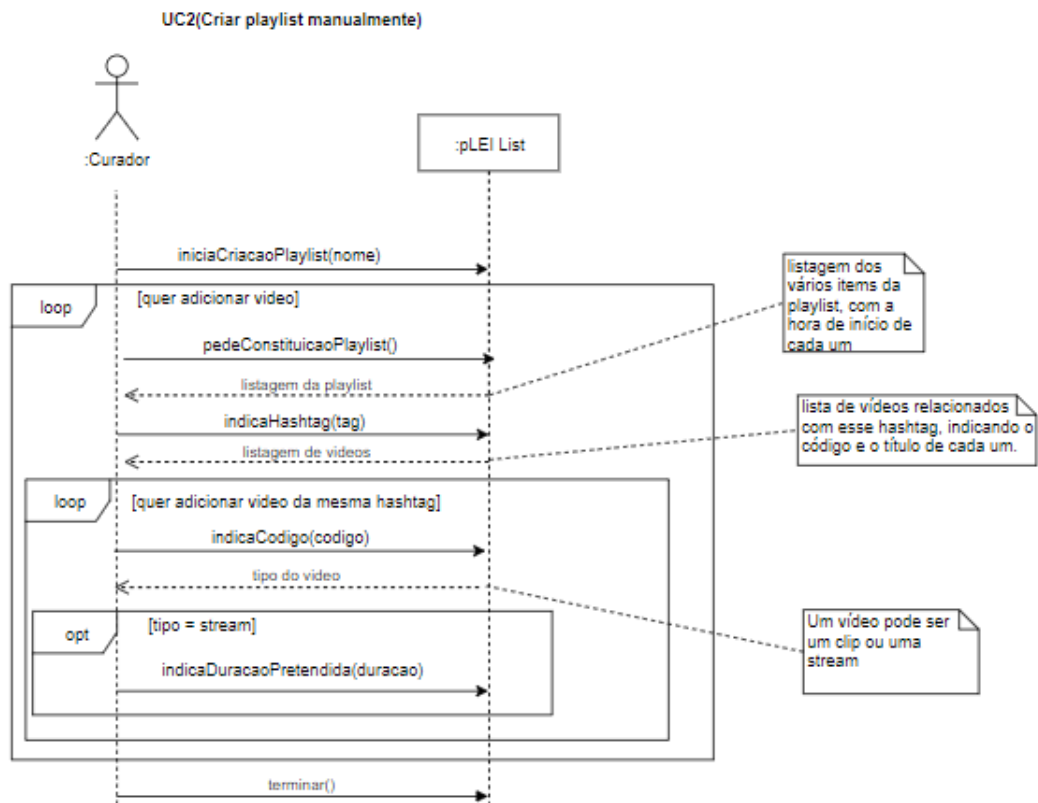
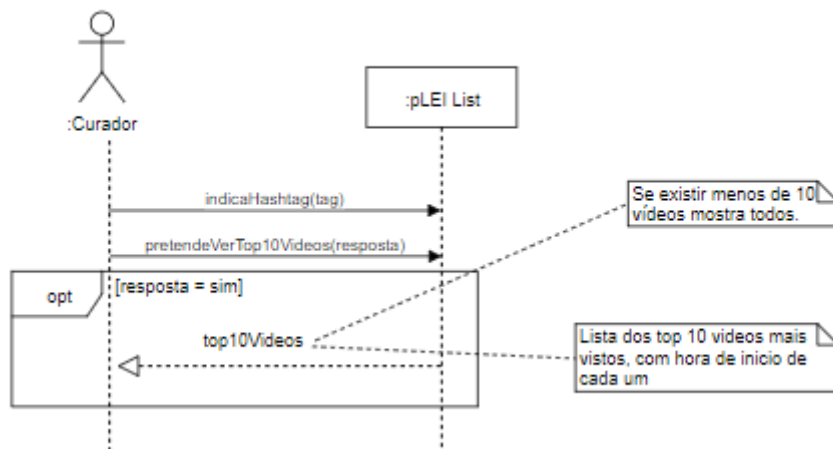


Ext 5a – Não se representa, é simples.

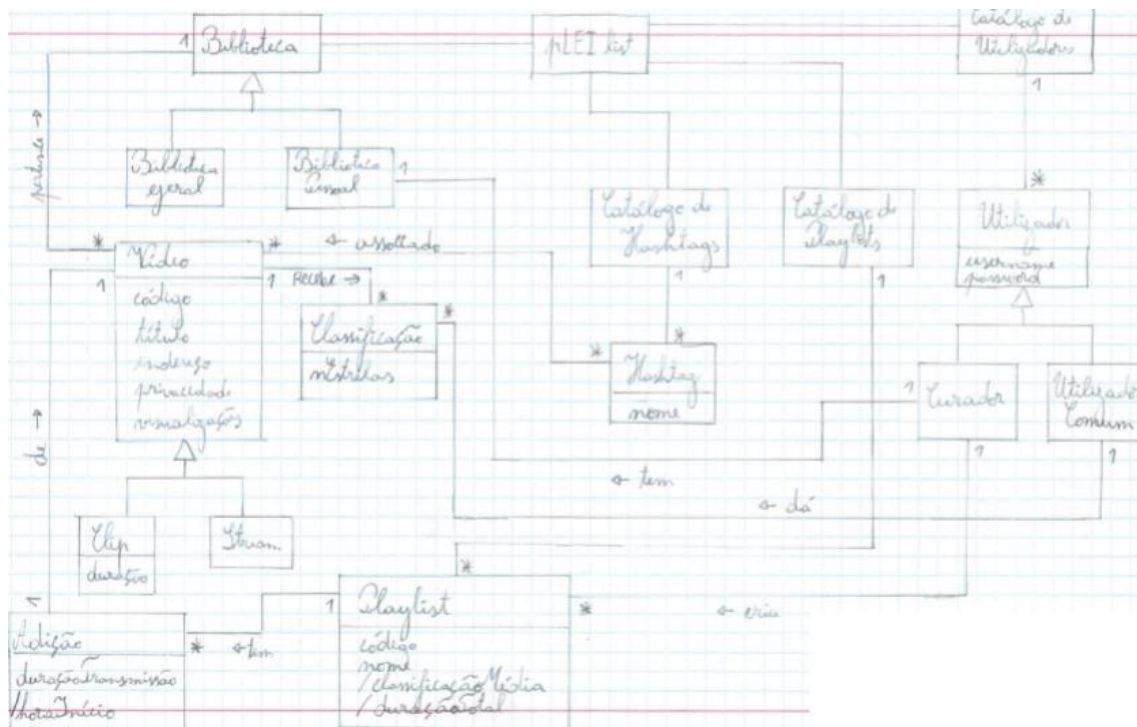
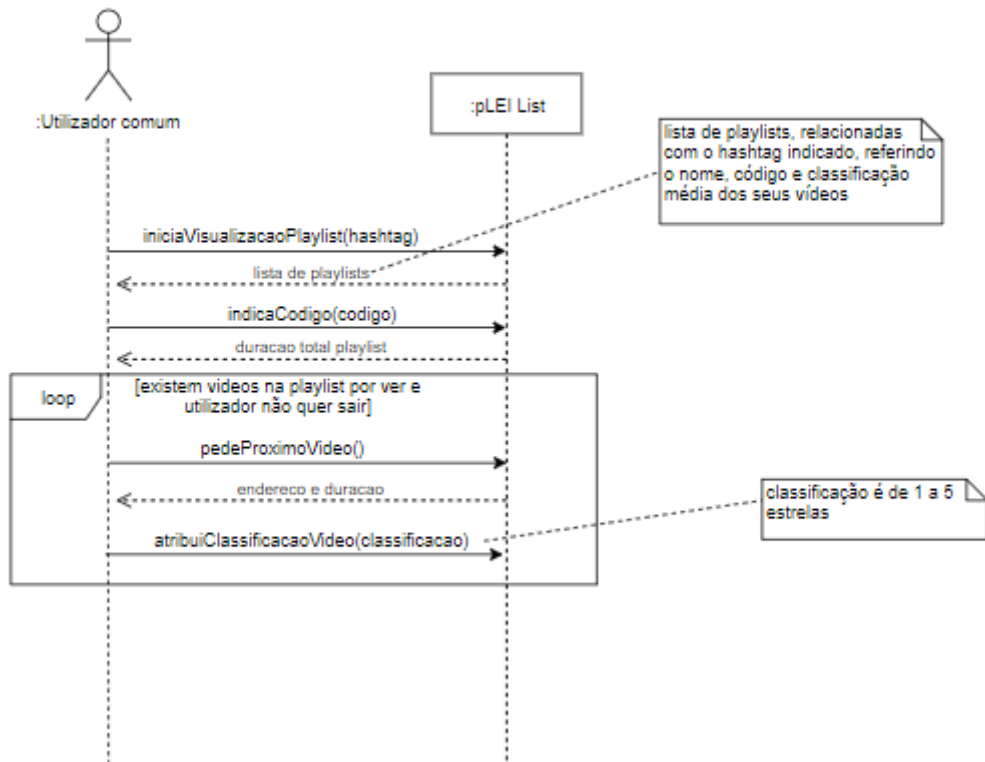


Ext 7a



Ext 8a – Não se representa, é simples.

UC4(Visualizar uma Playlist)



Contratos:

UC1-Adicionar vídeo à biblioteca

Operação: iniciaAdicaoVideo()

Pré-condições: Existe c:Curador autenticado e corrente.

Pós-condições: N/A

Operação: indicaTipoVideo(tipo)

Pré-condições: Existe c:Curador autenticado e corrente .

Pós-condições: É criado v:Clip ou v:Stream consoante tipo = clip ou tipo = stream, respetivamente.

Operação: identificaVideo(titulo, endereco)

Pré-condições: Existe c:Curador autenticado e corrente e v:Video corrente.

Pós-condições: v.titulo = titulo e v.endereco = endereço.

Operação: indicaDuracao(duracao)

Pré-condições: Existe c:Curador autenticado e corrente e v:Clip corrente.

Pós-condições: v.duracao = duracao.

Operação: associaHashtag(tag)

Pré-condições: Existe c:Curador autenticado e corrente e v:Video corrente.

Pós-condições: Seja h:Hashtag tal que h.nome = tag ou é criado h:Hashtag com h.nome = tag, é associado h a v.

Operação: indicaPrivacidade(privacidade)

Pré-condições: Existe c:Curador autenticado e corrente e v:Video corrente.

Pós-condições: v.privacidade = privacidade, v.codigo = novo codigo, v é adicionado à biblioteca geral ou pessoal consoante a privacidade = publica ou privacidade = privada, respetivamente.

UC2-Criar playlist manualmente

Operação: iniciaCriacaoPlaylist(nome)

Pré-condições: Existe c:Curador autenticado e corrente.

Pós-condições: É criada p:Playlist tal que p é a playlist corrente, p.nome = nome e p.codigo = novo codigo.

Operação: pedeConstituicaoPlaylist()

Pré-condições: Existe c:Curador autenticado e corrente e p:Playlist corrente.

Pós-condições: N/A

Operação: indicaHashtag(tag)

Pré-condições: Existe c:Curador autenticado e corrente e p:Playlist corrente.

Pós-condições: N/A

Operação: indicaCodigo(codigo)

Pré-condições: Existe c:Curador autenticado e corrente e p:Playlist corrente.

Pós-condições: Seja v:Video tal que v.codigo = codigo, v é o vídeo corrente.

É criado ad:Adicao tal que ad é a adicao corrente.

Se v for um clip (v:Clip) $ad.duracaoTransmissao = v.duracao$, $ad.horaInicio = p.duracaoTotal$, ad é associado a v e a p e $p.duracaoTotal = p.duracaoTotal + ad.duracaoTransmissao$.

Operação: indicaDuracaoPretendida(duracao)

Pré-condições: Existe c:Curador autenticado e corrente, p:Playlist corrente, v:Stream corrente e ad:Adicao corrente.

Pós-condições: $ad.duracaoTransmissao = duracao$, $ad.horaInicio = p.duracaoTotal$ e ad é associado a v e a p.

$p.duracaoTotal = p.duracaoTotal + ad.duracaoTransmissao$.

Operação: terminar()

Pré-condições: Existe c:Curador autenticado e corrente e p:Playlist corrente.

Pós-condições: N/A

Ext 7a

Operação: indicaHashtag(tag)

Pré-condições: Existe c:Curador autenticado e corrente e p:Playlist corrente.

Pós-condições: N/A

Operação: pretendeVerTop10Videos(resposta)

Pré-condições: Existe c:Curador autenticado e corrente e p:Playlist corrente.

Pós-condições: N/A

UC4-Visualizar uma playlist

Operação: iniciaVisualizacaoPlaylist(hashtag)

Pré-condições: Existe u:UtilizadorComum autenticado e corrente.

Pós-condições: N/A

Operação: indicaCodigo(codigo)

Pré-condições: Existe u:UtilizadorComum autenticado e corrente.

Pós-condições: Seja p:Playlist tal que p.codigo = codigo, p é a playlist corrente.

Operação: pedeProximoVideo()

Pré-condições: Existe u:UtilizadorComum autenticado e corrente e p:Playlist corrente.

Pós-condições: O próximo v:Video passa a ser o video corrente.

Operação: atribuiClassificacaoVideo(classificacao)

Pré-condições: Existe u:UtilizadorComum autenticado e corrente, p:Playlist corrente e v:Video corrente.

Pós-condições: Se já existir cl:Classificacao associada a u e a v, cl.nEstrelas = classificacao e cl é a classificação corrente. Se ainda não existir, é criado cl:Classificacao tal que cl é a classificação corrente, cl.nEstrelas = classificacao e cl é associado a u e a v.

p.classificacaoMedia é atualizada.