

Projecte Final de Carrera
Control físicament realista d'un robot bíped
virtual

Autor: Miquel Vergés Vilarrubia

Director: Joaquim Casulleras Ambrós

Departament: Física i Enginyeria Nuclear

Facultat: Facultat d'Informàtica de Barcelona

Universitat: Universitat Politècnica de Catalunya

Agraïments

Aquest projecte no hagués estat possible gràcies a moltes persones del meu voltant. Els hi dedico la primera plana del projecte com a mostre del meu agraïment.

Primer de tot voldria agrair al meu tutor Joaquim Casulleras Ambrós tot l'esforç i el temps que ha dedicat en mi perquè aconseguís els objectius que ens havíem proposat, sense ell aquest projecte no hagués estat possible. Gràcies per la paciència que has tingut amb mi.

Agrair també a Lluís Ametller Congost i Marta Jiménez Castells, President i Vocal del meu projecte, moltes gràcies per la paciència que han tingut durant 2 anys.

Moltes gràcies a la meva parella i a tota la meva família pel gran punt de suport que han estat, de vosaltres he pogut treure la constància que requeria un projecte d'aquest tipus. Gràcies per animar-me en els fracassos i està al meu costat durant tot el projecte.

Finalment, un agraïment per aquells que s'han interessat pel projecte, amics, companys i coneguts... i pels que han ajudat al desenvolupament del mateix, ja sigui donant idees o senzillament, donant la seva opinió personal.

Índex

Agraïments	3
1 Requisits del sistema	25
1.1 Propòsit del programa	25
1.2 Motivacions	25
1.3 Descripció	26
1.4 Funcionalitats de l'aplicació	27
1.5 Arquitectura de l'aplicació	28
1.6 Requeriments	28
1.6.1 QT Creator	29
1.6.2 OpenGL	30
1.6.3 Vídeo	30
1.6.4 Sistema de fitxers	30
1.7 Estudi de la Física de cossos articulats	30
1.7.1 Matrius homogènies	30
1.7.2 Operacions de composició amb matrius homogènies	32
1.7.3 Anàlisi i descripció de sistemes articulats	32
1.7.4 Modelització de sistemes articulats. Taula D-H	35
1.7.5 Obtenció dels paràmetres de la taula D-H	40
1.7.6 Descripció del problema i solució	47
Pas previ: Generalització taula D-H	47
Pas 1: Trajectòria descriptiva d'entrada i editada per l'usuari	50
Pas 2: Càlcul del moment angular i de la força de contacte	56
Pas 2: Càlcul del moviment mitjançant el Jacobià i les modelitzacions anteriors	61
Pas 3: Càlcul del moviment complet. Algoritme final	64
Pas 4: Càlcul dels paràmetres físics que estudien el moviment	66

2	Anàlisi de l'aplicació	69
2.1	Actors	69
2.2	Funcionalitat 1: Generació de models	69
2.2.1	Necessitat	69
2.2.2	Cassos d'ús	70
	Cas d'ús 1.1: Crear model	70
	Cas d'ús 1.2: Afegir Conjunt de cadenes	70
	Cas d'ús 1.3: Afegir cadena al model	70
	Cas d'ús 1.4: Afegir un "Link" al model	70
	Cas d'ús 1.5: Afegir una massa al model	70
	Cas d'ús 1.6: Afegir un recolzador al model	71
2.3	Funcionalitat 2: Modificació de models	71
2.3.1	Necessitat	71
2.3.2	Cassos d'ús	71
	Cas d'ús 2.1: Eliminar model	71
	Cas d'ús 2.2: Moure model	71
	Cas d'ús 2.3: Rotar model	72
	Cas d'ús 2.4: Eliminar cadena	72
	Cas d'ús 2.5: Moure cadena	72
	Cas d'ús 2.6: Rotar cadena	73
	Cas d'ús 2.7: Eliminar "Conjunt de cadenes"	73
	Cas d'ús 2.8: Eliminar "Link"	73
	Cas d'ús 2.9: Rotar "Link"	74
	Cas d'ús 2.10: Modificar la llargada del "Link"	74
	Cas d'ús 2.11: Eliminar Massa	74
	Cas d'ús 2.12: Rotar Massa	74
	Cas d'ús 2.13: Rotar Recolzador	75
	Cas d'ús 2.14: Eliminar Recolzador	75
	Cas d'ús 2.15: Modificar la llargada d'un Recolzador	75
	Cas d'ús 2.16: Modificar l'amplada d'un Recolzador	75
2.4	Funcionalitat 3: Generació d'escenes	75
2.4.1	Necessitat	75
2.4.2	Cassos d'ús	76
	Cas d'ús 3.1: Afegir un objecte	76
2.5	Funcionalitat 4: Modificació d'escenes	76
2.5.1	Necessitat	76
2.5.2	Cassos d'ús	76
	Cas d'ús 4.1: Rotar objecte	76
	Cas d'ús 4.2: Escalar objecte	77
	Cas d'ús 4.3: Moure objecte	77
	Cas d'ús 4.4: Duplicar objecte	77

	Cas d'ús 4.5: Eliminar objecte	78
	Cas d'ús 4.6: Modificar paràmetre objecte	78
2.6	Funcionalitat 5: Generació d'animacions	78
2.6.1	Necessitat	78
2.6.2	Cassos d'ús	78
	Cas d'ús 5.1: Afegir punt clau	78
	Cas d'ús 5.2: Modificar paràmetre del model articulat	79
	Cas d'ús 5.3: Afegir punt de contacte	79
	Cas d'ús 5.4: Assignar instant de temps	79
	Cas d'ús 5.5: Generar animació	80
2.7	Funcionalitat 6: Modificació d'animacions	80
2.7.1	Necessitat	80
2.7.2	Cassos d'ús	81
	Cas d'ús 6.1: Eliminar punt clau	81
	Cas d'ús 6.2: Modificar instant de temps	81
	Cas d'ús 6.3: Eliminar punt de contacte	81
	Cas d'ús 6.4: Modificar punt de contacte	81
	Cas d'ús 6.5: Modificar fotograma de l'animació	82
	Cas d'ús 6.6: Modificar paràmetre d'una etapa	82
	Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau	82
2.8	Funcionalitat 7: Gestió de la visió de l'espai tridimensional	83
2.8.1	Necessitat	83
2.8.2	Cassos d'ús	83
	Cas d'ús 7.1: Pan	83
	Cas d'ús 7.2: Zoom	83
	Cas d'ús 7.3: Rotació	84
2.9	Funcionalitat 8: Gestió de la informació	84
2.9.1	Necessitat	84
2.9.2	Cassos d'ús	85
	Cas d'ús 8.1: Guardar model	85
	Cas d'ús 8.2: Carregar model	85
	Cas d'ús 8.3: Guardar escena	85
	Cas d'ús 8.4: Carregar escena	85
	Cas d'ús 8.5: Crear projecte	86
	Cas d'ús 8.6: Guardar projecte	86
	Cas d'ús 8.7: Carregar projecte	86
	Cas d'ús 8.8: Afegir gràfica	86
	Cas d'ús 8.9: Eliminar gràfica	87
	Cas d'ús 8.10: Visualitzar gràfica	87

3.1	Model de cassos d'ús. Contractes	89
3.1.1	Cas d'ús 1.1: Crear Model	89
3.1.2	Cas d'ús 1.2: Afegir "Conjunt de cadenes"	90
3.1.3	Cas d'ús 1.3: Afegir cadena al model	90
3.1.4	Cas d'ús 1.4: Afegir un "Link" al model	92
3.1.5	Cas d'ús 1.5: Afegir una Massa al model	93
3.1.6	Cas d'ús 1.6: Afegir un Recolzador	94
3.1.7	Cas d'ús 2.1: Eliminar model	95
3.1.8	Cas d'ús 2.2: Moure model	96
3.1.9	Cas d'ús 2.3: Rotar model	97
3.1.10	Cas d'ús 2.4: Eliminar cadena	98
3.1.11	Cas d'ús 2.5: Moure cadena	98
3.1.12	Cas d'ús 2.6: Rotar cadena	99
3.1.13	Cas d'ús 2.7: Eliminar "Conjunt de cadenes"	100
3.1.14	Cas d'ús 2.8: Eliminar "Link"	101
3.1.15	Cas d'ús 2.9: Rotar "Link"	102
3.1.16	Cas d'ús 2.10: Modificar la llargada del "Link"	103
3.1.17	Cas d'ús 2.11: Eliminar Massa	104
3.1.18	Cas d'ús 2.12: Rotar Massa	104
3.1.19	Cas d'ús 2.13: Rotar Recolzador	105
3.1.20	Cas d'ús 2.14: Eliminar Recolzador	106
3.1.21	Cas d'ús 2.15: Modificar la llargada del Recolzador	107
3.1.22	Cas d'ús 2.16: Modificar l'amplada del Recolzador	108
3.1.23	Cas d'ús 3.1: Afegir un objecte	109
3.1.24	Cas d'ús 4.1: Rotar objecte	109
3.1.25	Cas d'ús 4.2: Escalar objecte	110
3.1.26	Cas d'ús 4.3: Moure objecte	111
3.1.27	Cas d'ús 4.4: Duplicar objecte	112
3.1.28	Cas d'ús 4.5: Eliminar objecte	113
3.1.29	Cas d'ús 4.6: Modificar paràmetre objecte	113
3.1.30	Cas d'ús 5.1: Afegir punt clau	114
3.1.31	Cas d'ús 5.2: Modificar paràmetre del model articulat	115
3.1.32	Cas d'ús 5.3: Afegir punt de contacte	116
3.1.33	Cas d'ús 5.4: Assignar instant de temps	116
3.1.34	Cas d'ús 5.5: Generar animació	117
3.1.35	Cas d'ús 6.1: Eliminar punt clau	118
3.1.36	Cas d'ús 6.2: Modificar instant de temps	118
3.1.37	Cas d'ús 6.3: Eliminar punt de contacte	119
3.1.38	Cas d'ús 6.4: Modificar punt de contacte	120
3.1.39	Cas d'ús 6.5: Modificar fotograma de l'animació	120
	Cas d'ús 6.6: Modificar paràmetre d'una etapa	121

	Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau . . .	122
3.1.40	Cas d'ús 7.1: Pan	123
3.1.41	Cas d'ús 7.2: Zoom	123
3.1.42	Cas d'ús 7.3: Rotació	124
3.1.43	Cas d'ús 8.1: Guardar model	124
3.1.44	Cas d'ús 8.2: Carregar model	125
3.1.45	Cas d'ús 8.3: Guardar escena	126
3.1.46	Cas d'ús 8.4: Carregar escena	127
3.1.47	Cas d'ús 8.5: Crear projecte	127
3.1.48	Cas d'ús 8.6: Guardar projecte	128
3.1.49	Cas d'ús 8.7: Carregar projecte	129
3.1.50	Cas d'ús 8.8: Afegir gràfica	130
3.1.51	Cas d'ús 8.9: Eliminar gràfica	130
3.1.52	Cas d'ús 8.10: Visualitzar gràfica	131
3.2	Representació esquemàtica dels cassos d'ús	133
3.2.1	Generació de models	133
3.2.2	Modificació de models	134
3.2.3	Generació d'escenes	134
3.2.4	Modificació d'escenes	135
3.2.5	Generació d'animacions	135
3.2.6	Modificació d'animacions	136
3.2.7	Gestió de la visió de l'espai tridimensional	136
3.2.8	Gestió de la informació	137
4	Model d'Anàlisis	139
4.1	Model Conceptual de les classes	139
4.1.1	Model Conceptual de la funcionalitat 1	139
4.1.2	Model Conceptual de la funcionalitat 2	141
4.1.3	Model Conceptual de la funcionalitat 3	142
4.1.4	Model Conceptual de la funcionalitat 4	143
4.1.5	Model Conceptual de la funcionalitat 5	143
4.1.6	Model Conceptual de la funcionalitat 6	144
4.1.7	Model Conceptual de la funcionalitat 7	145
4.1.8	Model Conceptual de la funcionalitat 8	145
4.1.9	Model Conceptual complet	147
4.2	Diagrames de seqüència i VOPC's	148
4.2.1	Cas d'ús 1.1: Crear Model	149
	VOPC	149
	Diagrama de seqüència	149
4.2.2	Cas d'ús 1.2: Afegir "Conjunt de cadenes"	152
	VOPC	152

	Diagrama de seqüència	152
4.2.3	Cas d'ús 1.3: Afegir cadena al model	154
	VOPC	154
	Diagrama de seqüència	154
4.2.4	Cas d'ús 1.4: Afegir un "Link" al model	156
	VOPC	156
	Diagrama de seqüència	156
4.2.5	Cas d'ús 1.5: Afegir una Massa al model	158
	VOPC	158
	Diagrama de seqüència	158
4.2.6	Cas d'ús 1.6: Afegir un Recolzador al Model	160
	VOPC	160
	Diagrama de seqüència	160
4.2.7	Cas d'ús 2.1: Eliminar model	162
	VOPC	162
	Diagrama de seqüència	162
4.2.8	Cas d'ús 2.2: Moure model	164
	VOPC	164
	Diagrama de seqüència	164
4.2.9	Cas d'ús 2.3: Rotar model	166
	VOPC	166
	Diagrama de seqüència	166
4.2.10	Cas d'ús 2.4: Eliminar cadena	168
	VOPC	168
	Diagrama de seqüència	168
4.2.11	Cas d'ús 2.5: Moure cadena	170
	VOPC	170
	Diagrama de seqüència	170
4.2.12	Cas d'ús 2.6: Rotar cadena	172
	VOPC	172
	Diagrama de seqüència	172
4.2.13	Cas d'ús 2.7: Eliminar "Conjunt de cadenes"	174
	VOPC	174
	Diagrama de seqüència	174
4.2.14	Cas d'ús 2.8: Eliminar "Link"	176
	VOPC	176
	Diagrama de seqüència	176
4.2.15	Cas d'ús 2.9: Rotar "Link"	178
	VOPC	178
	Diagrama de seqüència	178
4.2.16	Cas d'ús 2.10: Modificar la llargada del "Link"	180

	VOPC	180
	Diagrama de seqüència	180
4.2.17	Cas d'ús 2.11: Eliminar Massa	182
	VOPC	182
	Diagrama de seqüència	182
4.2.18	Cas d'ús 2.12: Rotar Massa	184
	VOPC	184
	Diagrama de seqüència	184
4.2.19	Cas d'ús 2.13: Rotar un Recolzador	186
	VOPC	186
	Diagrama de seqüència	186
4.2.20	Cas d'ús 2.14: Eliminar un Recolzador	188
	VOPC	188
	Diagrama de seqüència	188
4.2.21	Cas d'ús 2.15: Modificar la llargada d'un Recolzador	190
	VOPC	190
	Diagrama de seqüència	190
4.2.22	Cas d'ús 2.16: Modificar l'amplada d'un Recolzador	192
	VOPC	192
	Diagrama de seqüència	192
4.2.23	Cas d'ús 3.1: Afegir un objecte	194
	VOPC	194
	Diagrama de seqüència	194
4.2.24	Cas d'ús 4.1: Rotar objecte	196
	VOPC	196
	Diagrama de seqüència	196
4.2.25	Cas d'ús 4.2: Escalar objecte	198
	VOPC	198
	Diagrama de seqüència	198
4.2.26	Cas d'ús 4.3: Moure objecte	200
	VOPC	200
	Diagrama de seqüència	200
4.2.27	Cas d'ús 4.4: Duplicar objecte	202
	VOPC	202
	Diagrama de seqüència	202
4.2.28	Cas d'ús 4.5: Eliminar objecte	204
	VOPC	204
	Diagrama de seqüència	204
4.2.29	Cas d'ús 4.6: Modificar paràmetre objecte	206
	VOPC	206
	Diagrama de seqüència	206

4.2.30	Cas d'ús 5.1: Crear punt clau	208
	VOPC	208
	Diagrama de seqüència	208
4.2.31	Cas d'ús 5.2: Modificar paràmetre del model articulat .	210
	VOPC	210
	Diagrama de seqüència	210
4.2.32	Cas d'ús 5.3: Afegir punt de contacte	212
	VOPC	212
	Diagrama de seqüència	212
4.2.33	Cas d'ús 5.4: Assignar instant de temps	214
	VOPC	214
	Diagrama de seqüència	214
4.2.34	Cas d'ús 5.5: Generar animació	216
	VOPC	216
	Diagrama de seqüència	216
4.2.35	Cas d'ús 6.1: Eliminar punt clau	218
	VOPC	218
	Diagrama de seqüència	218
4.2.36	Cas d'ús 6.2: Modificar instant de temps	220
	VOPC	220
	Diagrama de seqüència	220
4.2.37	Cas d'ús 6.3: Eliminar punt de contacte	222
	VOPC	222
	Diagrama de seqüència	222
4.2.38	Cas d'ús 6.4: Modificar punt de contacte	224
	VOPC	224
	Diagrama de seqüència	224
4.2.39	Cas d'ús 6.5: Modificar fotograma de l'animació	226
	VOPC	226
	Diagrama de seqüència	226
	Cas d'ús 6.6: Modificar paràmetre d'una etapa	228
	VOPC	228
	Diagrama de seqüència	228
	Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau . . .	230
	VOPC	230
	Diagrama de seqüència	230
4.2.40	Cas d'ús 7.1: Pan	232
	VOPC	232
	Diagrama de seqüència	232
4.2.41	Cas d'ús 7.2: Zoom	233
	VOPC	233

	Diagrama de seqüència	233
4.2.42	Cas d'ús 7.3: Rotació	234
	VOPC	234
	Diagrama de seqüència	234
4.2.43	Cas d'ús 8.1: Guardar model	235
	VOPC	235
	Diagrama de seqüència	235
4.2.44	Cas d'ús 8.2: Carregar model	237
	VOPC	237
	Diagrama de seqüència	237
4.2.45	Cas d'ús 8.3: Guardar escena	239
	VOPC	239
	Diagrama de seqüència	239
4.2.46	Cas d'ús 8.4: Carregar escena	241
	VOPC	241
	Diagrama de seqüència	241
4.2.47	Cas d'ús 8.5: Crear projecte	243
	VOPC	243
	Diagrama de seqüència	243
4.2.48	Cas d'ús 8.6: Guardar projecte	245
	VOPC	245
	Diagrama de seqüència	245
4.2.49	Cas d'ús 8.7: Carregar projecte	246
	VOPC	246
	Diagrama de seqüència	246
4.2.50	Cas d'ús 8.8: Afegir gràfica	248
	VOPC	248
	Diagrama de seqüència	248
4.2.51	Cas d'ús 8.9: Eliminar gràfica	250
	VOPC	250
	Diagrama de seqüència	250
4.2.52	Cas d'ús 8.10: Visualitzar gràfica	252
	VOPC	252
	Diagrama de seqüència	252
5	Model de Disseny	255
5.1	Introducció	255
5.2	Model de classes normalitzat	255
5.3	Aplicació de patrons	257
5.3.1	Arquitectura de l'aplicació	257
5.3.2	Capa de presentació	257

5.3.3	Capa de domini	259
5.3.4	Capa de gestió de dades	260
5.4	Diagrames de seqüència modificats	261
5.4.1	Cas d'ús 1.1: Crear Model	261
5.4.2	Cas d'ús 1.2: Afegir "Conjunt de cadenes"	262
5.4.3	Cas d'ús 1.3: Afegir cadena al model	263
5.4.4	Cas d'ús 1.4: Afegir un "Link" al model	264
5.4.5	Cas d'ús 1.5: Afegir una Massa al model	265
5.4.6	Cas d'ús 1.6: Afegir un Recolzador al model	266
5.4.7	Cas d'ús 2.1: Eliminar model	267
5.4.8	Cas d'ús 2.2: Moure model	268
5.4.9	Cas d'ús 2.3: Rotar model	269
5.4.10	Cas d'ús 2.4: Eliminar cadena	270
5.4.11	Cas d'ús 2.5: Moure cadena	271
5.4.12	Cas d'ús 2.6: Rotar cadena	272
5.4.13	Cas d'ús 2.7: Eliminar "Conjunt de cadenes"	273
5.4.14	Cas d'ús 2.8: Eliminar "Link"	274
5.4.15	Cas d'ús 2.9: Rotar "Link"	275
5.4.16	Cas d'ús 2.10: Modificar la llargada del "Link"	276
5.4.17	Cas d'ús 2.11: Eliminar Massa	277
5.4.18	Cas d'ús 2.12: Rotar Massa	278
5.4.19	Cas d'ús 2.13: Rotar un Recolzador	279
5.4.20	Cas d'ús 2.14: Eliminar un Recolzador	280
5.4.21	Cas d'ús 2.15: Modificar la llargada d'un Recolzador	281
5.4.22	Cas d'ús 2.16: Modificar l'amplada d'un Recolzador	282
5.4.23	Cas d'ús 3.1: Afegir un objecte	283
5.4.24	Cas d'ús 4.1: Rotar objecte	284
5.4.25	Cas d'ús 4.2: Escalar objecte	285
5.4.26	Cas d'ús 4.3: Moure objecte	286
5.4.27	Cas d'ús 4.4: Duplicar objecte	287
5.4.28	Cas d'ús 4.5: Eliminar objecte	288
5.4.29	Cas d'ús 4.6: Modificar paràmetre objecte	289
5.4.30	Cas d'ús 5.1: Crear punt clau	290
5.4.31	Cas d'ús 5.2: Modificar paràmetre del model articulat	291
5.4.32	Cas d'ús 5.3: Afegir punt de contacte	292
5.4.33	Cas d'ús 5.4: Assignar instant de temps	293
5.4.34	Cas d'ús 5.5: Generar animació	294
5.4.35	Cas d'ús 6.1: Eliminar punt clau	295
5.4.36	Cas d'ús 6.2: Modificar instant de temps	296
5.4.37	Cas d'ús 6.3: Eliminar punt de contacte	297
5.4.38	Cas d'ús 6.4: Modificar punt de contacte	298

5.4.39	Cas d'ús 6.5: Modificar fotograma de l'animació	299
	Cas d'ús 6.6: Modificar paràmetre d'una etapa	300
	Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau	301
5.4.40	Cas d'ús 7.1: Pan	302
5.4.41	Cas d'ús 7.2: Zoom	302
5.4.42	Cas d'ús 7.3: Rotació	303
5.4.43	Cas d'ús 8.1: Guardar model	304
5.4.44	Cas d'ús 8.2: Carregar model	305
5.4.45	Cas d'ús 8.3: Guardar escena	306
5.4.46	Cas d'ús 8.4: Carregar escena	307
5.4.47	Cas d'ús 8.5: Crear projecte	308
5.4.48	Cas d'ús 8.6: Guardar projecte	309
5.4.49	Cas d'ús 8.7: Carregar projecte	310
5.4.50	Cas d'ús 8.8: Afegir gràfica	311
5.4.51	Cas d'ús 8.9: Eliminar gràfica	312
5.4.52	Cas d'ús 8.10: Visualitzar gràfica	313
5.5	Disseny de la interfície	314
6	Pla de test, proves i recerca realitzada	317
6.1	Introducció	317
6.2	Pla de test	317
6.3	Proves d'exemple	322
6.3.1	Exemple 1: Comprovació del funcionament del cas d'ús Crear Model	322
6.3.2	Exemple 2: Comprovació del càlcul de la derivada de la transformació que comporta una articulació al llarg d'una cadena	325
6.4	Recerca	328
7	Cost econòmic i planificació	331
8	Conclusions	333

Índex de figures

1.1	Articulació revolució	33
1.2	Representació d'un humanoide	34
1.3	Representació d'un humanoide amb un conjunt de cadenes	35
1.4	Exemple senzill del mètode proposat per Denavit i Hartenberg	36
1.5	Exemple de taula DH	37
1.6	Model articulat senzill: Tronc superior d'un humanoide	38
1.7	Taules D-H dels braços de la figura 1.6	39
1.8	Rotacions a partir d'un sistema de referència fixe	43
1.9	Rotacions a partir del sistema de referència que descriu el cos	44
1.10	cas 1: Modelat de la connexió entre una massa amb una altre massa	45
1.11	cas 2: Modelat de la connexió entre una massa amb un "link"	46
1.12	Exemple Senzill: Cames	49
1.13	Etapa 1: Impuls	51
1.14	Etapa 2: Vol	51
1.15	Etapa 3: Recepció	52
1.16	Pseudocodi del càlcul de la Trajectòria del Centre de Massa en una Etapa de Contacte	55
1.17	Envolupant convexa d'un conjunt de punts	59
3.1	Esquema de la funcionalitat 1: Generació de models	133
3.2	Esquema de la funcionalitat 2: Modificació de models	134
3.3	Esquema de la funcionalitat 3: Generació d'escenes	134
3.4	Esquema de la funcionalitat 4: Modificació d'escenes	135
3.5	Esquema de la funcionalitat 5: Generació d'animacions	135
3.6	Esquema de la funcionalitat 6: Modificació d'animacions	136
3.7	Esquema de la funcionalitat 7: Gestió de la visió de l'espai tridimensional	136
3.8	Esquema de la funcionalitat 8: Gestió de la informació	137
4.1	Model Conceptual de la funcionalitat 1: Generació de models	140

4.2	Model Conceptual de la funcionalitat 2: Modificació de models	141
4.3	Model Conceptual de la funcionalitat 3: Generació d'escenes	142
4.4	Model Conceptual de la funcionalitat 4: Modificació d'escenes	143
4.5	Model Conceptual de la funcionalitat 5: Generació de l'animació	144
4.6	Model Conceptual de la funcionalitat 6: Modificació de l'animació	145
4.7	Model Conceptual de la funcionalitat 7: Gestió de l'espai tridimensional	145
4.8	Model Conceptual de la funcionalitat 8: Gestió de Projectes	146
4.9	Model Conceptual complet	147
4.10	VOPC de Crear Model	149
4.11	Diagrama de seqüència de Crear Model	151
4.12	VOPC de Afegir "Conjunt de cadenes"	152
4.13	Diagrama de seqüència de Afegir "Conjunt de cadenes"	153
4.14	VOPC de Afegir cadena al model	154
4.15	Diagrama de seqüència de Afegir cadena al model	155
4.16	VOPC de Afegir un "Link" al model	156
4.17	Diagrama de seqüència de Afegir un "Link" al model	157
4.18	VOPC de Afegir una Massa al model	158
4.19	Diagrama de seqüència de Afegir una Massa al model	159
4.20	VOPC de Afegir un Recolzador al Model	160
4.21	Diagrama de seqüència d'Afegir un Recolzador al Model	161
4.22	VOPC de Eliminar model	162
4.23	Diagrama de seqüència de Eliminar model	163
4.24	VOPC de Moure model	164
4.25	Diagrama de seqüència de Moure model	165
4.26	VOPC de Rotar model	166
4.27	Diagrama de seqüència de Rotar model	167
4.28	VOPC de Eliminar cadena	168
4.29	Diagrama de seqüència de Eliminar cadena	169
4.30	VOPC de Moure cadena	170
4.31	Diagrama de seqüència de Moure cadena	171
4.32	VOPC de Rotar cadena	172
4.33	Diagrama de seqüència de Rotar cadena	173
4.34	VOPC de Eliminar "Conjunt de cadenes"	174
4.35	Diagrama de seqüència de Eliminar "Conjunt de cadenes"	175
4.36	VOPC de Eliminar "Link"	176
4.37	Diagrama de seqüència de Eliminar "Link"	177
4.38	VOPC de Rotar "Link"	178
4.39	Diagrama de seqüència de Rotar "Link"	179
4.40	VOPC de Modificar la llargada del "Link"	180

4.41	Diagrama de seqüència de Modificar la llargada del "Link"	181
4.42	VOPC de Eliminar Massa	182
4.43	Diagrama de seqüència de Eliminar Massa	183
4.44	VOPC de Rotar Massa	184
4.45	Diagrama de seqüència de Rotar Massa	185
4.46	VOPC de Rotar un Recolzador	186
4.47	Diagrama de seqüència de Rotar un Recolzador	187
4.48	VOPC de Eliminar un Recolzador	188
4.49	Diagrama de seqüència de Eliminar un Recolzador	189
4.50	VOPC de Modificar la llargada d'un Recolzador	190
4.51	Diagrama de seqüència de Modificar la llargada d'un Recolzador	191
4.52	VOPC de Modificar l'amplada d'un Recolzador	192
4.53	Diagrama de seqüència de Modificar l'amplada d'un Recolzador	193
4.54	VOPC de Afegir un objecte	194
4.55	Diagrama de seqüència de Afegir un objecte	195
4.56	VOPC de Rotar objecte	196
4.57	Diagrama de seqüència de Rotar objecte	197
4.58	VOPC de Escalar objecte	198
4.59	Diagrama de seqüència de Escalar objecte	199
4.60	VOPC de Moure objecte	200
4.61	Diagrama de seqüència de Moure objecte	201
4.62	VOPC de Duplicar objecte	202
4.63	Diagrama de seqüència de Duplicar objecte	203
4.64	VOPC de Eliminar objecte	204
4.65	Diagrama de seqüència de Eliminar objecte	205
4.66	VOPC de Modificar paràmetre objecte	206
4.67	Diagrama de seqüència de Modificar paràmetre objecte	207
4.68	VOPC de Crear punt clau	208
4.69	Diagrama de seqüència de Crear punt clau	209
4.70	VOPC de Modificar paràmetre del model articulat	210
4.71	Diagrama de seqüència de Modificar paràmetre del model articulat	211
4.72	VOPC de Afegir punt de contacte	212
4.73	Diagrama de seqüència de Afegir punt de contacte	213
4.74	VOPC de Assignar instant de temps	214
4.75	Diagrama de seqüència de Assignar instant de temps	215
4.76	VOPC de Generar animació	216
4.77	Diagrama de seqüència de Generar animació	217
4.78	VOPC de Eliminar punt clau	218
4.79	Diagrama de seqüència de Eliminar punt clau	219
4.80	VOPC de Modificar instant de temps	220

4.81	Diagrama de seqüència de Modificar instant de temps	221
4.82	VOPC de Eliminar punt de contacte	222
4.83	Diagrama de seqüència de Eliminar punt de contacte	223
4.84	VOPC de Modificar punt de contacte	224
4.85	Diagrama de seqüència de Modificar punt de contacte	225
4.86	VOPC de Modificar fotograma de l'animació	226
4.87	Diagrama de seqüència de Modificar fotograma de l'animació	227
4.88	VOPC de Modificar paràmetre d'una etapa	228
4.89	Diagrama de seqüència de Modificar paràmetre d'una etapa	229
4.90	VOPC de Modificar paràmetre d'un Punt Clau	230
4.91	Diagrama de seqüència de Modificar paràmetre d'un Punt Clau	231
4.92	VOPC de Pan	232
4.93	Diagrama de seqüència de Pan	232
4.94	VOPC de Zoom	233
4.95	Diagrama de seqüència de Zoom	233
4.96	VOPC de Rotació	234
4.97	Diagrama de seqüència de Rotació	234
4.98	VOPC de Guardar model	235
4.99	Diagrama de seqüència de Guardar model	236
4.100	VOPC de Carregar model	237
4.101	Diagrama de seqüència de Carregar model	238
4.102	VOPC de Guardar escena	239
4.103	Diagrama de seqüència de Guardar escena	240
4.104	VOPC de Carregar escena	241
4.105	Diagrama de seqüència de Carregar escena	242
4.106	VOPC de Crear projecte	243
4.107	Diagrama de seqüència de Crear projecte	244
4.108	VOPC de Guardar projecte	245
4.109	Diagrama de seqüència de Guardar projecte	245
4.110	VOPC de Carregar projecte	246
4.111	Diagrama de seqüència de Carregar projecte	247
4.112	VOPC de Afegir gràfica	248
4.113	Diagrama de seqüència de Afegir gràfica	249
4.114	VOPC de Eliminar gràfica	250
4.115	Diagrama de seqüència de Eliminar gràfica	251
4.116	VOPC de Visualitzar gràfica	252
4.117	Diagrama de seqüència de Visualitzar gràfica	253
5.1	Model Conceptual Normalitzat	256
5.2	Model Conceptual de la capa de presentació	258
5.3	Model Conceptual de la capa de domini	260

5.4	Model Conceptual de la capa de gestió de dades	260
5.5	Diagrama de seqüència de Crear Model	261
5.6	Diagrama de seqüència de Afegir "Conjunt de cadenes"	262
5.7	Diagrama de seqüència de Afegir cadena al model	263
5.8	Diagrama de seqüència de Afegir un "Link" al model	264
5.9	Diagrama de seqüència de Afegir una Massa al model	265
5.10	Diagrama de seqüència de Afegir un Recolzador al model	266
5.11	Diagrama de seqüència de Eliminar model	267
5.12	Diagrama de seqüència de Moure model	268
5.13	Diagrama de seqüència de Rotar model	269
5.14	Diagrama de seqüència de Eliminar cadena	270
5.15	Diagrama de seqüència de Moure cadena	271
5.16	Diagrama de seqüència de Rotar cadena	272
5.17	Diagrama de seqüència de Eliminar "Conjunt de cadenes"	273
5.18	Diagrama de seqüència de Eliminar "Link"	274
5.19	Diagrama de seqüència de Rotar "Link"	275
5.20	Diagrama de seqüència de Modificar la llargada del "Link"	276
5.21	Diagrama de seqüència de Eliminar Massa	277
5.22	Diagrama de seqüència de Rotar Massa	278
5.23	Diagrama de seqüència de Rotar un Recolzador	279
5.24	Diagrama de seqüència de Eliminar un Recolzador	280
5.25	Diagrama de seqüència de Modificar la llargada d'un Recolzador	281
5.26	Diagrama de seqüència de Modificar l'amplada d'un Recolzador	282
5.27	Diagrama de seqüència de Afegir un objecte	283
5.28	Diagrama de seqüència de Rotar objecte	284
5.29	Diagrama de seqüència de Escalar objecte	285
5.30	Diagrama de seqüència de Moure objecte	286
5.31	Diagrama de seqüència de Duplicar objecte	287
5.32	Diagrama de seqüència de Eliminar objecte	288
5.33	Diagrama de seqüència de Modificar paràmetre objecte	289
5.34	Diagrama de seqüència de Crear punt clau	290
5.35	Diagrama de seqüència de Modificar paràmetre del model articulat	291
5.36	Diagrama de seqüència de Afegir punt de contacte	292
5.37	Diagrama de seqüència de Assignar instant de temps	293
5.38	Diagrama de seqüència de Generar animació	294
5.39	Diagrama de seqüència de Eliminar punt clau	295
5.40	Diagrama de seqüència de Modificar instant de temps	296
5.41	Diagrama de seqüència de Eliminar punt de contacte	297
5.42	Diagrama de seqüència de Modificar punt de contacte	298
5.43	Diagrama de seqüència de Modificar fotograma de l'animació	299

5.44	Diagrama de seqüència de Modificar paràmetre d'una etapa	300
5.45	Diagrama de seqüència de Modificar paràmetre d'un Punt Clau	301
5.46	Diagrama de seqüència de Pan	302
5.47	Diagrama de seqüència de Zoom	302
5.48	Diagrama de seqüència de Rotació	303
5.49	Diagrama de seqüència de Guardar model	304
5.50	Diagrama de seqüència de Carregar model	305
5.51	Diagrama de seqüència de Guardar escena	306
5.52	Diagrama de seqüència de Carregar escena	307
5.53	Diagrama de seqüència de Crear projecte	308
5.54	Diagrama de seqüència de Guardar projecte	309
5.55	Diagrama de seqüència de Carregar projecte	310
5.56	Diagrama de seqüència de Afegir gràfica	311
5.57	Diagrama de seqüència de Eliminar gràfica	312
5.58	Diagrama de seqüència de Visualitzar gràfica	313
5.59	Interfície gràfica principal	315
6.1	Captura de l'aturada del programa amb Qt Creator que mostra el cas d'ús Crear Model	325
6.2	Captura de l'obtenció de la derivada d'un gir respecte l'eix Z amb Matlab	326
6.3	Captura de l'obtenció de la derivada d'un gir respecte l'eix Z amb el programa	327

Índex de taules

1.1	Taula D-H de l'esquena de la figura 1.6	39
1.2	Taula D-H del cas 1	45
1.3	Taula D-H del cas 2	46
7.1	Distribució de les hores en una setmana normal	331
7.2	Distribució de les hores en perfils i càlcul dels costos a partir dels sous	332

Capítol 1

Requisits del sistema

1.1 Propòsit del programa

La funció principal del programa consisteix en la realització d'animacions on intervé un o més models de manera que inicialment es proporcionen els punts de contacte amb alguna superfície o superfícies d'una escena generada anteriorment. Una vegada entrada la informació el sistema calcularà tot el moviment d'una manera realista.

El sistema ha de permetre de manera molt intuïtiva que l'usuari doni els punts claus d'un moviment complex i que el sistema calculi el moviment interpolat, primer a partir d'una interpolació lineal, i després mitjançant les lleis de la física per realitzar el moviment que passi per aquest punt. Una vegada generada aquesta animació, el sistema ha de permetre modificar, a l'usuari, certs punts per crear un moviment dependent de la naturalesa de cada cos de la realitat i que el sistema recalculi el moviment.

1.2 Motivacions

El fet de poder calcular un moviment mitjançant un conjunt limitat de punts clau per passar aquest moviment és una manera de simplificar molt el disseny en animacions, per exemple. I si a més, s'aconsegueix un moviment molt realista i proper a un cos concret, es podria arribar al punt de que es podria estudiar el moviment sense ús de sensors, ja que les dades les calcularia

el sistema. O en un punt extrem, aconseguir estudiar si la possibilitat de realitzar un moviment és possible o no per a un model articulat determinat.

Aquest sistema, podria tenir cabuda en camps com el disseny gràfic i multimèdia i seria útil en el camp de l'esport físic i de les seves patologies.

1.3 Descripció

El programa haurà de tenir un interfície gràfica amb la qual l'usuari pugui interactuar per tal de, generar models, editar escenes, modificar els models generats de manera que pugui donar diferents estats del sistema articulat en un moment determinat i poder generar animacions amb uns paràmetres físics concrets. Com també poder moure's per tot l'entorn tridimensional per buscar la manera més còmode de realitzar les funcionalitats nombrades anteriorment. Per tot això caldrà un sistema que gestioni els models, escenes i animacions mitjançant projectes.

Pel que fa a la generació de models articulats, ha de permetre generar-los a partir de dos mecanismes, el primer ha de ser a partir d'una taula descriptiva del model, on l'usuari va entrant els valors corresponents. I un segon mecanisme que seria a partir de la interfície gràfica, més intuïtiva i senzilla de cara a l'usuari. Aquest últim mecanisme consistirà en donar la informació del model a partir del ratolí introduït en cada moment si interessa una massa o un link, i a quina massa o link va lligada.

Per altre banda, la generació de escenes, part important ja que ens donarà els punts de recolzament que tindrà el model amb l'exterior. Necessitem un sistema que ens ajudi editar escenes de manera senzilla, així doncs, aquesta part ha de permetre introduir objectes en l'espai de treball, poder-los moure, escalar, o rotar segons convingui i modificar les seves característiques.

Una funcionalitat més que podem tenir és la modificació del model que ens donarà les diferents posicions o estats que tindrà el model en un temps determinat i poder aportar la informació necessària per a realitzar l'animació. Com també, més tard, poder modificar els punts concrets que interessin de l'animació per fer un moviment encara més realista.

A la part de generació d'animacions, el sistema ha de poder realitzar a partir de les posicions donades en els instants marcats un moviment físicament realista. Com després poder donar l'opció a l'usuari de modificar els paràmetres calculats en cada instant de temps de cada una de les posicions o estats del model articulat per tal de poder modificar el moviment, un moviment genèric i calculat a partir de l'optimització de certs paràmetres i que no representen amb exactitud la realitat. Però si que dóna a l'usuari quin seria el moviment ideal donats els diferents punts clau.

També caldrà un sistema de navegació per l'entorn tridimensional. L'objectiu d'aquesta funcionalitat és fer que l'usuari busqui la millor visió per generar i editar tant el model com l'escena i buscar millors posicions per dissenyar l'animació. Com també veure des de diferents punts de vista aquesta animació.

I finalment, s'ha de permetre a l'usuari guardar models articulats, escenes i animacions, dit pròpiament projectes d'aquesta aplicació i poder-les carregar en qualsevol moment. Per tant, caldrà un sistema d'emmagatzematge de la informació que l'usuari hagi introduït. Com també caldrà poder veure informació calculada per el sistema, ja sigui forces, velocitats...

1.4 Funcionalitats de l'aplicació

Les principals funcionalitats que tindrà el nostre sistema per tant, seran:

1. Generació/Modificació de models
2. Generació/Modificació d'escenes
3. Generació/Modificació animacions
4. Gestió de la visió de l'espai tridimensional
5. Gestió de la informació d'entrada

Aquestes funcionalitats són les principals perquè s'ha de passar per elles per arribar a la generació del moviment complet final esperat i amb cap d'aquestes funcionalitats es podria fer.

1.5 Arquitectura de l'aplicació

L'aplicació seguirà el patró de les tres capes. Tindrà una primera capa en aquest cas molt important ja que ha de permetre l'usuari interaccionar amb el sistema de moltes maneres diferents per fer lo mateix per tal de buscar la forma més còmode de fer-la, aquesta és la capa de presentació. La capa de presentació tindrà una part de formularis (botons, cel·les d'entrada de valors...) on l'usuari entrarà paràmetres més propis de lo que és l'animació en si i no el seu contingut, per exemple, mentre que tindrà una altre part de gràfics que serà la part on l'usuari generarà els models, escenes i podrà modificar-les com també els punts claus de l'animació.

Una capa per sota a la de presentació, la de domini, en aquesta capa existiran un conjunt de classes encarregades de recollir la informació que rep la capa de presentació i fer-ho consistent amb la informació anterior. També és la part encarregada dels càlculs que s'hagin de fer per generar l'animació, com també garantir la validesa de les dades provinents de la capa de presentació. Es comunica amb la capa de gestió de dades.

Aquesta capa de gestió de dades s'encarrega de guardar la informació que es va generant en la capa de domini, el que fa falta és un sistema de fitxers o un sistema de gestió de base de dades amb el qual li pugui anar passant tota la informació i anar-la guardant de manera consistent. Cal doncs que aquesta capa s'encarregui de que una vegada guardada la informació i tornada a carregar l'estat del programa sigui el mateix que en el moment que s'ha guardat.

1.6 Requeriments

Una vegada explicat les característiques que tindrà l'aplicació, és el moment de veure quines són les tecnologies que necessitarà el nostre sistema per tal de cobrir les necessitats definides anteriorment.

Per tal de poder generar models i modificar-los, necessitarem una interfície que recollí la informació que l'usuari entra, en aquest cas, el que es farà serà mitjançant un kit de desenvolupament anomenat **Qt Creator** que ajuda a

crear aplicacions en C++ de manera còmode i senzilla. Aquest kit, inclou un dissenyador de interfícies molt fàcil d'ús.

Per altre banda, l'aplicació ha de tenir una manera de representar models en 3D, ajudar a crear-los, modificar-los i eliminar-los. Una de les raons per les quals s'ha decidit crear una aplicació en C++ i no amb un altre llenguatge és perquè tenim una llibreria en C++ molt completa per tal de utilitzar **OpenGL**, una interfície per tal de poder programar les targetes gràfiques i que facin que es pugui generar imatges virtuals en 3D.

Pel que fa a la generació d'animacions, el programa ha de garantir que una vegada creada l'animació es pugui generar un vídeo amb l'animació final creada, per tant, fa falta una llibreria que pugui generar vídeos a partir de les imatges que es puguin anar recollint de l'animació generada amb el programa.

Si volem que un usuari pugui deixar el procés a mitges i continuar-lo després s'ha de crear un sistema de fitxers que pugui guardar informació a fitxers i després poder-los carregar per tal de no perdre la feina feta. Per tant, a partir de llibreries que tractin amb fitxers, en C++, caldrà dissenyar i implementar un sistema que ajudi a cobrir aquesta necessitat.

I finalment, tindrem una part molt important de la Física de cossos amb articulacions, per tant, caldrà fer un estudi profund de les necessitats que necessitem cobrir. Caldrà doncs determinar els paràmetres de la física que ens interessin i buscar la manera de introduir aquesta part al sistema de manera consistent.

1.6.1 QT Creator

Qt Creator és un entorn de desenvolupament integrat per tal de desenvolupar aplicacions amb la biblioteca **Qt**. Qt és una biblioteca que té la principal propietat que ajuda al programador a crear interfície gràfica per una aplicació qualsevol de manera senzilla, tot hi que també es poden fer aplicacions sense interfície gràfica usant aquesta biblioteca.

1.6.2 OpenGL

OpenGL és una especificació estàndard que defineix una API (Interfície de programació d'aplicacions) per crear programes que generin gràfics en 2D o 3D. En C++ tenim una llibreria que té les funcions implementades i que es poden usar tal i com OpenGL especifica.

1.6.3 Vídeo

La idea és poder guardar les imatges i ajuntar-les per poder realitzar un vídeo. Hi ha molts programes que fan això i que es podrien utilitzar cridant-los a mode de crida a sistema.

1.6.4 Sistema de fitxers

Farà falta un sistema que mantingui la consistència i que permeti de manera fàcil escriure i llegir a fitxers per tal de guardar informació. En C++ tenim mètodes fàcils per realitzar aquestes operacions, caldrà doncs usar-les per poder implementar un sistema senzill per guardar informació i llegir-la.

1.7 Estudi de la Física de cossos articulats

En aquesta secció, el que es farà serà presentar la teoria de la Física per poder portar a terme l'aplicació, com s'ha plantejat el problema i com s'abor- darà per solucionar-lo.

1.7.1 Matrius homogènies

Les matrius homogènies són matrius 4x4 que permeten descriure una translació i/o una rotació en una mateixa matriu, és a dir, si tenim un sistema de referència {A} i un altre {B} i el que ens interessa és tenir la descripció dels punts de {B} descrits des de {A}. Aleshores, es vol, per simplificar els càlculs, fer una operació que tindria la següent forma:

$${}^A P = {}^A_B T * {}^B P \quad (1.1)$$

on ${}^A P$ és un punt descrit des de $\{A\}$, la ${}^A_B T$ és la transformació que porta d'un sistema de referència a l'altre i ${}^B P$ és la descripció del punt respecte del sistema de coordenades $\{B\}$. Així doncs, si la transformació consisteix en una rotació i una translació sabem que seria el mateix que aplicar una rotació i seguida de una translació en una matriu de 3×3 de manera que primer multiplicariem el punt descrit des de $\{B\}$ i li sumariem la translació, això el que comportaria seria que necessitariem una matriu 3×3 i una translació i que per tant, el que s'hauria de fer sempre seria seguir el procediment de aplicar la rotació primer i després fer la suma del vector translació, amb la matriu homogènia el que fa és limitar la transformació a una sola multiplicació amb una matriu 4×4 que conté precisament el cas anterior. Per veure realment la forma que té una matriu homogènia, es descriu primer la rotació i la translació com a:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Llavors, la fórmula que descriuriem si tinguéssim aquesta forma, seria:

$${}^A P = R * {}^B P + T \quad (1.2)$$

això pot comportar errors en el moment de si hem d'aplicar primer la translació o no, per exemple, la solució de la matriu homogènia consisteix en crear una matriu 4×4 , de manera que els punts descrits, necessàriament se'ls haurà d'incloure una quarta component per tal de poder efectuar la multiplicació. Així doncs, el que fa és que aquesta quarta component prengui el valor de 1, i la forma de la matriu homogènia seria la següent:

$${}^A_B T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Per tant, si observem la fórmula 1.1 l'únic que cal fer és la multiplicació de la matriu 4×4 i ens farà una rotació i una translació, en aquest ordre, en una sola operació, mentre que l'equació 1.2 necessita dues operacions i saber l'ordre en que es fan les coses. Es veurà més endavant per quina altre raó varen sorgir les matrius homogènies i que realment val la pena dominar aquest tipus de matrius per expressar les transformacions d'una manera molt clara.

1.7.2 Operacions de composició amb matrius homogènies

Com he comentat en l'apartat anterior, l'ús de matrius homogènies per tal de descriure diferents canvis de sistemes de referència, és molt útil no només per descriure dos transformacions consecutives com poden ser la rotació i la translació, sinó que també permet representar de manera molt senzilla canvis entre diferents sistemes de referència que aparentment no tinguin res en comú. Per exemple, si es coneix com passar d'un sistema de referència $\{A\}$ a un $\{X\}$ i d'un $\{X\}$ a un $\{B\}$, amb una matriu homogènia diferent per cada una d'aquestes dues, es pot trobar una matriu homogènia que porti de $\{A\}$ a $\{B\}$ de manera directa. Expressat en llenguatge matemàtic, el que tenim és:

$${}^A P = {}^A_X T * {}^X P \quad (1.3)$$

$${}^X P = {}^X_B T * {}^B P \quad (1.4)$$

aleshores, per la propietat de composició de matrius homogènies, el que es pot dir és que:

$${}^A P = {}^A_X T * {}^X_B T * {}^B P \quad (1.5)$$

per tant, tindrem una matriu homogènia que de manera directa ens porta del sistema de referència $\{A\}$ al $\{B\}$. Per tant, si el model permet incloure sistemes de referència intermitjos $\{X\}$ capaços de portar d'una sistema de referència $\{A\}$ al $\{B\}$.

1.7.3 Anàlisi i descripció de sistemes articulats

Un sistema articulat és un conjunt d'articulacions lligades per un link o connexió rígida que les uneix i que li permet que diferents parts d'aquest sistema es puguin moure en l'espai 2D o 3D. Així doncs, usarem els sistemes articulats per representar models de la realitat. Exemple, una persona es pot veure com un sistema articulat si pensem en les diferents parts mòbils del cos, com els genolls, colzes, hombros... com un conjunt de masses mòbils impulsades per les accions de les articulacions, les quals estan interconnectades entre sí i en aquest cas per links o connexions com els ossos i els músculs, que uneixen per exemple el genoll amb el turmell.

Matemàticament, la distribució de massa de cada element rígida pot descriure's pel seu tensor de inèrcia, o equivalentment, per un conjunt de 6 masses puntuals que es mouen solidàriament amb l'element rígida. En el cas d'elements d'estructura longitudinal, el conjunt de masses que caracteritza

la distribució pot reduir-se a només dues, que situarem en els dos extrems de l'element.

En el nostre estudi usarem articulacions de revolució (que per abús de llenguatge anomenarem masses mòbils) que permeten als dos elements connectats tenir una mobilitat en 2D, el que implica que només tingui un grau de llibertat: només cal un paràmetre per a descriure-la, que és l'angle que forma respecte un eix arbitrari. Això no comporta cap pèrdua de generalitat, ja que per exemple el moviment en l'espai 3D originat per una articulació de dos graus de llibertat es pot descriure amb dues articulacions de un paràmetre units per un link de longitud 0.

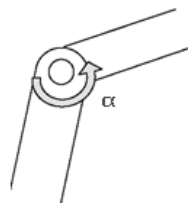


Figura 1.1: Articulació revolució

En la figura 1.1 veiem un exemple de articulació de revolució. Pot moure els "links" en l'espai de 2D perquè en el moment en que l'angle α varia els "links" enganxats a l'articulació són els que es mouen en 2D, així doncs, la funció dels "links" és la transmissió del moviment generat per una articulació d'un extrem a l'altre. També es diu que aquesta articulació té un grau de llibertat, en concret el grau α en aquest cas.

Per altre banda, una vegada explicat la massa mòbil i la connexió o link, cal veure que per si sol el sistema només podria tenir un únic inici i un únic final. Cal dir doncs que molts sistemes, com poden ser el dels humanoïdes, tenen fins a cinc terminacions, el cap, dues mans i dos peus. Així doncs, cal incorporar el concepte de punt de bifurcació de cadenes. El primer d'ells pot associar-se (de manera descriptiva) al sistema Centre del Model, del qual parteixen totes les cadenes de links i articulacions. Cal dir que es manté estàtica en el sentit de que els links connectats a ell no es poden moure, però això comporta que si se li aplica algun moviment o acció a aquest centre del model, el que provoca és que aquesta mateixa transformació afecta a

tot el sistema. Per tant, en aquest estudi els elements principals que poden representar un sistema articulat de la realitat, vindran a ser:

1. Masses mòbils o articulacions
2. "Links" o connexions
3. Sistema Centre del Model

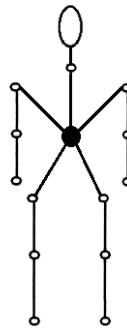


Figura 1.2: Representació d'un humànoide

En la Figura 1.2, es pot veure clarament el centre del model (rodona negra per dins), els links (línies) i les masses mòbils (rodones), aquest últim pot representar més d'una massa mòbil una sobre de l'altre, en el cas de tenir links de longitud zero. Cal remarcar que s'ha dibuixat el cap perquè es veïés quina era la orientació del sistema. Cal dir també que aquest sistema és molt senzill, segurament un sistema més real tindria més articulacions i per tant links, i que segurament el centre del model no estaria en aquella posició. Per altre banda, com es pot veure en la Figura 1.2, la representació de l'humànoide no és del tot correcte, ja que un moviment de l'esquena no afectaria als braços, per exemple. Per a aquestes situacions en que tenim punts de bifurcació utilitzem l'element conjunt de cadenes, que serà un element com el centre del model, que d'ella parteixen, també, un conjunt de cadenes. La diferència amb el centre del model és que d'aquest últim es parteix d'ell per descriure tot el model, mentre que l'altre només tindrà accés a una part parcial de la descripció del model, les cadenes unides a ell. Aleshores, una representació de l'humànoide vist anteriorment seria:

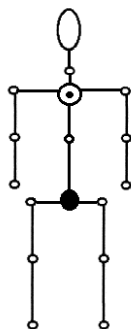


Figura 1.3: Representació d'un humanoide amb un conjunt de cadenes

Un element més és el conjunt de cadenes representat com a una rodona amb un punt al mig.

Així doncs, una vegada s'ha trobat com representar un model articulat i s'ha analitzat els diferents elements que pot tenir cal veure com es modelitza el sistema de manera que es pugui representar a l'espai i treballar amb els seus graus de llibertat.

1.7.4 Modelització de sistemes articulats. Taula D-H

Una vegada es té analitzat quins són els elements d'un sistema articulat i s'han descrit, cal veure com es modelitzen els sistemes articulats i obtenir una descripció del model sense perdre informació pel camí. El mètode de modelització que se seguirà serà el descrit per Denavit i Hartenberg mitjançant la taula D-H. El que indica aquesta taula són un conjunt de transformacions.

Cal, però, primer descriure com funciona el mètode de modelització per poder entendre més tard quina informació aporta la taula D-H. Denavit i Hartenberg descriuen un model articulat com una successió d'articulacions i connexions, per tant, una articulació o massa no admet més d'una connexió ja que el mètode no reflecteix aquest cas a priori.

La manera de com s'arriba del model articulat a l'espai a una simple taula de valors, és la següent. Primer es col·loca un sistema de referència a cada una de les masses que té el model de manera que l'eix Z es col·locarà en perpendicular amb el moviment de rotació de la massa, de manera que

aplicant una rotació sobre aquest eix aconseguim descriure el moviment que tindria una certa articulació. L'eix X tindrà la direcció de la recta de mínima distància que uneix les dues rectes definides per els eixos Z's de la massa que estem tractant amb la seva anterior. Segons com col·loquem els sentits dels eixos, l'angle haurà de tractar-se positiu o negatiu segons s'escaigui. Una figura a mode d'exemple de com funciona aquest mètode és el següent:

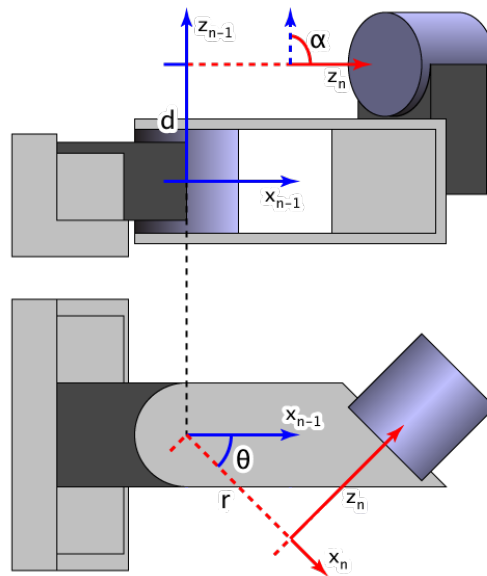


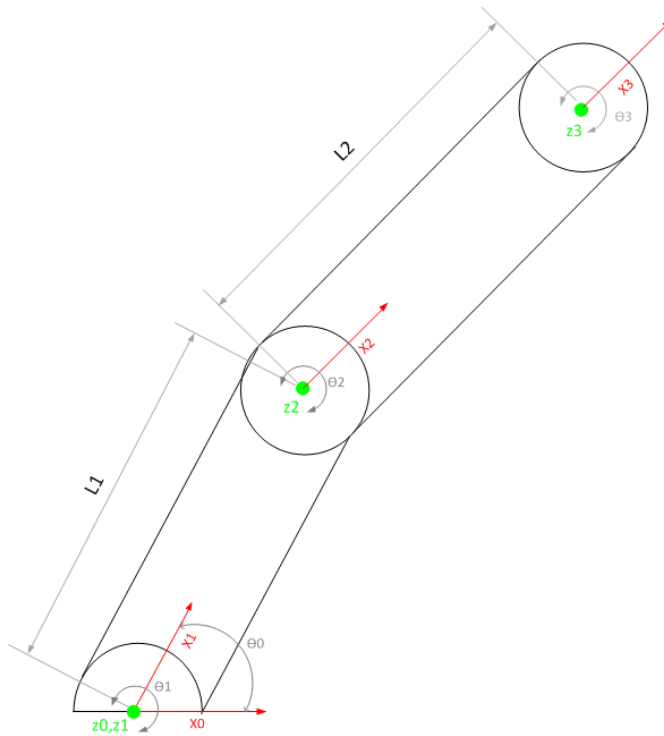
Figura 1.4: Exemple senzill del mètode proposat per Denavit i Hartenberg

Cal dir que existeix un primer sistema de coordenades del qual parteix el model articulat i que no té perquè està subjecte a una massa. Aquest sistema base del model és el sistema pel qual es comença a descriure en la taula D-H. Una vegada hem col·locat tots els sistemes de referència, la taula D-H descriu les transformacions necessàries per anar d'un sistema de referència $i - 1$ al i . Aquesta taula té quatre columnes que representen una translació i una rotació sobre l'eix X i translació i una rotació sobre l'eix Z. L'aplicació de les transformacions es realitza en aquest ordre i se solen anomenar A_{i-1} , α_{i-1} , D_i i θ_i .

Joint	A_{i-1}	α_{i-1}	D_i	θ_i
1	$r * A_w$	0	D_w	$r * \Theta_{k1} - (\pi / 2) * j$
2	A_1	$\pi/2$	0	$\Theta_{k2} + \Phi_2$
3	A_2	0	0	$\Theta_{k3} + \Phi_3$
4	A_3	$-\pi/2$	D_3	0

Figura 1.5: Exemple de taula DH

La primera columna indica l'articulació i . Un model senzill a mode d'exemple és el següent:



(a) Exemple senzill: Esquema

Massa i	A_{i-1}	α_{i-1}	D_i	θ_i
1	0	0	0	θ_0
2	L_1	0	0	θ_1
3	L_2	0	0	θ_2

(b) Exemple senzill: Taula D-H

Amb aquest petit exemple es pot veure com es descriu una cadena, un conjunt de masses i "links" lligats uns amb els altres sense formar una cadena tancada. Però, un sistema articulat està format per un conjunt de cadenes i no per una sola, en altres paraules, vol dir que en algun punt del model articulat tenim més d'una cadena que parteix d'aquell punt. Com s'ha explicat en punts anteriors això justifica que s'hagi d'usar l'element conjunt de cadenes per poder descriure aquest comportament.

Cal doncs, que en aquests conjunts de cadenes tinguem la taula o les taules que van del centre del model articulat (on hi ha el sistema de referència base del que parteix el model) fins a les masses d'aquests conjunts de cadenes. La idea serà tenir una estructura de dades que pugui mantenir aquesta informació i permeti fer les consultes necessàries, com determinar les transformacions per arribar a una massa concreta. Una cas simple a mode d'exemple podria ser el tronc superior d'un humanoide:

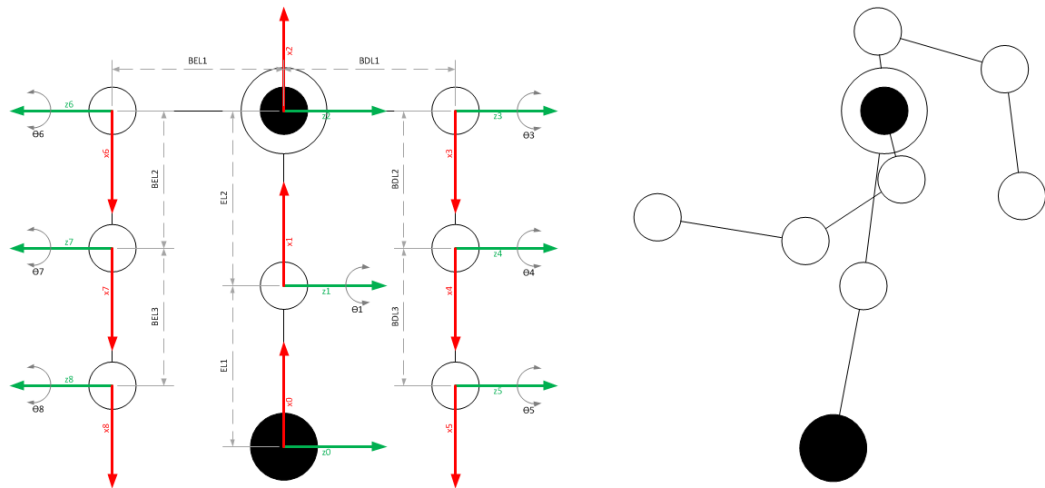


Figura 1.6: Model articulat senzill: Tronc superior d'un humanoide

En aquest exemple, es pot veure com existeixen tres cadenes, una que surt del sistema de referència base (rodona negra) i acaba al conjunt de cadenes (rodona amb un punt al mig) que té com a sistema de referència el 2, i que vindria a ser l'esquena. Mentre que les dues altres cadenes, els braços, surten del conjunt de cadenes. A la figura 1.6 ens mostra les mides i els sistemes de referència ja col·locats en les masses, en el centre de massa i el conjunt de cadenes. Aleshores, la taula D-H que descriu l'esquena és la següent:

Massa i	A_{i-1}	α_{i-1}	D_i	θ_i
1	EL1	0	0	θ_1
2	EL2	0	0	0

Taula 1.1: Taula D-H de l'esquena de la figura 1.6

Una vegada s'ha determinat l'esquena, les transformacions que s'aplicaran per arribar al conjunt de cadenes formen part de la descripció dels braços i no es pot omitir. A continuació es mostra les taules dels braços amb la informació parcial, és a dir, la informació per arribar del conjunt de cadenes descrit amb el sistema de referència 2 fins a la última massa que contenen. Així doncs, les taules tindran una forma com la següent:

Massa i	A_{i-1}	α_{i-1}	D_i	θ_i
6	0	-180	BEL1	$180 + \theta_6$
7	BEL2	0	0	θ_7
8	BEL3	0	0	θ_8

(a) Taula D-H del braç esquerra

Massa i	A_{i-1}	α_{i-1}	D_i	θ_i
3	0	0	BDL1	θ_3
4	BDL2	0	0	θ_4
5	BDL3	0	0	θ_5

(b) Taula D-H del braç dret

Figura 1.7: Taules D-H dels braços de la figura 1.6

Amb aquest exemple, dóna unes nocions de quina estructura podem tenir per emmagatzemar tot això. Podria ser en forma d'arbres on té una arrel principal amb l'element base del model articulat, nodes que representin les masses i els "links", que només tindrien un fill i un pare, i els nodes que representarien un conjunt de cadenes que tindrien un sol pare, i un conjunt finit de fills. De totes maneres, d'això en parlaré més endavant, en els capítols/apartats d'anàlisi i disseny.

Una vegada tenim com parametritzar un model articulat, es necessita un mecanisme que de manera automàtica ens calculi la taula D-H del model.

L'usuari no té perquè saber com parametritzar un model, ni conèixer el significat de cada un dels paràmetres de la taula D-H. Així doncs, caldrà analitzar quines són les possibilitats que ens podem trobar en una cadena pel que fa a les relacions d'elements, ja siguin masses o "links".

1.7.5 Obtenció dels paràmetres de la taula D-H

Cal partir de la idea que el sistema permetrà entrar a l'usuari els angles de rotació (R_x, R_y, R_z) entre dos sistema de referència, el nou element que s'afegeixi respecte l'últim entrat en una cadena. Com també les translacions que pertocuin. Així doncs, en el moment en que l'usuari desitgi introduir un element qualsevol, el sistema sabrà en tot moment quina és la posició inicial que tindrà respecte el sistema de referència de l'element anterior a l'afegit i l'usuari només haurà d'introduir els paràmetres que li interessin per obtenir el model que està dissenyant. El sistema automàticament identificarà en quin cas es troba i reconstruirà la taula D-H per poder després consultar-la i poder realitzar càlculs complexos s'una manera més senzilla.

Com s'ha comentat en el paràgraf anterior, els angles de rotació aniran en funció de l'últim sistema de referència existent a la cadena, i representaran l'orientació de l'element que s'afegeixi a continuació i en conseqüència també ho farà el sistema de referència que el defineix. Així doncs, les transformacions que s'aplicaran al nou element que s'afegeixi aniran en funció d'un sistema de referència fixe.

L'avantatge d'aquesta manera d'entrar aquestes dades, és que l'usuari de manera intuïtiva introdueix informació pròpia de l'estructura del model articulats i no necessita tenir coneixement de les transformacions geomètriques que es necessiten per descriure el model. Per altre banda, el fet de que l'usuari no necessiti conèixer tota aquesta informació cal que el sistema faci aquesta feina. El que comporta tot això és que necessitem passar d'angles de rotació en els tres eixos a descriure-ho en diferents rotacions però només en dos eixos, l'X i el Z, que són els que permeten la taula D-H. Aquí entra en joc els angles d'Euler, que no només donen els angles de rotació només en els eixos X i Z, sinó que també són els angles que s'apliquen en un sistema de referència mòbil. El que significa és que una transformació s'aplica al sistema de referència modificat i no al fixe com els angles que introdueix al principi l'usuari. Aquesta última característica és imprescindible ja que a la

taula D-H totes les transformacions es fan en funció del sistema de referència mòbil. Així doncs, el problema que cal resoldre és trobar com passar d'angles donats en XYZ i un sistema de referència fixe a angles d'Euler amb un sistema de referència mòbil.

Per poder veure com es troben els angles d'Euler, es presenta un codi en MATLAB que ho calcula:

Listing 1.1: Codi Matlab que passa els angles XYZ d'una rotació amb un S.I. fix a Angles d'Euler

```

clear all;

%1. Valors d'entrada, angles en XYZ en funcio del sistema de referencia
%global i fix

angleZ = pi/4; %45 graus
angleY = pi; %180 graus
angleX = pi/2; %90 graus

%2. Construccio de la matriu de transformacio
SR = [1 0 0 0
      0 1 0 0
      0 0 1 0
      0 0 0 1];

Rotz = [cos(angleZ) -sin(angleZ) 0 0
        sin(angleZ) cos(angleZ) 0 0
        0 0 1 0
        0 0 0 1];

Roty = [cos(angleY) 0 sin(angleY) 0
        0 1 0 0
        -sin(angleY) 0 cos(angleY) 0
        0 0 0 1];

Rotx = [1 0 0 0
        0 cos(angleX) -sin(angleX) 0
        0 sin(angleX) cos(angleX) 0
        0 0 0 1];

```

```

SR1 = SR*Rotx*Roty*Rotz; %Essent una transformacio amb sistema de
                          %referencia fix esmultiplica per la dreta

%3. Calculem els angles d'Euler
if(angleX == 0 && angleY == 0)
    euler(1) = 0;
    euler(2) = 0;
    euler(3) = angleZ;
else
    euler(1) = -atan2(SR1(3,1), -SR1(3,2));
    euler(2) = -acos(SR1(3,3));
    euler(3) = -atan2(SR1(1,3), SR1(2,3));
end

%4. Comprovacio
Rotz1 = [cos(euler(1)) -sin(euler(1)) 0 0
        sin(euler(1))  cos(euler(1)) 0 0
         0              0              1 0
         0              0              0 1];

Rotx = [1 0 0 0
        0 cos(euler(2)) -sin(euler(2)) 0
        0 sin(euler(2))  cos(euler(2)) 0
        0 0 0 1];

Rotz2 = [cos(euler(3)) -sin(euler(3)) 0 0
        sin(euler(3))  cos(euler(3)) 0 0
         0              0              1 0
         0              0              0 1];

SR2 = Rotz2*Rotx*Rotz1*SR; %Al ser un sistema de referencia mobil
                          %si li apliquem transformacions es
                          %multiplica per l'esquerra

error = 0;
for i = 1:1:3
    for j = 1:1:3
        diff = SR1(i,j) - SR2(i,j);
        if(sqrt(diff*diff) > 0.000001)
            error = -1;
        end
    end
end

```

end

error %0 si el calcul es correcta, -1 si no ho es

Com es pot veure, obtenim els angles de transformació en els eixos ZZZ, d'aquesta manera, podem també construir la taula D-H amb aquests mateixos angles, ja que només té en compte 2 eixos, i a més a més, són transformacions que fan referència a un sistema de referència que canvia amb cada una de les transformacions que descriuen aquestes tres rotacions. Per tant, només caldrà determinar dos cassos, quan hi ha translació o no. Sense translació els paràmetres que descriuran la taula DH vindran determinats per aquest mateix mètode i amb translació, el que es farà és determinar els paràmetres sense la translació i després, en última instància, realitzar la translació.

Per poder veure millor quin serà el mecanisme descrit anteriorment es presenta el següent exemple visual:

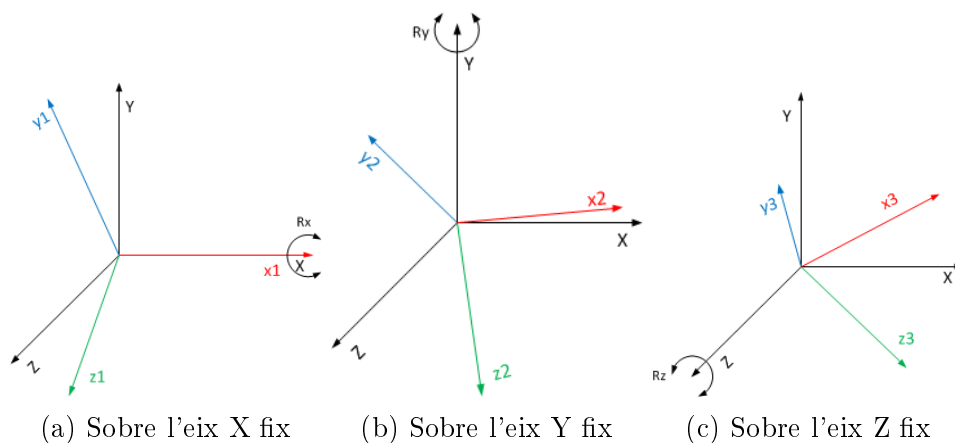


Figura 1.8: Rotacions a partir d'un sistema de referència fixe

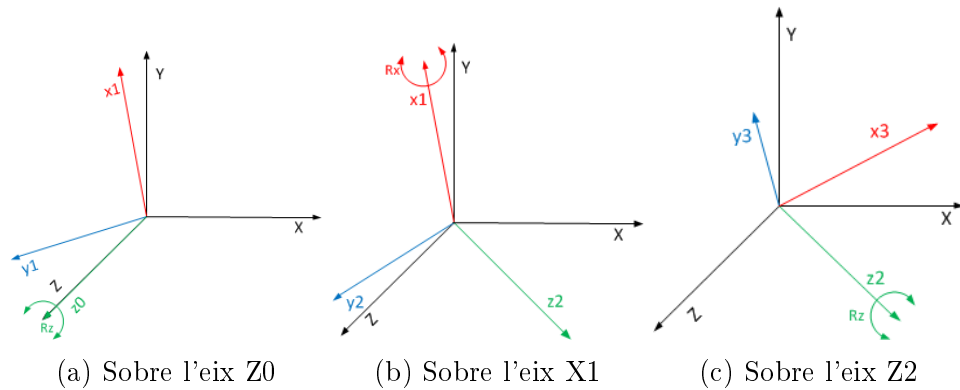


Figura 1.9: Rotacions a partir del sistema de referència que descriu el cos

Es pot veure que en el primer cas es pren un sistema de referència fix del qual es realitzen les transformacions. En el segon cas, les transformacions es realitzen sobre el sistema de referència que descriu el cos i que per tant es modifica en cada una de les transformacions, però com es pot veure en aquest cas la transformació seguint el procediment del primer cas (sistema de referència fix) dóna el mateix resultat que aplicant el del segon cas. Així doncs, el cas pràctic per al qual usarem aquest cas consistirà en obtenir els angles XYZ descrits en el primer cas, i calcularem els angles $ZX'Z''$ amb els quals podem usar per descriure el model articulat a la taula D-H.

A continuació es presentaran dos cassos en els quals es poden trobar i que cobreixen la resta de cassos. El primer cas consisteix en una connexió entre una massa i una altre massa. La següent figura es mostra un esquema d'aquest primer cas:

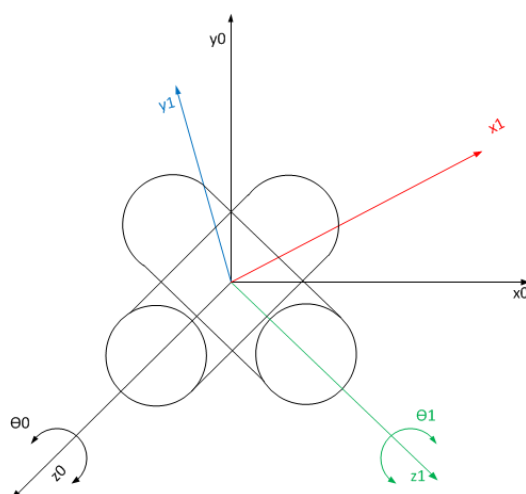


Figura 1.10: cas 1: Modelat de la connexió entre una massa amb una altre massa

Partim del cas que tenim els angles XYZ i que obtenim els valors de α , β i γ corresponents als angles de les rotacions sobre l'eix Z , X' i Z'' respectivament del sistema de referència mòbil que defineix el cos. Cal comentar, que a la figura es veu reflectit l'angle θ corresponen a la mobilitat de la massa i en la taula que es presentarà a continuació no es veurà reflectit, ja que es parla dels angles que descriuen l'estructura rígida del sistema articulat. S'han presentat per poder parlar de la diferència conceptual que existeix entre la rotació de Z que donarà l'usuari per descriure l'estructura del model articulat i l'angle que descriu el recorregut de la massa que li correspon. Així doncs, la taula D-H corresponen a aquesta connexió és la següent:

Massa i	A_{i-1}	α_{i-1}	D_i	θ_i
1	0	0	0	α
1	0	β	0	γ

Taula 1.2: Taula D-H del cas 1

El segon cas consisteix en incloure una translació, és a dir, en connexions "massa-link", "link-link" o "link-massa". La idea és la mateixa, caldrà trobar el sistema de referència just abans de la translació, col·locant l'eix X en la direcció de de la translació i després reflectir-la a la taula D-H. La figura següent mostra un cas concret, el de "massa-link":

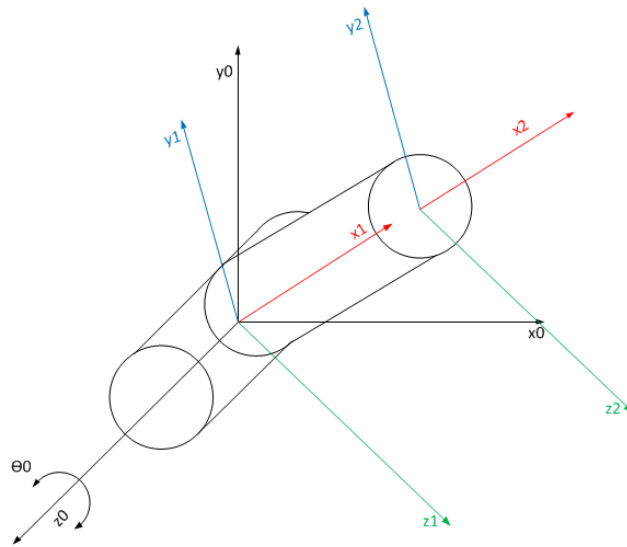


Figura 1.11: cas 2: Modelat de la connexió entre una massa amb un "link"

Si observem aquest cas, la descripció del model és el mateix que en el cas 1 amb la única diferència que cal afegir una translació aplicada a l'eix x_1 en aquest cas. Així doncs, la taula D-H que descriu aquest cas, suposant que la longitud del "link" és L_1 , és la següent:

Massa i	A_{i-1}	α_{i-1}	D_i	θ_i
1	0	0	0	α
1	0	β	0	γ
2	L_1	0	0	0

Taula 1.3: Taula D-H del cas 2

A partir d'aquí, es pot modelitzar mitjançant una interfície interactiva on l'usuari de manera intuïtiva pot construir el model sense tenir coneixement sobre el conveni descrit per Denavit i Hartenberg. I en tot moment conèixer la taula D-H del sistema i treure de manera directa les transformacions que porten d'un punt a l'altre.

Una vegada explicat com modelitzar un sistema a voluntat i tenir com s'interpreten els diferents paràmetres que els descriuen podem abordar el problema que ens plantegem.

1.7.6 Descripció del problema i solució

El problema consisteix en determinar el moviment que seguirà model articulat a partir de diferents posicions i estats del model en uns instants de temps determinats. Per tant, per cada instant de temps cal donar el valor dels angles del sistema articulat que en determinen la mobilitat de les seves masses associades a una articulació.

Cal tenir en compte que el sistema ha d'interpretar col·lisions i donar-li a l'usuari la capacitat de donar informació en aquestes col·lisions. Se'ls ha anomenat punts de contacte. L'usuari ha de permetre indicar aquestes dades al sistema perquè aquest pugui calcular el moviment. Així doncs, les dades que tenim d'entrada per cada un dels instants de temps del moviment són:

- El valor dels diferents graus de llibertat del model
- La posició i la orientació del model articulat
- El punt o punts de contacte del model amb l'entorn

Cal dir que a tot això se li afegeix tota la descripció del model, és a dir, la seva estructura. Les taules D-H que descriuen el model donen informació vital per poder resoldre el problema. De totes maneres una vegada s'ha dissenyat el model deixa de ser un element variable i passa a ser una estructura de la qual no varia amb el temps a diferència de les dades nomenades anteriorment.

La manera com s'abordarà el problema i com es tractaran les dades d'entrada s'explica a continuació. Primer es farà un anàlisi de com tractar la informació referent a la modelització del model articulat i després els diferents passos que seguirà per arribar a la solució final.

Pas previ: Generalització taula D-H

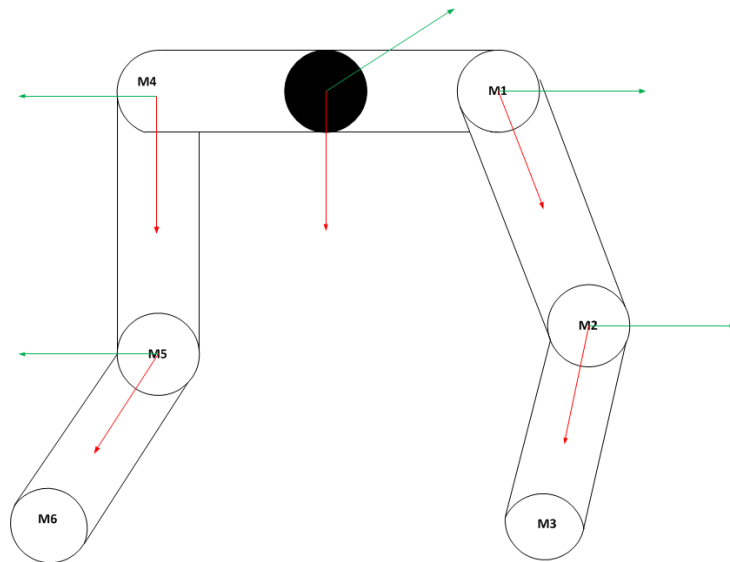
La interpretació del model consistirà en tenir N_m punts de massa i N graus de llibertat, inclouen els 6 graus de llibertat que corresponen al sistema base del model, els tres per la translació (x, y, z) i els tres de rotació (R_x, R_y, R_z) .

Ja s'ha descrit en apartats anteriors com un model articulat pot tenir un conjunt de taules D-H, però per descriure una sola massa es pot fer ús d'una sola taula que porti del sistema base al sistema de referència de la massa m_i que es vol descriure. Així doncs, cada una de les masses m_i per $i = 1..N_m$ té la seva pròpia taula $T(i)$ on,

$$T(i) = {}^0_iT = {}^0_BT * {}^B_1T * {}^1_2T * \dots * {}^{i-1}_iT =$$

$${}^0_BT(x, y, z, \theta_z^E, \theta_y^E, \theta_x^E) * T^B(\alpha_B) * T(\theta_r) * T(\alpha_1) * T(\theta_2) * \dots * T(\alpha_{i-1}) * T(\theta_i) * T(\alpha_i)$$

on $T(\alpha)$ són les matrius constants, equivalents a les parts de la taula D-H no variables. A mode d'exemple per poder entendre què és el que es pretén descriure tenim el següent cas:



(a) Esquema del model d'unes cames

M_i	A_{i-1}	α_{i-1}	D_i	θ_i
1	0	90	L1	θ_1
2	L2	0	0	θ_2
3	L3	0	0	0

(b) Taula D-H de la cama dreta

M_i	A_{i-1}	α_{i-1}	D_i	θ_i
4	0	-90	L1	θ_3
5	L2	0	0	θ_4
6	L3	0	0	0

(c) Taula D-H de la cama esquerra

Figura 1.12: Exemple Senzill: Cames

Una vegada veiem aquest exemple senzill podem començar a identificar les transformacions que es descriuen en els paràgrafs anteriors. Una primera transformació que cal destacar és la que porta del sistema de referència del món al sistema de referència del model. Per tant, tindrem una primera transformació ${}^0_B T$:

$$M_B : {}^0_B T = \left\{ \begin{array}{c|c} R^{Euler} & \begin{array}{c} x \\ y \\ z \end{array} \\ \hline 0 & 0 & 0 & 1 \end{array} \right\}$$

Una vegada es té la transformació base que porta a la massa base del model, i seguint la informació que aporten les taules D-H descrites anteriorment, es pot donar les transformacions que porten a cada una de les masses que conté aquest model presentat:

$$\begin{aligned} m1 : T(1) &= {}^0_B T(x, y, z, \theta_z^E, \theta_y^E, \theta_x^E) * R(x, 90) * T(z, L1) \\ m2 : T(2) &= T(1) * R(z, \theta_1) T(x, L2) \\ m3 : T(3) &= T(2) * R(z, \theta_2) T(x, L3) \\ m4 : T(4) &= {}^0_B T(x, y, z, \theta_z^E, \theta_y^E, \theta_x^E) * R(x, -90) * T(z, L1) \\ m5 : T(5) &= T(4) * R(z, \theta_1) T(x, L2) \\ m6 : T(6) &= T(5) * R(z, \theta_2) T(x, L3) \end{aligned}$$

D'aquesta manera el que aconseguim és que per cada massa m_i tenim una matriu de transformació $T(i)$ constant i una altre variable degut al grau de llibertat θ_i . Una vegada explicat això, podem prosseguir en el problema i veure els passos que se seguirà per a la seva resolució.

Pas 1: Trajectòria descriptiva d'entrada i editada per l'usuari

En aquest punt l'usuari ha introduït ja tota la informació prèvia, relativa a l'estructura del sistema estudiat. Per a generar un moviment determinat, l'usuari entrarà informació addicional que consistirà en diferents paràmetres que a continuació es descriuran i que determinaran la trajectòria del Centre de Massa que l'usuari vol que segueixi el model, així com també el punt o punts de contacte amb l'entorn que té el model en un instant donat. Cal tenir en compte que les característiques de la trajectòria del CM depenen fortament del nombre de punts de contacte (de recolzament, en definitiva) amb l'exterior. Per exemple, podem tenir fases de impuls, o de vol lliure. En aquestes, el moviment del CM queda determinat per les condicions inicials, sense que el model pugui alterar-la. En canvi, en una fase de impuls el model pot actuar amb l'entorn per a guiar el moviment en la trajectòria desitjada. Les forces de contacte, però, no només modifiquen la trajectòria del CM, sinó que modifiquen també el valor del moment angular. Amb tota aquesta informació es pot calcular el moviment, tot hi que poden aparèixer efectes no esperats per l'usuari i que fan que el moviment no es faci com aquest espera, mantenint, de totes maneres, que el moviment compleix la física del sistema.

A continuació s'explicarà quins paràmetres pot determinar l'usuari per cada un dels instants de temps determinats i fins hi tot, la combinació de diferents configuracions del model, es poden agrupar per formar etapes.

Durant l'especificació del problema es parlarà de punts clau, un Punt Clau és un configuració del model, que consisteix en determinar el valor de cada un dels seus graus de llibertat (posició, orientació i valor de les articulacions del model) en un temps determinat i amb cap, un o més d'un punt de contacte. També, apareixerà el concepte de etapa. Una etapa està constituïda per més d'un Punt Clau, que tenen en comú que durant aquesta etapa el conjunt de punts de contacte, no varia. Que no varii un conjunt de punts de contacte, implica que una extremitat del model estigui constantment en contacte en el mateix punt. A la pràctica, es pot considerar que un Punt Clau pot formar part de dues etapes, si considerem que la mateixa configuració podria tenir

una representació del conjunt dels punts de contacte compatible amb les dues etapes. A mode d'exemple d'un Punt Clau, d'una Etapa i d'un Punt Clau que contingui una configuració compatible amb dos Etapes diferents es presenten les següents il·lustracions d'un moviment que consisteix en prendre impuls amb els dos peus i saltar amunt en vertical:

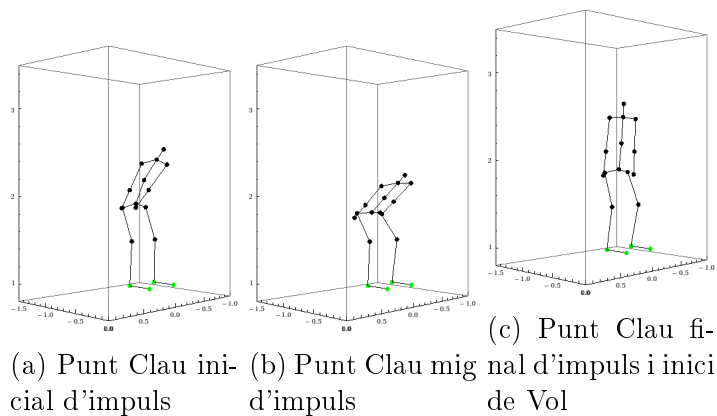


Figura 1.13: Etapa 1: Impuls

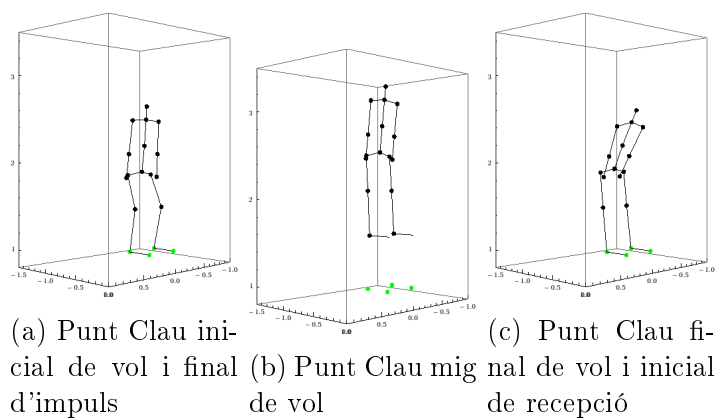


Figura 1.14: Etapa 2: Vol

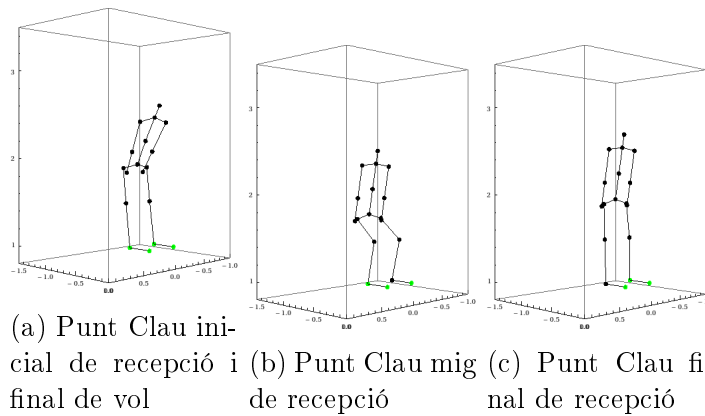


Figura 1.15: Etapa 3: Recepció

En aquest exemple, tenim tres etapes, la de impuls, de vol i la de recepció. Amb tres punts claus cada un. Com que tenim dos punts claus comuns a dues etapes diferents, tenim un total de 7 punts claus. Aleshores, per determinar les etapes, cal mirar si el conjunt de punts de contacte és vàlid durant una etapa en concret, a la etapa 1, es pot veure amb verd, els punts de contacte que l'usuari ha entrat i que són comuns en tots tres punts claus. Per altre banda, veiem que en l'etapa 2, els punts de contacte no són els mateixos, però per definició una etapa conté punts claus que el conjunt de punts de contacte o el subconjunt d'aquests és comú o es pot trobar una combinació comuna en cada un dels punts claus, en aquest cas, el conjunt comú a tots tres punts claus, és el buit, és a dir, que no hi ha punts de contacte, d'aquí la definició de etapa de vol.

En el moment en que apareix un seguit d'etapes i que tenim diferents punts clau per cada una de les etapes, apareix la necessitat d'interpol·lar mitjançant algun mecanisme. La manera com es farà és modelitzar el Centre de Massa del model de tal manera que per cada temps i tenim un valor del Centre de Massa. La posició del Centre de Massa d'un model és calcula de la següent manera:

$$\vec{R}_{CM} = \frac{\sum_{i=1}^{N_{masses}} m_i \vec{r}_i}{\sum_{i=1}^{N_{masses}} m_i} \quad (1.6)$$

La posició del Centre de Massa es calcula com la suma de la posició per la massa de totes les partícules dividit per la massa total. Una vegada sabem la definició del Centre de Massa, cal determinar quins paràmetres defineixen una etapa i que en determinen la trajectòria del Centre de Massa i la modelització d'aquest. Cal veure, que quan el conjunt de punts de contacte en una etapa és buit, la trajectòria queda determinada només amb els Centres de Massa que es dedueixen de la configuració que l'usuari ha entrat. En aquest exemple, l'usuari ha introduït 3 posicions del Centre de Massa per tal de determinar la trajectòria completa. L'equació que segueix en una Etapa de Vol és:

$$(rx, ry, rz)(t) = (rx_0, ry_0, rz_0) + (vx_0, vy_0, vz_0)t + \frac{(0, g, 0)tc^2}{2} \quad (1.7)$$

Prenent com a exemple la figura 1.14, (rx_0, ry_0, rz_0) correspon al Centre de Massa del Punt Clau inicial de l'Etapa 2. Per altre banda, podem obtenir la velocitat inicial tenint en compte que en t_{max} (temps en la que altura és màxima) la component vertical (en el nostre sistema és la component y) és 0. Aleshores podem deduir que:

$$t1 = \sqrt{2(Ry_{CM}^{Mitg} - Ry_{CM}^{Inicial})/g} \quad (1.8)$$

$$t2 = \sqrt{2(Ry_{CM}^{Mitg} - Ry_{CM}^{Final})/g} \quad (1.9)$$

$$Vx_{CM}^{Inicial} = \frac{(Rx_{CM}^{Final} - Rx_{CM}^{Inicial})}{t1 + t2} \quad (1.10)$$

$$Vy_{CM}^{Inicial} = \sqrt{2g(Ry_{CM}^{Mitg} - Ry_{CM}^{Inicial})} \quad (1.11)$$

$$Vz_{CM}^{Inicial} = \frac{(Rz_{CM}^{Final} - Rz_{CM}^{Inicial})}{t1 + t2} \quad (1.12)$$

Cal tenir en compte aquest cas que queda determinat en principi només amb la determinació de les configuracions dels punts claus per part de l'usuari.

Per altre banda, cal considerar com modelitzar el Centre de Massa quan el conjunt de punt de contacte durant una Etapa no és buit. En aquest cas, cal considerar trajectòries que interpolin primer els Centres de Massa per poder enllaçar les diferents etapes i que les velocitats inicials i finals de les Etapes corresponguin amb les finals i inicial de les etapes anterior i posterior d'una etapa. Així doncs, cal considerar una trajectòria que deixi com a paràmetres lliures el Centre de Massa inicial i final, com també les velocitats inicial i final del Centre de Massa. I per poder controlar el Centre de Massa intermitja d'aquest tipus d'etapes, calen certs paràmetres lliures amb els que l'usuari pugui modificar la trajectòria. Aquests paràmetres són els següents:

- **T1:** El temps que dura la fase que va del Punt Clau inicial al mig.
- **T2:** El temps que dura la fase que va del Punt Clau mig al final.
- **Dm:** Distància del Centre de Massa al punt intermitg de l'àrea que determina el conjunt de punt de contactes en temps igual a T1 en coordenades locals.
- **Thm:** Velocitat de pas en temps igual a T1 en coordenades locals.
- **Th1:** Inclinió respecte l'eix Z en el temps igual a T1.
- **Th2:** Inclinió respecte l'eix X en el temps igual a T1.

Per entendre aquest tipus d'interpolació, cal tenir en compte que es parteix de la base que és modelitza agafant com a centre de coordenades el punt de contacte promig d'una àrea de contacte per una etapa concreta. Així doncs, usarem un Spline per modelitzar el Centre de Massa que tingui com a paràmetres lliures tots els descrits anteriorment. Cal primer determinar els Centres de Massa respecte el nou origen del Spline. Després calcular els eixos del nou sistema de coordenades. I finalment, calcular el Centre de Massa en coordenades locals i aplicar-l'hi les transformacions necessàries per passar-lo al sistema de laboratori. A mode d'exemple d'aquest cas, es mostra en pseudocodi, de manera que es pugui veure més clar el tractament d'aquest tipus de trajectòries. Cal veure per això que es separa l'etapa en dues fases, abans de la configuració intermitja donada per l'usuari i després d'aquesta:

```

(*Primera part determinació de l'origen de coordenades del nou sistema de coordenades*)
FaseIpContacte = (PCOPuntCOB + PCOPuntCOP + PCOPuntCIB + PCOPuntCIP) / 4;
valors = {0.3, 0.15, 0.5, 0.94, 0, 0};
t1 = valors[[1]]; t2 = t1 + valors[[2]]; tm = t1; t10 = t1; t21 = t2 - t1;
dmOpt = valors[[4]]; vtOpt = valors[[3]]; e1Opt = valors[[5]]; e2Opt = valors[[6]];
VcmRepos = {0, 0, 0};

(*Càlcul dels Centres de Massa respecte el nou centre de coordenades i càlcul de eixos (n1,n31,n3)*)
r1D3 = {Rcm0 - FaseIpContacte} // Flatten; v1D3 = {0, 0, 0} // Flatten; r2D3 = {Rcm2 - FaseIpContacte} // Flatten; v2D3 = {Vcm2} // Flatten;
n1 = r1D3 / Sqrt[r1D3.r1D3]; n2 = r2D3 / Sqrt[r2D3.r2D3]; n3 = Normalize[n1 * n2]; n31 = n3 * n1; n32 = n3 * n2;
(*Càlcul de l'"Spline" amb paràmetres lliures posició i velocitat en instant t, amb Dm i Tthm(vt) lliures*)
posicio3D[dm_, vt_, x1_, y1_, z1_, vx1_, vy1_, vz1_, t1_, t_] :=
{dm + t^2 * (-vt^2 / (2 dm)) - (6 dm^2 - t1^2 vt^2 + 2 dm t1 vx1 - 6 dm x1) / (2 dm t1^2) +
t^3 ((vt (2 t1 vt + t1 vy1 - 3 y1)) / (dm t1^2) - (-2 dm^2 + 2 t1^2 vt^2 - dm t1 vx1 + t1^2 vt vy1 + 2 dm x1 - 3 t1 vt y1) / (dm t1^3)),
t vt - (t^2 (2 t1 vt + t1 vy1 - 3 y1)) / t1^2 +
t^3 (-vt^3 / (6 dm^2)) - (vt (6 dm^2 - t1^2 vt^2 + 2 dm t1 vx1 - 6 dm x1)) / (2 dm^2 t1^2) -
(-12 dm^2 t1 vt + t1^3 vt^3 - 3 dm t1^2 vt vx1 - 3 dm^2 t1 vy1 + 9 dm t1 vt x1 + 6 dm^2 y1) / (3 dm^2 t1^3)),
(t^3 (t1 vz1 - 2 z1)) / t1^3 + (t^2 (-t1 vz1 + 3 z1)) / t1^2}
trajmod[ec_, bc_, dm_, vt_, r1_, v1_, t1_] := Module[{x1, y1, z1, vx1, vy1, vz1, reslocal, d2sign, result, comp, comp2},
{x1, y1, z1} = Transpose[Transpose[{n1, n31, n3}.r3z[ec].r3x[bc]].r1];
{vx1, vy1, vz1} = Transpose[Transpose[{n1, n31, n3}.r3z[ec].r3x[bc]].v1];
reslocal = posicio3D[dm, vt, x1, y1, z1, vx1, vy1, vz1, t1, tt];
result = Transpose[{n1, n31, n3}.r3z[ec].r3x[bc].reslocal;
Return[{result, d2sign}];]

(*Càlcul de la fase 1 i 2 prenent com a Punt Clau Mitg amb t = 0*)
trajfe = (trajmod[e1, e2, dm, vt, r2D3, v2D3, te][[1]]);
trajfp = (trajmod[e1, e2, dm, vt, r1D3, v1D3, -tp][[1]]);
(*Preparació de la trajectòria per un millor ús*)
trajectoria1a = trajfp /. {e1 -> e1Opt, e2 -> e2Opt, dm -> dmOpt, vt -> vtOpt, tp -> t10};
trajectoria2a = trajfe /. {e1 -> e1Opt, e2 -> e2Opt, dm -> dmOpt, vt -> vtOpt, te -> t21};
trajectoria1[t_] := (trajectoria1a /. {tt -> t}) + FaseIpContacte;
trajectoria2[t_] := (trajectoria2a /. {tt -> t}) + FaseIpContacte;
trajectoria = If[t <= tm, trajectoria1[t - tm], trajectoria2[t - tm]];

```

Figura 1.16: Pseudocodi del càlcul de la Trajectòria del Centre de Massa en una Etapa de Contacte

L'algorisme descrit en el Pseudocodi, l'únic que fa és dir-li que en l'instant t_1 es compleixi que el Centre de Massa i la velocitat d'aquesta siguin les d'entrada a la funció i que en t igual a 0 es compleixi que la distancia i el t_{hm} (vt) prengui els valors determinats per l'usuari. Una vegada cridada aquesta funció només cal aplicar les transformacions necessàries per obtenir la coordenada en el sistema de laboratori. Cal després determinar en quin temps es dona aquest valor, ja que la funció dona valors de temps compresos entre $-t_1$ i t_2 , tal i com estant definides. Per això, el que es fa és que a un temps concret durant l'animació, se li resta el valor de temps del primer Punt Clau d'una Etapa concreta i després es distingeix de quina fase es tracta i se li resta el temps t_1 . D'aquesta manera, la trajectòria és exportable a qualsevol temps i no s'ha d'estar pendent de si abans d'aquesta etapa n'hi havia una altre o no.

Una vegada determinada tota l'entrada que farà l'usuari, cal veure com de tota aquesta informació s'en deriva la següent i que es necessària pels càlculs que comporta l'algoritme, com el moment angular o les forces de contacte que estant actuant en cada instant de temps.

Pas 2: Càlcul del moment angular i de la força de contacte

En el cas de tenir un o més punts de contacte amb l'entorn, el model podrà originar forces de contacte exercides per l'entorn en cada un d'aquests punts, de la mateixa manera que en l'experiència diària podem preparar un salt o una altra acció física. Si bé pel que fa al moviment del Centre de massa, el conjunt de forces de contacte és equivalent a la força resultant total, l'afectació causada sobre el moment angular depèn dels punts on cada una d'aquestes forces actua. Si només hi ha un punt de contacte, llavors l'efecte sobre el moment angular queda completament determinat per la força resultant i aquest punt de contacte. En cas de dos o més punts de contacte, l'efecte queda també determinat per la força resultant, més un punt de contacte equivalent, que queda localitzat en una regió de l'entorn delimitada entre la totalitat de punts de contacte. Aquest punt de contacte equivalent, tant pot ser proporcionat per l'usuari com determinat pel mateix programa. En aquest pas es veurà com s'obté el moment angular passant per la determinació del punt de contacte òptim que minimitza aquest moment angular.

El moment angular en sistemes articulats bé definit per diferents fórmules:

$$\vec{L}(t) = \sum_{i=1}^{N_{masses}} m_i \vec{r}_i(t) \times \vec{v}_i(t) \quad (1.13)$$

$$\vec{L}(t) = \vec{L}_{entorn} + \vec{L}_{CM} \quad (1.14)$$

$$\vec{L}_{entorn}(t) = \sum_{i=1}^{N_{masses}} m_i (\vec{r}_i(t) - \vec{R}_{CM}(t)) \times \vec{v}_i(t) \quad (1.15)$$

$$\vec{L}_{CM}(t) = \sum_{i=1}^{N_{masses}} m_i \vec{R}_{CM} \times \vec{V}_{CM} \quad (1.16)$$

on,

$$\vec{V}_{CM} = \frac{\sum_{i=1}^{N_{masses}} m_i \vec{v}_i}{\sum_{i=1}^{N_{masses}} m_i} \quad (1.17)$$

Com es pot veure, podem obtenir el moment angular a partir dels valors de posició i velocitat de cada una de les masses que contingui el model. De totes maneres, si no es té el moviment tal hi com es voldria, a partir d'aquestes fórmules no es pot obtenir el moment angular. De totes maneres, com es pot veure a l'equació 1.14 i l'equació 1.16, podem calcular una part a partir de la trajectòria que s'hagi obtingut en cada una de les etapes. Tot hi això cal buscar un altre camí que ens porti a calcular l'altre part del moment angular que només es pot calcular a partir de la posició i velocitat de cada una de les masses. Aquí hi entra el moment de les forces. El moment de les forces ens indica el canvi que es produeix en el moment angular en un temps determinat. La fórmula general per a calcular el moment de les forces consisteix en la suma del producte vectorial entre una força \vec{F}_i i el punt d'aplicació \vec{r}_i :

$$\vec{NF}(t) = \sum_{i=1}^{N_{forces}} \vec{r}_i(t) \times \vec{F}_i(t) \quad (1.18)$$

o separant per moment de les forces del Centre de Masses i moment entorn del Centre de Masses, es pot escriure i és equivalent:

$$\vec{N}F(t) = \sum_{i=1}^{N_{forces}} (\vec{r}_i(t) - \vec{R}_{CM}(t)) \times \vec{F}_i(t) + \vec{R}_{CM} \times \vec{F}_T(t) \quad (1.19)$$

on,

$$\vec{F}_T(t) = \sum_{i=1}^{N_{forces}} \vec{F}_i(t) \quad (1.20)$$

o també

$$\vec{F}_T(t) = \left(\sum_{i=1}^{N_{masses}} m_i \right) \vec{A}_{CM}(t) \quad (1.21)$$

En el nostre problema, apareixen dues forces, la força de contacte i la força de la gravetat. La segona s'aplica a cada una de les masses i és equivalent a dir que s'aplica sobre el Centre de Massa del model. En canvi la primera, s'aplica en el punt de la superfície on el model està recolzat i fa la força per realitzar els moviments.

Una vegada vist tot això i deduir que la força de contacte és la força total menys la força de la graveta, la única variable que ens queda és la posició del punt de contacte. Cal veure abans que separant el moment de les forces en dues parts, obtenim la següent fórmula:

$$\vec{N}F(t) = ((\vec{r}_{puntContacte}(t) - \vec{R}_{CM}(t)) \times \vec{F}_{contacte}(t)) + \vec{R}_{CM} \times (\vec{F}_g + \vec{F}_{contacte}(t))) \quad (1.22)$$

Vist això, cal tractar el cas de la posició del punt d'aplicació de la força de contacte. Hi ha dues maneres una que l'usuari ens la doni, o la calculem nosaltres. De totes maneres, cal determinar una àrea per la qual el sistema o l'usuari pugui col·locar el punt d'aplicació de la força de contacte. Estudiem el cas, l'usuari podrà entrar per una extremitat amb recolzador el punt on vol col·locar la punta del recolzador o la base d'aquest, la base és on comença el recolzador. Com indica el nom, el recolzador és l'element per on es recolza el sistema, en el nostre cas i per simplificar podria ser un peu o una ma, tot

hi que es possible amb altres. Així doncs, un conjunt de punts de contacte, consistirà en un conjunt de punts sobre una superfície plana. És a dir, una superfície, que per tots els punts la component vertical és la mateixa (una simplificació més, degut a la complexitat del cas). Per determinar una àrea, el que es fa és calcular la envolupant convexa de tots aquests punts i es determina una àrea. L'envolupant convexa és un polígon que conté tots els punts al seu interior, de manera que cada un dels angles dels vèrtexs són convexos tal hi com indica el nom. Un exemple d'envolupant convexa és el següent:

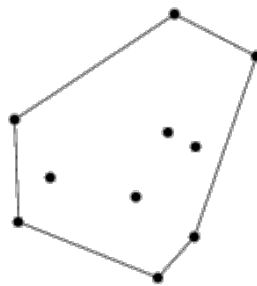


Figura 1.17: Envolupant convexa d'un conjunt de punts

Per resoldre aquest problema s'usarà una llibreria externa anomenada CGAL de tractament de geometria àmpliament usat.

Una vegada determinat com calcular l'àrea hàbil per el punt d'aplicació de la força de contacte, l'usuari ja podrà entrar un punt dins d'aquesta zona, l'únic que haurà de comprovar el sistema és que el punt que entri l'usuari estigui dins d'aquesta zona. Per altre banda, si l'usuari vol que el sistema calculi automàticament aquest punt, caldrà calcular quin és el punt d'aplicació de la força de manera que el moment de les forces es minimitzi o s'anul·li el moment angular entorn del Centre de Massa.

Si el sistema necessita calcular el punt d'aplicació de la força de contacte que minimitza o anul·la el moment angular entorn del Centre de Massa, el que cal veure és l'equació que determina el moment de les forces on les dues parts del moment de les forces estant separats en el que correspon al Centre de Massa i al de l'entorn del Centre de Massa. I veure que la part que es vol anul·lar, consisteix en la primera part on:

$$\vec{N}F_{entorn}(t) = ((\vec{r}_{puntContacte}(t) - \vec{R}_{CM}(t)) \times \vec{F}_{contacte}(t)) = \vec{0} \quad (1.23)$$

Cal resoldre doncs l'equació 1.23, sabent que coneixem la component vertical del punt d'aplicació, ja que, s'ha simplificat que tot objecte sempre té la mateixa component vertical per cada un dels seus punts, podem escriure que la resolució de la equació serà:

$$(rx_{puntContacte}(t), ry_{puntContacte}, rz_{puntContacte}(t)) = \vec{R}_{CM}(t) + \lambda \vec{F}_{contacte}(t) \quad (1.24)$$

Coneixent el valor de $ry_{puntContacte}$, podem calcular λ i després acabar de trobar els valors de les altres components del punt d'aplicació. Una vegada tenim això, el que s'obté és el punt que anul·la el moment de les forces entorn del Centre de Masses i pot ser que no estigui dins de l'àrea en la qual el model es recolza. El que cal aleshores, és portar el punt d'aplicació dins de l'àrea hàbil de manera que minimitzi aquest moment de les forces. En concret cal trobar el punt més proper al punt trobat i que estigui contingut dins de l'àrea. Aquest problema està resolt també amb la llibreria CGAL i és el problema genèric de trobar la mínima distància entre dos conjunts convexos de punts, en els quals un conjunt és un punt i l'altre l'àrea calculada a partir de l'envolupant convexa.

Una vegada tenim tot això determinat, ja podem abordar el problema del càlcul del moment angular per cada un dels instants de temps esperats. Així doncs, per poder calcular $L(t + \Delta t)$ es pot calcular de la manera següent:

$$L(t + \Delta t) = L(t) + \left(\frac{NF(t) + NF(t + \Delta t)}{2} \right) \Delta t \quad (1.25)$$

Es realitza una mitjana entre els dos moments de forces per poder tenir més precisió, si s'en volgués més es podrien fer servir altres tipus. Tot hi així d'aquesta manera es pot obtenir per cada instant de temps el moment angular sempre hi quan es tingui el moment angular en el temps igual a 0. Per simplificar, es considera que es parteix del repòs i per tant el moment angular és $(0, 0, 0)$.

Arribats a aquest punt, tenim modelitzat el Centre de Massa, el Moment Angular i els conjunt de punts de contacte per cada una de les etapes per les que passarà el model per realitzar el moviment. A continuació es presentarà quina és l'equació que ha de complir el model per tal de passar d'un instant de temps al següent.

Pas 2: Càlcul del moviment mitjançant el Jacobià i les modelitzacions anteriors

En aquest apartat, el que es presentarà és l'equació que s'ha de complir de tal manera que el model realitzi el moviment a partir de les variables modelitzades en els passos anteriors. Es veurà què és el Jacobià, com està compost i quin significat té. Com resoldre el sistema d'equacions amb el Jacobià i quins paràmetres de sortida obtindrem.

Per tal de presentar l'equació, cal primer, presentar el Jacobià i per quines parts està compost. Primer de tot, el Jacobià és una matriu que conté les derivades parcials primeres d'un conjunt de funcions. Aquestes funcions, són les del Moment Lineal (P), la del Moment Angular (L), ja vista, i la de la posició de les extremitats que estant en contacte en un moment determinat (PC_i). Per altre banda, el que s'esta buscant és quins increments s'han de realitzar en els graus de llibertat perquè la nova configuració correspongui amb els valors modelitzats. Per tant, tindrem per tantes columnes com graus de llibertat tingui el model i tantes files com funcions necessitem modelitzar. Aleshores, el Jacobià tindrà una forma semblant a la següent:

$$\begin{bmatrix} \frac{dP}{d\Theta_0} x & \frac{dP}{d\Theta_1} x & \dots & \frac{dP}{d\Theta_m} x \\ \frac{dP}{d\Theta_0} y & \frac{dP}{d\Theta_1} y & \dots & \frac{dP}{d\Theta_m} y \\ \frac{dP}{d\Theta_0} z & \frac{dP}{d\Theta_1} z & \dots & \frac{dP}{d\Theta_m} z \\ \frac{dL}{d\Theta_0} x & \frac{dL}{d\Theta_1} x & \dots & \frac{dL}{d\Theta_m} x \\ \frac{dL}{d\Theta_0} y & \frac{dL}{d\Theta_1} y & \dots & \frac{dL}{d\Theta_m} y \\ \frac{dL}{d\Theta_0} z & \frac{dL}{d\Theta_1} z & \dots & \frac{dL}{d\Theta_m} z \\ \frac{dPC_0}{d\Theta_0} x & \frac{dPC_0}{d\Theta_1} x & \dots & \frac{dPC_0}{d\Theta_m} x \\ \vdots & \vdots & \dots & \vdots \\ \frac{dPC_n}{d\Theta_0} z & \frac{dPC_n}{d\Theta_1} z & \dots & \frac{dPC_n}{d\Theta_m} z \end{bmatrix}$$

La construcció del Jacobià consistirà en detectar primer en quina fila es fa el càlcul del Jacobià per determinar-ne la funció i després calcular la derivada

parcial respecte el grau de llibertat que es correspongui en aquella columna. Així doncs, les derivades parcials per cada una de les funcions es pot representar de la manera següent:

$$\frac{dP_a}{d\Theta_j} = \sum_{i=1}^{N_{masses}} m_i \frac{dT(i)}{d\Theta_j}(a, 4) \quad (1.26)$$

$$\frac{dL_a}{d\Theta_j} = \sum_{i=1}^{N_{masses}} \varepsilon_{abc} m_i T(i)(b, 4) \frac{dT(i)}{d\Theta_j}(c, 4) \quad (1.27)$$

$$\frac{dPCk_a}{d\Theta_j} = \frac{dTPC(k)}{d\Theta_j}(a, 4) \quad (1.28)$$

En les equacions anteriors, cal mencionar que $T(i)$ correspon a la transformació que porta fins a la massa i , i que $TPC(k)$ correspon a la transformació que porta fins l'element recolzador k .

Una vegada explicat tot això, i sabem com construir el Jacobià, cal presentar l'equació que seguirà el nostre algoritme per arribar a completar tot el moviment:

$$Jacobia(Conf)\Delta\Theta = \Delta X \quad (1.29)$$

El que vol dir aquesta equació, és que a partir del Jacobià que s'obté mitjançant la configuració en un instant de temps del model i dels increments interns, podem obtenir els increments externs. En aquests cas els increments interns, correspondria amb els graus de llibertat del model, mentre que els increments externs correspondria a increments del moment lineal, angular i dels punts de les extremitats del model.

Com es pot apreciar, no tenim els increments interns, de fet, és el que necessitem per calcular la següent configuració del model. El que tenim són els increments externs, el moment angular, lineal i de la posició dels punts de contacte (que seran 0 o molt pròxims). Per tant, el sistema que es vol resoldre implica fer la inversa del Jacobià. Típicament per resoldre aquest sistema, el que es fa servir és la Pseudoinversa. En increments petits en ΔX

donarà errors petits a la solució, com que podem fer que calculi el moment angular i lineal en increments de temps tant petits com volem la solució la podem triar tant precisa com vulguem, tot hi que pot comportar més temps.

La Pseudoinversa es calcula de la manera següent:

$$Pseudoinversa(Jacobia(C)) = Jacobia(C)^T (Jacobia(C) Jacobia(C)^T)^{-1} \quad (1.30)$$

Per tant, podem calcular els increments interns de la següent manera:

$$Pseudoinversa(Jacobia(C)) \Delta X = \Delta \Theta \quad (1.31)$$

Una vegada determinat el sistema, queda explicar com calcularem les ΔX . Per definició, tenim que ΔX quedarà de la següent manera:

$$\begin{bmatrix} \Delta X_0 = P_x \Delta t \\ \Delta X_1 = P_y \Delta t \\ \Delta X_2 = P_z \Delta t \\ \Delta X_3 = L_x \Delta t \\ \Delta X_4 = L_y \Delta t \\ \Delta X_5 = L_z \Delta t \\ \Delta X_6 = \Delta PC0_x \\ \Delta X_7 = \Delta PC0_y \\ \dots \\ \Delta X_n = \Delta PCm_z \end{bmatrix}$$

Com podem veure el moment angular que hem calculat en passos anteriors, el podríem col·locar en la posició corresponent en els increments de les variables externes multiplicat per l'increment de temps triat per l'usuari. Per altre banda, també podem calcular els increments de les posicions de les extremitats, tenint en compte on ha dit l'usuari que vol els punts de contacte i a on estant en una configuració concreta. Però el Moment Lineal no el tenim. Sabem que el moment Lineal es pot calcular com:

$$\vec{P} = \sum_{i=1}^{N_{masses}} m_i \vec{v}_i \quad (1.32)$$

Si dividim P per la Massa total del model, obtenim el la V_{CM} i si ho multipliquem per un increment de temps, obtenim el canvi de la posició del Centre de Massa, que si ens hi fixem, també el podem obtenir a partir de la modelització de la trajectòria del Centre de Massa feta en els passos anteriors. De manera que el ΔX ens quedaria de la següent manera:

$$\left[\begin{array}{l} \Delta X_0 = R_{CM}x(t + \Delta t) - R_{CM}x(t) \\ \Delta X_1 = R_{CM}y(t + \Delta t) - R_{CM}y(t) \\ \Delta X_2 = R_{CM}z(t + \Delta t) - R_{CM}z(t) \\ \Delta X_3 = L_x \Delta t \\ \Delta X_4 = L_y \Delta t \\ \Delta X_5 = L_z \Delta t \\ \Delta X_6 = \Delta PC0_x \\ \Delta X_7 = \Delta PC0_y \\ \dots \\ \Delta X_n = \Delta PCm_z \end{array} \right]$$

Cal remarcar, que per poder fer això, cal aplicar el mateix procediment a l'equació 1.29 de dividir per la Massa total en el càlcul de la derivada primera parcial del moment lineal respecte el grau de llibertat j .

Pas 3: Càlcul del moviment complet. Algoritme final

Una vegada s'ha explicat tot el procediment per calcular el canvi que s'ha de produir en les variables internes del model, es bastant directe quin serà l'algoritme que determinarà tot el moviment per complet. A més a més, es presentarà alguna tècnica per intentar mantenir les configuracions marcades per l'usuari o complir les restriccions sobre els valors dels graus de llibertat.

L'algoritme final es presenta a continuació:

1. Inicialitzar variables i calcular moment angular i trajectòria de Centre de Massa.
2. Calcular els increments de les variables externes.
3. Calcular els increments de les variables internes a partir de l'anterior mitjançant la tècnica del jacobiana.
4. Sumar els increments de les variables internes a la configuració actual.

5. Si no acaba agafar com actual el següent i tornar al pas 2 sinó pas 6.
6. Calcula els paràmetres físics que pugui interessar a l'usuari.

Cal veure també com es pot fer que el sistema tendeixi cap a un cert estat o com fer que el model no sobrepassi possibles restriccions sobre els graus de llibertat del model. Per als dos cassos, cal fixar-se que es pot accelerar o frenar el ritme de canvi d'un grau de llibertat si s'aplica de la següent manera presentat amb pseudocodi:

JacobiaQ = Jacobia.DiagonalMatrix[QsValor];

PseudoInversaQ = PseudoInverse[JacobiaQ];

CanviEstatQ = DiagonalMatrix[QsValor].(PseudoInversaQ.DeltaX);

Cal vigilar en aquests cassos, ja que les Qs necessiten ser positives de lo contrari es podria comportar de manera no desitjada. Per al primer cas, si volem que durant el càlcul del moviment, el model tendeixi anar a una certa posició final, es pot fer els càlculs de les Qs de la següent manera descrit en pseudocodi:

Jacobia = CalculJacobia[estat0];

PseudoInversa = PseudoInverse[Jacobia];

CanviEstat1 = (PseudoInversa.DeltaXPhysNum);

QsValor = ((EstatDesti – EstatAct)/CanviEstat1)//Quiet;

Do[QsValor[[k]] = If[QsValor[[k]] ∈ Reals&&QsValor[[k]] > 0, Sqrt[QsValor[[k]]], 0], {k, 1, Ndof}];

De manera resumida, el que fa és veure quin increment li faria sense les Qs per cada un dels graus de llibertat i veure quin increment necessita per assolir el destí i dividir el segon per el primer per tal d'accelerar si encara li falta o frenar quan es passa.

L'altre cas el que cal intentar és que el model no tingui algun grau de llibertat fora del rang de valors possibles indicat per l'usuari. En aquest cas valdria qualsevol funció que tingués una Q petita quan s'aproximi al límit i una Q relativament gran quan estigui pel mig del rang. El procediment d'aplicació després de les Q s seria exactament igual al primer fragment de codi en pseudocodi presentat.

Donada la trajectòria genèrica $\theta_i(t)$, que s'ha calculat en el pas 1 com a una trajectòria interpolada, es calcula les forces generalitzades $F(i)$ per $i = 1, \dots, N$ de cada articulació incloent la força resultant total $F_t^a(t)$.

Les lleis que s'usaran per trobar les forces F_α^a exteriors són:

$$F_t^a(t) = \sum_{i=1}^{N_m} m_i g^a + \sum_{\alpha=1}^{N_r} F_\alpha^a(t) \quad (1.33)$$

$$\frac{dL_t^a}{dt} = \sum_{i=1}^{N_m} m_i r_i^b g^c \epsilon^{abc} + \sum_{\alpha=1}^{N_r} \epsilon^{abc} r_\alpha^b F_\alpha^c(t) \quad (1.34)$$

Pas 4: Càlcul dels paràmetres físics que estudien el moviment

Arribats a aquest punt, tenim la trajectòria definida per cada un dels temps determinats per l'usuari, això implica que per cada instant de temps tenim la configuració que tindrà el model. Això comporta que puguem calcular certs paràmetres que després ens seran útils per veure com actuen les forces durant un moviment o quina articulació és la que pateix més... Una vegada obtenim els valors que han de tenir els graus de llibertat, pot ser interessant guardar informació de cara a l'anàlisi del moviment. Així doncs, per obtenir les velocitats com també les acceleracions en cada una de les articulacions es pot usar les següents operacions:

$$\dot{\theta}_i(t) = \frac{\theta_i(t + \Delta t) - \theta_i(t - \Delta t)}{2\Delta t} \quad (1.35)$$

$$\ddot{\theta}_i(t) = \frac{\theta_i(t + \Delta t) - 2\theta_i(t) + \theta_i(t - \Delta t)}{\Delta t^2} \quad (1.36)$$

Tenint aquestes dades, podem obtenir les acceleracions i les velocitats de rotació de les masses. Per altre banda, si realment el que interessa és la direcció de la velocitat i l'acceleració que puguem obtenir en una massa, a partir del vector posició de la massa es pot obtenir de la següent manera:

$$\vec{r}_i = T(i) = T(\theta_1) * \dots * T(\theta_i) \quad (1.37)$$

$$\vec{v}_i = \frac{d}{dt}T(i) = \sum_{j=1}^i [T(\theta_1) \dots \frac{dT(\theta_j)}{d\theta_j} \dots T(\theta_i)]\dot{\theta}_j \quad (1.38)$$

$$\begin{aligned} \vec{a}_i = \frac{d}{dt}\left(\frac{d}{dt}T(i)\right) = \sum_{j_1=1, j_2=1}^i [T(\theta_1) \dots \frac{dT(\theta_{j_1})}{d\theta_{j_1}} \dots \frac{dT(\theta_{j_2})}{d\theta_{j_2}} \dots T(\theta_i)]\dot{\theta}_{j_1}\dot{\theta}_{j_2} + \\ \sum_{j=1}^i [T(\theta_1) \dots \frac{dT(\theta_j)}{d\theta_j} \dots T(\theta_i)]\ddot{\theta}_j \end{aligned} \quad (1.39)$$

Una vegada tenim calculades totes les posicions, velocitats i acceleracions de cada una de les masses, podem calcular el moment angular i el lineal a partir de les fórmules 1.13 i 1.32 respectivament. I finalment, podem calcular el moment generalitzat de cada grau de llibertat, com també l'esforç de cada un d'ells de la manera següent:

$$P_i(t) = \frac{\partial K}{\partial \dot{\theta}_i} = \text{moment generalitzat de l'articulació } i \quad (1.40)$$

$$F_j = \frac{dP_j(t)}{dt} - \frac{\partial K}{\partial \theta_j} - \sum F_i^{ext}(i) \frac{\partial T(i)}{\partial \theta_j} = \text{esforç de l'articulació } j \quad (1.41)$$

essent, K igual a l'energia cinètica total del model, que es calcula com:

$$K(t) = \frac{1}{2} \sum_{i=1}^{N_{masses}} m_i \vec{v}_i(t) \cdot \vec{v}_i(t) \quad (1.42)$$

o també, a partir de les transformacions de la taula DH per cada una de les cadenes del model:

$$K(t) = \frac{1}{2} \sum_{i=1, j_1=1, j_2=1}^{N_{masses}} m_i \left[\frac{dT(i)}{d\theta_{j_1}} \right]_{(1..3,4)}(t) \cdot \left[\frac{dT(i)}{d\theta_{j_2}} \right]_{(1..3,4)}(t) \dot{\theta}_{j_1}(t) \dot{\theta}_{j_2}(t) \quad (1.43)$$

tenint en compte com es calcula K , podem deduir el moment d'inèrcia i el moment generalitzat:

$$\frac{dK}{d\theta_i}(t) = \sum_{j=1, j_1=1, j_2=1}^{N_{masses}} m_j \left[\frac{dT(j)}{d\theta_i d\theta_{j_1}} \right]_{(1..3,4)}(t) \cdot \left[\frac{dT(j)}{d\theta_{j_2}} \right]_{(1..3,4)}(t) \dot{\theta}_{j_1}(t) \dot{\theta}_{j_2}(t) \quad (1.44)$$

$$\vec{P}_i(t) = \frac{dK}{d\dot{\theta}_i}(t) = \sum_{j=1, j_1=1}^{N_{masses}} m_j \left[\frac{dT(j)}{d\theta_i} \right]_{(1..3,4)}(t) \cdot \left[\frac{dT(j)}{d\theta_{j_1}} \right]_{(1..3,4)}(t) \dot{\theta}_{j_1}(t) \quad (1.45)$$

Una vegada tenim aquestes equacions definides, el que cal fer és calcular per cada instant de temps el moment generalitzat per cada un dels graus de llibertat, d'aquesta manera tindrem el $P_i(t)$ definit per cada un dels graus de llibertat i podrem calcular F_i . Per calcular el primer terme de la equació 1.41, es pot fer de la següent manera coneixen el $P_i(t)$:

$$\frac{dP(t)}{dt} = \frac{P_i(t + \Delta t) - P_i(t - \Delta t)}{2 \Delta t} \quad (1.46)$$

Una vegada tenim calculat aquest terme, i sabent com es calcula el segona, obtenim F_i . I finalment per acabar, també podem calcular dades interessants com la posició, velocitat i acceleració del Centre de Massa a partir de cada un dels estats calculats.

Una vegada arribat a aquest punt, queda vista tota la física que cal tenir en compte per realitzar aquest projecte. En el següent capítol, s'analitzarà tota la informació que l'usuari necessita entrar, com derivar dades a partir d'aquesta informació i quina estructura ha de tenir el programa de manera que els càlculs que cal fer siguin el màxim de consistents possibles.

Capítol 2

Anàlisi de l'aplicació

En aquest capítol s'analitzaran els requisits i la informació necessària per crear aquesta aplicació per tal de buscar les necessitats més específiques per cobrir les funcionalitats nombrades en el capítol anterior. Es determinaran els actors que intervenen en l'ús de l'aplicació i s'especificaran els cassos concrets per portar a terme les funcionalitats principals i es donarà una petita idea de com ha de ser el conjunt de classes o objectes que intervenen en el programa.

2.1 Actors

En aquest cas, només tenim un actor i és l'usuari, definit com aquella persona que fa ús del programa.

2.2 Funcionalitat 1: Generació de models

2.2.1 Necessitat

Com s'ha comentat anteriorment, la necessitat de crear models tridimensionals que simbolitzin un cos articulat a la realitat és un punt important en aquest projecte. Així doncs, necessitem un conjunt d'operacions o situacions d'interacció usuari-sistema per portar a terme la creació de models acurats i propers a la realitat. L'objectiu de tots i cada un dels cassos que es descriuran a continuació tenen aquesta finalitat, que no les exclou d'altres, però el conjunt de tots ells ha de permetre crear models reals.

2.2.2 Cassos d'ús

Cas d'ús 1.1: Crear model

Explicació En el moment en que l'usuari desitja crear un nou model articulat cal indicar-ho al programa i aquest ha de donar la capacitat a l'usuari a triar la posició del centre del model. Una vegada fet això, el sistema ha de guardar la informació i preparar-la per altres accions que permetin seguir amb la generació del model.

Cas d'ús 1.2: Afegir Conjunt de cadenes

Explicació El Conjunt de cadenes és un punt en el que un conjunt de cadenes surten d'ell per poder crear models més complexos i propers a la realitat. És un punt d'ancoratge del qual en surten diferents cadenes.

Cas d'ús 1.3: Afegir cadena al model

Explicació Cal indicar-li al sistema, una vegada creat el model, que es desitja incorporar un seguit de "links" i masses', que aniran lligades unes amb les altres, al punt inicial de construcció del model o un conjunt de cadenes.

Cas d'ús 1.4: Afegir un "Link" al model

Explicació En el moment que tenim un model creat amb alguna cadena, cal afegir la resta d'elements que es va descriure en el capítol 1 sobre el model en aquest cas l'element "Link", caldrà que l'usuari indiqui que vol afegir un "link" al model i el sistema li ha de permetre col·locar-lo en l'espai 3D a voluntat. Cal mantenir la concordança amb la taula D-H que descriu el model articulat.

Cas d'ús 1.5: Afegir una massa al model

Explicació És el mateix cas que l'anterior, però en aquest cas caldrà afegir una massa, vinculada a una articulació o només per poder obtenir informació d'aquell punt concret del model, el procediment d'incloure un o altre, respecte del cas d'ús anterior, es farà de manera diferent, però en aquest punt no és moment de pensar com fer-ho, ja que el que interessa és que s'incorpori al

model i prou. De la mateixa manera que en el cas d'ús anterior, cal mantenir la mateixa informació, que es veu en la visualització tridimensional, que en la taula D-H.

Cas d'ús 1.6: Afegir un recolzador al model

Explicació Cal remarcar que el sistema necessita conèixer quines parts del model són hàbils per poder tenir contacte amb algun objecte. Si permetéssim que el model pogués recolzar-se amb qualsevol part del cos, el problema seria altament complex. Per això la incorporació d'elements anomenats recolzadors que són aquelles parts més naturals per on un cos es recolzaria, per exemple el ser humà tindria els peus o les mans com a recolzadors.

2.3 Funcionalitat 2: Modificació de models

2.3.1 Necessitat

Una vegada creat un model la necessitat de modificar-lo és quasi directa, un error en el moment de generació del model o la intenció de crear un nou model a partir d'un ja guardat, són excusa suficient com per donar la capacitat al sistema de poder modificar models una vegada creats.

2.3.2 Cassos d'ús

Cas d'ús 2.1: Eliminar model

Explicació En qualsevol punt s'ha de poder eliminar el model si l'usuari ho desitja de manera que si ha de rectificar tot el disseny del model no hagi d'anar eliminant element per element per tornar a començar de zero. Així doncs l'usuari ha de poder indicar el model que vol eliminar i confirmar-ho al sistema.

Cas d'ús 2.2: Moure model

Explicació Hi ha varies raons per la qual el sistema necessita garantir que es pugui moure un model. Una primera podria ser per comoditat de

l'usuari a que el model es col·loqui apartat del centre de l'escena per poder dissenyar-la o una altre per poder donar posicions diferents al model ja que el moviment que es vol calcular ho requereix. Així doncs, el sistema ha de permetre a l'usuari poder seleccionar un model qualsevol que ha creat i deixar-li modificar la posició bàsicament de dues maneres, una que seria a partir de l'element gràfic, que mostra el model tridimensional, de la interfície gràfica del sistema o a través de formulari indicant les coordenades X, Y i Z que es desitja que tingui el model.

Cas d'ús 2.3: Rotar model

Explicació De la mateixa manera que el model es pot moure, cal també que es pugui rotar per poder donar diferents orientacions al model degut a que el moviment que es vol calcular ho requereix. Així doncs, el procediment serà de la mateixa manera que en el cas d'ús anterior, primer l'usuari indicarà al sistema quin model es vol rotar, i després podrà indicar l'angle o els angles per cada un dels eixos fixes de rotació a partir de dos mecanismes, un primer a través de l'element gràfic de la interfície gràfica del programa, i l'altre a través de formulari.

Cas d'ús 2.4: Eliminar cadena

Explicació Una cadena s'ha descrit com un conjunt d'elements Masses i "Links" que surten del centre del model o d'un conjunt de cadenes, així doncs, si l'usuari desitja eliminar una cadena sencera i no element per element, el sistema li permetrà fer-ho. El mètode de procediment, serà que l'usuari indicarà quina cadena desitja eliminar, i després li ho confirmarà, el sistema ha de garantir la concordança amb el nou model que desitja l'usuari.

Cas d'ús 2.5: Moure cadena

Explicació Moure una cadena implica que el punt d'unió del centre del model amb aquesta cadena es veu afectat. Dit això, el procediment és el mateix, l'usuari indica quina cadena desitja moure de posició, l'usuari amb el ratolí o mitjançant formulari indicarà la posició exacte mitjançant els angles de rotació que portin del centre del model o d'un conjunt de cadenes al sistema de referència inicial de la cadena. Oportunament, el sistema també li permetrà canviar la cadena a un altre centre de cadena. Una vegada que l'usuari acabi el sistema ha de mantenir la concordança modificant els

paràmetres corresponents que descriuen el model articulat que s'està dissenyant.

Cas d'ús 2.6: Rotar cadena

Explicació En aquest cas el que es fa es rotar el primer element de la cadena a partir del punt que està unit amb el centre de massa i per tant, la resta d'elements també es veuen afectats a causa d'aquesta transformació. Per tant, de la mateixa manera que s'ha descrit per moure una cadena, en aquí es podrà fer a partir de dues maneres, la primera de la mateixa manera que en el cas anterior, que és a partir de l'element gràfic de la interfície i el ratolí, i una altre que indicarà l'angle de rotació per cada un dels eixos.

Cas d'ús 2.7: Eliminar "Conjunt de cadenes"

Explicació La necessitat d'eliminar un conjunt de cadenes bé donada degut a que si un model mal dissenyat conté un conjunt de cadenes al qual no es té la intenció d'afegir cap més cadena i només en té una de la qual surt d'ella, al eliminar-la, podem unir les dues cadenes que estaven unides per aquest conjunt de cadenes. En el cas de que tingués més d'una cadena que surt del conjunt de cadenes eliminat, s'eliminarien totes les cadenes que sortissin d'ella. Així doncs, l'usuari podrà seleccionar el centre del model que vulgui i dir-li al sistema que es vol eliminar. Automàticament el sistema tractarà la informació que se li ha encarregat, l'actualitzarà segons convingui.

Cas d'ús 2.8: Eliminar "Link"

Explicació De la mateixa manera que es pot eliminar una cadena o un model directament, cal també poder eliminar elements que no siguin el centre de massa del model, ja que si s'elimina aquest, és necessari eliminar tot el model. Així doncs, en el moment de eliminar un "Link", el que tingui a banda i banda passa a estar unit i per tant, el sistema caldrà que ho enregistri i interpreti el que està passant. Així doncs, el sistema de funcionament de cas serà el següent, l'usuari indicarà al sistema quin element desitja eliminar i ho confirmarà, seguidament, el sistema enregistrarà i mantindrà la concordança i la consistència que convingui.

Cas d'ús 2.9: Rotar "Link"

Explicació Un dels mecanismes per poder modificar l'estructura d'un model articulat és directament modificar la unió entre elements, així doncs, en el moment de rotar un "Link", el que es fa és rotar a partir del punt de connexió amb l'element que està més a prop del centre de massa, sense modificar les seves característiques que el determinen (la llargada). El mecanisme que descriu el funcionament d'aquest cas és el següent, primer l'usuari selecciona el "Link" que vol rotar, i a partir del ratolí o d'un formulari indica al sistema quins són els valors dels angles, per cada un dels eixos, que l'usuari desitja que s'apliquin a l'element.

Cas d'ús 2.10: Modificar la llargada del "Link"

Explicació Una altre manera per modificar l'estructura és modificar els elements rígids que aguanten el sistema articulat, és a dir els "links". En aquest cas només tenen un paràmetre rígid que els descriu i és la llargada. Aleshores, el mecanisme és el següent, l'usuari indica el sistema quin "Link" li desitja canviar la llargada i el sistema a través de formulari li permetrà de modificar. Una vegada introduïda la dada, el sistema ha de guardar-la i fer-la notar en el model articulat, ja sigui en el model tridimensional i la taula D-H que descriu el sistema articulat.

Cas d'ús 2.11: Eliminar Massa

Explicació De la mateixa manera que es pot eliminar un "Link", una Massa també i el mecanisme és el mateix que en el cas d'ús 2.7, en el moment d'eliminar els dos elements connectats a aquesta Massa' passen a estar units i el sistema reconstrueix la informació per tal de tenir el model descrit de manera coherent. El mecanisme de funcionament és el mateix que el cas d'ús 2.7 però enlloc d'un "Link" és una Massa.

Cas d'ús 2.12: Rotar Massa

Explicació Exactament igual que en el cas d'ús 2.8. Però la única diferència és que el punt de contacte amb l'element anterior a la "Joint" es troba just al centre d'aquest, per tant, la rotació es realitzarà en el mateix centre de la Massa. A nivell d'aplicació la seqüència d'accions és la mateixa descrita que en el cas d'ús 2.8.

Cas d'ús 2.13: Rotar Recolzador

Explicació Aquest cas es comporta exactament que l'anterior, l'orientació bé descrita des de la base del recolzador i és aquesta la que es modifica, exactament de la mateixa manera que en el cas anterior.

Cas d'ús 2.14: Eliminar Recolzador

Explicació Cal remarcar, que en aquest cas, és més senzill de realitzar, ja que un Recolzador mai tindrà elements que pengin d'ell i per tant, eliminant un Recolzador no caldrà comprovar si hi ha res després i unir-ho al anterior, ja que no ni haurà cap. El procediment d'actuació entre usuari i sistema serà el mateix que en el cas d'ús 2.8 i 2.11.

Cas d'ús 2.15: Modificar la llargada d'un Recolzador

Explicació Un Recolzador té una superfície per la qual està en contacte amb alguna altre superfície i l'usuari ha de poder editar aquesta zona i poder fer-la més gran o més petita per tal de donar més o menys estabilitat al model. Aleshores, el que hauria de fer l'usuari és indicar quin Recolzador vol editar i donar-li la dimensió de la llargada del recolzador.

Cas d'ús 2.16: Modificar l'amplada d'un Recolzador

Explicació És el mateix cas que l'anterior, però l'única diferència és que en comptes de modificar la llargada es modificarà l'amplada del Recolzador indicat.

2.4 Funcionalitat 3: Generació d'escenes

2.4.1 Necessitat

Per tal de cobrir la necessitat de que el model articulats tingui punts de recolzament rígids amb els quals interactuar, cal crear una escena amb objectes que l'usuari pugui modificar i col·locar a consciència. Així doncs,

necessitarem mètodes que permetin la creació d'objectes a priori senzills, figures regulars, per exemple, que l'usuari només hagi de col·locar i modificar l'orientació, la mida...

2.4.2 Cassos d'ús

Cas d'ús 3.1: Afegir un objecte

Explicació Per generar una escena l'únic necessari és anar afegint objectes, d'aquesta manera acabes tinguen un escenari amb un conjunt d'objectes. Ara és cert que no només amb això acabes tinguen l'escenari que un desitjaria, però amb això és lo mínim. Cal remarcar també que el concepte d'objecte és deixa molt obert i pot ser qualsevol objecte, per això el sistema haurà de proporcionar un conjunt d'objectes que l'usuari pugui afegir. El mètode per afegir un objecte a una escena és el següent, l'usuari indica quin objecte vol incloure i depenent de quin objecte vulgui incloure (esfera, cub, pla...) tindrà un mecanisme o un altre, en el sentit de que pot fer falta un o dos clics de ratolí, arrearar o no... Tot hi així, al final de que l'usuari hagi indicat la forma de l'objecte, el sistema haurà de incloure l'objecte a l'escena.

2.5 Funcionalitat 4: Modificació d'escenes

2.5.1 Necessitat

Per tal de modificar una escena, n'hi ha prou en modificar els objectes que la componen, en aquesta aplicació, en principi el modelat de les escenes és molt senzill, ja que, l'objectiu principal no consisteix en crear una escena lo més real possible, sinó que cal crear un moviment físicament realista, i per tant, les principals formes de modificar un objecte, seran rotar, escalar o modificar la seva posició.

2.5.2 Cassos d'ús

Cas d'ús 4.1: Rotar objecte

Explicació Per poder modificar l'orientació d'un objecte el sistema ha de permetre poder rotar un objecte qualsevol de l'escena. El mecanisme de

procediment a seguir és el següent, primer l'usuari indica que vol seleccionar un objecte i indica quin, a continuació el sistema li ha de permetre a l'usuari introduir els angles de rotació per cada un dels eixos. Aquest pas pot ser a partir de l'element gràfic de la interfície gràfica o a través de formulari. Una vegada introduïdes les dades, el sistema ho ha d'enregistrar.

Cas d'ús 4.2: Escalar objecte

Explicació De la mateixa manera que es pot canviar l'orientació també el sistema ha de poder canviar la mida de qualsevol objecte. El procediment serà el mateix que en el cas d'ús anterior, l'únic que enlloc dels angles de rotació per cada eix, el que es dona és el valor pel qual s'ha de escalar per cada un dels eixos.

Cas d'ús 4.3: Moure objecte

Explicació Una altre manera de modificar una escena seria poder moure els objectes que conté de les seves posicions originals. Per tant, el sistema permetrà d'una manera senzilla canviar la posició de l'objecte, sempre considerant que la posició de l'objecte es troba en el centre de la caixa que engloba tots els punts de l'objecte. Així doncs, el mecanisme de funcionament d'aquest cas d'ús serà el mateix que els dos anteriors, l'únic que per cada eix el que es donarà serà el nou valor de la coordenada.

Cas d'ús 4.4: Duplicar objecte

Explicació Una mecanisme que agilitza el modelatge d'escenes, és el duplicat d'objectes. En moltes ocasions, tenim objectes aparentment iguals i que per tant es poden treure d'un objecte que ja existeix a l'escena i duplicar-lo fa més ràpid el procés de modelatge d'una escena i després modificant la posició o escalat o orientació ja sembla tenir un objecte diferent a l'original. El mecanisme de funcionament varia una mica respecte els altres 3 cassos d'ús anteriors, cal primer indicar quin és l'objecte que es vol duplicar, després duplicar-lo i el sistema just després d'això reclamarà una nova posició per aquest objecte, ja que podria quedar sol-lapat amb l'original i semblar que no hagués fet el duplicat. El sistema una vegada més haurà d'enregistrar els canvis.

Cas d'ús 4.5: Eliminar objecte

Explicació Per poder editar una escena, cal també poder eliminar objectes d'aquesta, un usuari es pot equivocar o no li agrada com ha quedat una escena... Per tant, el sistema ha de poder garantir a l'usuari que pot eliminar objectes que ell ha afegit i/o modificat. El mecanisme serà que l'usuari indiqui al sistema quin objecte es vol eliminar seleccionant-lo, li indiqui que el vol eliminar i el sistema ha d'eliminar l'objecte de l'aplicació.

Cas d'ús 4.6: Modificar paràmetre objecte

Explicació Tot objecte, té uns paràmetres concrets que el defineixen, com per exemple, en un cub, doncs la llargada dels costats és el paràmetre que descriu l'objecte, o el radi d'una esfera. L'usuari ha de poder donar valor a aquests paràmetres a voluntat fent d'aquest mecanisme una manera més de modificar una escena i tenir una varietat més ampla de possibilitats.

2.6 Funcionalitat 5: Generació d'animacions

2.6.1 Necessitat

Pel que fa a la generació d'animacions, el que interessa és que el sistema permeti entrar punts claus amb els seus temps i els seus punts de contacte o restriccions. Per tant, els cassos d'ús que estiguin dins d'aquesta funcionalitat, consistiran en cobrir aquestes necessitats, ni més ni menys.

2.6.2 Cassos d'ús

Cas d'ús 5.1: Afegir punt clau

Explicació Un moviment sempre té uns moments o instants de temps que marquen el moviment i en aquests punts hi ha un canvi de trajectòria del model que està realitzant aquest moviment. Per tant, cal donar a l'usuari la capacitat de donar aquests punts claus del moviment. En aquests punts claus, és on es donen els punts de recolzament sobre el terreny, o punts de contacte.

Cas d'ús 5.2: Modificar paràmetre del model articulat

Explicació A part de poder moure el model articulat, com també canviar-li la orientació, cal poder canviar els paràmetres que descriuen els graus de llibertat del model, és a dir, les articulacions, o masses associades a una articulació (Masses Mòbils), l'angle que formen els elements que uneix. Això és així, perquè en molts casos interessarà que el moviment tingui no només uns punts de contacte, sinó una aproximació a un estat del model articulat concret que no sigui l'estat de relaxació, on aquests paràmetres estant a 0. Aleshores, caldran uns mecanismes que ajudin a l'usuari poder modificar aquests graus de llibertat. Aquest cas d'ús tracta només un grau de llibertat, però és vàlid per tots, el mecanisme consisteix en que l'usuari selecciona l'element que se li relaciona el grau de llibertat, en aquest cas les masses articulades, aleshores, el sistema li permetrà modificar el grau de llibertat a partir de dos mètodes, el primer a través de l'element gràfic de la interfície gràfica i a través de formulari d'aquesta mateixa interfície. Finalment, el sistema a de garantir que la informació recollida ha estat enregistrada i es veu reflectida a l'element gràfic.

Cas d'ús 5.3: Afegir punt de contacte

Explicació Una de les parts principals d'aquesta aplicació, són els punts de contacte, són moments en el transcurs del moviment en que el model articulat interactua amb l'escena, i per tant, es generen forces. Cal doncs que l'aplicació doni a l'usuari la capacitat de col·locar punts de contacte o de recolzament del model sobre l'escena de manera senzilla, més aviat que l'ajudi. Un punt de contacte és una connexió que hi ha entre el model i l'escena i això és un conjunt reduït de l'espai si tenim en compte tot l'espai real que existeix en l'escena, per tant, un punt fora del conjunt reduït no interessa. Dit això, el mecanisme consistirà en que l'usuari pugui agafar qualsevol Recolzador i col·locar-lo en qualsevol punt d'algun objecte de l'escena. El sistema l'ajudarà en el sentit de que automàticament el sistema re-col·locarà el model tenint en compte el moviment del ratolí sempre intuït cap on vol anar l'usuari.

Cas d'ús 5.4: Assignar instant de temps

Explicació En un moment donat, quan tinguem els punts de contacte entrats per un moment donat del moviment, caldrà donar-li un temps dins de

la seqüència del moviment, donar-li un ordre dins l'espai de temps. Per tant, una vegada l'usuari cregui convenient, li indicarà al sistema que li vol introduir un instant de temps, i aquest li ha de permetre a l'usuari quin instant de temps és. Una vegada finalitzada la interacció, el sistema ha de enregistrar els canvis i que es vegin reflectits en la interfície gràfica de l'aplicació.

Cas d'ús 5.5: Generar animació

Explicació Després de tot el procés de crear les diferents posicions intermitges que el model té al llarg del moviment l'usuari ha de poder generar una animació amb aquestes posicions. En el moment en que ho desitgi l'usuari li dirà al sistema que ja pot fer el càlcul del moviment, en aquest moment, el sistema realitzarà un conjunt de càlculs que donarà els paràmetres del model per cada un dels fotogrames que tindrà l'animació, fent que sembli una pel·lícula. En aquest moment, el sistema ha de permetre a l'usuari poder modificar tot el que vulgui l'usuari per poder acabar de fer un moviment encara més realista i re-calcular els canvis de manera realista. Així doncs el procediment queda descrit d'aquesta manera, primer l'usuari indica al sistema que ja pot iniciar els càlculs i després aquest últim ha de permetre a l'usuari visualitzar tot el moviment complet i poder-lo editar a voluntat. Cal que en cada moment guardi la informació recollida i calculada per poder consultar-la i/o modificar-la quan es necessiti.

2.7 Funcionalitat 6: Modificació d'animacions

2.7.1 Necessitat

Si és cert que per generar animacions amb els cassos d'ús anteriors en tindríem prou, però caldria que l'usuari dissenyés el moviment a la primera i sense marge de error, per això cal poder modificar animacions per aconseguir el resultat esperat per l'usuari.

2.7.2 Cassos d'ús

Cas d'ús 6.1: Eliminar punt clau

Explicació En qualsevol moment l'usuari ha de poder eliminar un punt clau del moviment. Per fer-ho, el sistema ha de permetre d'una manera còmode seleccionar un punt clau i eliminar-lo. L'usuari seleccionarà el punt clau i indicarà que vol eliminar-lo, una vegada finalitzat, el sistema ha de enregistrar els canvis i si s'escau re-calcular el moviment.

Cas d'ús 6.2: Modificar instant de temps

Explicació Un punt clau té assignat un temps, així doncs, un usuari ha de poder editar-lo. Així doncs, l'usuari seleccionarà el punt clau al qual es vol modificar el temps, seguidament el sistema li permetrà modificar el temps, l'usuari introduirà la dada i seguidament el sistema enregistrarà els canvis i si s'escau re-calcular el moviment.

Cas d'ús 6.3: Eliminar punt de contacte

Explicació Cal poder eliminar punts de contacte ja que sinó no es permetria errors de l'usuari en el moment de dissenyar el moviment que ha de fer el model articulat. Així doncs a partir de l'element gràfic de la interfície gràfica o a través de formulari, l'usuari seleccionarà el punt de contacte i li indicarà al sistema que es desitja eliminar, al acabar, el sistema enregistra els canvis.

Cas d'ús 6.4: Modificar punt de contacte

Explicació De la mateixa manera que s'ha de poder eliminar un punt de contacte, també s'ha de poder modificar, ja que sinó obligaríem a l'usuari a col·locar un punt de contacte a la primera. Aleshores, l'usuari seleccionarà el punt de contacte mitjançant l'element gràfic de la interfície gràfica o a partir del la interfície gràfica mitjançant alguna llista i a partir de l'element gràfic, movent el ratolí, es modificarà la posició del punt de contacte. El sistema enregistrarà els canvis una vegada finalitzat el cas d'ús.

Cas d'ús 6.5: Modificar fotograma de l'animació

Explicació Una vegada generada l'animació tindrem en un interval petit de temps un fotograma al qual caldrà que el modifiquem, en aquest cas, el sistema ha de permetre realitzar una modificació com si es tractés d'un punt clau del moviment. Per tant, el sistema garantirà els cassos d'ús 5.1, 5.2, 6.1, 6.2, 6.3 i 6.4. Una vegada un d'aquest cassos d'ús hagi passat, el sistema ha d'enregistrar els canvis que hagin passat.

Cas d'ús 6.6: Modificar paràmetre d'una etapa

Explicació Una etapa, com ja hem vist conté un conjunt de paràmetres editables de manera que l'usuari pugui modificar ja sigui la trajectòria del Centre de Massa o la posició del punt d'aplicació de la força de contacte. Com hem vist, els paràmetres modificables, són en una etapa en la que no hi hagi punts de contacte, purament l'usuari només podrà modificar el Centre de Massa per cada un dels Punts Clau que estigui format. Mentre que en una etapa que contingui un conjunt de punts de contacte no buit, l'usuari hem vist que podia modificar el Centre de Massa de cada un dels Punts Clau com també la velocitat inicial i final de l'etapa sempre hi quan no estigui precedida o seguida per una etapa de vol i també els termes de durada de la primera fase de l'etapa i de la segona, el terme de distància mínima, de velocitat en el punt de distància mínima i els angles Th1 i Th2 vistos també seran editables. Finalment, l'usuari podrà dir al sistema si vol que el punt d'aplicació de la força de contacte estigui calculat per el sistema o ell dóna el valor d'aquest punt. Cal remarcar, que una etapa estarà composta per 3 Punts Clau i que tindran un subconjunt comú de punts de contacte que es manté fixa durant tota l'etapa. Si no es compleix això últim, el sistema considerarà no vàlida l'etapa i no deixarà modificar a l'usuari cap d'aquests paràmetres. Una vegada modificats, el sistema haurà de fer els canvis oportuns de manera que es faci consistent i coherents els canvis.

Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau

Explicació En un punt clau, queda definida tota la configuració del model i a més a més informació sobre els punts de contacte. L'usuari ha de ser capaç de modificar-ho per aconseguir diferents comportaments en el moviment que es vulgui determinar. Poder modificar un Punt Clau implica tenir la capacitat de modificar la posició i orientació del model, com també els graus d'obertura

de cada una de les articulacions que formen el model, i a més a més l'usuari ha de poder modificar els punts de contacte. En aquest últim cas, es farà servir el cas d'ús ja definit per modificar el punt de contacte.

2.8 Funcionalitat 7: Gestió de la visió de l'espai tridimensional

2.8.1 Necessitat

Per una millor edició a l'usuari, el sistema permetrà donar diferents punts de vista a l'usuari per poder treballar en l'animació. Aquests punts de vista han de usar uns mecanismes intuïtius i que l'usuari pugui comprendre de manera ràpida. Alguns d'aquest mecanismes són el Pan, el Zoom o la Rotació de tota l'escena. El Pan, dóna la sensació de que mou tota l'escena en bloc, el Zoom aproxima o allunya l'escena al punt de vista de l'observador, i la Rotació, com indica el nom rotar tota l'escena en el sentit que vulgui l'usuari. En principi amb aquests tres mecanismes, l'usuari ha de poder explorar tot l'espai tridimensional que es veu reflectit en l'element gràfic incorporat a la interfície gràfica.

2.8.2 Cassos d'ús

Cas d'ús 7.1: Pan

Explicació El Pan és un mecanisme que ajuda a moure tota l'escena respecte el punt de vista de l'observador. La manera de procedir en aquest cas, consistirà que a través de l'element gràfic de la interfície gràfica, l'usuari indicarà que vol realitzar un Pan i el sistema li permetrà primer prémer i després arrearar amb el ratolí tota l'escena visualitzada en 3D. Una vegada es deixa de prémer el botó el cas d'ús finalitza i el sistema manté la imatge que l'usuari ha deixat just abans de deixar de prémer el botó del ratolí.

Cas d'ús 7.2: Zoom

Explicació El Zoom és molt útil quan es vol veure amb més detall els objectes tridimensionals o veure tota l'escena, acostar o allunya l'escena a voluntat de l'usuari respecte el punt de vista de l'observador. Així doncs, la

manera hem com es durà a terme serà molt semblant a l'anterior, l'usuari indicarà al sistema la necessitat de fer un Zoom a l'escena actual i aquest li permetrà prémer el ratolí i després arrearar i veure com allunya o acosta l'escena a voluntat de l'usuari en el moment en que deixa de prémer el cas d'ús finalitza i el sistema guarda com ha quedat el punt de visió de l'element gràfic.

Cas d'ús 7.3: Rotació

Explicació La rotació, com indica el nom rota la imatge en, aquest cas, uns eixos determinats, ja que tenim el senyal del ratolí que ens indica quan s'ha desplaçat amunt o avall i quan s'ha desplaçat a la dreta o a l'esquerra, i per tant, tenim dos valors d'entrada que no es pot extrapolar a tres angles de tres eixos. Per tant, el que es fa és modificar l'angle de rotació de dos eixos i amb això es permetrà que amb la rotació es pugui veure tota l'escena, i cada un dels seus racons. Així doncs, el sistema ha de poder determinar que l'usuari vol fer una rotació de l'escena i aquest donar després clicant i desplaçant el ratolí per l'escena de manera que pugui anar rotant la imatge. Una vegada l'usuari deixa de prémer el botó del ratolí el sistema a de mantenir el punt de visió que ha quedat just abans de aixecar el dit del ratolí.

2.9 Funcionalitat 8: Gestió de la informació

2.9.1 Necessitat

Com s'ha parlat en el primer capítol, la necessitat de poder guardar i tornar a recuperar el treball fet és importantíssim en aquesta aplicació, ja que, essent un projecte d'àmbit experimental en el que donarà informació concreta respecte de un moviment d'un model articulat concret, com la força, l'angle d'obertura de les articulacions en cada instant, la velocitat de diferents parts del model... cal poder guardar la informació i no perdre tot el treball fet i haver-lo de tornar a fer cada vegada que es vulgui repetir l'experiment.

2.9.2 Cassos d'ús

Cas d'ús 8.1: Guardar model

Explicació Una de les parts importants d'un projecte fet amb aquesta aplicació és el model articulat, en el moment en que l'usuari ho desitgi, el sistema li permetrà realitzar. Primer de tot, l'usuari indicarà que vol guardar el model i aleshores, el sistema li proporcionarà la manera de introduir el nom de l'arxiu que contindrà aquest model i la carpeta a on ho vulgui guardar. Una vegada l'usuari confirmi aquesta informació el sistema crearà l'arxiu i escriurà la informació.

Cas d'ús 8.2: Carregar model

Explicació En qualsevol moment, l'usuari ha de poder carregar un model que ha guardat anteriorment i afegir-lo al projecte de treball, de manera que no hagi de tornar aquell mateix model. Així doncs, en aquest cas, l'usuari indicarà que vol carregar un model i el sistema li permetrà seleccionar un arxiu de tipus model que entengui mitjançant un navegador de carpetes i en el moment en que l'usuari seleccioni un arxiu, el sistema el carregarà a l'aplicació i la farà visible en l'element gràfic de la interfície gràfica.

Cas d'ús 8.3: Guardar escena

Explicació De la mateixa manera que es pot guardar un model, també s'ha de poder guardar l'escena, per poder-la usar en un altre moment i no haver-la de tornar a dissenyar de zero. El mecanisme en aquest cas és el mateix que amb el model, el sistema deixa a l'usuari introduir el nom i la direcció on vol guardar-la i el sistema ho enregistra generant un nou fitxer amb el contingut de l'escena.

Cas d'ús 8.4: Carregar escena

Explicació Per tal de no generar una escena des de zero cada vegada, el sistema haurà de permetre a l'usuari carregar escenes que ell mateix ha creat. El procediment és el mateix que amb els models, l'usuari ha de poder seleccionar un arxiu que representi una escena i el sistema el carregarà automàticament visualitzant-lo a la interfície gràfica de la mateixa manera que en l'últim moment que havia guardat l'escena.

Cas d'ús 8.5: Crear projecte

Explicació Cal unificar tota la informació que l'usuari va entrant en un únic element de manera que no hagi de carregar un model, després una escena i entrar tota la informació corresponen amb els punts de contacte que puguin aparèixer entre aquests dos. Així doncs, cal fer que l'usuari pugui crear projectes, que sigui un element que manegi tota la informació per evitar que l'usuari realitzi feines repetitives cada vegada que vulgui crear una animació concreta. Així doncs, l'usuari podrà crear un projecte i afegir-li models i escenes de manera que pugui després entrar informació dels punts de contacte. En el moment en que això passi, el sistema ha d'haver creat un projecte i fer-lo consistent.

Cas d'ús 8.6: Guardar projecte

Explicació No només n'hi ha prou en crear un projecte, sinó que també cal guardar tota la informació que es va entrant en cada moment. Així doncs, l'usuari quan indiqui el sistema que vol guardar un projecte, el sistema tindrà dos cassos d'actuació, un primer que si un projecte no té un fitxer de bolcat d'informació associat, és a dir, un fitxer, caldrà que el sistema doni a l'usuari la capacitat de poder donar-li el nom i la direcció on vulgui guardar, després el sistema enregistrarà els canvis en el fitxer. I si ja existeix aquest fitxer, el sistema directament enregistra els canvis.

Cas d'ús 8.7: Carregar projecte

Explicació En el moment en que l'usuari engega l'aplicació i vol tornar a seguir manipulant un projecte que havia guardat abans, el sistema ha de permetre a l'usuari carregar aquest projecte. Així doncs, l'usuari a través de la interfície podrà buscar el fitxer que conté el projecte que vol seguir manipulant i l'hi indicarà el sistema que vol carregar-lo. Automàticament, el sistema carregarà la informació del projecte i la farà visible de cara a l'usuari.

Cas d'ús 8.8: Afegir gràfica

Explicació Un dels punts importants d'aquest programa ha de ser la capacitat de poder veure informació sobre l'evolució del model en certs punts, com la posició del model, la orientació, la velocitat del model o de certes parts

del cos... Per tant, el sistema permetrà a l'usuari associar una gràfica a un paràmetre d'algun element del model i en el moment en que l'usuari generi el moviment ha d'actualitzar les dades de la gràfica perquè més tard l'usuari les pugui visualitzar. Així doncs, en el moment en que l'usuari vulgui afegir una gràfica, aquest li indicarà els paràmetres, i el sistema crearà una relació entre aquesta dada i la gràfica i la farà constar de cara a l'usuari afegint una gràfica al projecte.

Cas d'ús 8.9: Eliminar gràfica

Explicació Cal poder eliminar una gràfica en el moment en que ja no interressi consultar les dades que contingui o per error i no es vol tenir tanta informació. Per tant, l'usuari indicarà al sistema que vol eliminar una gràfica i el sistema enregistrarà els canvis i ho farà consistent amb la informació que ja es tingui.

Cas d'ús 8.10: Visualitzar gràfica

Explicació Quin sentit té tenir una gràfica creada i no poder consultar-la, d'aquí apareix la necessitat de tenir aquest cas d'ús, una vegada l'usuari tingui el moviment generat, podrà consultar les gràfiques que li interessin. Així doncs, l'usuari indicarà quina gràfica li interessa visualitzar i el sistema carregarà la informació necessària i la mostrarà a l'usuari en forma de gràfica.

Capítol 3

Especificació del sistema

3.1 Model de cassos d'ús. Contractes

3.1.1 Cas d'ús 1.1: Crear Model

Precondició:

Cap

Postcondició:

El sistema ha creat un nou model a l'aplicació amb un centre de massa sense cap cadena que surti d'ella.

Activació:

S'inicia en el moment en que l'usuari vol afegir un nou model al sistema.

Flux bàsic:

1. L'usuari necessita iniciar la generació d'un nou model hi ho indica al sistema.
2. El sistema permet a l'usuari indicar que vol crear un model.
3. Finalment, el sistema crea el nou model amb el centre del sistema de referència del model al centre del sistema de referència del programa.

3.1.2 Cas d'ús 1.2: Afegir "Conjunt de cadenes"

Precondició:

Ha de existir un model abans.

Postcondició:

S'ha afegit un Conjunt de cadenes al model.

Activació:

En el moment en que l'usuari li interessi afegir un Conjunt de cadenes per crear una bifurcació en l'estructura del model.

Flux bàsic:

1. L'usuari necessita indicar al sistema que vol afegir un Conjunt de cadenes al model.
2. L'usuari indica al sistema a quin element s'afegirà el Conjunt de cadenes, mitjançant formulari o gràficament.
3. El sistema enregistra el nou centre de cadenes incorporada al model.

Flux Alternatiu:

Element-centre-del-model: L'element seleccionat és un centre del model (2)

3. Acaba el cas d'ús sense enregistrar res.

Element-conjunt-de-cadenes: L'element seleccionat és un conjunt de cadenes (2)

3. Acaba el cas d'ús sense enregistrar res.

Model-no-existeix: El model no existeix (2)

3. El cas d'ús no es pot realitzar.

Element-no-existeix: No existeix l'element seleccionat (2)

3. El cas d'ús acaba.

3.1.3 Cas d'ús 1.3: Afegir cadena al model

Precondició:

Ha de existir un model abans.

Postcondició:

S'ha afegit una cadena al model.

Activació:

En el moment en que l'usuari li interessi afegir elements al model que no formen part d'una cadena que conté el model o simplement perquè encara no ha entrat cap element.

Flux bàsic:

1. L'usuari necessita indicar al sistema que vol generar una nova cadena al model.
2. El sistema pregunta quin tipus d'element és el primer que apareix a la cadena, una Massa o un "Link".
3. L'usuari decideix una Massa, aleshores, el sistema li permet a l'usuari col·locar la Massa, a partir del ratolí, al model de manera que el centre de la Massa serà el mateix que el centre del model o del conjunt de cadenes que s'hagi seleccionat. I sempre es col·locarà la Massa amb el seu eix de rotació (el grau de llibertat) perpendicular a la línia que forma el centre del model amb el punt de connexió amb la Massa. I l'eix Y perpendicular a l'eix de rotació estarà orientat de la mateixa manera que l'eix Y de l'escena.
4. El sistema enregistra la nova Massa incorporada al model.

Flux Alternatiu:

Decideix-Link: L'usuari decideix que el primer element serà un "Link" (2)

3. L'usuari decideix una "Link", aleshores, el sistema li permet a l'usuari col·locar el "Link de manera que primer premerà el botó esquerre del ratolí en el punt del centre del model o de cadenes on el vol col·locar i després determinarà la llargada sense deixar de prémer el botó. El "Link sempre es col·locarà a l'inici de manera que l'eix que passa per dins del "Link", en aquest cas l'eix Y, tindrà la mateixa direcció i sentit que el vector que va del centre del centre del model cap el punt a on va unit el "Link".
4. El sistema enregistra la nova "Link" incorporada al model.

Selecciona-Massa: L'usuari selecciona una Massa del model (3)

1. Acaba el cas d'ús sense registrar res.

Selecciona-Link: L'usuari selecciona un "Link" del model (3)

1. Acaba el cas d'ús sense registrar res.

Model-no-existeix: El model no existeix (1)

2. El cas d'ús s'acaba.

3.1.4 Cas d'ús 1.4: Afegir un "Link" al model

Precondició:

Ha d'existir un model, amb el seu centre i amb una cadena definida.

Postcondició:

S'ha afegit el "Link" a la cadena indicada.

Activació:

El cas comença en el moment en que l'usuari vol incorporar un "Link" al model.

Flux bàsic:

1. L'usuari indica al sistema que vol afegir un "Link" al model.
2. El sistema li permet incorporar
3. L'usuari selecciona un element, aleshores, el sistema li permet a l'usuari col·locar el "Link de manera que primer premerà el botó esquerre del ratolí en el punt del centre del model on el vol col·locar i després determinarà la llargada sense deixar de prémer el botó. El "Link sempre es col·locarà a l'inici de manera que l'eix que passa per dins del "Link", en aquest cas l'eix Y, tindrà la mateixa direcció i sentit que el vector que va del centre del centre de massa cap el punt a on va unit el "Link".
4. El sistema enregistra el nou "Link" incorporada al model.

Flux Alternatiu:

Selecciona-centre-del-model: L'usuari selecciona el centre del model (3)

1. El sistema enregistra el nou "Link" com a el primer element d'una nova cadena.

Selecciona-conjunt-de-cadenes: L'usuari selecciona un conjunt de cadenes del model (3)

1. El sistema enregistra el nou "Link" com a el primer element d'una nova cadena.

Model-no-existeix: El model no existeix (1)

2. El cas d'ús acaba.

3.1.5 Cas d'ús 1.5: Afegir una Massa al model

Precondició:

Ha de existir un model, amb el seu centre i amb una cadena definida.

Postcondició:

S'ha afegit la Massa a la cadena indicada.

Activació:

El cas comença en el moment en que l'usuari vol incorporar una Massa al model.

Flux bàsic:

1. L'usuari indica al sistema que vol afegir una Massa al model.
2. El sistema li permet incorporar
3. L'usuari decideix afegir una Massa, aleshores, el sistema li permet a l'usuari seleccionar l'element al qual anirà connectat la nova Massa.
4. L'usuari selecciona un "link", aleshores, la manera de com anirà connectat serà la següent, l'eix de coordenades d'aquesta nova Massa estarà amb l'eix X orientat de la mateixa manera de com estava en el "link" i l'eix Z també.
5. El sistema enregistra la nova Massa incorporada al model.

Flux Alternatiu:

Selecciona-Massa: L'usuari selecciona una Massa per afegir la massa (4)

4. L'usuari selecciona una Massa, per tant, l'eix X estarà connectat de la mateixa manera que en la Massa anterior i després l'eix Z perquè no coincideixin estarà en perpendicular amb l'eix Z de la Massa anterior.

5. El sistema enregistra la nova Massa incorporada al model.

Selecciona-centre-del-model: L'usuari selecciona el centre del model per afegir la massa (4)

4. L'usuari selecciona el centre de massa, per tant, l'eix X estarà connectat de la mateixa manera que en la Massa anterior i després l'eix Z perquè no coincideixin estarà en perpendicular amb l'eix Z de la Massa anterior.

5. El sistema enregistra la nova Massa incorporada al model.

Selecciona-conjunt-de-cadenes: L'usuari selecciona un centre de cadenes per afegir la massa (4)

4. L'usuari selecciona el centre de massa, per tant, l'eix X estarà connectat de la mateixa manera que en la Massa anterior i després l'eix Z perquè no coincideixin estarà en perpendicular amb l'eix Z de la Massa anterior.

5. El sistema enregistra la nova Massa incorporada al model.

Model-no-existeix: El model no existeix (1)

1. El cas d'ús acaba.

3.1.6 Cas d'ús 1.6: Afegir un Recolzador

Precondició:

Ha de existir un model, amb el seu centre i amb una cadena definida.

Postcondició:

S'ha afegit el Recolzador a la cadena indicada.

Activació:

El cas comença en el moment en que l'usuari vol incorporar un Recolzador al model.

Flux bàsic:

1. L'usuari indica al sistema que vol afegir un Recolzador al model.
2. El sistema li permet incorporar

3. L'usuari decideix afegir un Recolzador, aleshores, el sistema li permet a l'usuari seleccionar l'element al qual anirà connectat la nova Massa.
4. L'usuari selecciona un "link", aleshores, la manera de com anirà connectat serà la següent, l'eix de coordenades d'aquest nou Recolzador estarà amb l'eix X orientat de la mateixa manera de com estava en el "link" i l'eix Z també.
5. El sistema enregistra el nou Recolzador incorporada al model.

Flux Alternatiu:

Selecciona-Massa: L'usuari selecciona una Massa per afegir el Recolzador (4)

4. L'usuari selecciona una Massa, per tant, l'eix X estarà connectat de la mateixa manera que en la Massa anterior i després l'eix Z perquè no coincideixin estarà en perpendicular amb l'eix Z de la Massa anterior.
5. El sistema enregistra la nova Massa incorporada al model.

Selecciona-centre-del-model: L'usuari selecciona el centre del model per afegir la massa (4)

4. L'usuari selecciona el centre de massa, per tant, l'eix X estarà connectat de la mateixa manera que en la Massa anterior i després l'eix Z perquè no coincideixin estarà en perpendicular amb l'eix Z de la Massa anterior.
5. El sistema enregistra la nova Massa incorporada al model.

Model-no-existeix: El model no existeix (1)

1. El cas d'ús acaba.

3.1.7 Cas d'ús 2.1: Eliminar model**Precondició:**

Ha d'existir un model com a mínim al sistema.

Postcondició:

El sistema ha eliminat el model.

Activació:

En el moment en que l'usuari desitja eliminar un model del sistema.

Flux bàsic:

1. L'usuari indica al sistema que vol eliminar un model.
2. L'usuari demana al sistema que elimini el model seleccionat i ho confirma
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: El model no existeix (1)

2. El cas d'ús acaba.

3.1.8 Cas d'ús 2.2: Moure model

Precondició:

Ha d'existir un model com a mínim.

Postcondició:

El sistema ha mogut el model l'espai que l'usuari hagi indicat.

Activació:

En el moment en que l'usuari vulgui moure un model per la finalitat que sigui.

Flux bàsic:

1. L'usuari indica al sistema que vol moure un model.
2. L'usuari indica al sistema quin model és prement el botó esquerra del ratolí sobre el nom del model que vol a la part de l'element gràfic de la interfície gràfica o sobre el formulari en un llistat dels noms dels models.
3. Mantenint premut el botó i desplaçant el ratolí l'usuari aconsegueix moure el model seleccionat.
4. En el moment en que l'usuari deixa de prémer el botó el model es queda a la posició que ha quedat el ratolí. O mitjançant el formulari entrar la nova posició.
5. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: El model no existeix (1)

1. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova posició del model (2)

3. El cas d'ús acaba.

3.1.9 Cas d'ús 2.3: Rotar model

Precondició:

Ha d'existir un model com a mínim.

Postcondició:

El sistema ha rotat el model, sobre si mateix, els angles que hagi indicat l'usuari.

Activació:

En el moment en que l'usuari desitgi modificar l'orientació del model.

Flux bàsic:

1. L'usuari indica al sistema que vol rotar un model.
2. L'usuari indica al sistema quin model és prement el botó esquerra del ratolí sobre el nom del model que vol a la part de l'element gràfic de la interfície gràfica o sobre el formulari en un llistat dels noms dels models.
3. Mantenint premut el botó i desplaçant el ratolí l'usuari aconsegueix rotar el model seleccionat.
4. En el moment en que l'usuari deixa de prémer el botó el model es queda a la posició que ha quedat el ratolí. O mitjançant el formulari entrar la nova posició.
5. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: El model no existeix (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova orientació del model (2)

3. El cas d'ús acaba.

3.1.10 Cas d'ús 2.4: Eliminar cadena**Precondició:**

Ha d'existir algun model amb alguna cadena.

Postcondició:

El sistema ha eliminat la cadena que l'usuari desitjava eliminar.

Activació:

En el moment en que l'usuari desitja eliminar una cadena degut a que no li interessa tenir-la o vol rectificar-la.

Flux bàsic:

1. L'usuari indica al sistema quina cadena de quin model li interessa eliminar, mitjançant l'element gràfic de la interfície gràfica o formulari.
2. L'usuari demana que l'elimini una vegada seleccionat i ho confirma.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Cadena-no-existeix: No existeix la cadena (1)

2. El cas d'ús acaba.

3.1.11 Cas d'ús 2.5: Moure cadena**Precondició:**

Ha d'existir algun model amb alguna cadena.

Postcondició:

El sistema ha mogut la cadena que l'usuari desitjava moure.

Activació:

En el moment en que l'usuari desitja moure una cadena.

Flux bàsic:

1. L'usuari indica al sistema quina cadena de quin model li interessa moure, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes les cadenes dels models.
2. L'usuari prem el botó esquerre del ratolí i sense deixar de pressionar el desplaça. En el moment en que l'usuari aixeca el dit del botó la cadena deixa de moure's amb el ratolí i es queda unit al centre de massa o Conjunt de cadenes fins on s'havia mogut. També ho pot indicar mitjançant formulari.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Cadena-no-existeix: No existeix la cadena (1)

2. El cas d'ús acaba.

Centre-del-model-no-existeix: No existeix el centre del model al qual es vol afegir moure la cadena (1)

2. El cas d'ús acaba.

Conjunt-de-cadenes-no-existeix: No existeix el Conjunt de cadenes al qual es vol moure la cadena (1)

2. El cas d'ús acaba.

3.1.12 Cas d'ús 2.6: Rotar cadena**Precondició:**

Ha d'existir algun model amb alguna cadena.

Postcondició:

El sistema ha rotat la cadena que l'usuari desitjava rotar.

Activació:

En el moment en que l'usuari desitja rotar una cadena.

Flux bàsic:

1. L'usuari indica al sistema quina cadena de quin model li interessa rotar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes les cadenes dels models.
2. L'usuari prem el botó esquerre del ratolí i sense deixar de pressionar i el mou. En el moment en que l'usuari aixeca el dit del botó la cadena deixa de rotar amb el ratolí i es queda a la posició fins on s'havia rotat. O bé mitjançant formulari també pot indicar els angles de rotació que tindrà respecte el centre de massa o de cadenes al qual estigui lligat. El sistema ha de garantir que la nova rotació compleix que la cadena queda unida al centre.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Cadena-no-existeix: No existeix la cadena (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova orientació de la cadena (2)

3. El cas d'ús acaba.

3.1.13 Cas d'ús 2.7: Eliminar "Conjunt de cadenes"**Precondició:**

Ha d'existir algun model amb algun Conjunt de cadenes.

Postcondició:

El sistema ha eliminat el Conjunt de cadenes esperat.

Activació:

En el moment en que l'usuari desitja eliminar un Conjunt de cadenes.

Flux bàsic:

1. L'usuari indica al sistema quin Conjunt de cadenes de quin model li interessa eliminar, mitjançant l'element gràfic de la interfície gràfica o el formulari.
2. L'usuari elimina i confirma.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Conjunt-de-cadenes-no-existeix: No existeix el Conjunt de cadenes (1)

2. El cas d'ús acaba.

3.1.14 Cas d'ús 2.8: Eliminar "Link"

Precondició:

Ha d'existir algun model amb alguna cadena que contingui algun "link".

Postcondició:

El sistema ha eliminat el "link" esperat.

Activació:

En el moment en que l'usuari desitja eliminar un "link".

Flux bàsic:

1. L'usuari indica al sistema quin "link" de quin model li interessa eliminar, mitjançant l'element gràfic de la interfície gràfica o un formulari on apareixen tots els "links" dels models.
2. L'usuari elimina i confirma.

3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Link-no-existeix: No existeix el "Link" (1)

2. El cas d'ús acaba.

3.1.15 Cas d'ús 2.9: Rotar "Link"**Precondició:**

Ha d'existir algun model amb alguna cadena que contingui algun "link".

Postcondició:

El sistema ha rotat el "link" esperat.

Activació:

En el moment en que l'usuari desitja rotar un "link".

Flux bàsic:

1. L'usuari indica al sistema quin "link" de quin model li interessa rotar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes els "links" dels models.
2. L'usuari mou el ratolí amb el botó esquerra premut. El "link" comença a rotar. En el moment en que l'usuari deixa de prémer el botó el "link" i tots els elements que venen a continuació paren de rotar. També es pot realitzar la rotació mitjançant el formulari indicant per cada un dels eixos de rotació l'angle.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Link-no-existeix: No existeix el "Link" (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova orientació del "link" (2)

3. El cas d'ús acaba.

3.1.16 Cas d'ús 2.10: Modificar la llargada del "Link"**Precondició:**

Ha d'existir algun model amb alguna cadena que contingui algun "link".

Postcondició:

El sistema ha modificat la llargada del "link" esperat.

Activació:

En el moment en que l'usuari desitja modificar la llargada d'un "link".

Flux bàsic:

1. L'usuari indica al sistema quin "link" de quin model li interessa allargar o escurçar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes els "links" dels models.
2. L'usuari mou el ratolí amb el botó esquerra premut. El "link" comença allargar-se o ha escurçar-se segons el punter del ratolí. En el moment en que l'usuari deixa de prémer el botó el "link" deixa de modificar la seva llargada. Mitjançant el formulari l'usuari també ha de poder donar la llargada del "Link".
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Link-no-existeix: No existeix el "Link" (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova llargada del "link" (2)

3. El cas d'ús acaba.

3.1.17 Cas d'ús 2.11: Eliminar Massa

Precondició:

Ha de existir algun model amb alguna cadena que contingui alguna Massa.

Postcondició:

El sistema ha eliminat la Massa esperada.

Activació:

En el moment en que l'usuari desitja eliminar una Massa.

Flux bàsic:

1. L'usuari indica al sistema quina Massa de quin model li interessa eliminar, mitjançant l'element gràfic de la interfície gràfica. O a partir del formulari.
2. L'usuari elimina i confirma.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Massa-no-existeix: No existeix el Massa (1)

2. El cas d'ús acaba.

3.1.18 Cas d'ús 2.12: Rotar Massa

Precondició:

Ha d'existir algun model amb alguna cadena que contingui alguna Massa.

Postcondició:

El sistema ha rotat la Massa esperada.

Activació:

En el moment en que l'usuari desitja rotar un Massa.

Flux bàsic:

1. L'usuari indica al sistema quina Massa de quin model li interessa rotar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes les Masses dels models.
2. L'usuari mou el ratolí amb el botó esquerra premut. La Massa comença a rotar. En el moment en que l'usuari deixa de prémer el botó la "joint" i tots els elements que venen a continuació paren de rotar. L'usuari també ha de poder donar els angles de rotació de cada un dels eixos mitjançant formulari.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Massa-no-existeix: No existeix el Massa (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova orientació de la Massa (2)

3. El cas d'ús acaba.

3.1.19 Cas d'ús 2.13: Rotar Recolzador

Precondició:

Ha d'existir algun model amb alguna cadena que contingui algun Recolzador.

Postcondició:

El sistema ha rotat el Recolzador esperat.

Activació:

En el moment en que l'usuari desitja rotar un Recolzador.

Flux bàsic:

1. L'usuari indica al sistema quin Recolzador de quin model li interessa rotar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari.
2. L'usuari mou el ratolí amb el botó esquerra premut. El Recolzador comença a rotar. En el moment en que l'usuari deixa de prémer el botó la "joint" i tots els elements que venen a continuació paren de rotar. L'usuari també ha de poder donar els angles de rotació de cada un dels eixos mitjançant formulari.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Recolzador-no-existeix: No existeix el Recolzador (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova orientació del Recolzador (2)

3. El cas d'ús acaba.

3.1.20 Cas d'ús 2.14: Eliminar Recolzador**Precondició:**

Ha d'existir algun model amb alguna cadena que contingui algun Recolzador.

Postcondició:

El sistema ha eliminat el Recolzador esperat.

Activació:

En el moment en que l'usuari desitja eliminar un Recolzador.

Flux bàsic:

1. L'usuari indica al sistema quin Recolzador de quin model li interessa eliminar, mitjançant l'element gràfic de la interfície gràfica. O a partir del formulari.
2. L'usuari elimina i confirma.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Massa-no-existeix: No existeix el Massa (1)

2. El cas d'ús acaba.

3.1.21 Cas d'ús 2.15: Modificar la llargada del Recolzador

Precondició:

Ha d'existir algun model amb alguna cadena que contingui algun Recolzador.

Postcondició:

El sistema ha modificat la llargada del Recolzador esperat.

Activació:

En el moment en que l'usuari desitja modificar la llargada d'un Recolzador.

Flux bàsic:

1. L'usuari indica al sistema quin recolzador de quin model li interessa allargar o escurçar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari.
2. L'usuari mou el ratolí amb el botó esquerra premut. El Recolzador comença allargar-se o ha escurçar-se segons el punter del ratolí. En el moment en que l'usuari deixa de prémer el botó el Recolzador deixa de modificar la seva llargada. Mitjançant el formulari l'usuari també ha de poder donar la llargada del Recolzador.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Link-no-existeix: No existeix el Recolzador(1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova llargada del Recolzador (2)

3. El cas d'ús acaba.

3.1.22 Cas d'ús 2.16: Modificar l'amplada del Recolzador**Precondició:**

Ha d'existir algun model amb alguna cadena que contingui algun Recolzador.

Postcondició:

El sistema ha modificat l'amplada del Recolzador esperat.

Activació:

En el moment en que l'usuari desitja modificar l'amplada d'un Recolzador.

Flux bàsic:

1. L'usuari indica al sistema quin recolzador de quin model li interessa estrènyer o eixamplar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari.
2. L'usuari mou el ratolí amb el botó esquerra premut. El Recolzador comença estrènyer-se o ha eixamplar-se segons el punter del ratolí. En el moment en que l'usuari deixa de prémer el botó el Recolzador deixa de modificar la seva amplada. Mitjançant el formulari l'usuari també ha de poder donar l'amplada del Recolzador.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Link-no-existeix: No existeix el Recolzador(1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova amplada del Recolzador (2)

3. El cas d'ús acaba.

3.1.23 Cas d'ús 3.1: Afegir un objecte

Precondició:

Cap.

Postcondició:

El sistema ha afegit un nou objecte a l'escena.

Activació:

En l'instant en que un usuari té la necessitat d'afegir un nou objecte a l'escena per tal de que el model o els models puguin interactuar amb ell.

Flux bàsic:

1. L'usuari indica al sistema que vol afegir un nou objecte indicant-li quin tipus vol afegir, un cub, una piràmide, un pla...
2. Mitjançant l'element gràfic de la interfície gràfica, l'usuari indica les mides de l'element i la posició que tindrà.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

3.1.24 Cas d'ús 4.1: Rotar objecte

Precondició:

Ha d'existir algun objecte a l'escena.

Postcondició:

El sistema ha rotat l'objecte esperat.

Activació:

En el moment en que l'usuari desitja rotar un objecte.

Flux bàsic:

1. L'usuari indica al sistema quin objecte li interessa rotar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes els objectes de l'escena.
2. L'usuari mou el ratolí amb el botó esquerra premut. L'objecte comença a rotar. En el moment en que l'usuari deixa de prémer el botó, l'objecte para de rotar. O també, l'usuari mitjançant un formulari indica els angles de rotació de cada un dels eixos de rotació de l'objecte.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Objecte-no-existeix: No existeix l'objecte (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova orientació de l'objecte (2)

3. El cas d'ús acaba.

3.1.25 Cas d'ús 4.2: Escalar objecte

Precondició:

Ha d'existir algun objecte a l'escena.

Postcondició:

El sistema ha escalat l'objecte esperat.

Activació:

En el moment en que l'usuari desitja escalar un objecte.

Flux bàsic:

1. L'usuari indica al sistema quin objecte li interessa escalar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes els objectes de l'escena.
2. L'usuari mou el ratolí amb el botó esquerra premut. L'objecte comença a escalar. En el moment en que l'usuari deixa de prémer el botó, l'objecte para d'escalar. O també, l'usuari mitjançant un formulari indica els factors d'escalat de cada un dels eixos que descriuen la posició i orientació de l'objecte.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Objecte-no-existeix: No existeix l'objecte (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova mida de l'objecte (2)

3. El cas d'ús acaba.

3.1.26 Cas d'ús 4.3: Moure objecte

Precondició:

Ha d'existir algun objecte a l'escena.

Postcondició:

El sistema ha mogut l'objecte esperat.

Activació:

En el moment en que l'usuari desitja moure un objecte.

Flux bàsic:

1. L'usuari indica al sistema quin objecte li interessa moure, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes els objectes de l'escena.
2. L'usuari mou el ratolí amb el botó esquerra premut. L'objecte comença a moure's. En el moment en que l'usuari deixa de prémer el botó, l'objecte

para de moure's. O també, l'usuari mitjançant un formulari indica la nova posició de l'objecte.

3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Objecte-no-existeix: No existeix l'objecte (1)

2. El cas d'ús acaba.

Flux Alternatiu:

Usuari-usa-formulari: L'usuari fa servir el formulari per indicar en valors absoluts quina és la nova posició de l'objecte (2)

3. El cas d'ús acaba.

3.1.27 Cas d'ús 4.4: Duplicar objecte

Precondició:

Ha d'existir algun objecte a l'escena.

Postcondició:

El sistema ha duplicat un objecte esperat.

Activació:

En el moment en que l'usuari desitja duplicar un objecte.

Flux bàsic:

1. L'usuari indica al sistema quin objecte li interessa duplicar, mitjançant l'element gràfic de la interfície gràfica, o mitjançant un formulari on apareixen totes els objectes de l'escena.
2. L'usuari mou el ratolí amb el botó esquerra premut. L'objecte duplicat comença a moure's. En el moment en que l'usuari deixa de prémer el botó, l'objecte duplicat para de moure's. O també, l'usuari mitjançant un formulari indica la nova posició de l'objecte duplicat.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Objecte-no-existeix: No existeix l'objecte (1)

2. El cas d'ús acaba.

3.1.28 Cas d'ús 4.5: Eliminar objecte

Precondició:

Ha d'existir algun objecte a l'escena.

Postcondició:

El sistema ha eliminat un objecte esperat.

Activació:

En el moment en que l'usuari desitja eliminar un objecte.

Flux bàsic:

1. L'usuari indica quin objecte vol eliminar mitjançant l'element gràfic de la interfície gràfica o un llistat en el formulari on apareguin tots els objectes de l'escena.
2. L'usuari elimina l'objecte i ho confirma.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Objecte-no-existeix: No existeix l'objecte (1)

2. El cas d'ús acaba.

3.1.29 Cas d'ús 4.6: Modificar paràmetre objecte

Precondició:

Ha d'existir algun objecte a l'escena.

Postcondició:

El sistema ha modificat el paràmetre desitjat de l'objecte esperat.

Activació:

En el moment en que l'usuari desitja modificar les característiques d'un objecte.

Flux bàsic:

1. L'usuari indica quin objecte vol modificar.
2. El sistema mostra els paràmetres de l'objecte.
3. L'usuari modifica el paràmetre desitjat i confirma.
4. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Objecte-no-existeix: No existeix l'objecte (1)

2. El cas d'ús acaba.

3.1.30 Cas d'ús 5.1: Afegir punt clau**Precondició:**

Ha d'existir algun model.

Postcondició:

El sistema ha creat un punt clau.

Activació:

En el moment en que l'usuari desitja crear un punt clau que tingui el moviment que vulgui calcular.

Flux bàsic:

1. L'usuari indica quin model és el que se li vol afegir un punt clau del moviment.
2. L'usuari indica al sistema que vol afegir un punt clau mitjançant un formulari d'entrada.
3. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

3.1.31 Cas d'ús 5.2: Modificar paràmetre del model articulat

Precondició:

Ha d'existir algun model i aquest ha de tenir algun paràmetre variable per la qual el model es pugui moure.

Postcondició:

El sistema ha modificat l'element variable del model.

Activació:

En el moment en que l'usuari desitja modificar un valor d'entrada del model per aconseguir posicions noves per aquest.

Flux bàsic:

1. L'usuari indica quin model vol que se li modifiqui un paràmetre d'entrada. Ja sigui mitjançant l'element gràfic de la interfície gràfica o mitjançant un llistat en el formulari de la interfície.
2. L'usuari selecciona l'element que vol modificar. El sistema ha de garantir que en tot moment l'usuari pot reconèixer quin és l'element que modificarà
3. Mitjançant l'element gràfic o formulari (donant el valor del paràmetre), l'usuari modificarà el paràmetre variable del model.
4. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Element-no-existeix: No existeix l'element (2)

3. El cas d'ús acaba.

Element-no-té-paràmetre: L'element que es selecciona no té cap paràmetre assignat (2)

3. El sistema ho indica a l'usuari i torna al punt (2).

3.1.32 Cas d'ús 5.3: Afegir punt de contacte

Precondició:

Ha d'existir algun model i algun objecte a l'escena.

Postcondició:

El sistema ha creat un punt de contacte entre el model i l'objecte de l'escena.

Activació:

En l'instant mateix en que l'usuari desitja que hi hagi una interacció entre el model i l'escena per tal de començar a generar forces.

Flux bàsic:

1. L'usuari indica que vol crear un punt de contracte.
2. L'usuari indica quin Recolzador vol que prengui contacte amb un objecte.
3. L'usuari selecciona l'objecte que vol modificar. També pot ser gràficament o per formulari.
4. El sistema dóna una possible posició de com podria quedar el model en el moment del contacte.
5. El sistema enregistra els canvis i ho actualitza en l'espai tridimensional.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

Objecte-no-existeix: No existeix l'objecte (3)

4. El cas d'ús acaba.

3.1.33 Cas d'ús 5.4: Assignar instant de temps

Precondició:

Ha d'existir algun model.

Postcondició:

El sistema ha creat una relació entre el temps i l'estat del model.

Activació:

En l'instant mateix en que l'usuari desitja crear un punt clau del moviment que el sistema tindrà en compte per generar-lo.

Flux bàsic:

1. L'usuari indica quin model és al qual se li vol assignar un instant de temps. Mitjançant formulari o element gràfic.
2. L'usuari mitjançant formulari indica el temps que se li vol assignar.
3. El sistema assigna el temps a l'estat del model que en aquell moment tenia.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

3.1.34 Cas d'ús 5.5: Generar animació

Precondició:

Ha d'existir algun model amb com a mínim dos instants de temps assignats, l'inicial i el final.

Postcondició:

El sistema ha generat una animació on el model passa per tots els estats dels instants de temps assignats. La interpolació que es fa de manera senzilla, seria el moviment esperat ideal.

Activació:

En el moment en que l'usuari decideixi que tots els punts claus han estat introduïts i que s'han donat tots els punts de contacte necessaris estant introduïts.

Flux bàsic:

1. L'usuari indica que vol generar l'animació mitjançant el formulari.
2. El sistema realitza els càlculs i mostra l'animació.

3. El sistema guarda tota la informació recollida i la guarda.

Flux Alternatiu:

Moviment-no-possible: El moviment que es vol calcular no es pot calcular
(2)

3. El sistema ho indica a l'usuari.

3.1.35 Cas d'ús 6.1: Eliminar punt clau**Precondició:**

Ha d'existir algun model amb algun punt clau assignat.

Postcondició:

El sistema ha eliminat el punt clau que sabia assignat al model i que es volia eliminar.

Activació:

Quan l'usuari desitgi eliminar un punt clau ja que considera que no li pertoca aquell temps o que el resultat en la generació de l'animació no és el que s'esperava.

Flux bàsic:

1. L'usuari, mitjançant formulari, selecciona el punt clau que desitja.
2. L'usuari indica que vol eliminar i confirma.
3. El sistema registra el canvi i ho fa visible a l'usuari.

Flux Alternatiu:

punt-clau-no-existeix: No existeix el punt clau (1)

2. El cas d'ús acaba.

3.1.36 Cas d'ús 6.2: Modificar instant de temps**Precondició:**

Ha d'existir algun model amb algun instant de temps assignat.

Postcondició:

El sistema ha modificat l'instant de temps que sabia assignat al model i que es volia modificar.

Activació:

Quan l'usuari desitgi modificar un instant de temps ja que considera que no li pertoca aquell temps o que el resultat en la generació de l'animació no és el que s'esperava.

Flux bàsic:

1. L'usuari, mitjançant formulari, selecciona la relació de instant de temps amb l'estat del model que desitja.
2. L'usuari indica el nou temps mitjançant el formulari i confirma.
3. El sistema registra el canvi i ho fa visible a l'usuari.

Flux Alternatiu:

Relacio-model-temps-no-existeix: No existeix la relació model-temps (1)

2. El cas d'ús acaba.

3.1.37 Cas d'ús 6.3: Eliminar punt de contacte**Precondició:**

Ha d'existir algun model amb algun punt de contacte assignat.

Postcondició:

El sistema ha eliminat el punt de contacte que sabia assignat al model i que es volia eliminar.

Activació:

Quan l'usuari desitgi eliminar un punt de contacte ja que considera que el resultat en la generació de l'animació no és el que s'esperava.

Flux bàsic:

1. L'usuari, mitjançant formulari o gràficament, selecciona el punt de contacte que desitja. El sistema ha de deixar clar en tot moment quin és el punt de contacte que ha seleccionat.

2. L'usuari indica que vol eliminar i confirma.
3. El sistema registra el canvi i ho fa visible a l'usuari.

Flux Alternatiu:

Punt-de-contacte-no-existeix: No existeix el punt de contacte (1)

2. El cas d'ús acaba.

3.1.38 Cas d'ús 6.4: Modificar punt de contacte

Precondició:

Ha d'existir algun model amb algun punt de contacte assignat.

Postcondició:

El sistema ha modificat el punt de contacte que sabia assignat al model i que es volia modificar.

Activació:

Quan l'usuari desitgi modificar un punt de contacte ja que considera que el resultat en la generació de l'animació no és el que s'esperava.

Flux bàsic:

1. L'usuari, mitjançant formulari, selecciona la relació de instant de temps amb l'estat del model que desitja. És a dir, selecciona el Punt Clau que vulgui.
2. L'usuari indica la nova posició del punt de contacte, ja sigui per formulari o a partir de l'element gràfic.
3. El sistema registra el canvi i ho fa visible a l'usuari.

Flux Alternatiu:

Punt-de-contacte-no-existeix: No existeix el punt de contacte (1)

2. El cas d'ús acaba.

3.1.39 Cas d'ús 6.5: Modificar fotograma de l'animació

Precondició:

Ha d'existir alguna animació generada.

Postcondició:

El sistema ha modificat el fotograma de l'animació que es volia modificar i ha regenerat l'animació per tal d'actualitzar el nou moviment.

Activació:

Una vegada generada l'animació l'usuari pot decidir que la física que segueix el model és diferent i per tant, poder modificar algun punt en concret, creant un nou punt clau important, és el detonant d'aquest cas d'ús.

Flux bàsic:

1. L'usuari, mitjançant formulari, selecciona l'instant concret de temps que interessa modificar.
2. L'usuari modifica el fotograma amb algun cas d'ús anterior, ja sigui afegint un punt de contacte, o modificant-lo o eliminant-lo, o també pot modificar el temps.
3. L'usuari confirma el canvi.
4. El sistema regenera l'animació
5. El sistema registra el canvi, la nova informació i mostra la nova animació a l'usuari.

Flux Alternatiu:

temps-no-existeix No existeix el temps a l'animació (1)

2. El sistema ho informa a l'usuari i torna al pas (1).

Cas d'ús 6.6: Modificar paràmetre d'una etapa**Precondició:**

Ha d'existir alguna etapa.

Postcondició:

Ha actualitzat l'etapa si eren possibles els canvis i els ha fet consistents al sistema.

Activació:

Quan l'usuari necessita modificar la trajectòria degut que la trajectòria no pren uns valors desitjats.

Flux bàsic:

1. L'usuari, mitjançant formulari, selecciona l'etapa que es vol modificar.
2. L'usuari modifica el paràmetre desitjat.
3. El sistema re-calcula la nova trajectòria o el nou punt d'aplicació si s'escau i si és necessari re-calculer la posició d'algun Punt Clau.
4. El sistema registra el canvi, la nova informació i mostra la nova trajectòria a l'usuari.

Flux Alternatiu:

valor-no-vàlid El sistema no pot re-calculer els canvis degut a que algun canvi en algun Centre de Massa de un Punt Clau no es pot assolir degut a les restriccions dels graus de llibertat(3)

4. El sistema ho informa a l'usuari i desfà els canvis fets per l'usuari.

Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau**Precondició:**

Ha d'existir algun Punt Clau.

Postcondició:

Ha actualitzat el Punt Clau si era possibles i els ha fet consistents al sistema.

Activació:

Quan l'usuari necessita modificar la configuració d'un model en un Punt Clau determinat.

Flux bàsic:

1. L'usuari, mitjançant formulari, selecciona el Punt Clau que es vol modificar.
2. L'usuari modifica el paràmetre desitjat.
3. El sistema modifica el Punt Clau i si és necessari re-calcula el conjunt de etapes i el fa consistent amb la nova configuració del Punt Clau.

4. El sistema registra el canvi, la nova informació i mostra la nova configuració a l'usuari.

Flux Alternatiu:

valor-no-vàlid El sistema no pot re-calculer els canvis degut a que la configuració no és compatible amb les restriccions dels graus de llibertat(3)

4. El sistema ho informa a l'usuari i desfà els canvis fets per l'usuari.

3.1.40 Cas d'ús 7.1: Pan**Precondició:**

Cap

Postcondició:

El punt de visió s'ha d'haver desplaçat cap on l'usuari vulgui.

Activació:

En el moment en que l'usuari vulgui canviar el punt de visió.

Flux bàsic:

1. L'usuari indicarà al sistema que necessita fer un Pan de la visió de l'espai 3D.
2. L'usuari indica la nova posició del punt de visió.
3. El sistema recull la informació i actualitza el punt de visió de l'espai 3D.

3.1.41 Cas d'ús 7.2: Zoom**Precondició:**

Cap

Postcondició:

El camp de visió s'ha d'haver modificat segons com l'usuari vulgui.

Activació:

En el moment en que l'usuari vulgui canviar l'angle de visió.

Flux bàsic:

1. L'usuari indicarà al sistema que necessita fer un Zoom de la visió de l'espai 3D.
2. L'usuari indica el nou angle de visió de l'espai 3D.
3. El sistema recull la informació i actualitza l'angle de visió de l'espai 3D.

3.1.42 Cas d'ús 7.3: Rotació**Precondició:**

Cap

Postcondició:

El punt de visió s'ha d'haver rotat cap on l'usuari vulgui.

Activació:

En el moment en que l'usuari vulgui canviar el punt de visió.

Flux bàsic:

1. L'usuari indicarà al sistema que necessita fer una Rotació de la visió de l'espai 3D.
2. L'usuari indica l'angle de rotació del punt de visió.
3. El sistema recull la informació i actualitza el punt de visió de l'espai 3D.

3.1.43 Cas d'ús 8.1: Guardar model**Precondició:**

Ha d'existir un model com a mínim.

Postcondició:

El sistema ha guardat el model en un fitxer de manera que es pugui carregar més tard.

Activació:

En el moment en que l'usuari vulgui guardar un model per la raó que sigui.

Flux bàsic:

1. L'usuari indicarà al sistema quin model desitja guardar.
2. L'usuari indica el nom del fitxer i la direcció on vol guardar.
3. El sistema guarda el model en un fitxer escrivint tota la informació necessària per poder reconstruir-lo més tard en el sistema.

Flux Alternatiu:

Model-no-existeix: No existeix el model (1)

2. El cas d'ús acaba.

direcció-no-vàlida: La direcció no és vàlida (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-ja-existeix: El nom existeix a la direcció especificada (2)

3. El sistema demana a l'usuari si vol sobre escriure el model que ja existeix. Si diu que NO torna a (2), si diu que SÍ va a (3).

3.1.44 Cas d'ús 8.2: Carregar model

Precondició:

Cap.

Postcondició:

El sistema ha carregat el model d'un fitxer de manera la informació reconstruïda és la mateixa que en el moment de guardar-la.

Activació:

En el moment en que l'usuari vulgui carregar un model que havia guardat prèviament.

Flux bàsic:

1. L'usuari indicarà al sistema quin model desitja carregar.
2. L'usuari indica el nom del fitxer i la direcció d'on es vol carregar.
3. El sistema carrega el model d'un fitxer carregant tota la informació necessària per reconstruir el model.

Flux Alternatiu:

direcció-no-vàlida: La direcció no és vàlida (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

3.1.45 Cas d'ús 8.3: Guardar escena**Precondició:**

Cap.

Postcondició:

El sistema ha guardat l'escena en un fitxer de manera que es pugui carregar més tard.

Activació:

En el moment en que l'usuari vulgui guardar una escena per la raó que sigui.

Flux bàsic:

1. L'usuari indicarà al sistema que desitja guardar l'escena.
2. L'usuari indica el nom del fitxer i la direcció on vol guardar.
3. El sistema guarda l'escena en un fitxer escrivint tota la informació necessària per poder reconstruir-lo més tard en el sistema.

Flux Alternatiu:

direcció-no-vàlida: La direcció no és vàlida (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-ja-existeix: El nom existeix a la direcció especificada (2)

3. El sistema demana a l'usuari si vol sobreescriure l'escena que ja existeix. Si diu que NO torna a (2), si diu que SÍ va a (3).

3.1.46 Cas d'ús 8.4: Carregar escena

Precondició:

Cap.

Postcondició:

El sistema ha carregat l'escena d'un fitxer de manera la informació reconstruïda és la mateixa que en el moment de guardar-la.

Activació:

En el moment en que l'usuari vulgui carregar una escena que havia guardat prèviament.

Flux bàsic:

1. L'usuari indicarà al sistema quina escena desitja carregar.
2. L'usuari indica el nom del fitxer i la direcció d'on es vol carregar.
3. El sistema carrega l'escena d'un fitxer carregant tota la informació necessària per reconstruir l'escena.

Flux Alternatiu:

direcció-no-vàlida: La direcció no és vàlida (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

3.1.47 Cas d'ús 8.5: Crear projecte

Precondició:

Cap.

Postcondició:

El sistema ha creat un projecte buit.

Activació:

En el moment en que l'usuari desitgi crear un projecte que constarà de un o més models amb un escena i un conjunt de punts claus, amb punts de contacte, que descriuin un moviment.

Flux bàsic:

1. L'usuari indicarà al sistema que desitja crear un nou projecte.
2. L'usuari indica el nom del projecte.
3. El sistema genera un nou projecte buit i el fa visible a l'usuari.

Flux Alternatiu:

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-ja-existeix: El nom existeix a la direcció especificada (2)

3. El sistema demana a l'usuari si vol sobre escriure el projecte que ja existeix. Si diu que NO torna a (2), si diu que SÍ va a (3).

3.1.48 Cas d'ús 8.6: Guardar projecte

Precondició:

Ha d'existir algun projecte al sistema.

Postcondició:

El sistema ha guardat el projecte en un fitxer de manera que es pugui carregar més tard.

Activació:

En el moment en que l'usuari vulgui guardar un projecte per la raó que sigui.

Flux bàsic:

1. L'usuari indicarà al sistema que desitja guardar el projecte.
2. L'usuari indica el nom del fitxer i la direcció on vol guardar.
3. El sistema guarda el projecte en un fitxer escrivint tota la informació necessària per poder reconstruir-lo més tard en el sistema.

Flux Alternatiu:

direcció-no-vàlida: La direcció no és vàlida (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-ja-existeix: El nom existeix a la direcció especificada (2)

3. El sistema demana a l'usuari si vol sobre escriure el projecte que ja existeix. Si diu que NO torna a (2), si diu que SÍ va a (3).

3.1.49 Cas d'ús 8.7: Carregar projecte**Precondició:**

Cap.

Postcondició:

El sistema ha carregat el projecte d'un fitxer de manera la informació reconstruïda és la mateixa que en el moment de guardar-la.

Activació:

En el moment en que l'usuari vulgui carregar un projecte que havia guardat prèviament.

Flux bàsic:

1. L'usuari indicarà al sistema quin projecte desitja carregar.
2. L'usuari indica el nom del fitxer i la direcció d'on es vol carregar.
3. El sistema carrega el projecte d'un fitxer carregant tota la informació necessària per reconstruir els models, l'escena i el moviment si s'escau.

Flux Alternatiu:

direcció-no-vàlida: La direcció no és vàlida (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

3.1.50 Cas d'ús 8.8: Afegir gràfica

Precondició:

Ha d'existir un projecte que contingui un model i una escena.

Postcondició:

El sistema ha creat una relació de temps amb un paràmetre de qualsevol part del model i s'ha afegit una gràfica que mostra aquesta relació.

Activació:

En el moment en que un usuari vulgui conèixer el valor de algun paràmetre del model en el moment de realitzar un moviment.

Flux bàsic:

1. L'usuari indicarà al sistema quin paràmetre de quin model desitjat monitoritzar.
2. L'usuari indica el nom del gràfic.
3. El sistema fa consistent aquesta informació.

Flux Alternatiu:

nom-no-vàlid: El nom no és vàlid (2)

3. El sistema informa a l'usuari i torna al pas (2).

nom-ja-existeix: El nom existeix a la direcció especificada (2)

3. El sistema demana a l'usuari si vol sobre escriure la gràfica que ja existeix. Si diu que NO torna a (2), si diu que SÍ va a (3).

3.1.51 Cas d'ús 8.9: Eliminar gràfica

Precondició:

Ha d'existir com a mínim una gràfica.

Postcondició:

El sistema ha eliminat la gràfica esperada.

Activació:

En el moment en que un usuari no vulgui saber més informació d'una gràfica, o senzillament per un error en el moment de crear-la.

Flux bàsic:

1. L'usuari indicarà al sistema quina gràfica vol eliminar.
2. L'usuari indica elimina i confirma.
3. El sistema fa consistent aquesta informació.

Flux Alternatiu:

Gràfica-no-existeix: No existeix la Gràfica (1)

2. El cas d'ús acaba.

3.1.52 Cas d'ús 8.10: Visualitzar gràfica

Precondició:

Ha d'existir com a mínim una gràfica.

Postcondició:

El sistema mostra la informació de la gràfica.

Activació:

En el moment en que un usuari vulgui conèixer la informació que mostra la gràfica.

Flux bàsic:

1. L'usuari indicarà al sistema quina gràfica vol visualitzar.
2. El sistema mostra la gràfica.

Flux Alternatiu:

Gràfica-no-existeix: No existeix la Gràfica (1)

2. El cas d'ús acaba.

3.2 Representació esquemàtica dels cassos d'ús

3.2.1 Generació de models

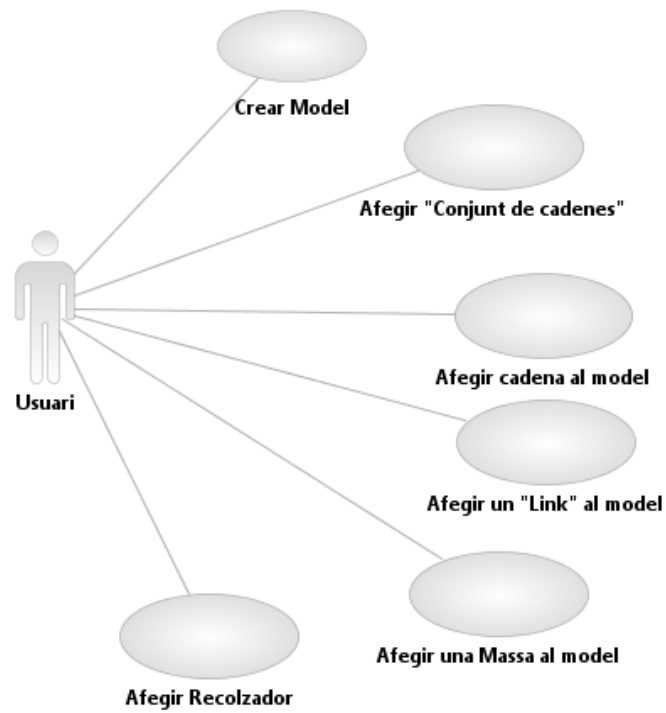


Figura 3.1: Esquema de la funcionalitat 1: Generació de models

3.2.2 Modificació de models

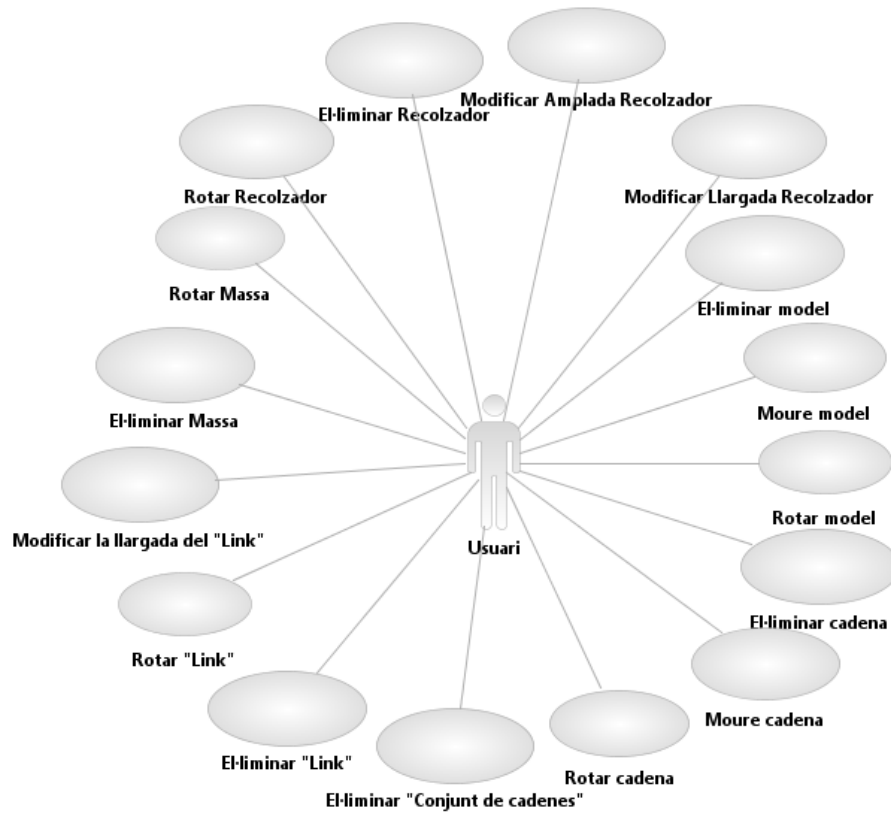


Figura 3.2: Esquema de la funcionalitat 2: Modificació de models

3.2.3 Generació d'escenes



Figura 3.3: Esquema de la funcionalitat 3: Generació d'escenes

3.2.4 Modificació d'escenes

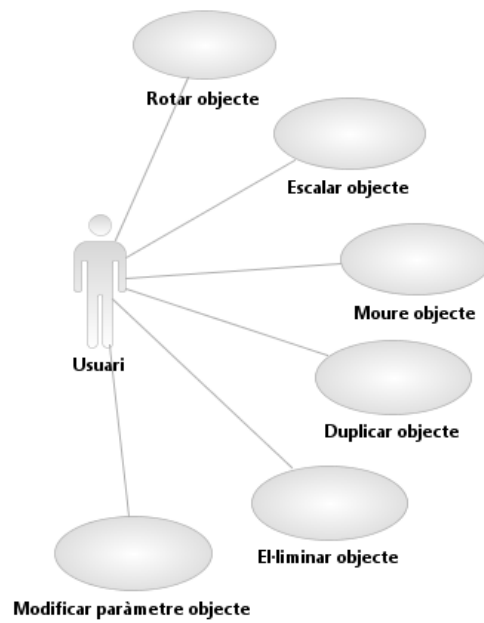


Figura 3.4: Esquema de la funcionalitat 4: Modificació d'escenes

3.2.5 Generació d'animacions

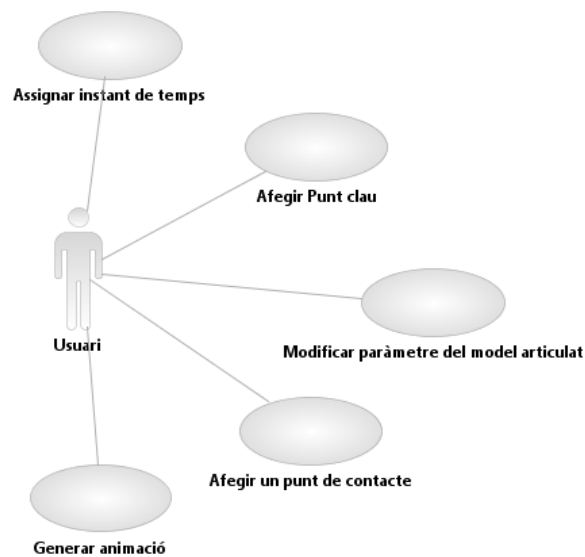


Figura 3.5: Esquema de la funcionalitat 5: Generació d'animacions

3.2.6 Modificació d'animacions

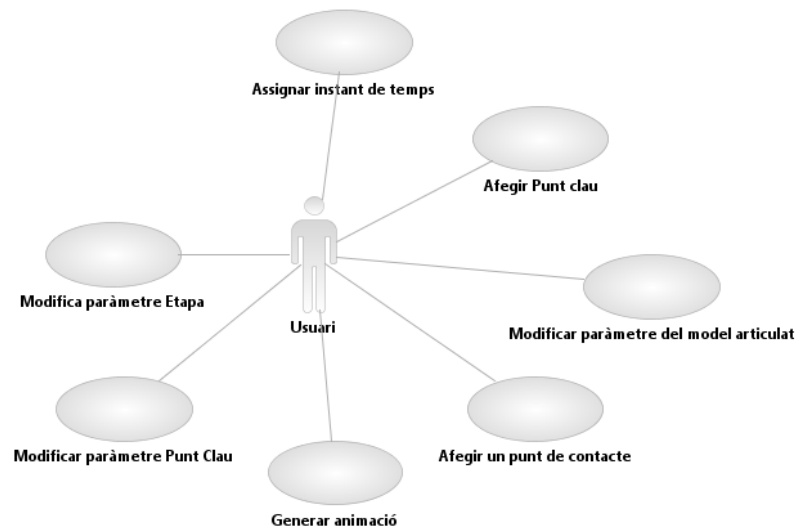


Figura 3.6: Esquema de la funcionalitat 6: Modificació d'animacions

3.2.7 Gestió de la visió de l'espai tridimensional

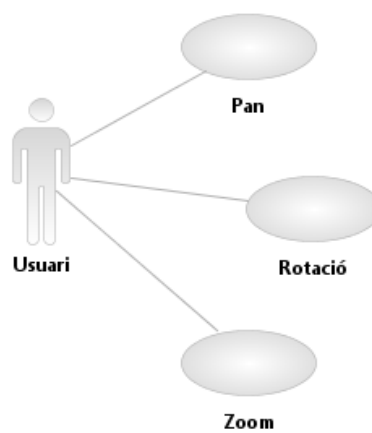


Figura 3.7: Esquema de la funcionalitat 7: Gestió de la visió de l'espai tridimensional

3.2.8 Gestió de la informació

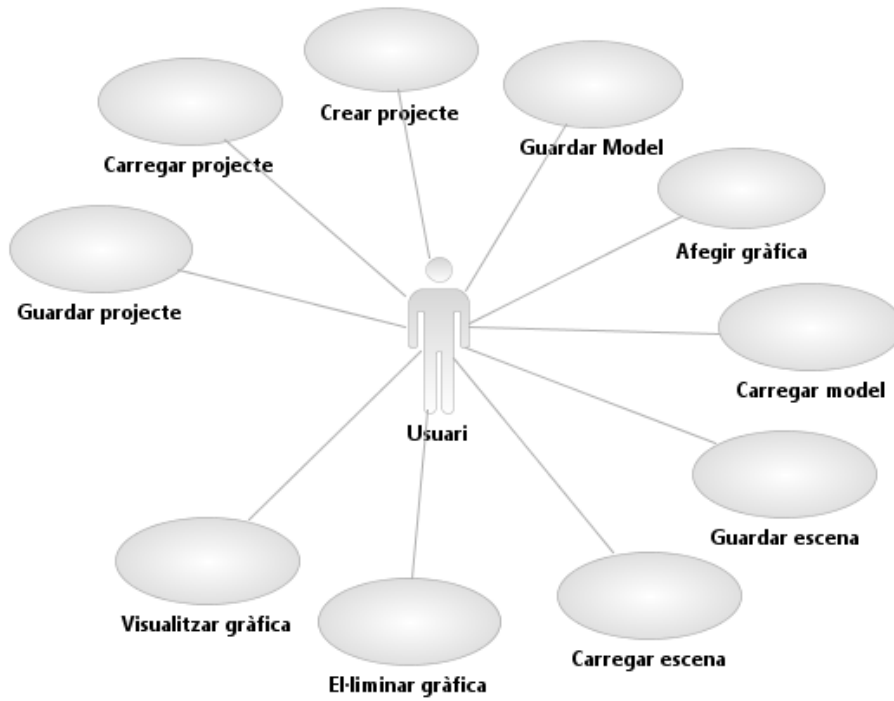


Figura 3.8: Esquema de la funcionalitat 8: Gestió de la informació

Capítol 4

Model d'Anàlisi

4.1 Model Conceptual de les classes

4.1.1 Model Conceptual de la funcionalitat 1

Per aquesta funcionalitat, el que cal tenir en compte és que tractem amb models principalment, que estant formats per cadenes d'elements i conjunt de cadenes, que alhora aquestes últimes tenen cadenes que surten d'elles. Per descriure aquestes cadenes, es té una taula amb valors, la Taula D-H. I cal tenir en compte també que els elements poden ser de dos tipus, masses o articulacions (Inertes o articulades depenent de si volem monitoritzar només informació en un cert punt de l'estructura o és que realment representa una articulació i té un grau de llibertat) i links. A la figura 4.1 es descriu tot el que s'ha dit anteriorment:

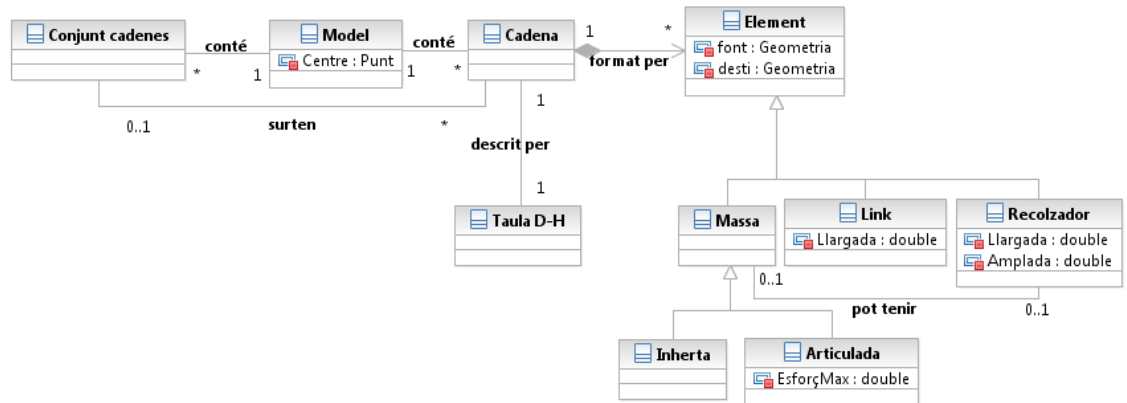


Figura 4.1: Model Conceptual de la funcionalitat 1: Generació de models

Cal remarcar que la Taula D-H és informació derivada dels elements que formen una cadena i que d'alguna manera la informació d'un lloc i l'altre és redundant. L'explicació es troba en el moment en que parlem de física, necessitem parlar de números i no d'elements. Així doncs, per poder realitzar els càlculs necessaris, cal tenir la Taula D-H de cada una de les cadenes d'un model.

4.1.2 Model Conceptual de la funcionalitat 2

En aquest cas, hi ha operacions de modificar el model, les cadenes i els elements concrets de cada un. Així doncs, es permet modificar posicions i orientacions. Així doncs cal tenir una classe que controli totes aquestes transformacions i que s'encarregui de l'espai tridimensional i del format de punts.

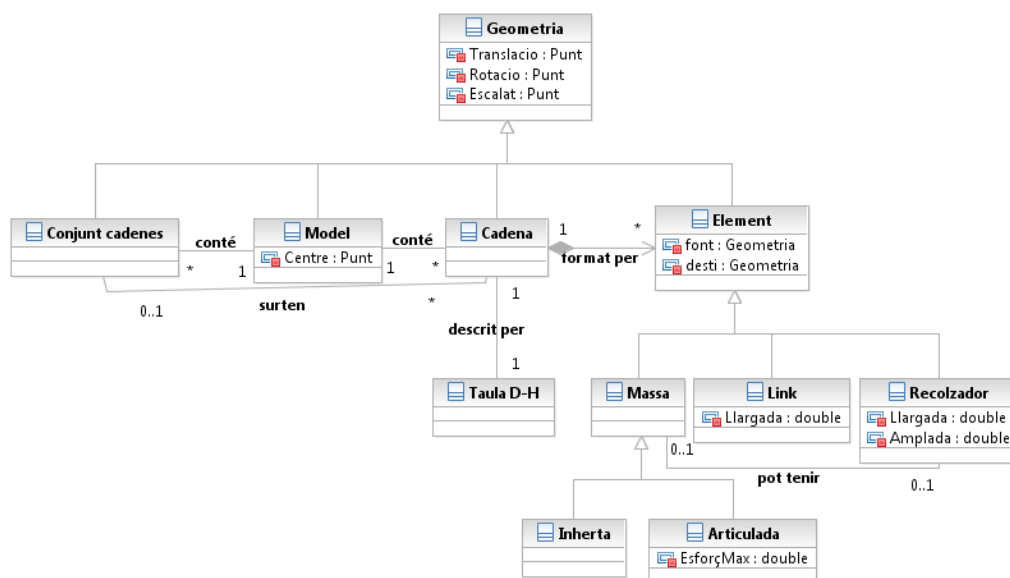


Figura 4.2: Model Conceptual de la funcionalitat 2: Modificació de models

La classe que conté la informació de les transformacions és la classe *Geometria*. Aquesta classe, és la super-classe de totes les classes que tenen una representació en l'espai tridimensional. Així doncs, podem saber en tot moment on es troba la geometria de cada un de les classes de la qual hereten la informació de la classe *Geometria*.

4.1.3 Model Conceptual de la funcionalitat 3

En aquesta funcionalitat, el que fa falta és un conjunt de classes encarregades de mantenir una escena amb tots els seus objectes. Així doncs una descripció de com seria les classes que s'encarreguen d'aquesta funcionalitat seria la que ens mostra la figura 4.3:

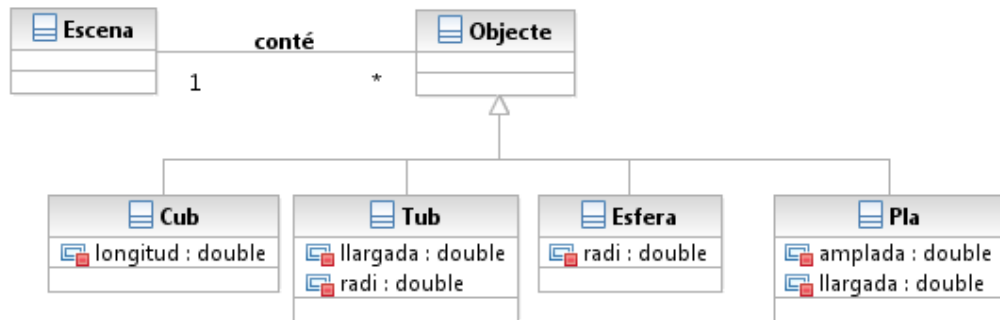


Figura 4.3: Model Conceptual de la funcionalitat 3: Generació d'escenes

Senzillament, cal tenir una classe escena que conté tots els objectes d'una escena i que aquests poden ser de diferents tipus (Cubs, tubs, Esferes, plans...) i que cada un d'ells tenen propietats que els descriuen.

4.1.4 Model Conceptual de la funcionalitat 4

De la mateixa manera que en la modificació de models, cal tenir una classe que controli tot el que fa referència a les transformacions tridimensionals que es donen en el nostre sistema. Així doncs:

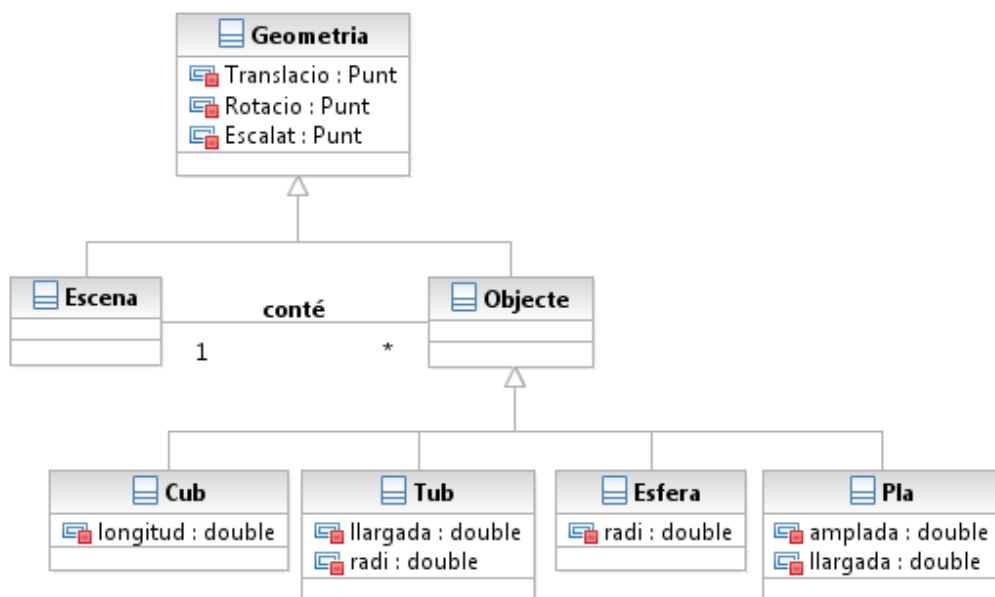


Figura 4.4: Model Conceptual de la funcionalitat 4: Modificació d'escenes

S'actua de la mateixa manera, totes les transformacions que es realitzin sobre la classe que sigui, queden guardades en els atributs que hereten les classes que tenen com a super-classe la classe Geometria. Un cas especial és l'escena ja que, és un element que pròpiament, no té una representació tridimensional en l'espai, però sí que té uns eixos de coordenades que s'han de respectar. En aquest cas, el que representa el seguit de transformacions, és el punt de visió de l'observador.

4.1.5 Model Conceptual de la funcionalitat 5

En aquest cas, el que cal per poder realitzar una animació és primer de tot tenir una classe que gestioni els diferents fotogrames que caldrà calcular. Aquests fotogrames tindran la plantilla d'una classe que tindrà les variables

del model, aquells components que si es modifiquen, canvia l'estat del model. Amb aquesta mateixa plantilla es forma els punts clau que representen la primera part del moviment i que en marquen el comportament. Se sap que aquests punts claus poden tenir punts de contacte i que són fruit de la interacció del model amb l'escena. Una representació de tot el que s'acaba de comentar es troba en la següent figura:

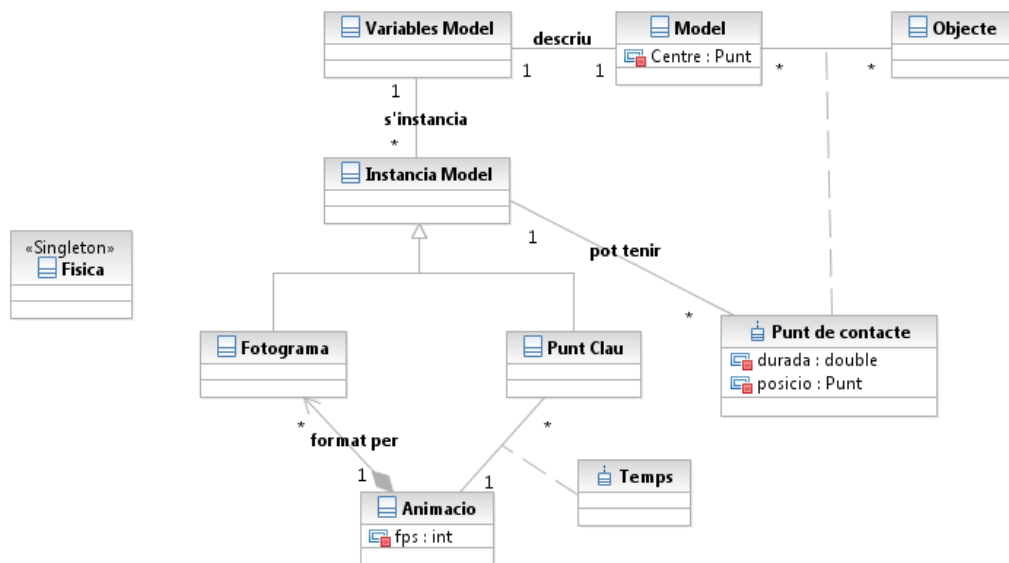


Figura 4.5: Model Conceptual de la funcionalitat 5: Generació de l'animació

4.1.6 Model Conceptual de la funcionalitat 6

Pel que fa a la modificació d'animacions, el que cal veure que les classes seran les mateixes, l'únic que es modificarà serà la informació d'un fotograma i caldrà re-calcular la resta per poder donar el nou moviment. Així doncs:

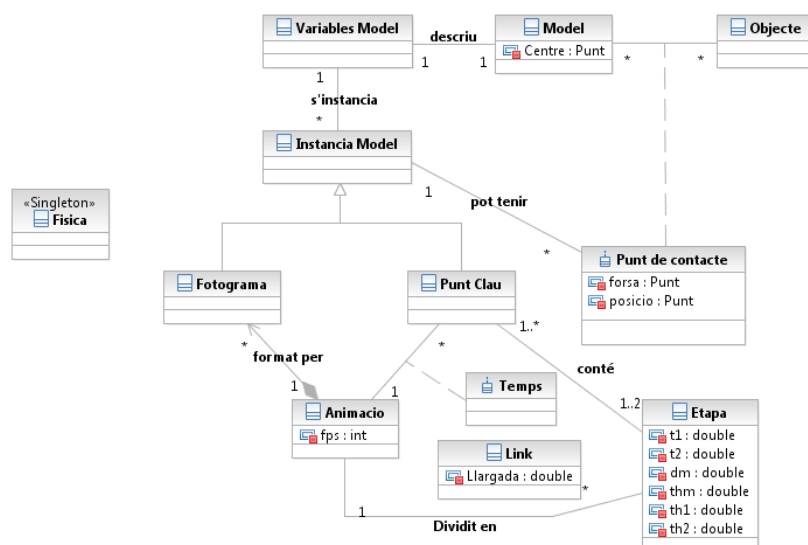


Figura 4.6: Model Conceptual de la funcionalitat 6: Modificació de l'animació

En aquest cas, cal la mateixa informació calculada per si fa falta refer l'animació, i l'únic per modificar cal donar l'usuari la capacitat de retocar els fotogrames calculats.

4.1.7 Model Conceptual de la funcionalitat 7

Per aquesta funcionalitat només cal una classe, la que guardi la informació referent al 3D i que el sistema sigui capaç de consultar i modificar en el moment que faci falta:



Figura 4.7: Model Conceptual de la funcionalitat 7: Gestió de l'espai tridimensional

4.1.8 Model Conceptual de la funcionalitat 8

I finalment en la part de gestió de la informació referent als projectes, les classes necessàries per mantenir la informació són la del model i l'escena amb

la seva informació i el projecte per englobar-ho tot. També caldrà una classe Gràfica que mostrarà informació del model o dels models del projecte que tinguin un moviment associat. Podem ajuntar models i escenes en un mateix projecte i fer que un model o una escena pertanyin a més d'un projecte:

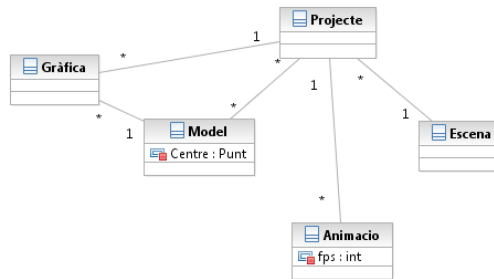


Figura 4.8: Model Conceptual de la funcionalitat 8: Gestió de Projectes

4.1.9 Model Conceptual complet

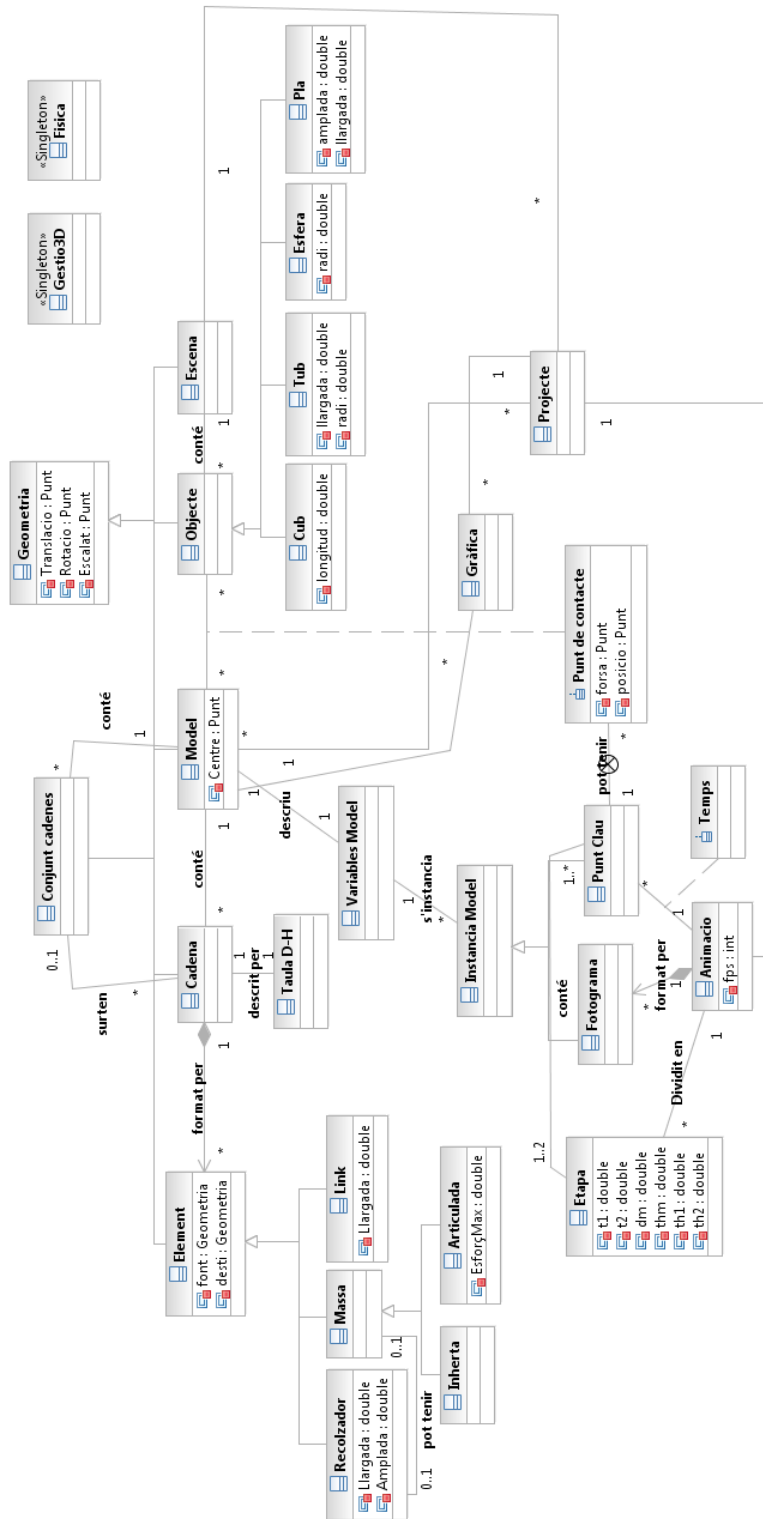


Figura 4.9: Model Conceptual complet

4.2 Diagrames de seqüència i VOPC's

Abans de posar tots els VOPC's i els diagrames de seqüència cal explicar una mica que significa cada cosa. Un VOPC ens indica quines classes intervenen en el cas d'ús i quines interactuen amb qui. També ens informa de quines classes són les que interactuen amb l'exterior (les Boundary, la simbologia és una línia vertical a l'esquerra amb una rodona i unides per una línia horitzontal), les que porten el control del cas d'ús (les Control, són les que tenen el símbol com a rodona i una fletxa que ressegueix la circumferència) i finalment les classes que representen la realitat (les Entity, una rodona sobre una línia horitzontal). Són fàcilment identificables en els dibuixos.

Per altre banda, tenim els diagrames de seqüència que representen el flux d'operacions internes que crida el sistema per tal de dur a terme el cas d'ús. Les classes queden representades en rectangles i es mostra amb un flux que va de d'alt cap avall com van cridant les diferents operacions, d'aquesta manera podem determinar un ordre d'execució.

A continuació per cada cas d'ús prioritzat es mostra els seu VOPC i el diagrama de seqüència.

4.2.1 Cas d'ús 1.1: Crear Model

VOPC

En aquesta figura, podem veure la classe `EntradaModel` i `CrearModel` que són la Boundary i la Control respectivament. També tenim la `Model` i `Projecte` que són Entity, és a dir, elements propis del programa. Com a primera figura que tenim d'un VOPC, podem veure que que tenim una classe que s'encarrega de recollir la informació necessària per portar a terme el cas d'ús, l'`EntradaModel`, una altre per controlar aquesta informació i administrar-la de manera que es faci consistent la informació fent ús de classes que representen la informació, en aquest cas la classe `CrearModel`, i d'un conjunt de classes que representen informació, com és el cas de `Model` i `Projecte`. Només veien aquesta figura, podem veure que la classe `EntradaModel` no té coneixement de les classes `Model` i `Projecte` i que no interactuarà amb elles directament, sinó que ho farà a partir de la classe `CrearModel`. Així doncs, de manera general tots els cassos d'ús seran així, tindrem una classe de recollida d'informació, una o algunes classes de control i varies classes Entity que representen informació o elements de la realitat.

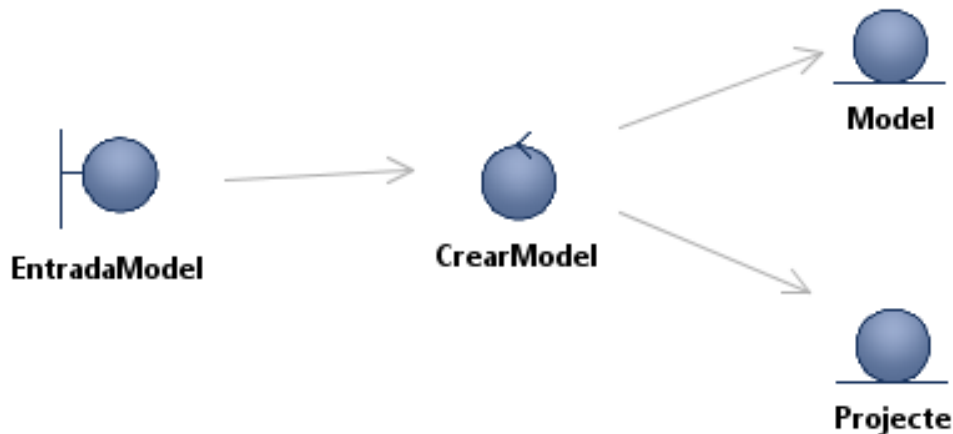


Figura 4.10: VOPC de Crear Model

Diagrama de seqüència

Com s'ha comentat anteriorment, els diagrames de seqüència mostren el flux d'informació i d'operacions que es porten a terme, a nivell software,

en el cas d'ús concret que descriuen. En aquest cas, tenim com la classe `EntradaModel` envia informació a la classe control `CrearModel` mitjançant l'operació `crearModel(...)`. En aquesta situació, per crear un model es necessita la posició i l'orientació. A més, i ja dins de l'operació `crearModel(...)`, cal tenir un identificador per saber en tot moment de quin model estem parlant. En aquest sistema, en tot moment, tindrem un identificador per tot el que faci falta, això inclou, models, masses, "links", cadenes, "conjunt de masses", objectes, gràfiques... Per tal de conèixer en tot moment quin element estem tractant. Com a primer diagrama de seqüència, es posa com seria una creació d'un element, passant-li la informació necessària a més de l'identificador, en els següents diagrames, l'identificador se sobre entén i s'omet, per tal de que hi hagi espai per tot. Una vegada creat el model, cal afegir-lo en el projecte que s'està editant amb el sistema.

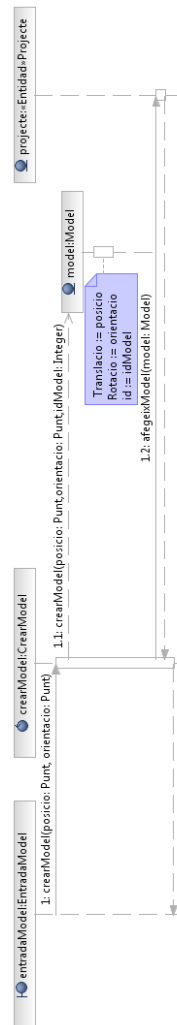


Figura 4.11: Diagrama de seqüència de Crear Model

4.2.2 Cas d'ús 1.2: Afegir "Conjunt de cadenes"

VOPC

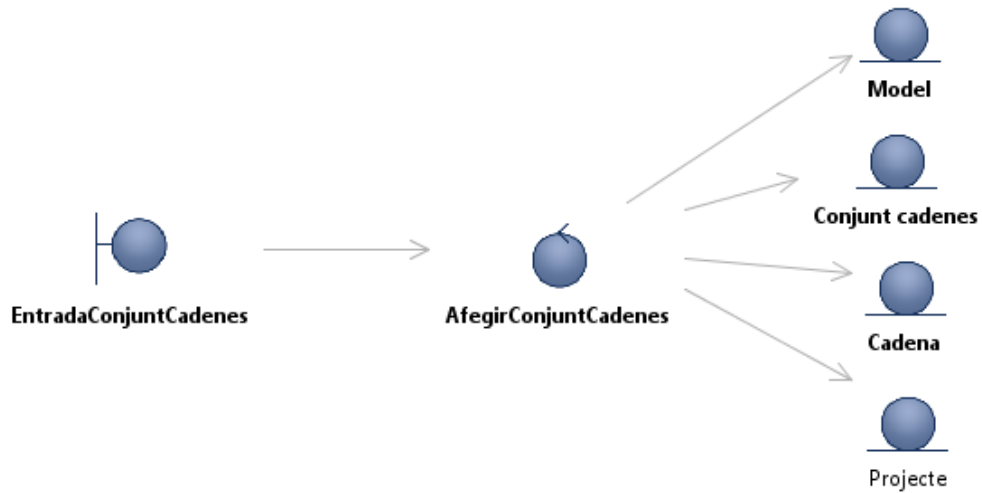


Figura 4.12: VOPC de Afegir "Conjunt de cadenes"

Diagrama de seqüència

En aquest cas concret cal afegir un "Conjunt de cadenes", com a idea general, cal tenir informació primer de la orientació que tindrà respecte l'últim sistema de referència de la cadena que conté aquest "Conjunt de cadenes". Així doncs, també cal conèixer la cadena que s'ha seleccionat i on s'afegirà el Conjunt. Per tant, caldrà crear el Conjunt, associar el conjunt amb la cadena i la cadena amb el conjunt com a "Cadena Font".

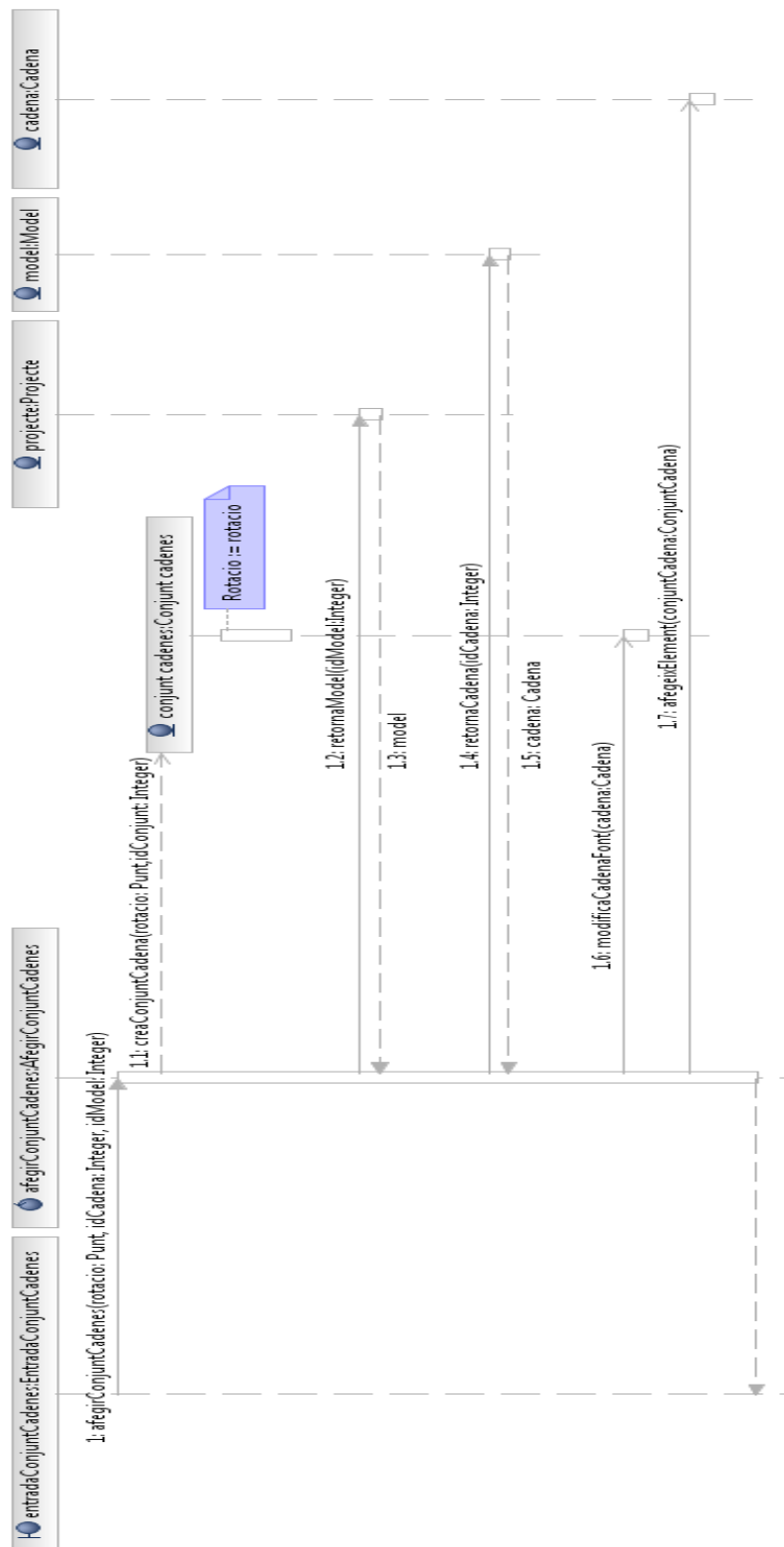


Figura 4.13: Diagrama de seqüència de Afegeir "Conjunt de cadenes"

4.2.3 Cas d'ús 1.3: Afegir cadena al model

VOPC

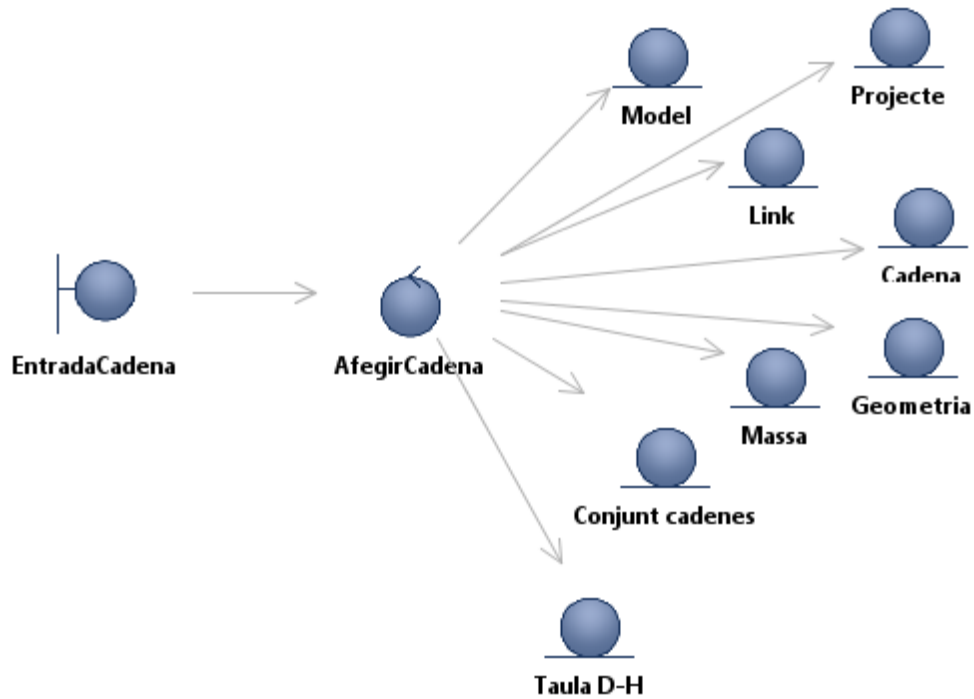


Figura 4.14: VOPC de Afegir cadena al model

Diagrama de seqüència

Afegir una cadena és un cas bastant complex, ja que cal saber a quin element s'afegeix, si directament al model o a un "Conjunt de cadenes". Després cal saber si la cadena comença amb una massa o amb un "link". Si és un "link" quina llargada té i si és una massa si és una massa associada a una articulació o només és un punt per conèixer informació respecte el moviment. Així doncs el que mostra el següent diagrama de seqüència és el comportament que est tindrà referent a una o altre situació.

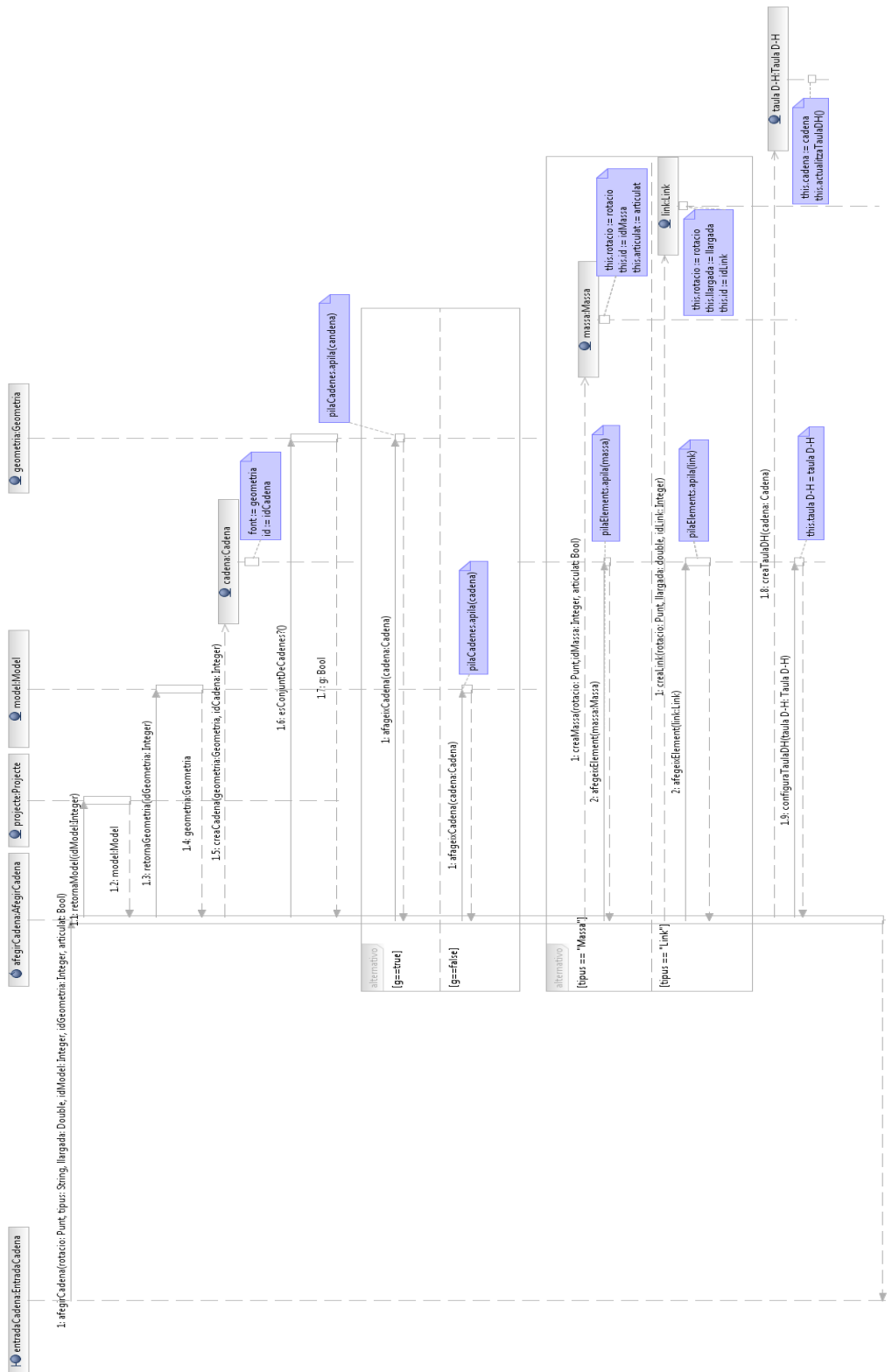


Figura 4.15: Diagrama de seqüència de Afegir cadena al model

4.2.4 Cas d'ús 1.4: Afegir un "Link" al model

VOPC

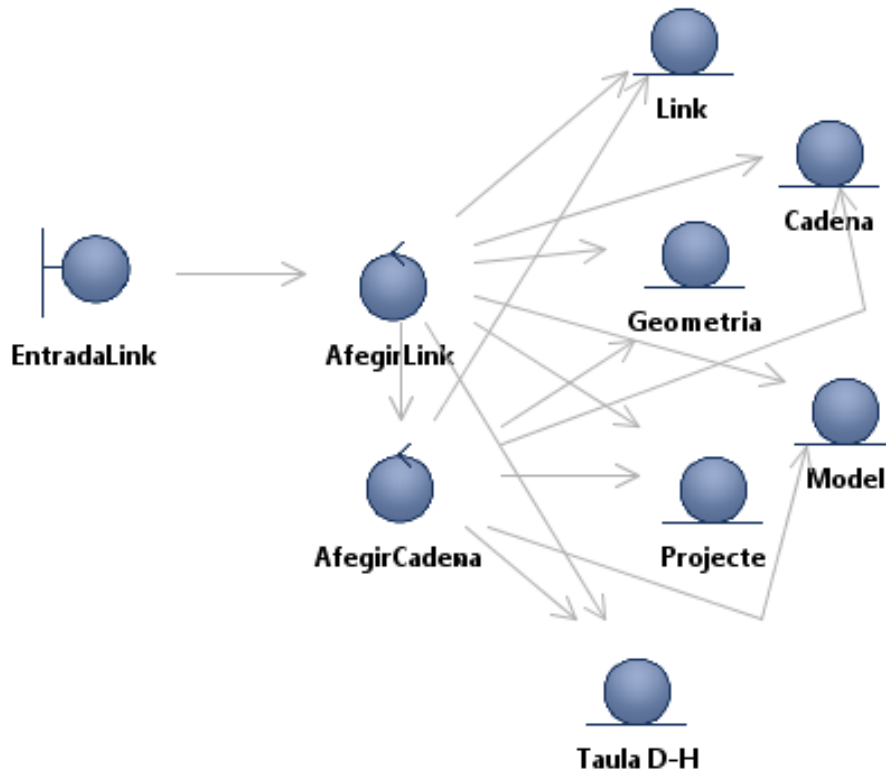


Figura 4.16: VOPC de Afegir un "Link" al model

Diagrama de seqüència

Afegir un "Link" pot ser que es trobi en tres situacions. La primera és que afegeix el "Link" a una cadena, l'altre és quan s'afegeix a un "Conjunt de cadenes" i l'altre és quan s'afegeix al model en sí, mitjançant el centre del model. En els dos últims cassos, cal crear una cadena. La complexitat del cas d'ús ve a la següent figura:

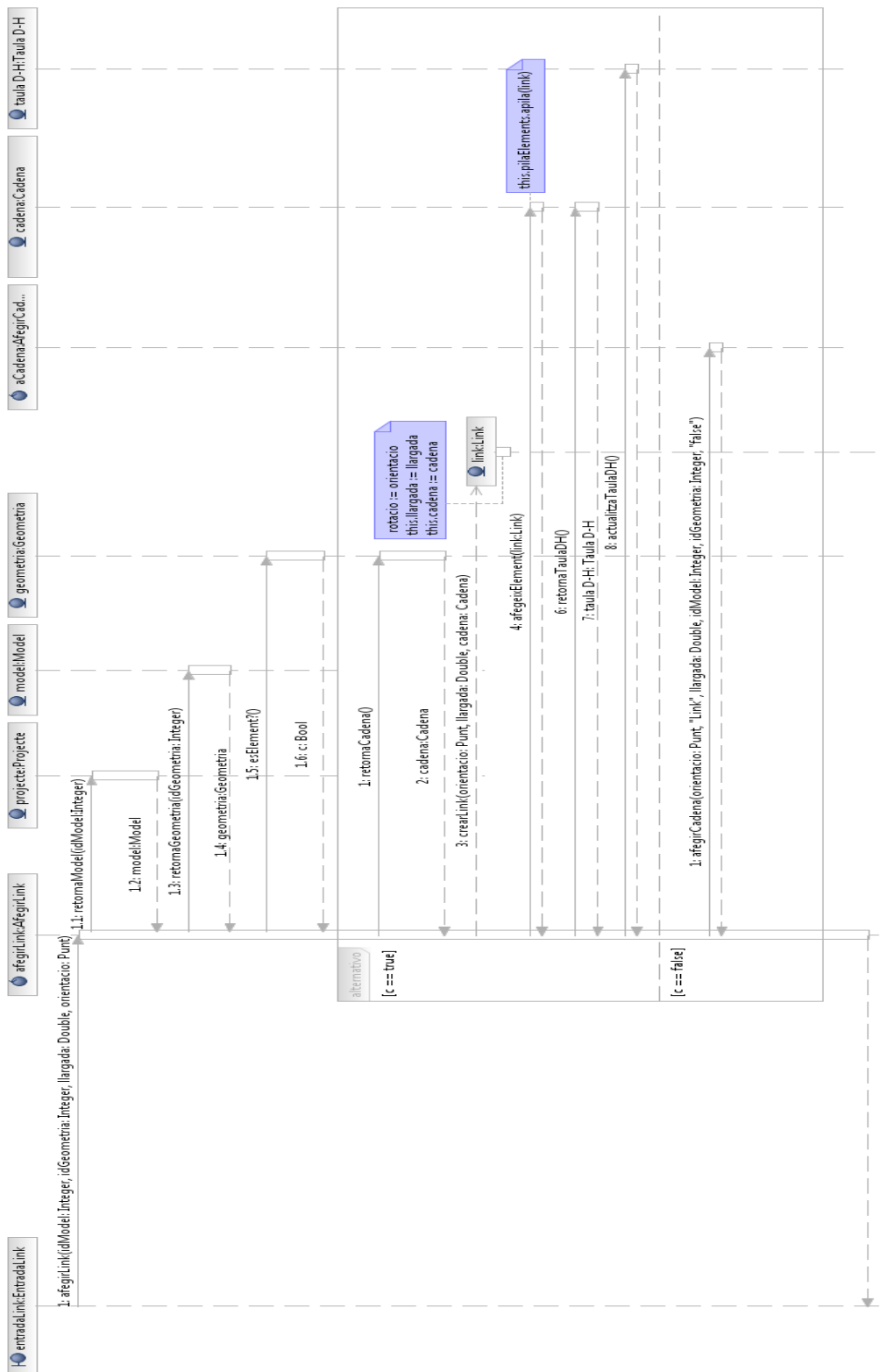


Figura 4.17: Diagrama de seqüència de Afegir un "Link" al model

4.2.5 Cas d'ús 1.5: Afegir una Massa al model

VOPC

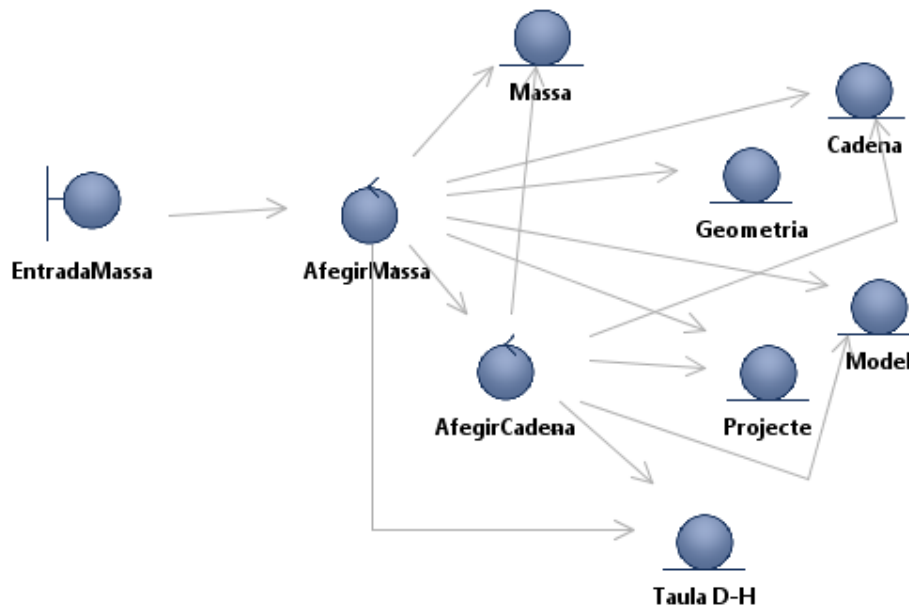


Figura 4.18: VOPC de Afegir una Massa al model

Diagrama de seqüència

És el mateix cas que afegir un "Link", però afegint una massa.

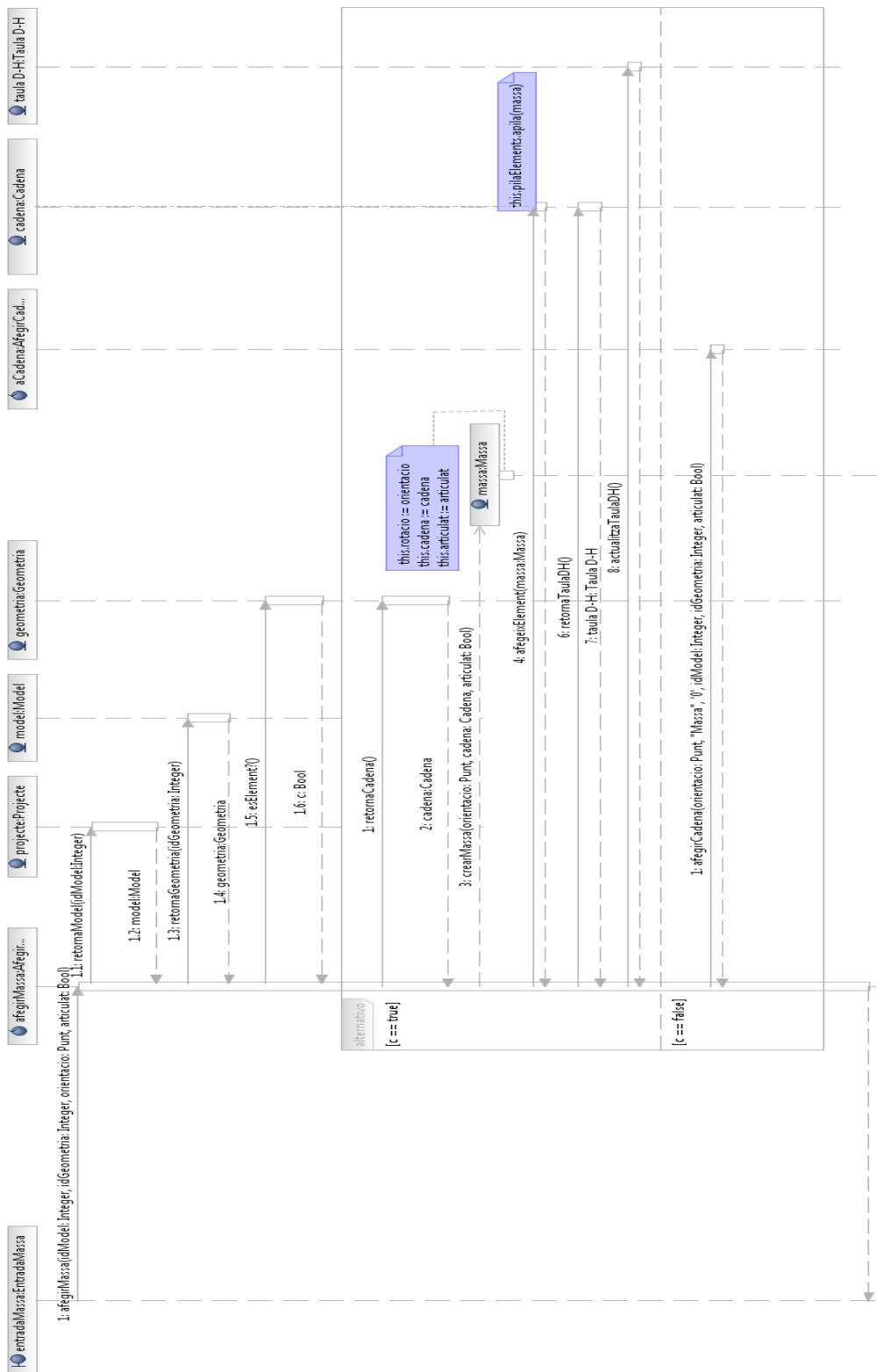


Figura 4.19: Diagrama de seqüència de Afegeix una Massa al model

4.2.6 Cas d'ús 1.6: Afegir un Recolzador al Model

VOPC

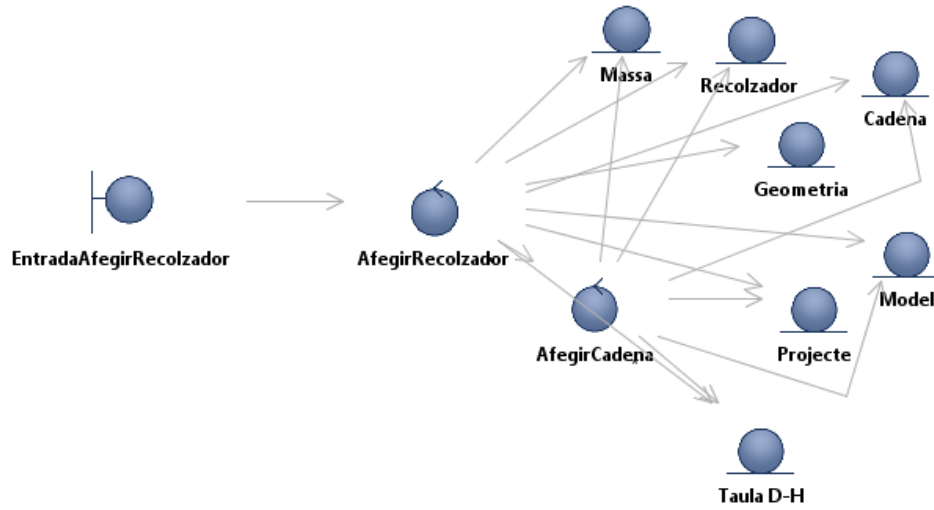


Figura 4.20: VOPC de Afegir un Recolzador al Model

Diagrama de seqüència

Idènticament que els dos casos anteriors, cal veure a quin punt d'ancoratge s'afegeix i crear una cadena o no.

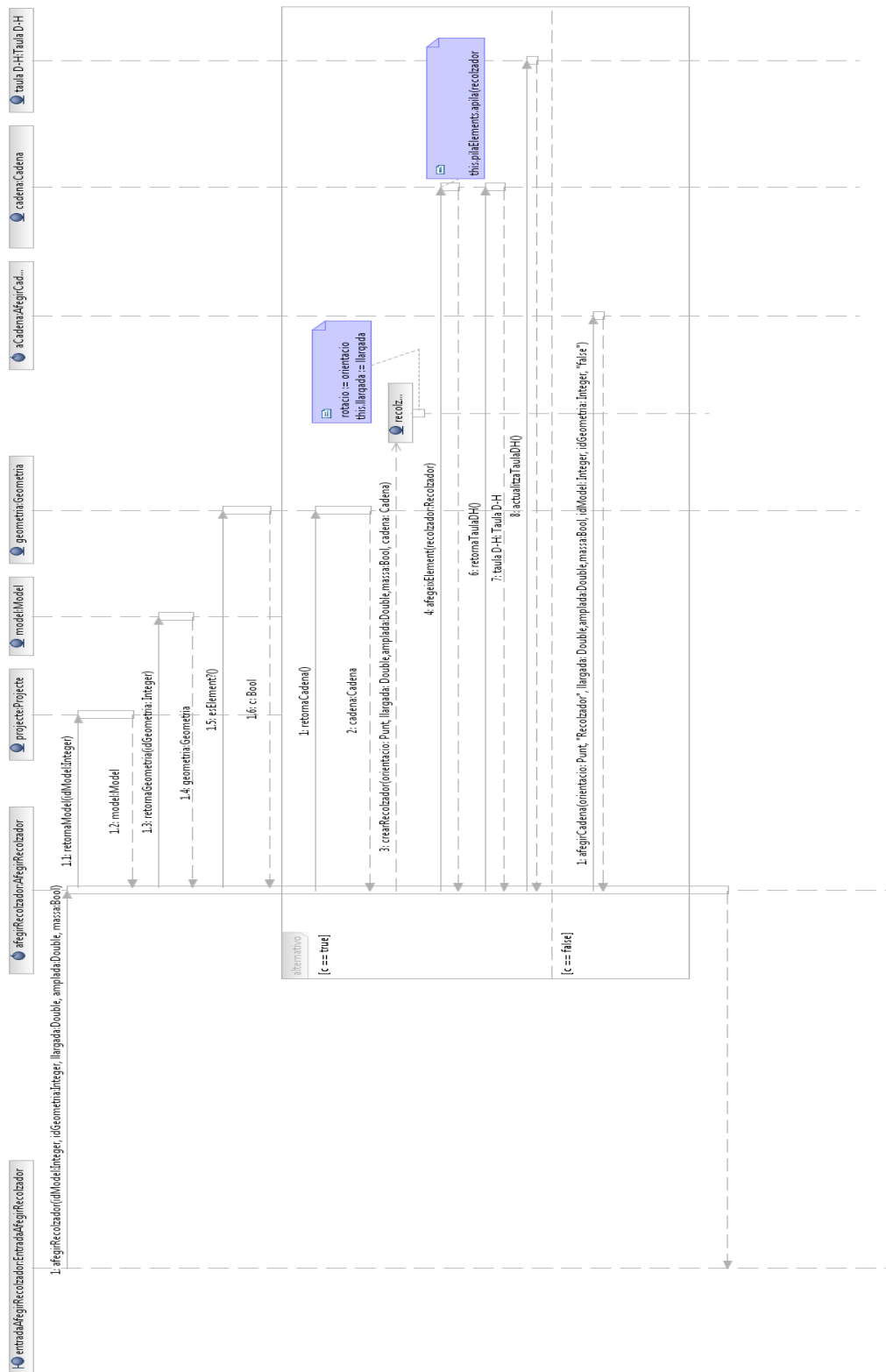


Figura 4.21: Diagrama de seqüència d'Afege un Recolzador al Model

4.2.7 Cas d'ús 2.1: Eliminar model

VOPC

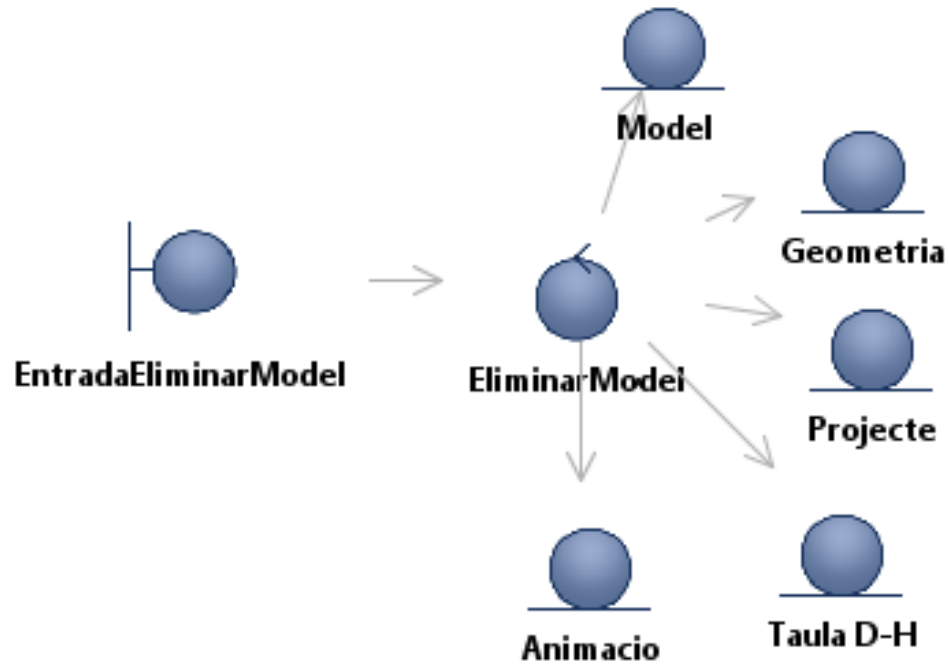


Figura 4.22: VOPC de Eliminar model

Diagrama de seqüència

Per eliminar un model, la principal informació necessària és quin model es vol eliminar. Una vegada es té aquesta informació, cal eliminar tots els elements (cadenes, masses, "links" i "Conjunt de cadenes") que continguin el model.

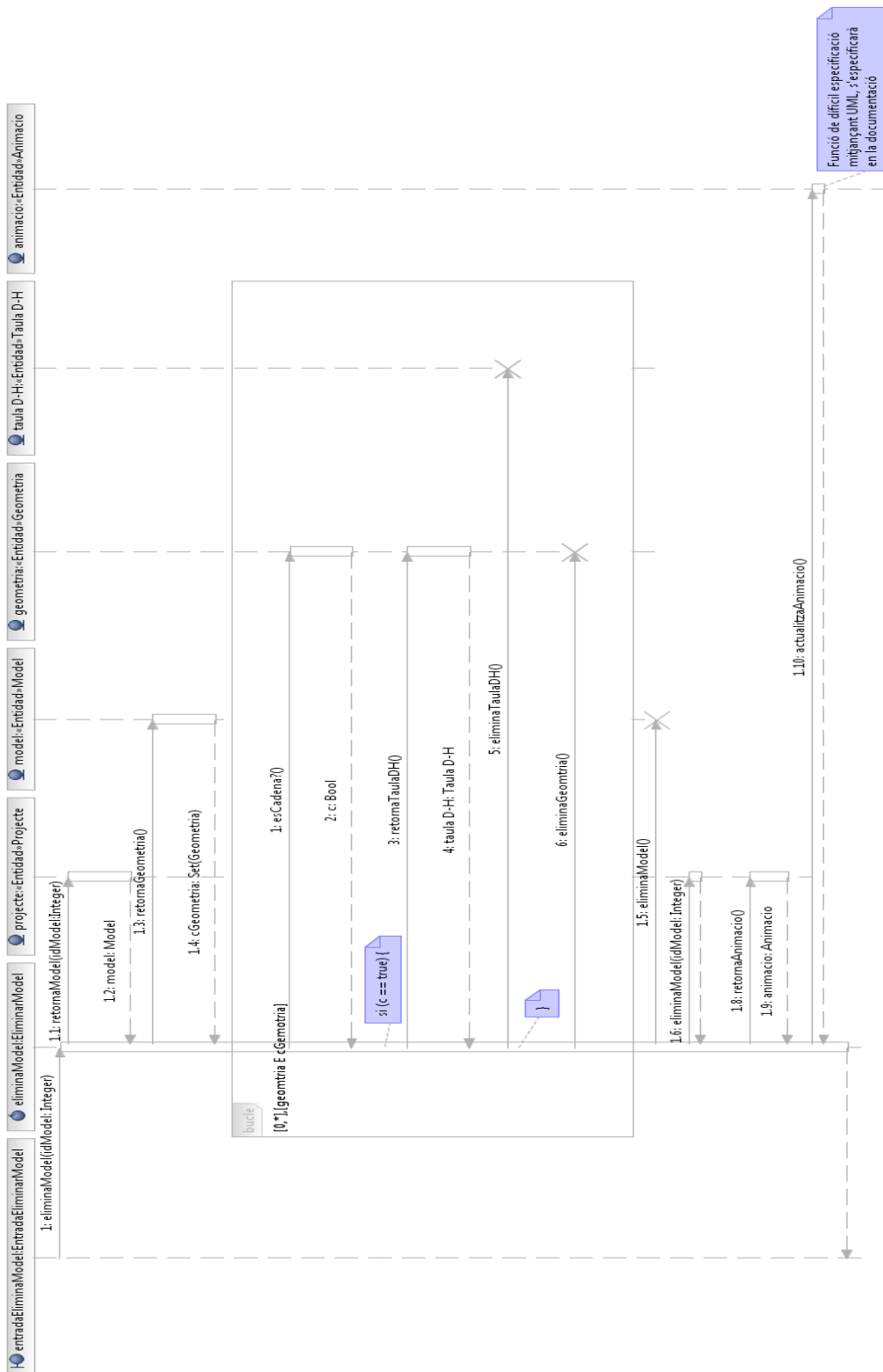


Figura 4.23: Diagrama de seqüència de Eliminar model

4.2.8 Cas d'ús 2.2: Moure model

VOPC

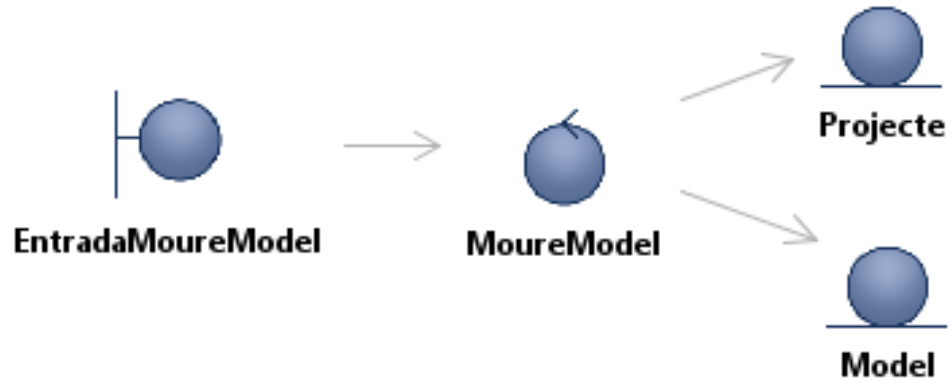


Figura 4.24: VOPC de Moure model

Diagrama de seqüència

En aquest cas, el sistema ha de conèixer el model que es vol moure i la translació que s'aplicarà al model. Una vegada és té aquesta informació el sistema ho registrarà.

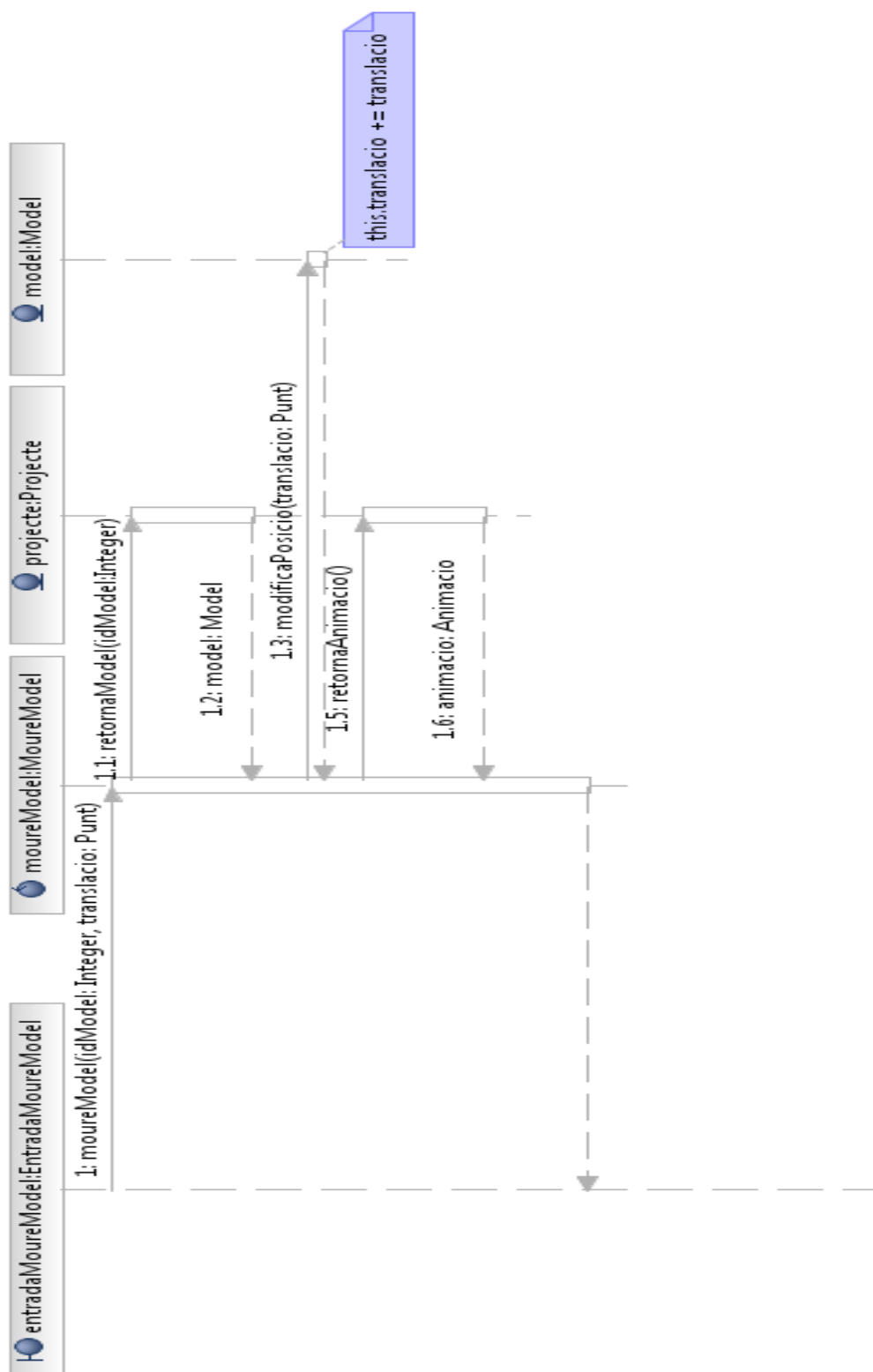


Figura 4.25: Diagrama de seqüència de Moure model

4.2.9 Cas d'ús 2.3: Rotar model

VOPC

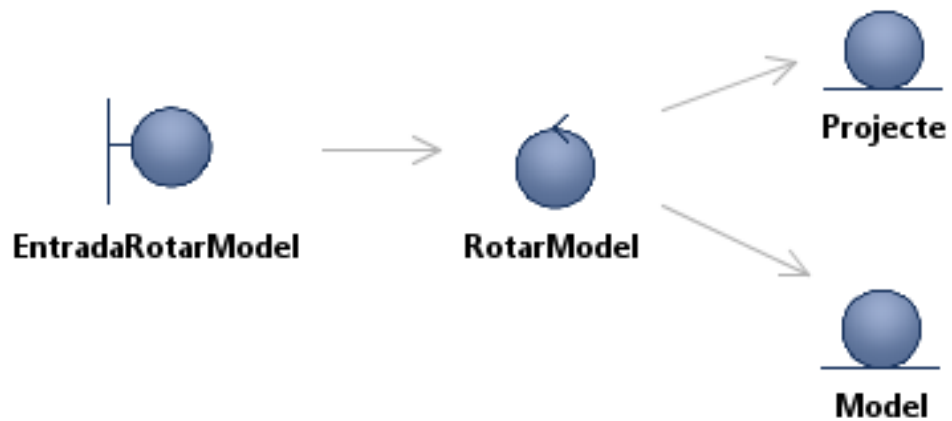


Figura 4.26: VOPC de Rotar model

Diagrama de seqüència

És el mateix cas que el cas d'ús de moure un objecte, però en aquest cas el que es modifica no és la posició sinó l'orientació del model.

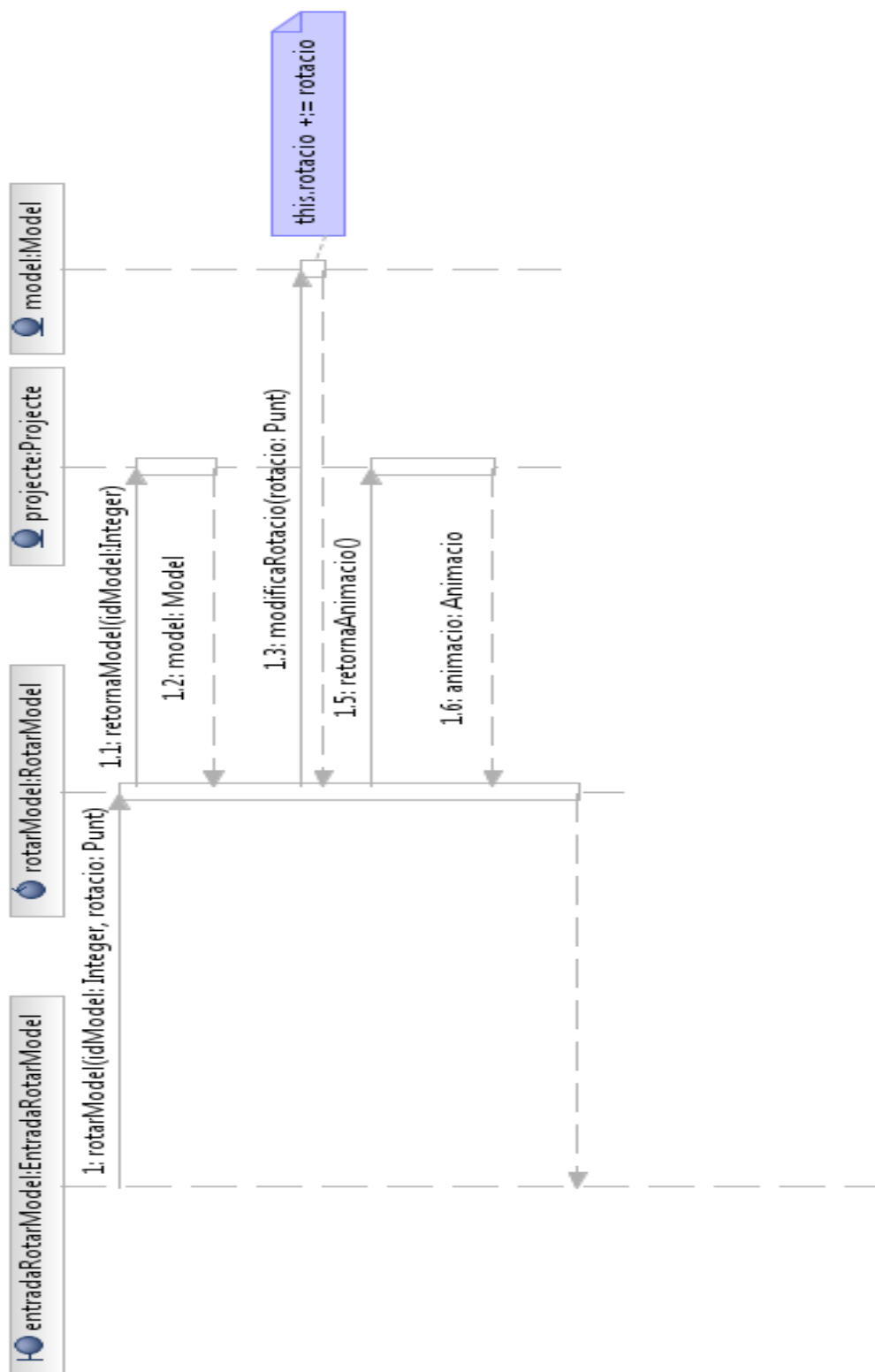


Figura 4.27: Diagrama de seqüència de Rotar model

4.2.10 Cas d'ús 2.4: Eliminar cadena

VOPC

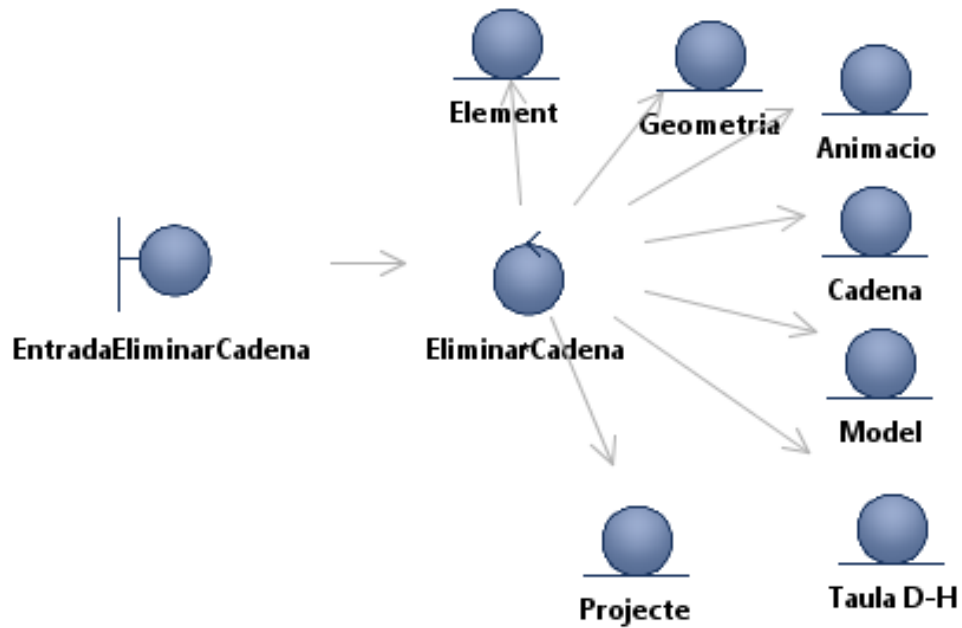


Figura 4.28: VOPC de Eliminar cadena

Diagrama de seqüència

El fet d'eliminar una cadena, comporta que tot el que penjava d'ella queda unit a la part d'on sortia la cadena. Aleshores, cal fer consistent aquesta informació, com també eliminar la taula D-H d'aquesta cadena. Cal actualitzar l'animació al final per si es pot salvar algun punt de contacte o realment cal eliminar-lo, ja que l'estructura del model s'ha modificat. Al següent diagrama de seqüència es pot veure què és el que fa exactament aquest cas d'ús:

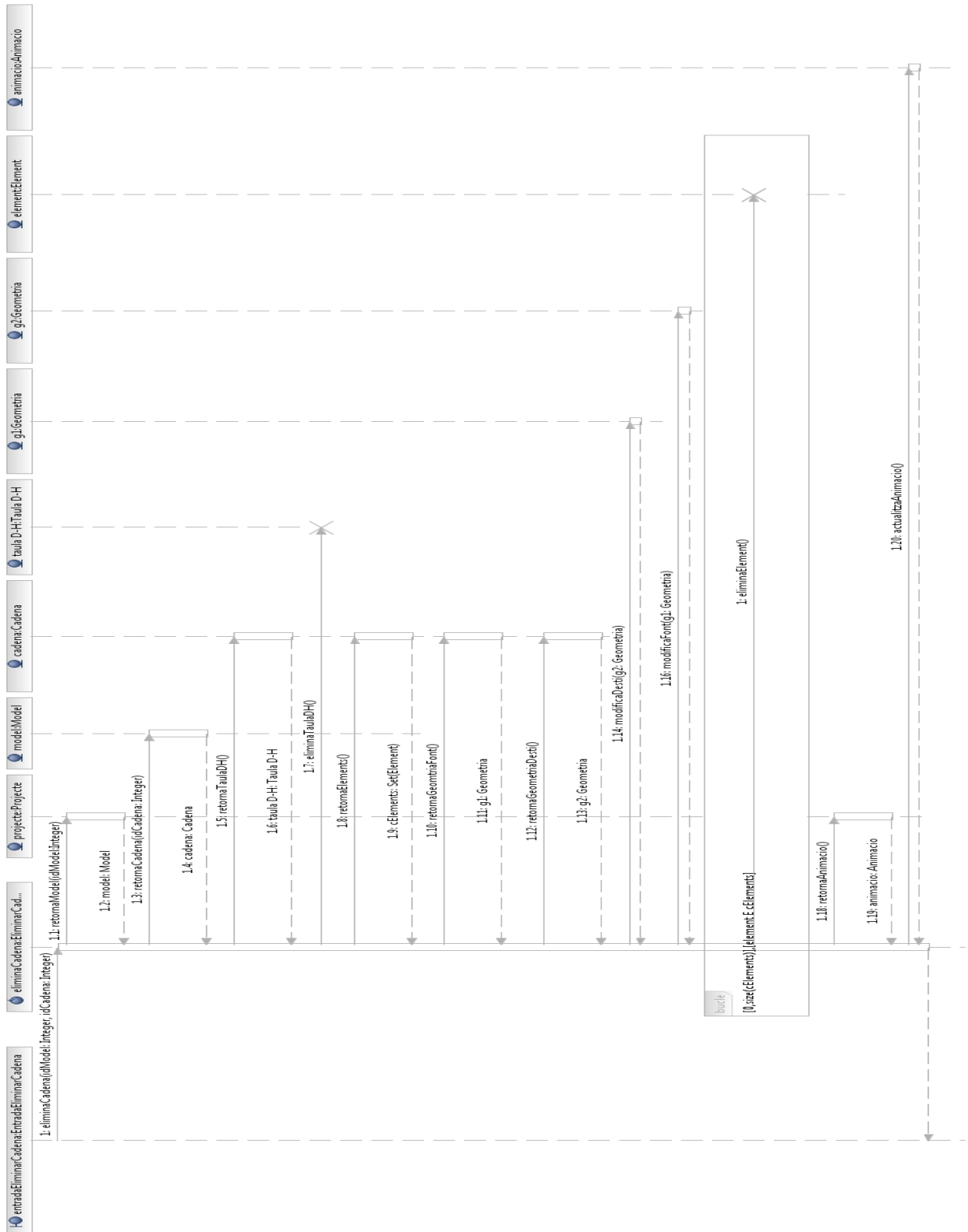


Figura 4.29: Diagrama de seqüència de Eliminar cadena

4.2.11 Cas d'ús 2.5: Moure cadena

VOPC

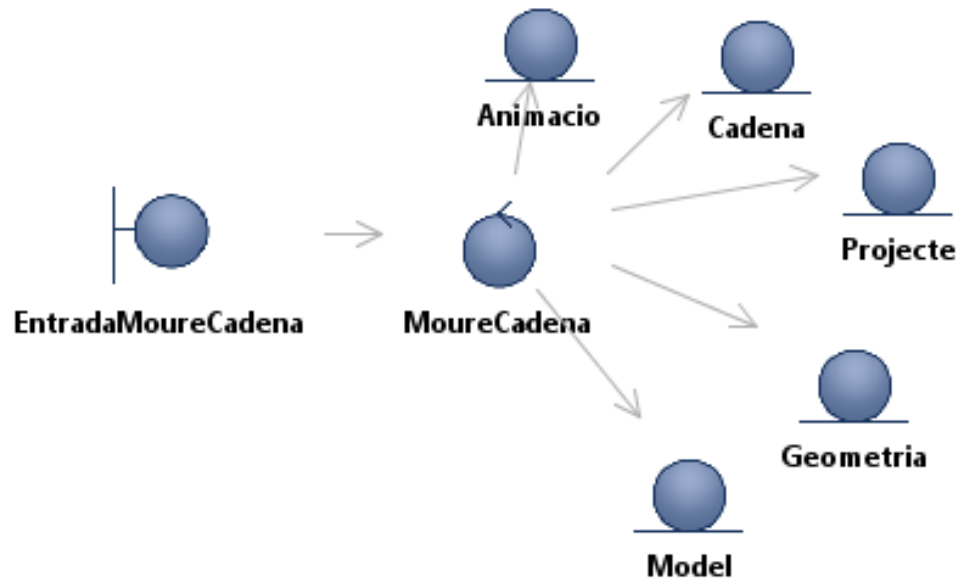


Figura 4.30: VOPC de Moure cadena

Diagrama de seqüència

El que comporta aquest cas d'ús és un canvi de la font d'una cadena, de l'element del que surt una cadena. En el moment en que es mou una cadena es mou tot el que estava lligat a ella. Cal actualitzar l'animació i re-calcular noves posicions pels punts de contacte si s'escau.

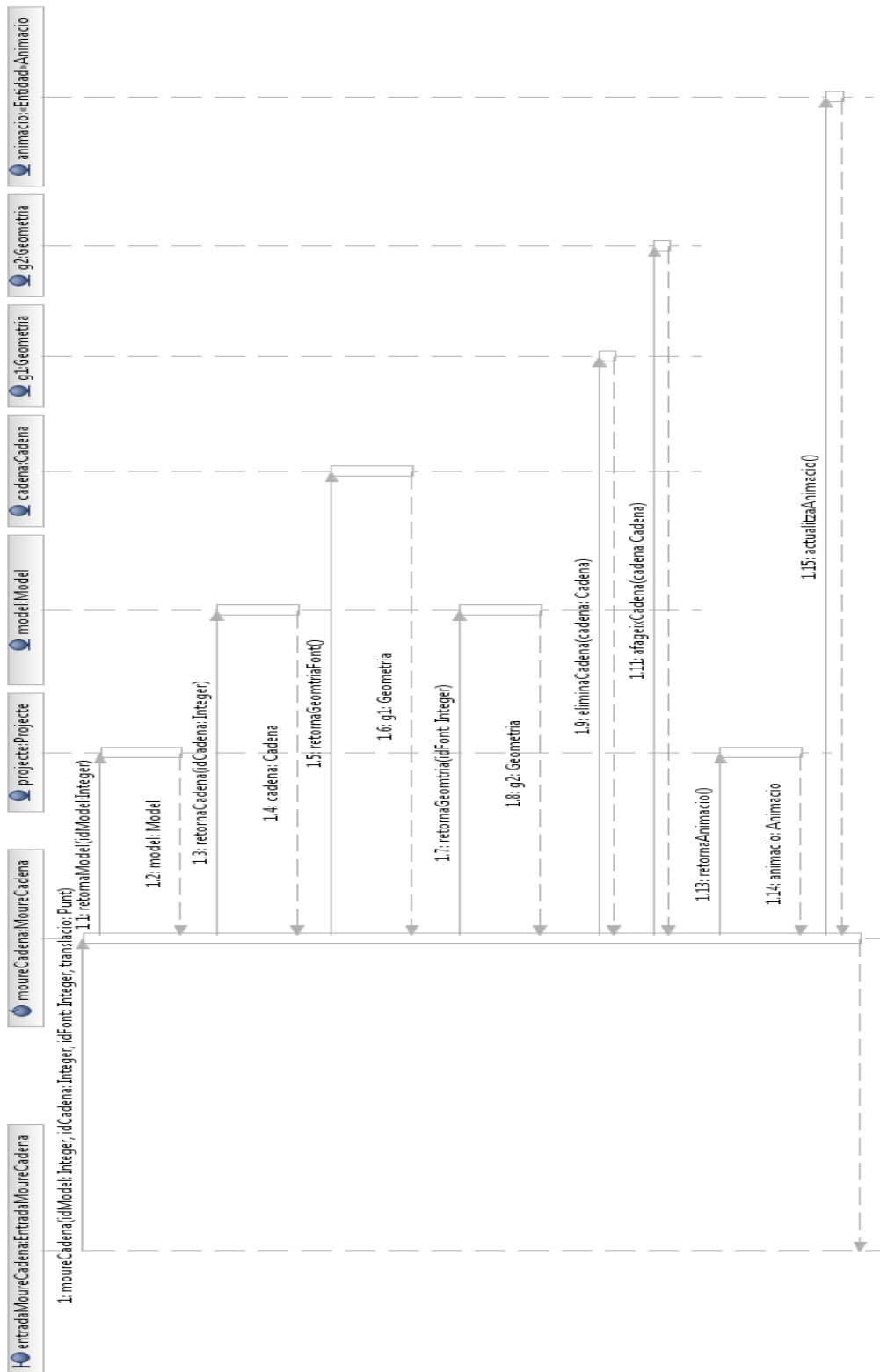


Figura 4.31: Diagrama de seqüència de Moure cadena

4.2.12 Cas d'ús 2.6: Rotar cadena

VOPC

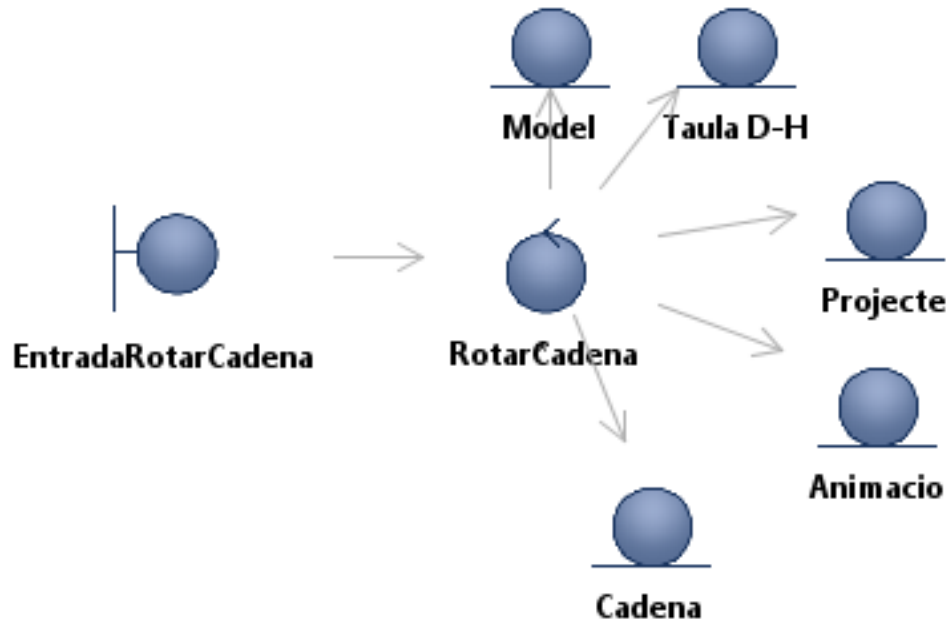


Figura 4.32: VOPC de Rotar cadena

Diagrama de seqüència

Rotar una cadena té el mateix efecte que tot el que bé lligat al final d'ella. Una vegada s'ha modificat, el que cal és actualitzar la taula D-H d'aquella cadena. El procés es veu reflectit en el següent cas d'ús:

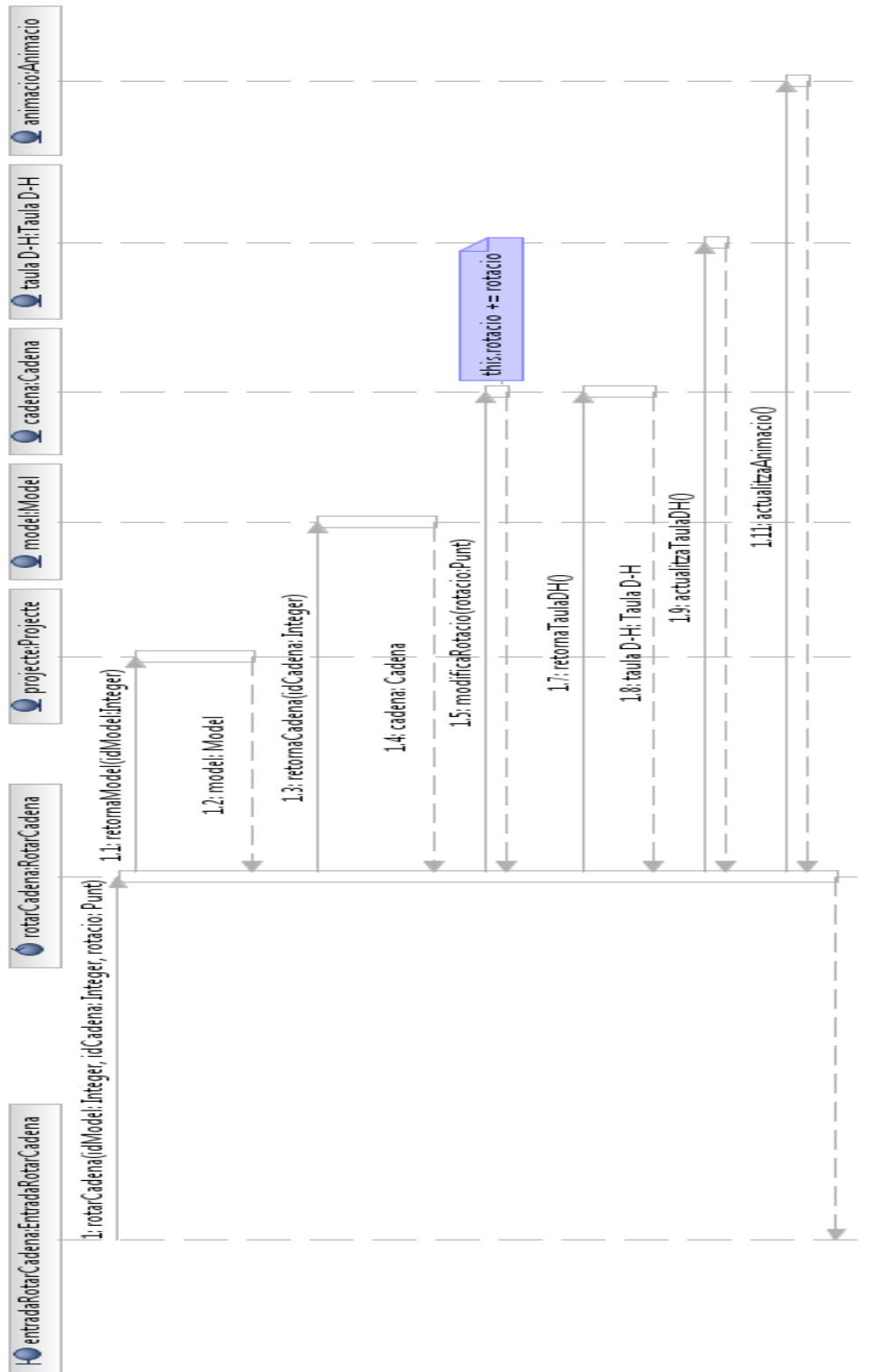


Figura 4.33: Diagrama de seqüència de Rotar cadena

4.2.13 Cas d'ús 2.7: Eliminar "Conjunt de cadenes"

VOPC

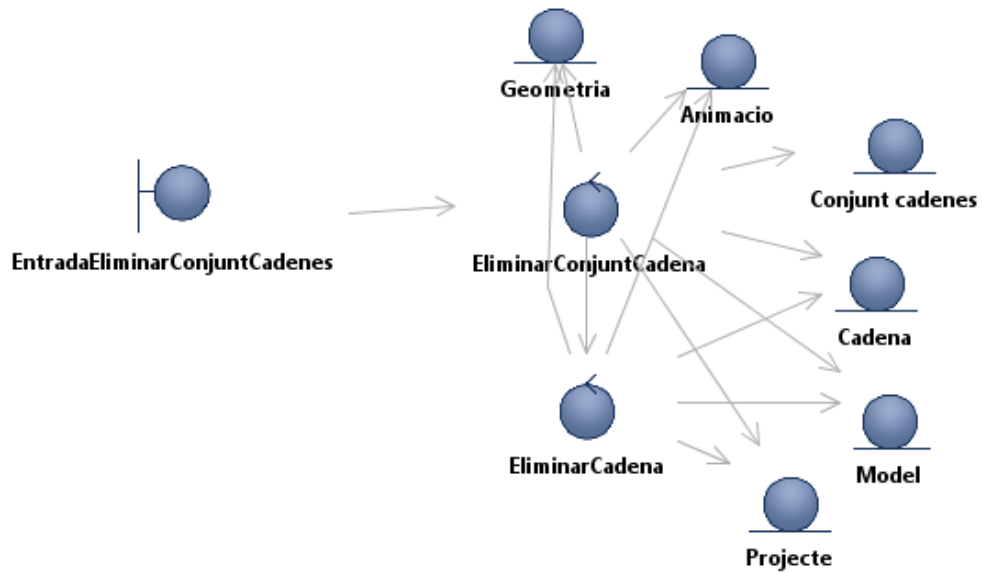


Figura 4.34: VOPC de Eliminar "Conjunt de cadenes"

Diagrama de seqüència

En el moment en que s'elimina un "Conjunt de cadenes" cal eliminar totes les cadenes i les seves respectives taules D-H, ja que, no hi hauria cap manera d'unir unes quantes cadenes amb una cadena. A continuació es presenta el conjunt d'operacions que s'han de realitzar per dur a terme el cas d'ús:

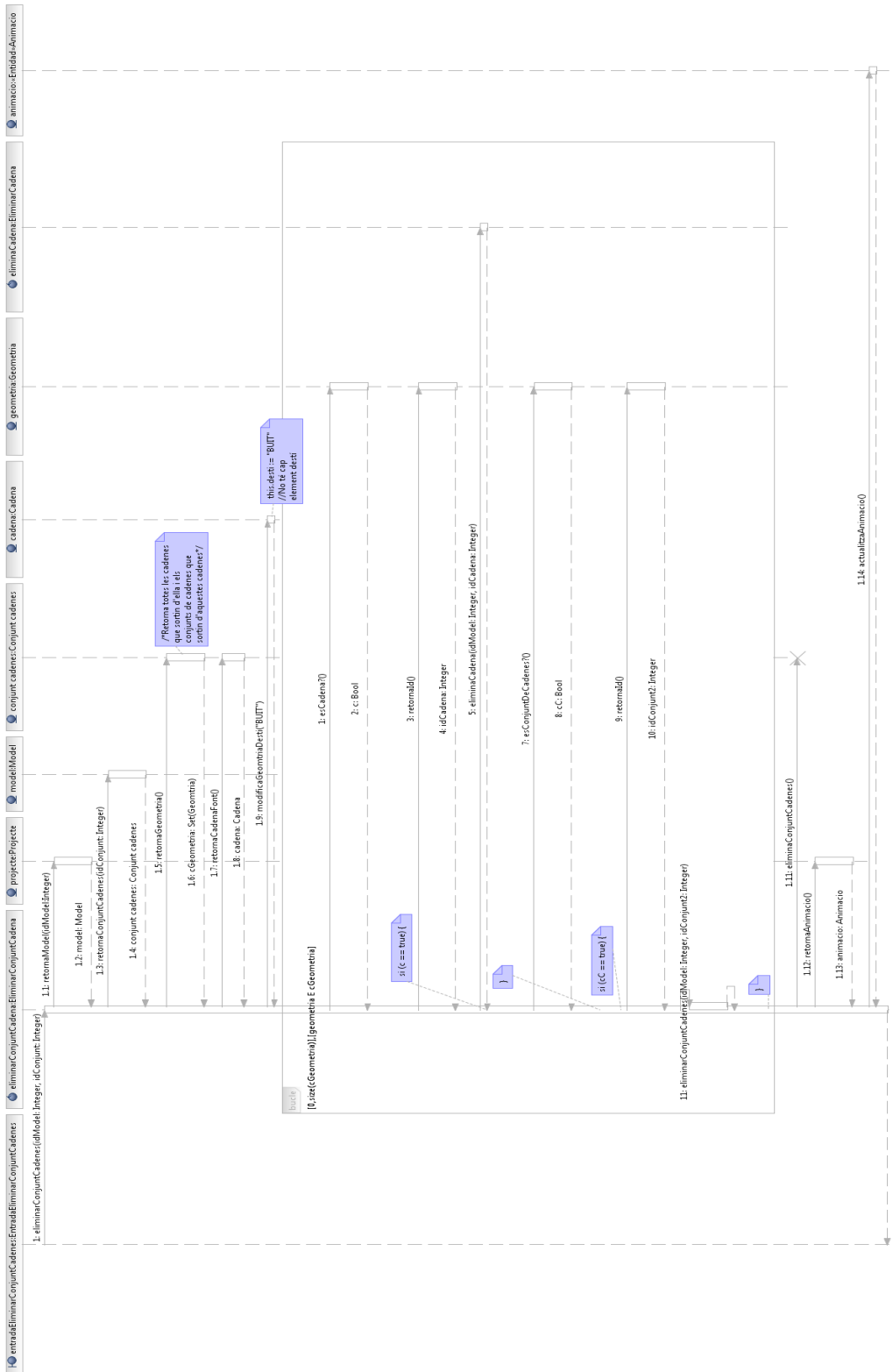


Figura 4.35: Diagrama de seqüència de Eliminar "Conjunt de cadenes"

4.2.14 Cas d'ús 2.8: Eliminar "Link"

VOPC

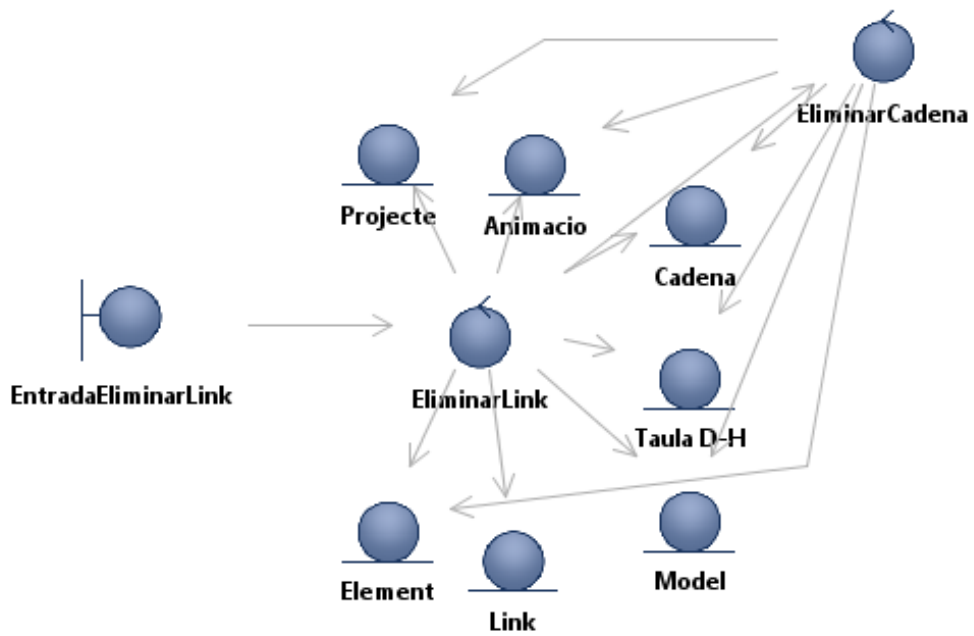


Figura 4.36: VOPC de Eliminar "Link"

Diagrama de seqüència

L'única complicació que comporta eliminar un "Link" és que cal unir els dos extrems que el "Link" unia i actualitzar la taula D-H. Per tant, el flux és el següent:

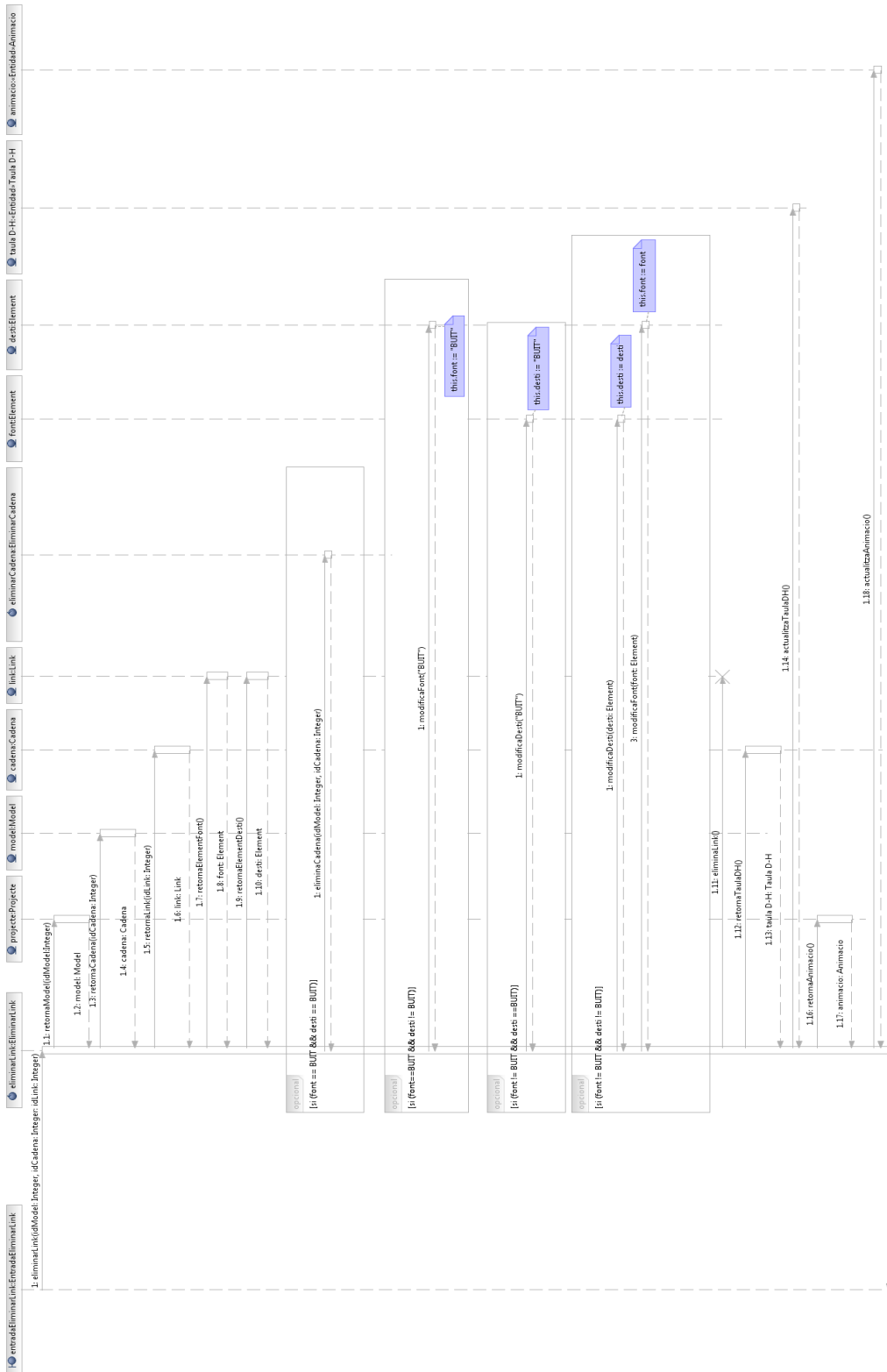


Figura 4.37: Diagrama de seqüència de Eliminar "Link"

4.2.15 Cas d'ús 2.9: Rotar "Link"

VOPC

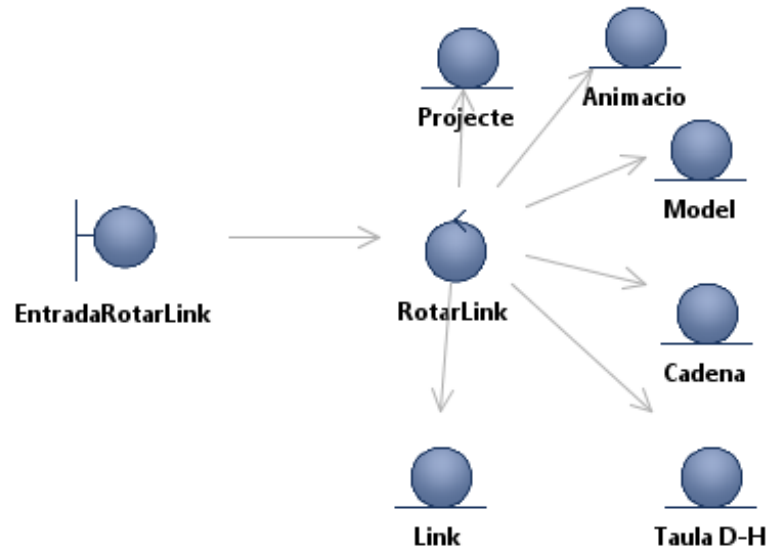


Figura 4.38: VOPC de Rotar "Link"

Diagrama de seqüència

Modificar la rotació d'un "Link" vol dir modificar els angles de rotació que porten del sistema de referència de l'element anterior al "Link" amb el sistema de referència que descriu el "Link". Caldrà doncs poder modificar aquests angles i després actualitzar la taula D-H:

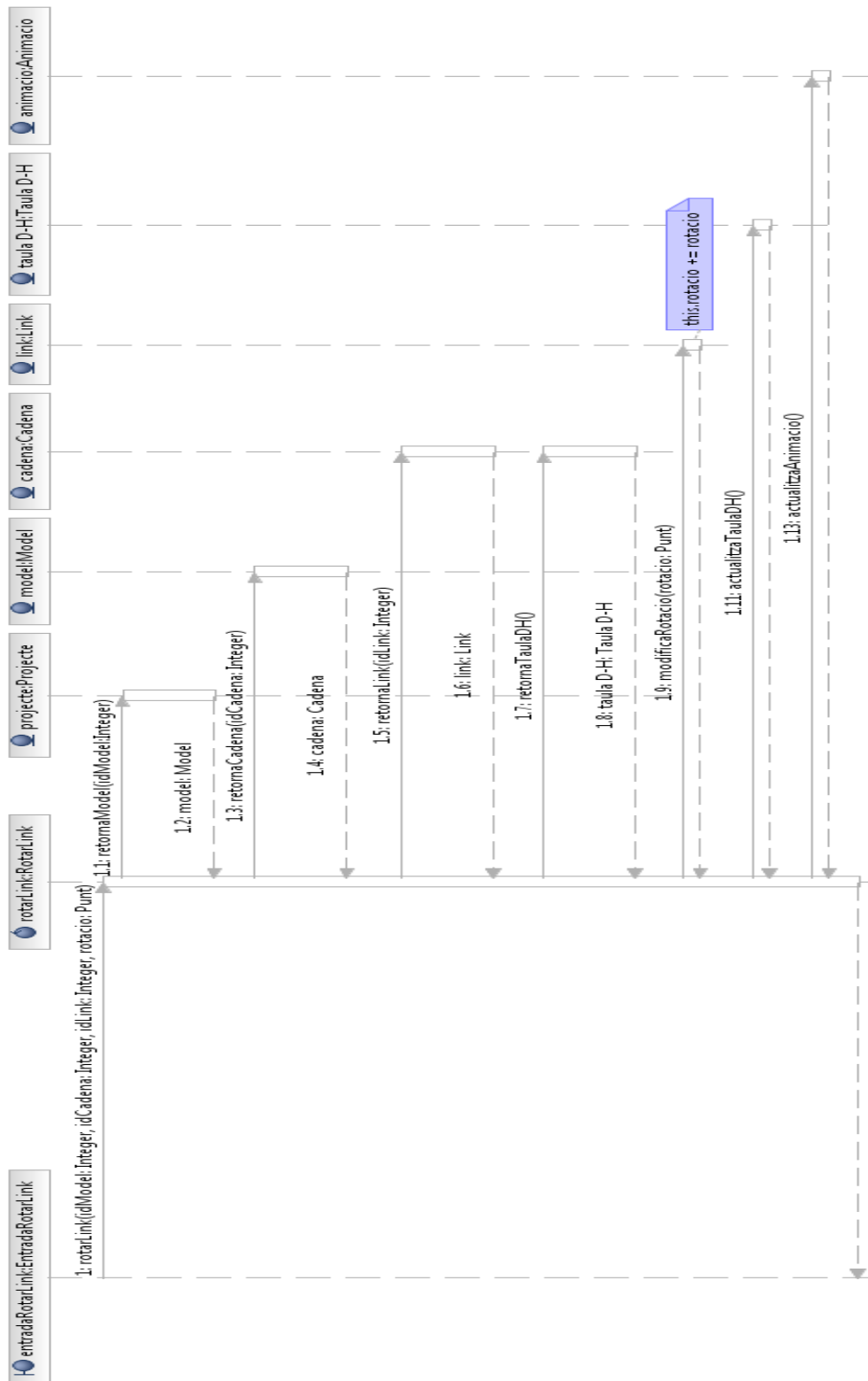


Figura 4.39: Diagrama de seqüència de Rotar "Link"

4.2.16 Cas d'ús 2.10: Modificar la llargada del "Link"

VOPC

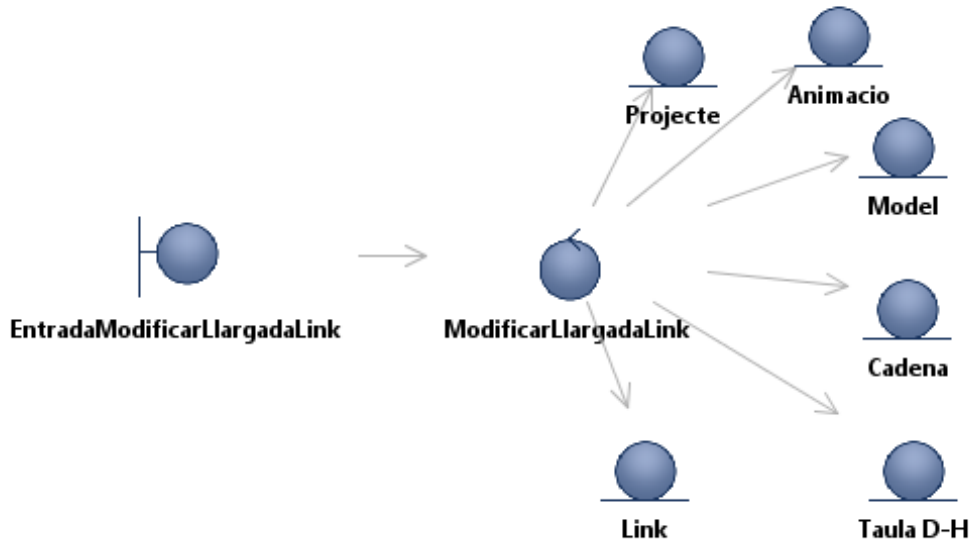


Figura 4.40: VOPC de Modificar la llargada del "Link"

Diagrama de seqüència

Bé a ser el mateix cas que el cas d'ús anterior, però el que es modifica en aquest cas és la translació que porta d'un sistema de referència a l'altre:

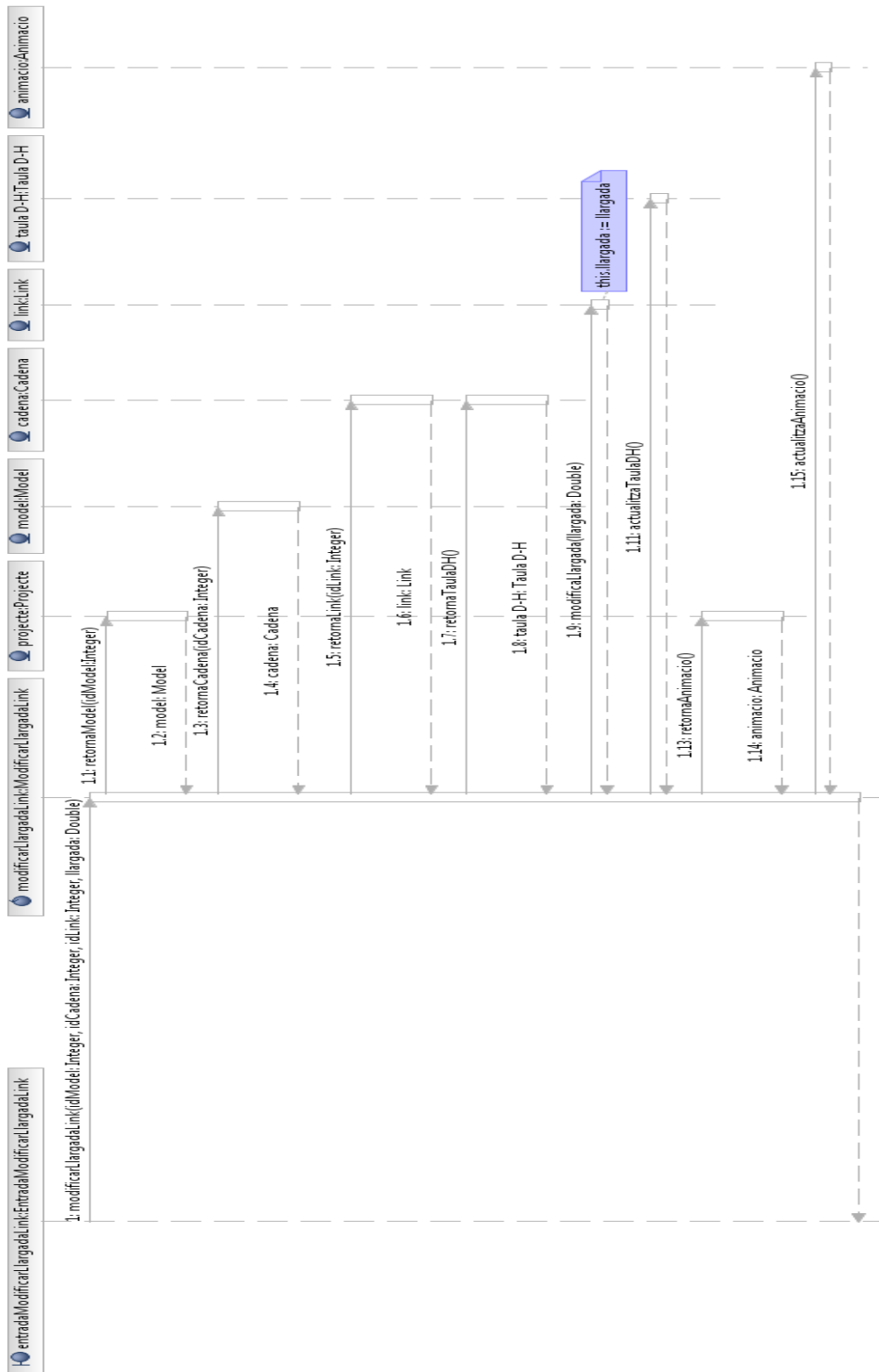


Figura 4.41: Diagrama de seqüència de Modificar la llargada del "Link"

4.2.17 Cas d'ús 2.11: Eliminar Massa

VOPC

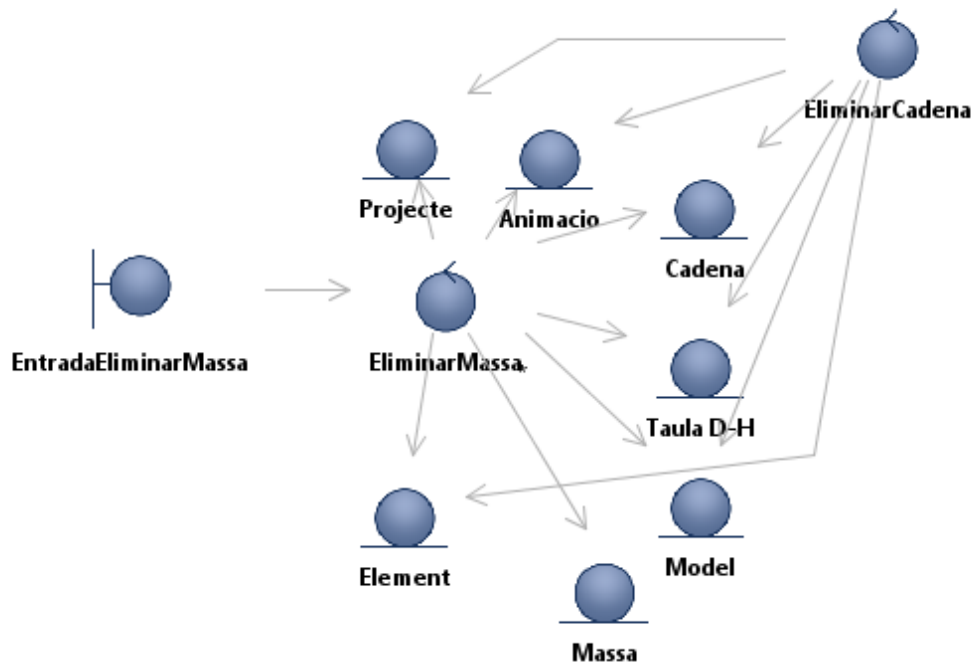


Figura 4.42: VOPC de Eliminar Massa

Diagrama de seqüència

És el mateix cas que en el cas d'ús d'eliminar un "Link". Així doncs el diagrama de seqüència és el següent:

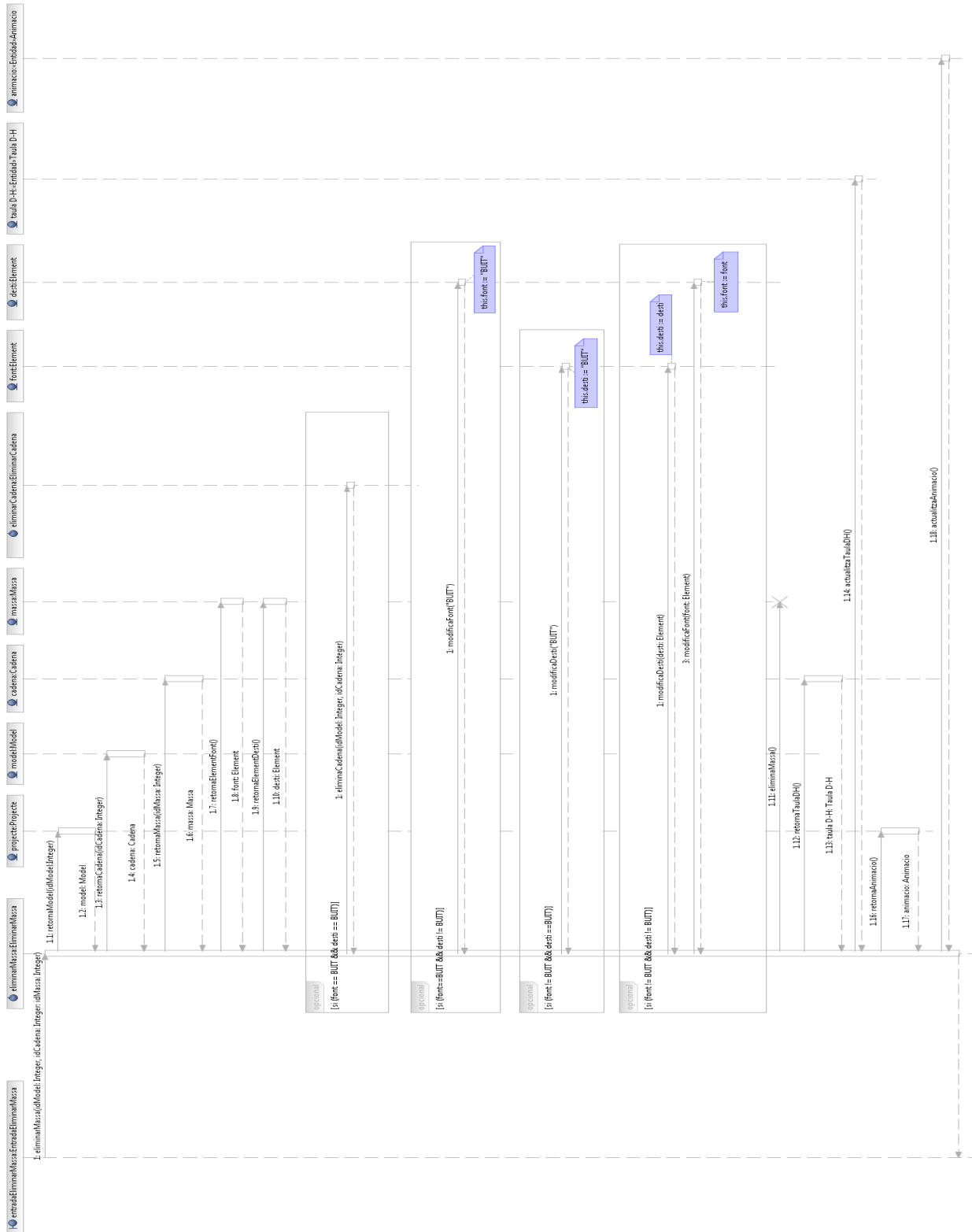


Figura 4.43: Diagrama de seqüència de Eliminar Massa

4.2.18 Cas d'ús 2.12: Rotar Massa

VOPC

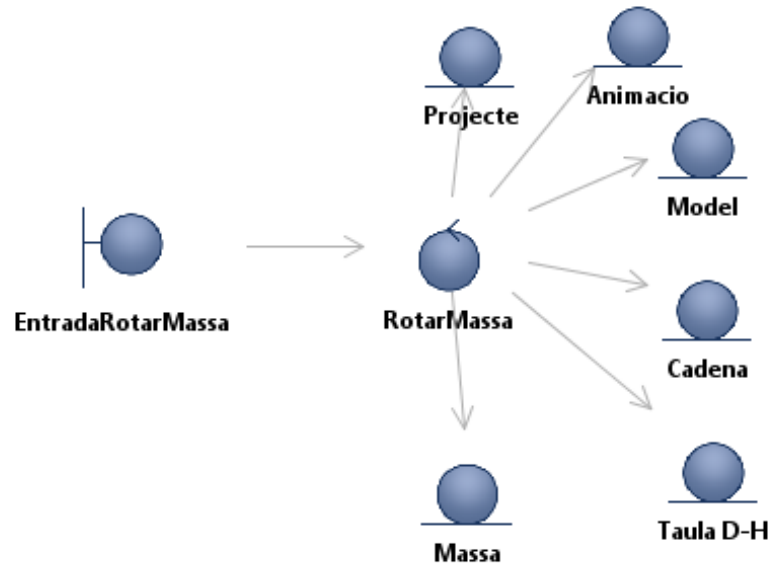


Figura 4.44: VOPC de Rotar Massa

Diagrama de seqüència

Segueix essent el mateix cas explicat que rotar un "Link", el procediment és el mateix, l'únic que en aquest cas és una massa. Com que s'ha modificat l'estructura del model, cal actualitzar l'animació. El procés es mostra en la següent figura:

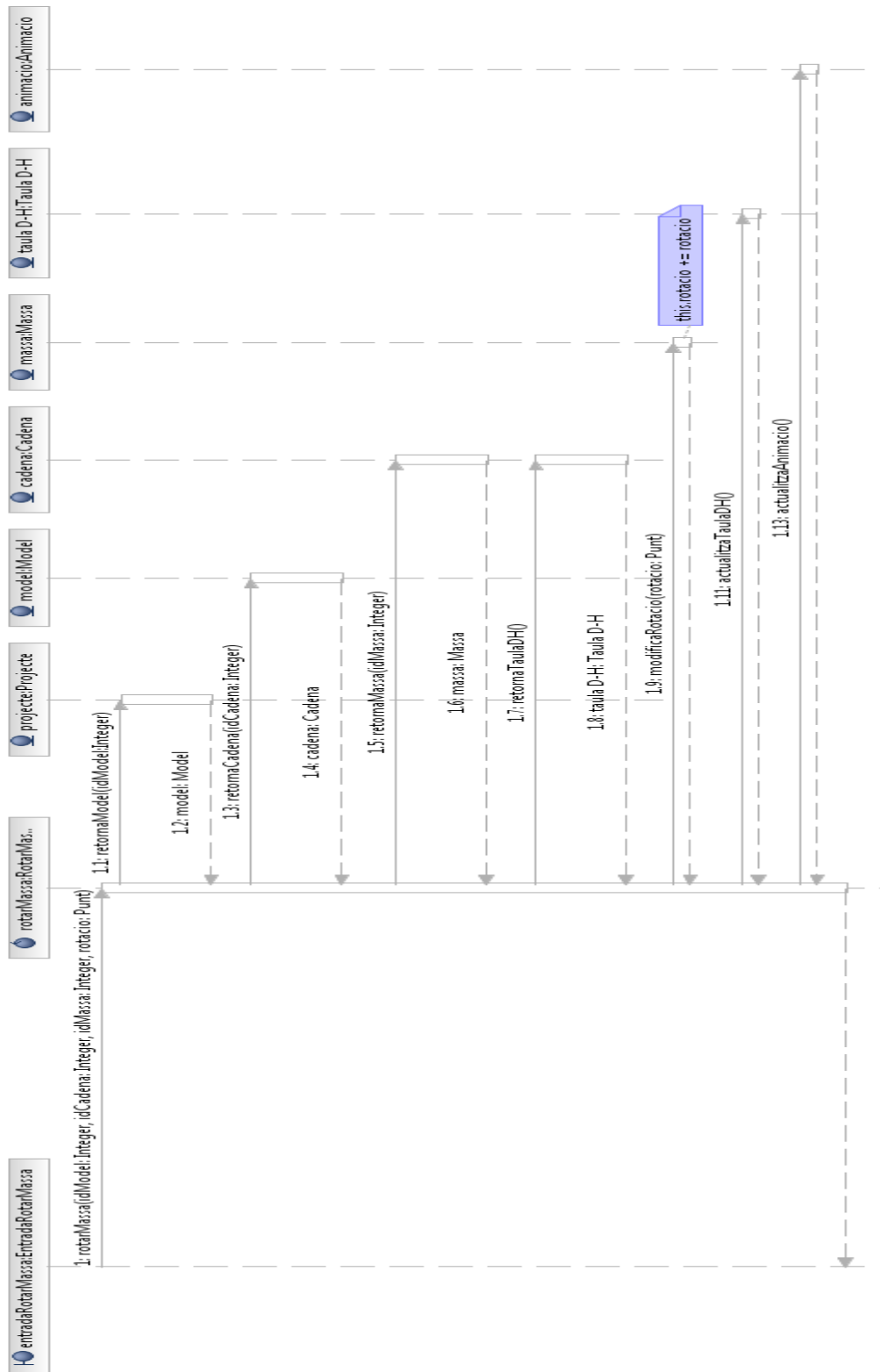


Figura 4.45: Diagrama de seqüència de Rotar Massa

4.2.19 Cas d'ús 2.13: Rotar un Recolzador

VOPC

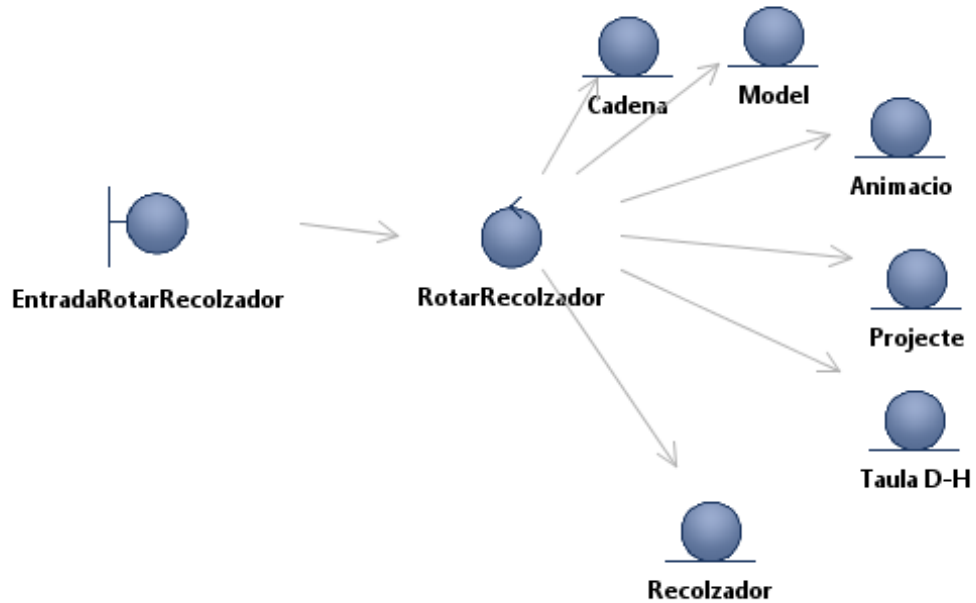


Figura 4.46: VOPC de Rotar un Recolzador

Diagrama de seqüència

Com a element d'un model, el Recolzador ha de poder-ser rotar respecte l'element anterior per buscar diversitat de models i aconseguir moviments de diferents tipus. La col·locació de peus i mans al món real és tant gran que una simple modificació de com està col·locat aquest element pot fer assolir un moviment o l'altre. El procediment d'actuació és el mateix respecte la rotació d'una massa o d'un "Link".

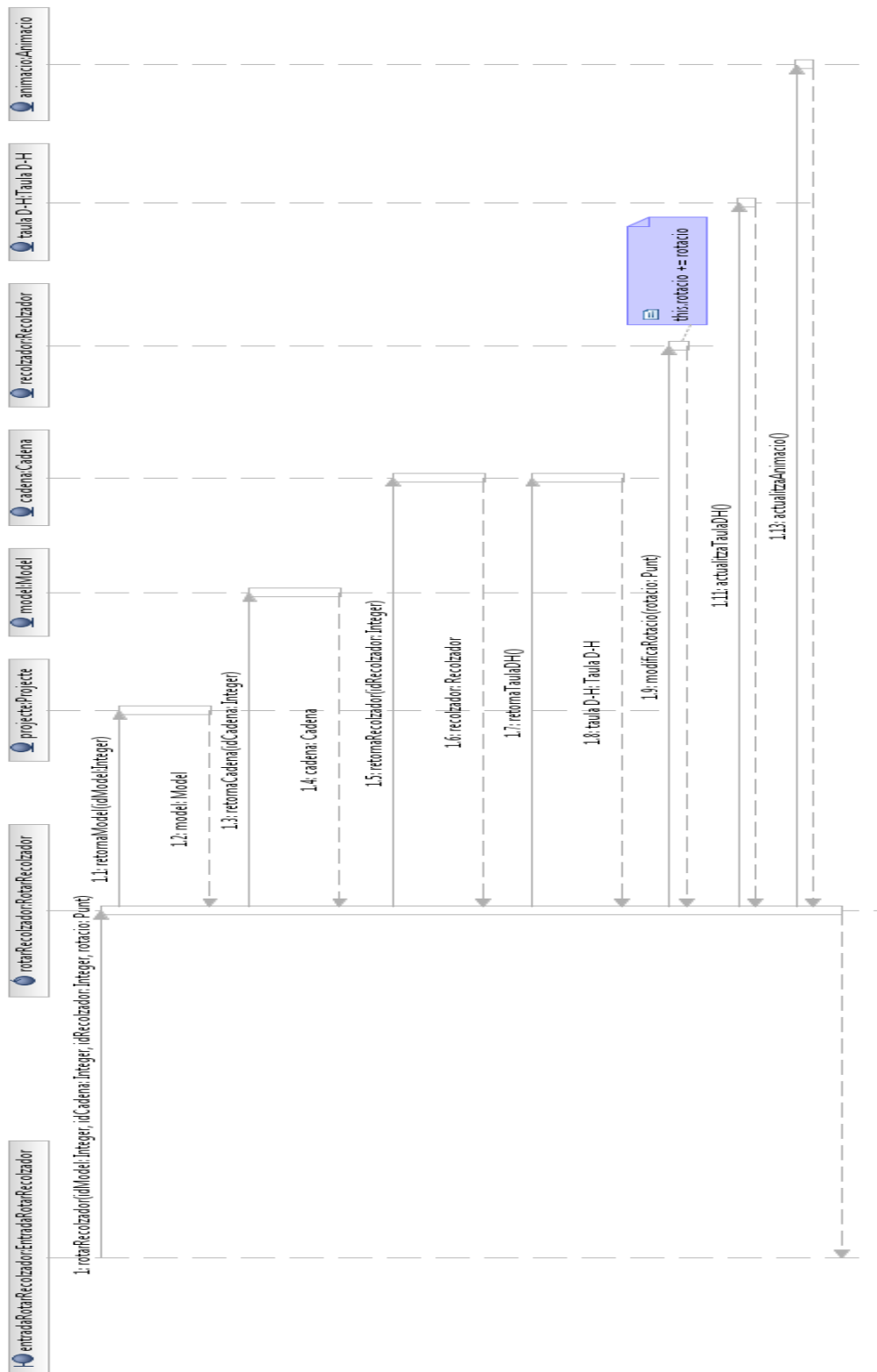


Figura 4.47: Diagrama de seqüència de Rotar un Recolzador

4.2.20 Cas d'ús 2.14: Eliminar un Recolzador

VOPC

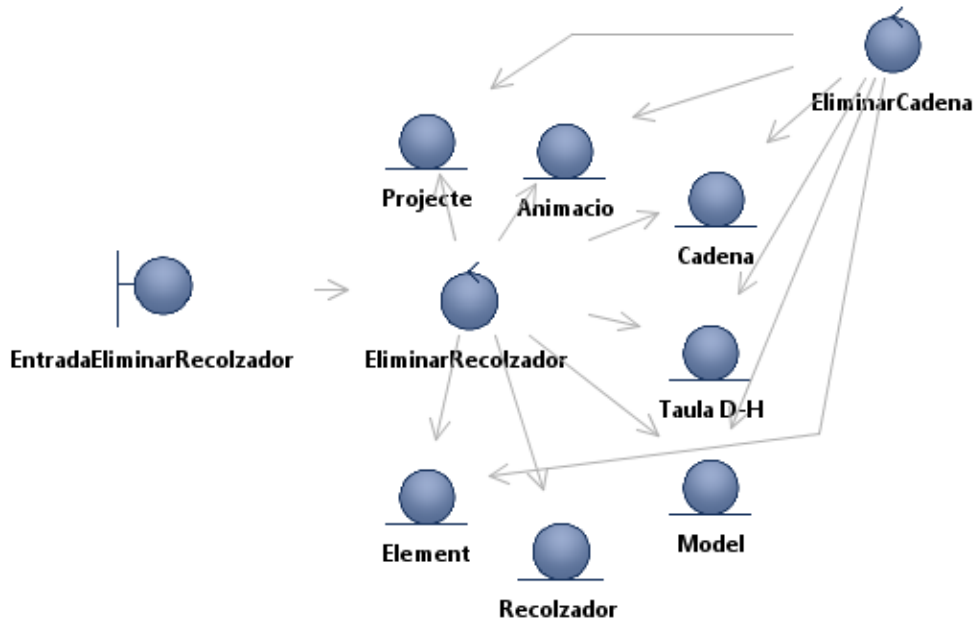


Figura 4.48: VOPC de Eliminar un Recolzador

Diagrama de seqüència

No deixa de ser el mateix que el cas 2.8 o 2.11, cal eliminar el Recolzador d'una cadena i si era l'únic element d'una cadena, aquesta cal ser eliminada. Una vegada fet això, cal actualitzar l'animació per tenir concordança amb el que s'ha modificat.

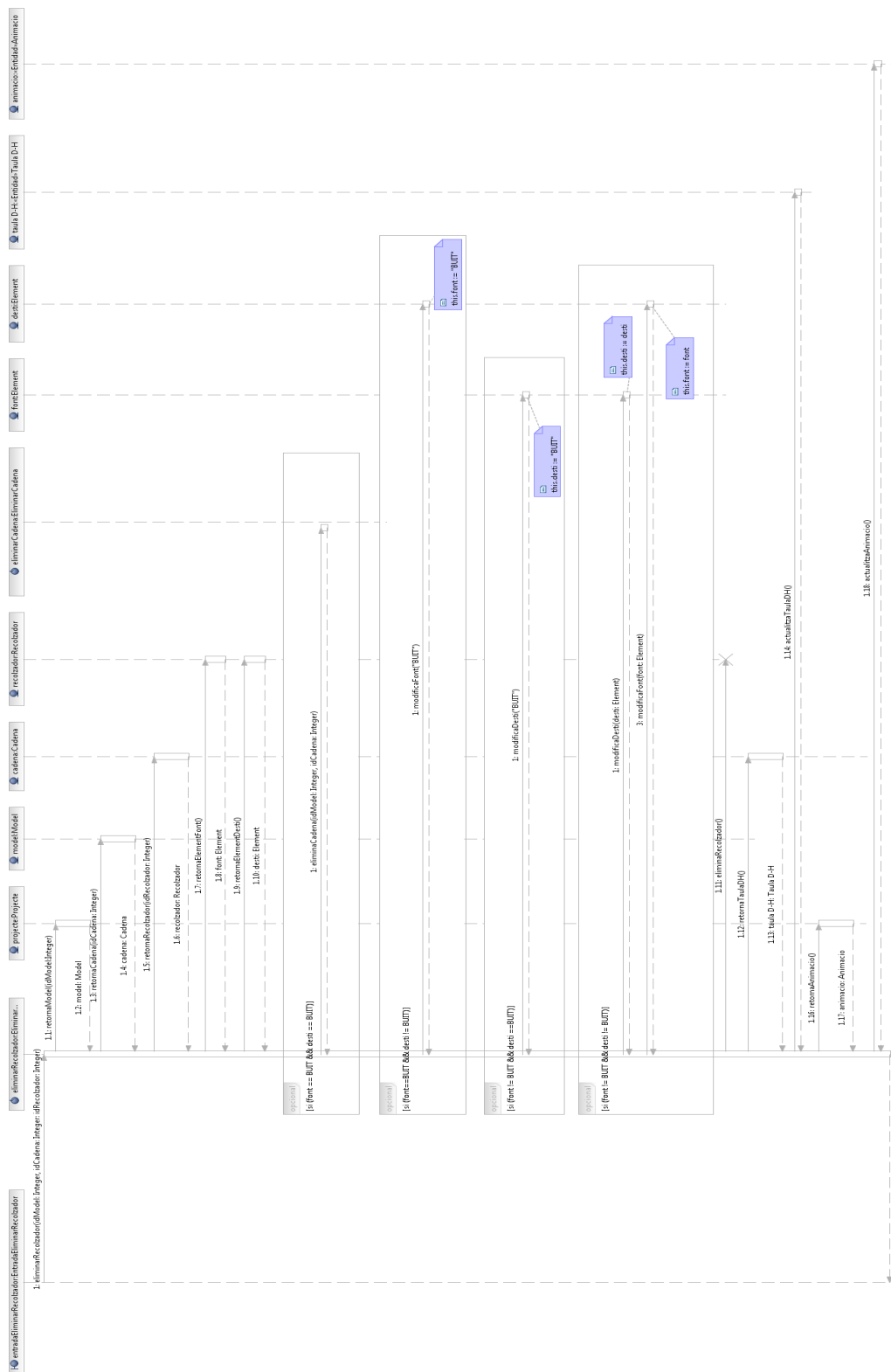


Figura 4.49: Diagrama de seqüència de Eliminar un Recolzador

4.2.21 Cas d'ús 2.15: Modificar la llargada d'un Recolzador

VOPC

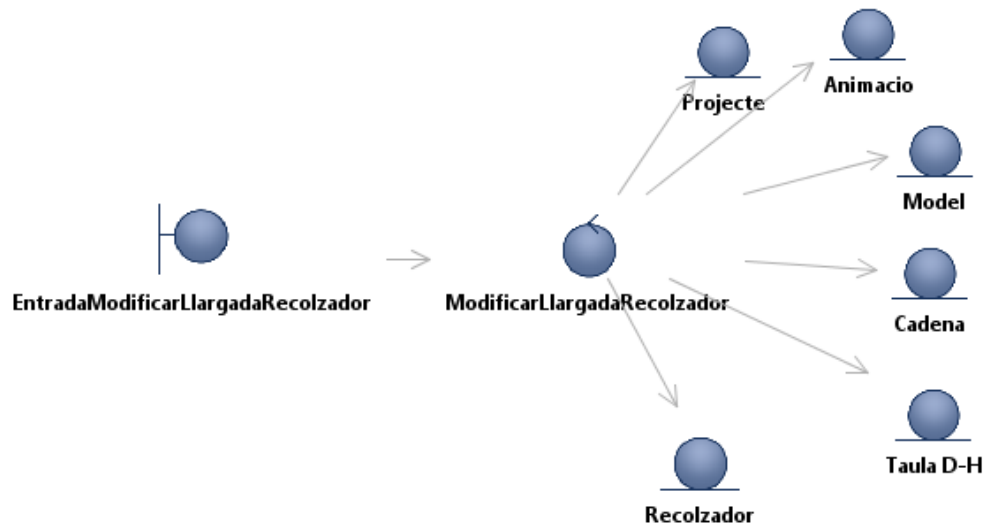


Figura 4.50: VOPC de Modificar la llargada d'un Recolzador

Diagrama de seqüència

De la mateixa manera que un "Link" pot modificar la seva llargada, en un Recolzador passa exactament el mateix, una vegada modificat, cal actualitzar la taula-DH de la cadena i actualitzar l'animació degut a que s'ha modificat l'estructura del model.

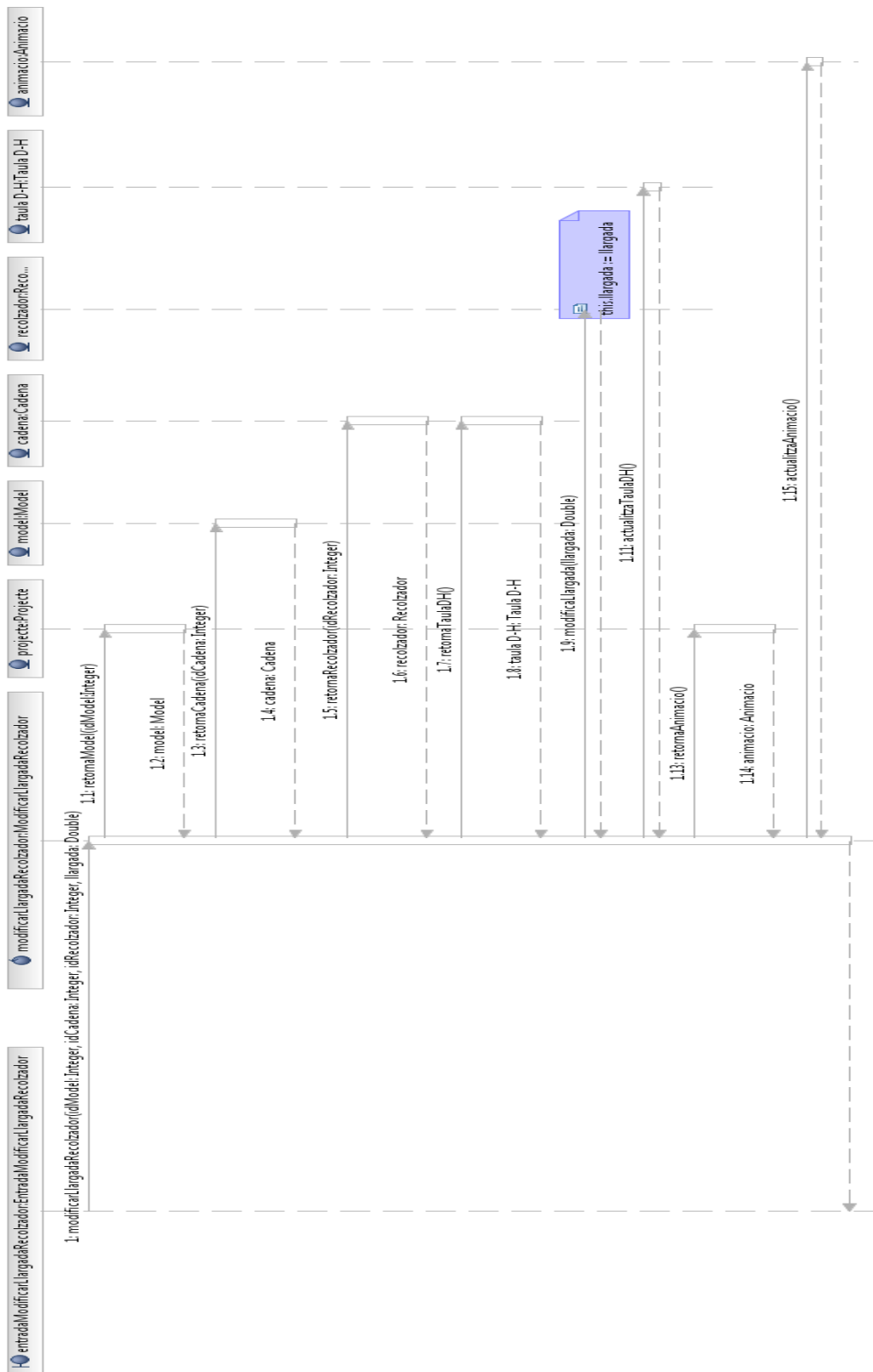


Figura 4.51: Diagrama de seqüència de Modificar la llargada d'un Recolzador

4.2.22 Cas d'ús 2.16: Modificar l'amplada d'un Recolzador

VOPC

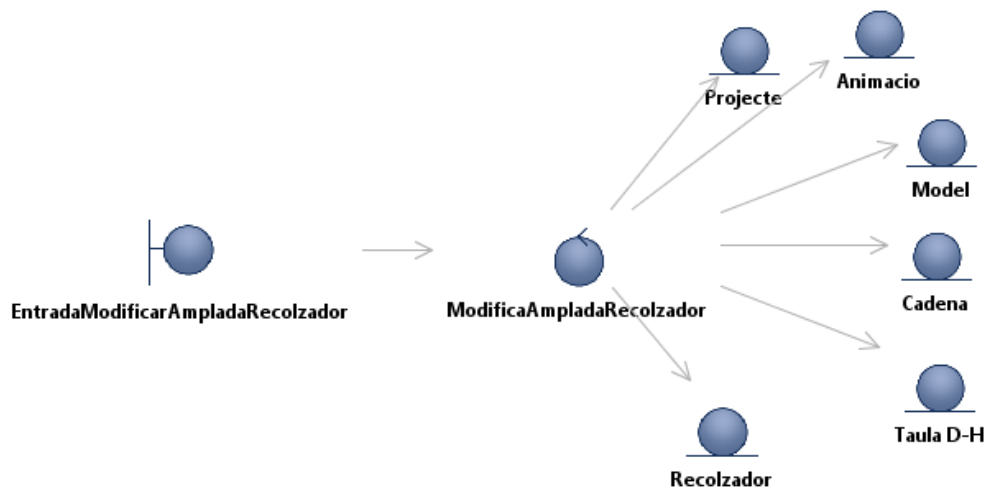


Figura 4.52: VOPC de Modificar l'amplada d'un Recolzador

Diagrama de seqüència

Exactament que el cas anterior, però en comptes de modificar la llargada es modifica l'amplada.

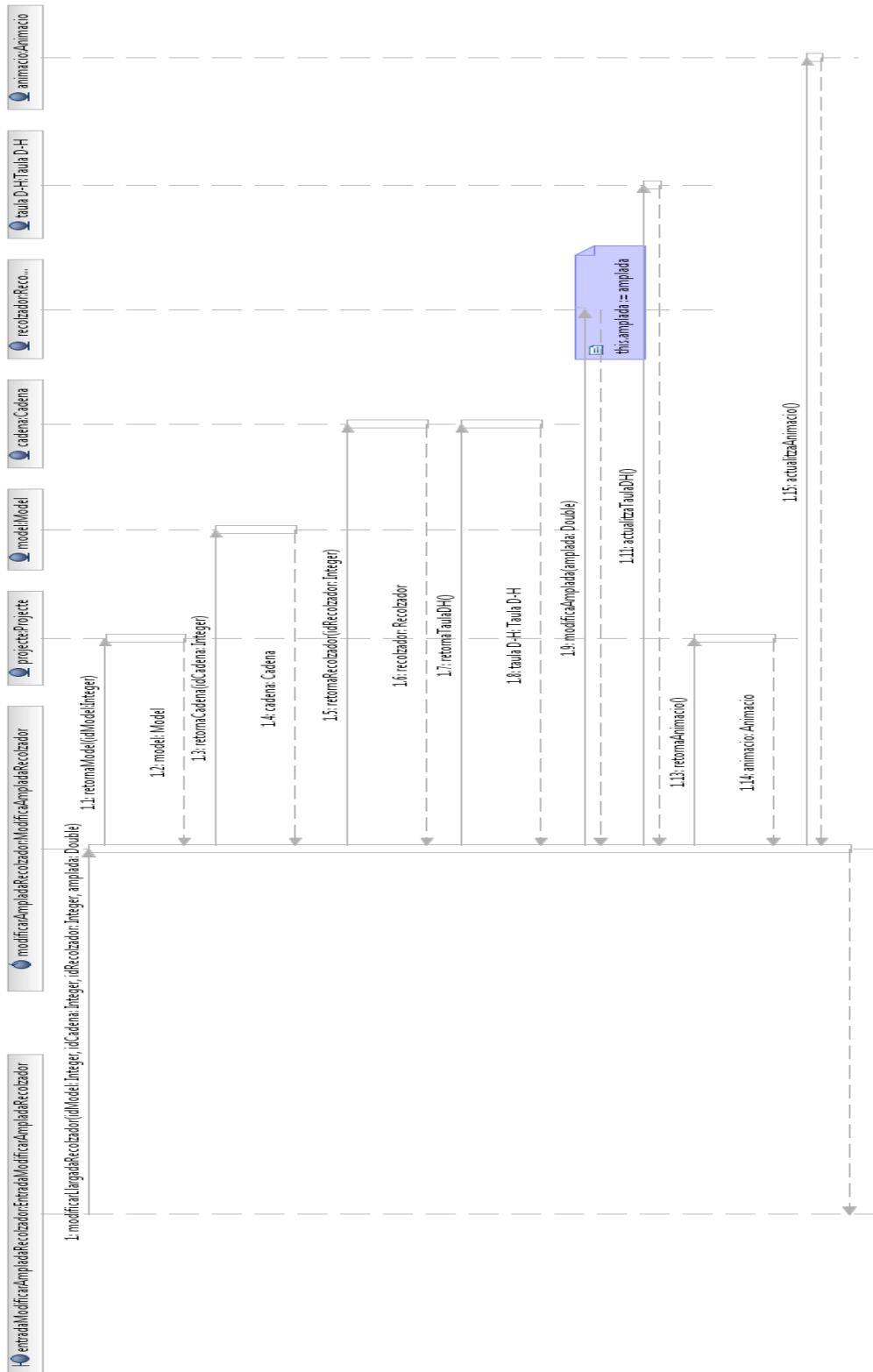


Figura 4.53: Diagrama de seqüència de Modificar l'ampada d'un Recolzador

4.2.23 Cas d'ús 3.1: Afegir un objecte

VOPC

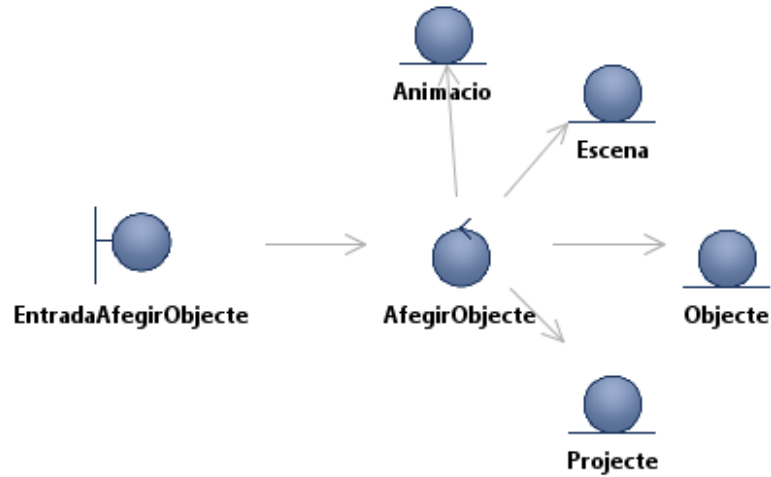


Figura 4.54: VOPC de Afegir un objecte

Diagrama de seqüència

Pel que fa a afegir un objecte, l'única complicació cau en recollir tota la informació per pintar-lo (rotació, posició, escalat i com també l'identificador que no apareix en explícitament en el diagrama) i després calcular si apareix algun punt de contacte en algun punt clau si s'en té algun. El diagrama següent mostra això:

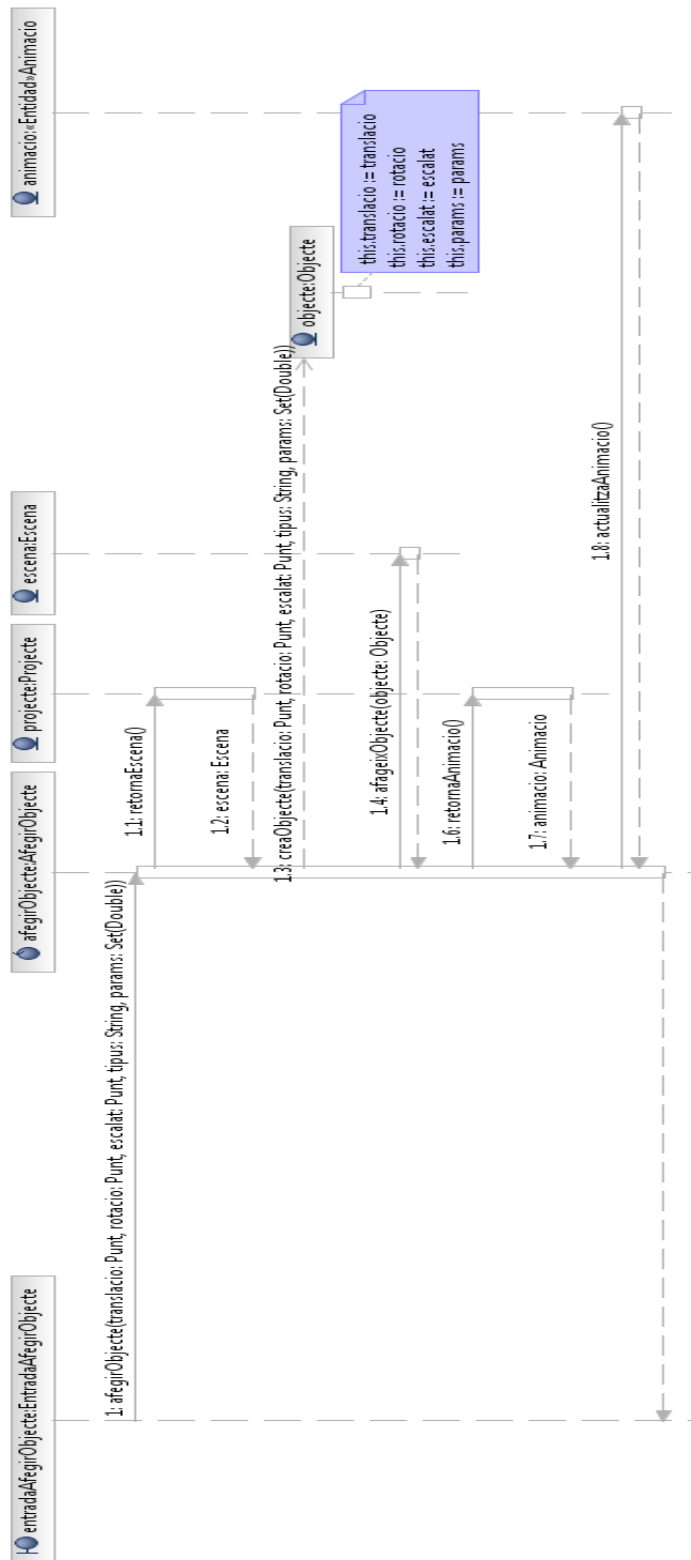


Figura 4.55: Diagrama de seqüència de Afegeix un objecte

4.2.24 Cas d'ús 4.1: Rotar objecte

VOPC

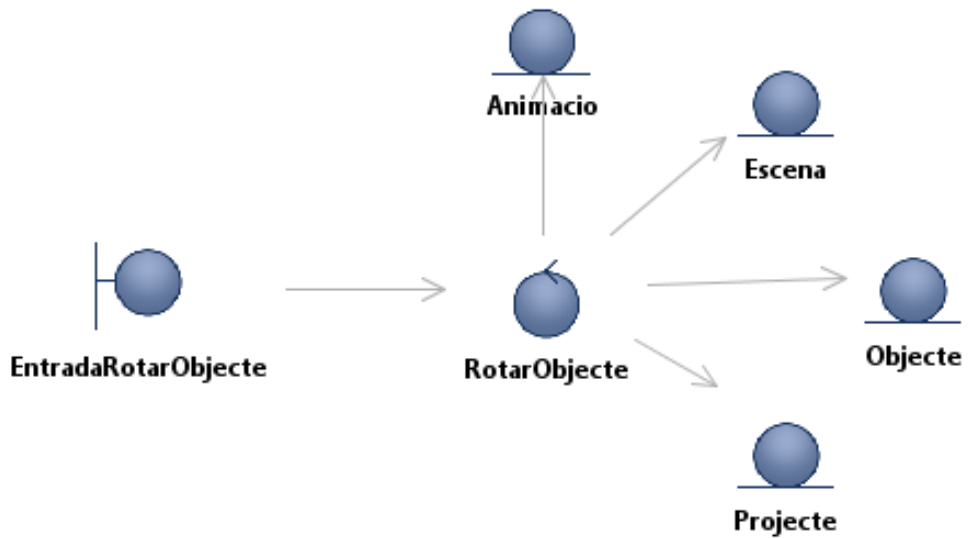


Figura 4.56: VOPC de Rotar objecte

Diagrama de seqüència

L'única informació necessària és quin objecte es vol rotar i la rotació que se li vol aplicar. Al final cal actualitzar l'animació, el diagrama següent mostra el procés que es duu a terme per realitzar el cas d'ús:

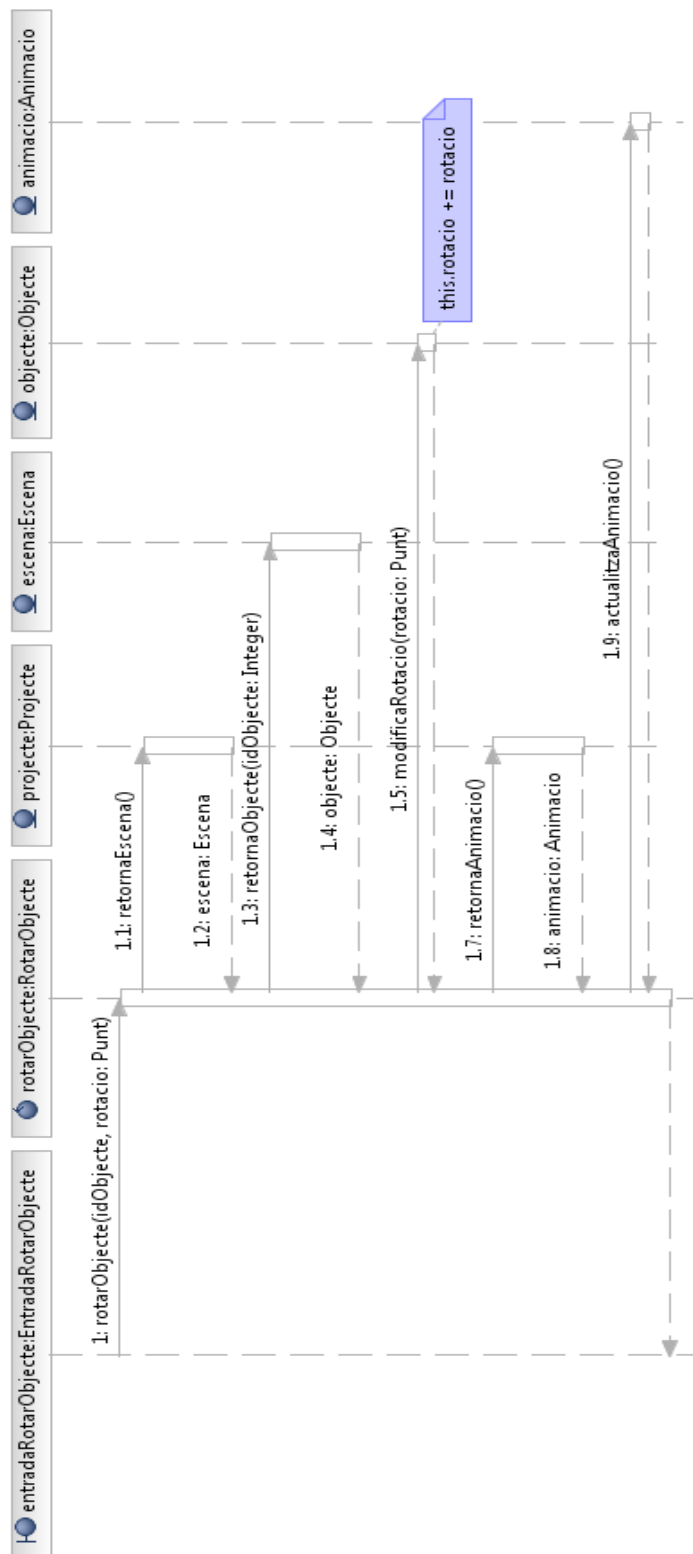


Figura 4.57: Diagrama de seqüència de Rotar objecte

4.2.25 Cas d'ús 4.2: Escalar objecte

VOPC

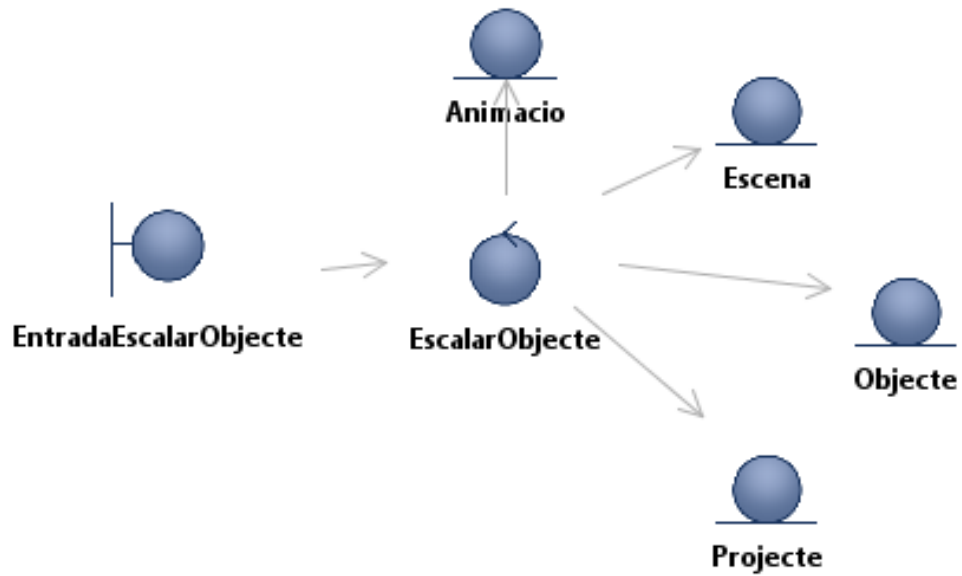


Figura 4.58: VOPC de Escalar objecte

Diagrama de seqüència

Lo mateix que el cas d'ús anterior, però en aquest cas es tracta l'escalat de l'objecte:

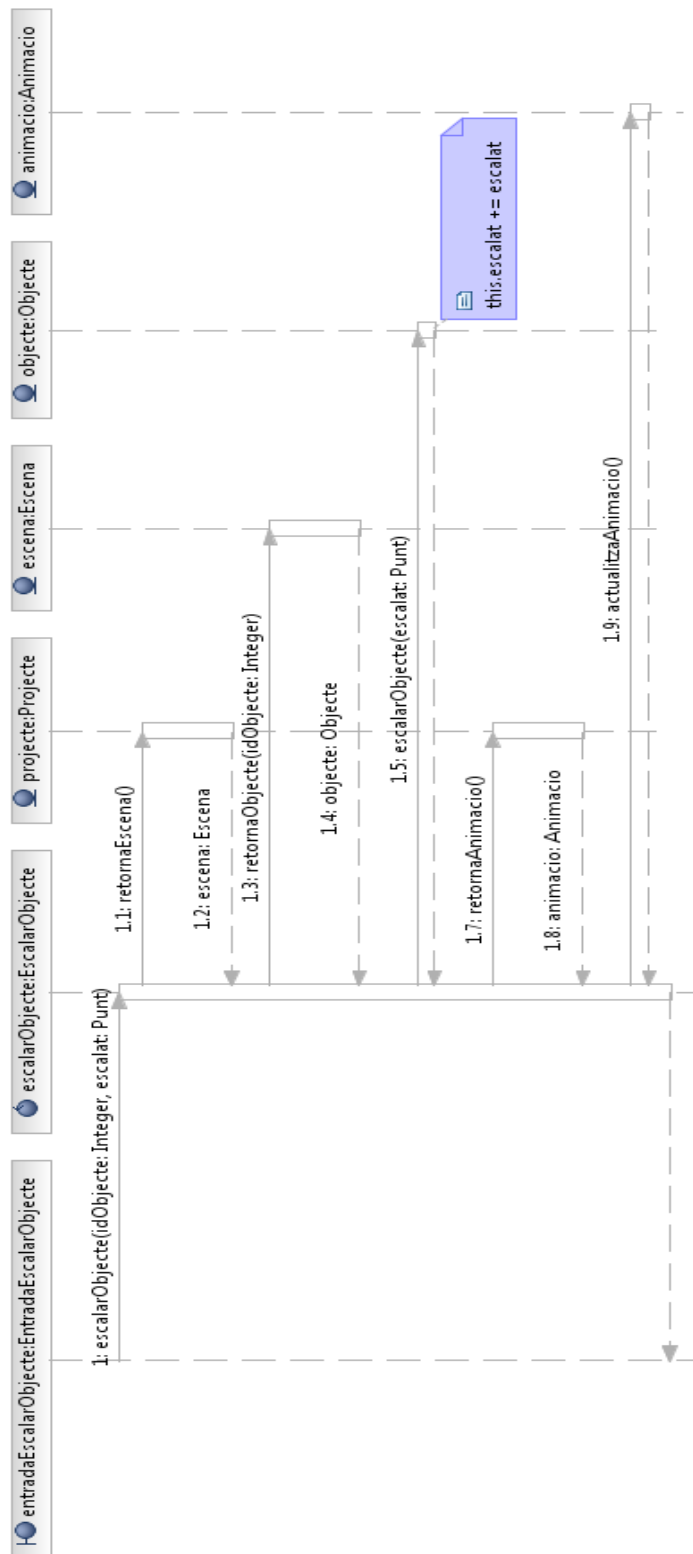


Figura 4.59: Diagrama de seqüència de Escalar objecte

4.2.26 Cas d'ús 4.3: Moure objecte

VOPC

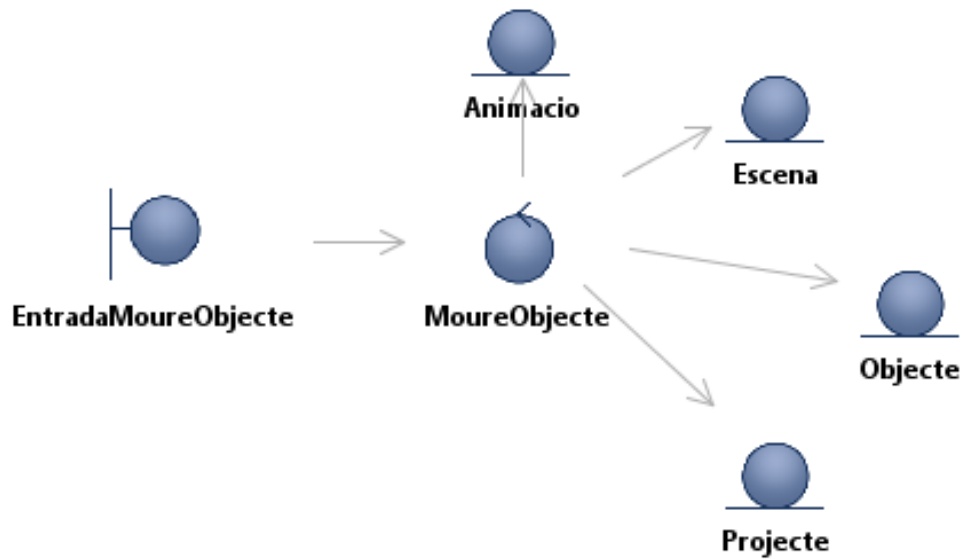


Figura 4.60: VOPC de Moure objecte

Diagrama de seqüència

Cas d'ús idèntic que els dos anteriors, però en aquest cas la posició de l'objecte es veu afectada.

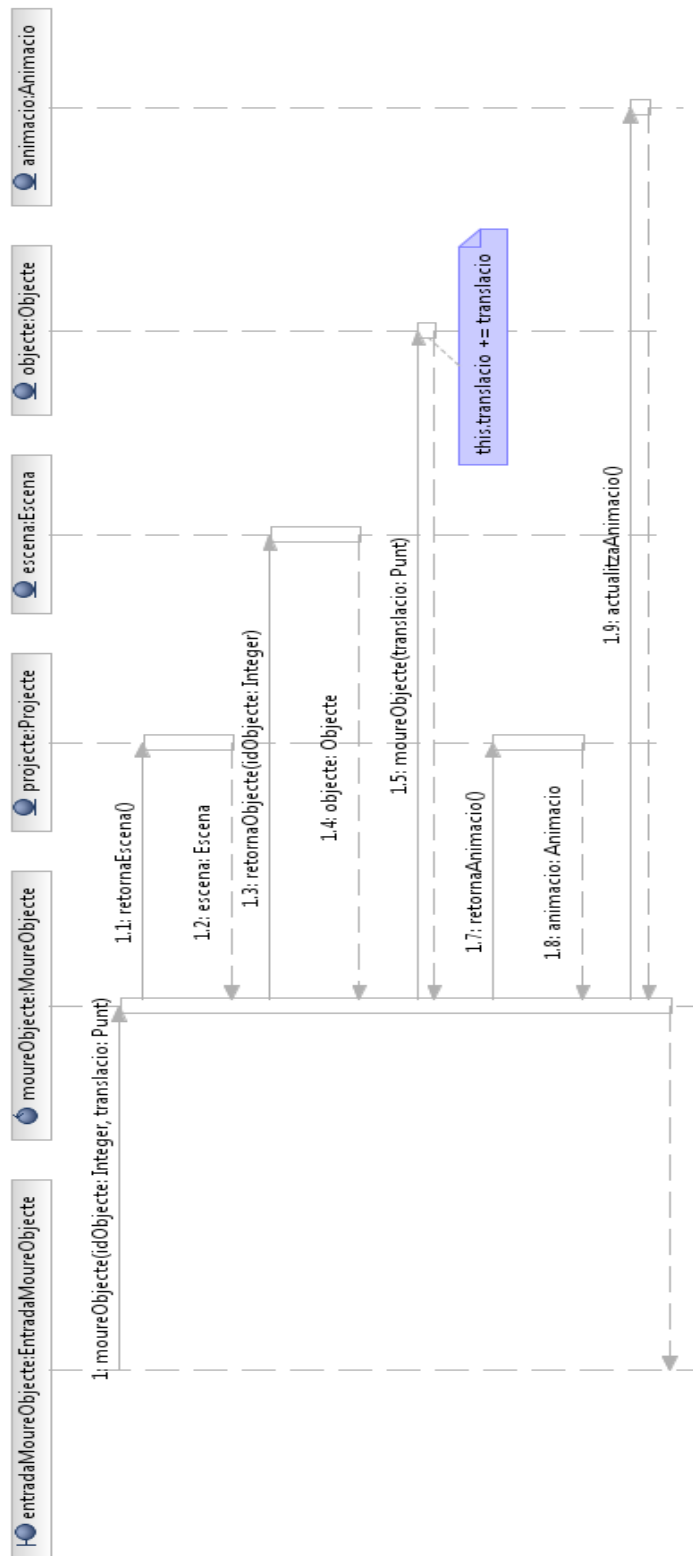


Figura 4.61: Diagrama de seqüència de Moure objecte

4.2.27 Cas d'ús 4.4: Duplicar objecte

VOPC

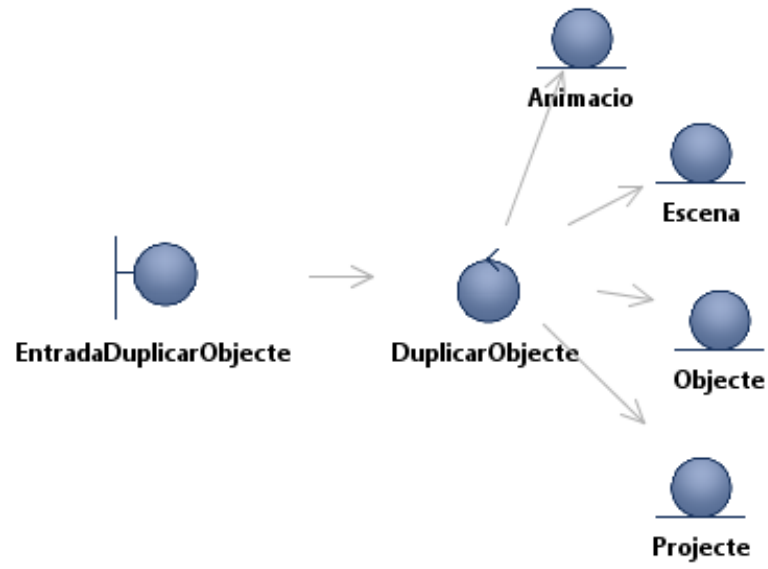


Figura 4.62: VOPC de Duplicar objecte

Diagrama de seqüència

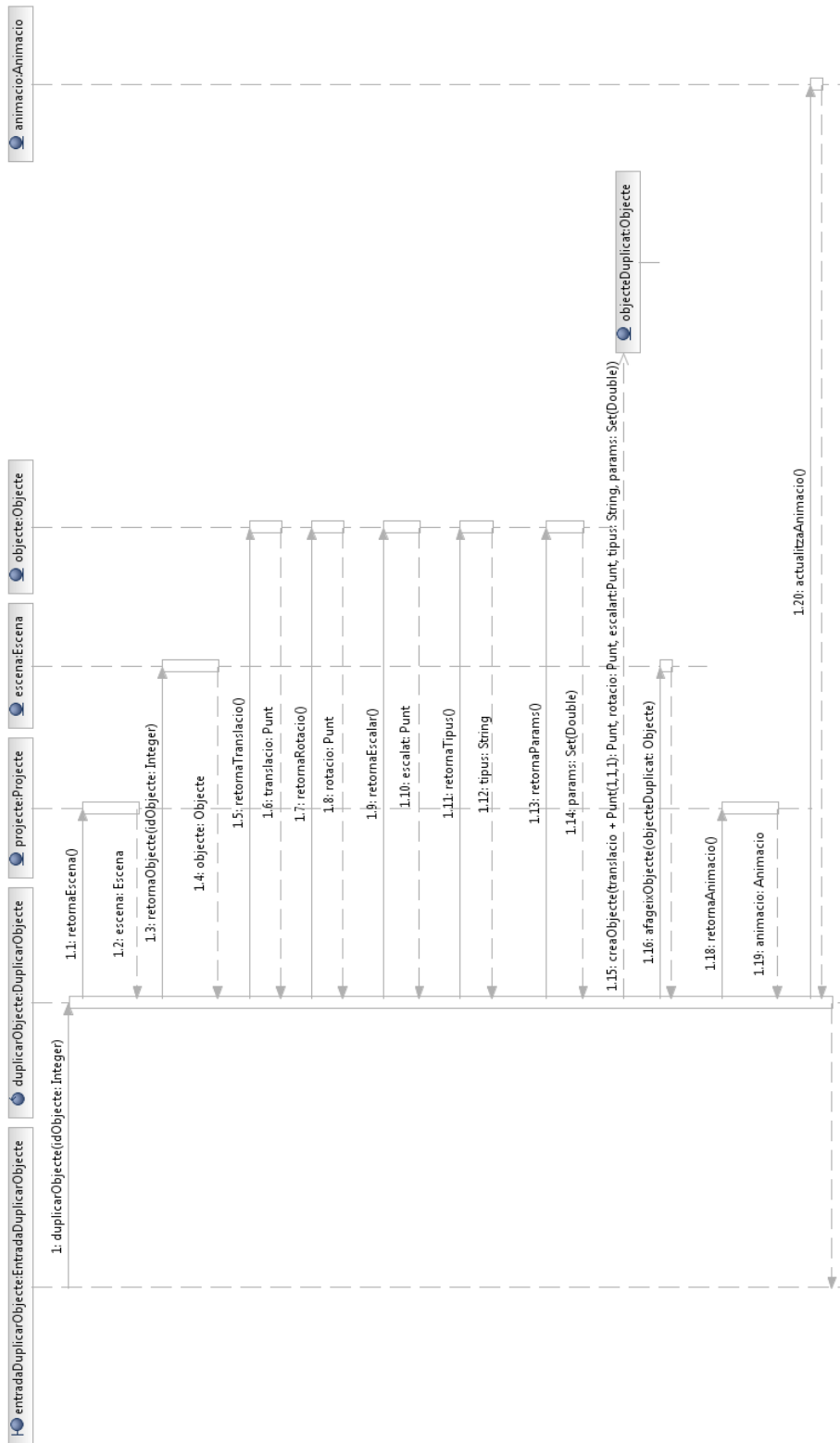


Figura 4.63: Diagrama de seqüència de Duplicar objecte

4.2.28 Cas d'ús 4.5: Eliminar objecte

VOPC

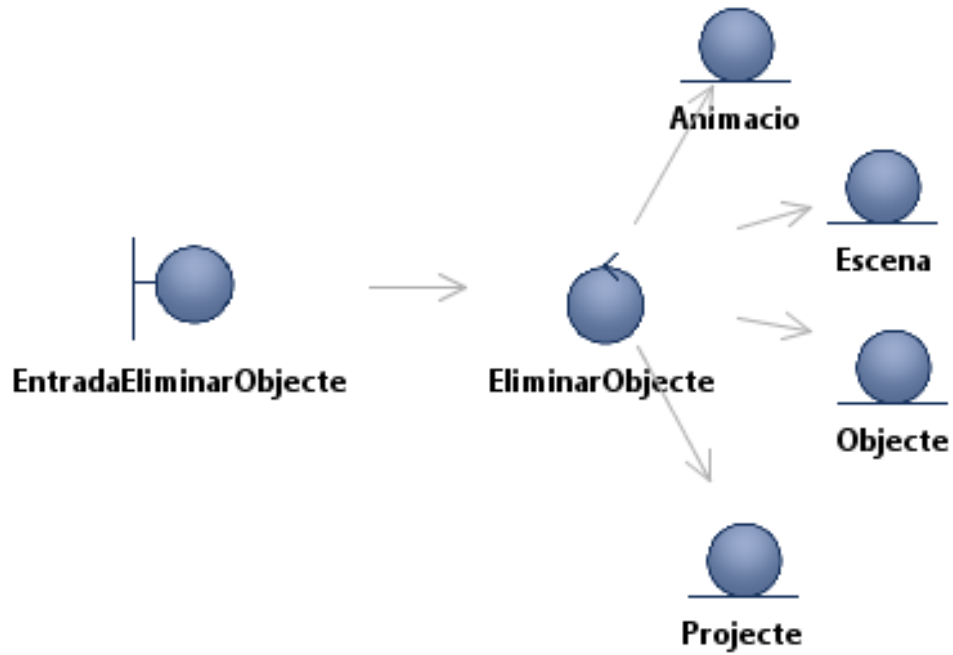


Figura 4.64: VOPC de Eliminar objecte

Diagrama de seqüència

És el mateix cas que afegir un objecte, amb la única diferència que l'objecte que es vol posar està a l'escena i es copien les mateixes característiques excepte la de la posició, ja que sinó quedarien sol·lapats. Així doncs, una vegada afegits, cal actualitzar l'animació per si apareix algun punt de contacte nou:

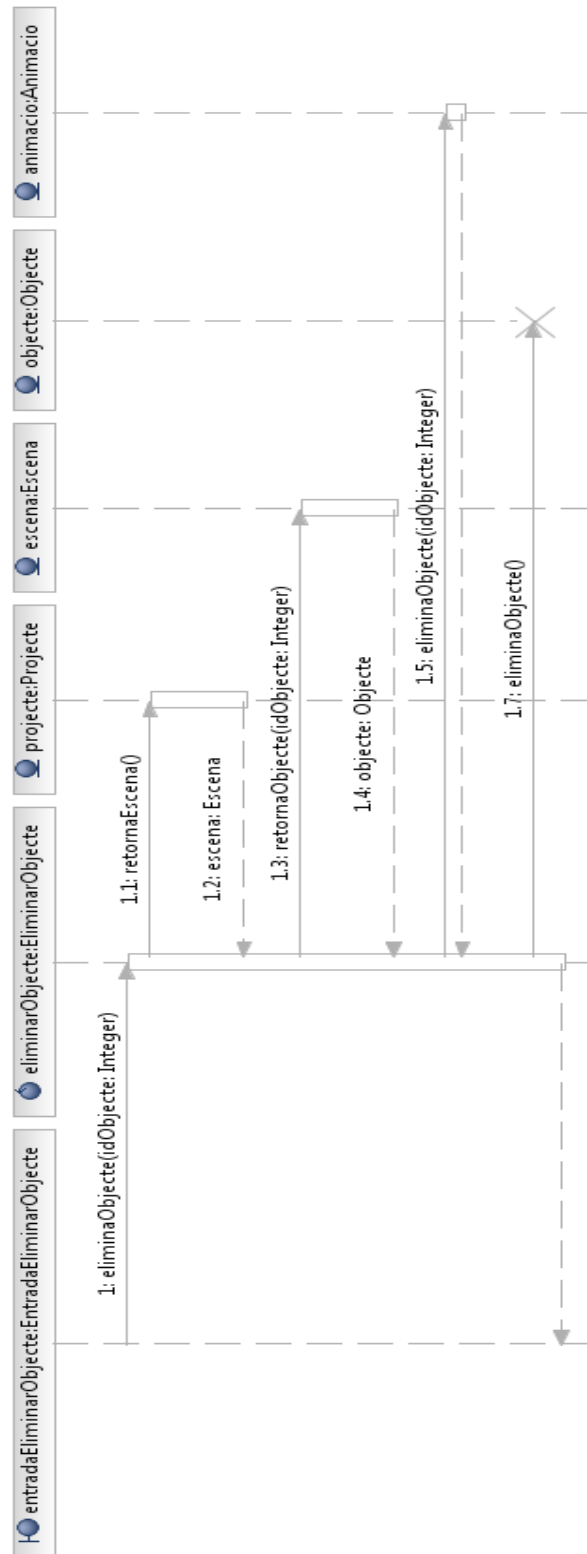


Figura 4.65: Diagrama de seqüència de Eliminar objecte

4.2.29 Cas d'ús 4.6: Modificar paràmetre objecte

VOPC

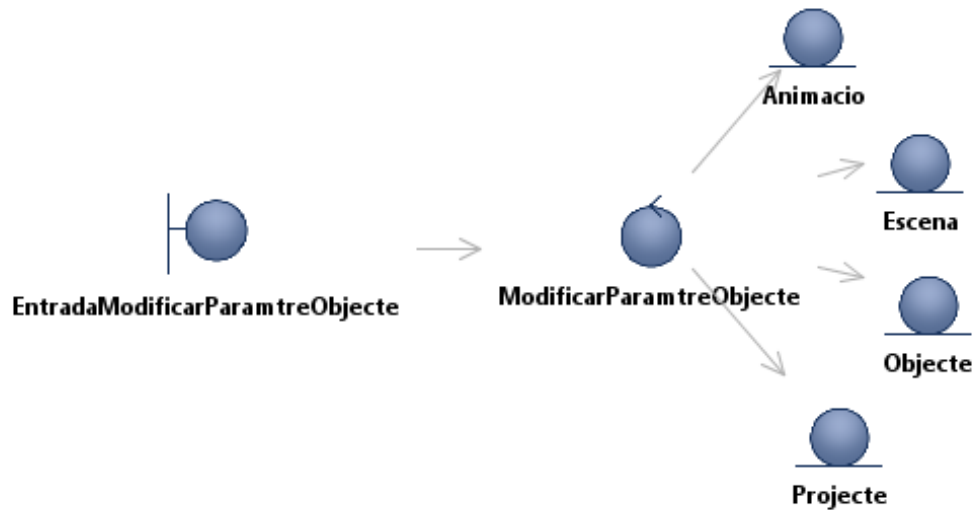


Figura 4.66: VOPC de Modificar paràmetre objecte

Diagrama de seqüència

El sistema deixa a l'usuari col·locar objectes tridimensionals, com esferes, plans, cubs... i aquests tenen uns paràmetres concrets com l'altura o el radi, per exemple, en aquest cas d'ús, l'usuari podrà modificar aquests paràmetres. Així doncs, el procés serà el següent:

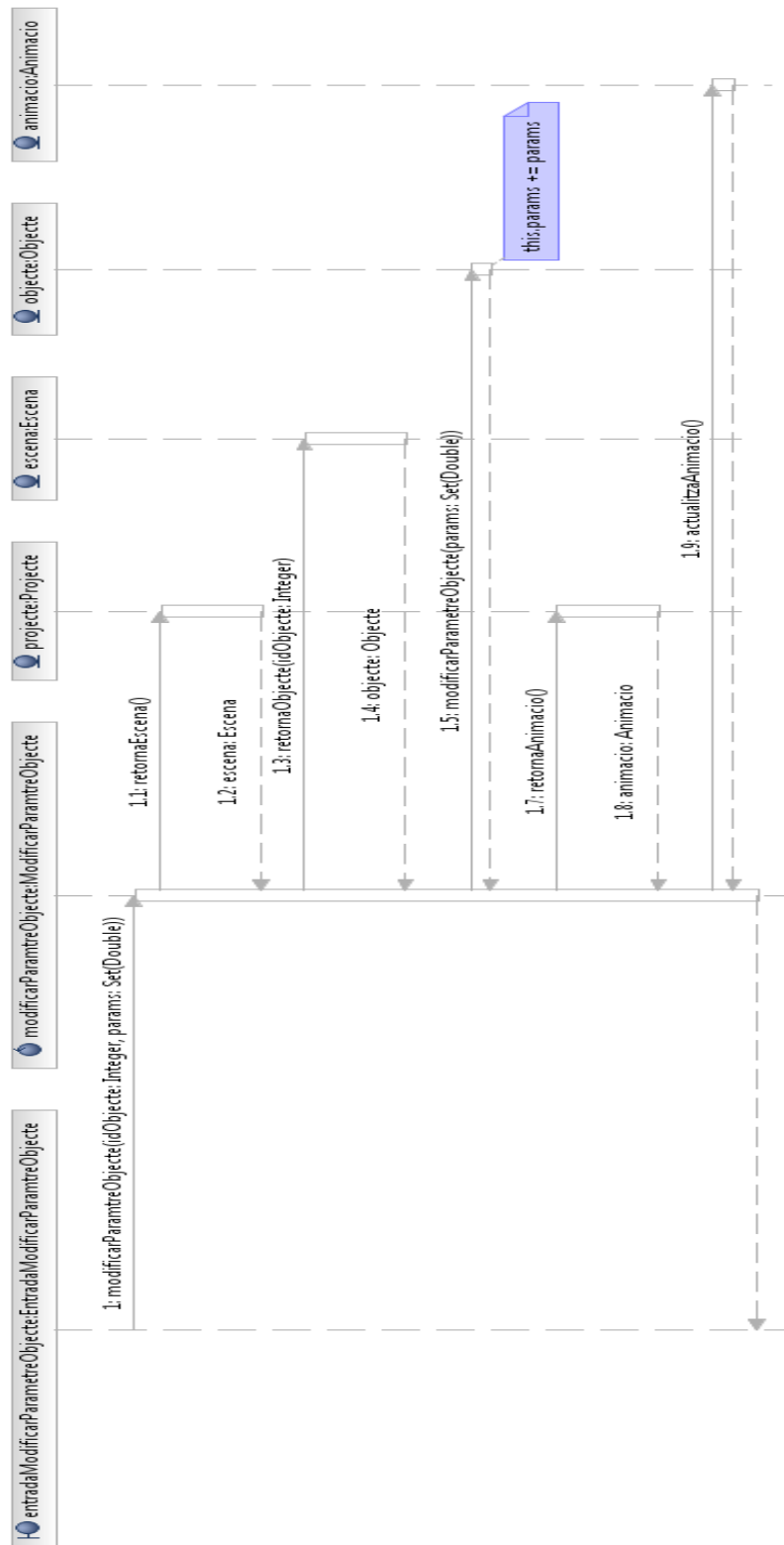


Figura 4.67: Diagrama de seqüència de Modificar paràmetre objecte

4.2.30 Cas d'ús 5.1: Crear punt clau

VOPC

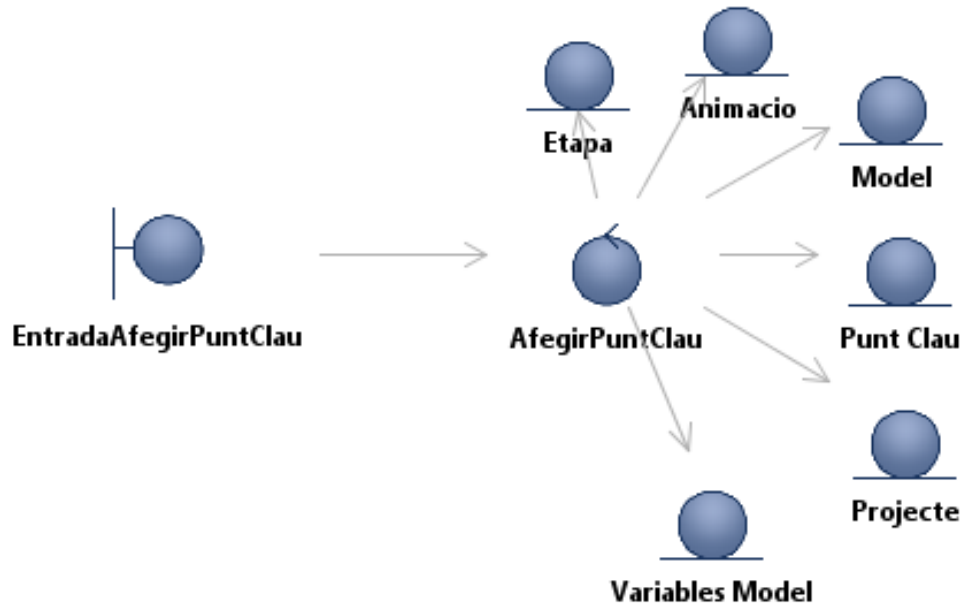


Figura 4.68: VOPC de Crear punt clau

Diagrama de seqüència

Una de les causes principals de que es pugui calcular tot el moviment sencer ve donat pels punts claus del moviment, aquells instants, en que hi ha un contacte o varis amb alguna superfície. L'usuari en el moment en que desitgi crear un punt clau, cal dir-li de quin model es tracta. Aleshores, totes les variables del model (posició, orientació i angles de les articulacions) s'agafaran de la instància presa del model (mitjançant un punt clau creat pel sistema i que no estarà inclosa en l'animació). Vegeu-ho a la figura següent:

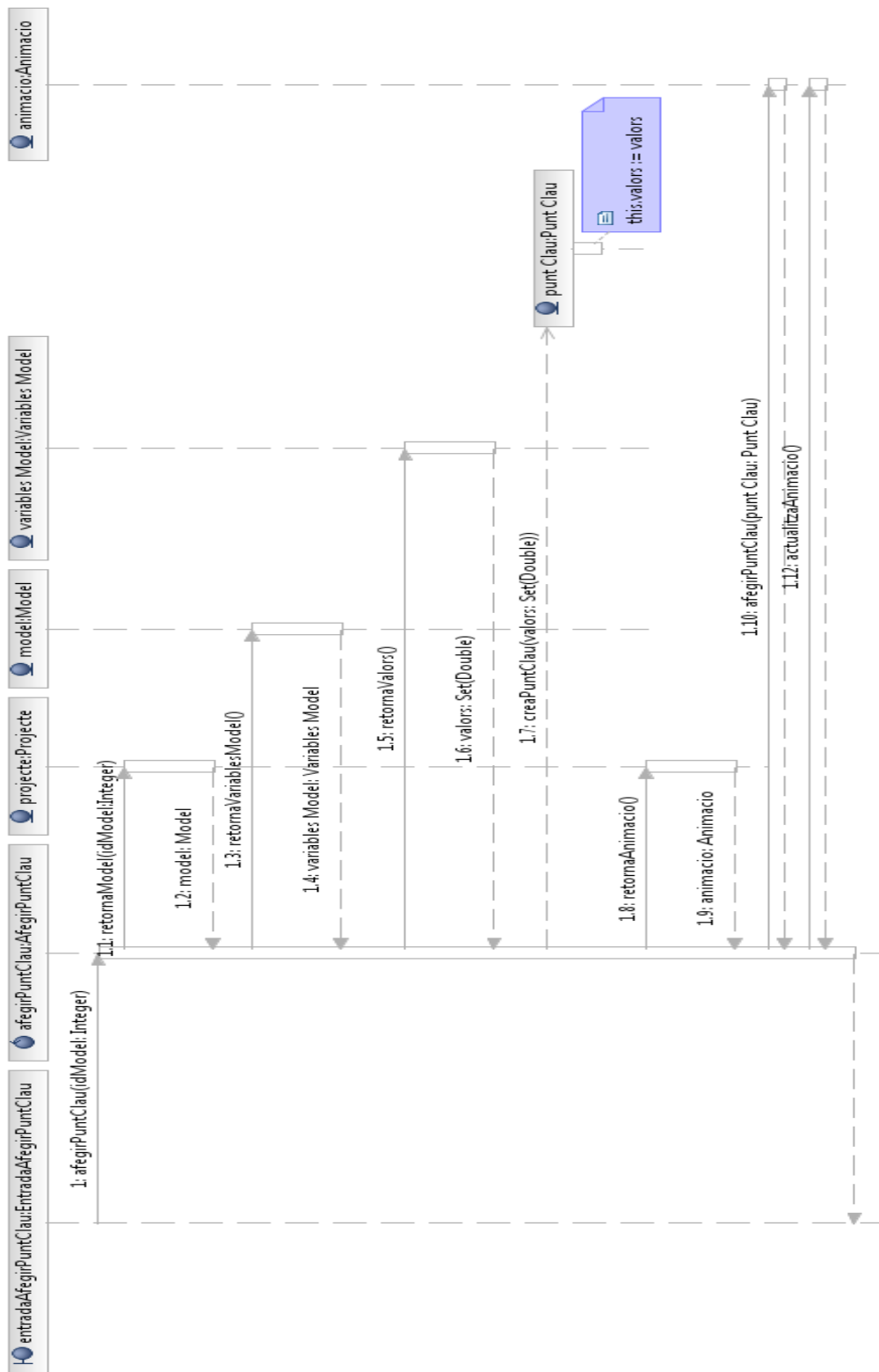


Figura 4.69: Diagrama de seqüència de Crear punt clau

4.2.31 Cas d'ús 5.2: Modificar paràmetre del model articulat

VOPC

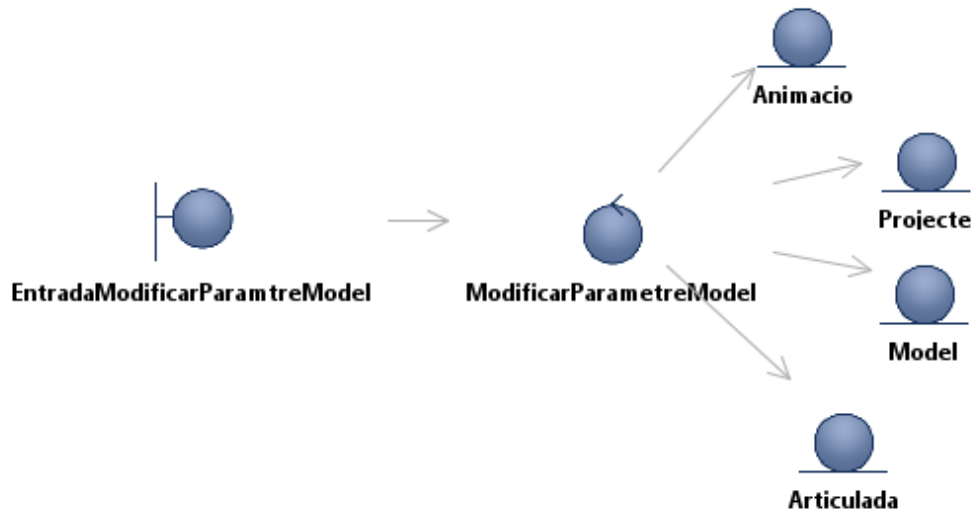


Figura 4.70: VOPC de Modificar paràmetre del model articulat

Diagrama de seqüència

En aquest cas, l'usuari podrà modificar a gust tots els paràmetres variables del model, com posició, rotació, angle d'una articulació, esforç, força... El sistema ho enregistrarà en el punt clau i ho tindrà en compte en el moment del càlcul:

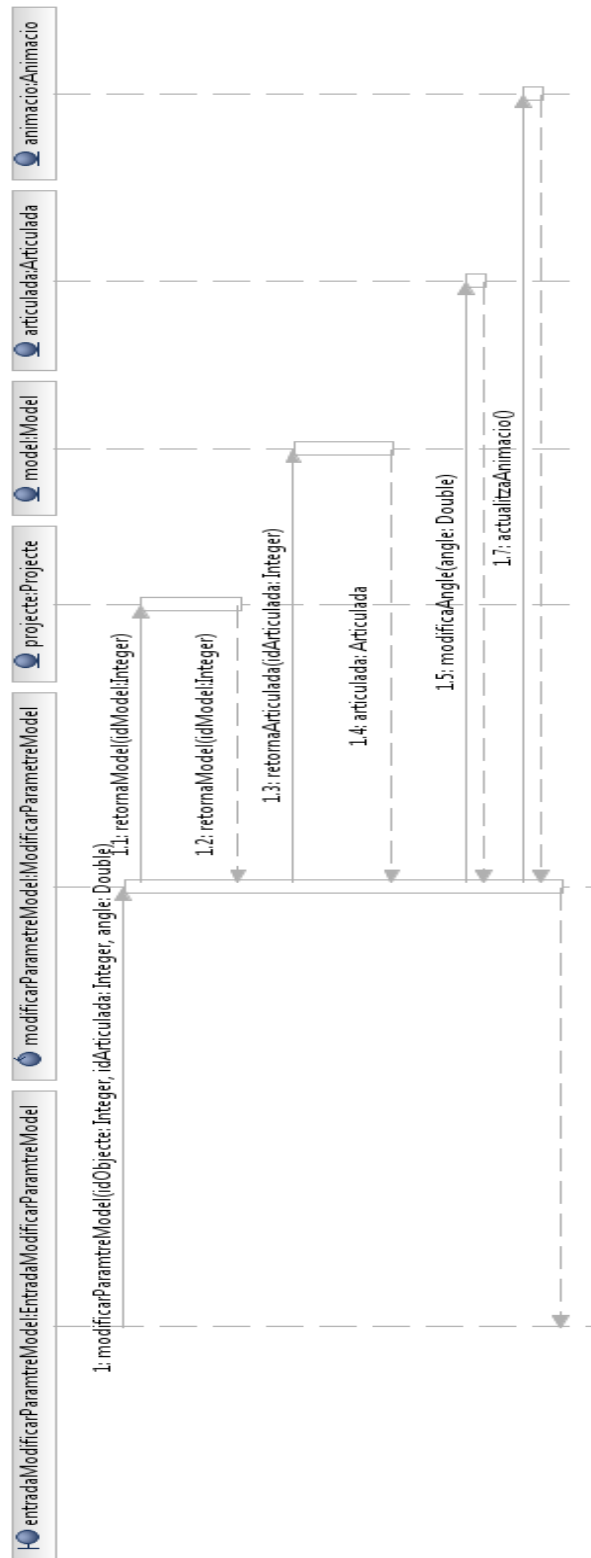


Figura 4.71: Diagrama de seqüència de Modificar paràmetre del model articulat

4.2.32 Cas d'ús 5.3: Afegir punt de contacte

VOPC

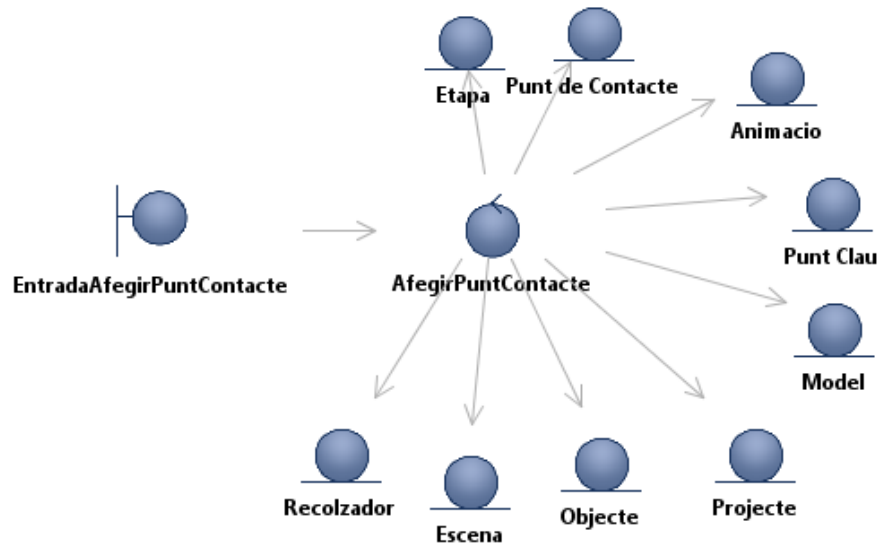


Figura 4.72: VOPC de Afegir punt de contacte

Diagrama de seqüència

Cal mitjançant la interfície poder donar la posició d'aquest punt de contacte i els elements que intervenen. El procés d'incloure tot això al sistema és el següent:

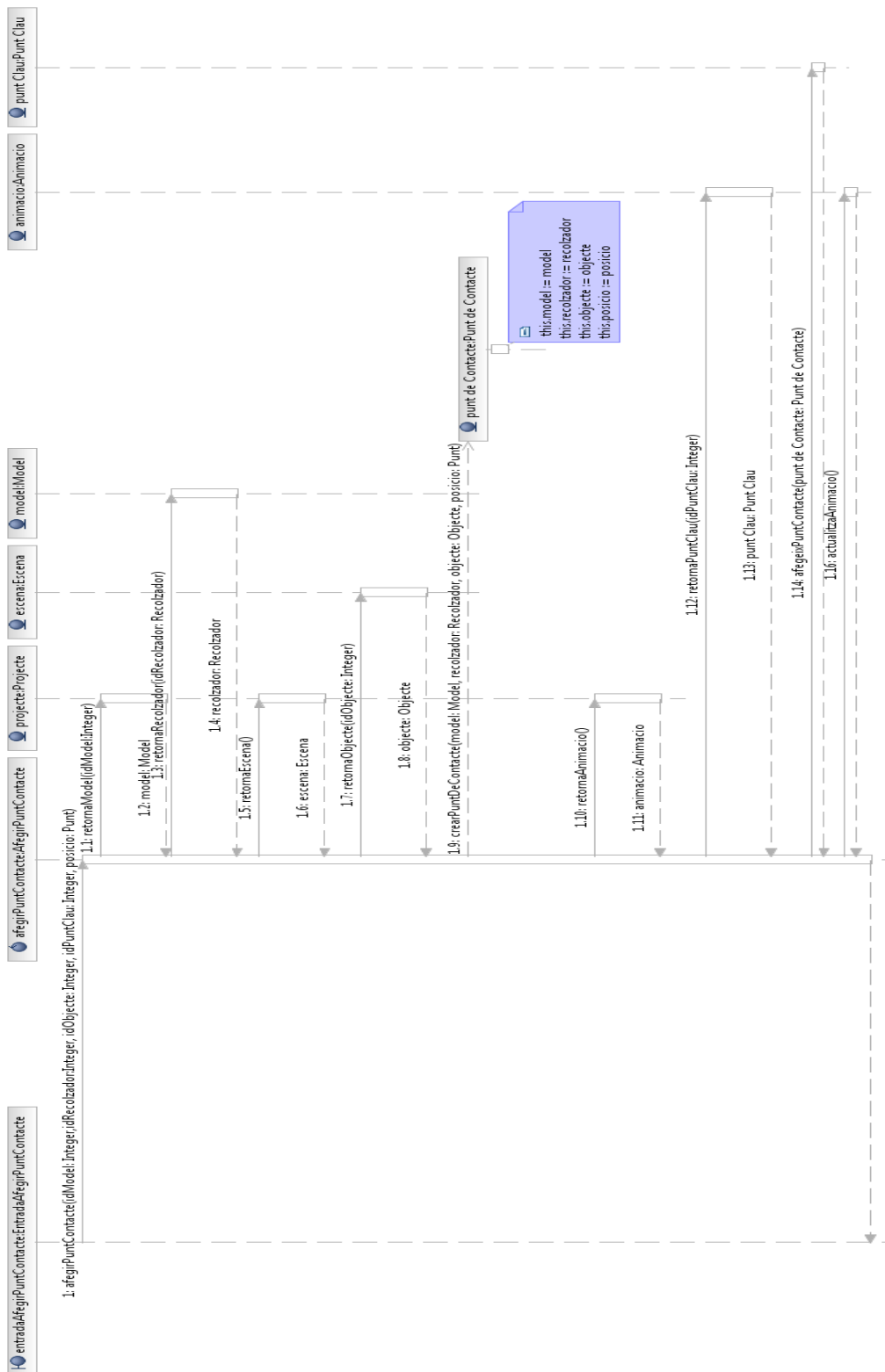


Figura 4.73: Diagrama de seqüència de Afegeix punt de contacte

4.2.33 Cas d'ús 5.4: Assignar instant de temps

VOPC

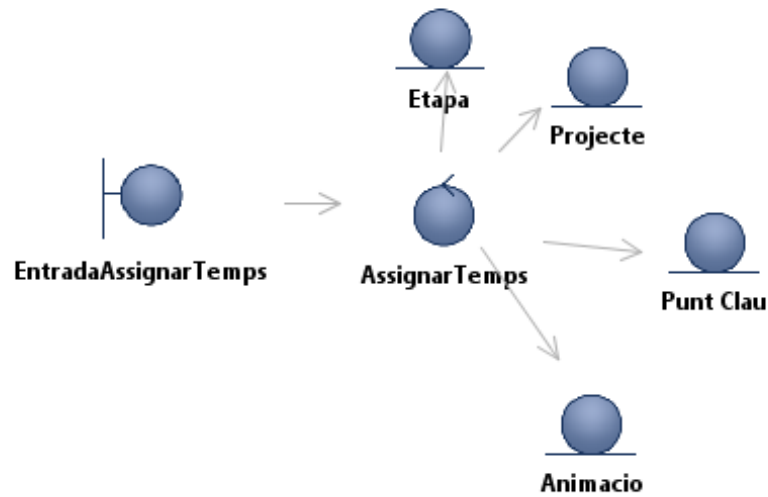


Figura 4.74: VOPC de Assignar instant de temps

Diagrama de seqüència

Per marcar l'ordre d'un punt clau a l'altre cal introduir un temps al punt clau, això és el que pretén aquest cas d'ús. L'usuari elegirà un punt clau, amb el seu identificador, i donarà un temps. El flux de la informació queda descrit en el següent diagrama:

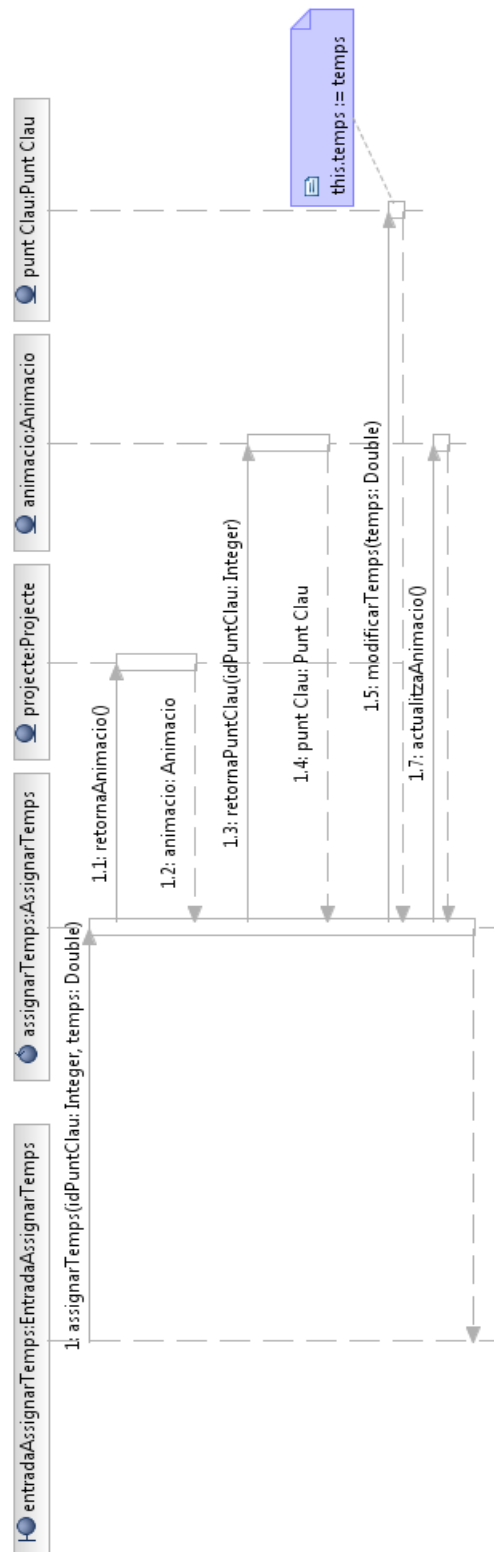


Figura 4.75: Diagrama de seqüència de Assignar instant de temps

4.2.34 Cas d'ús 5.5: Generar animació

VOPC

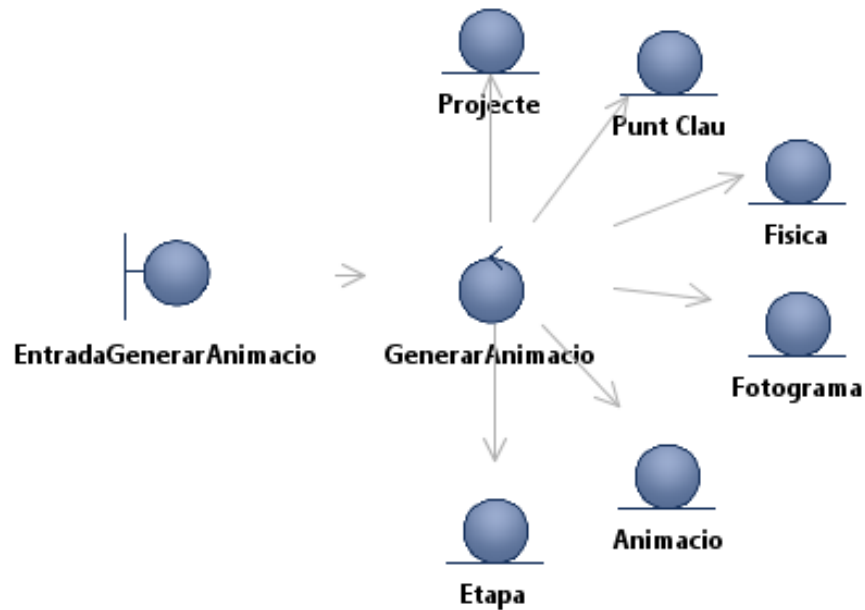


Figura 4.76: VOPC de Generar animació

Diagrama de seqüència

La funció del sistema en aquest cas, és cridar una funció del motor físic del sistema perquè calculi tots els fotogrames a partir dels punts claus i els afegeixi a l'animació:

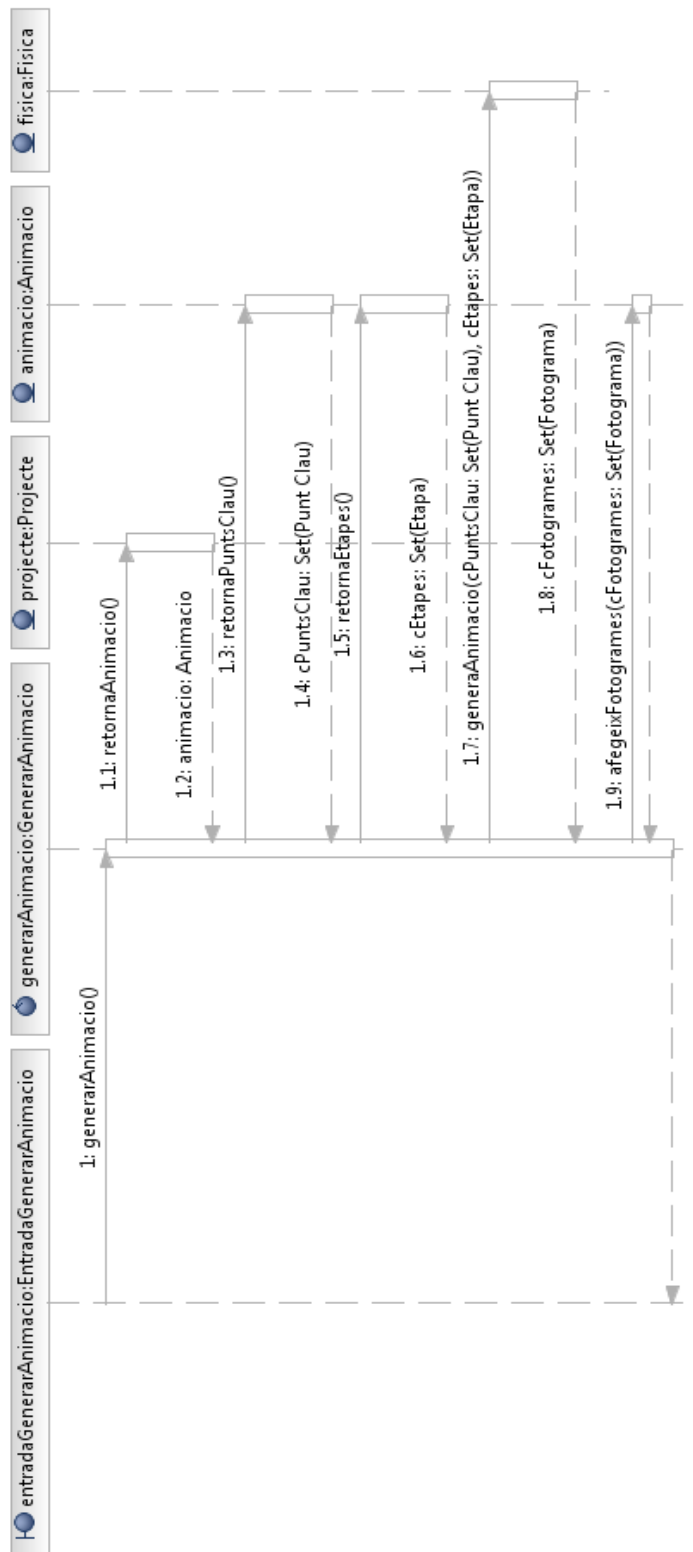


Figura 4.77: Diagrama de seqüència de Generar animació

4.2.35 Cas d'ús 6.1: Eliminar punt clau

VOPC

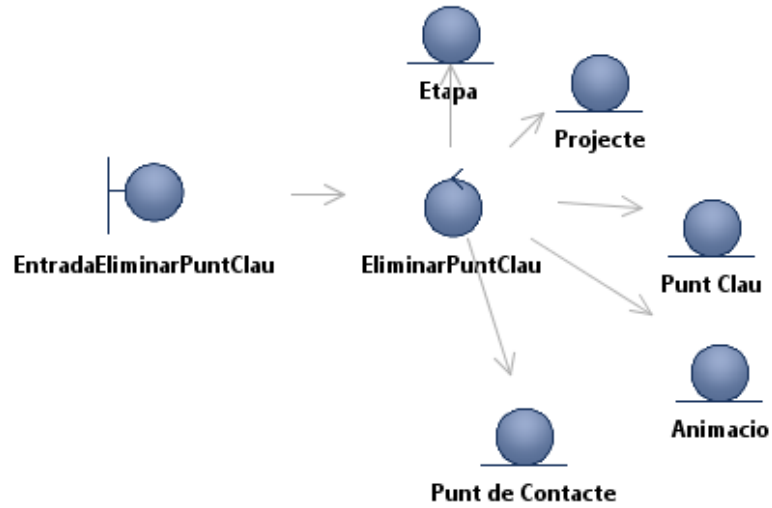


Figura 4.78: VOPC de Eliminar punt clau

Diagrama de seqüència

En aquest cas cal saber quin punt clau es vol eliminar i actualitzar l'animació per tal de que elimini aquest punt clau:

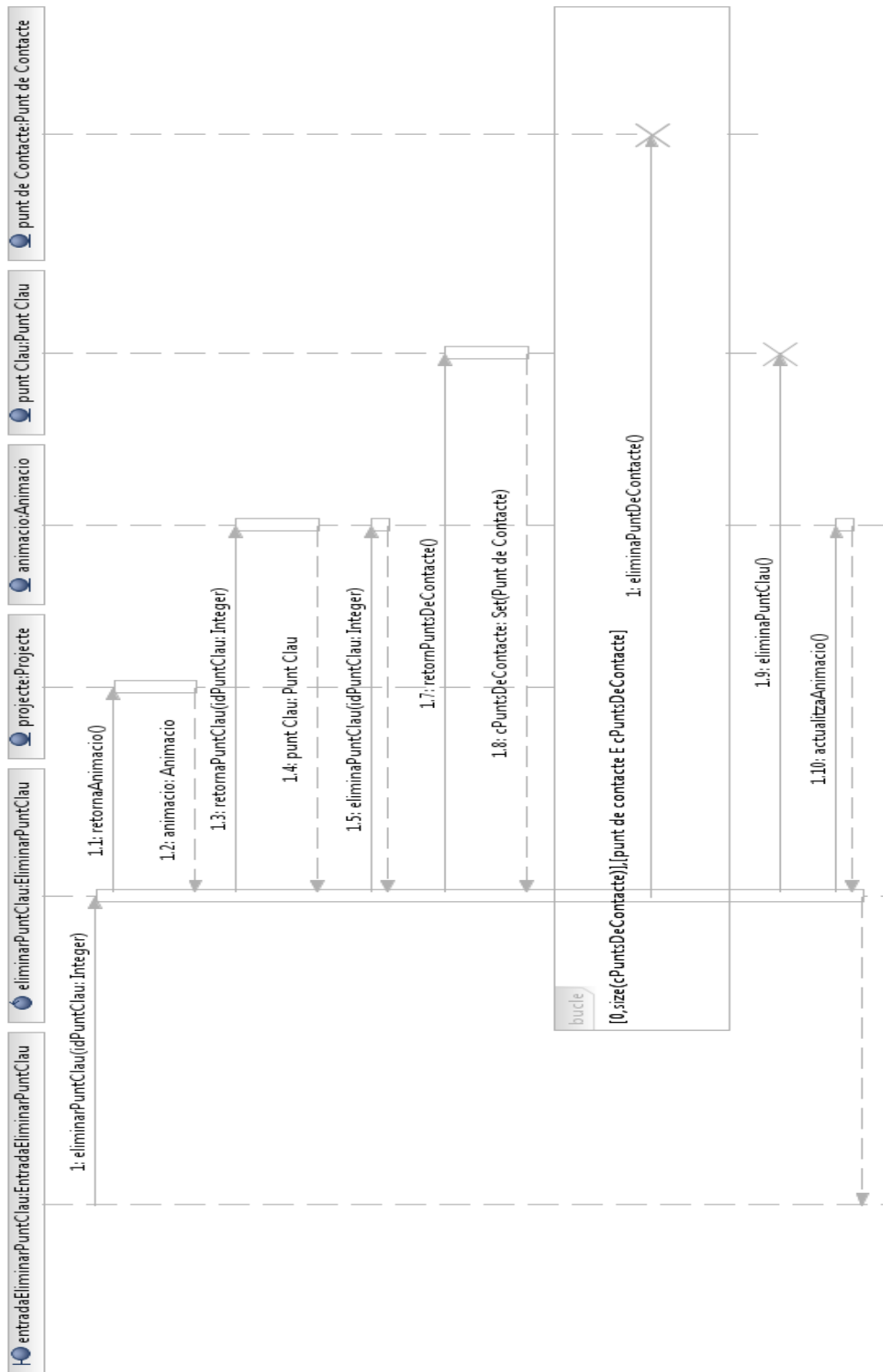


Figura 4.79: Diagrama de seqüència de Eliminar punt clau

4.2.36 Cas d'ús 6.2: Modificar instant de temps

VOPC

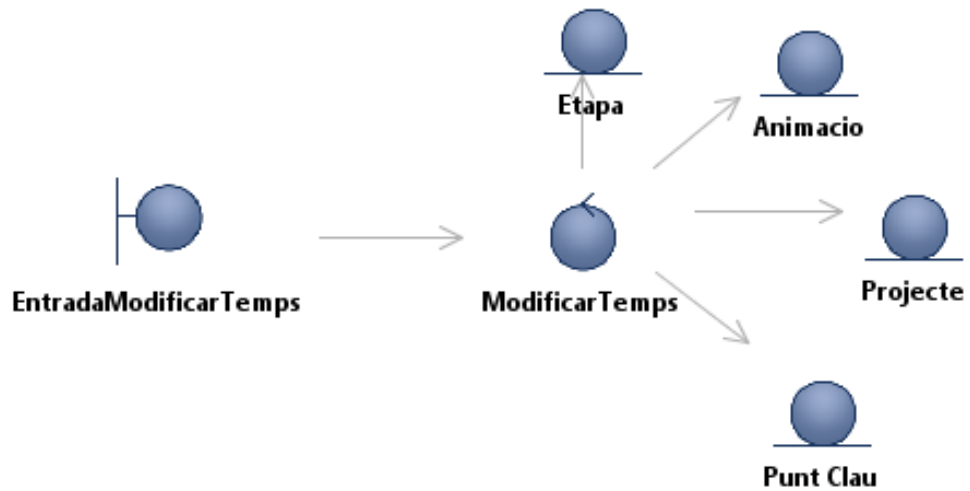


Figura 4.80: VOPC de Modificar instant de temps

Diagrama de seqüència

Cal posar el nou temps al punt clau que se li assigna i si s'escau actualitzar l'animació per si cal reordenar els punts claus i les re-calcular les etapes. En la següent diagrama de seqüència es pot veure:

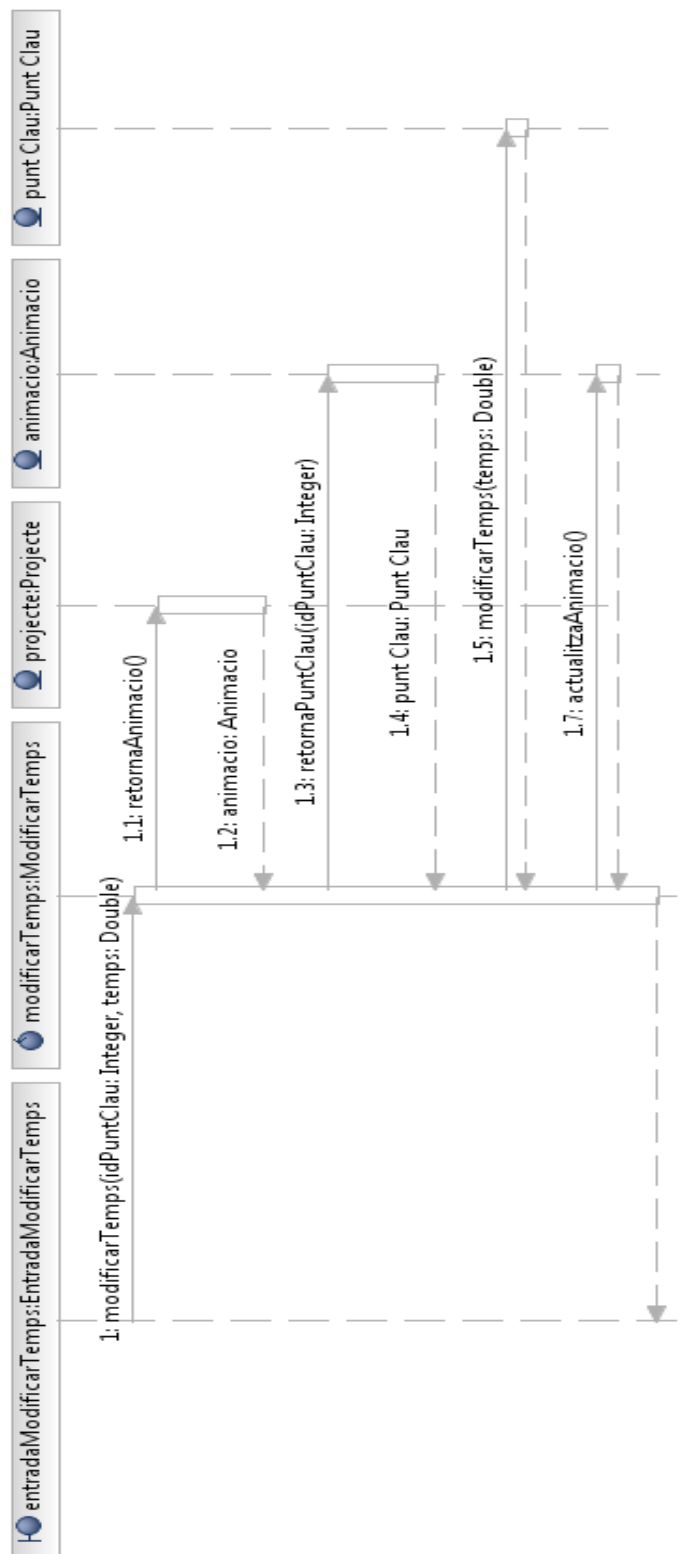


Figura 4.81: Diagrama de seqüència de Modificar instant de temps

4.2.37 Cas d'ús 6.3: Eliminar punt de contacte

VOPC

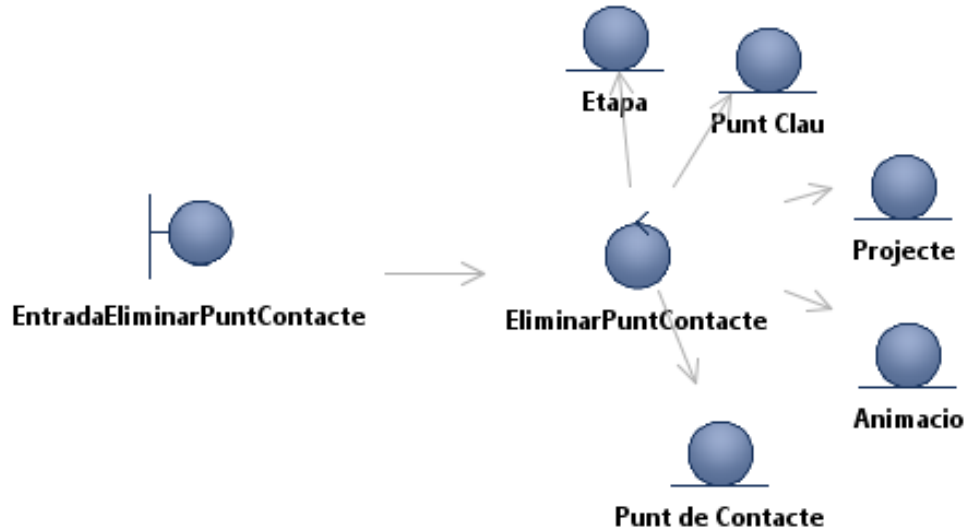


Figura 4.82: VOPC de Eliminar punt de contacte

Diagrama de seqüència

El fet de que a l'usuari li interessi no tenir un objecte en compte a l'animació, o que senzillament ja no hi hagi un punt de contacte degut a la modificació de l'estructura. Només cal saber de quin punt de contacte es tracta i de quin punt clau intervé. Caldrà finalment actualitzar l'animació de manera que recalculi les etapes.

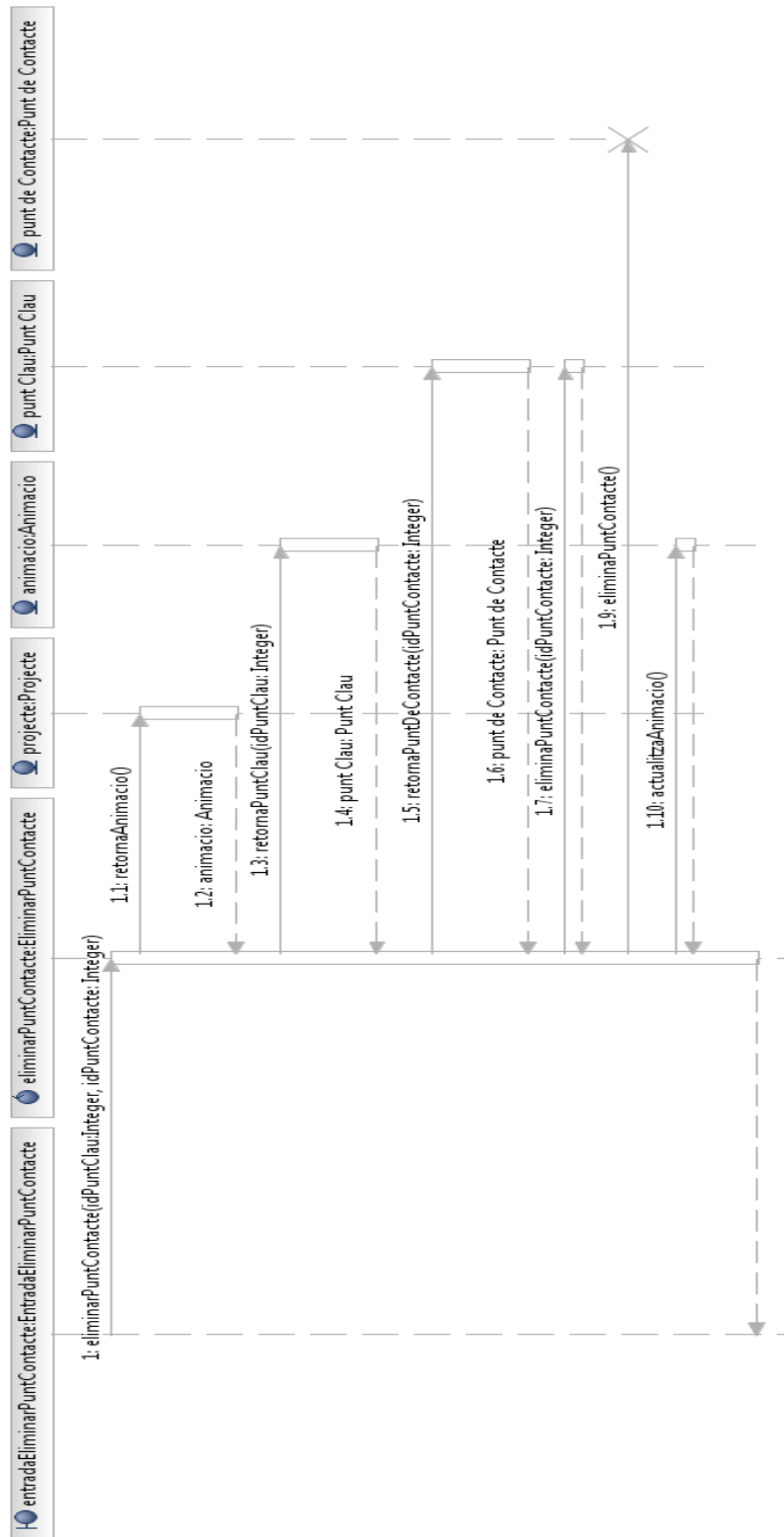


Figura 4.83: Diagrama de seqüència de Eliminar punt de contacte

4.2.38 Cas d'ús 6.4: Modificar punt de contacte

VOPC

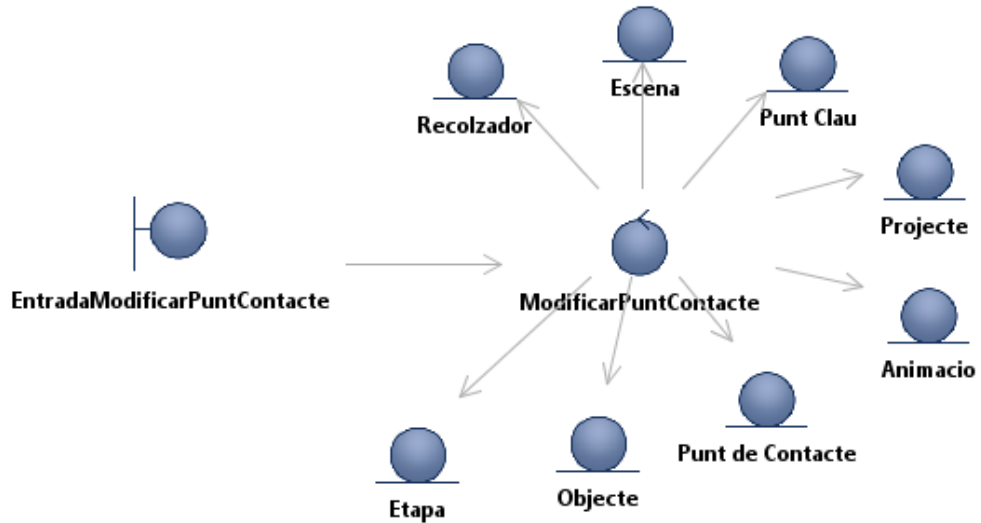


Figura 4.84: VOPC de Modificar punt de contacte

Diagrama de seqüència

Modificar el punt de contacte vol dir la posició o l'objecte al qual se li referia anteriorment, així doncs, caldrà actualitzar la informació com mostra el següent diagrama:

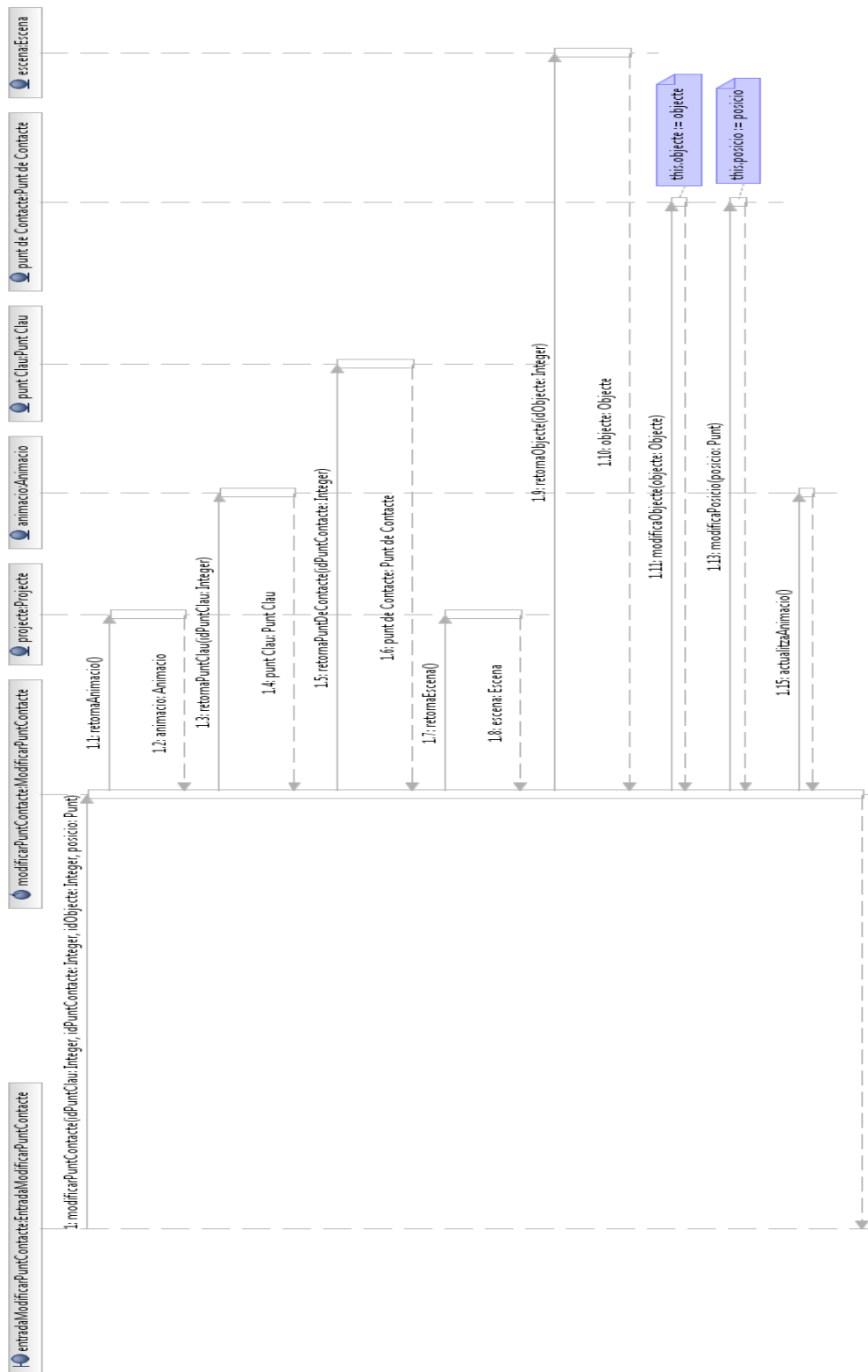


Figura 4.85: Diagrama de seqüència de Modificar punt de contacte

4.2.39 Cas d'ús 6.5: Modificar fotograma de l'animació

VOPC

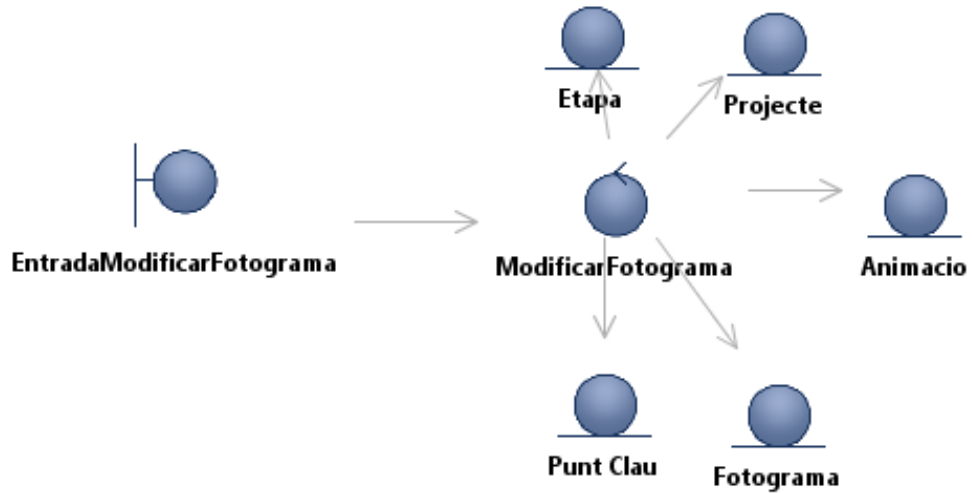


Figura 4.86: VOPC de Modificar fotograma de l'animació

Diagrama de seqüència

Una vegada l'usuari hagi generat una animació, aquest podrà modificar el fotograma que vulgui, la idea és que en el moment que això es faci, aquest fotograma passarà a ser un punt clau i l'usuari haurà de re-calculer l'animació. Com mostra el diagrama que ve a continuació:

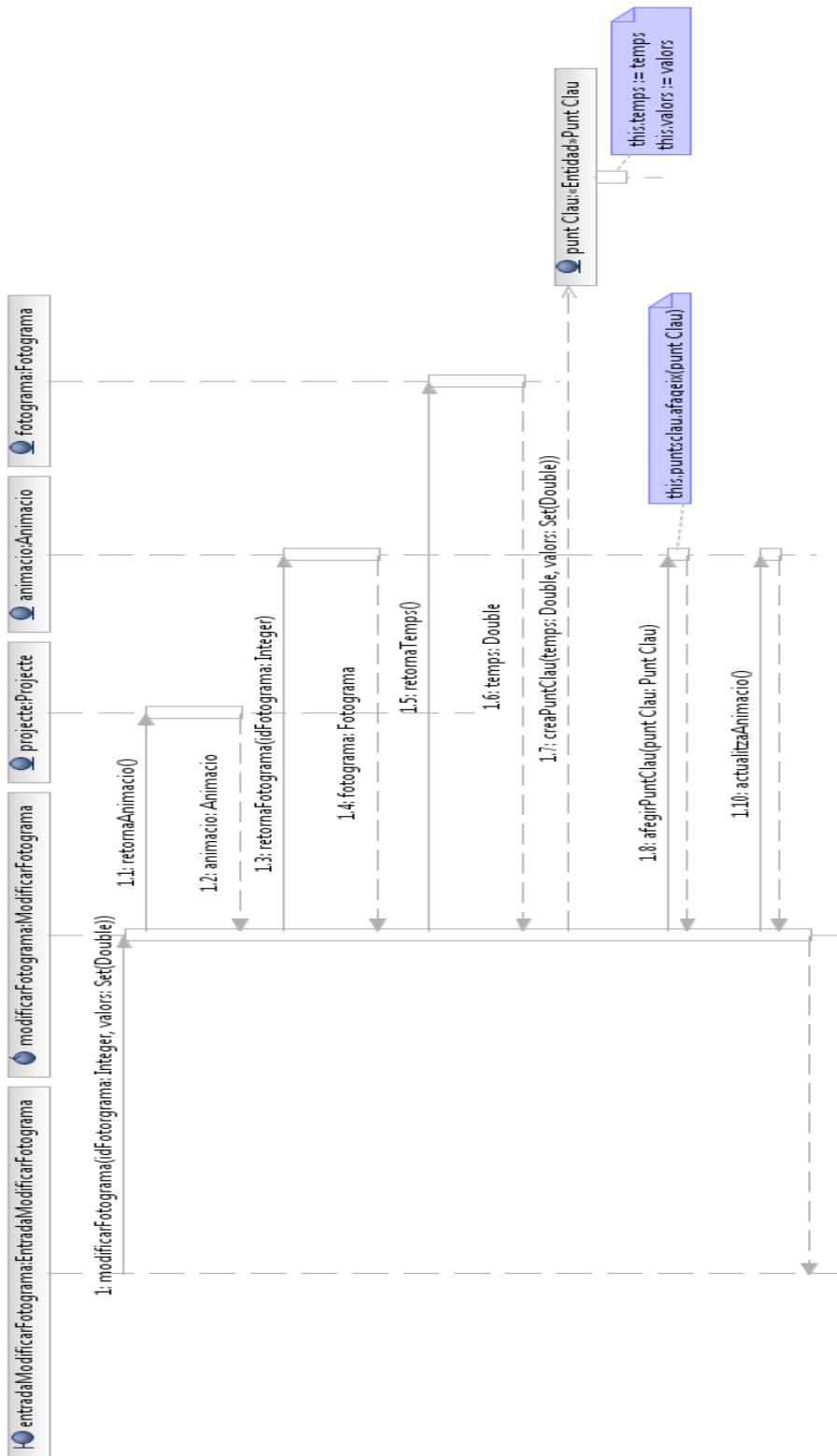


Figura 4.87: Diagrama de seqüència de Modificar fotograma de l'animació

Cas d'ús 6.6: Modificar paràmetre d'una etapa

VOPC

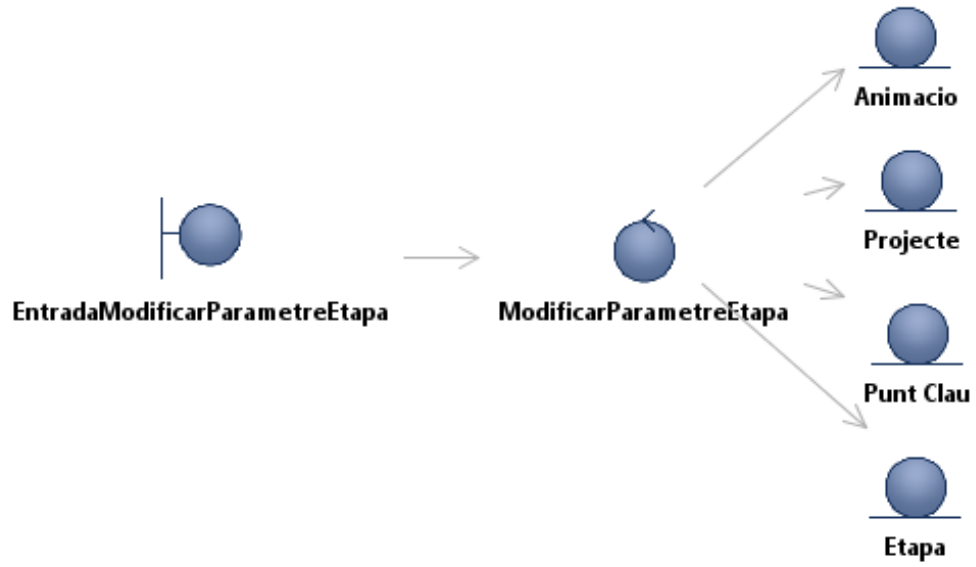


Figura 4.88: VOPC de Modificar paràmetre d'una etapa

Diagrama de seqüència

A mesura que l'usuari vagi afegint Punts Clau, el sistema anirà re-calculant les possibles etapes que conté l'animació descrita. A mesura que vagin apareguen etapes, l'usuari tindrà la capacitat de modificar certs paràmetres que descriuen una etapa. En aquesta secció es descriu el comportament intern que tindrà el sistema:

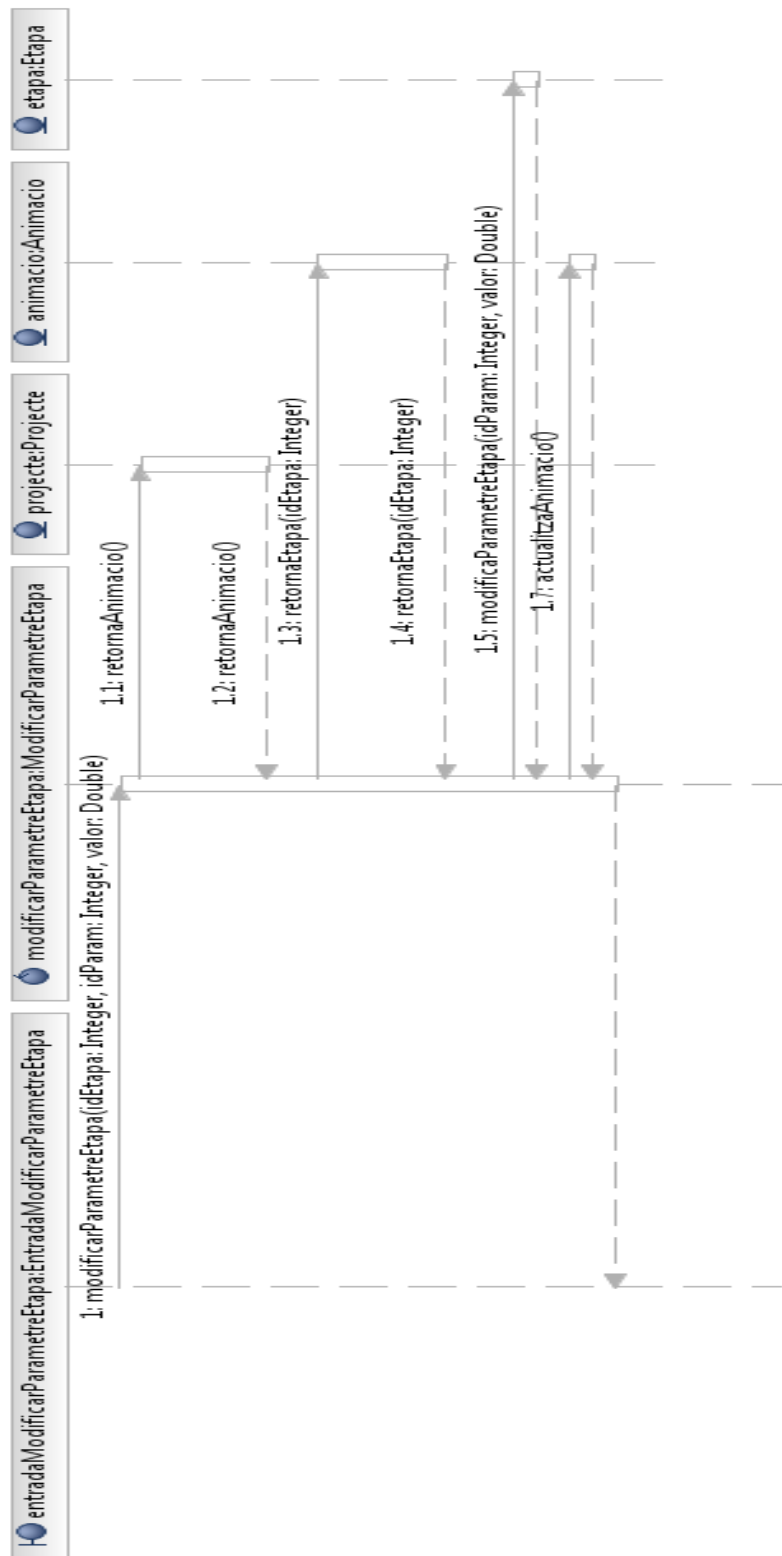


Figura 4.89: Diagrama de seqüència de Modificar paràmetre d'una etapa

Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau**VOPC**

De la mateixa manera que l'usuari pot modificar els valors del model, en un Punt Clau que és una captura d'una certa configuració del mateix model, l'usuari ha de poder modificar tots aquests paràmetres i que siguin consistents, a continuació es pot veure quines classes actuen i com ho fan:

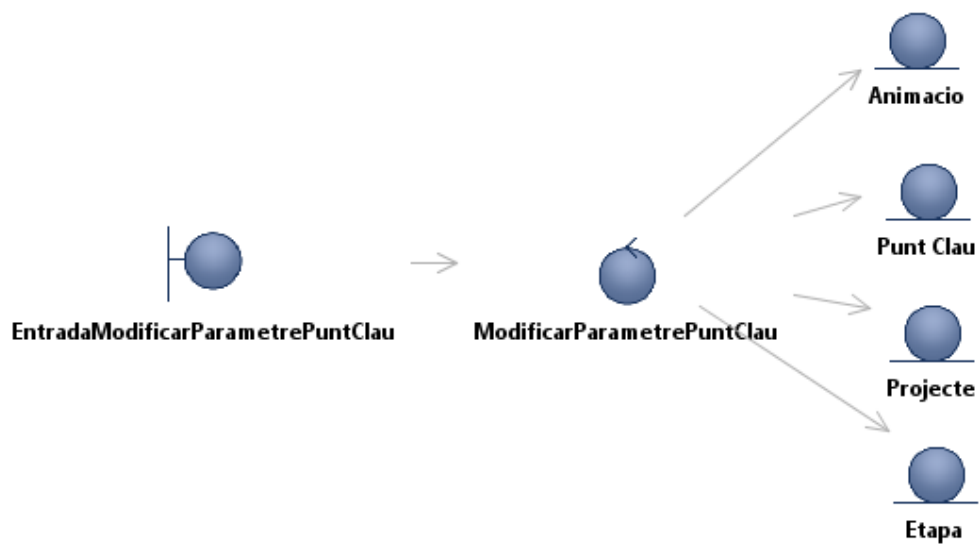


Figura 4.90: VOPC de Modificar paràmetre d'un Punt Clau

Diagrama de seqüència

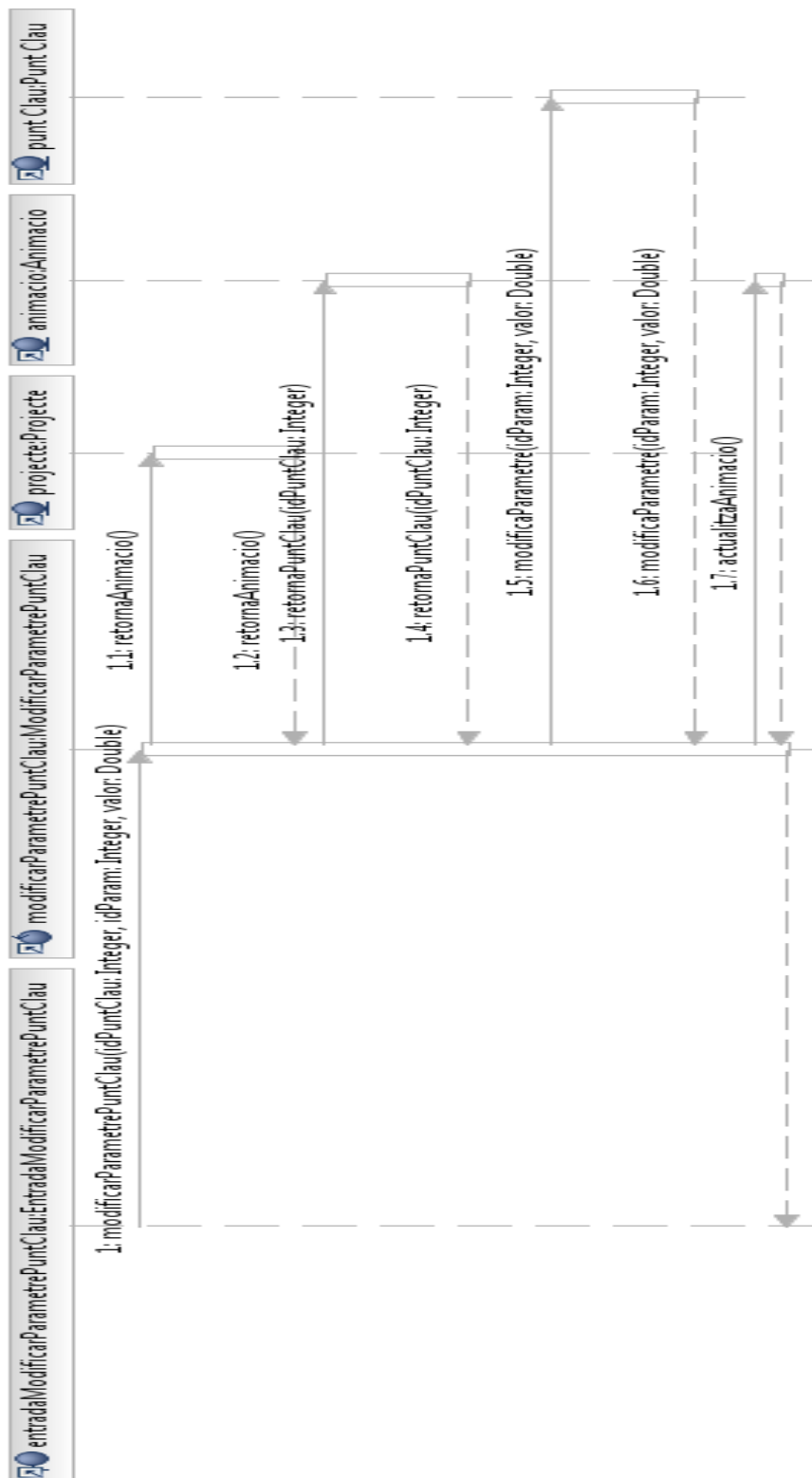


Figura 4.91: Diagrama de seqüència de Modificar paràmetre d'un Punt Clau

4.2.40 Cas d'ús 7.1: Pan

VOPC

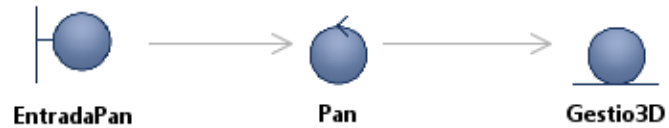


Figura 4.92: VOPC de Pan

Diagrama de seqüència

L'únic que es fa en aquest cas, serà modificar paràmetres de la càmera que vindrà definida en el mòdul de gestió 3D. En aquest cas el paràmetre que es modifica és el de posició del punt de referència de la càmera.

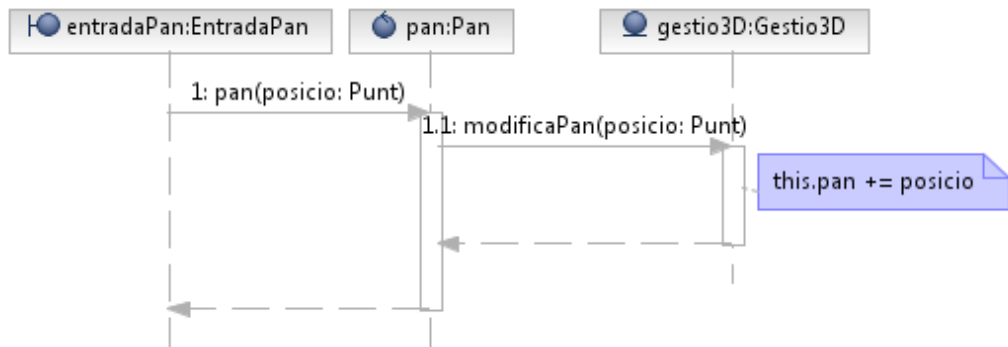


Figura 4.93: Diagrama de seqüència de Pan

4.2.41 Cas d'ús 7.2: Zoom

VOPC

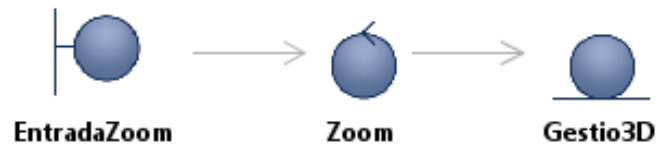


Figura 4.94: VOPC de Zoom

Diagrama de seqüència

En aquest cas el que es modifica és l'angle de visió de la càmera.

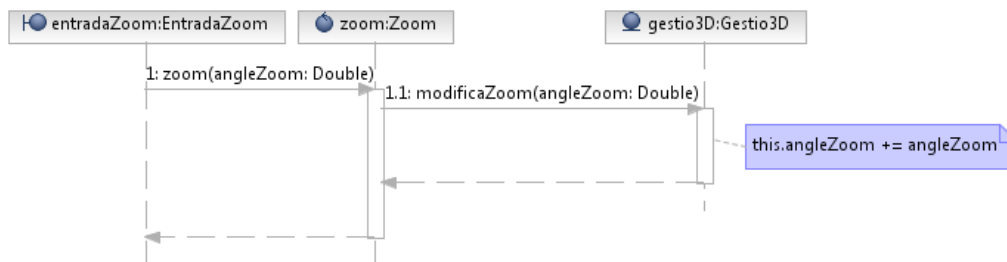


Figura 4.95: Diagrama de seqüència de Zoom

4.2.42 Cas d'ús 7.3: Rotació

VOPC

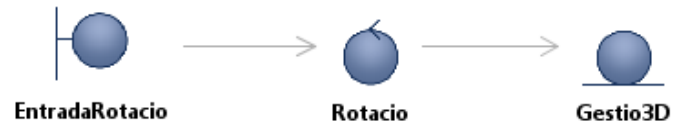


Figura 4.96: VOPC de Rotació

Diagrama de seqüència

Finalment en aquest última cas d'ús referent a la gestió 3D es modifica l'orientació de la càmera.

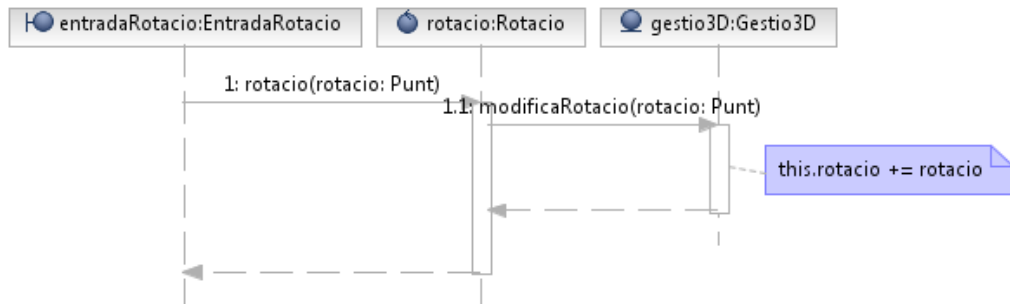


Figura 4.97: Diagrama de seqüència de Rotació

4.2.43 Cas d'ús 8.1: Guardar model

VOPC

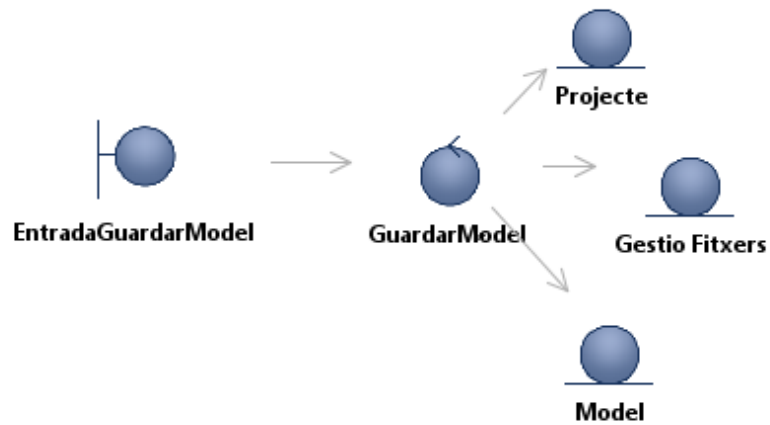


Figura 4.98: VOPC de Guardar model

Diagrama de seqüència

Una vegada es té el nom i la direcció del model i de quin model estem parlant, es pot guardar un model com es mostra a continuació:

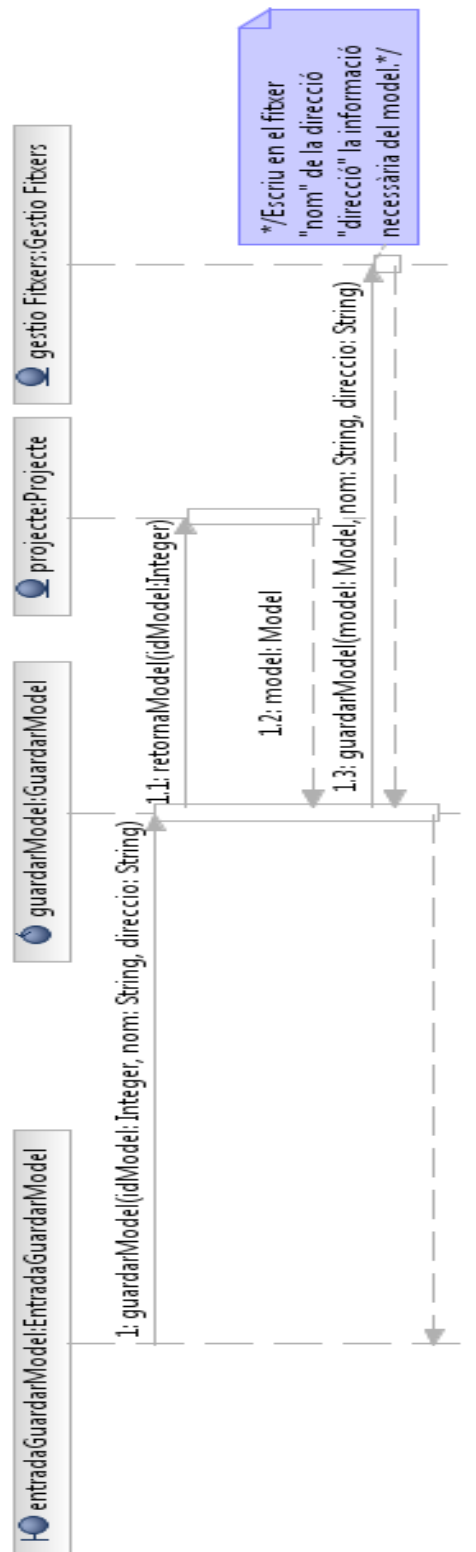


Figura 4.99: Diagrama de seqüència de Guardar model

4.2.44 Cas d'ús 8.2: Carregar model

VOPC

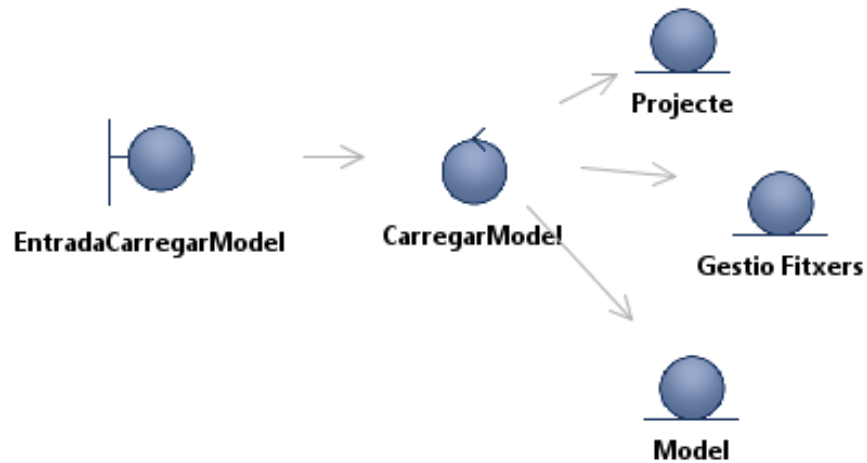


Figura 4.100: VOPC de Carregar model

Diagrama de seqüència

Quan es vol carregar un model, el que interessa és tenir el nom i la direcció del fitxer on està guardat el model, aleshores el sistema actuarà com es mostra a continuació:

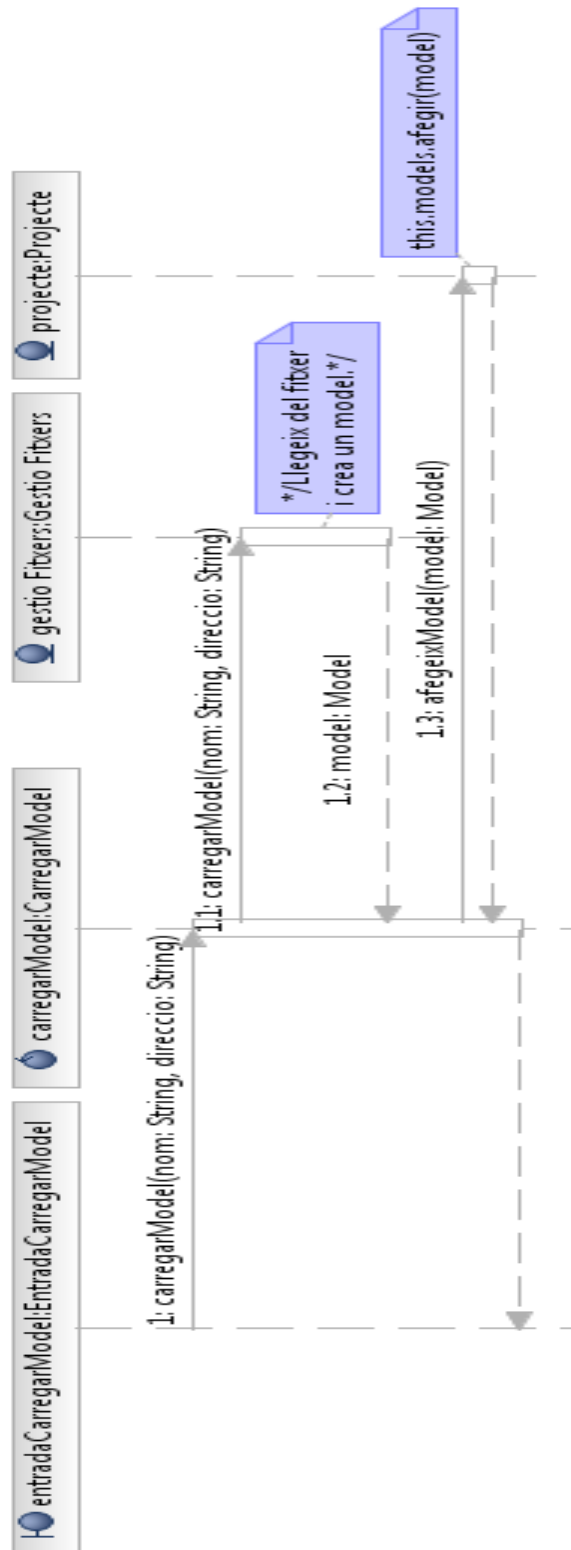


Figura 4.101: Diagrama de seqüència de Carregar model

4.2.45 Cas d'ús 8.3: Guardar escena

VOPC

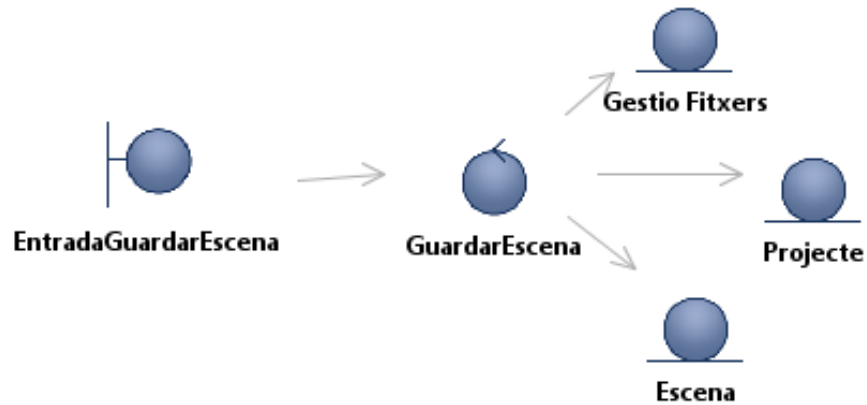


Figura 4.102: VOPC de Guardar escena

Diagrama de seqüència

És el mateix cas que el de guardar un model, però en aquest cas una escena, només cal el nom i la direcció on es vol guardar:

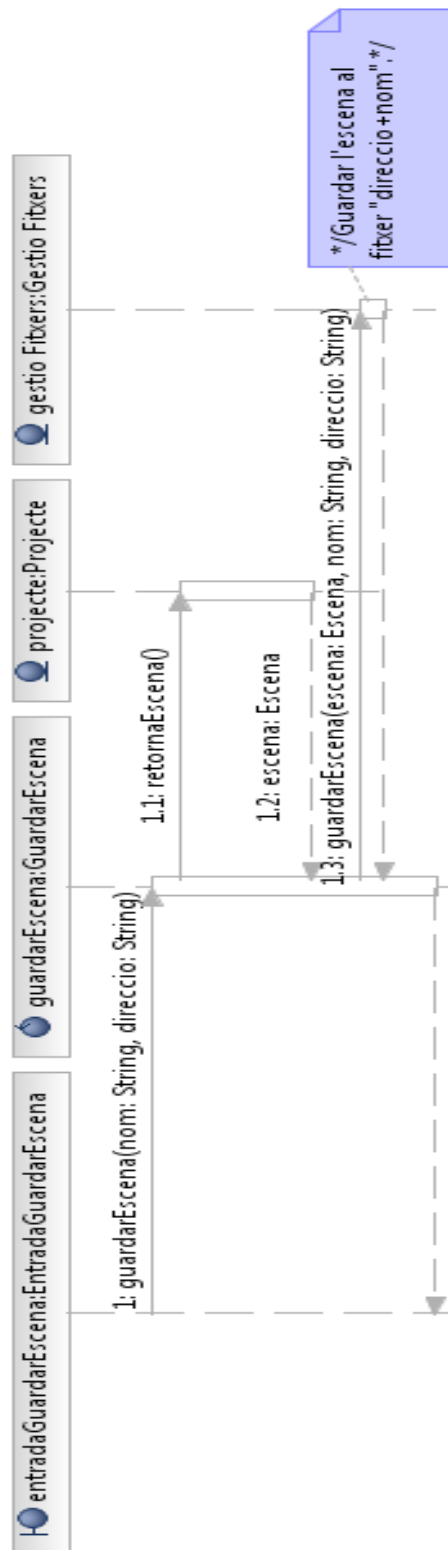


Figura 4.103: Diagrama de seqüència de Guardar escena

4.2.46 Cas d'ús 8.4: Carregar escena

VOPC

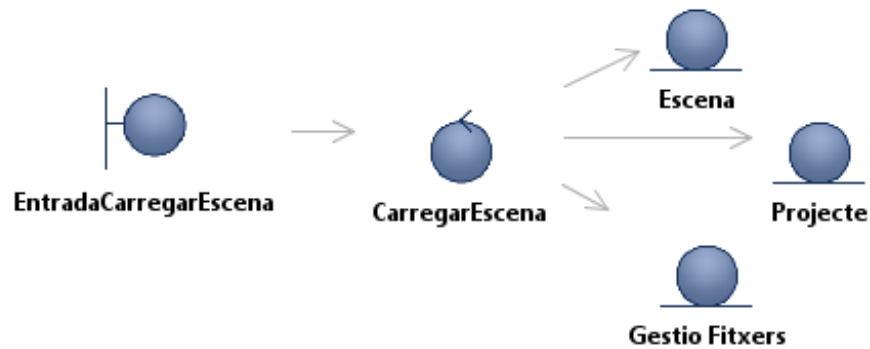


Figura 4.104: VOPC de Carregar escena

Diagrama de seqüència

En aquesta ocasió, carregar una escena no és el mateix que carregar un model, ja que d'escena només se'n té una i en el moment de carregar una escena substitueix a la que hi havia. En el diagrama següent es veu reflectit:

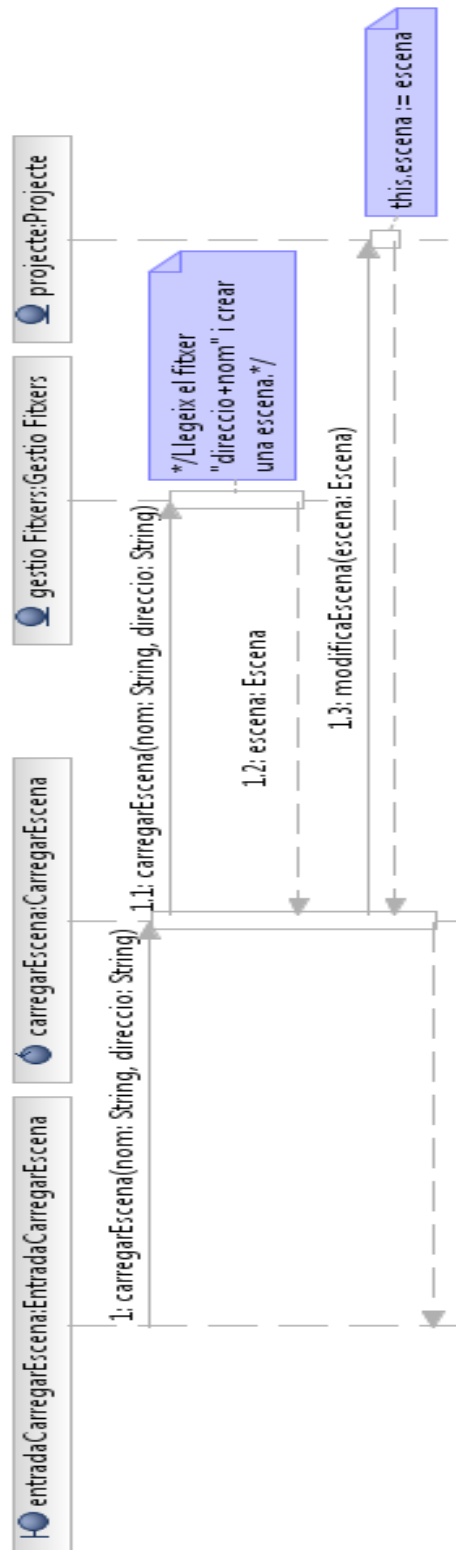


Figura 4.105: Diagrama de seqüència de Carregar escena

4.2.47 Cas d'ús 8.5: Crear projecte

VOPC

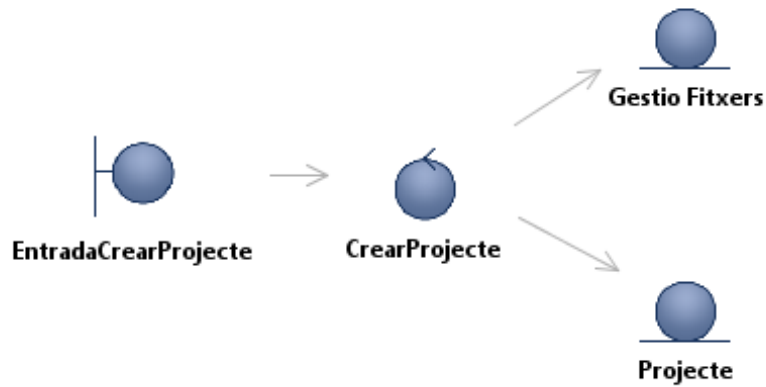


Figura 4.106: VOPC de Crear projecte

Diagrama de seqüència

En aquest cas, en el moment de crear un projecte substitueix al que ja hi havia al programa, per això, abans de borrar-lo cal guardar-lo i després carregar l'altre donant-li ja d'inici el nom i la direcció d'on es vol guardar. Com es mostra a la figura següent:

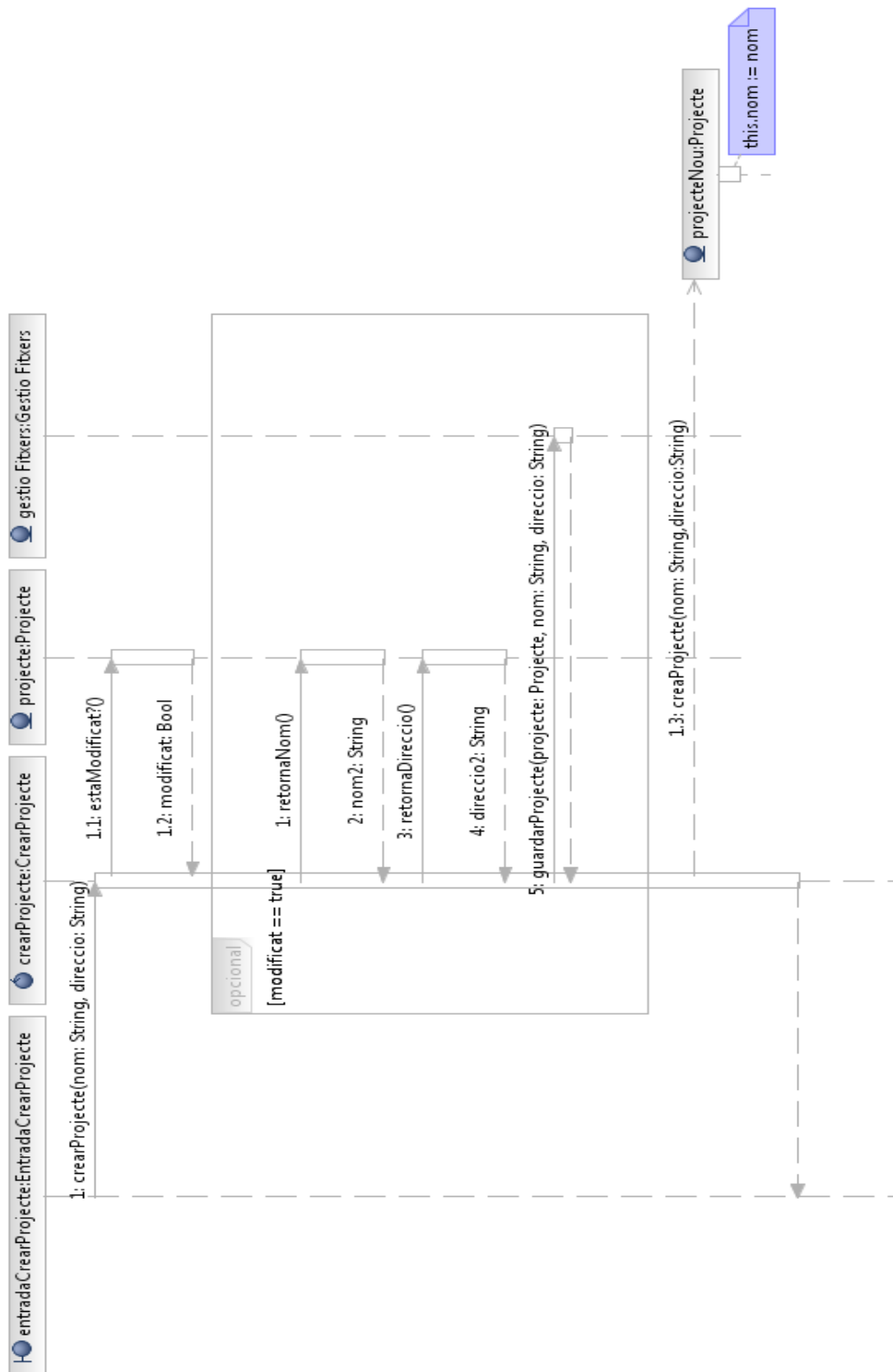


Figura 4.107: Diagrama de seqüència de Crear projecte

4.2.48 Cas d'ús 8.6: Guardar projecte

VOPC

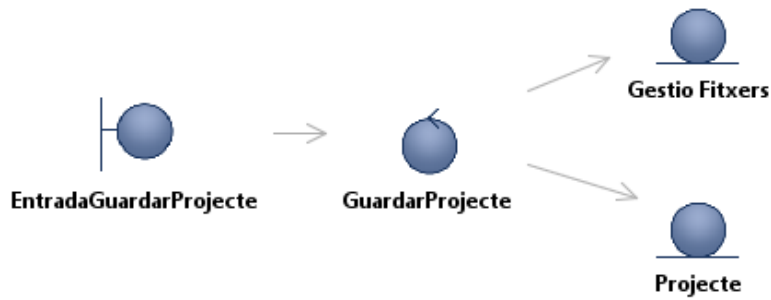


Figura 4.108: VOPC de Guardar projecte

Diagrama de seqüència

Senzillament cal donar el nom i la direcció del projecte i guardar-lo on s'escaigui. El diagrama següent ens ho mostra:

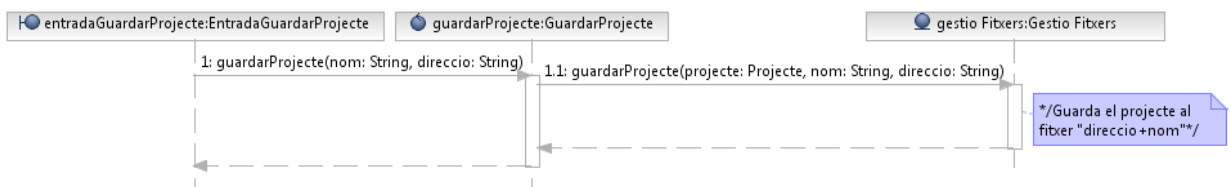


Figura 4.109: Diagrama de seqüència de Guardar projecte

4.2.49 Cas d'ús 8.7: Carregar projecte

VOPC

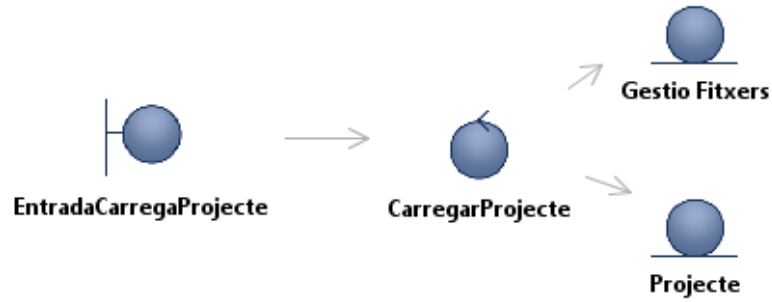


Figura 4.110: VOPC de Carregar projecte

Diagrama de seqüència

De la mateixa manera que es crea un projecte, en el moment en que es carrega, el projecte que hi havia es veu substituït pel nou projecte, per tant, cal guardar-lo si està modificat. Com en el diagrama següent:

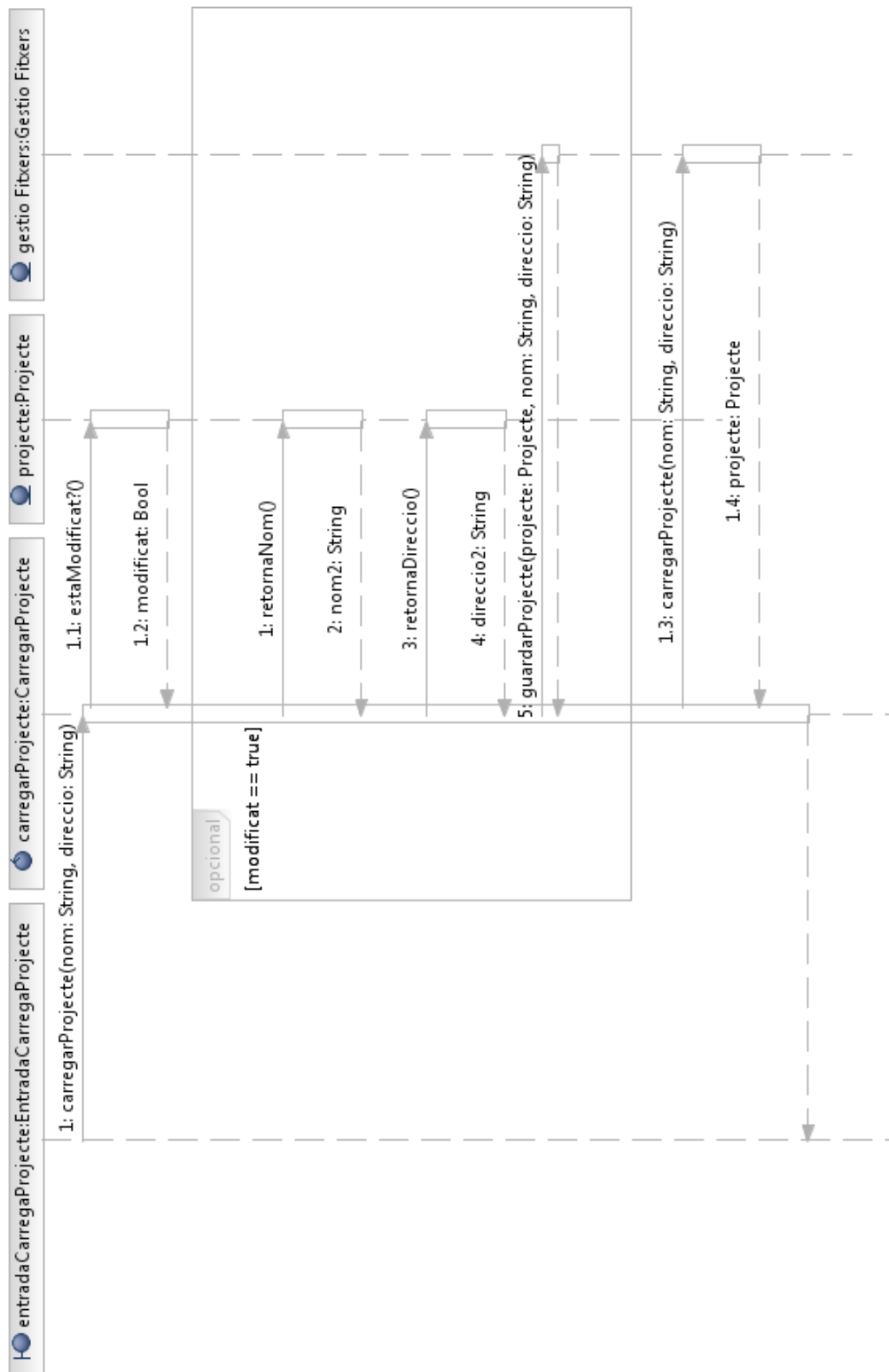


Figura 4.111: Diagrama de seqüència de Carregar projecte

4.2.50 Cas d'ús 8.8: Afegir gràfica

VOPC

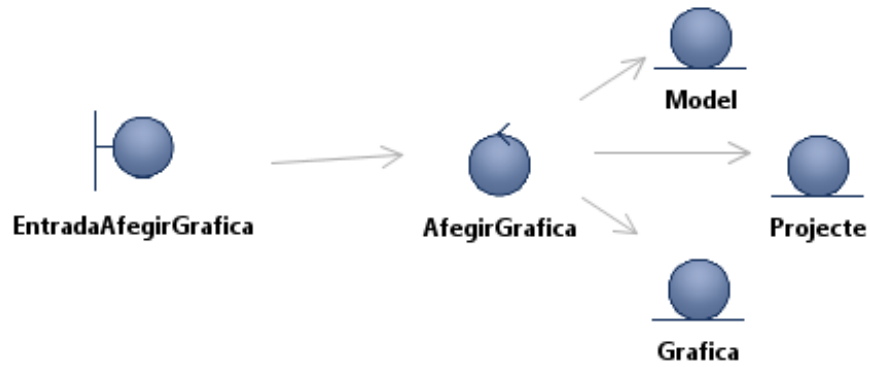


Figura 4.112: VOPC de Afegir gràfica

Diagrama de seqüència

Afegir una gràfica que doni informació a l'usuari implica que aquest a de dir de quin paràmetre es vol que es faci la gràfica, com un nom per distingir-les de la resta. El diagrama de seqüència següent reflecteix el que fa aquest cas d'ús:

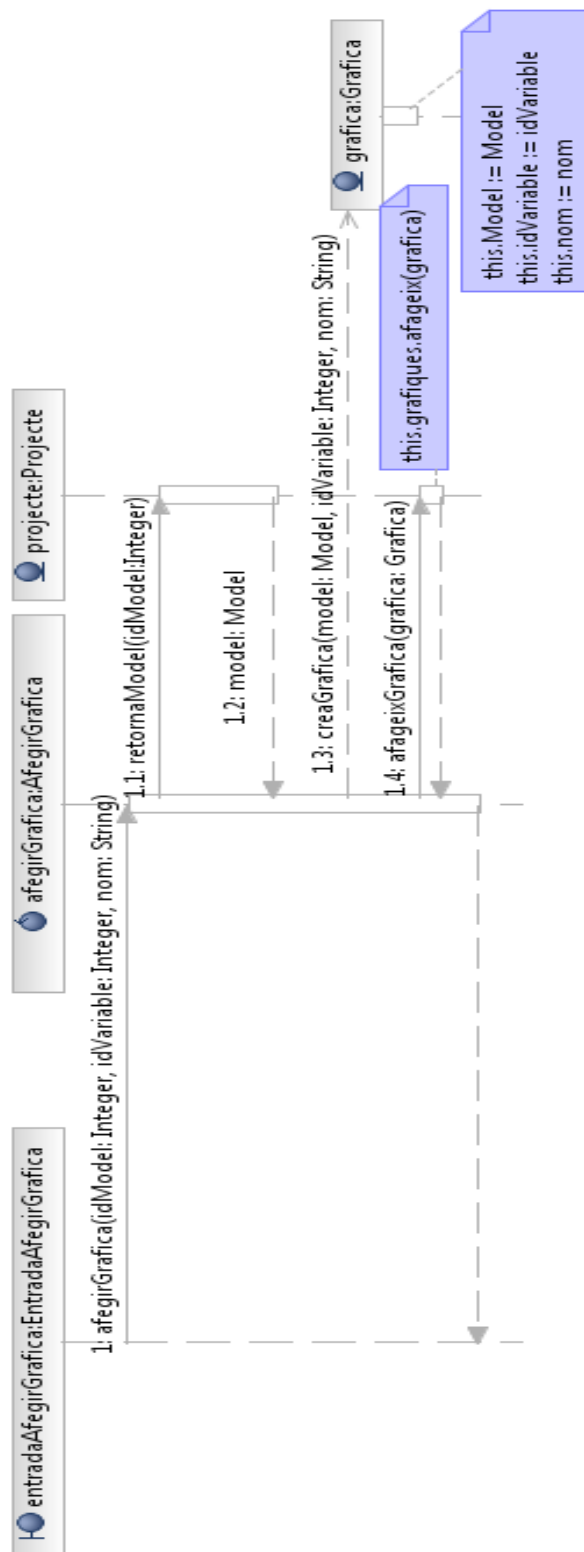


Figura 4.113: Diagrama de seqüència de Afegir gràfica

4.2.51 Cas d'ús 8.9: Eliminar gràfica

VOPC

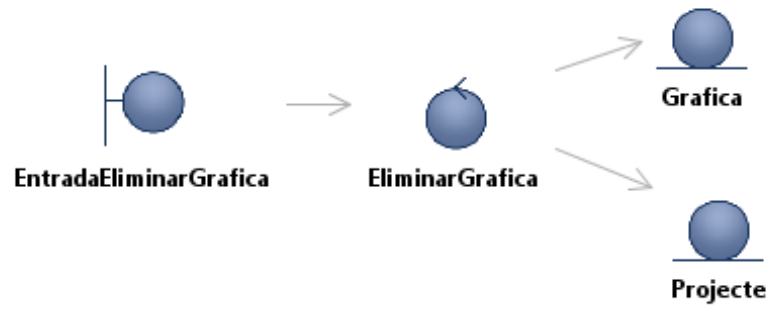


Figura 4.114: VOPC de Eliminar gràfica

Diagrama de seqüència

El sistema només coneixen l'identificador de la gràfica pot determinar quina es vol eliminar: Així doncs:

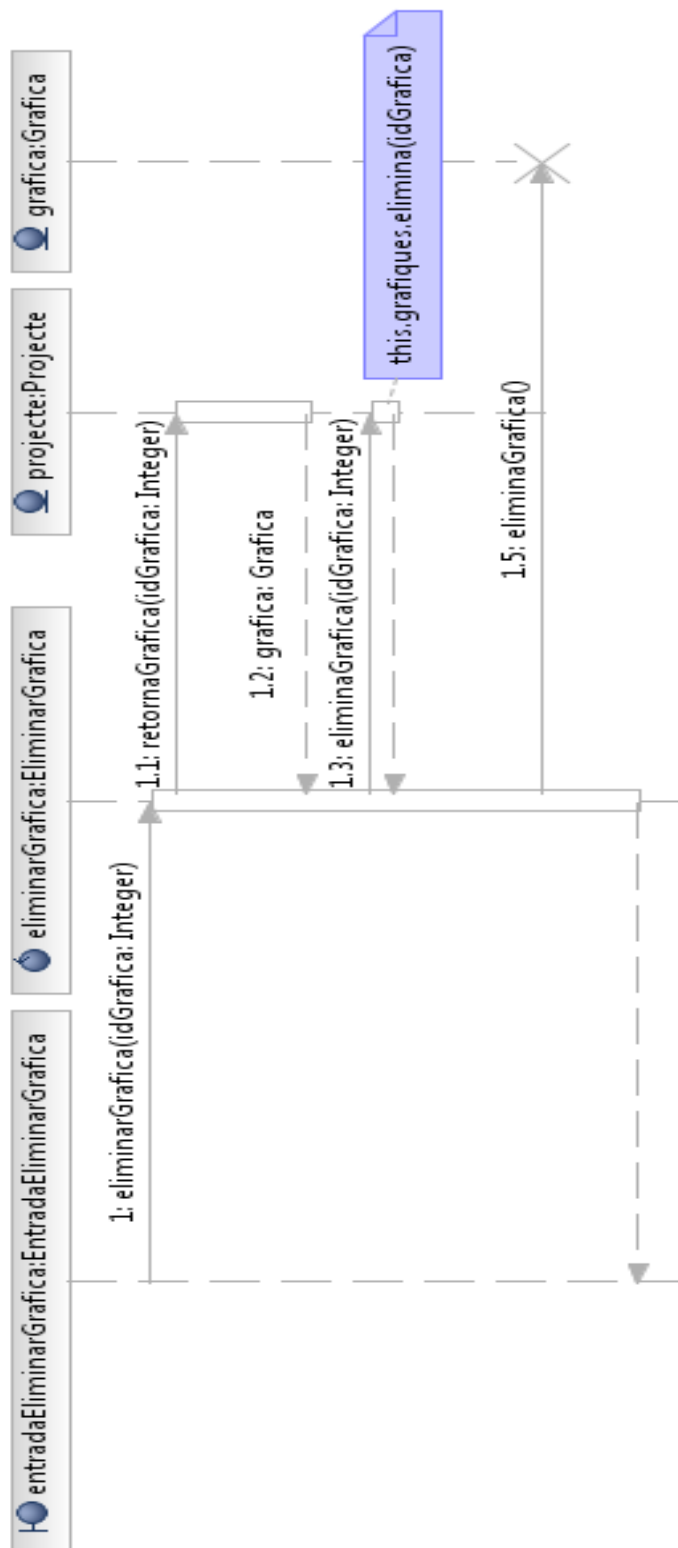


Figura 4.115: Diagrama de seqüència de Eliminar gràfica

4.2.52 Cas d'ús 8.10: Visualitzar gràfica

VOPC

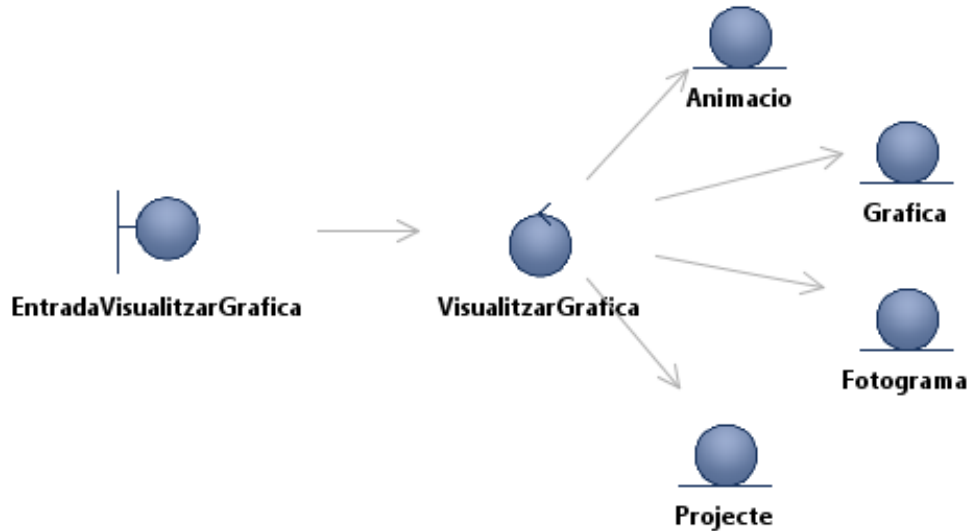


Figura 4.116: VOPC de Visualitzar gràfica

Diagrama de seqüència

L'usuari una vegada ha generat l'animació, només haurà de dir al sistema quina gràfica vol veure i automàticament la generarà a partir de la informació de cada un dels fotogrames que ha calculat el sistema. La manera concreta de com ho va es veu en el següent diagrama:

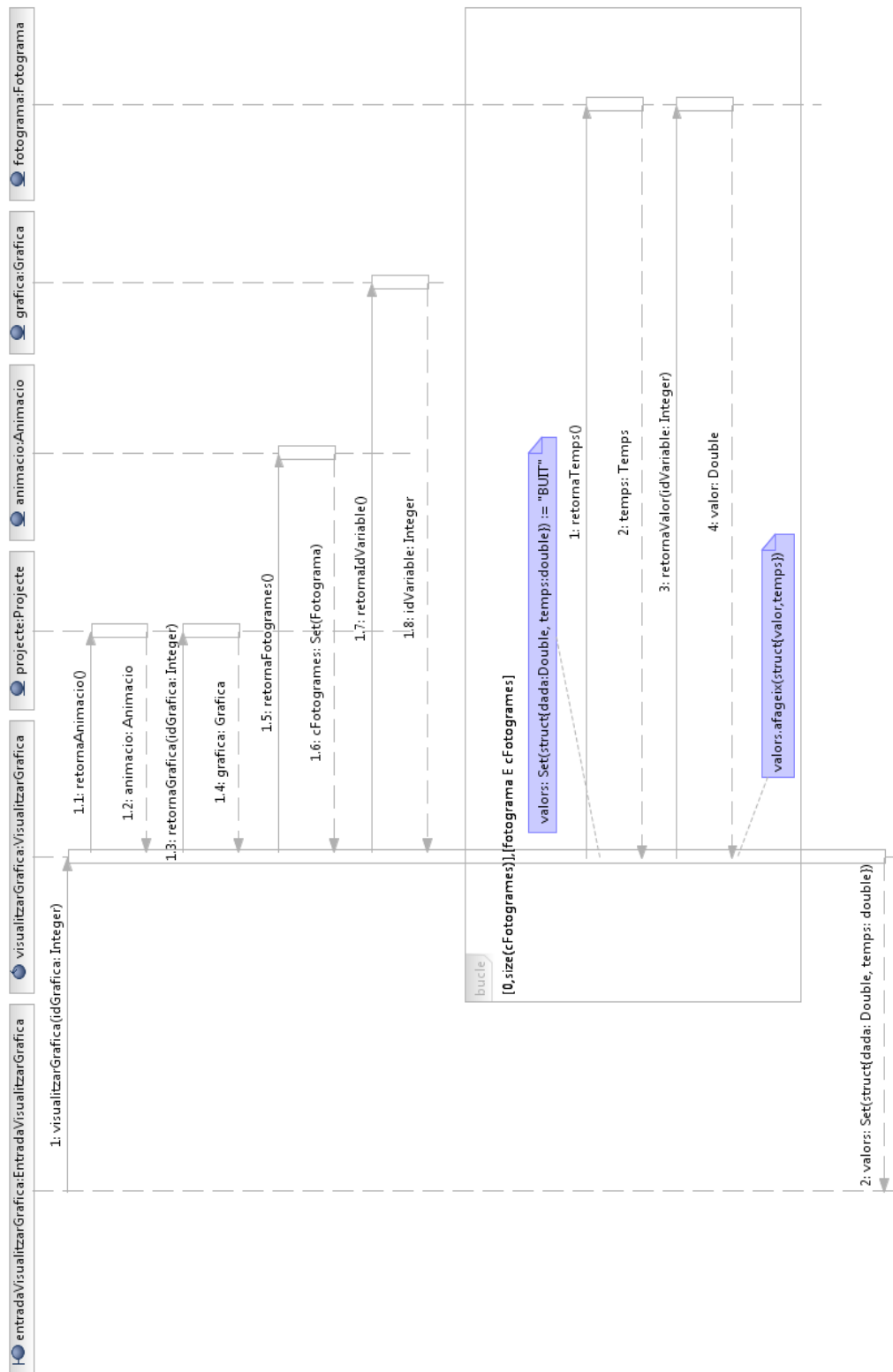


Figura 4.117: Diagrama de seqüència de Visualitzar gràfica

Capítol 5

Model de Disseny

5.1 Introducció

En aquesta part, s'especificarà el comportament intern de l'aplicació de cada una de les capes que es varen descriure en el capítol 1, la capa de presentació, la de domini i la de dades. S'aplicaran un conjunt de patrons de disseny per explicar millor el comportament de la nostra aplicació i poder complir els requisits no funcionals, com l'escalabilitat i robustesa de l'aplicació. Finalment, s'especificarà l'aparença externa de l'aplicació mostrant com tindrà distribuïdes les funcionalitats i la informació que podrà consultar i editar l'usuari.

5.2 Model de classes normalitzat

En aquest apartat es presenta el model conceptual normalitzat, aquest model representa la majoria de classes de domini que tenen a veure amb la informació que es necessita i les classes per obtenir la informació que s'espera, però pel que fa a les classes que s'encarreguen de cada un dels cassos d'ús queda encara veure la seva especificació:

5.3 Aplicació de patrons

5.3.1 Arquitectura de l'aplicació

En el capítol 1 es va explicar quina arquitectura s'usaria i de manera explícita ha estat així en els diagrames de seqüència i els VOPC's presentats, ara només cal especificar per cada una de les capes del patró arquitectònic de tres capes quins altres patrons s'han aplicat per acabar de descriure la nostra aplicació. Per recordar una mica quines capes eren es presenten a continuació:

- **Capa de presentació:** En aquesta capa, es capturen les accions de l'usuari, recullen la informació que introdueixi i mostra la informació que es té entrada al sistema.
- **Capa de domini:** S'encarrega de tenir tota la informació el màxim de consistent possible i de mantenir l'estat en un moment donat del programa. Es reben les peticions de l'usuari i s'encarrega de calcular les respostes a aquestes peticions.
- **Capa de dades:** S'encarrega de la persistència. Guarda i carrega dades de fitxers.

5.3.2 Capa de presentació

En aquesta capa, s'han aplicat els següents patrons:

- El patró MVC (model-vista-controlador): La manera en com funcionarà aquest patró és el següent, la única vista que tindrà el sistema recollirà les dades en funció de quin cas d'ús es trobi. Una vegada finalitzat, el controlador de la capa de presentació actualitzarà el model i prepararà la vista per a la següent acció de l'usuari.
- Patró plantilla: Una plantilla controlador consisteix en crear una classe pare amb les operacions necessàries i els atributs, i un conjunt de classes que heretin aquestes funcions i atributs bàsics.
- Patró estat: Serveix per poder controlar el canvi d'estat d'una classe que pot ser de més d'un subtipus.

La següent figura mostra el model conceptual de la capa presentació:

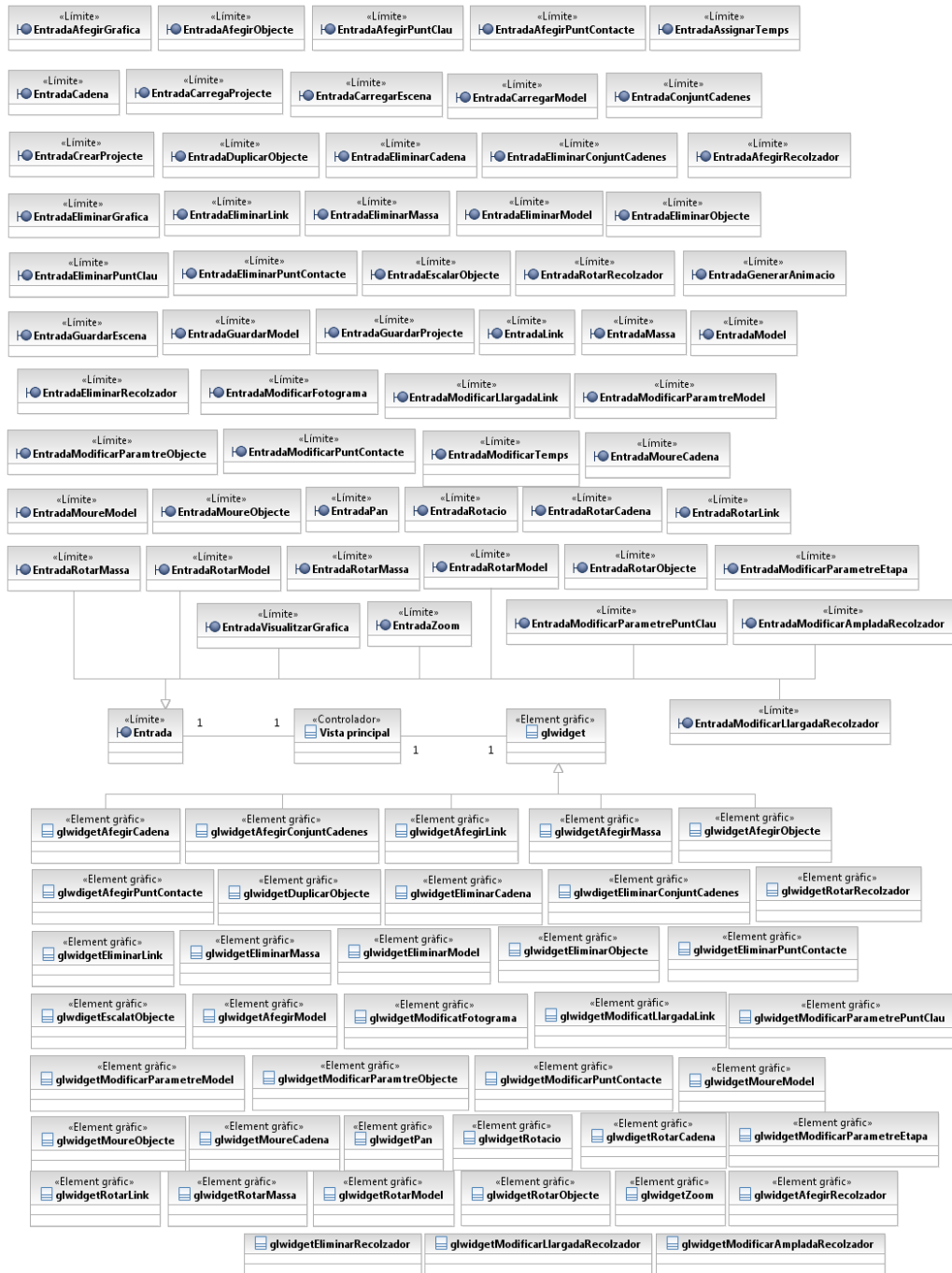


Figura 5.2: Model Conceptual de la capa de presentació

En aquest model, tenim una classe principal vista, que és la vista del programa. Té dues relacions, una el tipus de element gràfic que pertoca a cada cas d'ús, i el controlador per cada un dels cassos d'ús.

5.3.3 Capa de domini

A part del model conceptual normalitzat, cal afegir totes les classes que controlen cada cas d'ús i que es comunica amb la capa de presentació i la capa de gestió. En aquesta capa, els patrons aplicats són:

- **Patró estat**
- **Patró plantilla**
- **Patró controlador:** per cada cas d'ús, existeix un controlador que és l'encarregat de relacionar-se amb les classes que formen part del cas d'ús. Això evita crear dependències entre classes i per tenir les operacions més complexes en parts concretes del codi per tal de localitzar més fàcilment els errors.
- **Patró singleton:** la instanciació d'una classe concreta és d'un sol objecte.

A la següent figura es mostra les classes que intervenen en el control i el manegament de les classes del model conceptual de l'aplicació:

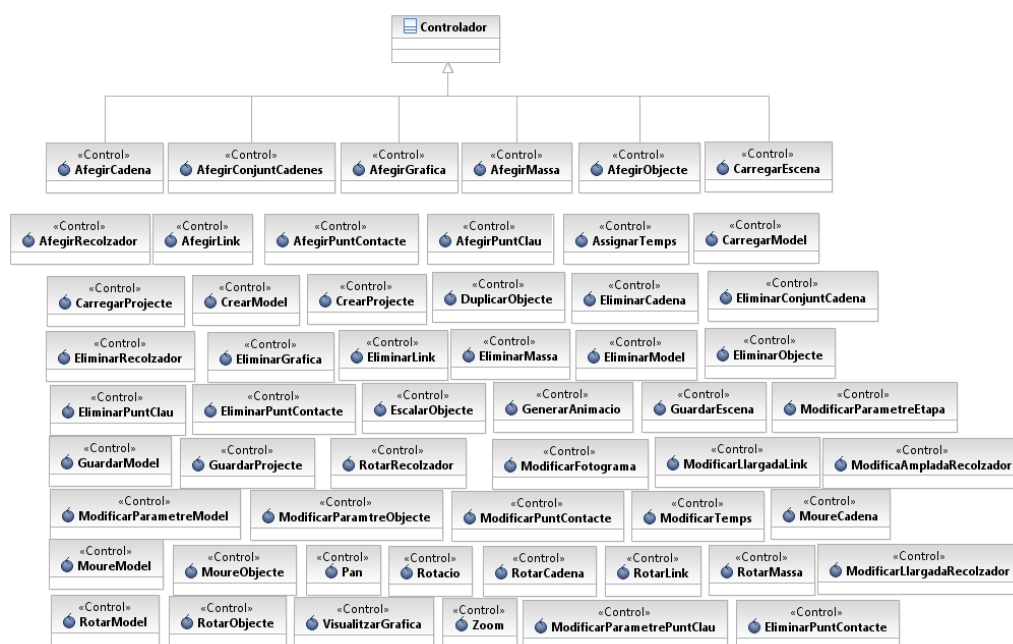


Figura 5.3: Model Conceptual de la capa de domini

Així doncs, tenim per cada un dels cassos d'ús descrits del nostre sistema una classe que s'encarrega de la gestió d'aquesta i de relacionar-se amb les classes que tenen i col·laboren en la generació de la informació referent a cada cas d'ús concret.

5.3.4 Capa de gestió de dades

En aquest cas, com que tenim poques classes que s'hagin de guardar o carregar i seran en moment puntuals, el sistema només tindrà una classe que s'encarregui de la gestió de fitxers en els quals crearà, escriurà i llegirà d'ells. Com s'ha especificat en el capítol anterior, només tenim tres classes les quals guardarem o carregarem dades: els projectes, les escenes i els models. Així doncs, l'únic patró s'ha aplicat és el **Singleton**.



Figura 5.4: Model Conceptual de la capa de gestió de dades

5.4 Diagrames de seqüència modificats

5.4.1 Cas d'ús 1.1: Crear Model

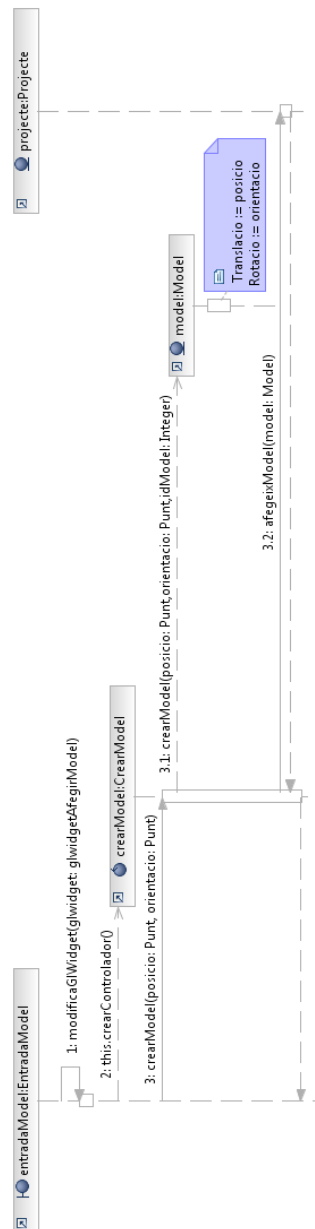


Figura 5.5: Diagrama de seqüència de Crear Model

5.4.2 Cas d'ús 1.2: Afegir "Conjunt de cadenes"

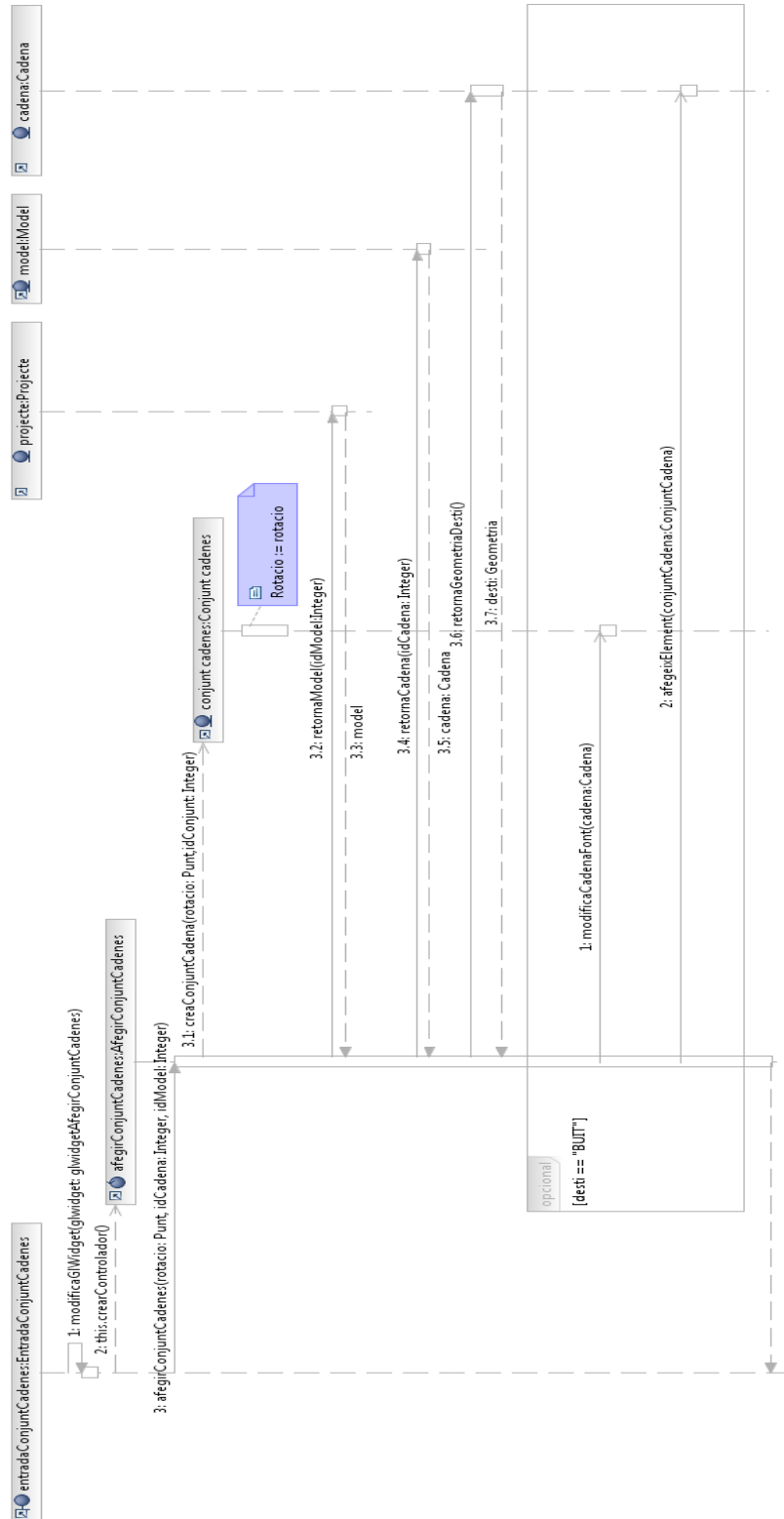


Figura 5.6: Diagrama de seqüència de Afegir "Conjunt de cadenes"

5.4.3 Cas d'ús 1.3: Afegir cadena al model

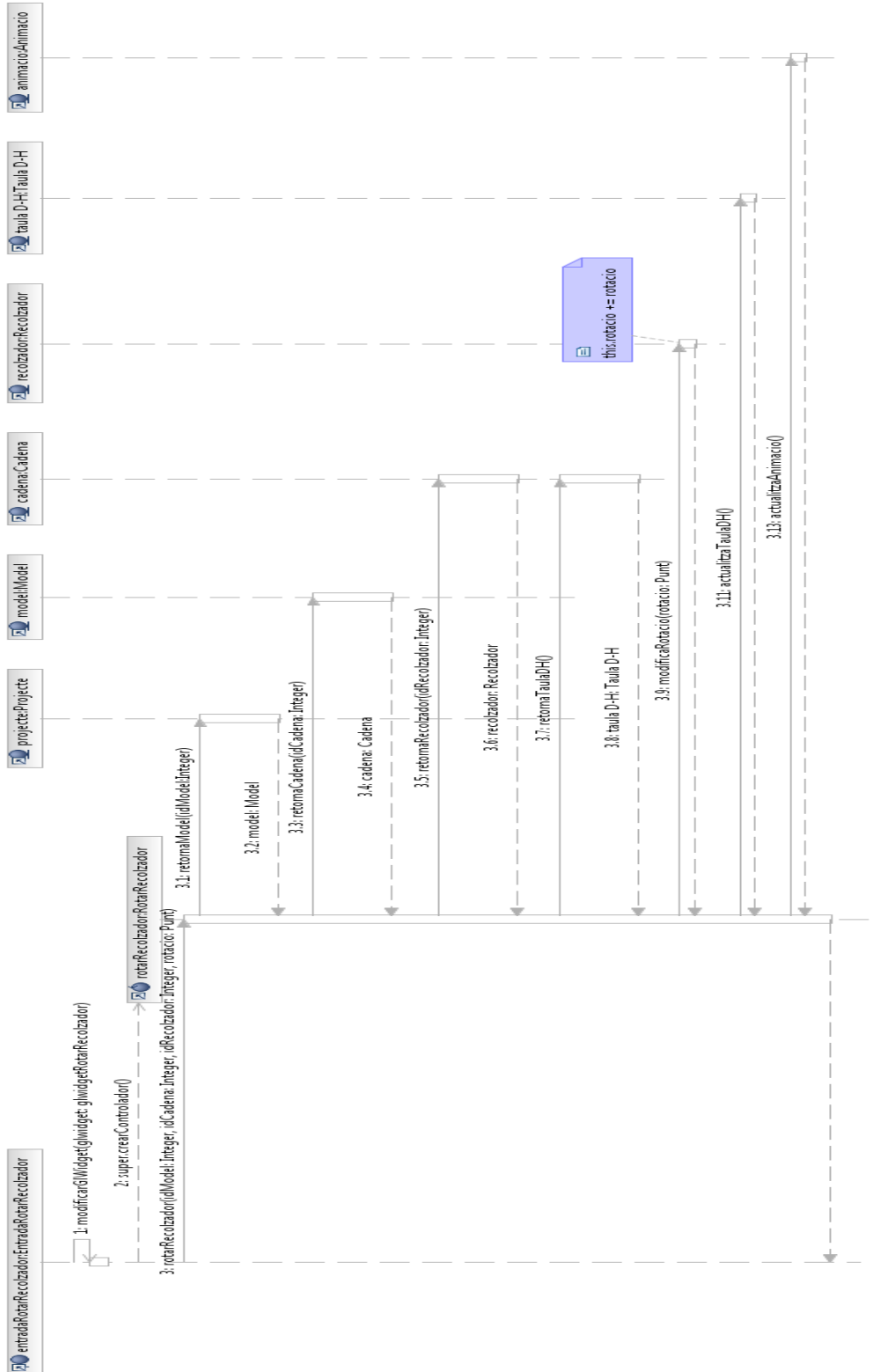


Figura 5.7: Diagrama de seqüència de Afegir cadena al model

5.4.4 Cas d'ús 1.4: Afegir un "Link" al model

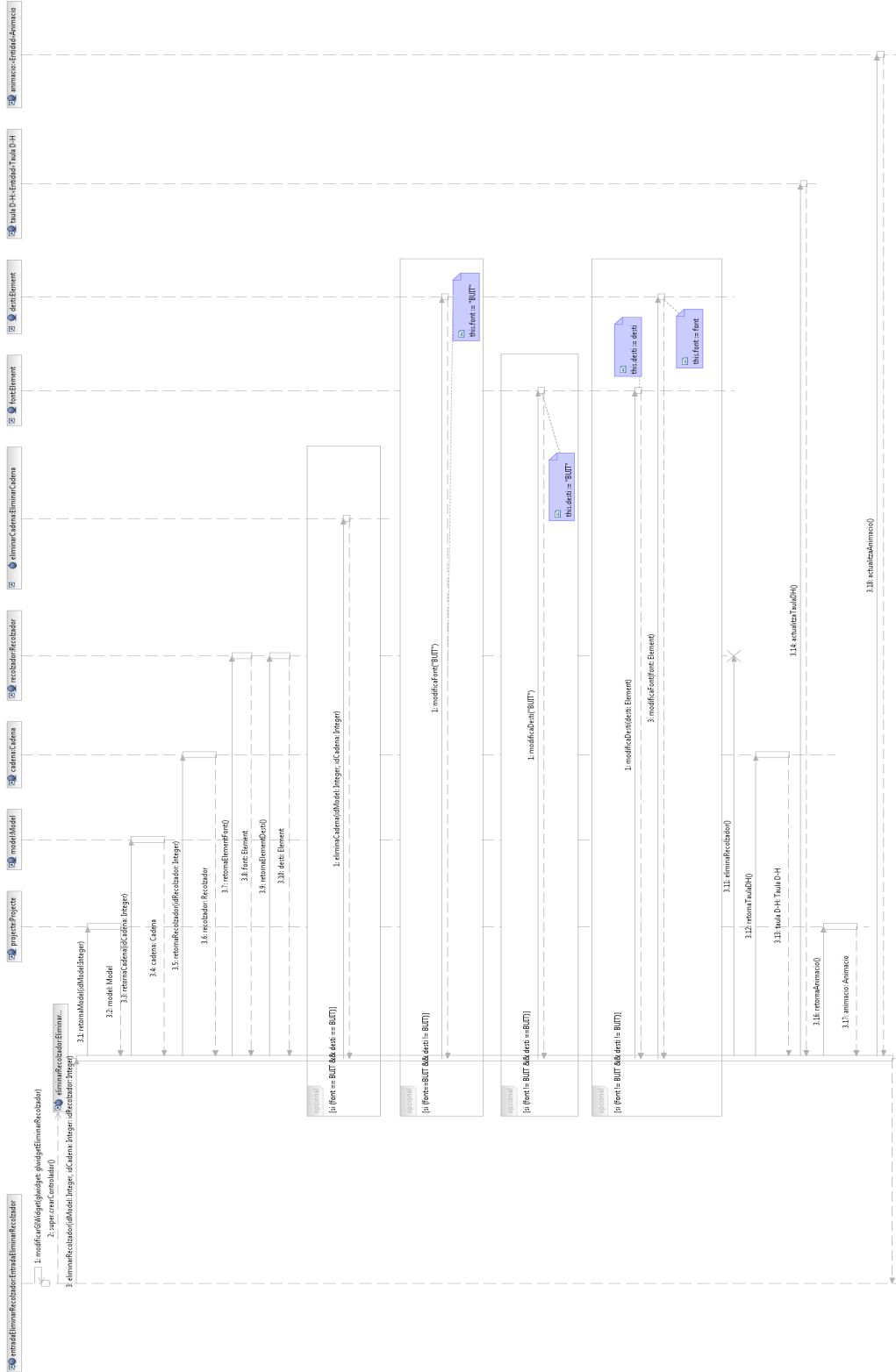


Figura 5.8: Diagrama de seqüència de Afegir un "Link" al model

5.4.5 Cas d'ús 1.5: Afegir una Massa al model

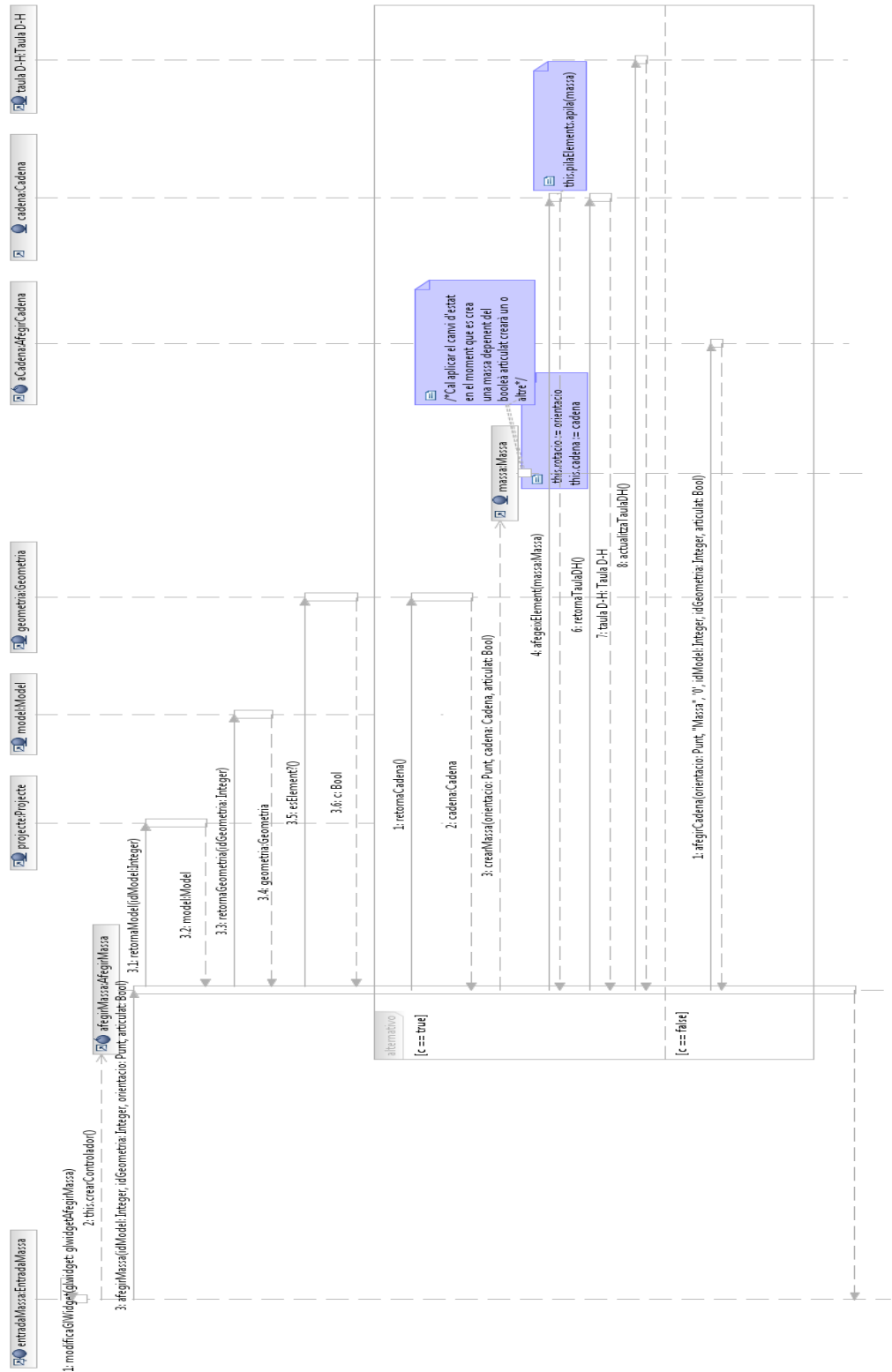


Figura 5.9: Diagrama de seqüència de Afegir una Massa al model

5.4.6 Cas d'ús 1.6: Afegir un Recolzador al model

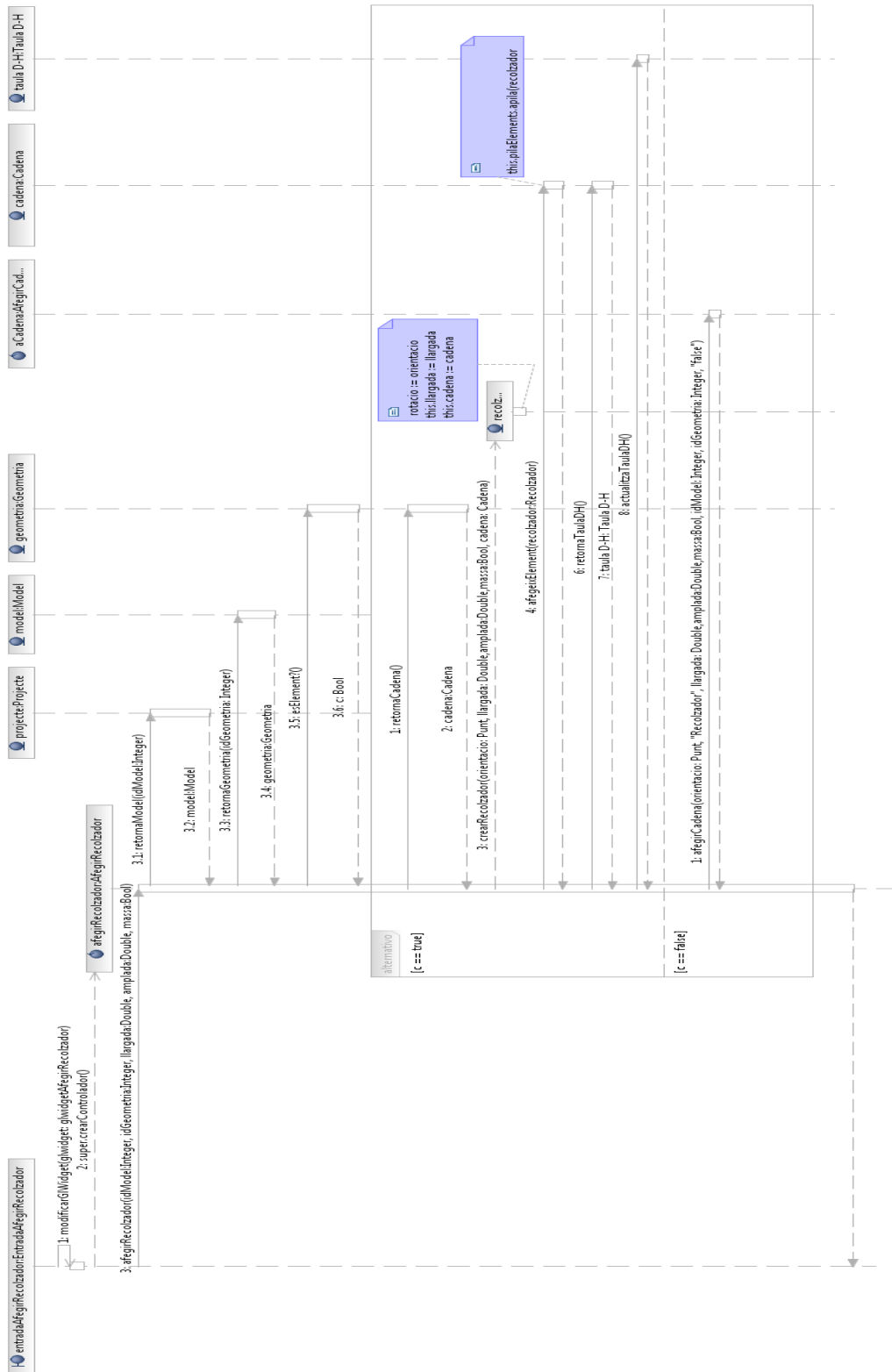


Figura 5.10: Diagrama de seqüència de Afegir un Recolzador al model

5.4.7 Cas d'ús 2.1: Eliminar model

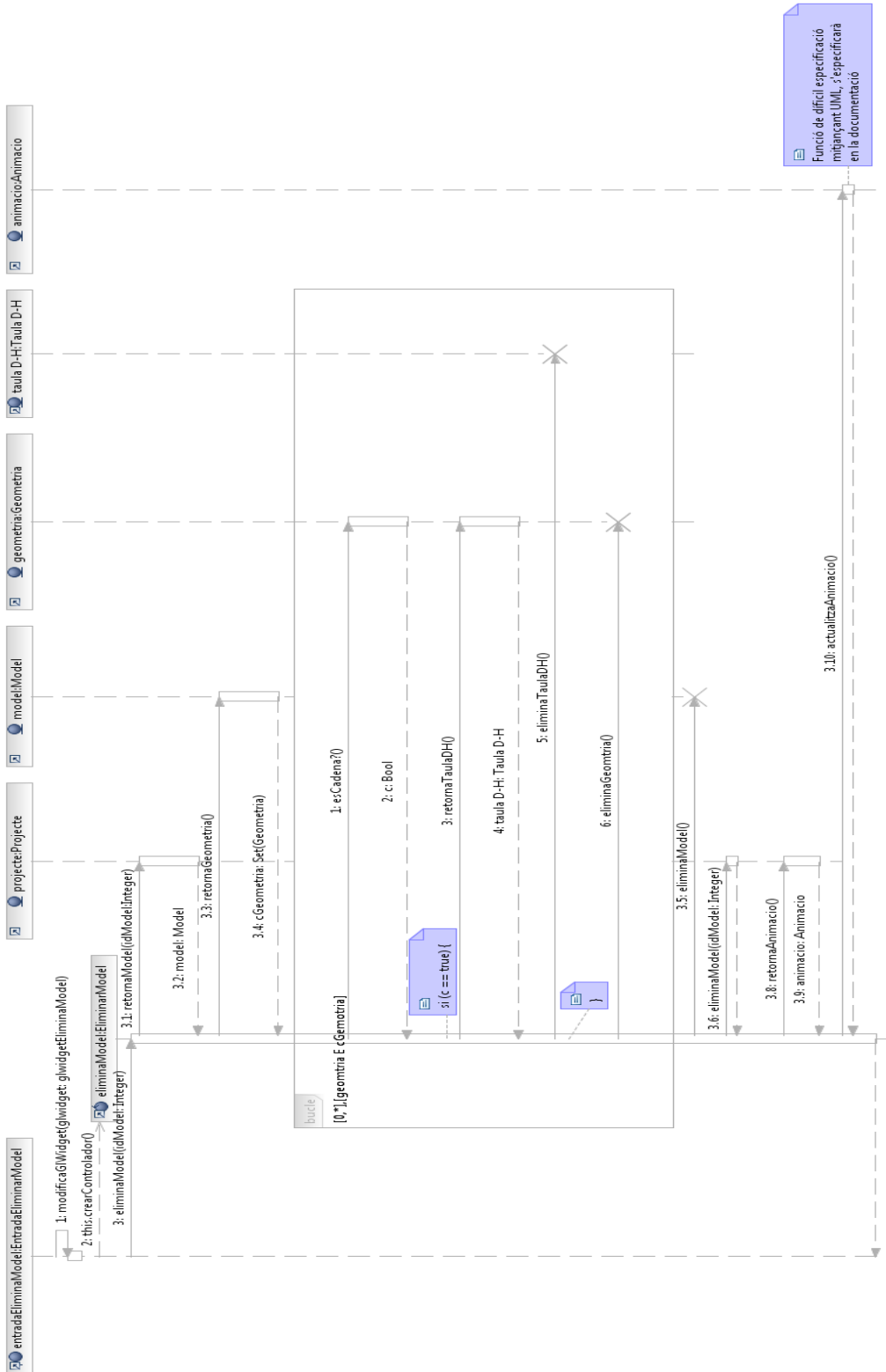


Figura 5.11: Diagrama de seqüència de Eliminar model

5.4.8 Cas d'ús 2.2: Moure model

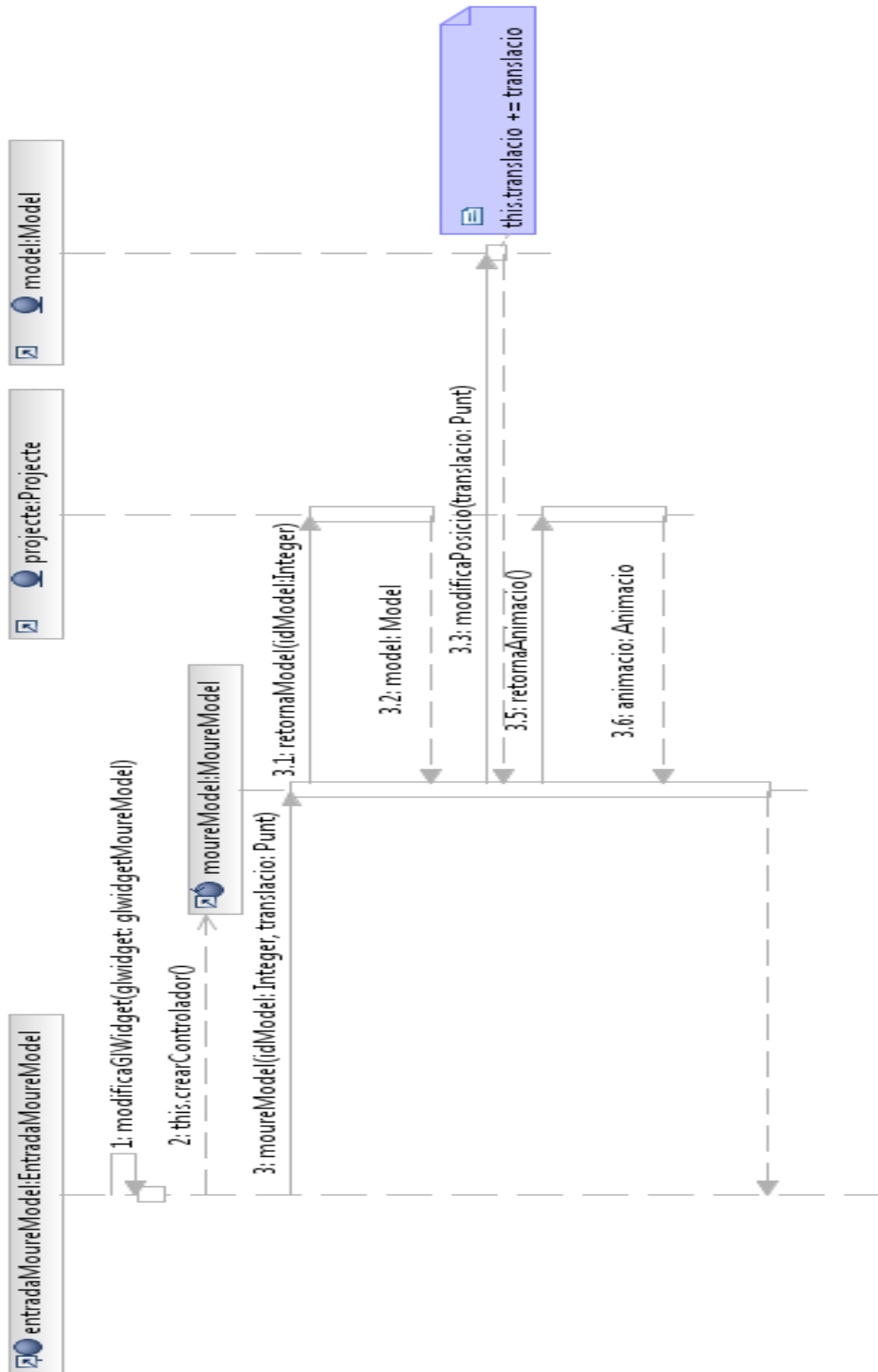


Figura 5.12: Diagrama de seqüència de Moure model

5.4.9 Cas d'ús 2.3: Rotar model

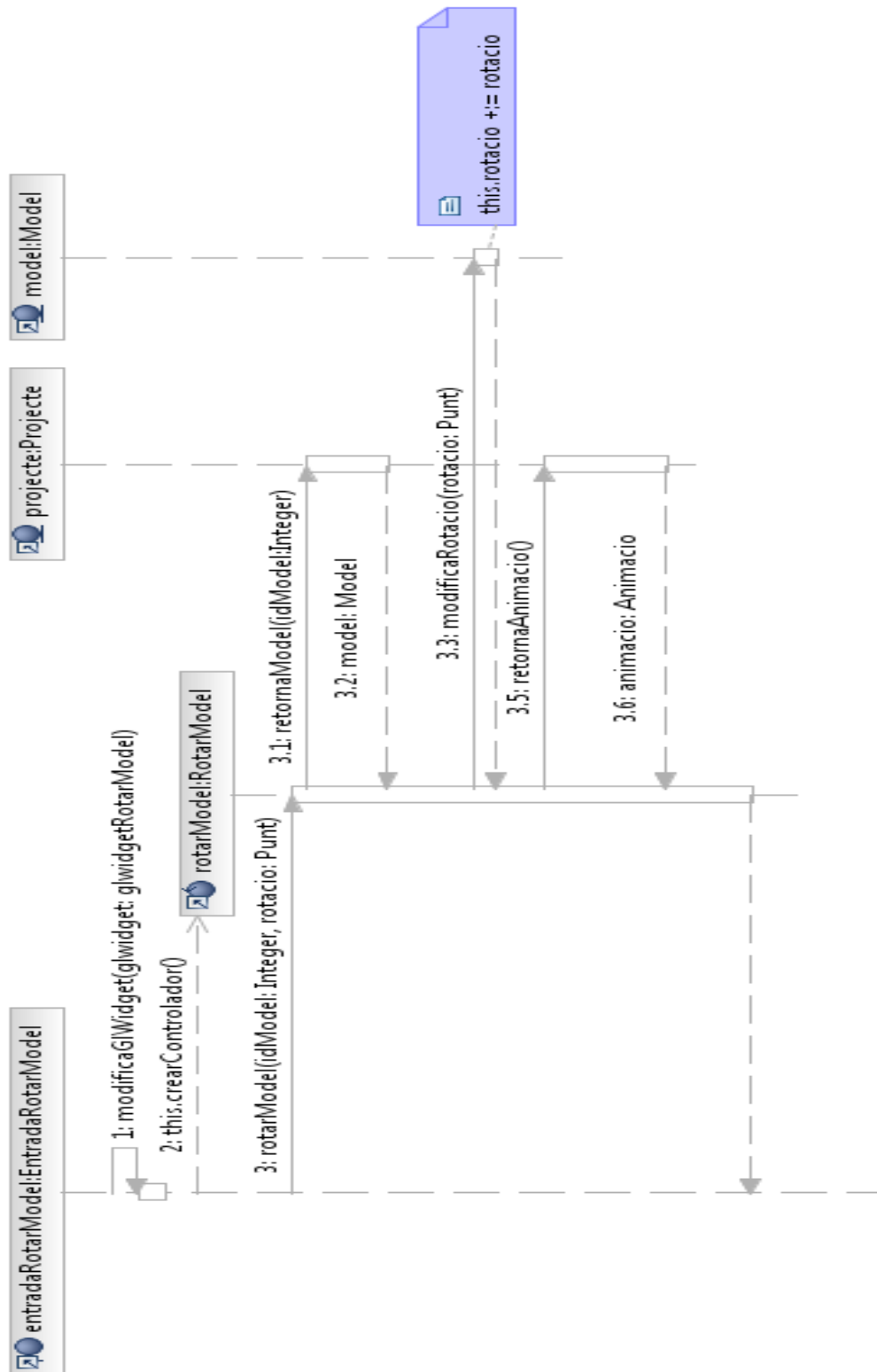


Figura 5.13: Diagrama de seqüència de Rotar model

5.4.10 Cas d'ús 2.4: Eliminar cadena

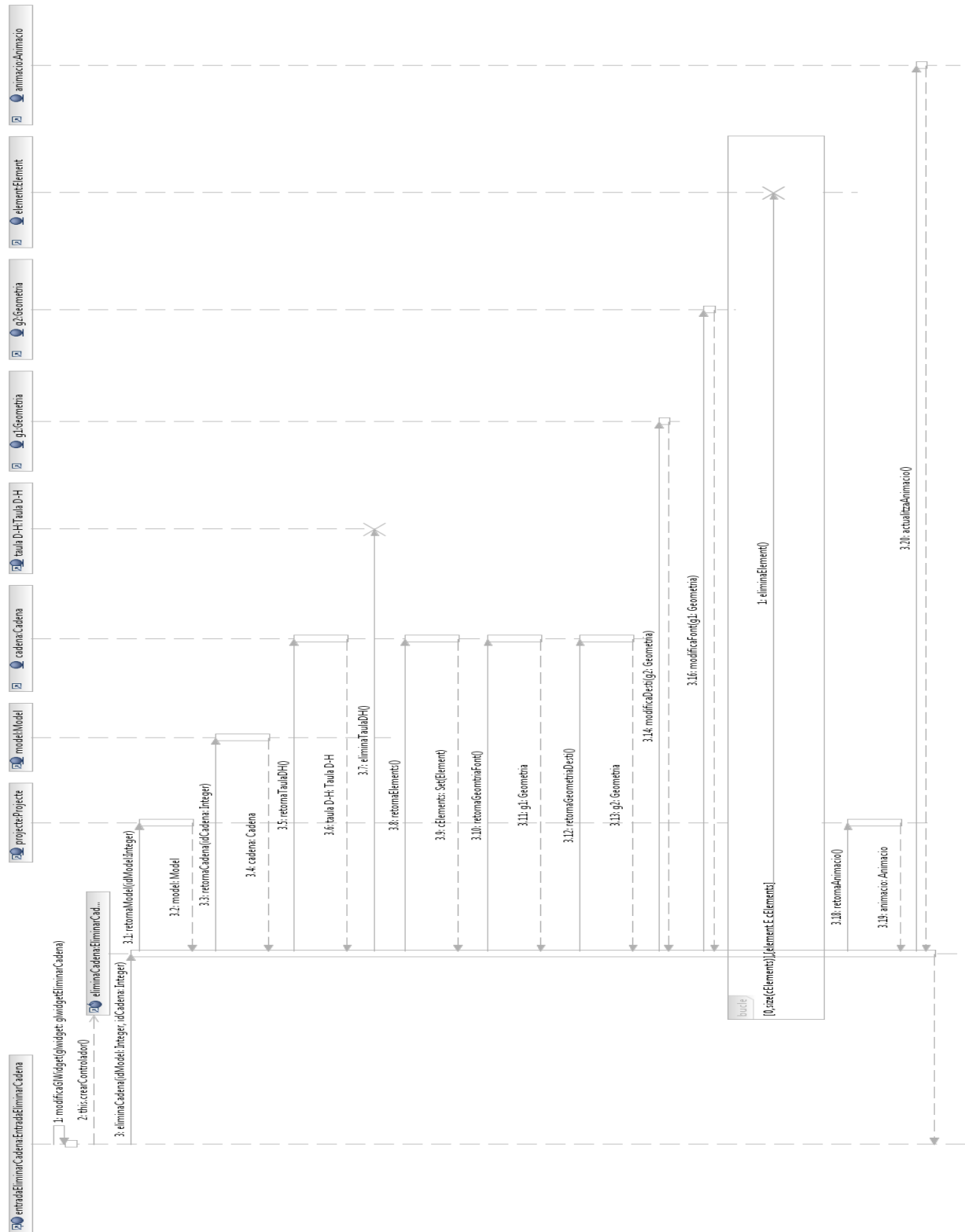


Figura 5.14: Diagrama de seqüència de Eliminar cadena

5.4.11 Cas d'ús 2.5: Moure cadena

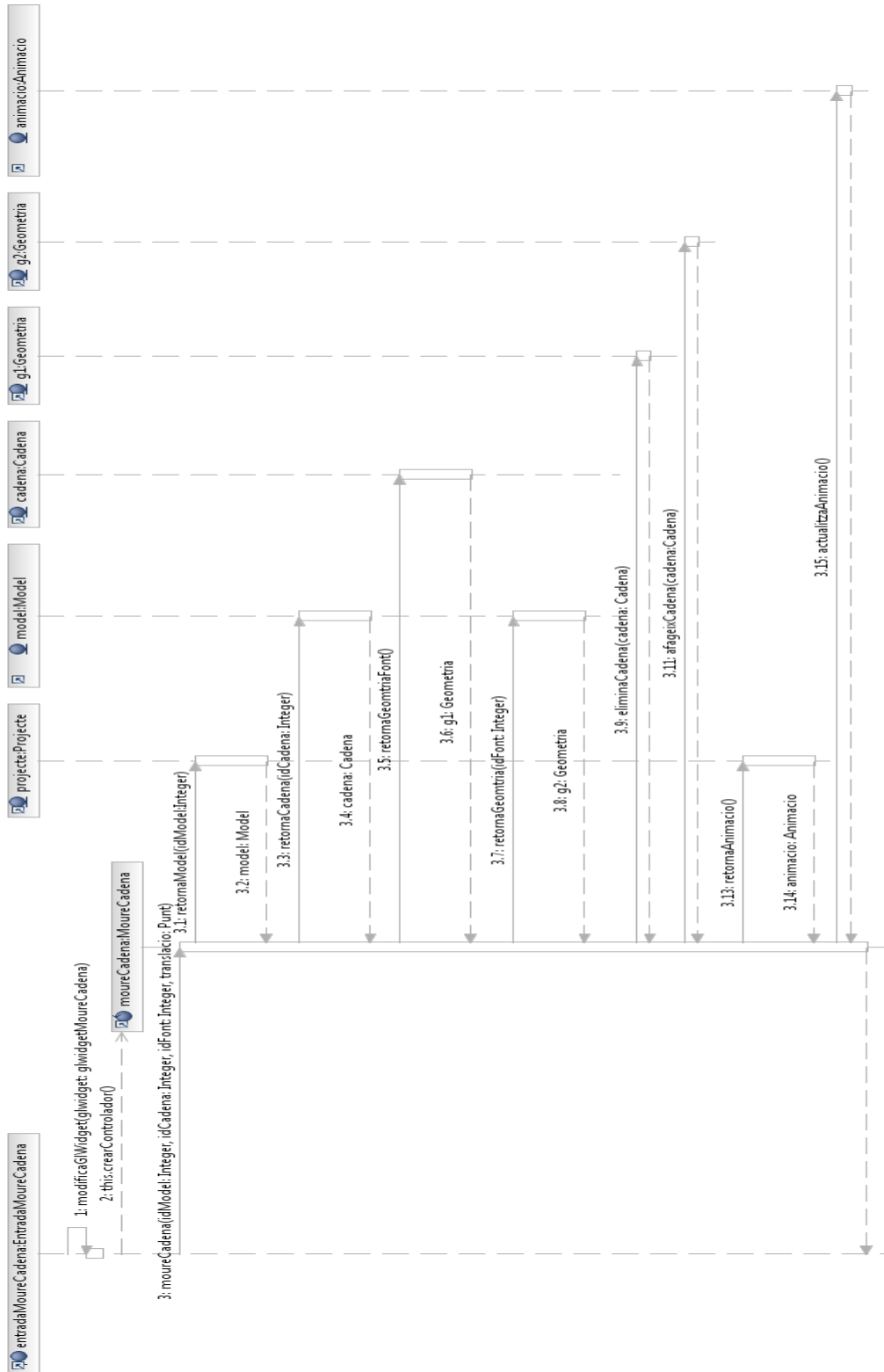


Figura 5.15: Diagrama de seqüència de Moure cadena

5.4.12 Cas d'ús 2.6: Rotar cadena

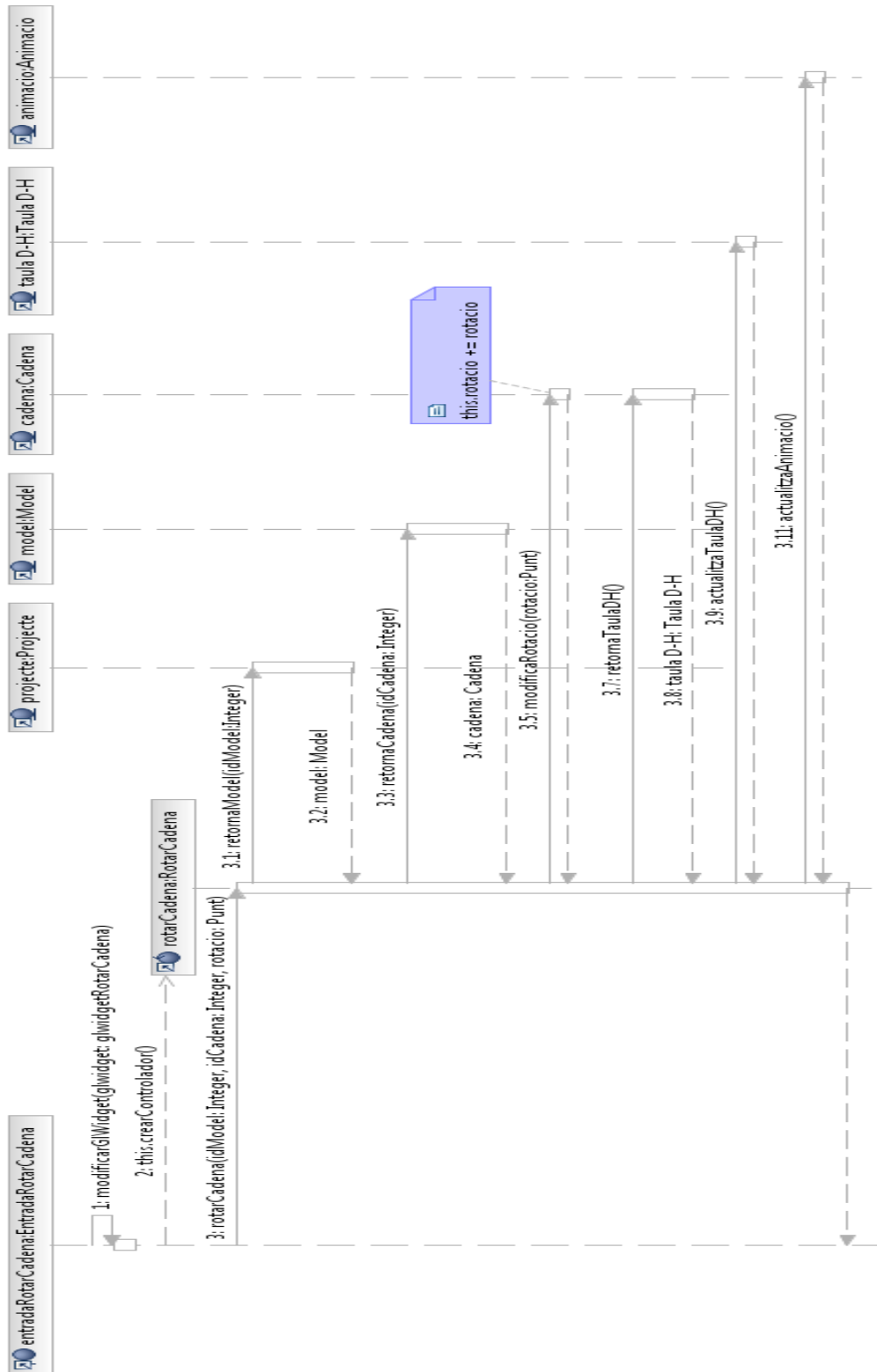


Figura 5.16: Diagrama de seqüència de Rotar cadena

5.4.13 Cas d'ús 2.7: Eliminar "Conjunt de cadenes"

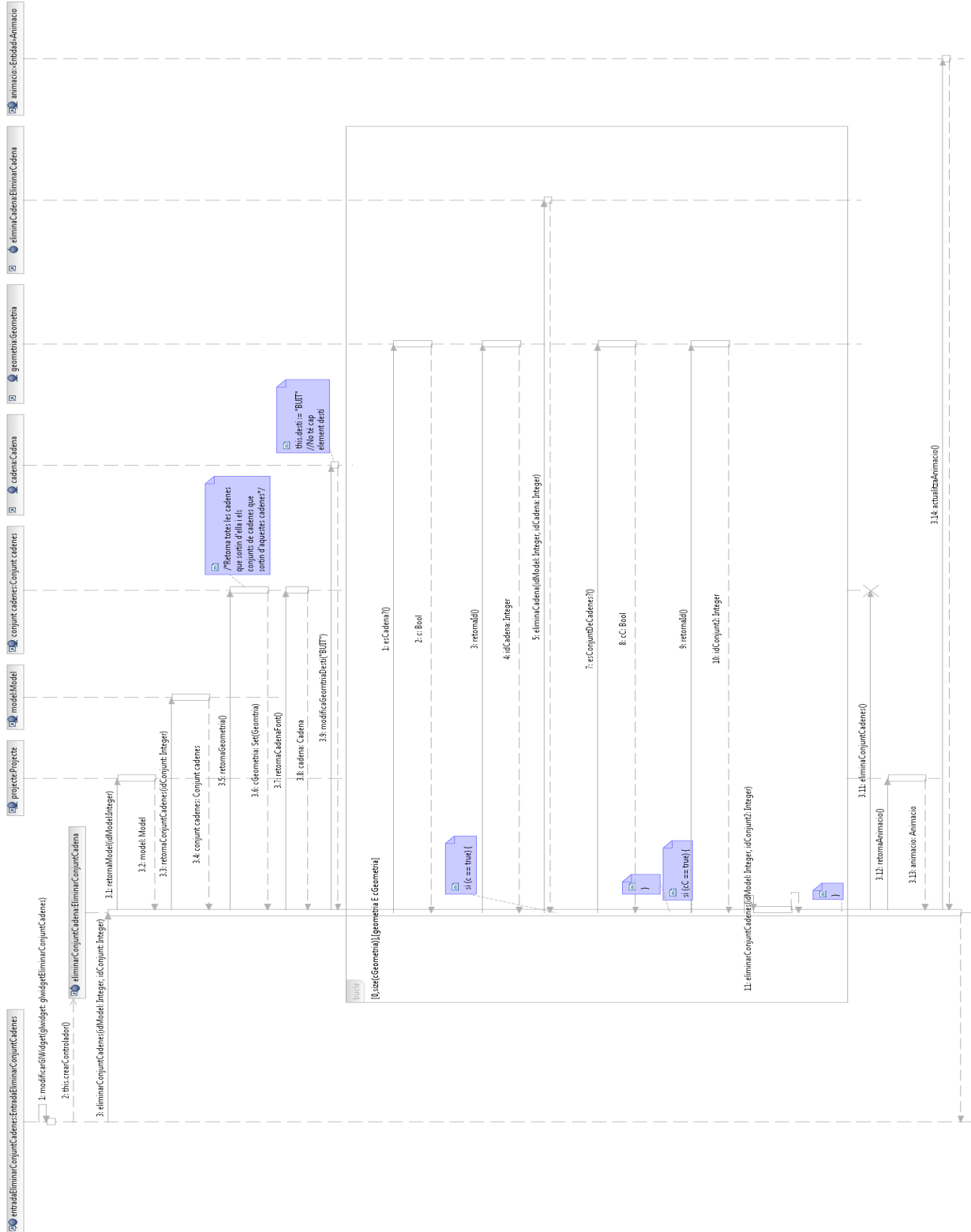


Figura 5.17: Diagrama de seqüència de Eliminar "Conjunt de cadenes"

5.4.14 Cas d'ús 2.8: Eliminar "Link"

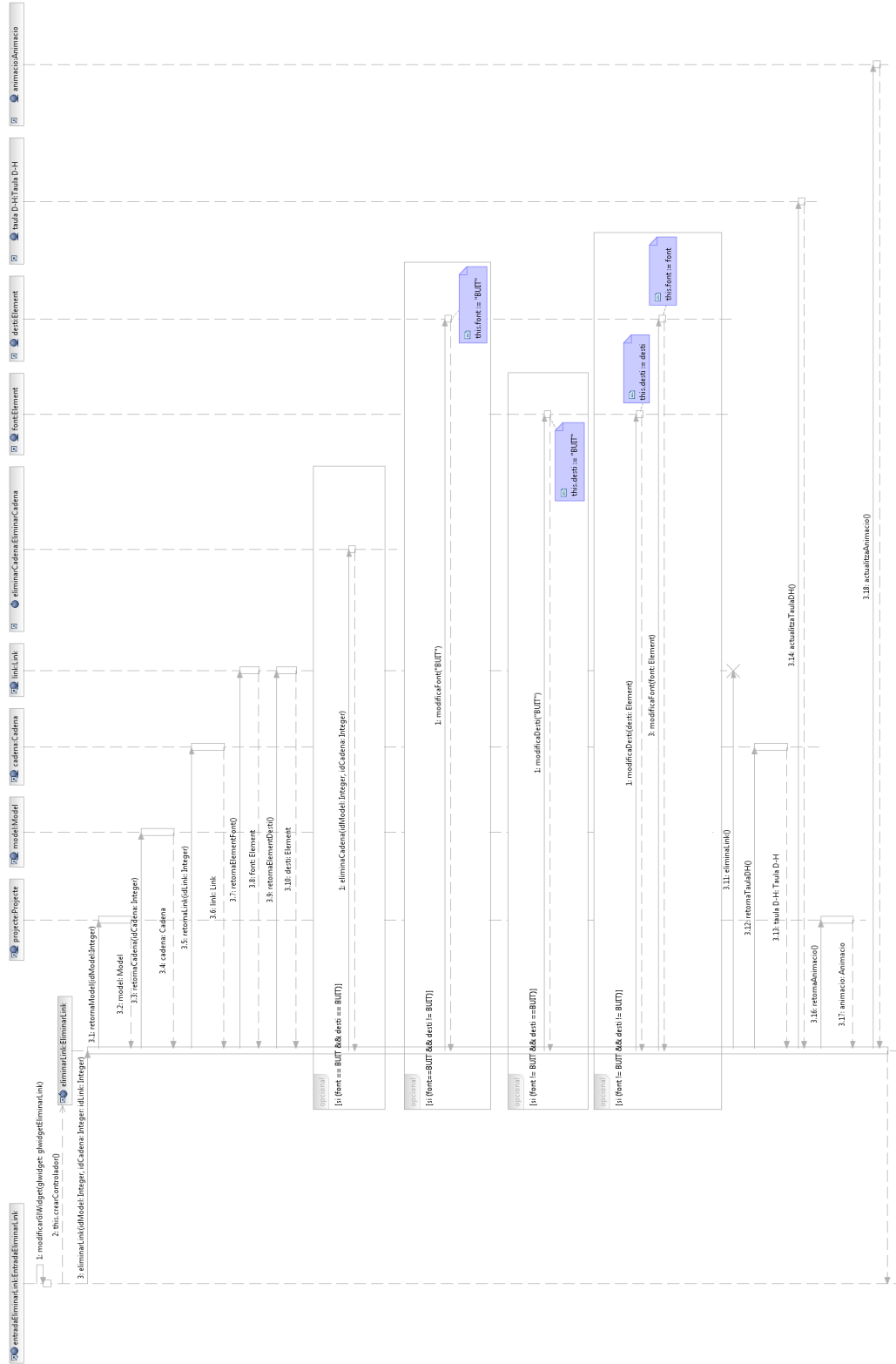


Figura 5.18: Diagrama de seqüència de Eliminar "Link"

5.4.15 Cas d'ús 2.9: Rotar "Link"

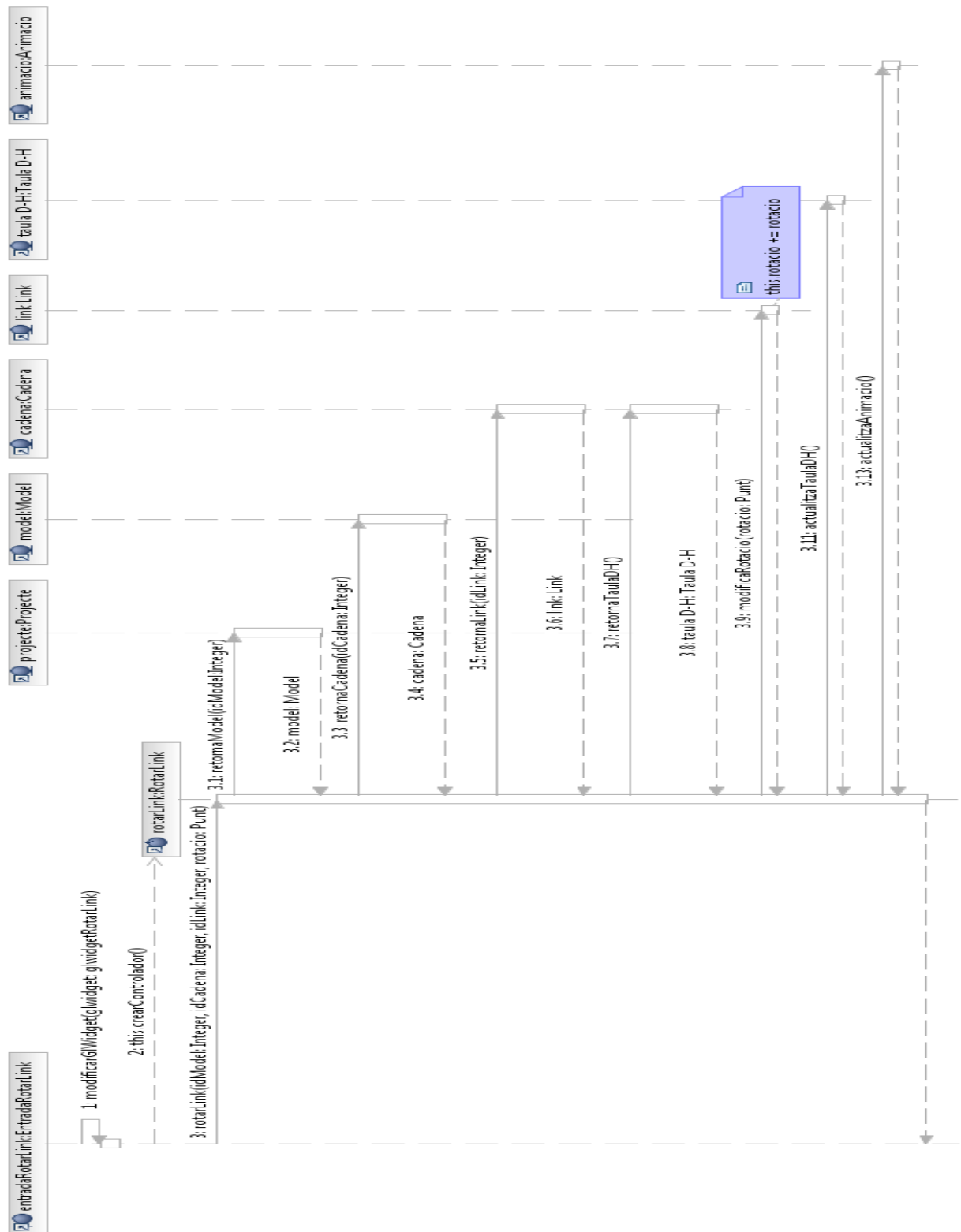


Figura 5.19: Diagrama de seqüència de Rotar "Link"

5.4.16 Cas d'ús 2.10: Modificar la llargada del "Link"

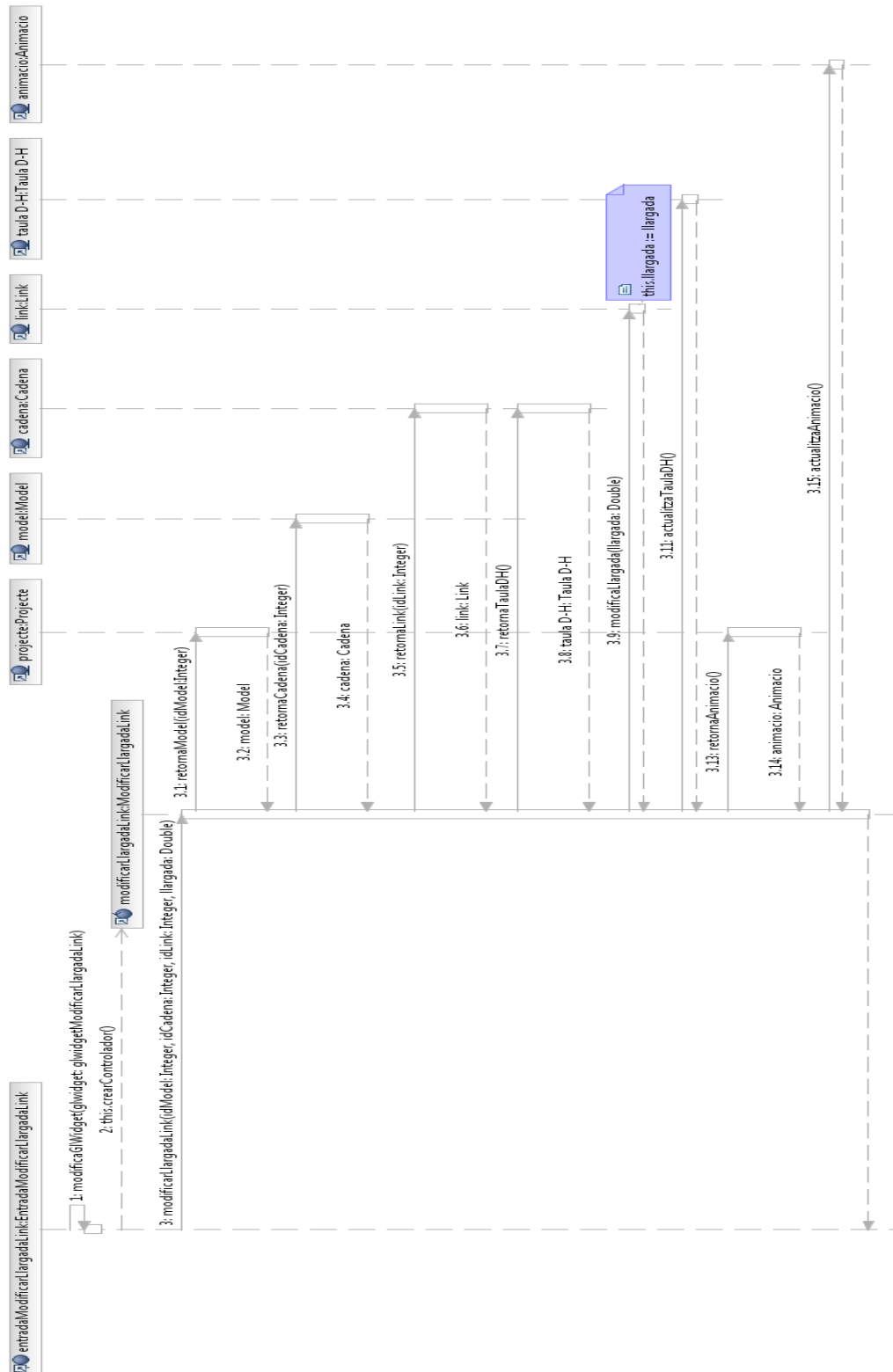


Figura 5.20: Diagrama de seqüència de Modificar la llargada del "Link"

5.4.17 Cas d'ús 2.11: Eliminar Massa

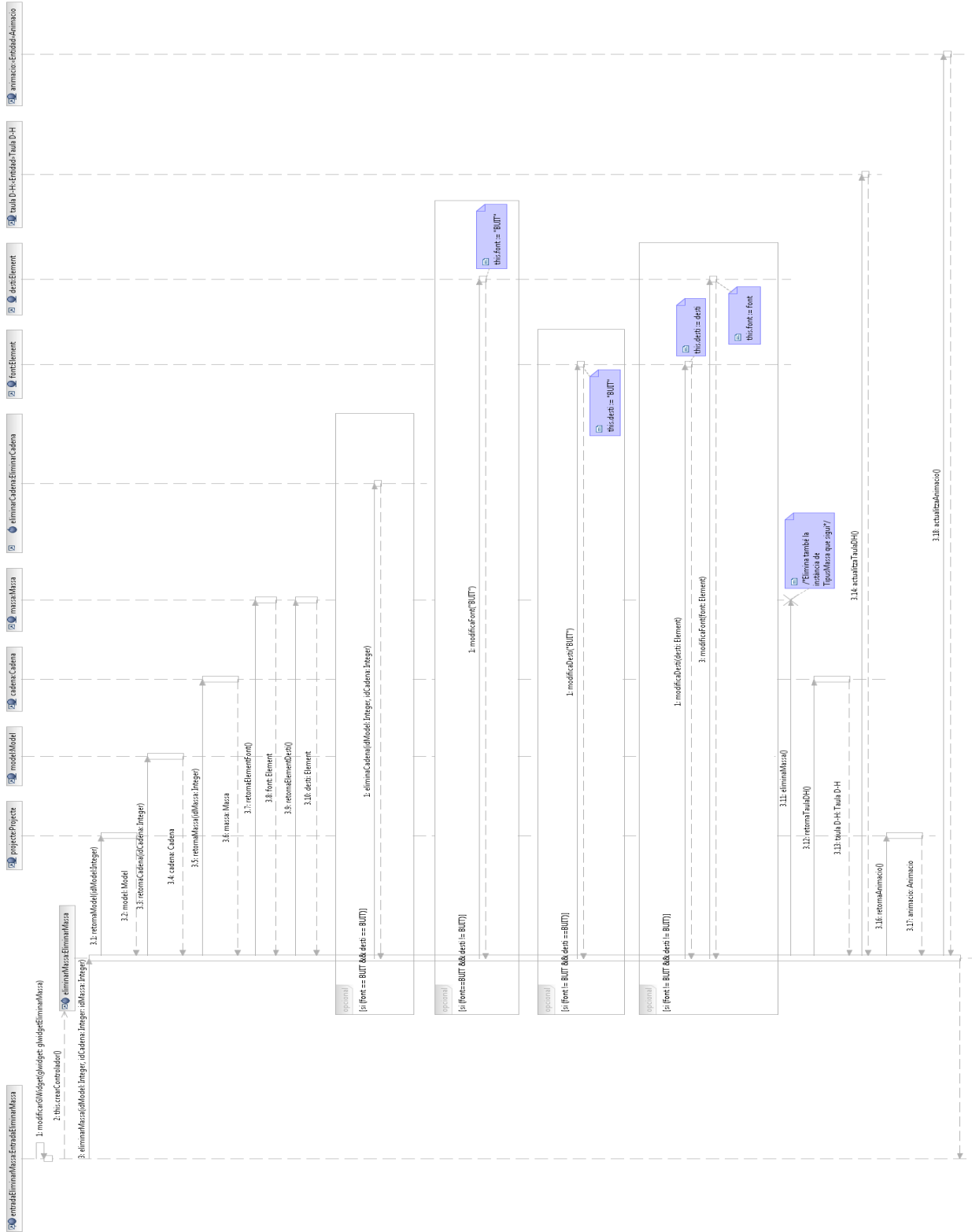


Figura 5.21: Diagrama de seqüència de Eliminar Massa

5.4.18 Cas d'ús 2.12: Rotar Massa

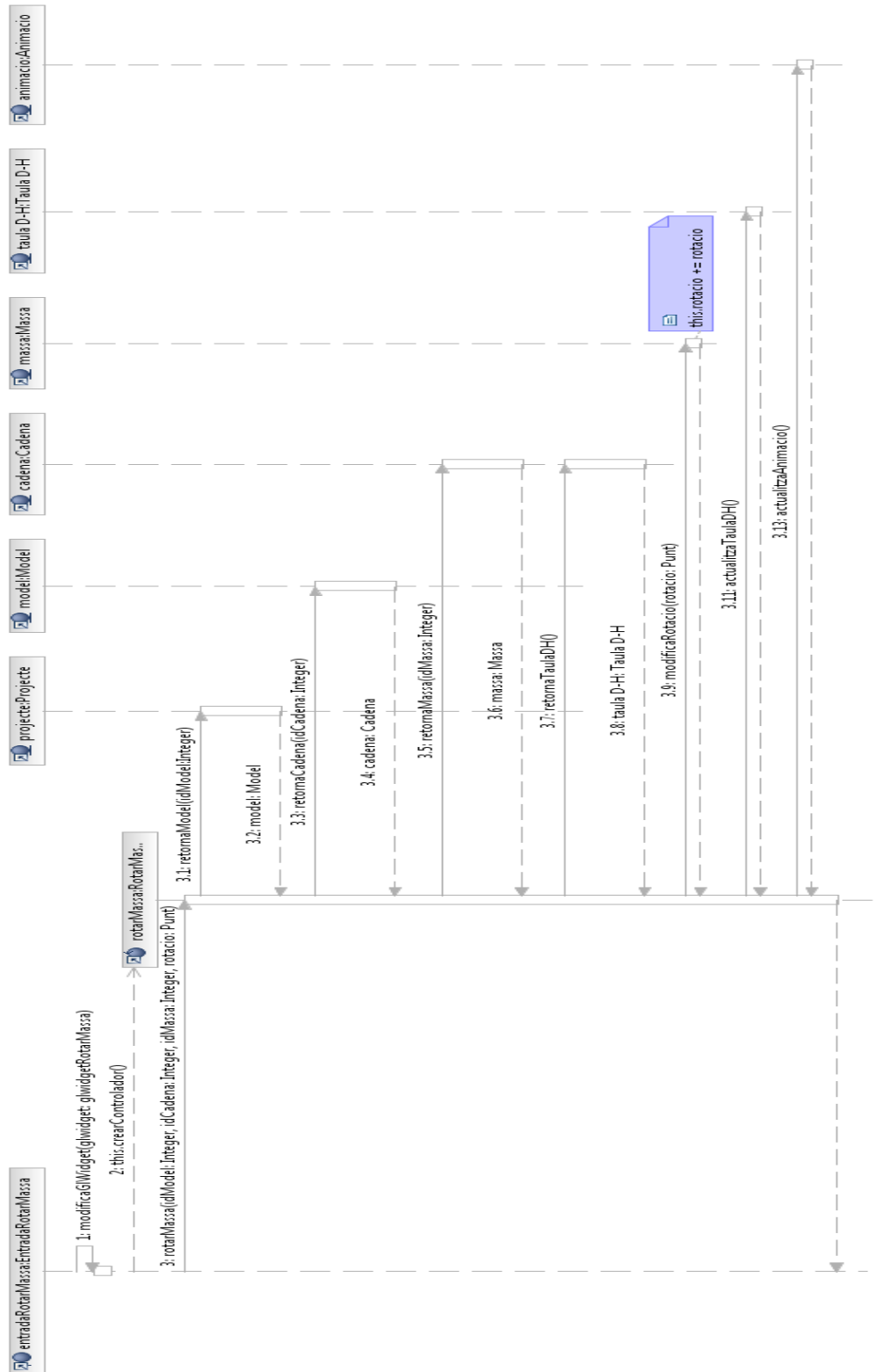


Figura 5.22: Diagrama de seqüència de Rotar Massa

5.4.19 Cas d'ús 2.13: Rotar un Recolzador

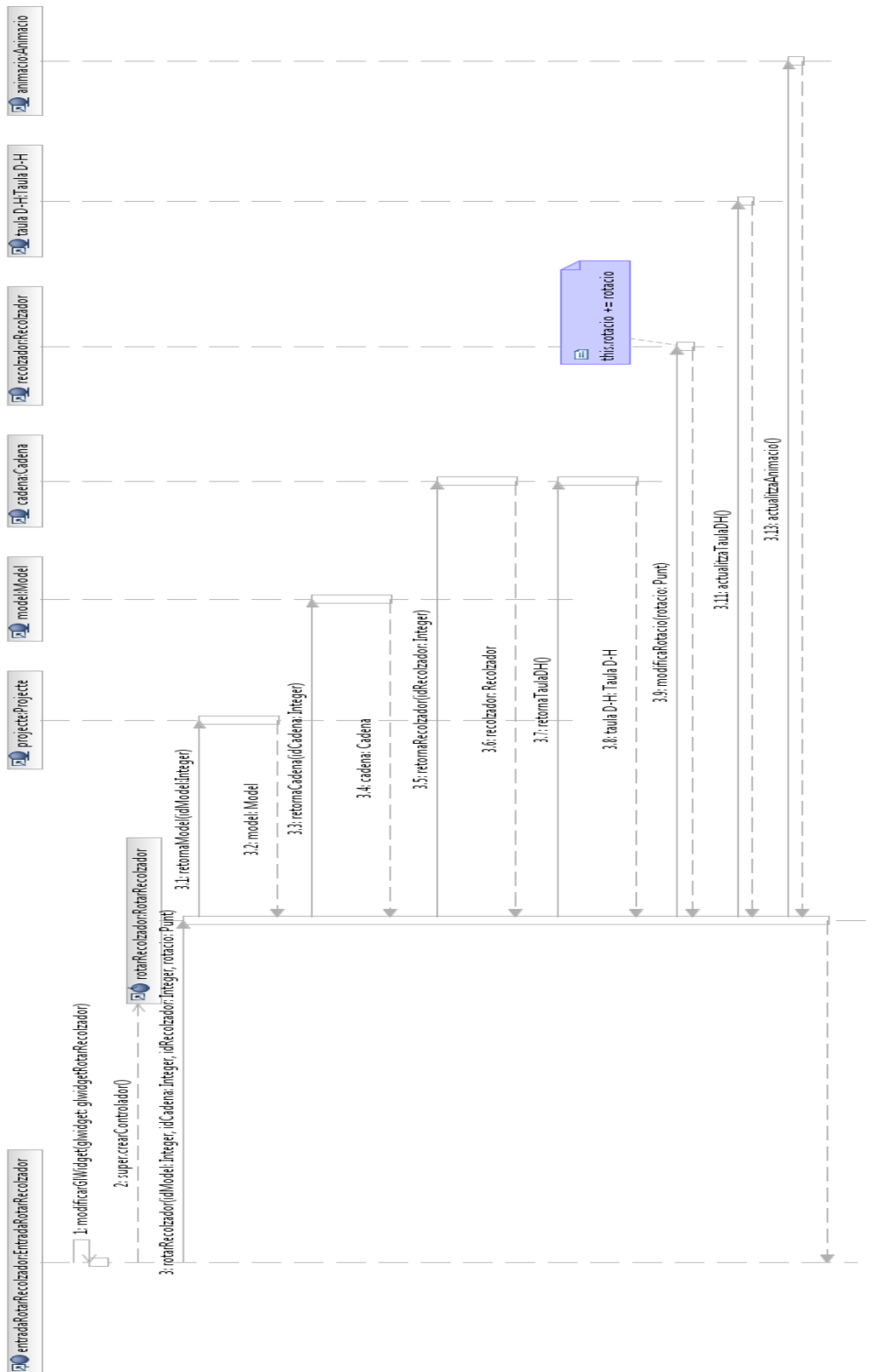


Figura 5.23: Diagrama de seqüència de Rotar un Recolzador

5.4.20 Cas d'ús 2.14: Eliminar un Recolzador

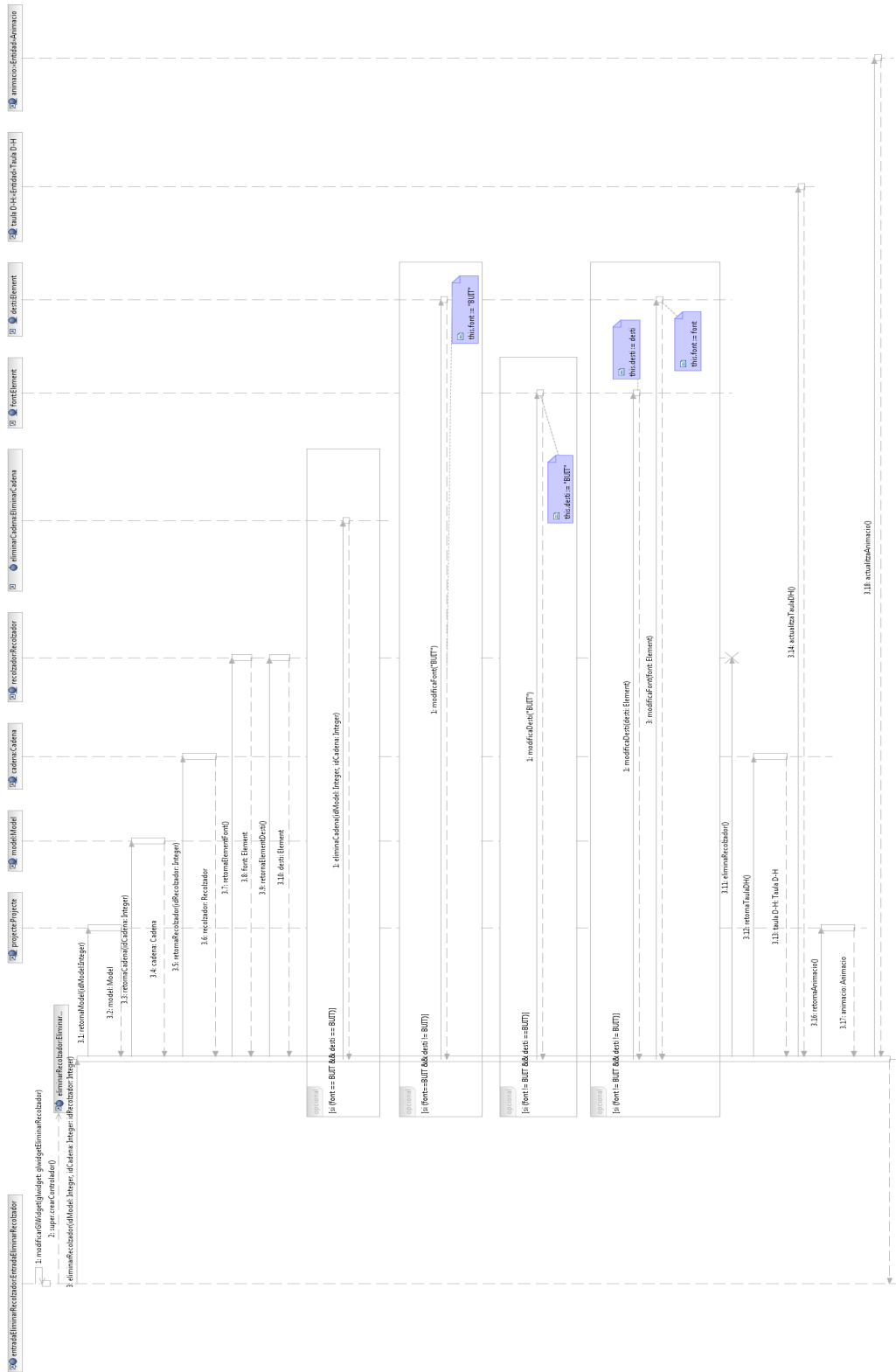


Figura 5.24: Diagrama de seqüència de Eliminar un Recolzador

5.4.21 Cas d'ús 2.15: Modificar la llargada d'un Recolzador

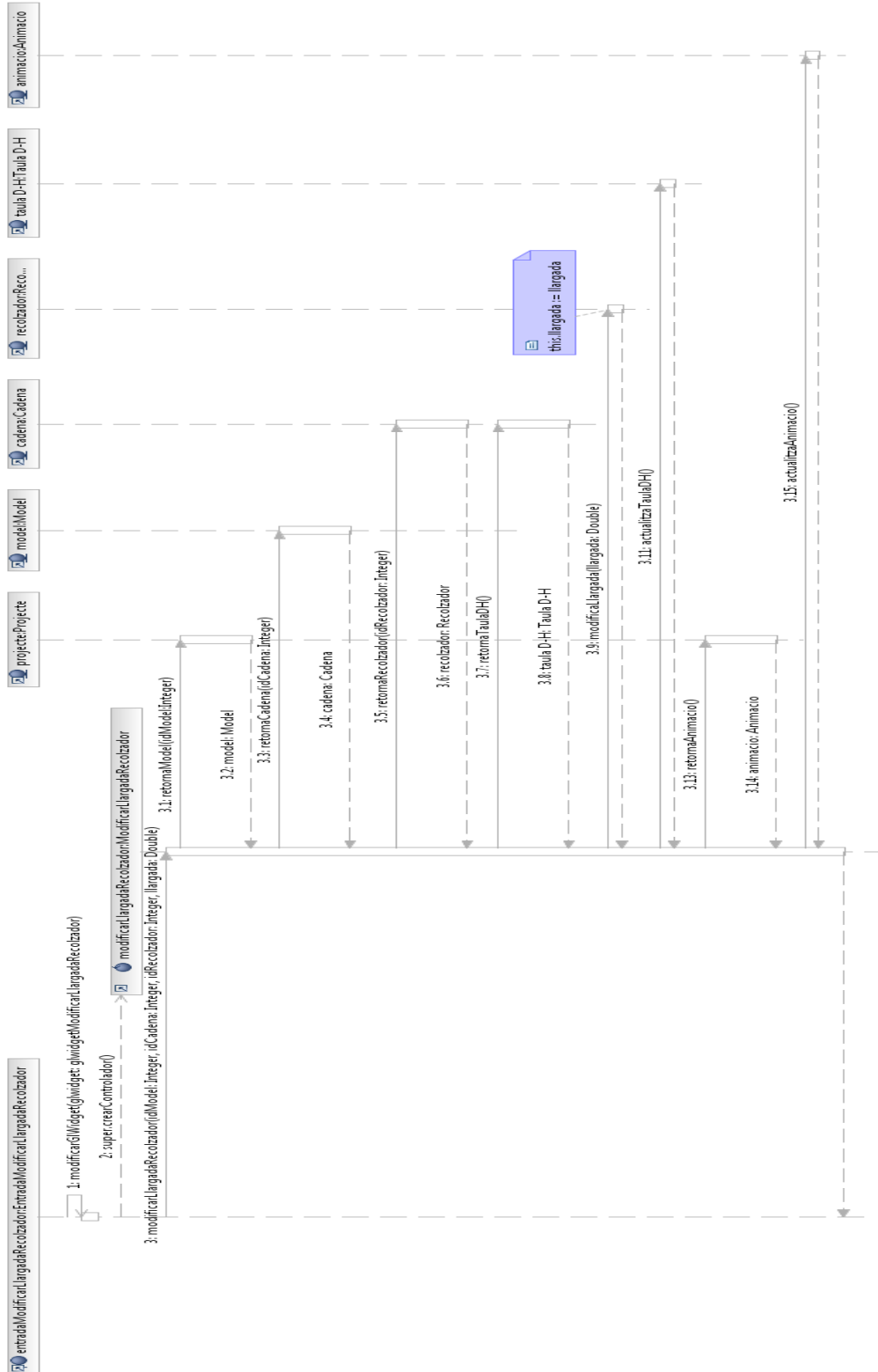


Figura 5.25: Diagrama de seqüència de Modificar la llargada d'un Recolzador

5.4.22 Cas d'ús 2.16: Modificar l'amplada d'un Recolzador

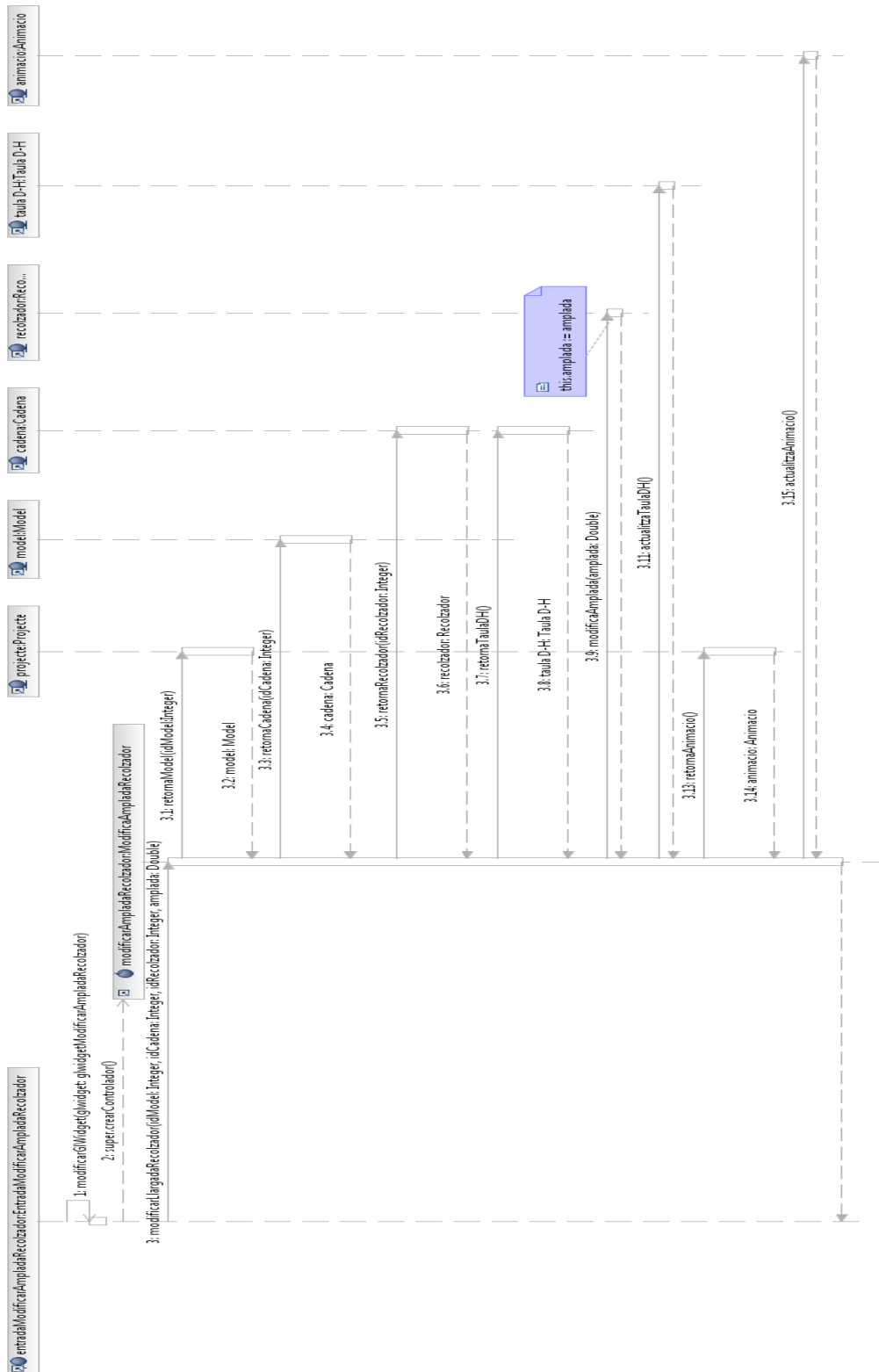


Figura 5.26: Diagrama de seqüència de Modificar l'amplada d'un Recolzador

5.4.23 Cas d'ús 3.1: Afegir un objecte

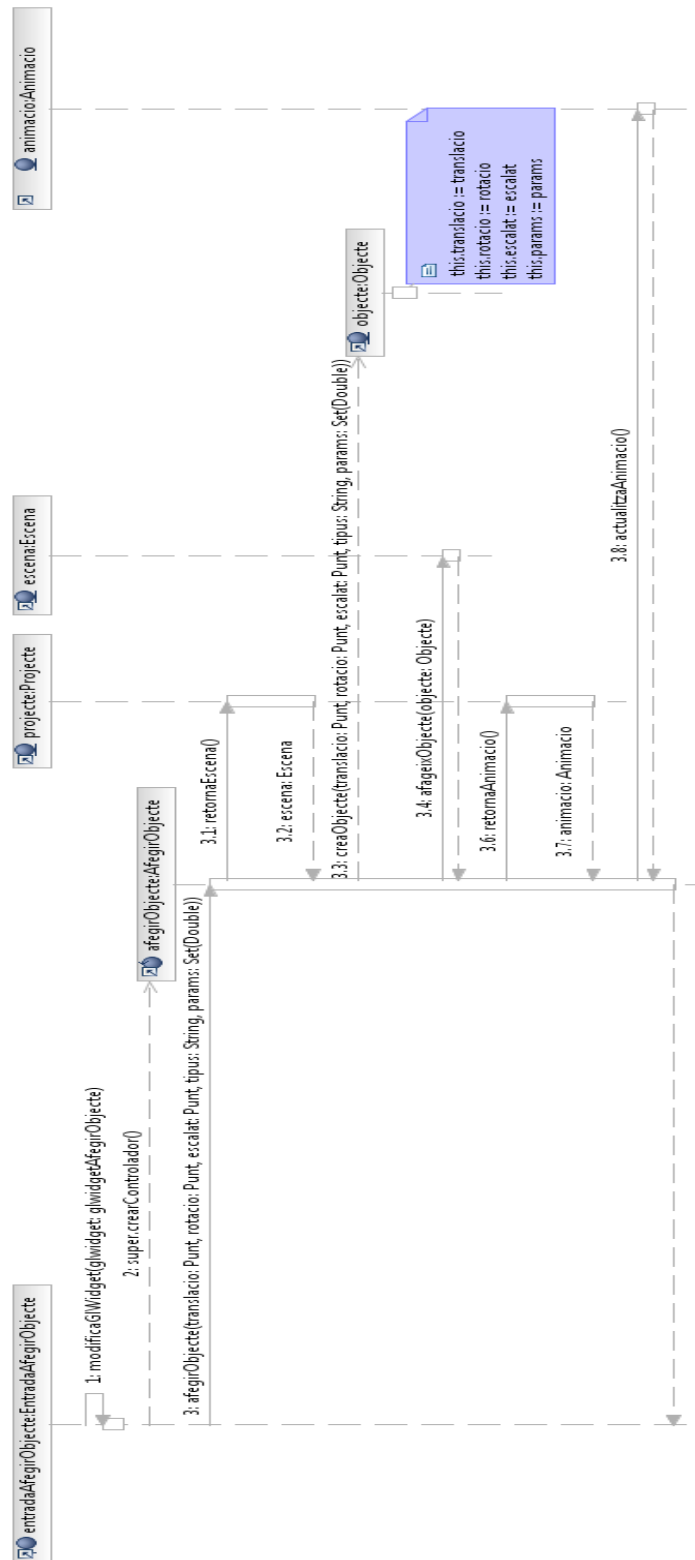


Figura 5.27: Diagrama de seqüència de Afegir un objecte

5.4.24 Cas d'ús 4.1: Rotar objecte

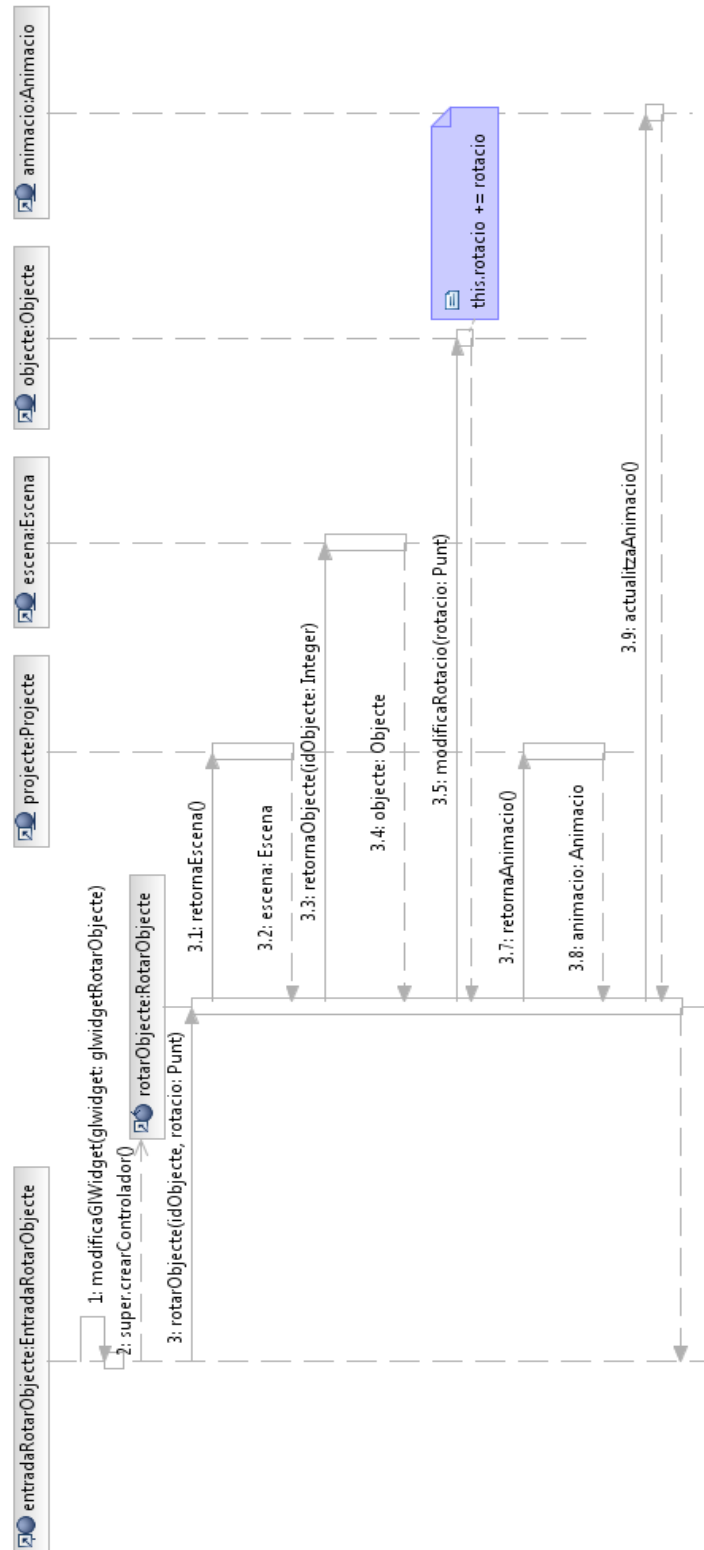


Figura 5.28: Diagrama de seqüència de Rotar objecte

5.4.25 Cas d'ús 4.2: Escalar objecte

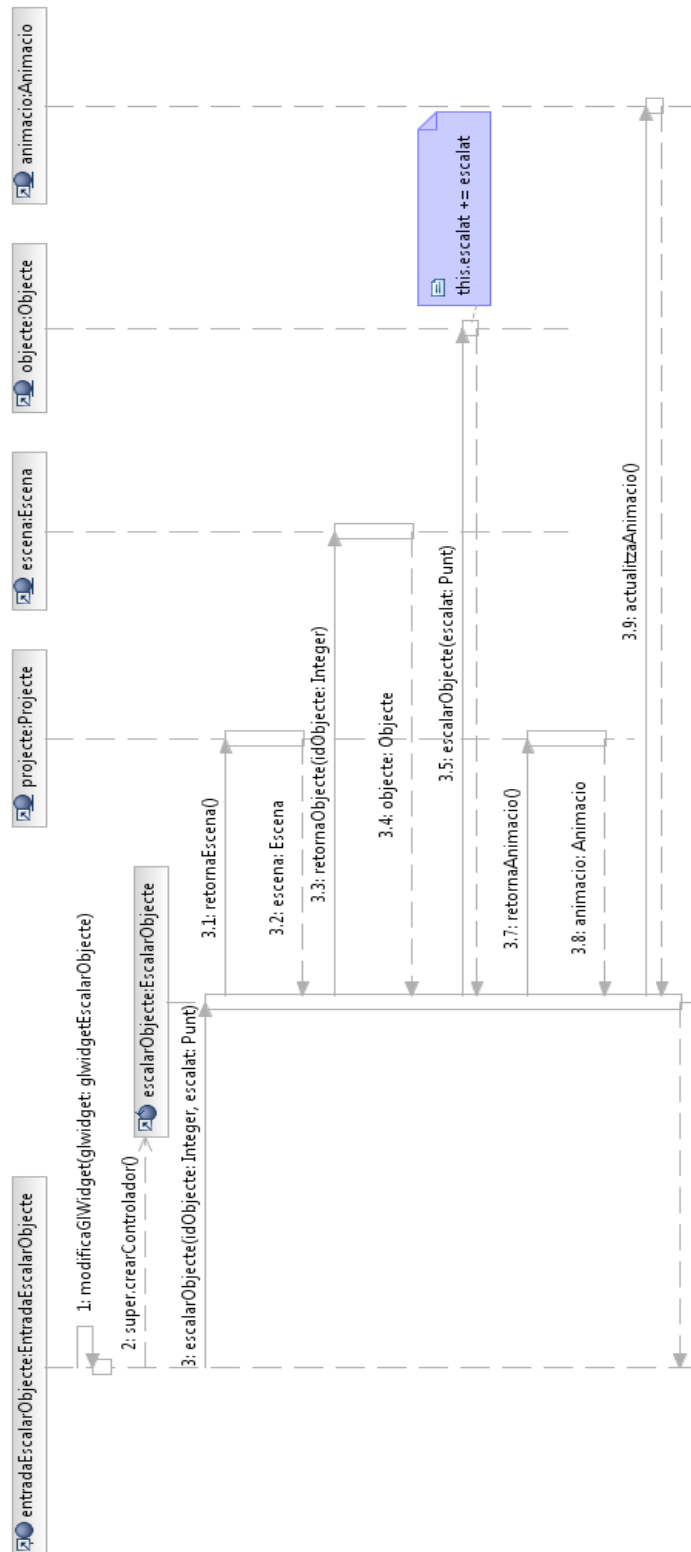


Figura 5.29: Diagrama de seqüència de Escalar objecte

5.4.26 Cas d'ús 4.3: Moure objecte

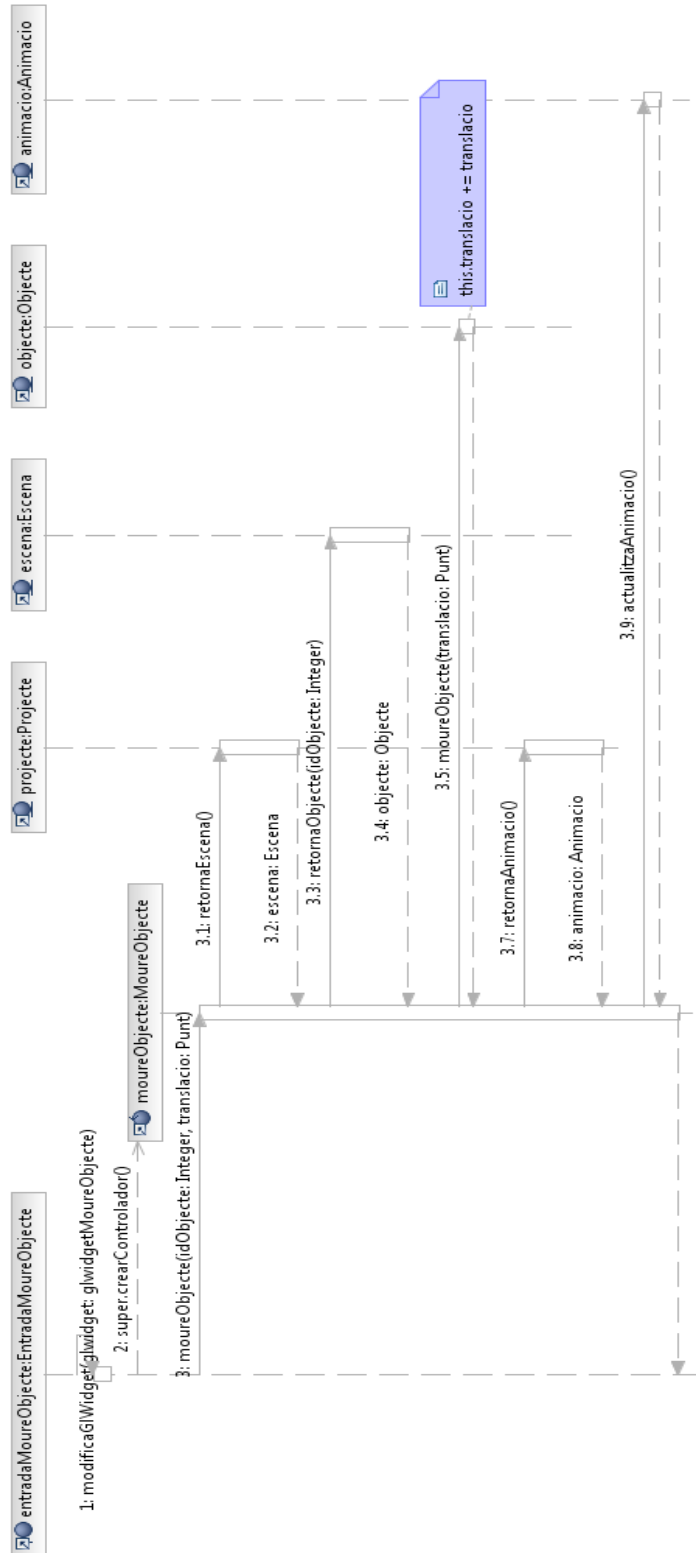


Figura 5.30: Diagrama de seqüència de Moure objecte

5.4.27 Cas d'ús 4.4: Duplicar objecte

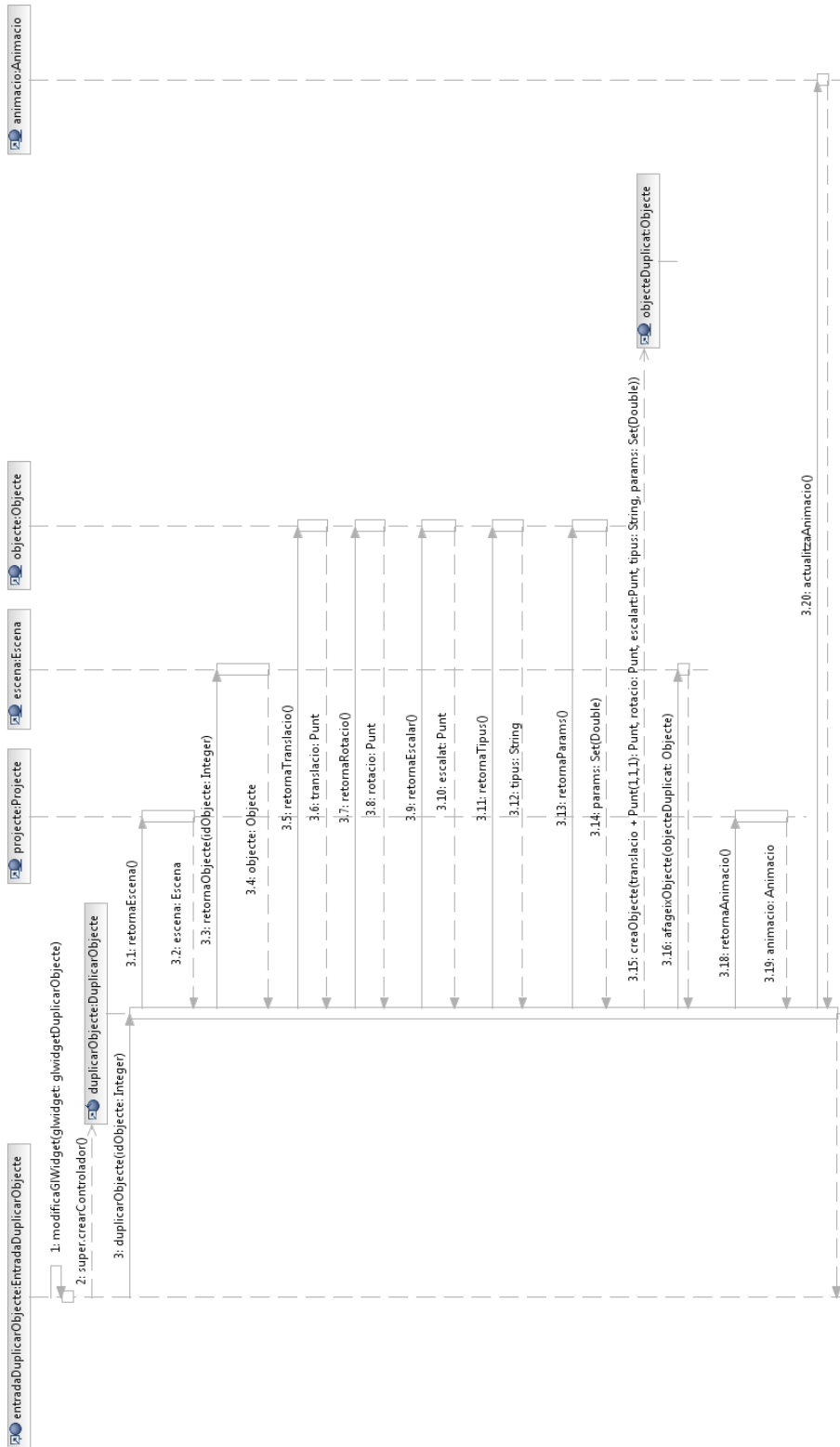


Figura 5.31: Diagrama de seqüència de Duplicar objecte

5.4.28 Cas d'ús 4.5: Eliminar objecte

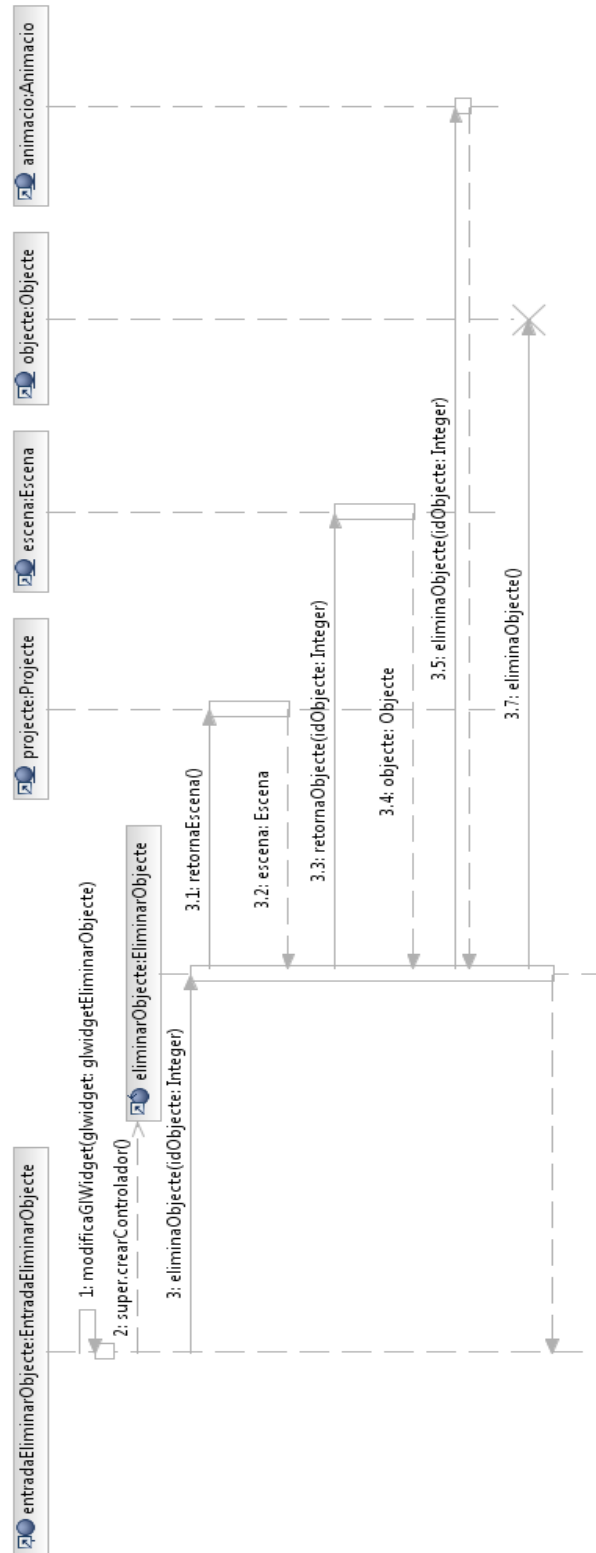


Figura 5.32: Diagrama de seqüència de Eliminar objecte

5.4.29 Cas d'ús 4.6: Modificar paràmetre objecte

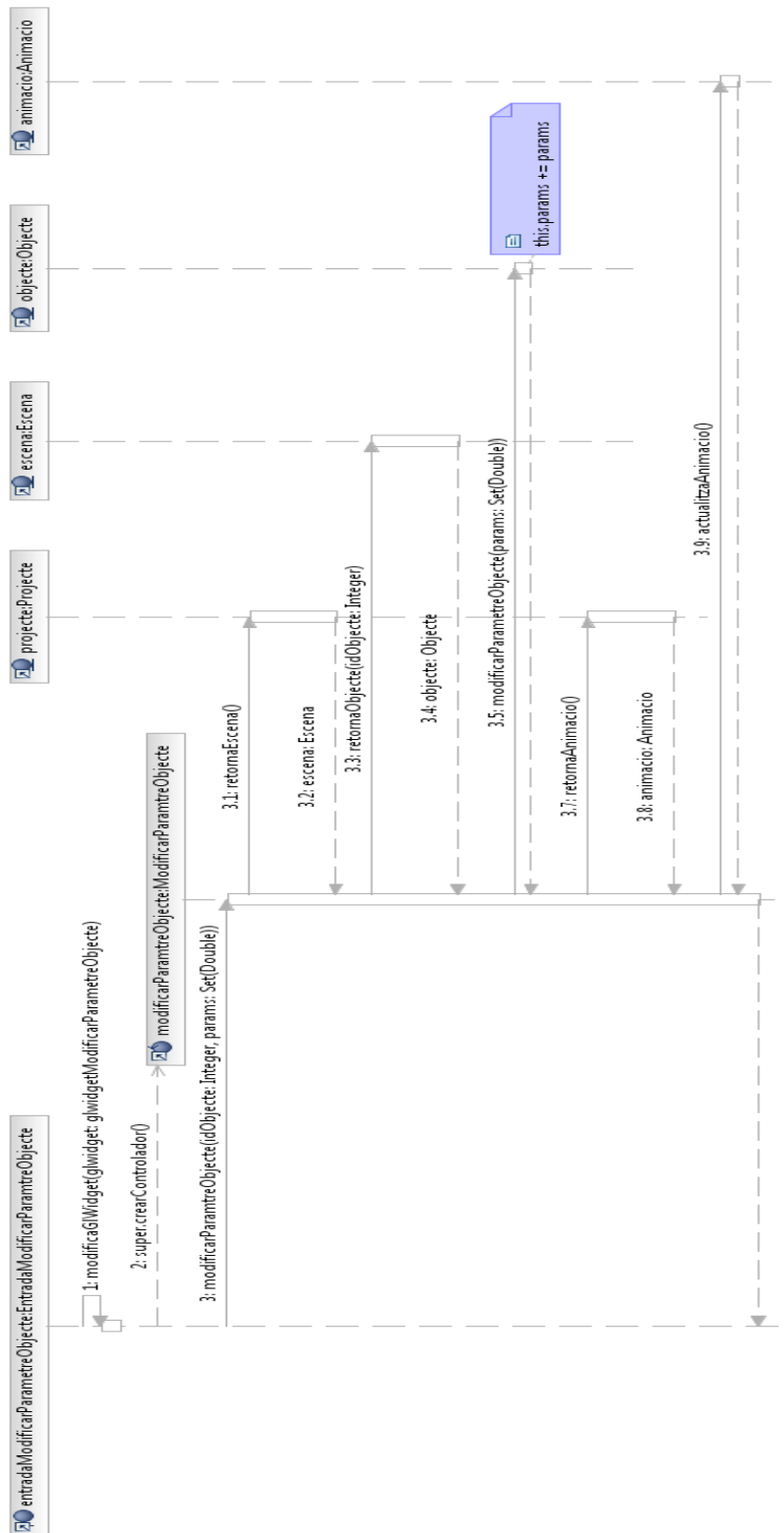


Figura 5.33: Diagrama de seqüència de Modificar paràmetre objecte

5.4.30 Cas d'ús 5.1: Crear punt clau

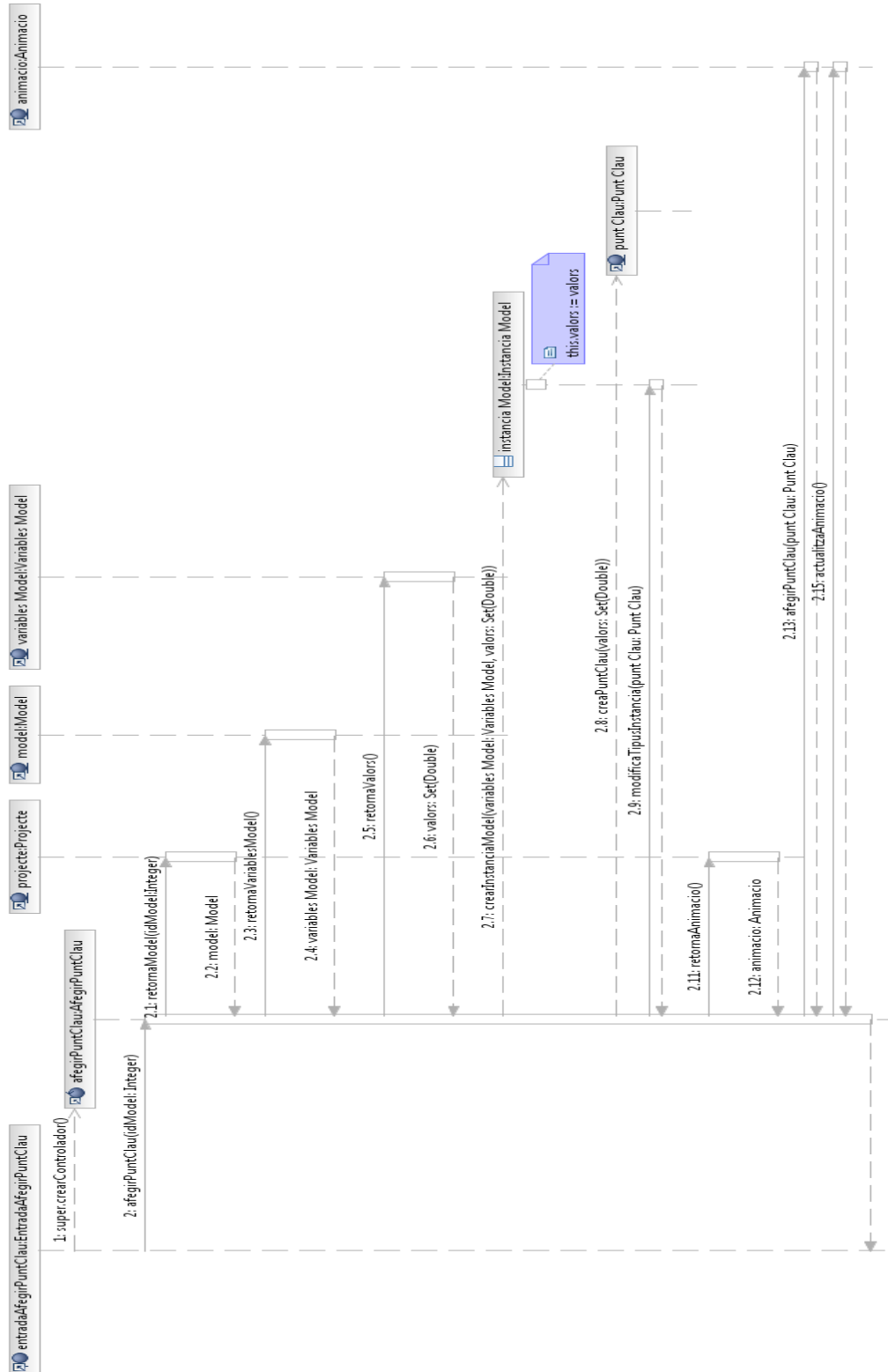


Figura 5.34: Diagrama de seqüència de Crear punt clau

5.4.31 Cas d'ús 5.2: Modificar paràmetre del model articulat

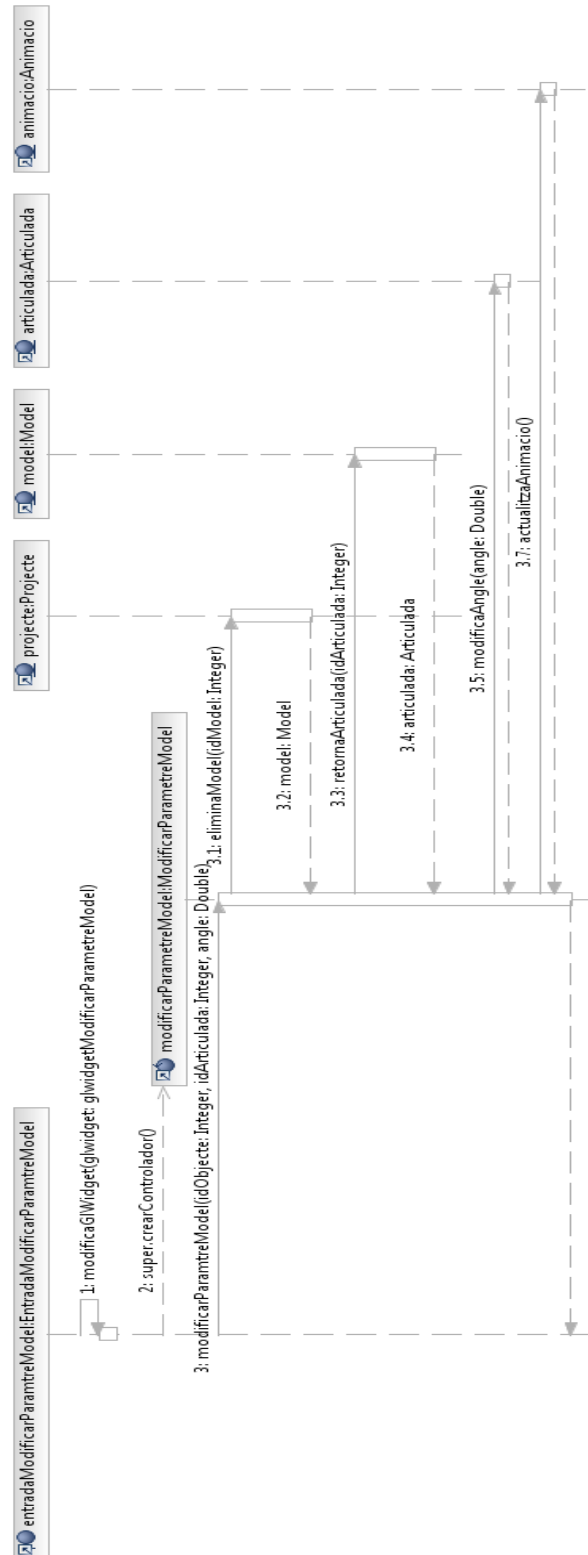


Figura 5.35: Diagrama de seqüència de Modificar paràmetre del model articulat

5.4.32 Cas d'ús 5.3: Afegir punt de contacte

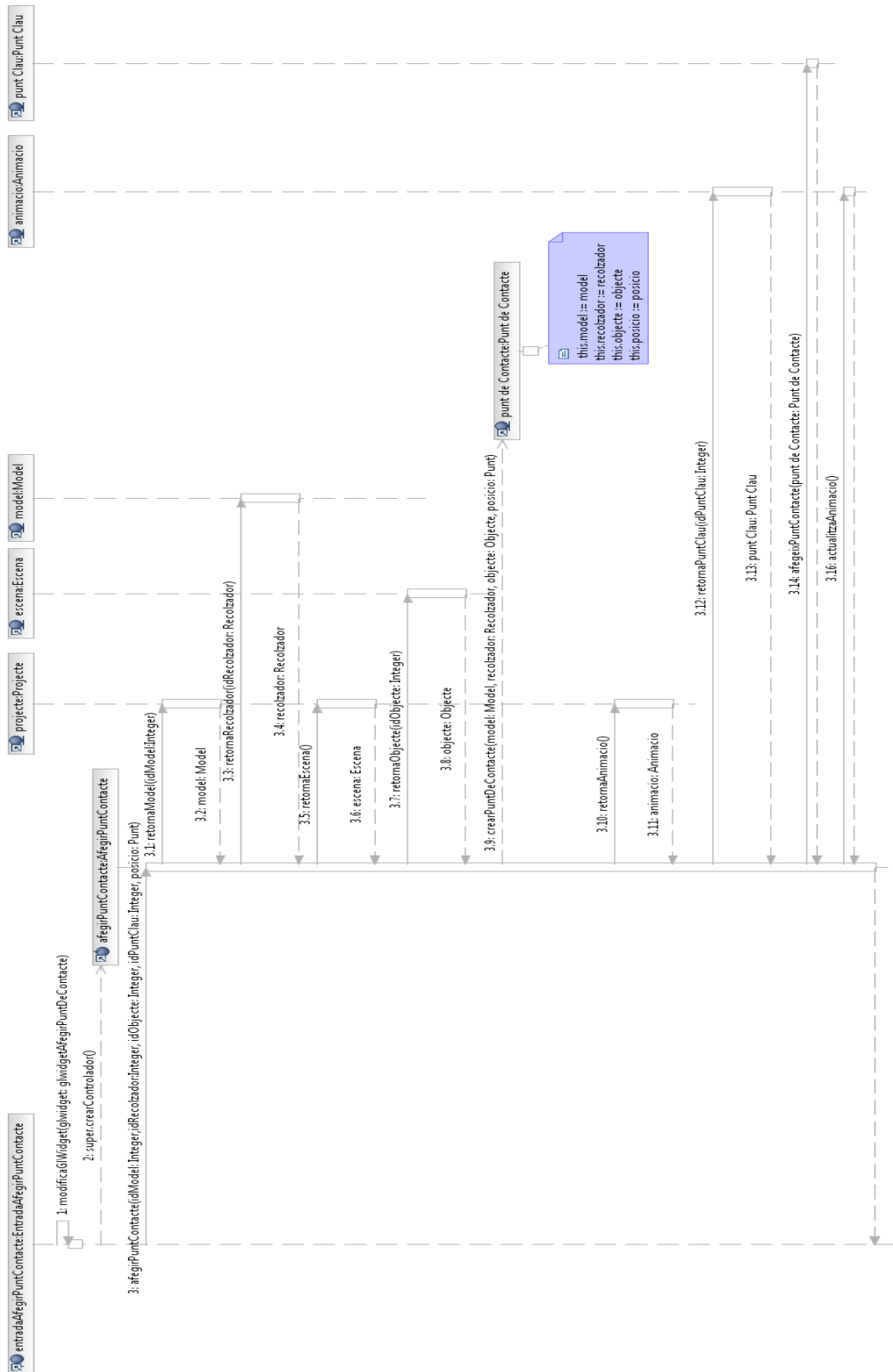


Figura 5.36: Diagrama de seqüència de Afegir punt de contacte

5.4.33 Cas d'ús 5.4: Assignar instant de temps

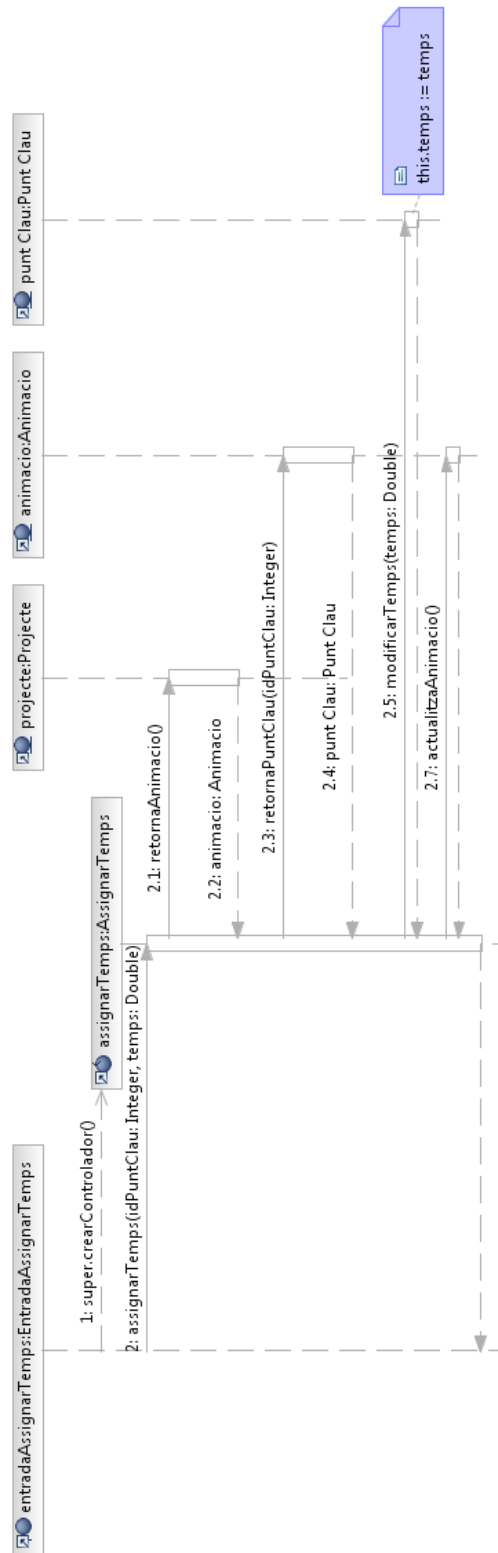


Figura 5.37: Diagrama de seqüència de Assignar instant de temps

5.4.34 Cas d'ús 5.5: Generar animació

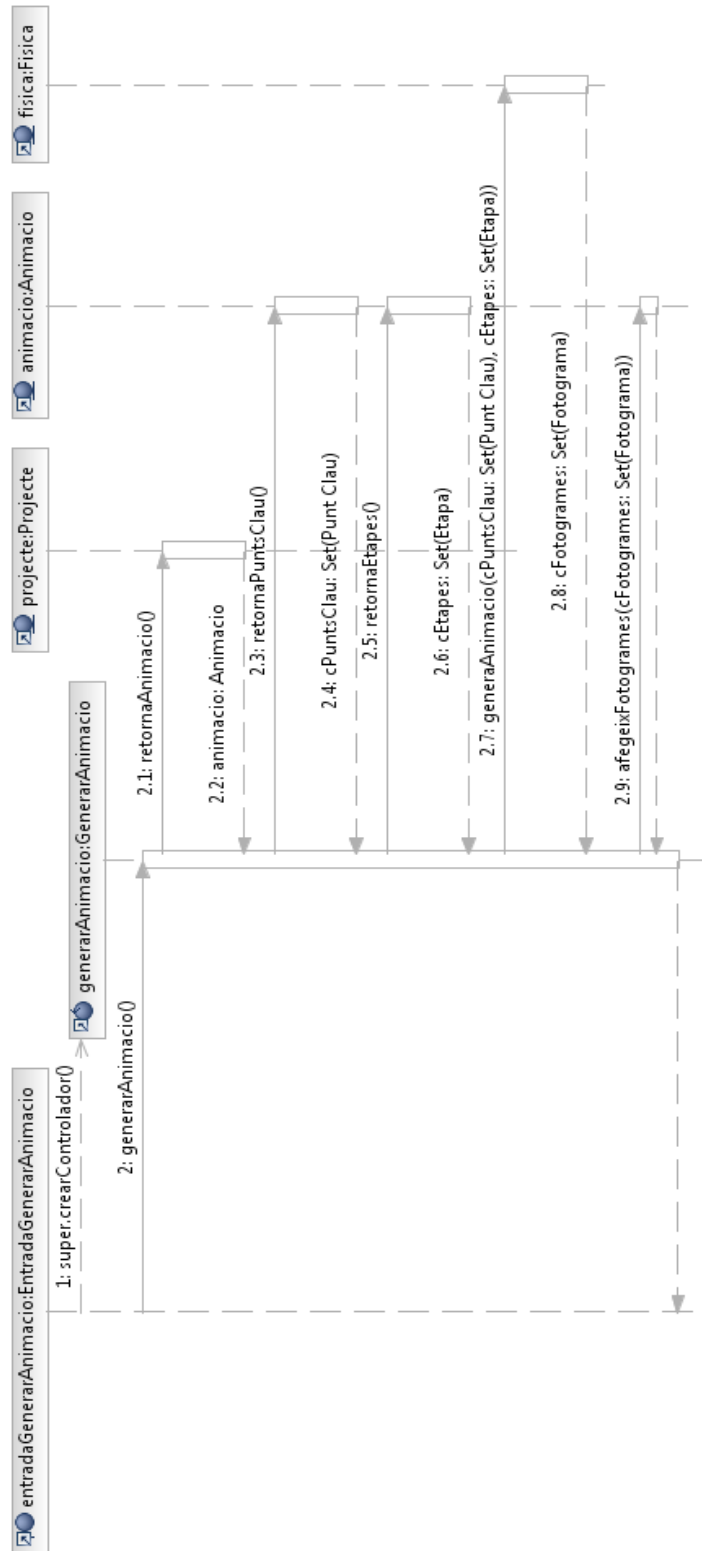


Figura 5.38: Diagrama de seqüència de Generar animació

5.4.35 Cas d'ús 6.1: Eliminar punt clau

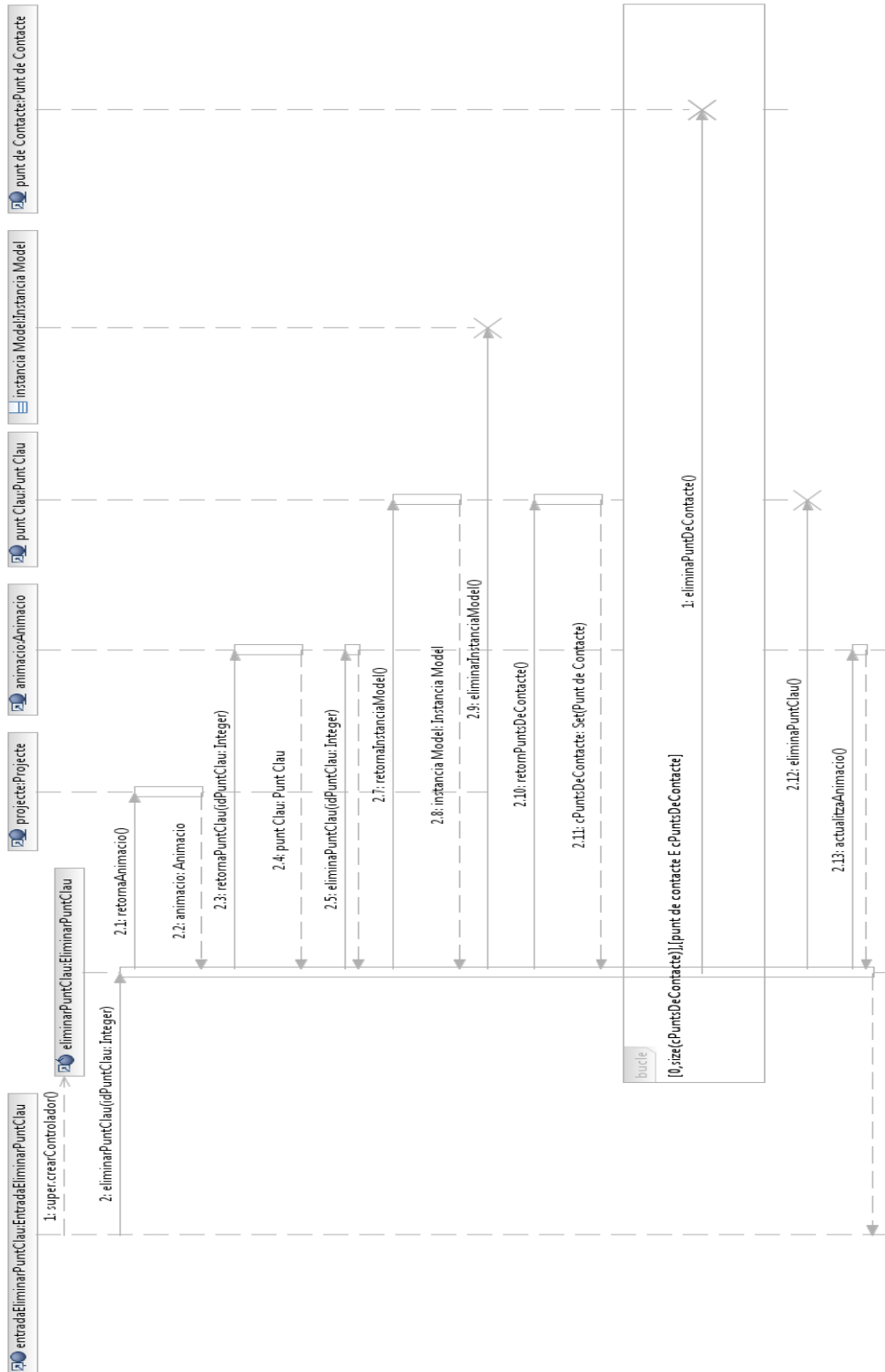


Figura 5.39: Diagrama de seqüència de Eliminar punt clau

5.4.36 Cas d'ús 6.2: Modificar instant de temps

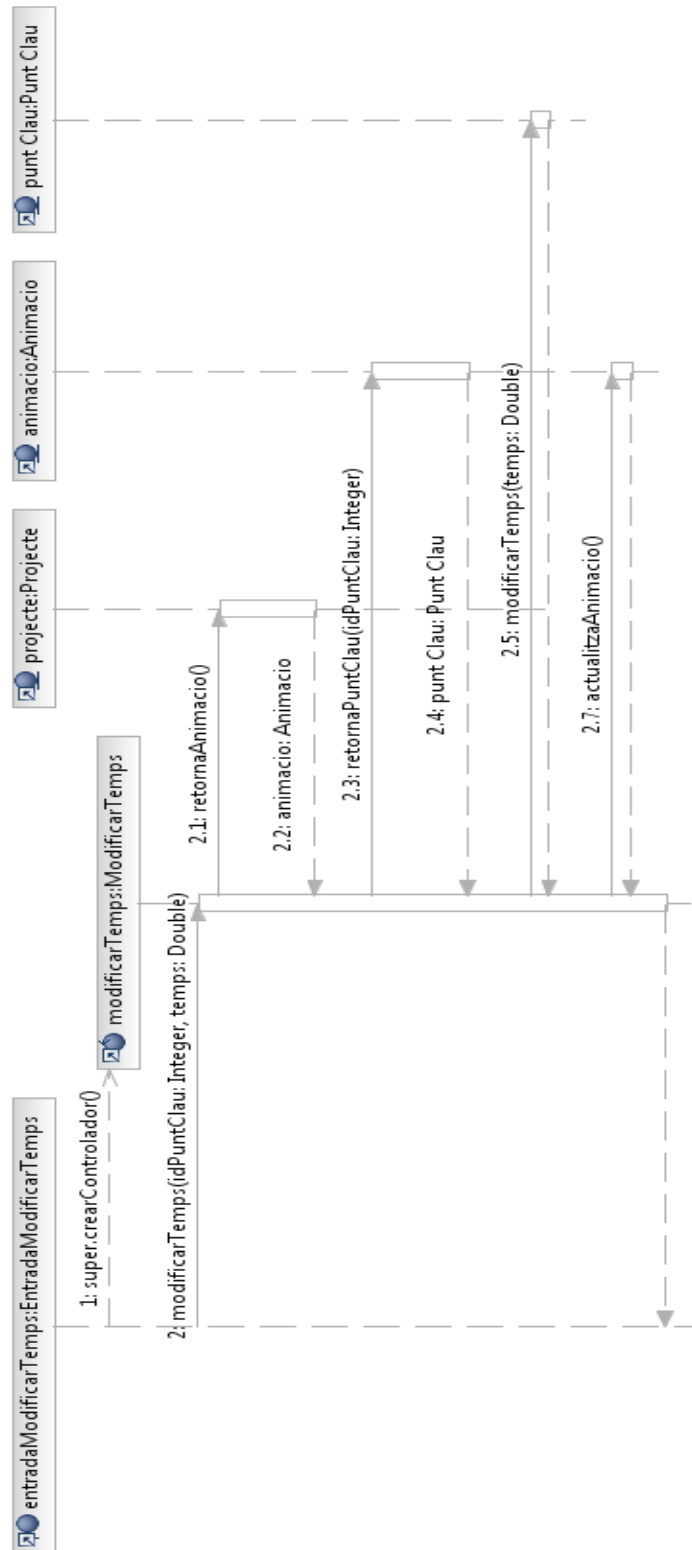


Figura 5.40: Diagrama de seqüència de Modificar instant de temps

5.4.37 Cas d'ús 6.3: Eliminar punt de contacte

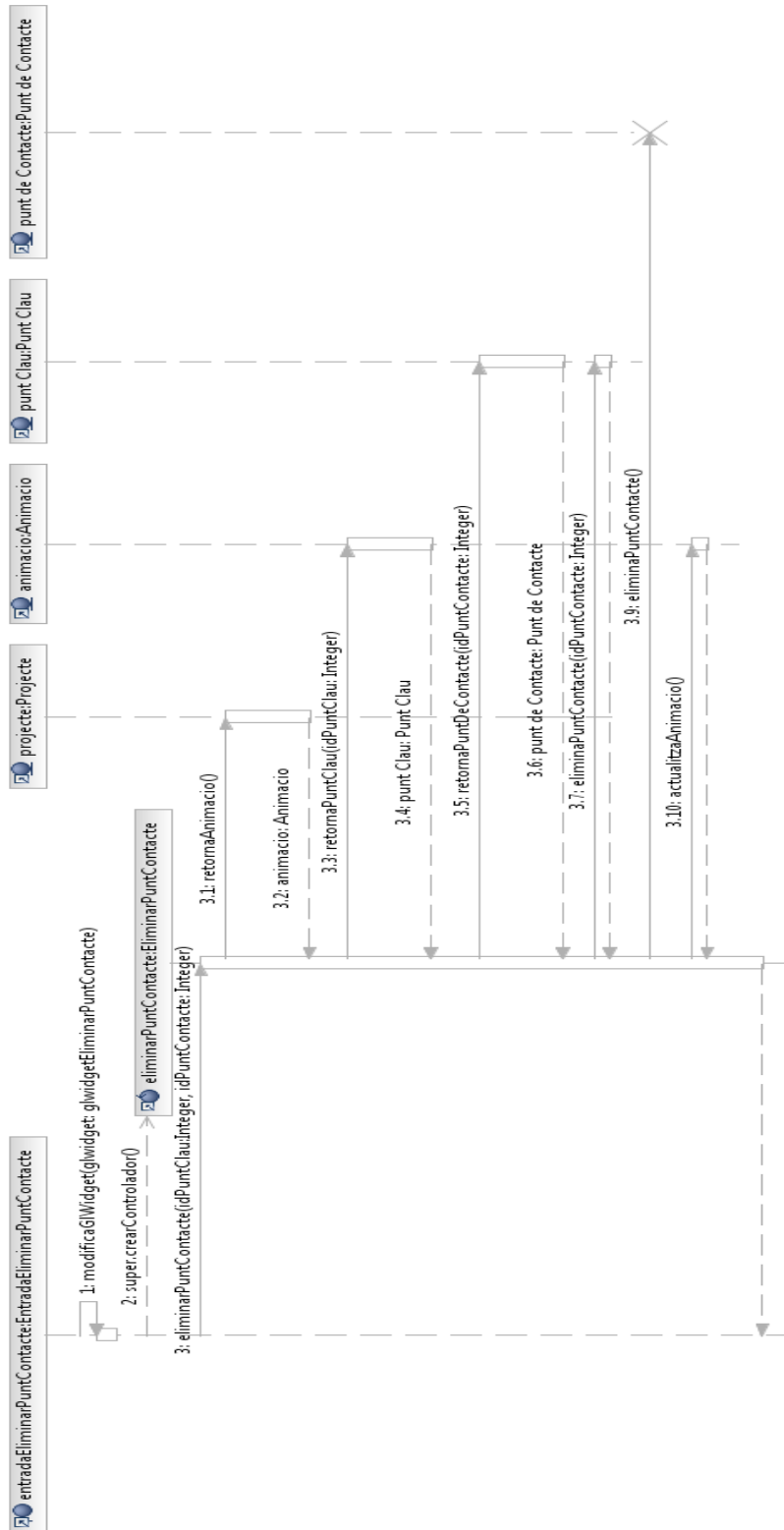


Figura 5.41: Diagrama de seqüència de Eliminar punt de contacte

5.4.38 Cas d'ús 6.4: Modificar punt de contacte

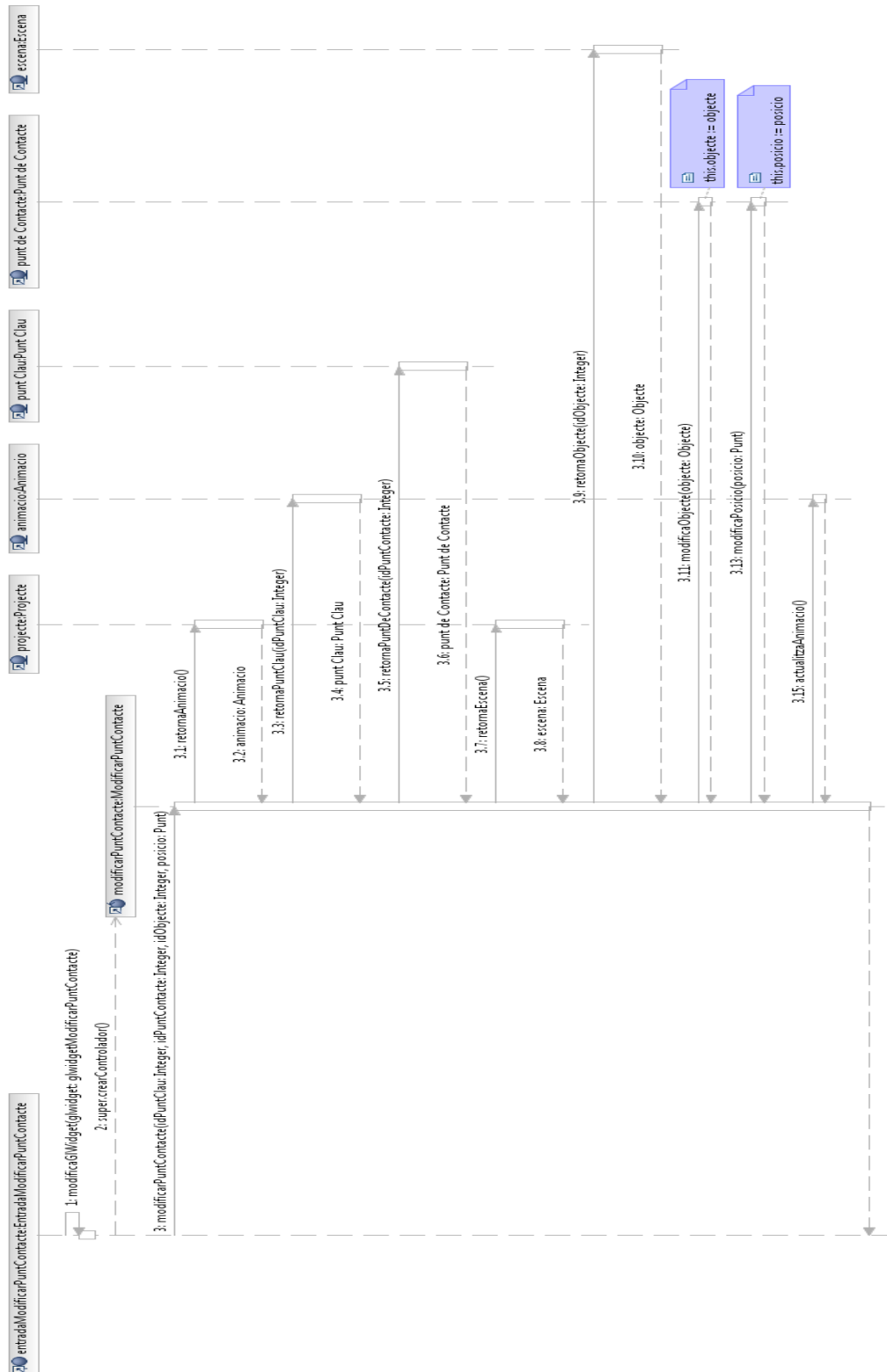


Figura 5.42: Diagrama de seqüència de Modificar punt de contacte

5.4.39 Cas d'ús 6.5: Modificar fotograma de l'animació

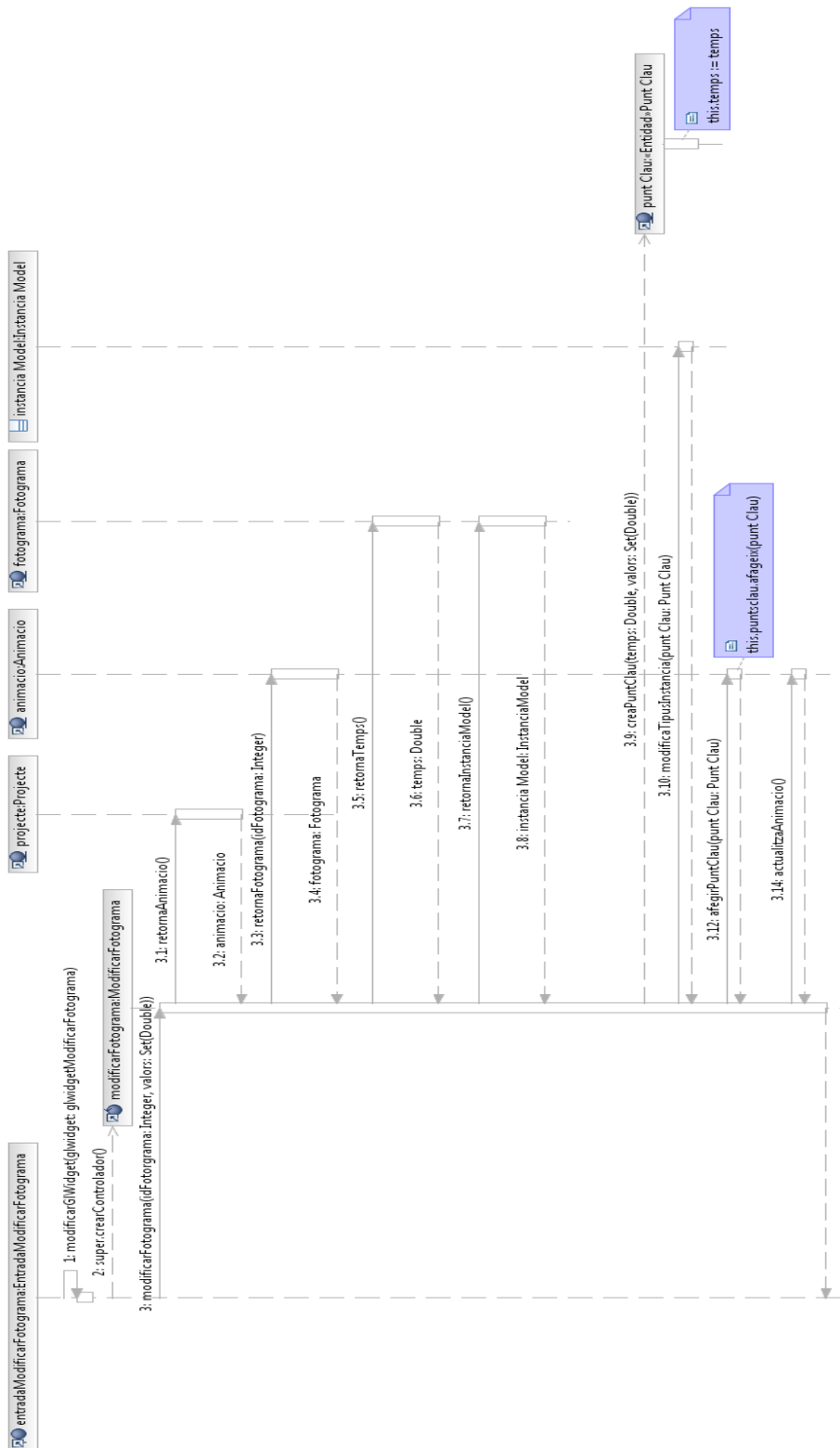


Figura 5.43: Diagrama de seqüència de Modificar fotograma de l'animació

Cas d'ús 6.6: Modificar paràmetre d'una etapa

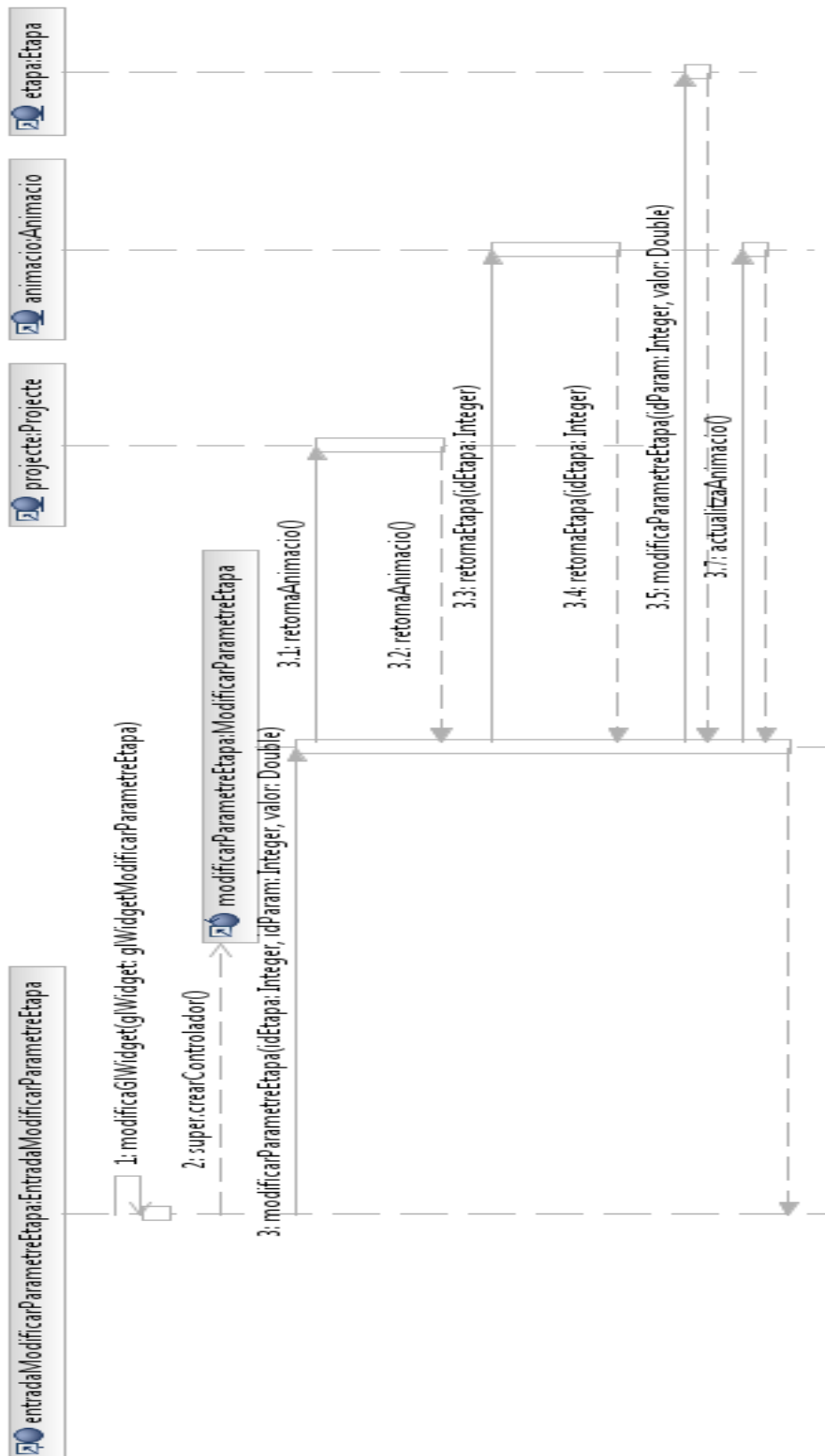


Figura 5.44: Diagrama de seqüència de Modificar paràmetre d'una etapa

Cas d'ús 6.7: Modificar paràmetre d'un Punt Clau

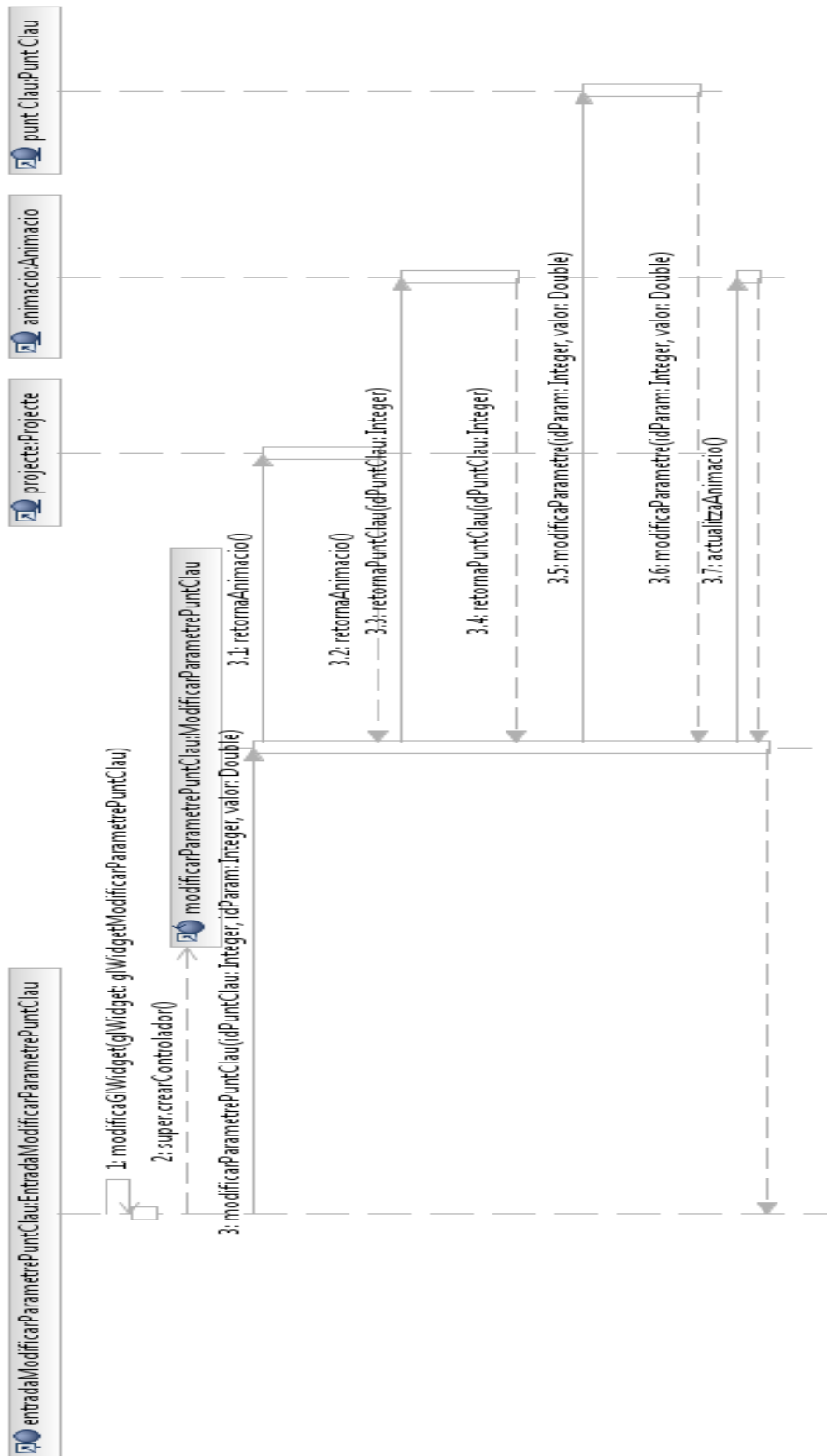


Figura 5.45: Diagrama de seqüència de Modificar paràmetre d'un Punt Clau

5.4.40 Cas d'ús 7.1: Pan

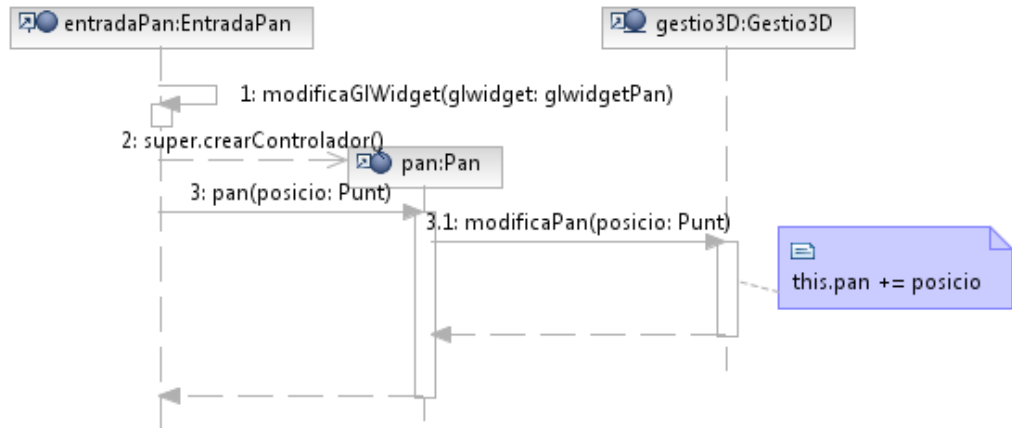


Figura 5.46: Diagrama de seqüència de Pan

5.4.41 Cas d'ús 7.2: Zoom

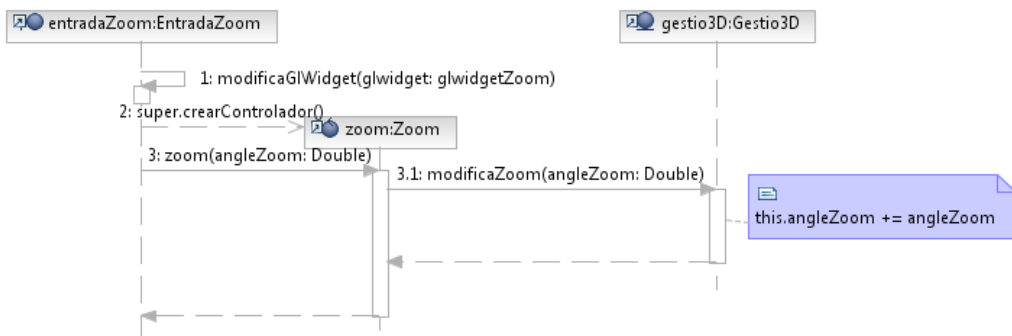


Figura 5.47: Diagrama de seqüència de Zoom

5.4.42 Cas d'ús 7.3: Rotació

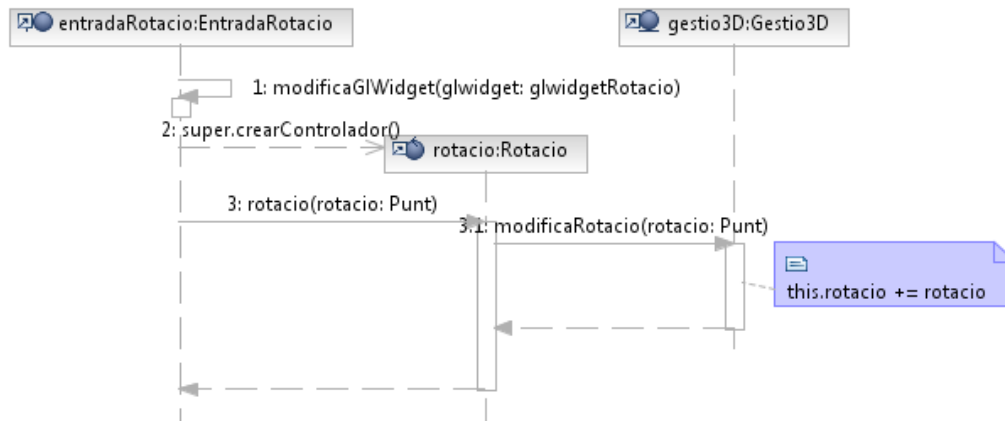


Figura 5.48: Diagrama de seqüència de Rotació

5.4.43 Cas d'ús 8.1: Guardar model

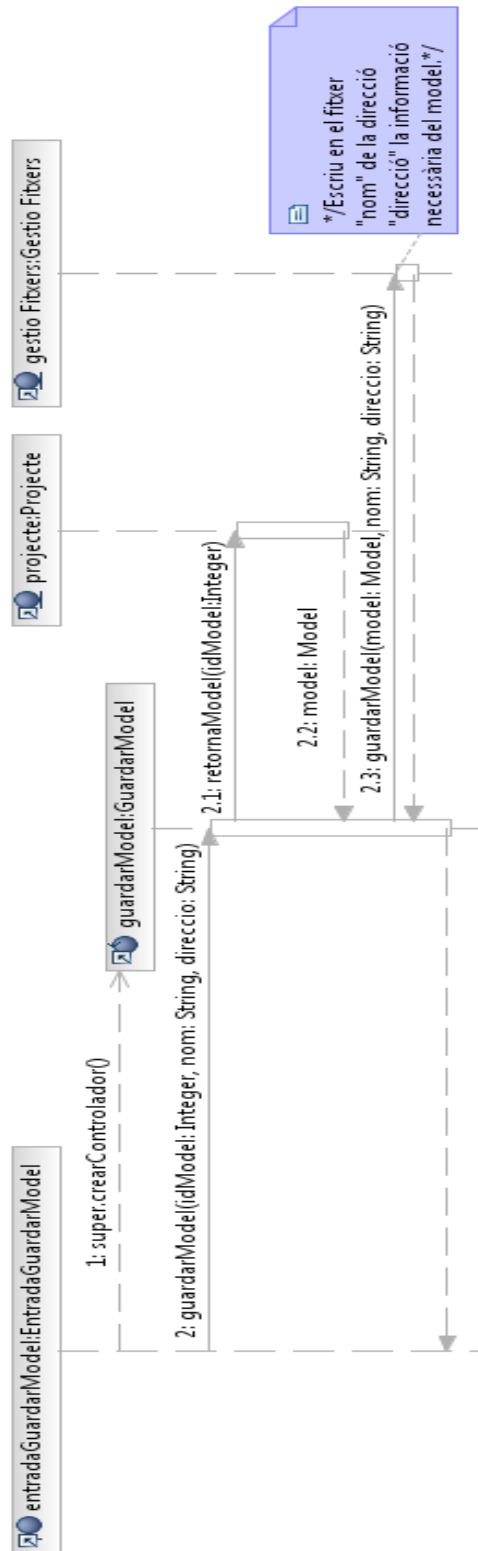


Figura 5.49: Diagrama de seqüència de Guardar model

5.4.44 Cas d'ús 8.2: Carregar model

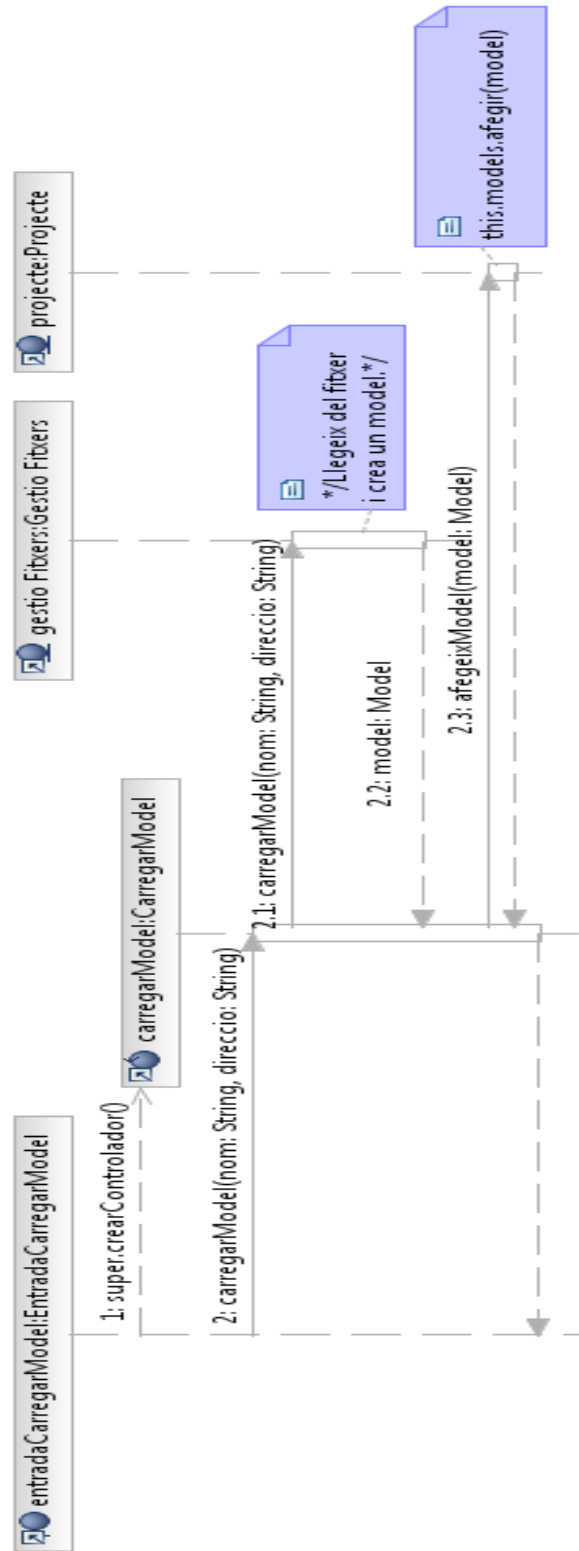


Figura 5.50: Diagrama de seqüència de Carregar model

5.4.45 Cas d'ús 8.3: Guardar escena

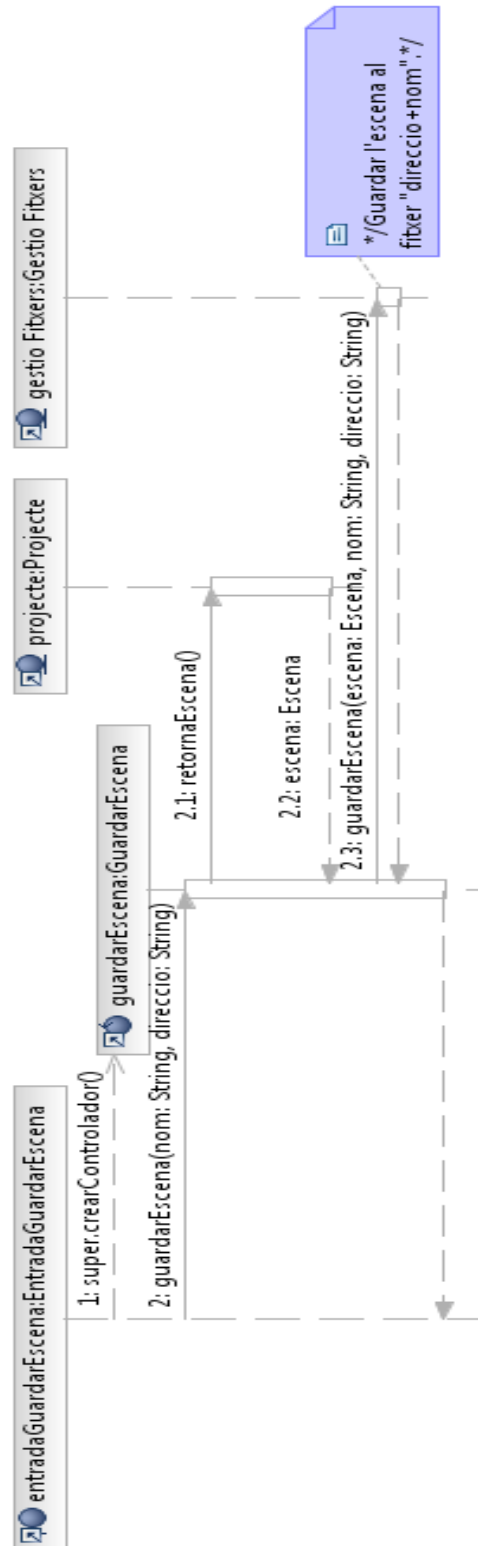


Figura 5.51: Diagrama de seqüència de Guardar escena

5.4.46 Cas d'ús 8.4: Carregar escena

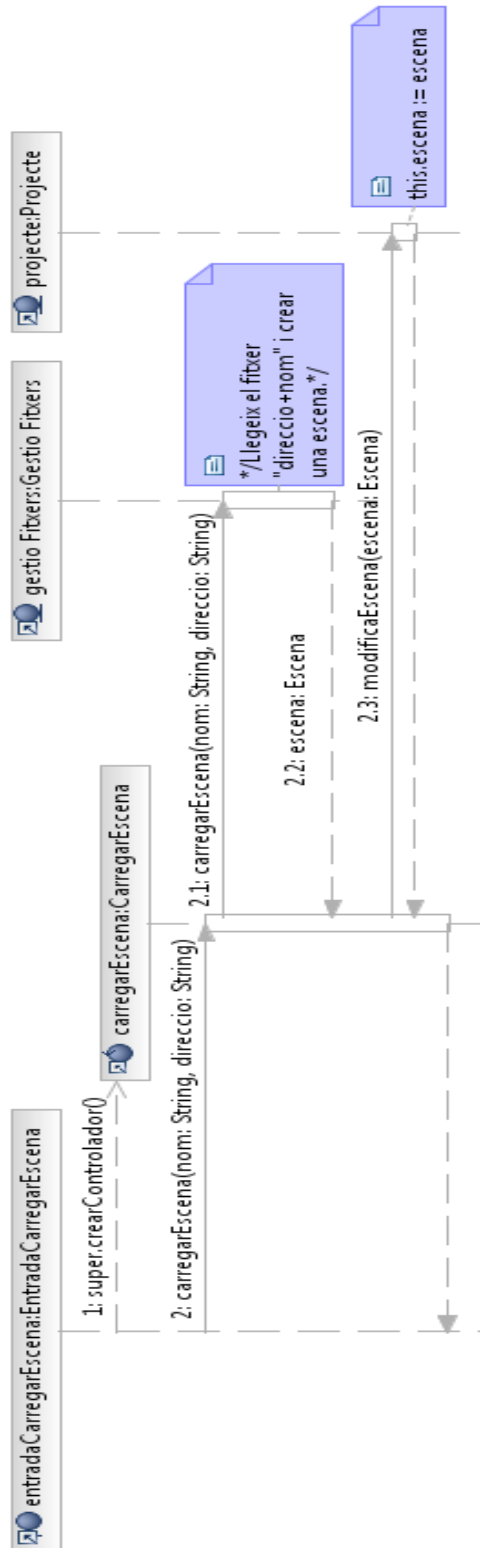


Figura 5.52: Diagrama de seqüència de Carregar escena

5.4.47 Cas d'ús 8.5: Crear projecte

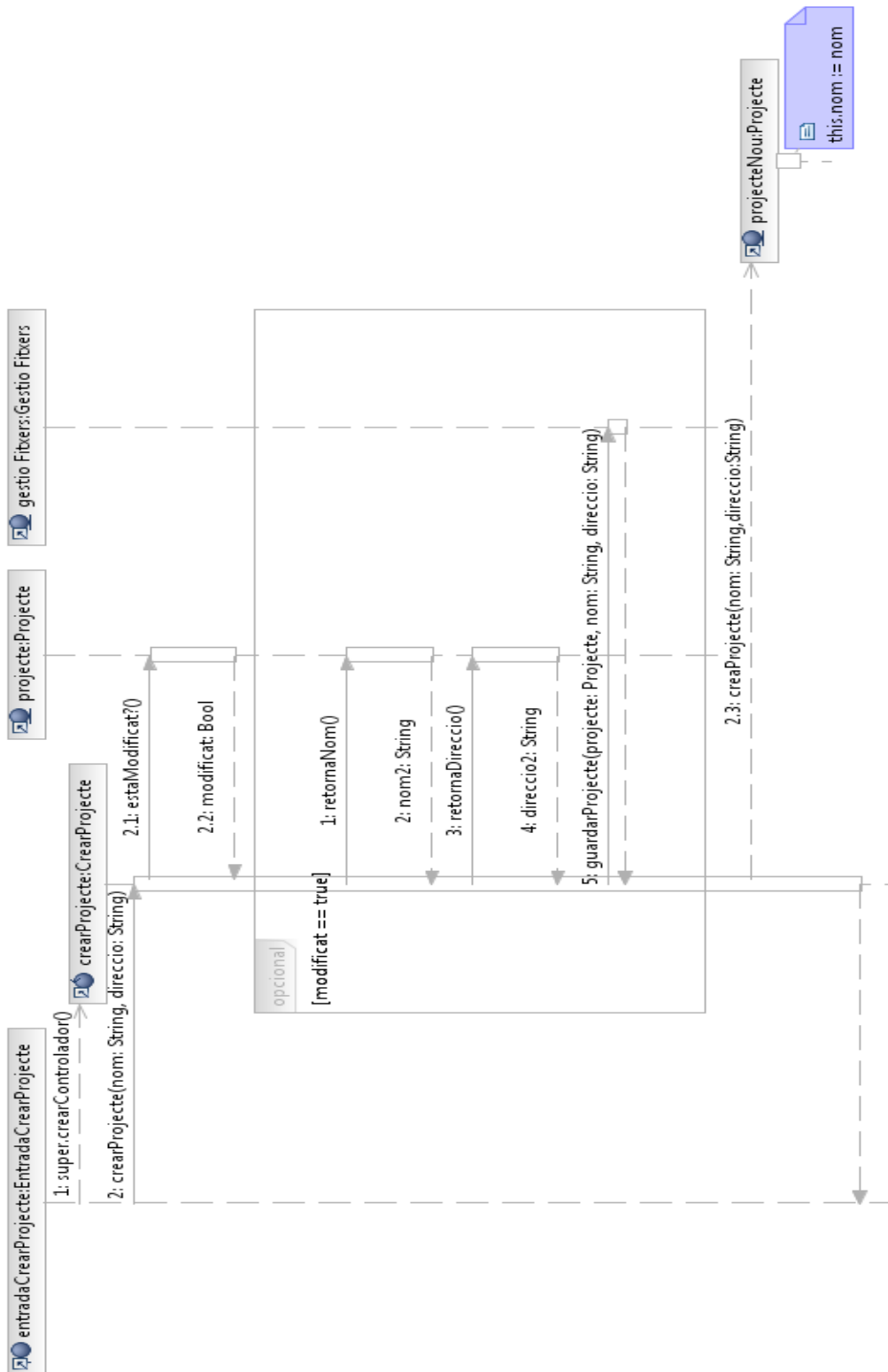


Figura 5.53: Diagrama de seqüència de Crear projecte

5.4.48 Cas d'ús 8.6: Guardar projecte

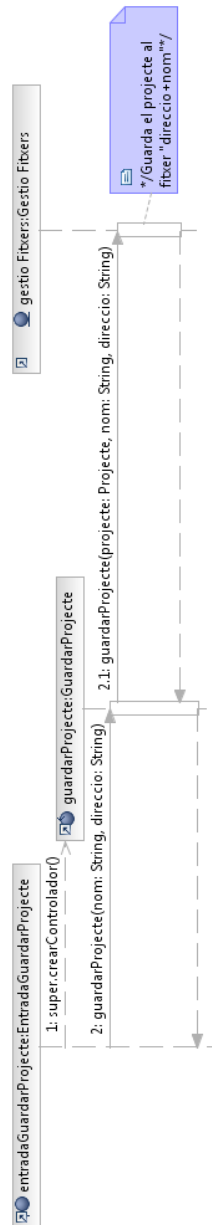


Figura 5.54: Diagrama de seqüència de Guardar projecte

5.4.49 Cas d'ús 8.7: Carregar projecte

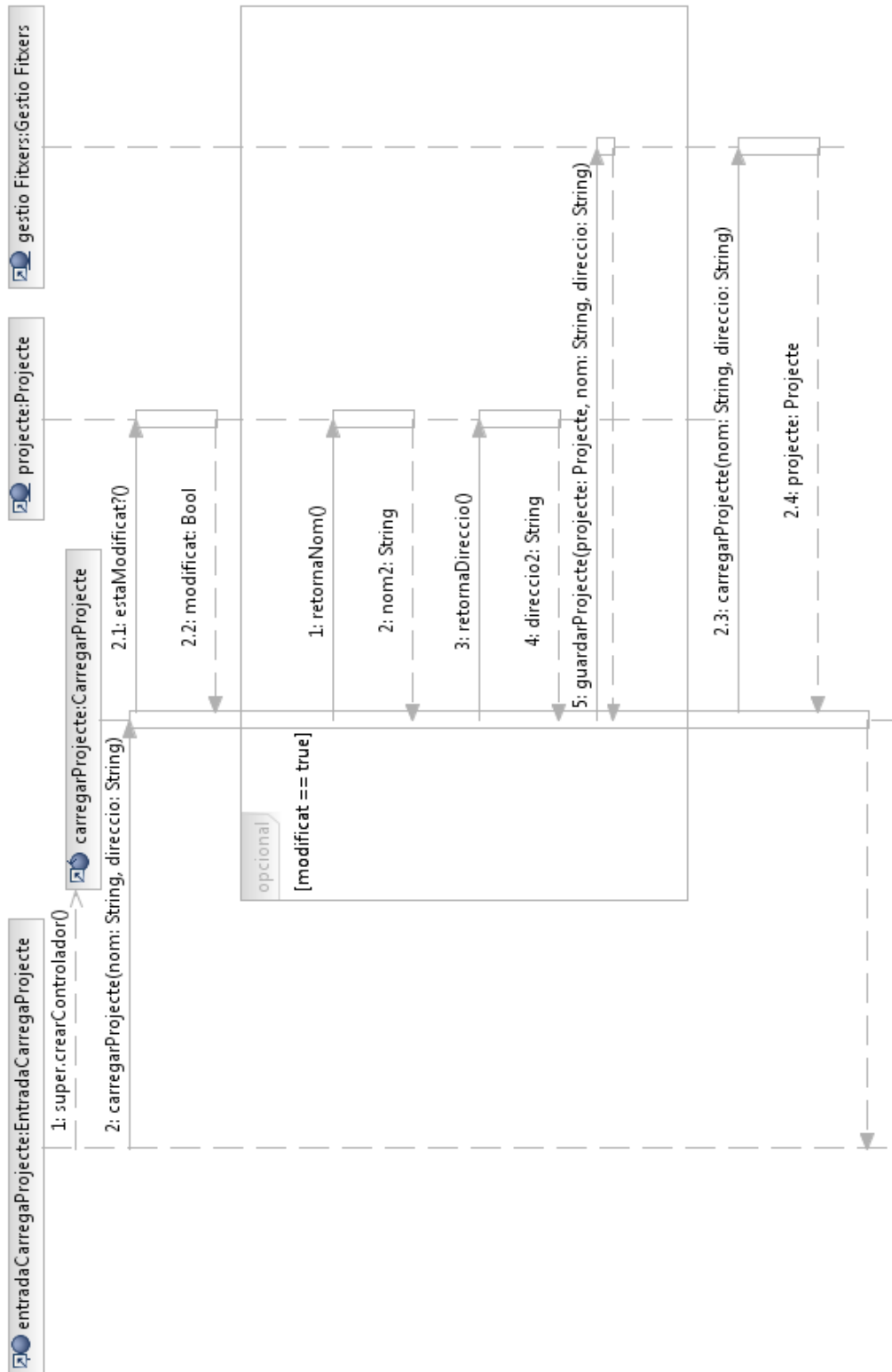


Figura 5.55: Diagrama de seqüència de Carregar projecte

5.4.50 Cas d'ús 8.8: Afegir gràfica

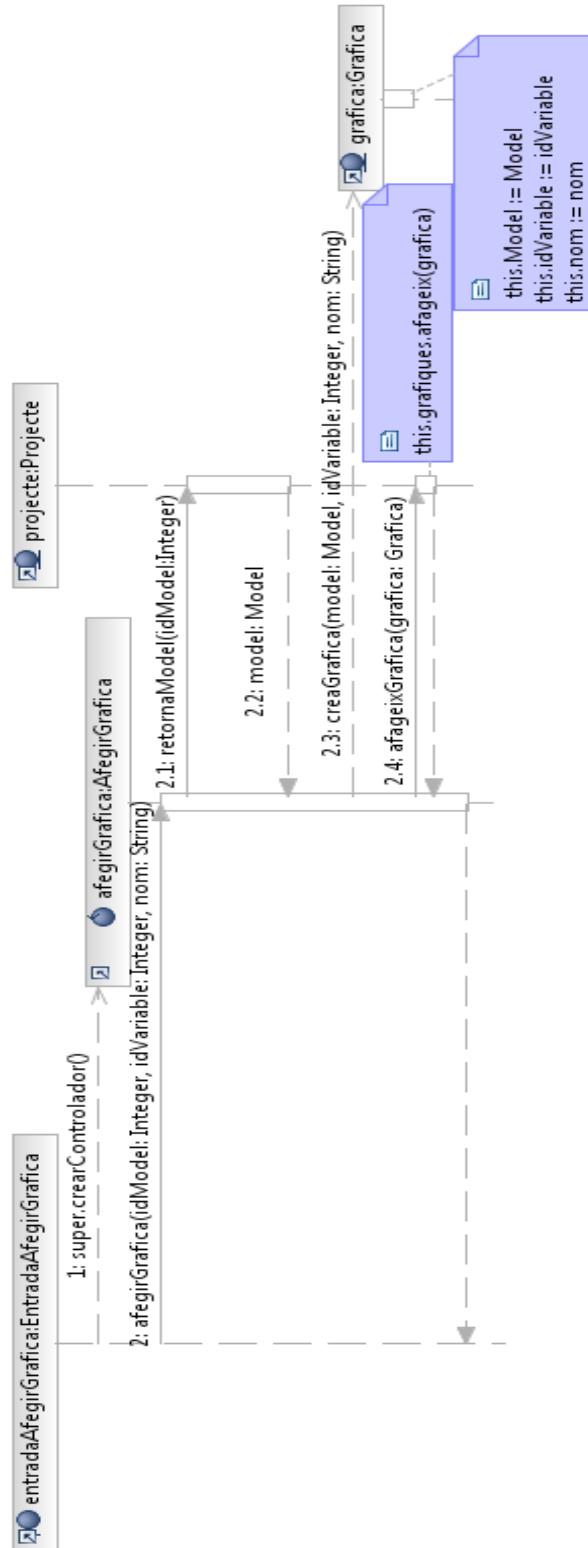


Figura 5.56: Diagrama de seqüència de Afegir gràfica

5.4.51 Cas d'ús 8.9: Eliminar gràfica

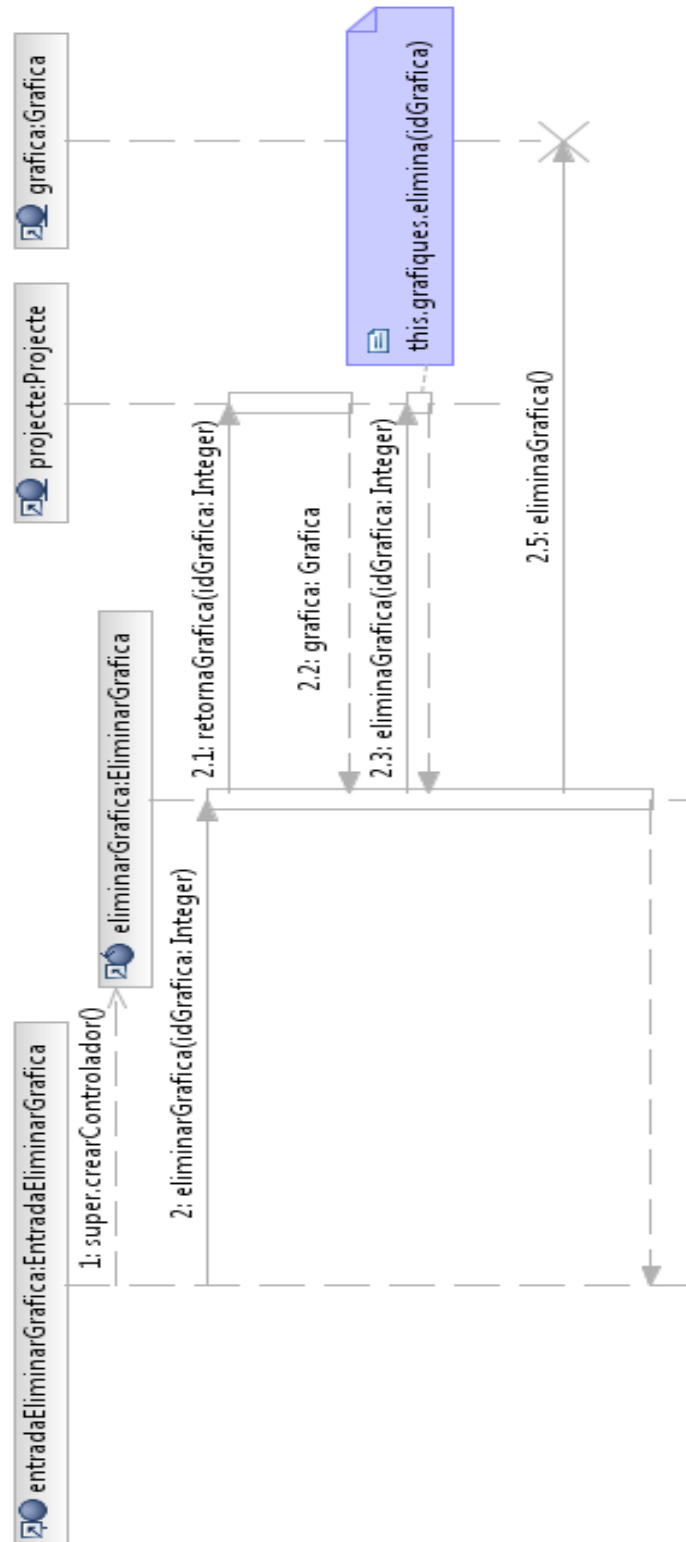


Figura 5.57: Diagrama de seqüència de Eliminar gràfica

5.4.52 Cas d'ús 8.10: Visualitzar gràfica

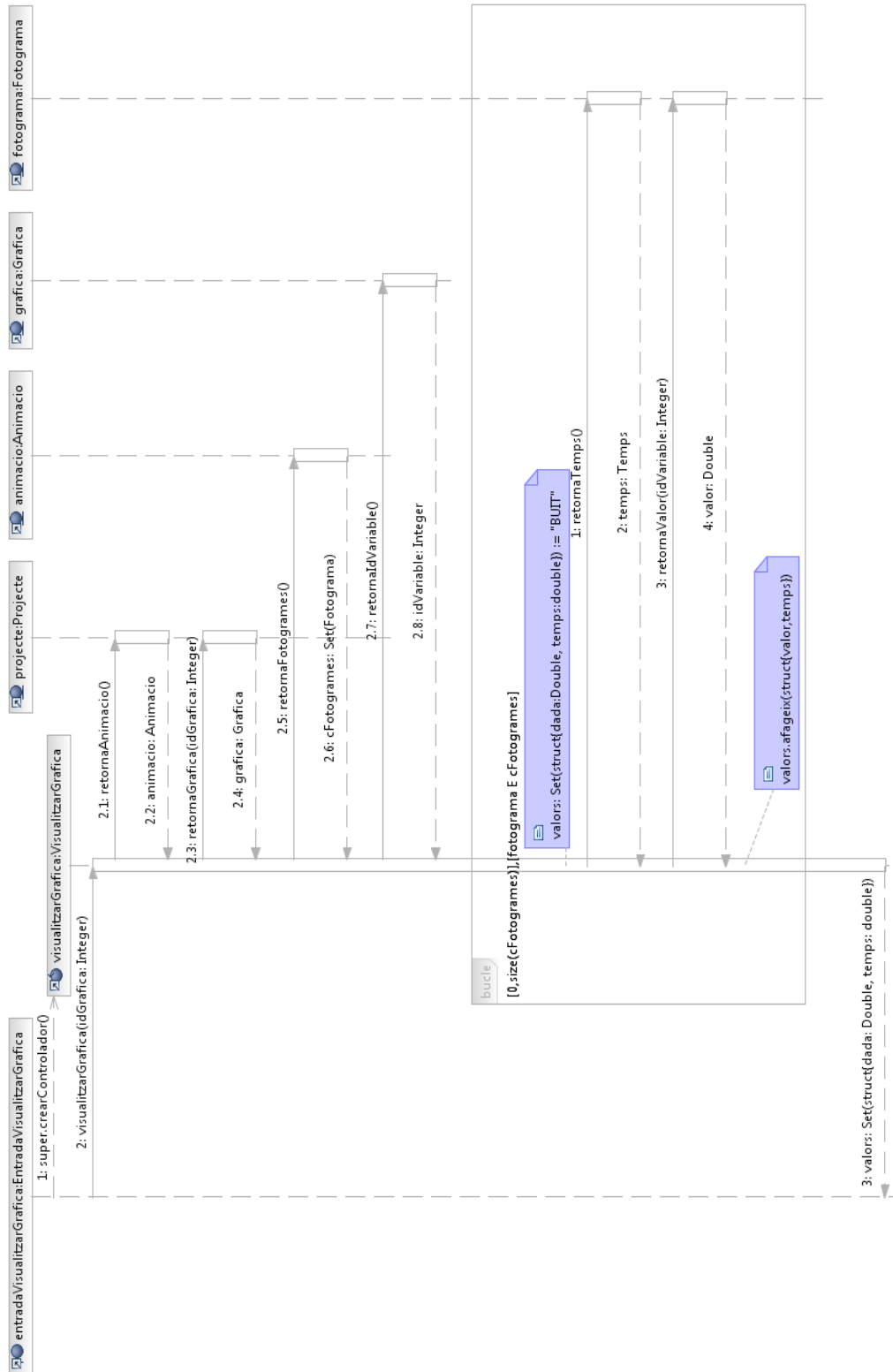


Figura 5.58: Diagrama de seqüència de Visualitzar gràfica

5.5 Disseny de la interfície

En aquesta secció es farà l'especificació sobre el disseny de la interfície gràfica mostrant l'aspecte extern que veurà l'usuari, quines possibilitats d'interacció té l'usuari i les diferents formes amb les que es pot trobar l'usuari.

La pantalla principal és la següent:

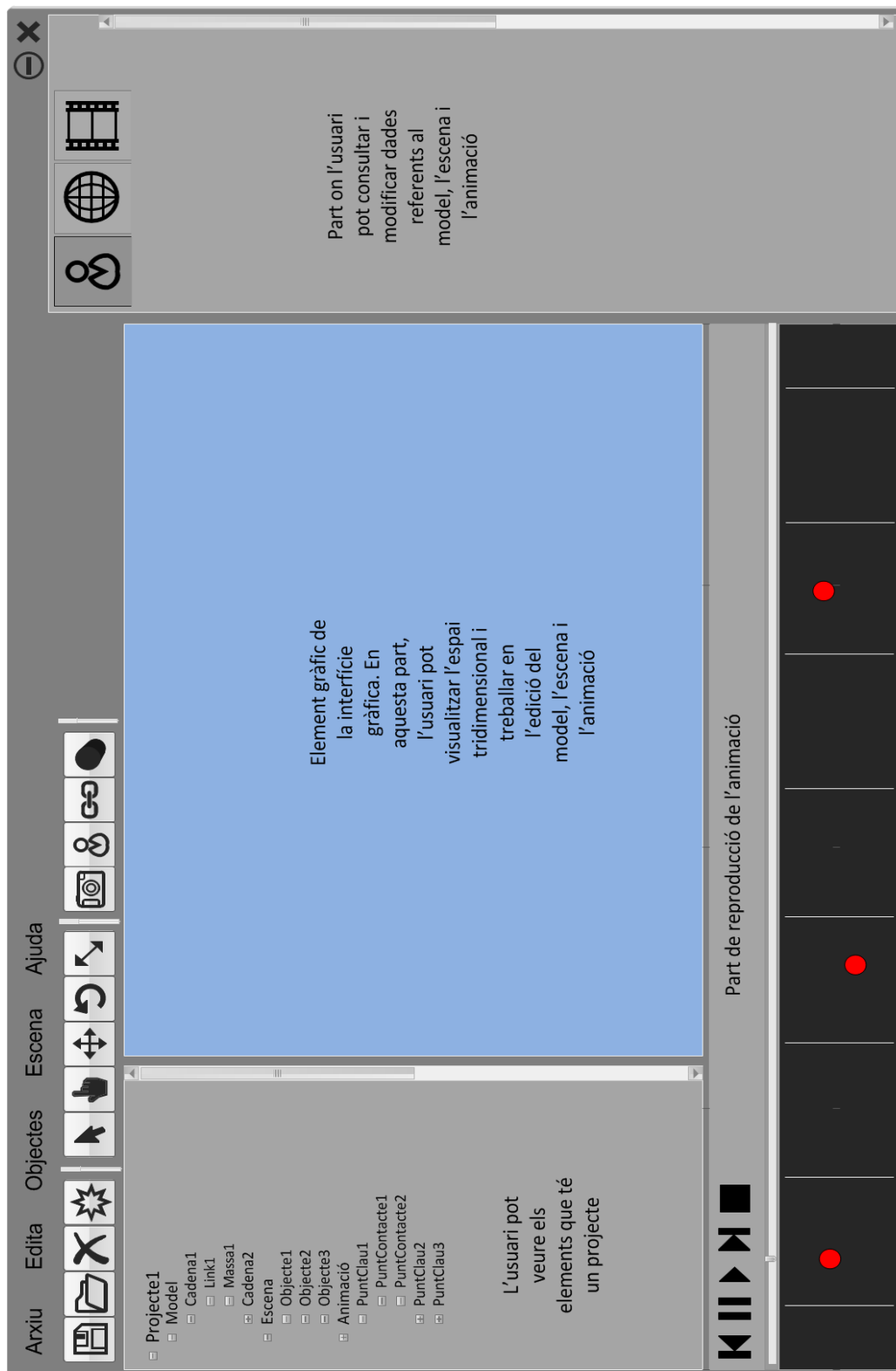


Figura 5.59: Interfície gràfica principal

En aquesta interfície es poden veure cinc zones importants de cara a l'usuari. Una primera on es troben els menús i els botons que donen a l'usuari les funcionalitats necessàries per realitzar el projecte. A la part esquerra, es té una finestreta que mostra els elements del projecte (escena, model i animació) on l'usuari pot seleccionar de manera més ràpida. La part central que conté la visió de l'espai tridimensional. La part de la dreta que és la finestra on es pot consultar i modificar informació en relació al model, l'escena i l'animació. I finalment la part de sota de tot, tenim el reproductor de l'animació on l'usuari pot veure els punts claus i moure's pels diferents fotogrames de l'animació.

Capítol 6

Pla de test, proves i recerca realitzada

6.1 Introducció

En aquest capítol el que s'explicarà serà quina metodologia s'ha seguit per comprovar el bon funcionament del programa, l'estat amb el que es troba actualment i amb què s'ha dedicat el temps de recerca. Per veure el tipus de proves i la metodologia es veurà un parell d'exemples de com s'ha aplicat la metodologia a la pràctica i a partir d'aquí entendre millor com s'han comprovat cada una de les dades que el programa proporcionava o que tractava. Finalment, s'explicarà amb quins temes s'ha investigat i quines conclusions s'han tret d'aquestes recerques.

6.2 Pla de test

En aquest tipus de projecte, s'han trobat bàsicament dos tipus de proves, aquelles que tenien a veure amb la interacció usuari-sistema i que el sistema havia de primer de tot satisfer les necessitats de l'usuari, i després s'hi era una entrada de dades o modificació d'aquestes havia de fer que la informació fos consistent i coherent amb lo que ja es tenia. I per altre banda, hi havia les proves referents als càlculs físics. Aquests últims calia una manera diferent de comprovació i requeria un altre sistema com el Matlab que ens ajudés a verificar els càlculs, sobretot els que feien referència al Jacobià, i al càlcul del moment angular.

Pel que fa a les proves del primer tipus, anomenades bàsiques, es centren en comprovar que el sistema garanteix el bon compliment dels cassos d'ús. Hi ha proves més o menys complexes, alguns cassos d'ús afecten a la capa de domini, i per tant, modifiquen el contingut del programa. N'hi ha d'altres, que només s'usen per visualitzar informació i per tant, no modifiquen la capa de domini i n'hi ha d'altres, que comporten llegir o escriure a fitxers i poden o no canviar l'estat de la capa de domini i la comprovació es fa més complicada. Per cada una d'aquestes proves bàsiques, el que s'ha fet és una plantilla amb la qual apuntar diferents dades per tal de tenir localitzat el problema i com resoldre'l.

La plantilla següent és la forma que tenia el full de detecció i resolució d'errors amb la qual s'ha treballat. En ella, s'ha de poder reflectir tot el que l'usuari ha de veure, el que ha modificat i el que hauria d'haver passat. Si no es compleix en una primera instància, caldrà fer una nova versió i comprovar si l'error persisteix. A continuació la plantilla:

Cas d'ús:

Nom prova:

Versió:

Supera?: ok/ko

Descripció:

Resultats esperats a la Capa de Presentació:

Resultats esperats a la Capa de Domini:

Resultats esperats a la Capa de Gestió:

Resultats a la Capa de Presentació:

Resultats a la Capa de Domini:

Resultats a la Capa de Gestió:

Problemes a resoldre:

Una vegada definida la plantilla que recull els resultats de les proves, el mecanisme que s'usa per resoldre aquest tipus de problema es basa en la capacitat que té Qt Creator per deixar en pausa l'execució del programa i poder veure el contingut de les variables de la Capa de Domini. Per altre banda, si es vol comprovar si la Capa de Presentació funciona correctament, es pot comprovar fent servir l'eina de Qt Creator, i també observant el comportament que pren. Poder parar el programa en un punt concret i poder observar quina informació s'ha guardat i quina no, ajuda molt a identificar problemes en el codi. No hi ha una metodologia exacte per aquests cassos, només saber on col·locar els punts on vols que es pari el programa per després poder observar i analitzar si és el que s'espera o no.

Per altre banda, com s'ha comentat, també s'ha hagut de testejar totes aquelles funcions referents a la part física i que són internes al programa. No són senzillament dades que l'usuari entra sinó que són dades que el sistema dedueix i que requereixen un tractament més complex. Com ja s'ha nomenat el Jacobià que sorgeix de l'estructura del model i la configuració que pren en un cert punt. En aquest cas, trobar errors en aquests cassos es fa més complex. Requereix l'ús d'un programa extern de simbologia matemàtica, o que pugui tractar fórmules i expressions simbòliques de manera que introduint les fórmules adequades puguem obtenir els resultats d'una manera més còmode i senzilla i després poder-ho contrastar amb els resultats del programa. En el nostre cas, s'ha fet servir el programa Matlab.

De totes maneres, cal entendre que aquest tipus de metodologia funciona bé quan es va poc a poc, és a dir, no es pot començar a comprovar el Jacobià si no concorden les matrius de transformació, o que el producte de matrius no és correcta. Cal doncs, pas a pas anar comprovant que tot funciona com toca per arribar a l'objectiu final.

La manera de com es comprova aquests tipus de problemes amb el programa pot variar de comprovar a ull amb l'eina d'aturada de Qt Creator, a escriure els valors per la sortida estàndard o amb un fitxer i contrastar-ho amb els valors del Matlab de la manera més còmode. Tot això varia en funció del grau de complexitat de les dades. No és el mateix comprovar una matriu 4×4 que un Jacobià de $n \times m$.

Per acabar, amb totes les proves que s'han fet, es pot dir que el programa està fora de errors bàsics. Està clar que encara falta molt per fer una Capa de Presentació còmode i que no requereixi molt coneixement per part de l'usuari, cal agafar més experiència en aquests sentit. S'ha intentat cobrir els cassos bàsics, però per la complexitat de la realitat de segur que no s'han cobert tots. Cal dir també que el sistema pot no donar valor per qualsevol entrada, buscarà la solució però pot ser que en algun moment es perdi (bàsicament, que se li demani al model un pas impossible), degut també a la gran llibertat que es dóna per generar moviments. De totes maneres, el programa es troba en fase beta, havent superat tots els testos bàsics i ara es troba en el punt d'agafar experiència per poder cobrir el màxim ventall de moviments.

6.3 Proves d'exemple

En aquest subapartat, es mostrarà algun exemple de com s'ha analitzat un cas d'ús i algun cas de la part física i a partir d'aquí poder entendre com s'ha fet tota aquesta part.

6.3.1 Exemple 1: Comprovació del funcionament del cas d'ús Crear Model

El procediment a seguir és el següent:

1. Escriure el nom del cas d'ús en aquest cas Crear Model.
2. Escriure el nom de la prova en aquest cas "Comprovació de la correcta inserció del Model".
3. Escriure el número de versió que s'ha fet d'aquesta prova en aquest cas 1.0.
4. Escriure a la descripció que el que es vol analitzar és que el model ha de ser afegit al projecte de manera consistent i coherent.
5. Als Resultats esperats de la Capa de Domini posarem que esperem que afegixi un model amb el valor de id 1000000, amb la posició inicial a (0, 0, 0) i una orientació final igual a (0, 0, 0), el nom ha de ser "Model", amb les Variables del model amb mida 0 i el número de cadenes a 0.
6. Col·locar el punter de pausa adequadament dins del codi del program i executar el cas d'ús i veure els valors.
7. Confirmar que es dona la situació ideal i donar per passat el programa.

A continuació es presenta la plantilla degudament omplerta i després la comprovació feta en punt 6:

Cas d'ús:

Crear Model

Nom prova:

Comprovació de la correcta inserció del Model

Versió:

1.0

Supera?: ok/ko

Descripció:

Amb aquesta prova es vol analitzar que el model que es vol afegir sigui exactament com està definit.

Resultats esperats a la Capa de Presentació:

Cap

Resultats esperats a la Capa de Domini:

El model afegit ha de tenir el valor de id 1000000, amb la posició inicial a (0,0,0) i una orientació final igual a (0,0,0), el nom ha de ser "Model", amb les Variables del model amb mida 0 i el número de cadenes a 0.

Resultats esperats a la Capa de Gestió:

Cap

Resultats a la Capa de Presentació:

Cap

Resultats a la Capa de Domini:

Els valors són els esperat. (Veure imatge de sota)

Resultats a la Capa de Gestió:

Cap

Problemes a resoldre:

Cap

Una vegada omplert el full, la comprovació de la prova queda per conclosa i es pot passar a la següent prova. A continuació es mostra la imatge que mostra que es compleix tots hi cada un dels valors demanats. Si en algun moment més endavant el programa fallés per aquest cas d'ús caldria generar una nova versió de prova i tornar a testejar-lo. La imatge que mostra els resultats és la següent:

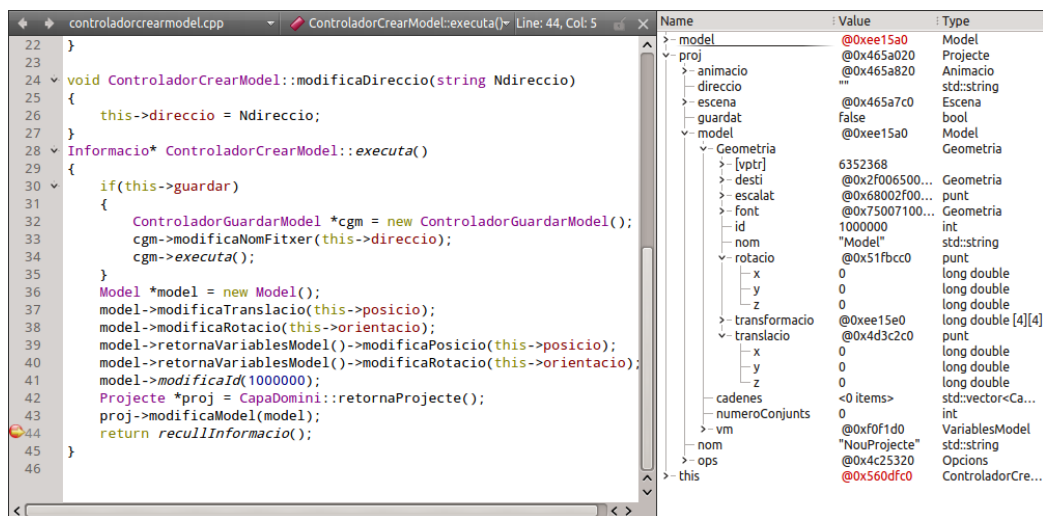


Figura 6.1: Captura de l'aturada del programa amb Qt Creator que mostra el cas d'ús Crear Model

Com es pot veure a la imatge, hi ha un bola vermella a la línia 44, així es com es marca per parar l'execució del programa en aquest punt. Quan hi ha una fletxa a sobre ens diu que l'execució està parat en aquest punt. A la part dreta de la imatge es pot veure com apareixen els valors dels atributs del model entre altres coses. Allà es pot comprovar cada una de les condicions que requeria el programa. Dóna molta llibertat, ja que pots col·locar tants punts vermells com es vulgui i si l'execució passa per aquell punt el sistema s'atura i deixa mostrar tota la informació que es pugui accedir en aquell moment.

Una vegada mostrat com se sol actuar en aquests cassos més bàsics, passem a una de les comprovacions que s'han fet per validar la part física.

6.3.2 Exemple 2: Comprovació del càlcul de la derivada de la transformació que comporta una articulació al llarg d'una cadena

Seguidament veurem la metodologia per comprovar que els càlculs fets amb el programa de la part física siguin els correctes. Farem un programa amb el Matlab per calcular la derivada de la transformació d'un articulació. Com que sabem que la transformació es correspon amb una rotació sobre

l'eix Z, la matriu de transformació es fàcilment calculable amb el Matlab. Cal tenir en compte que la funció del Matlab tindrà com entrada un angle que correspondrà al valor de l'articulació, així doncs:

```

1 - clear;
2 - x = sym('x');
3 - mat = [cos(x) -sin(x) 0 0;
4 -       sin(x) cos(x) 0 0;
5 -       0 0 1 0;
6 -       0 0 0 1];
7 - matD = diff(mat,x);
8 - angle = 17.6*pi/180;
9 - subs(matD,x,angle)
10
11

```

Command Window

```

ans =

    -0.3024    -0.9532         0         0
     0.9532    -0.3024         0         0
         0         0         0         0
         0         0         0         0

```

Figura 6.2: Captura de l'obtenció de la derivada d'un gir respecte l'eix Z amb Matlab

Com podem veure, definim el gir sobre l'eix Z com una matriu homogènia i al aplicar-li la derivada obtenim el valor de la matriu. Per obtenir el valor

de la matriu del programa, es pot fer un mòdul de comprovacions amb el qual es cridi la funció quan es necessiti fer aquest tipus de comprovacions. A continuació es mostra una captura mostrant els valors de la matriu aturant l'execució:

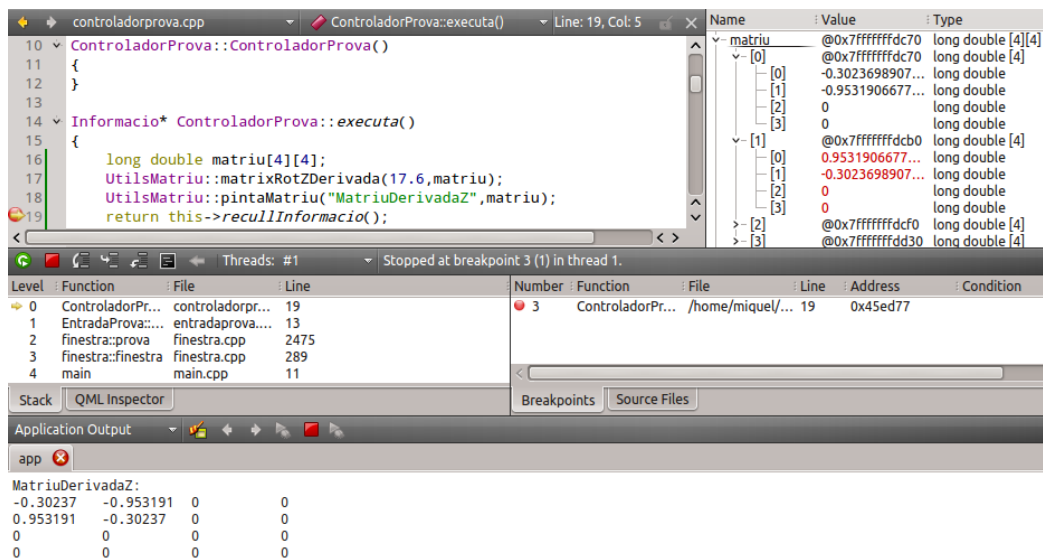


Figura 6.3: Captura de l'obtenció de la derivada d'un gir respecte l'eix Z amb el programa

Vist el procediment, és fàcil veure que per comprovar si es calcula bé el Jacobià o fins hi tot una animació, els fitxers de Matlab es fan més complexos, però no tant, en comparació amb els fitxers del programa que per si sols seria molt complicat trobar si existeix algun error.

Vistos els dos exemples, queda més clar quina és la metodologia per comprovar totes i cada una de les funcions que té el programa. Finalment, s'explicarà a grans trets la feina de recerca que s'ha fet i quins aspectes s'han estudiat amb més profunditat per intentar millorar els càlculs de l'animació i que l'usuari hagi de proporcionar menys informació per aconseguir el mateix resultat.

6.4 Recerca

La part d'investigació ha ocupat una part important del projecte, s'ha investigat sobretot el comportament, en diferents cassos, de tècniques per generar animacions. Hi ha tècniques que han estat assequibles i d'altres que no semblaven funcionar, però totes intentaven optimitzar el moviment i fer més estable el model durant el moviment.

Una de les coses amb que més s'ha jugat, ha estat el tema de les Qs que apareix al capítol 1. Les Qs, és una tècnica per la qual ens pot donar un gran ventall de solucions a l'equació que descriu la física del model del sistema. Es poden calcular diferents Qs que ajudin al model a aconseguir arribar a posicions estables mitjançant l'acceleració o la reducció del creixement dels graus de llibertat, o Qs que facin disminuir el creixement de la obertura d'una articulació si aquestes arriben al seu límit.

S'han fet diferents proves amb diferents models i amb diferents moviments, en el primer tipus de Qs, la conclusió que s'arriba és que és una tècnica útil quan el model és capaç d'arribar a aquella configuració que descriuen les Qs, però sinó el sistema es perd. La segona és més fiable, però tot hi així si un moviment té una dependència molt gran d'una articulació que està al límit del seu rang de valors tampoc assolirà l'objectiu.

Per altre banda, també s'ha investigat en el tema de la trajectòria que segueix el Centre de Massa del model durant una etapa, en aquest cas, s'ha intentat minimitzar el moment angular, minimitzar la distància del punt d'aplicació de la força de contacte òptim amb el real i s'ha intentat minimitzar el moment angular entorn del Centre de Massa. En totes aquestes solucions, amb certs cassos és vàlid, però la realitat dicta que un model mal posionat necessita un moment angular i un moment lineal que l'ajudi a recuperar l'equilibri i de moment el sistema no garanteix això perquè no entén de mal posicionament o de mal equilibri en un model. Seria un punt mol interessant a seguir investigant més endavant.

La conclusió a la que s'arriba del resultat del procés de recerca és que la majoria de tècniques són vàlides, l'única cosa és que s'ha de saber quan aplicar-les, de fet a la realitat quan un neix ha de aprendre a controlar el

seu cos i en certa manera aprendre a fer servir tècniques que ens l'ajuden a millorar el seu moviment.

Capítol 7

Cost econòmic i planificació

En aquesta part, s'explicarà el cost econòmic que ha tingut el projecte i la seva distribució de la feina. Cal remarcar que el projecte en total ha durat 4 quadrimestres amb total activitat en els 4 descansant només durant l'estiu del 2013. Cal dir també que uns dies més o uns dies menys però de mitjana es calcula que s'han dedicat unes 40 hores setmanals. En una setmana normal, la planificació en hores era la següent:

Dia de la setmana	Horari
Dilluns	9h-13h,15h-20h
Dimarts	9h-13h,15h-17h
Dimecres	9h-13h,15h-20h
Dijous	9h-13h,15h-17h
Divendres	9h-13h,15h-17h

Taula 7.1: Distribució de les hores en una setmana normal

Ja se sap que un dia es fa més i un altre menys, però l'horari que s'intentava complir era aquest, són més de 40 hores, però se'n contarán només 40. Si tenim en compte que un quadrimestre aproximadament està compost per 16 setmanes, podem concloure que les hores totals de treball seran 2560 hores dedicades a realitzar aquest projecte. La planificació, es centra bàsicament en 4 perfils de treball, primer com analista i dissenyador, segon com a programador, tercer com a cap de projecte i finalment com a investigador. Cada un d'aquests perfils té un preu per hora segons la importància i l'impacte que té en el producte final. Així doncs, la distribució aproximada de les hores dedicades amb cada un dels perfils amb el preu per hora i el seu total queda:

Perfil	Hores	Preu per hora	Total
Analista i Dissenyador	640 hores	20 Euros/hora	12800 Euros
Programador	960 hores	15 Euros/hora	14400 Euros
Cap de projecte	560 hores	25 Euros/hora	14000 Euros
Investigador	400 hores	22 Euros/hora	8800 Euros
			37400 Euros

Taula 7.2: Distribució de les hores en perfils i càlcul dels costos a partir dels sous

Segurament el cost final, sembla excessiu però de fet, s'han dedicat moltes hores a aquest projecte i és això el que fa pujar el seu cost. De totes maneres, cal tenir en compte que és un sistema que no requereix hardware addicional per obtenir dades de l'exterior.

Capítol 8

Conclusions

Primer de tot dir, que després de realitzar el projecte, estic molt satisfet i orgullós de tota la feina realitzada. S'han assolit els objectius de anàlisi, disseny i implementació. El programa està en fase Beta, i és l'etapa final d'un programa. El que queda ara és mantenir i actualitzar-lo.

Pel que fa a la part més de coneixements de Física s'han assolit a nivell intel·lectual i s'han aplicat al programa de manera correcta. No era una part trivial, venint per part meva del sector de la Informàtica, però he sabut adaptar-me en la major part de les situacions i he acabat entenent els conceptes que el meu tutor m'anava ensenyant.

S'han aplicat els coneixements adquirits en assignatures referents a l'Enginyeria del Software, a Projectes de Programació i de Visualització Gràfica i s'han combinat per realitzar el projecte. Pel que fa a la part informàtica del projecte han estat útils tots els coneixements adquirits i no ha calgut buscar informació més enllà dels apunts d'aquestes mateixes assignatures, o consultant algun manual específic necessari per a la programació.

És un projecte tendre encara, però ambiciós, la porta que obra a estudiar nous camps de la física relacionats amb el que treballa aquest projecte és molt gran degut a la no limitació del tipus de models. El sistema és capaç de tractar diferents tipus de models i fer-los interactuar amb l'escena i obtenir moviments i estudiar-los cas per cas.

Com a conclusió final, estic convençut que totes hi cada una de les hores dedicades en aquest projecte no han estat envà i que molts altres les aprofitaran per si algun dia, necessiten explorar nous moviments amb nous models.

Bibliografia

- Apunts de l'assignatura Projecte d'Enginyeria del Software i Base de Dades.
- Apunts de Física del Modelatge i Animació Realista.
- Apunts de Visualització i Interacció Gràfica.
- Apunts de Visualització Avançada.