**Caso de Uso 1 – Adicionar vídeo a biblioteca**

* Operação: adicionaVideo()
* Pré-condições: Existe c: Curador autenticado.
* Pós-condições: É criado v: Video, em que v é o vídeo corrente.
* Operação: ehClipe(resposta)
* Pré-condições: Existe c: Curador autenticado. Existe v: Video corrente.
* Pós-condições: Se resposta == true, então é criado vc: Clip, em que vc é o clipe corrente,

e chama-se a operação indicaDuracao(duracao).

Se resposta == false, então é criado vs: Stream, em que vs é stream corrente.

* Operação: indicaConteudo(titulo, endereco)
* Pré-condições: Exista c: Curador autenticado. Existe v: Video corrente.
* Pós-condições: Se v.titulo = titulo e v.endereco = endereco, v está em condições de ser

adicionado.

* Operação: indicaDuracao(duracao)
* Pré-condições: Existe c: Curador autenticado. Existe vc: Clip corrente.
* Pós-condições: Seja vc.duracao = duracao, em que vc.duracao corresponde à duração

do clipe corrente.

* Operação: associaHashtag(tag)
* Pré-condições: Existe v: Video corrente.
* Pós-condições: Seja h: Hashtag, tal que h.nome = tag, ou é criado h: Hashtag com

h.nome = tag. h é associado a v.

* Operação: indicaPrivacidade(privacidade)
* Pré-condições: Exista c: Curador autenticado. Existe v: Video corrente.
* Pós-condições: Seja v.privacidade = privacidade.

Se v.privacidade == privado, seja bp: BibliotecaPessoal, tal que bp está

associada a ca, e bp e v estão associados.

Se v.privacidade == publico, seja bg: BibliotecaGeral, tal que bg está

associada a ca, e bg e v estão associados.

v é adicionado ao catálogo de vídeos.

É devolvido v.codigo pelo sistema.

**Caso de Uso 2 – Criar Playlist Manualmente**

* Operação: criarPlaylist(nome)
* Pré-condições: Existe c: Curador autenticado corrente
* Pós-condições: É criada p: Playlist com p.nome = nome e p.codigo gerado pelo sistema.

p é a playlist corrente.

* Operação: mostrarPlaylist()
* Pré-condições: Existe c: Curador autenticado corrente. Existe p: Playlist corrente.
* Pós-condições: Seja v: Video um vídeo associado a p. No ecrã mostra-se v.nome e

v.hora\_Inicio para todo v associado a p.

* Operação: indicaHastag(tag)
* Pré-condições: Exista c: Curador autenticado corrente. Existe p: Playlist corrente.
* Pós-condições: Seja h: Hashtag tal que h.nome = tag. Seja v: video, seja bg: BibliotecaGeral

e seja bp: BibliotecaPessoal tal que bp e c estão associados.

Se v estiver associado a h e estiver associado ou a bg ou a bp, está em

condições de ser mostrado.

É mostrado v.codigo e v.nome para todo v que satisfaça estas condições.

Caso contrário, envia-se uma mensagem de tag desconhecida e chama-se a

operação verTop10(resposta).

* Operação: introduzirVideo(código)
* Pré-condições: Existe c: Curador autenticado corrente. Existe p: Playlist corrente.
* Pós-condições: Seja v: Video, seja h: Hashtag tal que h e v estão associados.

Seja bg: BibliotecaGeral e seja bp: BibliotecaPessoal tal que bp e c estão

associados.

Se v.codigo == codigo , v for clipe e v estiver associado ou a bg ou a bp, v

está em condições de ser associado a p, e p está em condições de ser

associado a todo h associado a v.

Se v.codigo == codigo, v for stream e v estiver associado ou a bg ou a bp,

chama-se a operação indicaDuracao(tempo).

Caso contrário envia-se uma mensagem de erro e chama-se a operação

introduzirVideo(codigo).

* Operação: indicaDuracao(tempo)
* Pré-condições: Existe c: Curador autenticado corrente. Existe p: Playlist corrente.

Existe s: Stream corrente.

* Pós-condições: Seja co: Contem tal que s está associado a p por co, com

co.duracao\_transmissao = tempo.

Seja h: Hashtag tal que h está associado a s. p fica associado a h, para

todo h associado a s.

* Operação: finalizarPlaylist()
* Pré-condições: Exista c: Curador autenticado corrente. Existe p: Playlist corrente
* Pós-condições: p deixa de ser a playlist corrente.
* Operação: verTop10(resposta)
* Pré-condições: Existe c: Curador autenticado corrente
* Pós-condições: Se resposta == true, mostra-se os 10 v: Video que têm maior v.visualizacao.

Se não existirem 10, mostram-se todos v.

Os v são ordenados de maior v.visualizacao para menor v.visualizacao.