



# Desenvolvimento de Sistemas Software

Aula Teórica 10 (suplemento)

## Análise da Fase 1 do Trabalho



# Números

- 43 trabalhos entregues (+2) - 43 propostas de solução
  - 43 Modelos de Domínio
  - 97 Use Cases distintos
  - Centenas de Mockups
- Que proposta(s) escolher?
  - Maioria cumpre os mínimos
  - Mas... só alguns realmente *seleccionáveis*
  - Diversos aspectos a melhorar



# Os bons trabalhos

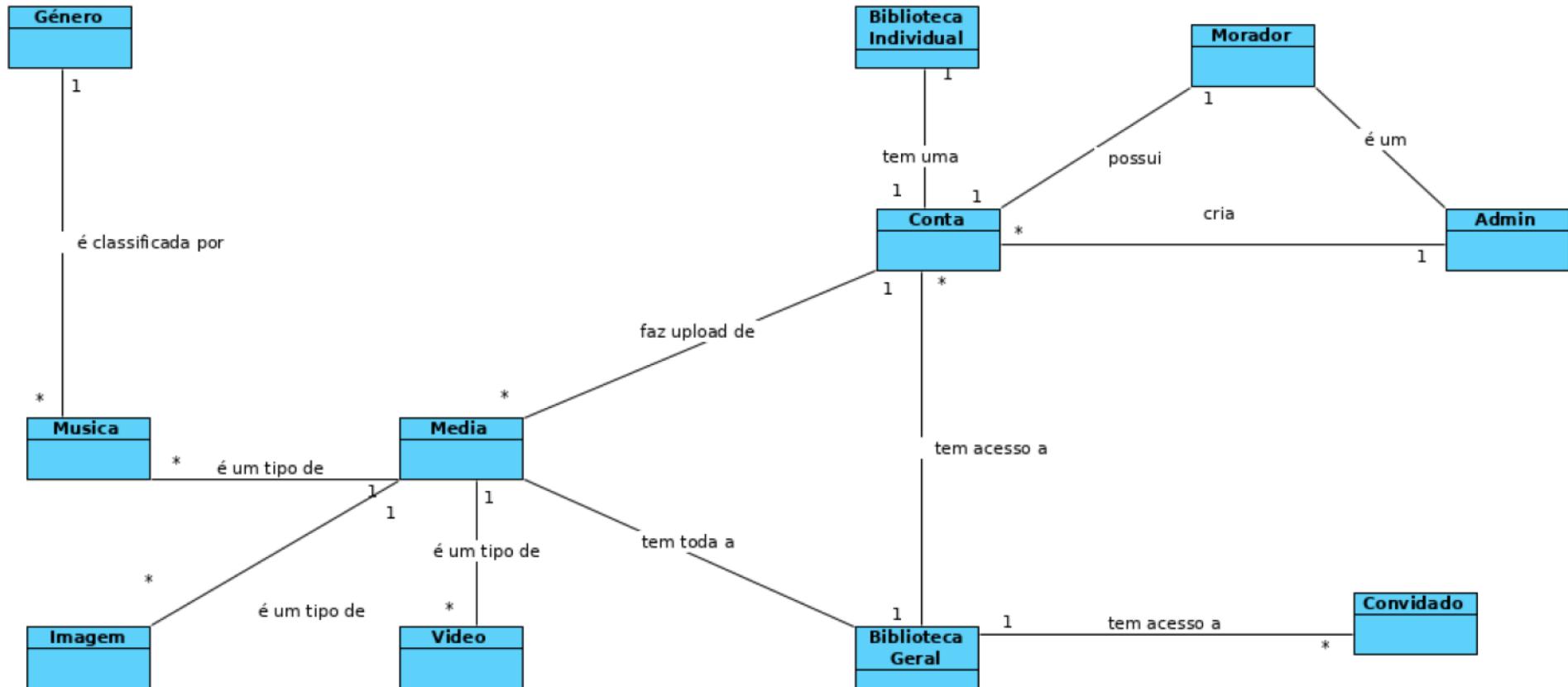
- Respeitam a(s) linguagem(ns)
  - Modelos de Domínio
  - Especificação dos Use Case
  - Diagramas UML (Use Case)
- São consistentes
  - Com os cenários
  - Internamente (Use Case vs. Modelo de Domínio, Use Cases vs Use Cases, Mockups vs Use Cases)
- São completos
  - Consideram a totalidade dos cenários propostos
  - e completam a funcionalidade de forma lógica



## Principais pontos a melhorar

- Utilização da(s) linguagem(ns)
  - Qualidade técnica deficiente causa *má impressão* e dificuldades de interpretação
- Inconsistências no(s) modelo(s)
- Modelos ambíguos
  - Difícil ou impossível perceber o que é proposto
- Funcionalidades em falta
  - Nem sempre os cenários apresentados estão plenamente considerados
- Funcionalidades que não pretendemos
  - Fim ultimo de desenvolver os modelos é discuti-los com o cliente

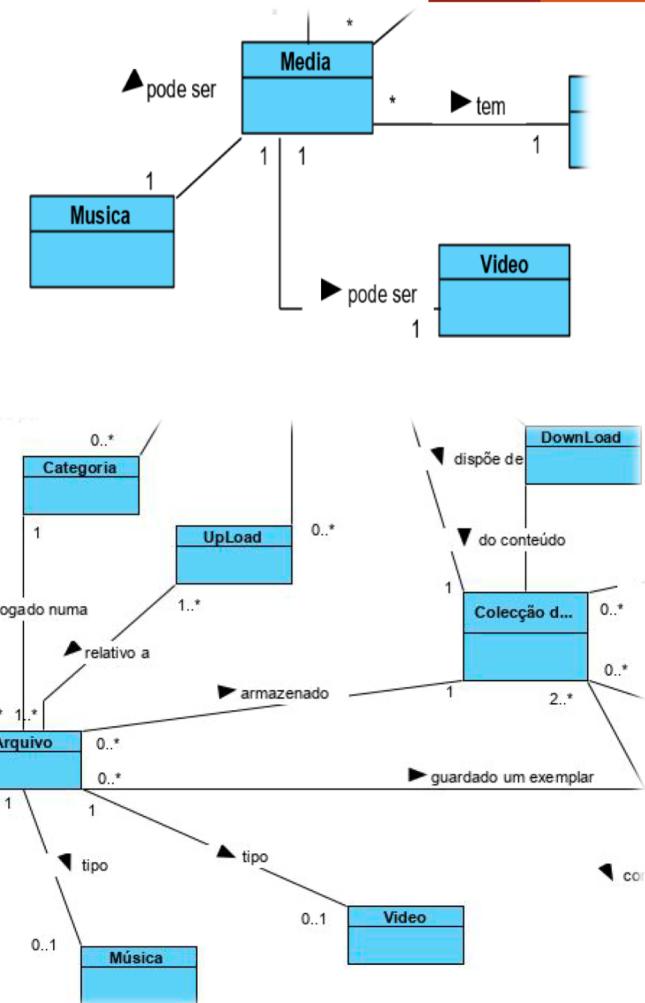
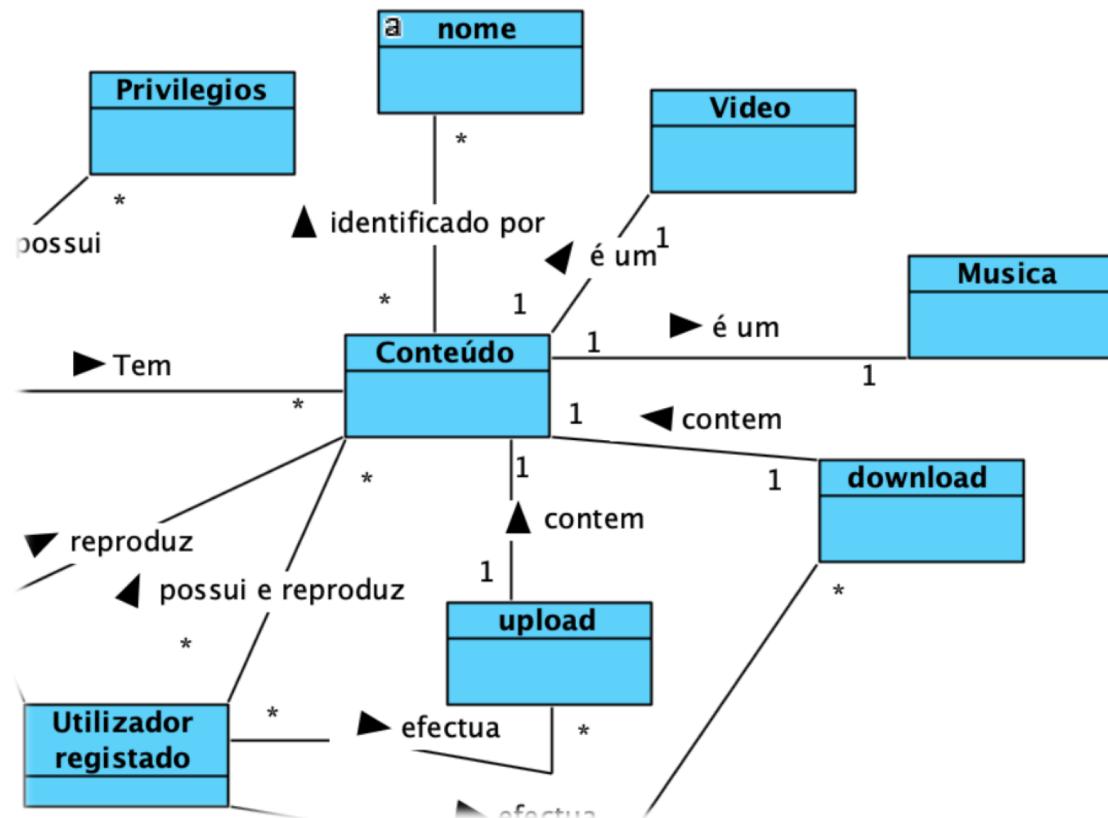
# Utilização da(s) linguagem(ns)



- Direcção de leitura das relações?!



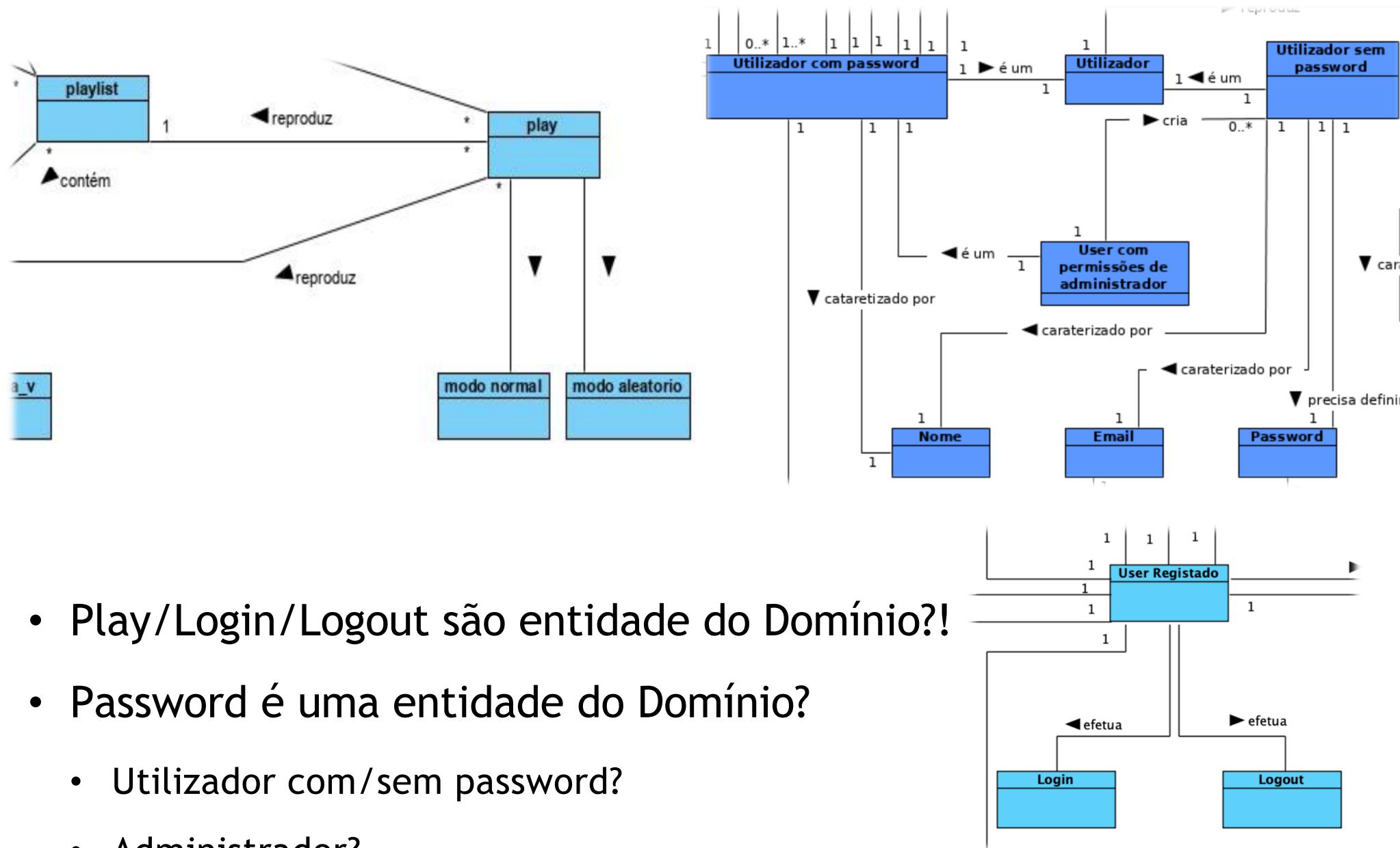
# Utilização da(s) linguagem(ns)



- Conteúdo é um Video e uma Música (em simultâneo)?!
  - Upload/Download contêm Conteúdo?!
  - Upload/Download são entidades do domínio?



# Utilização da(s) linguagem(ns)



- Play/Login/Logout são entidade do Domínio?!
- Password é uma entidade do Domínio?
  - Utilizador com/sem password?
  - Administrador?



# Utilização da(s) linguagem(ns)

Use Case:	Logout	
Descrição:	Permite o fim de sessão no sistema	
Actor:	User	
Pré condição:	User autenticado	
Pós condição:	User deixa de estar autenticado	
Cenário	Actor input	System response
Normal	1. User seleciona opção de logout	2. Mostra mensagem de logout

- Linguagem de especificação de Use Case?



# Utilização da(s) linguagem(ns)

- Use Case: Editar uma Playlist
  - Descrição: Editar as características de uma playlist.
  - Cenário: **O Ricardo quer editar o nome uma playlist.**
  - Pre-condições: Conta autenticada; **A playlist pertence ao utilizador.**
  - Pos-condições: A playlist é editada.
  - Fluxo Normal:
    1. O utilizador seleciona a opção "Playlists" a reproduzir.
    2. O sistema apresenta todas as playlists associadas a conta.
    3. O utilizador **seleciona o botão à frente da playlist que deseja editar.**
    4. O sistema apresenta as opções Apagar Playlist e Editar Playlist.
    5. O utilizador **seleciona a opção Editar Playlist.**
    6. O sistema apresenta as opções Editar Nome e Editar Descrição.
    7. O utilizador seleciona a opção Editar Nome.
    8. O sistema pede o novo nome da playlist.
    9. O utilizador inser novo nome.
    10. O sistema altera o nome e notifica o sucesso da operação.
  - Fluxo Alternativo: [Editar descrição]
    - 7.1 O utilizador seleciona a opção Editar Descrição.
    - 7.2 O sistema pede para inserir uma nova descrição.
    - 7.3 O utilizador insere a nova descrição.
    - 7.4 O sistema altera a descrição e notifica o sucesso da operação.
  - Fluxo de Exceção 1: [Não existe conteúdo]
    - 1.1 O sistema avisa o utilizador que não existe conteúdo na biblioteca.
- **Como se valida a pré-condição?**
- **Use Case não deve descrever a interface com o utilizador**
  - Interface surge para satisfazer requisites e não o contrário
- **Falta uma exceção no passo 5**
  - Actor escolhe Apagar Playlist...?

# Utilização da(s) linguagem(ns)

- Use Case: Classificar albuns
  - Pré-condição: O album não está classificado
  - Pós-condição: O album fica classificado a gosto do utilizador
  - Fluxo normal:
    1. O utilizador autentica a sua conta.
    2. O utilizador seleciona o album que pretende classificar.
    3. O utilizador classifica o album.
- Questões:
  - Como se valida a pré-condição?!
  - O Utilizador autentica a sua conta como?
    - Existe um Use Case Autenticar Conta: <<include>>?? (boa solução?)
  - Como fica o album classificado? (O Sistema não parece fazer nada)

# Utilização da(s) linguagem(ns)

<b>Use Case</b>	Aceitar pedido de amizade	
<b>Descrição</b>	Um utilizador que tenha recebido um pedido de amizade aceita o mesmo.	
<b>Pré-Condição</b>	O utilizador tem sessão iniciada.	
<b>Pós-Condição</b>	O Utilizador que enviou o pedido de amizade é adicionado à lista de amigos.	
<b>Comportamento Normal</b>	<b>Autor</b> <b>1</b> Utilizador aceita pedido de amizade.  <b>2</b>  <b>3</b>	<b>Sistema</b>    <b>Sistema regista que o pedido foi aceite.</b>  <b>Sistema adiciona o outro utilizador à lista de amigos.</b>
<b>Alternativo 1 [Utilizador não aceita pedido de amizade] (passo 1)</b>	<b>Autor</b> <b>1.1</b> Utilizador não aceita pedido de amizade.  <b>2.1</b>  <b>3.1</b>	<b>Sistema</b>    <b>Sistema regista que o pedido não foi aceite.</b>  <b>Sistema envia notificação ao outro utilizador de que o pedido não foi aceite.</b>

- Alternativa ou excepção?!

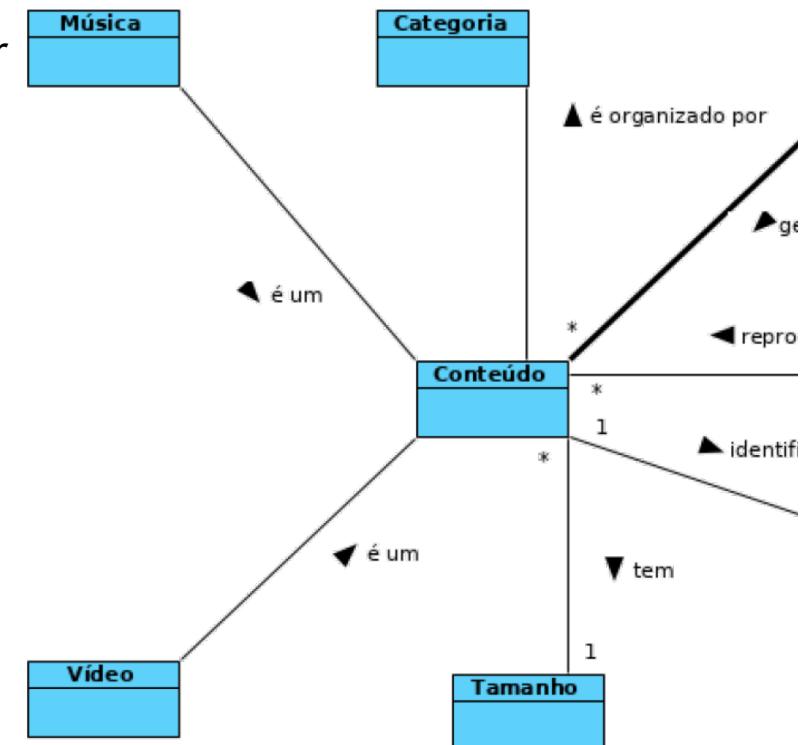


## Principais pontos a melhorar

- Utilização da(s) linguagem(ns)
  - Qualidade técnica deficiente causa *má impressão* e dificuldades de interpretação
- Inconsistências no(s) modelo(s)
- Modelos ambíguos
  - Difícil ou impossível perceber o que é proposto
- Funcionalidades em falta
  - Nem sempre os cenários apresentados estão plenamente considerados
- Funcionalidades que não pretendemos
  - Fim ultimo de desenvolver os modelos é discuti-los com o cliente

# Inconsistências no(s) modelo(s)

- Use Case: **Classificar** conteúdo
  - Descrição: O utilizador quer classificar um certo conteúdo da sua biblioteca
  - Cenário: O Manuel quer classificar uma musica que adicionou ao Media Center
  - Pré-condição: O utilizador está logged in
  - Pós-condição: O conteúdo é **classificado pelo utilizador**
  - Fluxo normal:
    1. O utilizador escolhe o conteúdo que quer classificar
    2. O utilizador escolhe a classificação do conteúdo
    3. O sistema define a classificação do conteúdo
- **O que é classificar conteúdo?**
  - Modelo de Domínio não esclarece o que é a classificação
- Ainda
  - **Pós-condição deve ser sobre o estado do Sistema**
  - “O Manuel quer...” não é verdadeiramente um cenário





## Principais pontos a melhorar

- Utilização da(s) linguagem(ns)
  - Qualidade técnica deficiente causa *má impressão* e dificuldades de interpretação
- Inconsistências no(s) modelo(s)
- Modelos ambíguos
  - Difícil ou impossível perceber o que é proposto
- Funcionalidades em falta
  - Nem sempre os cenários apresentados estão plenamente considerados
- Funcionalidades que não pretendemos
  - Fim ultimo de desenvolver os modelos é discuti-los com o cliente

# Modelos ambíguos

- Use Case: Adicionar Conteúdo à playlist.
  - Descrição: Adiciona conteúdo a uma playlist.
  - Cenários: O Manuel pretende adicionar uma musica a uma playlist.
  - Pré-condição: Sessão iniciada como membro registado, **ser o criador da playlist**.
  - Pós-condição: Conteúdo adicionado a playlist.
  - Fluxo Normal:
    1. Utilizador seleciona a opção de adicionar conteúdo à playlist.
    2. **Sistema pergunta qual a playlist**
    3. Utilizador especifica qual a playlist.
    4. Sistema valida se o conteúdo é repetido.
    5. Sistema adiciona o conteúdo a playlist.
  - Fluxo Alternativo 1: [Conteúdo já se encontra na playlist] (**passo 2**)
    - 2.1. Sistema alerta que o conteúdo já se encontra na playlist.
  - Para além dos problemas com a utilização da linguagem... **Que conteúdo é adicionado?**

# Modelos ambíguos

Use Case:	Adição de conteúdo	
Ator:	Utilizador registado	
Pré-condição:	Ator tem sessão iniciada com a sua conta	
Pós-condição:	Sistema de bibliotecas alterado	
	Autor	Sistema
Fluxo Normal	1. Seleciona biblioteca	
	2. Seleciona conteúdo	
		3. Apresenta opções relacionadas com o conteúdo selecionado
	4. Seleciona a opção adicionar a playlist já existente	
		5. Apresenta playlist onde o conteúdo pode ser adicionado
	6. Escolhe playlist	
		6. Adiciona conteúdo à playlist indicada
		7. Mensagem de sucesso

Use Case:	Alterar género	
Ator:	Utilizador registado	
Pré-condição:	Ator tem sessão iniciada com a sua conta	
Pós-condição:	Género do media selecionado foi alterado	
	Autor	Sistema
Fluxo Normal	1. Seleciona conteúdo a alterar	
		2. Verifica que o conteúdo selecionado foi carregado pelo utilizador
	3. Insere género a alterar	
		4. Altera género
Fluxo de Exceção 1	2.1 Verifica que o conteúdo selecionado não pertence ao utilizador	
	2.2 Mensagem de erro	

- Adição de conteúdo... onde?!
  - O que é a biblioteca? (não está no Modelo de Domínio)
  - O que é conteúdo? (não está no Modelo de Domínio) - Media?
- Alterar género... de quê?
  - O que é o género? (não está no Modelo de Domínio) - Categoria (Musical/de Video)?
  - Fluxo de exceção: assumindo que é categoria, consistente com cenários fornecidos? (Modelo de Domínio não esclarece relação de pertença)



## Principais pontos a melhorar

- Utilização da(s) linguagem(ns)
  - Qualidade técnica deficiente causa *má impressão* e dificuldades de interpretação
- Inconsistências no(s) modelo(s)
- Modelos ambíguos
  - Difícil ou impossível perceber o que é proposto
- Funcionalidades em falta
  - Nem sempre os cenários apresentados estão plenamente considerados
- Funcionalidades que não pretendemos
  - Fim ultimo de desenvolver os modelos é discuti-los com o cliente



# Funcionalidades em falta

- Use Cases mais comuns
  1. Registar utilizador - 98%
  2. Iniciar sessão - 95%
  3. Fazer upload de conteúdo - 98%
  4. Fazer download de conteúdo - 91%
  5. Reproduzir conteúdo - 91%
  6. Alterar categoria de conteúdo - 81%
  7. Terminar sessão - 74%
  8. Eliminar utilizador - 44%
  9. Adicionar amigo - 37%
  10. Aceder como convidado - 35%
  11. Alterar password - 33%
  12. Remover conteúdo - 33%

- Nem todos incluiram todos os Use Case necessários aos cenários fornecidos. E.g.
  - Alterar categoria de conteúdo no Cenário 1
  - Terminar Sessão/logout no Cenário 2



## Principais pontos a melhorar

- Utilização da(s) linguagem(ns)
  - Qualidade técnica deficiente causa *má impressão* e dificuldades de interpretação
- Inconsistências no(s) modelo(s)
- Modelos ambíguos
  - Difícil ou impossível perceber o que é proposto
- Funcionalidades em falta
  - Nem sempre os cenários apresentados estão plenamente considerados
- **Funcionalidades que não pretendemos**
  - Fim último de desenvolver os modelos é discuti-los com o cliente



# Funcionalidades que (não) pretendemos

- Um total de 97 Use Cases
- Em alguns casos, granularidade demasiado *fina*
  - Editar conta
    - Alterar username
    - Alterar e-mail
    - Alterar foto de perfil
    - Alterar Password como Administrador (?)
    - Alterar privilégios (?)
  - Alterar categoria de conteúdo
    - Alterar categoria de música
    - Alterar categoria de video
- Reproduzir Conteúdo
  - Parar Conteúdo
  - Avançar Conteúdo
  - Retroceder Conteúdo
  - Ativar Modo Shuffle
  - Desativar Modo Shuffle
  - Alterar volume
- Noutros casos, Use Cases desadequados
  - [Fazer] Pedido de criação de conta / Solicitar registo
  - Enviar mensagem
  - Ligar
  - Desligar



# Funcionalidades que (não) pretendemos

- Lista de amigos - duas abordagens
  - Adição imediata à lista de amigos
    - Use Case: Adicionar amigo
  - Sistema de convite e aceitação
    - Use Case: Fazer Convite
    - Use Case: Responder a Convite
  - Adoptar segunda versão!
- Acesso dos convidados - duas abordagens
  - Processo de acesso (login) específico
  - Acesso imediato (mas limitado) ao conteúdo
  - Adoptar segunda versão!



# Lista de Use Case a Considerar para a Fase 2

- Utilizadores
  1. Registar utilizador
  2. Iniciar sessão
  3. Terminar sessão
  4. Editar utilizador (inclui password)
  5. Eliminar utilizador
- Media
  6. Fazer upload de conteúdo
  7. Alterar categoria de conteúdo
  8. Remover conteúdo
  9. Reproduzir conteúdo (não é necessário ter sessão iniciada!)
- Playlists
  10. Criar playlist (devem ser consideradas diferentes possibilidades: aleatória, por género, por artista, ...)