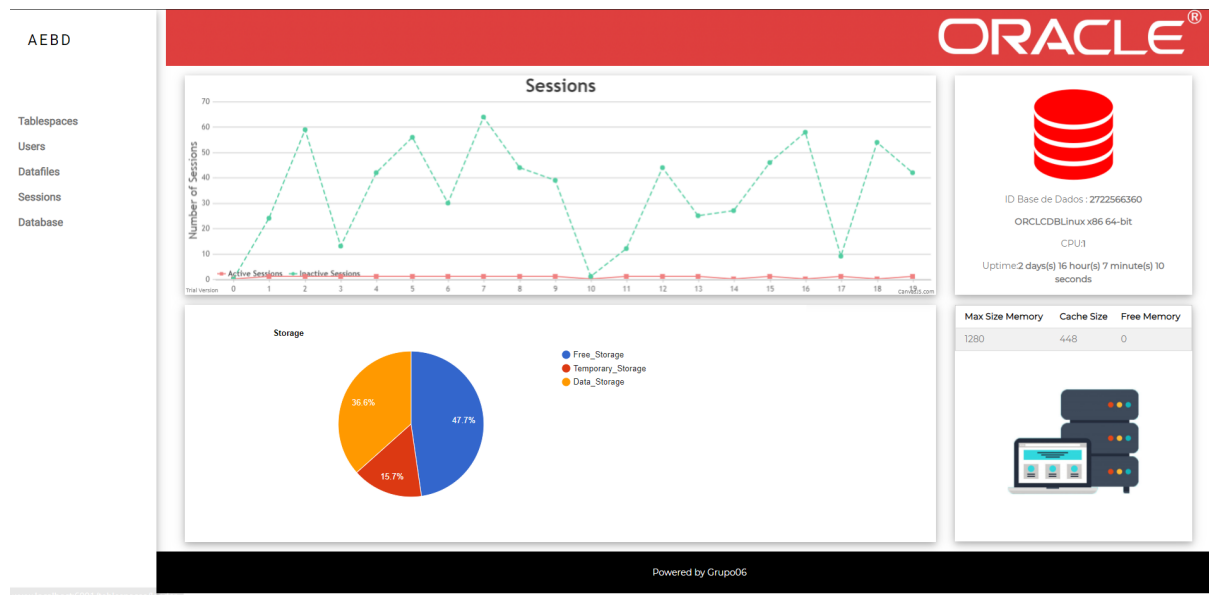


Figura 12: Representação visual da página inicial

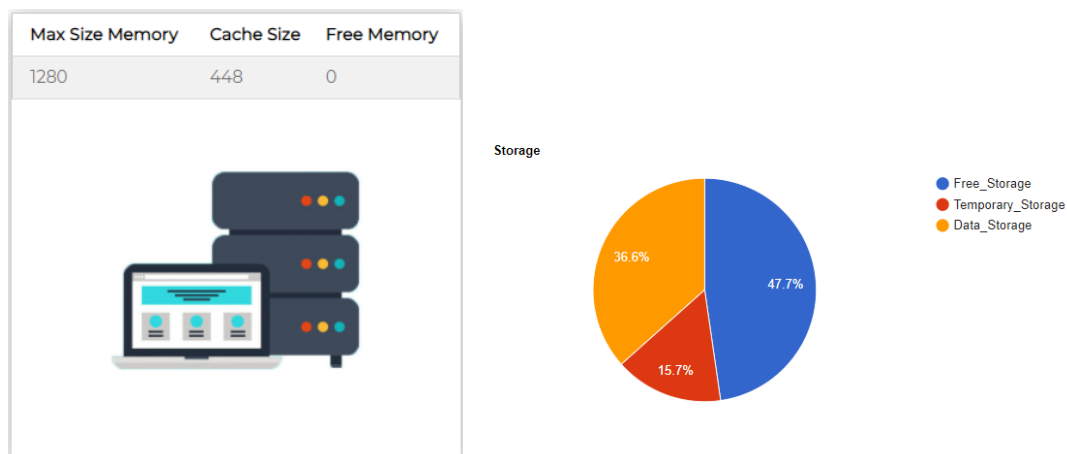


Figura 13: Dados de memória e *storage*



## Tablespaces

Tablespace ID	Tablespace Name	Allocated Space	Free Space	Used Space	Type	Timestamp
1164	AEBD_TABLES	500	308	192	PERMANENT	2021-01-27T16:25:28.040Z
1165	AEBD_TEMP	50	49	1	PERMANENT	2021-01-27T16:25:28.040Z
1166	HR_TABLES	500	497	3	PERMANENT	2021-01-27T16:25:28.040Z
1167	SYS_AUX	640	32	608	PERMANENT	2021-01-27T16:25:28.040Z
1168	SYSTEM	280	4	276	PERMANENT	2021-01-27T16:25:28.040Z
1169	TP_TABLES	500	497	3	PERMANENT	2021-01-27T16:25:28.040Z
1170	UNDOTBS1	180	99	81	UNDO	2021-01-27T16:25:28.040Z
1171	USERS	10	9	1	PERMANENT	2021-01-27T16:25:28.040Z
1172	HR_TEMP	2	499	1	TEMPORARY	2021-01-27T16:25:28.040Z
1173	TEMP	6	63	1	TEMPORARY	2021-01-27T16:25:28.040Z
1174	TP_TEMP	3	499	1	TEMPORARY	2021-01-27T16:25:28.040Z

Figura 14: Representação alusiva às informações relativas às Tablespaces

## Users

User Id	User Name	Account Status	Expiration Date	Profile	User Type	Created Data	Default Tablespace	Temporary Tablespace
123	JOB	OPEN		DEFAULT	NO	2021-01-08T00:00:00.000Z	42	25
122	C##JOB	OPEN		DEFAULT	YES	2021-01-08T00:00:00.000Z	43	30
121	TP	OPEN		DEFAULT	NO	2020-11-21T00:00:00.000Z	44	28
120	SH	OPEN		DEFAULT	NO	2020-11-21T00:00:00.000Z	43	23
119	FICHA04	OPEN		DEFAULT	NO	2020-11-21T00:00:00.000Z	43	23
118	HR	OPEN		DEFAULT	NO	2020-11-20T00:00:00.000Z	42	25
117	C##GRUPO6	OPEN		DEFAULT	YES	2020-11-20T00:00:00.000Z	43	30
114	NOVOUSER	OPEN		DEFAULT	NO	2020-11-20T00:00:00.000Z	43	30
113	C##1	OPEN		DEFAULT	YES	2020-11-20T00:00:00.000Z	43	30
112	TESTE	OPEN		DEFAULT	NO	2020-11-06T00:00:00.000Z	42	25
111	SYS1	OPEN		DEFAULT	NO	2020-10-30T00:00:00.000Z	43	30
106	FLows_FILES	EXPIRED & LOCKED		DEFAULT	YES	2017-03-02T00:00:00.000Z	43	26
108	APEX_050000	EXPIRED & LOCKED		DEFAULT	YES	2017-03-02T00:00:00.000Z	43	26
107	APEX_PUBLIC_USER	EXPIRED & LOCKED		DEFAULT	YES	2017-03-02T00:00:00.000Z	43	30
1279990	DVSYs	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	26
103	DVF	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	26
101	LBACSYS	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	27
98	SPATIAL_CSW_ADMIN_USR	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	30
96	MDDATA	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	30
93	OLAPSYS	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	26
90	MDSYS	EXPIRED & LOCKED		DEFAULT	YES	2017-01-26T00:00:00.000Z	43	26

Figura 15: Representação alusiva às informações relativas às Users

## Datafiles

Datafile Id	File Name	Status	Datafile Size	Used Space	Timestamp	Tablespace Id
1025	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/eebd_tables_01.dbf	AVAILABLE	500	190.625	2021-01-27T16:25:20.040Z	23
1026	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/system01.dbf	AVAILABLE	280	274.75	2021-01-27T16:25:20.040Z	27
1027	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/HR_FILES_01.dbf	AVAILABLE	500	2.375	2021-01-27T16:25:20.040Z	25
1028	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/TP_FILES_01.dbf	AVAILABLE	500	1.625	2021-01-27T16:25:20.040Z	28
1029	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/sysaux01.dbf	AVAILABLE	640	607.0625	2021-01-27T16:25:20.040Z	26
1030	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/undotbs01.dbf	AVAILABLE	180	80.3125	2021-01-27T16:25:20.040Z	29
1031	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/users01.dbf	AVAILABLE	10	0	2021-01-27T16:25:20.040Z	30
1032	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/eebd_temp_01.dbf	AVAILABLE	50	0	2021-01-27T16:25:20.040Z	24
1033	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/HR_TEMP_01.dbf	ONLINE	500	0	2021-01-27T16:25:20.040Z	42
1034	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/TP_TEMP_01.dbf	ONLINE	500	0	2021-01-27T16:25:20.040Z	44
1035	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/temp012017-03-02_07-54-38-075-AM.dbf	ONLINE	64	0	2021-01-27T16:25:20.040Z	43

Figura 16: Representação alusiva às informações relativas aos Datafiles

## Sessions

Session Id	Maximum Sessions	Current Sessions	Status	Timestamp	User Id
7988	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7987	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7986	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7985	472	26	INACTIVE	2021-01-27T16:25:19.040Z	9
7984	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7983	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7982	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7981	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7980	472	26	ACTIVE	2021-01-27T16:25:19.040Z	0
7979	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7978	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7977	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7976	472	26	ACTIVE	2021-01-27T16:25:19.040Z	0
7975	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7974	472	26	ACTIVE	2021-01-27T16:25:19.040Z	9
7973	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7972	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7971	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7970	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7969	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121
7968	472	26	INACTIVE	2021-01-27T16:25:19.040Z	121

Figura 17: Representação alusiva às informações relativas às Sessions

## Database

Database Id	Database Name	Operating System	Number CPU	Uptime
2722566360	ORCLCDB	Linux x86 64-bit	1	2 days(s) 16 hour(s) 7 minute(s) 10 seconds

Figura 18: Representação alusiva às informações relativas à Database

As figuras que se seguem são representativas do antes e depois do aumento da carga na base de dados, isto tornou-se possível com recurso ao *SwingBench*.

Deve-se ter em conta que, vários dos picos de sessões que são observáveis nos gráficos devem-se aos selects à *Root* e inserts na *PDB*, daí as subidas e descidas das mesmas.

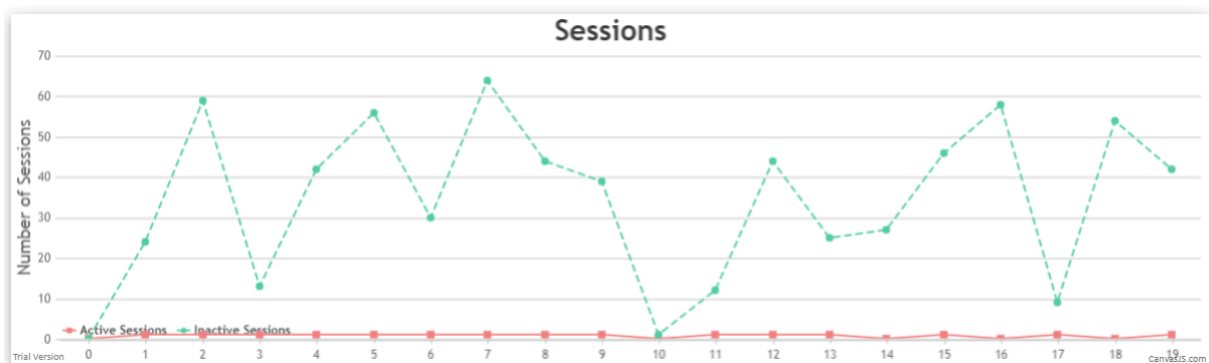


Figura 19: Gráfico representativo das sessões ativas/inativas antes do aumento da carga

Como é claramente visível, após a injeção de carga no sistema verifica-se um aumento do número de sessões.

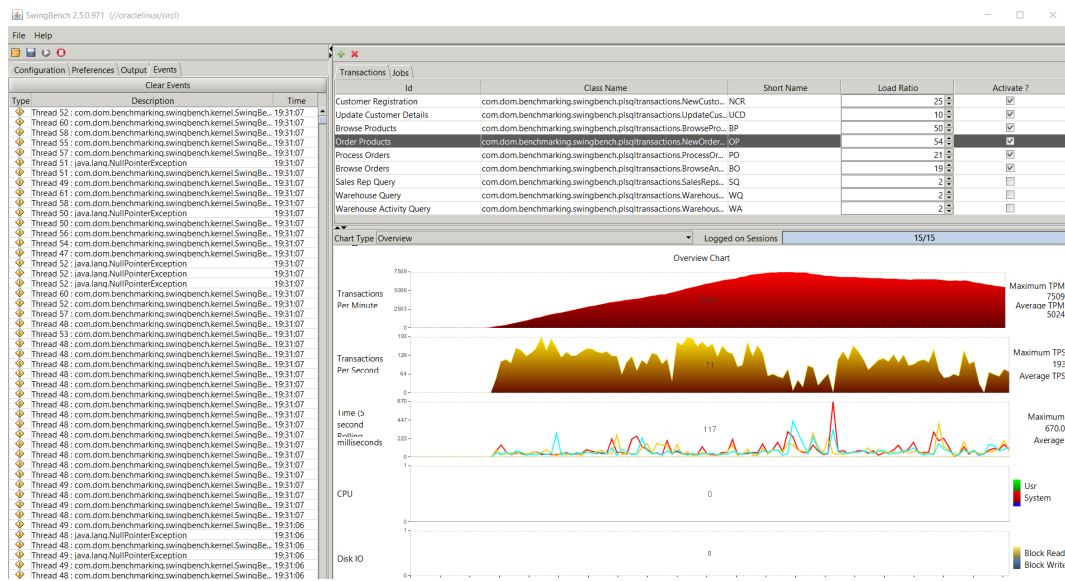


Figura 20: Aumento de carga a partir do *SwingBench*

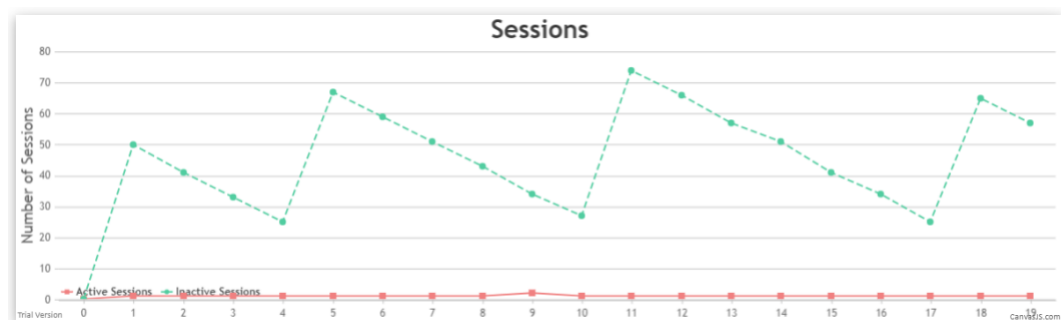


Figura 21: Gráfico representativo das sessões ativas/inativas depois do aumento da carga

Com o intuito de comprovar a veracidade dos valores obtidos, consultou-se o *SQL Developer* e, desta forma, foi possível confirmar, graficamente, que o número de sessões estão semelhantes, tal com é visível na figura seguinte:

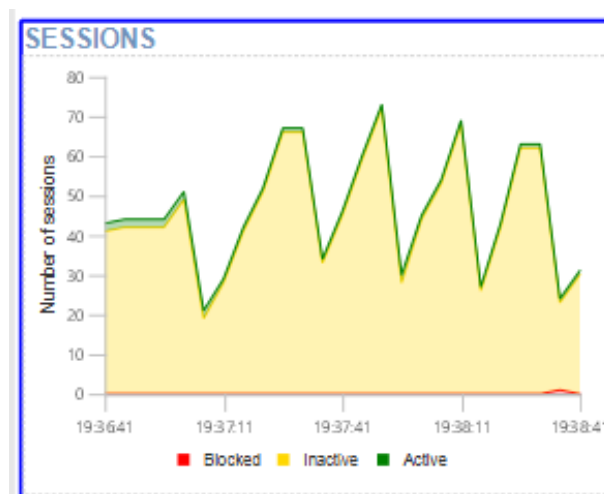


Figura 22: Sessões ativas/inativas depois do aumento da carga

## 6 Conclusão

Em virtude do que foi mencionado, o grupo considera que o projeto desenvolvido vai de encontro ao perspectivado aquando do início minuciosamente ponderado do mesmo.

Durante a realização do projeto, teve-se em consideração o principal objetivo que visa o desenvolvimento de um monitor de base de dados que apresente, de forma simples, os principais parâmetros de avaliação de performance de uma base de dados Oracle.

Este projeto revelou-se bastante enriquecedor, uma vez que incentivou não só a aprendizagem relativa à administração e exploração de uma base de dados, como também mitigou o manuseamento de ferramentas como o *SQL Developer* e *NodeJS* inerentes à elaboração do projeto.

Foi também possível a criação de uma interface *web* cujo principal propósito assenta na apresentação de todas as informações recolhidas. Como tal, foi exequível um *frontend* interativo e de fácil compreensão, tornando assim o programa visualmente mais interessante para os seus utilizadores.

Não obstante a futuros aprimoramentos, concebeu-se código inteligível para promover a vontade e ousadia necessária à realização de posteriores alterações/modificações.

Por fim, compreende-se a utilidade e importância da administração e exploração de bases de dados, visto que as empresas têm cada vez mais gastos com softwares que podem não gerar dados concisos. Assim sendo, a administração de dados é um recurso de muito valor, torna possível a deteção das necessidades de informações futuras e, desta forma, pode melhorar o planeamento das bases de dados para atender às necessidades do mundo empresarial.

## Referências

- [1] Mullins, C. *Database Administration: The Complete Guide to Practices and Procedures*, Addison Wesley Pub, 2002
- [2] Connolly, Thomas M., and Carolyn E. Begg *Database systems: a practical approach to design, implementation, and management*. Pearson Education, 2005
- [3] [https://hpeixoto.me/2020-11-16-performin\\_tasks\\_Oracle\\_multitenant/](https://hpeixoto.me/2020-11-16-performin_tasks_Oracle_multitenant/)