

Checkpoint 3 – 1TDSPG 2024

Computational Thinking with Python - 14/05/2024

Instruções:

- Crie arquivos .py, modularizando e importando as funções
- A prova será desconsiderada nos casos de:
 - Repetições desnecessárias (mesma função escrita mais de uma vez)
 - Código 100% em 1 arquivo
 - Código que não executa
 - Código sem testes
- Entregas similares serão desconsideradas
- Você terá até o dia 23/05 às 11h59 para enviar sua solução.

Menu de cadastro de usuários

Validação de login e senha

A forma mais básica de autenticação de usuários é por meio dos pares de **login** e **senha**. O login pode ser um e-mail, nome de usuário ou algum documento, como o CPF e RG. Já a senha, pode possuir alguns requisitos como tamanho mínimo e quantidade mínima de alguns caracteres, como alfabéticos, maiúsculas, números, etc. Tendo isso em mente, **implemente uma função** em Python que crie um login e senha para um usuário, **recebendo** e **validando** os dados obtidos por meio de inputs. Requisitos:

- Deverá necessariamente implementar com funções
- Login: ser um dos 3 tipos a seguir
 - email no formato: xxxxx@yyy.zzz
 - user-name: apenas conter caracteres alfabéticos e numéricos. A única exceção para caractere especial é o caractere “_”
 - documentos:
 - CPF: XXX.XXX.XXX-XX ou XXXXXXXXXXXX
 - RG: XX.XXX.XXX-X ou XXXXXXXXXX
- Senha: **ao menos 15** caracteres e possuir **pelo menos 3** caracteres de cada tipo:
 - Numérico: 0-9
 - Alfabético maiúsculo: A-Z
 - Alfabético minúsculo: a-z
 - Especiais: !@#\$%&*()[\];,,:./\|

Cadastrando usuários

Utilizando a rotina de validação da questão 1, vamos agora cadastrar um usuário, sua senha e sua “role”. Para isso, utilize uma estrutura de **lista de tuplas** para capturar todos os dados do usuário. Cada tupla, deverá conter os seguintes dados:

- “login”
- “tipo-login”
- “e-mail”
- “nome”
- “RG”: nullable se CPF
- “CPF”: nullable se RG
- “data-nascimento”
- “senha”
- “endereço”: nullable
- “role”: **admin** ou **user**

Na sequência, crie um menu em que a primeira opção seja cadastrar um usuário. A opção de cadastro **deverá pedir o login e a senha** para validar o usuário que está tentando cadastrar e só deverá permitir que usuários do tipo “**admin**” realizem o cadastro. Lembre-se de sempre validar os dados inseridos, inclusive o campo de datas.

Dicas: utilizar a biblioteca datetime e reutilizar as funções da Questão 1.

Atualizando o login e senha

Ofereça a opção de atualizar o login e a senha cadastrados na questão 2. Note que apenas o próprio usuário poderá alterar sua senha. Além disso, um usuário **admin** poderá alterar a senha de qualquer outro usuário, menos de outro admin.

Proteja as senhas

Utilize uma boa prática de segurança, que é salvar a senha criptografada. Aqui, você poderá utilizar **qualquer** função de criptografia, como por exemplo a hash que criamos em aulas passadas. Garanta que sua autenticação funcionará mesmo com a criptografia.

User ID

Atribua um ID para seu usuário e insira essa informação no seu registro de forma automática. Dica: incremente automaticamente, com base no índice da lista.

Navegação de dados e usabilidade

Implemente uma navegação de dados no menu, na qual o usuário possa interagir com os dados cadastrados. Seu menu deverá permitir:

- Acesso aos dados dos usuários
 - Impressão de todo o banco de dados (dica: utilizar pprint)
 - Impressão do resultado de buscas no BD, para isso, implemente funções de busca, nas quais o usuário selecionará qual (ou quais) campos ele tem interesse.
- Atualizar os dados de um usuário
 - O menu não deverá exigir que o usuário saiba seu id, mas sim oferecer funções que nos permitam navegar nos registros e selecionar qual usuário será trabalhado. Exemplo: listar todos os usuários com nome iniciando com a letra “R”. Dica: utilizar as funções acima.
- Deletar um usuário
 - Atente-se aos requisitos do item anterior

Lembre-se de respeitar as premissas de **role** e definições de quais usuários podem alterar determinado registro.

Extras:

- Utilizar arquivos “.json” para persistir as informações criadas.
- Utilizar estruturas distintas para credenciais (usuário e senha) e dados de usuários