Trabalho - Computação 2

$EPT/EP1 - 2016.2$

Prof. Brunno Goldstein 22/11/2016

1 Introdução

Neste trabalho o aluno deverá implementar uma *To-Do list* (lista de tarefas) utilizando os conceitos apresentados em sala. O trabalho deverá ser subdivido em duas partes. Na primeira, o aluno implementará o *back-end* da aplicação, ou seja, todas as classes necessárias para o funcionamento da aplicação. Já na segunda parte, o aluno irá implementar o *front-end* utilizando os conceitos de interface gráfica (Tkinter).

Observações:

- Todo código deve ser documentado (classes, atributos e métodos);
- A não documentação do código acarretará em nota zero, mesmo que a implementação esteja correta;
- Exceções devem ser previstas e tratadas corretamente;
- O trabalho a ser entregue deverá conter o *front-end* e o *back-end*.

2 Back-end

O *back-end* da aplicação é composto por três classes: Usuario, Projeto e Tarefa. Elas irão modelar um usuário real que possui um conjunto de projetos, onde cada projeto possui um conjunto de tarefas. Tal relação entre as classes, bem como seus atributos, são apresentados na Figura 1*. O aluno deverá implementar métodos necessários para manipular os atributos definidos.

Com as classes criadas, o aluno deverá implementar um módulo Python para instanciar os objetos e persistí-los em um "banco de dados"(arquivo) utilizando o módulo ***pickle***. Por fim, o objeto persistido deverá ser carregado para verificação de corretude do programa.

*Explicações sobre o diagrama de classes podem ser encontradas na Seção 4.

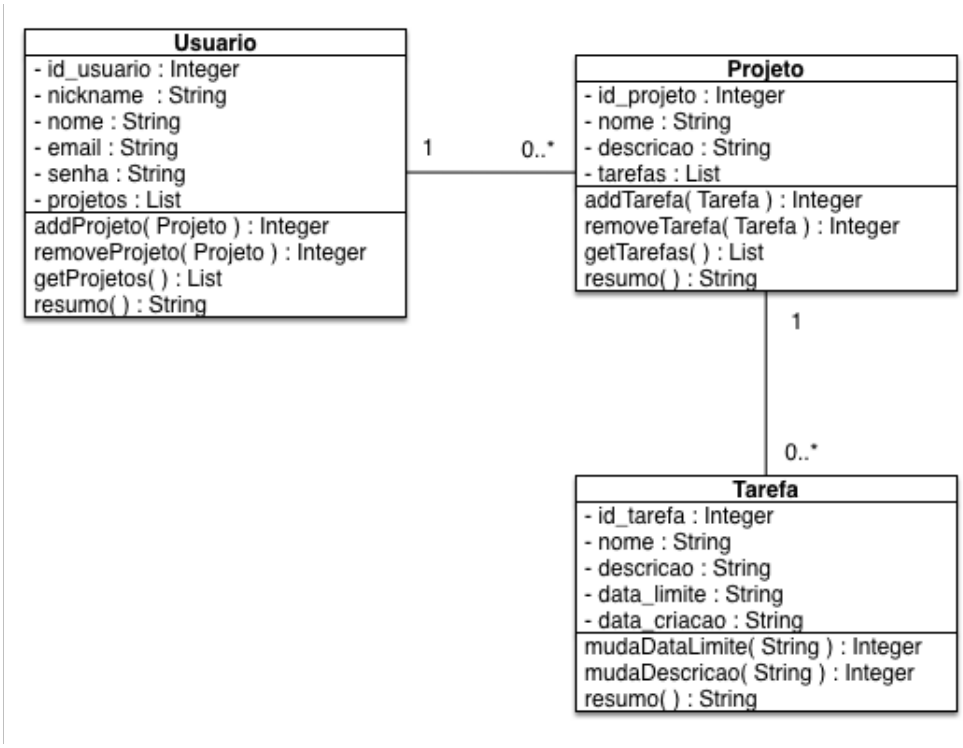

Figura 1: Diagrama de classes da aplicação To-do list.

3 Front-end

O *fron-end* da aplicação será implementado utilizando o módulo Tkinter ensinado em sala. A interface principal deverá conter os quatro componentes descritos abaixo e exemplificados na Figura 2.

- **TreeView**: Lista que irá exibir os projetos e tarefas criados;
- **Botão "Criar Projeto"**: Botão que irá exibir o pop-up para criação de novos projetos;
- **Botão "Criar Tarefa"**: Botão que irá exibir o pop-up para criação de novas tarefas. Tarefas só deverão ser criadas se houver projetos criados e associados a ela;
- **Label "Usuário"**: Label que irá informar qual é o nome do usuário corrente.

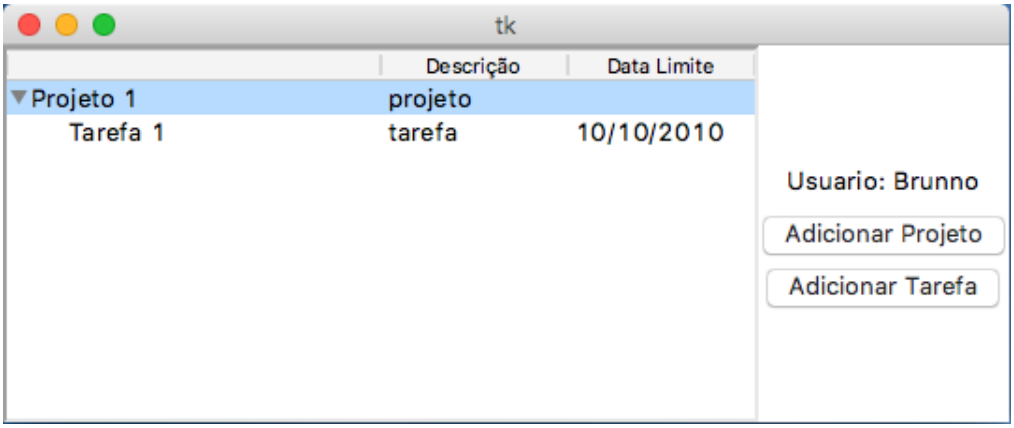

Figura 2: Exemplo de interface principal da aplicação.

As Figuras 3, 4 e 5 exibem exemplos de pop-ups para criação de novos usuários, projetos e tarefas respectivamente. É importante ressaltar que o pop-up para criação de tarefas só será exibido caso

exista algum projeto criado previamente. Caso contrário, um pop-up com mensagem de erro deverá ser exibido informando que o usuário precisa criar primeiro um projeto.

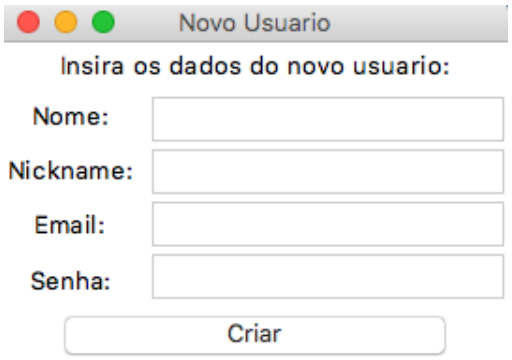

Figura 3: Exemplo de pop-up para criação de novo usuário para a aplicação.

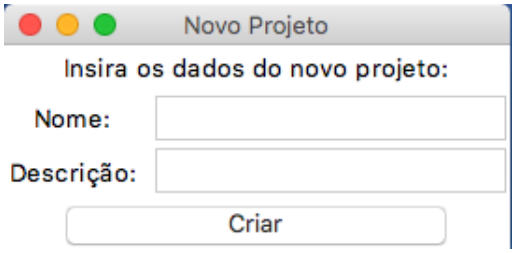

Figura 4: Exemplo de pop-up para criação de um novo projeto para a aplicação.

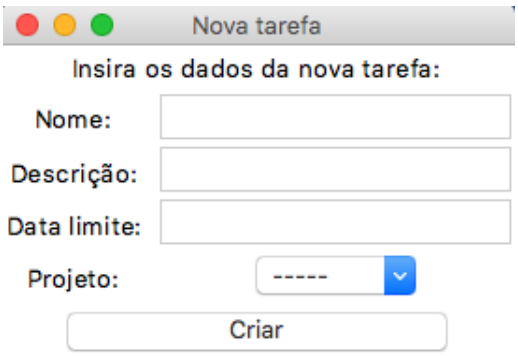

Figura 5: Exemplo de pop-up para criação de uma nova tarefa para a aplicação. Tal pop-up só será exibido caso exista um projeto.

O fluxo de execução da interface está descrito na Figura 6. Ao iniciar o programa, a interface deverá verificar se o arquivo de banco de dados (arq.bd no caso) já existe. Em caso de afirmativo, a interface irá prosseguir com a exibição da tela principal. Caso o arquivo de banco não exista, a interface deverá omitir a tela principal e exibir o pop-up da Figura 3 para a criação de um novo usuário.
Durante a execução da interface principal, o usuário poderá clicar nos dois botões disponíveis ("Criar Tarefa"e "Criar Projeto"). Ao clicar no botão "Criar Projeto", a interface deverá exibir o pop-up da Figura 4, onde o usuário irá digitar as informações necessárias para criação do projeto. Ao clicar no botão "Criar Tarefa", a interface deverá verificar se existe algum objeto projeto na lista de projeto do usuário corrente. Caso exista, a interface irá prosseguir exibindo a tela de criação de nova tarefa (Figura 5). Caso contrário, um pop-up de erro deverá ser exibido.

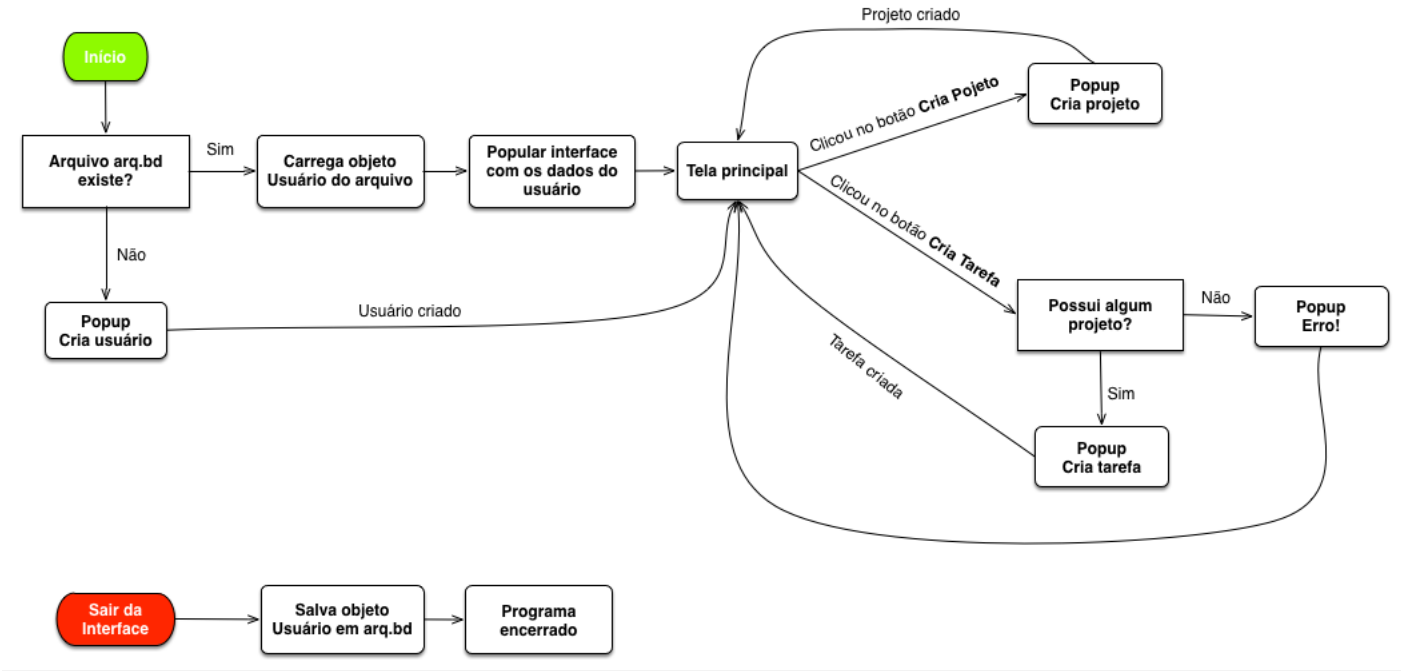

Figura 6: Fluxo de execução do front-end em conjunto com o back-end.

4 Diagrama de Classes

A Figura 7 descreve como o diagrama de classes deve ser interpretado. Cada caixa representa uma classe, onde uma lista de atributos e métodos são apresentados. Cada atributo é representado pelo seu nome e tipo, separados pelo símblo **:** . Os métodos são descritos pelo seu nome, o tipo dos parâmetros de entrada e o tipo da variável de retorno daquele método.
A relação entre as classes é descrita por uma linha interligando as caixas. Tal linha possui números em suas extremidades. Esses números representam quantos objetos daquela classe se relacionam com a outra.

Exemplo: Se na Figura 7 considerarmos que a classe no quadrante superior é a Classe_1 e a outra a Classe_2, podemos dizer que Classe_1 possui zero ou mais (**0..***) objetos da Classe_2. Ou seja, um dos atributos da Classe_1 armazena uma lista (**0..***) de objetos da Classe_2.

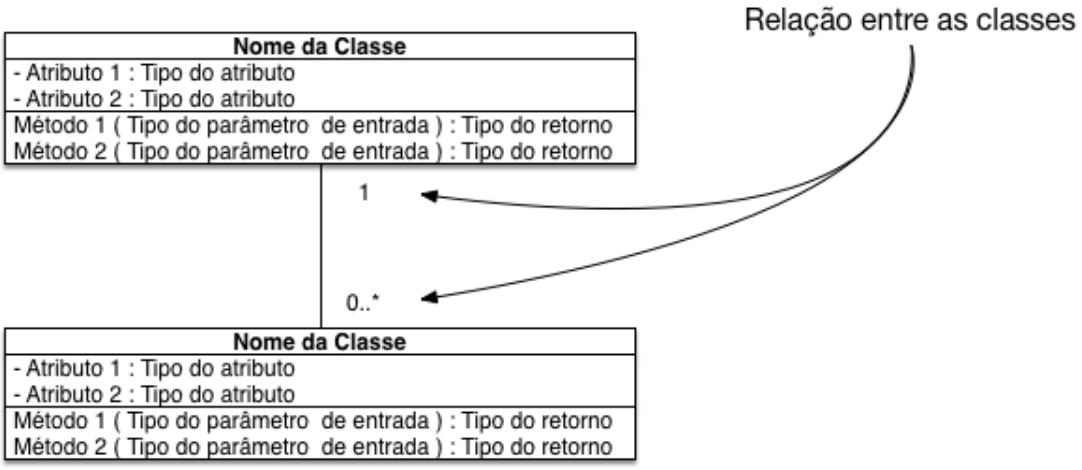

Figura 7: Descrição do diagrama de classes.

5 Informações adicionais

- Todo o código deverá ser documentado utilizando os conceitos ensinados em sala. Classes com descrição completa e métodos com breve descrição.
- O código deverá ser testado pelo aluno antes do envio. Trabalhos que não executarem não serão avaliados e receberam nota zero.
- O aluno deverá tratar todas as exceções que julgar necessário. Porém, três exceções devem ser tratadas obrigatoriamente. São elas:

  1. Erro ao criar tarefa sem projeto;
  2. Erro ao não selecionar um projeto para tarefa;
  3. Erro ao criar projeto ou tarefa com mesmo nome;***

  *** Como não estou pedindo a criação de um identificador único para cada objeto tarefa e projeto criado, vamos considerar que todo projeto e tarefa vão ter nomes distintos. Ou seja, os objetos projeto e tarefa serão identificados a partir de seus nomes durante as fases de recuperação de dados.
- O aluno deverá utilizar o conceito de persistência de objetos utilizando o módulo pickle. Tal persistência deverá ser feita toda vez que o usuário fechar a interface (balão vermelho na Figura 6).
- O aluno deverá implementar o carregamento do arquivo persistido pelo módulo pickle, bem como o preenchimento de todo dado carregado na interface.
- O aluno poderá utilizar os códigos implementados em sala e disponibilizados no site para o criação do seu trabalho.
- A data limite de entrega do trabalho está disponível no site da disciplina.