U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

T2 - Gestão de uma AppStore

(Parte 2)

João Nogueira – up201303882@fe.up.pt
Luís Oliveira – up201304515@fe.up.pt
Pedro Fraga – up201303095@fe.up.pt

Índice

Novas Estruturas de Dados

Nesta segunda parte do projecto foi necessário implementar três novas estruturas de dados, conforme é pedido no enunciado, e integrá-las na anterior implementação: Binary Search Tree (BST), tabela de dispersão (unordered_set) e fila de prioridade (priority_queue).

Solução Implementada

Binary Search Tree

A Binary Search Tree contém elementos da classe AppToBst que contém o "id" da App a que se refere, e os tres membros que servem como termos de comparação. No início do programa, são lidas as Apps para o vetor AllTimeApps e consequentemente são lidas as informações das Apps que estáo na BST.

Desta forma, os elementos da BST são apenas referencias para as Apps, sendo necessário ir ao vetor AllTimeApps sempre que se necessita de algum elementos da App, como por exemplo, o nome. Sempre que é criada uma App, eliminada ou alterada, as mesmas alterações surtem efeito na BST.

No que diz respeito ao menu dos CRUD, não fazia sentido, por exemplo, dar ao administrador liberdade para adicionar à BST uma App que na realidade não existe, portanto o grupo decidiu permitir ao utilizador as seguinter alterações: Adicionar à BST Apps que lá não estejam presentes, Remover da BST alguma App e listá-las a todas. As Apps são ordenadas na BST pelos critérios fornecidos, ou seja, em primeiro lugar pela classificação, se a classificação for igual pelo preço e assim consequentemente.

Priority Queue

A priority_queue contém AppForSubmission e tem o nome de validation. Todas as Apps criadas pelos Developers/Companies vão agora para validation e só o Admin as pode aprovar (ou não) e colocar à venda. Se o Admin aprovar uma App ela é posta à venda e depois disso o Developer/Company tem a liberdade de a retirar da venda aos clientes e voltar a colocar, sem que volte a precisar da aprovação do Admin da AppStore.

A classe AppForSubmission tem como membros-dado uma app (classe App), um SubDate (data de submissão da app para aprovação - struct Date) e uma string username (onde é guardado o Username do Developer/Company que submeteu a App para aprovação). Esta classe, obviamente, tem o overload do operator< para que a priority_queue se possa organizar. Primeiro são comparadas as datas, depois o custo da App e por fim o nome da App.

Hash Table

A hash table é uma estrutura de dados que guarda apontadores de Apps que não estão à venda, elas continuam a existir no vector de apps, mas com um booleano “on sale” que está definido como falso e como tal não é visível em nenhum dos menus, a não ser nos CRUD. Esta hash table é atualizada cada vez que o developer muda o estado de venda de cada uma das suas aplicações, e cada vez que se execute qualquer privilégio (além do read) da conta CRUD.

Casos de Utilização

- Inscrever-se como Developer, Company ou Client;

- Como cliente:
  - Comprar Apps;
  - Classificar e comentar Apps;
  - Carregar saldo;
  - Utilizar desconto na compra;
  - Procurar Apps;
  - Comprar várias Apps ao mesmo tempo usando um carrinho de compras;
  - Alterar a password;
  - Mostrar um Top10 das apps;

- Como Developer/Company:
  - Criar App;
  - Ver as apps validadas;
  - Ver as apps que esperam aprovação;
  - Alterar a morada ou a password;
  - Mostrar um Top10 das apps;

- Como Admin:
  - CRUD – App;
  - CRUD – Client;
  - CRUD – Developer;
  - CRUD – Company;
  - Read, Update e Delete das apps que esperam aprovação (o create é substituído pelo Validate);
  - Listar todas as vendas
  - Mostrar um Top10 das apps.

Dificuldades

Uma das dificuldades que encontramos ao longo do desenvolvimento da segunda parte deste trabalho foi o facto de o código desenvolvido na primeira parte do trabalho não estar estruturado da melhor forma para aplicar as alterações pretendidas. Embora saibamos que se trata de algo complicado pois até ao final da Unidade curricular não conhecemos as estruturas que temos de implementar na sua totalidade, reconhecemos que o trabalho teria sido facilitado se soubéssemos que teriamos de implementar novas estruturas com determinada função.

Distribuição de trabalho pelos membros do grupo

Todos nós participamos ao longo do desenvolvimento do trabalho, seja na primeira parte, seja na segunda. No entanto, para melhor nos conseguirmos organizar cada um de nós ficou responsável por uma das estruturas a implementar nesta segunda parte, e consequentes alterações da seguinte forma:

- Binary Search Tree – João Nogueira;
- Priority Queues – Luís Oliveira;
- Hash tables – Pedro Fraga.