



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL

TÉCNICAS DE PROGRAMAÇÃO – 2018.2 – PROF. WENDLEY

TRABALHO PRÁTICO: 02

Individual - Utilize Python

Entrega: até quarta-feira, 21/11/2018, 23h55, via GITHUB (com link no SIGAA)

Sobre os atrasos: com o final do semestre, serão tolerados atrasos de no máximo 24h, com a seguinte regra: até 12h de atraso, redução de 1,0 ponto, de 12 a 24h de atraso, redução de 3,0 pontos. Evitem os atrasos.

A solução para o trabalho prático (individual) descrito abaixo deve ser entregue via link no SIGAA, indicando o endereço para os arquivos que devem ser submetidos no repositório GitHub (<https://github.com/>) até o prazo descrito acima. A correção se dará fazendo um CLONE da sua pasta no repositório e executando o passo-a-passo descrito no seu “readme.md”. Reforçando: o código-fonte não deve ficar no SIGAA, e sim no GitHub.

Quem ainda não possui conta no GitHub, deve criar (é gratuito). Fiquem atentos que o GitHub registra a hora/data de envio e modificação de cada um dos arquivos, e vocês só poderão modificar os arquivos até a data de entrega (e depois do lançamento das notas finais no SIGAA). No diretório do seu trabalho, dentro do GitHub, deve ter um arquivo “readme.md” com a explicação para a execução do seu código até a tela inicial, similar a um pequeno manual de uso. Se os avaliadores seguirem a sua descrição e o programa não executar a tela inicial, é grande a probabilidade do seu trabalho ser enquadrado com a nota de “não funcionou/não executou” (ou seja, zero), por isso, façam a descrição de forma sucinta e objetiva, mas completa. No “Readme.md” também deve constar o nome da dupla que fez o trabalho.

O objetivo do trabalho, além da fixação de conhecimento na linguagem de programação Python, é estimular as habilidades de pesquisa dos alunos na temática de programação para solucionar problemas, dessa forma, dúvidas pontuais podem ser consultadas com os monitores, mas as principais pesquisas devem ser realizadas pelos próprios alunos.

Neste último trabalho da disciplina, o aluno fará a implementação de um sistema simples de locação de veículos, usando Python.

ESPECIFICAÇÕES (MÍNIMAS) SOBRE O SISTEMA BÁSICO DE LOCAÇÃO (ALUGUEL) DE VEÍCULOS:

1. Sistema sem interface gráfica, somente utilizando o terminal;
2. A “tela” principal deve mostrar uma listagem com, no mínimo, os seguintes módulos operacionais, que poderão ser escolhidos pelo usuário:
 - a. Consultar veículos,
 - b. Adicionar veículos,
 - c. Alugar/reservar veículos,
 - d. Devolver/liberar veículos,

- e. Excluir veículos,
- f. Avançar data atual; e
- g. Sair;

e também deve mostrar as seguintes informações:

- a. “Data atual: dd/mm/aaaa”;
- b. “Quantidade de veículos cadastrados: xx”;
- c. “Quantidade de veículos alugados: yy”; e
- d. “Quantidade de atrasos: hh”.

Inicie o sistema assumindo que a data do seu sistema é igual à data do sistema operacional do computador, e que não há veículos cadastrados. Obs.: para facilitar, pode considerar o sistema com informações de apenas dia, mês e ano, sem considerar os horários;

3. No módulo “**Consultar veículos**”, devem ser exibidos todos os veículos já cadastrados, com informação imediata do código do veículo, modelo e o *status* (se alugado, reservado, disponível ou em atraso), e com opção de consultar mais detalhes (marca, ano e valor da diária).
4. No módulo “**Adicionar veículos**”, deve ser permitido ao usuário adicionar a marca, modelo, ano e valor da diária do veículo. Imediatamente após adicionar, o veículo já fica disponível para aluguel e o código para o primeiro veículo é 001, sendo sequencial à medida que novos veículos são cadastrados;
5. No módulo “**Alugar/Reservar veículos**”, deve ser permitido alugar para uma determinada data um veículo que esteja disponível, e serão solicitados o nome do locatário (quem está alugando), e o prazo de locação (quantos dias). Use o código do veículo para alugar/reservar. Aluguel e reservas somente poderão ser realizadas para no máximo 30 dias adiante (considerando a “data atual”). Depois de alugado, o veículo escolhido será mostrado na “Consulta” como alugado ou reservado. Se a “data atual” do sistema coincidir com uma data do veículo reservado, então o sistema deve considerá-lo como alugado. Atenção: se um mesmo veículo estiver reservado, por exemplo, para os dias 04, 05 e 06, e um cliente deseja alugar no dia 03 (do mesmo mês/ano) por 2 dias, esse módulo deve indicar que há sobreposição de datas e deve ser escolhido outro carro ou outra data, retornando ao início do módulo;
6. No módulo “**Devolver/liberar veículos**”, o sistema deve considerar a “data atual” e listar os veículos que estão alugados ou reservados e permitir liberar a reserva ou efetivar a devolução para algum deles. Use o código do veículo para devolver/liberar a reserva. O nome do cliente que alugou ou reservou deve ser exibido. Caso seja devolução, o módulo deve calcular o valor a ser pago, considerando o tempo que o cliente ficou com o carro. Se houver atrasos, o cliente paga uma diária majorada em 100% por cada dia que ficou em atraso.
7. No módulo “**Excluir veículos**”, o sistema deve permitir remover da base de dados um determinado veículo que não tenha reservas;
8. No módulo “**Avançar data atual**”, o usuário poderá avançar em incrementos de 1 dia a data atual a ser considerada no seu sistema. Não há opção “retroceder data”. Esse módulo é para simular o avanço do tempo e verificar o funcionamento do sistema.
9. Sempre após a execução de qualquer um dos módulos, o sistema deve voltar para a “tela” principal, listando todas as operações possíveis e atualizando todas as informações;
10. Não é obrigatório o uso de banco de dados, mas pode utilizar arquivos TXT, CSV ou DAT para salvar os dados. Programas que utilizem apenas a memória (RAM), excepcionalmente, também serão aceitos.