Instituto Politécnico do Cávado e do Ave
Escola Superior Tecnologia

Sistema Diagnosticar o Stress como uma doença

Trabalho efetuado pelo grupo:

23548 Ana Pinto

21206 Flávio Carvalho

21211 João Sampaio

26 de novembro de 2023

Esta página foi intencionalmente deixada em branco.

Resumo

O principal objetivo do trabalho realizado na UC de Armazenamento e Acesso a Dados é criar uma plataforma que permita diagnosticar o *stress* como uma doença.

Apresenta-se no restante documento uma exposição sobre as principais ideias estabelecidas para o projeto.

Índice

Índice de Figuras

Índice de Tabelas

Não foi encontrada nenhuma entrada do índice de ilustrações.

Lista de siglas e acrónimos

- ER: Entidade Relação;
- TI: Tecnologias de Informação;
- UC: Unidade Curricular;

Introdução

Este projeto enquadra-se na Unidade Curricular de Armazenamento e Acesso a Dados, lecionada pelo docente Joaquim Gonçalves, no curso de Engenharia Informática Médica no Instituto Politécnico do Cávado e do Ave, tendo como objetivo a criação de uma plataforma que permite diagnosticar o *stress* como uma doença. Ademais, irá incorporar os conteúdos lecionados em aula nesta disciplina, assim como algum conhecimento através de pesquisa.

Este trabalho prático tem controlo de versão disponível no *GitHub*.

Objetivos

Neste tópico é descrito os objetivos do trabalho proposto. As finalidades incluem guardar a informação sobre um conjunto de dados através de dispositivos com sensores, tais como frequência cardíaca, temperatura da pele, frequência respiratória, entre outros; permitir o registo de respostas a questionários específicos que serão também utilizados como informação para a consulta.

Estrutura documento

Relativamente à estrutura do documento, irá ser dividido em dois tópicos principais, sendo estes Diagrama Entidade Relação e a Implementação em *PostgreSQL*, no programa *pgAdmin4*. No tópico sobre o Diagrama ER, irá ser demonstrado o diagrama, a sua descrição e explicação. Por último, na Implementação, irá ser demonstrado exemplos das criações das tabelas, inserção de dados, entre outras funcionalidades.

Ferramentas utilizadas

- **Discord** – Ferramenta para comunicar com pessoas, rápida e fácil. Conversar por voz, vídeo e texto.

- **WhatsApp** – Ferramenta que permite enviar mensagens privadas, fiáveis e gratuitas para todo o mundo.

- **Visual Paradigm** – Ferramenta que fornece um conjunto de instrumentos de *design*, análise e gerenciamento para impulsionar o desenvolvimento de projetos de TI e transformação digital.

- **Trello** – Ferramenta que permite a consolidação e reunião de tarefas, colegas de equipa e instrumentos. Estes são mantidos em um só lugar para que a equipa de trabalho tenha acesso às tarefas.

- **PostgreSQL** – Ferramenta que consiste num sistema de base de dados de código aberto.

1. Diagrama Entidade Relação

Neste tópico iremos apresentar e descrever o diagrama entidade relação que desenvolvemos para o projeto.

Apresentação do diagrama ER

Com o intuito de criar uma base de dados que guarde a informação relevante para diagnosticar o *stress* como uma doença, desenhou-se o diagrama Entidade Relação apresentado na Figura 1.

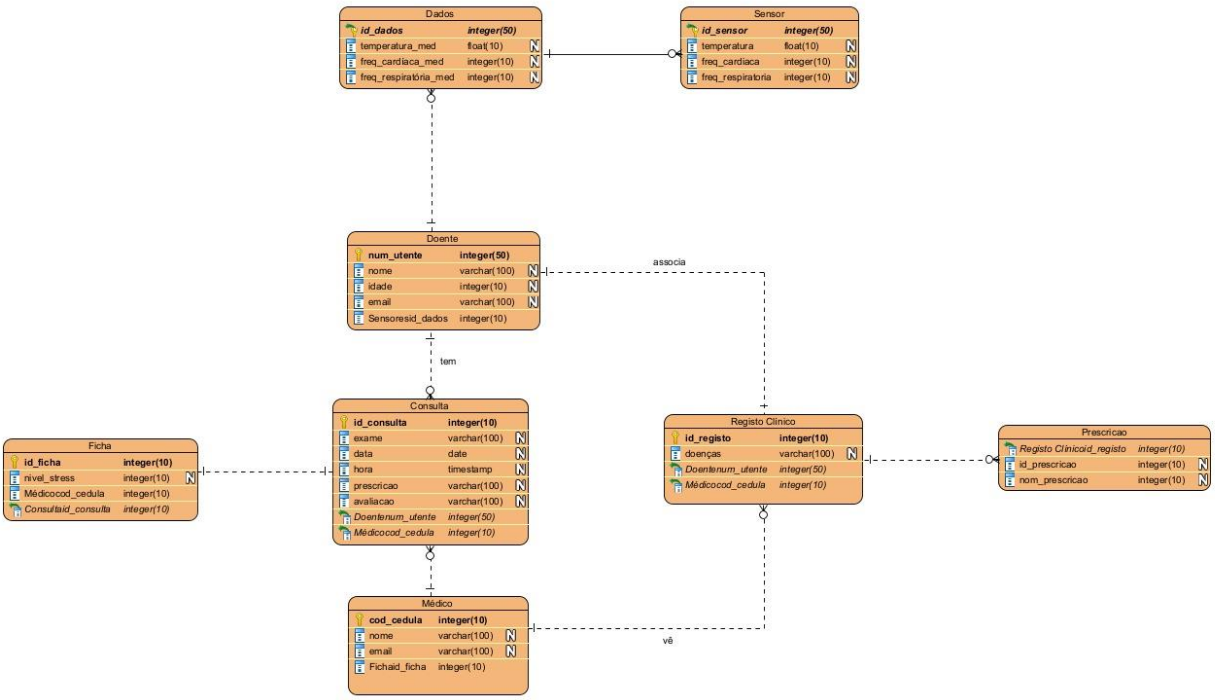

Figura 1 - Diagrama ER

Descrição

A estrutura de dados representada na Figura 1, contempla toda a área de armazenamento dos dados relativos ao *stress*, desde a utilização dos sensores para obter os dados dos doentes assim como a marcação de uma consulta.

Inicialmente, obtém-se os dados de um doente como, por exemplo, medidas de temperatura, frequência cardíaca e respiratória. Estes dados são obtidos por sensores. Os mesmos estão associados a um doente, sendo que cada um é identificado pelo número de utente. Após uma consulta, que é realizada numa data específica e respetiva hora, e tem um médico associado, identificado pelo seu código de cédula. Ademais, o médico têm acesso ao registo clínico do doente, para poder ver mais informações como exames já feitos, prescrições já passadas e doenças diagnosticadas. Anotações e descrições são registadas numa ficha técnica efetuadas pelo médico bem como o nível de *stress*, que o doente apresenta.

2. Implementação

Neste capítulo é descrito o trabalho de implementação, salientando as funcionalidades mais relevantes para o projeto. Em cada subcapítulo é descrito e explicado as mesmas.

Create Tables

Para a criação das tabelas, foi utilizada a declaração "*CREATE TABLE*" da seguinte forma:

```
--Criar tabela Sensor
CREATE TABLE "TP_AAD".Sensor (
    id_sensor integer PRIMARY KEY,
    temperatura float,
    freq_cardiaca integer,
    freq_respiratoria integer
);

--Criar tabela Dados
CREATE TABLE "TP_AAD".Dados (
    id_dados integer PRIMARY KEY,
    temperatura_med float,
    freq_cardiaca_med integer,
    freq_respiratoria_med integer,
    id_sensor integer REFERENCES "TP_AAD".Sensor(id_sensor) ON DELETE SET NULL
);

-- Criar tabela Doente
CREATE TABLE "TP_AAD".Doente (
    num_utente integer PRIMARY KEY,
    nome varchar(100),
    idade integer,
    email varchar(100),
    Sensoresid_dados integer REFERENCES "TP_AAD".Dados(id_dados) ON DELETE SET NULL
);

-- Criar tabela Ficha
CREATE TABLE "TP_AAD".Ficha (
    id_ficha integer PRIMARY KEY,
    nivel_stress integer,
    Medico_cod_cedula integer,
    Consulta_id_consulta integer
    -- As chaves estrangeiras para Medico e Consulta sao adicionadas mais tarde
depois de criar essas tabelas
);

-- Criar tabela Medico
CREATE TABLE "TP_AAD".Medico (
    cod_cedula integer PRIMARY KEY,
    nome varchar(100),
    email varchar(100),
    Fichaid_ficha integer REFERENCES "TP_AAD".Ficha(id_ficha) ON DELETE SET NULL
);

-- Dicionar chave estrangeira em Ficha que faz referencia a Medico
ALTER TABLE "TP_AAD".Ficha ADD CONSTRAINT fk_medico FOREIGN KEY (Medico_cod_cedula)
REFERENCES "TP_AAD".Medico(cod_cedula) ON DELETE SET NULL;

-- Criar tabela Consulta
```

```
CREATE TABLE "TP_AAD".Consulta (
    id_consulta integer PRIMARY KEY,
    exame varchar(100),
    data date,
    hora timestamp,
    prescricao varchar(100),
    avaliacao varchar(100),
    Doente_num_utente integer REFERENCES "TP_AAD".Doente(num_utente) ON DELETE
CASCADE,
    Medico_cod_cedula integer REFERENCES "TP_AAD".Medico(cod_cedula) ON DELETE SET
NULL
);

-- Addicionar chave estrangeira em Ficha que faz referencia a Consulta
ALTER TABLE "TP_AAD".Ficha ADD CONSTRAINT fk_consulta FOREIGN KEY
(Consulta_id_consulta) REFERENCES "TP_AAD".Consulta(id_consulta) ON DELETE SET NULL;

-- Criar tabela Registo Clinico
CREATE TABLE "TP_AAD".Registro_Clinico (
    id_registro integer PRIMARY KEY,
    doencas varchar(100),
    Doente_num_utente integer REFERENCES "TP_AAD".Doente(num_utente) ON DELETE
CASCADE,
    Medico_cod_cedula integer REFERENCES "TP_AAD".Medico(cod_cedula) ON DELETE SET
NULL
);

-- Criar tabela Prescricao
CREATE TABLE "TP_AAD".Prescricao (
    id_prescricao integer PRIMARY KEY,
    nom_prescricao VARCHAR(20),
    Registro_Clinicoid_registro integer REFERENCES
"TP_AAD".Registro_Clinico(id_registro) ON DELETE CASCADE
);
```

Pode-se observar também que foi necessário adicionar as chaves estrangeiras para tabelas que contem ligações com outras tabelas.

```
CREATE TABLE "TP_AAD".Doente (
    num_utente integer PRIMARY KEY,
    nome varchar(100),
    idade integer,
    email varchar(100),
    Sensoresid_dados integer REFERENCES "TP_AAD".Dados(id_dados) ON DELETE SET NULL
);
```

Por exemplo, na tabela “Doente” tem uma referência para “Dados” visto que o doente será associado com os seus dados recolhidos, para isso, tem de se usar a chave estrangeira de “Dados” para fazer essa ligação.

Inserção de Dados

Para inserir dados nas tabelas criadas, utiliza-se a declaração "*INSERT INTO*" da seguinte forma:

```
-- Inserir dados na tabela Sensor
INSERT INTO "TP_AAD".Sensor (id_sensor, temperatura, freq_cardiaca,
freq_respiratoria) VALUES
(1, 36.5, 70, 16),
(2, 37.2, 80, 18),
(3, 36.8, 65, 20),
(4, 37.5, 85, 22);

-- Inserir dados na tabela Dados
INSERT INTO "TP_AAD".Dados (id_dados, temperatura_med, freq_cardiaca_med,
freq_respiratoria_med, id_sensor) VALUES
(1, 36.7, 72, 17, 1),
(2, 37.0, 78, 19, 2),
(3, 36.9, 68, 21, 3),
(4, 37.3, 82, 20, 4);

-- Inserir dados na tabela Doente
INSERT INTO "TP_AAD".Doente (num_utente, nome, idade, email, Sensoresid_dados)
VALUES
(12345, 'João Silva', 30, 'joaosilva@example.com', 1),
(67890, 'Maria Souza', 25, 'mariasouza@example.com', 2),
(34567, 'Carlos Teixeira', 40, 'carlosteixeira@example.com', 3),
(45678, 'Ana Gomes', 35, 'anagomes@example.com', 4);

-- Inserir dados na tabela Ficha
-- Nota: Inserir depois de `Medico` and `Consulta` porque tem chaves estrangeiras
dessas tabelas.

-- Inserir dados na tabela Medico
INSERT INTO "TP_AAD".Medico (cod_cedula, nome, email) VALUES
(111, 'Dr. Ana Pereira', 'anapereira@example.com'),
(222, 'Dr. Lucas Martins', 'lucasmartins@example.com'),
(333, 'Dr. Sofia Cardoso', 'sofiacardoso@example.com'),
(444, 'Dr. Miguel Neto', 'miguelneto@example.com');

-- Inserir dados na tabela Consulta
INSERT INTO "TP_AAD".Consulta (id_consulta, exame, data, hora, prescricao, avaliacao,
Doente_num_utente, Medico_cod_cedula) VALUES
(1, 'Exame de sangue', '2024-01-10', '2024-01-10 09:00:00', 'Vitamin D', 'Normal',
12345, 111),
(2, 'Raio X', '2024-01-11', '2024-01-11 14:00:00', 'Calcium', 'Fratura detectada',
67890, 222),
(3, 'Eletrocardiograma', '2024-02-15', '2024-02-15 10:30:00', 'Beta-blocker',
'Batimentos irregulares', 34567, 333),
(4, 'Teste de esforço', '2024-02-20', '2024-02-20 11:00:00', 'Rehydration', 'Bom
desempenho', 45678, 444);

-- Inserir dados na tabela Ficha
INSERT INTO "TP_AAD".Ficha (id_ficha, nivel_stress, Medico_cod_cedula,
Consulta_id_consulta) VALUES
(1, 5, 111, 1),
(2, 3, 222, 2),
(3, 4, 333, 3),
```

```
(4, 2, 444, 4);

-- Inserir dados na tabela Registo Clinico
INSERT INTO "TP_AAD".Registro_Clinico (id_registro, doencas, Doente_num_utente,
Medico_cod_cedula) VALUES
(1, 'Hipertensão', 12345, 111),
(2, 'Asma', 67890, 222),
(3, 'Diabetes', 34567, 333),
(4, 'Colesterol Alto', 45678, 444);

-- Inserir dados na tabela Prescricao
INSERT INTO "TP_AAD".Prescricao (id_prescricao, nom_prescricao,
Registro_Clinicoid_registro) VALUES
(1, 'Benuron', 1),
(2, 'Brufen', 2),
(3, 'Ilvico', 3),
(4, 'Brufen', 4);
```

Aqui pode-se observar que se tem todos os dados inseridos na base de dados.

Queries

Para verificar o bom funcionamento, foram criadas algumas *queries* para uma melhor análise dos dados inseridos bem como as tabelas.
Essas *queries* podem ser utilizadas para identificar problemas com os doentes em consultas, por exemplo, verificar quando um doente apresenta frequências cardíacas demasiado altas.

```
-- Doentes com dados de sensores que excedem as médias críticas
SELECT
    D.num_utente,
    D.nome AS patient_name,
    DS.id_dados,
    DS.temperatura_med AS avg_temperatura,
    DS.freq_cardiaca_med AS avg_freq_cardiaca,
    DS.freq_respiratoria_med AS avg_freq_respiratoria
FROM "TP_AAD".Doente D
JOIN "TP_AAD".Dados DS ON D.Sensoresid_dados = DS.id_dados
WHERE
    DS.temperatura_med > 38.5 OR
    DS.freq_cardiaca_med > 100 OR
    DS.freq_respiratoria_med > 20;
```

	num_utente integer	patient_name character varying (100)	id_dados integer	avg_temperatura double precision	avg_freq_cardiaca integer	avg_freq_respiratoria integer
1	34567	Carlos Teixeira	3	36.9	68	21

Figura 2 - Doente com dados críticos

Com esta *querie*, foi possível identificar um doente que apresente uma frequência respiratória acima do normal.

```
--Dados das consultas de cada doente
SELECT
  D.nome AS nome_doente,
  D.email AS email_doente,
  C.data AS data_consulta,
  C.hora AS tempo_consulta,
  C.avaliacao AS avaliacao
FROM
  "TP_AAD".Doente D
JOIN "TP_AAD".Consulta C ON D.num_utente = C.Doente_num_utente
WHERE
  C.data = (SELECT MAX(data) FROM "TP_AAD".Consulta WHERE Doente_num_utente =
D.num_utente);
```

	nome_doente character varying (100)	email_doente character varying (100)	data_consulta date	tempo_consulta timestamp without time zone	avaliacao character varying (100)
1	João Silva	joaosilva@example.com	2024-01-10	2024-01-10 09:00:00	Normal
2	Maria Souza	mariasouza@example.com	2024-01-11	2024-01-11 14:00:00	Fratura detectada
3	Carlos Teixeira	carlosteixeira@example.com	2024-02-15	2024-02-15 10:30:00	Batimentos irregulares
4	Ana Gomes	anagomes@example.com	2024-02-20	2024-02-20 11:00:00	Bom desempenho

Figura 3 - Dados de Consultas

Com esta *querie*, consegue-se obter dados sobre as consultas de cada doente, como o nome do doente que teve a consulta, o seu email, a data da consulta, a hora em que a consulta foi feita e a avaliação médica feita.

```
--Todas as consultas com doentes e doutores incluindo dados dos sensores durante a
consulta
SELECT
  D.nome AS nome_doente,
  M.nome AS nome_doutor,
  C.data AS data_consulta,
  Da.temperatura_med AS dados_temperatura,
  Da.freq_cardiaca_med AS dados_freq_cardiaca,
  Da.freq_respiratoria_med AS dados_freq_respiratoria
FROM
  "TP_AAD".Consulta C
JOIN "TP_AAD".Doente D ON C.Doente_num_utente = D.num_utente
JOIN "TP_AAD".Medico M ON C.Medico_cod_cedula = M.cod_cedula
JOIN "TP_AAD".Dados Da ON D.Sensoresid_dados = Da.id_dados
JOIN "TP_AAD".Sensor S ON Da.id_sensor = S.id_sensor;
```

	nome_doente character varying (100)	nome_doutor character varying (100)	data_consulta date	dados_temperatura double precision	dados_freq_cardiaca integer	dados_freq_respiratoria integer
1	João Silva	Dr. Ana Pereira	2024-01-10	36.7	72	17
2	Maria Souza	Dr. Lucas Martins	2024-01-11	37	78	19
3	Carlos Teixeira	Dr. Sofia Cardoso	2024-02-15	36.9	68	21
4	Ana Gomes	Dr. Miguel Neto	2024-02-20	37.3	82	20

Figura 7 - Dados sobre médicos, doentes e sensores

Com esta *querie,* consegue-se ver as consultas com os doentes, médicos e os dados recolhidos pelos sensores de cada doente.

3. Conclusão

Este trabalho teve como objetivo consolidar e praticar todos os conhecimentos adquiridos ao longo das aulas lecionadas, tendo a funcionalidade de tentar interligar todo esse conhecimento. Sendo assim, este projeto permitiu perceber melhor o funcionamento em relação a todos os objetivos que este trabalho prático exigiu, conseguindo uma melhor compreensão e consolidação dos conteúdos abordados pelo docente. Ademais, foi um trabalho que nos exigiu algum tempo, uma vez que foram surgindo várias dificuldades que foram ultrapassadas e nunca deixadas para depois, investindo-se para obter resultados que, no final, fossem os desejados. Foi um trabalho muito estimulante e bastante desafiante de realizar.

Os objetivos propostos para foram cumpridos, com ajuda de alguma pesquisa e consulta por parte do cada elemento do grupo. É de realçar que foram encontradas algumas adversidades, como mencionado anteriormente, ultrapassadas com auxílio do docente da UC e alguma pesquisa.

Em suma, teve-se uma melhor perceção do que se aborda na cadeira de Armazenamento e Acesso a Dados que, num futuro próximo, poderá ser bastante útil.

4. Webgrafia

https://www.visualparadigm.com/support/documents/vpuserguide/3563/3564/85375_drawngentit.html

https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento

https://www.w3schools.com/postgresql/postgresql_create_table.php

https://www.w3schools.com/postgresql/postgresql_select.php