

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Creació i optimització de mecanismes mecànics senzills

Document: Memòria

Alumne: David Bruguera Tornés

Tutor: Dr. Gustavo Patow / Dr. Jordi Regincós

Departament: Informàtica, Matemàtica Aplicada i Estadística

Àrea: LSI

Convocatòria (mes/any) Juny / 2017

1.- Introducció	3
1.1 Motivacions	5
1.2 Propòsits	5
1.3 Objectius	5
2.- Estudi de viabilitat	6
2.1.- Recursos humans	7
2.2.- Recursos tècnics	8
2.3.- Cost econòmic	9
2.4.- Recapitulacions	12
3.- Metodologia	14
4.- Planificació	19
5.- Marc de treball i conceptes previs	21
5.1.- Simulació de maquinària	22
5.2.- Algorismes genètics	23
6.- Requisits del sistema	25
6.1 Plantejament del problema	26
6.2 Plantejament de la solució	27
6.3 Requisits funcionals	27
6.4 Requisits no funcionals	28
7.- Estudi i decisions	29
7.1.- Autodesk Inventor	29
7.2.- C#	31
7.3.- Microsoft Visual Studio 2017	35
7.4.- Llibreria genètica AForge	36
7.4.- Sistema operatiu	37
8.- Anàlisi i disseny	38
8.1.- Fitxes i diagrames de cas d'ús	38
8.2.- Diagrama de classes	41
9.- Implementació i proves	46
9.1.- Fitxer de configuració	47
9.2 Inicialització de l'aplicació	49
9.3 Botó AG optimizer	50
9.3.1 Inicialització de les variables	51
9.4 Problemes durant el treball	55
10.- Resultats	56
10.1 Prova 1	56
10.2 Prova 2	60

11.- Conclusions finals	64
12.- Treball de futur	68
13.- Bibliografia i webgrafia	70
14.- Annexos	71
14.1 Requisits per a Autodesk Inventor 2017	71
15.- Manual d'usuari i instal·lació	78
15.1 Instal·lació	79
15.2 Entorn de desenvolupament	79
15.3 Execució	80
15.4 Alertes habituals	84
15.3 Disseny de Fitxers XML	86

1.- Introducció

En aquest primer apartat del document és on se situa l'espai del projecte i en el qual s'exposen els motius que en justifiquen el desenvolupament; així mateix, s'hi estableixen els objectius inicials.

Des de ja fa unes dècades, podem adonar-nos, sense massa dificultats, que en el món de la informàtica s'han produït canvis que han comportat una evolució molt important, significativa i transcendental, tant entre les persones usuàries com entre els professionals i les empreses. La finalitat de canvis tan notables ha consistit en la voluntat de millorar-los la vida professional, però, també, la més domèstica i quotidiana. No obstant això, tot i que un dels principals objectius de les noves tecnologies és el de facilitar la millora del dia a dia dels llocs de treball i, alhora, realitzar i desenvolupar eines que siguin intel·ligents, fàcils d'utilitzar i que, a més a més, facin estalviar temps, esforç i, en conseqüència, diners, cal anotar que totes aquestes millores i tots aquests avantatges que s'han anat desenvolupant, no s'apliquen amb la mateixa intensitat a tots els camps en els quals serien possibles d'aplicar; un exemple d'això el podem localitzar, sense cap mena de dubte, en un àmbit tan quotidià i habitual com és en la construcció de mecanismes.

En aquesta línia, el treball que realitzarem i presentarem té com a finalitat la creació i optimització de mecanismes mecànics senzills (com ara, per exemple, el que es pot observar a la Figura 1). En el fons, seguint les recomanacions del nostre tutor, el Dr. Gustavo Patow, hem desenvolupat una continuació d'un anterior treball de fi de grau que, exitosament, va defensar el curs 2015, a la Universitat de Girona, l'Aura Ruiz

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

Duñach amb el seu projecte titulat *Llenguatge d'assemblatge per a la construcció i simulació de mecanismes senzills*. De fet, des d'un punt de vista de conceptes, continguts i d'estructura de treball, aquesta recerca ens ha resultat una eina molt útil, alhora que fonamental, per realitzar la nostra pròpia proposta, raó per la qual les referències que s'hi faran en el decurs de la nostra memòria seran habituals ja que, en el fons, com hem suggerit anteriorment, el nostre pretén ser un complement al que hem esmentat més amunt, seguint les recomanacions que ens va donar el nostre tutor.



Figura 1- Exemple de mecanisme d'un rellotge, basat en engranatges.

Finalment, volem indicar que la documentació i la informació que s'inclou a la present memòria s'estructura en quinze punts, la totalitat dels quals abraça, òbviament, des de la generació inicial de la idea fins a les conclusions finals del projecte, sense ometre, com correspon, la planificació, el desenvolupament i, finalment, la fase de testeig.

1.1 Motivacions

Per al desplegament i la realització del disseny d'una màquina, cal dur a terme un bon estudi de disseny, de viabilitat i d'assemblatge. D'aquesta manera, per al disseny dels mecanismes és convenient de desenvolupar tot un procés i unes tècniques d'optimització amb la finalitat darrera de poder optimitzar l'acció i l'efecte d'assemblatge d'un programa i tenir, amb el consegüent estalvi de temps, més marges de beneficis; no obstant això, tot aquest procés encara se sol realitzar a mà o bé fent ús d'uns paràmetres predefinits, en el fons, poc àgils. És en aquesta línia, des d'aquesta perspectiva de treball, que nosaltres desenvolupem el projecte amb els propòsits i objectius finals que tot seguit apuntem.

1.2 Propòsits

El propòsit d'aquest treball consisteix, fonamentalment, en el fet de facilitar, ajudar i ser un bon suport als dissenyadors de mecanismes basats en engranatges, amb la finalitat darrera, en el fons, d'anar reduint el temps i els diners que fins ara s'han hagut d'anar invertint en aquestes fases inicials del procés de construcció de mecanismes, la qual cosa implicaria que s'hi hauria produït una significativa optimització en el decurs del procés que resultaria, considerem, més àgil i senzilla i, per tant, molt menys costosa.

1.3 Objectius

L'objectiu principal d'aquest treball és pensar, reflexionar i dissenyar un procés ràpid que automatitzi tot l'esmentat procés de disseny d'assemblatge mecànic. De manera

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

fàcil, econòmica i senzilla, permetre de formar un conjunt d'engranatges concrets, posant un objectiu (relació de transmissió) a assolir amb els pertinents engranatges donats i, finalment, facilitar de modificar els paràmetres de la funció de *fitness* amb l'objectiu final de poder controlar la correcta execució del programa.

Amb aquesta finalitat, doncs, el treball es durà a terme mitjançant un llenguatge que permeti algorismes genètics i que, al mateix temps, permeti i faciliti, posteriorment, la simulació del sistema d'engranatges resultant. Per a assolir-ho, les diverses tasques que caldrà anar realitzant són les següents:

- Estudi acurat del funcionament de l'API d'Autodesk Inventor.
- Estudi del funcionament dels sistemes d'engranatges i de la seva transmissió de relació.
- Desenvolupament d'un algorisme genètic que permeti l'optimització de sistemes d'engranatges.
- Desenvolupament de les estructures de dades adequades per tal que tot funcioni correctament.
- Finalització i arrodoniment de la documentació del treball realitzat.

2.- Estudi de viabilitat

En aquest segon apartat de la memòria és on es recullen tots els paràmetres que han fet possible el desenvolupament del projecte, tant els recursos tècnics com els humans així com també les tecnologies que hi han estat emprades.

2.1.- Recursos humans

Des d'un punt de vista referit a les persones que cal que hi participin (o de la plantilla de personal en casos professionals), el projecte ha requerit de diferents rols per a poder realitzar-se adequadament i correctament, tals com:

- **Client:** és el que posa en marxa, qui origina el projecte i, al cap i a la fi, qui en defineix les especificacions finals.
- **Analista:** es tracta de l'encarregat d'entendre i interpretar les necessitats plantejades pel client.
- **Arquitecte de software:** és l'encarregat de transformar els requisits de l'analista en requisits tècnics.
- **Arquitecte de sistema:** és el responsable de pensar el sistema mitjançant el qual s'executarà el programari.
- **Desenvolupador de software:** és l'encarregat d'escriure el codi i de fer tots els *refactorings* adequats i corresponents una vegada es consideri com a finalitzat el projecte.
- **Tester:** o provador de software, és la persona que té cura de realitzar els testos amb la finalitat de provar que el projecte no contingui *bugs* i, per tant, de validar els resultats que es vagin obtenint.

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

- **Gerent del projecte:** és el responsable de planificar el projecte i de cercar les solucions adequades als problemes localitzats que poguessin presentar-se i que, al mateix temps, és possible que arribessin a posar en perill el correcte acompliment del projecte.

Malgrat que hem definit tots aquests diferents rols indispensables per a l'adequada realització i posterior desenvolupament del projecte, en realitat, per facilitar-ne la feina, en el nostre cas, tractant-se d'un projecte de fi de grau, ens els hem repartit entre l'alumne mateix (David Bruguera, que ha realitzat les tasques d'arquitecte de software, arquitecte de sistema, desenvolupador de software i *tester*) i el cap de projecte (Gustavo Patow, al qual li ha correspost de fer de client, d'analista i de gerent del projecte).

2.2.- Recursos tècnics

Els recursos tècnics dels quals ens hem anat servint per a fer possible l'execució i el desplegament del projecte són els següents:

- Autodesk Inventor: recurs tècnic que ens ha servit per a poder realitzar totes les diverses simulacions dels resultats obtinguts.
- Visual Studio: recurs que hem utilitzat amb la finalitat de poder programar amb C# i, al mateix temps, per poder controlar l'entorn de simulació de l'Autodesk Inventor

Aquest programari utilitzat s'executa sobre el sistema operatiu Windows 10 de 64 bits amb un processador i7 de 4a generació de la sèrie HQ amb una gràfica gtx

860M i amb 8Gb de memòria RAM, és a dir, en realitat ho hem executat amb un PC estàndard amb l'únic condicionant que s'ajusti als requeriments de hardware, amb això ja n'hi ha prou.

2.3.- Cost econòmic

En aquest apartat, detallarem, en la mesura del possible, el cost econòmic que considerem que comporta tot el que hem anat esmentant anteriorment, tant des del punt de vista de la tecnologia, com dels recursos humans, diferenciant, però, el que ha resultat per a nosaltres com a estudiants universitaris del que podria representar la seva aplicació en una empresa.

En referència a l'aspecte dels recursos tècnics, hem d'anotar que, des d'un principi, ja disposàvem de tots els recursos tecnològics necessaris que hem esbossat a l'apartat anterior, per la qual cosa, tot plegat no ens va representar inicialment cap tipus de despesa o inversió econòmiques, tot hi així contarem com si n'haguéssim comprat un ja que faria falta per a tenir-lo en compte així que usarem la següent fórmula:

$$\text{Amortització} = \frac{\text{Preu recurs} * \text{mesos feina}}{36 \text{ mesos}}$$

L'amortització pel nou ordinador, amb un preu total de 960€, i una integració en el projecte de 12 mesos, és de:

$$\text{Amortització} = \frac{960 * 12}{36} = 320 \text{ €}$$

En referència a l'aspecte del programari, cal indicar que, en disposar d'un compte d'estudiant gràcies a la Universitat de Girona, també ens va resultar tot el software

Creació i optimització de mecanismes mecànics senzills

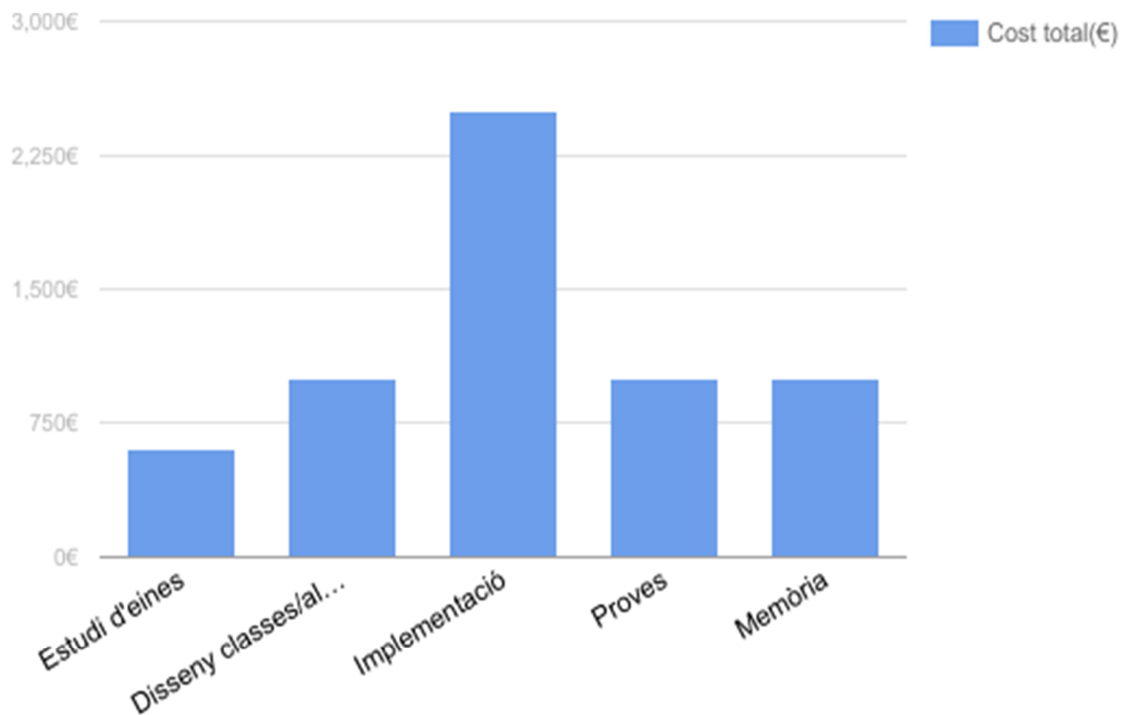
David Bruguera Tornés

gratuït, partint de la base, però, que aquestes llicències tenen limitacions en tractar-se de llicències per a estudiants, la qual cosa vol dir que, si s'apliqués professionalment el projecte s'haurien de tenir en consideració altres paràmetres d'índole econòmica, o sigui, que, indubtablement, les despeses econòmiques variarien. D'aquesta manera, per exemple, una subscripció anual professional a Autodesk Inventor costa 2.060 euros (segons consulta feta a dia 29 de maig de 2017) tal i com hem recollit a <https://www.autodesk.es/products/inventor/subscribe> i l'adquisició del Visual Studio Professional 2017 té un preu de 574 euros i una subscripció anual bàsica es troba a partir de 918 euros (segons consulta realitzada a dia 29 de maig de 2017) , tal i com hem pogut consultar i comprovar a https://www.microsoftstore.com/store/mseea/es_ES/list/Visual-Studio/categoryID.66235600

Des del punt de vista de recursos humans, de persones treballant-hi, cada professional comporta els seus costos, a un preu que oscil·la entre els deu i vint euros l'hora, així que, per fer-los més visibles, els definim a la següent taula (vegeu Taula 1) amb la seva corresponent gràfica (vegeu Gràfica 1):

Taula 1. Costos dels recursos humans dividits en els diversos professionals (Font: elaboració pròpia)

	Hores	Cost(€/h)	Cost total(€)
Estudi d'eines	30	20	600€
Disseny classes/algorismes	50	20	1.000€
Implementació	250	10	2.500€
Proves	100	10	1.000€
Memòria	50	20	1.000€
COST TOTAL	480		6.100€

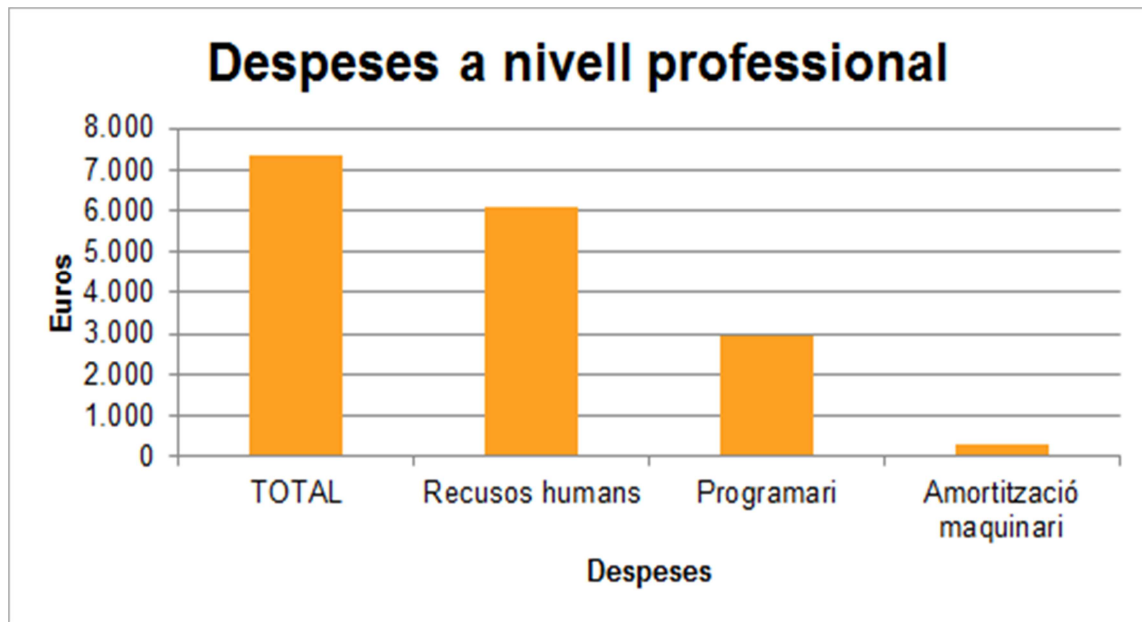


Gràfica 1. Despeses en recursos humans on podem comprovar que el més costós és la implementació (Font: elaboració pròpia)

Tanmateix, i com ja avançàvem a l'inici d'aquest apartat, malgrat que les despeses per a nosaltres hagin resultat a cost zero perquè disposàvem dels recursos tècnics propis -a banda de la despesa derivada de la utilització de l'ordinador que, evidentment li provoca un desgast que a mitjà o llarg termini comportarà el seu canvi-, com a estudiants gaudíem del programari gratuït i entre el tutor del treball i nosaltres mateixos vam realitzar tots els rols humans que calia sense cap tipus de remuneració econòmica; si això s'hagués d'aplicar professionalment, en un àmbit professional, costaria, aproximadament, uns 6100 euros en recursos humans i 2978 euros en software -sempre que es tractés de subscripcions anuals a Autodesk Inventor i Studio Professional-, per tant, un total final aproximat de 7.078 euros i,

sumant-hi els costos d'amortització del maquinari, quedaria en 7.398 euros.(vegeu

Gràfica 2)



Gràfica 2. Despeses aproximades que comportaria el projecte en un àmbit professional: un total de 7.398 euros dividits en 6.100 de recursos humans, 2.978 de programari i uns 320 d'amortització de maquinari.

2.4.- Recapitulacions

Tal i com hem pogut constatar a partir de la lectura dels apartats anteriors del treball, a nivell tècnic i tecnològic el resultat final considerem que està garantit atès que ja es posseeixen les eines disponibles per a la seva realització i bon funcionament; a banda que també existeixen les llicències d'estudiants amb la finalitat d'abaratir costos; unes despeses que, malgrat això, variarien si es realitzés un desenvolupament professional ja que, aleshores, hauríem d'afegir-hi les subscripcions a Autodesk Inventor i a Visual Studio i, alhora, a nivell de recursos humans, de presència de professionals, si haguéssim hagut de contractar alguna altra persona, a part de la despesa d'amortització del maquinari, és a dir, que si la

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

tasca no l'haguéssim pogut desenvolupar íntegrament entre nosaltres mateixos i el tutor del projecte, les despeses es podrien considerar prou significatives ja que estarien al voltant dels set mil cinc-cents euros. Però, tractant-se de nosaltres mateixos els qui realitzem les diferents tasques, el cost final ens ha resultat zero, a banda de la indicada amortització del maquinari..

3.- Metodologia

En aquest tercer punt, és en el qual plantejem la metodologia que s'ha utilitzat durant el desenvolupament del projecte que s'ha escollit i, de la mateixa manera, els motius pels quals s'ha realitzat aquesta elecció en detriment d'altres possibilitats també existents..

Són moltes i diverses les metodologies de desenvolupament que existeixen a l'actualitat, moltes d'elles prou eficients i amb característiques pròpies que les diferencien de les altres, algunes ja amb molta tradició, com per exemple la de *Waterfall* o Model de Desenvolupament en Cascada, que tenim esmentada des de l'any 1956 (vegeu Figura 2), *Spiral* o Model Espiral, documentada des de 1986 (vegeu Figura 3) i d'altres de més actuals i modernes, en les quals la interactuació i el treball en equip és fonamental, com ara *Agile*, o Àgil, documentada des de l'any 2001 (vegeu Figura 4)

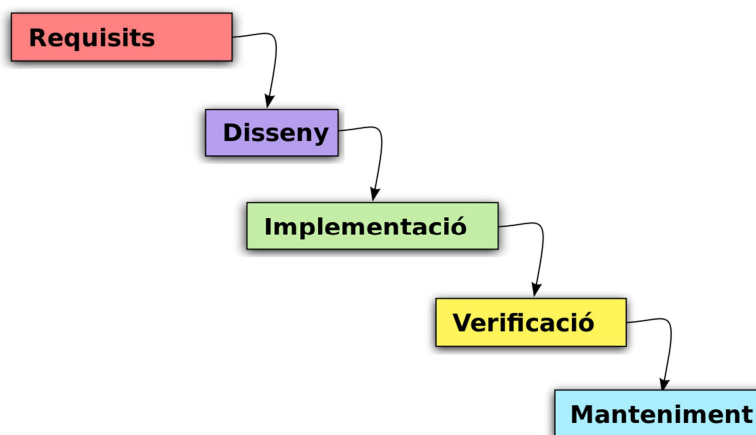


Figura 2. Exemple de metodologia de desenvolupament Waterfall (Font: Viquipedia.org)

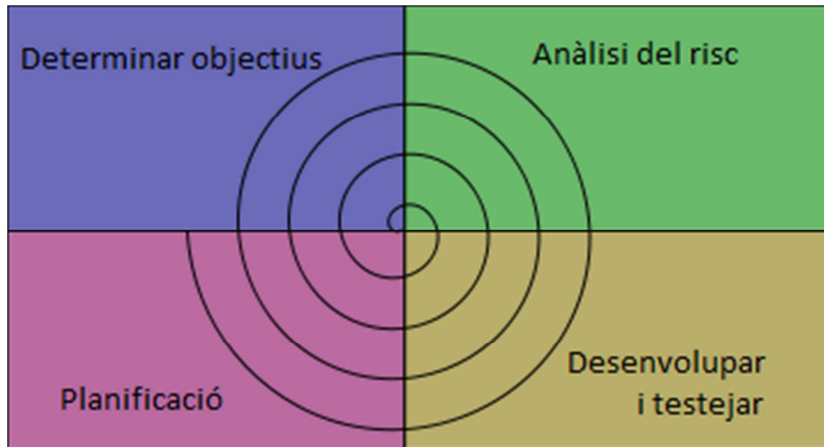


Figura 3. Exemple de metodologia de desenvolupament Spiral (Font: Viquipedia.org)

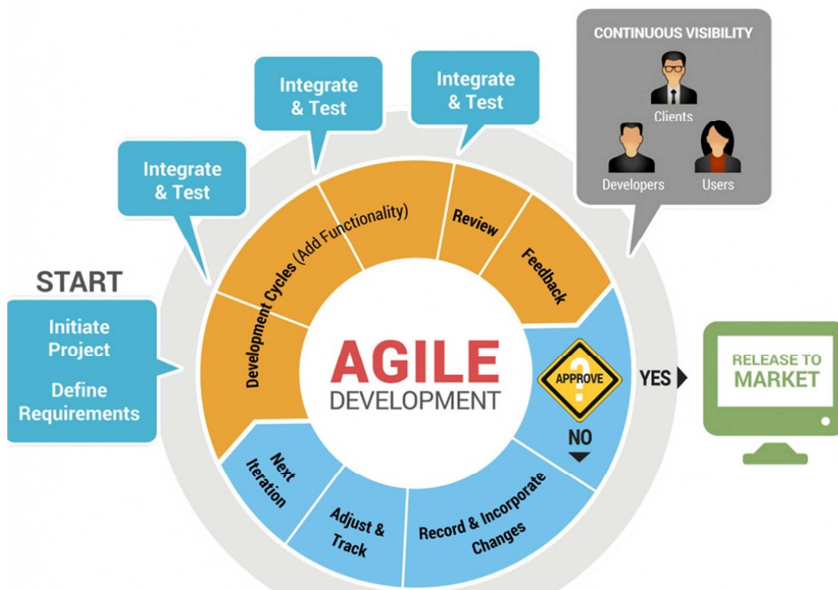


Figura 4. Exemple de desenvolupament de metodologia Agile (Font: <http://comunidad.iebschool.com/>)

Malgrat això que hem indicat, la metodologia que hem aplicat a la realització i desenvolupament del nostre projecte no pot considerar-se que segueixi exactament cap paràmetre metodològic únic o concret dels ja existents (com els indicats anteriorment, *Waterfall* o Desenvolupament en Cascada, *Agile*, Àgil (amb les

diverses opcions que s'hi apropen, com *Scrum* o *Iterative*) o *Spiral* o Model Espiral), sinó que hem anat emprant una metodologia pròpia basant-nos, això sí, en la família de les metodologies àgils citades anteriors, anomenada *metodologia SkylineEngine*, la qual consideràvem la més adient i adaptable als objectius del nostre projecte i que, al cap i a la fi, és la que ens va recomanar i aconsellar de seguir el tutor del projecte, Dr. Gustavo Patow, i que, ahora, és el mateix que ja havia seguit Aura Ruiz Duñach al seu projecte de l'any 2015. D'aquesta manera, s'ha anat definint un tipus de metodologia que s'hagi adaptat adequadament al projecte, tal i com podem observar i constatar al següent diagrama de flux (vegeu Figura 5):

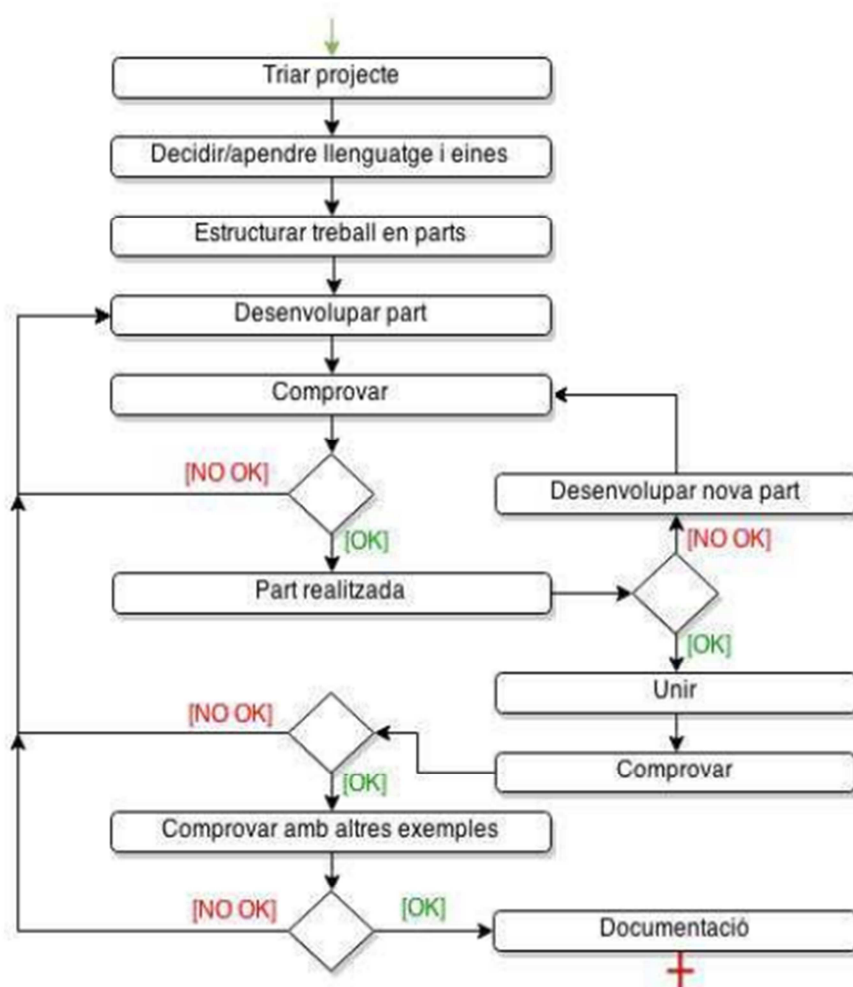


Figura 5. Diagrama de flux per a la metodologia SkylineEngine, metodologia que hem seguit per al desenvolupament del nostre projecte i que es basa en la família de metodologies àgils (Font, Ruiz Duñach, 2015)

Com es pot observar a partir del diagrama de flux de la Figura 4, els passos que se segueixen per a la realització d'un treball amb l'esmentada metodologia són els que anotem tot seguit::

1. Es tria el treball a desenvolupar.
2. Es decideix el llenguatge de programació i les eines que hi caldrà utilitzar.
3. Cal aprendre el llenguatge de programació i el funcionament de les eines escollides.
4. S'estructura el treball en parts segons les funcions que s'han de realitzar.
5. Es desenvolupa la part corresponent seguint l'ordre de l'estructura del treball.
6. Es realitzen les comprovacions per tal de confirmar que el funcionament és correcte en finalitzar cada part.
 - 6.1. Si el resultat no és l'esperat, es torna al cinquè per tal de realitzar els canvis oportuns de la darrera part desenvolupada, o parts anteriors, si fos necessari
 - 6.2. Si el resultat és l'esperat, s'agafa la part següent i es torna al punt 5 per a desenvolupar-la. En cas que s'hagin finalitzat les parts amb les comprovacions respectives, s'inicia el punt 7.
7. S'uneixen totes les parts desenvolupades i es comprova que el funcionament sigui el correcte
 - 7.1. Si el resultat no és l'esperat, es torna al punt 5 per realitzar els canvis oportuns de l'última part desenvolupada, o anteriors, si calgués.
 - 7.2. Si el resultat és l'esperat, s'inicia el punt 8
8. Es generen diversos models d'exemple per comprovar que el funcionament global sigui l'adequat.

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

- 8.1. Si el resultat no és l'esperat, es torna al punt 5 per realitzar els canvis oportuns de l'última part desenvolupada, o anteriors, si s'escaigüés.
- 8.2. Si el resultat és l'esperat, s'inicia el punt 9
9. Es redacta i es poleix la documentació del projecte

Com podem comprovar, es tracta d'una metodologia que consisteix a dividir el projecte en mòduls i anar establint una determinada quantitat de temps per al seu desenvolupament, per a la verificació i per a la posterior correcció del projecte quan és necessari. Durant el transcurs del projecte, s'estableixen unes pautes de seguiment, normalment setmanals, supervisades mitjançant reunions amb el tutor del projecte malgrat que, com és comprensible, no es tractarà del mateix seguiment a totes les etapes del projecte; d'aquesta manera, a les etapes inicials del projecte, així com a les de documentació, les reunions no seran tan habituals i freqüents com a les altres etapes.

L'objectiu, a l'hora de donar per finalitzat un mòdul, és el de no haver-lo de retocar amb posterioritat, que el puguem considerar com a definitiu, de manera que garantim, majoritàriament, que els errors que puguin anar sorgint formin part del darrer dels mòduls en desenvolupament i/o comprovacions, i no pas de parts anteriors, o, com a mínim, que afectin el mínim possible el projecte sense comprometre'n el progrés, atès que, d'aquesta manera, anem cloent parts a mesura que anem avançant en mòduls, la qual cosa el fa un mètode força fiable i que impedeix que a les acaballes del treball s'hagi de tornar molt enrere si algun aspecte no es pot acabar de verificar.

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

Per al procés de disseny, ens servirem del llenguatge que és l'estàndard *de facto* en el camp de l'enginyeria del software, l'UML (*Unified Modeling Language*). L'UML s'utilitza per definir un sistema, per detallar-ne els seus elements, per a la documentació i per a la construcció. Per poder arribar a assolir aquest concepte, l'UML disposa de molts tipus de diagrames, els quals mostren un nombre molt significatiu d'aspectes dels elements que es van representant.

Els motius, les raons, que ens han conduït a escollir aquesta metodologia i no alguna altra han estat conseqüència de les característiques del propi projecte. En primer lloc, el projecte es tracta d'un projecte executat i desenvolupat per una única persona, de manera que no és necessari detallar amb excés cada part per coordinar-la entre diversos integrants de l'equip si fos el cas d'un treball grupal o col·lectiu. Per tant, com indica Ruiz Duñach (2015, p.13): "aquesta metodologia ens permet prioritzar el procés de desenvolupament en lloc d'invertir molt temps en el disseny previ". Tanmateix, com és lògic, això no exclou que s'hagi establert un disseny inicial amb anterioritat al fet de començar a desenvolupar el projecte.

4.- Planificació

Aquesta quarta etapa és aquella en la qual es defineixen les dates aproximades de realització i evolució del projecte, és a dir, el pla que s'ha seguit per a realitzar les tasques que es pretenien desenvolupar. Igualment, s'hi inclou el temps estimat per a la realització de cadascuna tasques que s'hagin anat realitzant.

El plantejament inicial d'aquest projecte es remunta cap al mes d'octubre de l'any 2016 amb la intenció de ser presentat i defensat el juny del 2017. En general, cal indicar que sempre hem anat realitzant la investigació i l'execució del projecte de forma paral·lela. El desenvolupament o temporització de la planificació ha estat la següent (vegeu Figura 6), partint de la base que durant tots els mesos que ha durat la realització i desenrotllament del projecte hem anat mantenint una entrevista setmanal d'entre trenta minuts i una hora amb el tutor del treball, el Dr. Gustavo Patow:

- Octubre de 2016 a desembre de 2016: creació de les estructures de dades adequades per tal de poder treballar amb els algorismes genètics.
- Desembre de 2016 a març de 2017: recerca i anàlisi de llibreries genètiques.
- Abril de 2017 a maig de 2017: programació de l'algorisme genètic.
- Maig de 2017: testeig, verificació i refinament del projecte desenvolupat.

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

- Maig de 2017 a juny de 2017: redacció de la memòria del projecte (amb el suport del Dr. Jordi Regincós)

Traslladat a un Diagrama de Gantt podem observar de forma més visual la planificació i l'evolució en la realització del projecte.

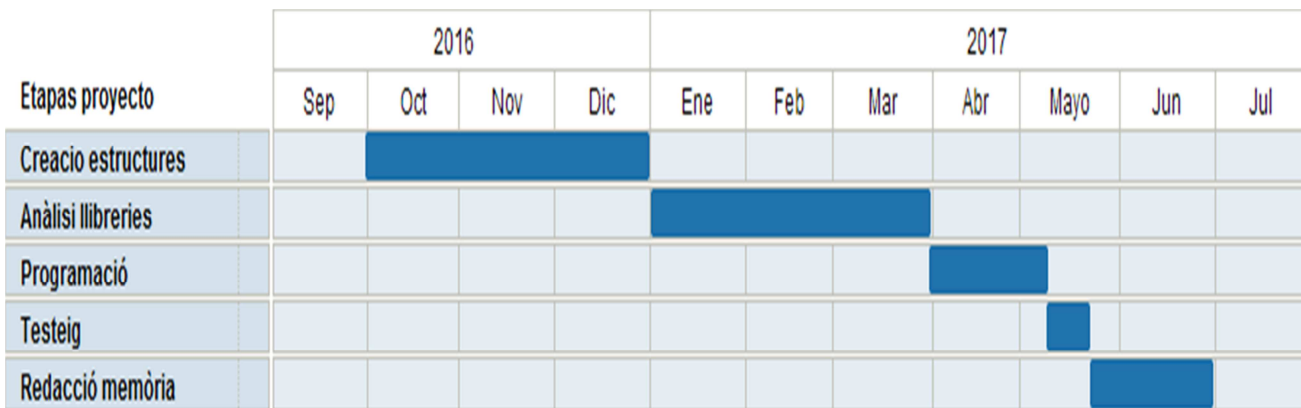


Figura 6. Diagrama de Gantt en el qual hi podem observar les diverses tasques realitzades per al desenvolupament del projecte amb la seva seqüenciació temporal (Font elaboració pròpia)

5.- Marc de treball i conceptes previs

En aquest cinquè apartat, hi anotarem la introducció a tots els conceptes relacionats amb la programació utilitzada per al desenrotllament del treball. Cal indicar que tots aquestes aspectes es consideren necessaris per aconseguir d'obtenir un adequat seguiment del projecte.

Tot seguit, definirem i explicarem els conceptes bàsics i fonamentals que considerem que calen per facilitar una millor comprensió del projecte que hem desenvolupat, els quals són, únicament, dos:

- Simulació de maquinària
- Algorismes genètics

5.1.- Simulació de maquinària

En el món actual, els ordinadors i les solucions i possibilitats que aquests ofereixen es troben molt presents en l'àmbit de l'enginyeria, per això han esdevingut tan importants els sistemes de simulació per a poder disposar de coneixement sobre quin serà o podria ser el resultat final d'un prototip que es troba en una fase de proves i, conseqüentment, si es considera possible i assumible dur a terme la seva construcció. Així mateix, d'aquesta manera, mitjançant la simulació (vegeu la Figura 7), també es redueix el número de prototips construïts relacionats amb un mateix producte que puguin resultar fallits o erronis per diverses raons o condicions de

funcionament ja que gràcies a ells podem conèixer els canvis que caldria aplicar al mecanisme en cas de construcció com a conseqüència de les seves condicions de funcionament.

Igualment, la simulació també ens facilita el fet de poder conèixer com s'aplicaran les lleis de la física i la termodinàmica en el nostre mecanisme. D'aquesta manera, els professionals i les empreses podran arribar a reduir el cost de les primeres etapes o nivells de disseny i proves inicials o estudis previs dels mecanismes.

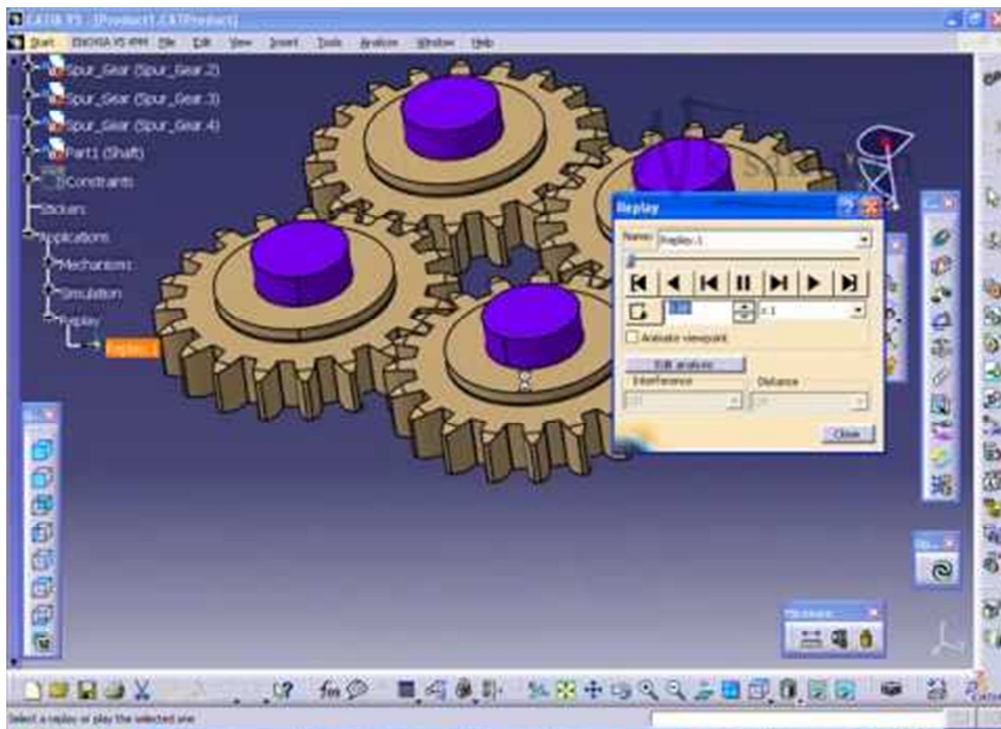


Figura 7. Exemple de simulació d'engranatges

5.2.- Algorismes genètics

Quan parlem d'un algorisme genètic, de l'anglès *genetic algorithm*, ens referim a un tipus d'algorisme que és utilitzat en la informàtica o en sistemes computacionals amb la finalitat de procurar de trobar solucions aproximades a problemes

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

d'optimització i de recerca. Es tracta d'un tipus d'algorismes evolutius que s'inspiren en l'evolució en biologia, és a dir, en l'evolució de les espècies presentada per Charles Darwin i les posteriors investigacions d'estudiosos com Gregor Mendel, així, per tant, en aspectes com poden ser l'herència, la mutació, la selecció i l'encreuament, és a dir, que es tracta d'una manera de cercar solucions seguint el model de la naturalesa (exemple d'execució a la Figura 8), de tal manera que s'implementen com una simulació informàtica en què unes representacions abstractes de solucions possibles a una problemàtica d'optimització evolucionen cap a les solucions més adequades. És com aplicar la modelació en informàtica de l'evolució genètica, tal i com es recull en uns apunts del Departament d'Electrònica, Informàtica i Automàtica de l'assignatura de Disseny de Sistemes de Supervisió de la Universitat de Girona que, a més a més, afegeixen:

"Els gens són símbols que tenen codificades les solucions possibles a un problema d'optimització, definint-se l'adaptació com una funció que es vol optimitzar, i els individus són possibles solucions que de forma evolutiva i agrupats en poblacions intenten adaptar-se al medi definit per l'esmentada funció. Els operadors de selecció, reproducció, mutació, i altres es defineixen a similitud biològica seleccionant individus de forma iterativa per a cada població, creuant codis genètics entre individus i mutant el codi dels individus seguint variables aleatòries."

(Universitat de Girona, http://rogiteam.udg.es/docencia/dss_2000/genetics/practica5_ag.pdf.

Consulta el dia 05 de juny de 2017)

Per a obtenir una més àmplia informació i, per tant, coneixement, a l'entorn dels algorismes genètics, es pot consultar l'obra de Melanie Mitchell, de l'any 1996,

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

titulada: *An introduction to genetic algorithms* que s'inclou a la llista de referències bibliogràfiques.

D'aquesta manera, per acabar, podem indicar que un algorisme genètic, bàsicament, està format per dues variables:

- Com es representen les solucions dins de l'algorisme
- La funció de *fitness* per tal de poder comparar dues solucions

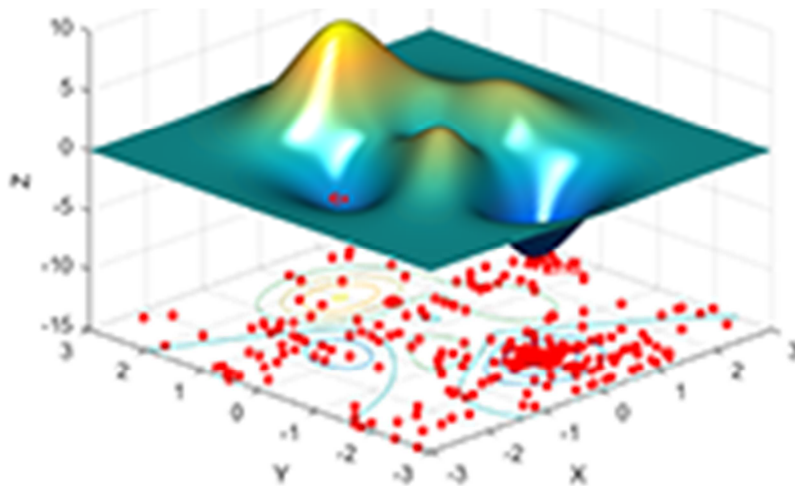


Figura 8. Exemple d'execució d'un algorisme genètic

6.- Requisites del sistema

Al sisè punt, hi descrivim els requisits que són necessaris del sistema (tant els funcionals com els no funcionals) per a arribar a conèixer amb exactitud tot allò que cal fer per a què el projecte acabi complint els objectius fixats inicialment.

O sigui, que en aquest apartat detallarem tots els requisits i condicionants que seran necessaris per al sistema, és a dir, tot allò que cal a fi i efecte d'aconseguir fer funcionar adequadament el projecte, des de tots els àmbits, tant des d'un punt de vista del programari com del maquinari i, per tant, que, al cap i a la fi, es puguin aconseguir de fer realitat els objectius inicials.

6.1 Plantejament del problema

La problemàtica que ens plantegem d'abordar i, per tant, que volem solucionar, fa referència al concepte d'optimitzar mecanismes senzills amb un objectiu concret d'automatització del procés de disseny d'assemblatge mecànic per fer-lo més ràpid, àgil i econòmic..

Un dels primers problemes amb el qual ens trobem consisteix en el fet de poder llegir les peces inicials des d'un fitxer ja escrit. Així mateix, també ens fa falta poder programar l'algorisme d'optimització i, tot seguit, poder-lo simular a l'Autodesk Inventor per a poder-ne observar el resultat.

6.2 Plantejament de la solució

La solució que es planteja és llegir des d'un XML totes les peces que l'usuari desitgi carregar usant el mateix llenguatge creat per l'Aura Ruiz Duñach en el seu treball de l'any 2015. El propi programa facilitarà totes les eines necessàries per a poder optimitzar el sistema d'engranatges donat. D'aquesta manera, l'usuari podrà optimitzar el mecanisme usant els paràmetres configurats en el fitxer.

6.3 Requisits funcionals

Quan al·ludim als requisits funcionals, ens referim als requisits i funcionalitats a desenvolupar i que són necessaris que realitzi l'aplicació:

- Càrrega de fitxers personalitzat
- Càrrega de fitxers en format XML
 - Generació d'assemblatges
 - Generació de peces
 - Connexió de peces i/o d'assemblatges
 - Assignació del motor del mecanisme
 - Estructura jeràrquica d'assemblatges
 - Càlcul de relació de transmissió
 - Optimització d'engranatges

6.4 Requisits no funcionals

En aquest apartat mostrarem les característiques que ha de contenir l'aplicació. Aquests requisits fan referència al quant i al què hem d'anar en compte a l'hora de dissenyar el projecte.

La plataforma en la qual s'executarà el codi en el nou llenguatge serà Autodesk Inventor 2017. Arran del fet que els models dels quals farà ús la nostra aplicació per generar noves peces no s'accepten en versions més antigues, un dels requisits de l'aplicació serà el d'optimitzar la interacció únicament amb la versió 2017 de l'Autodesk Inventor. A més a més, cal indicar que els requisits de l'ordinador del qual farem ús seran els més bàsics que exigeix aquesta plataforma, que són els que segueixen:

- Sistema Operatiu Windows 7 (SPI)
- 8 GB de memòria RAM
- Tipus de CPU Intel o AMD

7.- Estudi i decisions

Tot seguit, desenvoluparem la descripció del maquinari i programari utilitzat i igualment, i per acabar, les seves dependències, per tant, en aquest apartat de la memòria descriurem el maquinari i el programari usats per a portar a terme el projecte així com les justificacions, les raons per les quals hem elegit aquests elements i no pas cap altre.

7.1.- Autodesk Inventor

L'Autodesk inventor es tracta d'un paquet de modelatge paramètric d'objectes sòlids, és a dir, que ofereix a l'usuari la facultat de dissenyar peces i fer-ne assemblatges amb la viabilitat de crear-ne altres de noves, és a dir, es tracta d'un programari que permet renderitzar objectes en 3D a través de paràmetres en els quals l'usuari gaudeix de la possibilitat d'interactuar amb l'editor, des d'objectes senzills, simples, fins a peces amb un alt grau de complexitat. Aquest programa ha estat desenvolupat per l'empresa Autodesk fent ús de tecnologia de Microsoft amb la finalitat de fer més ràpid, àgil i senzill el procés de desenvolupament d'un producte a l'hora de realitzar proves mecàniques sobre la base d'una proposta virtual, és a dir, que facilita de forma molt eficient el pas d'un disseny inicial en dues dimensions a volums i objectes. Altres programaris de disseny assistit per a ordinador semblants que existeixen en el mercat i que ens haurien pogut resultar útils, però que, tanmateix, hem descartat per al nostre projecte, són el SolidWorks, Pro/ENGINEER, CATIA o el Solid Edge.

Aquest programari es presenta amb dues versions:

- L'Autodesk Inventor Series (AIS): En aquesta versió, s'hi inclouen les eines bàsiques per a poder modelar les peces i el seu assemblatge o acoblament, entre d'altres funcions fonamentals.
- L'Autodesk Inventor Professional (AIP): Aquest model facilita a l'usuari l'oportunitat d'utilitzar l'AutoCAD a més a més de poder fer ús de totes les possibilitats que ofereix l'Autodesk Inventor Series..

Motiu de selecció

Hem decidit d'utilitzar aquest programari atès que, a banda que ens permet d'usar-lo de manera gratuïta perquè hem pogut obtenir una llicència d'estudiant universitari, també cal destacar que aquest programari és un dels més potents que existeix en el seu sector, a més a més que té una gran tradició i ha anat evolucionant amb les diverses versions que s'han anat desenvolupant, ja que la primera versió data de 1999, raó per la qual està perfectament estudiat i desenvolupat de manera que difícilment pot provocar sorpreses inesperades durant el seu ús. A més a més, ofereix la possibilitat de programar-hi amb molts i diversos llenguatges de programació. A part d'això, alhora, cal indicar que es tracta del mateix programari que havia elegit l'Aura Ruiz Duñach en el seu projecte de l'any 2015; no obstant això, nosaltres ens vam decidir d'utilitzar la versió 2017 ja que aquesta és la darrera de la qual disposàvem, compatible amb l'ordinador amb el qual havíem de realitzar el nostre projecte. Malgrat això, cal afegir que ja s'ha començat a desenvolupar i comercialitzar una nova versió, l'anomenada versió 2018, tal i com hem pogut

constatar a [.https://www.autodesk.es/campaigns/inventor](https://www.autodesk.es/campaigns/inventor) (consulta d'1 de juny de 2017)



7.2.- C#

C# es tracta d'un llenguatge de programació que va ser desenvolupat per Microsoft com a part integrant i estandarditzada de la seva plataforma de .NET, a partir dels anys 1999-2000, dissenyat per la *Common Language Infrastructure*, és a dir, que es tracta d'una especificació estàndard que descriu un entorn virtual per a l'execució d'aplicacions en múltiples i diversos llenguatges. Va ser normalitzat per ECMA (*European Computer Manufacturers Association*) i recollit com a un estàndard per ISO (*International Organization for Standardization*).

La seva sintaxi deriva del llenguatge de programació C/C++ (vegeu la Figura 9) i fa ús del model d'objectes que s'utilitza a la plataforma .NET, molt semblant al llenguatge de programació de Java, encara que inclou moltes millores respecte d'altres llenguatges ja que s'hi han aplicat aspectes òptims que en provenen, com

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

ara les influències que ha rebut de Delphi, Modula-3 o Eiffel, entre d'altres. Es tracta, per tant, d'una sintaxi força senzilla i assimilable per a l'usuari si aquest disposa d'uns certs coneixements de programació informàtica. Tanmateix, malgrat que forma part de la plataforma .NET, C# és un llenguatge de programació independent que ha estat dissenyat amb la finalitat última de generar programes sobre aquesta pròpia plataforma .NET.

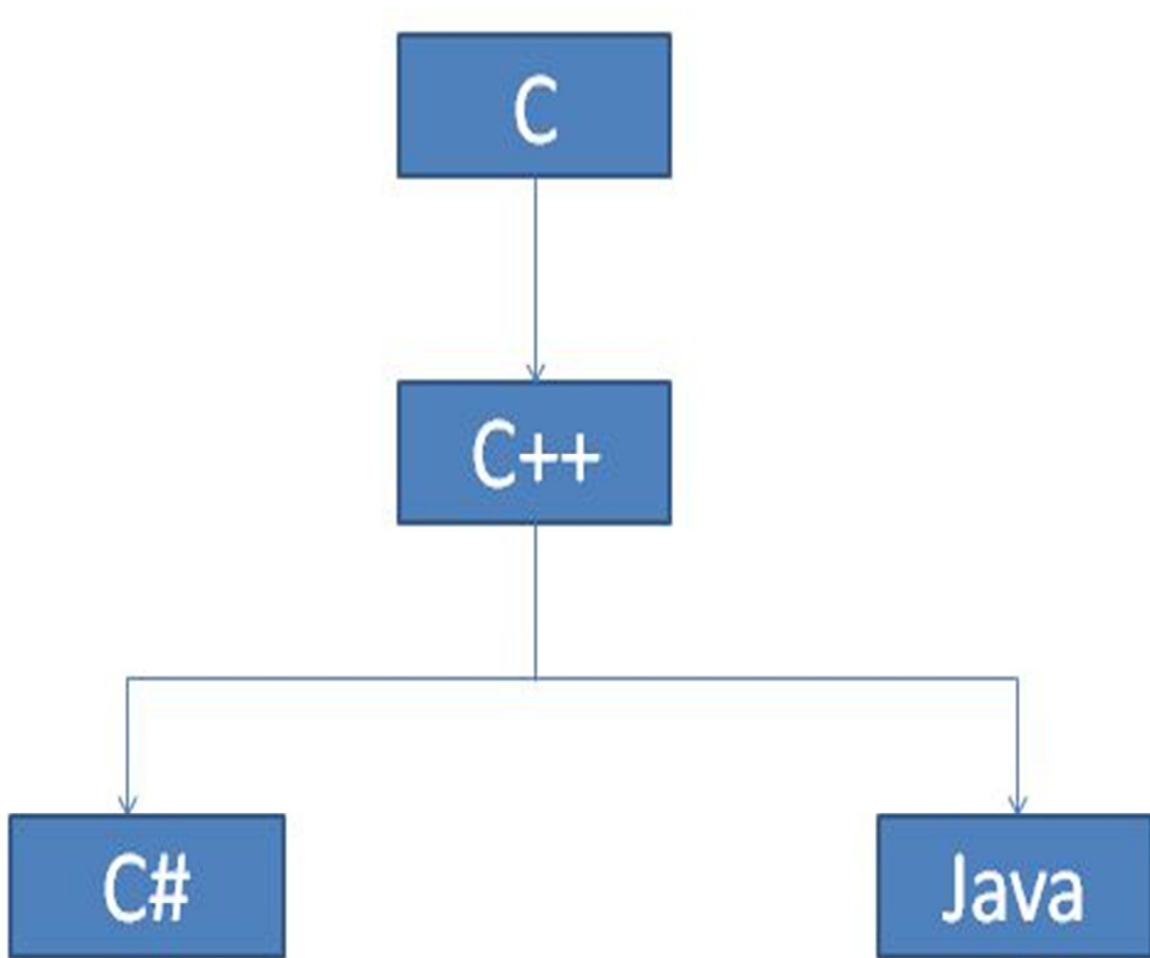


Figura 9. Bases del C#

C# es considera que és un llenguatge de programació eficaç i funcional orientat a objectes i que té molts i diversos tipus de variables i, alhora, també disposa de tipus de valor i tipus de referència.

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

Com que el projecte l'estem fent sobre C# i a l'hora d'optimitzar és important conèixer els rangs i precisió dels diferents tipus de variables, els números enters que serveixen per a representar números són::

Tipo de datos de enteros				
Tipo	Equivalente BCL	Tamaño	Rango	Significado
byte	System.Byte	8-bit (1-byte)	0 a 255	Entero sin signo
sbyte	System.SByte	8-bit (1-byte)	-128 a 127	Entero con signo
short	System.Int16	16-bit (2-byte)	-32.768 a 32.767	Entero corto con signo
ushort	System.UInt16	16-bit (2-byte)	0 a 65.535	Entero corto sin signo
int	System.Int32	32-bit (4-byte)	-2.147.483.648 a 2.147.483.647	Entero medio con signo
uint	System.UInt32	32-bit (4-byte)	0 a 4.294.967.295	Entero medio sin signo
long	System.Int64	64-bit (8-byte)	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Entero largo con signo
ulong	System.UInt64	64-bit (8-byte)	0 a 18.446.744.073.709.551.615	Entero largo sin signo

Tipus de coma flotant, que serveix per a representar números amb decimals:

Tipo de datos de coma flotante				
Tipo	Equivalente BCL	Tamaño	Rango	Significado
float	System.Single	32-bit (4-byte)	$\pm 1.401298E-45$ a $\pm 3.402823E+38$	Coma flotante corto
double	System.Double	64-bit (8-byte)	$\pm 4.94065645841246E-324$ a $\pm 1.79769313486232E+308$	Coma flotante largo
decimal	System.Decimal	128-bit (16-byte)	$-7.9228162514264337593543950335$ a $+7.9228162514264337593543950335$	Coma flotante monetario

Tipus caràcter, per a guardar caràcters, com per exemple lletres:

Tipo de datos de caracteres				
Tipo	Equivalente BCL	Tamaño	Rango	Significado
char	System.Char	16-bit (2-byte)	'\u0000' a '\uFFFF'	Carácter unicode

Tipus lògic:

Tipo de datos lógicos				
Tipo	Equivalente BCL	Tamaño	Rango	Significado
bool	System.Boolean	8-bit (1-byte)	true o false	Verdadero o falso

Motius de selecció

Malgrat que en alguns fòrums es dóna molt més suport a Visual Basic que no pas a C#, atès que a la Universitat de Girona s'ensenya el C++ i el C# hi està basat, ens ha resultat molt més pràctic, àgil i senzill de programar amb C#. A més a més, C# permet desenvolupar aplicacions amb una certa facilitat, la qual cosa l'ha fet molt útil per al nostre treball



7.3.- Microsoft Visual Studio 2017

Visual Studio és un conjunt d'eines i d'altres tecnologies, un entorn integrat de desenvolupament, creat amb la finalitat de poder desenvolupar programari de manera eficient. Aquest entorn de desenvolupament integrat de Microsoft també serveix, alhora, per a poder desenvolupar amb altres llenguatges de programació que pot integrar. És una eina que, alhora, permet de desenvolupar altres eines, aplicacions i pàgines web.

Motius de selecció

Hem decidit de fer ús d'aquest entorn integrat de desenvolupament pel fet que és un IDE molt important i significatiu quant a comunitat d'internet i de funcionalitat i, a més, amb molta experiència de programació per part de Microsoft, per tant, això implica que té múltiples funcionalitats a disposició del desenvolupador i molta comunitat al darrere que pot col·laborar amb tu a l'hora de cercar solucions davant d'algun inconvenient o problema.



7.4.- Llibreria genètica AForge

Aquesta llibreria és un *framework*, un entorn o marc de treball de C# dissenyat per a desenvolupadors i investigadors en el camp de la intel·ligència artificial i visió per computador.

AForge té un set de llibreries internes, la qual cosa implica que disposa de moltes característiques i diverses i útils possibilitats:

:

- AForge.Imaging: Llibreria per a processar imatges i posar filtres.
- AForge.Vision: Visió per computador.
- AForge.Video: Conjunt de llibreries o biblioteques per a processament de vídeos.
- AForge.Neuro: Llibreria per a crear xarxes neurals.
- AForge.Genetic: Llibreria per a fer programació genètica
- AForge.Fuzzy: Llibreria per a càlculs difusos.
- AForge.Robotics: Biblioteca per a poder donar suport a *kits* de robòtica
- AForge.MachineLearning: Llibreria per a poder fer aprenentatge automàtic, en anglès *machine learning*.

Motius de selecció

Malgrat que no disposa d'una gran comunitat al darrera i que la documentació no és excessivament àmplia ni rica, AForge.Genetic és una llibreria de codi obert i es pot llegir sense massa problemes el codi font a part que està prou ben comentat.



7.4.- Sistema operatiu

El sistema operatiu que hem utilitzat per a la realització del nostre projecte ha estat la darrera versió del sistema operatiu Windows, és a dir, el Windows 10.

Motius de selecció

Hem utilitzat aquest sistema operatiu perquè és el més compatible amb l'Autodesk Inventor i amb la resta de programari usat.



8.- Anàlisi i disseny

En aquest vuitè apartat, hi desenvolupem l'explicació de l'anàlisi que s'encarrega de totes les necessitats detallades del sistema, a part, d'un disseny del sistema amb una solució. De la mateixa manera, en aquesta fase també detallarem el model de les dades i el model de processos i, així mateix, hi mostrarem les interfícies d'usuaris, els models físics i els models d'objectes.

D'aquesta manera, en aquesta part, podrem observar de manera més precisa, concreta i detallada les necessitats del sistema per a la realització i desenvolupament del nostre projecte i, alhora, les solucions que hem anat aplicant per a cadascuna d'aquestes necessitats.

L'objectiu d'aquest projecte és, al cap i a la fi, poder desenvolupar un programari que permeti a l'usuari, de forma senzilla i pràctica, parametritzar l'algorisme amb la finalitat de poder optimitzar un sistema d'engranatges i, tot seguit, poder-lo observar i estudiar simulat dins de l'Autodesk Inventor i, d'aquesta manera, poder copsar quin seria a la realitat el seu funcionament.

8.1.- Fitxes i diagrames de cas d'ús

Un cas d'ús és una seqüència d'interaccions que es desenvoluparan entre un sistema i els seus actors en resposta a un esdeveniment que inicia un actor principal sobre el mateix sistema amb la finalitat de poder conèixer les necessitats que seran precises per a un projecte indicant com hauria d'interactuar l'usuari amb el sistema

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

amb la finalitat d'arribar a assolir la solució que es consideraria, finalment, el resultat més òptim i desitjable.

Els diagrames de casos d'ús tenen com a utilitat especificar la comunicació i el comportament d'un sistema mitjançant la seva interacció amb els usuaris i/o altres sistemes, de la manera com es pot comprovar a la Figura 10.

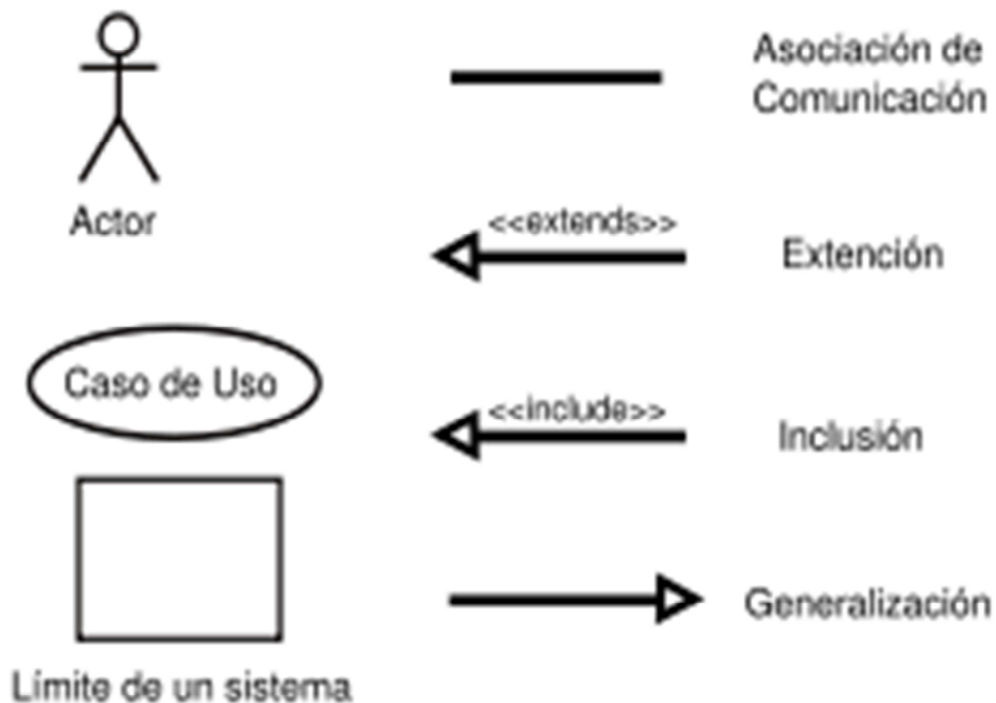


Figura 10 Notació de cas d'ús (Font: Viquipedia.org).

De totes maneres, cal indicar que en el projecte que hem desenvolupat, el cas d'ús és molt simple tal i com es pot comprovar a la Figura 11:

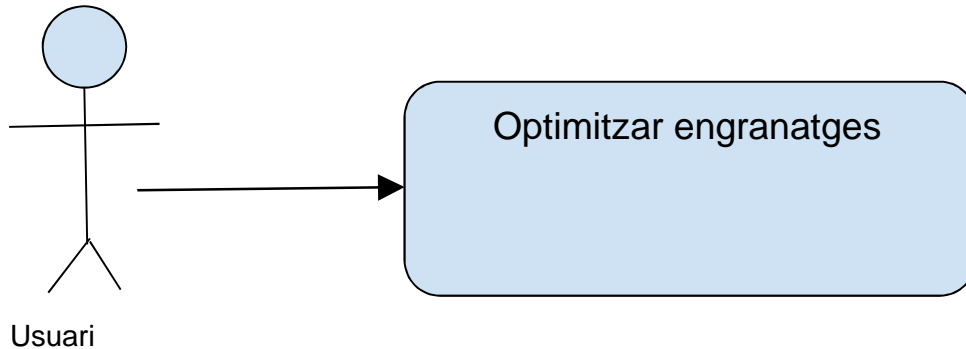


Figura 11 Cas d'ús del nostre projecte (Font elaboració pròpia)

El cas d'ús consta de les següents parts o elements:

- **Actor:** un actor és una entitat que fa ús del nostre programari, els actors no formen part del sistema, no obstant això, com a tals, actuen sobre d'ell.
- **Comunicació:** és allò que transmet un cas d'ús a un actor.
- **Escenari:** és un conjunt de casos d'usos i interaccions entre un conjunt d'actors i el sistema.

Per a especificar allò que realitza cada acció dins del sistema, s'hi utilitzen el que s'anomena fitxa de cas d'ús, la qual cosa consisteix, fonamentalment, a escriure'l; cal indicar que, en el nostre cas, només existeix una única fitxa de cas d'ús, que anotem tot seguit:

Cas d'ús:	Optimitzar engranatges
Descripció	L'actor vol optimitzar un sistema d'engranatges
Actor	Usuari
Precondició	Disposa d'un arxiu de configuració i un arxiu XML amb les peces a tractar
Flux del programa	<ol style="list-style-type: none"> 1. Clicar el botó d'optimització 2. Obrir el fitxer de configuració 3. Veure el resultat a la finestra emergent
Postcondició	Les peces ja optimitzades apareixen en el programa d'Autodesk Inventor i es mostra un TXT amb el sistema d'engranatges

8.2.- Diagrama de classes

Un diagrama de classes és allò que defineix les relacions internes que existeixen, és a dir, “un tipus de diagrama d'estructura estàtica que descriu l'estructura d'un sistema mostrant-ne els atributs, les classes i les relacions entre elles” (Ruiz Duñach, 2015, p.35)

8.2.1.- Mòduls funcionals

En el cas del nostre programa, fonamentalment, l'aplicació es divideix en tres mòduls funcionals o Llibreries de C#.(vegeu la Figura 12), els quals estan dividits en tres colors segons on es troba ubicat el codi

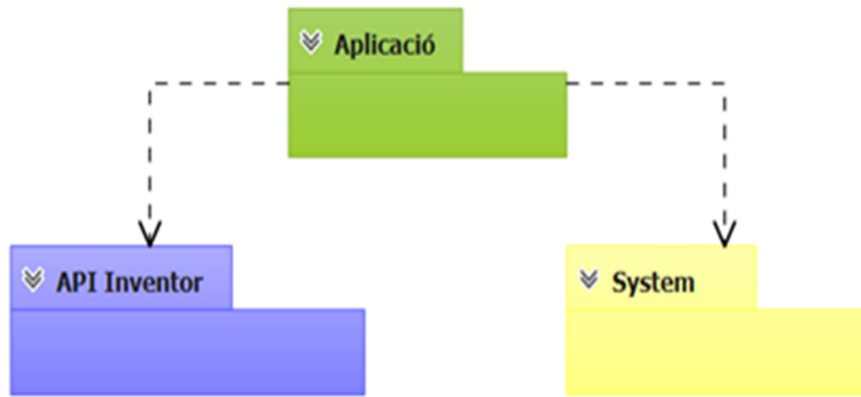


Figura 12 Mòduls del programari

Cada mòdul funcional es troba dividit per classes, que, com hem anotat més amunt, es diferencien per colors que ens permeten de tenir coneixement del mòdul al qual cadascun pertany. El diagrama, en la seva totalitat és francament complex (vegeu Figura 13), raó per la qual, posteriorment l'hem dividit en tres blocs, cadascun dels quals correspon a cadascuna de les classes que el conformen (vegeu Diagrames 13.1, 13.2 i 13.3).

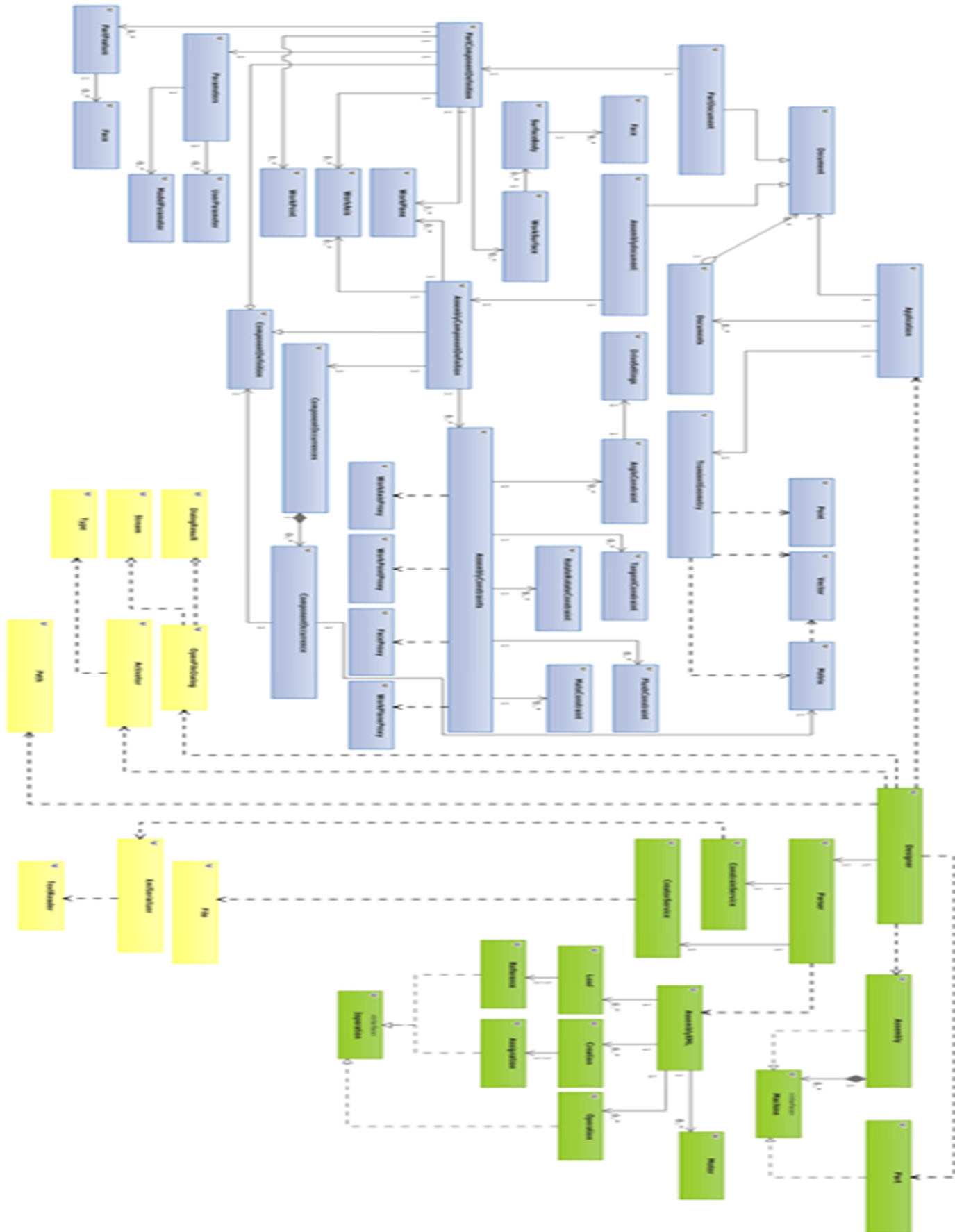


Figura 13 Diagrama de classes (Font Ruiz Duñach 2015)

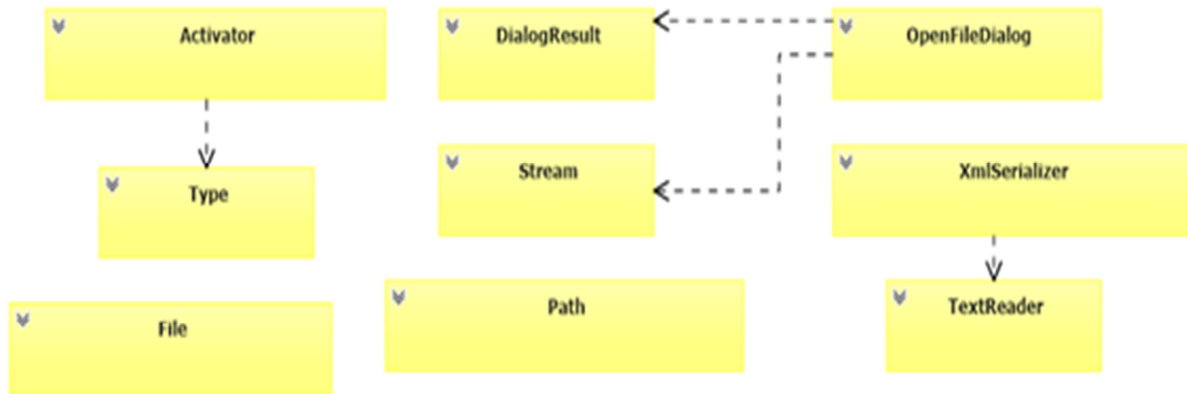


Figura 13.1 Diagrama de classes, mòdul de System (Font Ruiz Duñach 2015)

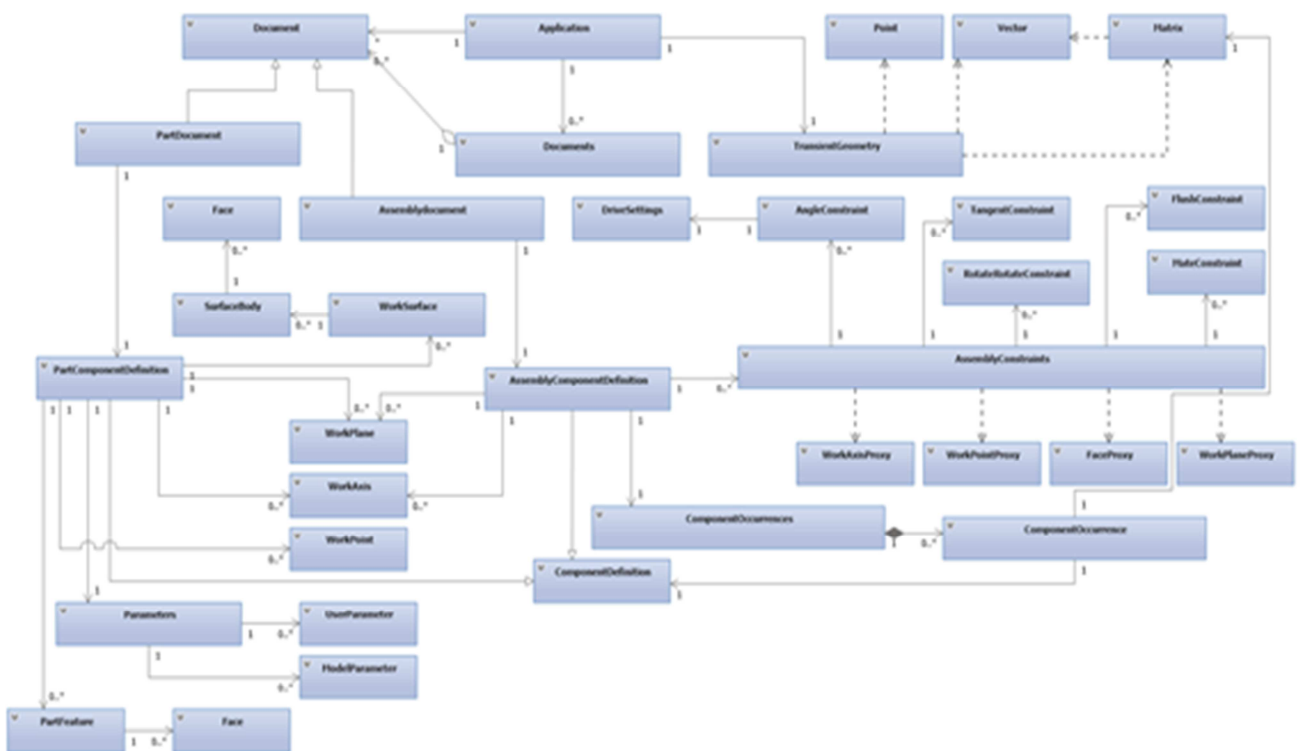


Figura 13.2 Diagrama de classes, mòdul d'API Inventor (Font Ruiz Duñach 2015)

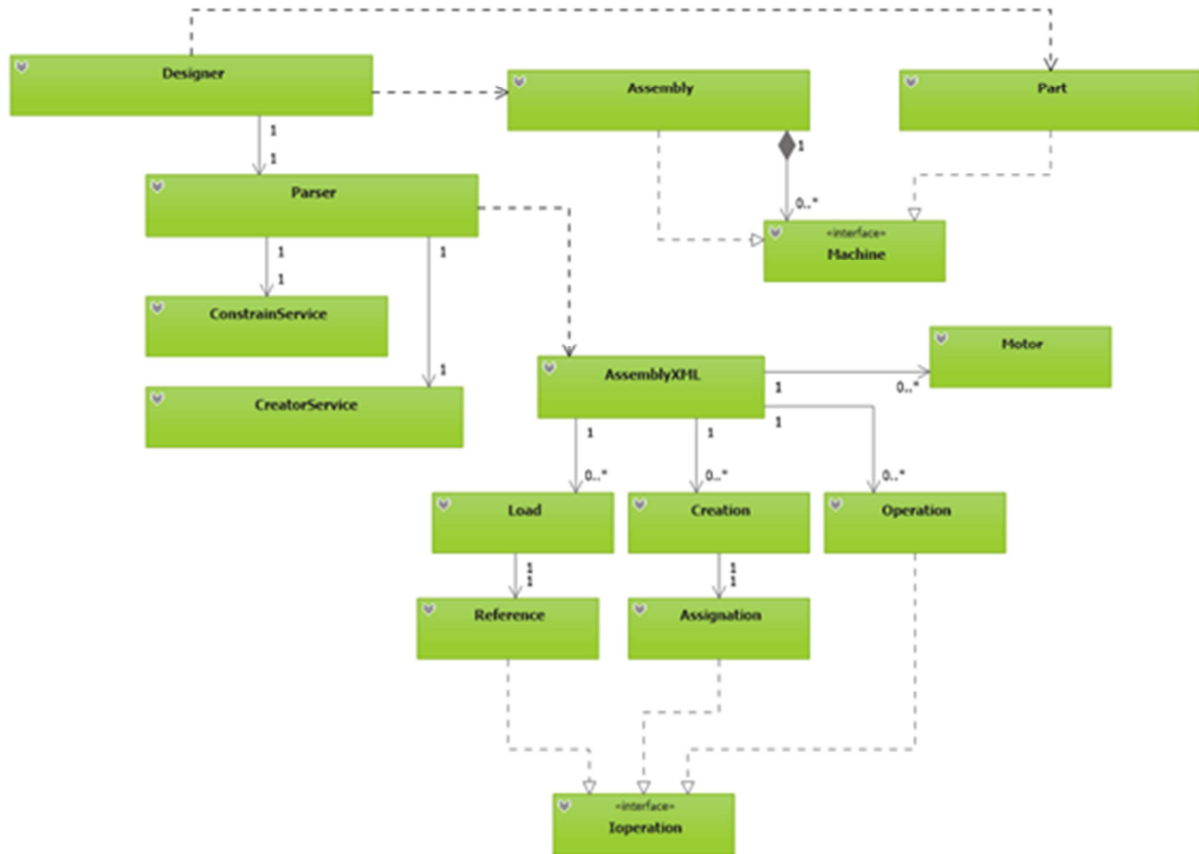


Figura 13.3 Diagrama de classes, mòdul d'aplicació (Font Ruiz Duñach 2015)

Partint de la presentació dels tres mòduls del programari (vegeu Figura 12) i del diagrama de classes (vegeu Figura 13) , els mòduls que cal concretar i explicar són els que segueixen:

- **Mòdul de System:** mòdul que forma part del llenguatge C# i que defineix les classes fonamentals i de base que expliquen l'ús de dels valors i referències. Aquest espai de noms defineix totes les funcions que ens ofereix el sistema per tal d'interactuar amb el sistema operatiu. En el diagrama de classes és el que apareix de color groc (vegeu Figura 13.1)

- **Mòdul d'API Inventor:** aquest mòdul és el que ens ve donat per la pròpia plataforma d'Autodesk Inventor amb la finalitat de poder enviar i rebre paràmetres d'aquest programa, de tal manera que puguem canviar, crear, etc les parts de la plataforma en la qual farem les simulacions. És força ampli i complex com es pot comprovar en el diagrama de classes en el qual apareix de color blauós (vegeu Figura 13.2)
- **Mòdul d'Aplicació:** aquest és el mòdul principal del treball ja que en aquest mòdul és en el qual podem localitzar-hi tot allò que ha estat programat i que, conseqüentment, treballa sobre els altres mòduls, per tant, és el que ajunta i fa ús dels mòduls de l'API Autodesk Inventor i de System per assolir les funcionalitats projectades. És el que en el diagrama de classes apareix de color verd (vegeu Figura 13.3)

9.- Implementació i proves

En aquest novè punt, hi detallem els problemes amb els quals ens hem trobat durant la realització i el progressiu desplegament del projecte així com les decisions preses per a solucionar-los.

D'aquesta manera, concretarem el procés que segueix el projecte amb la finalitat de generar un fitxer per a poder optimitzar.

9.1.- Fitxer de configuració

El format del fitxer de configuració és un format *Custom*, és a dir, que no segueix cap dels estàndards actuals de programació ja que es tracta d'un format personalitzat. En realitat, consisteix en un format propi per tal que pugui adaptar-se de la manera més òptima i eficaç possible a les necessitats del nostre projecte. Aquest fitxer segueix el següent format:

- **<nom de la variable>;<valor que se li vol donar>**

D'aquesta manera, podem aconseguir que aquells usuaris o persones que no estiguin massa familiaritzats amb la programació informàtica puguin fer ús, igualment, del nostre programari. A la nostra aplicació podrem fer ús dels següents paràmetres:

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

- **Population:** aquest paràmetre es fa servir amb la finalitat de decidir la població inicial de l'algorisme genètic, la qual, per defecte, és 200.
- **Parts:** fitxer amb format XML amb totes les peces que es vulguin carregar al programa; no té valor per defecte, per la qual cosa caldrà posar-lo per a que el programa funcioni adequadament.
- **Objective:** es tracta del valor objectiu que volem que aconseguixi, per defecte el valor és 0,5.
- **Reescalat:** aquest paràmetre afecta directament la funció de *fitness*, en aquest cas defineix com de superior és el valor màxim de la funció.
- **Tolerància:** aquest paràmetre també afecta directament la funció de *fitness*, tanmateix, en aquest cas, consisteix en el marge de tolerància que admet.
 - La funció de *fitness* és el següent: $\text{reescalat} / ((x-N)^2 + \text{tolerància})$
 - Essent x el valor actual i N el valor objectiu
- **EpochTimes:** aquest paràmetre fa referència a les vegades que s'executa l'algorisme genètic amb la finalitat d'intentar optimitzar; per defecte el valor és 50.
- **Output:** es tracta del lloc on es localitzarà el fitxer de sortida amb la millor solució trobada, per defecte crea un fitxer de sortida amb .TXT al costat de l'executable.

El patró del fitxer seria aquest:

```
population;...  
parts;...  
objective;...  
reescalat;...  
tolerancia;...  
epochTimes;...  
output;...
```

Un fitxer d'exemple del nostre projecte podria ser com el que segueix:

```
population;201  
parts;C:\Assembler\Demos\prova2-2gears.xml  
objective;0,5  
reescalat;5001  
tolerancia;201  
epochTimes;1
```

9.2 Inicialització de l'aplicació

Un cop disposem de l'arxiu de configuració, cal executar el programa per tal d'optimitzar els engranatges desitjats, per a inicialitzar el programa usarem els següents *usings*:

```
using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Runtime.InteropServices;  
using System.IO;  
using System.Xml;  
using System.Xml.Serialization;  
using System.Xml.Schema;  
using Inventor;  
using AForge.Genetic
```

9.3 Botó AG optimizer

Fonamentalment, el nostre projecte s'engloba en un únic botó; aquest botó s'anomena "AG optimizer" (*Algorism optimizer*, vegeu Figura 14):

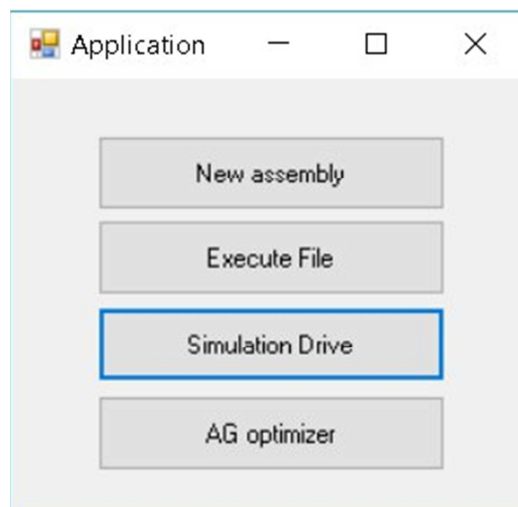


Figura 14 Botons programari

El programa, el dividim en diversos *blocks* funcionals:

9.3.1 Inicialització de les variables

```
Stream myStream = null;
OpenFileDialog openFileDialog1 = new OpenFileDialog();
openFileDialog1.InitialDirectory = PATH_DEMOS;
openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
openFileDialog1.FilterIndex = 2;
openFileDialog1.RestoreDirectory = true;
Tuple<Assembly, List<Part>> retur;
int sPopulation = 200, epoches = 50;
String filePiece = "", fileOut = "aqui.xml";
String path_assem = "";
Double objective = 0, reescalat = 5000, tolerancia = 200;
```

Tal i com es pot observar, en aquest *block* podem comprovar-hi la declaració i assignació dels valors per defecte.

```
int counter = 0;
string line;
System.IO.StreamReader file = new
System.IO.StreamReader(openFileDialog1.FileName);
path_assem =
System.IO.Path.GetDirectoryName(openFileDialog1.FileName) +
"\\\\";
while ((line = file.ReadLine()) != null)
```



```
{
    System.Console.WriteLine(line);
    String[] split = line.Split(';');
    switch (split[0])
    {
        case "population":
            sPopulation = Convert.ToInt32(split[1]);
            break;
        case "parts":
            filePiece = split[1];
            break;
        case "objective":
            objective = Convert.ToDouble(split[1]);
            break;
        case "reescalat":
            reescalat = Convert.ToDouble(split[1]);
            break;
        case "tolerancia":
            tolerancia = Convert.ToDouble(split[1]);
            break;
        case "epochTimes":
            epoches = Convert.ToInt32(split[1]);
            break;
        case "output":
            fileOut = split[1];
            break;
        default:
            Console.WriteLine("Default case");
            break;
    }
    counter++;
}
file.Close();
```

En aquest apartat triem el fitxer de configuració i, posteriorment, el llegim amb la finalitat de carregar els valors desitjats.

```
bool needToStop = false;
int i = 0;
Popury population = new Popury(sPopulation, new
chromosomeTree(allParts), new fitnessPiece(objective,
reescalat, tolerancia), (ISelectionMethod)new
EliteSelection());

while (!needToStop)
{
    // run one epoch of genetic algorithm
    population.RunEpoch();

    i++;

    if (i > epoches)
    {
        needToStop = true;
    }
}
```

I arribats aquí, aquesta part es porta a terme just quan s'acaba l'execució de l'algorisme genètic: bàsicament el que realitza aquesta part és aplegar la millor solució i escriure-la en un fitxer.

```
chromosomeTree show =
(chromosomeTree)(population.BestChromosome);
Part mot = Part.toPart(show.tree.getMotor());
List<Part> lp = show.tree.getAllData();
List<opera> lo = show.tree.iGetOperations();
writeToFile(fileOut, lp, lo, mot);
System.Diagnostics.Process.Start("notepad.exe", fileOut);
_invsApp.Documents.Add(DocumentTypeEnum.kAssemblyDocumentObject);
return parser.parserFile(fileOut, "Arrel", "", true,
path_assem);
```

```
machine = retur.Item1;  
allParts = retur.Item2;  
MessageBox.Show("Solution found, writting to \"" + fileOut +  
"\".");
```

Tal i com es pot observar, la funció “*writteToFile*” és una funció molt bàsica per a escriure el fitxer de sortida:

```
StreamWriter writetext = new StreamWriter(fileName);  
writetext.WriteLine("<?xml version = \"1.0\" encoding =  
\"ISO-8859-1\" ?>");  
writetext.WriteLine("");  
writetext.WriteLine("<assembly  
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"  
xsi:noNamespaceSchemaLocation=\"assembly.xsd\">");  
foreach (Part par in lp)  
{  
    writetext.WriteLine(" <creation>");  
    writetext.WriteLine(" <name>" + par.nom + "</name>");  
    writetext.WriteLine(" <part_info>");  
    writetext.WriteLine(" <part>new gear</part>");  
    writetext.WriteLine(" <param>" + par.diameter + "</param>");  
    writetext.WriteLine(" </part_info>");  
    writetext.WriteLine(" </creation>");  
}  
foreach (opera par in lo)  
{  
    writetext.WriteLine(" <operation>");  
    writetext.WriteLine(" <constraint>join</constraint>");  
    writetext.WriteLine(" <param>" + par.parameters[0].nom +  
"</param>");  
    writetext.WriteLine(" <param>" + par.parameters[1].nom +  
"</param>");  
    writetext.WriteLine(" <param>" + par.graus + "</param>");  
    writetext.WriteLine(" </operation>");  
}  
writetext.WriteLine(" <motor>");  
writetext.WriteLine(" <motor_name>" + mot.nom + "</motor_name>");  
writetext.WriteLine(" </motor>");  
writetext.WriteLine("</assembly>");  
writetext.Close();  
writetext.Dispose();
```

9.4 Problemes durant el treball

Des d'un punt de vista de programari, els problemes amb els quals ens hem trobat s'exposen a l'apartat de conclusions. No obstant això, podríem esmentar aspectes diversos com ara el fet que l'API d'Autodesk Inventor no sigui de codi obert, raó per la qual en alguns casos hem hagut de dirigir-nos al fòrum del programari per solucionar alguns problemes, o bé que l'IDE de Microsoft Visual Studio 2017 en algunes ocasions li costa de posar-se en funcionament ja que ha de carregar molts fitxers de configuració o que AForge.Genètic donava problemes en la configuració quan menys t'ho esperaves perquè semblava, la que hi havia, una configuració perfectament establerta, etc. Tanmateix, aquests obstacles o problemes els vam anar podent solucionar amb temps, paciència i ajuda del nostre tutor..

En referència a l'apartat humà, el problema més greu amb el qual ens hem trobat ha fet referència al fet que un mes abans d'haver de presentar la present memòria, el nostre tutor de projecte, el Dr. Gustavo Patow, va patir un important problema de salut, del qual, a poc a poc es va recuperant, i vam haver de posar-nos en contacte amb algú altre de la Universitat de Girona perquè ens ajudés a acabar el projecte i, sobretot, la memòria. Finalment, aquestes persones van ser els doctors Jordi Regincós Isern i Fernando Julián Pérez, als quals volem agrair la seva ajuda i consells des d'aquestes pàgines.

10.- Resultats

En aquesta etapa, s'hi desplega una descripció detallada del procés de desenvolupament del projecte. També s'hi mostren els resultats obtinguts així com el grau d'assoliment dels objectius.

10.1 Prova 1

El fitxer de configuració d'entrada del qual hem fet ús ha estat el que es pot observar tot seguit:

```
population;201
parts;C:\Assembler\Demos\prova1-3Gears.xml
objective;0,5
reescalat;5001
tolerancia;201
epochTimes;10
```

Així mateix, l'XML "prova1-3Gears.xml" és el següent:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<assembly xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="assembly.xsd">
  <creation>
    <name>g1</name>
    <part_info>
      <part>new gear</part>
```

```

        <param>5</param>
    </part_info>
</creation>
<creation>
    <name>g2</name>
    <part_info>
        <part>new gear</part>
        <param>10</param>
    </part_info>
</creation>
<creation>
    <name>g3</name>
    <part_info>
        <part>new gear</part>
        <param>15</param>
    </part_info>
</creation>
<operation>
    <constraint>join</constraint>
    <param>g1</param>
    <param>g2</param>
    <param>30</param>
</operation>
<operation>
    <constraint>join</constraint>
    <param>g2</param>
    <param>g3</param>
    <param>60</param>
</operation>
<motor>
    <motor_name>g1</motor_name>
</motor>
</assembly>

```

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

Tal i com es pot copsar, en aquest XML es carreguen tres engranatges. Amb posterioritat al fet de passar l'algorisme d'optimització, s'ha trobat una solució (vegeu Figura 15):

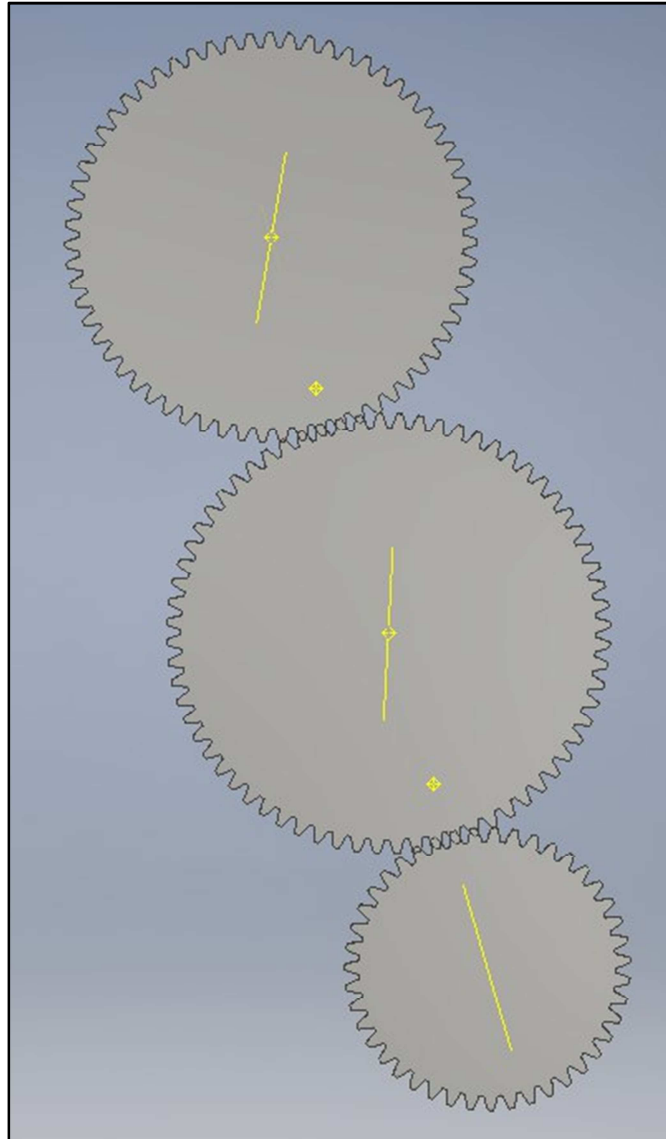


Figura 15 Els tres engranatges de sortida

I el resultat, vist amb format XML és el següent::

```
<?xml version = "1.0" encoding = "ISO-8859-1" ?>
<assembly xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="assembly.xsd">
  <creation>
    <name>g2</name>
    <part_info>
      <part>new gear</part>
      <param>13</param>
    </part_info>
  </creation>
  <creation>
    <name>g3</name>
    <part_info>
      <part>new gear</part>
      <param>14</param>
    </part_info>
  </creation>
  <creation>
    <name>g1</name>
    <part_info>
      <part>new gear</part>
      <param>9</param>
    </part_info>
  </creation>
  <operation>
    <constraint>join</constraint>
    <param>g2</param>
    <param>g3</param>
    <param>64</param>
  </operation>
  <operation>
    <constraint>join</constraint>
    <param>g3</param>
    <param>g1</param>
    <param>64</param>
  </operation>
  <motor>
    <motor_name>g2</motor_name>
  </motor>
</assembly>
```


Tal i com podem constatar mitjançant l'observació de la resposta XML, està compost pels següent paràmetres:

- Engranatge g2
- Engranatge g3
- Engranatge g1
- Connexió entre g3 i g2
- Connexió entre g3 i g1
- Com a motor posem el g2

10.2 Prova 2

El fitxer de configuració d'entrada del qual hem fet ús ha estat el que es pot observar tot seguit:

```
population;201
parts;C:\Assembler\Demos\prova2-2gears.xml
objective;0,5
reescalat;5001
tolerancia;201
epochTimes;1
```

Així mateix, l'XML "prova2-2gears.xml" és el següent:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<assembly xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="assembly.xsd">
  <creation>
    <publicPrivate>public</publicPrivate>
    <name>g1</name>
    <part_info>
      <part>new gear</part>
      <param>10</param>
    </part_info>
  </creation>
  <creation>
    <publicPrivate>public</publicPrivate>
    <name>g2</name>
    <part_info>
      <part>new gear</part>
      <param>5</param>
    </part_info>
  </creation>
  <operation>
    <constraint>join</constraint>
    <param>g1</param>
    <param>g2</param>
    <param>30</param>
  </operation>
  <motor>
    <motor_name>g2</motor_name>
  </motor>
</assembly>
```

Tal i com es pot copsar en aquest XML es carreguen dos engranatges. Amb posterioritat al fet de passar l'algorisme d'optimització, s'ha trobat una solució (vegeu Figura 16):

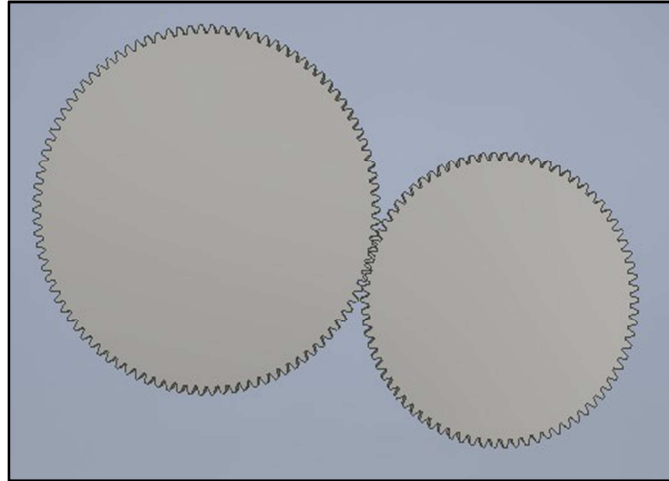


Figura 16 Els dos engranatges de sortida

I el resultat, vist amb format XML és aquest:

```
<?xml version = "1.0" encoding = "ISO-8859-1" ?>
<assembly xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="assembly.xsd">
  <creation>
    <name>g1</name>
    <part_info>
      <part>new gear</part>
      <param>20</param>
    </part_info>
  </creation>
  <creation>
    <name>g2</name>
    <part_info>
      <part>new gear</part>
      <param>16</param>
    </part_info>
  </creation>
  <operation>
    <constraint>join</constraint>
    <param>g1</param>
    <param>g2</param>
  </operation>
</assembly>
```

```
<param>16</param>
</operation>
<motor>
  <motor_name>g1</motor_name>
</motor>
</assembly>
```

Tal i com podem constatar amb l'observació de la resposta XML, està compost pels següent paràmetres:

- Engranatge g2
- Engranatge g1
- Connexió entre g2 i g1
- Com a motor posem el g1

11.- Conclusions finals

L'onzè punt és aquell en el qual presentem una reflexió final sobre l'assoliment dels requisits establerts, l'explicació de les possibles desviacions de la planificació inicial i, per acabar, la crítica dels resultats obtinguts.

A l'inici del treball es van definir diferents propòsits per tal de desenvolupar el present projecte, l'objectiu principal del qual era el de crear un algorisme genètic amb la finalitat de poder optimitzar un sistema d'engranatges mecànics. Per arribar a aconseguir aquesta finalitat, vam dividir el projecte en diverses parts en referència a les més significatives de les quals anirem presentant les conclusions que hem pogut anar extraient

- Estudi del funcionament de l'API d'Autodesk Inventor:
 - Majoritàriament, en aquest apartat s'ha anat estudiant i assimilant al mateix ritme que anàvem avançant en el projecte. L'API té bona documentació i posseeix tot un fòrum al darrere amb una bona comunitat, la qual cosa ens ha permès d'anar avançant en el coneixement de l'Autodesk Inventor, el qual, finalment, hem pogut comprendre amb una certa minuciositat per a les tasques d'execució que exigia el nostre projecte..
 - Malgrat això, en no tractar-se d'un programari de codi obert i, per tant, no poder observar el codi, el programari en algunes ocasions ha generat problemes de programació, raó per la qual en algunes

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

ocasions hem hagut de preguntar o demanar informació en el fòrum per tal que col·laboressin amb nosaltres a l'hora de solucionar la complicació que ens havia sorgit.

- Lenguatge de programació C#
 - Aquest llenguatge de programació tot i que originàriament provenia i era semblant a C i a C++, en realitat hem pogut comprovar que hi disposa de moltes diferències ja que, al contrari que C++ , conté *garbage collection*, és a dir, recollida de memòria brossa, raó per la qual no fa falta gestionar de la mateixa manera els objectes.
 - Per altra banda, recull un conjunt d'instruccions molt més complet que no pas el de C++, raó per la qual ens hem hagut d'anar llegint la documentació proporcionada per Microsoft amb la finalitat de poder comprendre adequadament el llenguatge.

- Microsoft Visual Studio 2017
 - En aquest IDE desenvolupat per Microsoft, es percep que hi ha molt temps, molta experiència de programació al darrere i, conseqüentment, posseeix moltes i diverses funcionalitats.
 - No obstant això, cal assenyalar que l'ús aquest IDE, en alguns moments, esdevé excessivament feixuc, a més a més que en algunes ocasions s'excedeix en el temps que cal per posar-se en funcionament i, alhora, inclou processos en *background* que agafen molts recursos

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

del sistema, la qual cosa, en algunes ocasions, afecta significativament el rendiment del programari.

- AForge.Genètic
 - Tot i que inicialment se'ns va fer complicat utilitzar aquesta llibreria pel fet que se'n localitza escassa documentació que s'hi refereixi i, alhora, que no disposi d'una comunitat activa al darrere, hem pogut anar solucionant totes les problemàtiques que se'ns han anat generant mirant el codi font. De la mateixa manera, cal indicar que ha resultat molt més senzill d'afegir al nostre projecte que d'altres *framework* atès que aporta unes instruccions molt clares de com afegir-lo en un projecte de C# sobre Visual Studio.
 - Aquesta llibreria és notablement completa en comparació a moltes d'altres. Malgrat això, s'ha d'estar molt alerta a la configuració que li donis atès que si no hi estàs molt alerta, pot no funcionar de forma correcta o adequada. Per tant, s'ha de anar molt amb compte ja que en treballar sobre algorismes genètics que són semialeatoris es pot donar el cas que amb la mateixa configuració en moltes ocasions funcioni correctament però que un determinat moment falli a causa d'algun problema de configuració.

En conclusió, a l'entorn del projecte que hem anat realitzant i configurant hem pogut observar que realment la solució obtinguda pel programa creat per nosaltres mateixos representa, com ens proposàvem inicialment, una certa millora en relació

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

amb la situació inicial per a l'optimització de mecanismes mecànics senzills, de tal manera que amb el nostre projecte es pot estalviar temps i, a la llarga, recursos, ja que s'agilitzen de manera prou significativa les etapes inicials d'un projecte. I, econòmicament, a nivell professional i/o empresarial considerem que aquesta aplicació podria resultar viable ja que el seu cost seria, aproximadament de 7398 euros incloent-hi els recursos humans, el programari i l'amortització de maquinari.

Tanmateix, com que no hem trobat altres tecnologies que realitzin un treball semblant al que hem presentat en el nostre projecte, no hem pogut realitzar comparacions amb d'altres tecnologies.

Per altra banda, i per anar acabant, el projecte que hem realitzat ens ha resultat molt útil i enriquidor en el sentit d'experimentar i testejar o validar de manera més profunda els algoritmes genètics que he pogut conèixer i estudiar mentre he cursat l'assignatura d'Intel·ligència Artificial. Així mateix, ens ha enriquit en el coneixement aprofundit d'alguns programaris que fins ara només coneixíem de forma més superficial. Així mateix, ens ha satisfet poder continuar una tasca endegada fa un temps per una companya, l'Aura Ruiz Duñach, seguint els consells del nostre tutor, el Dr. Gustavo Patow. Evidentment obtenir finalment un resultat adequat en relació als propòsits inicials també ens ha satisfet notablement.

Quant a aspectes que ens han resultat més complexos o feixucs podríem esmentar els que ja han quedat anotats anteriorment en relació als programaris que no sempre responien com esperàvem.

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

Finalment, voldríem afegir que si ara tornés a iniciar el projecte i pogués substituir alguna cosa de les realitzades en relació amb el seu desenvolupament, potser el que faria seria passar-me més temps observant l'API d'Autodesk Inventor i llegint la documentació de l'AForge ja que això, possiblement, em facilitaria de no haver de dedicar, posteriorment, quan l'execució del projecte ja estigués en un punt força avançat, temps a aquestes tasques, sinó centrar-nos pròpiament en la consolidació del projecte.

I ja per acabar, no voldríem cloure aquesta memòria sense agrair tot el suport que hem rebut de la Universitat de Girona i, en especial del nostre tutor, Dr. Gustavo Patow, així com els que ens han ajudat a poder acabar la present memòria després de la seva malaltia, el Dr. Jordi Regincós i el Dr. Fernando Julián.

12.- Treball de futur

En aquest punt, hi realitzem l'anàlisi i valoració de les possibles ampliacions o millores del projecte per a un treball futur partint del treball que nosaltres hem realitzat.

Tot i que amb el desenvolupament del projecte hem assolit els objectius plantejats inicialment, hi ha uns quants aspectes que es poden continuar treballant per a evolucionar i millorar-lo:

- Per una part, al final del projecte ens hem adonat que podríem aplicar diferents tipus d'optimització, en el cas actual només mirem la relació de transmissió entre tots els engranatges per tal d'obtenir la velocitat i força desitjats en la sortida. En treballs futurs, es podrien implementar altres sistemes d'optimitzacions com ara, per exemple, que ocupi un espai concret, que minimitzi el número d'engranatges que s'utilitzen...
- Un altre possible projecte o treball final seria poder instal·lar una interfície d'usuari per a poder generar els fitxers de configuració de tal manera que resulti molt més simple el fet de poder treballar amb aquest programari.

13.- Bibliografia i webgrafia

MITCHELL, Melanie (1996): *An introductipn to genetic algorithms*. Massachussets Institute of Technology. A Bratford Book The MIT Press.
<http://www.boente.eti.br/fuzzy/ebook-fuzzy-mitchell.pdf> (Consulta 06 de juny 2017).

RUIZ DUÑACH, Aura (2015). Memòria del projecte *Llenguatge d'assemblatge per a la construcció i simulació de mecanismes senzills*. Treball Fi de Grau dirigit pel Dr. Gustavo Patow. Universitat de Girona, <http://dugi-doc.udg.edu/bitstream/handle/10256/12023/1.%20Memoria.pdf?sequence=1>
(Darrera consulta, 06 de juny de 2017)

Aforge.NET :: Computer Vision, Artificial Intelligence, <http://www.aforgenet.com/>, 2017

Inventor | Mechanical Design & 3D CAD Software | Autodesk, <https://www.autodesk.com/products/inventor/overview>, 2017

C# | Microsoft Docs, <https://docs.microsoft.com/es-es/dotnet/csharp/csharp>, 2017

I

DE de Visual Studio, editor de código, Team Services y Mobile Center, <https://www.visualstudio.com>, 2017

Universitat de Girona > UdG, <https://www.udg.edu/ca/>, 2017

14.- Annexos

En aquest catorzè i penúltim apartat, hi recollim la informació complementària derivada del projecte.

14.1 Requisits per a Autodesk Inventor 2017

Aquesta informació referida als requisits que exigeix la instal·lació del programari Autodesk Inventor en la seva versió de l'any 2017 ha estat extreta de l'enllaç <https://goo.gl/NdEJjn>

System Requirements for Autodesk Inventor 2017 Windows	
Operating System ¹	<ul style="list-style-type: none"> • Microsoft® Windows® 10 (desktop OS) • 64-bit Microsoft Windows 8.1 with Update KB2919355 • 64-bit Microsoft Windows 7 SP1
CPU Type ²	<p>Recommended:</p> <p>Intel® Xeon® E3 or Core i7 or equivalent, 3.0 GHz or greater</p> <p>Minimum:</p> <p>64-bit Intel or AMD, 2 GHz or faster</p>

<p>Memory ³</p>	<p>Recommended:</p> <p>20 GB RAM or more</p> <p>Minimum:</p> <p>8 GB RAM for less than 500 part assemblies</p>
<p>Disk Space</p>	<p>Installer plus full installation: 40 GB</p>
<p>Graphics</p>	<p>Recommended:</p> <p>Microsoft Direct3D 11® or capable graphics card or higher</p> <p>Minimum:</p> <p>Microsoft Direct3D 10® capable graphics card or higher</p> <p>See the Certified Hardware</p>
<p>Display Resolution</p>	<p>1280 x 1024 or higher.</p> <p>Recommended scaling: 100%, 125%, 150% or 200%.</p>

Other	<ul style="list-style-type: none">● DVD-ROM □● Internet connection for Autodesk® 360 functionality, web downloads, and Subscription Aware access● Adobe® Flash® Player 15 □● Microsoft Internet Explorer® 11 or equivalent● Full local install of Microsoft® Excel 2010, 2013 or 2016 for iFeatures, iParts, iAssemblies, thread related commands, clearance/threaded hole creation, Global BOM, Parts Lists, Revision Tables, spreadsheet-driven designs and Studio animation of Positional Representations. Excel Starter®, Online Office 365® and OpenOffice® are not supported.● 64-bit Microsoft Office is required to export Access 2007, dBase IV, Text and CSV formats.● Microsoft .NET Framework 4.6 or later● Virtualization supported on Citrix® XenApp™ 7.7 and 7.8; Citrix XenDesktop™ 7.7 and 7.8 (requires Inventor Network Licensing).
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

For Complex Models, Complex Mold Assemblies, and Large Assemblies (typically more than 1,000 parts)	
CPU Type	Recommended: Intel Xeon E3 or Core i7 or equivalent, 3.30 GHz or greater ²
Memory	Recommended: 24 GB RAM or greater ³
Graphics	Recommended: Microsoft Direct3D 11 or capable graphics card or higher See the Certified Hardware

Notes:

1. Autodesk Inventor and Autodesk Vault must both be the same language on a given computer. English versions of these applications will run on any language operating system. Other language versions of these applications will run on operating systems of that same language.
2. Autodesk Inventor 2017 is optimized to take advantage of the SSE2 extended instruction sets supported on Pentium 4, AMD Athlon 64, and AMD Opteron processors. Inventor 2017 will not install on computers that do not support

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

SSE2. Several utilities are available on the Internet that report CPUID including supported instructions sets.

- Autodesk recommends settings which allow Microsoft Windows to manage virtual memory as needed. There should always be at least twice as much free hard disk space as system memory (RAM).
- DVD-ROM drive is not required if installing from USB or software download.
- The multimedia learning components of the Inventor Help system, such as the UI Video Tour, Command Reference, and Show Me animations, require Adobe Flash Player 10. If it is not already installed, you can download it from the Adobe website.

Es pot instal·lar el programari Autodesk Inventor Professional en un ordinador Mac en una partició de Windows. El sistema ha d'utilitzar Apple Boot Camp® per gestionar una configuració de doble sistema operatiu i complir amb els requisits mínims del sistema:

:

Boot Camp	
Mac OS	Recommended: Mac OS® X 10.10.x

	<p>Minimum:</p> <p>Mac OS X 10.9.x</p>
CPU Type	Intel Core 2 Duo or greater, 3.0 GHz or greater
Memory	<p>Recommended:</p> <p>16 GB RAM or greater</p> <p>Minimum:</p> <p>8 GB RAM</p>
Partition Size	<p>Recommended:</p> <p>500 GB or greater</p> <p>Minimum:</p> <p>200 GB</p>
Windows OS	<ul style="list-style-type: none"> • Microsoft Windows 10 (desktop OS) • 64-bit Microsoft Windows 8.1 with Update KB2919355 • 64-bit Microsoft Windows 7 SP1

Autodesk Inventor Professional es pot utilitzar en el Mac a través de “Parallels Desktop” per Mac sense haver d’arrancar en el sistema operatiu de Windows, fet pel

qual és fàcil canviar entre plataformes. El sistema ha de complir els requisits següents:

Mac Virtualization on Parallels Desktop	
Mac OS	Recommended: Mac OS X 10.10.x Minimum: Mac OS X 10.9.x
Parallels	Parallels Desktop 10 or higher
CPU Type	Intel Core 2 Duo or greater, 3.0 GHz or greater
Memory	Recommended: 16 GB RAM or greater Minimum: 8 GB RAM
Free Disk Space	Recommended: 250 GB or greater Minimum: 100 GB

Windows OS	<ul style="list-style-type: none">• Microsoft Windows 10 (desktop OS)• 64-bit Microsoft Windows 8.1 with Update KB2919355• 64-bit Microsoft Windows 7 SP1
-------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

15.- Manual d'usuari i instal·lació

En aquest darrer punt és en el qual s'explica la manera com s'ha d'instal·lar i utilitzar l'aplicació. Cal indicar que hem partit del projecte de d'Aura Ruiz Duñach (2015) perquè es tracta exactament del mateix manual d'usuari i d'instal·lació que el que ella va anotar per al seu projecte (Ruiz Durach, 2015, p. 122-126)

15.1 Instal·lació

Per al correcte funcionament del projecte necessitem instal·lar les següents aplicacions:

Descarregar i instal·lar Autodesk Inventor:

- <http://www.autodesk.com/education/free-software/inventor-professional>
(sense cost econòmic si es tracta d'un estudiant universitari)

Descarregar i instal·lar el Visual Studio:

- <https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs>

15.2 Entorn de desenvolupament

Com que el projecte està pensat per a continuar essent desenvolupat, no hi ha un procés d'instal·lació, sinó d'execució seguint les següents pautes:

1. Copiar la carpeta Assembler, inclosa en el disc adjunt, al directori c:\.
2. Obrir el projecte amb un dels mètodes indicats:
 - a. Mètode 1: obrir el fitxer Assembler.sln del directori C:\Assembler\Assembler.
 - b. Mètode 2: Executar el *Visual Studio* i anar al File->Open->Project/Solution... . Dirigir-se a l'adreça C:\Assembler\Assembler i seleccionar el fitxer Assembler.sln.

15.3 Execució

Disposant del projecte obert al Visual Studio, s'inicia el programa amb la icona . Acte seguit, s'obrirà l'aplicació d'*Autodesk Inventor* (Vegeu figura 16), en cas que no ho estigués, juntament amb un fitxer d'assemblatge .iam buit, i es mostrarà el menú de l'aplicació (vegeu figura 17).

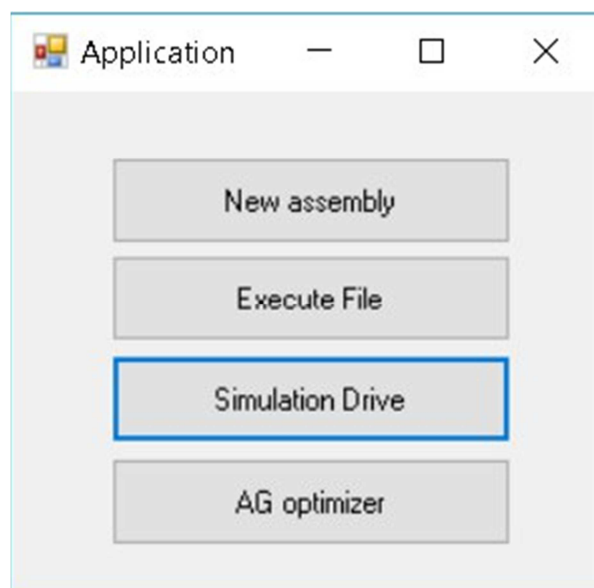


Figura 17 Menú de l'aplicació

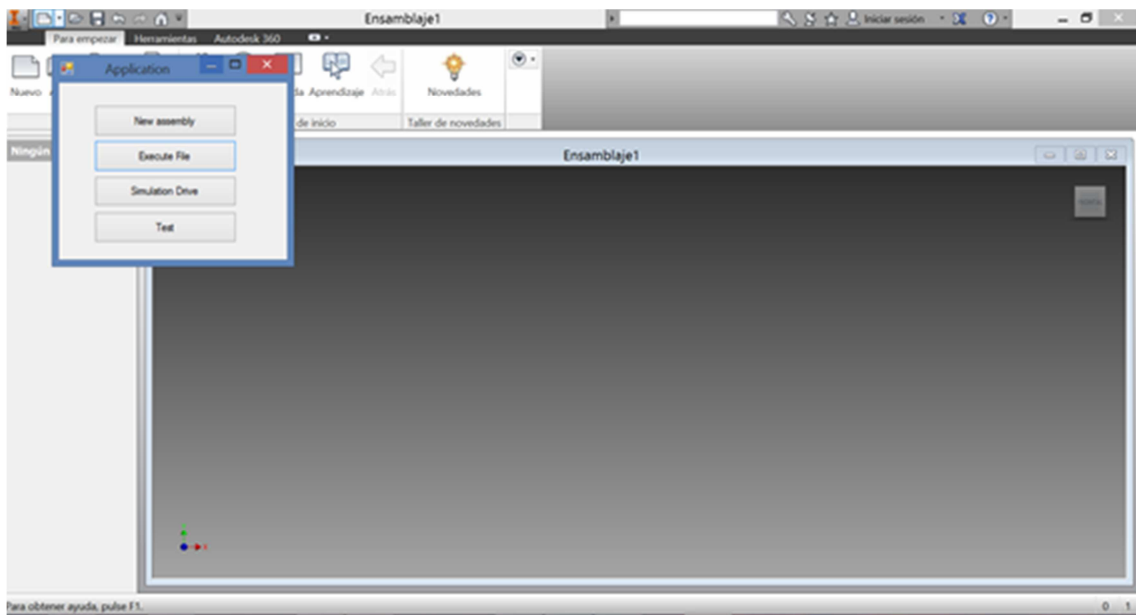


Figura 18 Inici de les aplicacions

Funcionalitats dels botons:

- Botó "New Assembly": s'obrirà un nou assemblatge buit igual com l'inicial (vegeu Figura 19).

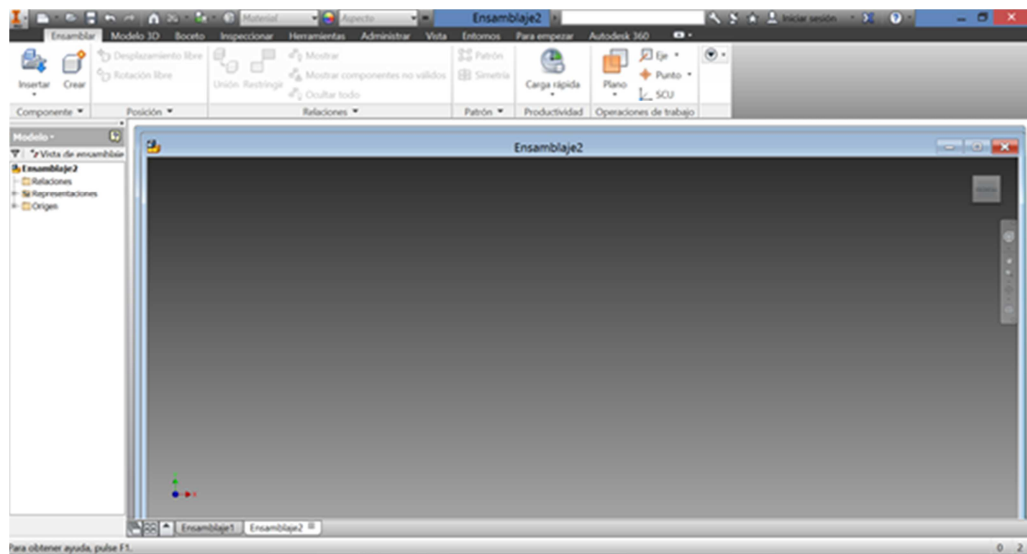


Figura 19 Nou assemblatge obert

Creació i optimització de mecanismes mecànics senzills

David Bruguera Tornés

- Botó "Execute File": s'obre el navegador per seleccionar el fitxer d'entrada (vegeu Figura 20).

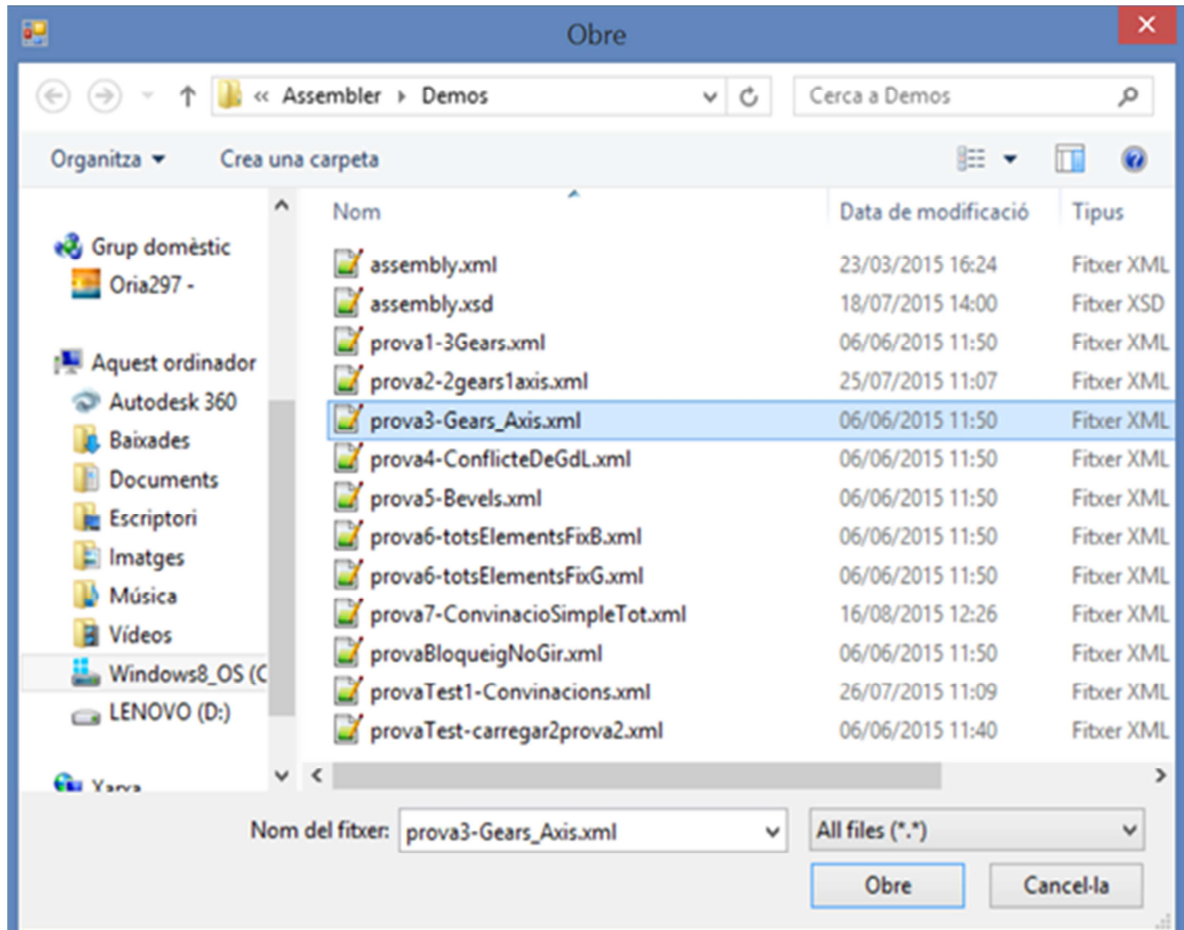


Figura 20 Carpeta inici al Demo

Un cop seleccionat el fitxer desitjat, es prem el botó "Obre" (vegeu Figura 21). El resultat final serà l'assemblatge carregat a l'inventor.

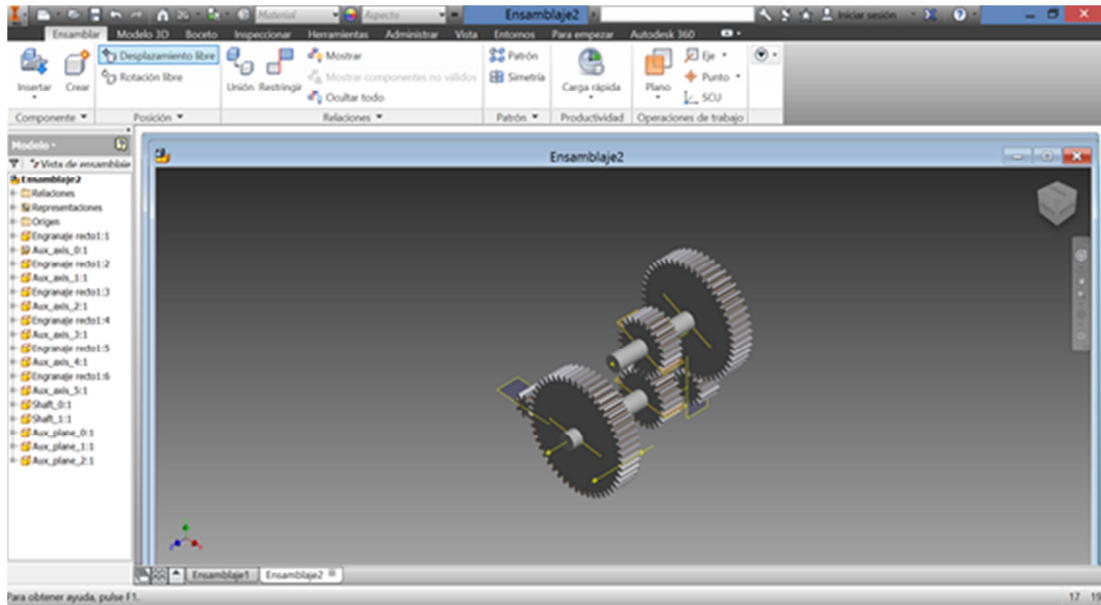


Figura 21 Assemblatge generat resultant

- Botó "Simulation Drive": la simulació prèviament carregada iniciarà el moviment.
- Botó "AG optimizer": és el nou botó implementat. Primer s'ha de triar el fitxer de configuració (vegeu Figura 20) amb el selector de fitxers i, posteriorment, es mostrarà el resultat (vegeu Figura 22).

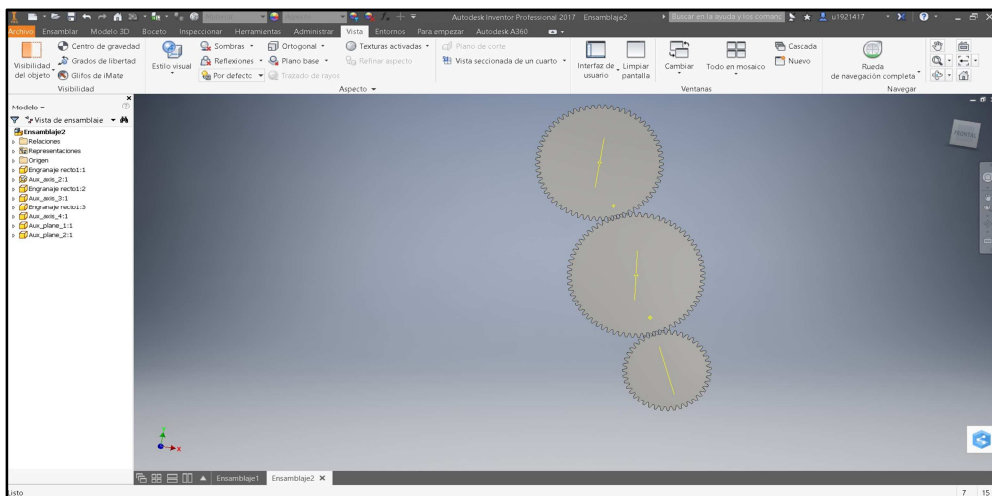


Figura 22 Exemple de sortida

15.4 Alertes habituals

En el procés d'utilització de l'aplicació poden aparèixer diverses pantalles d'avís. A continuació es mostren quines alertes són, quan apareixen i la solució proposada en aquests casos.

Falta càrrega de l'assemblatge

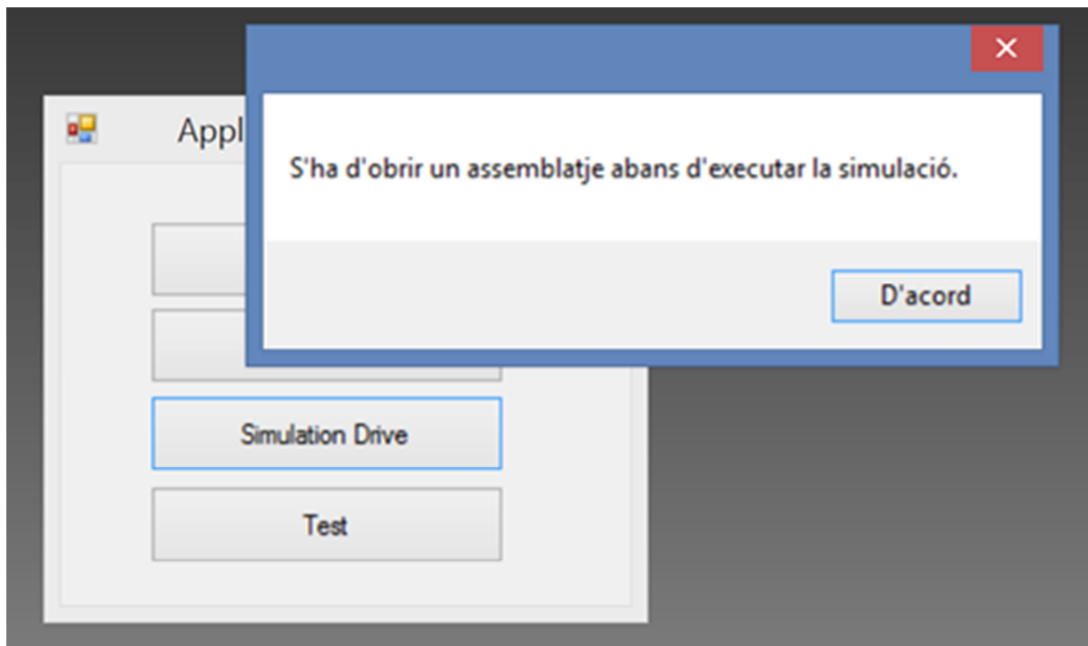


Figura 23 Error no s'ha obert

Aquest avís acostuma a aparèixer quan s'intenta executar el "Simulation Drive" sense tenir cap assemblatge prèviament carregat o, en cas contrari, l'assemblatge o acoblament no conté cap element amb la funcionalitat de motor.

Solució proposada: executa un fitxer XML que contingui les etiquetes pròpies pel motor.

Falta assemblatge buit

L'avís apareix indicant l'absència d'un assemblatge i sorgeix quan s'intenta seleccionar un fitxer XML amb l'"Execute File" sense tenir cap fitxer .iam obert a l'Autodesk *Inventor*.

Solució proposada: executa el botó "New Assembly" abans de seleccionar l'"Execute File".

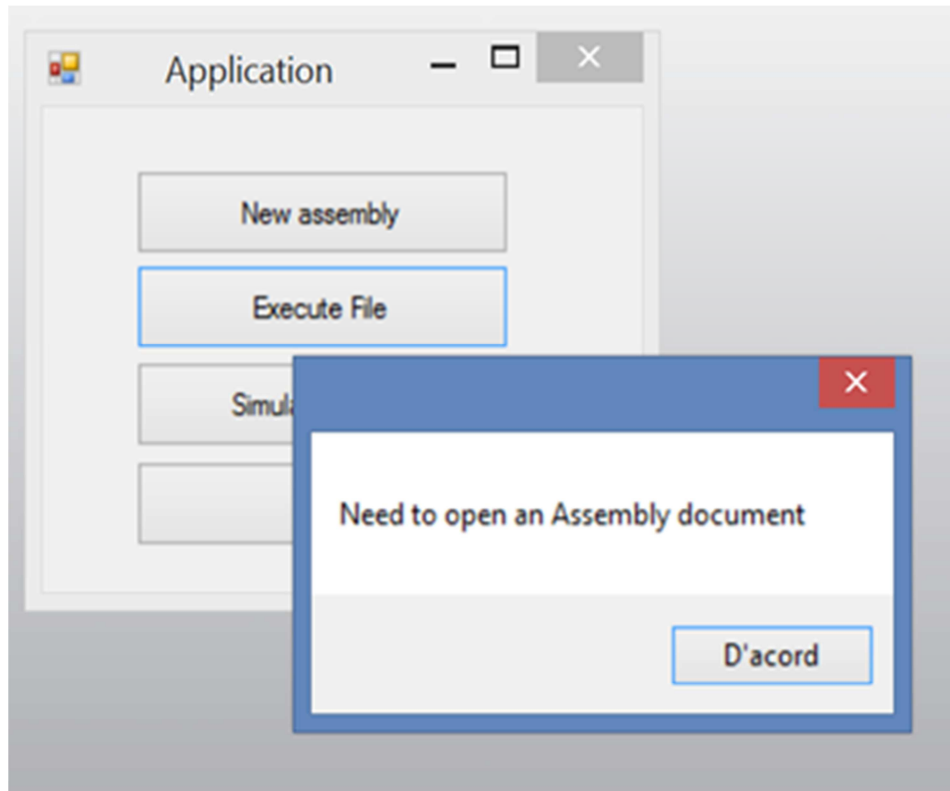


Figura 24 Error assemblatge buit

15.3 Disseny de Fitxers XML

Cal conèixer l'estructura i les etiquetes que necessiten els fitxers XML que pot acceptar l'aplicació. Les bases necessàries per a dissenyar correctament un fitxer d'entrada XML d'assemblatges es troba al punt 9.1 d'aquesta documentació.