StreamZ

• • •

Tema 3 - StreamZ (Parte 1)

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

Descrição do Problema

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

Descrição da Solução

- Utilização de classes adequadas para representação das entidades envolvidas;

- Utilização correta de herança e polimorfismo;

- Tratar convenientemente de possíveis exceções;

- Usar algoritmos de pesquisa e ordenação;

- Escrita e leitura de ficheiros.

Algoritmos Relevantes

Search

- Sequential Search
- Binary Search

Sort
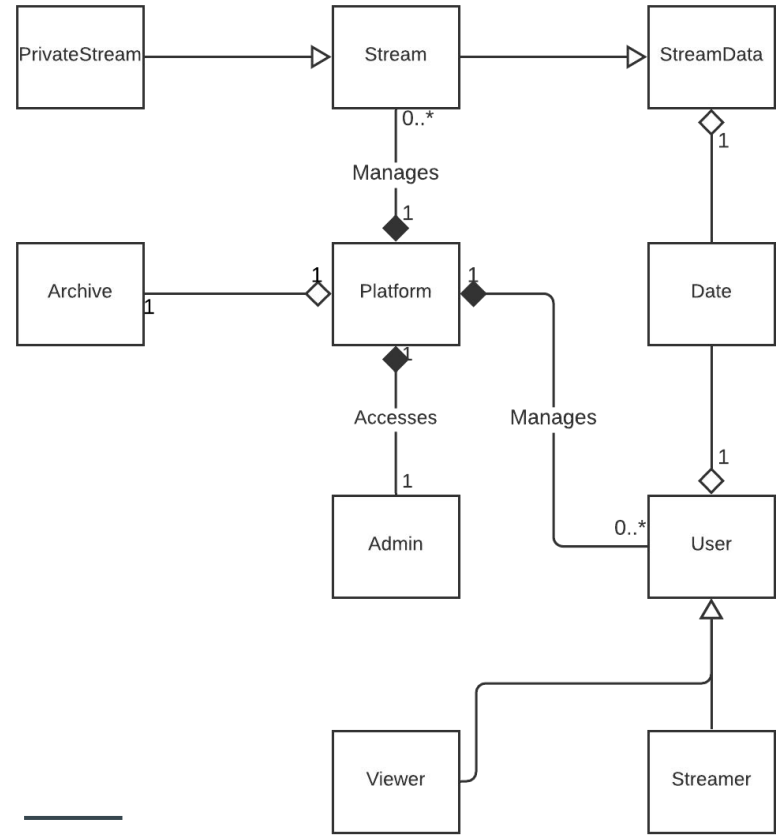
- Sort (STL)
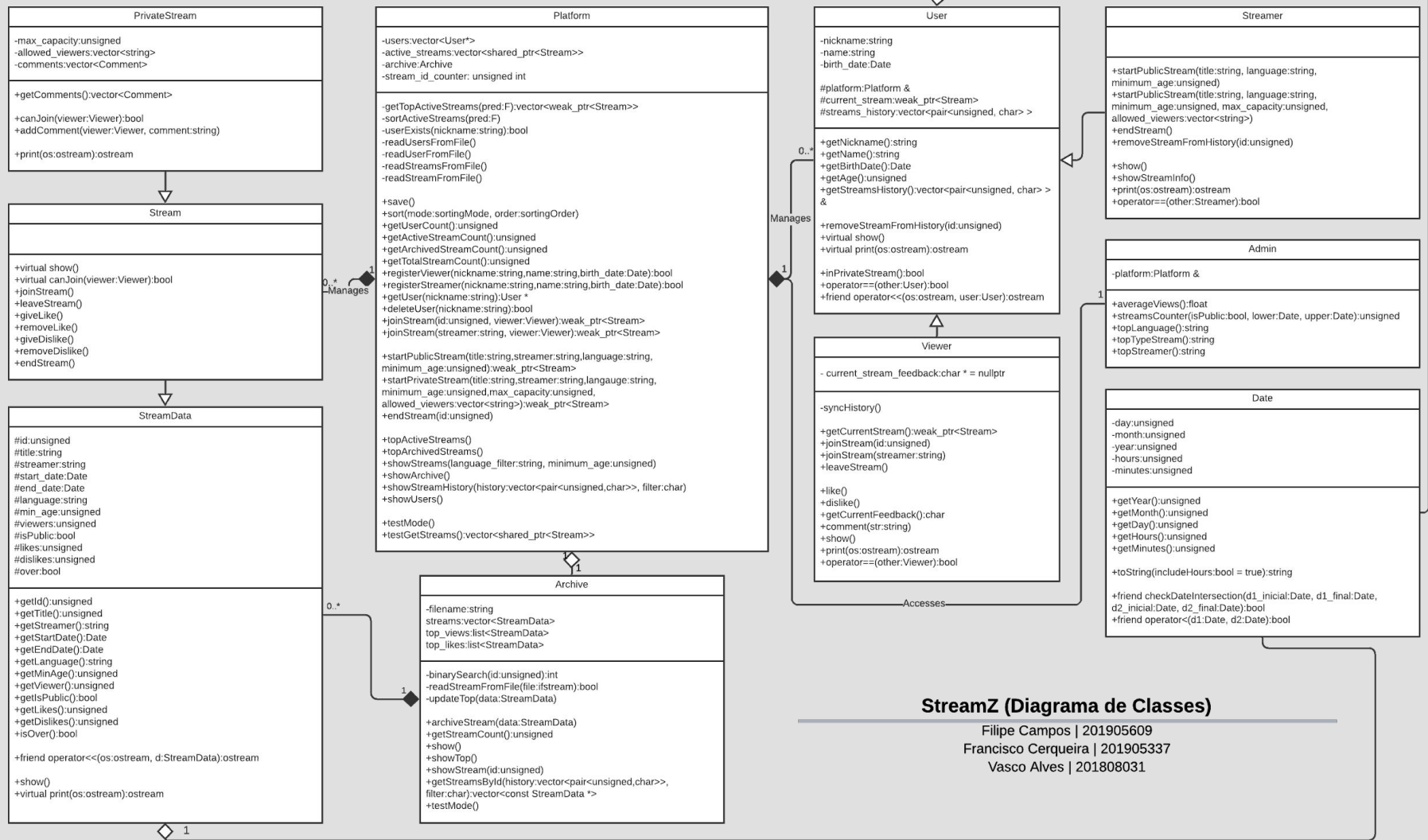- Insertion Sort
  Parcial

```cpp
template <typename F>
std::vector<std::weak_ptr<Stream>> Platform::getTopActiveStreams(F pred) {
    std::vector<std::weak_ptr<Stream>> top10;
    int n_elements = active_streams.size() > 10 ? 10 : active_streams.size();
    top10.reserve(n_elements);
    std::stack<decltype(active_streams.begin())> s;

    // Custom insertion sort, where 10 greatest elements are inserted into the first 10 index
    for(auto it = active_streams.begin(); it != active_streams.begin() + n_elements; ++it){
        auto max_it = std::max_element(it, active_streams.end(), pred);
        s.emplace(max_it);
        std::iter_swap(it, max_it);
    }
    for(int i = 0; i < n_elements; ++i){
        top10.emplace_back(active_streams[i]);
    }
    // Unwind the changes made, restoring original vector state
    auto it = active_streams.begin() + n_elements - 1;
    for(int i = 0; i < n_elements; ++i, --it){
        std::iter_swap(s.top(), it);
        s.pop();
    }
    return top10;
}
```

Diagrama de Classes

PrivateStream

-max_capacity:unsigned
-allowed_viewers:vector<string>
-comments:vector<Comment>

+getComments():vector<Comment>

+canJoin(viewer:Viewer):bool
+addComment(viewer:Viewer, comment:string)

+print(os:ostream):ostream

Stream

+virtual show()
+virtual canJoin(viewer:Viewer):bool
+joinStream()
+leaveStream()
+giveLike()
+removeLike()
+giveDislike()
+removeDislike()
+endStream()

StreamData

#id:unsigned
#title:string
#streamer:string
#start_date:Date
#end_date:Date
#language:string
#min_age:unsigned
#viewers:unsigned
#isPublic:bool
#likes:unsigned
#dislikes:unsigned
#over:bool

+getId():unsigned
+getTitle():unsigned
+getStreamer():string
+getStartDate():Date
+getEndDate():Date
+getLanguage():string
+getMinAge():unsigned
+getViewer():unsigned
+getIsPublic():bool
+getLikes():unsigned
+getDislikes():unsigned
+isOver():bool

+friend operator<<(os:ostream, d:StreamData):ostream

+show()
+virtual print(os:ostream):ostream

Platform

-users:vector<User*>
-active_streams:vector<shared_ptr<Stream>>
-archive:Archive
-stream_id_counter: unsigned int

-getTopActiveStreams(pred:F):vector<weak_ptr<Stream>>
-sortActiveStreams(pred:F)
-userExists(nickname:string):bool
-readUsersFromFile()
-readUserFromFile()
-readStreamsFromFile()
-readStreamFromFile()

+save()
+sort(mode:sortingMode, order:sortingOrder)
+getUserCount():unsigned
+getActiveStreamCount():unsigned
+getArchivedStreamCount():unsigned
+getTotalStreamCount():unsigned
+registerViewer(nickname:string,name:string,birth_date:Date):bool
+registerStreamer(nickname:string,name:string,birth_date:Date):bool
+getUser(nickname:string):User *
+deleteUser(nickname:string):bool
+joinStream(id:unsigned, viewer:Viewer):weak_ptr<Stream>
+joinStream(streamer:string, viewer:Viewer):weak_ptr<Stream>

+startPublicStream(title:string,streamer:string,language:string,
minimum_age:unsigned):weak_ptr<Stream>
+startPrivateStream(title:string,streamer:string,langauge:string,
minimum_age:unsigned,max_capacity:unsigned,
allowed_viewers:vector<string>):weak_ptr<Stream>
+endStream(id:unsigned)

+topActiveStreams()
+topArchivedStreams()
+showStreams(language_filter:string, minimum_age:unsigned)
+showArchive()
+showStreamHistory(history:vector<pair<unsigned,char>>, filter:char)
+showUsers()

+testMode()
+testGetStreams():vector<shared_ptr<Stream>>

Archive

-filename:string
streams:vector<StreamData>
top_views:list<StreamData>
top_likes:list<StreamData>

-binarySearch(id:unsigned):int
-readStreamFromFile(file:ifstream):bool
-updateTop(data:StreamData)

+archiveStream(data:StreamData)
+getStreamCount():unsigned
+show()
+showTop()
+showStream(id:unsigned)
+getStreamsById(history:vector<pair<unsigned,char>>,
filter:char):vector<const StreamData *>
+testMode()

User

-nickname:string
-name:string
-birth_date:Date

#platform:Platform &
#current_stream:weak_ptr<Stream>
#streams_history:vector<pair<unsigned, char> >

+getNickname():string
+getName():string
+getBirthDate():Date
+getAge():unsigned
+getStreamsHistory():vector<pair<unsigned, char> >
&

+removeStreamFromHistory(id:unsigned)
+virtual show()
+virtual print(os:ostream):ostream

+inPrivateStream():bool
+operator==(other:User):bool
+friend operator<<(os:ostream, user:User):ostream

Viewer

- current_stream_feedback:char * = nullptr

-syncHistory()

+getCurrentStream():weak_ptr<Stream>
+joinStream(id:unsigned)
+joinStream(streamer:string)
+leaveStream()

+like()
+dislike()
+getCurrentFeedback():char
+comment(str:string)
+show()
+print(os:ostream):ostream
+operator==(other:Viewer):bool

Streamer

+startPublicStream(title:string, language:string,
minimum_age:unsigned)
+startPublicStream(title:string, language:string,
minimum_age:unsigned, max_capacity:unsigned,
allowed_viewers:vector<string>)
+endStream()
+removeStreamFromHistory(id:unsigned)

+show()
+showStreamInfo()
+print(os:ostream):ostream
+operator==(other:Streamer):bool

Admin

-platform:Platform &

+averageViews():float
+streamsCounter(isPublic:bool, lower:Date, upper:Date):unsigned
+topLanguage():string
+topTypeStream():string
+topStreamer():string

Date

-day:unsigned
-month:unsigned
-year:unsigned
-hours:unsigned
-minutes:unsigned

+getYear():unsigned
+getMonth():unsigned
+getDay():unsigned
+getHours():unsigned
+getMinutes():unsigned

+toString(includeHours:bool = true):string

+friend checkDateIntersection(d1_inicial:Date, d1_final:Date,
d2_inicial:Date, d2_final:Date):bool
+friend operator<(d1:Date, d2:Date):bool

Manages

Manages

Accesses

0..*   1

0..*

1

1

1

1

0..*

1

1

StreamZ (Diagrama de Classes)

Filipe Campos | 201905609
Francisco Cerqueira | 201905337
Vasco Alves | 201808031

Estrutura de Ficheiros

```
(93) ezarjSlne03ue4
    by juelz304
    public PT 4393 viewers 332 likes  916 dislikes
    27/12/2018 12:47 - 27/12/2018 16:56
    minimum_age: 17
(94) 9aenmhdelgAmmnao1i
    by mohammed19
    private IT 2135 viewers 822 likes  452 dislikes
    14/02/2013 18:42 - 14/02/2013 23:53
    minimum_age: 13
```
archive.txt

```cpp
bool Archive::readStreamFromFile(std::ifstream &file) {
    std::string str;
    std::vector<std::istringstream> lines;
    for(int i = 0; i < 5; ++i){
        if(!getline( &: file >> std::ws,  &: str)){
            return false;
        }
        lines.emplace_back(str);
    }
    char discard;
    unsigned id, likes, dislikes, views, minimum_age;
    std::string title, streamer, stream_type, language, start_date, end_date;

    lines[0] >> discard >> id >> discard;
    std::getline( &: lines[0] >> std::ws,  &: title);
    lines[1] >> str >> streamer;
    lines[2] >> stream_type >> language >> views >> str >> likes >> str >> dislikes;
    start_date = lines[3].str().substr( pos: 0,  n: 16); end_date = lines[3].str().substr( pos: 19,  n: 16);
    lines[4] >> str >>  minimum_age;
    streams.emplace_back(id, title, streamer, start_date, end_date, language, views,
                         isPublic: stream_type=="public", minimum_age, likes, dislikes,  over: true);
    updateTop( data: *(streams.rbegin()));
    return true;
}
```
Read stream from archive.txt

```
(104) A title!
    by brody295
    public IT 152 viewers 95 likes  52 dislikes
    15/08/2020 18:30
    minimum_age: 12
(105) First livestream
    by rosalyn259
    private EN 15 viewers 12 likes  2 dislikes
    01/05/2019 13:45
    minimum_age: 24
    maximum_capacity: 24 allowed_users: danna398 macie464 guillermo9 owen108 vincenzo38 landon24
      01/05/2019 15:19 owen108 "Great stream!"
      01/05/2019 16:32 guillermo9 "Meh, could be better"
```
active_streams.txt

```
(viewer) kolton149 Kolton
    129 10/05/1978
    history: 43L 52D 25L 73- 129L
(streamer) audrey359 Audrey
    153 06/11/1922
    history: 1- 3L
```
users.txt

Tratamento de Exceções

```cpp
class UserDoesNotExist : public std::exception{...};

class UserAlreadyExists : public std::exception{...};

class InvalidAge : public std::exception{...};

class InsufficientAge : public std::exception{...};

class StreamNoLongerActive : public std::exception {...};

class StreamDoesNotExist : public std::exception {...};

class StreamerNotStreaming : public std::exception {...};

class InvalidAction : public std::exception {...};

class InvalidDate : public std::exception {...};
```

Exemplo: UserDoesNotExist

```cpp
Menu * LoginUserMenu::getNextMenu() {
    if(!logged_in){
        std::string username;
        if(!input::get( & username)){
            return invalidOption();
        }
        try{
            User * ptr = platform.getUser(username);
            Streamer * streamer_ptr = dynamic_cast<Streamer *>(ptr);
            if(streamer_ptr){
                logged_in = true;
                return new StreamerMenu( & platform, streamer_ptr);
            }
            Viewer * viewer_ptr = dynamic_cast<Viewer *>(ptr);
            if(viewer_ptr){
                logged_in = true;
                return new ViewerMenu( & platform, viewer_ptr);
            }
        }
        catch(const UserDoesNotExist &e){
            std::cout << "User " << e.getNickname() << " does not exist" << std::endl;
            input::waitEnter();
        }
    }
    return nullptr;
}
```

Funcionalidades Implementadas

C	R	U	D
Create	Read	Update	Delete
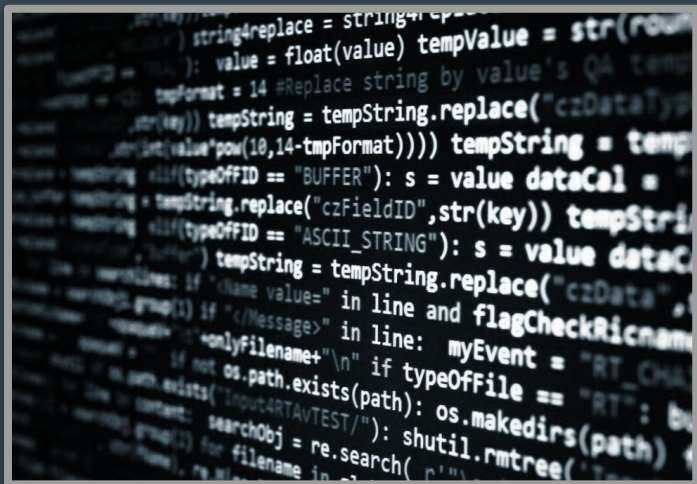
- Adicionar/remover utilizadores:
  - Streamer
  - Viewer
- Adicionar/remover streams:
  - Public Stream
  - Private Stream
- Visualizar informações sobre utilizadores e streams:
  - Histórico de streams
  - Feedback (Comentários/Likes/Dislikes)
- Ler/escrever informações de/em ficheiros
  - Active/Archive Streams
  - Users
- Administrador
  - Acesso Privilegiado

Todas as funcionalidades encontram-se completas

Funcionalidades Implementadas

Listagem / Pesquisa

```cpp
float averageViews() const;
unsigned int streamsCounter(bool isPublic, const Date &lower, const Date &upper) const;
std::string topLanguage() const;
std::string topTypeStream() const;
std::string topStreamer() const;
```

```cpp
void topActiveStreams();
void topArchivedStreams() const;
void sort(sortingMode mode, sortingOrder order);
void showStreams(const std::string &language_filter = "", unsigned minimum_age = 999) const;
void showArchive();
void showStreamHistory(const std::vector<std::pair<unsigned int, char>> &history, char filter=0) const;
void showUsers() const;
```

```cpp
enum sortingMode{
    views,
    id,
    minimum_age,
    likes
};
enum sortingOrder{
    ascending,
    descending
};
```

Todas as funcionalidades encontram-se completas

Funcionalidade Destaque

Comentários

```cpp
struct Comment{
    Date date; /**< Comment date */
    std::string nickname; /**< Viewer nickname */
    std::string comment; /**< Comment message */
};
```

```cpp
void PrivateStream::addComment(const Viewer &viewer, const std::string &comment){
    Comment c;
    c.date = Date();
    c.nickname = viewer.getNickname();
    c.comment = comment;
    comments.push_back(c);
}
```

```
Comments:
 (19/11/2020 22:29)  ronin144 "This stream is very nice!"
 (19/11/2020 22:29)  journi196 "Meh, it could be better."
```

Principais Dificuldades

De modo a garantir o correto funcionamento da aplicação foi necessário garantir que, após um streamer terminar a sua stream, todos os viewers ligados sejam removidos.
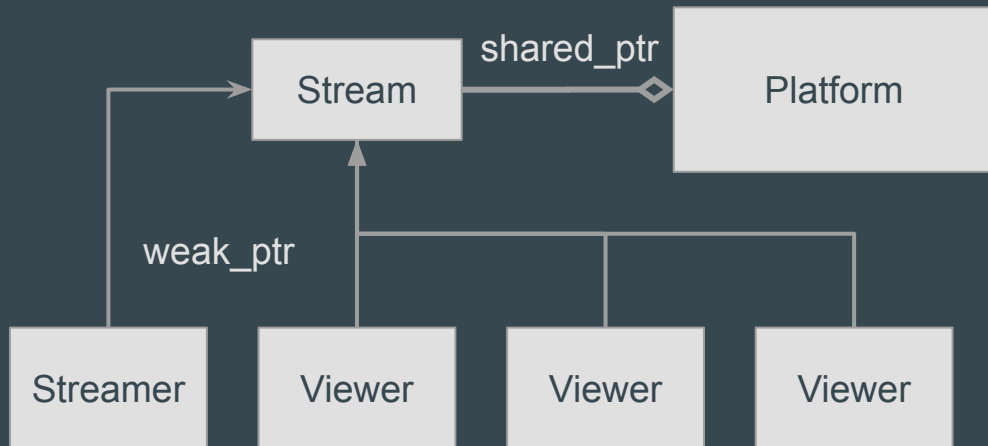

```cpp
class Platform {
private:
    std::vector<std::shared_ptr<Stream>> active_streams;
```

```cpp
class User {
protected:
    std::weak_ptr<Stream> current_stream;
```

Exemplos de Execução

```
 _     _     _     _     _ _____
(_)  _(_)   _(_)   (_)  (_|___  |
  \_(_)    (_)      \_____/  / /
         StreamZ            / /
                          _/ /_
                         |_____|

Main Menu:

[1] Login User
[2] Register User
[3] Administrator
[4] Show information
[5] Manual Save

[0] Exit
> 2▯
```

Menu Principal

```
Information required: username, name, birthdate, type of user
username
 > example123

name
 > Example Name

birth date
 > 27/7/1984

type of user (viewer or streamer)
 > viewer

Registered with success!

Press enter to continue...
▯
```

Menu de registo de utilizadores

2

1

1

```
[1] Join Stream
[2] View stream history
[3] View liked streams history
[4] Remove stream from history
[5] Delete Account

[0] Exit
> 1▯
```

Menu de Viewer que não se encontra em Stream

```
 _____ _____ _____ _____ _____
|   __|_   _| __  |   __|  _  |  _  |
|__   | | | |    -|   __|     | | | |
|_____| |_| |__|__|_____|__|__|_|_|_|
                               STREAM
```

```
[1] Leave Stream
[2] Like
[3] Dislike
[4] Comment
[5] Delete Account

[0] Exit
> ▯
```

Menu de Viewer ligado a uma Stream

Exemplos de Execução

```
[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] Sort active streams

[0] Exit
> 
```

Menu de
informação

2

7

```
All Users:
Viewer    kohen12      Kohen
Viewer    melina256    Melina
Streamer  emiliano29   Emiliano
Viewer    darwin195    Darwin
Streamer  mckenzie27   Mckenzie
```

Lista de utilizadores

```
Archive Top 10
Top by Views:
    streamer    type     id   title              lang min_age views likes dislikes start            end
 1: jose36      private  44   sE6ejroi3c         RU   12      2274  134   533      13/04/2017 04:33 13/04/2017 14:59
 2: ford451     private  31   orf4B1lreila5d     IT   20      2178  613   789      17/12/2016 12:21 17/12/2016 12:32
 3: mohammed19  private  94   9aenmhdelgAmmnao1i IT   13      2135  822   452      14/02/2013 18:42 14/02/2013 23:53
 4: leia30      private  35   ee3arlt0liCtaho    GB   13      2030  800   50       03/06/2013 03:15 03/06/2013 14:19
 5: gunner369   public   73   ngen3raub9Ihmri6   RU   19      2007  231   18       24/05/2011 09:19 24/05/2011 14:33
 6: harlow460   private  52   arFio6lw0hn4oa     IT   17      2006  139   21       25/12/2013 11:40 25/12/2013 20:57
 7: gia326      private  10   32i6andnagLe       ES   25      1994  215   292      01/03/2014 02:22 01/03/2014 04:49
 8: avah217     private  80   2hGa7eevieevnva1   EN   15      1965  865   672      09/07/2011 22:16 09/07/2011 23:49
 9: vivaan234   private  23   Bia43ednva2iov     PT   19      1952  441   303      01/09/2012 05:18 01/09/2012 22:59
10: mckenzie27  public   2    2kie7nlmiCozecelpa RU   23      1941  770   981      20/09/2016 08:20 20/09/2016 13:37

Top by Likes:
    streamer    type     id   title              lang min_age views likes dislikes start            end
 1: gunner369   public   73   ngen3raub9Ihmri6   RU   19      2007  231   18       24/05/2011 09:19 24/05/2011 14:33
 2: gia326      private  10   32i6andnagLe       ES   25      1994  215   292      01/03/2014 02:22 01/03/2014 04:49
 3: aspen432    public   46   as2e4iaapnlinn3L   RU   16      3715  153   461      28/06/2011 04:19 28/06/2011 12:41
 4: alia295     private  7    a5a2ilinM9a        RU   21      3253  151   150      07/12/2018 09:12 07/12/2018 14:26
 5: selena345   public   28   4l35aealnhseaepR   IT   12      736   141   182      21/11/2014 16:11 21/11/2014 17:56
 6: harlow460   private  52   arFio6lw0hn4oa     IT   17      2006  139   21       25/12/2013 11:40 25/12/2013 20:57
 7: jose36      private  44   sE6ejroi3c         RU   12      2274  134   533      13/04/2017 04:33 13/04/2017 14:59
 8: audrey359   public   1    oynvir9id5n3aeGua  EN   19      3530  132   535      21/09/2017 01:47 21/09/2017 04:56
 9: mckenzie27  public   96   mJikcoamz7nnies2e  GB   25      4635  124   662      22/11/2012 07:46 22/11/2012 09:47
10: juliette63  public   5    6uoitme3riljeatK   GB   24      4462  114   440      13/08/2018 15:55 13/08/2018 20:58

Press enter to continue...
```

Top Streams arquivadas

Exemplos de Execução

```
[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
[0] Exit
> []
```
Menu admin

3

```
Information required: streams type (public or private) and lower and upper time interval limit
Stream type
 > public

Lower date
 > 23/12/2019

Upper date
 > 05/01/2020

Number of public streams between 23/12/2019 and 05/01/2020: 52

Press enter to continue...
]
```
Número de streams num dado período de tempo

4

```
> 4

Most used language in streams: ES

Press enter to continue...
```
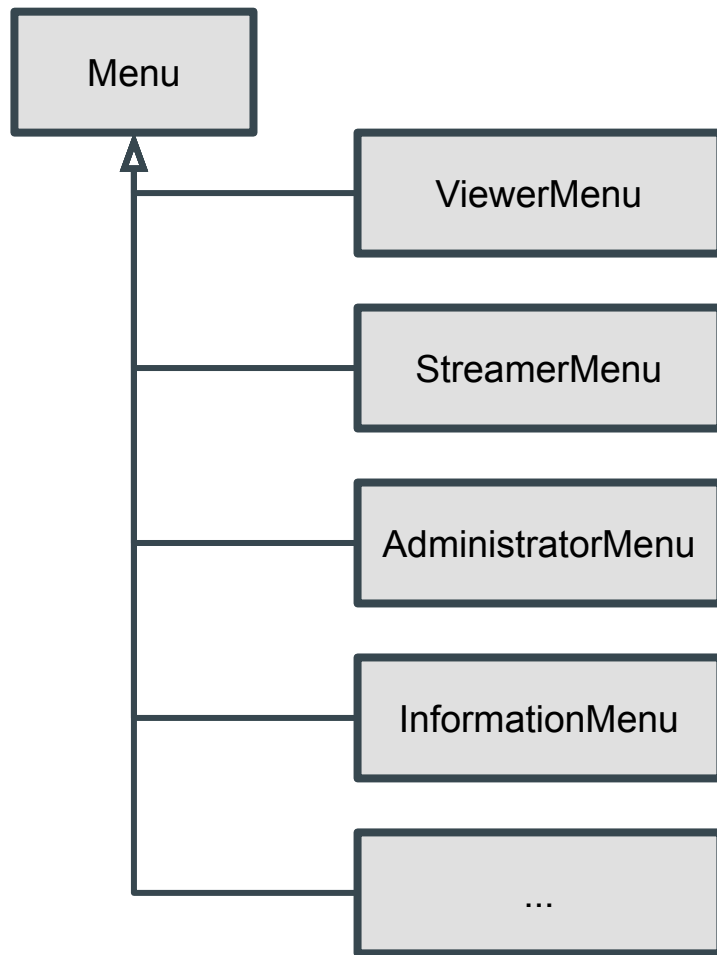Linguagem mais usada

5

```
Most used stream type in streams: public

Press enter to continue...
```
Tipo de stream mais usado

Outras Observações
Menus

```cpp
int main() {
    Platform platform;
    std::stack<Menu *> menus;
    menus.push( x: new MainMenu( &: platform));

    while(!menus.empty()){
        menus.top()->show();
        Menu * ptr = menus.top()->getNextMenu();
        if(ptr){
            if(ptr != menus.top()){
                menus.push(ptr);
            }
        }
        else{
            delete menus.top();
            menus.pop();
        }
    }
    return 0;
}
```

Esforço de cada elemento

36% 33% 31%

Filipe   Francisco   Vasco