



Relatório de Projeto

CET de Redes e Gestão de Sistemas de Informáticos

pyTunes

Frederico Mateus Martins

5087, 5117, 5118, 5199 - Introdução à Programação

Formador: João Galamba

2015

Introdução

Este trabalho foi desenvolvido no âmbito dos módulos 5091, 5117, 5118 e 5119 de Introdução à Programação, para o professor João Galamba. O objetivo principal requisitado pelo professor foi a elaboração de duas aplicações, utilizando a linguagem de programação Python, de modo a fazer um *media player*, que recebesse músicas por *stream* de um servidor e as reproduzisse no cliente.

Deste modo o trabalho teve que ser dividido em duas partes, o lado servidor, que gere e envia a informação consoante o requisitado pelo cliente, e o cliente, que recebe as informações do servidor e reproduz música de forma eficiente e segura.

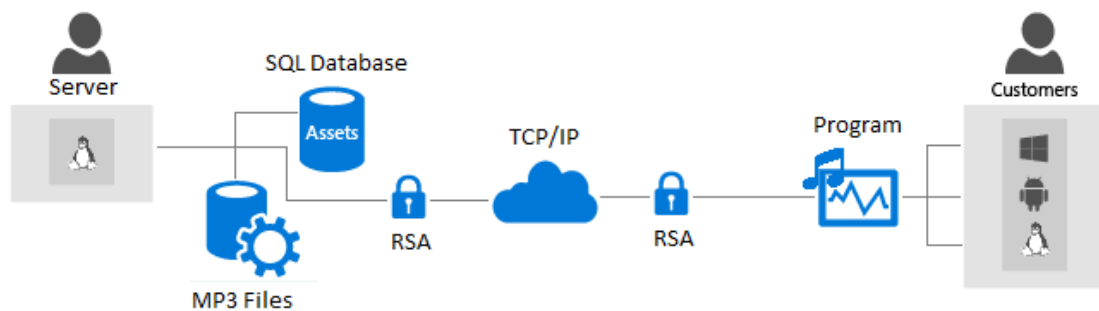


Figura 1 - Arquitetura da ligação entre Cliente e Servidor

Servidor

Compatibilidade

O servidor desenvolveu-se com principal objetivo de correr em sistemas operativos baseados em GNU/Linux e servir diferentes clientes simultaneamente. Este possui uma consola de administração acessível através de um terminal que permite fazer diferentes interações. Esta está dividida maioritariamente em duas partes, a comunicação através de TCP/IP com um pequeno protocolo desenvolvido sobre TCP e a base de dados que organiza e gere a biblioteca de músicas e as diferentes contas de utilizador.

Através da comunicação é possível acabar com determinadas ligações clientes, verificar as diferentes ligações existentes atualmente e desligar/ligar todas comunicações futuras e existentes temporariamente para depuração de erros ou motivos de segurança.

Na parte da base de dados é possível introduzir novas músicas manualmente (apesar deste ir buscar automaticamente as músicas quando o programa é iniciado), inserir ou modificar utilizadores registados, modificar *playlists* de determinados utilizadores, entre muitos outros comandos possíveis.

A parte da base de dados foi utilizado o módulo built-in de Python, SQLite3. Que nos possibilitou uma ligação eficaz entre o programa do servidor e as tabelas da base de dados.

```

root@formando-vb: /home/frederico/pyTunes Server
Bem-vindo ao pyTunes Server Administration Console

Último login: 2015-09-16 14:40:38
Usar exit, para sair do programa.
Usar help, para ajuda sobre o programa.
SQLite3 3.8.7.4 v.

root# select * from sqlite_master where type='table'
-----
| type | name | tbl_name | rootpage | sql |
-----
| table | Users | Users | 2 | CREATE TABLE Users (Username text primary key not NULL, Password text not NULL, Email text) |
| table | Library | Library | 4 | CREATE TABLE Library (ID integer primary key, Artist text, Music text, Album text, Duration text, Directory text) |
| table | Playlist | Playlist | 5 | CREATE TABLE Playlist (ID integer primary key, User text, Name text) |
| table | Track | Track | 6 | CREATE TABLE Track (ID integer primary key, LibrabryID integer, PlaylistID integer) |
-----
Tempo: 0.00139 s

root#
  
```

Figura 2 - Tabelas principais existentes na base de dados

Servidor

Segurança

A comunicação entre o servidor e os clientes é feita por *sockets*, enviando pacotes de 1024 bytes e recebendo a mesma quantidade, pacotes estes que se encontram encriptados por encriptação assimétrica RSA, com a exceção do envio de ficheiros que são enviados em aberto.

Por questões de segurança o servidor não guarda nenhuma palavra-chave sem fazer antes o *hash* (SHA512) da mesma, impossibilitando assim o conhecimento dessas mesmas palavras-chaves para possíveis intrusos. São ainda guardados registos de logins feitos ao servidor e clientes com as datas e horas, IP e tentativas de login.

É também guardado um histórico dos comandos usados na consola de administração por questões de possíveis lapsos da parte do administrador ou ainda acessos desconhecidos

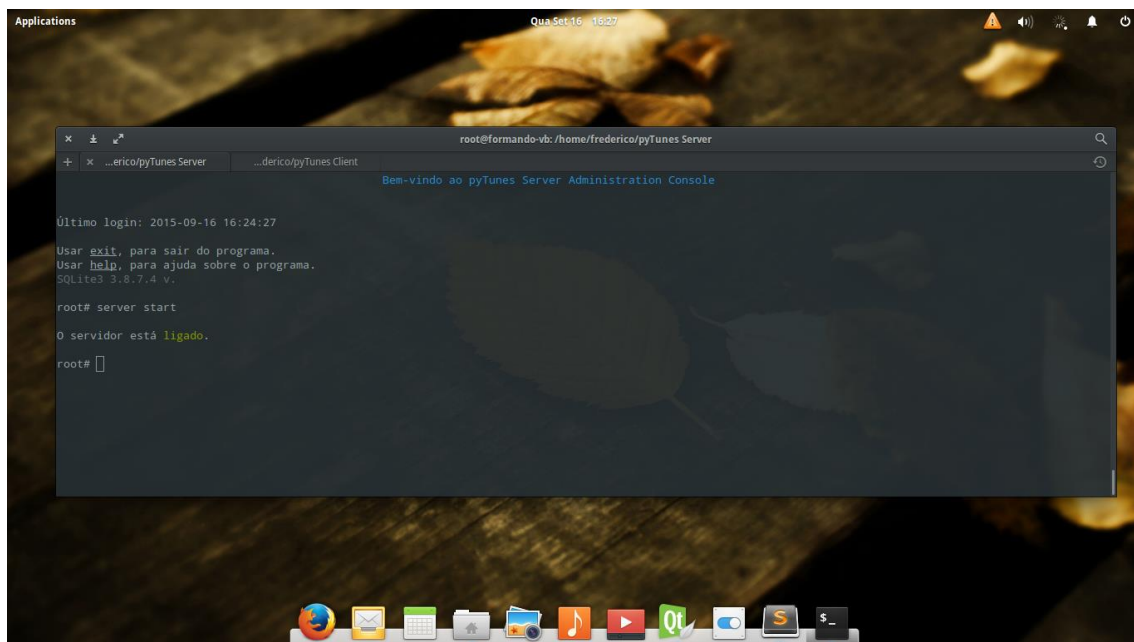
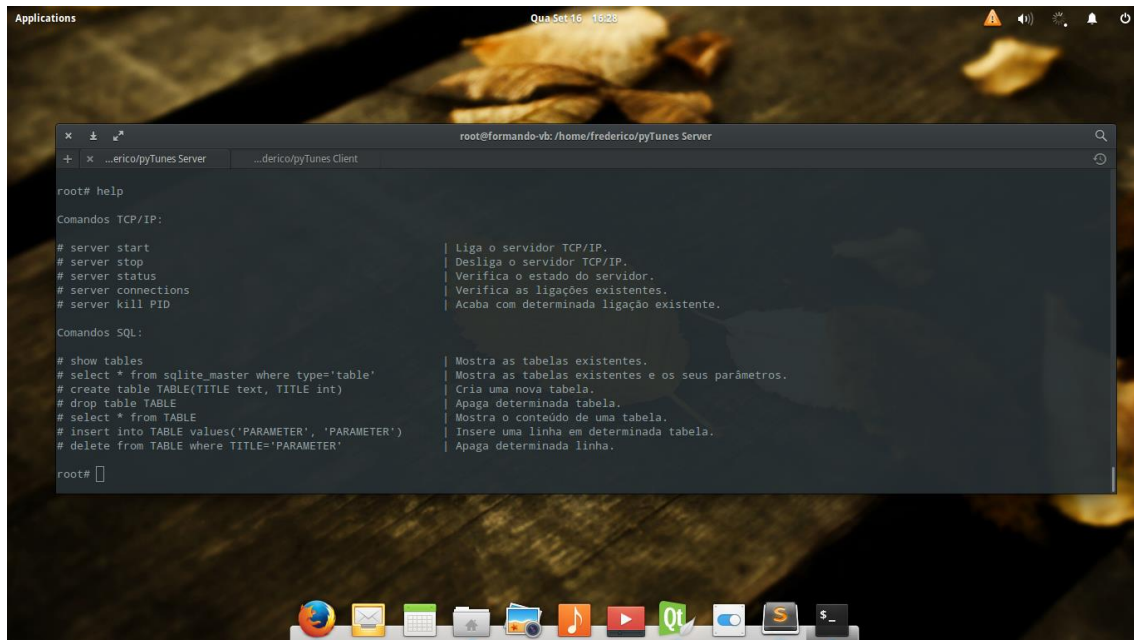


Figura 3 - Início do processo do serviço de TCP/IP

Servidor

Facilidade

A consola dispõe de um comando 'help' que tem como propósito facilitar o uso da mesma e mostrar as diferentes ações possíveis. Este tem um pequeno campo de explicação à frente dos comandos para dar um conceito geral e básico dos mesmos.




```
root@formando-vb: /home/frederico/pytunes Server
root# help
Comandos TCP/IP:
# server start          | Liga o servidor TCP/IP.
# server stop           | Desliga o servidor TCP/IP.
# server status         | Verifica o estado do servidor.
# server connections    | Verifica as ligações existentes.
# server kill PID       | Acaba com determinada ligação existente.

Comandos SQL:
# show tables           | Mostra as tabelas existentes.
# select * from salite_master where type='table' | Mostra as tabelas existentes e os seus parâmetros.
# create table TABLE(TITLE text, TITLE int)   | Cria uma nova tabela.
# drop table TABLE    | Apaga determinada tabela.
# select * from TABLE | Mostra o conteúdo de uma tabela.
# insert into TABLE values('PARAMETER', 'PARAMETER') | Inseire uma linha em determinada tabela.
# delete from TABLE where TITLE='PARAMETER' | Apaga determinada linha.

root#
```

Figura 4 - Comando 'help'

Servidor



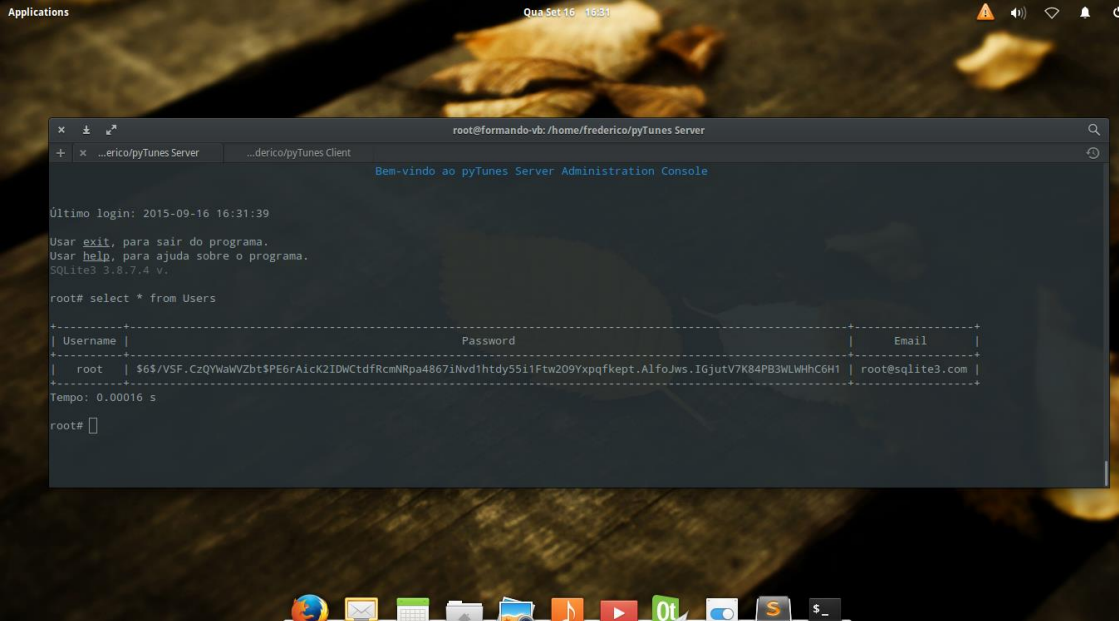
```

root@fermando-vb: /home/frederico/pyTunes Server
# server start          | Liga o servidor TCP/IP.
# server stop           | Desliga o servidor TCP/IP.
# server status         | Verifica o estado do servidor.
# server connections    | Verifica as ligações existentes.
# server kill PID       | Acaba com determinada ligação existente.

Comandos SQL:
# show tables           | Mostra as tabelas existentes.
# select * from sqlite_master where type='table' | Mostra as tabelas existentes e os seus parâmetros.
# create table TABLE (TITLE text, TITLE int) | Cria uma nova tabela.
# drop table TABLE    | Apaga determinada tabela.
# select * from TABLE | Mostra o conteúdo de uma tabela.
# insert into TABLE values('PARAMETER', 'PARAMETER') | Insere uma linha em determinada tabela.
# delete from TABLE where TITLE='PARAMETER' | Apaga determinada linha.

root# server connections
IP - 192.168.0.184 | PID - 8676
root#
  
```

Figura 5 - Ligações existentes com o IP e o número do processo, que neste caso é apenas um cliente que está ligado



```

root@fermando-vb: /home/frederico/pyTunes Server
 Bem-vindo ao pyTunes Server Administration Console

Último login: 2015-09-16 16:31:39

Usar exit, para sair do programa.
Usar help, para ajuda sobre o programa.
SQLite3 3.8.7.4 v.

root# select * from Users
+-----+-----+-----+
| Username | Password | Email |
+-----+-----+-----+
| root | $6$/V$F.CzQWwMVZbt$PE6rAick2IDWctdFRc0nRpa4867iNvd1htdy5Si1Ftw209Yxpfkept.AlfoJws.IGjutV7K84PB3WLWlhC6H1 | root@sqlite3.com |
+-----+-----+-----+

Tempo: 0.00016 s

root#
  
```

Figura 6 - Hashing das palavras-chaves dos utilizadores

Cliente

Compatibilidade

O programa cliente, em relação ao servidor, é feito para ser corrido pelo máximo de clientes possíveis e tem que ser assim o mais independente possível da plataforma onde é executado, por este motivo foi feito para ser compatível com versões do Windows e de Linux, ambos de 64 bits.

Ao contrário do servidor que é executado em consola, este possui uma interface gráfica para melhor conforto e simplicidade do cliente, oferecendo *features*, que são ao mesmo tempo úteis, mas simples. Desenvolvida em PyQt 5.5, que oferece compatibilidade com os mais diversos sistemas operativos.

Para poder se entrar na parte principal do programa e se puder começar a ouvir música, é necessário antes fazer a autenticação do utilizador, para este aceder às suas playlists e à sua conta, autenticação esta que é feito do lado do servidor e apenas se for bem-sucedida é que possibilita o login.

Caso o utilizador em questão ainda não esteja registado, pode ainda se registar para depois fazer o login com a sua nova conta e começar a navegar.

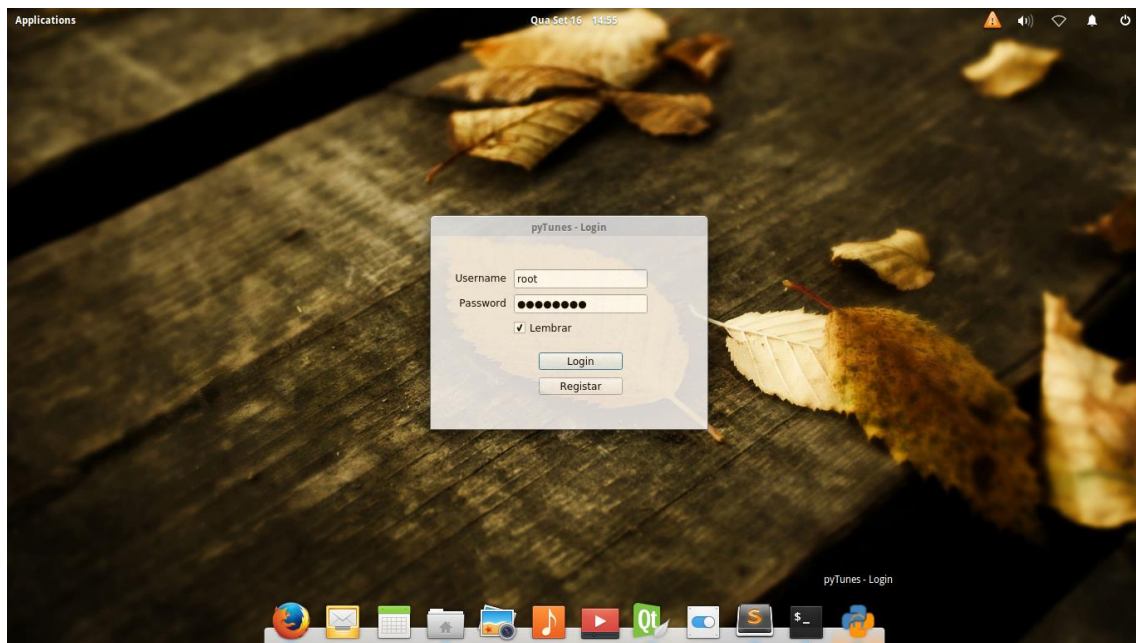


Figura 7 - Login no programa cliente

Cliente

Segurança

A comunicação entre o cliente e o servidor é também encriptada assimetricamente por RSA, trocando chaves públicas no início de cada login.

Os ficheiros como dito anteriormente não são enviados encriptados, isto por questões de tempo e menos sobrecarga para o lado do servidor. Depois de recebidos, os ficheiros de imagens são guardados em uma pasta temporária e não voltando a pedir a mesma imagem uma segunda vez, enquanto as músicas são guardadas no *buffer* por questões de direitos de autor, para não serem pirateadas.

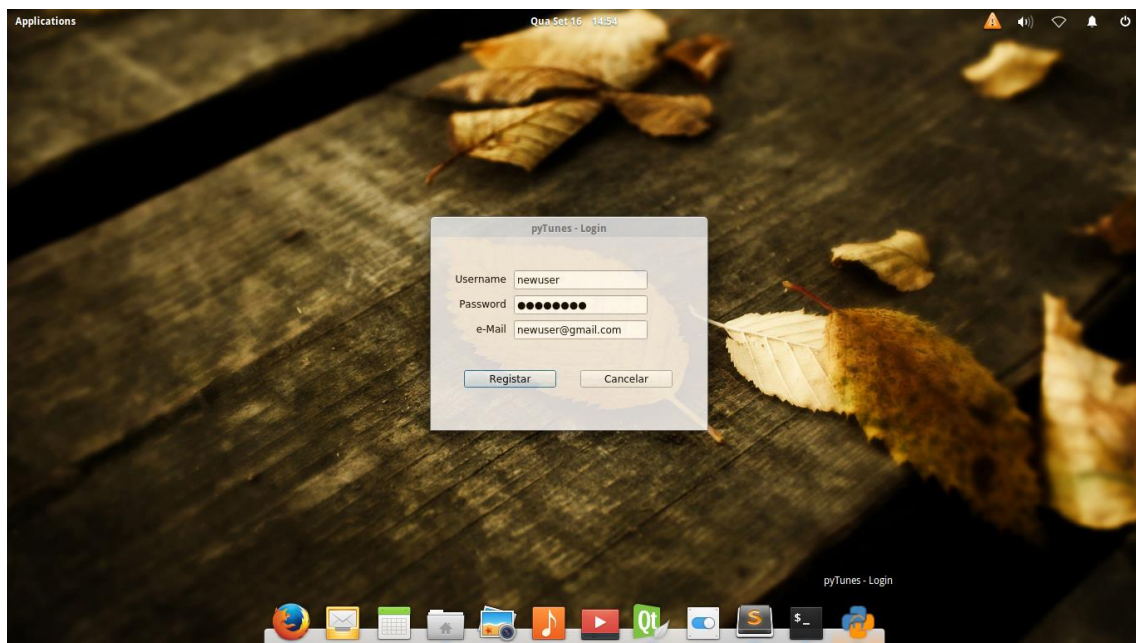


Figura 8 - Registo de um novo utilizador

Cliente

Facilidade

A ideia por trás do programa e de qualquer outro programa, é oferecer ao utilizador uma experiência o mais simples possível e com pouca complexidade, não necessitando de grande esforço para a entender e poder assim aproveitar do programa em qualquer ocasião.

Foi nesta base que o programa foi construído, sendo um programa bastante simples, mas compacto e seguro.

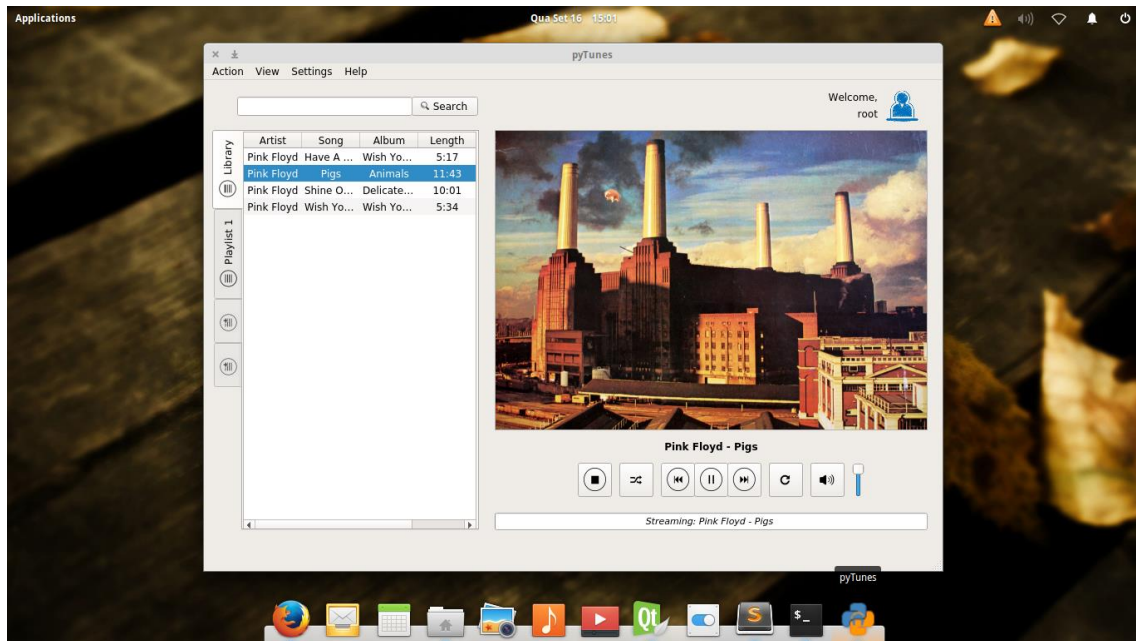


Figura 9 - pyTunes interface principal