

**Mestrado em Engenharia Informática**

**Unidade Curricular Complementar**

**Base de Dados NoSQL**

Ficha de Exercícios 02

4º Ano, 1º Semestre

Ano letivo 2020/2021

Diogo Alexandre Rodrigues Lopes

PG42823

Joel Costa Carvalho

PG42837

Ana Margarida da Rocha Ferreira

PG44412

Bruno Xavier Brás dos Santos

PG44414

**Conteúdo**

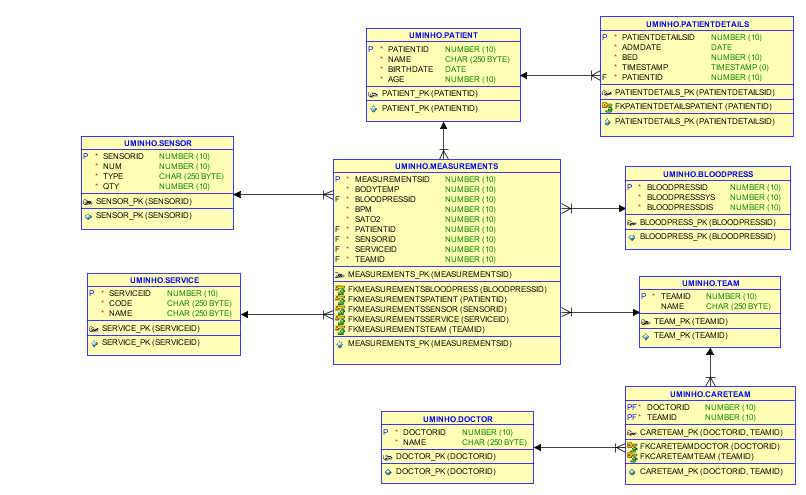
[Tarefa 1. Desenho de um esquema relacional que albergue os dados recolhidos por cada um dos sensores.](#_heading=h.gjdgxs) 3

[Tarefa 2. Criação do schema para albergar os dados, recorrendo ao container Oracle.](#_heading=h.3soz06oyfkn9) 3

[Tarefa 3. Criação das tabelas necessárias para acolher os dados tendo em conta o diagrama relacional definido, recorrendo ao container Oracle.](#_heading=h.36eg4b9w53b3) 3

[Tarefa 4. Construção de um programa, recorrendo à linguagem de programação Python, capaz de ler dados da API REST e guardar na base de dados criada.](#_heading=h.47cjrc50nag2) 7

# **Tarefa 1.** Desenho de um esquema relacional que albergue os dados recolhidos por cada um dos sensores.



# **Tarefa 2.** Criação do schema para albergar os dados, recorrendo ao container Oracle.

CREATE TABLESPACE uminho\_tables DATAFILE 'UMINHO\_FILES\_01.dbf' SIZE 500m;

CREATE USER uminho IDENTIFIED BY "uminho2020" DEFAULT TABLESPACE uminho\_tables QUOTA UNLIMITED ON uminho\_tables;

GRANT CONNECT, RESOURCE, CREATE VIEW, CREATE SEQUENCE TO uminho;

# **Tarefa 3.** Criação das tabelas necessárias para acolher os dados tendo em conta o diagrama relacional definido, recorrendo ao container Oracle.

CREATE TABLE BloodPress (

BloodPressID number(10) NOT NULL,

BloodPressSys number(10) NOT NULL,

BloodPressDis number(10) NOT NULL,

CONSTRAINT BloodPress\_PK PRIMARY KEY (BloodPressID));

CREATE TABLE Service (

ServiceID number(10) NOT NULL,

Code char(250) NOT NULL,

Name char(250) NOT NULL,

CONSTRAINT Service\_PK PRIMARY KEY (ServiceID));

CREATE TABLE Doctor (

DoctorID number(10) NOT NULL,

Name char(250) NOT NULL,

CONSTRAINT Doctor\_PK PRIMARY KEY (DoctorID));

CREATE TABLE CareTeam (

DoctorID number(10) NOT NULL,

TeamID number(10) NOT NULL,

CONSTRAINT CareTeam\_PK PRIMARY KEY (DoctorID,

TeamID));

CREATE TABLE Measurements (

MeasurementsID number(10) NOT NULL,

BodyTemp number(10) NOT NULL,

BloodPressID number(10) NOT NULL,

BPM number(10) NOT NULL,

Sato2 number(10) NOT NULL,

PatientID number(10) NOT NULL,

SensorID number(10) NOT NULL,

ServiceID number(10) NOT NULL,

TeamID number(10) NOT NULL,

CONSTRAINT Measurements\_PK PRIMARY KEY (MeasurementsID));

CREATE TABLE Patient (

PatientID number(10) NOT NULL,

Name char(250) NOT NULL,

Birthdate date NOT NULL,

Age number(10) NOT NULL,

CONSTRAINT Patient\_PK PRIMARY KEY (PatientID));

CREATE TABLE PatientDetails (

PatientDetailsID number(10) NOT NULL,

AdmDate date NOT NULL,

Bed number(10) NOT NULL,

TimeStamp timestamp(0) NOT NULL,

PatientID number(10) NOT NULL,

CONSTRAINT PatientDetails\_PK PRIMARY KEY (PatientDetailsID));

CREATE TABLE Sensor (

SensorID number(10) NOT NULL,

Num number(10) NOT NULL,

Type char(250) NOT NULL,

Qty number(10) NOT NULL,

CONSTRAINT Sensor\_PK PRIMARY KEY (SensorID));

CREATE TABLE Team (

TeamID number(10) NOT NULL,

Name char(250),

CONSTRAINT Team\_PK PRIMARY KEY (TeamID));

ALTER TABLE Measurements ADD CONSTRAINT FKMeasurementsBloodPress FOREIGN KEY (BloodPressID) REFERENCES BloodPress (BloodPressID);

ALTER TABLE Measurements ADD CONSTRAINT FKMeasurementsSensor FOREIGN KEY (SensorID) REFERENCES Sensor (SensorID);

ALTER TABLE Measurements ADD CONSTRAINT FKMeasurementsService FOREIGN KEY (ServiceID) REFERENCES Service (ServiceID);

ALTER TABLE Measurements ADD CONSTRAINT FKMeasurementsTeam FOREIGN KEY (TeamID) REFERENCES Team (TeamID);

ALTER TABLE Measurements ADD CONSTRAINT FKMeasurementsPatient FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE CareTeam ADD CONSTRAINT FKCareTeamDoctor FOREIGN KEY (DoctorID) REFERENCES Doctor (DoctorID);

ALTER TABLE CareTeam ADD CONSTRAINT FKCareTeamTeam FOREIGN KEY (TeamID) REFERENCES Team (TeamID);

ALTER TABLE PatientDetails ADD CONSTRAINT FKPatientDetailsPatient FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

CREATE SEQUENCE BloodPress\_sequence;

CREATE OR REPLACE TRIGGER BloodPress\_on\_insert

BEFORE INSERT ON BloodPress

FOR EACH ROW

BEGIN

SELECT BloodPress\_sequence.nextval

INTO :new.bloodpressid

FROM dual;

END;

CREATE SEQUENCE Service\_sequence;

CREATE OR REPLACE TRIGGER Service\_on\_insert

BEFORE INSERT ON Service

FOR EACH ROW

BEGIN

SELECT Service\_sequence.nextval

INTO :new.serviceid

FROM dual;

END;

CREATE SEQUENCE Measurements\_sequence;

CREATE OR REPLACE TRIGGER Measurements\_on\_insert

BEFORE INSERT ON Measurements

FOR EACH ROW

BEGIN

SELECT Measurements\_sequence.nextval

INTO :new.measurementsid

FROM dual;

END;

CREATE SEQUENCE PatientDetails\_sequence;

CREATE OR REPLACE TRIGGER PatientDetails\_on\_insert

BEFORE INSERT ON PatientDetails

FOR EACH ROW

BEGIN

SELECT PatientDetails\_sequence.nextval

INTO :new.patientdetailsid

FROM dual;

END;

CREATE SEQUENCE Team\_sequence;

CREATE OR REPLACE TRIGGER Team\_on\_insert

BEFORE INSERT ON Team

FOR EACH ROW

BEGIN

SELECT Team\_sequence.nextval

INTO :new.teamid

FROM dual;

END;

# **Tarefa 4.** Construção de um programa, recorrendo à linguagem de programação Python, capaz de ler dados da API REST e guardar na base de dados criada.

O programa desenvolvido encontra-se no ficheiro anexado com o nome noSQL\_FE02\_GR07.py.