Índex

1. Introducció 3
   1.1. Presentació 3
   1.2. Motivació 4
   1.3. Conceptes bàsics 5
       1.3.1. Fitxer de dades 5
       1.3.2. Visualitzador de molècules 6
       1.3.3. Aigua 7
       1.3.4. Lípid 8
       1.3.5. Sistema estel·lar 9

2. Objectius 11

3. Estudi 13
   3.1. Eines i conceptes 13
   3.2. Metodologia de desenvolupament i llenguatge de programació 16
   3.3. Accés al sistema 18

4. Antecedents i Factibilitat 19
   4.1. Eines i conceptes 19
   4.2. Factibilitat 22
       4.2.1. Factibilitat tècnica 23
       4.2.2. Factibilitat operativa 25
       4.2.3. Factibilitat econòmica 26

5. Desenvolupament tècnic 31
   5.1. Especificació 32
       5.1.1. Descripció general 32
       5.1.2. Requeriments funcionals 34
       5.1.3. Requeriments no funcionals 44
   5.2. Anàlisi de l’aplicació 53
       5.2.1. Diagrama de context 54
       5.2.2. Casos d’ús 56
       5.2.3. Diagrama de classes 72
       5.2.4. Atributs y operacions de classes 75
   5.3. Disseny de l’aplicació 93
5.3.1. Diagrama de seqüència
5.3.2. Disseny interfície

6. Investigació
6.1. Estudi i observació de les llibreries gràfiques necessàries per a la visualització del sistema (Directx9)
6.1.1. Direct3D
6.1.2. DXUT
6.2. Diàlegs estàndard per a poder obrir o guardar fitxers
6.3. Opció de visualització d’una paleta de colors per a modificacions de colors
6.4. Estudi i recerca de llibreries gràfiques per a l’exportació d’imatges
6.5. Creació de fitxers de vídeo

7. Resultats experimentats
7.1. Connexió dinàmica
7.2. Visualització
7.3. Exportació d’imatges
7.4. Generació de vídeos

8. Anàlisi econòmic global
8.1. Planificació inicial
8.2. Planificació final
8.3. Costos del projecte

9. Conclusió i futur de la plataforma

10. Annex
10.1. Manual d’instal·lació
10.2. Manual de l’usuari
10.2.1. Visualització
10.2.2. Exportació a fitxers d’imatge
10.2.3. Exportació a fitxers de vídeo

11. Bibliografia
1. Introducció

1.1. Presentació

El projecte s’anomena Animació de Sistemes de Partícules amb Connexió dinàmica. És un projecte final de carrera de la modalitat A del pla 91 proposat per un professor del departament de Física i Enginyeria Nuclear, amb una càrrega lectiva de 30 crèdits.

Aquest projecte sorgeix de la necessitat de crear una eina de representació gràfica on-line de la dinàmica de sistemes clàssics i més concretament de comunicar la sortida de dades d’una altra eina amb la nova a través d’un fitxer estructurat.

Una altra necessitat és la de poder guardar la representació creada en pantalla en un fitxer d’imatge, ja sigui format jpg, bmp, gif, etc. i la de generar a partir d’una seqüència de fitxers, i per tant d’imatges, una pel·lícula de vídeo, en format avi.

En general, es podrà aplicar a sistemes de n partícules governats per les lleis de la mecànica clàssica, com per exemple, líquids, gasos o sistemes estel·lars.

Gràcies a l’ajut i al seguiment de la feina per part de la figura del director del projecte, ha fet possible desenvolupar aspectes tant importants com la comunicació o facilitat de treball en grup.

S’ha de destacar la gran labor d’investigació que s’ha necessitat per a realitzar el desenvolupament i la documentació del projecte, ja que ha contribuït a ampliar coneixements en programació orientada a objectes, programació gràfica i generació de documentació.
1.2. Motivació

Com ja s’ha comentat en l’apartat anterior la principal motivació de la creació d’aquesta eina és la necessitat de crear una eina de representació gràfica on-line de la dinàmica de sistemes clàssics i més concretament de comunicar la sortida de dades d’una altra eina amb la nova a través d’un fitxer estructurat.

És un projecte molt interessant ja que es barregeu diferents aspectes entorn al seu desenvolupament. S’ha d’adquirir coneixement de programació orientada a objectes, de gràfics, i d’exportació a fitxers d’imatges o de vídeo.

La finalitat és poder generar els fitxers d’imatge o de vídeo desitjats i poder veure aquest resultat en qualsevol màquina.

També es vol poder guardar la representació creada en pantalla en un fitxer d’imatge, ja sigui format jpg, bmp, gif, etc. i generar, a partir d’una seqüència de fitxers una pel·lícula de vídeo, en format avi, per a que sigui compatible amb diferents sistemes operatius.

L’eina que es pretén crear facilitarà molt la feina del client ja que només indicant un fitxer d’entrada de dades i una configuració obtindrà la visualització del sistema. I només indicant un nom de fitxer, podrà exportar la visualització en una fitxer d’imatge. Tanmateix, indicant un conjunt de fitxers podrà generar un vídeo amb la seqüència de les imatges, simplement amb un clic.
1.3. Conceptes bàsics

En aquest punt veurem el conceptes bàsics necessaris per entendre com funciona el programa.

1.3.1. Fitxer de dades

El fitxer de dades és el conté totes les posicions de les molècules que s’han de visualitzar. Aquest fitxer s’ha generat a partir d’un programa de representació física que disposa el departament de Física i Enginyeria Nuclear.

Aquests fitxers seran llegits pel programa i posteriorment visualitzats.

El fitxer de dades sempre conté la mateixa informació, és a dir, sigui quin sigui el sistema a visualitzar sempre s’estructura de la mateixa manera. Cada línia del fitxer conté les coordenades \( x, y, z \) dels punts a dibuixar.

En aquest exemple es pot observar un fitxer que conté 3 àtoms. El valor final és el mida de la capsà contenidora.
1.3.2. Visualitzador de molècules

El visualitzador de molècules és la principal funcionalitat del programa, ja que permet mostrar a l’usuari les molècules a través d’un món tridimensional. Per a poder visualitzar les molècules és necessari haver indicat abans una configuració i el fitxer a obrir.

El visualitzador pot representar qualsevol sistema de partícules, però s’estudiaran 3 exemples diferents per a veure la seva versatilitat. Un exemple serà una molècula d’aigua, on serà necessari mostrar els seus components que són àtoms i enllaços; en canvi en els altres exemples, com el lípid o el sistema estel·lar només serà necessari mostrar àtoms o estels.

Quan el visualitzador presenti un àtom, haurà de tenir en compte la posició donada pel fitxer de dades, i el color i el mida (radi) donat per la configuració establerta de la molècula. De totes maneres aquestes configuracions són modificables en tot moment de la visualització.

Existeixen diferents tipus d’enllaços químics (iònic, covalent, metàl·lic, etc.) que mantenien units els àtoms que formen la molècula. En tots els casos un enllaç és una força que uneix àtoms, i per tant, si es volgués veure a simple vista una molècula tan sols podríem veure els àtoms, ja que els enllaços serien invisibles. De totes maneres el visualitzador presenta els enllaços dibuixant un cilindre entre els àtoms que comparteixen l’enllaç.

L’aplicació permet també configurar la visualització a través d’operacions geomètriques com són la rotació, el zoom o la translació, fent així el sistema molt més navegable i amigable.
1.3.3. L’Aigua

La molècula d’aigua (H₂O) és el principal exemple que s’ha fet servir per posar en marxa el sistema. Com indica la fórmula, l’aigua és una molècula formada per dos àtoms d’hidrogen i un àtom d’oxigen, units per dos enllaços covalents.

Un enllaç covalent es forma quan la diferència d’electronegativitat no és suficientment gran com per efectuar una transferència d’electrons, llavors els àtoms comparteixen un o més parells electrònics en un nou tipus d’orbital, anomenat orbital molecular. A diferència de l’enllaç ionic, en el que es produeix la transferència d’electrons d’un àtom a un altre, en l’enllaç químic covalent, els electrons d’enllaç són compartits pels dos àtoms.

La distància entre un oxigen i un hidrogen és aproximadament un Angström, i l’angle intern molecular format pels dos enllaços és d’aproximadament 105 graus.

Ens els exemples estudiats es mostres conjunts de 216 molècules d’aigua.
1.3.4. El lípid

Com a segon exemple d’ús de l’aplicació s’ha usat el lípid.

El lípid es representa per cadenes de cinc components, on una d’elles actua com a cap i les altres quatre com a cua.

![Diagrama del lípid](image)

Es representen 76 lípids envoltats per 600 molècules d’aigua. Les relacions entre ells són de tipus atractiu entre el cap del lípid i l’aigua, i de tipus repulsiu entre la cua i l’aigua.

En aquest exemple, encara que sembli molt senzill, és capaç de modelar l’estructura i la dinàmica de les membranes biològiques i els seus moviments.
1.3.5. Sistema estel·lar

Com a tercer exemple de visualització s’ha usat un sistema estel·lar. El sistema estel·lar es presenta com a un únic component.

Les estrelles que formen un sistema estel·lar estan vinculades físicament a través dels efectes de la seva gravitació mútua, però en la visualització no es veurà reflectida la seva vinculació.

En els casos estudiats es veuran uns 76 estels.
2. **Objectius**

Es pretén crear una eina de representació gràfica "on-line" de la dinàmica de sistemes clàssics. En general, es podrà aplicar a sistemes de n partícules governats per les lleis de la mecànica clàssica, com per exemple, líquids, gasos o sistemes estel·lars.

El projecte consisteix en elaborar un codi que permeti generar una interfície gràfica que pugui visualitzar el sistema físic segons les especificacions de les posicions donades. També es pretén que pugui generar imatges fites en fomat gif, jpg, bmp ... del que es visualitza en pantalla, i de manera opcional poder generar seqüències animades en format vídeo (.avi) a mida que es desenvolupa la representació del sistema, per a que sigui compatible amb diferents sistemes operatius.

Com a millora es vol intentar poder llegir un seguit de fitxers d'un determinat directori i a mida que es va generant la interfície gràfica es vagi generant un fitxer de vídeo (.avi també) amb les seqüències dibuixades.

Com que el departament ja compta amb diferents projectes realitzats en una plataforma Linux es va pensar fer-lo en Windows per a que fos totalment diferent de tots els altres, sobretot en l'àmbit de la interfície gràfica.
3. Eines i Conceptes

3.1. Arquitectura del sistema

Per a desenvolupar el nostre sistema s’ha decidit una programació orientada a objectes, ja que és un llenguatge molt usat en la programació gràfica.

La programació orientada a objectes defineix els programes en termes de “classes d’objectes” que són entitats que combinen estats (dades), comportament (mètodes) i identitat (propietat de l’objecte). La programació orientada a objectes expressa un programa com un conjunt d’aquests objectes, que col·laboren entre ells per a la realització de tasques. Això permet fer el programes o mòduls més fàcils d’escriure, mantenir i reutilitzar.

D’aquesta manera, un objecte conté tota la informació (atributs) que permet definir-lo i identificar-lo respecte altres objectes (de la mateixa o d’altres classes). També disposa de mecanismes d’interacció (mètodes) que s’encarreguen de la comunicació entre els objectes, ja siguin de la mateixa o diferent classe, i del canvi d’estat dels propis objectes. Aquesta característica fa que es tractin com a unitats indivisibles en la que no se separa la informació ni els mètodes.
Les principals característiques de la programació orientada a objectes són:

- **Abstracció**
  Cada objecte en el sistema serveix com a model d’un “agent” abstracte que pot realitzar una determinada tasca, informar i canviar el seu estat, i “comunicar-se” amb altres objectes en el sistema sense mostrar com s’implementen les seves característiques.

- **Encapsulament**
  Significa reunir tots els elements que puguin considerar-se que pertanyen a una mateixa entitat, al mateix nivell d’abstracció. Això permet augmentar la cohesió dels components del sistema.

- **Principi d’ocultació**
  Cada objecte està aïllat de l’exterior, i cada tipus d’objecte mostra als altres objectes una interfície que específica com pot interactuar amb els objectes de la classe. L’aïllament protegeix les propietats d’un objecte de la seva modificació sobre qui no tingui dret a fer-ho, només els propis mètodes interns de l’objecte poden accedir al seu estat. Això assegura que els altres objectes no poden canviar l’estat intern d’un objecte de manera inesperada.
• Polimorfisme
Comportaments diferents, associats a objectes diferents, poden compartir el mateix nom, i al cridar-los per aquest nom s'utilitzarà el comportament corresponent a l'objecte que s'estigui usant. És a dir, les referències i les col·leccions d'objectes poden contenir objectes de diferents tipus, i la crida d'un comportament en una referència produirà el comportament correcte per al tipus real de l'objecte referenciat.

• Herència
Les classes no estan aïllades, sinó que es relacionen entre sí, formant una jerarquia de classificació. Els objectes hereten les propietats i el comportament de totes les classes a les que pertanyi. L'herència organitza i facilita el polimorfisme i l'encapsulament permetent als objectes ser definits i creats com a tipus especialitzats d'objectes preexistents. Aquests poden compartir (i extendre) el seu comportament sense haver d'implementar el seu comportament. Això normalment es fa agrupant els objectes en classes i aquestes en arbres que reflecteixen un comportament comú. Si un objecte hereta de més d'una classe es diu que hi ha herència múltiple. No tots els llenguatges de programació suporten aquest tipus d'herència.
3.2. Metodologia de desenvolupament i llenguatge de programació

En aquest apartat es descriurà quin ha estat del model de desenvolupament que s’ha adoptat per desenvolupar el sistema i quins llenguatges de programació s’han utilitzat.

Com a metodologia de desenvolupament s’ha optat l’orientació a objectes amb notació en UML(*) . L’UML és un llenguatge gràfic per a visualitzar, especificar i documentar cadascuna de les parts que comprenen el desenvolupament d’un sistema software. A través de l’UML es proposa una metodologia de treball que facilita la tasca de desenvolupament i que garanteix un software amb un cert grau de qualitat.

Com a llenguatge de programació s’ha optat per C++, ja que una de les principals característiques és el suport per a la programació orientada a objectes, a més de ser un llenguatge molt robust. C++ està considerat pels experts com el llenguatge de programació més potent, degut a que permet treballar tant en alt com en baix nivell, en canvi, però és a la vegada un dels que menys automatismes té (obliga a fer gairebé tot manualment, igual que el llenguatge C) i això és el que "dificulta" molt el seu aprenentatge.

(*) UML: Unified Modeling Language (Llenguatge Unificat de Modelat).
Finalment, s'ha utilitzat un sèrie d'Apis per a la creació de la interfície gràfica. Aquestes Apis són DirectX i estan especialment creades per a facilitar la programació de videojocs en una plataforma Windows. Aquest kit de desenvolupament el distribueix gratuïtament per Microsoft i ja va inclòs en el propi Windows.

Hi ha moltes Apis en aquest paquet:

- **Direct3D:** per a dibuixar imatges en dos dimensions (planes), i per a representació d’imatges en tres dimensions.
- **DirectInput:** utilitzat per a processar dades del teclat, ratolí, joystick i altres controls per a jocs.
- **DirectPlay:** per a comunicacions en xarxa.
- **DirectSound:** per a reproducció i gravació de so.
- **DirectMusic:** per a la reproducció de pistes musicals compostes amb DirectMusic Producer.
- **DirectShow:** per a reproduir àudio i vídeo amb transparència de xarxa.
- **DirectSetup:** per a l’instal·lació de components DirectX.

Encara que hagi estat desenvolupat exclusivament per a la plataforma Windows, una implementació de la seva API es troba en progrés per a sistemes Unix (en particular Linux) i X Window System coneguda com Cedega, orientada a l’execució de jocs desenvolupats per al Windows sota sistemes Unix.
3.3. Accés al sistema

Com que el projecte s'ha realitzar exclusivament per al departament del Física i Enginyeria nuclear de la UPC i el programa s'instal·larà a l'equip on es farà servir, de moment no se li ha restringit l'accés al sistema. L'accés ve restringit per l'usuari de connexió a l'equip en qüestió.

Potser es podria tenir en compte alhora de fer les ampliacions del sistema i tenir en compte els usuaris que poden executar el programa i fins i tot restringir algunes funcionalitats segons un perfil, però no s'ha cregut convenient implementar-ho en aquest moment.
4. Antecedents i Factibilitat

4.1. Anàlisi d’antecedents

El departament de Física i Enginyeria Nuclear, va suggerir fer un anàlisi d’antecedents per a poder observar i comparar les seves funcionalitats i poder aconseguir una bona guia per a realitzar la implementació dels objectius plantejats.

Les aplicacions que s’havien utilitzat fins el moment i han de servir d’exemple són: Rasmol, Molden i gOpenMol.

Es farà un estudi d’aquestes aplicacions per a veure les seves funcionalitats i la metodologia usada.

Rasmol

![RasMol](image)

RasMol és una eina científica important per a la visualització de les molècules creada per Roger Sayle el 1992. RasMol s’utilitza per a veure les macromolècules i per a preparar imatges de la publicació. Sembla ser que l’accés al codi font és lliure, perquè segons comenten, permet adaptar les eines ràpidament i amb eficàcia.
Molden

Es un programa pre i post processador d'estructures moleculars i electròniques.

Formats de sortida:

- Xwindows
- Postscript
- OpenGL
- PovRay
- Tek4010, HPGL, ...
gOpenMol és una eina per a la visualització i l’anàlisi d’estructures moleculars i de les seves característiques químiques. El programa utilitza el motor d’escripting Tcl/Tk i pot ser ampliat fàcilment sense modificar codi del nucli. També es pot ampliar escrivint extensions utilitzant objectes compatibles (Linux/Unix) i els mòduls dinàmics d’intercanvi de dades (Windows).

gOpenMol s’utilitza per mostrar i analitzar:

- Estructures i característiques moleculars calculades amb programes externs
- Trajectòria molecular de la dinàmica
- Superfícies orbitals, moleculars i densitats d’electrons
- Tallar plans i fer animacions curtes
- Es pot usar amb potencials electrostàtics d’altres programes, com GaussianXX.
4.2. Factibilitat

Quan les necessitats bàsiques i la funcionalitat ja s’han definit s’ha de fer un anàlisi de factibilitat, aquest s’utilitza per ajudar a decidir al client si seguir o no endavant amb el projecte d’enginyeria del software, és per això que s’haurà de dur a terme durant tot el cicle de vida. Els anàlisis de factibilitat també determinen els riscos associats amb el projecte que han de ser efectuats si el projecte s’executa.

Existeixen tres tipus d’anàlisis:

- Factibilitat tècnica
- Factibilitat operativa
- Factibilitat econòmica

Anem a veure en detall cadascun d’ells:
4.1.1. Factibilitat tècnica

És un anàlisi tecnològic de tots aquells factors que justifiquen la millor combinació d'aquests per determinar la viabilitat del projecte.

Elecció del hardware

- Establir requeriments globals
- Establir la filosofia de processament
- Definir l’arquitectura
- Pautar un creixement per al mig o llarg plaç
- Pautar envergadura del processament
- Definir el grau de sofisticació tècnica

Elecció del software

La tecnologia en la qual es desenvolupa el projecte és Windows, que no és gratuïta, però que la majoria dels ordinadors que es venen en el mercat el porten incorporat. I les llibreries de DirectX 9 ja venen incorporades en els Windows que es distribueixen actualment. Per tant en un ordinador convencional que es ven en el mercat el projecte funcionaria correctament.

Aquesta tecnologia compleix uns paràmetres molt importants que ens permeten assegurar el funcionament correcte del sistema:

- Maduresa: L’existència del sistema operatiu Windows data des de molts anys enrere, no tant la de les llibreries DirectX 9, però és un llenguatge que s’utilitza en moltes ocasions, i cada vegada més, per a l’elaboració de jocs.
• Fàcil instal·lació: Avui en dia aquest sistema ja ve incorporat en els
ordinador que es venen en el mercat, per tant no ha de comportar
cap mena de problemàtica.

• Àmplia difusió: Com s’ha dit anteriorment, el Windows és un sistema
que està difós per tot el món. Qui no té un ordinador amb el aquest
sistema operatiu a casa?

La factibilitat tècnica obliga a estudiar si la incorporació de la nova
technologia al client és possible o pot presentar problemes. Però com
quan es va plantejar als inicis del projecte, el departament disposa d’un
ordinador amb el sistema operatiu necessari i les llibreries de Directx9
instal·lades. Per tant només serà necessari afegir l’aplicació i ja estarà
llesta per a funcionar, no farà falta instal·lar cap més llibreria ni cap
programa auxiliar.

És tant simple, que l’usuari final pot instal·lar el projecte i fer-lo
funcionar sense cap tipus de problema ni complicació alguna.
4.1.2. Factibilitat operativa

És l'evaluació de l'impacte del projecte sobre l'organització.

L'ús d'aquest projecte no implica canvis a nivell organitzatiu, ja que es tracta d'una eina nova que ofereix una funcionalitat extra respecte a les eines usades pel client final. Per tant no ha d’haver cap figura nova per a fer servir aquesta eina, sinó que es pot compaginar a les tasques actuals. A més amb l’ús d’aquesta eina es pretén agilitzar moltes de les tasques realitzades fins el moment en el client final.

L’únic rol i funció a definir és la de l’usuari final que usará aquesta eina, i com és molt senzilla d’usar aquest no ha de notar cap impacte important, tot el contrari, el que ha de notar és una millora en el rendiment laboral.
4.1.3. **Factibilitat econòmica**

L’estudi de la factibilitat econòmico-financera és l’eina imprescindible per a conèixer la totalitat de les despeses que sorgiran per a la incorporació del nou sistema, com també l’increment de les despeses per les càrregues d’estructura que demanarà el seu funcionament després de la implementació.

S’ha de considerar diferents tipus de costos:

- Despeses de desenvolupament del sistema
- Despeses d’operació del sistema
  - Costos fixes
  - Costos variables
- Despeses del sistema

**Despeses de desenvolupament del sistema**

En les despeses s’ha de tenir en compte els salaris, la capacitació, la compra, etc. Aquest elements no tenen gran importància al tractar-se d’un projecte final de carrera, ja que aquest no s’ha desenvolupat per obtenir uns beneficis econòmics.

S’hauria de tenir en compte el cost del sistema operatiu i de l’eina usada per al desenvolupament del projecte, però el sistema operatiu ja va integrat en l’ordinador destinat a l’ús del programa, i l’eina de desenvolupament al ser per a ús acadèmic no té despesa alguna.

El cost del hardware tampoc es tindrà en compte, ja que com s’ha dit anteriorment el departament ja disposa un ordinador amb aquestes característiques.
També s’ha de tenir en compte les despeses de departament/empresa de desenvolupament (edifici, personal administratiu, etc.) que de la mateixa manera que abans no farà falta tenir-los en compte.

Despeses d’operació del sistema
Les despeses d’operació del sistema són aquelles despeses que són contínues durant tot el cicle de vida del sistema.

Es divideixen entre fixes i variables:

- **Costos fixes:** Apareixen a intervals regulars i amb taxes generalment estables.

  Poden ser:
  - Lloguers i llicències de software
  - Salaris del personal
  - Assegurances
  - Serveis associats, com per exemple, l’acés a internet

  En aquest cas no s’haurà de tenir en compte cap punt, ja que tant salaris com assegurança no són necessaris i llicències i internet ja es disposa.

- **Costos variables:** Apareixen en proporció a un factor particular.

  Poden ser:
  - Paper, tinta, tóner, cd, etc.
  - Costos addicionals com energia, correu, equipament addicional, etc.
  - Ús d’ordinadors
En aquest cas només serà necessari un ordinador. Això comporta unes despeses com el manteniment o l’energia elèctrica.

**Despeses del sistema**

En aquest punt s’ha de fer referència a les despeses indirectes del projecte i les infraestructures a nivell de departament o d’edificis.

Un exemple seria el lloguer d’una oficina y les seves despeses que comporta. Com que el sistema s’implanta en un departament i unes oficines ja existents on ja controlen totes les despeses, aquestes tampoc s’han de tenir en compte.

**Beneficis tangibles**

Són els beneficis fàcils de quantificar, com ara:

- Disminució d’errors
- Increment de la rendibilitat
- ReduCCIó dels costos anteriors (fixes o variables)

**Beneficis intangibles**

Són els beneficis que en el moment de l’anàlisi no es poden quantificar, com ara:

- Satisfacció del client
- Publicitat
- Millora en a presa de decisions

Aquests beneficis s’explicaran més endavant ja que en el moment de fer l’anàlisi són valors totalment desconeguts.
Uns altres temes que es tracten en l’anàlisi de despeses/beneficis són si el sistema proposat és efectiu en relació a la despesa, si els beneficis poden superar les despeses. Però aquests temes no tenen gaire importància ja que la finalitat del projecte no és un benefici econòmic, sinó que és purament pràctic i educatiu.

Finalment s’ha d’analitzar la inversió davant la posició financera del client, establir els beneficis totals i analitzar el retorn de la inversió (VAN – Valor actualitzat net, TIR – Tasa interna de retorn).
5. Desenvolupament tècnic

Aquest apartat s’ha dividit en tres fases:

- Especificació
- Anàlisi
- Disseny
5.1. Especificació

5.1.1. Descripció general

El sistema ha de complir les pautes indicades en la definició d’objectius que s’ha fet a l’inici del projecte. En aquest cas, ha de tenir les següents funcionalitats:

- Comunicació i entrada de dades a partir d’aplicacions físiques del departament ja existents.
- Visualitzar un conjunt de molècules a través d’un sistema navegable, amigable i d’un aspecte elegant.
- Poder apropar o allunyar el sistema presentat per a una millor visualització o estudi.
- Modificar característiques de la visualització per a un millor tractament posterior, com per exemple, canviar el color de fons per a una millor impressió.
- Exportar fitxers d’imatge fàcilment en diferents formats i extensions.
- Visualitzar i extreure seqüències de vídeo d’animacions dels conjunts de molècules que puguin ser utilitzades en qualsevol sistema operatiu.

Després d’aquesta introducció sobre el que es pretén desenvolupar amb aquest projecte es necessari realitzar l’especificació dels requeriments del nostre sistema que es divideixen bàsicament en requeriments funcionals i requeriments no funcionals.
Definim els requeriments funcionals com aquells que descriuen les entrades i sortides del sistema i la relació que existeix entre ambedues, o el que és el mateix, són les necessitats o funcionalitats que ha d’assolir el sistema per a que funcioni de forma correcta.

Per una altra banda els requeriments no funcionals els definim com aquells que defineixen les qualitats que el sistema haurà de tenir com per exemple factors de qualitat, rendiment, seguretat, etc.

Per tal de determinar quins són els requeriments de cada sistema existeixen 4 estratègies: preguntar a l’usuari, extreure informació d’un sistema software anterior, sintetitzar-los a partir del funcionament de l’empresa i experimentar amb un prototipus i veure la reacció de l’usuari.
5.1.2. Requeriments funcionals

Per tal d’entendre quins son els requeriments funcionals del nostre sistema, primer de tot, hem d’entendre com serà el funcionament del mateix i quins actors hi participaran i quin paper tindran.

Descripció del funcionament del sistema
Per explicar el funcionament del sistema es farà desglossat per grups per a una millor explicació:

1. Visualització de molècules
   - **Interfície gràfica**
     La interfície gràfica ha de contenir una àrea de visualització del sistema físic, i uns panells (botons i desplegables) per a poder operar amb l’escena i modificar-la i poder importar i exportar resultats.

![Example of graphical interface](image-url)
• **Visualització de l’escena**

La visualització de l’escena es fa a partir d’unes APIs que conté el Directx ja que es va decidir fer l’entorn gràfic en un sistema Windows.

• **Transformacions geomètriques**

Sobre el sistema ha de ser possible moure’ns o rotar en qualsevol dels eixos de l’espai 3D, apropar-nos o allunyar-nos, així com traslladar-nos per l’escena. També hi ha l’opció de mostrar els eixos de coordenades per a poder veure en tot moment quina és l’orientació de l’escena.
• **Opcions de visualització**

A través del panell és possible mostrar tota l'escena a pantalla completa, escollir el color de fons i canviar el color dels àtoms (o si es tracta d'un sistema estel·lar, canviar el color dels estels).

Com es veurà a continuació, les propietats dels àtoms, colors i mida de les esferes es poden configurar abans d'iniciar l'execució mitjançant la configuració predeterminada del tipus d'escena, però també es pot modificar qualsevol d'aquestes característiques en qualsevol moment de la visualització.
• **Arxiu**

Des del panell és possible obrir arxius que contenen la informació per a la visualització del sistema de partícules, exportar informació una vegada visualitzada l'escena a fitxers d'imatge i creació de fitxers en format vídeo a partir de seqüències d'imatges.

Els arxius que contenen la informació són fitxers amb extensió .data, però la finestra modal que permet escollir el fitxer a obrir, permet escollir qualsevol tipus d'extensió.
Comunicació i protocol

Mitjançant les llibreries de Directx és possible visualitzar el sistema gràfic que segueix el protocol establert anteriorment. Ha estat inevitable un procés de coordinació entre el programa de representació existent i el projecte encarregat. El resultat ha estat la elaboració d'un protocol de comunicació entre els dos programes.

Concretament s'ha fet ús d'un fitxer que conté les coordenades del sistema de partícules i s'ha necessitat a creació de diferents configuracions segons el sistema que es vol visualitzar.

• Configuració

És la informació referent al tipus de sistema de partícules que es vol visualitzar, i segons el tipus de visualització ha de tenir uns paràmetres o uns altres.

Les dades de configuració són les següents:

- Número que identifica el tipus del sistema de partícules
  - 1 -> Aigua
  - 2 -> Lípid
  - 3 -> Sistema estel·lar

- Mida de les boles
  - 0.4 -> Petites
  - 0.6 -> Mitjanes
  - 0.8 -> Grans
  - 1.0 -> Gegants
• 1.2 -> Estel·lars

  o Color de les boles
    • Aigua -> Blanc i vermell
    • Lípid -> Groc, blau i blanc
    • Sistema estel·lar -> Verd, groc, vermell

  o Si s'ha de dibuixar les unions o no
    • Aigua -> Si
    • Lípid -> No
    • Sistema estel·lar -> No

  o Àtoms que formen la molècula
    • Aigua -> 3
    • Lípid -> 5
    • Sistema estel·lar -> 1

• **Fitzar de sistemes de partícules**

El fitxer de sistemes de partícules sempre té la mateixa estructura, sigui quin sigui el tipus de visualització que es vol obtenir.

L'estructura global del sistema físic és la següent:

➢ Posició de l'àtom 1
➢ Posició de l'àtom 2
➢ Etc ...
➢ Posició de l'àtom n
➢ Mida de la capsà contenidora
Un exemple de fitxer format per només tres àtoms seria el següent:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2034</td>
<td>13.4241</td>
<td>11.3684</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.2588</td>
<td>13.1021</td>
<td>11.4511</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.1040</td>
<td>14.3542</td>
<td>10.9956</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.9458</td>
<td>1.5636</td>
<td>7.5909</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.2894</td>
<td>2.3237</td>
<td>7.9904</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.3430</td>
<td>1.1279</td>
<td>8.4438</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.1729</td>
<td>16.8421</td>
<td>13.8441</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.0636</td>
<td>17.2983</td>
<td>14.7501</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.3244</td>
<td>16.3289</td>
<td>13.6717</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.62603010000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exportació d’imatges

Com ja s’ha indicat, des del programa es pot obrir arxius amb
extensió .data que contenen l’estat del sistema de partícules. A més,
s’haurà de poder exportar imatges a arxius amb diferents formats d’imatge.

Els diferents formats d’imatge que permet exportar són:

- **BMP**  Windows bitmap format
- **JPG**  Estàndard JPEG
- **GIF**  Graphics Interchange Format
- **TGA**  Targa o Truevision Graphics Adapter Format
- **PNG**  PNG format
- **DDS**  Direct Draw surface file format. Emmagatzema textures
- **PPM**  Portable Pixmap format
- **DIB**  Windows DIB
- **HDR**  HDR format
- **PFM**  Portable float map format
Generació i exportació de vídeos

L'extracció de seqüències de vídeo, és una opció que se li ha donat molta importància en el projecte, d'aquí ve el nom de "connexió dinàmica".

L'objectiu és poder visualitzar una seqüència d'imatges enllaçades de tal manera que es vegin consecutives en la visualització del programa i que paral·lelament es generi un fitxer d'extensió .avi per a ser usat i exportat a qualsevol sistema operatiu.

És un procés bastant complex, però el seu ús es fa de la manera més senzilla possible, el que es pretén és que hi hagi una opció (botó) que permetés indicar quins fitxers es volen introduir a la seqüència.

Així no faria falta ni indicar el número de fotogrames a mostrar, només marcant els fitxers desitjats el programa ha d'anar llegir tots els fitxers del directori i generant el fitxer de vídeo.
**Llistat de funcionalitats del sistema**

Modificació dels paràmetres de funcionament: Es pot modificar la visió de l'escena ja que és possible moure'ns o rotar en qualsevol dels eixos de l'espai 3D, apropar-nos, allunyar-nos o traslladar-nos. També es pot mostrar els eixos de coordenades per a poder veure en tot moment quina és l'orientació de l'escena.

També es pot mostrat tota l'escena a pantalla completa, escollir el color de fons i canviar el color dels àtoms o estels, modificar la mida de les esferes, tant abans d'iniciar l'execució com en qualsevol moment de la visualització.
5.1.3. Requeriments no funcionals

Els requeriments no funcionals són aquells que defineixen les qualitats generals que ha de tenir el sistema al realitzar la seva funció. És a dir, aspectes com la qualitat, la plataforma d’explotació, plataforma de desenvolupament, seguretat, perfil de l’usuari, rendiment extern, modificabilitat, eficiència, portabilitat, etc.

Qualitat

Segons la terminologia de la IEEE, la qualitat d’un sistema s’obté en funció del compliment dels requeriments inicals específicats pel client o usuari final. Les especificacions de la qualitat d’un producte de software han estat objecte de treball de varis grups d’investigació dels quals un dels més destacats és el model de McCall. Aquest model estableix tres àrees principals que intervenen en la qualitat del software:

- Qualitat en l’operació del producte: Requereix que el software pugui ser entès fàcilment, que operi eficientment i que els resultats obtinguts siguin els requerits inicialment per l’usuari.
- Revisió de la qualitat del producte de software: Té com a objectiu realitzar revisions durant el procés de desenvolupament per a detectar els errors que afecten a l’operació del producte.
- Qualitat en el procés: Requereix de la definició d’estàndards i procediments que serveixen de base per al desenvolupament del software.
Una autre model important és el de Boehm. Aquest model destaca per ser un dels millors definits. El model és de naturalesa jeràrquica i els criteris de qualitat es presenten en tres grans divisions Portabilitat, Usabilitat i Mantenibilitat.

Model de Boehm per a Classificar els Criteris de Qualitat

El projecte s’ha dut a terme seguint els principals atributs de qualitat del software i ara es comentaran a continuació cadascun d’ells:
Plataforma de desenvolupament

En aquest apartat es descriuen quins elements del hardware i el software són necessaris per desenvolupament de l’aplicació.

El projecte s’ha desenvolupat en dos ordinadors diferents, encara que les proves s’han realitzat en tres de diferents.

Anem a veure, però els dos equip usats per a la implementació del projecte:

➤ Hardware

<table>
<thead>
<tr>
<th></th>
<th>Sobretaula</th>
<th>Portàtil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processador:</td>
<td>Pentium 4</td>
<td>Pentium 4</td>
</tr>
<tr>
<td>Memòria</td>
<td>1Gb</td>
<td>512 Mb</td>
</tr>
<tr>
<td>Tarjeta gràfica</td>
<td>32Mb</td>
<td>64Mb</td>
</tr>
<tr>
<td>Disc dur</td>
<td>200 Gb</td>
<td>80Gb</td>
</tr>
</tbody>
</table>

➤ Software

<table>
<thead>
<tr>
<th></th>
<th>Windows XP</th>
<th>Windows Vista</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO</td>
<td>DirectX9</td>
<td>DirectX9</td>
</tr>
<tr>
<td>Llibreries:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S’han necessitat les llibreries del llenguatge de programació utilitzat per a desenvolupar el sistema:

➤ Visual C++

A més s’han fet servir per a desenvolupar vídeo diferents llibreries auxiliars que s’han afegit al sistema:

➤ AviFile.h
➤ DxToMovie.h
➤ RenderTarget.h
També s’han estudiat diferents programes per a l’estudi d’antecedents, previs a la implementació del sistema:

- gOpenMol
- Molden
- Rasmol

També s’han necessitat diferents editors de texte, tant per al desenvolupament del sistema com per a la realització de la memòria final, i també per a la lectura de diferents manuals en format electrònic:

- WordPad
- UltraEdit
- Microsoft Word
- Acrobat reader (pdf)
- Microsoft project
- Pacestar UML Diagrammer
- Microsoft PowerPoint

Per a la visió d’imatges creades pel programa i configuració d’imatges per a la documentació s’ha necessitat diferents programes:

- ACDSee 7.0
- Paint

Per a visualitzar el vídeo creat per l’aplicació s’han fet servir el següents programes:

- Reproductor de windows media
- Windows Media Player

Per a buscar tota la informació online necessària per al desenvolupament de l’aplicació s’han fet servir el següents navegadors:

- Internet Explorer
Perfil de l’usuari

L’aplicació està dirigida a professors del departament de Física i Enginyeria Nuclear (FEN) de la Universitat Politècnica de Catalunya (UPC).

Aquest programa s’ha realitzat a mida per al departament, ja que requereixen un software científic a mida que els permeti la visualització de sistemes de partícules en general.

No són necessaris coneixements previs per a la utilització del programa ja que el significat semàntic del problema a visualitzar és independent del programa, i per tant se li pot donar qualsevol ús.