Aplicação para inscrição em turnos

Introdução

Pretende-se desenvolver uma aplicação que permita fazer a inscrição de alunos de um UC nos respectivos turnos. A aplicação processa de forma independente turnos teóricos e práticos e ignora completamente os horários.

Os turnos são identificados por um nome e é necessário especificar a capacidade de cada um. A aplicação necessita também da lista de alunos inscritos na UC, para validação das marcações e detecção de estatutos prioritários (i.e., estudantes-trabalhadores e portadores de deficiência).

Um ciclo de utilização da aplicação compreende 3 fases diferentes:

- Configuração – fase inicial em que se inserem os turnos e alunos inscritos
- Marcação – período em que os alunos podem indicar as suas preferências por turnos
- Geração – fase final em que são produzidas as listas de alunos colocados em cada turno

Devido à eventual extinção e/ou a alteração das vagas dos turnos, a aplicação poderá ser utilizada para repetir a geração dos turnos da mesma UC sem perda dos dados das marcações anteriores.

Requisitos

Funcionais:

1. Os dados de entrada são:
    a. Lista de alunos inscritos na UC
    b. Lista de turnos, com respectivo nome e número de vagas
    c. Preferências dos alunos pelos turnos (ordenação dos turnos), etiquetadas temporalmente.
2. Os dados de saída são as listagens ordenadas de alunos em cada turno, incluindo a indicação de alunos não inscritos, e alunos não colocados por falta de vagas. A ordem é a da data-hora das marcações.
3. Só permite realizar uma fase do ciclo de utilização se existirem os dados da fase anterior, e permite limpar todos os ficheiros de dados.
4. Cada aluno é colocado num só turno e, se se inscrever mais do que uma vez, conta a última marcação.
5. As vagas são ocupadas por ordem da marcação e da preferência. Os alunos com estatuto de estudante-trabalhadore e portador-de-deficiência têm prioridade, ou seja, colocados antes dos alunos com outros estatutos.
6. Na eventualidade de ser necessário repetir a geração de turnos:
    a. Os alunos para os quais seja possível (colocados abaixo do novo limite), podem congelar previamente a sua colocação.
    b. Todos os alunos podem alterar a sua marcação, ficando a contar a nova data-hora.

Não funcionais:

- O código deve ser estruturado de forma orientada-aos-objectos.
- As informações que necessitem de ser persistentes devem ser guardadas em ficheiros.
- O formato dos ficheiros de entrada e saída deve ser compatível com o Excel.

Plano

A aplicação será desenvolvida em duas versões:

- Uma tradicional aplicação consola (tarefas 1 a 4).
- Um serviço web e um conjunto de páginas HTML estáticas simples que permitam aos utilizadores interagirem com o serviço (tarefas 5 e 6).

1. Configuração

Definir o formato dos ficheiros de configuração com os turnos e alunos inscritos.

Leitura dos 2 ficheiros, validação do formato e carregamento da informação em estruturas de dados adequadas.

Implementação de uma interface simples que permita receber os ficheiros e fornecer um sumário final, no mínimo com o número de turnos e alunos.

Gravação dos dados em 2 ficheiros próprios da aplicação para validação. O formato pode/deve ser o mesmo do input.

2. Marcações

Implementar a recepção de marcações utilizando uma interface que deve ser o mais simples possivel. Esta interface deve permitir inserir marcações individuais e em lote (através da indicação de um ficheiro).

Estudar a API da biblioteca padrão C++ para tratar e representar data-e-hora. Na marcação individual é o programa que coloca a etiqueta temporal em tempo-real.

Validar cada marcação comparando com os dados dos alunos inscritos e mostrar na consola o resultado:

- Na marcação individual
  - se for inválida, uma mensagem descritiva do erro;
  - se for de um aluno prioritário, indicar/confirmar o turno em que ficou colocado (assumir estes alunos cabem nos turnos da sua primeira preferência);
  - senão, confirmar a recepção da marcação.
- Na marcação em lote
  - para cada inscrição, se for inválida, uma mensagem descritiva do erro;
  - um sumário final, no mínimo com o número de inscrições correctas.

Guardar os dados das marcações válidas em ficheiro, para permitir a repetição do ciclo de inscrição. O formato pode/deve ser o mesmo da marcação em lote.

3. Estruturação Orientada-aos-Objectos

Concepção do conjunto de classes da aplicação, seguindo os princípios e práticas que têm sido leccionadas. É valorizada a utilização adequada de polimorfismo e padrões de desenho.

Descrição na forma de um diagrama de classes UML, incluindo as variáveis e funções membro de cada classe.

Mover o código já desenvolvido para a estrutura de classes escolhida, fazendo as necessárias alterações. O código deve compilar e executar tal como nas fases anteriores.

4. Geração de turnos

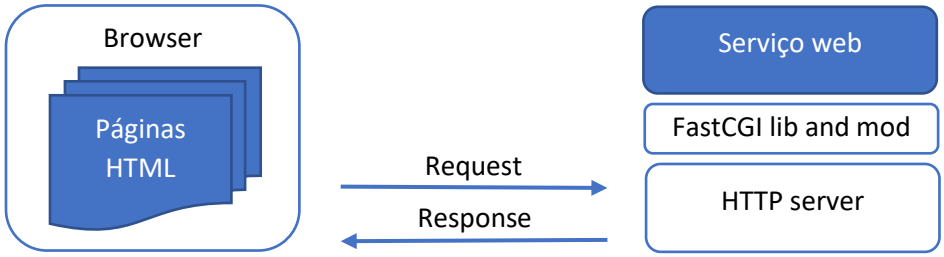
Definir o formato dos ficheiros com a constituição dos turnos.

Implementação do que falta das classes concebidas na fase anterior, incluindo o algoritmo de geração dos turnos. Num programa orientado a objectos é normal que um algoritmo envolva (esteja distribuído por) várias classes/objectos.

Descrição do algoritmo de geração dos turnos num diagrama de actividades. Experimentar a importação/abertura do ficheiros gerados no Excel.

5. Serviço web e respectivos pedidos

A figura abaixo ilustra a arquitectura da versão web, baseada na tecnologia FastCGI.



Instalação do servidor HTTP e configuração de acordo com a tecnologia web a utilizar.

Estudo da tecnologia web utilizada para implementar o serviço, incluindo os principais aspectos do protocolo HTTP e HTML básico.

Definição dos pedidos HTTP que o serviço suportará:

1. configurar os alunos inscritos
2. configurar os turnos e vagas
3. submeter inscrições em lote
4. fazer a inscrição de um aluno
5. consultar a inscrição de um aluno
6. alterar a inscrição de um aluno
7. obter os turnos
8. congelar a colocação no turno atribuído
9. eliminar os dados armazenados

Implementação de um serviço que receba os referidos pedidos e forneça uma resposta de sucesso para os pediso válidos e de erro nos outros casos. Teste do serviço utilizando uma ferramenta como o Postman ou utilizando páginas HTML.

6. Cliente web e respostas do serviço

Migração da parte lógica do código da aplicação consola para o serviço web e testes de validação através dos ficheiros produzidos.

Estudo da linguagem HTML orientado para os elementos que permitam a construção de páginas simples para as 3 fases do ciclo de utiliação da aplicação: configuração, marcações e geração de turnos.

Implementação das páginas de interacção com serviço web, nomeadamente gerando os pedidos HTTP definidos na tarefa anterior.

Implementação das respostas do serviço aos pedidos definidos na tarefa anterior, produzindo uma página HTML com o(s) resultado(s) do respectivo pedido. Pela sua importância, destacam-se as seguintes respostas aos pedidos:

2. sucesso (e o turno ocupado no caso de aluno prioritário) ou insucesso da marcação
5. dados da marcação feita anteriormente e as opções de alterá-la ou congelar o turno atribuído
7. listagem dos turnos gerados ou download de ficheiro zip respectivo, consoante a opção escolhida no pedido.

É de realçar também que a página de resposta ao pedido 5 é a que permite a realização dos pedidos 6 e 8, ou seja, os pedidos 6 e 8 não têm uma página HTML estática (i.e., ficheiro) respectiva.

Avaliação

O projecto será avaliado de acordo com três componentes:

1. Cumprimento de requisitos
2. Cumprimento das datas de entrega
3. Qualidade do software, nomeadamente:
    a. classes devidamente encapsuladas
    b. dependências entre classes reduzidas
    c. distribuição coerente e equilibrada das funcionalidades, ou seja, respectivamente, funções com relações estreitas devem estar juntas e classes não devem ser demasiado grandes.