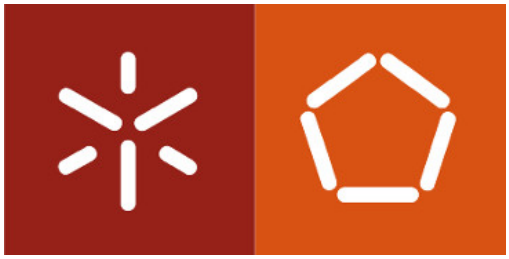


Universidade do Minho

Mestrado em Engenharia Informática

Trabalho Prático

Carolina Resende Marques, PG42818

Francisco Borges, PG42829

Rui Pereira, PG42853

Vasco António Lopes Ramos, PG42852

Administração e Exploração de Bases de Dados
4º Ano, 1º Semestre
Departmento de Informática

29 de janeiro de 2021

Índice

Lista de Figuras

Lista de Tabelas

Lista de Códigos

1 Introdução

Um sistema de informação é um sistema formal de armazenamento e processamento de informação. A maioria das empresas utiliza atualmente uma base de dados para automatizar os seus sistemas de informação. Uma base de dados é uma colecção organizada de informação tratada como uma unidade. O objetivo de uma base de dados é recolher, armazenar, e recuperar informação relacionada para a sua utilização por aplicações de bases de dados.

O modelo relacional é a base para um sistema de gestão de bases de dados relacionais (RDBMS). Um RDBMS move dados para uma base de dados, armazena os dados, e recupera os dados para que as aplicações os possam manipular.

Uma base de dados Oracle é uma RDBMS . Consiste em uma ou mais unidades lógicas de armazenamento denominados *tablespaces*, que de forma unificada permitem o armazenamento de toda a informação de uma Base de Dados. Cada *tablespace* consiste em um ou mais ficheiros físicos denominados *datafiles*. Estes ficheiros são estruturas físicas presentes no sistema operativo do servidor no qual a base de dados oracle está a correr. [1]

Utilizando os conhecimentos adquiridos na componente prática e teórica da UC de Administração e Exploração de Base de Dados foi construido um monitor de BD que apresenta os principais parâmetros de avaliação de performance de uma BD Oracle.

Este relatório descreve o procedimento usado para construir esta ferramenta de monitorização, desde o agente de recolha de dados e a base de dados associada, passando pela camada de servir os dados (Rest API) até à visualização das métricas recolhidas (interface).

2 Criação e Esquema da Base de Dados

Para suportar e persistir a recolha de informação relativa à base de dados *Oracle*, fazia todo o sentido criar uma nova *PDB (Pluggable Database)*. Parte deste processo inclui criar os *tablespaces* e respetivos *datafiles*, permanentes e temporários, bem como um utilizador com as permissões necessárias à operacionalização do sistema.

2.1 Criação de Base de Dados de suporte ao sistema

Desta forma recorreu-se ao seguinte processo de criação:

1. No sistema de ficheiros e como ***root***, criou-se o espaço físico para a PDB:

```
cd /home/uminho/dockers/data/oracle
cd u02/app/oracle/oradata/ORCL/
mkdir orclmonitor
chown oracle:oinstall orclmonitor/
```

Código 1: Criação de espaço para a PDB

2. Dentro da *shell* da instância de *Oracle*, criou-se a PDB:

```
sqlplus sys/Oradoc_db1 as sysdba

CREATE pluggable database orclmonitor
 admin user aebd_admin IDENTIFIED BY aebd
 roles = (DBA)
 FILE_NAME_CONVERT=('/u02/app/oracle/oradata/ORCL/pdbseed',
     '/u02/app/oracle/oradata/ORCL/orclmonitor');

ALTER pluggable database orclmonitor open;
```

Código 2: Criação da PDB

3. Após criar a PDB, é necessário criar os *tablespaces* e respetivos *datafiles*:

```
connect sys/Oradoc_db1@localhost:1521/orclmonitor.localdomain
  AS sysdba

CREATE tablespace orclmonitor_data datafile
  '/u02/app/oracle/oradata/ORCL/orclmonitor/permmonitor01.dbf'
  SIZE 10M AUTOEXTEND ON;

CREATE temporary tablespace orclmonitor_temp tempfile
  '/u02/app/oracle/oradata/ORCL/orclmonitor/tempmonitor01.dbf'
  SIZE 10M AUTOEXTEND ON;
```

Código 3: Criação dos *Tablespaces* e *Datafiles*

4. Por fim, falta apenas proceder à criação do utilizador e conceder-lhe as permissões necessárias à operacionalização do sistema:

```
CREATE user orcl_monitor IDENTIFIED BY secret
  DEFAULT TABLESPACE orclmonitor_data
  TEMPORARY TABLESPACE orclmonitor_temp
  QUOTA UNLIMITED ON orclmonitor_data;

GRANT CREATE MATERIALIZED VIEW, UNLIMITED TABLESPACE,
  CREATE SESSION, RESOURCE, ALTER ANY MATERIALIZED VIEW,
  DROP ANY MATERIALIZED VIEW, DROP ANY VIEW,
  CREATE ANY VIEW TO orcl_monitor;
```

Código 4: Criação do utilizador

2.2 Criação do Esquema de Dados

Tendo em conta os requisitos do trabalho e considerando a pesquisa feita pelo grupo, decidiu-se recolher os seguintes dados:

- PDBs existentes;
- Utilizadores;
- Sessões;

- Utilização de CPU;
- Utilização da memória;
- *Tablespaces* e *Datafiles*.

Assim, cirou-se um esquema de base de dados que alberga estes dados e relaciona os dados que são relacionáveis. A figura 1 mostra o esquema final utilizado para guardar os valores recolhidos e servir a *Rest API*.

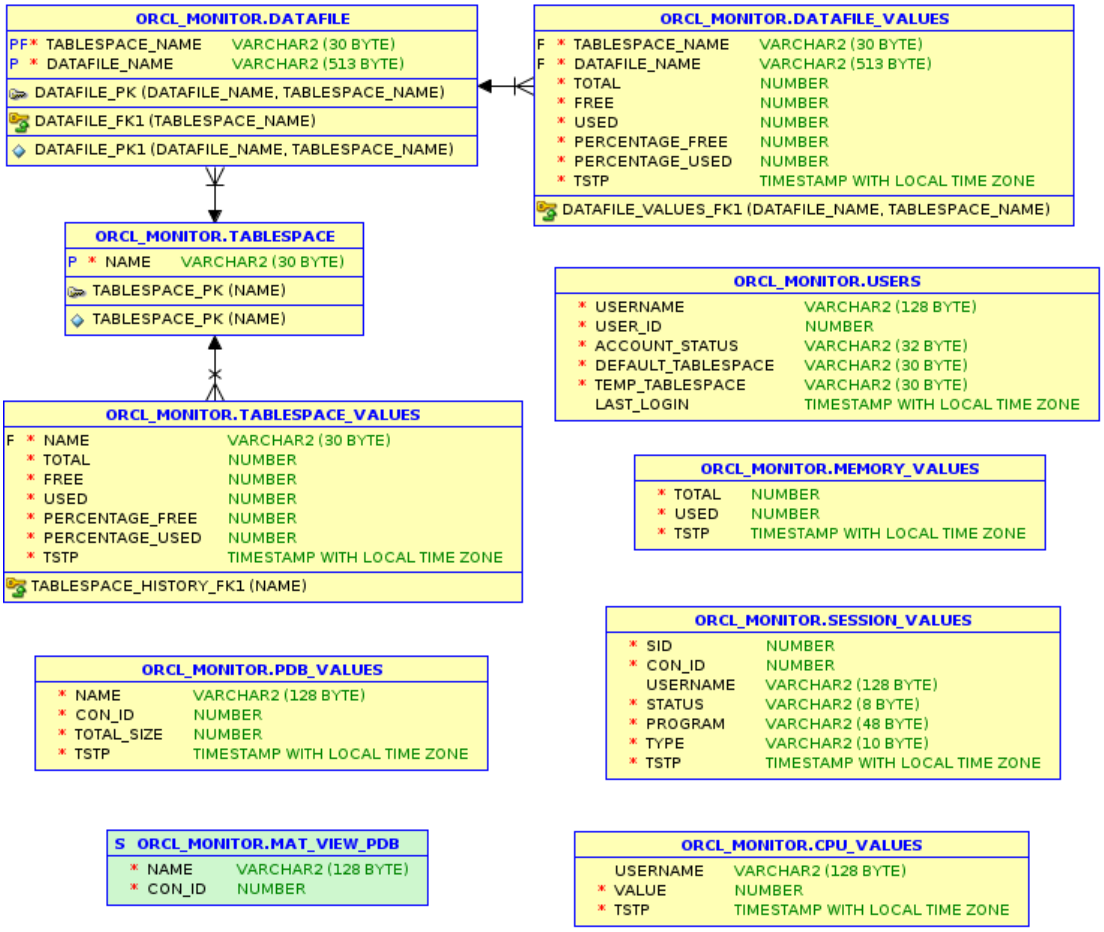


Figura 1: Esquema da base de dados

Para além do esquema representado na figura 1, foram, também, criadas algumas *Views* para permitir mais facilmente pesquisar os dados de forma agregada consoante o tempo (por minuto, hora, dia, mês e ano). Este tipo de *views* foram

aplicadas aos relativos a: *CPU*, *Datafiles*, *Tablespaces*, *Memory*, *PDB* e *Sessions*.

3 Agente de Recolha de Informação

De forma a conseguir recolher a informação necessária para monitorizar a instância da base de dados *oracle*, é neessário ter um agente que continuamente recolha métricas e valores relacionados com a base de dados a vários níveis. Para tal, criou-se um agente em Python, utilizando a biblioteca, documentada em [2].

Para termos os dados necessários, atlernámos entre ligações (utilizadores) de **root**, à CDB, e ligações às PDBs. Por fim, de forma a termos uma frequente fonte de dados, preparámos o agente, em *python*, de forma a correr de 30 em 30 segundos, pelo que conseguimos ter 2 instâncias de dados, por cada minuto.

Na porção de código 5 é possível ver a estrutura principal de execução do agente e na porção de código 6 é possível ver o exemplo das funções de recolha e registo associadas as valores das PDBs.

```
def run_agent():
  try:
    pdb_query()
    session_query()
    memory_query()
    users_query()
    tablespaces_query()
    datafiles_query()
    cpu_query()
  except cx_Oracle.Error as error:
    print("Error occurred: " + error)

def main():
  while True:
    run_agent()
    time.sleep(30)
```

Código 5: Função *main* do agente, em *python*

```
def pdb_query():
  with cx_Oracle.connect(
    config.username_root, config.password_root, config.dsn_root,
    cx_Oracle.SYSDBA, encoding=config.encoding,
  ) as connection:
    with connection.cursor() as cursor:
      cursor.execute(pdb_sql)
      while True:
        rows = cursor.fetchmany(batch_size)
        if not rows:
          break
        insert_pdb_entries(rows)

def insert_pdb_entries(rows):
  sql = "insert into pdb_values(name, con_id, total_size, tstp)
           values(:name, :con_id, :total_size, :tstp)"
  with cx_Oracle.connect(
    config.username2, config.password2,
    config.dsn2, encoding=config.encoding
  ) as connection:
    with connection.cursor() as cursor:
      cursor.setinputsizes(None, None, None, cx_Oracle.TIMESTAMP)
      cursor.executemany(sql, rows)
      connection.commit()
```

Código 6: Funções de captação e registo dos dados das *PDBs*

4 Rest API

A terceira etapa era criar uma camada de consumo dos dados que fosse capaz de abstrair o esquema da base de dados. O desenvolvimento de uma ***Rest API***, para servir os dados num formato **JSON** faz precisamente isso.

Por uma questão de familiaridade, a Rest API foi desenvolvida em *Express.js*, com base de *Node.js* e a comunicação com a base de dados foi feita através do *package* ***oracledb***, documentado em [3].

Para melhor organizar a estrutura da API, decidiu-se dividir a sua implementação em três principais responsabilidades:

- ***Resources*:** lida com a conexão à base de dados e a execução de *queries*;
- ***Routes*:** lida com: o roteamento dos pedidos, o tratamento dos mesmos e a sua respetiva resposta;
- ***Controllers*:** abstrai as *queries* à base de dados com componentes específicos para cada tipo de valores.

Para facilitar o desenvolvimento da interface e permitir que o trabalho fosse feito assíncronamente entre os vários elementos foi criada uma página de documetanção da Rest API, no ponto de vista de utilização. Esta página foi criado através do *package* ***swagger-ui-express***, documentado em [4].

Na figura 2 é possível ver a página de receção da documentação *Swagger* da *Rest API* e na figura 3 consegue-se ver o detalhe de um dos *endpoints* implementados na *Rest API*.

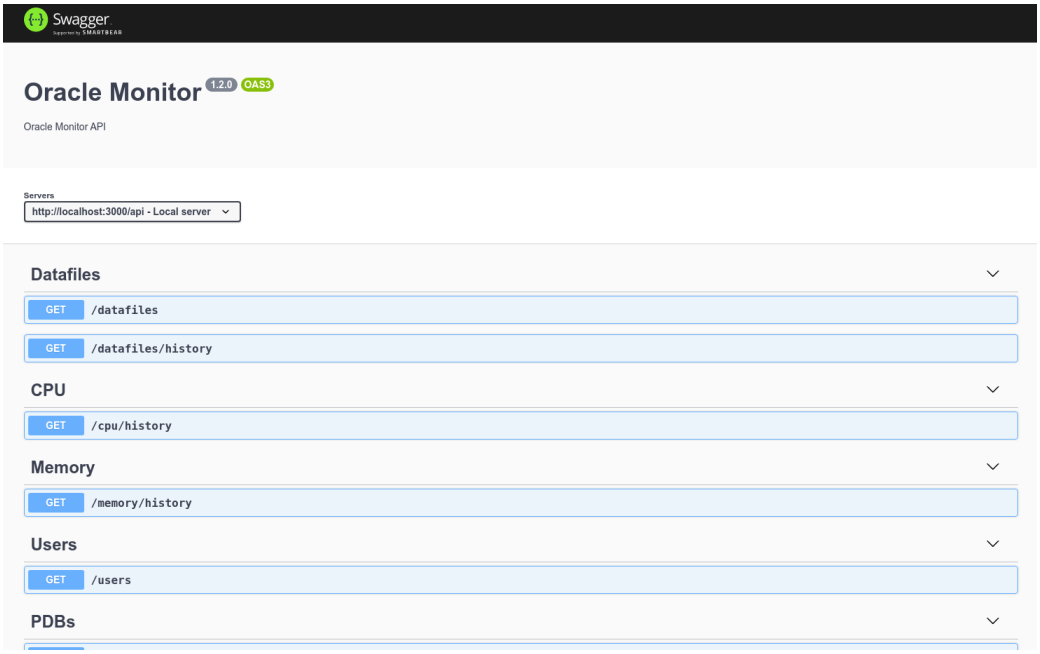


Figura 2: Página geral da documentação da *Rest API*

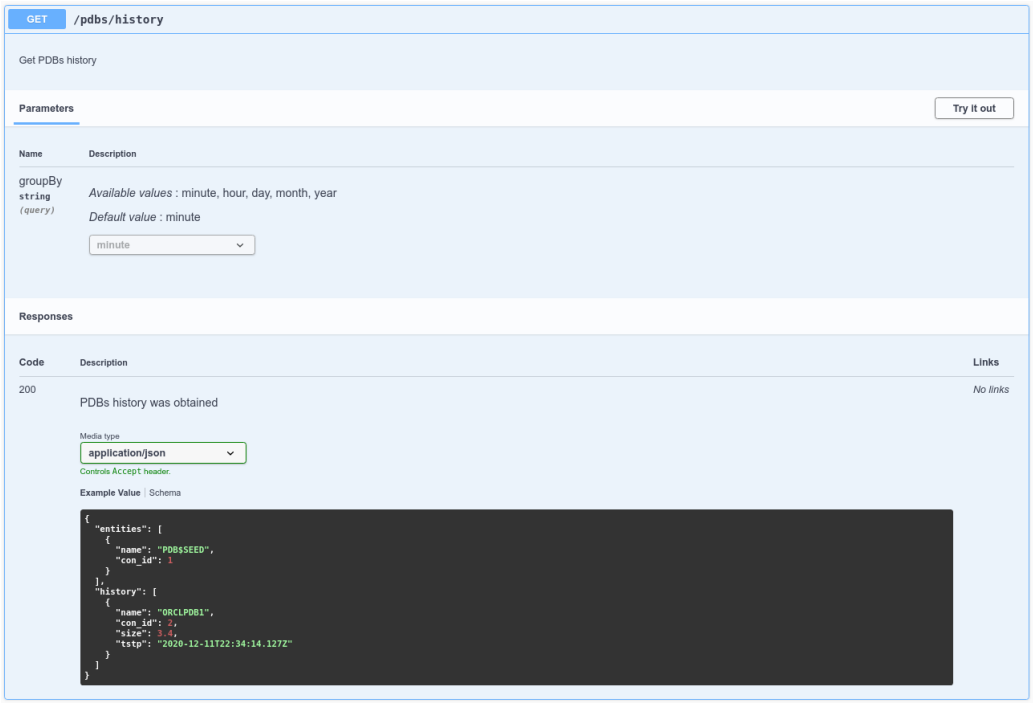


Figura 3: Detalhe de um *endpoint* da *Rest API*

Por fim, para questões de documentação apresenta-se a tablea 1, com a listagem dos endpoints existentes na Rest API, e o propósito de cada um:

Endpoint (URL)	**Ação**
/cpu/history	Lista os valores associados à utilização do CPU, ao longo do tempo.
/memory/history	Lista os valores associados à utilização da memória RAM, ao longo do tempo.
/users	Lista os utilizadores existentes no sistema de base de dados.
/pdbs	Lista as *PDBs* existentes no sistema de base de dados,
/pdbs/history	Lista os valores associados às *PDBs*, ao longo do tempo.
/sessions/history	Lista os valores associados às sessões existentes no sistema de base de dados, ao longo do tempo.
/sessions/total/history	Lista o total de sessões existentes no sistema de base de dados, ao longo do tempo.
/tablespaces	Lista os *tablespaces* existentes no sistema de base de dados.
/tablespaces/history	Lista os valores associados aos *tablespaces* existentes no sistema de base de dados, ao longo do tempo.
/datafiles	Lista os *datafiles* existentes no sistema de base de dados.
/datafiles/history	Lista os valores associados aos *datafiles* existentes no sistema de base de dados, ao longo do tempo.

Tabela 1: Listagem dos *endpoints* existentes no serviço da *Rest API*

5 Interface Web

A última etapa deste trabalho trata-se da criação de uma interface *web* que apresente os dados recolhidos na tarefa 3. A interface criada está dividida em 3 páginas principais:

1. Na **Página Inicial** foi desenvolvida uma dashboard com 4 gráficos com o objetivo de permitir a visualização da distribuição da utilização de memória e do CPU, o total de sessões e o tamanho das diferentes PDBs ao longo do tempo. De 30 em 30 segundos esta informação é atualizada e no caso de se querer imediatamente ver a nova informação, o botão *Force Refresh* poderá ser usado.

Figura 4: Página Inicial

Estes dados podem ser vistos ao longo do tempo em minutos, horas, dias, meses e anos. Por predefinição todos as informações apresentadas encontram-se por minutos. Para além disto, é apresentada uma listagem de todos os utilizadores na BD. Informações como o nome, *status*, o *tablespace* por *default*, o *tablespace* temporário e o *last login* deste *user* podem ser consultadas.

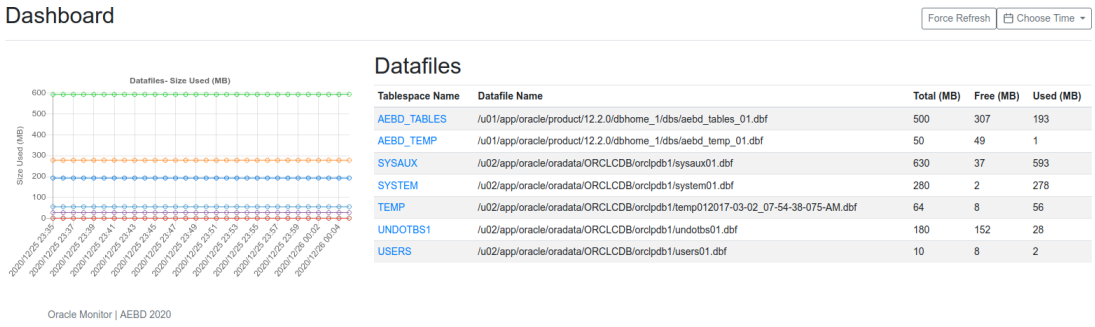
Figura 5: Dashboard em Horas

Users

Name	Status	Default Tablespace	Temporary Tablespace	Last Login
SYS	OPEN	TEMP	SYSTEM	null
SYSTEM	OPEN	TEMP	SYSTEM	2020-12-26T00:03:22.000Z
XS$NULL	EXPIRED & LOCKED	TEMP	SYSTEM	null
LBACSYS	EXPIRED & LOCKED	TEMP	SYSTEM	null
OUTLN	EXPIRED & LOCKED	TEMP	SYSTEM	null
DBSNMP	EXPIRED & LOCKED	TEMP	SYSAUX	null
APPQOSSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
DBSFWUSER	EXPIRED & LOCKED	TEMP	SYSAUX	null
GGSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
ANONYMOUS	EXPIRED & LOCKED	TEMP	SYSAUX	null
FLOWS_FILES	EXPIRED & LOCKED	TEMP	SYSAUX	null
CTXSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
SI_INFORMTN_SCHEMA	EXPIRED & LOCKED	TEMP	SYSAUX	null
DVSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
DVF	EXPIRED & LOCKED	TEMP	SYSAUX	null
GSMADMIN_INTERNAL	EXPIRED & LOCKED	TEMP	SYSAUX	null
ORDPLUGINS	EXPIRED & LOCKED	TEMP	SYSAUX	null
APEX_050000	EXPIRED & LOCKED	TEMP	SYSAUX	null
MDSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
OLAPSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
ORDDATA	EXPIRED & LOCKED	TEMP	SYSAUX	null
XDB	EXPIRED & LOCKED	TEMP	SYSAUX	null
WMSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
ORDSYS	EXPIRED & LOCKED	TEMP	SYSAUX	null
GSMCATUSER	EXPIRED & LOCKED	TEMP	USERS	null

Figura 6: Listagem dos utilizadores presentes

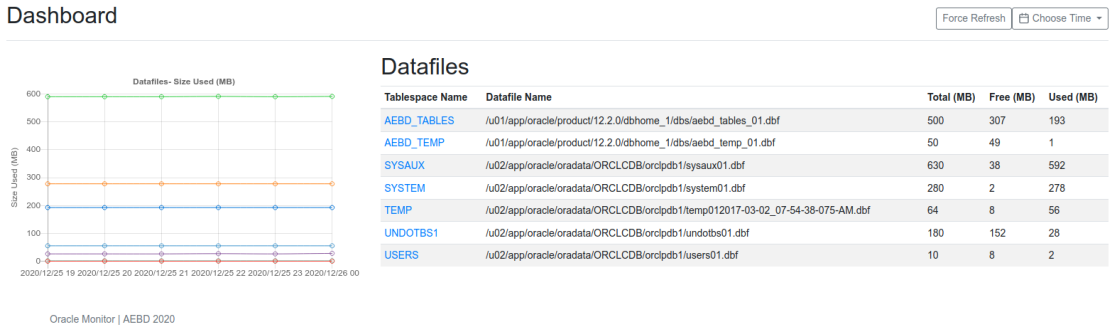
2. A **Página de *Datafiles*** apresenta um gráfico com o tamanho usado por cada *datafile* ao longo do tempo e uma tabela com a informação sobre todos os *datafiles*. Inclui o nome do *tablespace* onde este se encontra, o nome deste *datafile*, o tamanho total dele, o espaço livre e ocupado (em MB).



Tablespace Name	Datafile Name	Total (MB)	Free (MB)	Used (MB)
AEBD_TABLES	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/aebd_tables_01.dbf	500	307	193
AEBD_TEMP	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/aebd_temp_01.dbf	50	49	1
SYSAUX	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/sysaux01.dbf	630	37	593
SYSTEM	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/system01.dbf	280	2	278
TEMP	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/temp012017-03-02_07-54-38-075-AM.dbf	64	8	56
UNDOTBS1	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/undotbs01.dbf	180	152	28
USERS	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/users01.dbf	10	8	2

Figura 7: Página de *Datafiles*

Como na página inicial, existe também a possibilidade de ver a evolução do tamanho usado de cada *datafile* em minutos, horas, dias, meses e anos.



Tablespace Name	Datafile Name	Total (MB)	Free (MB)	Used (MB)
AEBD_TABLES	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/aebd_tables_01.dbf	500	307	193
AEBD_TEMP	/u01/app/oracle/product/12.2.0/dbhome_1/dbs/aebd_temp_01.dbf	50	49	1
SYSAUX	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/sysaux01.dbf	630	38	592
SYSTEM	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/system01.dbf	280	2	278
TEMP	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/temp012017-03-02_07-54-38-075-AM.dbf	64	8	56
UNDOTBS1	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/undotbs01.dbf	180	152	28
USERS	/u02/app/oracle/oradata/ORCLCDB/orclpdb1/users01.dbf	10	8	2

Figura 8: Página de Datafiles em Horas

3. Finalmente, a **Página de *Tablespaces***. Esta é muito semelhante à página de *datafiles* onde pode ser observado um gráfico com a evolução do tamanho usado por cada *tablespace* e uma tabela com o nome de cada *tablespace*, o tamanho total, livre e usado em MB.

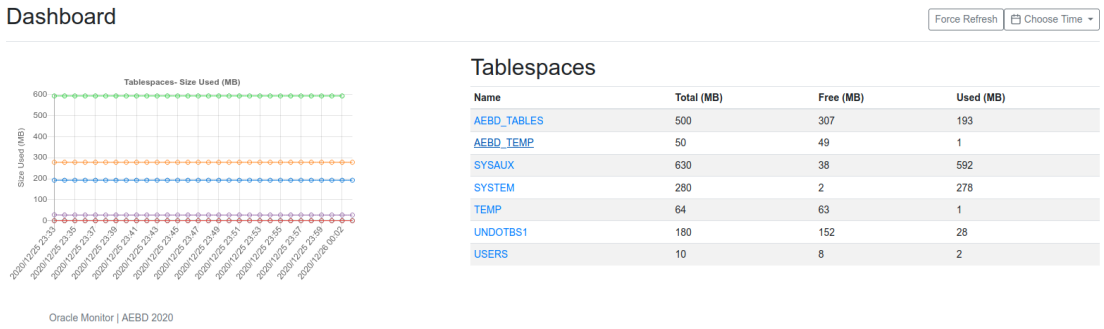


Figura 9: Página de *Tablespaces*

Novamente, esta evolução pode ser observada noutras métricas de tempo.

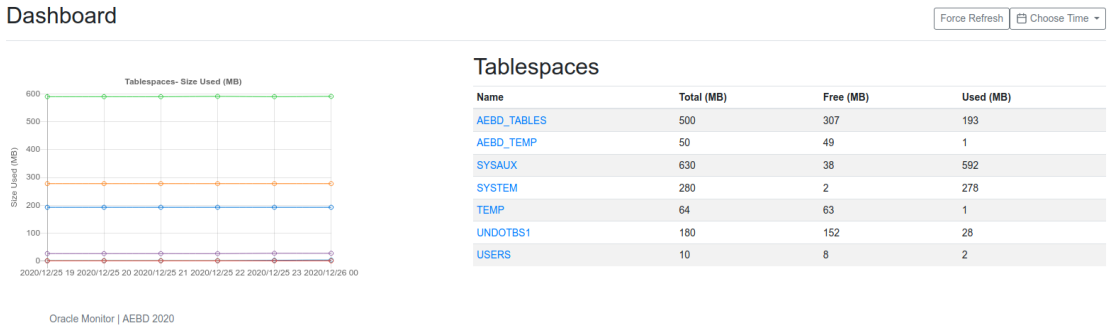


Figura 10: Página de *Tablespaces* em Horas

Uma página de detalhes para um dado *tablespace* foi também adicionada.

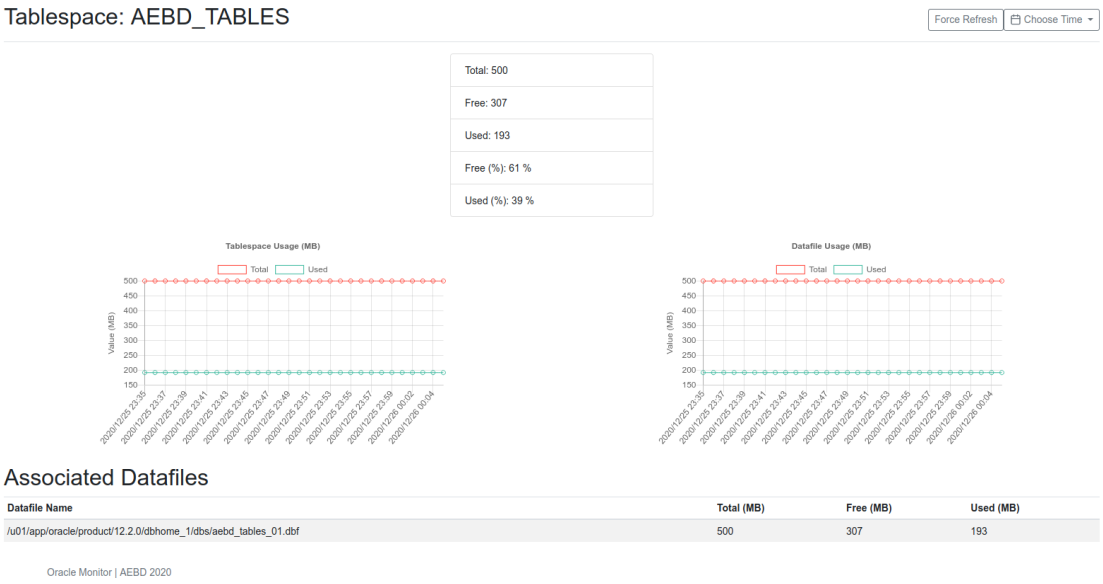


Figura 11: Informação sobre o Tablespace AEBD_TABLES

Na figura 11, informação sobre o *tablespace* AEBD_TABLES é apresentada. O tamanho total, usado e livre (em MB) é novamente apresentado, para além das respetivas percentagens. Os dois gráficos mostram o tamanho livre e usado dos *datafiles* associados a este *tablespace* e do *tablespace* em si. Esta página de detalhes apresenta finalmente uma tabela com todos os *datafiles* associados e a sua informação.

6 Conclusão

Atualmente *Oracle Enterprise Manager Database Express*, também referido como *EM Express*, é uma ferramenta baseada na web para gerir a Base de Dados Oracle 12c. Construído dentro do servidor de base de dados, oferece suporte para tarefas administrativas básicas, tais como armazenamento e gestão de utilizadores, e fornece soluções abrangentes para diagnóstico e ajuste de desempenho. O projeto desenvolvido para a UC de Administração e Exploração de Base de Dado teve como base o que é apresentado no *EM Express* e transformou isso em algo mais simples mas sucinto e claro.

Com este projeto foi possível consolidar conhecimentos como criação de PDBs, *users* e respetivos acessos, ter uma melhor perceção de como funciona a arquitetura de uma instância de Base de Dados Oracle, bem como explorar algumas áreas mais relacionadas com a Rest API e Interface.

Por fim, o grupo considera ter cumprido todos os requisitos com sucesso e que o trabalho desenvolvido reflete não só o trabalho desenvolvido no decorrer da disciplina, mas também os objetivos delineados para o mesmo de uma forma fidedigna.

Referências

[1] *Documentação - Oracle Database*, "https://docs.oracle.com/en/database/oracle/oracle-database/21/index.html", Acedido: 26-12-2020.

[2] *Documentação - cx Oracle*, "https://cx-oracle.readthedocs.io/en/latest", Acedido: 18-12-2020.

[3] *Documentation for the Oracle Database Node.js Add-on*, "https://oracle.github.io/node-oracledb/doc/api.html", Acedido: 20-12-2020.

[4] *Swagger UI Express | NPM Documentation*, "https://www.npmjs.com/package/swagger-ui-express", Acedido: 10-12-2020.