Ciências ULisboa | Informática

Aplicações Distribuídas

2015/16

Projeto 3

1. Descrição geral

A avaliação da cadeira de aplicações distribuídas está divida em quatro projetos. O projeto 3 não tem ligação com os anteriores.

O objetivo geral do projeto será concretizar um serviço WEB para gerir um sistema simplificado de inscrição de alunos em turmas. A implementação vai utilizar o estilo arquitetural REST [1] e uma base de dados relacional acessível pela linguagem SQL. Para este efeito no servidor será utilizada a *framework* de desenvolvimento WEB *Flask* [2] e o motor de base de dados SQL *sqlite* [3]. O programa cliente utilizará o módulo *requests* [4] para implementar a interação cliente/servidor baseada no HTTP.

2. Esquema da base de dados

A definição da base de dados assenta nos conceitos envolvidos: aluno, disciplina, turma e inscrição. Cada aluno inscreve-se numa ou mais turmas e cada turma está associada a uma disciplina (podendo várias turmas estarem associadas à mesma disciplina). Cada conceito corresponde a uma tabela de acordo com a figura seguinte, onde também se ilustram as várias relações.

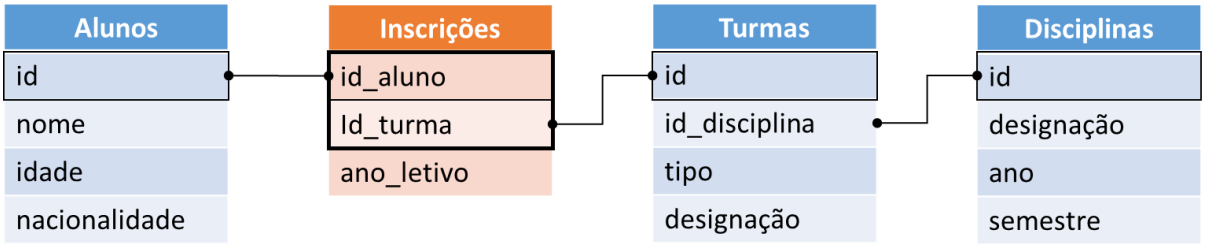

Figura 1 - Esquema da base de dados.

Para criar a base de dados poderá ser utilizado o seguinte código em SQL (continua na próxima página):

```
PRAGMA foreign_keys = ON;

CREATE TABLE alunos (
      id                INTEGER PRIMARY KEY,
      nome              TEXT,
      nacionalidade     TEXT,
      idade             INTEGER
);

CREATE TABLE disciplina (
      id          INTEGER PRIMARY KEY,
      designacao  TEXT,
      ano         INTEGER,
      semestre    INTEGER
);
```

```
CREATE TABLE turma (
      id                INTEGER PRIMARY KEY,
      id_disciplina     INTEGER,
      tipo              TEXT,
      designacao        TEXT,
      FOREIGN KEY(id_disciplina) REFERENCES disciplina(id)
);

CREATE TABLE inscricoes (
      id_aluno    INTEGER,
      id_turma    INTEGER,
      ano_letivo  TEXT,
      PRIMARY KEY (id_aluno, id_turma),
      FOREIGN KEY(id_aluno) REFERENCES alunos(id),
      FOREIGN KEY(id_turma) REFERENCES turma(id)
);
```

A primeira linha desta listagem serve para que o *sqlite* possa suportar chaves estrangeiras. Por omissão, na instalação de alguns sistemas operativos, essa opção está desabilitada.

A aplicação pretendida deverá contemplar uma rotina de inicialização que verifica se a base de dados já existe. Caso não exista ela deverá ser criada e inicializada com o código apresentado.

3. O programa cliente

O programa cliente aceita interactivamente três operações e seus parâmetros (introduzidos via teclado numa consola), e comunica com o servidor para que este processe as operações e armazene a informação numa base de dados. A tabela 1 mostra detalha as operações que o cliente deverá suportar.

Tabela 1 - Lista de operações que o cliente aceita e parâmetros correspondentes.

Operação	Parâmetros	Observações
ADD	ALUNO <nacionalidade> <idade> <nome> ou DISCIPLINA <ano> <semestre> <designação> ou TURMA <id_disciplina> <tipo> <designação> ou <id_aluno> <id_turma>	ano = 1 | 2 | 3 | 4 | 5 semestre = 1 | 2 tipo = T|TP|PL|O|OT A última variante serve para inscrever o aluno <id_aluno> na turma <id_turma>
REMOVE ou SHOW	ALUNO <id_aluno> ou DISCIPLINA <id_disciplina> ou TURMA <id_turma> ou <id_aluno> <id_turma> ou ALL <ALUNOS | DISCIPLINAS | TURMAS> ou ALL ALUNOS <id_turma> ou ALL ALUNOS <id_disciplina> ou ALL TURMAS <id_disciplina>	As três últimas variantes permitem remover ou obter: • todos os alunos inscritos numa turma; • todos os alunos inscritos numa disciplina; • todas as turmas de uma disciplina;

Convém relembrar que os id's (chaves primárias) dos elementos nas tabelas de alunos, disciplinas e turmas são números inteiros.

O cliente comunicará com o servidor através de mensagens em HTTP e usará uma representação utilizando JSON [5]. Para este efeito os alunos utilizarão o módulo *requests* [4]. As mensagens terão de respeitar a API REST definida pelo serviço WEB de inscrição de alunos.

4. O serviço WEB

O serviço WEB será implementado com recurso à *framework* Flask [2] e a API REST será disponibilizada através de três URLs de base:

1. /alunos
   Para operações relativas a alunos.
2. /disciplinas
   Para operações relativas a disciplinas.
3. /turmas
   Para operações relativas a turmas.

É muito importante que os alunos planeiem a API REST antes de iniciarem a implementação. Sugere-se que façam uma tabela onde definam a correspondência entre as operações suportadas, o método do HTTP, as URLs dos recursos, os parâmetros das operações, e as possíveis respostas HTTP com que o serviço responderá ao cliente.

Quando o serviço recebe uma mensagem de um cliente, a operação deverá ser implementada sobre a base de dados e a resposta será preparada segundo os padrões REST utilizando JSON para transmitir a representação dos recursos ou o conteúdo de outras mensagens.

Para o acesso à base de dados, os alunos devem procurar na documentação sobre Flask a forma de fazer com que a ligação à base de dados exista de forma automática sempre que a aplicação recebe um pedido.

5. Tratamento de erros

Sempre que a operação não possa ser executada ou que algum erro inesperado ocorra, o serviço deverá responder ao cliente com uma resposta HTTP incluindo uma descrição detalhada do problema segundo o formato apresentado na aula TP08 sobre JSON. O cliente apresentará a informação da descrição detalhada na consola.

6. Entrega

A entrega do projeto 3 consiste em colocar todos os ficheiros *.py* do projeto numa diretoria cujo nome deve seguir exatamente o padrão **grupoXX** (por exemplo grupo01 ou grupo23). Juntamente com os ficheiros *.py* deverá ser enviado um ficheiro de texto README.txt (não é .pdf nem .rtf nem .doc nem .docx) onde os alunos podem relatar 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 e o ficheiro README.txt, qualquer outro ficheiro vai ser ignorado**.

O prazo de entrega é sábado, dia 30/04/2016, até às 22:00hs.

7. Referências

[1] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[2] http://flask.pocoo.org/

[3] https://www.sqlite.org/

[4] http://docs.python-requests.org/en/master/

[5] http://www.json.org/