

Universidade do Minho

DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

MEDIA CENTER

Grupo 1



(a) A85227 João Azevedo



(b) A85729 Paulo Araújo



(c) A83719 Pedro Machado



(d) A89983 Paulo Lima

Braga, 18 de Dezembro de 2019

Conteúdo

1	Introdução	2
2	Fase 1	3
2.1	Análise de Requisitos	3
2.2	Modelação de Domínio	3
2.3	Diagrama de <i>Use Cases</i>	4
2.4	Especificação de Use Cases e <i>Mockups</i>	6
2.4.1	Autenticar um Utilizador [Iniciar Sessão] - <u>(Implementado)</u>	6
2.4.2	Criar novo Utente - <u>(Futura impl.)</u>	10
2.4.3	Remover Utente - <u>(Futura impl.)</u>	11
2.4.4	Upload de Conteúdo - <u>(Implementado)</u>	12
2.4.5	Download Conteúdo - <u>(Futura impl.)</u>	14
2.4.6	Criar Álbuns - <u>(Futura impl.)</u>	15
2.4.7	Eliminar Álbum - <u>(Futura impl.)</u>	17
2.4.8	Editar Álbuns - <u>(Futura impl.)</u>	18
2.4.9	Mover Conteúdo - <u>(Futura impl.)</u>	19
2.4.10	Reproduzir Conteúdo - <u>(Implementado)</u>	20
2.4.11	Sair do Sistema (Log-out) - <u>(Implementado)</u>	25
2.5	Outros <i>Mockups</i>	25
3	Fase 2	27
3.1	Seleção dos <i>Use Cases</i>	27
3.2	Modelação de interação	29
3.2.1	Diagramas de sequência	30
3.3	Modelação estrutural	30
3.3.1	Diagrama de Classe	30
3.3.2	Diagrama de Package	31
4	Fase 3	34
4.1	Síntese das Fases 1 e 2 e próximas decisões	34
4.2	<i>Data Access Objects</i> e persistência de dados	34
4.2.1	Base de dados em MySql	34
4.2.2	DAOs	35
4.3	Redefinição dos diagramas de Sequência	36
5	Modelação final	43
5.1	Diagrama de Use Case	43
5.2	Diagrama de Classes	44
5.3	Diagrama de Package	45
5.4	Packages e Classes	46
6	Conclusões e trabalho futuro	47

Capítulo 1

Introdução

Este relatório visa apresentar a elaboração e implementação de um sistema de *software* proposto como projeto final para a Unidade Curricular de Desenvolvimento de Sistemas de Software.

O Sistema proposto incide na implementação de um *Media Center* para partilha de música/vídeos (*media*) num apartamento guiado por um conjunto de cenários definidos pelo Cliente (enunciado). A análise dos cenários surgiu como ponto de partida para o desenvolvimento da aplicação.

Assim, ao longo deste documento, vão ser descritos os diferentes processos adotados e toda a modelação produzida nas diferentes fases do trabalho.

Por fim, o objetivo principal deste projeto é mostrar a importância de uma análise faseada, independente e por camadas do sistema através de todas as ferramentas UML disponibilizadas para a construção de uma aplicação sustentável, estruturada e consistente, algo muito importante num processo de desenvolvimento de *software*.

Capítulo 2

Fase 1

2.1 Análise de Requisitos

Numa primeira fase, analisamos os cenários propostos e, a partir destes, identificamos os vários requisitos necessários para a aplicação ‘Media Center’ cumprir os cenários indicados.

Para além dos requisitos identificados apenas pelos cenários descritos, pudemos adicionar outros que se mostraram interessantes e adequados para o funcionamento da aplicação.

Vejamos então, alguns dos cenários propostos e a sua posterior análise:

- ”(...) poderia fazer upload da sua **música** e **vídeos**, que passariam a estar disponíveis para todos os residentes (...);”;
- ”Cada um só pode fazer **download** das suas próprias músicas e vídeos (...);”;
- ”A Isabel, pediu à Paula, que **tinha permissões de administração**, que criasse a **conta** do Manuel no media center. Com os dados fornecidos pelo Manuel (**nome** e **e-mail**) a Paula criou um novo **utilizador** e o Manuel definiu a sua **palavra-passe**.“
- ”Mesmo assim, o conteúdo ficou disponível nas **suas coleções** e todos os residentes com quem ele tinha conteúdo em comum foram adicionados à sua lista de **potenciais amigos**.“
- ”O Manuel notou que o sistema gerara automaticamente um conjunto de **categorias para organizar o conteúdo** (...)"
- ”Um dos novos residentes tinha adicionado dois ou três **álbuns** classificados como Jazz (...)"
- ”o Ricardo fez logout do media center e o Rui acedeu como **convidado**.“

Os cenários acima apresentados descrevem um conjunto de entidades que são necessárias para estabelecer todas as interações e toda a modelação comportamental da aplicação. Entidades como **Música**, **Vídeo**, **Utilizador**, **Utente**, **Administrador**, **Convidado**, **Credenciais**, **Conteúdo**, **Categoria**, **Álbuns**, entre outras, foram consideradas na modelação.

2.2 Modelação de Domínio

A partir das mesmas, conseguimos estabelecer as suas relações que se encontram descritas no diagrama UML que descreve o seguinte **Modelo de Domínio**:

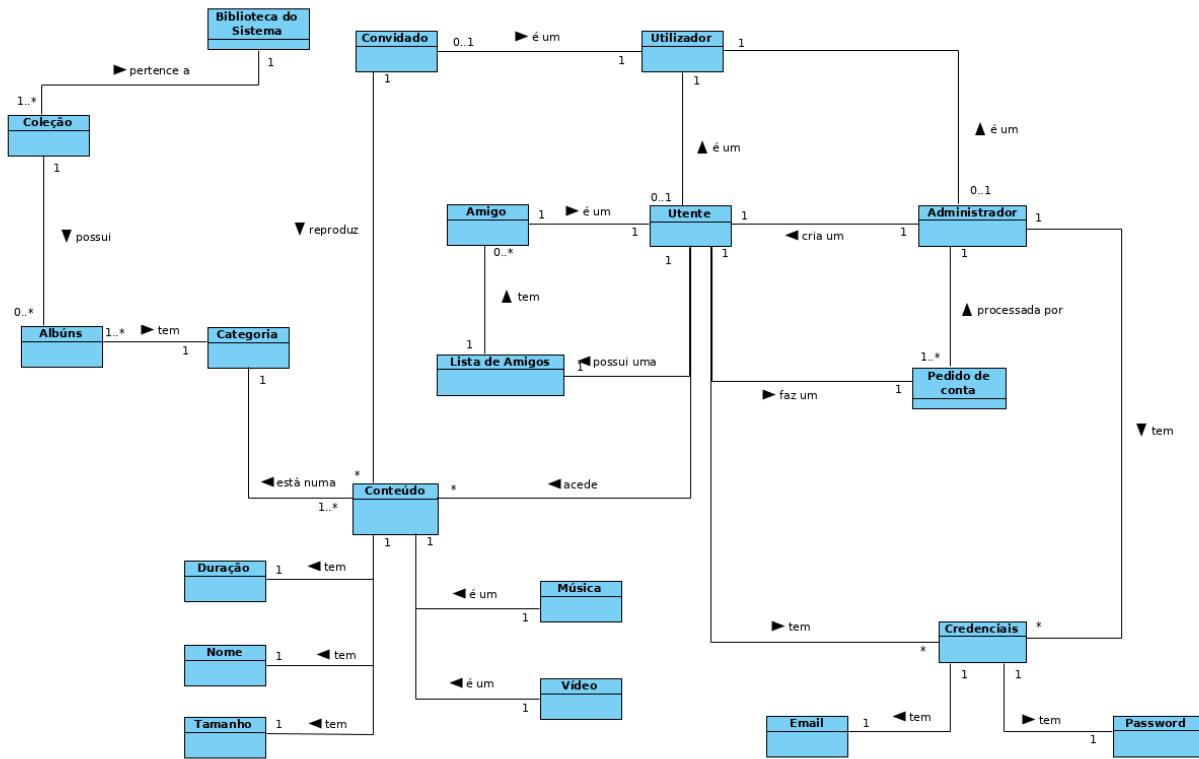


Figura 2.1: Modelo de domínio.

Este modelo foi estabelecido, numa primeira fase, como o modelo base para as fases seguintes. No entanto, algumas das entidades consideradas aqui não foram repercutidas para a fase final do projeto, entre elas, a **Duração**, **Tamanho** e **Pedido de Conta** que não se revelaram importantes para suportar a modelação da aplicação mas considera-se que, numa futura aplicação, possam vir a ser necessárias para estabelecer uma estrutura mais robusta deste sistema.

2.3 Diagrama de *Use Cases*

A partir dos cenários definimos um conjunto de funcionalidades para descrever como os diferentes **Atores** atingem os seus objetivos no sistema. Os *Use Cases* foram definidos a partir dos cenários apresentados, seguem-se alguns exemplos:

- ”(...) só pode fazer download das suas próprias músicas e vídeos (...)”
 - * **O Ator Utente** pode fazer download de media (da media que está na sua conta).
 - ”(...) o Manuel descarregou a sua biblioteca de media para um disco externo (...)”
 - * **O Ator Utente** pode fazer download para um disco;
 - ”A Isabel, pediu à Paula, que tinha permissões de administração, que criasse a conta do Manuel no media center. (...)”
 - * **O Ator Administrador** cria Utentes (é o único com esta funcionalidade);
 - ”O sistema detectou que algum do conteúdo já estava presente, não o tendo carregado para evitar duplicações (...)”, ”Mesmo assim, o conteúdo ficou disponível nas suas colecções (...)”
 - * **O Sistema** gere automaticamente conteúdo repetido e não o carrega.
 - ”(...) Colocou-o (o Media Center) a tocar Jazz em modo aleatório”
 - O Ator Utente** colocou a música. Possivelmente, terá um modo aleatório;

Estes exemplos constituem alguns dos exemplos de funcionalidades comportamentais extraídas dos cenários apresentados pelo Cliente (enunciado).

Assim, apresenta-se a primeira abordagem da modelação de **Use Cases**:

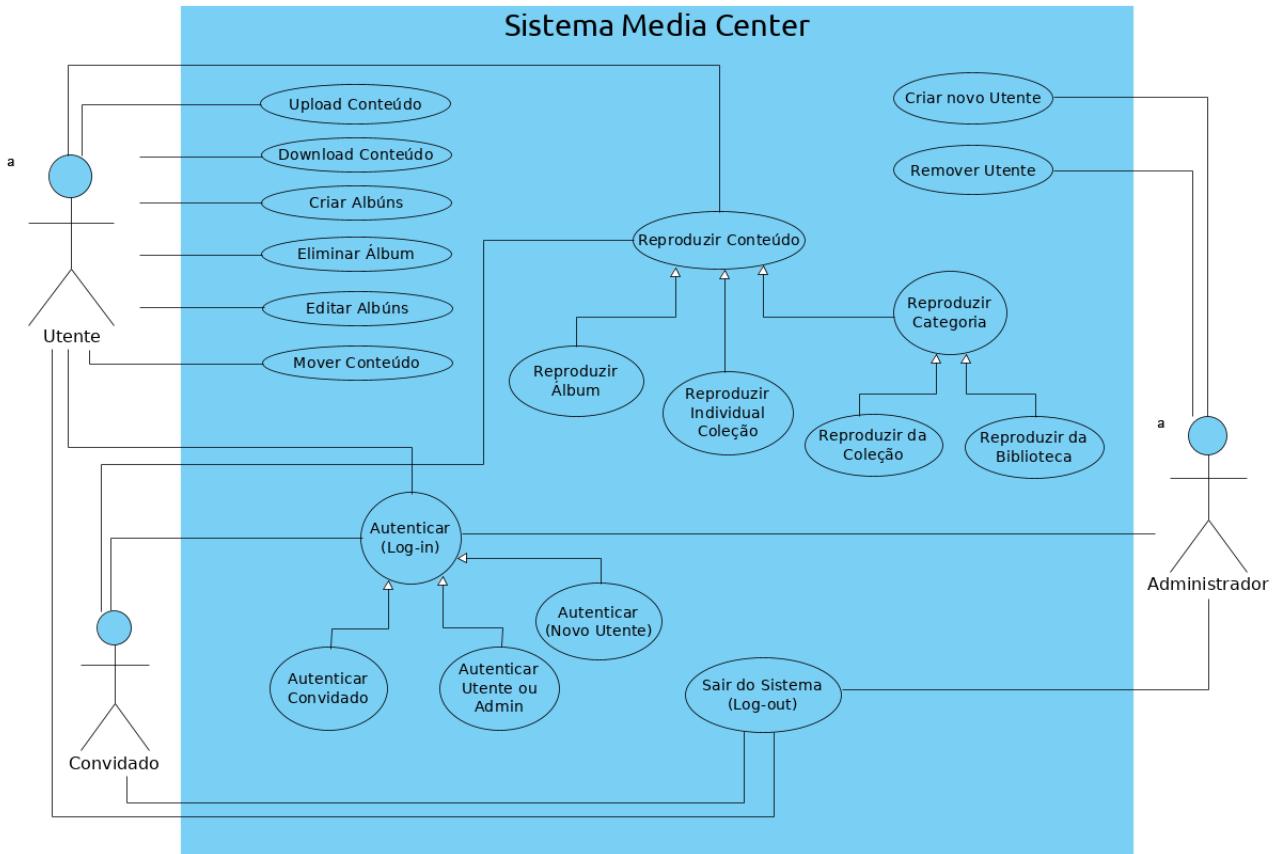


Figura 2.2: Diagrama de Use Cases.

Neste modelo casos de uso, conseguimos dividir o sistema em **3 Atores** e **16 Use Cases** iniciais. A divisão dos Use Cases passa por:

- **Reprodução:** Reproduzir Álbum, Reproduzir Individual Coleção, ...;
- **Autenticação:** Autenticar Convidado, Autenticar Utente ou Admin e Autenticar Novo Utente;
- **Conteúdo:** Upload, Dowload, Criar Álbuns...;
- **Registo:** Registar Utentes e remover Utentes;
- **Outros:** Sair do Sistema (Log-out).

Agora, tendo estabelecido os Use Cases a implementar, seguem-se aqueles que foram desenvolvidos na primeira fase. Junto aos mesmos, e para facilitar a visualização, têm associados um "(Implementado)" caso tenham sido implementados. De realçar que alguns erros cometidos inicialmente na conceção dos mesmos foram corrigidos.

2.4 Especificação de Use Cases e *Mockups*

2.4.1 Autenticar um Utilizador [Iniciar Sessão] - (Implementado)

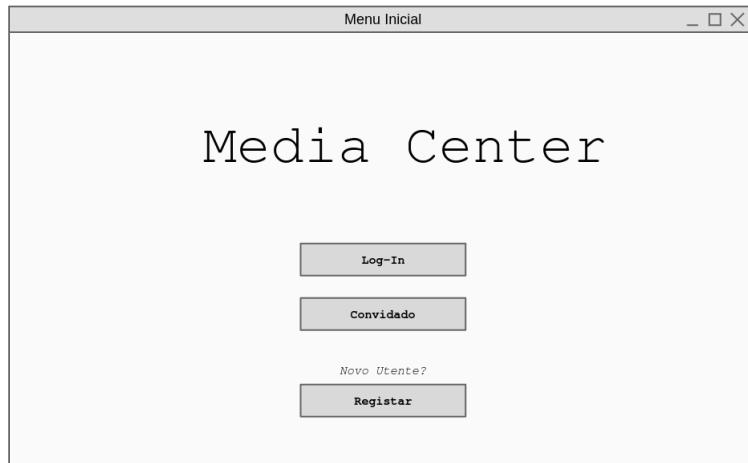


Figura 2.3: Interface de Log-in para Utilizadores.

Autenticar Utente ou Administrador - (Implementado)

- **Descrição:**

Autenticar um Utente ou Administrador no Sistema.

- **Cenários:**

”O Ricardo chegou a casa e ligou o media center.”

Cenário geral: Um Utente ou Administrador pretende usar o Media Center e terá, necessariamente, de se autenticar como tal.

- **Pré-Condição:**

O Utilizador tem de estar registado no Sistema e não pode haver outro Utilizador no Sistema.

- **Pós-Condição:**

O Utilizador fica autenticado no Sistema como Utente ou Administrador.

- **Fluxo Normal:**

1. O Sistema apresenta as opções de Autenticação;
2. O Utilizador escolhe a opção de entrar como Utente ou Administrador e insere as credenciais;
3. O Sistema valida as credenciais inseridas;
4. O Utilizador autentica-se no Sistema;

-
- **Fluxo de exceção 1:** [As credenciais são inválidas] (passo 3)

- 3.1. O Sistema avisa o Utilizador que as credenciais são inválidas;
- 3.2. O Utilizador sai do Sistema.

Menu de Autenticação

Log-In

E-Mail:

Password:

Figura 2.4: Interface de Log-in para Utente ou Administrador.

Menu de Autenticação

Credenciais inválidas! (Falha na autenticação)

⚠️ O E-mail inserido não se encontra registado ou inseriu incorretamente a password.

E-Mail:

Password:

Figura 2.5: Interface de Log-in para Utente ou Administrador (**Fluxo de Excepção 1**).

Autenticar Convidado - (Não considerado na fase final - desnecessário)

- **Descrição:**

Autenticar um Convidado no Sistema.

- **Cenários:**

”(...) o Rui acedeu como convidado.”

- **Pré-Condição:**

Não haver outro Utilizador no Sistema.

- **Pós-Condição:**

O Utilizador fica autenticado no Sistema como Convidado.

.....

- **Fluxo Normal:**

1. O Sistema apresenta as opções de Autenticação;
2. O Utilizador seleciona a opção de entrar como convidado;
3. O Utilizador entra no Sistema;

Autenticar Novo Utente - (Futura impl.)

- **Descrição:**

Autenticar um Utente temporário no Sistema.

- **Cenários:**

”Com os dados fornecidos pelo Manuel (nome e email) a Paula criou um novo utilizador e o Manuel definiu a sua palavra passe.”

- **Pré-Condição:**

O Novo Utente tem de estar registado no Sistema como Utente temporário pelo Administrador.

- **Pós-Condição:**

O Utilizador fica autenticado e registado no Sistema.

.....

- **Fluxo Normal:**

1. O Sistema apresenta as opções de Autenticação;
 2. O Utilizador seleciona a opção de entrar como novo Utente;
 3. O Sistema apresenta um menu de inserção das suas credenciais e definição de password;
 4. O Utilizador insere as credenciais e define a password;
 5. O Sistema regista e autentica o novo Utente;
-

- **Fluxo de exceção 1:** [O E-mail não está registado ou a password não foi corretamente inserida] (passo 5)
 - 5.1. O Sistema avisa o Utilizador que as credenciais são inválidas;
 - 5.2. O Utilizador sai do Sistema.

The screenshot shows a window titled "Novo Utente". Inside, there is a form with the following fields:
 - A text input field labeled "E-Mail:"
 - A text input field labeled "Defina a sua password:"
 - A text input field labeled "Repita a password:"
 - Two buttons at the bottom: "Cancelar" and "Registrar".

Figura 2.6: Interface para registar um novo Utente.

The screenshot shows a window titled "Novo Utente". Inside, there is a form with an error message box and a red-bordered password field:
 - An error message box titled "Credenciais inválidas! (Falha na autenticação)" containing the text "O E-mail inserido não se encontra registado ou inseriu incorretamente a password."
 - A red-bordered text input field labeled "Repita a password:".
 - Two buttons at the bottom: "Cancelar" and "Registrar".

Figura 2.7: Interface para registar um novo Utente (**Fluxo de Excepção 1**).

2.4.2 Criar novo Utente - (Futura impl.)

- **Descrição:**

Um novo Utente é registado no sistema pelo Administrador.

- **Cenários:**

A Isabel, pediu à Paula, que tinha permissões de administração, que criasse a conta do Manuel no media center.

- **Pré-Condição:**

O Administrador tem de estar autenticado no Sistema.

- **Pós-Condição:**

O novo Utente é adicionado ao Sistema.

.....

- **Fluxo Normal:**

1. O Administrador tem um pedido pendente para criar um novo Utente, juntamente com os dados (Nome e E-Mail), aceita esse pedido e insere os dados de registo;
 2. O Sistema valida os dados e regista um novo Utente temporário;
 3. O novo Utente define a sua password;
 4. O Sistema regista um novo Utente efetivo (registado no Sistema);
-

- **Fluxo de excepção 1:** [O Nome ou E-mail não são válidos (já existe esse E-Mail)] (passo 2)

- 2.1. Os dados fornecidos ao Administrador são inválidos;
- 2.2. O Administrador não cria a nova conta;

The screenshot shows a window titled "Criar Utente - Administrador". The main title is "Criar Utente". Below it, there are two input fields: one labeled "Nome:" and another labeled "E-Mail:". Both fields have empty text boxes. At the bottom of the window are two buttons: "Cancelar" (Cancel) on the left and "Criar" (Create) on the right.

Figura 2.8: Interface para criar um novo Utente.

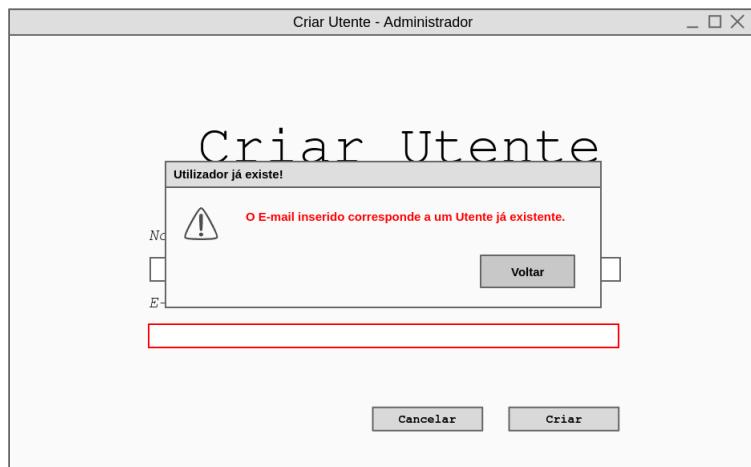


Figura 2.9: Interface para criar um novo Utente (**Fluxo de Excepção 1**).

2.4.3 Remover Utente - (Futura impl.)

- **Descrição:**

Um Utente é removido do sistema pelo Administrador.

- **Cenários:**

Cenário Geral: Para facilitar a gestão de Utentes no Sistema, o Administrador tem a possibilidade de remover Utentes;

- **Pré-Condição:**

O Administrador tem de estar autenticado no Sistema.

- **Pós-Condição:**

Um Utente é removido do Sistema.

- ***Fluxo Normal:***

1. O Administrador escolhe a opção de "Remover Utente";
2. O Sistema apresenta uma lista de todos os Utentes registados;
3. O Administrador seleciona o(s) Utente(s) que pretende remover;
4. O Sistema remove o(s) Utente(s) da lista de Utentes do Media Center e todo o conteúdo que seja exclusivamente do(s) Utente(s) selecionado(s);

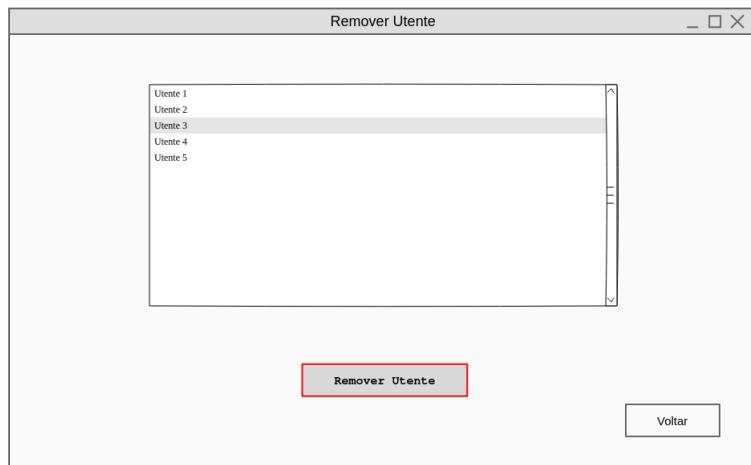


Figura 2.10: Interface para remover um Utente.

2.4.4 Upload de Conteúdo - (Implementado)

- **Descrição:**

Carregar Conteúdo do disco para o Sistema originando um novo Álbum.

- **Cenários:**

”(...) poderia fazer upload da sua música e vídeos, que passariam a estar disponíveis para todos os residentes (...)”

”(...) utilizou-o para fazer upload do conteúdo para a sua conta no media center (...)”

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente.

- **Pós-Condição:**

O novo Álbum é adicionado ao sistema.

- ***Fluxo Normal:***

1. O Utilizador seleciona a opção ”Upload Conteúdo”;
2. Sistema pergunta qual o ”Path”(caminho) virtual para os ficheiros a inserir;
3. O Utilizador indica o caminho virtual para os ficheiros;
4. O Sistema adiciona todos os ficheiros indicados num Álbum, mas apenas se cumprirem os requisitos (o tipo de conteúdo (media) inserido, Ex: .mp3 ou .mp4 etc...);
5. O Sistema automaticamente verifica conteúdo duplicado adicionando apenas para o Media Center aquele que é novo. No entanto as músicas ficam disponíveis na conta pessoal do Utente;
6. O Sistema automaticamente adiciona o nome ao Álbum (nome do primeiro conteúdo inserido) e categoriza o conteúdo adicionado (com base na informação já existente no sistema) e adiciona potenciais amigos ao Utente, sendo estes, outros Utentes com conteúdo em comum;

- **Fluxo de exceção 1:** [Caminho para ficheiros inválido ou caminho sem conteúdo disponível] (passo 4)

4.1. O Sistema deteta um caminho inválido avisando o Utilizador e não permite o Upload de Conteúdo, saindo deste Use Case.

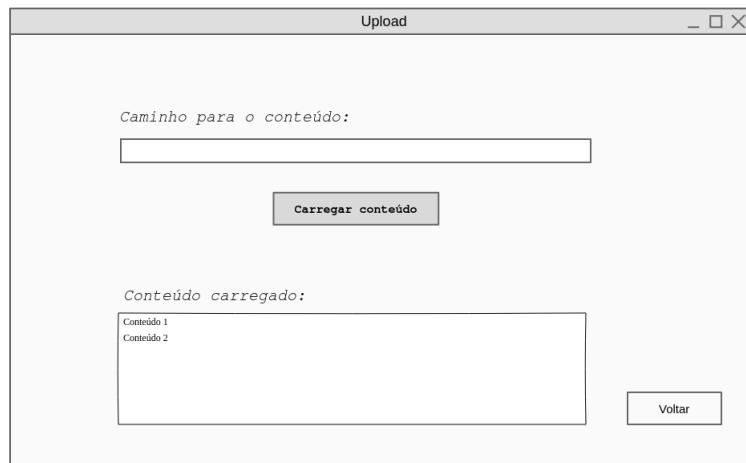


Figura 2.11: Interface para carregar conteúdo no Sistema.

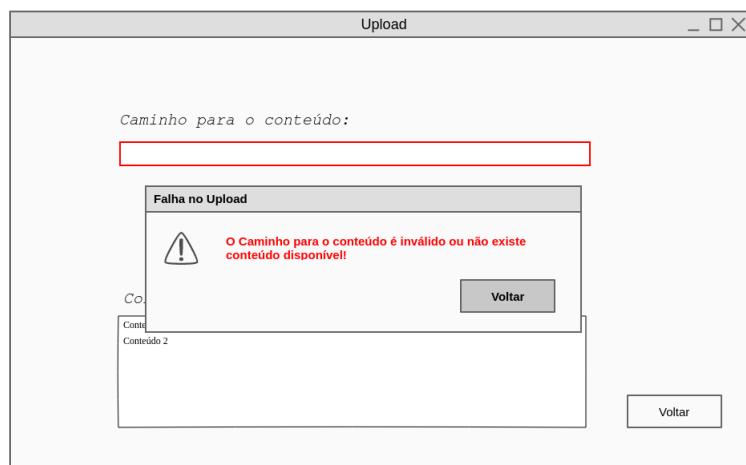


Figura 2.12: Interface para carregar conteúdo no Sistema (**Fluxo de Excepção 1**).

2.4.5 Download Conteúdo - (Futura impl.)

- **Descrição:**

Descarregar Conteúdo do Sistema para o disco.

- **Cenários:**

”(...) só pode fazer download das suas próprias músicas e vídeos (...)"

”(...) o Manuel descarregou a sua biblioteca de media para um disco externo (...)"

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente e tem de ter conteúdo na sua conta pessoal.

- **Pós-Condição:**

O Conteúdo é descarregado do sistema para o seu disco.

.....

- **Fluxo Normal:**

1. O Utilizador seleciona a opção ”Download Conteúdo”;
 2. O Sistema apresenta as opções de *download* (Álbum ou Coleção);
 3. O Utilizador seleciona a opção de transferir o Álbum;
 4. O Sistema apresenta os álbuns disponíveis;
 5. O Utilizador seleciona o Álbum que pretende transferir;
 6. O Sistema pergunta qual o ”Path”(caminho) virtual para guardar os ficheiros;
 7. O Utilizador indica o caminho virtual para os ficheiros;
 8. O Sistema descarrega todos os ficheiros indicados;
-

- **Fluxo Alternativo 1:** [O Utilizador seleciona a opção de transferir a coleção] (passo 3)

- 3.1. O Utilizador seleciona a opção de transferir a coleção;
 - 3.2. O Sistema pergunta qual o ”Path”(caminho) virtual para guardar os ficheiros;
 - 3.3. O Utilizador indica o caminho virtual para os ficheiros;
 - 3.4. O Sistema descarrega todos os ficheiros indicados;
-

- **Fluxo de excepção 1:** [Caminho para ficheiros inválido] (passo 8)

- 8.1. O Sistema deteta um caminho inválido avisando o cliente e não permite o Download de Conteúdo, saindo deste Use Case;

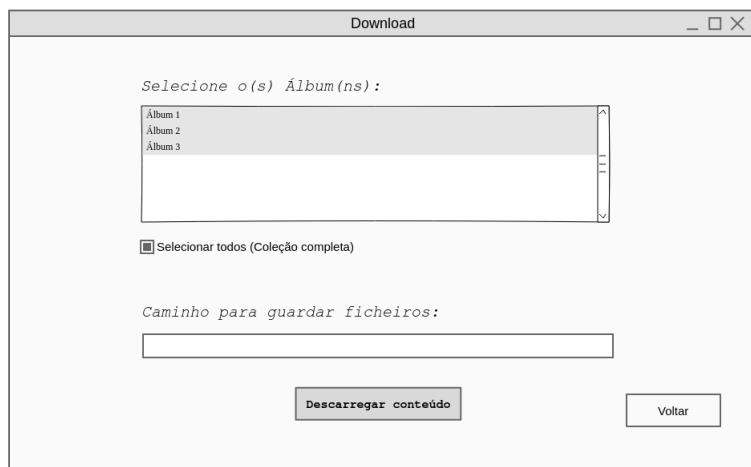


Figura 2.13: Interface para descarregar conteúdo do Sistema.

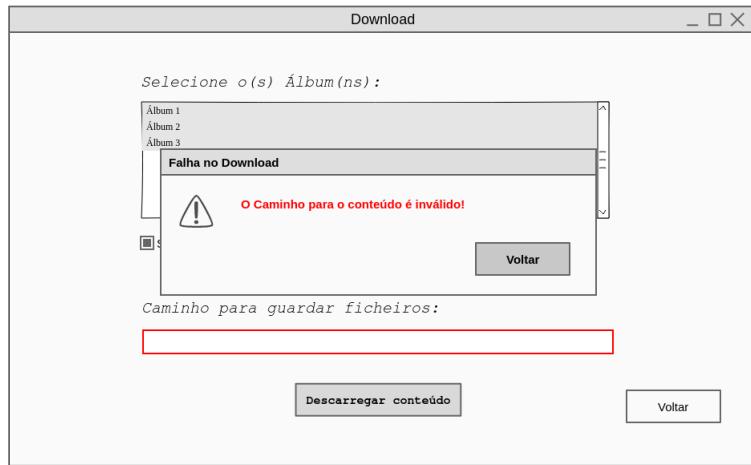


Figura 2.14: Interface para descarregar conteúdo do Sistema (**Fluxo de Excepção 1**).

2.4.6 Criar Álbuns - (Futura impl.)

- **Descrição:**

Criar albúns de conteúdo no Sistema.

- **Cenários:**

"(...) gastou algum tempo a organizar o conteúdo de acordo com a sua preferência (...)"

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente e tem de ter conteúdo na sua conta pessoal.

• **Pós-Condição:**

- O Utilizador tem um Álbum novo.
-

• **Fluxo Normal:**

1. O Utilizador seleciona a opção "Criar Álbum";
 2. O Sistema apresenta a lista de conteúdo da sua conta pessoal e espera seleção;
 3. O Utilizador seleciona o conteúdo (da sua conta pessoal);
 4. O Sistema agrupa o conteúdo num Álbum e pergunta se quer manualmente categorizar o conteúdo ou gerar automaticamente;
 5. O Utilizador seleciona a opção manual insere os dados (nome e categoria da coleção);
 6. O Sistema cria o novo Álbum e guarda-o na conta pessoal do Utilizador.
-

• **Fluxo Alternativo 1:** [A categoria é gerada automaticamente] (passo 5)

- 5.1. O Utilizador seleciona a opção de gerar automaticamente;
- 5.2. O Sistema pergunta qual o nome a atribuir ao Álbum;
- 5.3. O Utilizador indica o nome a atribuir;
- 5.4. O Sistema gera automaticamente a categoria do Álbum e guarda-o na conta pessoal do Utilizador com o nome definido pelo mesmo.

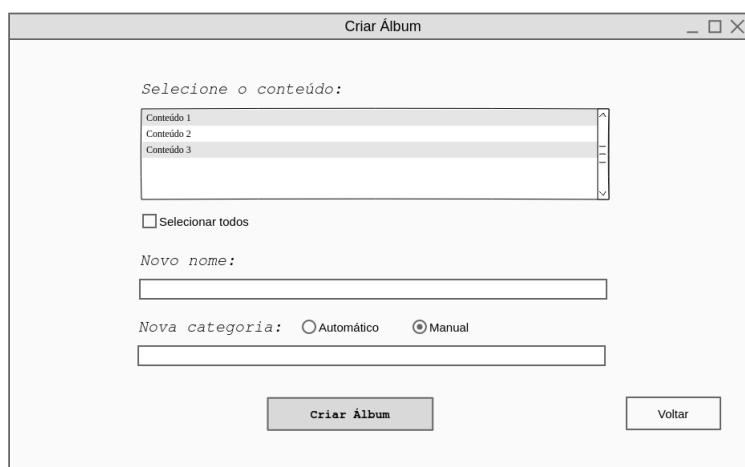


Figura 2.15: Interface para criar Álbuns.

2.4.7 Eliminar Álbum - (Futura impl.)

- **Descrição:**

Eliminar álbuns de um Utente.

- **Cenários:**

Cenário Geral: Para facilitar a gestão de conteúdo pelos Utentes acrescenta-se a possibilidade de remover álbuns.

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente e tem de ter conteúdo na sua conta pessoal.

- **Pós-Condição:**

- O Utilizador removeu um álbum da sua coleção.

.....

- **Fluxo Normal:**

1. O Utilizador seleciona a opção "Eliminar Álbum";
 2. O Sistema apresenta a lista de álbuns da sua conta pessoal e espera seleção;
 3. O Utilizador seleciona o(s) álbum(ns) (da sua conta pessoal);
 4. O Sistema remove o(s) álbum(ns) selecionado(s);
-

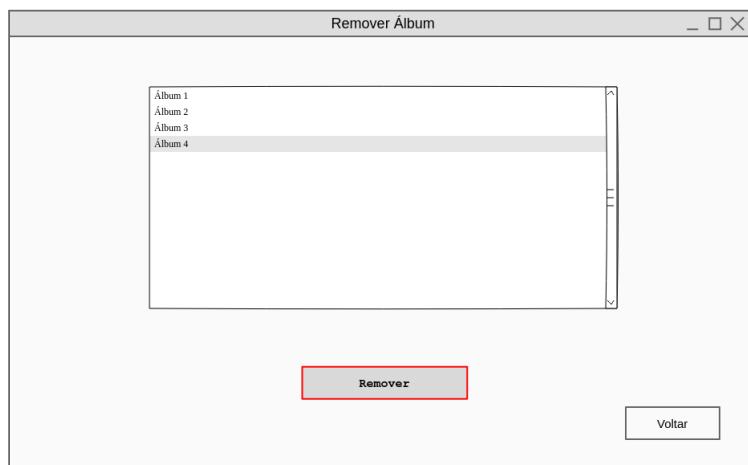


Figura 2.16: Interface para remover Álbuns por um Utente.

2.4.8 Editar Álbuns - (Futura impl.)

- **Descrição:**

Editar Álbuns (alterar o seu nome e categoria) do Sistema.

- **Cenários:**

”(...) gastou algum tempo a organizar o conteúdo de acordo com a sua preferência (...)”

”(...) tinha adicionado dois ou três álbuns classificados como Jazz, mas que para ele eram de Acid Jazz, e ainda não se tinha dado ao trabalho de lhes alterar a classificação (...)”

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente e tem de ter conteúdo na sua conta pessoal.

- **Pós-Condição:**

O Utilizador alterou o nome e/ou categoria de um Álbum.

.....

- **Fluxo Normal:**

1. O Utilizador seleciona a opção ”Editar Álbum”;
2. O Sistema apresenta a lista Albúns e espera seleção;
3. O Utilizador seleciona o Álbum;
4. O Sistema pergunta qual o novo nome e categoria que deve ser estabelecido no Álbum;
5. O Utilizador insere os dados pedidos;
6. O Sistema altera os dados do Álbum;

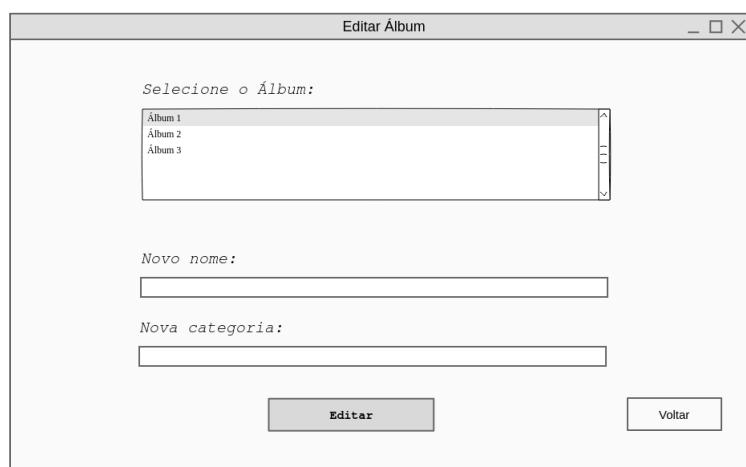


Figura 2.17: Interface para editar Álbuns.

2.4.9 Mover Conteúdo - (Futura impl.)

- **Descrição:**

Mover conteúdo presente no Sistema de um Álbum para outro da sua coleção.

- **Cenários:**

”(...) gastou algum tempo a organizar o conteúdo de acordo com a sua preferência (...)”

- **Pré-Condição:**

O Utilizador tem de se autenticar e deve ter conteúdo para ser movido.

- **Pós-Condição:**

O Sistema moveu um Conteúdo de um Álbum para outro.

.....

- **Fluxo Normal:**

1. O Utilizador seleciona a opção ”Mover Conteúdo”;
2. O Sistema apresenta a lista de conteúdo presente na sua coleção;
3. O Utilizador seleciona um elemento da coleção;
4. O Sistema apresenta a lista de Álbuns da sua coleção;
5. O Sistema espera seleção do Álbum destino;
6. O Utente seleciona o Álbum destino;
7. O Sistema move o conteúdo de um álbum para o outro;

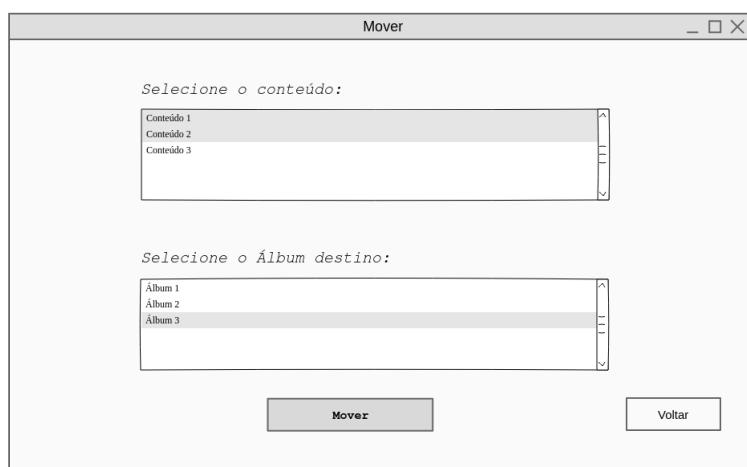


Figura 2.18: Interface para mover conteúdo.

2.4.10 Reproduzir Conteúdo - (Implementado)



Figura 2.19: Interface para reproduzir conteúdo (Convidado).



Figura 2.20: Interface para reproduzir conteúdo (Utente).

Reproduzir Álbum - (Implementado)

- **Descrição:**

Reproduzir um Álbum da coleção do Utilizador.

- **Cenários:**

Um Utente pretende reproduzir um Álbum da sua coleção.

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente;

- **Pós-Condição:**

O Sistema reproduziu o Conteúdo selecionado pelo Utente;

- **Fluxo Normal:**

1. O Utente seleciona a opção "Reproduzir Álbum";
2. O Sistema apresenta a lista de Álbuns da coleção do Utente;
3. O Utente seleciona reproduzir o Álbum ou um conteúdo do Álbum;
4. O Sistema reproduz o conteúdo;

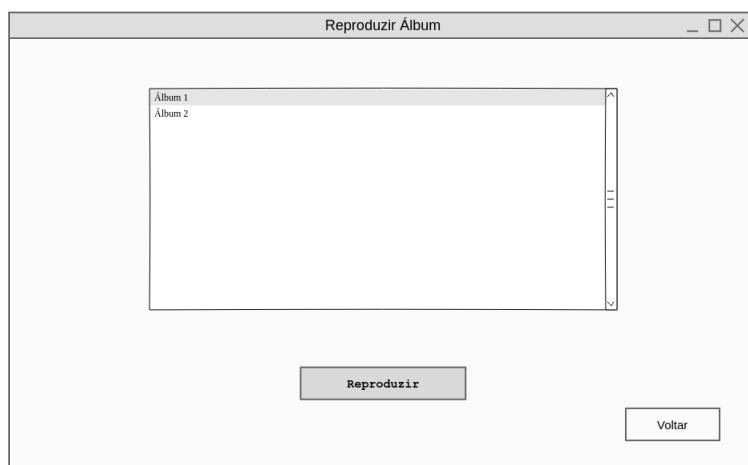


Figura 2.21: Interface para reproduzir Álbuns.

Reproduzir Conteúdo Individual - (Implementado)

- **Descrição:**

Reproduzir um conteúdo individual da biblioteca do sistema.

- **Cenários:**

Um Utente pretende reproduzir um conteúdo individual da Biblioteca do Media Center.

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente.

- **Pós-Condição:**

O Sistema reproduziu o Conteúdo selecionado pelo Utente.

.....

- **Fluxo Normal:**

1. O Utente seleciona a opção "Biblioteca do Media Center";
2. O Sistema apresenta a lista de Conteúdo da Biblioteca;
3. O Utente seleciona o conteúdo a reproduzir;
4. O Sistema reproduz o conteúdo;

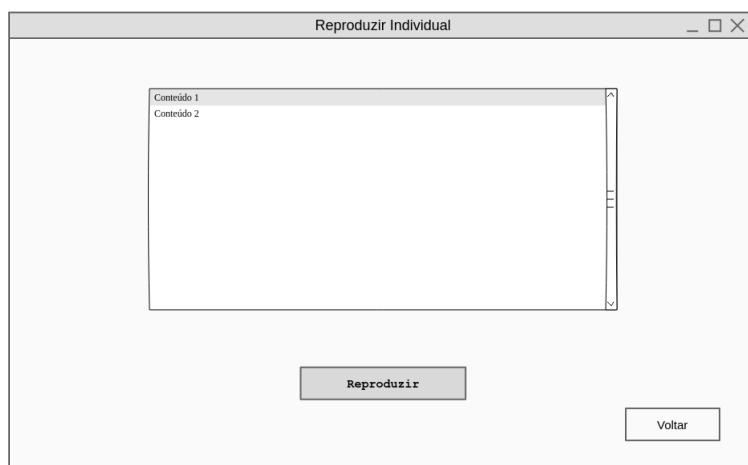


Figura 2.22: Interface para reproduzir um conteúdo individual.

Reproduzir Categoria da coleção - (Futura impl.)

- **Descrição:**

Reproduzir uma categoria da coleção do Utilizador.

- **Cenários:**

Um Utente pretende reproduzir uma categoria de conteúdo da sua coleção.

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente.

- **Pós-Condição:**

O Sistema reproduziu uma categoria da coleção selecionado pelo Utente.

.....

- **Fluxo Normal:**

1. O Utente seleciona a opção "Reproduzir Categoria da Coleção";
 2. O Sistema apresenta uma lista de categorias presentes;
 3. O Utente seleciona a Categoria;
 4. O Sistema filtra a coleção e pergunta se quer reproduzir em modo aleatório ou normal;
 5. O Utente seleciona o modo normal;
 6. O Sistema reproduz o conteúdo filtrado em modo normal.
-

- **Fluxo Alternativo 1:** [O Utente escolhe modo aleatório] (passo 6)

- 5.1. O Utente seleciona a opção modo aleatório;
- 5.2. O Sistema reproduz o conteúdo filtrado em modo aleatório.

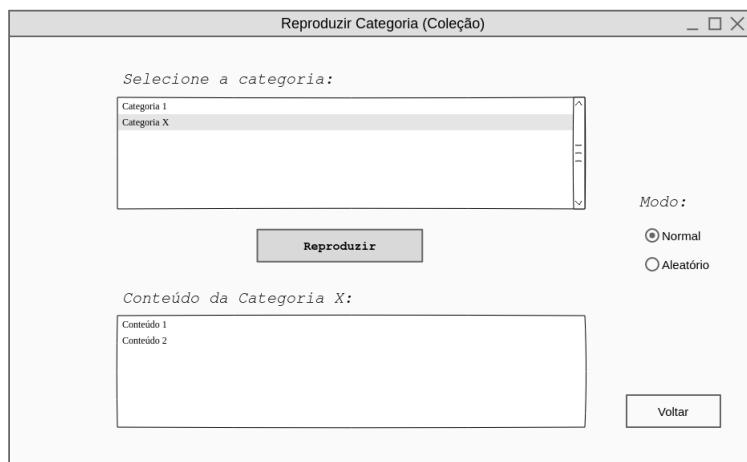


Figura 2.23: Interface para reproduzir um conteúdo da coleção do Utente.

Rep. uma categoria da biblio. do Media Center - (Futura impl.)

- **Descrição:**

Reproduzir uma categoria da biblioteca do Media Center.

- **Cenários:**

Um Utente ou Convidado pretende reproduzir uma categoria de conteúdo do Media Center.

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente ou Convidado.

- **Pós-Condição:**

O Sistema reproduziu uma categoria da biblioteca do Media Center.

.....

- **Fluxo Normal:**

1. O Utente seleciona a opção "Reproduzir Categoria do Media Center";
 2. O Sistema apresenta uma lista de categorias presentes;
 3. O Utente seleciona a Categoria;
 4. O Sistema filtra a biblioteca e pergunta se quer reproduzir em modo aleatório ou normal;
 5. O Utente seleciona o modo normal;
 6. O Sistema reproduz o conteúdo filtrado em modo normal.
-

- **Fluxo Alternativo 1:** [O Utente escolhe modo aleatório] (passo 6)

- 5.1. O Utente seleciona a opção modo aleatório;
- 5.2. O Sistema reproduz o conteúdo filtrado em modo aleatório.

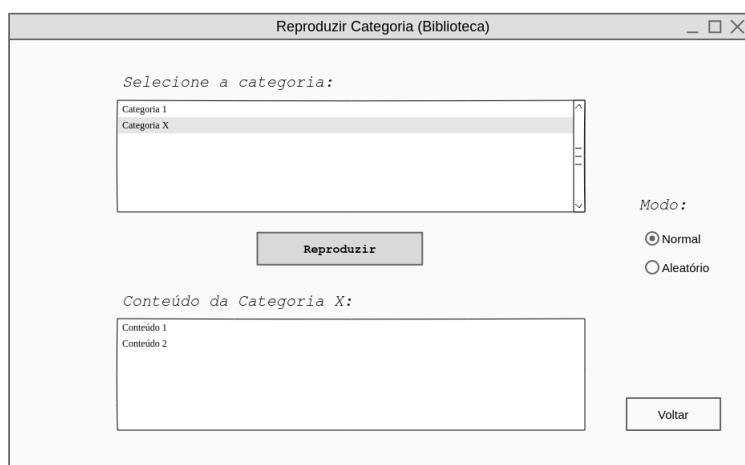


Figura 2.24: Interface para reproduzir um conteúdo da Biblioteca do Media Center.

2.4.11 Sair do Sistema (Log-out) - (Implementado)

- **Descrição:**

Sair do Sistema Media Center.

- **Cenários:**

(...) o Ricardo fez logout do media center (...)

- **Pré-Condição:**

O Utilizador tem de estar autenticado no sistema.

- **Pós-Condição:**

O Utilizador saiu do sistema

- **Fluxo Normal:**

1. O Utente seleciona a opção "Log-out";
2. O Sistema faz Log-out do Utilizador.

2.5 Outros Mockups

Por fim, são apresentados os protótipos de interfaces "*intermediárias*" referidas ao longo da especificação dos *Use Cases* apresentada no capítulo anterior e definidos na **Fase 1**.

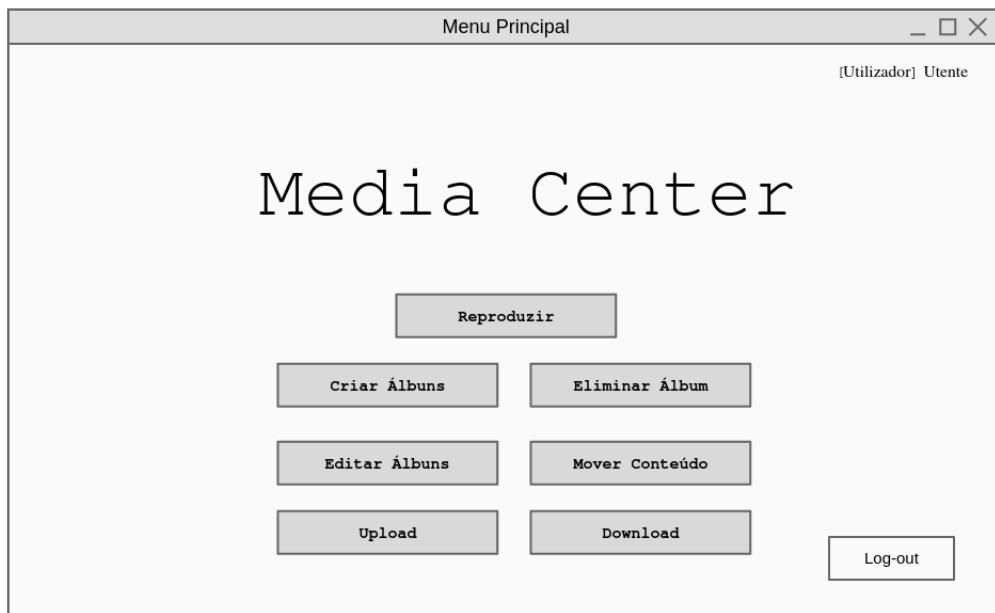


Figura 2.25: Interface que representa o Menu principal de um Utente.

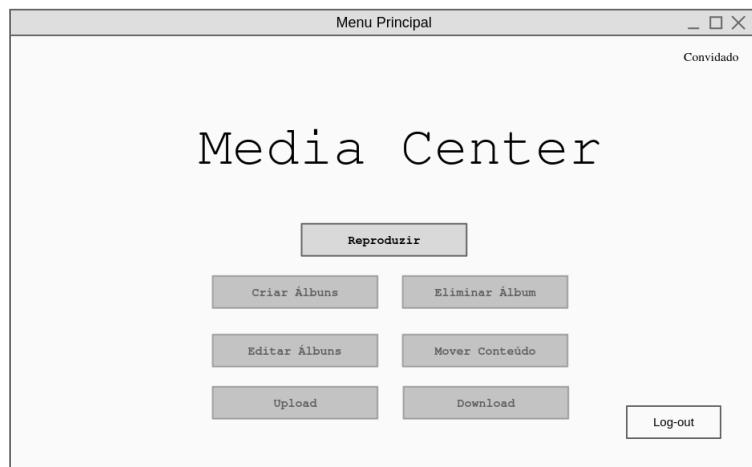


Figura 2.26: Interface que representa o Menu principal de um Convidado.

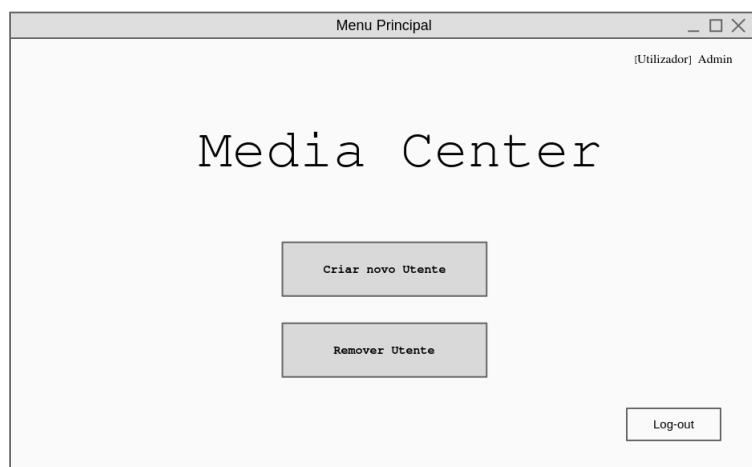


Figura 2.27: Interface que representa o Menu principal de um Administrador.

Capítulo 3

Fase 2

Na segunda fase, foi feita uma seleção dos *Use Cases* mais relevantes para a futura implementação da aplicação. Com essa seleção partimos para a modelação comportamental e arquitetural do sistema e a partir da divisão do modelo de domínio em partes lógicas definimos as coleções principais do nosso sistema para armazenar todos os dados e as operações mais importantes sobre as diferentes classes do *Media Center* de modo a satisfazer e tornar todos os processos descritos nos *Use Cases* mais eficientes.

3.1 Seleção dos *Use Cases*

Nesta secção vamos apresentar a seleção efetuada pela equipa docente e posteriormente pelo grupo do conjunto de *Use Cases* mais relevantes para esta segunda fase. A figura seguinte mostra o descrito anteriormente:

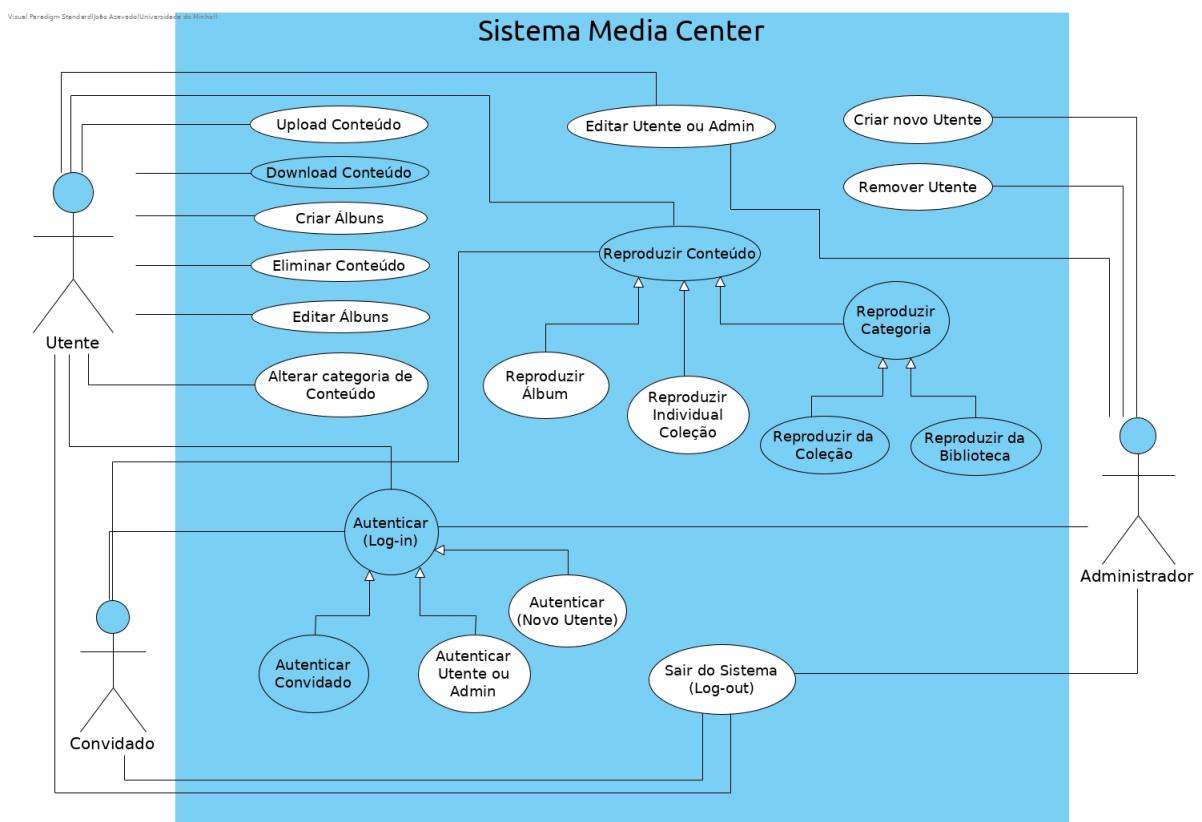


Figura 3.1: Seleção dos *Use Cases* considerados (a branco).

Aproveitamos também para generalizar todos os casos de uso propostos pela equipa docente naqueles implementados pelo nosso grupo para um maior entendimento dos mesmos, i.e. a correspondência entre os nomes dados aos *Use Cases* que se encontra na tabela seguinte:

Use Case da equipa docente	Use Case do grupo
Registrar Utilizador	Autenticar (Novo Utente)
Iniciar Sessão	Autenticar Utente ou Admin
Terminar Sessão	Sair do Sistema (Log-out)
Editar utilizador	Editar Utente ou Admin
Eliminar utilizador	Remover Utente
Fazer upload de conteúdo	Upload Conteúdo
Alterar categoria de conteúdo	Ler Nota 1
Remover conteúdo	Eliminar Conteúdo
Reproduzir conteúdo	Repr. Álbum ou Repr. Individual Coleção (Biblioteca)
Criar playlist	Criar Álbuns
Extra	Editar Álbuns

Tabela 3.1: Correspondência de Use Cases.

Assim, após a definição dos *Use Cases* a implementar, procedemos à sua especificação através de diagramas de interação, mais propriamente, definimos os *diagramas de sequência* que descrevem a interação (mensagens) entre as diferentes classes que compõem o sistema *Media Center*.

Nota 1: Inicialmente, o *Use Case Alterar categoria de conteúdo*, não foi considerado pelo grupo, pelo que foi especificada uma alternativa ao mesmo, sendo essa, o *Use Case Editar Álbuns/Playlist*. Deste modo sentimos a necessidade de proceder à especificação deste “novo” *Use Case*. A sua especificação encontra-se de seguida:

Alterar categoria de conteúdo

- **Descrição:**

Alterar categoria de um conteúdo da coleção pessoal do Utilizador.

- **Cenários:**

”(...) gastou algum tempo a organizar o conteúdo de acordo com a sua preferência (...)”

- **Pré-Condição:**

O Utilizador tem de estar autenticado no Sistema como um Utente.

- **Pós-Condição:**

O Sistema alterou a categoria de um conteúdo selecionado pelo Utilizador.

- **Fluxo Normal:**

1. O Utilizador seleciona a opção ”Alterar categoria de conteúdo”;
2. O Sistema apresenta uma lista de conteúdo da coleção do Utilizador;
3. O Utente seleciona a nova categoria do conteúdo;
4. O Sistema altera a categoria do conteúdo selecionado;

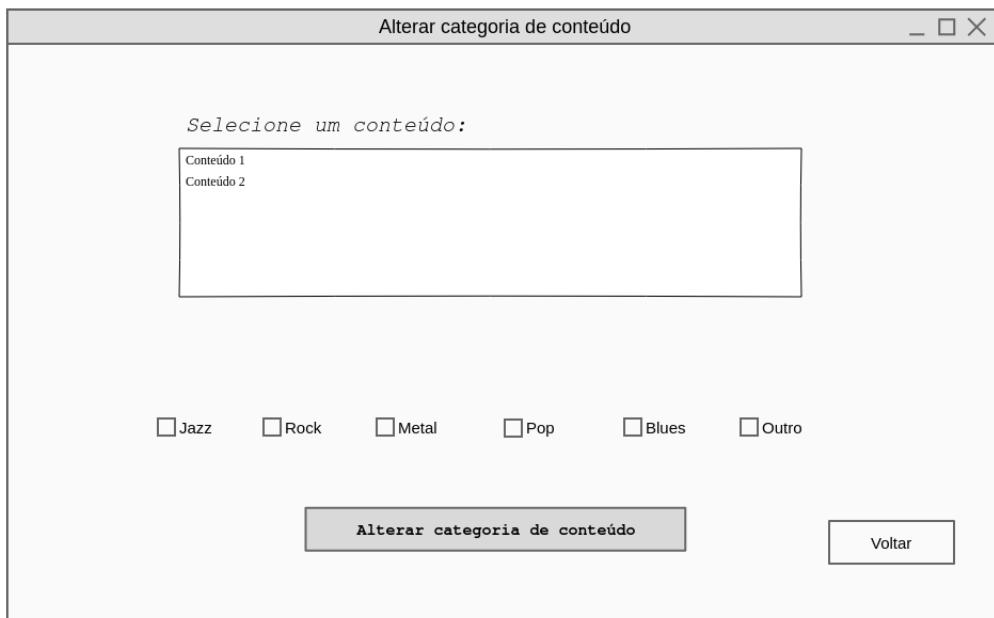


Figura 3.2: Menu Alterar Categoria de conteúdo.

3.2 Modelação de interação

A fase de modelação que enfrentamos agora segue no sentido de prevenir decisões precipitadas na implementação, pelo que nos permite modelar o sistema de forma a estabelecer as classes principais e as mensagens que as mesmas trocam entre elas.

Assim, o primeiro passo seria definir as classes principais do sistema. O grupo decidiu que seria necessário ter um *Facade* para abstrair e estabelecer a API fundamental por de trás de toda a lógica de negócio. Esta classe principal guardaria todos os dados necessários a carregar a aplicação no seu estado anterior.

De modo a clarificar as ideias, temos definido nesta **Fase 2**:

1. **Facade**: ”*MediaCenter_LN*”, que contém uma API para:

- Gestão de Utilizadores;
- Gestão de Sessão atual;
- Gestão de estruturas de dados e persistência dos mesmos (será falado mais à frente).

2. **GUI**: ”*MediaCenter_GUI*”, que contém:

- Interfaces da aplicação;
- Interação e IO;
- Reprodução de media **mp3** e **mp4**.

3. **Outras** : ”*Utilizador, Album e Conteúdo*”, que contém:

- **Utilizador**: Gestão de um Utilizador: credenciais, permissões no sistema, informação pessoal e gestão do seu conteúdo;

- **Conteúdo:** Gestão de um Conteúdo: identificação, localização do ficheiro, tipo de conteúdo e gestão de categoria;
- **Álbuns:** Gestão de uma coleção de conteúdo associado a um Utilizador, identificação e gestão de categorias associadas.

Tendo definido estas entidades principais, que posteriormente poderiam sofrer alterações na sua API, passamos à implementação das interações entre as mesmas e posterior definição de diagramas de interação e estrutura.

3.2.1 Diagramas de sequência

Os diagramas de sequência que modelam os *Use Cases* definidos na segunda fase podem ser visualizados num **pdf anexo** com o nome **anexo-diagramas-de-sequencia.pdf**. Estes constituem uma primeira abordagem ao sistema para uma análise sem serem consideradas Bases de Dados para persistir os dados.

3.3 Modelação estrutural

Será apresentada a concepção da arquitetura que, à priori, é capaz de suportar o conjunto de funcionalidades, i.e. de *Use Cases*, visto que foi desenvolvida nesse mesmo sentido. Assim, apresentam-se no Diagrama de Classes todos os métodos associados às diferentes classes que compõem o nosso sistema. O desenvolvimento do diagrama de package permitiu reduzir a complexidade do número crescente de classes que foram surgindo dividindo a aplicação em diferentes camadas lógicas, de controlo e de interface. Permitiu ainda estabelecer as diferentes dependências entre as diferentes camadas da aplicação no sentido de verificar a validade das mesmas.

3.3.1 Diagrama de Classe

Nota: A definição a seguir apresentada constitui a **Primeira Versão** estabelecida para a arquitetura do sistema, sendo que, mais à frente, a mesma será atualizada.

(ver diagrama de classe na próxima página)

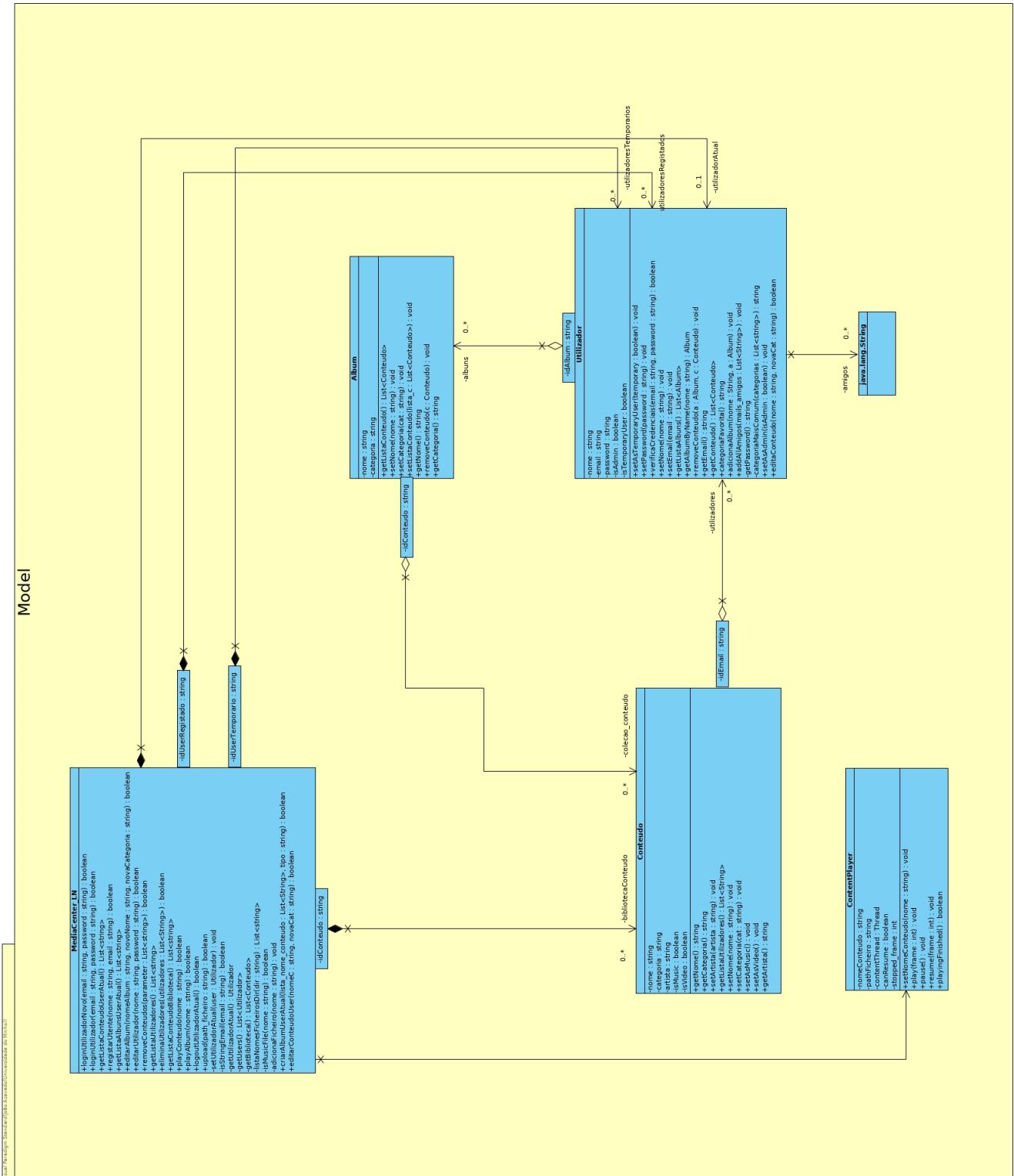


Figura 3.3: Diagrama de Classes (Camada de Aplicação) - Versão 1

3.3.2 Diagrama de Package

(ver diagrama de package na próxima página)

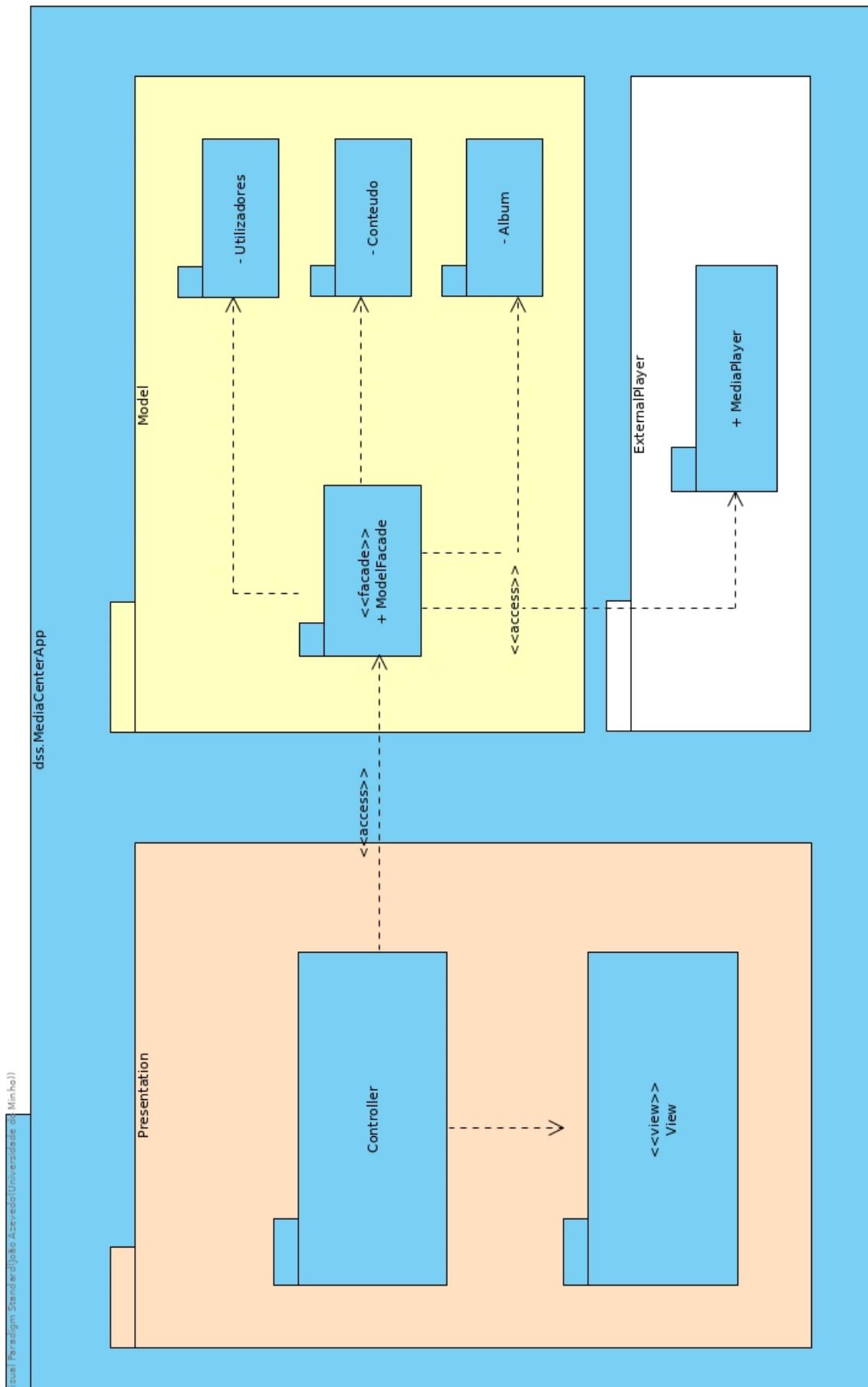


Figura 3.4: Diagrama de Package.

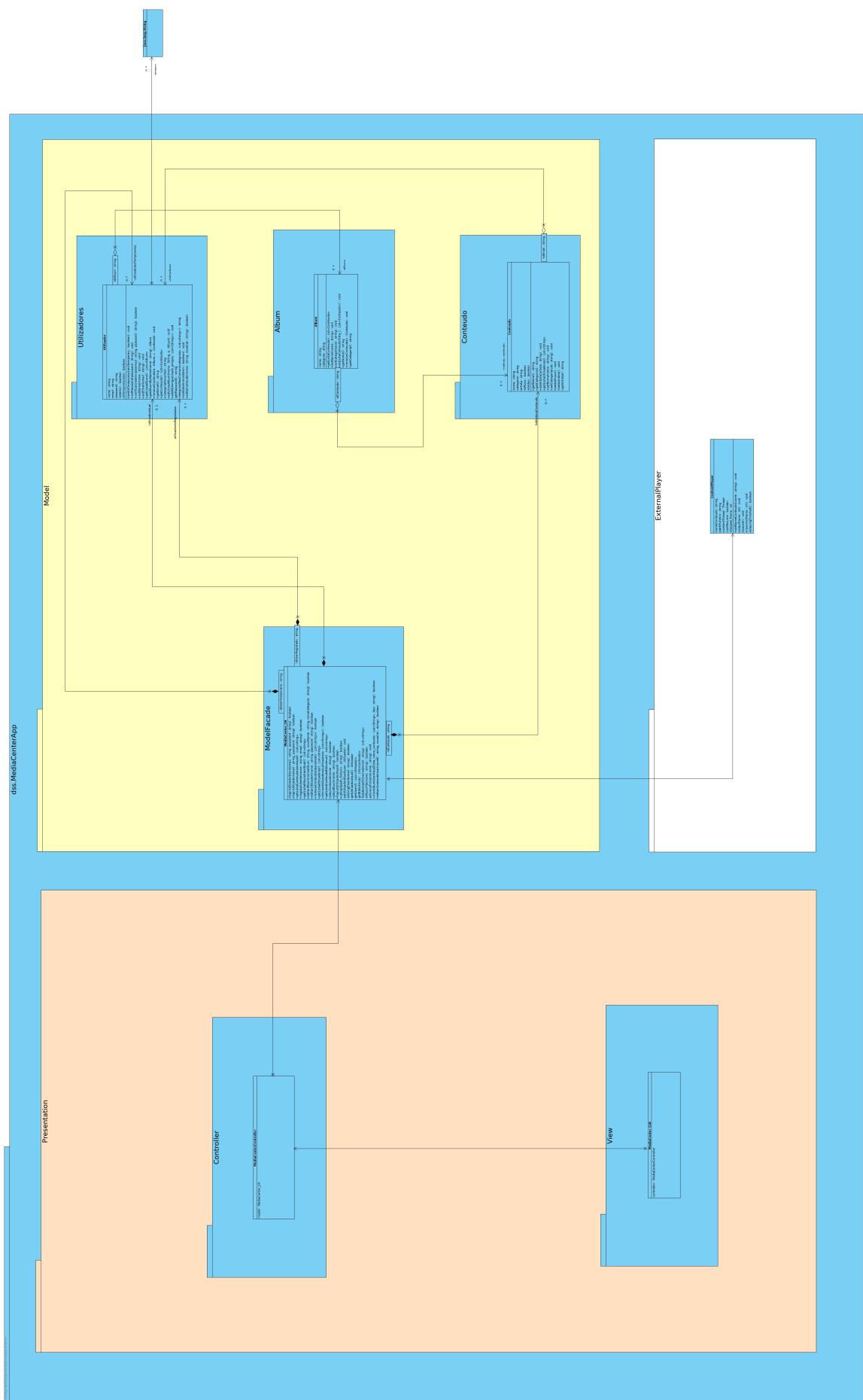


Figura 3.5: Representação das diferentes classes no diagrama de package.

Capítulo 4

Fase 3

4.1 Síntese das Fases 1 e 2 e próximas decisões

Até agora passamos de uma visão abstrata da análise de cenários fornecidos por um Cliente (enunciado do projeto) para uma visão estruturada da aplicação em entidades, posteriormente classes e estabelecendo mensagens entre as mesmas.

Um aspecto fundamental é o seguinte: Todos os dados até agora são armazenados e geridos por estruturas definidas pelo **Java**, como *Maps*, *Lists*, etc..., que precisam de ser armazenadas em ficheiros *object* para posteriormente serem carregadas. Este processo de guardar e carregar as estruturas de dados tem um *overhead* que é rapidamente escalável para aplicações de grande escala, sendo que, esta aplicação não é exceção, nem deve ser considerada como tal.

A partir desta prespetiva, o próximo passo será garantir que as limitações de acesso aos dados, procura, organização é mantida fora da memória principal do computador e não precisa de ser carregada na totalidade mas sim por partes quando for preciso ao longo de toda a implementação.

Assim, nada melhor que criar uma Base de Dados relacional que simula os Mapeamentos até agora considerados e a persistência dos mesmos dados.

4.2 Data Access Objects e persistência de dados

Nesta secção, iremos explicar as decisões tomadas a nível da criação de toda a base de dados relacional associada a esta aplicação.

4.2.1 Base de dados em MySql

Deste modo, o primeiro passo foi pegar nas estruturas principais até agora consideradas que servirão de tabelas de dados na Base de dados:

1. **Utilizadores:** Os Utilizadores são identificados unicamente pelo seu e-mail e, assim, esta propriedade será a nossa **Primary Key** da tabela de utilizadores. Os outros atributos associados a cada Utilizador são os considerados até agora: nome, password e permissões.
2. **Álbuns:** Cada Utilizador terá referências para os seus álbuns através de uma referência pelo nome do álbum e pelo id do Utilizador que o possui através de uma relação **n para n**. Este tuplo (idUtilizador, idAlbum) identifica unicamente o álbum que pertence a um dado Utilizador.
3. **Conteúdo:** O Conteúdo não poderá existir se não existir um álbum, e consequentemente, existirá se um utilizador possuir um álbum, pelo menos. Assim o Conteúdo pertence a um Álbum e está associado por uma relação **n para n**.
4. **Categoria:** A categoria é propriedade de um Conteúdo e de um Álbum pois ambos possuem esta característica. Por outro lado, de modo a limitar a quantidade de categorias existentes no sistema, e não termos ambiguidades como uma categoria chamada "rock" e

outra "Rock", decidimos criar uma tabela de Categorias identificadas pelo seu nome identificador e, na escolha da categoria, por parte de um utilizador, é limitada.

5. Amigo: Os amigos, apesar de não pertencerem à implementação final da aplicação, deverão ser considerados de modo a suportar alguns casos de uso considerados nas fases 1 e 2. Para isso criamos uma relação **n para n** de Utilizador para Amigo, criando uma tabela de associações intermediária para guardar a associação entre um amigo (email) e um Utilizador (id do Utilizador).

Definimos assim, o seguinte modelo relacional:

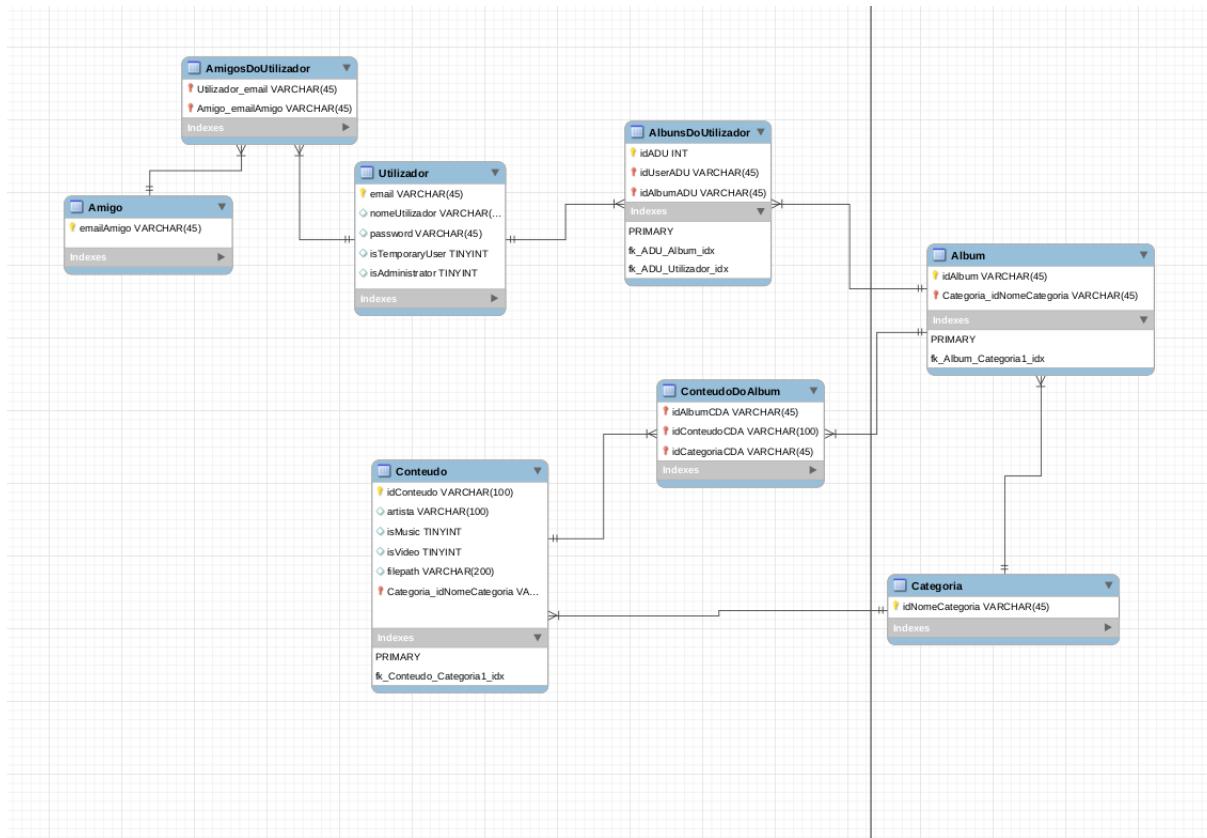


Figura 4.1: Modelo relacional da Base de Dados do Media Center.

4.2.2 DAOs

Para aceder à base de dados através de uma linguagem como **Java** é importante seguir um padrão associado a linguagens orientadas a objetos, padrão esse que se baseia na criação de *Data Access Objects*.

Os DAOs expõem um API de acesso, atualização e inserção de dados na Base de Dados da nossa aplicação através da criação de *statements* que executam *queries* em **sql**.

O próximo passo será definir quais os DAOs necessários para dividir a lógica de acesso à base de dados para garantir a persistência da estruturação desta aplicação por camadas independentes.

Assim, definimos os seguintes DAOs:

1. UtilizadorDAO: Este DAO está associado a todos os dados relativos a Utilizadores, anteriormente associado a um **Map<String, Utilizador>** que nos permite associar, procurar e atualizar um Utilizador identificado pelo seu ID (E-Mail). Este DAO expõem uma API de funções exportadas da interface **Map** como:

- + containsKey(email:String) : boolean; //Se existe um Utilizador

- + get(email:String) : Utilizador; //Ir buscar um Utilizador dado o email
 - ...
2. **BibliotecaDAO:** Este DAO está associado a todos os dados relativos ao conteúdo pertencente à Biblioteca do Media Center, ou seja, o antigamente considerado no nosso diagrama de classes Map<String, Conteudo> permitindo identificar, procurar e adicionar novos conteúdos à Base de Dados. Algumas das funções disponíveis são as seguintes:
- + keySet() : Set<String>; //Lista todos os nomes dos conteúdos
 - + size() : int; //nº de conteúdos diferentes
 - + getOwners(nomeConteudo:String) : List<String>; //Emails dos donos do conteúdo
3. **ConteudoPessoalDAO:** Este DAO, por fim, está associado ao conteúdo pessoal de um Utilizador, no qual se inclui os Álbuns do Utilizador e o Conteúdo associado ao mesmo e, como, na base de dados, um Conteúdo não existe se não existir um Álbum e consequentemente um Utilizador, decidimos incluir, neste DAO, toda a API relativa às operações em Álbuns e Conteúdo referentes a um dado Utilizador (a um dado email). Assim, podemos definir operações como:

- + getListaConteudo() : List<String>; //Listar o nome do conteúdo de um User
- + adicionaAlbum(a : Album) : void; //Adicionar um álbum à coleção do User
- + getListaAlbuns() : List<String> //Listar o nome dos Álbuns do User
- + categoriaFavorita() : String; //Categoria mais comum de um Utilizador

4.3 Redefinição dos diagramas de Sequência

Após garantir a persistência dos dados redefinimos os diagramas de Sequência até agora implementados, limitando a definição apenas aos casos de uso limitados na terceira fase do projeto, entre eles:

- **Iniciar Sessão;**
- **Terminar Sessão;**
- **Upload Conteúdo;**
- **Alterar Categoria de Conteúdo;**
- **Reproduzir Conteúdo;**

Nota: No Reproduzir consideramos duas formas de o fazer: Reproduzir conteúdo da biblioteca do Media Center e reproduzir um Álbum do Utilizador.

(A definição dos DSS encontra-se na próxima página)

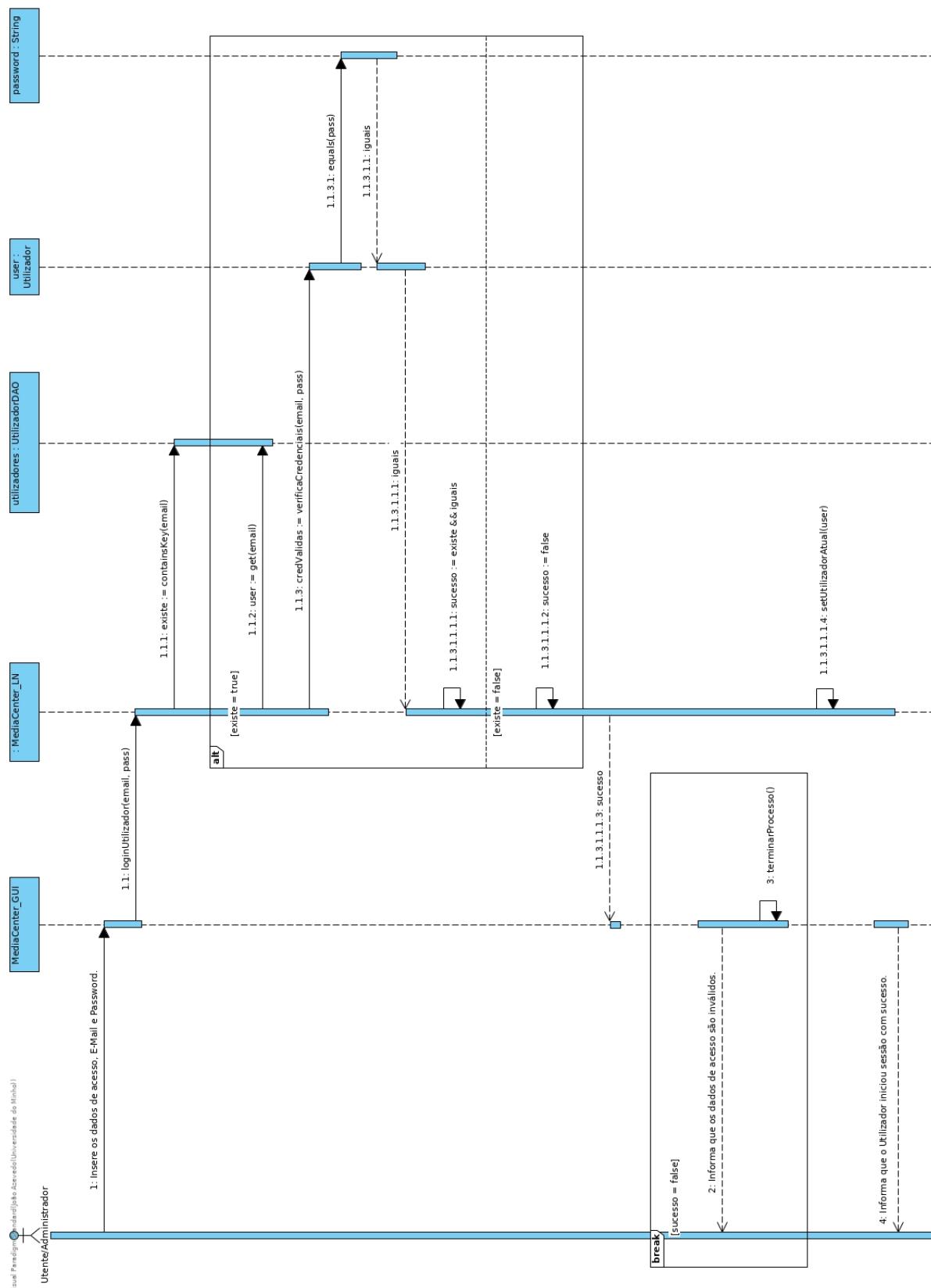


Figura 4.2: Iniciar Sessão - DSS.

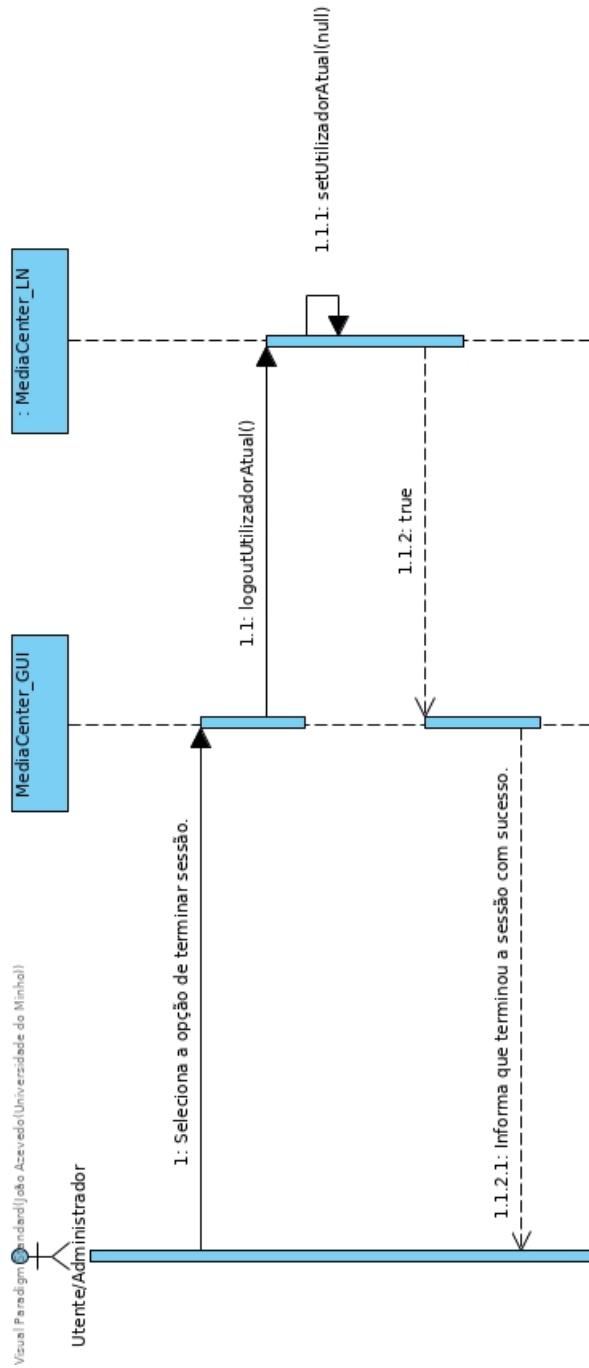


Figura 4.3: Terminar Sessão - DSS.

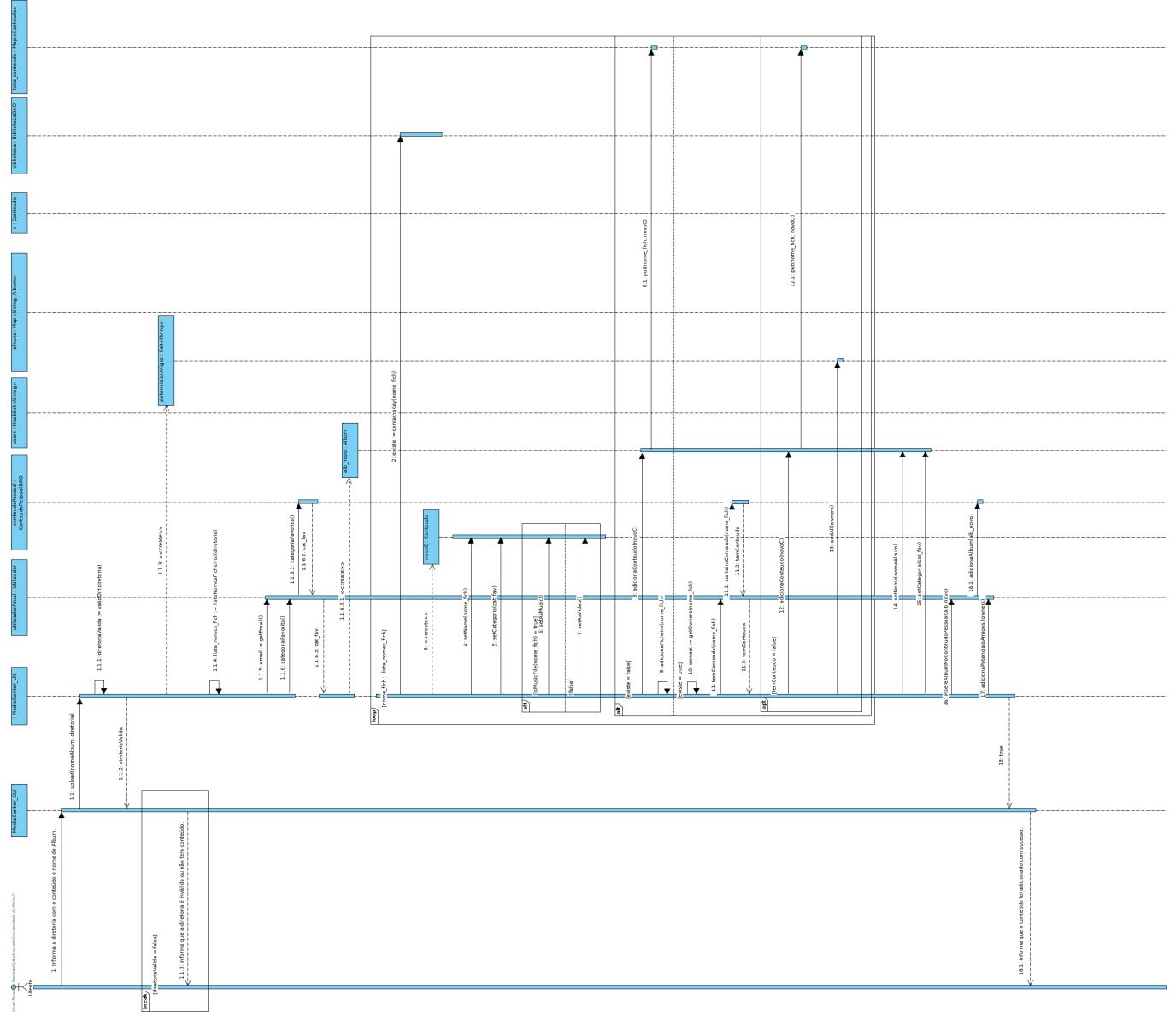


Figura 4.4: Upload Conteúdo - DSS.

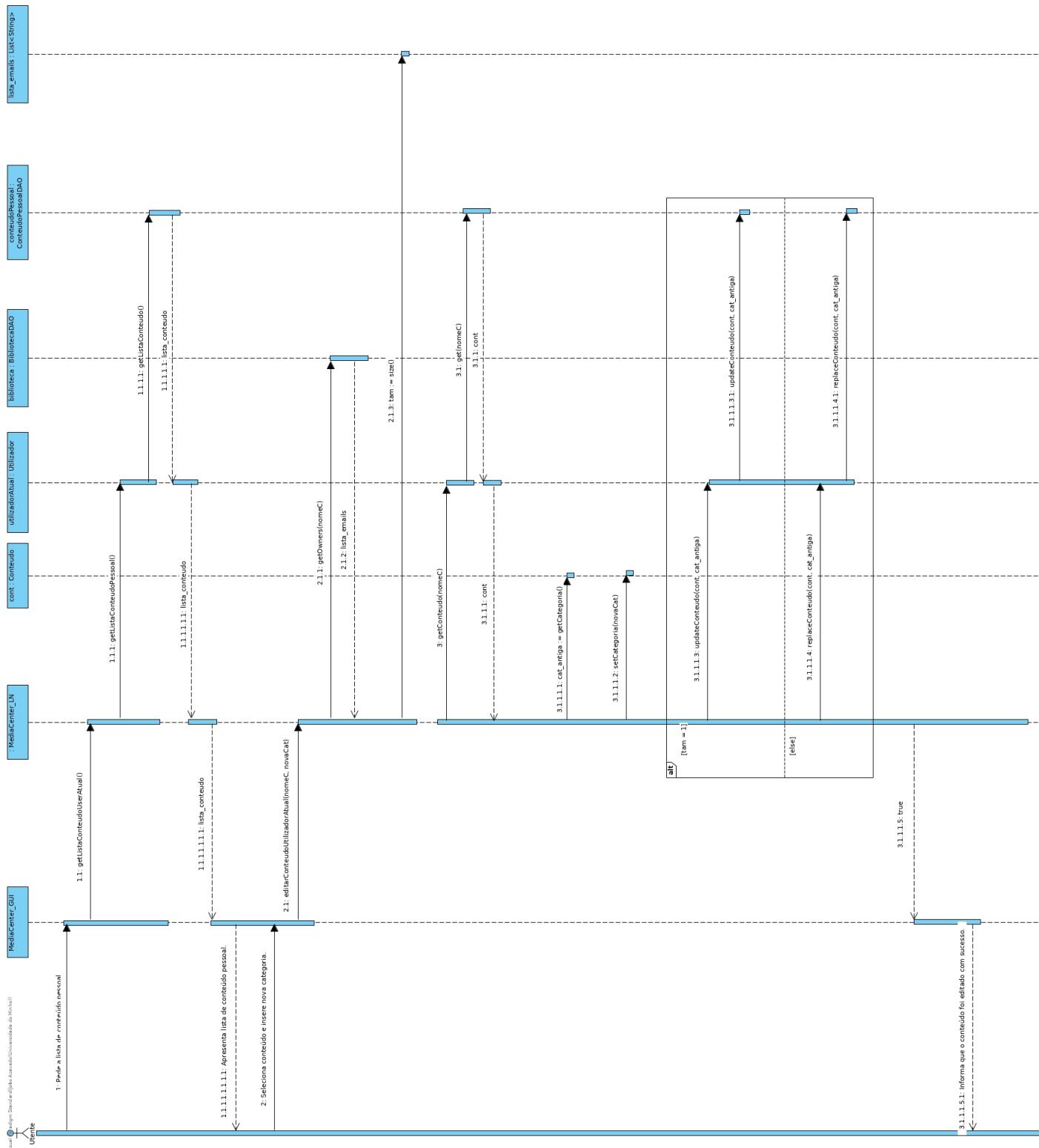


Figura 4.5: Alterar categoria de Conteúdo - DSS.

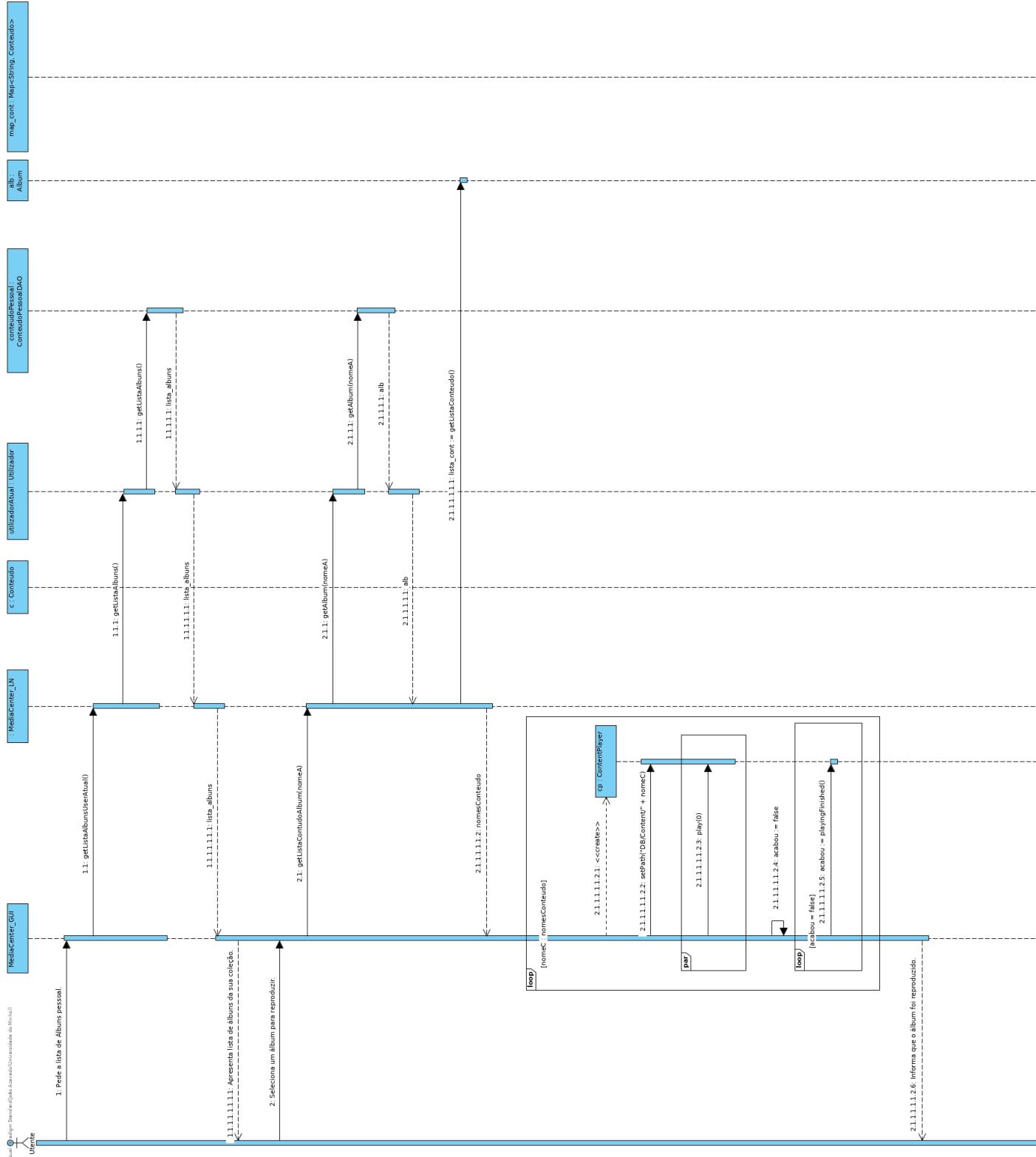


Figura 4.6: Reproduzir Álbum - DSS.

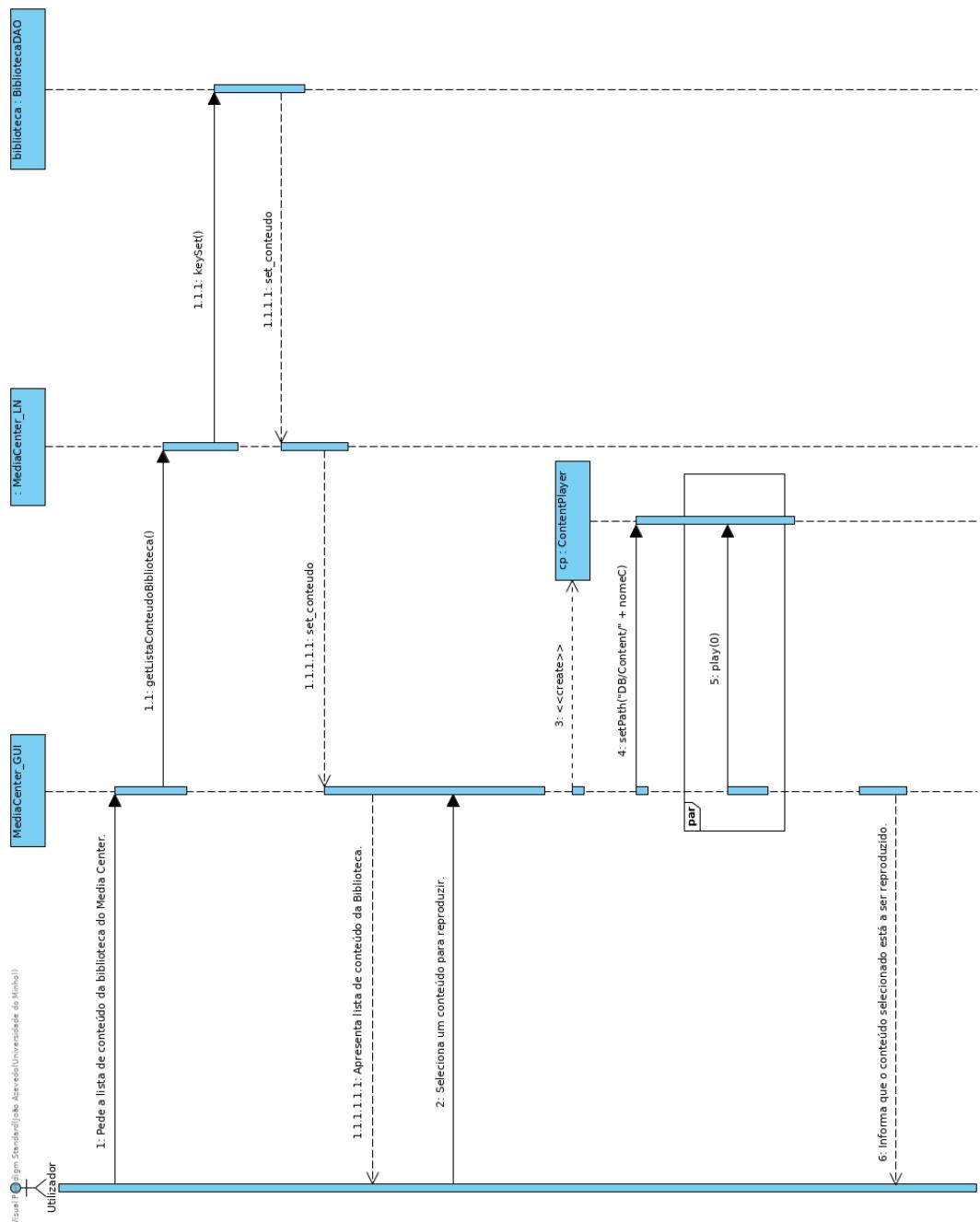


Figura 4.7: Reproduzir conteúdo da Biblioteca - DSS.

Capítulo 5

Modelação final

Após todas as iterações deste trabalho, chegamos à definição final, à partida, da modelação que compõem este sistema. Obviamente, toda esta definição está a incluir os casos de uso que foram limitados na última fase e consequentemente o conjunto de métodos que compõem a API até agora definida. No entanto, a aplicação evoluiu de forma a conseguir ser estruturada de forma independente, de tal modo que, uma alteração num componente deste sistema não altera a conceção de outro componente.

5.1 Diagrama de Use Case

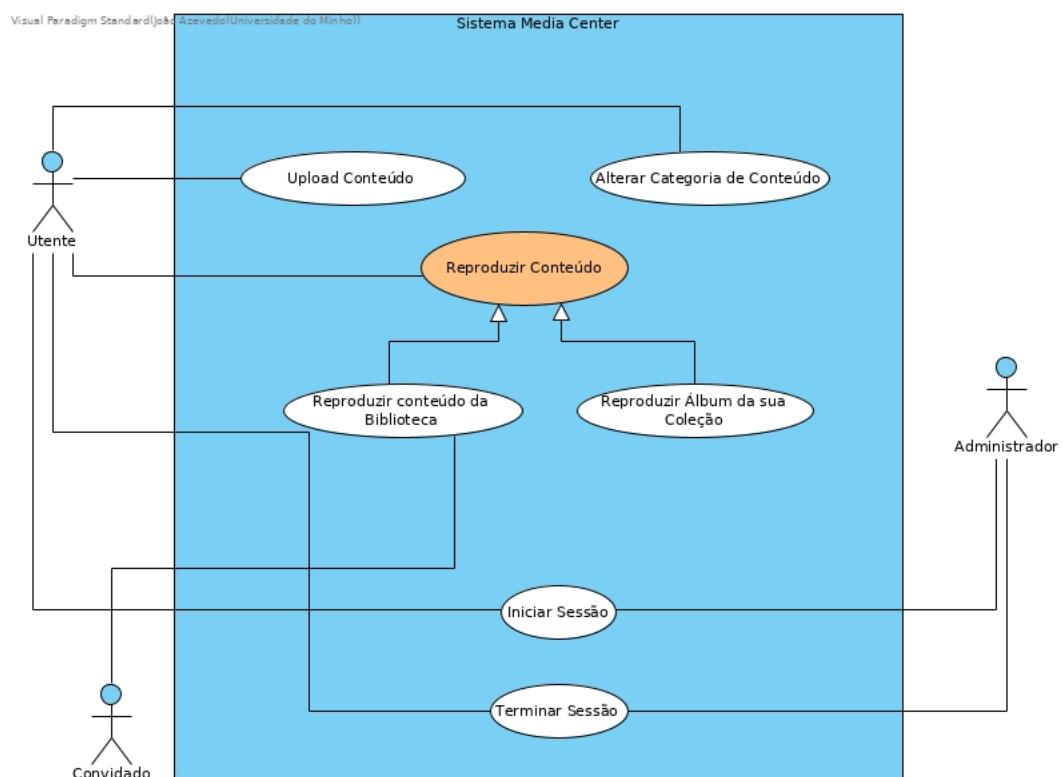


Figura 5.1: Diagrama de Use Case para a implementação final.

5.2 Diagrama de Classes

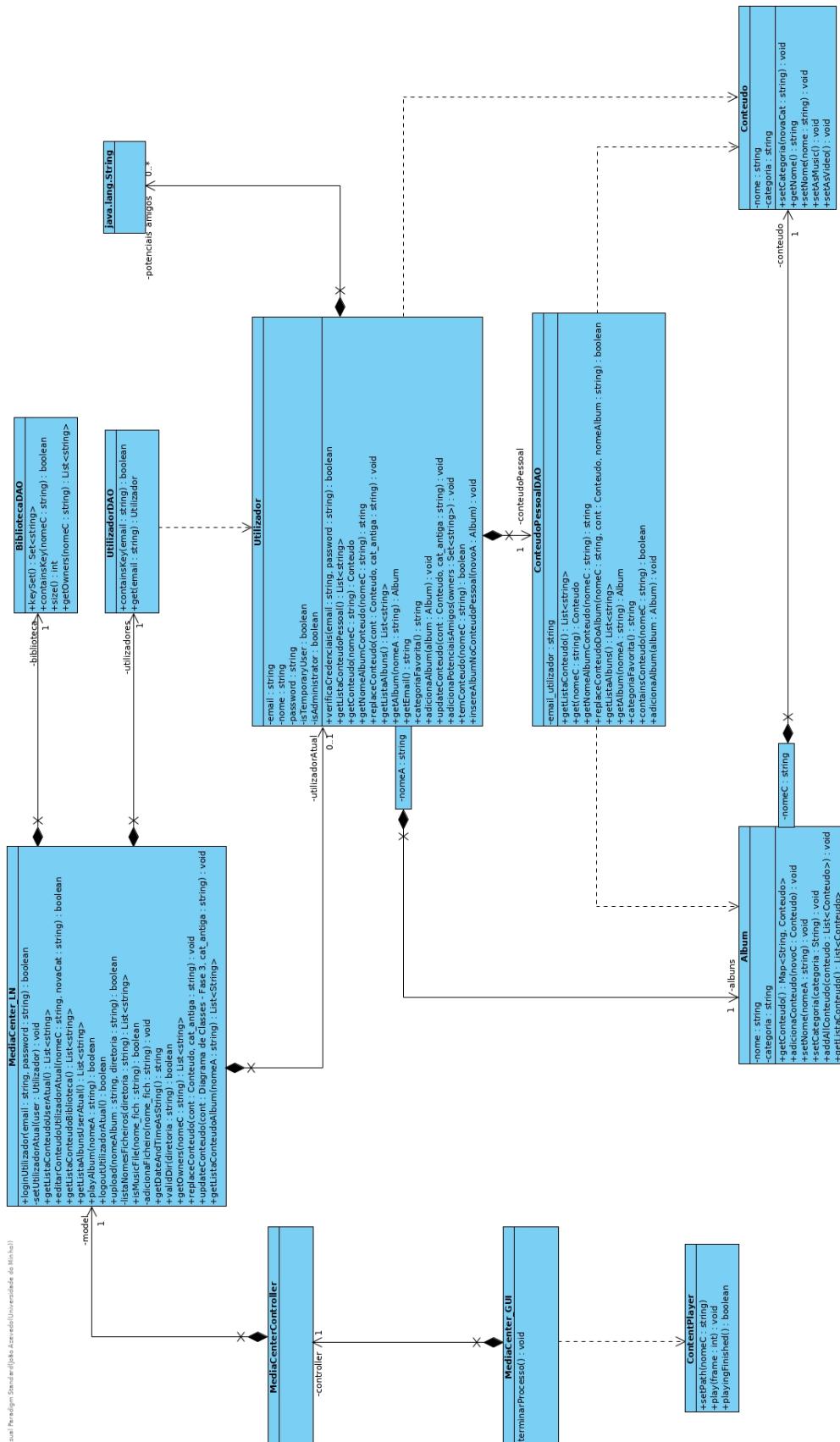


Figura 5.2: Diagrama de Classes para a implementação final.

5.3 Diagrama de Package

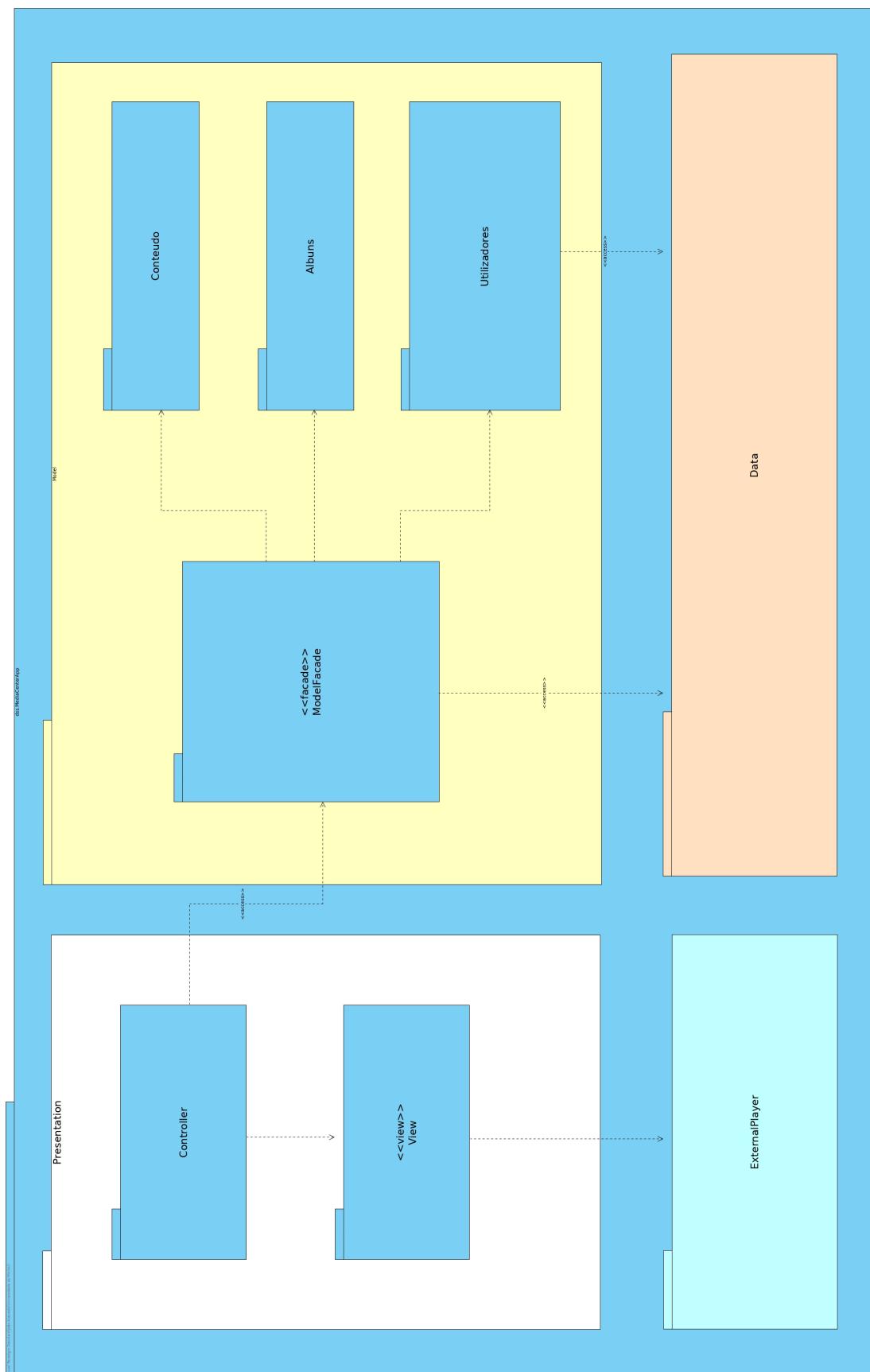


Figura 5.3: Diagrama de Package para a implementação final.

5.4 Packages e Classes

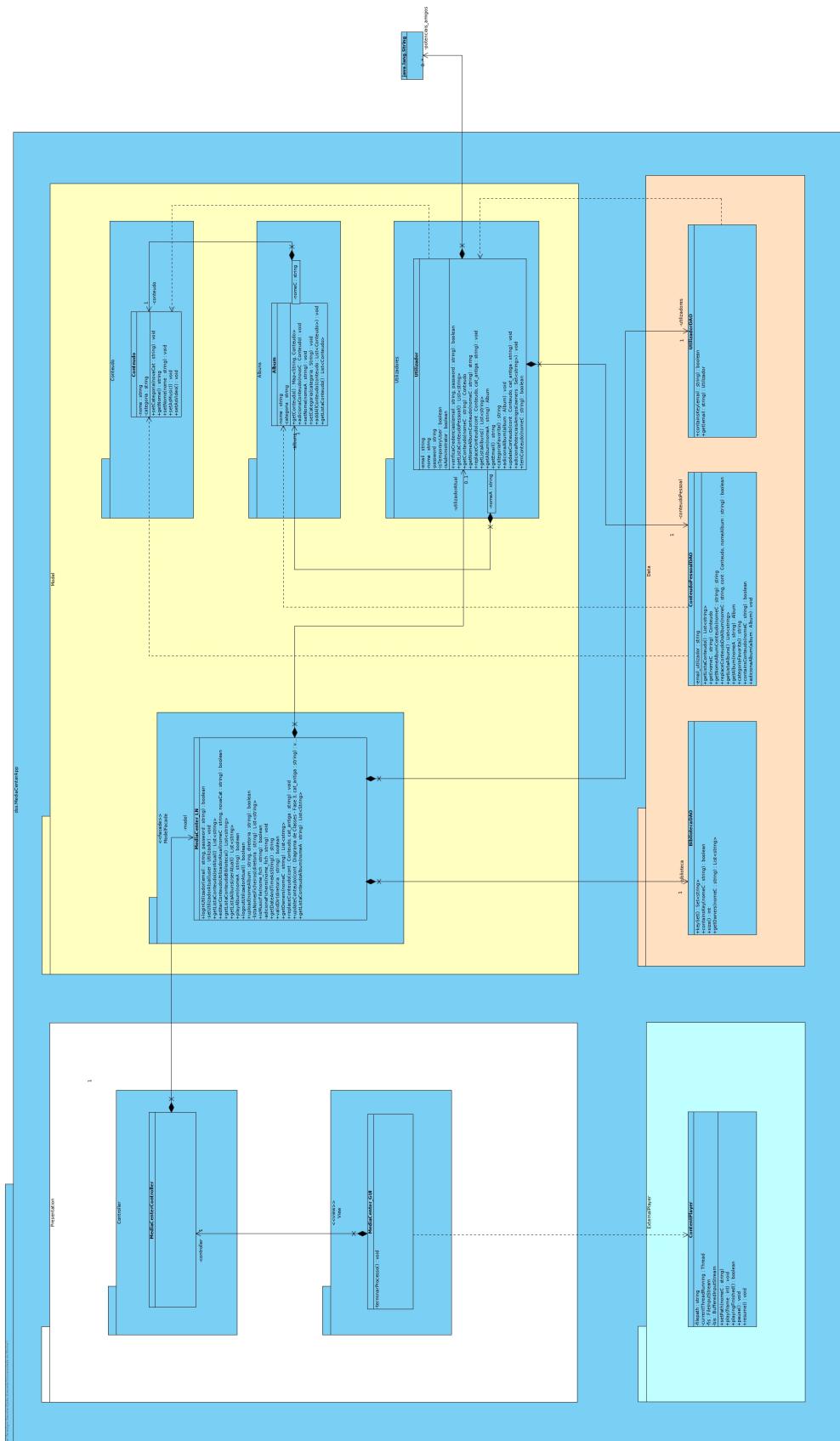


Figura 5.4: Diagrama de Package com classes para a implementação final.

Capítulo 6

Conclusões e trabalho futuro

Primeiramente é necessário clarificar alguns pontos referidos ao longo do projeto relativamente a decisões da estrutura que compõem este projeto que não foram consideradas mas que poderiam ter sido consideradas.

Por um lado, nem todas as considerações iniciais foram implementadas, como descritas em vários casos de uso iniciais, como **Editar Álbuns**, **Criar Álbuns**, que o grupo considera que seriam importantes para estruturar mais a liberdade de edição de conteúdo por parte do Utilizador. Outra consideração a referir é relativa à existência de Convidados, Utentes e Administradores que não se encontra totalmente estruturada neste projeto, i.e., um Administrador, visto que não estão implementados Use Cases para o mesmo, não possui vantagem na implementação final mas sim para uma implementação futura.

Por outro lado, os DAOs criados cumprem parcialmente o objetivo por de trás dos mesmos mas não implementam todos os métodos associados à interface **Map** respetiva uma vez que a sua implementação não traz vantagens para a implementação final, p. ex., métodos como `entrySet():Set<...>`, `clear()`,... Assim, o nosso projeto está longe de estar totalmente completo devido ao tempo gasto na estruturação e modelação que garantiu que o que já foi implementado cumpre os requisitos da aplicação pedida.

Assim, consideramos que este projeto