

# Apresentação - Projeto Introdução Engenharia de Software

Gustavo Juliano Borges   Victor Emanuel Almeida   Marco  
Aurélio Guerra Pedroso

Outubro - 2020

# Conteúdo da apresentação

- Nome: DataStreamingVideoSystem (*DSVS*)
- Linguagem: C++

# Diagrama

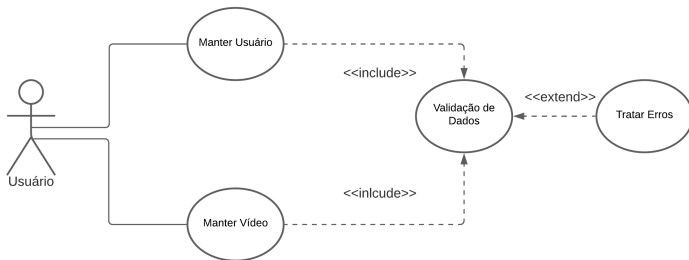


Figure: Diagrama UML de casos de uso de um sistema de streaming

# Conteúdo do main

O nosso arquivo main é dividido em:

- 1 leitura do arquivo;
- 2 menu principal;
- 3 escrita no arquivo;

# leitura do arquivo

```
// função que faz a leitura do arquivo de usuários de todos os usuário
UserListDocumentStructure readFileUserList(string file_name) {
    ifstream inputFile;
    UserListDocumentStructure vetor;

    inputFile.open(file_name);

    if( !inputFile.is_open() ) {
        cout << "Falha ao abrir o arquivo, nome inválido\n";
        exit(0);
    }

    int i;

    for (i = 0; findBeginOfStructureUser(inputFile); i++) {
        vetor.usersList[i] = readFileUser(inputFile);
    }
    vetor.numberOfUsers = i;
    return vetor;
}
```

```
// função que faz a leitura no arquivo do video todos os videos
VideoListDocumentStructure readFileVideoList(string file_name) {
    ifstream inputFile;
    VideoListDocumentStructure videoList;

    inputFile.open(file_name);

    if( !inputFile.is_open() ) {
        cout << "Falha ao abrir o arquivo, nome inválido\n";
        exit(0);
    }

    int i;
    for (i = 0; findBeginOfStructureVideo(inputFile); i++) {
        videoList.videosList[i] = readFileVideo(inputFile);
    }
    videoList.numberOfVideos = i;
    return videoList;
}
```

Figure: Leitura dos vetores do arquivo

# menu principal

```
void menuPrincipalController(UserListDocumentStructure &userList, VideoListDocumentStructure &videolist) {  
    int option = 0;  
    char selected;  
  
    system("clear");  
    menuPrincipalMessage(option);  
  
    selected = getChar();  
  
    while( true ){  
        if( selected == UP_DEFAULT && option > 0 )  
            option--;  
  
        if( selected == DOWN_DEFAULT && option < 2 )  
            option++;  
  
        if( selected == ENTER ) {  
            // acessar menu de usuários  
            if( option == 0 )  
                usersMenuController(userList);  
  
            // acessar menu de vídeo  
            if( option == 1 )  
                videosMenuController(videolist);  
  
            // rotina para finalizar o programa  
            if( option == 2 )  
                return;  
        }  
  
        system("clear");  
        menuPrincipalMessage(option);  
  
        selected = getChar();  
    }  
}
```

Figure: Interface do Menu principal

# escrita no arquivo

```
// função que imprime no arquivo de usuários todos os usuário
void printFileUserList(UserListDocumentStructure vetor, string file_path) {

    ofstream output_file;
    output_file.open(file_path);

    for (int i = 0; i < vetor.numberOfUsers; i++) {
        output_file << "{\n";
        printFileUser(vetor.usersList[i], output_file);
        output_file << "}\n";
    }

    output_file.close();
}
```

```
// função que imprime no arquivo de video todos os videos
void printFileVideoList(VideoListDocumentStructure vet, string file_path) {

    ofstream output_file;
    output_file.open(file_path);

    for (int i = 0; i < vet.numberOfVideos; i++) {
        output_file << "{\n";
        printFileVideo(vet.videosList[i], output_file);
        output_file << "}\n";
    }

    output_file.close();
}
```

Figure: Escrita dos vetores do arquivo