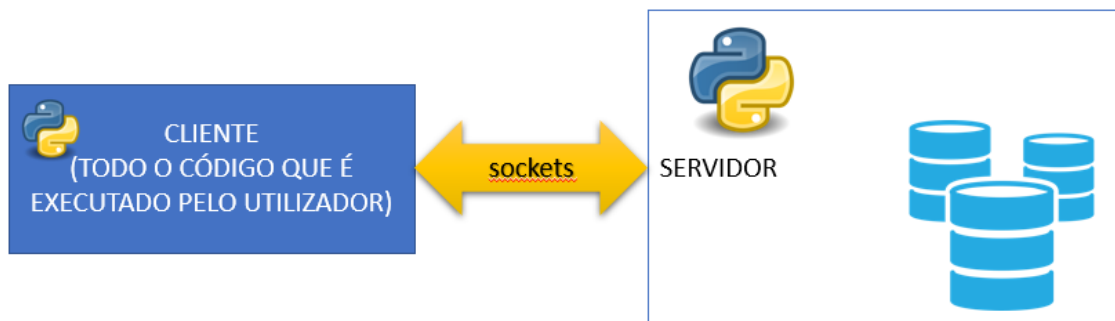


Projeto Programação II

O Instituto Politécnico decidiu desenvolver um novo sistema de gestão de turmas e solicitou à equipa de Gestão de Sistemas de Informação para programar o software que irá ser usado para fazer esta gestão.

Foram entregues os requisitos técnicos que se pretendem ver ser desenvolvidos nesta aplicação, os alunos têm o objetivo de realizarem uma aplicação com uma linguagem de programação dinâmica: Python. A escrita de código adicional na linguagem de programação C/C++ com o eventual uso de bibliotecas adicionais é possível, desde que não ultrapasse 20% do código total do projeto.

Uma porção do programa funcionará nos computadores clientes, e outra parte do código irá funcionar num servidor, que contém também uma base de dados, seguindo então o seguinte esquema:



O utilizador apenas pode interagir com o código que se encontra do lado do cliente, e todo o código executado no servidor deverá de funcionar de forma “passiva”, que reage à informação que chega do código do lado do cliente. Deverá ser possível então que cliente e servidor possam trocar informação.

Outro dos requisitos estipulados foi ter o SQLite como Sistema de Gestão de Base de Dados no servidor.

Requisitos Funcionais

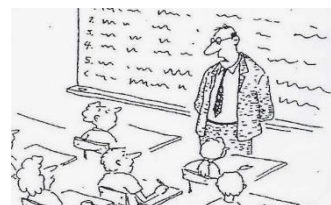
Cliente

Os requisitos funcionais a desenvolver do lado do cliente serão os seguintes:

- Desenvolver um menu que tenha estas opções:

- 1) Criar Disciplina
- 2) Listar Disciplinas
- 3) Eliminar Disciplina
- 4) Criar Aluno
- 5) Inscrever Aluno
- 6) Eliminar Aluno
- 7) Listar Alunos
- 8) Listar Alunos inscritos numa dada disciplina
- 9) Criar Professor
- 10) Adicionar professor a uma disciplina
- 11) Importar alunos de um ficheiro
- 0) Sair

Para cada uma das opções serão dadas as informações a desenvolver.



1) Criar Disciplina

O programa deverá de solicitar ao utilizador que insira um nome de uma disciplina, e essa disciplina deverá ficar guardada em memória (lista).

Deverá ser enviado esta informação para o servidor para ser guardado numa tabela de base de dados.

2) Listar Disciplinas

O programa deverá de Listar no terminal (Shell do programa) todos os nomes de disciplinas que foram inseridas no software.

A listagem deverá aparecer cada disciplina individualmente, precedida de um número, ou seja, se tivermos a seguinte lista: ('Matemática', 'Português', 'Programação') no terminal deverá aparecer o resultado da seguinte maneira:

Disciplinas existentes:

- 0 – Matemática
- 1 – Português
- 2 – Programação

3) Eliminar Disciplina

O programa **deverá executar o mesmo código desenvolvido para o "Listar Disciplinas"** e deverá de ter uma funcionalidade acrescida de solicitar ao utilizador qual é a disciplina que pretende eliminar.

O programador pode escolher em desenvolver o programa para pedir o número da disciplina, ou então pedir o nome da disciplina para eliminar. Ambas as opções são válidas.

Exemplo, se pretender eliminar a disciplina de Matemática, o utilizador teria de inserir ou "0" ou então "Matemática".

No final deverá aparecer uma listagem atualizada com as disciplinas que ainda existem, semelhante à primeira.

4) Criar Aluno

Na opção de criação de aluno, o utilizador é solicitado a inserir os seguintes dados: Nome completo, idade, morada, número de cartão de cidadão, número de aluno.

As informações todas referentes aos alunos deverão de ser guardadas numa Lista geral, que permita guardar todos os alunos até agora inseridos.

Rever o conceito de classes.

Deverá ser enviado para o servidor apenas o Nome e número do aluno.

5) Inscrever Aluno

O início desta operação apresenta as disciplinas existentes de forma igual à opção de “listar disciplinas”, e permite ao utilizador seleccionar qual a disciplina onde pretende inscrever um aluno.

Após o utilizador seleccionar a disciplina, o programa mostra uma listagem dos alunos existentes, e o programa permite escolher qual dos alunos pretende inscrever numa disciplina.

Atenção: Quando guardarmos a informação das opções do utilizador, a “associação” à disciplina será o número de aluno. Isto é, o utilizador selecciona a disciplina de “Programação”, e quando selecciona o aluno que pretende ter inscrito na disciplina, ficará algures registado o número de aluno associado a essa disciplina.

6) Listar Alunos

Esta operação permite listar o nome de todos os alunos e respetivos números de aluno no terminal.

7) Eliminar Aluno

Esta operação lista todos os alunos existentes atualmente na instituição de ensino e permite ao utilizador escolher qual o aluno que pretende eliminar do programa.

Nota: O aluno que for removido do programa deverá também ser removido de todas as disciplinas em que estiver inscrito! Deverá igualmente ser removido a referência deste aluno na base de dados do Servidor.

8) Listar Alunos inscritos numa dada disciplina

Com esta opção, deverá aparecer a listagem de todas as disciplinas existentes, e posteriormente o utilizador selecciona qual a disciplina que pretende, e o programa lista todos os alunos associados a essa disciplina.

9) Criar Professor

Semelhante às funções de criação de Alunos, durante a criação de professor, o programa deverá solicitar ao utilizador a seguinte informação: Nome do Professor, idade, Morada, Categoria Profissional, anos de experiência Profissional, número de professor.

O nome e número de professor deverão de ser enviados para o servidor para que fiquem guardados em base de dados.

10) Adicionar Professor a uma disciplina

À semelhança dos conceitos anteriormente definidos, deverão ser listadas as disciplinas todas que já foram criadas, e o utilizador tem a hipótese de escolher qual a disciplina que irá modificar.

Após selecionar a disciplina, deverá surgir uma listagem com todos os professores existentes, e o utilizador seleciona qual o professor que pretende adicionar à disciplina. Na disciplina fica guardado o nome do professor que lhe está associado.

11) Importar alunos de um ficheiro

Esta opção deverá permitir ao utilizador fazer a importação de alunos de um ficheiro e incorporar todos os nomes presentes nesse ficheiro numa disciplina à escolha do utilizador.

0) Sair

A aplicação termina.

Recomendações:

A “Disciplina”, o “Aluno” e o “Professor” **podem ser classes**, para que mais facilmente se vejam quais os atributos que cada um tem.

- Não esquecer que uma Disciplina é composta por uma **lista** de alunos.
- Cada disciplina tem **apenas 1 professor**.

Em Python é possível termos listas de objetos! Ou seja, se criarmos o objeto “aluno” a partir de uma classe, nós podemos guardar este objeto dentro de uma lista, o que auxilia na questão das Listagens que são pedidas nos problemas.

Modo de Funcionamento

O trabalho, em todas as vertentes, deve ser efetuado em cima de um sistema de controlo de versões (ex: mercurial, svn, git, etc...). O tema do trabalho é idêntico para todos os alunos. A avaliação é individual.

Sistema Operativo

O sistema operativo escolhido para a realização do trabalho pode ser um dos seguintes:

- Windows;
- Linux numa das suas variantes;
- OSX numa das suas variantes.

Sugere-se que o trabalho seja desenvolvido numa máquina virtual ligeira e com um editor de texto de sistema em modo de consola, por exemplo: vi, emacs, nano, etc.

A utilização de ambientes integrados de desenvolvimento, vulgarmente conhecidos por IDE, podem ser usados, mas os alunos deverão de possuir o conhecimento para também desenvolverem código sem o recurso a um IDE.

É recomendado aprender a funcionar com linha de comandos.

Sistema de Controlo de Versões

O trabalho deve ser realizado com controlo de versões. O sistema pode ser escolhido pelo próprio aluno, no entanto na realização do trabalho deverá este ser acompanhado de um relatório dos “commits” do próprio sistema de versões utilizado.

Data e Modo de Entrega

O projeto estará disponível imediatamente no início do semestre e a data limite para a entrega do trabalho será divulgada na página da disciplina.

Não há tolerância com a data e hora de entrega pelo que se aconselha o planeamento atempado da submissão.

A entrega deverá ser através da plataforma moodle.

Os alunos devem entregar o trabalho num ficheiro .zip com a seguinte estrutura.

Nome ficheiro zip: **nomeAluno_nº_ProgII.zip**

O ficheiro **nomeAluno_nº_ProgII.zip** deverá conter:

relatório.pdf (obrigatório) : correspondente ao relatório do trabalho;

src (obrigatório): diretório/pasta onde reside o código fonte da aplicação, ou seja, todo o código desenvolvido deverá de estar dentro desta pasta “src”;

manual.pdf (facultativo): correspondente ao manual de utilizador da aplicação, para caso seja necessário declarar módulos, estruturas ou bibliotecas que tenham usado e que seja necessário dar a indicação de como tais módulos funcionam;

srcdoc.pdf (obrigatório): correspondente à documentação do código.

É **importante** que os alunos que pretendam ser avaliados na disciplina enviem uma mensagem no fórum do moodle a informar dessa pretensão. Será questionado a todos os alunos, através de um post, quais

efetivamente irão entregar o trabalho para avaliação. Este post será inserido no moodle até uma semana antes da data limite de entrega.

Componentes da Avaliação

O trabalho é composto pelas seguintes parcelas que serão avaliadas:

- E* - Aplicação e scripts executáveis;
- C* - Código da aplicação;
- R* - Relatório do trabalho;
- D* - Documentação do código da aplicação;
- A* - Apresentação.

É importante salientar duas questões fundamentais:

- a originalidade e inventividade das soluções;
- a honestidade na realização e na atribuição dos créditos intelectuais.

Aplicação Executável

A funcionalidade geral da aplicação, a obediência, e eventual ultrapassagem dos requisitos será levada em conta. Fatores de robustez e segurança na utilização da aplicação também serão importantes.

Código da Aplicação

No código da aplicação, entre outros aspetos, serão tomados em conta:

- a clareza do código;
- a estrutura do código;
- os conhecimentos de programação específicos para cada linguagem;
- a utilização correta do sistema de controlo de versões;
- a originalidade das soluções propostas;
- a referência adequada das fontes de inspiração do código.

Elaboração de relatório

Para a elaboração do relatório, deverá de ser usado o template disponibilizado na página da disciplina para o efeito.

O relatório desenvolvido deve respeitar as normas presentes no template, tal como se fosse apresentar o trabalho numa revista científica de referência, e deverá de respeitar a seguinte estrutura:

- Autor/ Título / Subtítulo do trabalho
- Índice do documento
- Índice de figuras / imagens (caso as haja)
- Índice de tabelas (caso as haja)
- Introdução – Escrever uma introdução ao projeto em Português, referindo também um pouco do estudo desenvolvido no Estado da Arte (mencionado adiante)
- Estudo do Estado da arte – Deve conter um estudo feito com base em artigos científicos de referência sobre o Python e a sua utilização em aplicações de gestão informática; deverá ser também estudado a interação com bases de dados open source.
- Todos os artigos, sites, livros ou locais que venham a ser usados para desenvolver o conteúdo do vosso trabalho deverá de ser mencionado nas referências.
- Implementação de uma proposta de software – Devem de explicar os passos de desenvolvimento do código do vosso programa. Técnicas, algoritmos, sites, livros ou qualquer outro tipo de fontes que sirvam de base para implementar alguma funcionalidade no trabalho deve ser mencionado nas referências.
- Deverão fazer alusão à lógica de programação que estão a usar.
- Conclusões – Deverão de mencionar as conclusões obtidas do desenvolvimento do projeto, as dificuldades sentidas e ultrapassadas e a perspectivas de possíveis projetos futuros em que o vosso código se possa inserir.

- Referências – Esta secção deverá conter TODAS as fontes de informação que serviram de base para o trabalho que desenvolveram.

É totalmente imperativo ter em conta os elementos:

- a qualidade técnica e científica da escrita;
- a utilização correta do português e/ou do inglês;
- a estrutura do documento;
- a utilização adequada das referências bibliográficas.

O formato final deve ser em PDF.

Documentação para o Utilizador

A documentação para o utilizador deverá ser de leitura fácil e tecnicamente correta. O formato final deve ser em PDF.

Documentação do Código

Encoraja-se a utilização de ferramentas de extração automática da documentação do código diretamente a partir do código fonte.

Exemplos de sistemas deste género estão disponíveis em

- https://www.ruby-toolbox.com/categories/documentation_tools;
- <http://www.naturaldocs.org/>;
- http://en.wikipedia.org/wiki/Comparison_of_documentation_generators;
- <http://sphinx-doc.org/>;
- <http://docs.python.org/2/library/pydoc.html>;
- <http://wiki.python.org/moin/DocumentationTools>.

A conversão de HTML para PDF é facilitada por ferramentas como:

- <http://html2pdf.fr/en/default>;
- <http://www.cyberciti.biz/open-source/html-to-pdf-freeware-linux-osx-windows-software/>;
- <http://code.google.com/p/wkhtmltopdf/>.

O formato final deve ser em PDF.

Apresentação

O aluno deverá realizar uma apresentação final do trabalho, que será tido em conta para a avaliação.

A apresentação deverá ser no máximo de 15 min, havendo posteriormente tempo para discussão com perguntas e respostas.

Classificação Final

A classificação final, F , será calculada por:

$$F = 25\% \times E + 25\% \times C + 30\% \times R + 10\% \times D + 10\% \times A.$$

E - aplicação executável;

C - código da aplicação;

R - relatório do trabalho;

U - documentação para o utilizador;

D - documentação do código da aplicação;

A - apresentação.

A classificação final é arredondada para unidades entre 0 e 20 valores.

Bom Trabalho!