

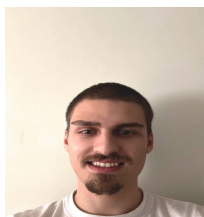


# Universidade do Minho

Mestrado Integrado em Engenharia Informática

## Administração e Exploração de Bases de Dados

### Grupo 7



Gonçalo Borges  
(pg42643)



Mariana Pereira  
(a81146)



Rita Rosendo  
(a84475)



Eduardo Semanas  
(a75536)

27 de janeiro de 2021

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Bases de Dados</b>	<b>4</b>
2.1	Vistas de desempenho . . . . .	4
2.2	Modelo Lógico . . . . .	5
2.2.1	Entidades . . . . .	6
2.2.2	Users . . . . .	8
2.2.3	Sessions . . . . .	9
2.2.4	CPU . . . . .	9
2.2.5	Users <sub>has_roles</sub> . . . . .	10
2.2.6	Profiles . . . . .	10
2.2.7	Memory . . . . .	10
2.2.8	Database <sub>instance</sub> . . . . .	10
2.2.9	Roles . . . . .	11
2.2.10	Datafiles . . . . .	11
2.2.11	Tablespaces . . . . .	11
2.3	Script de povoamento . . . . .	12
<b>3</b>	<b>API Rest</b>	<b>14</b>
3.1	Lógica de controlo . . . . .	14
3.2	Acessos à BD . . . . .	14
<b>4</b>	<b>Interface WEB</b>	<b>16</b>
4.1	Instance . . . . .	16
4.2	CPU . . . . .	17
4.3	Datafile . . . . .	17
4.4	Memory . . . . .	18
4.5	Profiles . . . . .	18
4.6	Roles . . . . .	19
4.7	Sessions . . . . .	20
4.8	TableSpaces . . . . .	21
4.9	Users . . . . .	22
<b>5</b>	<b>Swingbench</b>	<b>23</b>
<b>6</b>	<b>Conclusão</b>	<b>25</b>

## 1 Introdução

Uma base de dados é um simples repositório de dados relacionados com determinado assunto. Ora um administrador de base de dados é responsável por configurar a mesma para garantir o melhor desempenho possível, com mínimo tempo de indisponibilidade, segurança no acesso aos dados e garantias de recuperação em caso de falha humana ou falha de hardware.

Este trabalho prático tem como objectivo monitorizar uma base de dados Oracle, através de diversas métricas disponibilizadas pelo **Oracle** (ex.: utilização de **cpu** por parte da base de dados, utilizadores e as suas permissões respectivas, **tablespaces** presentes na mesma, etc).

Ao longo deste relatório são descritos os processos efetuados e métodos adotados para execução das diferentes etapas propostas para o projeto (criação de uma base de dados **Oracle** capaz de armazenar as informações relativas às métricas anteriormente definidas, implementação de um agente/*script* capaz de obter as informações necessárias para preencher a nossa base de dados a partir de uma outra base de dados **Oracle** de origem, implementação de uma **API rest** capaz de processar os dados presentes na nossa base de dados e de uma interface capaz de apresentar esses mesmos dados de forma perceptível).

## 2 Bases de Dados

Após a criação da nossa Base de Dados Oracle, *TrabalhoPDB*, como ponto de partida utilizamos dados retirados da Base de Dados *SYS*. Efetuamos uma análise às views e coletamos os dados que se adequavam à nossa necessidade.

### 2.1 Vistas de desempenho

Em baixo, apresentam-se as vistas consideradas no nosso trabalho para efetuarmos a nossa monitorização.

Tabela 1: Views e respetivos dados retirados.

Views	Dados
DBA_HIST_DATABASE_INSTANCE	DBID INSTANCE_NUMBER STARTUP_TIME VERSION DB_NAME INSTANCE_NAME PLATFORM_NAME DB_UNIQUE_NAME
DBA_USERS	USERNAME USER_ID ACCOUNT_STATUS EXPIRY_DATE DEFAULT_TABLESPACE TEMPORARY_TABLESPACE CREATED PROFILE COMMON LAST_LOGIN
DBA_DATA_FILES	FILE_NAME FILE_ID TABLESPACE_NAME BYTES AUTOEXTENSIBLE ONLINE_STATUS
DBA_TABLESPACES	TABLESPACE_NAME STATUS CONTENTS SEGMENT_SPACE_MANAGEMENT
DBA_FREE_SPACE	FREE_BYTES FILE_ID

DBA_HIST_SQLSTAT	DBID SQL_ID EXECUTIONS_DELTA DISK_READS_DELTA BUFFER_GETS_DELTA CPU_TIME_DELTA ELAPSED_TIME_DELTA IOWAIT_DELTA APWAIT_DELTA
V\$SGA	NAME VALUE
DBA_HIST_ACTIVE_SESS_HISTORY	SESSION_ID SAMPLE_TIME SQL_ID SQL_OPNAME SQL_PLAN_OPERATION WAIT_CLASS WAIT_TIME SESSION_TYPE SESSION_STATE TIME_WAITED USER_ID
DBA_ROLES	ROLE AUTHENTICATION_TYPE COMMON
DBA_PROFILES	PROFILE RESOURCE_NAME RESOURCE_TYPE LIMIT

## 2.2 Modelo Lógico

Depois de analisarmos as *views* da BD *orclpdb1* procedemos à criação de um Modelo Lógico com os diversos atributos, os seus tipos, e as chaves primárias e estrangeiras de cada tabela.

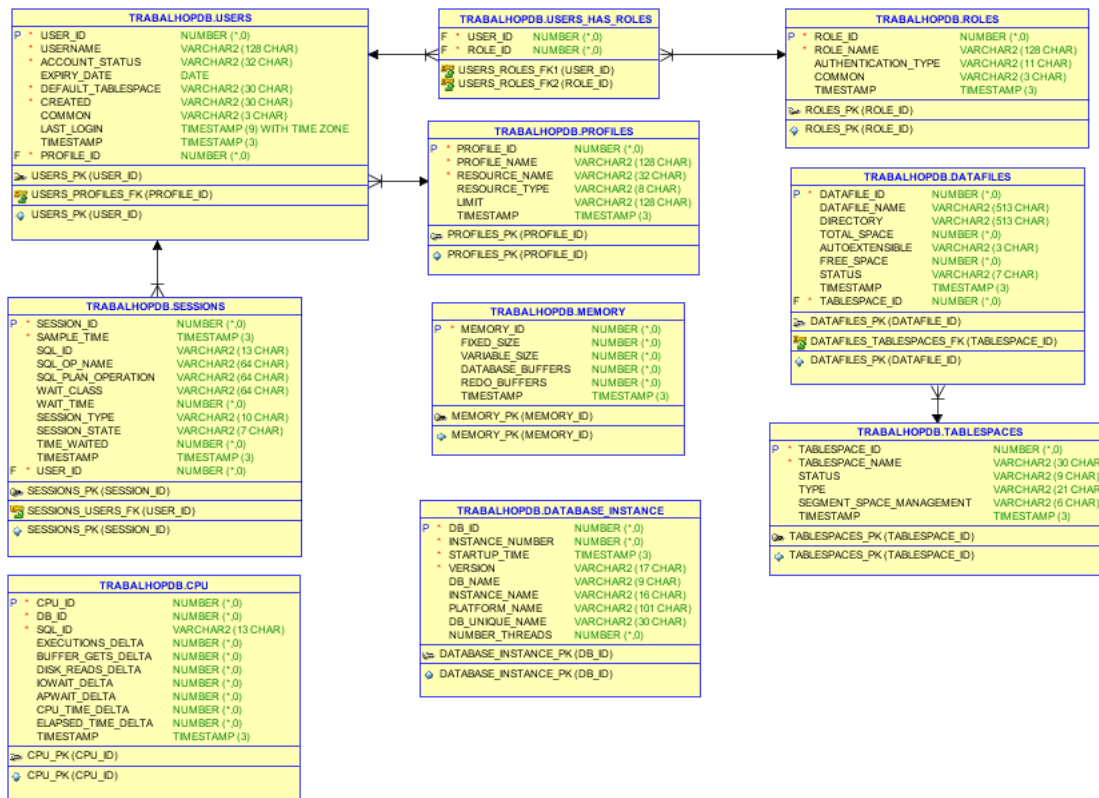


Figura 1: Modelo Relacional

### 2.2.1 Entidades

- **Tablespaces** - Contém informações lógicas relativas às *tablespaces*;
- **Datafiles** - Contém as informações físicas dos ficheiros relativos às *tablespaces*;
- **Users** - Contém informações relativas aos utilizadores;
- **Roles** - Contém informações relativas aos roles existentes;
- **Users Has Roles** - Contém informações sobre quais os roles que um utilizador tem;
- **Sessions** - Contém as informações da atividade das sessões;
- **Profiles** - Contém a informação relativa aos perfis;
- **CPU** - Contém a informação relativa a um histórico de estatísticas;
- **Memory** - Contém informação sobre a memória em uso;
- **Database Instance** - Contém a informação sobre as bases de dados e as instâncias;

Tabela 2: Descrição de cada tabela

Entidade	Atributos	Tipo e tamanho	Anulável
USERS	USER_ID	NUMBER	Não
	USERNAME	VARCHAR2 (128 CHAR)	Não
	ACCOUNT_STATUS	VARCHAR2 (32 CHAR)	Não
	EXPIRY_DATE	DATE	Sim
	DEFAULT_TABLESPACE	VARCHAR2 (30 CHAR)	Não
	CREATED	VARCHAR2 (30 CHAR)	Não
	COMMON	VARCHAR2 (3 CHAR)	Sim
	LAST_LOGIN	TIMESTAMP (9) WITH TIMEZONE	Sim
	TIMESTAMP	TIMESTAMP (3)	Sim
	PROFILE_ID	NUMBER	Não
USERS_HAS_ROLES	USER_ID	NUMBER	Não
	ROLE_ID	NUMBER	Não
PROFILES	PROFILE_ID	NUMBER	Não
	PROFILE_NAME	VARCHAR2 (128 CHAR)	Não
	RESOURCE_NAME	VARCHAR2 (32 CHAR)	Não
	RESOURCE_TYPE	VARCHAR2 (8 CHAR)	Sim
	LIMIT	VARCHAR2 (128 CHAR)	Sim
	TIMESTAMP	TIMESTAMP (3)	Sim
ROLES	ROLE_ID	NUMBER	Não
	ROLE_NAME	VARCHAR2 (128 CHAR)	Não
	AUTHENTICATION_TYPE	VARCHAR2 (11 CHAR)	Sim
	COMMON	VARCHAR2 (3 CHAR)	Sim
	TIMESTAMP	TIMESTAMP (3)	Sim
SESSIONS	SESSION_ID	NUMBER	Não
	SAMPLE_TIME	TIMESTAMP (3)	Não
	SQL_ID	VARCHAR2 (13 CHAR)	Sim
	SQL_OP_NAME	VARCHAR2 (64 CHAR)	Sim
	SQL_PLAN_OPERATION	VARCHAR2 (64 CHAR)	Sim
	WAIT_CLASS	VARCHAR2 (64 CHAR)	Sim
	WAIT_TIME	NUMBER	Sim
	SESSION_TYPE	VARCHAR2 (10 CHAR)	Sim
	SESSION_STATE	VARCHAR2 (7 CHAR)	Sim
	TIME_WAITED	NUMBER	Sim
	TIMESTAMP	TIMESTAMP (3)	Sim
	USER_ID	NUMBER	Não

CPU	CPU_ID DB_ID SQL_ID EXECUTIONS_DELTA BUFFER_GETS_DELTA DISK_READS_DELTA IOWAIT_DELTA APWAIT_DELTA CPU_TIME_DELTA ELAPSED_TIME_DELTA TIMESTAMP	NUMBER NUMBER VARCHAR2 (13 CHAR) NUMBER NUMBER NUMBER NUMBER NUMBER NUMBER NUMBER TIMESTAMP (3)	Não Não Não Sim Sim Sim Sim Sim Sim Sim Sim
MEMORY	MEMORY_ID FIXED_SIZE VARIABLE_SIZE DATABASE_BUFFERS REDO_BUFFERS TIMESTAMP	NUMBER NUMBER NUMBER NUMBER NUMBER TIMESTAMP (3)	Não Sim Sim Sim Sim Sim
DATABASE_INSTANCE	DB_ID INSTANCE_NUMBER STARTUP_TIME VERSION DB_NAME INSTANCE_NAME PLATFORM_NAME DB_UNIQUE_NAME NUMBER_THREADS	NUMBER NUMBER TIMESTAMP (3) VARCHAR2 (17 CHAR) VARCHAR2 (9 CHAR) VARCHAR2 (16 CHAR) VARCHAR2 (101 CHAR) VARCHAR2 (30 CHAR) NUMBER	Não Sim Sim Sim Sim Sim Sim Sim Sim
DATAFILES	DATAFILE_ID DATAFILE_NAME DIRECTORY TOTAL_SPACE AUTOEXTENSIBLE FREE_SPACE STATUS TIMESTAMP TABLESPACE_ID	NUMBER VARCHAR2 (513 CHAR) VARCHAR2 (513 CHAR) NUMBER VARCHAR2 (3 CHAR) NUMBER VARCHAR2 (7 CHAR) TIMESTAMP (3) NUMBER	Não Sim Sim Sim Sim Sim Sim Sim Não
TABLESPACES	TABLESPACE_ID TABLESPACE_NAME STATUS TYPE SEGMENT_SPACE_MANAGEMENT TIMESTAMP	NUMBER VARCHAR2 (30 CHAR) VARCHAR2 (9 CHAR) VARCHAR2 (21 CHAR) VARCHAR2 (6 CHAR) TIMESTAMP (3)	Não Não Sim Sim Sim Sim

### 2.2.2 Users

- USER\_ID - Identificador único de cada utilizador;
- USERNAME - Nome de cada utilizador;



- ACCOUNT\_STATUS - Estado da conta (open, expired, locked, etc);
- EXPIRY\_DATE - Data em que a conta expira;
- DEFAULT\_TABLESPACE - Tablespace pré-definido para os dados;
- CREATED - Data de criação do utilizador;
- COMMON - Indica se o utilizador é frequente;
- LAST\_LOGIN - Data do último login;
- TIMESTAMP - Data e hora a que os dados foram recolhidos;
- PROFILE\_ID - Identificador do perfil a que o utilizador está associado.

### 2.2.3 Sessions

- SESSION\_ID - Identificador atribuído a cada sessão;
- SAMPLE\_TIME - Duração de um exemplo;
- SQL\_ID - Identificador da declaração SQL que está a ser executada;
- SQL\_OP\_NAME - Nome da operação SQL;
- SQL\_PLAN\_OPERATION - Nome da operação interna realizada (update, select statement, table access);
- WAIT\_CLASS - Nome da classe do evento para cada sessão que estava à espera da amostra;
- SESSION\_TYPE - Tipo de sessão (foreground, background);
- SESSION\_STATE - Estado da sessão (waiting, on cpu);
- TIME\_WAITED - Tempo total que a sessão atual ficou à espera do evento;
- TIMESTAMP - Data e hora a que os dados foram recolhidos;
- USER\_ID - Identificador do utilizador associado à sessão.

### 2.2.4 CPU

- CPU\_ID - Identificador único do CPU;
- DB\_ID - Identificador da base de dados;
- SQL\_ID - Identificador da instrução SQL para a qual foram medidos os parâmetros do CPU;
- EXECUTIONS\_DELTA - Número de execuções que foram medidas para a instrução específica do SQL;
- BUFFER\_GETS\_DELTA - Número de gets feitos para a instrução específica do SQL;
- DISK\_READS\_DELTA - Número de reads feitos no disco para a instrução específica do SQL;
- IOWAIT\_DELTA - Medição de I/O físico para cada instrução SQL específica, fornecendo características importantes sobre a dada instrução;

- APWAIT\_DELTA - Medição do tempo de espera na aplicação;
- CPU\_TIME\_DELTA - Valor de tempo em CPU despendido no parsing, execução e fetch do resultado da instrução SQL específica, em microssegundos;
- ELAPSED\_TIME\_DELTA - Valor de tempo passado no parsing, execução e fetch do resultado da instrução SQL específica, em microssegundos;
- TIMESTAMP - Data e hora a que os dados foram recolhidos;

#### 2.2.5 *Users\_roles*

- USER\_ID - Identificador do utilizador;
- ROLE\_ID - Identificador do perfil.

#### 2.2.6 *Profiles*

- PROFILE\_ID - Identificador único do perfil;
- PROFILE\_NAME - Identificador do nome do perfil
- RESOURCE\_NAME - Identificador do recurso em uso;
- RESOURCE\_TYPE - Identificador do tipo recurso em uso;
- LIMIT - Identificador do limite;
- TIMESTAMP - Data e hora a que os dados foram recolhidos;

#### 2.2.7 *Memory*

- MEMORY\_ID - Identificador único atribuído à memória;
- FIXED\_SIZE - Valor, em bytes, do tamanho fixo;
- VARIABLE\_SIZE - Valor, em bytes, do tamanho variável;
- DATABASE\_BUFFERS - Valor, em bytes, dos database buffers;
- REDO\_BUFFERS - Valor, em bytes, dos redo buffers;
- TIMESTAMP - Data e hora a que os dados foram recolhidos;

#### 2.2.8 *Database\_instance*

- DB\_ID - Identificador único da base de dados;
- INSTANCE\_NUMBER - Número da instância;
- STARTUP\_TIME - Tempo de início da instância;
- VERSION - Versão da base de dados;
- DB\_NAME - Nome da base de dados;

- INSTANCE\_NAME - Nome da instância;
- PLATFORM\_NAME - Nome da plataforma onde estão a ser recolhidos os dados;
- DB\_UNIQUE\_NAME - Nome único da base de dados;
- NUMBER\_THREADS - Número de threads.

### 2.2.9 Roles

- ROLE\_ID - Identificador único do role;
- ROLE\_NAME - Identificador do nome do role;
- AUTHENTICATION\_TYPE - Indica o mecanismo de autenticação (none, external, global, application, password);
- COMMON - Indica se o role é frequente;
- TIMESTAMP - Data e hora a que os dados foram recolhidos;

### 2.2.10 Datafiles

- DATAFILE\_ID - Identificador único de cada datafile;
- DATAFILE\_NAME - Nome do datafile;
- DIRECTORY - Diretoria onde o datafile está guardado;
- TOTAL\_SPACE - Tamanho do ficheiro em bytes;
- AUTOEXTENSIBLE - Indica se o datafile é autoextensível;
- FREE\_SPACE - Espaço livre do datafile;
- STATUS - Estado online do datafile (sysoff, system, offline, online, recover);
- TIMESTAMP - Data e hora a que os dados foram recolhidos;
- TABLESPACE\_ID - Identificador do tablespace onde o datafile se encontra.

### 2.2.11 Tablespaces

- TABLESPACE\_ID - Identificador único de cada tablespace;
- TABLESPACE\_NAME - Nome do tablespace;
- STATUS - Estado do tablespace (online, offline, read only);
- TYPE - Tipo do tablespace (undo, permanent, temporary);
- SEGMENT\_SPACE\_MANAGEMENT - Indica se o espaço livre e ocupado no tablespace é gerido por free lists (manual) ou bitmaps (automático);
- TIMESTAMP - Data e hora a que os dados foram recolhidos;

## 2.3 Script de povoamento

Após a base de dados criada para o trabalho estar devidamente estruturada, foi necessário povoá-la com a informação presente na BD *orclpdb1*. Para tal, recorremos à linguagem de programação Python onde criámos um script que recolhe os dados pretendidos da bases de dados *orclpdb1* e os guarda na nova bases de dados *TrabalhoPDB*. Para além da informação recolhida foram também guardados os *timestamps* em tal ocorreu, de modo a criar um histórico de dados.

Para tornar possível todas as operações previamente descritas recorremos à biblioteca do python *cx\_Oracle*.

Em resumo, foram criadas duas conexões, uma para a bases de dados *orclpdb1* sendo que, neste caso, o user utilizado foi o "system" com a respetiva password, e a outra para a bases de dados *TrabalhoPDB* sendo que, nesse caso, o user utilizado foi o que também foi criado por nós, o *TRABALHOPDB*. Foram também criados dois cursores, um para cada conexão, responsáveis por executar as respetivas queries. Estas queries foram desenvolvidas para cada tabela sendo que consistem em queries do tipo select, insert e update.

```
username = "system"
password = "Oradoc_db1"

username2 = "TRABALHOPDB"
password2 = "trab"
#faz a conexão à BD orclpdb1 usando username, password, connectionString e o tipo de encoding
conn_pdb1 = cx_Oracle.connect(username, password, "///127.0.0.1/orclpdb1.localdomain", encoding="UTF-8")

#cursor responsável por executar as queries
cur_pdb1 = conn_pdb1.cursor()

#Faz a conexão à BD TrabalhoPDB para fazer os Inserts
conn_pdb2 = cx_Oracle.connect(username2, password2, "///127.0.0.1/TrabalhoPDB.localdomain", encoding="UTF-8")

#cursor responsável por executar as queries
cur_pdb2 = conn_pdb2.cursor()
```

Figura 2

Exemplo de uma inserção/update numa tabela, neste caso, a Roles.

```
#Query9
# (select da informação da trabalhoPDB)
sql9_1 = "SELECT * FROM ROLES "
# (select da informação da orclpdb1 )
sql9_2 = "SELECT ROLE, AUTHENTICATION_TYPE, COMMON FROM DBA_ROLES"
```

Figura 3: Queries select

```
res = cur_pdb1.execute(sql9_2)
columns = [col[0] for col in res.description]
res.rowfactory = Lambda *args: dict(zip(columns, args))
res = res.fetchall()
for row in res:
    if list(row.keys())[0] == 'ROLE':
        try:
            ct = datetime.datetime.now()
            row['TIMESTAMP'] = ct
            row['ROLE_ID'] = (randint(0,1000))
            print(row)
            insert_sql = "INSERT INTO ROLES (ROLE_NAME,AUTHENTICATION_TYPE, COMMON, TIMESTAMP, ROLE_ID) \
values (:1, :2, :3, :4, :5)"
            cur_pdb2.prepare(insert_sql)
            cur_pdb2.setinputsizes(cx_Oracle.DB_TYPE_VARCHAR, cx_Oracle.DB_TYPE_VARCHAR,
cx_Oracle.DB_TYPE_VARCHAR, cx_Oracle.DB_TYPE_TIMESTAMP, cx_Oracle.DB_TYPE_NUMBER,)
            cur_pdb2.execute(None,list(row.values()))
            conn_pdb2.commit()
        except cx_Oracle.IntegrityError:
            aux = row['ROLE_ID']
            del row['ROLE_ID']
            row['ROLE_ID'] = aux
            insert_sql = "UPDATE ROLES SET ROLE_NAME = :1, AUTHENTICATION_TYPE = :2, \
COMMON = :3, TIMESTAMP = :4 WHERE ROLE_ID = :5"
            cur_pdb2.prepare(insert_sql)
            cur_pdb2.setinputsizes(cx_Oracle.DB_TYPE_VARCHAR, cx_Oracle.DB_TYPE_VARCHAR,
cx_Oracle.DB_TYPE_VARCHAR, cx_Oracle.DB_TYPE_TIMESTAMP, cx_Oracle.NUMBER )
            cur_pdb2.execute(None,list(row.values()))
            conn_pdb2.commit()
        except cx_Oracle.IntegrityError:
            break
    else:
        break
s = cur_pdb2.execute(sql9_1)
print("\nInfo da roles no trabalho\n")
columns = [col[0] for col in s.description]
s.rowfactory = Lambda *args: dict(zip(columns, args))
s = s.fetchall()
```

Figura 4: Inserção e, por vezes, update na tabela Roles

## 3 API Rest

No enunciado no ponto 3 era-nos pedido a criação de uma API REST que se conecte à PBD criada anteriormente. Para a realização desse ponto optámos pela criação de uma API em Node Js com express.

### 3.1 Lógica de controlo

De forma a que esta API fosse capaz de responder aos pedidos que vimos como necessários para o bom funcionamentos da nossa interface implementámos as seguintes rotas:

- /: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Database.Instance";
- /cpu: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Cpu";
- /datafiles: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Datafiles";
- /memory: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Memory";
- /profiles: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Profiles";
- /roles: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Roles";
- /sessions: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Sessions";
- /tablespaces: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Tablespaces";
- /userhasroles: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Users\_has\_Roles". Esta rota acabou por não ser utilizada pois as informações relevantes desta tabela foram incluídas nas rotas referentes às tabelas de "Roles" e "Users";
- /users: Rota responsável por chamar a função de controlo responsável pela obtenção da informação referente à nossa tabela "Users". O objeto de retorno desta rota vai incluir informação não só referente a esta tabela mas também ao nome do "Role" associado a cada "User";

Todas estas rotas vão chamar funções de um mesmo ficheiro de controlo, funções estas que acabaram todas por se tornar extremamente simples tendo apenas que chamar funções de acesso à base de dados presentes em ficheiros do tipo "*model*" por nós criados.

### 3.2 Acessos à BD

De forma a que as rotas anteriormente referidas sejam capazes de retornar a informação necessária no formato pretendido, que neste caso será **JSON**, foi necessária a criação de vários ficheiros do tipo "*model*" que efetua esta operação para as suas tabelas respetivas, tendo sido criado um ficheiro deste género para cada tabela.

Um exemplo deste tipo de "modelos" é o ficheiro **users.js** que se pode observar de seguida, optámos por este exemplo visto ser aquele anteriormente referido que faz uso das tabelas "Users\_Has\_Roles" e "Roles" de forma a juntar a variável "Role\_Name" ao seu resultado final:

```
const db = require("../config/db").connection;

module.exports.list = async () => {
  var conn;
  const res = [];
  try {
    conn = await db;
    const query = `SELECT U.USER_ID, U.USERNAME, U.ACCOUNT_STATUS, U.EXPIRY_DATE, \
U.DEFAULT_TABLESPACE, U.CREATED, U.COMMON, U.LAST_LOGIN, U.TIMESTAMP, U.PROFILE_ID, \
R.ROLE_NAME \
FROM TRABALHOPDB.USERS U \
INNER JOIN TRABALHOPDB.USERS_HAS_ROLES H ON U.USER_ID = H.USER_ID \
INNER JOIN TRABALHOPDB.ROLES R ON H.ROLE_ID = R.ROLE_ID \
ORDER BY TIMESTAMP DESC \
FETCH FIRST 300 ROWS ONLY`;
    return conn.execute(query).then((data) => {
      data.rows.forEach((row) => {
        const aux = {};
        data.metaData.forEach((header, index) => {
          aux[header.name] = row[index];
        });
        res.push(aux);
      });
      return res;
    });
  } catch (e) {
    console.log(e);
  }
};
```

De notar que o **Oracle** tem uma estrutura de retorno, para este caso, não ideal contendo um objeto do tipo *array* **metaData**, contendo os *headers* (ou seja as *keys*/identificadores pretendidos), em conjunto com um outro objeto do mesmo tipo **rows** contendo a informação associada a estes *headers*. Estes dois arrays criam uma matrix de associação, a qual é percorrida pelo nosso algoritmo criando um só objeto do tipo *array* em formato **JSON** habitual ( headers[index]: row[index] ).

A variável **db** presente neste ficheiro, assim como em todos os outros ficheiros "modelo", é a variável responsável por executar a conexão à nossa Base de Dados. Para este efeito implementá-mos o seguinte código, fazendo uso do "node-module" **oracledb**:

```
const oracledb = require("oracledb");

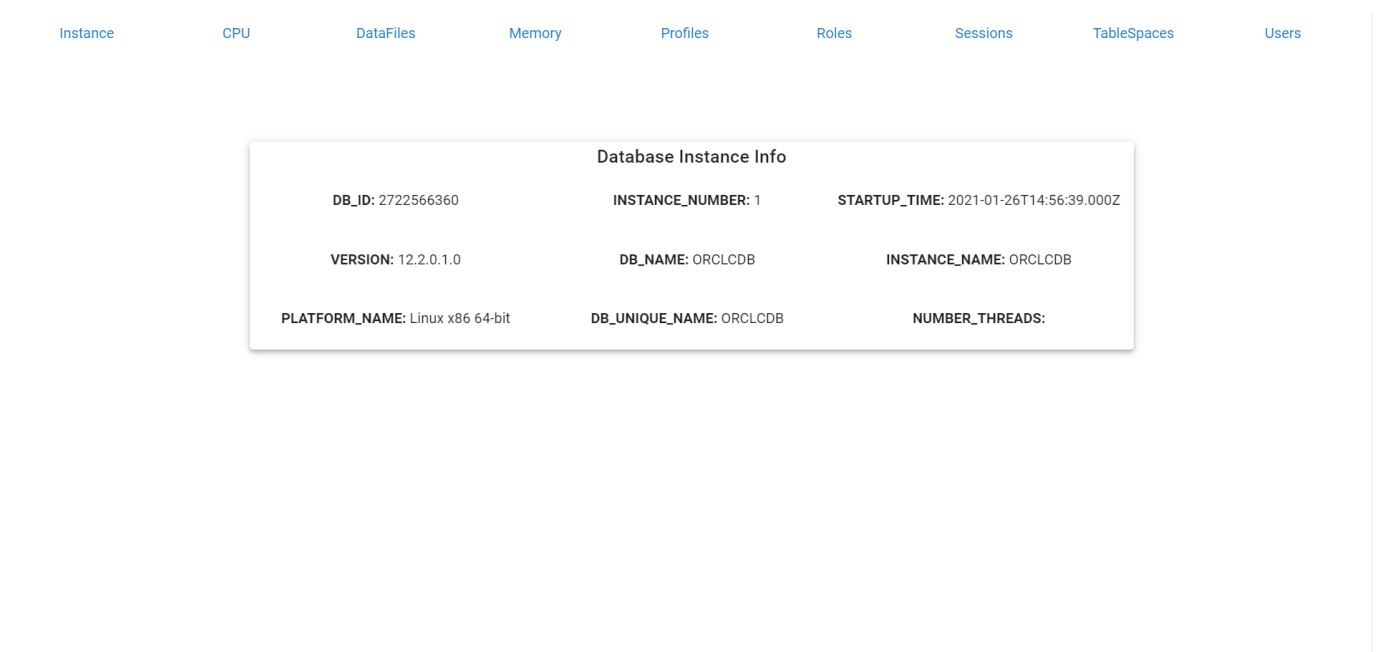
module.exports.connection = oracledb.getConnection({
  user: "system",
  password: "Oradoc_db1",
  connectString: "//127.0.0.1/TrabalhoPDB.localdomain",
});
```

## 4 Interface WEB

No enunciado foi-nos pedida uma interface web para a demonstração dos dados recolhidos nos pontos anteriores. Nesta secção vamos passar a apresentar exemplos destes mesmos resultados.

### 4.1 Instance

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação da Database Instance, interface esta que servirá de "*homepage*" para o nosso projeto.





## 4.2 CPU

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação ao CPU.

Instance		CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users	
CPU_ID	DB_ID	SQL_ID	EXECUTIONS_DELTA	BUFFER_GETS_DELTA	DISK_READS_DELTA	IOWAIT_DELTA	APWAIT_DELTA	CPU_TIME_DELTA	ELAPSED_TIME_DELTA	TIMESTAMP
276	2722566360	35ra2cdwvqg1w	382	475	1	5718	0	25635	52278	2021-01-27T17:10:12.350Z
613	2722566360	8swypbbr0m372	328	2922	126	440358	0	75194	476872	2021-01-27T17:10:12.342Z
52	2722566360	3wrrjm9qtr2my	2	223	63	35801	0	16309	139988	2021-01-27T17:10:12.338Z
841	2722566360	fn0snbuqyccq5	14	12326	778	2881123	0	1373894	8477655	2021-01-27T17:10:12.310Z
784	2722566360	5u7g54s63p4ts	545	7365	32	9076	0	62116	118708	2021-01-27T17:10:12.306Z
365	2722566360	3un99a0zwp4vd	328	5345	236	853724	0	177006	1155243	2021-01-27T17:10:12.280Z
358	2722566360	6h1ysp5jm8h8m	1	37347	149	617648	0	8547519	23399999	2021-01-27T17:10:12.248Z
435	2722566360	3jcm0tmm1wan4	1	541	5	23116	0	148978	181875	2021-01-27T17:10:12.243Z
379	2722566360	8594ra3zcassdf	4	102	0	0	0	153680	156061	2021-01-27T17:10:12.239Z
778	2722566360	2xyb5d6xg9srh	284	284	0	0	0	15284	13799	2021-01-27T17:10:12.235Z
783	2722566360	34cd4y8mbqvsk	14	184058	0	0	0	1715693	8239256	2021-01-27T17:10:12.211Z
910	2722566360	fn0snbuqyccq5	6	8852	771	2533666	0	849386	4182052	2021-01-27T17:10:12.183Z
972	2722566360	6h1ysp5jm8h8m	1	31408	75	120693	0	5777893	12765148	2021-01-27T17:10:12.148Z

## 4.3 Datafile

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação ao Datafile.

Instance	CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users
DATAFILE_NAME	DIRECTORY	TOTAL_SPACE	AUTOEXTENSIBLE	FREE_SPACE	STATUS	TABLESPACE_ID	TIMESTAMP	
/u02/app/oracle/oradata/ORCLCDB/orclpdb1/undotbs01.dbf		188743680	YES	137822208	ONLINE	609	2021-01-27T17:10:19.559Z	
/u02/app/oracle/oradata/ORCLCDB/orclpdb1/sysaux01.dbf		650117120	YES	38535168	ONLINE	225	2021-01-27T17:10:19.549Z	
/u02/app/oracle/oradata/ORCLCDB/orclpdb1/system01.dbf		293601280	YES	6094848	SYSTEM	769	2021-01-27T17:10:19.544Z	
/u02/app/oracle/oradata/ORCLCDB/pdata01.dbf		10485760	NO	9437184	ONLINE	156	2021-01-27T17:10:19.540Z	
/u02/app/oracle/oradata/ORCLCDB/orclpdb1/users01.dbf		10485760	YES	9437184	ONLINE	913	2021-01-27T17:10:19.533Z	

4.4 Memory

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação ao memory.

Instance	CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users
FIXED_SIZE	VARIABLE_SIZE		DATABASE_BUFFERS		REDO_BUFFERS		TIMESTAMP ↓	
					7983104		2021-01-27T17:10:09.479Z	
			469762048				2021-01-27T17:10:09.477Z	
	855639592						2021-01-27T17:10:09.474Z	
8792536							2021-01-27T17:10:09.471Z	
					7983104		2021-01-27T17:08:07.715Z	
			469762048				2021-01-27T17:08:07.713Z	
	855639592						2021-01-27T17:08:07.710Z	
8792536							2021-01-27T17:08:07.709Z	
					7983104		2021-01-27T16:17:28.149Z	
			469762048				2021-01-27T16:17:28.146Z	
	855639592						2021-01-27T16:17:28.136Z	
8792536							2021-01-27T16:17:28.118Z	

4.5 Profiles

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação ao profile.

Instance	CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users
PROFILE_NAME	RESOURCE_NAME		RESOURCE_TYPE		LIMIT	TIMESTAMP		
ORA_STIG_PROFILE	INACTIVE_ACCOUNT_TIME		PASSWORD		35	2021-01-27T17:10:03.446Z		
ORA_STIG_PROFILE	PASSWORD_GRACE_TIME		PASSWORD		5	2021-01-27T17:10:03.444Z		
ORA_STIG_PROFILE	PASSWORD_LOCK_TIME		PASSWORD		UNLIMITED	2021-01-27T17:10:03.424Z		
ORA_STIG_PROFILE	PASSWORD_VERIFY_FUNCTION		PASSWORD		ORA12C_STIG_VERIFY_FUNCTION	2021-01-27T17:10:03.412Z		
ORA_STIG_PROFILE	PASSWORD_REUSE_MAX		PASSWORD		10	2021-01-27T17:10:03.410Z		
ORA_STIG_PROFILE	PASSWORD_REUSE_TIME		PASSWORD		365	2021-01-27T17:10:03.407Z		
ORA_STIG_PROFILE	PASSWORD_LIFE_TIME		PASSWORD		60	2021-01-27T17:10:03.389Z		
ORA_STIG_PROFILE	FAILED_LOGIN_ATTEMPTS		PASSWORD		3	2021-01-27T17:10:03.351Z		
ORA_STIG_PROFILE	PRIVATE_SGA		KERNEL		DEFAULT	2021-01-27T17:10:03.323Z		
ORA_STIG_PROFILE	CONNECT_TIME		KERNEL		DEFAULT	2021-01-27T17:10:03.291Z		
ORA_STIG_PROFILE	IDLE_TIME		KERNEL		15	2021-01-27T17:10:03.256Z		
ORA_STIG_PROFILE	LOGICAL_READS_PER_CALL		KERNEL		DEFAULT	2021-01-27T17:10:03.214Z		
ORA_STIG_PROFILE	LOGICAL_READS_PER_SESSION		KERNEL		DEFAULT	2021-01-27T17:10:03.212Z		

## 4.6 Roles

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação ao roles.

Instance	CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users
ROLE_NAME			AUTHENTICATION_TYPE		COMMON	TIMESTAMP		
HR_MANAGER			NONE		NO	2021-01-27T17:10:01.375Z		
C##R1			NONE		YES	2021-01-27T17:10:01.354Z		
APEX_GRANTS_FOR_NEW_USERS_ROLE			NONE		YES	2021-01-27T17:10:01.348Z		
APEX_ADMINISTRATOR_ROLE			NONE		YES	2021-01-27T17:10:01.318Z		
DV_REALM_RESOURCE			NONE		YES	2021-01-27T17:10:01.302Z		
DV_POLICY_OWNER			NONE		YES	2021-01-27T17:10:01.287Z		
DV_DATAPUMP_NETWORK_LINK			NONE		YES	2021-01-27T17:10:01.278Z		
DV_AUDIT_CLEANUP			NONE		YES	2021-01-27T17:10:01.254Z		
DV_GOLDENGATE_REDO_ACCESS			NONE		YES	2021-01-27T17:10:01.238Z		
DV_XSTREAM_ADMIN			NONE		YES	2021-01-27T17:10:01.216Z		
DV_GOLDENGATE_ADMIN			NONE		YES	2021-01-27T17:10:01.213Z		
DV_STREAMS_ADMIN			NONE		YES	2021-01-27T17:10:01.209Z		
DV_PATCH_ADMIN			NONE		YES	2021-01-27T17:10:01.206Z		

4.7 Sessions

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação as sessions.

Instance	CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users		
SAMPLE_TIME ↑	SQL_ID	SQL_OP_NAME	SQL_PLAN_OPERATION	WAIT_CLASS	WAIT_TIME	SESSION_TYPE	SESSION_STATE	TIME_WAITED	USER_ID	TIMESTAMP
2021-01-26T23:04:23.647Z	fn0snbuqycq5	SELECT	PX COORDINATOR		1057	FOREGROUND	ON CPU	0	0	2021-01-27T17:10:09.148Z
2021-01-26T22:01:01.413Z	fn0snbuqycq5	SELECT	PX COORDINATOR	Other	0	FOREGROUND	WAITING	21024	0	2021-01-27T17:10:09.145Z
2021-01-16T10:33:32.752Z				System I/O	0	BACKGROUND	WAITING	445	0	2021-01-27T17:10:09.141Z
2021-01-16T17:31:42.826Z	6h1ysp5jm8h8m	UPSERT		Scheduler	0	FOREGROUND	WAITING	19289	0	2021-01-27T17:10:09.135Z
2021-01-16T16:31:41.805Z	34cd4y8mbqvsk	SELECT	HASH JOIN	Scheduler	0	FOREGROUND	WAITING	11926	0	2021-01-27T17:10:09.128Z
2021-01-16T14:51:19.594Z	6h1ysp5jm8h8m	UPSERT	MERGE STATEMENT	Scheduler	0	FOREGROUND	WAITING	11078	0	2021-01-27T17:10:09.124Z
2021-01-16T14:31:18.914Z	6h1ysp5jm8h8m	UPSERT		Scheduler	0	FOREGROUND	WAITING	23581	0	2021-01-27T17:10:09.119Z
2021-01-16T16:31:31.775Z	6h1ysp5jm8h8m	UPSERT	HASH JOIN	Scheduler	0	FOREGROUND	WAITING	19907	0	2021-01-27T17:10:09.116Z
2021-01-16T15:51:31.009Z	34cd4y8mbqvsk	SELECT		Scheduler	0	FOREGROUND	WAITING	11833	0	2021-01-27T17:10:09.112Z
2021-01-16T15:51:41.060Z	3qkhfbf2kyvhk	SELECT	XMLTABLE EVALUATION	Scheduler	0	FOREGROUND	WAITING	35189	0	2021-01-27T17:10:09.107Z
2021-01-16T13:31:26.459Z	cyju584tyg8mc	SELECT		Scheduler	0	FOREGROUND	WAITING	9420	0	2021-01-27T17:10:09.103Z
2021-01-16T12:50:55.716Z	34cd4y8mbqvsk	SELECT		Scheduler	0	FOREGROUND	WAITING	11251	0	2021-01-27T17:10:09.099Z
2021-01-16T12:50:55.716Z	6h1ysp5jm8h8m	UPSERT		Scheduler	0	FOREGROUND	WAITING	15905	0	2021-01-27T17:10:09.096Z

## 4.8 TableSpaces

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação aos tablespaces.

Instance	CPU	DataFiles	Memory	Profiles	Roles	Sessions	TableSpaces	Users
TABSPACE_ID	TABSPACE_NAME	STATUS	TYPE	SEGMENT_SPACE_MANAGEMENT	TIMESTAMP			
411	TEMP_PDB3	ONLINE	TEMPORARY	MANUAL	2021-01-27T17:10:19.311Z			
270	PDATA	ONLINE	PERMANENT	AUTO	2021-01-27T17:10:19.300Z			
315	AEBD_TEMP	ONLINE	TEMPORARY	MANUAL	2021-01-27T17:10:19.290Z			
607	AEBD_TABLES	ONLINE	PERMANENT	AUTO	2021-01-27T17:10:19.276Z			
918	USERS	ONLINE	PERMANENT	AUTO	2021-01-27T17:10:19.269Z			
707	TEMP	ONLINE	TEMPORARY	MANUAL	2021-01-27T17:10:19.260Z			
221	UNDOTBS1	ONLINE	UNDO	MANUAL	2021-01-27T17:10:19.257Z			
76	SYSAUX	ONLINE	PERMANENT	AUTO	2021-01-27T17:10:19.253Z			
152	SYSTEM	ONLINE	PERMANENT	MANUAL	2021-01-27T17:10:19.250Z			
510	TEMP_PDB3	ONLINE	TEMPORARY	MANUAL	2021-01-27T17:08:18.813Z			
839	PDATA	ONLINE	PERMANENT	AUTO	2021-01-27T17:08:18.801Z			
237	AEBD_TEMP	ONLINE	TEMPORARY	MANUAL	2021-01-27T17:08:18.795Z			
549	AEBD_TABLES	ONLINE	PERMANENT	AUTO	2021-01-27T17:08:18.793Z			

## 4.9 Users

A figura seguinte demonstra a interface web criada para a demonstração dos dados recolhidos em relação aos users. Nesta interface é possível verificar o resultado do modelo de "users", referido anteriormente na secção "API Rest", estando presente na tabela a coluna "ROLE\_NAME" que representa o "role" que cada utilizador tem associado na base de dados de origem.

Instance		CPU	DataFiles	Memory	Profiles		Roles	Sessions	TableSpaces	Users
USER_ID	USERNAME	ACCOUNT_STATUS	EXPIRY_DATE	DEFAULT_TABLESPACE	CREATED	COMMON	LAST_LOGIN	PROFILE_ID	ROLE_NAME	TIMESTAMP
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	JAVAUSERPRIV	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	PROVISIONER	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	PROVISIONER	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	PROVISIONER	2021-01-27T17:10:05.475Z
9	SYSTEM	OPEN	2021-04-21T18:53:33.000Z	SYSTEM	TEMP	YES	2021-01-27T02:45:30.000Z	19	PROVISIONER	2021-01-27T17:10:05.475Z

## 5 Swingbench

O Swingbench é uma ferramenta para testar a performance de uma base de dados Oracle com a geração de carga nesta.

Depois da conclusão das fases anteriores aplicamos esta ferramenta com a intenção de verificar se a implementação estava correta, sendo possível ver isso através da interface web.

Para realizar os testes à *orclpdb1*, colocamos a seguinte informação nas configurações: , em *username* colocamos "system", em *password* colocamos "Oradoc\_pdb1" e em *connect string* colocamos "//127.0.0.1/orclpdb1.localdomain". Os testes foram realizados com 5 utilizadores.

The screenshot shows the Swingbench Configuration window with the following settings:

- Configuration** (selected tab):
  - User Details** (selected sub-tab):
    - Username: system
    - Password: [masked]
    - Connect String: //127.0.0.1/orclpdb1.localdomain
    - Driver Type: Oracle jdbc Driver
    - ☐ Collect database statistics
    - ☐ Take AWR snapshots at start and end (10g/11g only)
    - System Username: [empty]
    - System Password: [empty]
  - Load** (selected sub-tab):
    - Number of Users: 5
    - Min. Inter Delay Between Transactions (ms): 0
    - Max. Inter Delay Between Transactions (ms): 0
    - Min. Intra Delay Within Transactions (ms): 5
    - Max. Intra Delay Within Transactions (ms): 5
    - Logon Delay (milliseconds): 1
    - Logon Group: 1
    - Wait Till All Sessions Log On: true
    - Logoff Post Transaction: false
    - Tx. per Reconnect: 0
    - Benchmark Run Time (hh:mm): 0:0
    - Record Statistics After (hh:mm): 0:0
    - Stop Recording After (hh:mm): 0:0

Figura 5: Configurações Swingbench

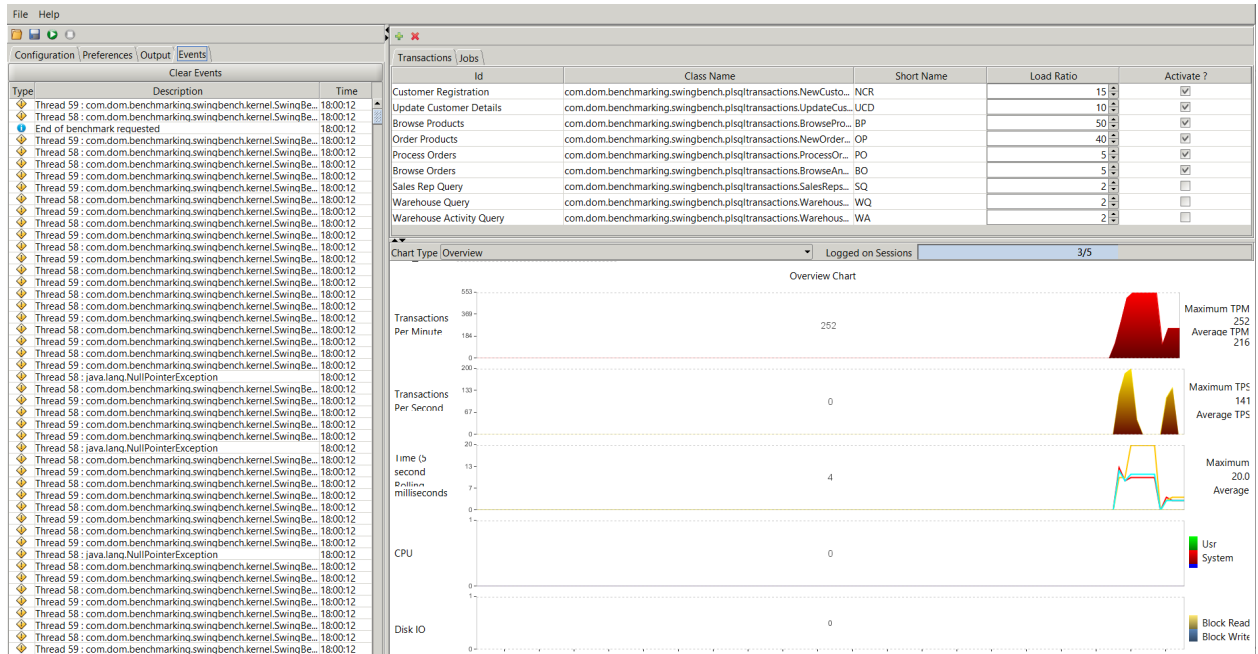


Figura 6: Swingbench



## 6 Conclusão

Com este projeto obtivemos uma melhor percepção do funcionamento de uma Base de Dados em **Oracle** assim como do conceito de "**Pluggable Databases**". Foi-nos ainda possível aprofundar o nosso conhecimento relativamente ao desenvolvimento de aplicações *web*, e consequentemente o nosso conhecimento de conceitos/*frameworks* como **NodeJS**, **Express** e **Vue**.

Para além deste conhecimento adquirimos ainda uma melhor compreensão da relevância do paradigma de administração de bases de dados, assim como as suas abrangentes possibilidades. Paradigma este que é cada vez mais importante na sociedade atual, visto o crescimento da quantidade de informação digital existente.

Apesar de algumas dificuldades, principalmente na construção do agente **Python** que efetua a ligação entre a Base de Dados de origem e a criada para este projeto, pensamos ter cumprido todos os objetivos estipulados tanto pela equipa docente como pelo grupo. Resta-nos agora esperar por oportunidades de aplicar o conhecimento obtido nesta UC em situações "reais".