



Universidade do Minho
Escola de Engenharia

Desenvolvimento de Sistemas de Software

Trabalho Prático - Relatório Final

Grupo 10

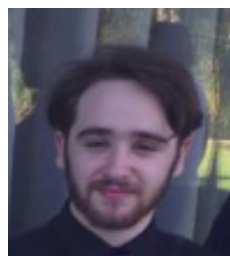
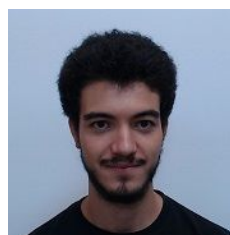
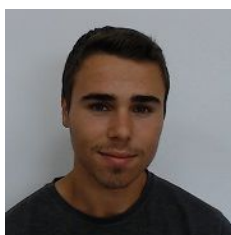
José Magalhães, A85852

Luís Ramos, A83930

Luís Vila, A84439

Ivo Baixo, A86579

Sérgio Gomes, A67645



Conteúdo

1	Introdução	4
2	Modelação	5
1	Modelo de Domínio	5
2	Modelo de <i>Use Cases</i>	6
2.1	Diagrama de <i>Use Cases</i>	6
2.2	Especificação de <i>Use Cases</i>	7
3	Protótipo da <i>Interface</i>	17
4	Diagramas relativos à Fase 2	20
4.1	Diagramas de sequência com subsistemas por <i>Use Case</i> .	20
4.2	Diagramas de <i>packages</i>	30
4.3	Diagrama de classe	31
4.4	Digrama de sequência das operações de cada subssistema .	32
3	Implementação	42
1	Diagrama de classe com DAOs	42
2	Diagramas de Sequência de Sistemas DAO	43
2.1	Iniciar Sessão	43
2.2	Terminar Sessão	43
2.3	Alterar Categoria	44
2.4	<i>Upload</i> de Conteúdo	45
2.5	Reproduzir Conteúdo	46
3	Máquinas de Estado	47
3.1	Início de Sessão	47
3.2	Alterar Categoria	48
3.3	<i>Upload</i>	49
3.4	Reproduzir Conteúdo	50
4	Diagrama de Instalação	51
5	Descrição da interface	52
5.1	Iniciar Sessão	52
5.2	<i>Upload</i> de conteúdo	54

5.3	Alterar Categoria	55
5.4	Reproduzir Conteúdo	56
6	Modelo da Base de Dados	59
4	Conclusões	60

Introdução

No âmbito da Unidade Curricular de Desenvolvimento de Sistemas de Software, foi-nos proposto elaborar um projeto faseado, que vai ao encontro do lecionado na disciplina.

O programa pretendido passa por implementar um *Media Center* em que haja partilha de música/vídeos através de múltiplos utilizadores, num determinado apartamento. Os utilizadores podem, além de reproduzir o conteúdo, dar *upload* dos seus conteúdos e alterar as categorias destes conforme pretenderem - podem, até, criar novas.

Além de utilizadores, a aplicação tem também a presença de um administrador, cujas permissões passam por gerir as contas, e a opção Convidado, que permite visualizar todos os conteúdos do sistema.

A primeira fase teve como objetivo a concretização de modelos de domínio e modelos dos vários *Use Cases* do programa, bem como a especificação dos mesmos.

Na segunda fase foi pedido para efetuar uma modelação arquitectural e comportamental de um sistema, capaz de suportar alguns *Use Cases* dados pela equipa docente - modelação essa que inclui vários tipos de diagramas.

A última fase, por sua vez, corresponde à implementação da solução final utilizando, para isso, linguagem *Java*.

De realçar que esta solução foi realizada coerentemente através dos diagramas previamente definidos, pelo que tornou-se extremamente importante um prévio levantamento de requisitos por parte do grupo.

Modelação

1 Modelo de Domínio

Numa primeira abordagem, representamos o problema em questão através de um modelo de domínio. Este permitiu a definição de alguns conceitos chave, ajudando a ter uma ideia melhor de como se iria desenvolver o projeto.

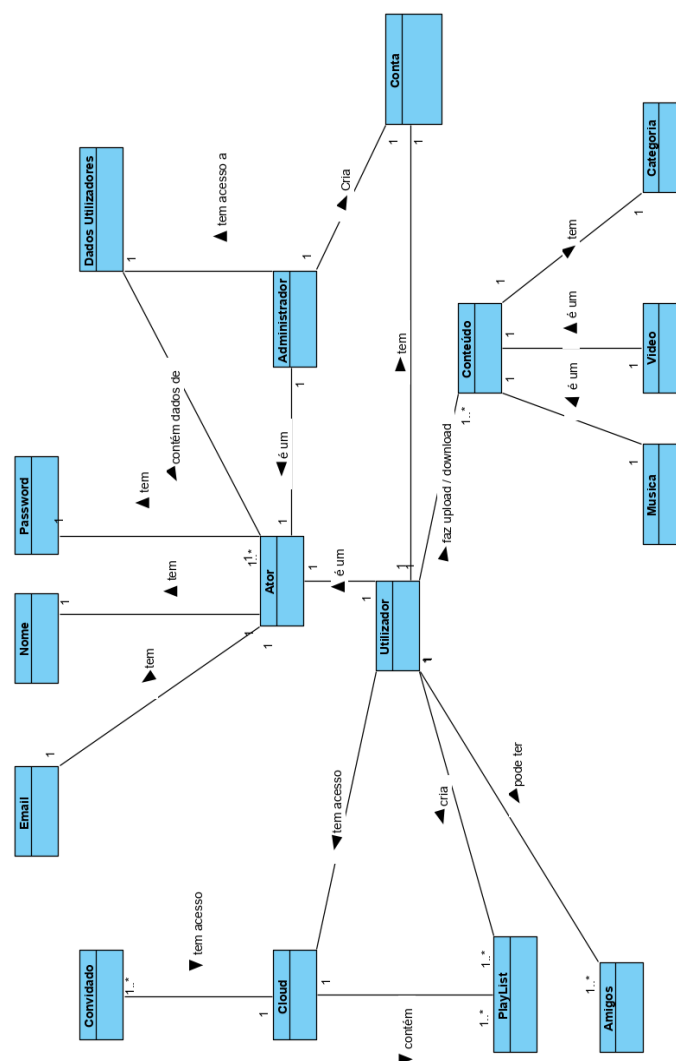


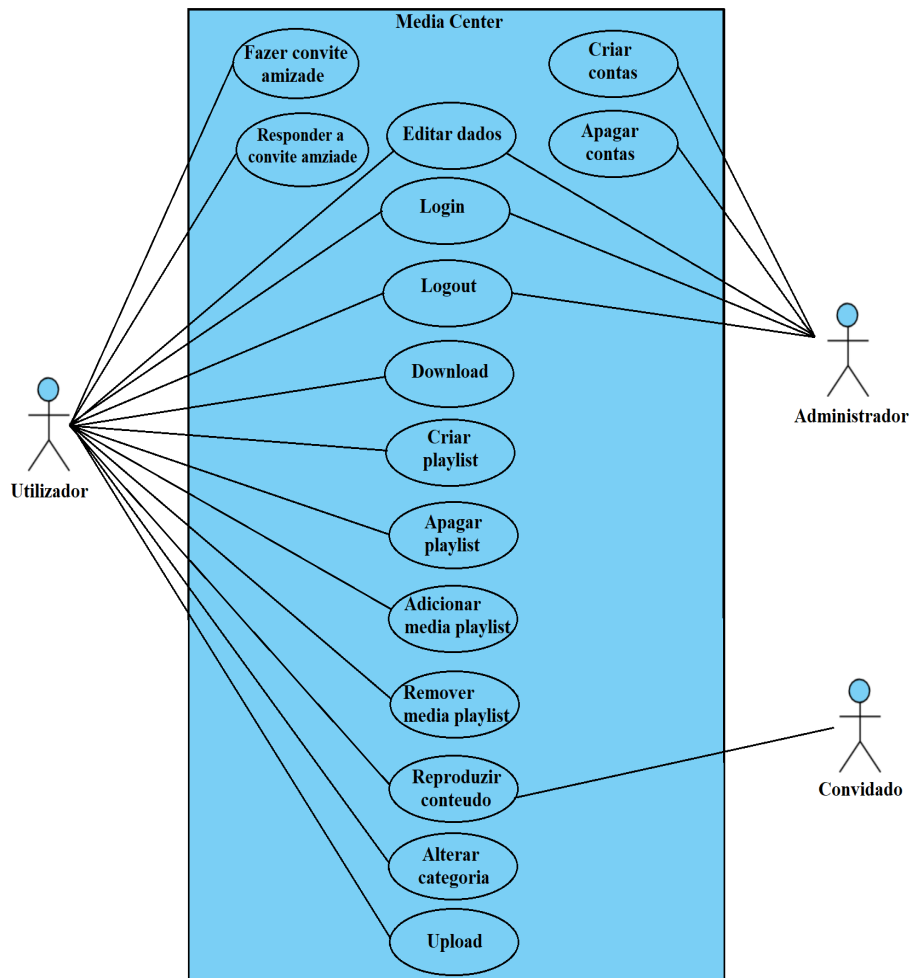
Figura 2.1: Modelo de Domínio

2 Modelo de *Use Cases*

2.1 Diagrama de *Use Cases*

O sistema suporta diversos casos, divididos por três atores: Utilizador, Administrador e, por fim, o Convidado.

O Administrador tem, além de poder da *login* e *logout* permissões para criar contas, e apagá-las e editar os seus próprios dados. O Convidado tem acesso a todo o Conteúdo presente no sistema, podendo reproduzi-lo. O Utilizador, por sua vez, auferir de tudo o que a aplicação pode oferecer, como demonstrado de seguida, exceto aos *Use Cases* relativos ao Administrador.



2.2 Especificação de *Use Cases*

UC 1: Criação de conta

Descrição: O administrador cria de conta.

Cenário: O José (administrador) cria uma conta para a Anabela (utilizador).

Pré-condição: O administrador está “*logged in*”.

Pós-condição: A Anabela já pode aceder à sua conta.

Fluxo normal:

1. Administrador seleciona opção ‘Criar conta’;
2. Sistema pede que insira os dados;
3. Administrador preenche os campos pedidos;
4. Conta criada com sucesso.

Fluxo alternativo:

- 4.1 O administrador preenche os dados de forma incorreta;
- 4.2 O sistema avisa o administrador para retificar os campos;
- 4.3 Salta para (2).

Fluxo exceção:

- 4.1 Já existe uma conta criada com estes dados;
- 4.2 O sistema avisa o administrador;
- 4.3 Criação de conta sem sucesso.

UC2: Eliminação de conta

Descrição: O administrador apaga a conta.

Cenário: O Zé (administrador) apaga a conta da Isabel.

Pré-condição: O administrador está “*logged in*”.

Pós-condição: Conta é eliminada.

Fluxo normal:

1. O administrador seleciona opção ‘Apagar conta’;
2. Sistema apresenta todas as contas existentes;
3. O administrador seleciona a conta que deseja apagar;
4. Sistema pede a *password* ao administrador a fim de confirmar esta ação;
5. Administrador insere a palavra-passe e seleciona ‘Confirmar’;
6. A conta é removida com sucesso.

Fluxo alternativo:

- 6.1 O administrador engana-se na sua password;
- 6.2 Sistema apresenta mensagem de erro;
- 6.3 Saltar para o passo (4).

UC3: Criar *Playlist*

Descrição: O utilizador cria uma nova playlist.

Cenário: O Jorge cria uma playlist com as suas músicas favoritas dos anos 80.

Pré-condição: O utilizador está “*logged in*”.

Pós-condição: O utilizador tem uma nova playlist na sua lista de playlists.

Fluxo normal:

1. Utilizador seleciona opção ‘Criar Playlist’;
2. Sistema pede ao utilizador que insira um nome;
3. Utilizador insere nome que deseja;
4. Utilizador fica com uma nova playlist na sua lista de playlists;

Fluxo alternativo 1:

- 4.1 O utilizador já tem uma playlist com esse nome;
- 4.2 Sistema avisa utilizador que o nome é repetido;
- 4.3 Salta para (2).

UC4: *Login*

Descrição: Um utilizador ou admin faz login.

Cenário: O Manuel entra na sua conta.

Pré-condição: Utilizador está a usar o sistema.

Pós-condição: O utilizador ou admin está "*logged in*".

Fluxo normal:

1. O utilizador insere o *email*;
2. O utilizador insere a *password*;
3. Sistema valida as credenciais;
4. O estado do utilizador muda de *off* para *on*;

Fluxo alternativo 1:

- 4.1 O utilizador engana-se na *password*;
- 4.2 O sistema avisa o utilizador para retificar os dados;
- 4.3 Salta para passo (1).

Fluxo alternativo 2:

- 4.1 Não existe conta com o *email* inserido pelo utilizador;
- 4.2 O sistema avisa o utilizador para retificar os dados;
- 4.3 Salta para passo (1).

UC5: Apagar *Playlist*

Descrição: Um utilizador apaga uma das suas *playlists*.

Cenário: O Jorge apaga a sua playlist Metal.

Pré-condição: O utilizador está "*logged in*".

Pós-condição: O utilizador fica sem a *playlist* na sua lista de *playlists*.

Fluxo normal:

1. Utilizador seleciona opção 'Apagar *Playlist*'.
2. Sistema pede ao utilizador que insira o nome da *playlist* a apagar;
3. Utilizador insere nome da *playlist* a apagar;
4. *Playlist* é removida com sucesso.

Fluxo exceção:

- 4.1 Não existe nenhuma *playlist* com esse nome;
- 4.2 Sistema avisa utilizador que não houve sucesso;
- 4.3 Remoção de *playlist* sem sucesso.

UC6: Adicionar media à playlist

Descrição: Um utilizador adiciona um media (música ou vídeo) a uma *playlist*.

Cenário: O Jorge adiciona 'Boyfriend-Ariana Grande' à sua *playlist* 'Basic Pop'.

Pré-condição: O utilizador está "logged in" e tem pelo menos uma *playlist* criada.

Pós-condição: O utilizador fica com mais um media adicionado a uma *playlist* sua.

Fluxo normal:

- 1. Utilizador seleciona opção 'Adicionar *Media*'.
- 2. Sistema pede ao utilizador que insira o nome da *playlist* onde deseja adicionar o conteúdo;
- 3. Utilizador insere nome da *playlist*.
- 4. Sistema verifica se *playlist* existe;
- 5. Sistema pede ao utilizador que o conteúdo que deseja adicionar;
- 6. Utilizador insere o conteúdo;
- 7. Conteúdo adicionado com sucesso.

Fluxo alternativo:

- 5.1 Não existe nenhuma *playlist* com esse nome.
- 5.2 Sistema avisa utilizador que o nome não existe.
- 5.3 Salta para passo (2).

Fluxo de exceção:

- 7.1 Não existe conteúdo;
- 7.2 Sistema avisa utilizador que o conteúdo não existe;
- 7.3 Adicionar conteúdo sem sucesso.

UC7: Upload

Descrição: Utilizador dá *upload* a uma música ou vídeo.

Cenário: O Manuel adiciona a música "*Always Ascending*" ao *Media Center*.

Pré-condição: Utilizador está "*logged in*".

Pós-condição: *Playlists* por categoria, *cloud* e *uploads* do utilizador são atualizados.

Fluxo normal:

- 1. Utilizador seleciona no menu a opção '*Upload*';
- 2. Sistema pede o nome, tipo, categoria e o conteúdo, do media;
- 3. Utilizador preenche os campos todos;
- 4. Sistema valida todos os campos;
- 5. Música/vídeo é adicionada com sucesso.

Fluxo alternativo:

- 3.1 Utilizador não encontra o género em que a música/vídeo se insere;
- 3.2 Sistema pede ao utilizador para inserir o nome da nova categoria;
- 3.3 Utilizador insere nome;
- 3.4 Sistema cria nova categoria;
- 3.5 Salta para passo (4).

Fluxo de exceção:

- 5.1 Os dados estão inválidos;
- 5.2 Upload sem sucesso.

UC8: *Download*

Descrição: Utilizador descarrega uma música/vídeo.

Cenários: O Manuel descarrega o vídeo da música "*Day I Die*".

Pré-condição: Utilizador está "*logged in*".

Pós-condição: Utilizador possui uma cópia da música/vídeo.

Fluxo normal:

1. Utilizador seleciona no menu a opção '*Download*';
2. Sistema pede ao utilizador para inserir o nome do media;
3. Utilizador escreve o nome;
4. Sistema verifica dados e permissões;
5. A música/vídeo é descarregada com sucesso.

Fluxo de exceção:

- 3.1 Utilizador não tem permissões de download;
- 3.2 *Download* sem sucesso.

UC9: Remover *media* da *playlist*

Descrição: Um utilizador remove um media (música ou vídeo) da playlist.

Cenário: O Jorge remove '*Boyfriend-Ariana Grande*' da sua *playlist* '*Pop*'.

Pré-condição: O utilizador está "logged in" e tem pelo menos uma playlist criada.

Pós-condição: O utilizador fica sem esse media na sua playlist.

Fluxo normal:

1. Utilizador seleciona opção '*Remover Media*';
2. Sistema pede ao utilizador que insira o nome da playlist onde deseja remover o ficheiro;
3. Utilizador insere nome da playlist;
4. Sistema pede ao utilizador que insira o nome do media que deseja remover;
5. Utilizador insere o nome do media;
6. Media removida da playlist com sucesso.

Fluxo alternativo 1:

- 4.1 Não existe nenhuma playlist com esse nome.
- 4.2 Sistema avisa utilizador que a playlist não existe.
- 4.3 Salta para passo (2).

Fluxo alternativo 2:

- 6.1 Não existe nenhuma media com esse nome.
- 6.2 Sistema avisa utilizador que o media não existe.
- 6.3 Salta para passo (4).

UC10: Responder a convite de amizade

Descrição: Um utilizador aceita um convite de amizade.

Cenário: O Jorge aceita o João à sua lista de amigos.

Pré-condição: O utilizador está “logged in”.

Pós-condição: O utilizador fica com mais um amigo na sua lista de amigos.

Fluxo normal:

- 1. Utilizador seleciona opção ‘Responder a convite de amizade’;
- 2. Sistema apresenta ao utilizador os pedidos de amizade por aceitar;
- 3. Utilizador aceita o convite;
- 4. A lista de amigos é atualizada.

Fluxo de exceção 1:

- 2.1 Não há pedidos;
- 2.2 Lista de amigos fica inalterada.

Fluxo de exceção 2:

- 3.1 Não aceita o pedido;
- 3.2 Lista de amigos fica inalterada.

UC11: Fazer convite de amizade

Descrição: Um utilizador quer arranjar amigos.

Cenário: É sugerido ao Jorge pessoas que tem musicas em comum com ele, podendo ele pedir amizade.

Pré-condição: O utilizador está "logged in".

Pós-condição: O utilizador envia convite de amizade.

Fluxo normal:

1. Utilizador seleciona opção 'Fazer convite de amizade';
2. Sistema apresenta ao utilizador as pessoas com músicas em comum;
3. Utilizador seleciona a pessoa a que quer efetuar o pedido;
4. Convite efetuado com sucesso.

Fluxo de exceção:

- 2.1 Não há pessoas com músicas em comum;
- 2.2 Nenhum convite é efetuado.

UC12: Logout

Descrição: Utilizador sai da sua conta.

Cenário: O Manuel sai da sua conta de utilizador.

Pré-condição: Utilizador tem sessão iniciada.

Pós-condição: Utilizador não tem sessão iniciada.

Fluxo normal:

1. Utilizador seleciona a opção Terminar Sessão;
2. A sessão do utilizador é terminada;
3. Sistema apresenta janela inicial de *login*.

UC13: Editar dados

Descrição: Um utilizador quer editar os seus dados.

Cenário: Jorge quer alterar a sua password.

Pré-condição: Utilizador está “logged in”.

Pós-condição: Dados do utilizador são atualizados.

Fluxo normal:

1. Utilizador seleciona opção ‘Editar dados’;
2. Sistema pede ao utilizador para inserir dados que pretende alterar;
3. Utilizador insere dados;
4. Dados são atualizados com sucesso.

UC14: Alterar categoria

Descrição: Um utilizador quer alterar categoria de um conteúdo.

Cenário: Jorge quer alterar a categoria da musica ”Zombie-Eleto- para *HardStyle*.

Pré-condição: O utilizador está ”logged in”.

Pós-condição: Categoria do conteúdo é alterada.

Fluxo normal:

1. Utilizador seleciona opção ‘Alterar Categoria’;
2. Sistema pede ao utilizador o tipo de conteúdo, categoria atual, nome do conteúdo e nova categoria;
3. Utilizador preenche os campos;
4. Sistema valida todos os campos;
5. Categoria alterada com sucesso.

Fluxo alternativo:

- 3.1 Utilizador não encontra a categoria que pretende;
- 3.2 Sistema pede que insira o nome da nova categoria;
- 3.3 Utilizador insere nome da nova categoria;
- 3.1 Sistema cria nova categoria;
- 3.3 Salta para passo (4).

Fluxo exceção:

- 5.1 Nova categoria é igual à categoria atual;
- 5.2 Sistema avisa utilizador;
- 5.3 Categoria não é alterada.

UC15: Reproduzir conteúdo

Descrição: Um utilizador quer reproduzir conteúdo.

Cenário: Jorge quer ouvir musica.

Pré-condição:

Pós-condição: Conteúdo é reproduzido.

Fluxo normal:

1. Utilizador seleciona opção 'Reproduzir conteúdo';
2. Sistema apresenta categorias disponíveis;
3. Utilizador escolhe categoria que pretende;
4. Sistema apresenta todos os conteúdos dessa categoria;
5. Utilizador seleciona conteúdo que pretende que seja reproduzido;
6. Conteúdo reproduzido com sucesso.

3 Protótipo da *Interface*

De seguida será mostrado o aspeto inicial idealizado para a nossa aplicação tendo este funcionado como base para a posterior implementação, que veremos no **ponto 3 - Implementação** deste relatório.



Figura 2.2: Menu inicial do sistema

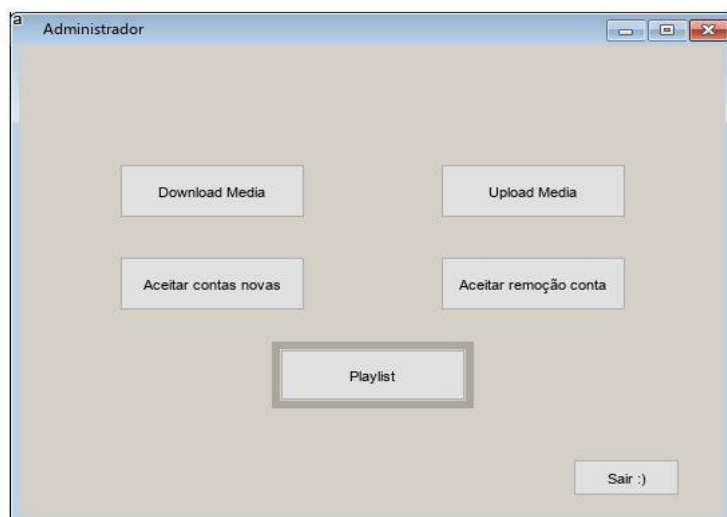


Figura 2.3: Menu referente ao Administrador

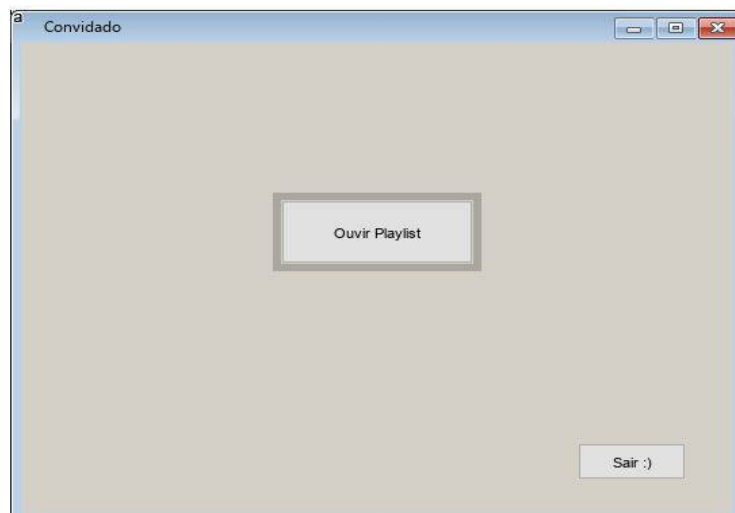


Figura 2.4: Menu referente ao Convidado

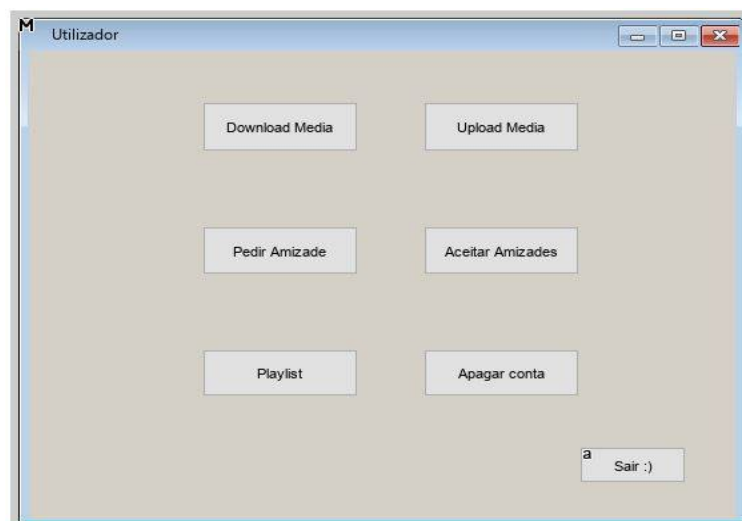


Figura 2.5: Menu referente ao Utilizador



Figura 2.6: Menu referente à opção *Playlist*



Figura 2.7: Menu referente à escolha de géneros

4 Diagramas relativos à Fase 2

4.1 Diagramas de sequência com subsistemas por *Use Case*

Os diagramas seguintes podem ser considerados simples, pois apenas efetuam a relação entre o ator e o sistema traduzindo as especificações dos *Use Cases*.

Registrar utilizador

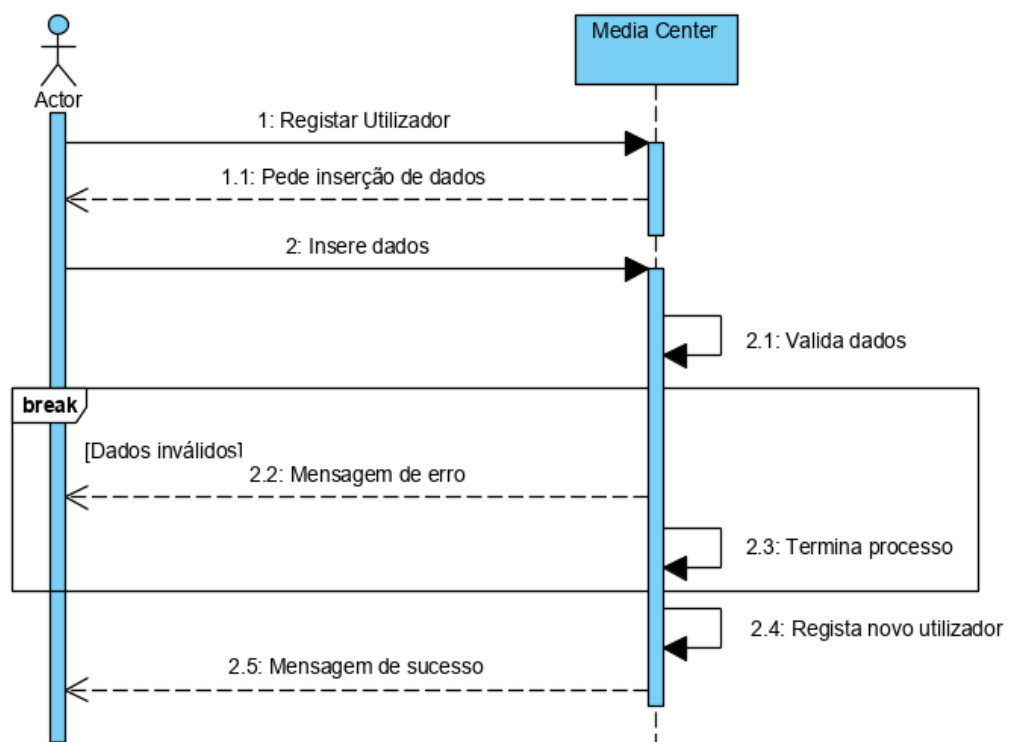


Figura 2.8: Registrar utilizador

Iniciar sessão

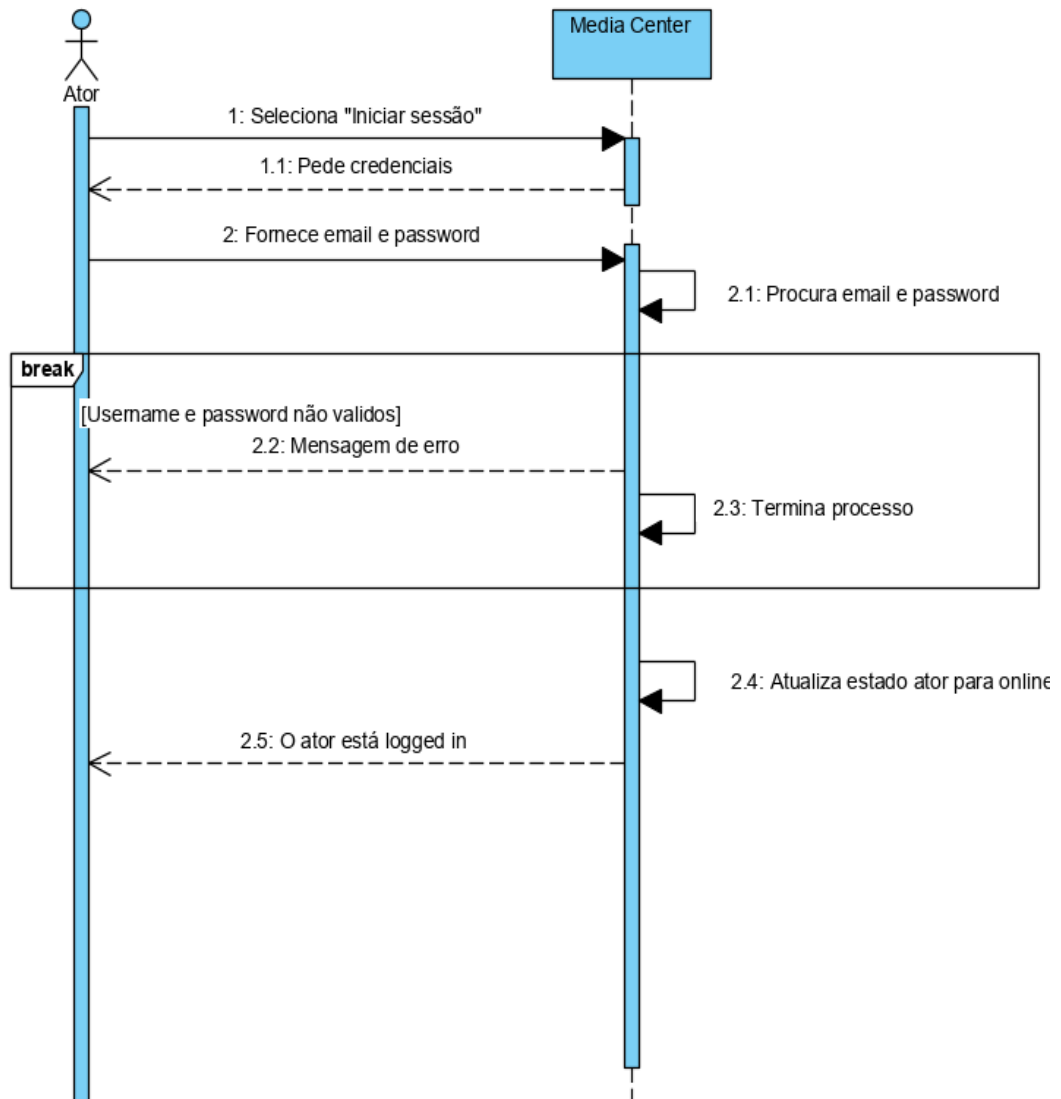


Figura 2.9: Iniciar sessão

Terminar sessão

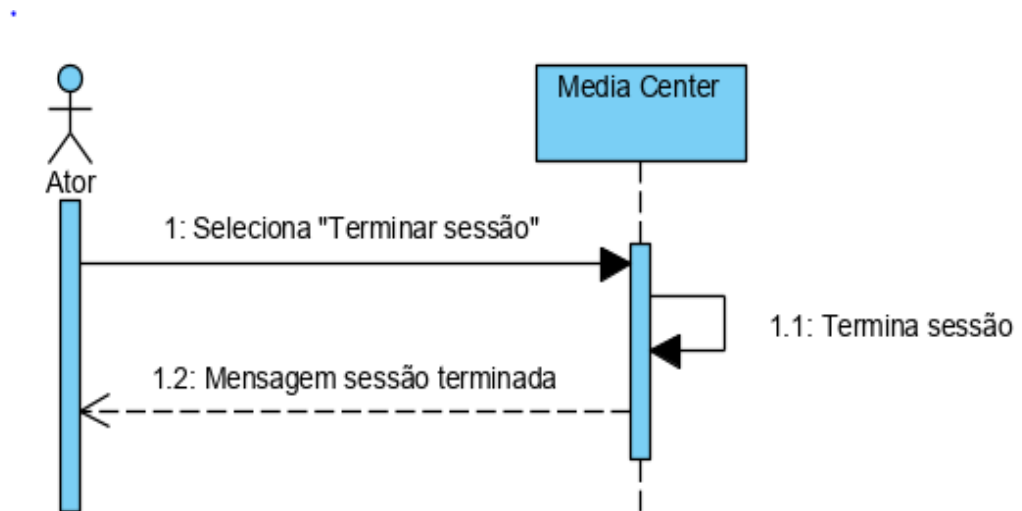


Figura 2.10: Terminar sessão

Editar utilizador

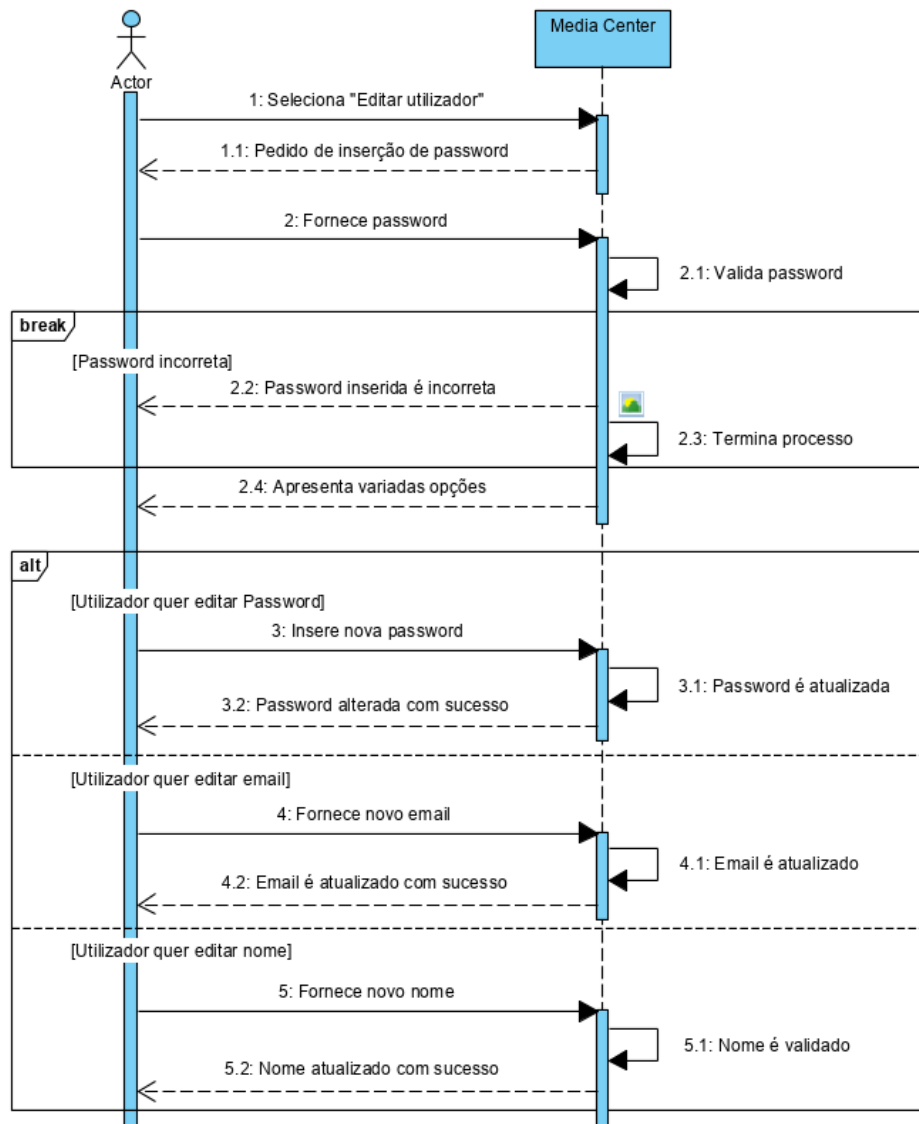


Figura 2.11: Editar utilizador

Eliminar utilizador

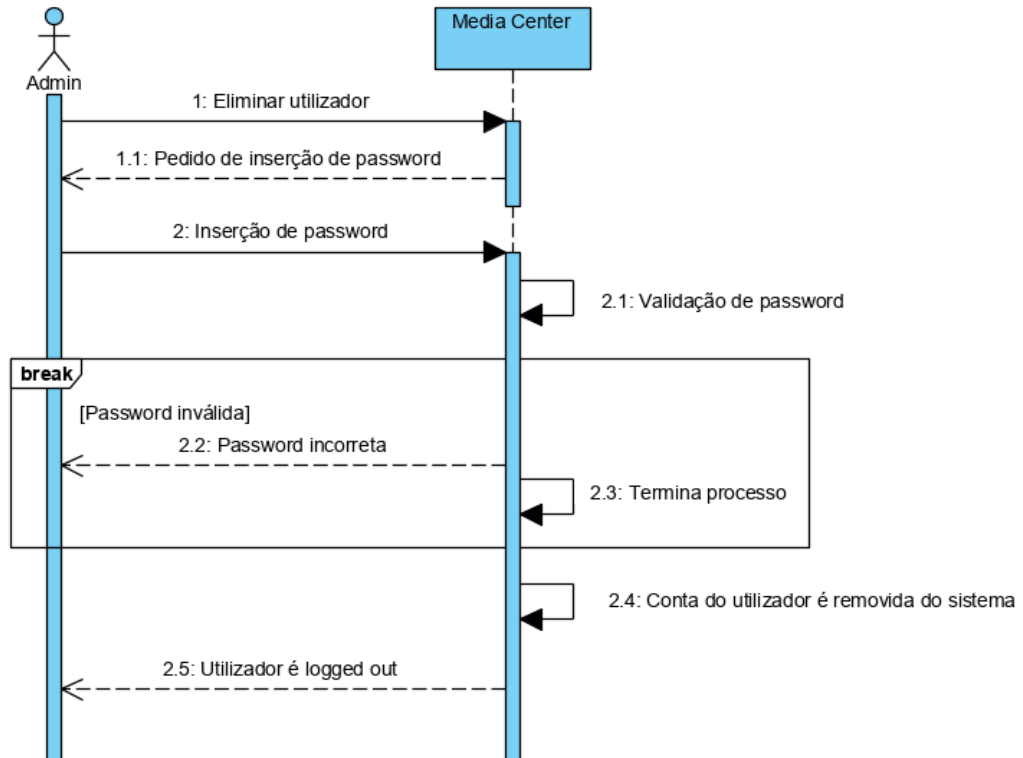


Figura 2.12: Eliminar utilizador

Fazer *upload* do conteúdo

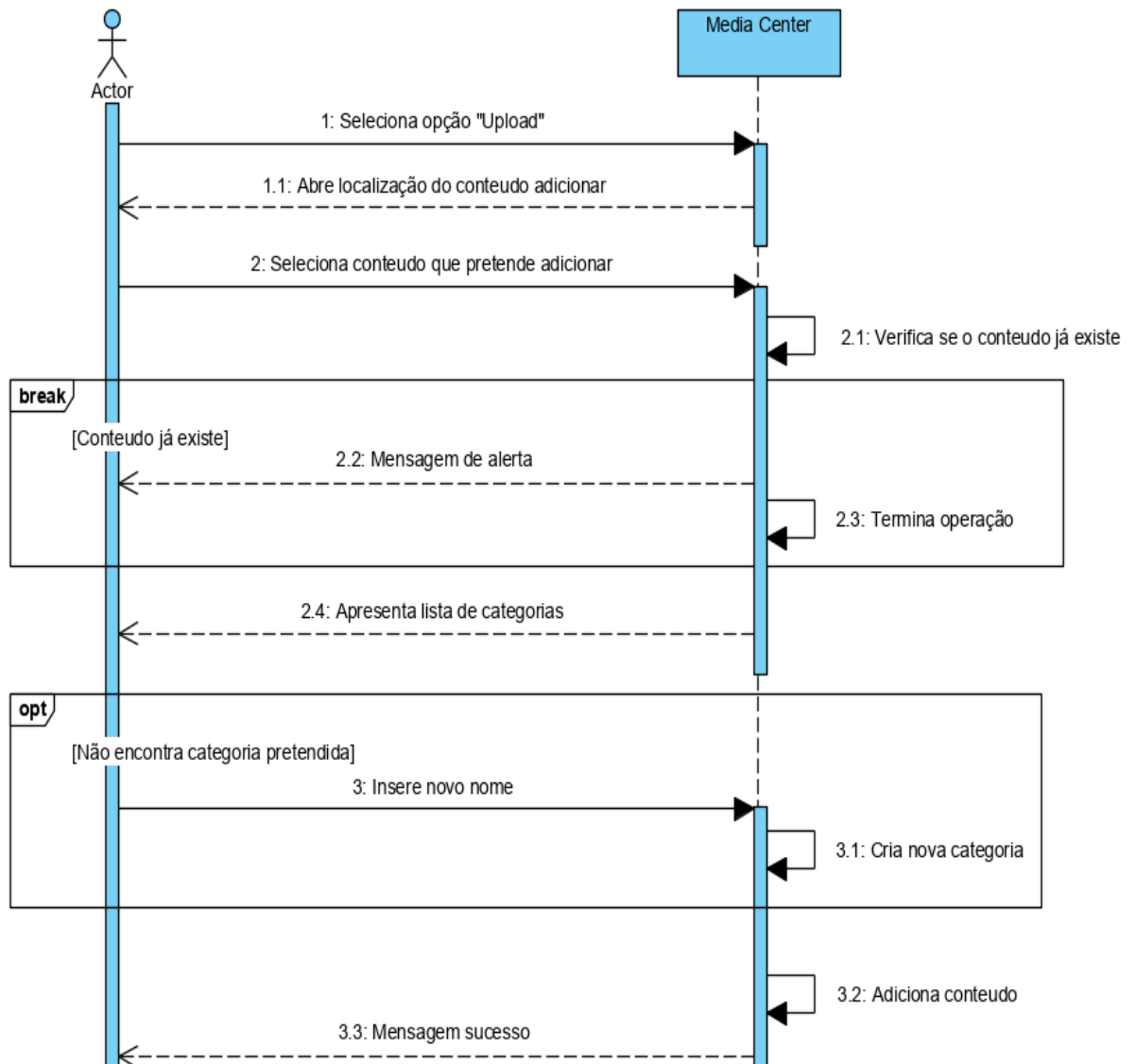


Figura 2.13: *Upload* de conteúdo

Alterar categoria do conteúdo

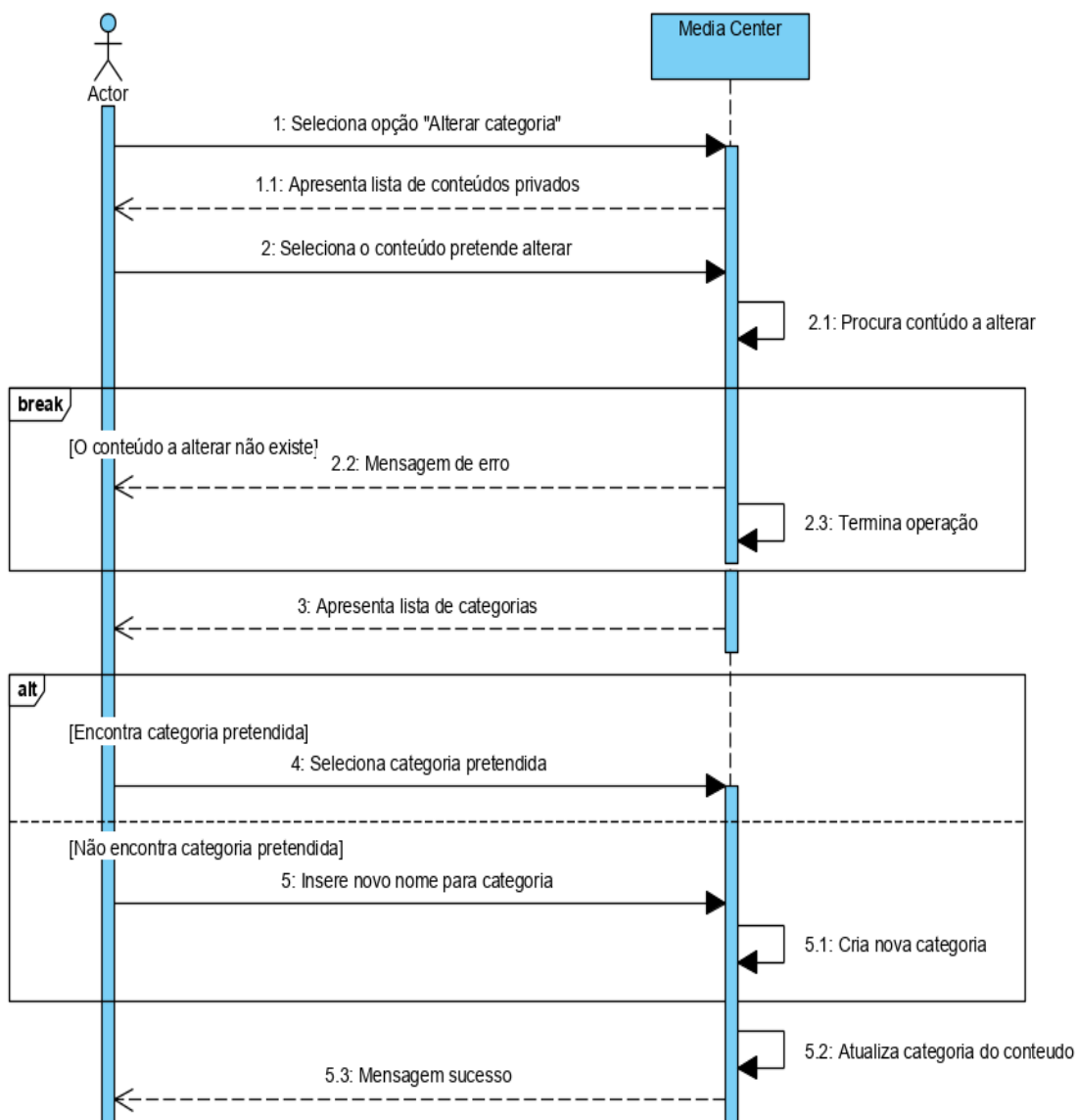


Figura 2.14: Alterar categoria

Remover conteúdo

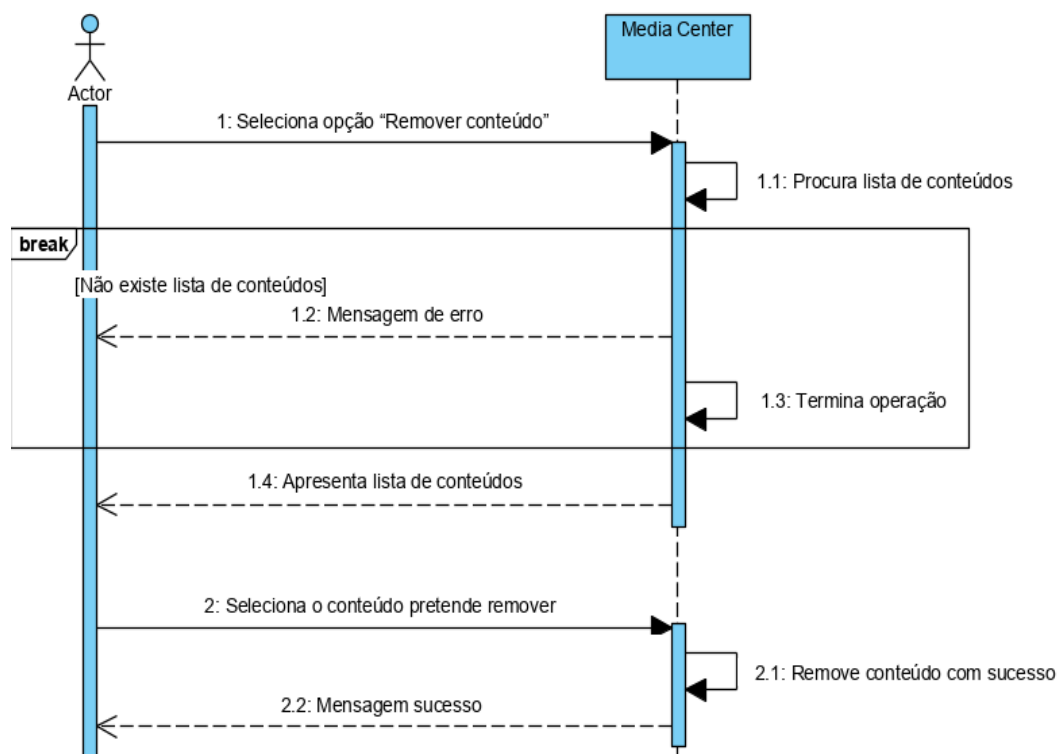


Figura 2.15: Remover conteúdo

Reproduzir conteúdo

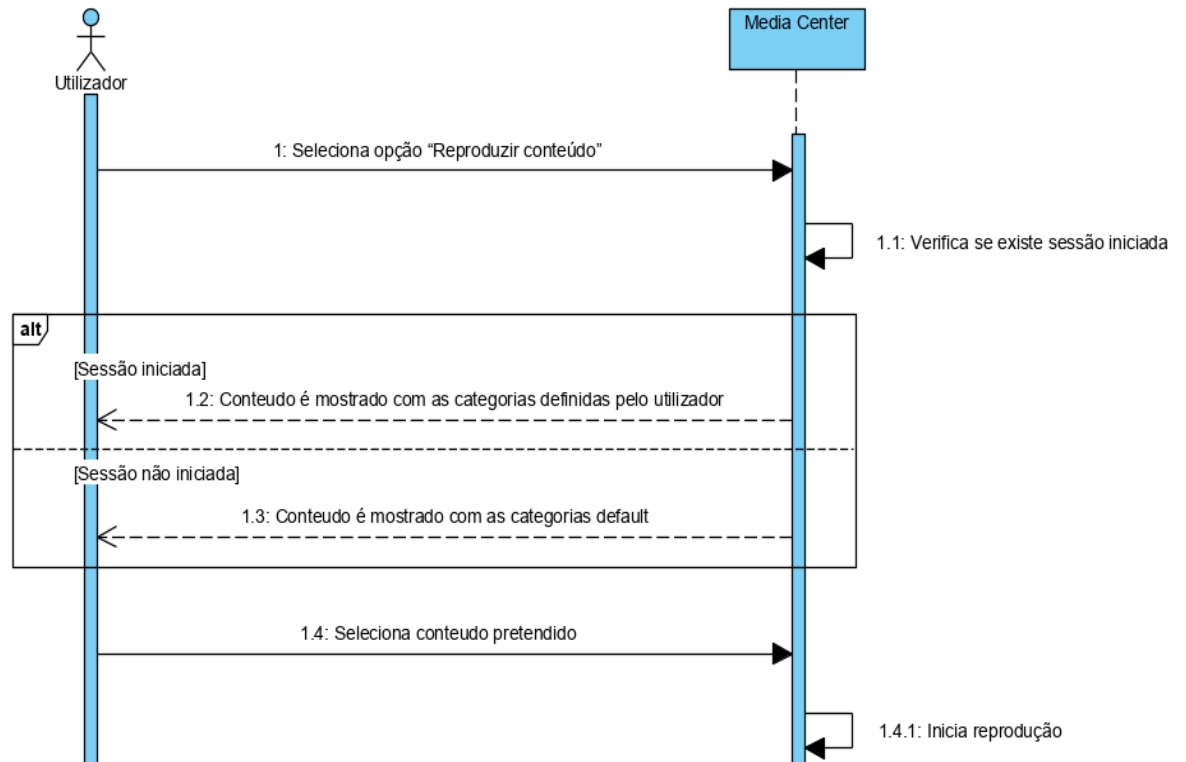


Figura 2.16: Reproduzir conteúdo

Criar *playlist*

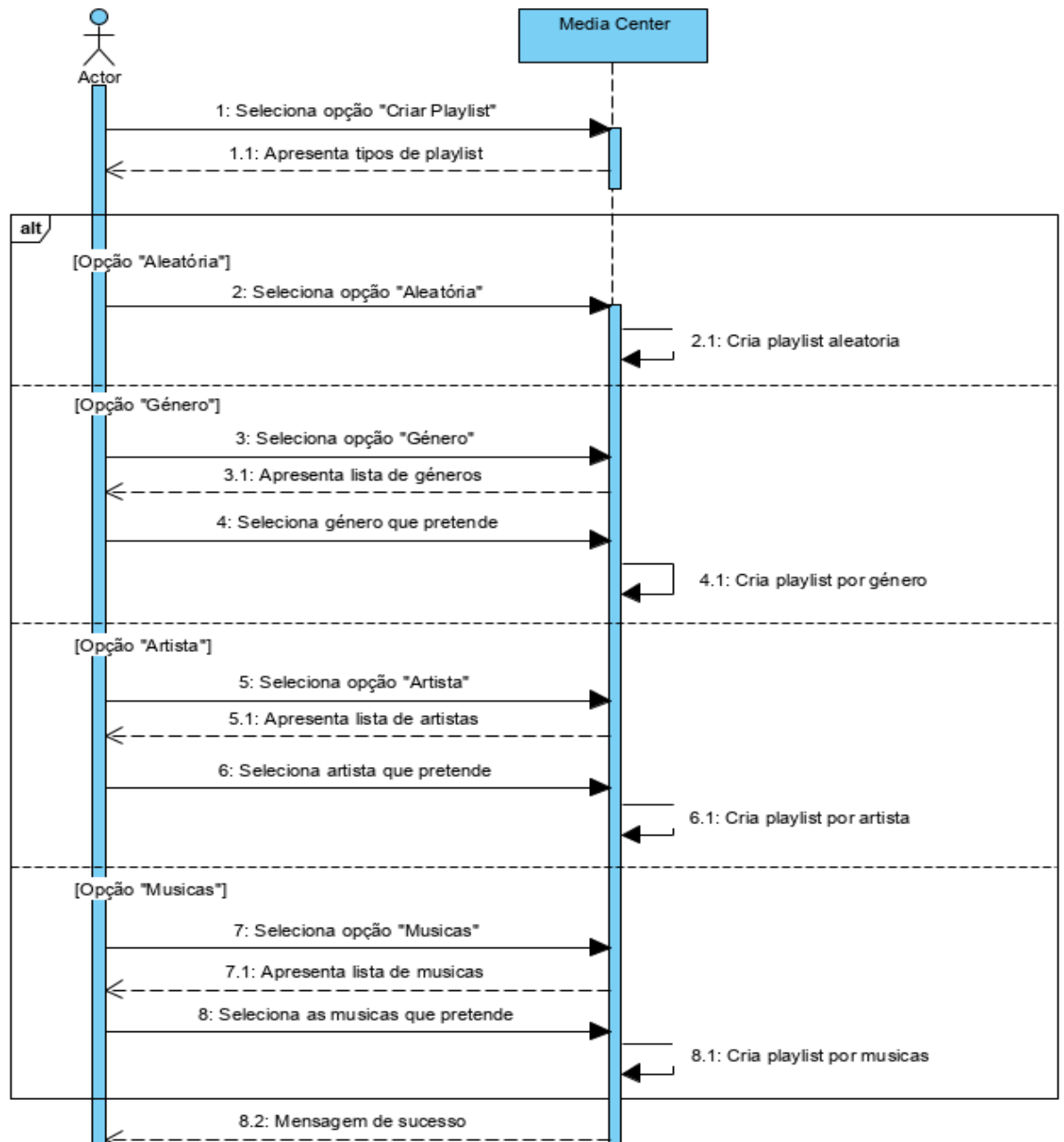


Figura 2.17: Criar *playlist*

4.2 Diagramas de *packages*

Para fazer uma boa gestão das classes do nosso projeto, utilizamos um diagrama de *package*, onde as classes são agrupadas em 3 pacotes diferentes: *'User Interface'*, *'Business Logic'* e *'Data'*, como demonstrado de seguida.

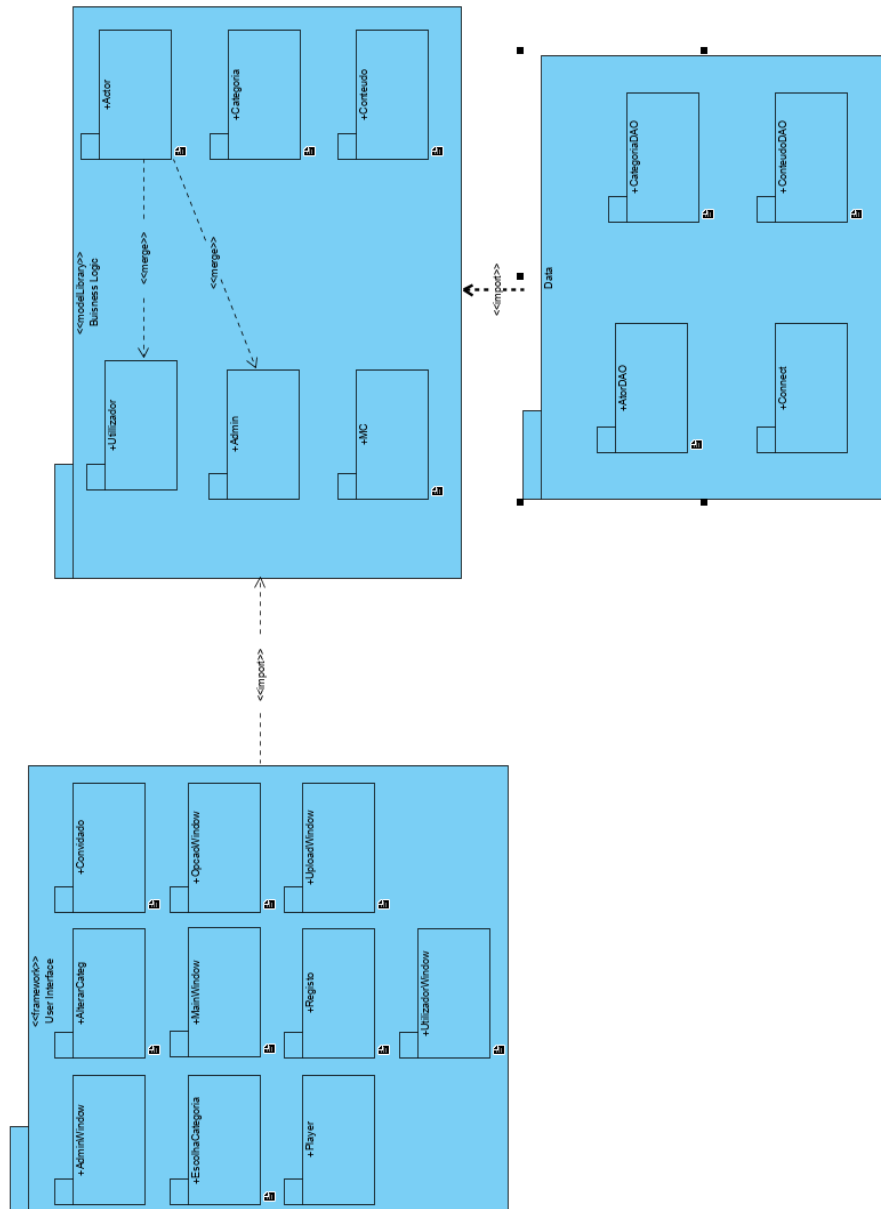


Figura 2.18: Diagrama de packages

4.3 Diagrama de classe

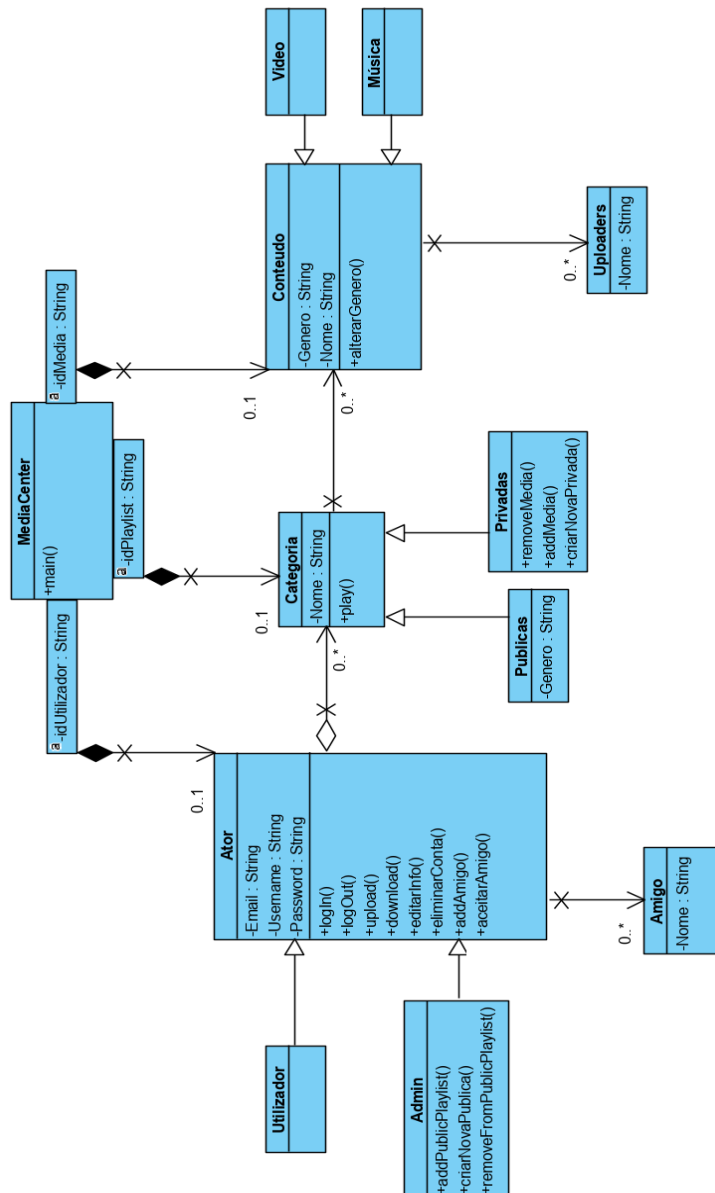


Figura 2.19: Diagrama de classe

4.4 Digrama de sequência das operações de cada subsistema

Diagramas mais complexos, que já fazem referência aos subsistemas que fazem parte do projeto, contendo já código *Java*.

Registrar utilizador

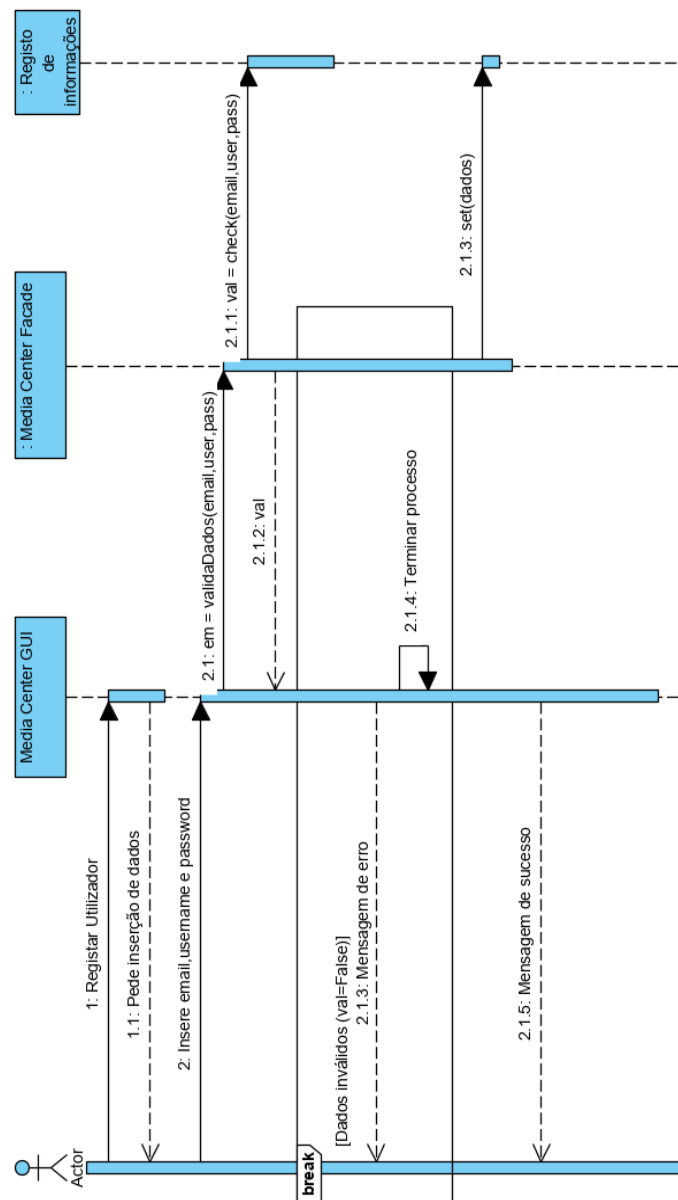


Figura 2.20: Registrar utilizador

Iniciar sessão

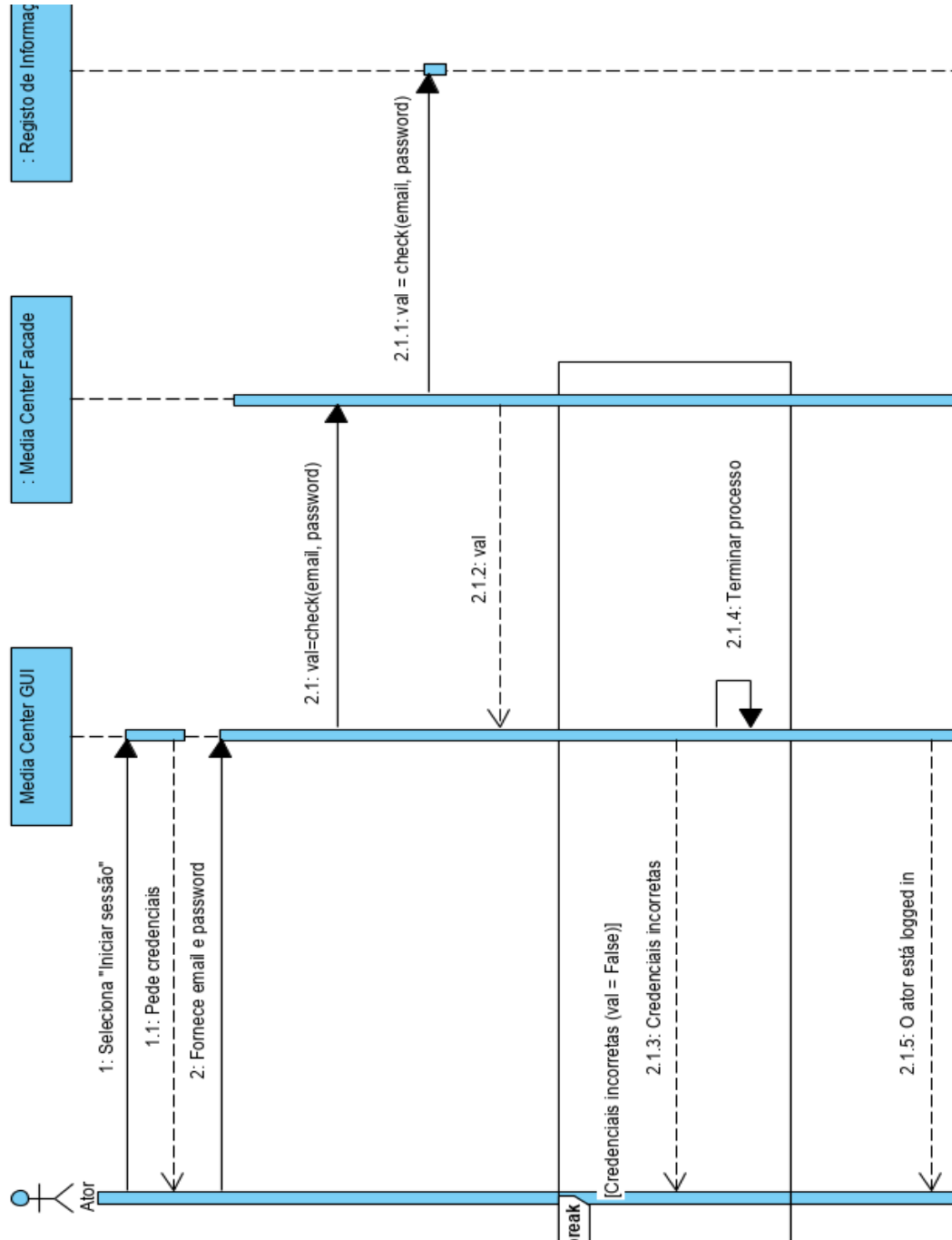


Figura 2.21: Iniciar sessão

Terminar sessão

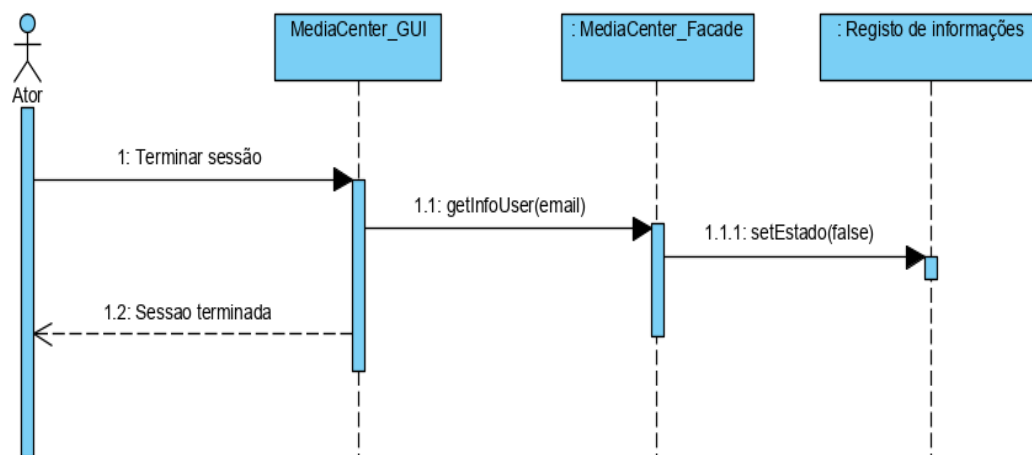


Figura 2.22: Terminar sessão

Editar utilizador

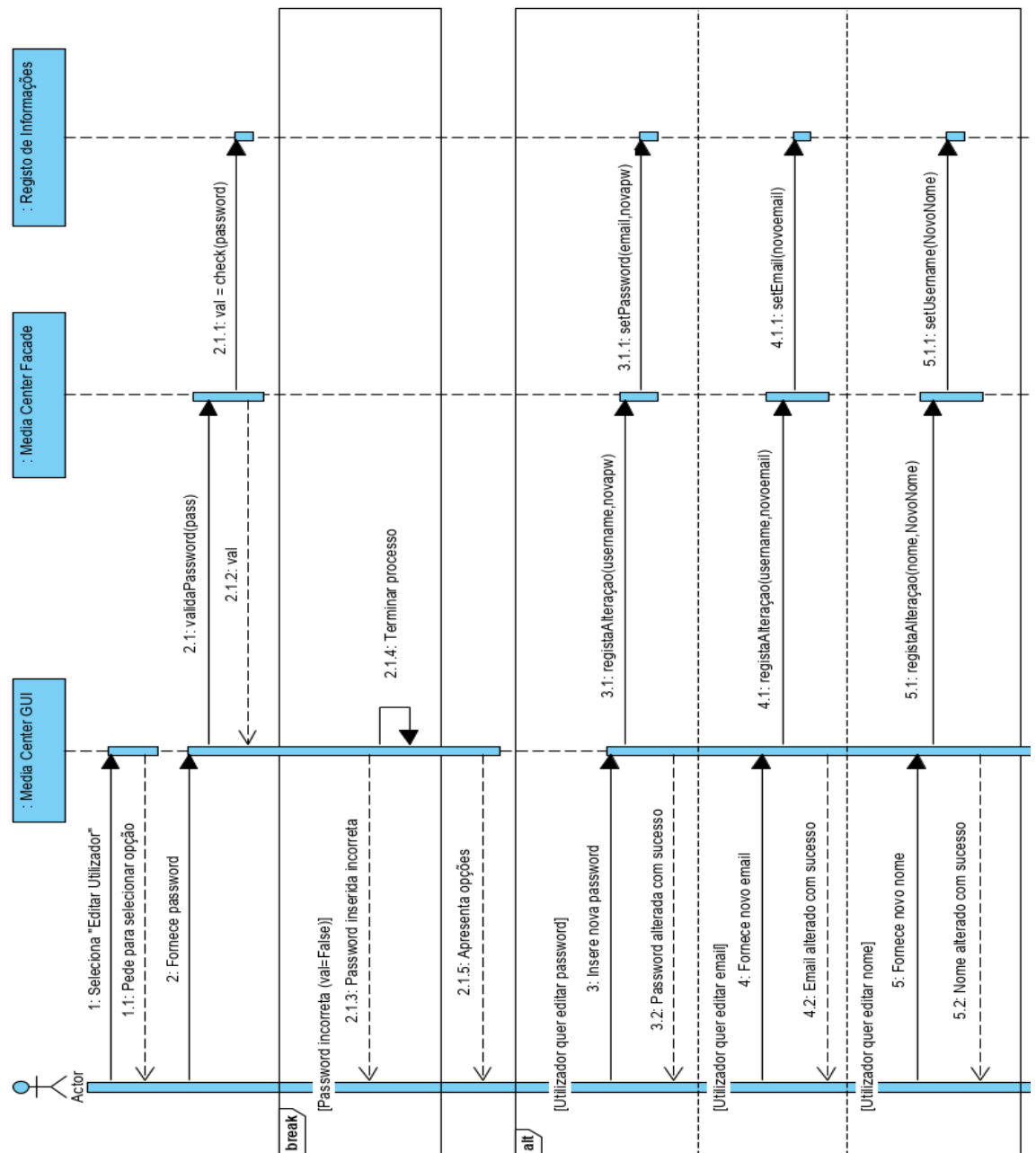


Figura 2.23: Editar utilizador

Eliminar utilizador

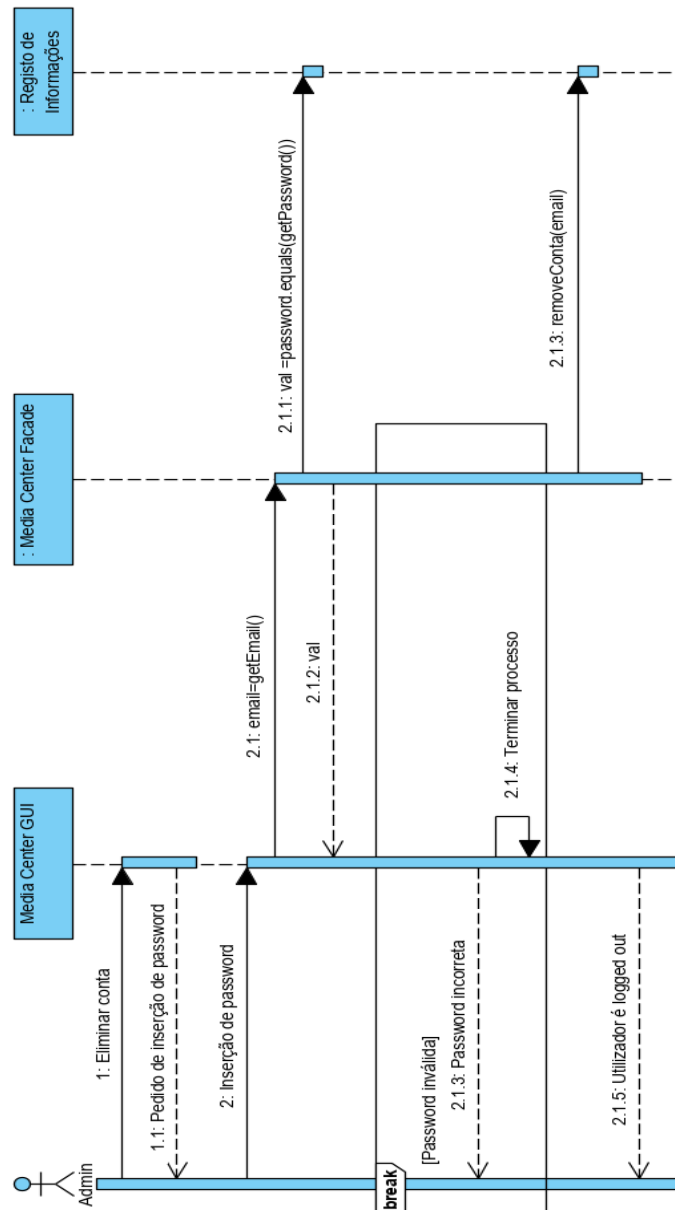


Figura 2.24: Eliminar utilizador

Fazer *upload* do conteúdo

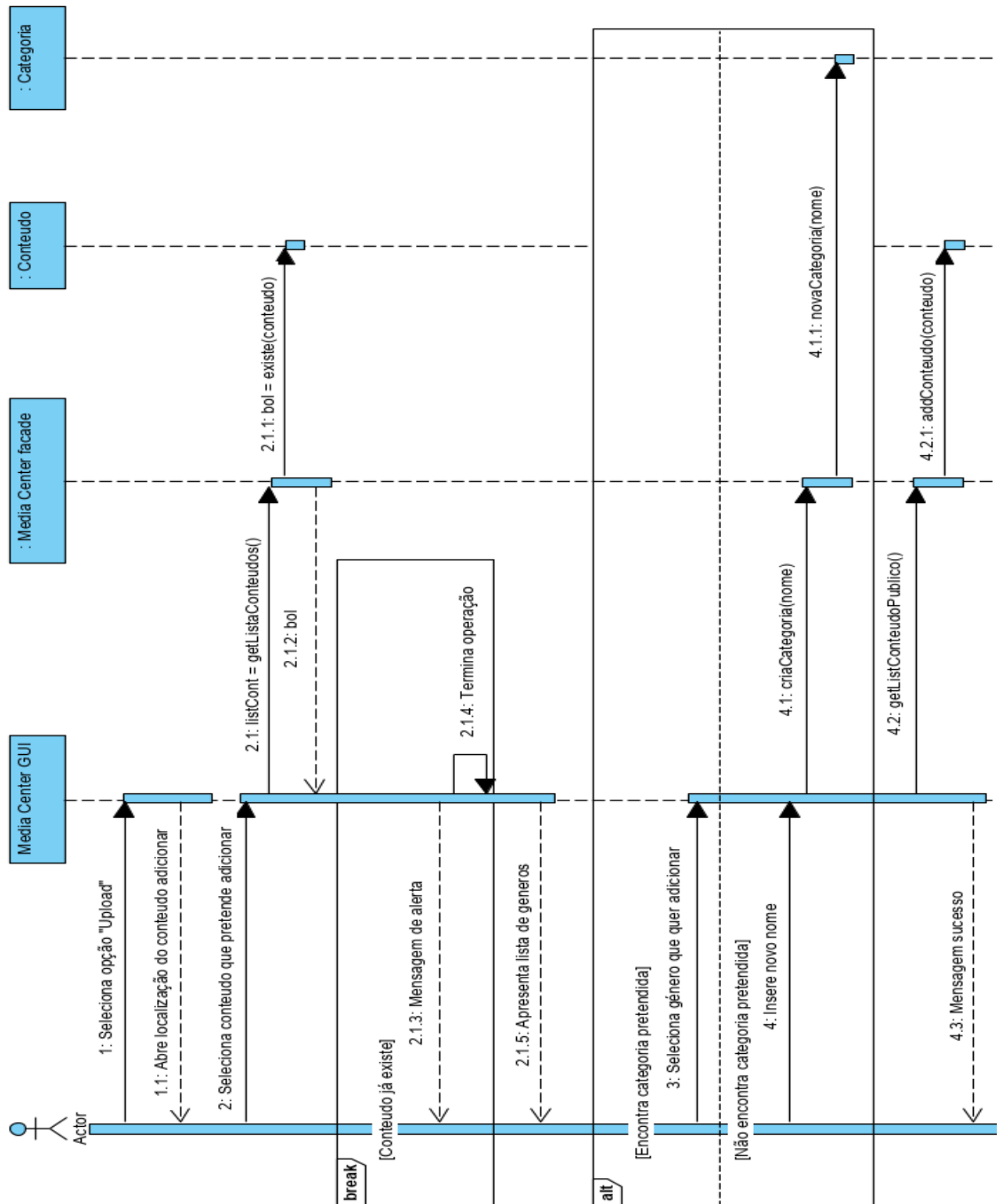


Figura 2.25: *Upload* de conteúdo

Alterar categoria do conteúdo

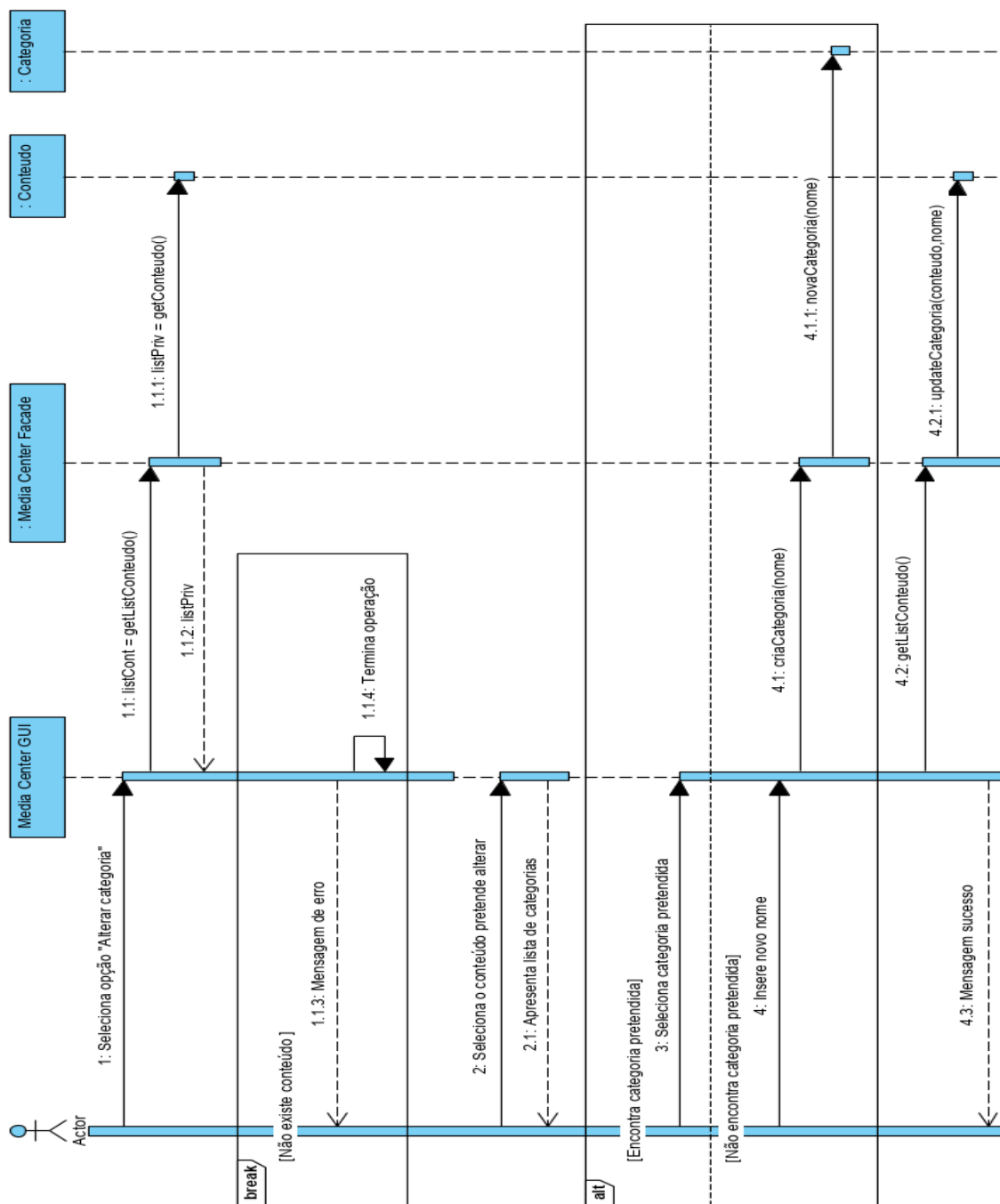


Figura 2.26: Alterar categoria

Remover conteúdo

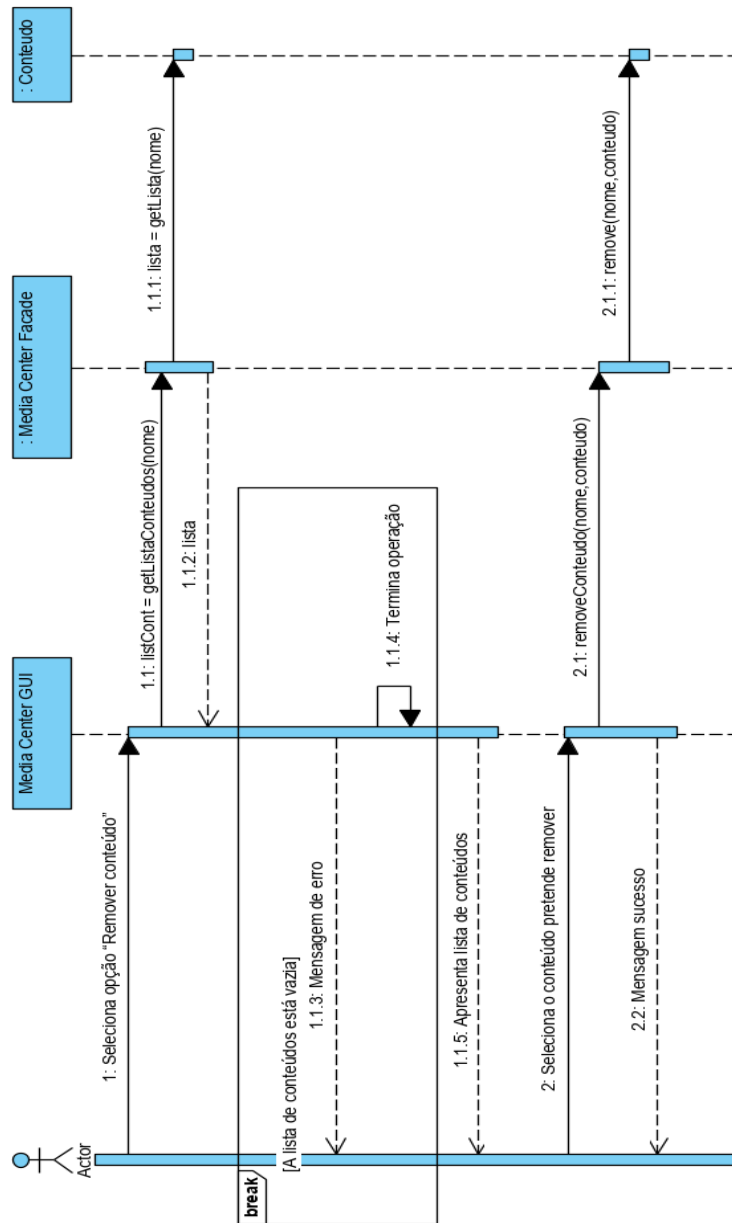


Figura 2.27: Remover conteúdo

Reproduzir conteúdo

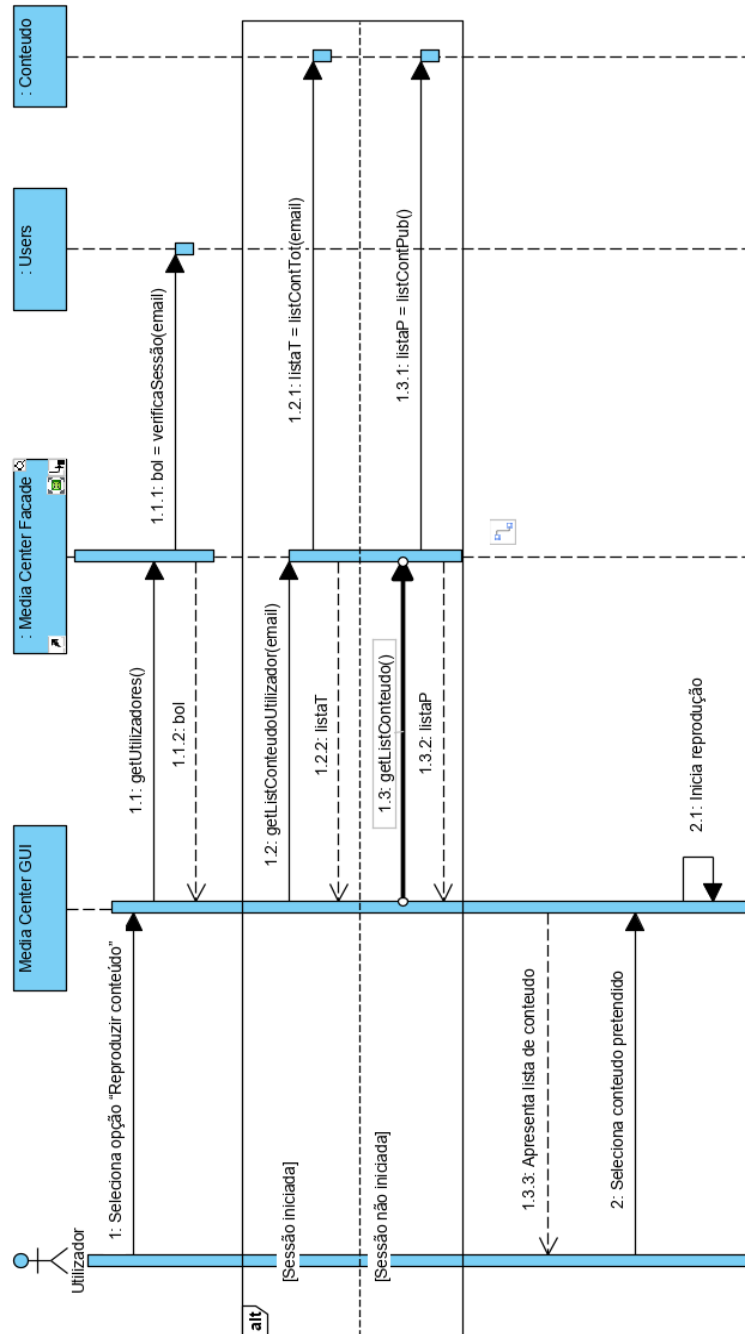


Figura 2.28: Reproduzir conteúdo

Criar *playlist*

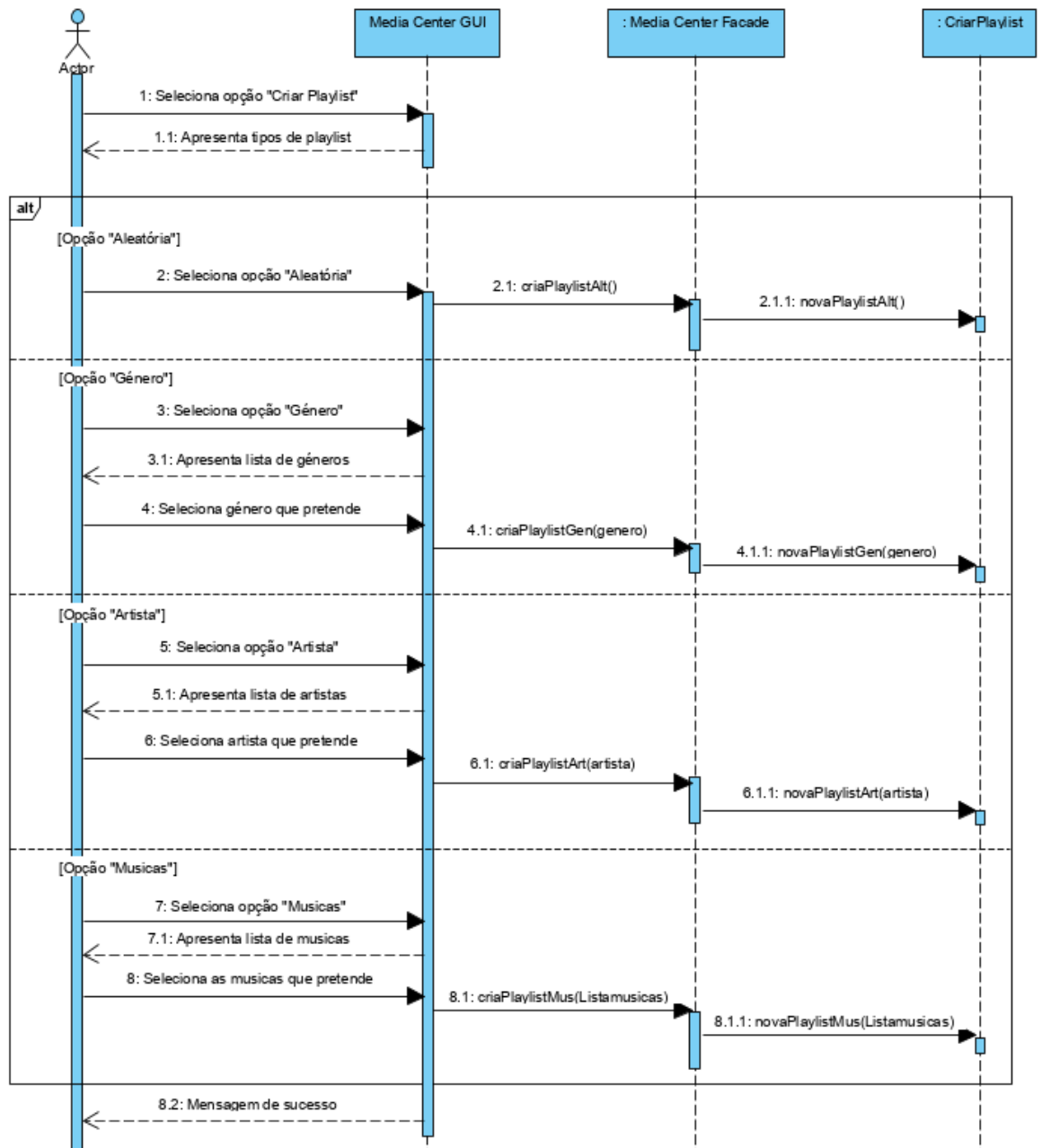
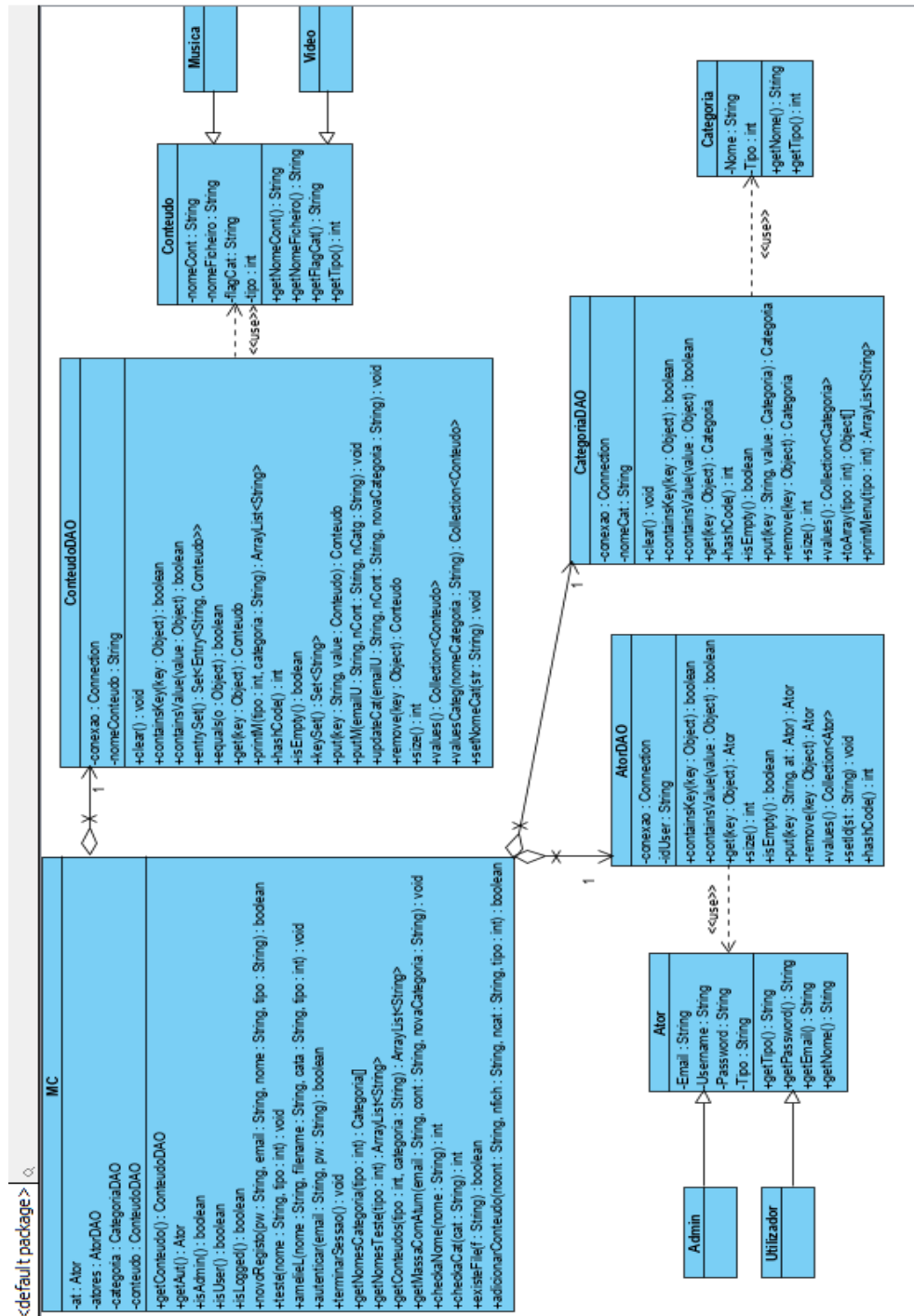


Figura 2.29: Criar *playlist*

Implementação

1 Diagrama de classe com DAOs



2 Diagramas de Sequência de Sistemas DAO

2.1 Iniciar Sessão

É relativo ao *Use Case* Iniciar Sessão e permite o acesso ao sistema.

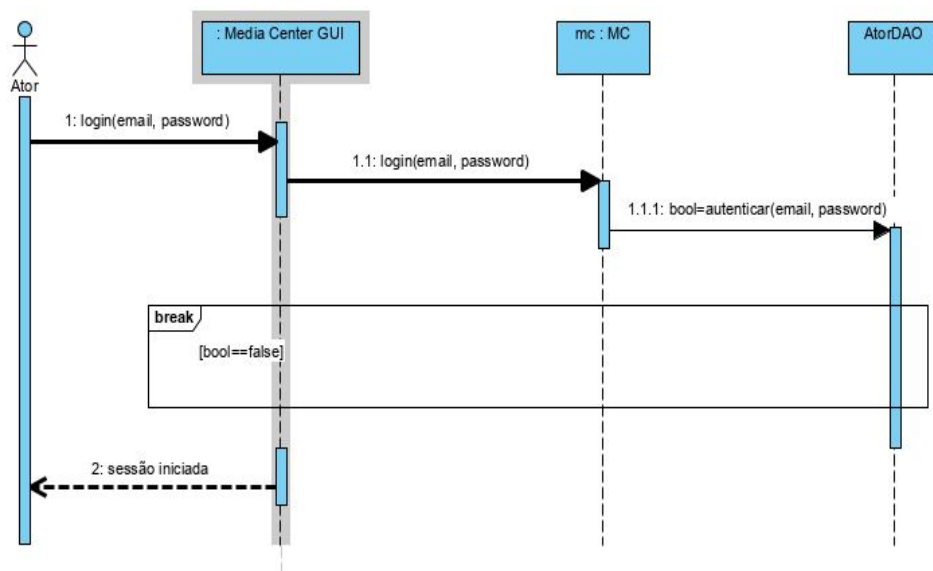


Figura 3.1: DSS-DAO Iniciar Sessão

2.2 Terminar Sessão

É relativo ao *Use Case* Terminar Sessão e fecha o acesso ao sistema.

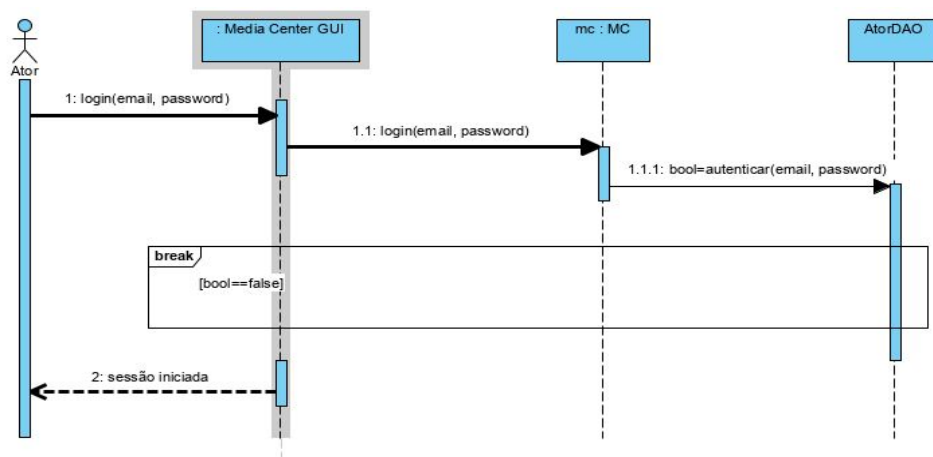


Figura 3.2: DSS-DAO Terminar Sessão

2.3 Alterar Categoria

É relativo ao *Use Case* Alterar Categoria e permitir mudar a categoria de determinado conteúdo.

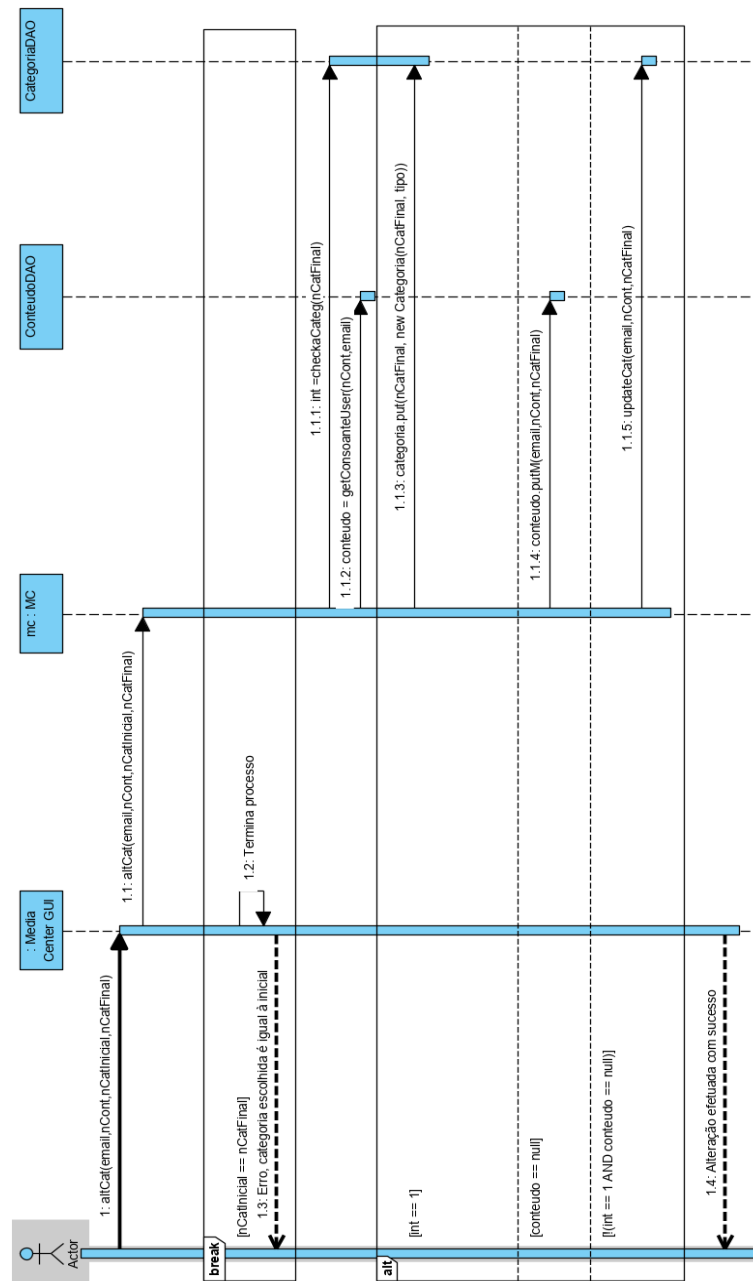


Figura 3.3: DSS-DAO Alterar Categoria

2.4 Upload de Conteúdo

Este diagrama é relativo ao *Use Case Upload* que permite a inserção de um ficheiro no sistema.

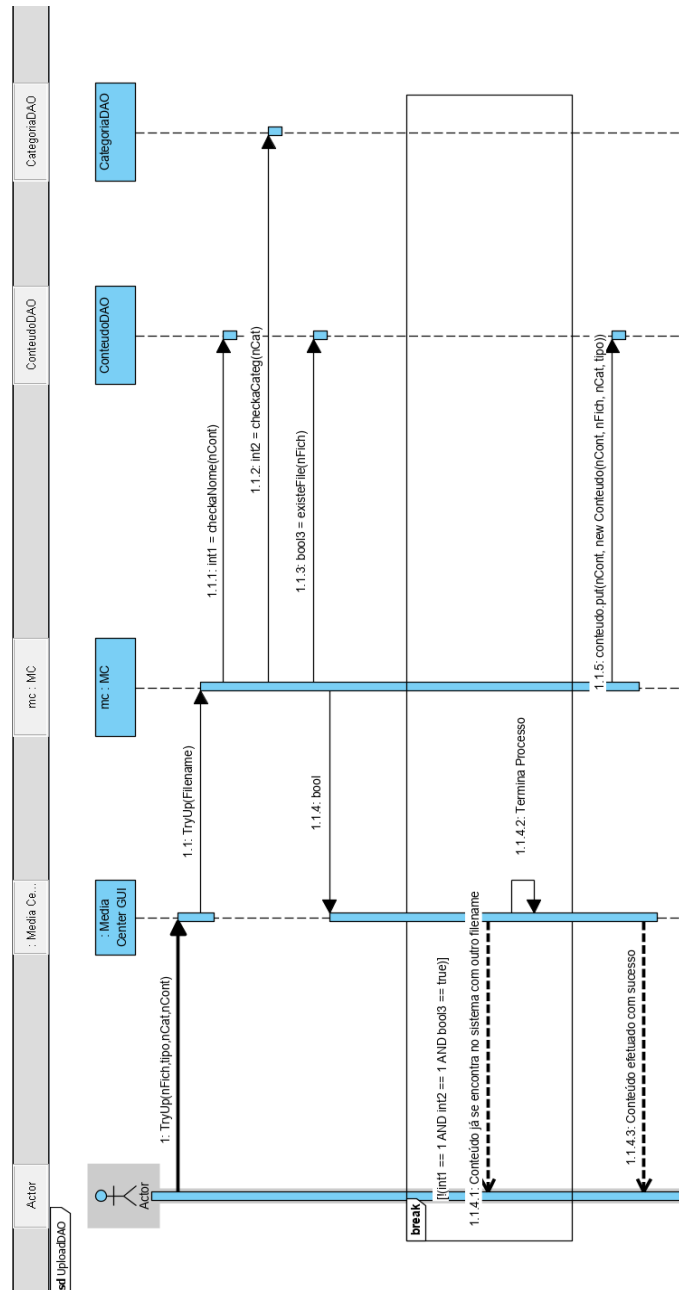


Figura 3.4: DSS-DAO *Upload*

2.5 Reproduzir Conteúdo

É relativo ao *Use Case* reproduzir conteúdo e é o método que reproduz o vídeo ou música selecionado.

sd Reproduzir conteúdo

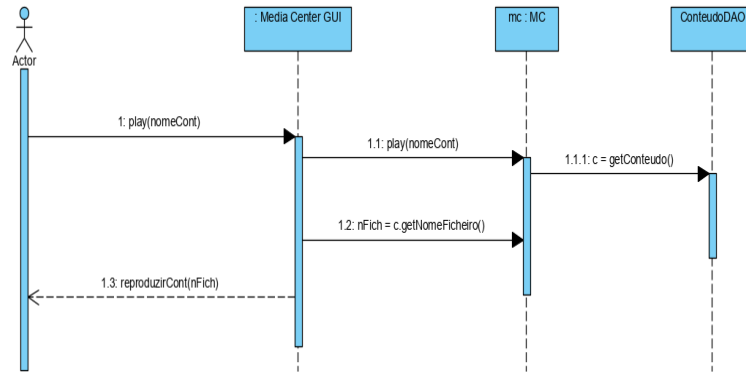
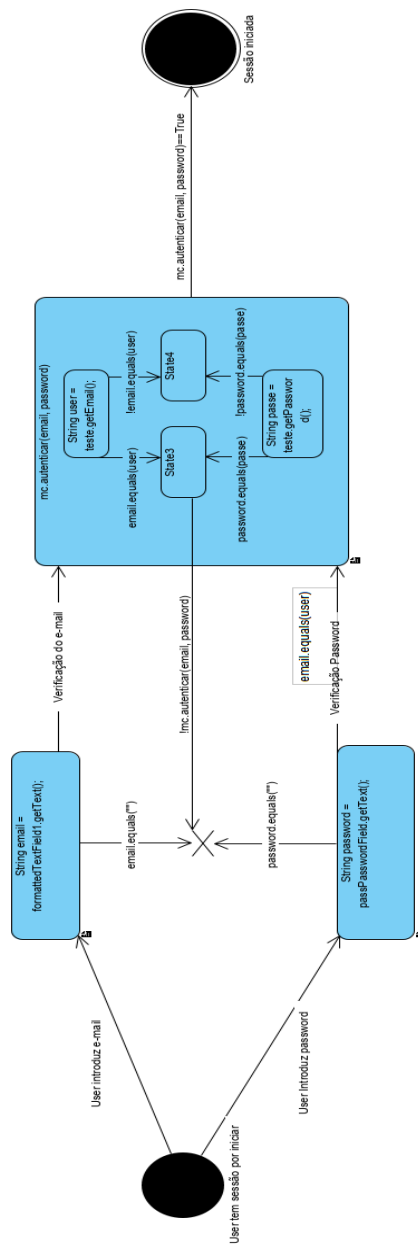


Figura 3.5: DSS-DAO Reproduzir Conteúdo

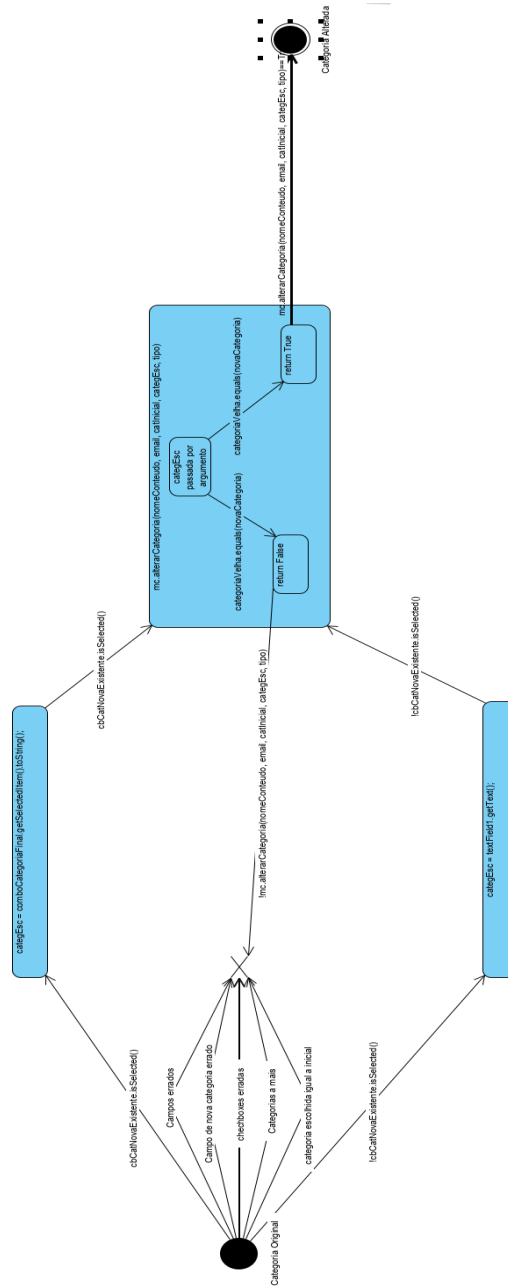
3 Máquinas de Estado

Os seguintes diagramas visualizar o comportamento do nosso sistema perante ou seja, de que forma este reage, de forma global, perante três *Use Cases* principais.

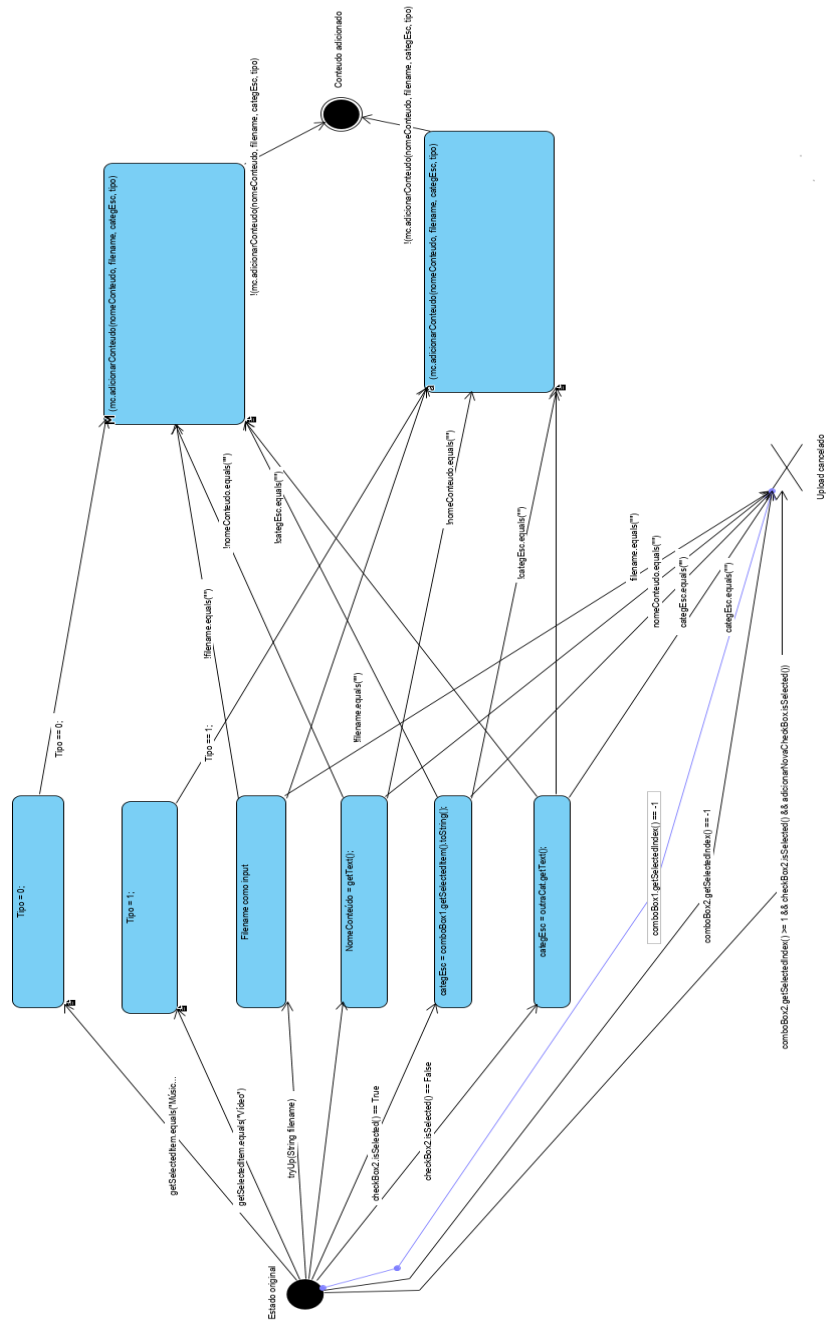
3.1 Início de Sessão



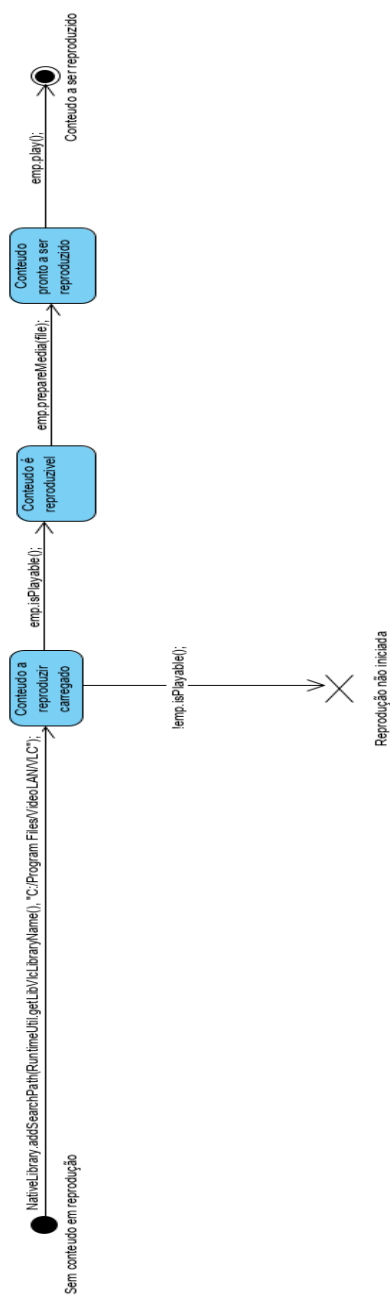
3.2 Alterar Categoria



3.3 Upload



3.4 Reproduzir Conteúdo



4 Diagrama de Instalação

Este diagrama serve o propósito de modular configuração do sistema desenvolvido.

Na nossa arquitetura, utilizamos dois nós que representam os "devices" usados (PC e Servidor da base de dados) e os ambientes de utilização (o Sistema Operativo (SO) e a Base de dados). Os ambientes de utilização permitem a execução de "software", pelo que o sistema que desenvolvemos é executado no SO de um PC, a partir de onde consegue aceder a uma base de dados.

Nesta são armazenados os dados necessários ao funcionamento do sistema. Estes dados são comunicados entre a base de dados e o sistema através de uma ligação TCP/IP.

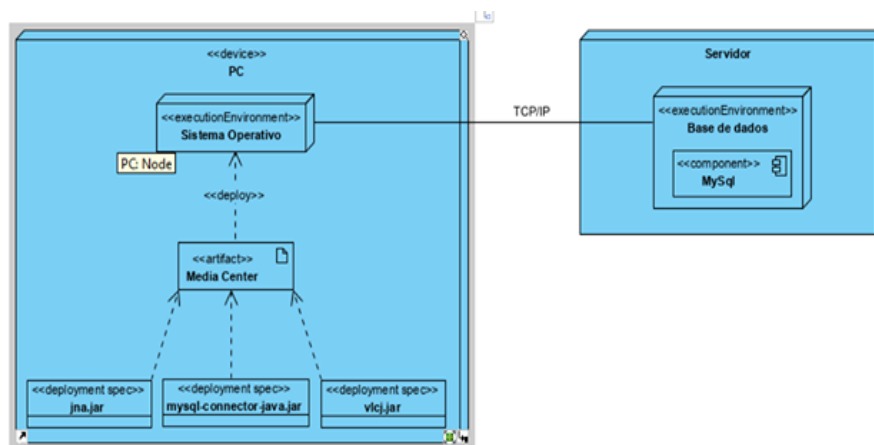


Figura 3.6: Diagrama de Instalação

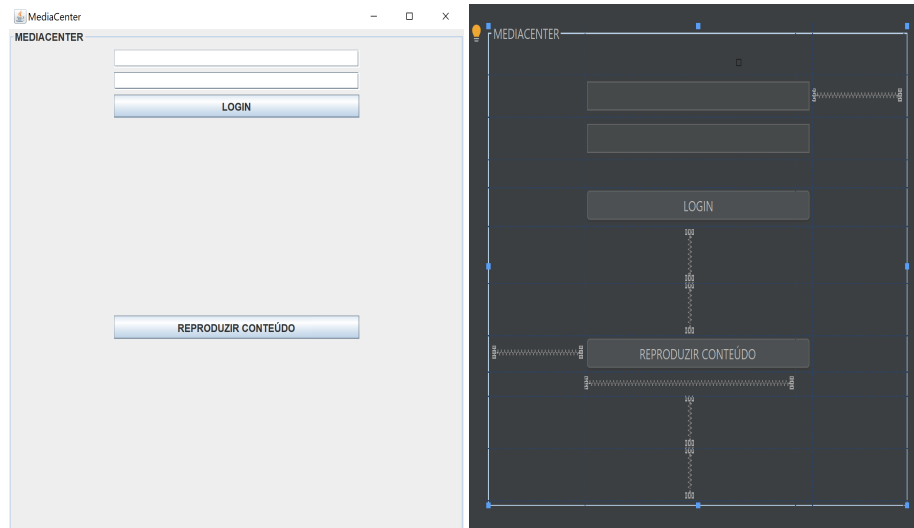
5 Descrição da interface

Para uma melhor descrição da implementação usada pelo grupo, iremos dividir a mesma pelos *Use Cases* pedidos pelo enunciado.

De realçar que para a construção dos Menus e consequente *Interface* da aplicação, bem como a interação entre o sistema e o utilizador, foi utilizada a ferramenta *GUI* presente no editor *IntelliJ* bem como a biblioteca *Swing*, pertencente à linguagem *Java*.

5.1 Iniciar Sessão

Após o início da aplicação, o primeiro menu que aparecer ao Ator é o de Iniciar Sessão podendo este, além de registar-se, reproduzir conteúdo na qualidade de convidado e ainda, se for o caso, introduzir os dados de início de sessão e usufruir do sistema.



Início de sessão como Administrador

Iniciando a sessão com dados correspondentes a um Administrador, então o sistema irá reagir apresentando o Menu correspondente, representado pelo seguinte:

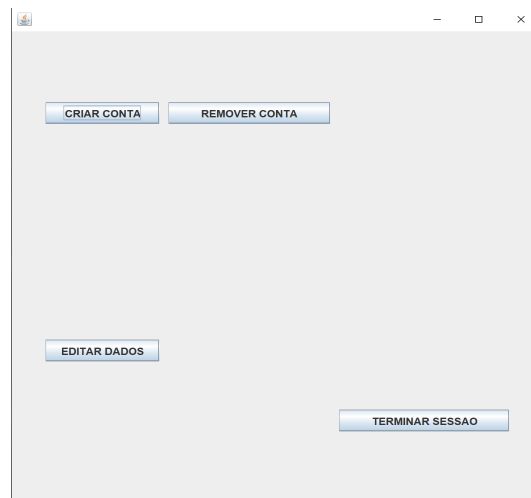


Figura 3.7: Menu do Administrador

Início de sessão como Utilizador

Iniciando a sessão com dados correspondentes a um determinado Utilizador, então o sistema irá reagir apresentando o Menu correspondente.

Cada Utilizador tem à sua disposição um leque de escolhas que se encontram explicitas na imagem seguinte:



Figura 3.8: Menu do Utilizador

5.2 Upload de conteúdo

Caso o Utilizador pretenda dar *Upload* de um conteúdo seu, então basta premir o respetivo botão para ser aberta a janela que irá permitir adicionar, então, o conteúdo ao sistema.

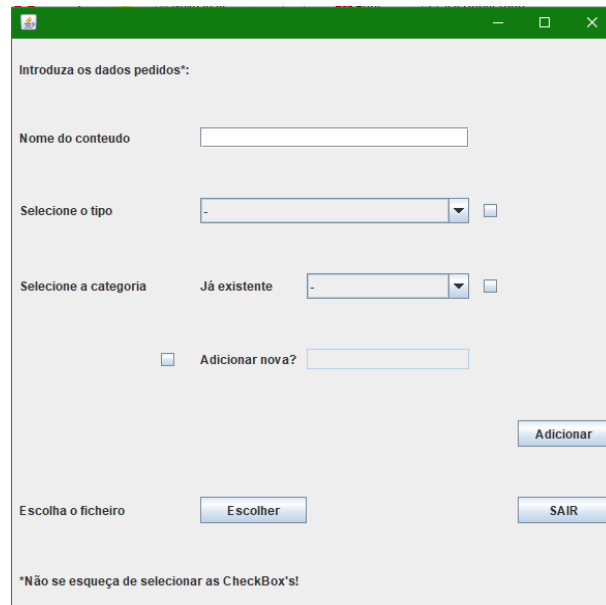


Figura 3.9: Menu do *Upload*

De realçar que se impôs algumas regras de modo a tornar a aplicação mais credível, entre elas:

- Caso já exista o nome do conteúdo fornecido, então o *upload* só será validado se o nome do ficheiro selecionado pelo Utilizador coincidir com o já presente na base de dados;
- É necessário selecionar apenas um tipo de conteúdo, seja ele Música ou Vídeo;
- É necessário atribuir ao conteúdo uma Categoria, seja ela já existente no sistema ou, então, adicionar uma nova, consoante a vontade;
- Não é possível adicionar um ficheiro que já conste na base de dados;
- Para uma maior eficácia é necessária a confirmação das ações, sendo que, para isso, é obrigatória a sinalização das várias *Checkbox's*.

5.3 Alterar Categoria

Outro dos *Use Cases* a ser implementado nesta fase corresponde ao de Alterar a Categoria de um dado Conteúdo.

Sendo esta uma aplicação aberta, então é possível a todo o Utilizador alterar qualquer Conteúdo para qualquer Categoria pelo que esta alteração é visível apenas para o próprio. Esta imposição permite que não haja problemas de apresentação para dois autenticados com perceções de Categoria diferentes.

O processo é simples, tendo apenas o *user* que seleccionar a Categoria atual do Conteúdo em questão e, de seguida, seleccionar a nova, podendo esta já existir ou até mesmo criar uma nova, a seu gosto.

Salienta-se que o sistema apenas veta esta alteração se for seleccionada a opção "Adicionar nova" e esta já estar no mesmo.

Selecione o tipo de conteúdo que pretende alterar

MUSICA VIDEO

Selecione a categoria que está atualmente atribuída ao conteúdo:

Guardar

Selecione o conteúdo que pretende alterar:

Guardar

Selecione a nova categoria:

Já existente: Guardar Confirmar

Adicionar nova SAIR

Figura 3.10: Menu para Alterar Categoria

5.4 Reproduzir Conteúdo

Os seguintes menus serão abertos selecionando a opção **Reproduzir Conteúdo**, tanto no menu inicial como no referente ao Utilizador.

A primeira janela a aparecer ao ator é a correspondente ao tipo de Conteúdo que pretende visualizar.



Figura 3.11: Menu para escolher tipo

Dependendo do tipo escolhido, a próxima janela a ser aberta contém todas as categorias existentes no sistema referentes ao tipo em questão. Realça-se que cada janela tem capacidade para apresentar até oito categorias e, para números superiores, implementou-se um sistema de páginas, a fim de facilitar o uso da *App*.

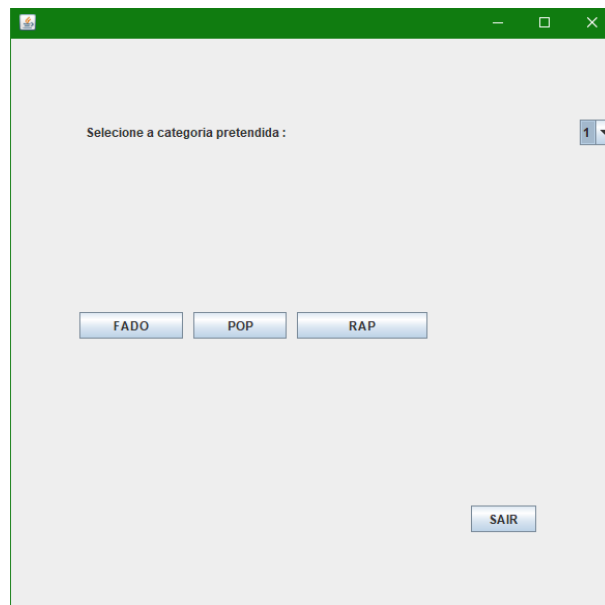


Figura 3.12: Menu para Escolher Categoria

De seguida, aquando a escolha da Categoria, será aberta a janela correspondente ao Conteúdo. Esta depende, mais uma vez, da categoria anteriormente escolhida e pode diferir não só de utilizador para convidado, mas também entre os próprios utilizadores.

Por exemplo, para o ator Convidado, o sistema assume as Categorias de modo *default*, ou seja, o Conteúdo terá associado a si a categoria dada através do primeiro *upload*.

Já no ator Utilizador a janela de conteúdos assumirá todas as alterações feitas por ele sendo que, para as que não efetuou alterações, então assume, também, as categorias *default*.

Quanto ao modo de apresentação, é utilizado o mesmo sistema aplicado às categorias. Na imagem seguinte é possível verificar o resultado após a seleção da Categoria **FADO**, presente na imagem anterior.



Figura 3.13: Menu para Escolher Conteúdo

Por fim, após premido o *PLAY* da Música/Vídeo pretendido, então abrirá uma nova janela, correspondente ao *Player*. Este possui uma nova interação, pois permite que o ator faça *Play* ou *Pause*.

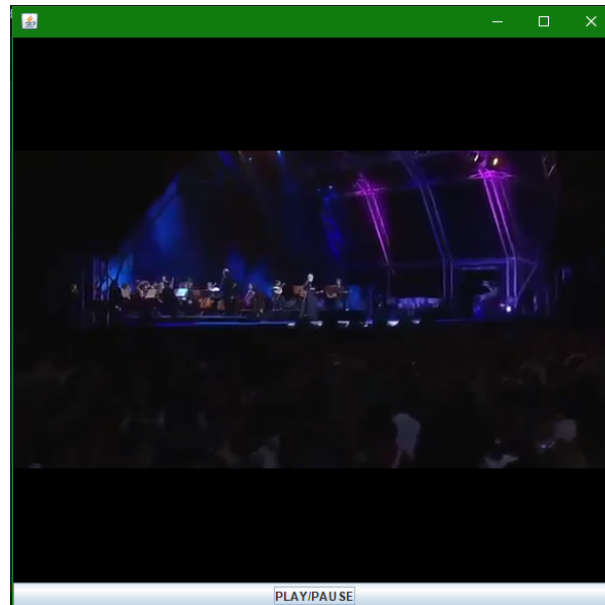


Figura 3.14: *Player*

6 Modelo da Base de Dados

Para uma eficiente gestão e manuseamento dos dados, foi utilizada uma Base de Dados relacional.

Começou-se por criar uma tabela para cada classe que precisasse de armazenar dados (Utilizador, Admin, Categoria e Conteúdo), processo o qual foi concretizado com facilidade através da consulta do Diagrama de Classes, previamente realizado. A Base de Dados foi implementada em *SQL* (usando o *MySQLWorkbench*) utilizando, para isso, um conector presente nas bibliotecas *Java*, nomeadamente o *mysql-conector-java.jar*. Realça-se que esta decisão foi tomada perante os conteúdos lecionados na aula teórica, no âmbito do tema *Java Database Connectivity*.

Por fim, com a base de dados criada, construiu-se com facilidade o Diagrama de Classes ORM, onde se substituiu os *Map's* por classes *DAO*, que têm como objetivo estabelecer a ligação à, então, base de dados.

Acrescenta-se, ainda, que a tabela *User_Conteudo* é fruto de uma relação **N para M** entre as entidades *Utilizador* e *Conteudo*.

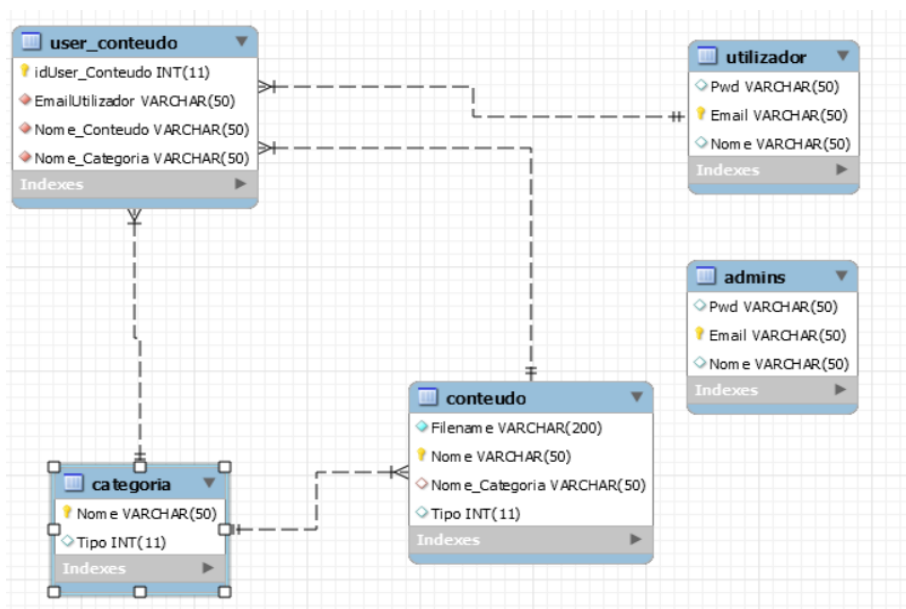


Figura 3.15: Modelo Relacional

Conclusões

Quanto à parte pedagógica, concluímos que este projeto foi muito enriquecedor para o nosso coletivo, pois este trabalho permitiu que melhorássemos as nossas competências a nível prático relacionadas com a UC de Desenvolvimento de Sistemas de Software.

Reconhecemos que o projeto foi feito com alguma dificuldade pois tentámos sempre garantir uma boa modelação para que esta facilitasse a implementação. Modelação essa que constituiu uma barreira extra, pois teria de albergar todos os casos do nosso programa e, por isso, era bastante complexa e incluía vários tipos de diagramas.

Quanto à construção da aplicação propriamente dita, fizemos um esforço para tentar seguir a modelação previamente entregue mas assumimos que em alguns casos tal não foi possível e, por isso, esta teve de ser refeita para haver uma coerência entre o idealizado e o efetivamente concretizado.

De salientar ainda o esforço do grupo para garantir que o resultado final fosse uma aplicação "aberta" e que, por exemplo, suportasse quantas Categorias/Conteúdos quantos o Utilizador quisesse adicionar, bem como o número de usufruidores do sistema. Optou-se, também, pela simplicidade em respeito à interação com o utilizador criando, assim, janelas de fácil interpretação.

Em suma, o esforço coletivo foi grande com o intuito de garantir boas soluções para o enunciado proposto, indo assim ao encontro dos objetivos definidos pelo grupo.