



# IPG

Politécnico  
da Guarda

Escola Superior  
de Tecnologia e Gestão

## OFICINA ONLINE

---

### BASE DE DADOS ONLINE PARA OFICINAS

---

---

<b>Curso(s):</b>	Computação Móvel
<b>Unidade(s) Curricular(es):</b>	Base de Dados Avançada
<b>Ano Letivo:</b>	2016/2017
<b>Docente:</b>	Dr. Carlos Fonseca
<b>Coordenador(a) da área disciplinar:</b>	Dr. Carlos Fonseca
<b>Data:</b>	03 de Fevereiro de 2017

## Conteúdo

Introdução	3
Modelo er.....	4
Dicionário de dados .....	5
BD_abastecimentos .....	5
BD_Anomalias.....	6
BD_Clientes .....	7
BD_VEICULOS .....	8
bd_mecanicos.....	9
bd_reparacoes_tipos .....	10
bd_ocorrencias.....	11
bd_Ocorrencias_estados .....	13
Bd_reparacoes_ocorrencias .....	14
Views .....	15
View SOMALITROS .....	15
View VEICULOSPORCLIENTES.....	15
view loginclientes.....	15
sequências .....	16
Procedimentos e package .....	17
WS_Inserereocorrencia.....	18
WS_REGISTAVEICULO .....	19
WS_verveiculos.....	20
WS_login.....	21
Sinónimos .....	22
FUNÇÕES.....	23
TRIGGERS.....	25
Segurança .....	27
ROLES.....	28
transações.....	29
Implementação .....	31
Auditoria .....	32
Conclusão .....	33
Bibliografia.....	34

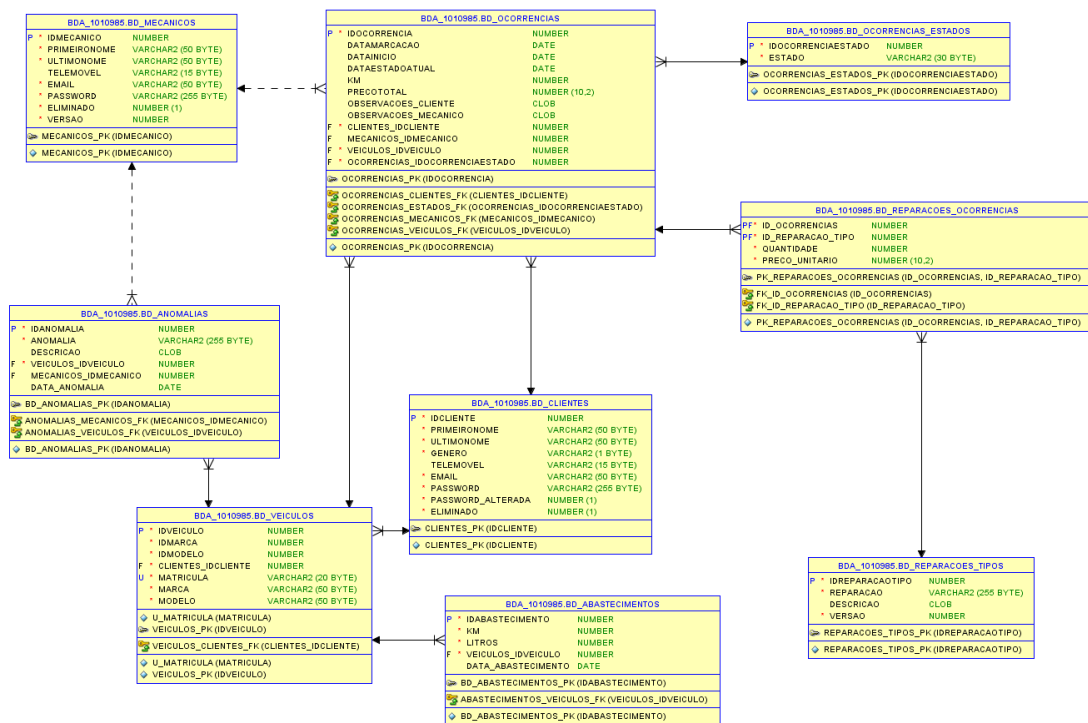
---

## INTRODUÇÃO

---

Este projeto tem como objetivo desenvolver uma aplicação que gere as manutenções e reparações de veículos dos clientes de uma oficina de automóveis. Assim, a oficina consegue manter um historial das intervenções efetuadas nos veículos e assim fornecer aos seus clientes um serviço de maior qualidade ao mesmo tempo que fidelizar o cliente. A aplicação pode ser utilizada pelos clientes para fazer marcações na oficina, registar e gerir novos veículos e registar anomalias dos veículos.

## MODELO ER



## DICIONÁRIO DE DADOS

### BD\_ABASTECIMENTOS

Tabela onde o utilizador vai registar os abastecimentos que efetua no(s) veículo(s)

ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDABASTECIMENTO	NUMBER	(PK) ID do abastecimento	Not Null
KM	NUMBER	Km do veículo quando o abastecimento foi efetuado	Not Null
LITROS	NUMBER	Quantidade (L) abastecidos	Not Null
VEICULOS_IDVEICULO	NUMBER	(FK) ID do veículo. BD_VEICULOS.IDVEICULO	Not Null
DATA	DATE	Data do abastecimento	Not Null

**Media\_Combustível**- CAMPO DESNORMALIZADO, DE MODO A FACILITAR A CONSULTA DA MEDIA TOTAL DE UM VEICULO.

SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
_id	INTEGER	(PK) ID do abastecimento	Not Null
Km	INTEGER	Km do veículo quando o abastecimento foi efetuado	Not Null
Litros	INTEGER	Quantidade (L) abastecidos	Not Null
IdVeiculo	INTEGER	(FK) ID do veículo. BD_VEICULOS.IDVEICULO	Not Null
DataAbastecimento	TEXT	Data do abastecimento	Not Null
Sync	INTEGER	Indica se o registo foi enviado p/ o servidor (0:não enviado   1:enviado)	Not Null

### BD\_ANOMALIAS

Tabela onde são registadas as anomalias do veículo (registos efetuados pelo cliente e mecânico.  
Ex: Ruído da direção ao virar p/ a esquerda)

#### ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDANOMALIA	NUMBER	(PK) ID da anomalia	Not Null
ANOMALIA	VARCHAR2(255)	Nome da anomalia	Not Null
DESCRICAO	CLOB	Descricao da anomalia	
VEICULOS_IDVEICULO	NUMBER	(FK) ID do veículo. BD_VEICULOS.IDVEICULO	Not Null
MECANICOS_IDMECANICO	NUMBER	(FK) ID do mecânico. Pode ser Null porque a anomalia pode ser adicionada por um cliente.	
DATA_ANOMALIA	DATE	Data da anomalia	Not Null

#### SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
_id	INTEGER	(PK) ID da anomalia	Not Null
Anomalia	TEXT	Nome da anomalia	Not Null
Desc	TEXT	Descricao da anomalia	
Idveiculo	INTEGER	(FK) ID do veiculo. bd_veiculos.IdVeiculo	Not Null
IdMecanico	INTEGER	(FK) ID do mecânico. Pode ser Null porque a anomalia pode ser adicionada por um cliente. bd_mecanicos.IdMecanico	
Sync	INTEGER	Indica se o registo foi enviado p/ o servidor (0:não enviado   1:enviado)	Not Null

## BD\_CLIENTES

Tabela onde são registados os clientes da oficina

ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDCLIENTE	NUMBER	(PK) ID do cliente	Not Null
PRIMIRONOME	VARCHAR2(50)	Primeiro nome	Not Null
ULTIMONOME	VARCHAR2(50)	Último nome	Not Null
TELEMOVEL	VARCHAR2(15)	Telemóvel	
EMAIL	VARCHAR2(50)	Email	Not Null
PASSWORD	VARCHAR2(255)	Palavra-passe encriptada	Not Null
PASSWORD_ALTERADA	NUMBER(1)	Indica se o utilizador já alterou a password	
ELIMINADO	NUMBER(1)	Indica se o utilizador foi eliminado (default: 0)	Not Null
GENERO	VARCHAR2(1)	Género do cliente (m/f)	Not Null, Check

SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
_id	INTEGER	(PK) ID do cliente	Not Null
PrimeiroNome	TEXT	Primeiro nome	Not Null
UltimoNome	TEXT	Último nome	Not Null
Telemovel	TEXT	Telemóvel	
Email	TEXT	Email	Not Null
Password	TEXT	Palavra-passe encriptada	Not Null
Sync	INTEGER	Indica se o registo foi enviado p/ o servidor (0:não enviado   1:enviado)	Not Null
Genero	TEXT	Género do cliente (m/f)	

## BD\_VEICULOS

Tabela onde são registados os veículos dos clientes

### ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDVEICULO	NUMBER	(PK) ID do veículo	Not Null
IDMARCA	NUMBER	(FK) ID da marca (web service)	Not Null
IDMODELO	NUMBER	(FK) ID do modelo (web service)	Not Null
CLIENTES_IDCLIENTE	NUMBER	(FK) ID do proprietário. BD_CLIENTES.IDCLIENTE	Not Null
MATRICULA	VARCHAR2(20)	Matrícula do veículo	Not Null, Unique
MARCA	VARCHAR2(50)	Marca do veículo	Not Null
MODELO	VARCHAR2(255)	Modelo do veículo	Not Null

### SQLITE

Nome do campo	Tipo de dados	Descrição	Restrições
_id	INTEGER	(PK) ID do veículo	Not Null
IdMarca	INTEGER	(FK) ID da marca (web service)	Not Null
IdModelo	INTEGER	(FK) ID do modelo (web service)	Not Null
IdCliente	INTEGER	(FK) ID do proprietário. BD_CLIENTES.IDCLIENTE	Not Null
Matricula	TEXT	Matrícula do veículo	Not Null
Marca	TEXT	Marca do veículo	Not Null
Modelo	TEXT	Modelo do veículo	Not Null



## BD\_MECANICOS

Tabela onde são registados os mecânicos da oficina

ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDMECANICO	NUMBER	(PK) ID do mecânico	Not Null
PRIMIRONOME	VARCHAR2(50)	Primeiro nome	Not Null
ULTIMONOME	VARCHAR2(50)	Último nome	Not Null
TELEMOVEL	VARCHAR2(15)	Telemóvel	
EMAIL	VARCHAR2(50)	Email	Not Null
PASSWORD	VARCHAR2(255)	Palavra-passe encriptada	Not Null
ELIMINADO	NUMBER(1)	Indica se o mecânico foi eliminado (default: 0)	Not Null
VERSAO	NUMBER	Indica a versão do registo. É incrementado quando existe uma alteração (default: 1)	Not Null

SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
_id	INTEGER	(PK) ID do cliente	Not Null
PrimeiroNome	TEXT	Primeiro nome	Not Null
UltimoNome	TEXT	Último nome	Not Null
Telemovel	TEXT	Telemóvel	
Email	TEXT	Email	Not Null
Versao	INTEGER	Indica a versão do registo. Se for diferente do servidor, o registo deve ser atualizado	

### BD\_REPARACOES\_TIPOS

Tabela onde são registados os tipos de reparações. Estes dados ficam disponíveis para seleção.  
Ex: mudança de óleo, troca de pneus, etc.

ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDREPARACAOTIPO	NUMBER	(PK) ID do tipo de reparação	Not Null
REPARACAO	VARCHAR2(255)	Identifica o tipo de reparação. Ex: Mudança de óleo	Not Null
DESCRICAO	CLOB	Descrição do tipo de reparação	
VERSAO	NUMBER	Indica a versão do registo. É incrementado quando existe uma alteração (default: 1)	Not Null

SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
_id	NUMBER	(PK) ID do tipo de reparação	Not Null
Reparacao	TEXT	Identifica o tipo de reparação. Ex: Mudança de óleo	Not Null
Descricao	TEXT	Descrição do tipo de reparação	
Versao	INTEGER	Indica a versão do registo. Se for diferente do servidor, o registo deve ser atualizado	

### BD\_OCORRENCIAS

Tabela onde são registados todos os dados das ocorrências. Ex: Mudança de óleo, datas, etc.

ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDOCORRENCIA	NUMBER	(PK) ID da ocorrência	Not Null
DATAMARCAÇAO	DATE	Data da marcação da ocorrência	
DATAINICIO	DATE	Data em que a ocorrência foi iniciada	
DATAESTADOATUAL	DATE	Data do estado atual	Not Null
KM	NUMBER	Km do do veiculo quando foi iniciada a reparação	
PRECOTOTAL	NUMBER(10,2)	Preço total da ocorrência	
OBSERVACOES_CLIENTE	CLOB	Observações colocadas pelo mecânico	
OBSERVACOES_MECANICO	CLOB	Observações colocadas pelo cliente	
CLIENTES_IDCLIENTE	NUMBER	(FK) ID do cliente. BD_CLIENTES.IDCLIENTE	Not Null
MECANICOS_IDMECANICO	NUMBER	(FK) ID do mecânico. BD_MECANICOS.IDMECANICO	
VEICULOS_IDVEICULO	NUMBER	(FK) ID do veiculo. BD_VEICULOS.IDVEICULO	Not Null
OCORRENCIAS_IDOCORRENCIAESTADO	NUMBER	(FK) ID da ocorrências_estados. BD_OCORRENCIAS_ESTADOS. IDOCORRENCIAESTADO	Not Null

**PREÇOTOTAL->** CAMPO DESNORMALIZADO DE MODO A FACILITAR A CONSULTA DO PREÇO TOTAL DE UMA ENCOMENDA.

SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
<b>_id</b>	INTEGER	(PK) ID da ocorrência	Not Null
<b>DataMarcacao</b>	TEXT	Data da marcação da ocorrência	
<b>DataInicio</b>	TEXT	Data em que a ocorrência foi iniciada	
<b>DataEstadoAtual</b>	TEXT	Data do estado atual	Not Null
<b>KM</b>	INTEGER	Km do do veiculo quando foi iniciada a reparação	
<b>PrecoTotal</b>	TEXT	Preço total da ocorrência	
<b>ObsMecanico</b>	TEXT	Observações colocadas pelo mecânico	
<b>ObsCliente</b>	TEXT	Observações colocadas pelo cliente	
<b>ClientesIdClientes</b>	INTEGER	(FK) ID do cliente. bd_clientes.IdCliente	Not Null
<b>MecanicosIdMecanicos</b>	INTEGER	(FK) ID do mecânico. bd_mecanicos.IdMecanico	
<b>VeículosIdVeiculo</b>	INTEGER	(FK) ID do veiculo	Not Null
<b>OcorrenciasIdOcorrenciasEstado</b>	INTEGER	(FK) ID da bd_ocorrencias_estados. bd_ocorrencias_estados.Id OcorrenciaEstado	Not Null

### BD\_OCORRENCIAS\_ESTADOS

Tabela onde são registados os estados das ocorrências. Ex: marcada.

#### ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
IDOCORRENCIAESTADO	NUMBER	(PK) ID da rel. ocorrências - estados	Not Null
ESTADO	VARCHAR2(30)	Nome do estado (marcada, efetuada, pendente, cancelada)	Not Null

#### SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
_id	INTEGER	(PK) ID da ocorrência	Not Null
Estado	TEXT	Data da marcação da ocorrência	Not Null

### BD\_REPARACOES\_OCORRENCIAS

Tabela onde são registadas as várias reparações de cada ocorrência.

ORACLE

Nome do campo	Tipo de dados	Descrição	Restrições
ID_OCORRENCIAS	NUMBER	(PK) ID da rel. reparações - ocorrências	Not Null
ID_REPARACAO_TIPO	NUMBER	(FK) ID do tipo de reparação. BD_REPARACOES_TIPOS. IDREPARACAOTIPO	Not Null
QUANTIDADE	NUMBER	Quantidade	Not Null
PRECO_UNITARIO	NUMBER(10,2)	Preço unitário	Not Null

SQLite

Nome do campo	Tipo de dados	Descrição	Restrições
IdOcorrencias	INTEGER	(PK) ID da ocorrência	Not Null
IdReparacaoTipo	INTEGER	Data da marcação da ocorrência	Not Null
Quantidade	INTEGER	Quantidade	Not Null
PrecoUnitario	TEXT	Preço unitário	Not Null

---

## VIEWS

---

### VIEW SOMALITROS

Esta View devolve o consumo total de litros de cada veículo.

```
create or replace view SOMALITROS as
SELECT v.IDVEICULO, sum(a.LITROS) litros
from BD_VEICULOS v full JOIN BD_ABASTECIMENTOS a on v.IDVEICULO =
a.VEICULOS_IDVEICULO
group by v.IDVEICULO with READ ONLY ;
```

### VIEW VEICULOSPORCLIENTES

Esta View devolve a marca, modelo, matricula, consumo total de litros e o cliente de cada veículo. O consumo de litros é devolvido pela View SOMALITROS.

```
create or replace view VEICULOSPORCLIENTES AS
select v.MARCA,v.MODELO,v.MATRICULA,v.CLIENTES_IDCLIENTE,s.litros
from BD_VEICULOS v join SomaLitros s on v.IDVEICULO = s.IDVEICULO
with READ ONLY ;
```

### VIEW LOGINCLIENTES

Esta View devolve os dados dos clientes para efeitos de login. A View foi feita par aumentar a segurança da aplicação devido ao atributo "READ ONLY".

```
create OR REPLACE view LOGINCLIENTES AS
select PRIMEIRONOME,ULTIMONOME,EMAIL,PASSWORD,IDCLIENTE from
BD_CLIENTES WITH READ ONLY;[1]
```

---

## SEQUÊNCIAS

---

Estas sequências foram criadas com o propósito de incrementar as chaves primárias das tabelas automaticamente.

```
CREATE SEQUENCE seq_idCliente
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idAbastecimento
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idAnomalia
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idMecanico
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idOcorrencia
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idOcorrenciaEstado
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idReparacaoTipo
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE SEQUENCE seq_idVeiculo
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```



---

## PROCEDIMENTOS E PACKAGE

---

Nesta secção, apresentamos os procedimentos e o package criados na nossa base de dados.

O seguinte package foi criado para organizar os nossos procedimentos de web service. Contém os procedimentos WS\_VERVEICULOS, WS\_INSEREOCORRENCIA, WS\_REGISTAVEICULO e WS\_LOGIN.

```
create or replace package WS as
procedure WS_VerVeiculos(id_cliente number,v_email
varchar2,v_password varchar2) ;
procedure WS_InsererOcorrencia (v_dMarcacao date,
    v_dInicio DATE,
    v_km NUMBER,
    v_obsC CLOB,
    v_obsM CLOB,
    v_idcliente NUMBER,
    v_idmecanico NUMBER,
    v_idveiculo number,
    v_estado number,
    v_email varchar2,
    v_password varchar2) ;
procedure WS_RegistaVeiculo (v_idmarca number, v_idmodelo number,
    v_marca varchar2, v_modelo varchar2, v_matricula varchar2,
    v_idcliente number, v_email varchar2, v_password varchar2) ;
procedure WS_LOGIN (vemail varchar2, vpassword varchar2) ;

end WS;
```

Nesta secção, apresentamos os procedimentos de web service, criados dentro do package anterior. Todos os procedimentos verificam a autenticação do utilizador.

### WS\_INSEREOCORRENCIA

Este webservice insere uma ocorrência na base de dados. Esta ocorrência pode ser um serviço efetuado pelo mecânico ou uma marcação efetuada pelo cliente.

```
create or replace procedure WS_InsererOcorrencia
(v_dMarcacao date,
 v_dInicio DATE,
 v_km NUMBER,
 v_obsC CLOB,
 v_obsM CLOB,
 v_idcliente NUMBER,
 v_idmecanico NUMBER,
 v_idveiculo number,
 v_estado number,
 v_email varchar2,
 v_password varchar2) is

    id_cliente number;
    resultado varchar2(10000);
BEGIN

    select idCliente into id_cliente from bd_clientes where
email=v_email and password=v_password;
    INSERT INTO BD_OCORRENCIAS

(DATAMARCACAO,DATAINICIO,KM,PRECOTOTAL,OBSERVACOES_CLIENTE,OBSERVACOE
S_MECHANICO,

CLIENTES_IDCLIENTE,MECHANICOS_IDMECHANICO,VEICULOS_IDVEICULO,OCORRENCIA
S_IDOCORRENCIAESTADO)
VALUES

(v_dMarcacao,v_dInicio,v_km,0,v_obsC,v_obsM,v_idcliente,v_idmecanico,
v_idveiculo,v_estado);
    resultado:=resultado || '{"sucesso": "1"}';
    http.print(resultado);
    EXCEPTION WHEN NO_DATA_FOUND THEN
        http.p('Acesso Negado');
    WHEN OTHERS THEN
        resultado:=resultado || '{"Erro": "SQLERRM"}';
        http.print(resultado);

end;
```

## WS\_REGISTAVEICULO

Este web service regista um veículo de um cliente na base de dados.

```
create or replace procedure WS_RegistaVeiculo (v_idmarca number,
v_idmodelo number,
v_marca varchar2, v_modelo varchar2, v_matricula varchar2,
v_idcliente number, v_email varchar2, v_password varchar2) is

id_cliente number;
resultado varchar2(10000);
BEGIN
select idCliente into id_cliente from bd_clientes where
email=v_email and password=v_password;

INSERT INTO BD_VEICULOS
(IDMARCA,IDMODELO,MARCA,MODELO,MATRICULA,CLIENTES_IDCLIENTE)
VALUES
(v_idmarca,v_idmodelo,v_marca,v_modelo,v_matricula,v_idcliente);

resultado:='';
resultado:=resultado || '{"Resposta": "Inserido com sucesso."}';
http.print(resultado);
EXCEPTION
WHEN NO_DATA_FOUND THEN
http.p('Acesso Negado');
WHEN OTHERS
THEN
resultado:= '{"Resposta": "Ocorreu um erro."}';
http.print(resultado);

end;
```

## WS\_VERVEICULOS

Este web service devolve todos os veículos de um utilizador.

```
create or replace procedure WS_VerVeiculos(id_cliente
number,v_email varchar2,v_password varchar2) is

    CURSOR cursor_veiculo IS
        select MARCA,MODELO,MATRICULA,LITROS
        from VeiculosPorClientes
        where CLIENTES_IDCLIENTE = id_cliente;

    c_marca VARCHAR2(255);
    c_modelo VARCHAR2(255);
    c_matricula VARCHAR2(255);
    c_litros VARCHAR2(255);
    v_idcliente number;
    resultado CLOB;
BEGIN
    select idCliente into v_idcliente from bd_clientes where
email=v_email and password=v_password;
    resultado:='[';
    OPEN cursor_veiculo;
    LOOP
        exit when cursor_veiculo%notfound;
        FETCH cursor_veiculo INTO c_marca, c_modelo,
c_matricula,c_litros;
        resultado:=resultado || '{"Marca": "' || c_marca || '", "Modelo":
" '
            || c_modelo || '", "Matricula": "' || c_matricula ||
            '" , "Litros": "' || c_litros || '"},';
    END LOOP;
    CLOSE cursor_veiculo;
    resultado:=substr(resultado,0,length(resultado)-1) || ']';
    http.print(resultado);
    EXCEPTION WHEN NO_DATA_FOUND THEN
        http.p('Acesso Negado');
    WHEN OTHERS THEN
        resultado:=resultado || '{"Erro": "' || SQLERRM || '"}';
        http.print(resultado);
end;
```

## WS\_LOGIN

Este web service devolve a informação de um cliente, se o username e password estiverem corretos.

```
create or replace procedure WS_LOGIN (vemail varchar2, vpassword  
varchar2) is
```

```
    c_idcliente VARCHAR2 (255) :=NULL;  
    c_pnome VARCHAR2 (255) :=NULL;  
    c_unome VARCHAR2 (255) :=NULL;  
    resultado varchar2 (10000) ;  
BEGIN  
    select IDCLIENTE,PRIMEIRONOME,ULTIMONOME  
    into c_idcliente,c_pnome,c_unome  
    from LOGINCLIENTES  
    where email=vemail  
           and password=vpassword;  
  
    resultado:='{ "idCliente": "||c_idcliente  
                ||", "primeiroNome": "||c_pnome  
                ||", "ultimoNome": "||c_unome  
                ||", "email": "||vemail  
                ||", "password": "||vpassword  
                ||"}';  
    http.print(resultado);  
EXCEPTION  
    WHEN no_data_found  
    THEN  
        c_idcliente := 0;  
        c_pnome:='0';  
        c_unome:='0';  
        resultado:='{ "idCliente": "||c_idcliente  
                    ||", "primeiroNome": "||c_pnome  
                    ||", "ultimoNome": "||c_unome  
                    ||"}';  
        http.print(resultado);  
  
end; [2]
```

---

## SINÓNIMOS

---

Criamos estes sinónimos para simplificar nomes de tabelas extensas:

```
CREATE SYNONYM REP_OCUR  
FOR BD_REPARACOES_OCORRENCIAS;
```

```
CREATE SYNONYM OCUR_ESTADO  
For BD_OCORRENCIAS_ESTADOS;
```

---

## FUNÇÕES

---

Esta função foi criada para calcular a média de combustível que gasta um veículo:

```
create or replace function fn_calc_media_combustivel (idveiculo
number)
return NUMBER
is mediaCombustivel number(10);

maxKM number;
sumLitros number;

BEGIN

select max(KM) , sum(LITROS)
into maxKM, sumLitros
from BD_ABASTECIMENTOS, BD_VEICULOS
where VEICULOS_IDVEICULO = idveiculo;

mediaCombustivel := maxKM/sumLitros;
return mediaCombustivel;

END;
```

Esta função encripta e retorna, em SHA-256, um valor passado em parâmetro:

```
create or replace FUNCTION MY_CRYPTO_SHA2
(VALUE IN VARCHAR2) RETURN VARCHAR2 AS
l_hash raw(20000);
BEGIN
l_hash := dbms_crypto.hash ( UTL_I18N.STRING_TO_RAW (value,
'AL32UTF8'), dbms_crypto.hash_sh256);
RETURN to_char(l_hash);
END MY_CRYPTO_SHA2;
```

Esta função retorna o número total de veículos que existem.

```
create or replace function fn_totalVeiculos
return NUMBER
is numeroVeiculos number(10);

BEGIN

Select count(idveiculo)
into numeroVeiculos
from bd_veiculos;

return numeroVeiculos;

END;
```

[3]

Esta função retorna o preço total de uma ocorrência.

```
CREATE OR REPLACE FUNCTION fn_preco_total_reparacao (p_id_ocorrencia
IN number)
RETURN NUMBER AS preco_total number(10,2);

BEGIN
  SELECT
    SUM(quantidade * preco_unitario) INTO preco_total
  FROM BD_REPARACOES_OCORRENCIAS
  WHERE id_ocorrencias = p_id_ocorrencia;
RETURN preco_total;
END;
```

[3]



---

## TRIGGERS

---

Estes são os triggers criados para inserir as chaves primárias de cada tabela:

```
create or replace TRIGGER T_IDCLIENTE
  BEFORE INSERT ON BD_CLIENTES
  FOR EACH ROW
BEGIN
  SELECT seq_idCliente.nextval
    INTO :new.IDCLIENTE
    FROM dual;
END;

create or replace TRIGGER T_IDABASTECIMENTO
  BEFORE INSERT ON BD_ABASTECIMENTOS
  FOR EACH ROW
BEGIN
  SELECT SEQ_IDABASTECIMENTO.nextval
    INTO :new.IDABASTECIMENTO
    FROM dual;
END;

create or replace TRIGGER T_IDANOMALIA
  BEFORE INSERT ON BD_ANOMALIAS
  FOR EACH ROW
BEGIN
  SELECT SEQ_IDANOMALIA.nextval
    INTO :new.IDANOMALIA
    FROM dual;
END;

create or replace TRIGGER T_IDMECANICO
  BEFORE INSERT ON BD_MECANICOS
  FOR EACH ROW
BEGIN
  SELECT SEQ_IDMECANICO.nextval
    INTO :new.IDMECANICO
    FROM dual;
END;

create or replace TRIGGER T_IDOCORRENCIA
  BEFORE INSERT ON BD_OCORRENCIAS
  FOR EACH ROW
BEGIN
  SELECT SEQ_IDOCORRENCIA.nextval
    INTO :new.IDOCORRENCIA
    FROM dual;
END;

create or replace TRIGGER T_IDOCORRENCIAESTADO
  BEFORE INSERT ON BD_OCORRENCIAS_ESTADOS
  FOR EACH ROW
BEGIN
  SELECT SEQ_IDOCORRENCIAESTADO.nextval
    INTO :new.IDOCORRENCIAESTADO
    FROM dual;
END;

create or replace TRIGGER T_IDREPARACAOTIPO
  BEFORE INSERT ON BD_REPARACOES_TIPOS
  FOR EACH ROW
```

```

BEGIN
  SELECT SEQ_IDREPARACAOTIPO.nextval
    INTO :new.IDREPARACAOTIPO
    FROM dual;
END;

create or replace TRIGGER T_IDVEICULO
  BEFORE INSERT ON BD_VEICULOS
  FOR EACH ROW
BEGIN
  SELECT SEQ_IDVEICULO.nextval
    INTO :new.IDVEICULO
    FROM dual;
END;

```

---

## SEGURANÇA

---

Nós decidimos criar três utilizadores únicos que têm acesso à base de dados: Administrador, Mecânicos e Clientes. O Administrador será utilizado para criar a base de dados e será desativado posteriormente. Os privilégios de cada utilizador encontram-se na seguinte tabela CRUD:

Tipo de utilizador da aplicação	Administrador	Mecânicos	Clientes
Utilizador Oracle	oracle_admin	oracle_mecanico	oracle_cliente
BD_OCORRENCIAS	CRUD	CRUD	CRD
BD_OCORRENCIAS_ESTADOS	CRUD	R	R
BD_REPARACOES_OCORRENCIAS	CRUD	CRUD	CRUD
BD_REPARACOES_TIPOS	CRUD	CRUD	R
BD_VEICULOS	CRUD	CRD	CRD
BD_CLIENTES	CRUD	RU	RU
BD_MECHANICOS	CRUD	RU	RU
BD_ANOMALIAS	CRUD	CRUD	CRUD
BD_ABASTECIMENTOS	CRUD	CRUD	CRUD

---

## ROLES

---

Foram criados os seguintes roles, consoante os privilégios anteriormente sugeridos:

```
CREATE role role_clientes;
grant CREATE,SELECT,DELETE on BD_OCORRENCIAS to role_clientes;
grant SELECT on BD_OCORRENCIAS_ESTADOS to role_clientes;
grant CREATE,SELECT,ALTER,DELETE on BD_REPARACOES_OCORRENCIAS to
role_clientes;
grant SELECT on BD_REPARACOES_TIPOS to role_clientes;
grant CREATE,SELECT,DELETE on BD_VEICULOS to role_clientes;
grant SELECT,ALTER on BD_CLIENTES to role_clientes;
grant SELECT,ALTER on BD_MECANICOS to role_clientes;
grant CREATE,SELECT,ALTER,DELETE on BD_ANOMALIAS to role_clientes;
grant CREATE,SELECT,ALTER,DELETE on BD_ABASTECIMENTOS to
role_clientes;

CREATE role role_mecanicos;
grant CREATE,SELECT,ALTER,DELETE on BD_OCORRENCIAS to role_mecanicos;
grant SELECT on BD_OCORRENCIAS_ESTADOS to role_mecanicos;
grant CREATE,SELECT,ALTER,DELETE on BD_REPARACOES_OCORRENCIAS to
role_mecanicos;
grant CREATE,SELECT,ALTER,DELETE on BD_REPARACOES_TIPOS to
role_mecanicos
grant CREATE,SELECT,DELETE on BD_VEICULOS to role_mecanicos;
grant SELECT,ALTER on BD_CLIENTES to role_mecanicos;
grant SELECT,ALTER on BD_MECANICOS to role_mecanicos;
grant CREATE,SELECT,ALTER,DELETE on BD_ANOMALIAS to role_mecanicos;
grant CREATE,SELECT,ALTER,DELETE on BD_ABASTECIMENTOS to
role_mecanicos;

CREATE role role_admin;
grant all on BD_OCORRENCIAS to role_admin;
grant all on BD_OCORRENCIAS_ESTADOS to role_admin;
grant all on BD_REPARACOES_OCORRENCIAS to role_admin;
grant all on BD_REPARACOES_TIPOS to role_admin
grant all on BD_VEICULOS to role_admin;
grant all on BD_CLIENTES to role_admin;
grant all on BD_MECANICOS to role_admin;
grant all on BD_ANOMALIAS to role_admin;
grant all on BD_ABASTECIMENTOS to role_admin; [4]
```

---

## TRANSAÇÕES

---

Nesta secção, apresentamos as transações criadas na nossa base de dados.

A seguinte transação delUser, apaga todos os dados de um utilizador de forma correta. É criado um savepoint no início da transação e, se ocorrer uma exceção, a base de dados faz um rollback até esse savepoint.

```
CREATE or REPLACE PROCEDURE delUser(idUser IN NUMBER) IS

BEGIN

savepoint erro;

DELETE
FROM BD_REPARACOES_OCORRENCIAS
WHERE BD_REPARACOES_OCORRENCIAS.ID_OCORRENCIAS IN
(
SELECT IDOCORRENCIA FROM BD_OCORRENCIAS WHERE CLIENTES_IDCLIENTE =
idUser
);

DELETE FROM BD_ANOMALIAS
where BD_ANOMALIAS.VEICULOS_IDVEICULO IN
(
select idveiculo
from BD_VEICULOS where CLIENTES_IDCLIENTE = idUser
);

DELETE FROM BD_ABASTECIMENTOS
where BD_ABASTECIMENTOS.VEICULOS_IDVEICULO IN
( select idveiculo
from BD_VEICULOS where CLIENTES_IDCLIENTE = idUser
);

DELETE FROM BD_OCORRENCIAS
where CLIENTES_IDCLIENTE = idUser;

DELETE FROM bd_veiculos
where CLIENTES_IDCLIENTE = idUser;

DELETE FROM bd_clientes
where IDCLIENTE = idUser;

exception when others then rollback to erro;

END;
/
```

A seguinte transação apaga uma reparação\_tipo, juntamente com as suas ocorrências.

```
CREATE or REPLACE PROCEDURE delReparacaoTipo(idRepT IN NUMBER) IS

BEGIN

savepoint erro;

DELETE
FROM BD_REPARACOES_OCORRENCIAS
WHERE BD_REPARACOES_OCORRENCIAS.ID_REPARACAO_TIPO = idRepT;

DELETE
FROM BD_REPARACOES_TIPOS
where BD_REPARACOES_TIPOS.IDREPARACAOTIPO = idRepT;

exception when others then rollback to erro;

END;
/
```

---

## IMPLEMENTAÇÃO

---

Até este momento, foram implementadas, numa aplicação Android, as funcionalidades de login, registar um veículo e ver os veículos.

A aplicação deixa os clientes fazerem login com o seu email e password e, através do web service WS\_LOGIN, a aplicação recebe os dados do cliente correspondente.

Depois do login, o cliente pode escolher entre registar um veículo e ver os veículos. Ao registar um veículo, o cliente é apresentado com os modelos e marcas de automóveis, que são devolvidas por uma API externa (<http://fipeapi.appspot.com>). Após seleccionar uma marca, um modelo e introduzir a matrícula, o veículo é registado na base de dados back-end através do web service WS\_REGISTAVEÍCULO.

Ao seleccionar a opção de ver os veículos, a aplicação vai buscar os veículos do cliente utilizando o web service WS\_VERVEICULOS e devolve uma lista com todos os veículos do cliente.

---

## AUDITORIA

---

Foram feitas auditorias automáticas às transações mais críticas da base de dados. Optámos por não fazer auditoria aos SELECTs de forma a não sobrecarregar a base de dados e fizemos auditoria apenas às operações que modificam a base de dados:

```
audit INSERT on bd_abastecimentos by access WHENEVER successful;
audit INSERT on bd_anomalias by access WHENEVER successful;
audit INSERT on bd_clientes by access WHENEVER successful;
audit INSERT on bd_veiculos by access WHENEVER successful;
audit INSERT on bd_mecanicos by access WHENEVER successful;
audit INSERT on bd_reparacoes_tipos by access WHENEVER successful;
audit INSERT on bd_ocorrencias by access WHENEVER successful;
audit INSERT on bd_ocorrencias_estados by access WHENEVER successful;
audit INSERT on bd__reparacoes_ocorrencias by access WHENEVER
successful;
```

```
audit UPDATE on bd_abastecimentos by access WHENEVER successful;
audit UPDATE on bd_anomalias by access WHENEVER successful;
audit UPDATE on bd_clientes by access WHENEVER successful;
audit UPDATE on bd_veiculos by access WHENEVER successful;
audit UPDATE on bd_mecanicos by access WHENEVER successful;
audit UPDATE on bd_reparacoes_tipos by access WHENEVER successful;
audit UPDATE on bd_ocorrencias by access WHENEVER successful;
audit UPDATE on bd_ocorrencias_estados by access WHENEVER successful;
audit UPDATE on bd__reparacoes_ocorrencias by access WHENEVER
successful;
```

```
audit DELETE on bd_abastecimentos by access WHENEVER successful;
audit DELETE on bd_anomalias by access WHENEVER successful;
audit DELETE on bd_clientes by access WHENEVER successful;
audit DELETE on bd_veiculos by access WHENEVER successful;
audit DELETE on bd_mecanicos by access WHENEVER successful;
audit DELETE on bd_reparacoes_tipos by access WHENEVER successful;
audit DELETE on bd_ocorrencias by access WHENEVER successful;
audit DELETE on bd_ocorrencias_estados by access WHENEVER successful;
audit DELETE on bd__reparacoes_ocorrencias by access WHENEVER
successful;
```



---

## CONCLUSÃO

---

Este projeto não só foi uma oportunidade para alargar os nossos conhecimentos em programação para PL/SQL com webservices, segurança na base de dados, permissões e acessos para utilizadores e auditoria de base de dados, mas também uma experiência educativa na implementação destes conceitos.

Apesar de só termos implementados uma pequena parte da aplicação, temos a certeza que aplicaremos estes conceitos no futuro, quer na interação entre a aplicação e a base de dados, quer na integridade da própria base de dados.

---

## BIBLIOGRAFIA

---

- [1] D. C. Fonseca, “Encriptação,” Guarda.
  - [2] “Managing Views,” Oracle, [Online]. Available:  
[https://docs.oracle.com/cd/B28359\\_01/server.111/b28310/views001.htm#ADMIN11774](https://docs.oracle.com/cd/B28359_01/server.111/b28310/views001.htm#ADMIN11774).  
[Acedido em 01 02 2017].
  - [3] D. C. Fonseca, “Oracle Access a Soap WebServices,” Guarda.
  - [4] D. C. Fonseca, “Privilégios e Roles”.
- 
-