

UNIVERSIDADE PAULISTA – UNIP

CURSO: TEC ANALISE DES/ SISTEMAS	COORDENAÇÃO: PROF. CRISTIANO
TUTMA: DS2A34/DS1A34	TURNIO: MATUTINO

DISCIPLINA: Matemática para Computação **PROFESSOR:** Cristiano Nascimento

EQUIPE	RA
FELIPE MATEUS B NASCIMENTO	N372HD-8
JOSE R C CARDOSO JUNIOR	N256BJ-3
LEVY DE LIRA GOMES	D7533H-9
LISVANETE RODRIGUES GARCIA	N2572H-7
VENNER MANOEL BERTINO	N33618-8
WILLIAM BENJAMIM M SAMPAIO	D7534B-7

EXERCICIOS – NP1

Para a NP1, foi solicitado a criação de uma algoritmo que:

1. Imprima o nome dos integrantes da equipe;
2. Imprima (desenha na tela) alguns robôs do filme *Star-Wars*;
3. Chame um sistema de medição de IMC, em que o resultado deve ser armazenado em um arquivo ou em uma Base de Dados;
4. Sistema que leia temperatura e humidade e escreva em uma Base de Dados com data e hora da entrada;
5. Sistema de conexão com Base de Dados.

A linguagem de programação escolhida para desenvolver o trabalho foi o C. O SGBD escolhido foi o MySQL, utilizamos também a biblioteca mysql de conexão com o banco de dados em C. Para escrever os códigos utilizamos a IDE Code::Blocks com o compilador GNU GCC Compiler.

Todos os exercícios foram feitos em um só programa, e podem ser chamados através de um menu simples. Foi criada uma biblioteca chamada "*biblioteca.h*" no qual é responsável por armazenar todas as funções e procedimentos do programa, constantes, variáveis globais e estruturas de dados.

Dentro do diretório raiz do programa está a pasta "*/sound*", onde fica um executável com nome "*sound.exe*", também escrito em C e tem a função de reproduzir um áudio "*Blue (Da Ba Dee) – Eiffel 65.wav*" e escrever um arquivo com o nome "*status*" que armazena o PID de sua execução. É executado logo quando o programa principal é iniciado:

```
WinExec("sound/sound.exe", SW_HIDE) ;
```

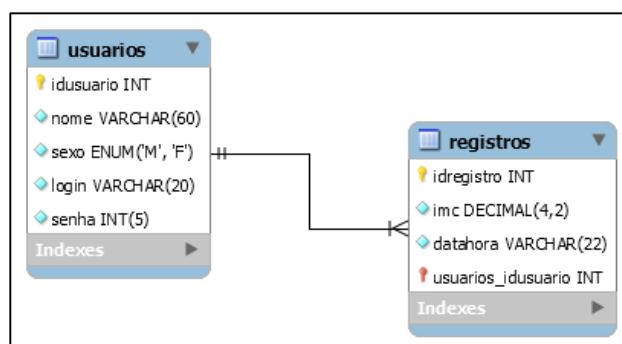
O primeiro parâmetro define o diretório de "*sound.exe*", o segundo parâmetro oculta a tela do programa "*sound.exe*", ou seja ele 'roda' em segundo plano. É importante explicar que quando "*sound.exe*" é aberto ele passa a ser executado de forma independente mas como um processo 'filho' do programa principal, ou seja,

apesar de não 'depender' do programa principal, se este for finalizado ele também é finalizado.

O primeiro exercício foi realizado sem nenhuma dificuldade. Foi criada uma procedimento em que é impresso o nome dos integrantes da equipe, e etc.

O segundo exercício também foi realizado sem nenhuma dificuldade, porem utilizamos para criar os desenhos de caracteres um software chamado *ASCII Generator v2.0.0* (Disponível em: <http://ascgendotnet.jmsoftware.co.uk> e <http://ascgen2.sourceforge.net>).

O terceiro exercício, Sistema de IMC, foi o foco principal do desenvolvimento. Foi criado um banco de dados com o nome "**imc01**", que possui duas tabelas:



A primeira tabela armazena os usuários, a segunda o armazena os registros de imc que são referenciados aos seus respectivos usuários através de uma *Chave-Estrangeira*.

idusuario	nome	sexo	login	senha
1	BONNER BONNER	F	bonner	123
3	ANTONIO SILVA	M	silva	123

idregistro	usuarios_idusuario	imc	datahora
19	1	15.00	Oct 23 2018 - 00:40:52
20	1	15.43	Oct 23 2018 - 00:42:37

Acima dois registros na tabela usuários, abaixo dois registros na tabela registros.

Foi criada as seguintes funcionalidades para o sistema IMC:

1. Entrar (login);
2. Cadastre-se (cria uma conta);

Após login, dentro do sistema o usuário 'logado' pode:

1. Registrar novo IMC;
2. Exibir todos os registros (IMC referentes a ele);
3. Alterar dados pessoais;
4. Voltar ao menu principal (do sistema IMC);
5. Sair;
6. Apagar cadastro.

O quinto e último exercício, sistema de conexão com o banco de dados, é solucionado através de 3 funções em nosso programa:

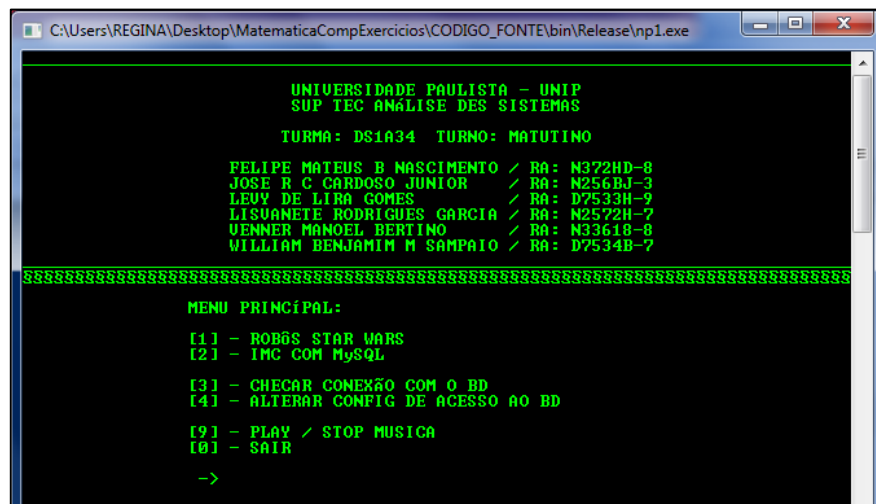
void _BD_checarCon(void) - Verifica conexão com banco de dados, e mostra através de uma caixa de mensagem o status da conexão, se foi possível estabelecer conexão ou não.

void _BD_configura(void) - Caso o arquivo “**configdb**”, que contém as configurações de acesso ao banco de dados, não exista ele cria e solicita ao usuário as configurações e escreve no arquivo.

void _BD_pegarConfig(BASEDEDADOS* bd) - Tem a função de pegar as configurações de acesso do arquivo “**configdb**”, recebe como parâmetro um ponteiro do tipo “**BASEDEDADOS**” onde serão guardadas por referência as informações coletadas do arquivo “**configdb**”.

void _BD_altConfig(void) - Edita, altera as configurações de acesso presentes no arquivo “**configdb**”.

Outra funcionalidade do programa é iniciar / parar a música por meio de uma função que verifica o PID no arquivo “**sound/status**”. Caso o PID seja zero o “**sound.exe**” é executado, caso contrário o processo “**sound.exe**” é finalizado através de seu PID.



Tela principal do programa.

*Obs: o quarto exercício **não foi feito pela equipe**, pois além da falta de recursos a equipe não conseguiu a tempo resolver o problema proposto, de tratar o recebimento dos números binários referentes a temperatura e humidade e armazená-los na Base de Dados.