Trabalho 1

PROBLEMA:
Implementar um sistema orientado a objetos em Python para simulação de uma urna eletrônica para votação em reitor e pró-reitores.

ESCOPO DO DESENVOLVIMENTO:
Em uma eleição da UFSC, cada centro possui uma urna eleitoral. Cada urna eleitoral deve ser configurada para um determinado turno (primeiro ou segundo) de uma eleição, bem como devem constar na urna os candidatos inscritos para os cargos de reitor e pró-reitores. Cada candidato concorre para um determinado cargo (reitor ou pró-reitor), sendo filiado a uma determinada chapa.

A urna estando homologada, no dia da realização de votação, o voto pode ser realizado. Cada eleitor deverá votar em um reitor e um pró-reitor para cada pró-reitoria (Graduação, Pesquisa e Extensão). Considere que a urna deve ser configurada para aceitar uma quantidade configurável de eleitores e candidatos. A lista completa dos votos deve ser guardada na urna.

Existem três grupos de eleitores: professores, técnicos administrativos e alunos.

Encerrada a votação, a urna totaliza os votos válidos (por candidato para cada cargo) e os votos inválidos (brancos e nulos). Além disso, o sistema deve totalizar quais foram os candidatos eleitos para cada cargo.

Considere algumas regras:

1. No cadastro do(a) candidato(a) deve ser informado seu número para votação. São válidos números de 01 a 98. A tentativa de cadastro de número fora desta regra ou o cadastro de um número já pertencente a outro candidato deve ser tratada pelo sistema.

2. Um voto válido é aquele cujo número do(a) candidato(a) digitado na urna é o número de um(a) candidato(a) cadastrado(a) para o cargo.

3. Um voto branco ocorre quando o eleitor não informa um número de candidato. Nesse caso, deve ser atribuído valor 00 como um número de candidato.

4. Um voto nulo ocorre quando o eleitor informa qualquer número de candidato não cadastrado para o cargo. Nesse caso deve ser atribuído valor 99 como um número de candidato.

5. Os votos registrados não podem permitir a identificação de qual candidato(a) foi votado(a) por cada eleitor(a). No entanto, o registro dos eleitores que já votaram deve ficar armazenado na urna, impedindo que o mesmo eleitor vote novamente.

6. A totalização dos votos deve levar em conta a proporcionalidade dos grupos de eleitores (professores, técnicos administrativos e alunos). Cada grupo possui um terço

dos votos na proporcionalidade de 40.000 alunos, 2.500 professores e 3.100 técnicos administrativos em educação.

RESTRIÇÕES DE ESCOPO:
Para simplificar este trabalho, o sistema contempla somente algumas das funcionalidades de uma urna eletrônica, não abordando todas as questões de segurança, criptografia e funcionalidades do sistema do mesário, por exemplo.

Para este tema padrão, serão considerados:

- **cadastros**: de chapas, candidatos(as), eleitores(as), urnas, etc.
- **registros**: de votos
- **relatório**: totalização dos votos por grupo de eleitores (quantos votos cada candidato recebeu), e resultado da eleição (levando em conta a proporcionalidade dos grupos).

ENTREGAS:
Parte 1: Deve ser postado um arquivo ZIP por equipe no Moodle até o dia **03/10/2022** às 18:00hs, contendo:

- Um documento em formato PDF com a listagem das telas (menus) do sistema e como serão utilizadas e a divisão das atividades do trabalho entre os membros da equipe;
- Todas as figuras com os diagramas de classes, seguindo a notação UML 2. No caso de mais de 10 classes, deve ser elaborado um diagrama de classes por funcionalidade do sistema, englobando: controladores, classes de apresentação/telas e as entidades.

Parte 2: Deve ser postado um arquivo ZIP por equipe no Moodle até o dia **31/10/2022** às 18hs, contendo:

- Código fonte completo do sistema orientado a objetos em Python.
- Figuras contendo os diagramas de classes atualizados seguindo a notação UML 2. No caso de mais de 10 classes, deve ser elaborado um diagrama de classes por funcionalidade do sistema, englobando: controladores, classes de apresentação/telas e as entidades.

APRESENTAÇÕES / DEFESAS DOS TRABALHOS:
A apresentação do trabalho será realizada **presencialmente** por cada grupo em sala de aula ou laboratório. Na apresentação deve ser **demonstrado o sistema desenvolvido em execução**. Cada um dos membros do grupo deverá **apresentar a parte do código-fonte que desenvolveu**, explicando o código-fonte implementado. A nota de cada membro da equipe será **individual**, dependendo da sua **participação individual no desenvolvimento do trabalho e na apresentação**. A **participação individual** do aluno no trabalho pode ser **comprovada** ao professor por meio da listagem dos *commits* na ferramenta de controle de versões do código-fonte (GitHub ou similar), ou outra forma de relatório que comprove de forma inequívoca que os códigos foram elaborados pelo aluno.

O aluno que não estiver presente na apresentação (salvo motivo regimentalmente justificável) receberá nota zero no trabalho.

CRITÉRIOS DE AVALIAÇÃO:

- Cadastro da urna, das chapas, candidatos e eleitores que votam naquela urna, contemplando para cada um: inclusão, exclusão, alteração e listagem (2,0 pontos).
- Registro dos votos, contemplando a identificação inicial do eleitor, o início da votação do eleitor, a inclusão do voto em reitor(a), a inclusão dos votos em pró-reitores e correção de cada voto enquanto o eleitor estiver votando (1,5 pontos).
- Encerramento da eleição, com geração da totalização do resultado da urna, listando o total de votos de cada candidato e o relatório dos vencedores da urna para reitoria e para pró-reitorias (1,5 pontos).

Além desses critérios funcionais, também serão avaliados:

- Documentação das telas (menus) do sistema e como serão utilizadas e a divisão das atividades (0,5 ponto)
- Qualidade, uso da notação e consistência do diagrama de classes (0,5 ponto)
- Utilização correta de: associação, agregação e composição (1,0 ponto)
- Utilização correta do MVC (1,0 ponto)
- Utilização correta de: herança e classes abstratas (1,0 ponto)
- Tratamento de todas as exceções (1,0 ponto)