



1. Descrição geral

A avaliação da disciplina de aplicações distribuídas está dividida em quatro projetos. O quarto projeto incide sobre **medidas de segurança** para o projeto anterior.

No terceiro projeto foi concretizado um serviço WEB para gerir um sistema simplificado de classificação de álbuns de música de utilizadores. Para esse efeito, foram utilizados no servidor a *framework* de desenvolvimento WEB *Flask* e o motor de base de dados SQL *sqlite*. O programa cliente utiliza o módulo *requests* para implementar a interação cliente/servidor baseada no HTTP.

2. Autenticação e confidencialidade da comunicação

O **protocolo SSL/TLS** deverá ser utilizado **com o *Flask* e com o módulo *requests*** (ver os parâmetros ***cert* e *verify***) para que o cliente e o servidor verifiquem mutuamente a autenticidade do interlocutor e para que a **comunicação seja confidencial (cifrada)**. Para este efeito, o cliente e o servidor deverão ter **certificados de chave pública assinados por uma *Certificate Authority* (CA)**. Ou seja, os certificados não devem ser auto-assinados (ver o Exercício 6 do Guião da PL08). A implementação no *Flask* será feita através da classe ***SSLContext* do módulo *ssl*** [1] da biblioteca padrão do *Python*. Recomenda-se passar um objeto *SSLContext* (ver o Guião da PL08) à variável `ssl_context` no arranque do *Flask* (ver o Guião da PL09).

3. Autorização

O protocolo *OAuth2* [2][3] deverá ser utilizado com o *Flask* e com o módulo *requests_oauthlib* [4][5] para que a aplicação possa pedir autorização para usar os recursos disponibilizados pelo servidor de recursos do Spotify. Para este efeito, o URI do servidor deverá estar registado na API de *OAuth* do Spotify de forma que o mesmo possa ser autenticado e autorizado por esta API (ver o Guião da PL09).

4. Entrega

A entrega do Projeto 4 consiste em colocar todos os ficheiros *.py* e *.db*, o ficheiro *README* e o ficheiro com o esquema da base de dados em SQL numa diretoria cujo nome deve seguir exatamente o padrão **grupoXX** (onde XX é o número do grupo). Deve-se criar três subdiretórias com os nomes **client**, **server** e **certs**. Nestas serão colocados os ficheiros referentes ao cliente (grupoXX/client), ao servidor (grupoXX/server/) e aos certificados (grupoXX/certs/).

Juntamente com os ficheiros *.py* e *.db* deverá ser enviado um ficheiro de texto README.txt (não é .pdf nem .rtf nem .doc nem .docx) onde os alunos descrevem o modo de utilização e funcionamento do projeto (software desenvolvido). Deverão relatar também toda a informação que acharem pertinente sobre a sua implementação do projeto (por exemplo, limitações).

A diretoria será incluída num ficheiro ZIP cujo nome deve seguir exatamente o padrão **grupoXX.zip**. Esse ficheiro será submetido num recurso a disponibilizar para o efeito na página de AD no moodle da FCUL.

Note que a **entrega deve conter apenas os ficheiros .py, .sql, certificados e o ficheiro README.txt, qualquer outro ficheiro vai ser ignorado.**

O prazo de entrega é domingo, dia 02/05/2021, até às 23:59hs.

5. Bibliografia

- [1] <https://docs.python.org/3/library/ssl.html>
- [2] <https://aaronparecki.com/oauth-2-simplified/>
- [3] <https://oauth.net>
- [4] <https://pypi.python.org/pypi/requests-oauthlib>
- [5] <http://requests-oauthlib.readthedocs.io/en/latest/index.html>