

**FACULDADE DE CIÊNCIAS E TECNOLOGIAS**

**MIEEC – MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES**

**-- Base de Dados--**

**Projeto (2ª Parte):**

**“Songs”**

****

**Docente:** Pedro Nuno San-Bento Furtado

**Realizado por:**

**Manuel Filipe Ribeiro Restolho Mateus Nº:2012169738   
Miguel Soares Maranha Tiago Nº:2012138309**

1. **Introdução**

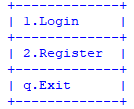
Nesta segunda parte do projeto da unidade curricular de Base de Dados, pretende-se implementar a base de dados normalizada obtida na primeira parte do trabalho.

Optamos por utilizar como ferramentas de trabalho o software Phyton onde foi feito todo o código, juntamente com o software XAMPP para podermos ligar o código com o a base de dados.

Para podermos ver e editar a base de dados usamos o site <http://localhost/phpmyadmin>.

Esta base de dados tem como objetivo principal conseguir-se organizar e gerir informação sobre músicas.

1. **Descrição da Aplicação- Manual do Utilizador**

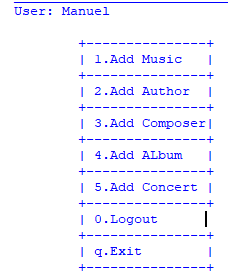
A aplicação desenvolvida ajuda a organizar e gerir a informação de várias músicas e por exemplo, ajudar um utilizador que esteja à procura de uma música específica e saber em que album está disponível.

Inicialmente a consola pede um login e caso o utilizador não esteja registado pede o seu registo, chamando a função(int\_main).

No caso do utilizador não estar registado, selecionando a opção “2”, o interface pede o nome, o username e uma password encriptada\*.

Caso já esteja registado é feito o login (opção “1”), com um username e password.

De seguida é nos apresentado o segundo menu, chamado através da função (editor\_menu):

Como demonstrado na imagem ao lado, apenas um utilizador registado e declarado com editor pode adicionar músicas, autores, compositores, álbuns, concertos.

Optamos por ter também uma opção de logout caso queiramos entrar com outro utilizador.

Em caso de logout, volta para o menu da imagem anterior.

Caso o utilizador não seja declarado com editor é lhe apresentado o seguinte menu (user\_ menu):

Neste caso, qualquer utilizador desde que esteja registado pode procurar músicas, autores, compositores, álbuns, concertos, fazer uma critica, criar uma playlist ou ver playlist.

Tal como no caso anterior também temos uma opção de logout caso queiramos entrar com outro utilizador.

Em caso de logout, volta para o menu inicial.

1. **Descrição da Aplicação- Manual de Instalação**

open mysql and import the result from ONDA.com for <bd6\_5estrelas.json>`

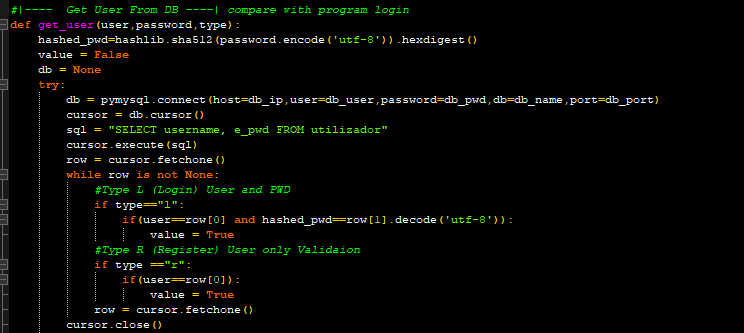
Install python3

pip3 install pymysql

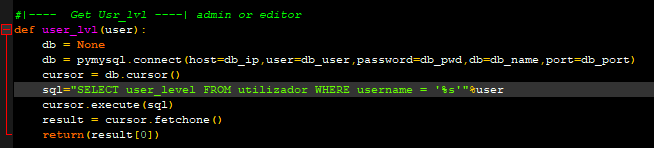
1. **Diagramas ER**

Optamos por fazer algumas alterações às tabelas através das linhas de código:

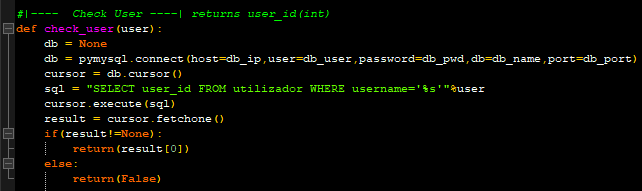
* ALTER TABLE `musica` CHANGE `tempo\_da\_musica` `tempo\_da\_musica` TIME NULL DEFAULT NULL;
* ALTER TABLE `membros` CHANGE `mebro\_id` `membro\_id` BIGINT(20) NOT NULL AUTO\_INCREMENT;
* ALTER TABLE album CHANGE ano ano YEAR NULL DEFAULT NULL;
* ALTER TABLE `album\_review` ADD `album\_album\_id` BIGINT(20) NOT NULL AFTER `data`, ADD INDEX `album\_review\_fk1` (`album\_album\_id`) USING BTREE;
* ALTER TABLE `album\_review` ADD FOREIGN KEY (`album\_album\_id`) REFERENCES `album`(`album\_id`) ON DELETE RESTRICT ON UPDATE CASCADE;
* ALTER TABLE `album\_review` ADD `utilizador\_user\_id` BIGINT(20) NOT NULL AFTER `album\_album\_id`, ADD INDEX `album\_review\_fk2` (`utilizador\_user\_id`) USING BTREE;
* ALTER TABLE `album\_review` ADD FOREIGN KEY (`utilizador\_user\_id`) REFERENCES `utilizador`(`user\_id`) ON DELETE RESTRICT ON UPDATE CASCADE;

1. **Algumas Funções**
   1. **Get\_user**

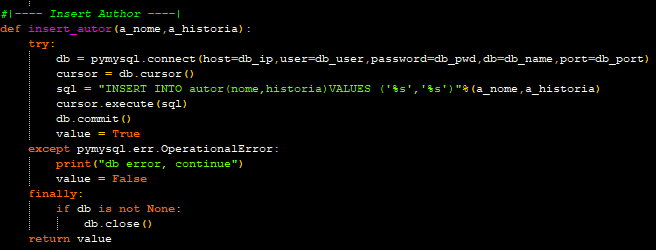
Dado o username, password e tipo faz uma comparação quando tipo=l indica que é um login e compara os parâmetros dados com os da base de dados. Quando é do tipo=r apenas verifica se o utilizador já está registado, devolvendo o valor (“true”).

* 1. **User\_lvl**

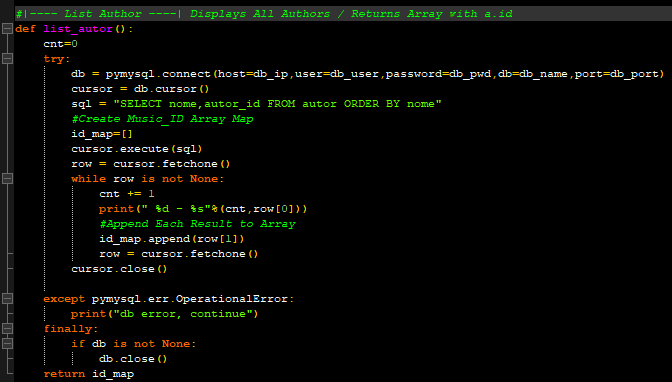
Vai a base de dados buscar o nível de edição á tabela utilizador onde o username é igual ao parâmetro passado(user). Esta retorna o valor que está na base de dados que é editor ou autor.

* 1. **Check\_user**

Pesquisa na tabela utilizador o user\_id= username dado por parâmetro, se for diferente de NULL, devolve o user\_id caso contrário devolve (“false”).

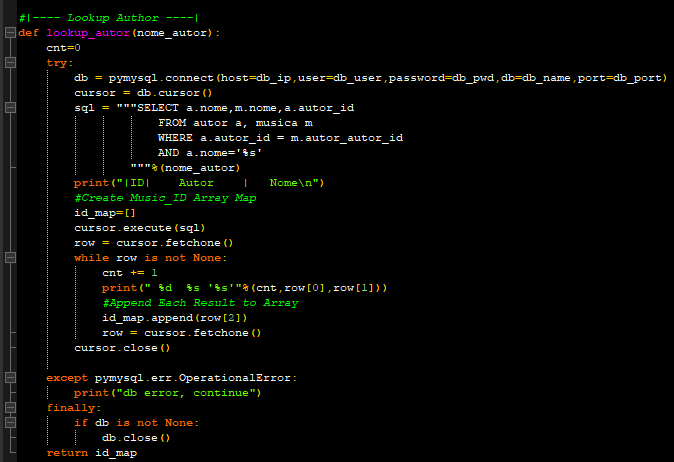
* 1. **Insert\_autor**

Insere na tabela autor os valores, nome de autor e história.

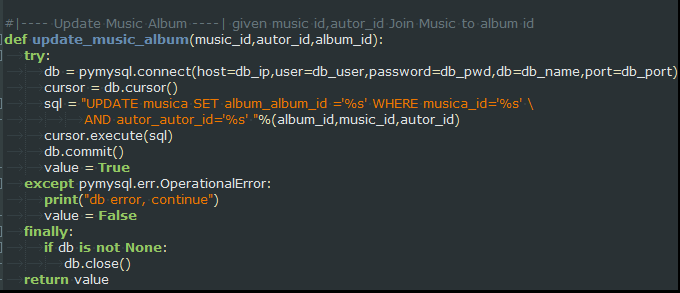
* 1. **List\_autor**

Vai à tabela autor, buscar o nome e o utilizador\_id e faz display a todos os resultados ordenados por nome.

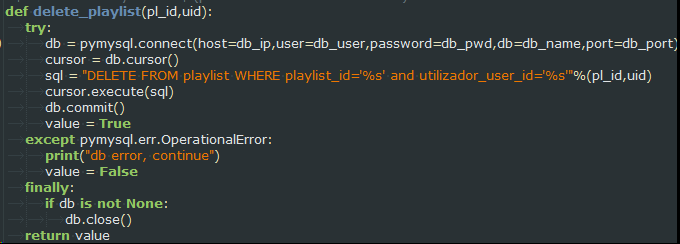
* 1. **lookup\_autor**

Seleciona o nome do autor, o nome da música e o autor\_id da tabela musica e autor com autor\_id=autor\_autor\_id que vêm da tabela música.

* 1. **Update\_music\_album**

Vai á tabela música inserir um album, onde a musica\_id e o autor\_id são passados por parâmetros. Isto é adiciona músicas ao album.

* 1. **Delete\_playlist**



Vai à tabela playlist procurar o user\_id e a playlist\_id dados por parâmetros e apaga os dados respectivos.

1. **Conclusão**

Neste trabalho teve-se a oportunidade de aprender vários conceitos e aplicá-los, relacionados com *PHP*, *Phyton*, *MySQL*.

A falta tempo fez com que a aplicação não tivesse muitas funcionalidades como poderiam ter sido implementados, mas o balanço continua a ser positivo, visto que se adquiriu muitos conhecimentos relacionados com a unidade curricular.