



## Fase3 - Cap 8 - Crie funções

### Crie suas próprias funções no Oracle

Proposta de automação

#### Integrantes

Elio Gonçalves de Lima - [elio.designer@hotmail.com](mailto:elio.designer@hotmail.com)

Andrei Vedovato Alves Rodrigues - [vedovato.andrei@gmail.com](mailto:vedovato.andrei@gmail.com)

Luis Carvalho dos Santos Junior - [luiss225@hotmail.com](mailto:luiss225@hotmail.com)

Vinicios Koji Yamamoto - [viniciosyamamoto@gmail.com](mailto:viniciosyamamoto@gmail.com)

RAFAELA ROSSO PACHECO - [rafaelarosso2011@hotmail.com](mailto:rafaelarosso2011@hotmail.com)

#### Github


GitHub - elioglima/fiap-fase3-cap8-funcoes-psql

MASCARA DE CEP, MASCARA DE RG, MASCARA DE CPF, MASCARA DE CNPJ, MASCARA DE TELEFONE, MASCARA DE CELULAR, FUNCAO PARA CALCULAR A IDADE FUNCAO PARA FORMATAR CPF FUNCAO PARA FORMATAR CNPJ TRIGGER

<https://github.com/elioglima/fiap-fase3-cap8-funcoes-psql.git>

elioglima/**fiap-fase3-cap8-funcoes-psql**

Contributor 1 Issues 0 Stars 0 Forks 0



#### Download Script SQL


fiap-fase3-cap8-funcoes-psql/script.sql at main · elioglima/fiap-fase3-cap8-funcoes-psql

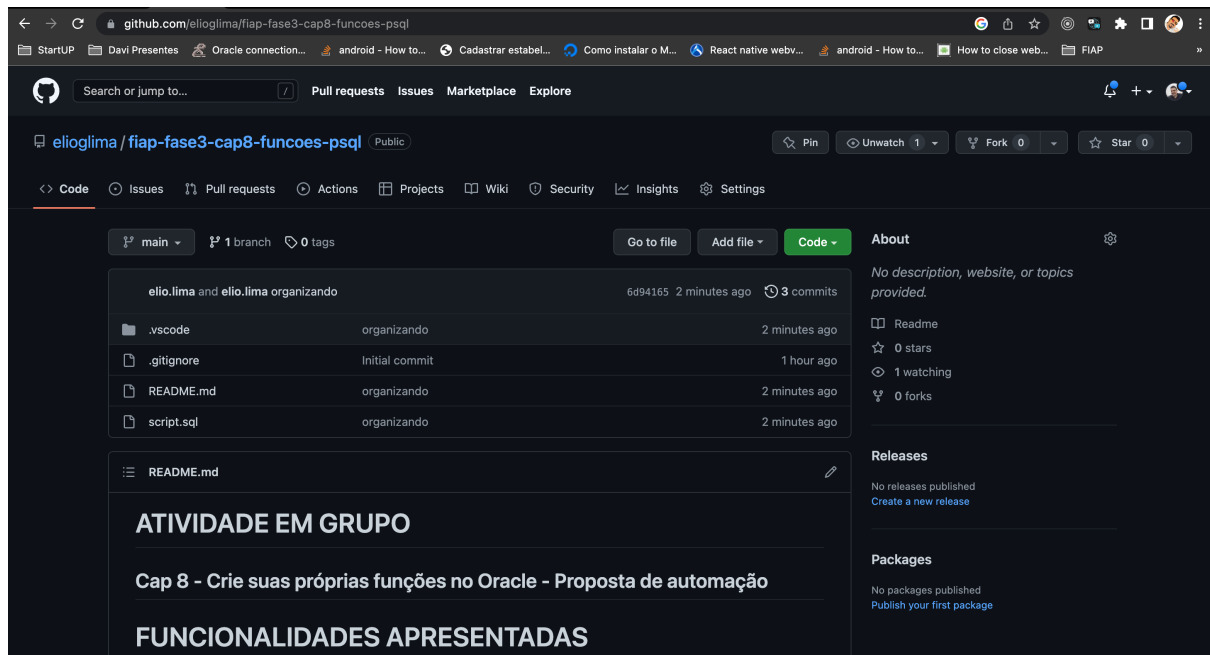
This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters. Learn more about bidirectional Unicode characters You can't perform that action at this time. You

<https://github.com/elioglima/fiap-fase3-cap8-funcoes-psql/blob/main/script.sql>

elioglima/**fiap-fase3-cap8-funcoes-psql**

Contributor 1 Issues 0 Stars 0 Forks 0





## FUNÇÕES

### Função genérica de formatação

```
CREATE OR REPLACE FUNCTION MASCARA
  (val IN VARCHAR2, mask IN VARCHAR2)
  RETURN VARCHAR2 AS masked VARCHAR(100) DEFAULT ' ';

  k INT DEFAULT 0;
  i INT DEFAULT 0;

  BEGIN WHILE i < LENGTH(mask) LOOP i: = i + 1;
    IF SUBSTR(mask, i, 1) = '#' THEN
      IF k < LENGTH(val) THEN k: = k + 1;
        masked: = CONCAT(masked, SUBSTR(val, k, 1));
      END IF;

    ELSE
      IF i < LENGTH(mask) THEN
        masked: = CONCAT(masked, SUBSTR(mask, i, 1));
      END IF;

    END IF;
  END LOOP;

  RETURN masked;

END MASCARA;
```

```
# FUNCAO GENERICA DE MASCARA

- MASK("12345123", "#####-###") AS cep,
- MASK("121231231", "###.###.###-#") AS rg,
- MASK("1212312312", "###.###.###-##") AS cpf,
```

```

- MASK("12123123123412", "##.###.###/####-##") AS cnpj,
- MASK("1212341234", "(##) ####-####") AS telefone,
- MASK("12123451234", "(##) #####-####") AS celular

```

**CEP:** #####-###

Função que converte um texto de um numero de cep para um cep formatado.

**RG:** ##.###.###-#

Função que converte um texto de um numero de cep para um cep formatado.

**CPF:** ##.###.###-##

Função que converte um texto de um numero de cep para um cep formatado.

**CNPJ:** ##.###.###/####-##

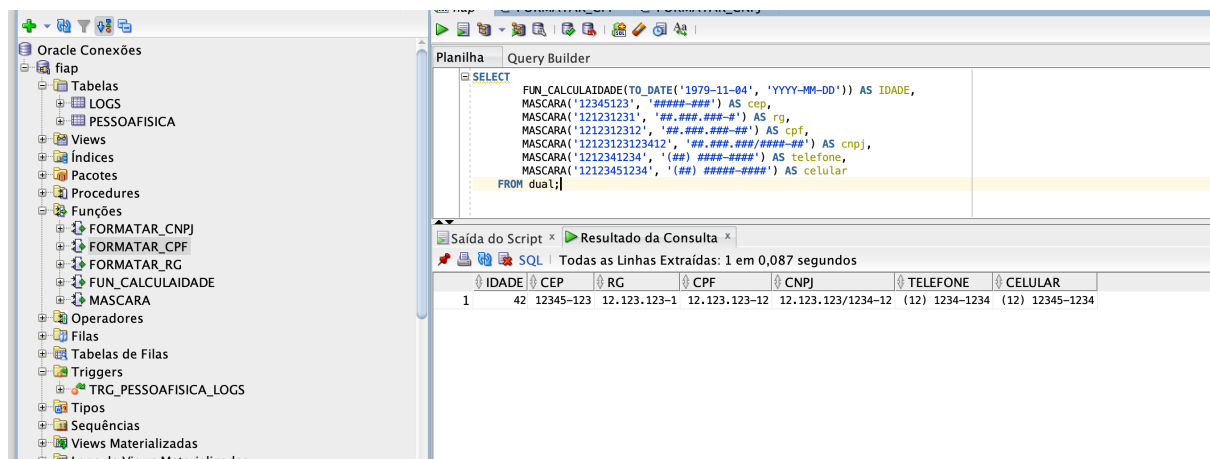
Função que converte um texto de um numero de cep para um cep formatado.

**TELEFONE:** #####-####

Função que converte um texto de um numero de cep para um cep formatado.

**CELULAR:** #####-####

Função que converte um texto de um numero de cep para um cep formatado.



## Criação das tabelas

# TABELA DE CADASTRO DE PESSOA FISICA

```

CREATE TABLE PESSOAFISICA (
  ID NUMBER(*, 0) GENERATED ALWAYS AS IDENTITY INCREMENT BY 1 MAXVALUE 999999 MINVALUE 1 CACHE 20 NOT NULL,
  NOME VARCHAR2(100 BYTE) NOT NULL,
  CPF VARCHAR2(20 BYTE) NOT NULL,
  RG VARCHAR2(20 BYTE) NOT NULL,
  TELEFONE VARCHAR2(15 BYTE) NOT NULL,
  CELULAR VARCHAR2(15 BYTE) NOT NULL,
  DATA_NASCIMENTO DATE NOT NULL,
  CONSTRAINT PESSOAFISICA_PK PRIMARY KEY (ID)
)
USING INDEX (
  CREATE UNIQUE INDEX PESSOAFISICA_PK ON PESSOAFISICA (ID ASC)
  LOGGING TABLESPACE TBSPC_ALUNOS
  PCTFREE 10
  INITRANS 2
  STORAGE (INITIAL 65536 NEXT 1048576
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED BUFFER_POOL DEFAULT)
)

```

```

        ) NOPARALLEL
    ) ENABLE
) LOGGING TABLESPACE TBSPC_ALUNOS
  PCTFREE 10
  INITRANS 1
  STORAGE (INITIAL 65536 NEXT 1048576
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED BUFFER_POOL DEFAULT
  ) NOCOMPRESS NO INMEMORY NOPARALLEL;

# TABELA DE LOGS
CREATE TABLE LOGS (
    ID NUMBER(*, 0) GENERATED ALWAYS AS IDENTITY INCREMENT BY 1 MAXVALUE 999999 MINVALUE 1 CACHE 20 NOT NULL,
    DESCRICAO VARCHAR2(1000 BYTE) NOT NULL,
    FUNCIONALIDADE VARCHAR2(45 BYTE) NOT NULL,
    DATA TIMESTAMP(6) DEFAULT CURRENT_TIMESTAMP NOT NULL,
    ID_REFERENCIA NUMBER(*, 0)
) LOGGING TABLESPACE TBSPC_ALUNOS PCTFREE 10 INITRANS 1 STORAGE (
  INITIAL 65536 NEXT 1048576
  MINEXTENTS 1 MAXEXTENTS UNLIMITED BUFFER_POOL DEFAULT
) NOCOMPRESS NO INMEMORY NOPARALLEL;

```

## Criação de Trigger`s

```

# TRIGGER DE REGISTRO DE LOGS ADICIONAR
CREATE OR REPLACE TRIGGER TRG_PESSOA FISICA_LOGS_ADICIONA
  AFTER INSERT ON PESSOA FISICA FOR EACH ROW BEGIN

  INSERT INTO
    LOGS (FUNCIONALIDADE, ID_REFERENCIA, DESCRICAO, DATA)
  VALUES (
    'NOVO CADASTRO DE PESSOA FISICA',
    :new.ID,
    :new.NOME,
    CURRENT_TIMESTAMP
  );

END;

# TRIGGER DE REGISTRO DE LOGS REMOVE
CREATE OR REPLACE TRIGGER TRG_PESSOA FISICA_LOGS_REMOVE
  AFTER DELETE ON PESSOA FISICA FOR EACH ROW BEGIN

  INSERT INTO
    LOGS (FUNCIONALIDADE, ID_REFERENCIA, DESCRICAO, DATA)
  VALUES
    (
      'REMOVE CADASTRO DE PESSOA FISICA',
      :old.ID,
      :old.NOME,
      CURRENT_TIMESTAMP
    );

END;

# TRIGGER DE REGISTRO DE LOGS ATUALIZA
CREATE OR REPLACE TRIGGER TRG_PESSOA FISICA_LOGS_ATUALIZA
  AFTER UPDATE ON PESSOA FISICA FOR EACH ROW BEGIN

  INSERT INTO
    LOGS (FUNCIONALIDADE, ID_REFERENCIA, DESCRICAO, DATA)
  VALUES (
    'NOVO CADASTRO DE PESSOA FISICA',
    :new.ID,
    :new.NOME,
    CURRENT_TIMESTAMP
  );

END;

```

```

INSERT INTO
    pessoafisica (nome, cpf, rg, telefone, celular, data_nascimento)
values
    (
        'Joao Almenda Brandao',
        '64443404031',
        '451089923',
        '1120823541',
        '11952550331',
        TO_DATE('1979-11-04', 'YYYY-MM-DD')
    );

INSERT INTO
    pessoafisica (nome, cpf, rg, telefone, celular, data_nascimento)
values
    (
        'Elisa Gomes da Silva',
        '98771181067',
        '228949609',
        '1120823541',
        '11952550331',
        TO_DATE('1986-03-11', 'YYYY-MM-DD')
    );

INSERT INTO
    pessoafisica (nome, cpf, rg, telefone, celular, data_nascimento)
values
    (
        'Pedro Alencar dos Santos',
        '05955141030',
        '358771663',
        '1120823541',
        '11952550331',
        TO_DATE('1955-09-07', 'YYYY-MM-DD')
    );

INSERT INTO
    pessoafisica (nome, cpf, rg, telefone, celular, data_nascimento)
values
    (
        'Abigail da Costa',
        '78766609089',
        '355373415',
        '1120823541',
        '11952550331',
        TO_DATE('1965-02-01', 'YYYY-MM-DD')
    );

```

## Consultas

```

# VISUALIZA OS LOGS

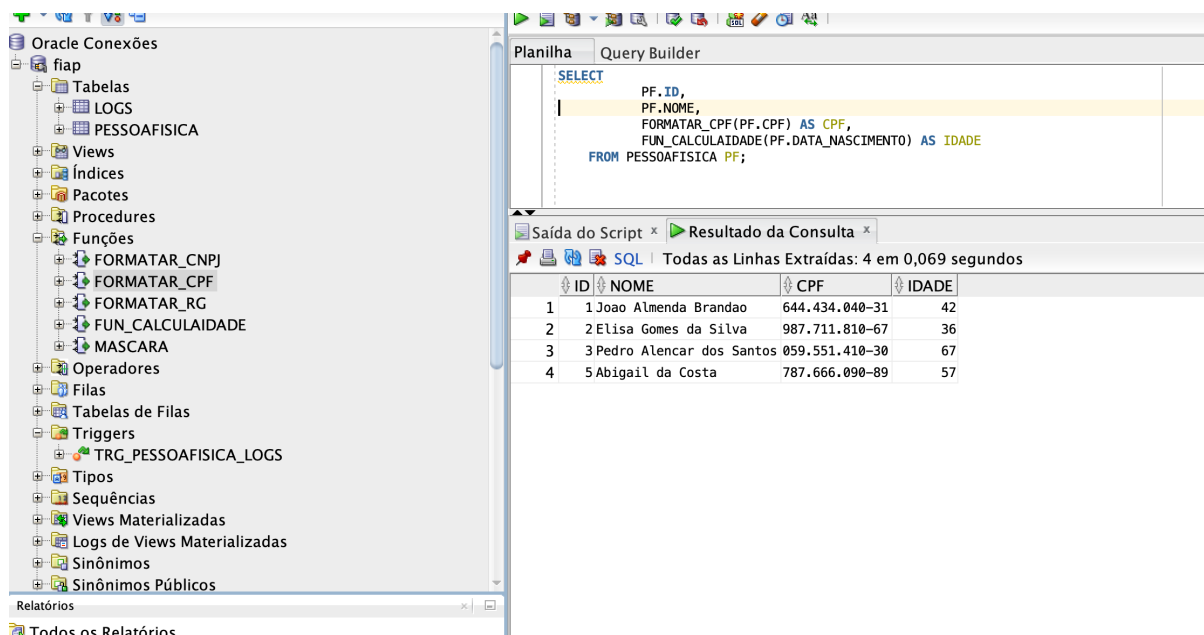
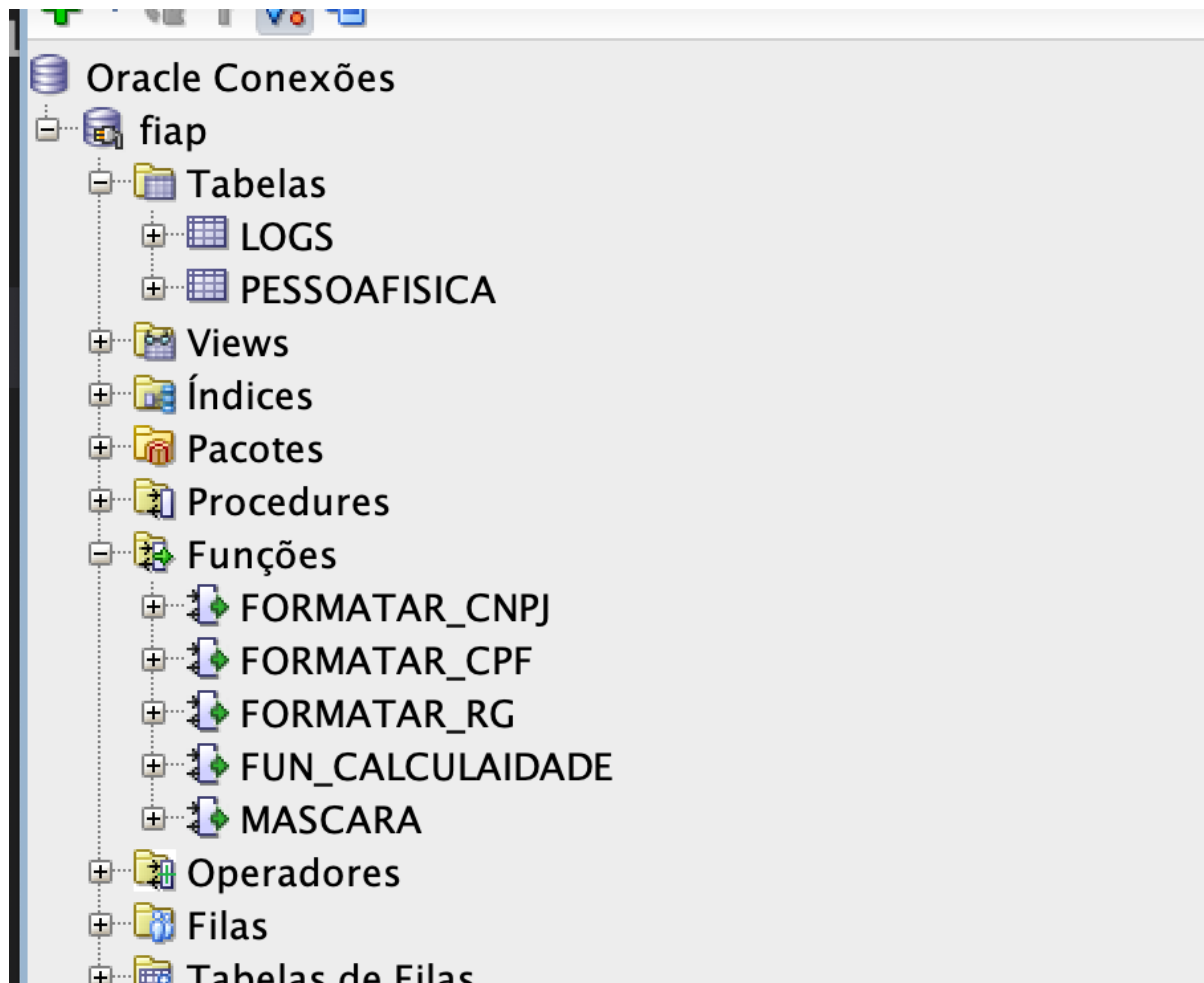
SELECT l.* FROM logs l;

# LISTA PESSOA FISICA

SELECT
    PF.*,
    FUN_CALCULAIDADE(PF.DATA_NASCIMENTO) AS IDADE
FROM PESSOAFISICA PF;

```

## Evidências e testes



Oracle Conexões

- fiap
  - Tabelas
    - LOGS
    - PESSOAFISICA
  - Views
  - Índices
  - Pacotes
  - Procedures
  - Funções
    - FORMATAR\_CNPJ
    - FORMATAR\_CPF
    - FORMATAR\_RG
    - FUN\_CALCULAIDADE
    - MASCARA
  - Operadores
  - Filas
  - Tabelas de Filas
  - Triggers
    - TRG\_PESSOA FISICA\_LOGS
  - Tipos
  - Sequências
  - Views Materializadas
  - Links de Views Materializadas

Planilha Query Builder

```

SELECT
  FUN_CALCULAIDADE(TO_DATE('1979-11-04', 'YYYY-MM-DD')) AS IDADE,
  MASCARA('12345123', '#####-###') AS cep,
  MASCARA('121231231', '##.###.###-#') AS rg,
  MASCARA('1212312312', '##.###.###-##') AS cpf,
  MASCARA('12123123123412', '##.###.###/####-##') AS cnpj,
  MASCARA('1212341234', '(##) ####-####') AS telefone,
  MASCARA('12123451234', '(##) #####-####') AS celular
FROM dual;

```

Saída do Script x Resultado da Consulta x

Todas as Linhas Extraídas: 1 em 0,087 segundos

IDADE	CEP	RG	CPF	CNPJ	TELEFONE	CELULAR
1	42 12345-123	12.123.123-1	12.123.123-12	12.123.123/1234-12	(12) 1234-1234	(12) 12345-1234