

Problema 3: Sistema Gerenciador de Tarefas

Problema

Atualmente, sistemas gerenciadores de tarefas vêm ganhando popularidade, tanto no mundo corporativo quanto entre usuários individuais. Empresas que visam organizar o trabalho de seus funcionários utilizam destas ferramentas para atribuir e monitorar as tarefas de seus empregados e até mesmo de suas equipes. Da mesma forma, estes sistemas também vêm recebendo adesão por muitos usuários individuais que visam obter maior produtividade e organização em seu cotidiano.

De modo geral, um sistema gerenciador de tarefas é uma ferramenta que serve para listar as tarefas que devem ser executadas por seus usuários. Estas tarefas, por sua vez, possuem prioridades distintas, o que demanda ordens de execução mais rápidas e urgentes para aquelas tarefas com maior prioridade ou mais lentas para as tarefas secundárias. Alguns exemplos de sistemas gerenciadores de tarefas atuais são o Trello, o Quire e o Asana.

Ao longo dos anos, a Coordenação do Colegiado do Curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS) tem notado a grande dificuldade de muitos estudantes calouros e veteranos do curso em organizar e administrar suas atividades e estudos. Isto tem trazido inúmeras consequências negativas aos discentes, como noites de sono perdidas, não execução e atraso de atividades, mau preparo para as avaliações, sobrecarga de estudo na véspera de avaliações. Visando atenuar este problema e oferecer uma possível ajuda aos estudantes, o coordenador do curso sugeriu o uso de uma ferramenta gerenciadora de tarefas simples e de livre acesso a todos os estudantes. Foi encaminhada esta sugestão ao Departamento de Ciências Exatas (DEXA), que resolveu solicitar aos alunos do módulo integrador de Algoritmos o desenvolvimento de tal ferramenta.

Como sugerido pelo coordenador do curso e, sendo esta a primeira versão do sistema, os estudantes devem desenvolver uma ferramenta simples e com funcionalidades básicas. O sistema terá apenas duas classes: Usuário e Tarefa. Em outras palavras, a ferramenta consistirá de um conjunto de usuários, cada um com uma lista contendo suas respectivas tarefas.

Seguem as especificações do sistema:

Ao iniciar o sistema o usuário deverá visualizar um *menu* com as seguintes opções:

- A. *Cadastrar novo usuário.* Nesta opção, devem ser fornecidos o nome/nickname do usuário e uma senha (sem restrições de senha). Ao finalizar o cadastro, o sistema deve verificar a singularidade do nome/nickname do usuário buscando este nome em um arquivo em formato texto contendo os usuários já cadastrados na ferramenta. Não deve ser

salva a senha do usuário, mas um *hash* da senha para evitar problemas de segurança. Caso não exista nenhum usuário com este nome/nickname, o novo usuário deve ser adicionado ao arquivo em formato texto de usuários cadastrados no sistema. Caso contrário, deve ser solicitado um novo nome ao usuário até que o cadastro seja completado.

- B. *Logar no sistema.* Esta opção deve solicitar o nome/nickname do usuário e sua senha. O sistema deve buscar pelo usuário no arquivo de usuários do sistema em formato texto, checando o nome e sua senha correspondente. Devem ser identificados os erros de usuário inexistente ou senha incorreta. Caso os dados fornecidos sejam válidos, um novo *menu* deve ser apresentado contendo as seguintes opções:

- a. *Cadastrar nova Tarefa.* Ao selecionar esta opção, o usuário deve fornecer os seguintes dados da tarefa que deseja cadastrar: Título, Descrição e Prioridade. A prioridade pode ser Alta, Média ou Baixa. Ao finalizar o cadastro, o sistema deve atribuir a esta tarefa uma ID numérica, indicando a ordem de criação da tarefa. Após atribuir a ID, esta tarefa deve ser inserida na coleção de tarefas do usuário seguindo as regras de inserção que serão apresentadas mais adiante e, em seguida, deve salvá-la em um arquivo binário de tarefas exclusivo para guardar todas as tarefas deste usuário.
- b. *Visualizar Tarefas.* Nesta opção, o sistema deve ser capaz de buscar a coleção de tarefas do respectivo usuário em um arquivo binário contendo as tarefas dele. As tarefas do usuário devem ser exibidas em ordem de maior prioridade e, em seguida, por ordem de ID. Caso não existam tarefas cadastradas, o sistema deve informar a não existência de tarefas. Devem ser exibidos a ID da tarefa, o título da tarefa, sua descrição e sua prioridade.
- c. *Alterar Tarefa.* Nesta opção, o usuário deve fornecer a ID da tarefa que deseja alterar e o sistema deve verificar a existência desta ID, retornando uma mensagem de erro caso esta não exista. Caso a ID fornecida exista, o sistema deve perguntar ao cliente qual informação deseja alterar: Título, Descrição ou Prioridade. O usuário deve ser capaz de modificar a informação desejada e, após a alteração, o arquivo de tarefas do usuário deve ser atualizado.
- d. *Excluir Tarefa.* O usuário deve ser capaz de excluir uma tarefa informando a ID da tarefa que deseja remover e o sistema deve buscar por esta tarefa e removê-la da

coleção de tarefas do usuário, atualizando o arquivo binário de tarefas do usuário. Deve ser informada uma mensagem de erro caso a ID não seja encontrada.

- e. *Sair*. O usuário pode escolher sair de sua conta e retornar ao *menu* inicial.

- C. *Sair do sistema*. Esta opção deve permitir que o sistema seja encerrado.

Regras de inserção de tarefas na coleção de tarefas do usuário: A nova tarefa criada deve ser inserida na coleção de maneira ordenada respeitando primeiramente a prioridade designada pelo usuário e, em seguida, a ID atribuída pelo sistema. Em outras palavras, a coleção de tarefas deve apresentar, primeiramente, tarefas de prioridade Alta em ordem crescente de IDs. Em seguida, as tarefas de prioridade Média, também em ordem crescente de IDs. Finalmente, as tarefas de prioridade Baixa também em ordem crescente de seus IDs.

As especificações do sistema ainda solicitam a implementação de um código bem modularizado, pois o sistema passará futuramente por atualizações e pela criação de novas versões por outros desenvolvedores.

Produto e Relatório

Você deverá desenvolver o código fonte do sistema **adequadamente modularizado** em Python e um relatório final, no formato de artigo da SBC, conforme modelo e instruções disponibilizados no site do MI. O código deve ser entregue até ao meio dia do dia **19/03/2020** e o relatório final até ao meio dia do dia **23/03/2020** (a entrega impressa do relatório final pode ser solicitada pelo tutor). O relatório final só será aceito mediante a entrega do código fonte.

O desempenho nas sessões tutoriais equivale a 30% da nota no Problema. O relatório equivale a 30% da nota e o código fonte (produto) equivale a 40%. Haverá penalidade de um ponto por descumprimento do prazo de entrega e um ponto por dia de atraso na entrega, até o máximo de cinco dias. Após este prazo, o trabalho não será mais aceito.

Tanto o código fonte quanto o relatório devem ser desenvolvidos individualmente. Deve constar no código fonte declaração de ausência de plágio.

Recursos para Aprendizagem

WAZLAWICK, R. S. Introdução a Algoritmos e Programação com Python. Elsevier, 2018.

BORGES, L. E. Python para Desenvolvedores. Novatec, 2014.

SUMMERFIELD, M. Programação em Python 3. Elsevier / Altabooks, 2015.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para Desenvolvimento de Programação. São Paulo: Érica, 1996.

FORBELLONE, A. V. L., EBERSPACHER, H. F. Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados. 2. ed. Makron Books, 2000.

DIERBACH, C. Introduction do Computer Science Using Python: A Computational Problem-Solving Focus. Wiley, 2012.

BEAZLEY, D.; JONES, B. K. Python Cookbook. O'Reilly, 2013.

BARRY, P. Use a Cabeça! Python. Elsevier / Alta Books, 2013.

Cronograma

Data	Sessão Tutorial
06/02/2020	Apresentação do Problema 3
13/02/2020	Sessão Tutorial - Problema 3
20/02/2020	Sessão Tutorial - Problema 3
27/02/2020	Sessão Tutorial - Problema 3
05/03/2020	Sessão Tutorial - Problema 3
12/03/2020	Sessão Tutorial - Problema 3
19/03/2020	Entrega do Código-Fonte.
23/03/2020	Entrega do Relatório.