U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

StreamZ

...

Tema 3 - StreamZ (Parte 2)

2MIEIC07_G8 - AEDA
Filipe Campos | up201905609
Francisco Cerqueira | up201905337
Vasco Alves | up201808031

Descrição do Problema

Parte 1:

Desenvolver uma plataforma de streaming com diferentes tipos de utilizadores e streams que seja capaz de efetuar diversas ações sobre estes

Parte 2:

Implementar um sistema de donativos, guardando toda a informação numa BST, um sistema de “merchandising”, gerindo produtos através de uma fila de prioridade e manter um registo de todos os streamers (ativos ou com conta eliminada), utilizando uma tabela de dispersão

Descrição da Solução

- Utilizar a aplicação desenvolvida na Parte 1, recorrendo a estruturas de dados não-lineares (BST's, tabelas de dispersão e filas de prioridades) para organizar a nova informação de forma eficiente
- Operações CRUD
- Listagens totais ou parciais, com critérios a definir pelo utilizador

Algoritmos Relevantes

Nesta segunda parte do trabalho, não foi utilizado nenhum algoritmo em específico, a não ser os que já se encontram implementados relativamente às estruturas de dados utilizadas.

Principais Dificuldades

Nesta segunda parte do projeto não enfrentamos qualquer dificuldade ou adversidade.

Operadores Associados às Estruturas de Dados Não-Lineares

- BST
  - Operator<(const Donation &d) const
  - Operator==(const Donation &d) const
- Hash Table
  - Operator()(const StreamerRecord &sr) const
  - Operator()(const StreamerRecord &sr1, const StreamerRecord &sr2) const
- Priority Queue
  - Operator<(const Order &o) const
  - Operator==(const Order &o) const

```cpp
struct StreamerRecordHash {
    int operator()(const StreamerRecord &sr) const {
        int v = 0;
        for (const char &c : sr.getNickname()) {
            v = 37 * v + c;
        }
        return v;
    }

    bool operator()(const StreamerRecord& sr1, const StreamerRecord& sr2) const {
        return sr1.getNickname() == sr2.getNickname();
    }
};
```

```cpp
bool Order::operator<(const Order &o) const {
    if(getSize() == o.getSize()){
        return getDisp() < o.getDisp();
    }
    return getSize() > o.getSize();
}
```

```cpp
bool Donation::operator<(const Donation &d) const {
    if (this->getValue() == d.getValue())
        return (this->getEvaluation() > d.getEvaluation());
    return this->getValue() > d.getValue();
}
```

Estrutura de Ficheiros

Ficheiros Modificados:
users.txt (Hash Table)

Ficheiros Criados:
donations.txt (BST)
inactive_users.txt (Hash Table)
stores.txt (Priority Queue)
orders.txt (Priority Queue)

```
example_viewer1 example_streamer 1 c
    Tshirt 29.99€
    Mug 5.99€
example_viewer1 example_streamer 2 p
    Mug 5.99€
example_viewer2 example_streamer 5 p
    Jacket 59.99€
    Tshirt 29.99€
```

orders.txt

```
5
example_streamer 2
    Tshirt 29.99
    Mug 5.99
    Jacket 59.99
example_streamer2 0
    Poster 3.99
```

stores.txt

```
(viewer) kolton149 Kolton
    129 10/05/1978
    history: 43L 52D 25L 73- 129L
(streamer) audrey359 Audrey
    153 06/11/1992 B
    history: 1- 3L
```

users.txt

```
oaken123   15€   5   1
streamer   10€   5   1
oaken123   5€    1   1
soul       1€    3   3
fighter    1€    2   1
```

donations.txt

```
gerald343
heidi203
tadeo439
mikeG
oaken306
vance240
fiona344
```

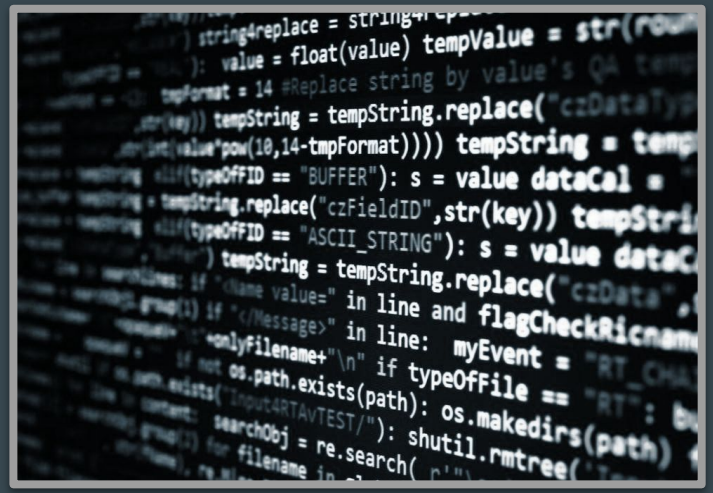
inactive_users.txt

Funcionalidades Implementadas

- Donativos
  - Capacidade de atribuir donativos a streamers
  - Leitura / escrita de informação acerca dos donativos
- Merchandising
  - Capacidade de adicionar / eliminar merchandise disponível para compra
  - Criação / deleção de pedidos de compra por parte dos visualizadores
  - Leitura / escrita de pedidos recebidos, submetidos e realizados.
- Streamers
  - Registo de todos os streamers (ativos ou com conta eliminada)
  - Atribuição de um bónus de 50 likes na primeira stream de um streamer que se volte a registar no programa
  - Leitura / escrita da informação relativamente a streamers
- Administrador
  - Alteração do limite máximo de produtos vendidos imposto pela Platform
  - Reset ao contador de produtos vendidos por cada streamer

C R U D

Create Read Update Delete



Todas as funcionalidades encontram-se completas

Funcionalidades Implementadas

Listagem / Pesquisa

```
vector<Donation> getDonationsEval(unsigned a1, unsigned a2, unsigned min_value = 0);
vector<Donation> getDonationsStreamer(string st);
vector<Donation> getDonationsValue(unsigned v);
```

- Donativos

Listagem e pesquisa de donativos com critérios a definir pelos administradores

- Merchandising

Listagem de pedidos presentes na fila de prioridade de uma Store.
Listagem de pedidos de compra pendentes e aceites de cada utilizador

```
void showOrders() const;
```

- Streamers

Listagem e pesquisa de streamers com critérios a definir pelo utilizador

```
void showStreamers(std::string active_filter = "") const;
```

Todas as funcionalidades encontram-se completas

Funcionalidade Destaque

Listagem de Streamers

```
void Platform::showStreamers(std::string active_filter) const {
    if (active_filter != "") {
        active_filter[0] = toupper( c: active_filter[0]);
        std::cout << active_filter << " ";
    }
    std::cout << "Streamers:" << std::endl;
    HashTableStreamerRecord::const_iterator it = streamerRecords.begin();
    while (it != streamerRecords.end()) {
        if (active_filter.empty() || (it->isActive() ? "Active" : "Inactive")  == active_filter) {
            if (active_filter.empty()) {
                std::cout << std::setw( n: 9) << (it->isActive() ? "Active" : "Inactive");
            }
            std::cout << it->getNickname() << std::endl;
        }
        ++it;
    }
}
```

Exemplos de Execução - BST

Menu admin

```
[1] Information
[2] Show average views
[3] Filter streams
[4] Show Top Language
[5] Show Top Stream Type
[6] Show Top Streamer
[7] Delete user
[8] Delete stream
[9] Change Max Product Limit
[10] Reset Products Sold for all Stores
[11] Show all donations
[12] Search donations between evaluation
[13] Search donations between evaluation and a minimum value
[14] Search donations by streamer
[15] Search donations by value

[0] Exit
>
```

14

```
[14] Search donations by streamer
[15] Search donations by value

[0] Exit
> 14

streamer:
> oaken123

oaken123  15€  5  1
oaken123  5€  1  1

Press enter to continue...
```

Menu de listagem de donations por streamer

11

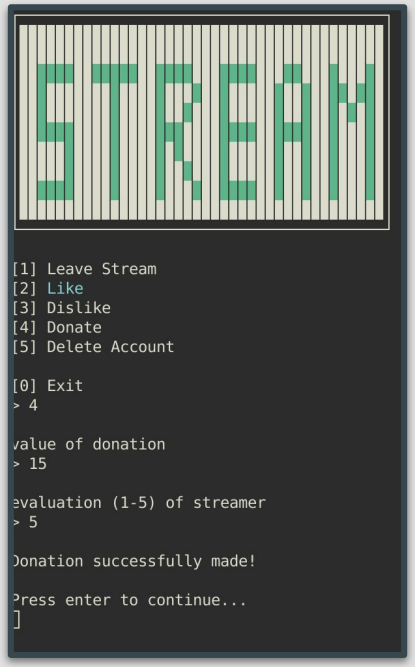
```
[11] Show all donations
[12] Search donations between evaluation
[13] Search donations between evaluation and a minimum value
[14] Search donations by streamer
[15] Search donations by value

[0] Exit
> 11

All donations:
oaken123  15€  5  1
streamer  10€  5  1
oaken123  5€  1  1
soul  1€  3  3
fighter  1€  2  1

Press enter to continue...
```

Menu de listagem de donativos



Donativo a um streamer

Exemplos de Execução - Hash Table

```
[1] Show top active streams
[2] Show top archived streams
[3] Show all active streams
[4] Search active streams by language
[5] Search active streams by minimum age
[6] Show all archived streams
[7] Show users
[8] Show streamers
[9] Sort active streams

[0] Exit
> 
```

Menu de informação

8

Menu usado para listar streamers

```
[1] Show top active streams
[2] Show top archived streams
[3] Show all active streams
[4] Search active streams by language
[5] Search active streams by minimum age
[6] Show all archived streams
[7] Show users
[8] Show streamers
[9] Sort active streams

[0] Exit
> 8

Information required: streamer type (active, inactive or leave it empty for no filter)
Streamer type
 > 
```

inactive

Listagem dos streamers

```
Information required: streamer type (active, inactive or leave it empty for no filter)
Streamer type
 > inactive

Inactive Streamers:
oaklee306
tadeo439
heidi203
bradley254
madisyn49
calum490
ameer246
```

Exemplos de Execução - Priority Queue

```
[1] Show Orders
[2] Process Orders
[3] Add Product
[4] Remove Product
[0] Exit

Products sold: 0/15
Available products:
1.  "Mug" - 5.99€
2.  "Tshirt" - 15.99€
3.  "Poster" - 3.99€
>
```

Menu de Store de um streamer

```
[1] Add Product to Order
[2] Remove Product from Order
[3] Change Availability
[4] Submit Order
[0] Exit

Available products:
1.  "Mug" - 5.99€
2.  "Tshirt" - 15.99€
3.  "Poster" - 3.99€

Current Order:
----------------------
viewer oaken123 3 p
     Tshirt 15.99€
     Poster 3.99€

Total price: 19.98€
----------------------
>
```

Menu usado para criar um novo pedido

```
[1] Show Orders
[2] Process Orders
[3] Add Product
[4] Remove Product
[0] Exit

Products sold: 0/15
Available products:
1.  "Mug" - 5.99€
2.  "Tshirt" - 15.99€
3.  "Poster" - 3.99€
> 1

keras oaken123 5 p
     Tshirt 15.99€
     Tshirt 15.99€

viewer oaken123 3 p
     Tshirt 15.99€
     Poster 3.99€


Press enter to continue...
```

Visualização de pedidos de compra

```
[1] Join Stream
[2] View stream history
[3] View liked streams history
[4] Remove stream from history
[5] View orders
[6] Remove order
[7] Access Store
[8] Delete Account

[0] Exit
> 5

Pending orders
1:
viewer oaken123 1 p
     Mug 5.99€

Completed orders
1:
viewer oaken123 3 c
     Tshirt 15.99€
     Poster 3.99€

Press enter to continue...
```

Registo de pedidos realizados e pendentes de um viewer

Esforço de cada elemento

33% 33% 33%

Filipe Francisco Vasco