

UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria Tècnica d'Informàtica de Sistemes

APLICACIÓ DE CALCULADORA SOLAR

Alumne/a: Belén Cruz García

Dirigit per: Ramon Català Pou

CURS 2012-13 (Febrer/Setembre)



Aquest treball està subjecte –excepte que s'indiqui el contrari– en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de *Creative Commons*. Podeu copiar-lo, distribuir-lo i transmetre'ls públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada. La llicència completa es pot consultar en

<http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>

*Voldria dedicar aquest treball fi de carrera als dos estels
que m'il·luminen amb més força que fins i tot la mateixa llum
del Sol. Que res ni ningú apagui la seva llum.*

RESUM

El present treball fi de carrera consisteix en la realització de l'estudi, disseny i implementació d'una aplicació informàtica dins la temàtica dels sistemes d'informació geogràfica (SIG) per a realitzar el càlcul de la quantitat de llum solar rebuda pels diferents edificis d'una ciutat. L'aplicació s'ha desenvolupat sota l'entorn de GeoMedia, una plataforma comercial per a tractar dades SIG, i ha estat programada en Visual Basic fent servir l'entorn Visual Studio 2012.

Les funcionalitats de l'aplicació són principalment tres. La primera consisteix en la generació dels polígons 3D dels edificis a partir d'una alçada donada i dels polígons 2D obtinguts a partir de mapes topogràfics descarregats de l'Institut Cartogràfic de Catalunya. La segona funcionalitat calcula el percentatge de llum solar sobre la coberta i la façana d'un edifici. Per a fer aquesta avaluació s'ha calculat si els edificis circumdants fan o no ombra a una sèrie de punts distribuïts sobre la coberta i la façana. Finalment, l'última funcionalitat és el càlcul d'hores de llum sobre la coberta i els diferents pisos d'un edifici per un període de temps donat, on també cal avaluar si els punts reben o no la llum directa del sol. Per a realitzar l'avaluació de llum solar s'han estudiat i implementat dos algorismes diferents: traçat de rajos (*ray tracing*) i proporció de triangles. El primer és més precís, mentre que el segon és més eficient en termes de càlcul computacional. La implementació de l'aplicació s'ha fet en Visual Basic fent servir interfícies de GeoMedia, on s'ha fet servir programació orientada a objectes explotant herència i polimorfisme. L'aplicació s'ha implementat com a comanda personalitzada (*custom command*) per a poder cridar-la des de dins de GeoMedia i així estendre les seves funcionalitats SIG i permetre avaluacions de grans superfícies amb una interfície senzilla.

Com a cas d'exemple per a depurar i validar l'aplicació s'ha escollit el municipi de Sant Esteve Sesrovires, situat a la comarca del Baix Llobregat. La capa amb la informació dels edificis s'ha descarregat de l'Institut Cartogràfic de Catalunya (fitxer *SHP* o de formes). Partint dels polígons 2D dels edificis, s'han afegit les alçades que, donat de no disposar de la informació real d'aquestes, s'ha optat per afegir-les aleatòriament. D'igual manera, el polígon corresponent a la façana s'ha escollit a l'atzar. Les altres dues funcionalitats, el càlcul del percentatge solar i el número d'hores totals, també s'han validat. A més, s'ha fet una comparativa dels dos algorismes implementats pel càlcul de llum solar (*ray tracing* i proporció de triangles), tant a nivell dels resultats obtinguts amb cadascun com de cost computacional. Els resultats obtinguts mostren la validesa de les funcionalitats implementades.

ÍNDEX

Índex de Figures	7
Índex de Taules	9
Taula d'Acrònims	9
1. Introducció	10
1.1 Marc del projecte	10
1.2 Objectius del TFC	10
1.3 Pla de treball del TFC	11
1.4 Sistemes d'informació geogràfica	12
1.4.1 Definició de sistema d'informació geogràfica	12
1.4.2 Funcions dels sistemes d'informació geogràfica	13
1.5 Programari Utilitzat	14
1.5.1 GeoMediaProfessional i GeoMedia 3D	14
1.5.2 Microsoft Visual Studio Express 2012	18
2. Anàlisi funcional de l'aplicació	20
2.1 Definició de casos d'ús	20
2.2 Model de dades	22
2.3 Definició dels algorismes	23
2.3.1 Generació dels polígons 3D	24
2.3.2 Càlcul de la posició del sol	25
2.3.3 Algorisme de <i>ray tracing</i>	26
2.3.4 Algorisme per proporció de triangles	28
2.3.5 Càlcul del percentatge de llum i del número total d'hores	30
2.4 Definició de la interfície	34
3. Disseny de l'Aplicació	35
3.1 Diagrama de Classes	35
3.1.1 Classes <i>CalculSolar</i> , <i>PercentatgeSolar</i> i <i>HoresTotals</i>	36
3.1.2 Classe <i>LlumSolar</i>	38
3.1.3 Classe <i>AfegirAlçades</i>	39
3.1.4 Classe <i>MyProgressBar</i>	39
3.1.5 Llibreria d'utilitats	40
3.2 Pla de Proves	40

4. Construcció del Sistema	41
4.1 Programant amb Visual Basic i GeoMedia	41
4.2 Consideracions sobre GeoMedia	43
4.3 Manual d'usuari de la Calculadora Solar	45
5. Resultats.....	49
5.1 Afegir alçades	49
5.2 Percentatge solar	49
5.3 Hores totals	52
5.4 Comparativa de Temps	54
6. Conclusions i Línies Futures	56
Bibliografia.....	58

ÍNDEX DE FIGURES

Figura 1. Estructura de la descomposició del treball del TFC.....	12
Figura 2. Pantalla per a la configuració del sistema de coordenades.....	16
Figura 3. Definició d'un arxiu de configuració de magatzem.	16
Figura 4. Definició d'una connexió nova.....	17
Figura 5. Importació d'una ortofoto dins la classe entitat <i>Ortofoto</i> creada anteriorment.	17
Figura 6. Pantalla principal de GeoMedia Professional on es poden veure les diferents capes importades. En aquest cas s'han desactivat alguns elements per a la visualització. L'ortofoto es mostra de fons amb una transparència del 50%.	18
Figura 7. Importació dels objectes 3D mitjançant la utilitat disponible a GeoMedia Professional.....	19
Figura 8. Visualització dels edificis importats del municipi de Sant Esteve Sesrovires amb Montserrat de fons. L'alçada d'extrusió s'ha fixat en aquest cas a 10 metres, ja que en produir-se un error en importar el fitxer KMZ, aparentment les alçades dels edificis no s'han guardat correctament a la base de dades.	19
Figura 9. Diagrama de casos d'ús de l'aplicació desenvolupada.....	21
Figura 10. Representació dels angles d'azimut i altitud del sol calculats respecte a l'observador. Font: [13].	26
Figura 11. Esquema de l'algorisme de <i>ray tracing</i> per a avaluar si l'edifici interferent (vermell) dona ombra o no sobre un punt donat de l'edifici a avaluar (blau).	27
Figura 12. Esquema de l'algorisme fent servir proporció de triangles per a avaluar si l'edifici interferent (vermell) dona ombra o no sobre un punt donat de l'edifici a avaluar (blau).	29
Figura 13. Esquema de l'algorisme per a saber si un edifici interferent (edifici vermell) es troba sobre la recta que uneix el sol i un punt sobre l'edifici d'interès (edifici blau). La figura de l'esquerra mostra una situació en la que l'edifici avaluat no interfereix, mentre que a la figura de la dreta sí succeeix.	30
Figura 14. Exemple on per a l'algorisme proporció de triangles l'edifici interfereix per molt poc i, per tant, a la pràctica és com si l'edifici fos completament entre el sol i el punt (rectangle verd).	30
Figura 15. Diagrama de la classe abstracta <i>CalculSolar</i> i les sub-classes corresponents als dos algorismes principals.....	35
Figura 16. Diagrama de la classe abstracta <i>LlumSolar</i> i les sub-classes corresponents.	38
Figura 17. Diagrama de la classe <i>AfegirAlcades</i>	39
Figura 18. Diagrama de la classe <i>BarraProgres</i>	39
Figura 19. Llibreries importades de GeoMedia dins l'aplicació.	41

Figura 20. Explorador de solucions de l'aplicació implementada.....	42
Figura 21. Disseny de l'aplicació per a les diferents funcionalitats. (a) Per a l'opció d'afegir alçada als polígons 2D. (b) Per a calcular el percentatge de llum solar sobre els edificis generats a l'opció anterior. (c) Per a calcular el número total d'hores d'un rang de dates donades.	46
Figura 22. Propietats d'estil per a poder visualitzar els polígons creats en 3D.	47
Figura 23. Configuració per a la visualització de la informació en forma de rang de dades. En aquest cas es tracta del percentatge solar i els valors van de 0% a 100% en increments del 10%.	48
Figura 24. Exemple de com filtrar els registres d'una taula per a poder visualitzar la informació desitjada. L'exemple està filtrant només les hores totals de la coberta.	48
Figura 25. Visualització dels polígons 3D obtinguts amb la funcionalitat d'afegir alçades al municipi de Sant Esteve Sesrovires. Les alçades s'han afegit aleatòriament amb valors que van de 5 a 30 metres. S'han afegit textures als costats i a la coberta. L'ortofoto amb 25 cm de resolució descarregada de l'ICC es mostra a sota.	49
Figura 26. Posició del sol el dia 1 de juny a les 17:00 UT (19:00 hora local). L'azimut és 280° i l'angle d'elevació són 23°. Les línies groga i vermella indiquen l'ortus i l'ocàs, respectivament. Font: [14].....	50
Figura 27. Percentatge de llum solar sobre la coberta al municipi de Sant Esteve Sesrovires a les 19:00 hores locals el dia 1 de juny de 2013. Calculat amb (amunt) <i>ray tracing</i> i (avall) proporció de triangles. La direcció d'observació de la càmera correspon aproximadament a la del sol.	51
Figura 28. Percentatge de llum solar sobre la façana dels edificis al municipi de Sant Esteve Sesrovires a les 19:00 hores locals el dia 1 de juny de 2013. Els rajos de sol venen aproximadament de la part inferior. Càlcul realitzat amb <i>ray tracing</i> . Les façanes de cada edifici estan indicades amb una línia negra sòlida. L'ortofoto es mostra a sota amb transparència.	51
Figura 29. Número d'hores totals sobre la coberta pel dia 1 de juny de 2013 al municipi de Sant Esteve Sesrovires. Calculat amb (amunt) <i>ray tracing</i> i (avall) proporció de triangles. Càmera orientada de nord a sud (nord cap avall).	52
Figura 30. Número d'hores totals sobre (amunt) planta baixa i (mig) tercera planta pel dia 1 de juny de 2013 al municipi de Sant Esteve Sesrovires. (Avall) Representació de les façanes (polígons negres). Càlcul realitzat amb <i>ray tracing</i>	53

ÍNDEX DE TAULES

Taula 1. Fites del TFC definides durant el pla de treball.	12
Taula 2. Comparativa del cost computacional pels dos algorismes de càlcul de llum solar implementats amb diferents opcions de configuració.	54

TAULA D'ACRÒNIMS

CUDA	<i>Compute Unified Device Architecture</i>
EDT	Estructura de la Descomposició del Treball
GPU	<i>Graphic Processor Unit</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
ICC	Institut Cartogràfic de Catalunya
IG	Informació Geogràfica
KML	<i>Keyhole Markup Language</i>
KMZ	<i>Keyhole Markup Language Zipped</i>
LST	<i>Local Sideral Time</i>
MrSID	<i>Multiresolution Seamless Image Database</i>
OLE	<i>Object Linking and Embedding</i>
OpenCL	<i>Open Computing Language</i>
SIG	Sistema d'Informació Geogràfica
TFC	Treball Fi de Carrera
UOC	Universitat Oberta de Catalunya
UT	<i>Universal Time</i>
UTM	<i>Universal Transverse Mercator</i>

1. INTRODUCCIÓ

1.1 Marc del projecte

Aquest treball fi de carrera (TFC) s'emmarca dins l'Enginyeria Tècnica d'Informàtica de Sistemes cursada a la Universitat Oberta de Catalunya (UOC), i s'ha desenvolupat dins l'àrea temàtica dels sistemes d'informació geogràfica (SIG).

1.2 Objectius del TFC

El projecte consisteix en l'estudi, disseny i implementació d'una aplicació informàtica per a realitzar diferents càlculs relacionats amb la quantitat de llum solar que reben els edificis d'una ciutat. Aquesta eina SIG té l'objectiu d'ajudar en el disseny i planificació de noves construccions per minimitzar el consum de recursos energètics, com per exemple, l'obtenció de la localització i orientació òptima de panells solars sobre els edificis per maximitzar-ne l'eficiència.

Les principals funcionalitats a implementar són les següents:

- Càlcul dels polígons 3D dels edificis a partir dels polígons 2D i les seves alçades. Aquesta funcionalitat també indica quin dels polígons correspon a la façana.
- Càlcul del percentatge de llum rebuda sobre la coberta i la façana dels edificis per a una hora i dia específics. Les dades d'entrada són els polígons 3D anteriorment generats.
- Càlcul del número total d'hores sobre la coberta i els diferents pisos de la façana per a un període de temps determinat. Les dades d'entrada són de nou els polígons 3D dels edificis.

Per a realitzar els càlculs relacionats amb la llum solar s'han implementat diversos algorismes. En primer lloc, s'han utilitzat algorismes existents posats a disposició dins l'assignatura per a calcular la posició del sol per a unes coordenades i dia i hora donats, incloent el càlcul de les hores de l'ortus i l'ocàs. A Internet es poden trobar aquestes aplicacions sota el nom de calculadores solars. El segon tipus d'algorismes tracten el problema de saber si un punt rep la llum directa del sol, on cal avaluar si els edificis circumdants bloquegen o no el raig de sol. Comunament es fan servir els anomenats algorismes de *ray tracing*, o traçat de rajos, per a resoldre aquest tipus de geometries. En aquest TFC s'ha implementat un algorisme senzill de *ray tracing* basat en la intersecció d'una recta amb un pla. També s'ha implementat un segon algorisme basat en la proporció del costats d'un triangle. D'aquesta manera s'ha pogut fer una comparativa dels dos algorismes tant a nivell de resultats com de temps de càlcul.

El desenvolupament de l'aplicació s'ha dut a terme seguint les passes típiques de l'enginyeria de programari: estudi, disseny i implementació. La part de l'estudi ha avaluat primer els requeriments de les funcionalitats per a després definir els casos d'ús, el model de dades i els algorismes abans esmentats. L'etapa de disseny ha definit el model de classes, les quals s'han especificat amb l'objectiu de generar un codi eficient i reutilitzable. Finalment, l'etapa d'implementació i depuració s'ha realitzat

en Visual Basic amb funcionalitats de GeoMedia, de manera que el producte final és una aplicació del tipus comanda personalitzada (*custom command*) que és cridada des del mateix GeoMedia.

L'aplicació i les seves funcionalitats s'han validat amb el municipi de Sant Esteve Sesrovires com a cas d'exemple, on els polígons 2D dels edificis s'han descarregat de l'Institut Cartogràfic de Catalunya (ICC) [6]. També destacar que la visualització de la informació obtinguda s'ha fet mitjançant una representació temàtica de rangs utilitzant GeoMedia 3D.

Aquest document està dividit en sis capítols. A continuació en aquest mateix capítol s'exposen el pla de treball, així com els fonaments dels sistemes d'informació geogràfica i el programari utilitzat durant la realització del TFC. El segon capítol aborda l'anàlisi funcional de l'aplicació, que inclou la definició dels casos d'ús, el model de dades, l'estudi i definició dels algorismes a implementar, i la interfície del programari. El tercer capítol presenta el disseny de l'aplicació, on principalment es mostren els diagrames de classe i la implementació final dels algorismes que duen a terme les funcionalitats principals del percentatge solar i el número d'hores totals. El capítol també inclou una descripció del pla de proves que s'ha executat per validar la correcta implementació dels algorismes. El quart capítol se centra en la construcció del sistema, principalment en els detalls sobre la implementació de la comanda personalitzada i la programació sota GeoMedia. També s'inclou una secció amb el manual d'usuari. El capítol cinquè presenta els resultats de l'aplicació implementada fent servir com a cas d'exemple el municipi de Sant Esteve Sesrovires, a la comarca del Baix Llobregat. S'exposen i s'analitzen els resultats de les tres funcionalitats implementades, així com una comparativa entre l'algorisme de *ray tracing* i proporció de triangles. Finalment, el sisè i últim capítol presenta les conclusions del projecte i es suggereixen noves línies de treball futures.

1.3 Pla de treball del TFC

A l'inici del TFC es va relitzar un pla de treball amb l'objectiu de planificar les diferents etapes del projecte dins el termini de tres mesos i mig. Aquest exercici és, en part, un dels objectius d'aprenentatge associats al TFC, donat que la preparació d'un pla de treball és una tasca habitual a l'hora de fer una proposta d'un projecte per a d'altres empreses o institucions [1]. A continuació es mostra un breu resum del pla de treball que es va realitzar.

En primer lloc es presenten les fites definides en el TFC, i que són les marcades al calendari de l'aula com a PACs, més la memòria final i el debat. La Taula 1 presenta les fites juntament amb les dates associades a les dates d'entrega dels esborranys.

Al pla de treball també es va incloure una estructura de la descomposició del treball (EDT) de les diferents tasques i sub-tasques, que es presenta a la Figura 1 i permet tenir una visió global del projecte. Aquest s'ha dividit en cinc grups principals de tasques, dels quals els primers quatre es poden associar directament als capítols d'aquest TFC. Així doncs, la tasca T100 i les seves sub-tasques corresponen a la introducció, la tasca T200 al capítol 2, la T300 al capítol 3, i la T400 al capítol 4. D'altra banda, els noms de les tasques ja donen una idea de les activitats que inclouen, però

en qualsevol cas associat a l'EDT normalment també s'inclouen altres dades de les tasques i sub-tasques, com la data d'inici i duració, la informació d'entrada requerida, les activitats a realitzar i els resultats esperats. Aquesta informació addicional no s'inclou en aquest document final per motius d'espai.

Fita	Data
Inici TFC	13/03/2013
Esborrany Pac 2	03/04/2013
Lliurament Pac 2	09/04/2013
Esborrany Pac 3	08/05/2013
Lliurament Pac 3	14/05/2013
Lliurament esborrany presentació i memòria	03/06/2013
Lliurament presentació i memòria	10/06/2013
Inici debat	26/06/2013
Fi debat	28/06/2013
Fi TFC	28/06/2013

Taula 1. Fites del TFC definides durant el pla de treball.

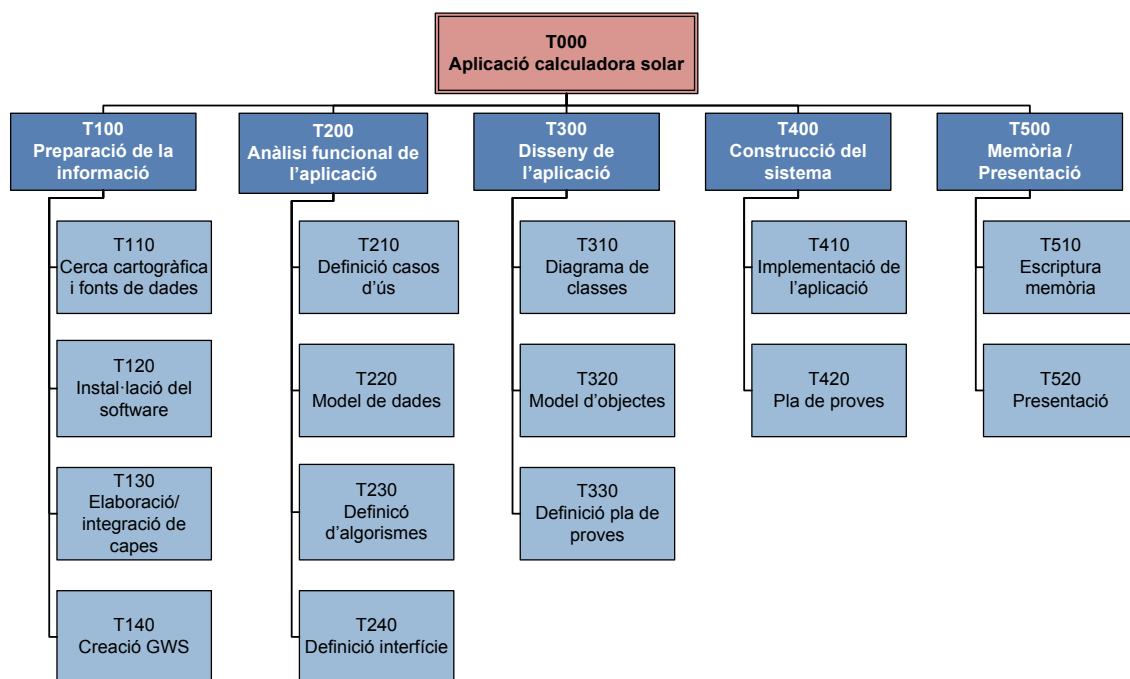


Figura 1. Estructura de la descomposició del treball del TFC.

Finalment, el pla de treball també va incloure un esborrany de l'estructura final del TFC, el diagrama de Gantt del projecte, i un pla d'incidències i riscos, els quals, de nou, no s'inclouen aquí per limitacions d'espai.

1.4 Sistemes d'informació geogràfica

En aquesta secció es presenten breument els principals conceptes relacionats amb els sistemes d'informació geogràfica (SIG).

1.4.1 Definició de sistema d'informació geogràfica

Per a definir què és un SIG, primerament es definirà quin és el concepte d'informació geogràfica (IG). Aquesta es pot definir com la informació sobre un element localitzat en

algun punt de la Terra [19]. Amb d'altres paraules, la informació o propietats sobre un objecte o entitat té associada la seva localització espacial. El tipus de localització té les següents propietats:

- És multidimensional, per exemple, fent servir coordenades Cartesianes o en latitud/longitud.
- Pot tenir diferents resolucions, de centímetres a quilòmetres.
- Pot estar associada a un volum.
- Es pot representar amb diferents formats digitals (binari, ASCII, etc.).
- S'acostuma a projectar, comunament sobre una superfície plana.

Quant al concepte *sistema*, es pot definir com el conjunt de persones, màquines o altres mètodes que permeten tractar (emmagatzemar, processar, transmetre, rebre, gestionar, distribuir, etc.) dades, les quals representen la informació d'utilitat [19].

Per tant, un sistema d'informació geogràfica, és un sistema que tracta –en el sentit més ampli– informació que té associada localització espacial. Donada l'enorme quantiat de dades a tractar pels SIG, la informació acostuma a guardar-se en magatzems de dades (*data warehouse* en anglès), compostes per bases de dades estructurades amb l'objectiu d'un ràpid accés a la informació.

1.4.2 Funcions dels sistemes d'informació geogràfica

Les funcions dels SIG segueixen un model de fluxe de la informació on les principals funcions són [19]:

- L'entrada de dades.
- El tractament de les dades d'entrada.
- La sortida de les dades.

Les dades d'entrada necessiten ser adquirides d'alguna manera, ja sigui perquè un individu o una entitat les ha generades ell mateix per al seu propi ús (*dades primàries*), o a través de distribuïdors externs (*dades secundàries*). Gràcies al gran nombre de bases de dades digitals accessibles, les dades secundàries es poden aconseguir de manera senzilla i poc costosa. D'altra banda, cal considerar els diferents formats de les dades digitals, tot i que hi ha una tendència a unificar formats i seguir estàndards per facilitar precisament l'ús d'aquest tipus de dades. Les maneres d'obtenir les dades d'entrada són molt diverses i depenen de la naturalesa de les dades a mesurar. D'altra banda, les dades espacials acostumen a adquirir-se mitjançant aparells mòbils equipats amb receptors GPS¹.

És interessant avaluar el paper de les dades en el marc de l'actual TFC. D'una banda, les dades secundàries són els polígons 2D i les alçades dels edificis, informació que s'ha descarregat de l'Institut Cartogràfic de Catalunya. D'altra banda, no hi ha dades primàries en sí que s'hagin adquirit, però la posició del sol calculada a través d'algorismes existents podria jugar aquest paper (veure Secció 2.3.2).

¹ *Global Positioning System*.

El segon pas dins de la cadena de processat de dades SIG es troba en el tractament de les dades, que inclou principalment l'emmagatzament, la recuperació, l'anàlisi i la consulta de les mateixes. Les dades prèviament adquirides s'han d'emmagatzemar, tant la informació geoespacial (geometria) com la informació associada (atributs). D'aquesta manera un usuari pot accedir-hi sempre que les necessiti. L'anàlisi i consulta de les dades es fa comunament mitjançant programari SIG, com per exemple GeoMedia (veure la Secció 1.5.1), i permeten a l'usuari el tractament, visualització i manipulació de les dades, i en última instància transformar-les en informació.

En el marc del TFC, l'aplicació que s'ha implementat juga el paper de l'anàlisi de les dades, on a partir de la informació d'entrada (polígons 2D i alçades dels edificis), es calculen una sèrie de paràmetres associats a la quantitat de llum rebuda sobre la coberta i la façana dels edificis, convertint aquesta informació en les dades de sortida.

I són precisament aquestes dades de sortida el producte final de la funcionalitat d'un SIG, i esdevenen, per tant, el seu producte principal. Tant la presentació com la visualització de les dades de sortida depenen de nou de la naturalesa de les dades obtingudes. Els mitjans més comuns per la visualització són la pantalla de l'ordinador, un projector, un plòter o una impresora. D'altra banda, la forma de presentació té moltes possibilitats, tot i que sempre caldrà buscar, més que l'espectacularitat, la utilitat, la funcionalitat i la facilitat per a la interpretació de les dades per part de l'usuari.

En el marc del TFC, la sortida de les dades es fa mitjançant una representació temàtica de rangs dins del mateix GeoMedia 3D, on a cada edifici se li associa un color en funció de la quantitat mesurada, per exemple, percentatge solar o número d'hores totals, i on a més l'edifici es representa en tres dimensions.

1.5 Programari Utilitzat

Per a la realització d'aquest TFC s'ha fet servir principalment GeoMedia® Professional [2]² i la seva extensió per a 3D, GeoMedia 3D, més l'entorn de programació Microsoft Visual Studio Express 2012 [3]³, que s'ha utilitzat per a implementar l'aplicació SIG en Visual Basic. Per a la definició dels casos d'ús i els diagrames de classe s'ha fet servir l'aplicació MagicDraw UML [4]⁴. A més a més, durant les fases d'implementació i optimització del codi també s'ha fet ús de Microsoft Access 2010 [5]⁵ per a modificar les taules dins la base de dades directament. A continuació es descriuen les dues principals aplicacions.

1.5.1 GeoMediaProfessional i GeoMedia 3D

GeoMedia Professional és una eina que permet la presentació i l'anàlisi de dades originals de diferents bases de dades en un mateix entorn SIG. Un dels grans avantatges de GeoMedia Professional és que les dades, que poden tenir diferents

² <http://geospatial.intergraph.com/products/GeoMedia/Details.aspx>

³ <http://www.microsoft.com/visualstudio>

⁴ <http://www.nomagic.com/products/magicdraw.html>

⁵ <http://office.microsoft.com>

formats, orígens de coordenades o projeccions cartogràfiques, no s'emmagatzemen en un format intern dins de GeoMedia, sinó que aquest accedeix directament a les dades de forma simultània, evitant així una duplicació d'aquestes. A més a més, GeoMedia Professional també és capaç de capturar i editar dades espacials. GeoMedia 3D és un conjunt d'eines que s'integren dins de l'aplicació GeoMedia Professional que permeten la presentació i l'anàlisi de les dades en tres dimensions.

Després de la instal·lació de GeoMedia Professional i GeoMedia 3D, es procedeix a la descàrrega de les dades cartogràfiques dels servidors de l'Institut Cartogràfic de Catalunya [6]⁶. La fulla seleccionada per a la realització del TFC és la 282-121 corresponent a una part del municipi de Sant Esteve Sesrovires a escala 1:5000 i amb el sistema geodèsic de referència ETRS89. La base cartogràfica escollida conté dues capes, la primera en format vectorial amb extensió SHP corresponent a *Esri Shapefile* i la segona en format ràster georeferenciat amb extensió SID corresponent a *Multiresolution Seamless Image Database* (MrSID), en el present cas, una ortofoto. A fi de disposar de l'atribut alçada del l'edifici també s'obté el fitxer del model tridimensional de les edificacions i les construccions en format KMZ (*Keyhole Markup Language Zipped*) corresponent a arxius comprimits del format KML (*Keyhole Markup Language*).

Seguidament es crea un nou espai de treball GeoMedia (*GeoWorkspace*) i un nou fitxer de sistema de coordenades del projecte que relaciona una coordenada geogràfica amb una coordenada del model a l'espai d'emmagatzament SIG (veure Figura 2). En el nostre cas i segons les Especificacions Tècniques de la base topogràfica de Catalunya 1:5000 (BT - 5M) v2.0 [7], el nostre fitxer del sistema de coordenades s'ha de configurar amb les següents dades:

- Sistema geodèsic de referència ETRS89, amb les coordenades geodèsiques positives al nord de l'Equador per la latitud i a l'est del meridià de Greenwich per a la longitud.
- Sistema de referència altimètric corresponent als registres del nivell mig del mar a Alacant.
- Sistema de representació plana corresponent a la projecció conforme *Universal Transversa de Mercator* (UTM) del fus 31, amb l'ordre de coordenades *Easting* (X), *Northing* (Y).

⁶ <http://www.icc.cat>

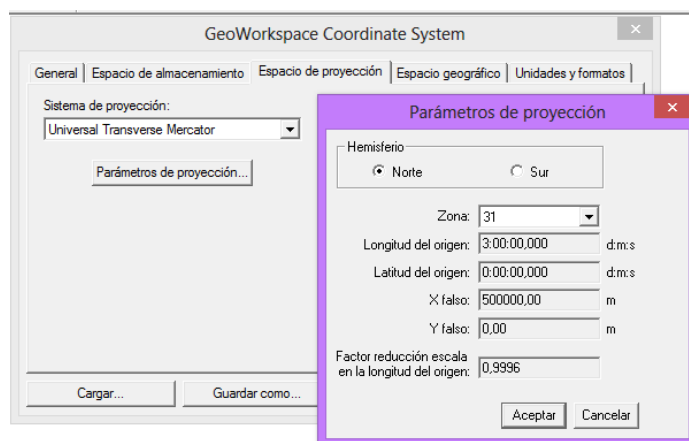


Figura 2. Pantalla per a la configuració del sistema de coordenades.

A continuació cal crear una base de dades on s'emmagatzemaran les dades que es generaran dins GeoMedia. Amb aquesta finalitat es crea un magatzem del tipus "Accés lectura i escriptura" amb l'opció de menú *Magatzem (Almacén)*. En el present cas, s'ha creat una base de dades Microsoft Acces a partir d'una plantilla pròpia de GeoMedia.

Després es genera un nou arxiu de configuració d'un magatzem per a la connexió ArcView que es crea posteriorment a aquest pas i que serveix per a incorporar la informació que pertany a la capa topogràfica vectorial, *ESRI Shapefile*. Aquest arxiu amb extensió INI facilita la configuracions de diferents magatzems amb diferents tipus d'arxius INI (veure Figura 3). Aquest pas es fa mitjançant una aplicació de GeoMedia dins la mateixa carpeta d'executables.

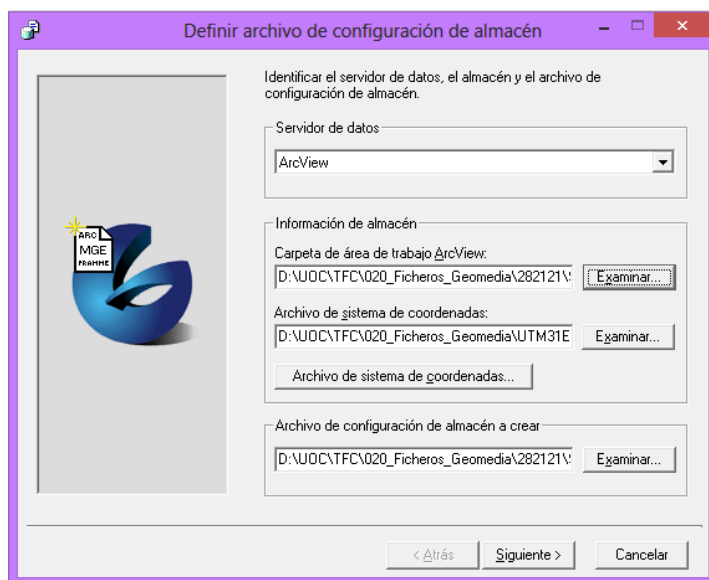


Figura 3. Definició d'un arxiu de configuració de magatzem.

A continuació es crea una nova connexió a un magatzem del tipus ArcView per a enllaçar les dades proveïdes pels fitxers *Esri Shapefile* (veure Figura 4).

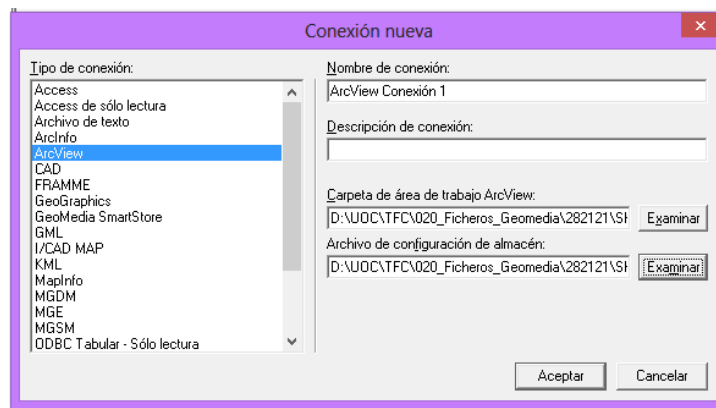
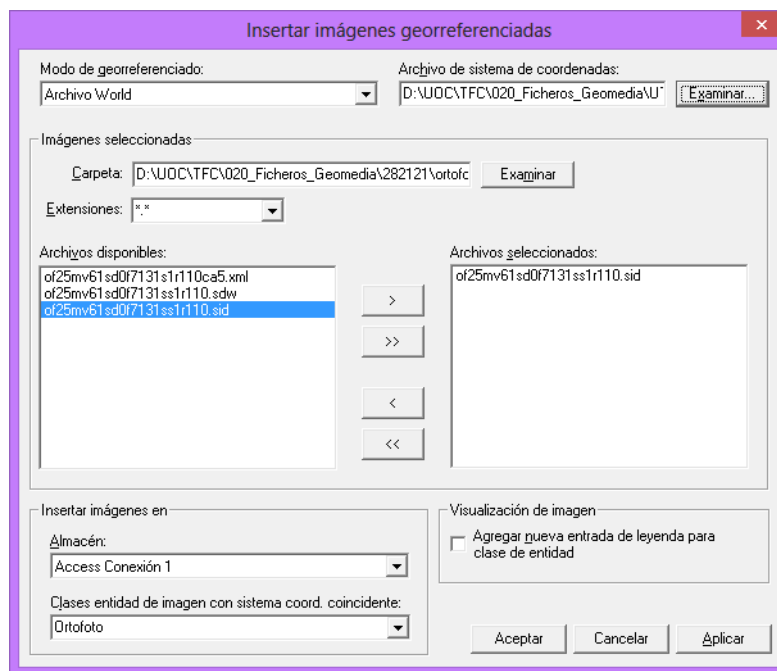


Figura 4. Definició d'una connexió nova.

Per a afegir la informació continguda a la capa ràster del format MrSID es crea al magatzem del GeoWorkspace un nova classe d'entitat del tipus *Imatge* i seguidament es pot inserir una imatge georeferenciada (en aquest cas una ortofoto) que estarà associada al magatzem creat anteriorment (veure Figura 5).


Figura 5. Importació d'una ortofoto dins la classe entitat *Ortofoto* creada anteriorment.

Finalment, s'activa la llegenda corresponent a les capes que es vulguin visualitzar. La Figura 6 mostra un exemple de la pantalla principal de GeoMedia Professional on es poden veure les diferents capes vectorials importades del fitxer *ESRI Shapefile*, les quals inclouen, entre d'altres, els edificis, les corbes de nivell, boscos, etc. Com a exemple d'imatge ràster es mostra l'ortofoto.

Per a importar un fitxer del tipus KMZ amb informació de les alçades dels edificis, cal primer crear una nova classe d'entitat, similar a com s'ha fet anteriorment amb l'ortofoto. Seguidament, es fa servir la utilitat de GeoMedia Professional *Importar objetos 3D*, la qual inclourà els registres necessaris dins la classe entitat creada

anteriorment. Fent servir l'opció *Replace feature class* la utilitat crearà els camps necessaris dins la taula, on cal destacar el camps *GM3D_Model*, i *origin_x*, *origin_y*, *origin_z*, que contenen la referència del model 3D i les coordenades centrals, respectivament. Malauradament, la utilitat *Importar objetos 3D* es queda penjada durant l'execució. Així i tot, es generen alguns fitxers i la geometria es pot visualitzar, tal i com mostra la Figura 8, amb la limitació que s'ha hagut de fixar una alçada constant per a tots els edificis. Es desconeix l'origen del problema, però podria estar relacionat amb la manca de compatibilitat de GeoMedia 3D amb Windows 8, que és el sistema operatiu sota el qual s'està treballant.

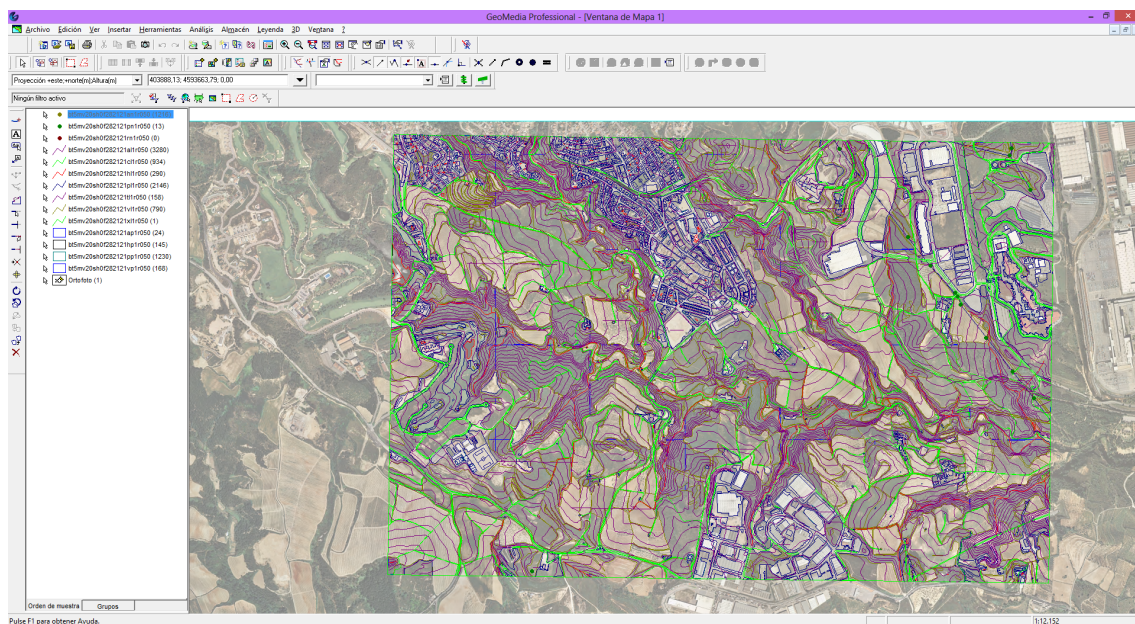


Figura 6. Pantalla principal de GeoMedia Professional on es poden veure les diferents capes importades. En aquest cas s'han desactivat alguns elements per a la visualització. L'ortofoto es mostra de fons amb una transparència del 50%.

1.5.2 Microsoft Visual Studio Express 2012

Amb la finalitat de programar l'aplicació SIG s'ha fet servir el llenguatge de programació Visual Basic dins l'entorn Microsoft Visual Studio Express 2012 [3]. L'objectiu és programar una aplicació personalitzada (*custom command*) de GeoMedia Professional. Fins a la versió 2008 de Visual Basic era possible fer servir una plantilla per a crear la comanda, però malauradament aquesta opció no existeix per a les versions posteriors a Visual Basic 2008. Tot i així, s'ha pogut trobar una plantilla a Internet [8]⁷, la qual ha permès desenvolupar la utilitat sota Visual Express 2012. La Secció 4.1 exposa més detalls sobre l'entorn Visual Studio i la programació d'aplicacions per a GeoMedia.

⁷ <http://www.intergraph-forum.de/forum/posts/list/141.page>

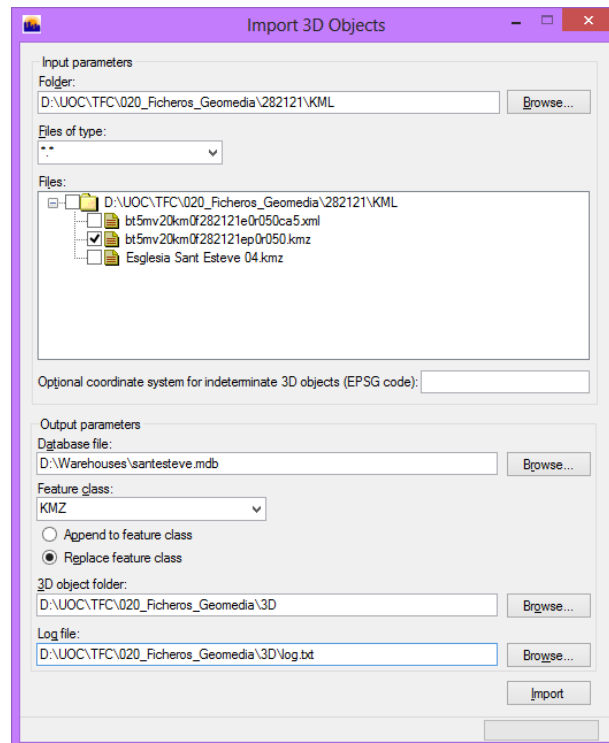


Figura 7. Importació dels objectes 3D mitjançant la utilitat disponible a GeoMedia Professional.

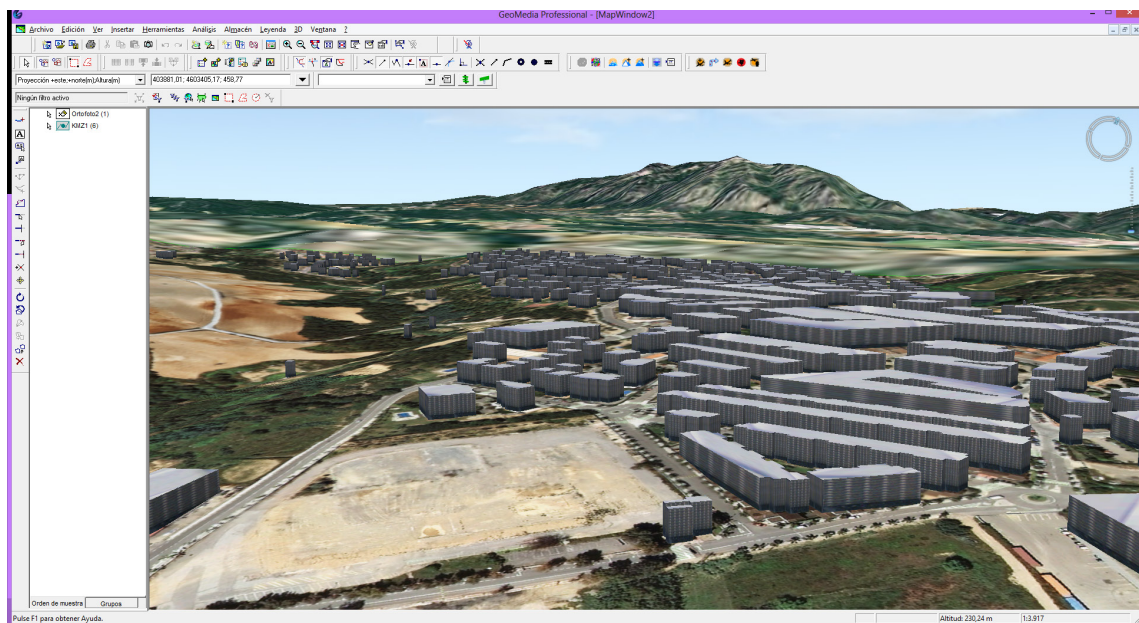


Figura 8. Visualització dels edificis importats del municipi de Sant Esteve Sesrovires amb Montserrat de fons. L'alçada d'extrusió s'ha fixat en aquest cas a 10 metres, ja que en produir-se un error en importar el fitxer KMZ, aparentment les alçades dels edificis no s'han guardat correctament a la base de dades.

2. ANÀLISI FUNCIONAL DE L'APLICACIÓ

L'anàlisi funcional de l'aplicació s'encarrega d'analitzar i definir les diferents funcionalitats de l'aplicació, així com l'estructura de les dades a emmagatzemar i la interfície amb l'usuari. Les següents subseccions mostren els casos d'ús, el model de dades, els algorismes a implementar, i la interfície d'usuari.

2.1 Definició de casos d'ús

Per a la present aplicació s'han definit quatre casos d'ús i un únic actor que és l'usuari del programari GeoMedia. La representació gràfica dels casos d'ús es presenta a la Figura 9, i la seva descripció és la següent:

- 1) **Generació d'una base de dades de tots els polígons dels edificis d'una zona seleccionada:** En aquest cas es calculen els polígons 3D a partir dels polígons 2D dels edificis i l'alçada dels mateixos. Per a cada edifici s'inclouen els casos *Identificació dels polígons de la façana principal i la coberta* i *l'Actualització de la base de dades amb els valors per a cada edifici*. D'aquesta manera s'emmagatzemaran les dades que es necessiten pels casos posteriors.
- 2) **Càlcul del percentatge de radiació solar sobre la coberta i la façana principal a partir d'un dia i hora concret per a tots els edificis de la zona seleccionada:** Aquest cas inclou el *Càlcul de l'azimut i l'altitud solar (per a una data i hora concreta)* i *una posició sobre l'edifici (longitud, latitud)*, *Càlcul de visió directa amb el sol dels N punts significatius del polígon donat* i *l'Actualització de la base de dades amb els valors per a cada edifici*, on en aquest darrer cas s'emmagatzemaran els percentatges.
- 3) **Càlcul de les hores totals de sol a la coberta i sobre els diferents pisos de la façana principal, per a un dia i/o mes d'un any concret, per a tots els edificis d'una zona seleccionada:** Aquest cas inclou *Càlcul de l'ortus i l'ocàs (per a un dia donat)*, *Càlcul de l'azimut i l'altitud solar (per a una data i hora concreta)* i *una posició sobre l'edifici (longitud, latitud)*, *Càlcul de visió directa amb el sol dels N punts significatius del polígon donat* i *l'Actualització de la base de dades amb els valors per a cada edifici*.
- 4) **Representació temàtica dels valors calculats per a tots els edificis:** Aquest cas genera les capes corresponents als casos 2) i 3), tot generant una entrada temàtica per rang de colors, per exemple, pel cas 2) una escala de grisos partint del negre (0%) fins a blanc (100%) amb increments del 10%.

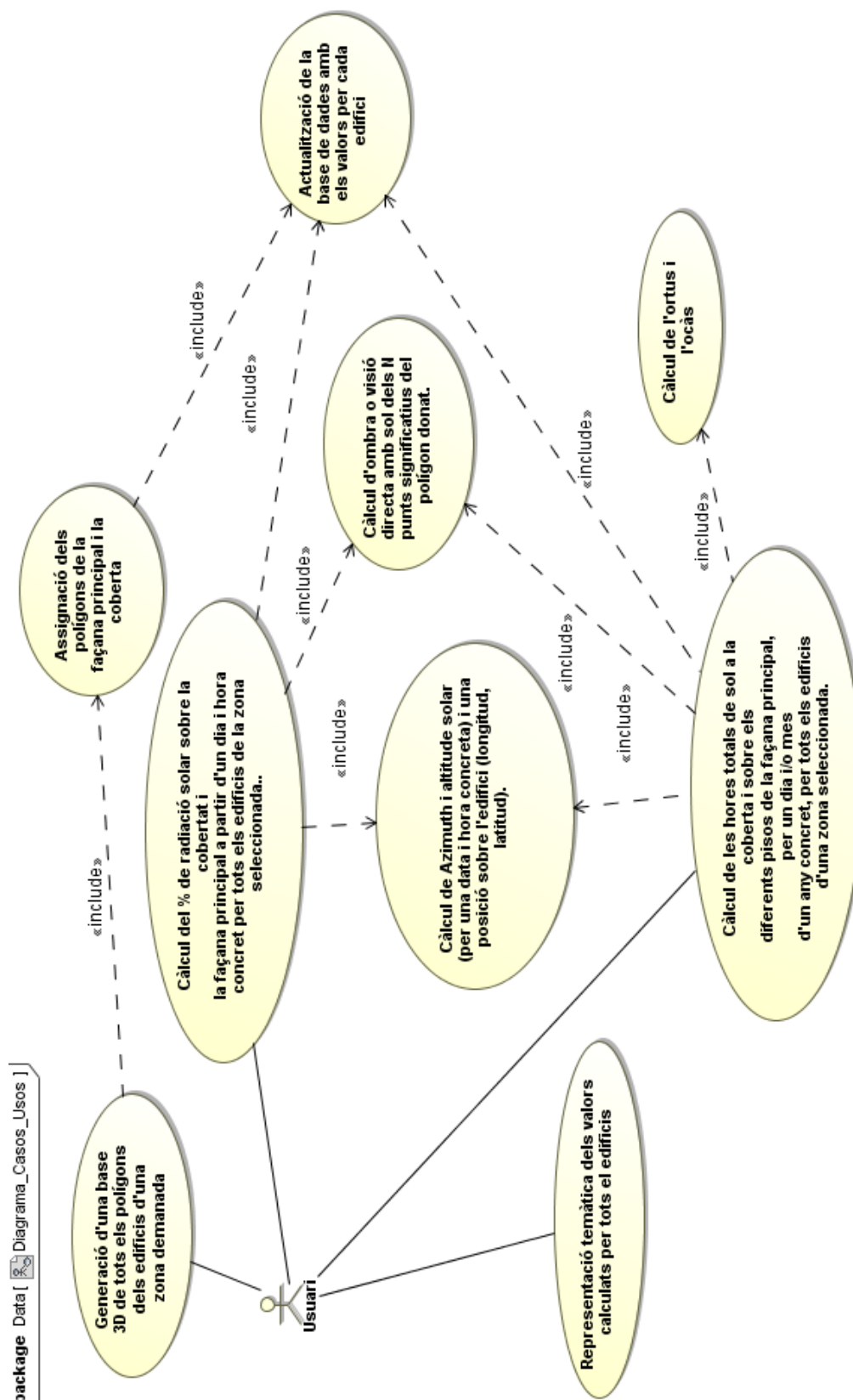


Figura 9. Diagrama de casos d'ús de l'aplicació desenvolupada.

2.2 Model de dades

El model de dades de l'aplicació es compon en principi de tres taules que es descriuen a continuació:

- 1) **Poligons3D**. Taula per a guardar els polígons 3D dels edificis. L'estructura és la següent:

Id1	Id_Edifici	Geometria	Tipus_Poligon	Altura
Autonumèric	Text(80)	OLE Object	Text(1)	Double

PRIMARY KEY (Id1)

Id1, Autonumèric enter {Comptador autonumèric identificador del polígon}

Id_Edifici, Text(80) {Identificador de l'edifici}

Geometria, OLE Object {Polígon dins l'edifici}

Tipus_Poligon, Text(1) {Identifica el polígon com a coberta 'C', façana principal 'F' o qualsevol de les altres façanes 'A'}

Altura, Double {alçada de l'edifici al que pertany el polígon}

Aquesta taula emmagatzema un registre per a cada polígon de l'edifici. S'està assumint que l'identificador de l'edifici és una cadena de text. El polígon s'emmagatzema dins el camp Geometria, que GeoMedia tracta com a objecte OLE. El Tipus_Poligon podrà ser 'C' o 'F' segons si es refereix a la coberta o a la façana principal i 'A' si no pertany a cap dels altres casos. Notar que el polígon original –el que es troba a nivell de terra– no s'emmagatzema a la taula. És a dir, només s'emmagatzemen les cares (inclosa la façana) i la coberta.

- 2) **PercentatgeSolar**. Taula per a emmagatzemar el percentatge de radiació solar sobre la coberta d'un edifici i la seva façana principal per a un dia i hora concret. L'estructura és la següent:

Id1	Id_Edifici	Data	Geometria	pc_rad_coberta	pc_rad_façana_p
Autonumèric	Text(80)	Date	OLE Object	Double	Double

PRIMARY KEY (Id)

ID, Autonumèric enter {Comptador autonumèric}

Id_Edifici, Text(80) {Identificador del edifici}

Data, Date {Data i hora demanades}

Geometria, OLE Object {Polígon de la coberta}

pc_rad_coberta, Double {% de radiació calculat sobre la coberta de l'edifici}

pc_rad_façana, Double {% de radiació calculat sobre la façana principal de l'edifici}

Aquesta taula emmagatzema un registre per a un dia i hora concret amb els valors calculats de radiació per a la coberta i la façana principal. Els punts interpolats necessaris per a calcular el tant per cent de radiació a la coberta o a la façana principal no s'emmagatzemen, sinó que es calculen al moment per a no haver de guardar-los. El camp Geometria conté la coberta de l'edifici i es fa servir per a poder visualitzar posteriorment els resultats dins de GeoMedia.

- 3) **HoresTotals**. Taula per a emmagatzemar les hores totals de sol a la coberta i sobre els diferents pisos de la façana principal d'un edifici per a un dia i/o mes d'un any concret. L'estructura de la taula és la següent:

Id	Id_Edifici	Periode	Planta	Geometria	HoresTotals
Autonumèric	Text(80)	Text	Text (1)	OLE Object	Double

PRIMARY KEY (Id)

Id, Autonumèric enter {Comptador autonumèric}

Id_Edifici, Text(80) {Identificador del edifici}

Periode, Text(80) {Cadena de text que inclou la data d'inici i la data final del periode avaluat}

Planta, Text(1) {Número de planta}

Geometria, OLE Object {Polígon de la coberta}

HoresTotals, Double {Hores totals calculades de sol sobre la planta corresponent del'edifici}

Aquesta taula emmagatzema un registre per cada planta d'un edifici. Les hores totals de sol de la coberta es guarden amb el número de planta igual al màxim del número de plantes de l'edifici més 1. Igual que abans, el camp Geometria s'emmagatzema per a poder visualitzar posteriorment els resultats dins de GeoMedia.

2.3 Definició dels algorismes

Les principals funcionalitats de l'aplicació que s'ha desenvolupat (veure Secció 1.1) es basen en una sèrie d'algorismes que esdevenen la principal contribució d'aquest TFC. A continuació s'enumeren els algorismes:

- Algorisme per a calcular l'azimut i l'altura del sol respecte un punt de referència per una hora i un dia donats.
- Algorisme per a calcular les hores corresponents a l'ortus i l'ocàs per un dia donat per un punt de referència donat.
- Algorisme de *ray tracing* per a saber si un punt donat rep la llum directa del sol o no.
- Alternativament a l'algorisme de *ray tracing* també s'ha implementat l'algorisme per proporció de triangles.

Amb aquests algorismes es podran realitzar les següents funcionalitats:

- Càlcul del percentatge de radiació sobre la coberta i la façana per a un dia i una hora donats.
- Càlcul de les hores totals de sol a la coberta i sobre els diferents pisos de la façana principal d'un edifici per a un dia i/o mes concret de l'any.

Independent als algorismes i funcionalitats esmentats es troba la primera de les funcionalitats que apareix a la introducció, és a dir, una eina per a afegir les alçades als polígons 2D i així obtenir els polígons 3D dels edificis que són la principal font de dades de cara als càlculs.

A continuació es descriuen les funcionalitats i els algorismes en detall.

2.3.1 Generació dels polígons 3D

Aquesta funcionalitat assumeix que a la base de dades hi ha una taula amb el nom *Poligons2D* que conté les geometries de les bases dels edificis. Per a saber com s'ha de generar a partir de fitxers SHP descarregats de l'Institut Cartogràfic de Catalunya [6], mirar la Secció 4.3 amb el manual d'usuari.

En general, la taula *Poligons2D* contindrà els diferent polígons que defineixen altres entitats i no només edificis. Per això, l'algorisme que afegeix les alçades filtra els edificis que tinguin dins el camp CAS l'identificador "EDI01". La resta de registres es corresponen a edificis en construcció o altres tipus de superfícies.

El pseudo-codi de l'algorisme que afegeix les alçades és el següent:

```
Obrir taula Poligons2D per a lectura (cursor objRSInput)
Obrir taula Poligons3D per a escriptura (cursor objRSOutput)
count = 0 'comptador d'edificis
Percada Poligon dins la taula Poligons2D fer
  Si Camp("CAS") = "EDI01" llavors fer
    Si opcioAlçadaRandom = True llavors alçada = CDb(Int( (AlçadaMàxima-AlçadaMínima) * Rnd() +
                                                                AlçadaMínima))

    sino alçada = alçadaConstant
    nomEdifici = "Edifici" + Convert.ToString(count + 1)
    AfegirAlçades(objRSInput, objRSOutput,nomEdifici,alçada)
    count++
  FiSi
  objRSInput.SegüentRegistre()
FiPer
```

L'alçada que s'afegeix pot ser aleatòria o constant, on els valors es defineixen a través de la interfície (veure Secció 2.4). S'ha decidit fer-ho així ja que l'alçada dels edificis no està inclosa dins del fitxer SHP. En aquest TFC no s'ha implementat una funcionalitat que determini les alçades reals dels edificis a partir d'una altra font de dades. Una primera opció que es va intentar va ser la d'importar les alçades del cadastre, però no va ser possible trobar la informació dins de la pàgina web. Una altra possibilitat seria fer servir el fitxer KMZ mencionat a la Secció 1.5.1, i a partir de les coordenades dels polígons de la taula *Poligons2D* i els corresponents al KMZ trobar les alçades. Recordar que la importació del fitxer KMZ va donar errors, tal i com s'ha comentat en la mateixa secció 1.5.1. En qualsevol cas, la contribució d'aquest TFC s'ha centrat en les altres dues funcionalitats.

La funció **AfegeixAlçadaEdifici()** crea els polígons de les cares i la coberta amb l'alçada corresponent per a un edifici. Donat N punts dins el polígon (on l'últim punt és el mateix que el primer, ja que els polígons dins Geomèdia han de ser tancats) es crearan també N polígons ($N - 1$ cares més la coberta). Les cares es creen agafant els punts a parells i afegint l'alçada (cadascun té un total de cinc punts), mentre que la coberta és agafar tots els punts originals i suma'ls-hi l'alçada. El pseudo-codi és el següent:

```
Procediment AfegeixAlçadaEdifici(nomEdifici, alçada)
  objEdifici = StorageToGeometry(objRSInput.GFields("Geometry").Value)
  Desde m=1 fins a objEdifici.Points.Count
    Si m = objEdifici.Points.Count llavors
      objPoligonNou = AfegirPoligonCoberta(objEdifici,alçada)
```

```

Tipus = "C"
sino
  objPoligonNou=AfegirPoligonCara(objEdifici.Points.Item(m), objEdifici.Points.Item(m+1),
    alçada)
  Si m=1 llavors Tipus "F" sino Tipus="A"
FiSi
AfegirPoligonATaula(objRSOutput,objPoligonNou,Tipus,nomEdifici)
FiDesde
FiProcediment

```

Donat que, a més de la coberta, també es necessari definir una façana, el que s'ha decidit en aquest cas és que la primera cara que es defineix ($m=1$) és la que s'identifica com a façana (Tipus = "F"), mentre que la resta son definides amb Tipus="A". A l'última iteració és quan s'afegeix la coberta dins la taula (Tipus = "C"). Les funcions **AfegirPoligonCoberta** i **AfegirPoligonCara** són molt senzilles, ja que només creen els polígons corresponents. El codi es pot trobar dins el codi adjunt al projecte.

Evidentment, en el futur seria interessant disposar de la informació verídica sobre quina cara de l'edifici és la façana, per exemple, a través d'altres bases de dades (cadastre o ajuntament), o bé calcular-la automàticament assumint, per exemple, que la façana principal és la paral·lela a l'eix del vial més proper a l'edifici.

És important destacar que aquestes funcions assumeixen que les dades es troben en un tipus de projecció on la variable Z és l'alçada, de manera que només cal sumar l'alçada desitjada a aquesta component.

2.3.2 Càlcul de la posició del sol

Un possible algorisme per a calcular els angles d'azimut i alçada del sol respecte a un punt es pot trobar fent servir el descrit a [11]⁸, el qual es basa en la descripció de la plana C24 del *Astronomical Almanac* de l'any 1996. La versió *online* es pot trobar a [12]⁹. La Figura 10 mostra aquests angles en el sistema de coordenades horitzontal, on es pot veure que l'angle azimut és donat respecte al nord i l'angle d'altitud respecte al pla horitzontal.

Els paràmetres d'entrada de l'algorisme són la data (any, mes i dia), les hores i els minuts (en UT, *Universal Time*), i les coordenades del punt desitjat en latitud i longitud (en graus decimals). Les passes es descriuen breument a continuació:

1. Calcular els dies respecte l'any J2000 (1 de gener del 2000 a las 12 hores de temps terrestre).
2. Calcular la longitud mitja del sol en el dia d'interès, L .
3. Calcular la anomalia mitja del sol en el dia d'interès, g .
4. Calcular la longitud eclíptica del sol, λ .
5. Calcular la obliquitat del pla eclíptic, ϵ .
6. Calcular l'ascensió recta, α , i la declinació, δ , del sol.
7. Calcular l'hora local sideral (*local sideral time*, LST) i l'hora angle del sol.

⁸ <http://www2.arnes.si/~gljsentvid10/sun.html>

⁹ <http://asa.usno.navy.mil/>

8. Calcular l'altitud, A , i l'azimut, Z .

S'ha de tenir en compte que aquest algorisme no considera la refracció dels rajos de sol a l'atmosfera ni el semidiàmetre angular del sol. Les fórmules implementades s'han verificat amb les calculadores solars posades a disposició dins l'assignatura o d'altres trobades a Internet.

D'igual manera també s'han implementat les fórmules per a calcular les hores de l'ortus i l'ocàs per a un dia donat. En aquest cas s'ha fet servir la pàgina HTML posada a disposició dins l'assignatura.

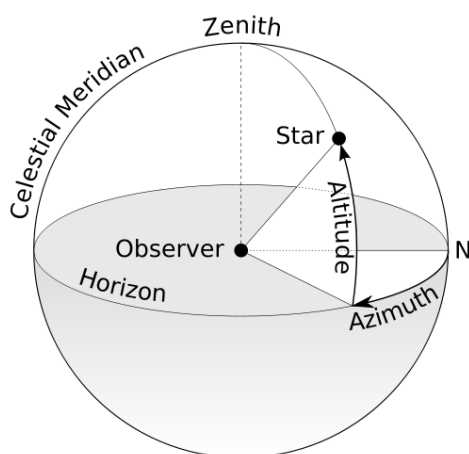


Figura 10. Representació dels angles d'azimut i altitud del sol calculats respecte a l'observador. Font: [13]¹⁰.

2.3.3 Algorisme de *ray tracing*

L'algorisme de *ray tracing* és un dels dos algorismes escollits per a calcular si un raig de sol arriba a un punt determinat (veure la següent secció per a l'algorisme de proporció de triangles). Com a punt de partida, es pot assumir que la funció que calcula l'azimut i l'alçada del sol presentat a la secció anterior és un sistema de coordenades esfèriques centrat al punt d'interès, \vec{p}_{ref} , situat, per exemple, sobre la coberta de l'edifici que es vol avaluar. Així doncs, en un sistema cartesià centrat a aquest mateix punt de referència \vec{p}_{ref} , el punt \vec{p}_{sol} ve donat per

$$\vec{p}_{sol} = (r \cdot \cos \varphi \cdot \sin \alpha, r \cdot \cos \varphi \cdot \cos \alpha, r \cdot \sin \varphi) \quad (1)$$

on α és l'angle azimut, φ l'angle d'altitud, i r és una distància suficientment gran per a poder assumir que els rajos que arriben a la terra són paral·lels. En aquest sentit, una distància d'uns 600 km és suficient, ja que el valor real donaria problemes de precisió de coma flotant. Notar que els angles definits a l'equació anterior difereixen dels donats a [9]¹¹ ja que en coordenades UTM la X és l'est i la Y el nord, i a més també s'ha de tenir en compte el signe de l'angle azimut.

¹⁰ http://en.wikipedia.org/wiki/Horizontal_coordinate_system

¹¹ http://en.wikipedia.org/wiki/Spherical_coordinate_system

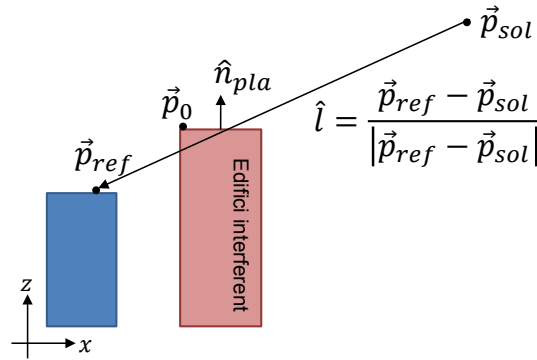


Figura 11. Esquema de l'algorisme de *ray tracing* per a avaluar si l'edifici interferent (vermell) dona ombra o no sobre un punt donat de l'edifici a avaluar (blau).

La Figura 11 mostra un esquema de la situació on es vol avaluar si un edifici interferent (color vermell) dona ombra o no sobre un punt de la coberta de l'edifici a avaluar (color blau). Donat un polígon de l'edifici interferent, per exemple la coberta, es calcula el seu vector normal, \hat{n}_{pla} . En primer lloc, es fan servir tres punts d'aquest polígon per a obtenir dos vectors i així poder calcular el producte vectorial

$$\vec{p}_{pla} = (\vec{p}_{pla,2} - \vec{p}_{pla,1}) \times (\vec{p}_{pla,3} - \vec{p}_{pla,1}), \quad (2)$$

i després es normalitza

$$\hat{n}_{pla} = \frac{\vec{p}_{pla}}{|\vec{p}_{pla}|}. \quad (3)$$

Seguidament es calcula el vector director de la recta entre el punt a avaluar, \vec{p}_{ref} , i el sol, donat per

$$\hat{l} = -\frac{\vec{p}_{sol}}{|\vec{p}_{sol}|}, \quad (4)$$

ja que es té un sistema de coordenades centrat al punt de referència. Amb aquest vector director, es pot calcular el punt d'intersecció d'aquesta recta amb el pla definit per la coberta de l'edifici interferent. La distància d respecte al sol en la direcció donada per \hat{l} on la recta intersecciona el pla ve donada per [10]¹²

$$d = \frac{(\vec{p}_0 - \vec{p}_{sol}) \cdot \hat{n}_{pla}}{\hat{l} \cdot \hat{n}_{pla}} \quad (5)$$

on " \cdot " representa el producte escalar i \vec{p}_0 és un punt dins el polígon de l'edifici interferent (qualsevol dels punts que formen el polígon). En el present cas la recta sempre interseccionarà el pla i per tant la fórmula no donarà cap indeterminació. Un cop calculat d , es pot calcular el punt dins el pla amb

$$\vec{p}_{pla} = d \cdot \hat{l} + \vec{p}_{sol} + \vec{p}_{ref}, \quad (6)$$

on s'ha afegit \vec{p}_{ref} per a tenir el punt en les coordenades del Workspace i poder fer el darrer càlcul. Efectivament, ara és necessari calcular si aquest punt cau dins del polígon de la cara o coberta de l'edifici interferent que s'està avaluant. Amb aquesta

¹² http://en.wikipedia.org/wiki/Line-plane_intersection

finalitat es pot fer servir la funció de GeoMedia *PointInPolygon* que retorna 1 si el punt ho està. Si és aquest el cas, el punt \vec{p}_{ref} és a l'ombra, en cas contrari, el sol l'il·lumina i es marca com a tal. Aquest procés s'ha de repetir per a cada punt a sobre la coberta i per a cada polígon de cada edifici interferent, tal i com es descriu a la Secció 2.3.5.

L'algorisme implementat realitza un parell de càlculs addicionals per a disminuir la càrrega computacional, que són:

- Al principi de l'algorisme es calcula l'angle entre el vector director del polígon i el raig de sol mitjançant el producte escalar. Si l'angle està entre 85° i 105° , el polígon es descarta completament, ja que es considera que el raig és gairebé paral·lel al pla i per tant no donarà cap ombra.
- Seguidament es mira si l'alçada màxima del polígon és més baixa que la coordenada z del punt que s'està avaluant. En cas afirmatiu el polígon no dona ombra a aquest punt.
- Després de calcular el punt \vec{p}_{pla} i abans de cridar la funció *PointInPolygon*, s'avalua si la coordenada z del punt és més gran que la màxima coordenada z del polígon que s'està avaluant. En cas afirmatiu, el punt és fora del polígon i per tant no dona ombra a aquest punt.
- Immediatament després s'avalua si el punt \vec{p}_{pla} es troba entre el punt essent avaluat i el sol al pla X-Y mitjançant el càlcul de distàncies. Si no és el cas, tampoc dona ombra.
- Finalment, es crida la funció *PointInPolygon*.

Un detall important a tenir en compte és la tolerància en la mesura a l'hora de fer els càlculs. Quan s'avalua si un punt és a dins del polígon, pot haver-hi errors degut a manca de precisió o limitacions de coma flotant. Per aquest motiu, no sempre serà 0 quan hauria de ser-ho. GeoMedia té un paràmetre dins l'objecte *GeoMathSvc.GeoMathService* anomenat *PositionalTolerance* que permet controlar aquest marge, i acostuma a ser de 0.0001 metres. Si s'incrementa aquest valor, s'incrementarà també la tolerància, però cal tenir en compte que la tolerància es fa servir també per a altres càlculs, i per tant no convé posar-la molt gran. En qualsevol cas, el valor per defecte de 0.0001 dona bons resultats.

2.3.4 Algorisme per proporció de triangles

El segon mètode implementat per a calcular si un punt rep llum és el de proporció de triangles, tal i com mostra la Figura 12. Donada la distància al pla X-Y entre el punt de referència i el sol, d_s , i la diferència d'alçada entre aquests, Δz_s , es pot calcular l'alçada que tindrà aquest triangle en funció de la distància que es vulgui avaluar. Si se sap que l'edifici interferent està a una distància d_i al pla X-Y del punt de referència, la diferència d'alçada mínima que ha de tenir l'edifici interferent ve donada per

$$h_{min} = \frac{d_i}{d_s} \Delta z_s, \quad (7)$$

de manera que si la diferència d'alçades entre els dos edificis, Δz_i , és igual o més gran que h_{min} llavors el punt de referència no rep la llum directa del sol.

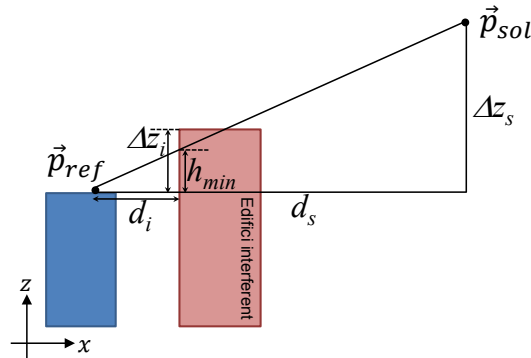


Figura 12. Esquema de l'algorisme fent servir proporció de triangles per a avaluar si l'edifici interferent (vermell) dona ombra o no sobre un punt donat de l'edifici a avaluar (blau).

Aquest algorisme és més eficient en termes computacionals que el de *ray tracing*, ja que no cal avaluar cada polígon de l'edifici independentment, però és menys precís. Efectivament, l'avaluació del punt a la mínima distància serà sempre el mateix per a un mateix edifici interferent i per tant l'algorisme no té en compte ni la forma ni l'orientació d'aquest. Un altre inconvenient afegit és que s'ha d'avaluar si l'edifici interferent es troba realment en la línia recta que uneix el punt de referència i el sol, fet que requereix càlculs addicionals i que està considerat inherentment a l'algorisme de *ray tracing*.

Un possible algorisme per a saber si un edifici es troba sobre la línia que uneix el sol i el punt d'interès és el següent. Donada la configuració de la Figura 13, es vol avaluar si l'edifici interferent (de color vermell) es troba sobre la línia que uneix el sol i un punt de l'edifici a avaluar (de color blau). El primer pas és generar els diferents vectors entre cadascun dels punts que formen la coberta de l'edifici interferent i el punt del sol

$$\vec{v}_i = \vec{p}_i - \vec{p}_{sol}, i \in 1 \dots N, \quad (8)$$

on N és el número de punts de la coberta. Llavors cal avaluar el producte vectorial entre cada vector \vec{v}_i i el vector \hat{l} que uneix el punt d'interès i el sol, on prèviament s'haurà posat la coordenada z dels dos vectors a 0, de manera que els vectors són a sobre del pla X-Y. El resultat ve donat per

$$\vec{e}_i = \vec{v}_i \times \hat{l}, i \in 1 \dots N. \quad (9)$$

El pas final consisteix a avaluar la coordenada z del vector resultant \vec{e}_i . Si l'edifici que s'està avaluant no es troba sobre la línia recta, llavors totes les coordenades z tindran el mateix signe (positiu si l'edifici es troba per sobre de la línia, o negatiu si es troba per sota). Aquesta és la situació donada en la Figura 13 (a). D'altra banda, si l'edifici es troba realment sobre la línia recta, com a la Figura 13 (b), llavors hi haurà valors positius i negatius en la coordenada z dels vectors \vec{e}_i calculats. Matemàticament això es pot expressar de la següent manera

- Si per cada i , $e_{z,i} > 0$ o $e_{z,i} < 0$, llavors l'edifici no és sobre la recta;
- Si $\min\{e_{z,i}\} < 0$ i $\max\{e_{z,i}\} > 0$, llavors l'edifici és sobre la recta.

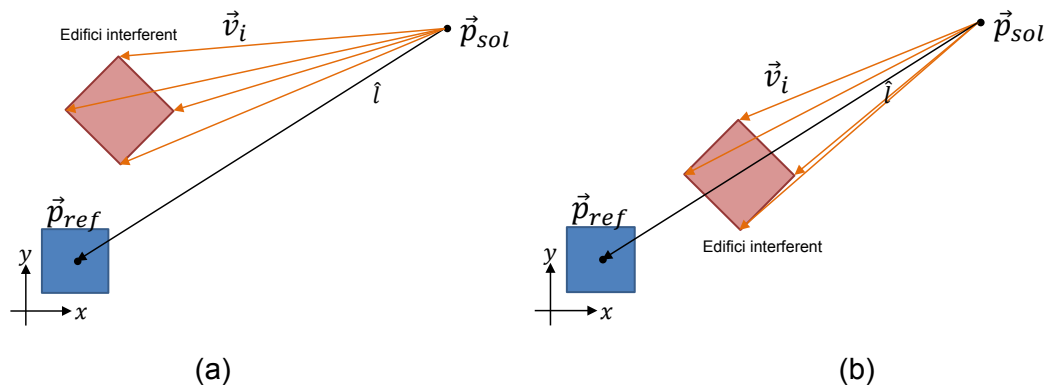


Figura 13. Esquema de l'algorisme per a saber si un edifici interferent (edifici vermell) es troba sobre la recta que uneix el sol i un punt sobre l'edifici d'interès (edifici blau). La figura de l'esquerra mostra una situació en la que l'edifici avaluat no interfereix, mentre que a la figura de la dreta sí succeeix.

Si l'edifici es troba sobre la recta, llavors es passa a avaluar l'alçada mínima que ha de tenir fent servir l'equació (7). Aquest algorisme es pot fer només un cop per edifici interferent, agafant llavors el punt central de l'edifici a avaluar (edifici blau), o es pot fer per cada punt sobre l'edifici a avaluar (ja sigui a la coberta o a la façana). El segon cas té un cost computacional una mica més alt i és substancialment més precís que la primera opció. Per aquest motiu s'ha decidit implementar aquesta segona opció. Cal notar però, que tot i que continua essent més eficient que l'algorisme de *ray tracing* exposat a la Secció 2.3.3, és menys precís ja que no té en compte la forma exacta de l'edifici interferent. A més a més, si un edifici és a sobre la línia entre el sol i el punt per pocs metres, a la pràctica serà com si estigués just al davant, la qual cosa produirà més resultats d'ombra que els que hi ha en realitat, tal i com es pot apreciar a la Figura 14. Al capítol de resultats es presenta una comparativa entre aquests dos mètodes.

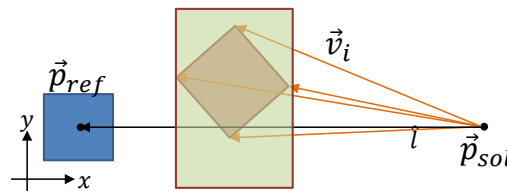


Figura 14. Exemple on per a l'algorisme proporció de triangles l'edifici interfereix per molt poc i, per tant, a la pràctica és com si l'edifici fos completament entre el sol i el punt (rectangle verd).

2.3.5 Càlcul del percentatge de llum i del número total d'hores

Les dues funcionalitats principals de l'aplicació a desenvolupar són el càlcul del percentatge de llum solar sobre cada edifici (coberta i façana principal) per a un dia i hora donats, i el número total d'hores sobre cada edifici (coberta i cada planta) per a un període de temps determinat. Donat que el primer és un cas particular del segon, s'ha optat per una implementació conjunta de les dues funcionalitats per a maximitzar la reutilització del codi. Aquest fet també s'ha tingut en compte a l'hora de dissenyar les classes, tal i com es descriu a la Secció 3.1. Tot i així, de cara a una descripció més senzilla de les dues funcionalitats, és més convenient presentar-les per separat, per a més endavant trobar una harmonització de totes dues més adient. Així doncs, a continuació es presenta el pseudo-codi de la funcionalitat percentatge solar sense tenir en compte l'altra funcionalitat:

```

Percada Edifici dins la taula fer (objecte Edifici)
  Escollir polígon de la coberta i polígon de la façana
  Psol = CalculaPosicioSol(PuntDeLaCoberta,Hora/Dia)
CreacioPunts(PoligonCoberta,PoligonFasana,PuntsCoberta,PuntsFaçana)
  Inicialitzar RaigSolCoberta() a 1
  Inicialitzar RaigSolFaçana() a 1
Per cada Edifici dins la taula fer (objecte EdificiInterferent)
  Si Edifici no és igual a EdificiInterferent llavors
    SiEdificiValid(EdificiInterferent,radi,puntCentreEdifici, Psol) llavors
      Per cada Poligon dins d'EdificiInterferent fer (objecte Poligon)
        Per cada PuntsCoberta fer
          Resultat= AvaluaSolDirecte(PuntsCoberta(Index),Poligon,Psol)
          Si Resultat igual a 0 llavors RaigSolCoberta(Index) = 0
        FiPer
        Per cada PuntsFaçana fer
          Resultat = AvaluaSolDirecte(PuntsFaçana(Index),Poligon,Psol)
          Si Resultat igual a 0 llavors RaigSolFaçana(Index) = 0
        FiPer
      FiPer
    FiSi
  FiSi
FiPer
'mirant polígons del mateix edifici sobre la façana
Per cada Poligon dins edifici excepte la Façana fer(Poligon)
Per cada PuntsFaçana fer
  Resultat = AvaluaSolDirecte(PuntsFaçana(Index),Poligon,Psol)
  Si Resultat igual a 0 llavors RaigSolFaçana(Index) = 0
FiPer

```

Les funcions a implementar són les següents:

- **CalculaPosicioSol**(PuntCentral, Hora/Dia): retorna la posició del sol en coordenades cartesianes centrades al punt central de la coberta per a un dia i una hora donades fent servir l'algorisme descrit a la secció 2.3.2.
- **CreacioPunts**(PoligonCoberta,PoligonFasana,PuntsCoberta,PuntsFaçana): retorna un màxim d'MxM punts repartits regularment sobre el pla de la coberta o sobre el pla de la façana, respectivament (veure Secció 2.3.5.2 pels detalls).
- **EdificiValid**(EdificiInterferent,radi,puntCentreEdifici,Psol): aquesta funció retorna *True* si cal calcular l'ombra d'aquest edifici sobre l'edifici d'interés. La Secció 2.3.5.1 descriu com es fa aquesta avaluació.
- **AvaluaSolDirecte**(PuntsCoberta(Index)/PuntsFaçana(Index), Poligon, Psol): retorna 1 si el punt sobre la coberta rep la llum directa del sol. Pot fer servir qualsevol dels dos algorismes descrits a les seccions 2.3.3 (*ray tracing*) i 2.3.4 (proporció de triangles), respectivament.
- **AvaluaMascaraRaigSolGuardaRegistre**(RaigSolCoberta,RaigSolFaçana): avalua les màscares, calcula els percentatges i crea un registre a la taula de resultats amb el percentatge de llum sobre la coberta per a l'edifici indicat.

Destacar que pel cas de la façana també s'avalua si les altres cares del propi edifici li fan ombra. Per a aquest cas s'ha optat per fer servir sempre l'algorisme de *ray tracing*, ja que el de proporció de triangles no donaria els resultats esperats degut a que la distància mínima del polígon seria en molts casos zero (les cares i la coberta tocant la façana). En qualsevol cas, el fer servir l'algorisme de *ray tracing* no implica un augment

significatiu del temps de càlcul, ja que el veritable coll d'ampolla és el bucle anterior on s'avaluen la resta d'edificis.

A continuació es presenta el pseudo-codi de la segona funcionalitat que ha de calcular el número d'hores sobre la coberta i els diferents pisos de la façana per a un rang de dates donades. En una primera aproximació la implementació que es presenta crida la funció descrita anteriorment per al càlcul del percentatge solar. Però de totes maneres, tal i com s'ha comentat, la implementació final difereix per motius d'eficiència (veure la Secció 3.1.1). El pseudo-codi és mostra a continuació:

```
Percada Edifici dins la taula fer (objecte Edifici)
CreacioPunts(PoligonCoberta,PoligonFasana,PuntsCoberta,PuntsFaçana)
HoresTotalsCoberta = 0
HoresTotalsPisos(nPisos) = 0
Percada Dia dins del rang fer
    CalcularOrtusOcas(PuntPrimerEdifici,horaOrtus,horaOcas)
    Inicialitzar HoresCoberta(nIntervals) a 1
    Inicialitzar HoresFaçana(nIntervalsTemps,nPisos) a 1
    Per cadainterval dins horaOrtus-horaOcas fer
        CalculHora(PuntsCoberta,PuntsFaçana,dia,hora)
        AvaluaMascaraHores(HoresCoberta,HoresFaçana)
    FiPer
    AvaluaHoresTotals(HoresTotalsCoberta,HoresTotalsPisos)
FiPer
GuardarHoresDinsTaula(Edifici,HoresTotalsCoberta,HoresTotalsPisos)
FiPer
```

Les funcions a implementar són les següents:

- **CreacioPunts**(PoligonFasana,PuntsCoberta,PuntsFaçana): en aquest cas, la funció crea els punts a avaluar, tant a la coberta com a les diferents plantes. (veure Secció 2.3.5.2 pels detalls).
- **CàlculOrtusOcas**(PuntPrimerEdifici,horaOrtus,horaOcas): la funció retorna l'hora de l'ortus i l'hora de l'ocàs (veure Secció 2.3.2).
- **CalculHora**(PuntsCoberta,PuntsFaçana,dia,hora): aquesta funció avalua si els punts reben o no la llum del sol per a un dia i hora donats. En principi crida una funció similar a la del càlcul de percentatge solar, on s'avalua cada edifici veí.
- **AvaluaMascaraHores**(HoresCoberta,HoresFaçana): aquesta funció calcula a partir de les màscares quantes hores de sol han rebut l'edifici i les diferents plantes. La decisió es pren de la següent manera: per a la coberta, al menys el 25% dels punts avaluats han d'haver rebut llum per a considerar que en aquell instant de temps la coberta estava sota el sol. Pel cas de cada pis, caldrà que l'únic punt que hi ha (veure Secció 2.3.5.3) rebi llum directa.
- **AvaluaHoresTotals**(HoresTotalsCoberta,HoresTotalsPisos): aquesta funció va acumulant el número d'hores per a l'edifici donat.
- **GuardarHoresDinsTaula**(Edifici,HoresTotals,TaulaHores): crea un registre a la taula de resultats amb el número d'hores totals a cada planta i sobre la coberta de l'edifici corresponent.

2.3.5.1 Selecció dels edificis a avaluar

Per a cada punt de l'edifici a avaluar, cal comprovar si els edificis veïns li fan ombra o no. És un fet, però, que no cal avaluar tots els edificis presents a la base de dades, ja que hi ha certes condicions necessàries que, si no es compleixen, poden permetre descartar d'entrada molts edificis i, per tant, accelerar el procés de càlcul. Les passes per a decidir si un edifici interferent ha de ser avaluat o no són les següents:

1. Si la distància entre el punt i el punt central de l'edifici interferent és més gran que un radi donat, llavors es retorna *False*.
2. Si el punt central de l'edifici interferent es troba més lluny del sol que el punt central de l'edifici essent avaluat en el pla X-Y, llavors es retorna *False*.
3. Arribat aquest punt, es retorna *True* i, per tant, cal avaluar aquest edifici.

El pseudo-codi de la funció és el següent:

```

Funcio EdificiValid(EdificiInterferent, Radi,puntCentreEdifici, pSol)
  pCentreInt = PuntCentral(EdificiInterferent)
  Si Distancia2D(pCentreInt, pCentreEdifici) > Radi llavors retorna False

  Si Distancia2D(pCentreInt,pSol) >Distancia2D(pCentreEdifici,pSol) llavors retorna False

  retorna True
FiFuncio

```

on la funció *PuntCentral()* retorna el punt central del polígon, i la funció *Distancia2D()* retorna la distància euclidiana al pla X-Y entre dos punts.

2.3.5.2 Creació dels Punts pel percentatge solar

Els punts sobre la coberta (o la façana) es calculen de la següent manera. Agafant per exemple la coberta, primer es calculen les màximes i mínimes coordenades en X i Y del polígon en qüestió. Partint del número de punts per costat, M , es generen $M \times M$ punts equidistants dins del rectangle definit pels valors màxims i mínims. Donat que l'edifici no sempre ha d'estar orientat correctament en el pla X-Y, és possible que alguns punts no caiguin dins el polígon. Per aquest motiu es farà servir també la funció *PointInPolygon* per a avaluar si s'agafa aquest punt o no. Com a resultat, no sempre es tindran el màxim número de punts, és a dir, $M \times M$, però això simplifica l'algorisme.

Pel cas de la façana també es fa de manera similar. Amb les coordenades màximes en X, Y i Z, es calculen primer els increments en totes tres direccions

$$\begin{aligned}
 \delta x &= \frac{x_{max} - x_{min}}{M}, \\
 \delta y &= \frac{y_{max} - y_{min}}{M}, \\
 \delta z &= \frac{z_{max} - z_{min}}{M},
 \end{aligned}
 \tag{10}$$

per a després generar els punts fent servir dos bucles. En aquest cas els punts estaran sempre dins el polígon, ja que la façana sempre serà un rectangle on els valors d'alçada no varien.

2.3.5.3 Creació dels punts pel número total d'hores

Pel cas de la coberta, s'agafen directament els punts que defineixen aquesta, és a dir, es trobaran al voltant del polígon i no al mig. Quant als punts de la façana, s'agafa només un punt centrat a la façana. El número de pisos vindrà donat per l'alçada de l'edifici dividit entre 3 metres, ja que assumim aquest valor d'alçada de les plantes. Així, el número de plantes serà $\lfloor h/3 \rfloor$ on h és l'alçada de l'edifici i $\lfloor \cdot \rfloor$ és l'operador que obté el número enter.

2.4 Definició de la interfície

La interfície de cara a l'usuari és molt senzilla, ja que es requereixen pocs paràmetres d'entrada per la seva part. El diagrama de casos d'ús presentat a la Secció 2.1 ja defineix quins són els paràmetres requerits, els quals depenen de les diferents funcionalitats desitjades. Així doncs, a continuació es defineix la interfície per a cada cas:

- Generació dels polígons 3D: donat que la base de dades amb els polígons 2D ja haurà estat creada (veure Secció 2.3.1), l'aplicació llegirà directament la taula amb el nom *Poligons2D* i crearà la nova amb el nom *Poligons3D*. Es donarà l'opció d'escollir les alçades a afegir: o bé aleatòriament, on es poden escollir l'alçada mínima i màxima, o bé una alçada constant per a tots els edificis.
- Càlcul del percentatge de radiació solar per a un dia i hora donats. En aquest cas només caldrà que l'usuari introdueixi la data i l'hora desitjats. Es donarà com a opcions el poder fixar el radi màxim per a descartar aquells edificis que estiguin més enllà d'aquest radi, l'algorisme a utilitzar per al càlcul, així com el poder definir el número de punts per costat per a calcular el número de punts sobre la coberta (o façana). Aquests valors tenen un impacte directe en el temps de càlcul. L'opció de tolerància té a veure amb el discutit a la secció 2.3.3 sobre la tolerància de GeoMedia a l'hora d'avaluar distàncies, però en qualsevol cas es recomana deixar el valor per defecte de GeoMedia.
- Càlcul del número d'hores totals de sol a la coberta i sobre els diferents pisos de la façana principal per a un rang de dates donades. En aquest cas caldrà que l'usuari defineixi el rang de dates desitjades per a fer el càlcul. A més de l'opció de fixar el radi de cerca dels edificis interferents, l'algorisme i la tolerància, a l'usuari també se li donarà l'opció de definir l'interval de temps per al càlcul, per exemple, cada 30 minuts o cada hora.

3. DISSENY DE L'APLICACIÓ

El disseny de l'aplicació s'ha realitzat partint de la base d'un codi modular i reutilitzable. Per aquest motiu, és possible que el disseny escollit no sigui el més eficient, ja que sempre existeix aquest compromís a l'hora de programar. En qualsevol cas, partint del model dissenyat sempre es pot intentar optimitzar el codi en una segona iteració, pas que no s'ha fet durant la realització d'aquest projecte però podria ser tema d'un futur TFC.

A continuació es mostra el disseny de l'aplicació a partir del diagrama de classes. El pla de proves escollit es mostra també com a part del disseny de l'aplicació.

3.1 Diagrama de Classes

Aquesta secció presenta els diagrames de les noves classes de l'aplicació que s'han creat. En aquest sentit, s'entén que no cal explicar les classes ja existents de GeoMedia, ni les pròpies de Visual Basic per a la generació dels formularis o similars.

Tal i com s'ha comentat durant la descripció de les dues principals funcionalitats, el pseudo-codi mostrat no es correspon amb el disseny final, ja que la seva funció és entendre les passes necessàries independentment. Per tant, s'ha fet un esforç de no duplicar codi, cosa que queda patent en el disseny final, especialment pel que respecta a la classe **CalculSolar**, que implementa les funcionalitats de percentatge solar i número total d'hores. La descripció de tallada de cada classe i el perquè de la implementació es mostra a les següents seccions.

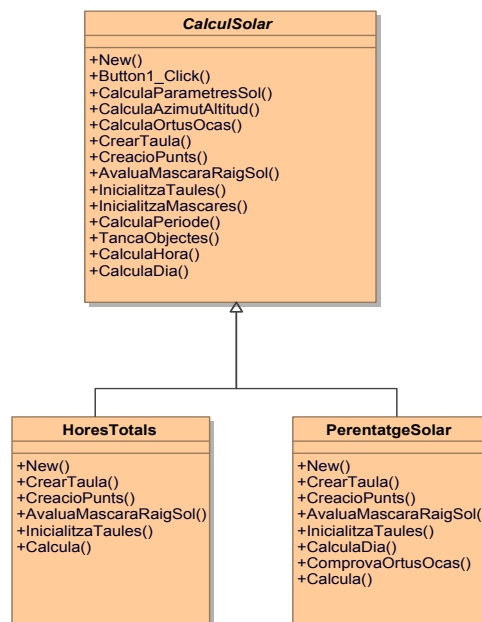


Figura 15. Diagrama de la classe abstracta **CalculSolar** i les sub-classes corresponents als dos algorismes principals.

3.1.1 Classes *CalculSolar*, *PercentatgeSolar* i *HoresTotals*

La classe abstracta ***CalculSolar*** té dues sub-classes, les quals es corresponen amb els dos algorismes principals: el del càlcul del percentatge solar i el del càlcul de les hores totals de sol, tal i com es pot veure a la Figura 15. La classe explota el polimorfisme i per això algunes funcions tenen el mateix nom dins les sub-classes.

El diagrama de flux escollit finalment, que barreja els algorismes mostrats a la Secció 2.3.5, es mostra a continuació de manera descriptiva, on els noms de les funcions són els reals dins les classes. El símbol \hookrightarrow a sota d'una funció indica que el codi amb sangria que estigui per sota s'executa dins de la mateixa (fins arribar a **FiFuncio**).

```

CalculaPeriode()
↳ InicialitzaTaules()
  ↳ CrearTaula()
FiFuncio
  Percada Edifici dins la taula fer
    CreacioPunts()
    Percada Dia dins del Període fer
      CalculaDia()
        ↳ CalculaOrtusOcas()
          Percada Interval dins de Dia fer
            InicialitzaMascares()
            CalculaHora()
              ↳ CalculaAzimutAltitud()
                Percada Edifici dins la taula fer(EdificiInterferent)
                  SiEdificiValid(EdificiInterferent) llavors
                    Per cada Polígon dins de EdificiInterferent fer
                      AvaluaSolDirecte(PuntsCoberta)
                      AvaluaSolDirecte(PuntsFasana)
                    FiPer
                  FiSi
                FiPer
              FiFuncio
                'avaluació màscares i actualització hores pel dia donat
              FiPer
                'avaluació hores totals pel dia donat
              FiFuncio
            FiPer
          AvaluaMascaraRaigSol()
        FiPer
      FiFuncio
    FiPer
  FiFuncio

```

Per a possibilitar l'ús d'aquest algorisme amb les dues funcionalitats, la classe implementa el polimorfisme. Està clar que la funció de percentatge solar no necessita fer un bucle ni per cada dia ni per cada interval. Per aquest motiu la funció *CalculaDia* està implementada a la sub-classe ***PercentatgeSolar***, on es fa la crida directament a la funció *CalculaHora*.

Una altra opció per a implementar l'algorisme hagués estat portar el primer bucle dels edificis a dins de la funció *CalculaHora*. Per al cas de la funcionalitat d'hores totals, això té l'inconvenient que cal anar guardant en memòria els resultats dels càlculs de cada edifici per a poder guardar-los al final, mentre que la implementació escollida fa tots els càlculs d'un únic edifici i emmagatzema els resultats seguidament (amb la crida a la funció *AvaluaMascaraRaigSol*), de manera que no cal guardar en memòria més que una matriu amb els dies i intervals d'aquest únic edifici.

Les funcions d'aquestes classe i sub-classes es descriuen a continuació (veure Figura 15). Les marcades amb * exploten el polimorfisme i per tant s'indica com s'han implementat per a cada sub-classe:

- *New()*: constructor
- *Button1_Click()*: per a controlar l'esdeveniment quan es cancel·la el procés de càlcul (veure Secció 3.1.4).
- *CalculaParametresSol()*: realitza diversos càlculs comuns per a calcular l'azimut/altura i l'ortus/ocàs.
- *CalculaAzimutAltitud()*: retorna la posició del sol pel dia, hora i coordenades donats.
- *CalculaOrtusOcas()*: retorna l'hora de l'ortus i l'ocàs pel dia i coordenades donats.
- *CrearTaula()**: crea la taula corresponent (PercentatgeSolar o HoresTotals). La funció primer verifica si la taula existeix o no. Si existeix i hi ha registres amb el mateix període de temps, llavors demana a l'usuari si vol continuar o cancel·lar. En cas de continuar s'esborraran els registres coincidents.
- *CreacioPunts()**: retorna els punts creats sobre la coberta i la façana. Les funcions dins de cada sub-classe implementen la creació dels punts tal i com s'ha descrit a la Seccions 2.3.5.2 i 2.3.5.3 per a la classe **PercentatgeSolar** i **HoresTotals**, respectivament.
- *AvaluaMascaraRaigSol()**: avalua la informació calculada i la guarda a la taula corresponent, de nou en funció del tipus de sub-classe al que correspongui l'objecte.
- *InicialitzaTaules()**: inicialitza taules per a escriptura/lectura, creant les taules si és necessari cridant la funció *CrearTaula*.
- *InicialitzaMascares()*: inicialitza les màscares pels punts de la coberta i la façana.
- *TancaObjects()*: llibera objectes.
- *CalculaPeriode()*: fa un bucle de dies pel període de temps donat. Dins de la funció es crida la funció *CalculaDia*.
- *CalculaDia()**: fa un bucle d'interval·ls pel dia donat entre l'ortus i l'ocàs. Aquesta funció està implementada dins la classe **CalculSolar** i només la classe **PercentatgeSolar** la implementa independentment fent servir polimorfisme i així cridar directament la funció *CalculaHora*.
- *CalculaHora()*: per al dia i hora donats, fa un bucle per a avaluar tots els edificis interferents.
- *Calcula()*: cada sub-classe té aquesta funció, la qual representa la crida principal per a executar la funcionalitat.
- *ComprovarOrtusOcas()*: aquesta funció només està implementada dins la sub-classe **PercentatgeSolar** per a comprovar que l'hora donada per l'usuari es troba dins de l'ortus i l'ocàs. En cas de no ser així, es cancel·la el càlcul.

Notar que no s'ha inclòs la funció *AvaluaSolDirecte*, ja que aquesta és part de la classe **LlumSolar** que es presenta a continuació.

3.1.2 Classe *LlumSolar*

Aquesta classe s'encarrega de calcular si un punt rep llum solar o no avaluant un polígon corresponent a un edifici interferent determinat. El seu diagrama apareix a la Figura 16. La classe abstracta ***LlumSolar*** té dues sub-classes, les quals es corresponen amb els dos algorismes possibles per a realitzar el càlcul, explotant de nou polimorfisme. D'aquesta manera es poden afegir més algorismes en un futur si així es creu oportú. Les funcions es descriuen a continuació:

- *EdificiValid()*: retorna *True* si l'edifici és vàlid per a ésser avaluat per l'algorisme de llum.
- *ComandaSelect()**: retorna la comanda a l'hora de seleccionar els edificis interferents. La funció a la classe ***LlumSolar*** selecciona tots els polígons d'un edifici donat, és a dir, els que avalua l'algorisme *ray tracing*. Per a l'algorisme proporció de triangles, però, només cal avaluar la coberta, i per això la classe ***ProporcioTriangles*** explota el polimorfisme per a modificar la selecció.
- *AvaluaSolDirecte()*: fa un bucle de tots els punts dins el polígon que està essent avaluat i crida la funció *AvaluaSolDirectePunt*.
- *AvaluaSolDirectePunt()**: avalua si un punt rep o no el raig de sol. Cada sub-classe implementa els algorismes corresponents descrits a les seccions 2.3.3 i 2.3.4 pel cas de *ray tracing* i proporció de triangles, respectivament.
- *EdificiSobreRecta()*: implementa l'algorisme per a saber si l'edifici es troba sobre la línia que uneix el sol i el punt essent avaluat, tal i com està descrit a la Secció 2.3.4. Només és necessari per a l'algorisme de proporció de triangles.

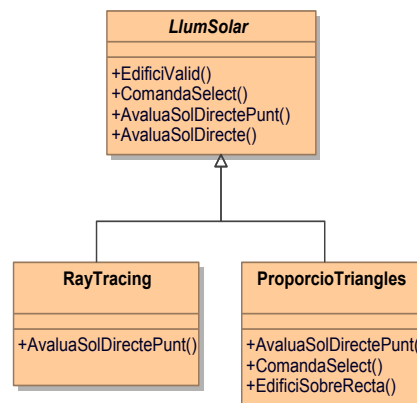


Figura 16. Diagrama de la classe abstracta ***LlumSolar*** i les sub-classes corresponents.

La classe ***CalculSolar*** conté dos objectes de les dues sub-classes. Un d'aquests objectes és sempre del tipus ***RayTracing***, ja que l'avaluació dels punts de la façana pels polígons del mateix edifici es fan sempre amb aquest algorisme. El tipus del segon objecte dependrà de l'algorisme que seleccioni l'usuari.

3.1.3 Classe AfegirAlçades

Aquesta classe implementa les funcionalitats per a afegir les alçades als polígons 2D i així crear la taula Poligons3D. El diagrama de la classe apareix a la Figura 17 i les funcions són les següents:

- *AfegirPoligonCara()*: crea un polígon corresponent a una cara a partir de dos punts consecutius del polígon base i l'alçada.
- *AfegirPoligonCoberta()*: crea el polígon de la coberta a partir del polígon base i l'alçada.
- *AfegeixAlcadaEdifici()*: crea els diferents polígons (coberta i cares) a partir del polígon base i una alçada donada (crida les dues funcions descrites anteriorment). També guarda els registres a la base de dades.
- *AfegeixAlcades()*: implementa el bucle principal que genera els polígons 3D de tots els polígons 2D dins la taula d'entrada.
- *CreaTaulaPoligons3D()*: crea la taula on s'emmagatzemaran els polígons 3D creats.

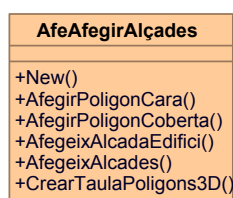


Figura 17. Diagrama de la classe **AfegirAlcades**.

3.1.4 Classe MyProgressBar

Aquesta classe implementa una barra de progrés senzilla que també dóna una estimació del temps restant. La Figura 18 mostra el seu diagrama. Les funcions que conté són:

- *New()*: constructor on s'inicialitzen el màxim número d'iteracions i el text d'informació.
- *Valor()*: rep l'índex, el qual es fa servir per a actualitzar la barra de progrés. El temps s'actualitza cada 1%.
- *Acabar()*: dóna un missatge sobre el temps total trigat en realitzar el càlcul.
- *Button1_Click()*: per a avaluar si l'usuari ha cancel·lat el càlcul.

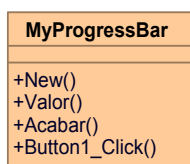


Figura 18. Diagrama de la classe BarraProgres.

Aquesta classe és part d'una classe **Form** i fa servir una classe **ProgressBar** de Visual Basic.

3.1.5 Llibreria d'utilitats

A més de les classes descrites anteriorment, també hi ha una sèrie d'utilitats que s'han inclòs dins del mòdul **Utilitats**. A continuació es descriuen aquestes funcions:

- *VectorsQuasiOrtogonal*(*:*): avalua si dos vectors formen un angle entre 85° i 105° .
- *Distancia*(*:*): retorna la distància euclidiana entre dos punts.
- *Distancia2D*(*:*): retorna la distància euclidiana al pla X-Y entre dos punts.
- *EsborraRegistres*(*:*): esborra els registres del cursor donat.
- *EsborraTotsElsRegistres*(*:*): esborra tots els registres d'una taula donada.
- *ExisteixTaula*(*:*): torna *True* si la taula existeix, *False* altrament.
- *Total*(*:*): torna la suma total dels components d'un *array*.
- *ValorCoordenada*(*:*): torna en forma d'*array* els valors d'una coordenada donada, ja sigui la x, y, o la z, d'un *array* de punts.
- *MaxMin*(vector, max, min): retorna els valors màxim i mínim de l'*array*.
- *AfegirPunt*(*:*): afegeix un punt al polígon donat.
- *PuntCentral*(*:*): retorna el punt mig d'un polígon donat.
- *VectorDirector*(*:*): retorna el vector director del polígon donat.

3.2 Pla de Proves

Els algorismes claus que cal verificar són els relacionats amb la posició del sol (càlcul d'azimut i elevació del sol i hora d'ortus i ocàs, Secció 2.3.2), i el de càlcul de la llum sobre un polígon fent servir *ray tracing* o proporció de triangles (seccions 2.3.3 i 2.3.4). A continuació es defineixen les proves d'aquests dos casos:

- Algorismes relacionats amb el sol: els dos algorismes avaluats (càlcul de la posició del sol, i càlcul de l'ortus i l'ocàs) han estat validats amb d'altres algorismes d'Internet i les eines posades a disposició al tauler de l'assignatura. En tots els casos els resultats eren o iguals o molt similars.
- Algorismes de *ray tracing* i proporció de triangles: per a la validació dels algorismes s'ha definit un escenari senzill amb dos edificis alineats en la recta est-oest i separats una distància d coneguda. Posant el sol exactament a l'est (azimut igual a 90°) a un cert angle d'elevació, és possible resoldre la geometria mitjançant el teorema de Pitàgores, i per tant verificar si els algorismes donen o no els resultats esperats.

Un cop els algorismes particulars per a calcular si un punt rep llum o no han estat avaluats, cal encara però avaluar el conjunt de les funcionalitats, que inclouen la resta d'algorismes descrits a la Secció 2.3. Això s'ha realitzat mitjançant la inspecció visual dels resultats obtinguts amb les geometries reals, on en molts casos era evident que amb la geometria donada els resultats no podien ser correctes del tot.

El pla de proves ha permès, doncs, avaluar i depurar els algorismes per a validar-los i comprovar que s'obtenen els resultats esperats.

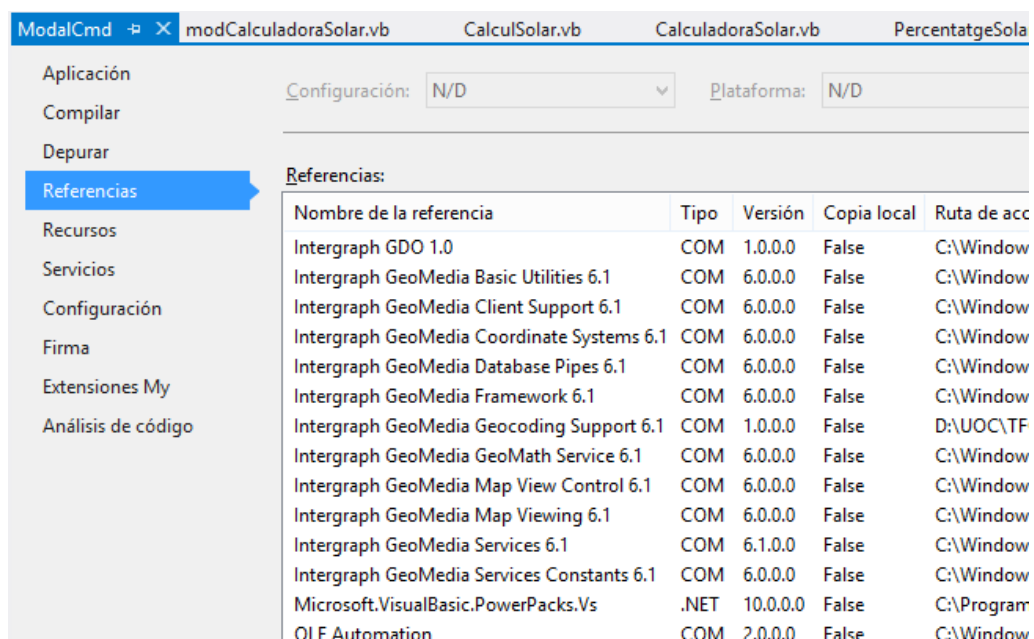
4. CONSTRUCCIÓ DEL SISTEMA

En aquest capítol s'exposen els detalls de la construcció del sistema, la qual inclou principalment la implementació de les funcionalitats en Visual Basic sota l'entorn de Microsoft Visual Express 2012, així com les principals interfícies de GeoMedia que s'han fet servir. Al final d'aquest capítol també es presenta un petit manual d'usuari de l'aplicació desenvolupada.

4.1 Programant amb Visual Basic i GeoMedia

En aquest TFC s'ha desenvolupat una aplicació amb l'objectiu de cridar-la des de dins de GeoMedia, és a dir, com a comanda personalitzada o *custom command*. Fins a la versió 2008 de Visual Studio era possible crear una comanda personalitzada a partir d'una plantilla dins mateix de Visual Basic. A partir de la versió 2010, però, aquesta opció ja no està disponible. Així i tot, s'ha trobat una plantilla vàlida a Internet [8], que ha permès desenvolupar l'aplicació satisfactòriament sota Visual Studio 2012. Aquesta plantilla és molt fàcil de modificar i en el present cas només cal accedir a la funció *Activate* dins la plantilla i posar el codi desitjat a dins. A partir d'aquí, doncs, només cal fer la implementació en Visual Basic.

El primer pas per a començar a programar una comanda *custom* és afegir les llibreries necessàries. La Figura 19 mostra una captura de pantalla amb les llibreries que s'han afegit a l'aplicació per a poder fer servir les diferents funcionalitats de GeoMedia, com per exemple la utilització dels tipus de variables bàsiques de geometria (polígon, punts, punt, etc.), accedir a les connexions i taules obertes a GeoMedia, utilitzar les funcionalitats de transformació de coordenades, accedir a la llegenda de GeoMedia, etc.



Nombre de la referencia	Tipo	Versión	Copia local	Ruta de acc
Intergraph GDO 1.0	COM	1.0.0.0	False	C:\Window
Intergraph GeoMedia Basic Utilities 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Client Support 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Coordinate Systems 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Database Pipes 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Framework 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Geocoding Support 6.1	COM	1.0.0.0	False	D:\UOC\TF
Intergraph GeoMedia GeoMath Service 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Map View Control 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Map Viewing 6.1	COM	6.0.0.0	False	C:\Window
Intergraph GeoMedia Services 6.1	COM	6.1.0.0	False	C:\Window
Intergraph GeoMedia Services Constants 6.1	COM	6.0.0.0	False	C:\Window
Microsoft.VisualBasic.PowerPacks.Vs	.NET	10.0.0.0	False	C:\Program
OLE Automation	COM	2.0.0.0	False	C:\Window

Figura 19. Llibreries importades de GeoMedia dins l'aplicació.

La implementació de l'aplicació ha seguit el disseny mostrat a la secció anterior, on s'ha fet servir un fitxer per a cada classe més un altre per a les utilitats. La Figura 20 mostra l'explorador de solucions del projecte de Visual Basic, on es poden veure els diferents fitxers (amb extensió vb). A part dels fitxers on s'han implementat les classes i que porten el mateix nom, també trobem els següents fitxers:

- frmCalculadoraSolar.vb: correspon al diàleg principal de l'aplicació que s'ha implementat.
- CalculadoraSolar.vb: és el fitxer principal de la comanda personalitzada que conté la funció *Activate* on es crida el diàleg principal de l'aplicació (veure Secció 4.3).
- modCalculadoraSolar.vb: mòdul que ve amb la plantilla per a declarar variables globals per a poder accedir-hi des de qualsevol mòdul. Principalment interessa la que es correspon amb l'objecte que gestiona l'aplicació GeoMedia, *GeoMedia.Application*.

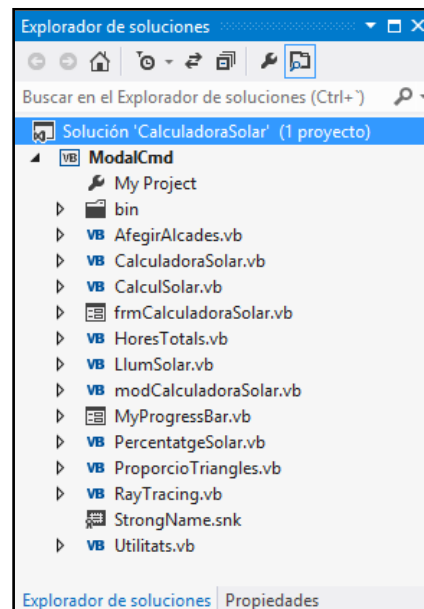


Figura 20. Explorador de solucions de l'aplicació implementada.

La compilació de la plantilla genera el fitxer *CalcladoraSolar.dll* que cal registrar a GeoMedia fent servir un fitxer *script* que acompanya la plantilla (*RegisterModalCmd.bat* per a registrar i *UnregisterModalCmd.bat* per a eliminar). El contingut d'aquests dos fitxers es mostra a continuació:

RegisterModalCmd.bat:

```
REM Registrat COM-Interop amb .NET 4.0
C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\RegAsm ModalCmd.dll /tlb /codebase

REM Registrat com a comanda d'aplicació (Application command)
"C:\Program Files (x86)\GeoMedia Professional\Program\InstallAppCmd" /cmdset GMCCommands /loc %cd% /dll %cd%

REM Registrat com a comanda d'usuari (User command)
REM "C:\Program Files (x86)\GeoMedia Professional\Program\InstallUsrCmd" ModalCmd.dll ModalCmd.ini
```

UnregisterModalCmd.bat:

```
REM Esborrar COM-Interop amb .NET 4.0
C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\RegAsm /u ModalCmd.dll /tlb /codebase

REM Esborrar com a comanda d'aplicació (Application command)
"C:\Program Files (x86)\GeoMedia Professional\Program\InstallAppCmd" /u /cmdset GMCCCommands /loc %cd% /dll %cd%

REM Esborrar com a comanda d'usuari (User command)
REM "C:\Program Files (x86)\GeoMedia Professional\Program\InstallUsrCmd" /u ModalCmd.dll ModalCmd.ini
```

On notar que es té l'opció de registrar com a comanda d'aplicació, de manera que tots els usuaris de l'estació de treball hi tinguin accés, o només com a comanda d'usuari, on únicament l'usuari en qüestió podrà executar la comanda. Per al primer cas es requereixen drets d'administrador. A l'exemple s'està registrant com a comanda d'aplicació.

El fitxer *CalculadoraSolar.xml* inclòs s'ha de modificar introduint el nom del fitxer DLL, així com una breu descripció de l'aplicació. Un cop fet això, només cal escollir l'aplicació dins de GeoMedia fent servir l'opció de menú *Ver→Barras de herramientas→Personalizar*, i al nou diàleg, a sota de *Categorías* escollim *Personalizado*. A les icones que apareixen a la dreta hauria de ser-hi la que es correspon a l'aplicació *Calculadora solar*. Prement sobre la icona es pot arrossegar fins la barra de tasques. Si tot ha anat correctament, la finestra de la Figura 21 (a) apareix quan es pren el botó que s'ha posat a la barra.

A l'inici de la implementació es va crear una aplicació externa a GeoMedia però fent servir les utilitats d'aquesta. Això va permetre en un primer moment familiaritzar-se millor amb les funcionalitats bàsiques de GeoMedia. Seguidament es va passar a fer servir la comanda personalitzada. Amb l'objectiu de poder depurar durant l'execució dins de GeoMedia, cal fer seguir les següents passes:

1. Compilar en mode *Debug* i generar el fitxer DLL prement F7.
2. Si encara no s'ha registrat la DLL, registrar-la fent servir l'*script RegisterModalCmd.bat*.
3. Obrir GeoMedia.
4. Si dins de GeoMedia no s'ha afegit la icona de l'aplicació a la barra de tasques o al menú, fer-ho com s'ha descrit anteriorment.
5. Dins de Visual Studio, escollir l'opció de Menú *Depurar→Associar a Proceso*. En la nova finestra que apareix cal escollir *GeoMedia.exe*.

Quan s'executi l'aplicació, Visual Studio es parará al primer punt d'interrupció que s'hagi posat, podent així depurar el codi com de costum.

4.2 Consideracions sobre GeoMedia

Un dels aspectes on més temps s'ha invertit és en entendre i aprendre com funciona la interfície amb GeoMedia. El fitxer d'ajuda *GM_auto61.chm* adjunt amb la distribució de GeoMedia és on estan descrites totes les possibilitats. El problema rau en que les funcionalitats són moltes i els noms no són sempre gaire clars, i per tant no sempre és fàcil trobar el que es necessita. Hi ha força exemples, però moltes vegades són insuficients. A més, la informació o existència d'exemples a Internet és limitada, i de

fet, molts dels que s'han pogut trobar corresponen a treballs fi de carrera realitzats a la UOC.

En qualsevol cas, a continuació es comenten les funcionalitats i interfícies més importants de GeoMedia que s'han fet servir a l'aplicació:

- Funcions per a l'administració de bases de dades. Inclou principalment els següents objectes i classes:
 - *objGeoApp.Document.Connections(1).Database* del tipus *GDatabase*: objecte per a l'accés a la base de dades oberta durant l'execució del codi a GeoMedia. En aquest cas s'està assumint que la base de dades amb les gemoetries d'interès es troba a la primera connexió.
 - *PCClient.GeometryStorageService*: classe que permet realitzar diferents transformacions. És important per a poder llegir la geometria d'un registre (funció *StorageToGeometry*) i viceversa, és a dir, guardar una geometria a un registre (funció *GeometryToStorage*).
 - La creació/modificació de taules/registres està ben exemplificada dins el fitxer d'ajuda de GeoMedia esmentat anteriorment, i per tant no es descriuen aquí les funcionalitats.
- Funcionalitat per a la transformació de coordenades. Principalment s'ha fet servir l'objecte *objGeoApp.ActiveWindow.MapView.CoordSystemsMgr.CoordSystem* que és del tipus *PCSS.CoordSystem* per a poder fer conversions de coordenades de la projecció de la finestra de mapa actual a qualsevol altra i viceversa. La funció en concret és *TransformPoint*, que en el marc del TFC s'ha fet servir per a transformar el punt en la projecció donada a latitud/longitud per a poder fer els càlculs de posició del sol.
- La funció *PointInPolygon*: la funció torna el valor -1 si el punt no és dins el polígon, 1 si és a dins, i 0 si és a sobre. Cal tenir en compte que, aparentment, a sobre vol dir a sobre en el pla X-Y, de manera que si el polígon és vertical a aquest pla (per exemple, una cara d'un edifici) es poden obtenir resultats inesperats. Cal tenir aquest comportament en compte per evitar resultats erronis. Per aquest motiu, dins l'algorisme de *ray tracing* abans de cridar *PointInPolygon* s'avalua primer si el punt on s'intersecciona el pla és més alt que la màxima alçada del polígon. En cas afirmatiu el polígon no bloqueja el raig de sol.

Finalment, només destacar que les interfícies i possibilitats de programar amb GeoMedia són moltes, però donada la manca de temps no s'ha pogut explotar al màxim aquest potencial. Hi ha certes funcionalitats que es poden programar i que estalviarien feina a l'usuari, com per exemple la creació de les *metadades* dins la base de dades per a que GeoMedia reconegui els continguts, o la generació automàtica de la llegenda per a la visualització de les dades un cop acabat el càlcul. En la següent secció s'explica, entre d'altres coses, com l'usuari ha de fer aquestes tasques manualment.

4.3 Manual d'usuari de la Calculadora Solar

Un cop instal·lada l'aplicació –després d'haver registrat la comanda personalitzada i de posar-la al menú de GeoMedia– cal seguir les següents passes per a utilitzar-la.

En primer lloc, cal carregar un fitxer amb les geometries dels edificis. L'aplicació assumeix que les dades d'entrada són els fitxers SHP descarregats de l'Institut Cartogràfic de Catalunya (ICC) [6]. El fitxer d'interès és el denominat "pp" (en el cas de la ciutat escollida, Sant Esteve Sesrovires, el nom és *bt5mv20sh0f282121pp1r050.shp*), que conté els polígons que defineixen els edificis. Primer cal crear una nova connexió ArcView per a tenir accés al fitxer SHP. Seguidament s'importa el fitxer SHP dins de la taula Access del GeoWorkspace tal i com s'ha descrit a la Secció 1.5.1. Si no hi ha cap taula Access oberta, llavors es crea una amb l'opció *Almacén→Almacén nuevo*. Per a importar la taula cal fer servir l'opció *Almacén→Sacar a clases de entidad* i seguidament escollir la taula amb el mateix nom que el fitxer desitjat. La taula creada dins la base de dades Access manté el nom original del fitxer SHP, i per tant cal canviar-lo a *Poligons2D* (*Almacén→Definición de clases de entidad→Editar*), ja que aquest és el nom que l'eina desenvolupada assumeix.

Un cop la taula *Poligons2D* ha estat creada, ja es pot fer servir la funcionalitat *Afegir Alçades*. Les opcions es poden veure a la Figura 21 (a). L'alçada es pot afegir o bé de manera aleatòria, on es poden escollir les alçades mínima i màxima, o bé una alçada constatat per a tots els edificis. Prement el botó *Afegir Alçades* s'afegiran les alçades i es crearà la taula *Poligons3D* amb els nous polígons. Per a visualitzar-la, primer cal crear la informació de *metadata* de la taula. Això es pot fer utilitzant l'aplicació de externa de GeoMedia *Utilidades de bases de datos*. Un cop oberta, s'escull la base de dades Access amb la que s'està treballant, i en el nou menú s'escull *Insertar metadatos de clase de entidad*. Després d'escollir la taula *Poligons3D*, s'accepta el sistema de coordenades per defecte i es prem acceptar. Cal dir que aquest pas de la inserció de les *metadades* només cal fer-lo un cop per base de dades. Quant a control d'errors, l'aplicació cancel·la l'operació si no troba la taula *Poligons2D*, i si la taula *Poligons3D* ja existeix, llavors s'avisa a l'usuari que si es continua s'esborraran els registres existents.

Després de reobrir la base de dades a GeoMedia (*Almacén→Conexiones*, seleccionar base de dades i opció *Reabrir conexión*), ja es pot escollir la taula per a incloure-la a la llegenda. La Figura 22 mostra una captura de pantalla amb les opcions per a poder visualitzar correctament les alçades dels edificis. Aquesta opció s'activa clicant amb el botó dret sobre l'entrada de la llegenda i escollint l'opció *Propiedades de estilo*.

Ara que la taula *Poligons3D* existeix, es pot fer servir qualsevol de les altres funcionalitats. La Figura 21 (b) mostra les opcions de la funcionalitat per al càlcul del percentatge de radiació solar per a un dia i hora donats. L'usuari ha d'introduir la data i hora desitjats, on l'hora s'introdueix en format 24 hores i en temps universal (UT), i com a opcions de càlcul es poden fixar el radi màxim per a descartar aquells edificis que estiguin més enllà d'aquesta distància; el número de punts per costat sobre la

coberta i la façana a avaluar, així com l'algorisme a utilitzar per fer el càlcul (*ray tracing* o proporció de triangles).



Calculadora Solar v1.0

Afegir Alçades | Percentatge solar | Hores de sol

Taula amb els polígons 2D: Polígons2D

Tipus d'alçada: Alçada aleatòria

Alçada mínima: 5

Alçada màxima: 30

Alçada constant: 30

Cancel·lar Afegir Alçades

Calculadora Solar v1.0

Afegir Alçades | Percentatge solar | Hores de sol

Selecció de data: sábado, 1 de junio de 2013

Hora: 12 Minuts: 00

Opcions de càlcul:

Radi de cerca d'edifici [metres]: 50

Punts / costat: 4

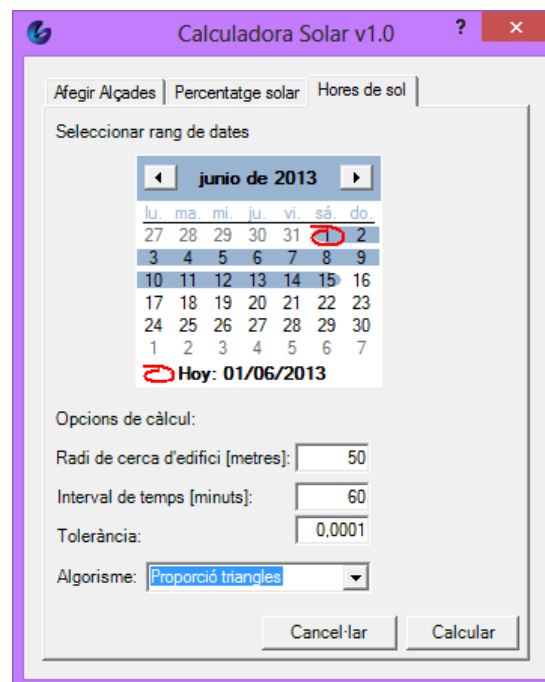
Tolerància: 0.0001

Algorisme: Ray tracing

Cancel·lar Calcular

(a)

(b)



Calculadora Solar v1.0

Afegir Alçades | Percentatge solar | Hores de sol

Selecció de rang de dates

junio de 2013

lu	ma	mi	ju	vi	sá	do
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Hoy: 01/06/2013

Opcions de càlcul:

Radi de cerca d'edifici [metres]: 50

Interval de temps [minuts]: 60

Tolerància: 0.0001

Algorisme: Proporció triangles

Cancel·lar Calcular

(c)

Figura 21. Disseny de l'aplicació per a les diferents funcionalitats. (a) Per a l'opció d'afegir alçada als polígons 2D. (b) Per a calcular el percentatge de llum solar sobre els edificis generats a l'opció anterior. (c) Per a calcular el número total d'hores d'un rang de dates donades.

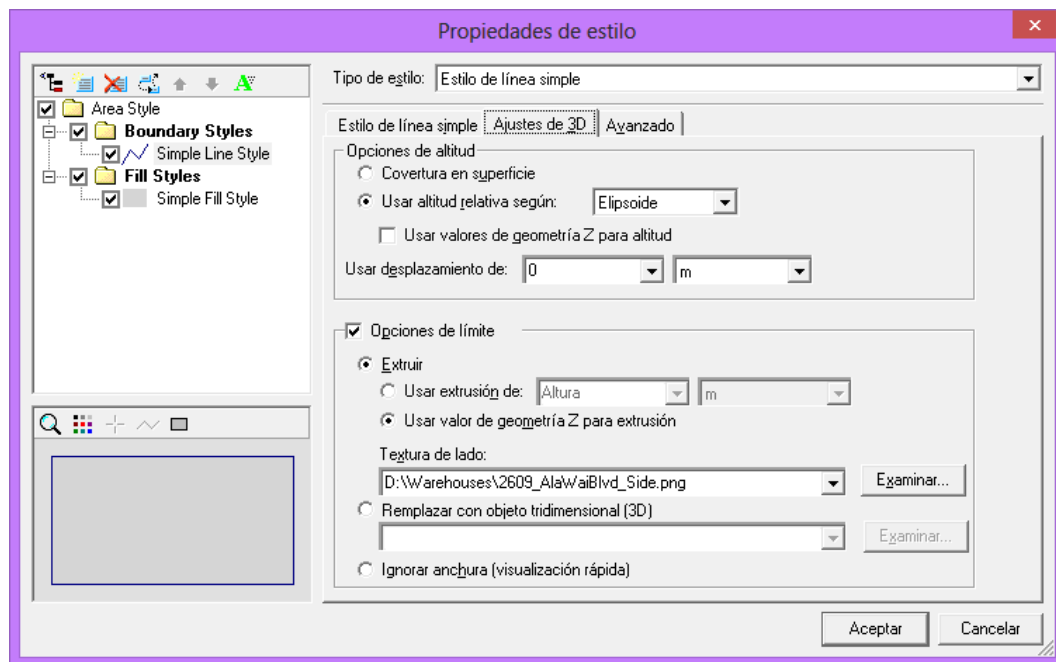


Figura 22. Propietats d'estil per a poder visualitzar els polígons creats en 3D.

L'opció de tolerància té a veure amb el discutit a la secció 2.3.3 sobre la tolerància de GeoMedia a l'hora d'avaluar distàncies. Es recomana deixar el valor per defecte de GeoMedia, però s'ha deixat per experimentar. La Figura 21 (c) mostra la interfície de la tercera funcionalitat, on cal definir un període de temps o escollir un únic dia. A més, s'inclou l'opció de definir l'interval de temps entre càlculs dins el mateix dia en minuts. Un valor més gran reduirà el temps de càlcul però serà menys precís. L'opció de punts per costat no apareix, ja que a la coberta s'agafen directament tots els punts del polígon, i per cada pis només es crea un punt al mig de la façana.

Per a les funcionalitats de percentatge solar i hores totals l'aplicació verifica que no existeixin registres amb la data i hora demanades, o amb el període de dies demanat, respectivament. Si existeixen s'avisarà a l'usuari que si es continua s'esborraran tots els registres corresponents.

Després d'executar cadascuna de les funcionalitats, es generen les taules *PercentatgeSolar* i *HoresTotals* per al percentatge solar i el número d'hores totals, respectivament. Després del càlcul, caldrà de nou inserir la informació *metadata* de la taula generada tal i com s'ha descrit per a la taula *Poligons3D*. La visualització a la llegenda es farà d'igual manera que amb la taula *Poligons3D*, però seguidament es clicarà amb el botó dret sobre l'entrada de la llegenda i s'escollirà l'opció *Propiedades de entrada de leyenda*. Cal escollir el tipus *Temático de rangos* per a poder configurar els valors desitjats. La Figura 23 mostra un exemple pel cas de percentatge solar, on els valors van de 0% a 100% en increments del 10%.

Donat que els càlculs es poden fer per a diferents dates o, com és el cas del número d'hores totals, per a diferents pisos, cal poder filtrar els continguts de la taula per a visualitzar correctament la informació. Això es pot fer mitjançant l'opció *Análisis→Consulta de atributo*. La Figura 24 mostra un exemple per a la funcionalitat d'hores totals, on s'estan filtrant els registres corresponents a la coberta. En acceptar

es crearà automàticament la llegenda, la visualització de la qual caldrà modificar de manera similar als casos anteriors per a poder visualitzar les dades incloent l'alçada dels edificis.

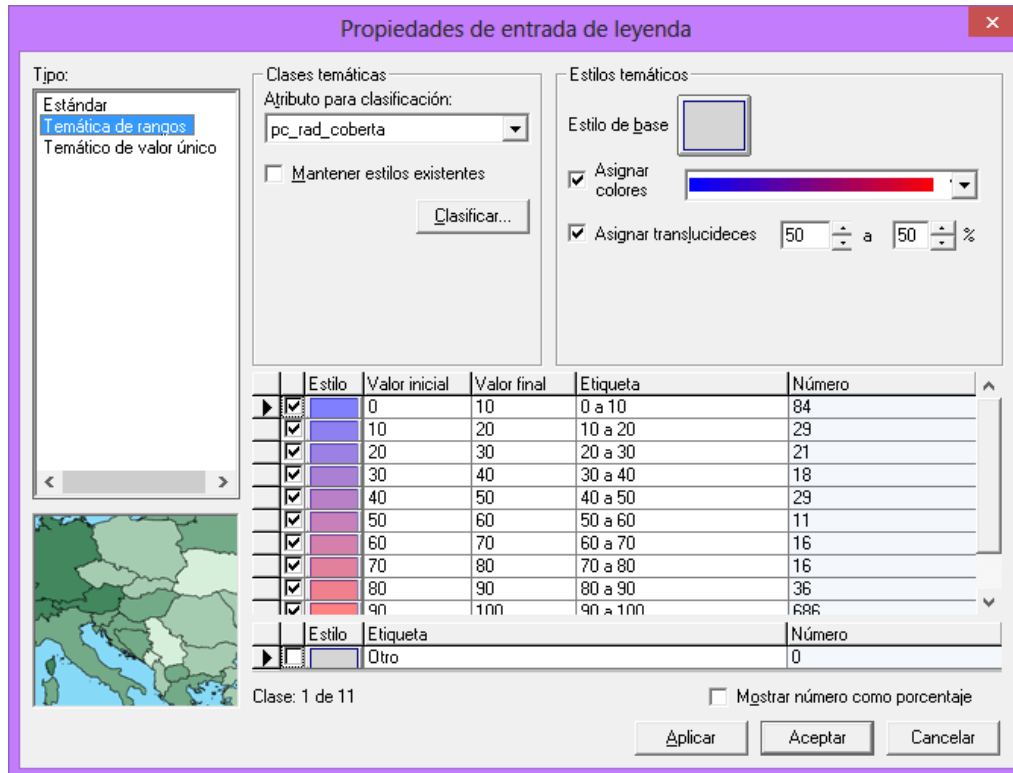


Figura 23. Configuració per a la visualització de la informació en forma de rang de dades. En aquest cas es tracta del percentatge solar i els valors van de 0% a 100% en increments del 10%.

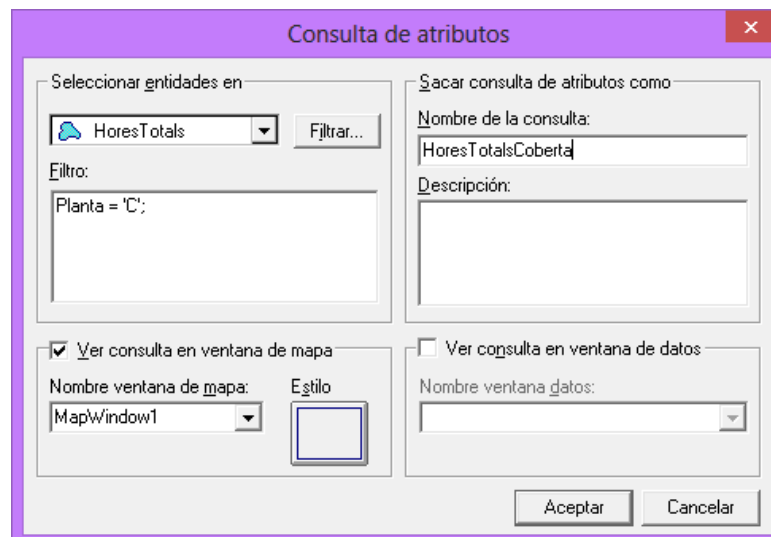


Figura 24. Exemple de com filtrar els registres d'una taula per a poder visualitzar la informació desitjada. L'exemple està filtrant només les hores totals de la coberta.

5. RESULTATS

En aquest capítol es presenten els resultats obtinguts amb l'aplicació implementada. A mode d'exemple s'ha escollit el municipi de Sant Esteve Sesrovires, situat a la comarca del Baix Llobregat. Els fitxers de formes i l'ortofoto s'han descarregat de la pàgina web de l'Institut Cartogràfic de Catalunya (ICC) [6].

5.1 Afegir alçades

La primera de les funcionalitats és afegir alçades als polígons 2D descarregats de l'ICC per a obtenir els polígons 3D. La Figura 25 mostra el resultat, on les alçades s'han afegit en aquest cas de manera aleatòria entre 5 m i 30 m.

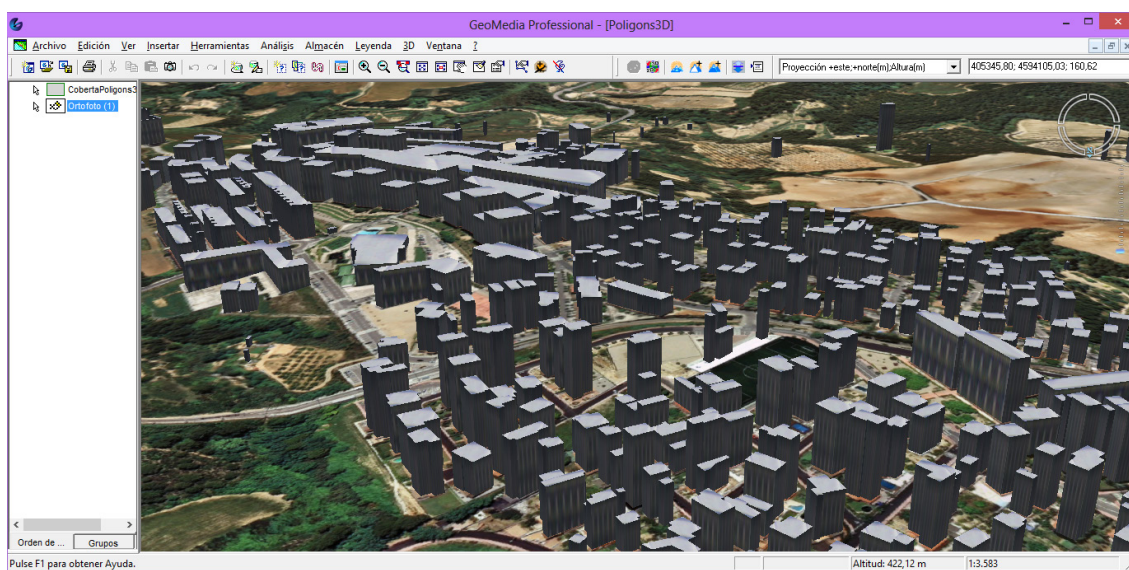


Figura 25. Visualització dels polígons 3D obtinguts amb la funcionalitat d'afegir alçades al municipi de Sant Esteve Sesrovires. Les alçades s'han afegit aleatòriament amb valors que van de 5 a 30 metres. S'han afegit textures als costats i a la coberta. L'ortofoto amb 25 cm de resolució descarregada de l'ICC es mostra a sota.

5.2 Percentatge solar

El càlcul del percentatge solar es mostra a continuació per als dos algorismes implementats. Per a tots dos la data escollida és l'1 de juny de 2013 a les 17:00 UT (19:00 locals). A aquesta hora, el sol es troba a un angle azimut de 280° respecte al nord i un angle d'elevació de 23° . La Figura 26, obtinguda a partir de la plana web [14]¹³, mostra la direcció del sol d'aquesta hora, així com la de l'ortus i l'ocàs. Les opcions pel càlcul que s'han escollit són d'un radi màxim de 100 metres i 8 punts per costat.

¹³ <http://suncalc.net>

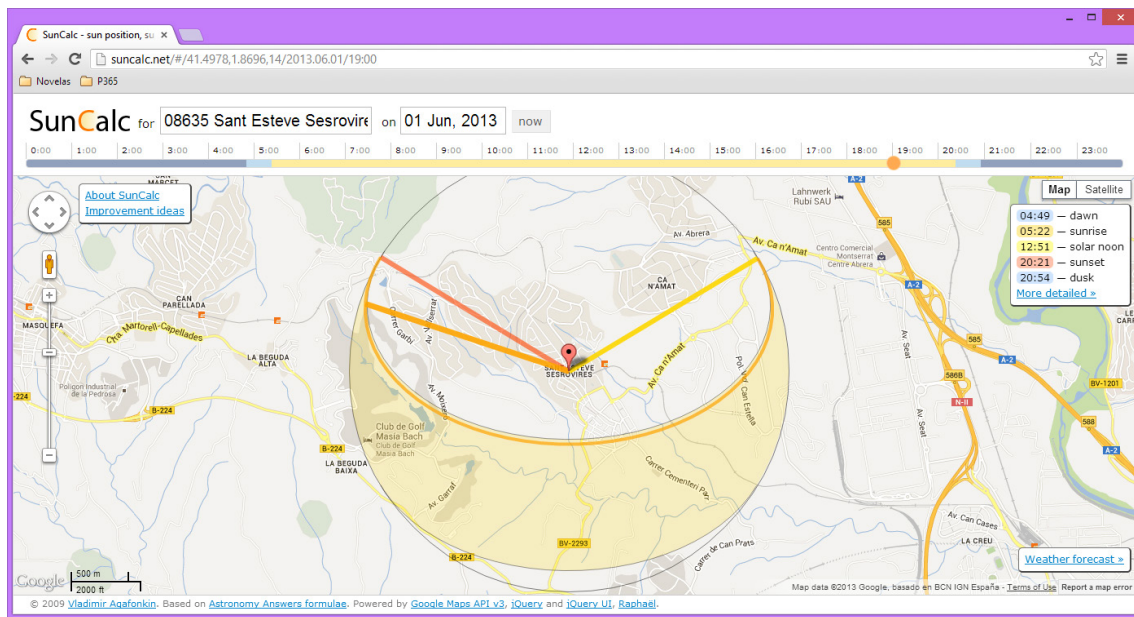


Figura 26. Posició del sol el dia 1 de juny a les 17:00 UT (19:00 hora local). L'azimut és 280° i l'angle d'elevació són 23°. Les línies groga i vermella indiquen l'ortus i l'ocàs, respectivament. Font: [14].

La Figura 27 mostra el percentatge de sol calculat amb els dos algorismes per a una porció de l'escena. La direcció d'observació de la càmera correspon aproximadament a la del sol (amb el sol a l'esquena de la càmera). La visualització per rangs s'ha fet en increments de 10%, i notar que la llegenda inclou el número d'edificis de l'escena completa dins de cada rang entre parèntesis. Com era d'esperar, els resultats amb proporció de triangles són més pessimistes en el sentit que tendeixen a donar ombra abans que llum, donat que el punt més proper sempre és el mateix i, per tant, en la pràctica no tenen en compte la forma de l'edifici interferent, tal i com s'ha comentat a la Secció 2.3.4. En qualsevol cas, per a tots dos algorismes la majoria de cobertes estan al sol, però això depèn evidentment de la distribució d'alçades de l'escena.

Els resultats de percentatge de llum sobre la façana s'han calculat només amb *ray tracing*, tal i com ja s'ha esmentat anteriorment. El resultat apareix a la Figura 28, on les façanes estan indicades amb una línia negra. Si mirem la llegenda i comparem el número d'edificis per rang amb els que s'obtenen en el cas de la coberta, és evident que hi ha més edificis amb la façana a l'ombra, on el sol és bloquejat en molts casos per la coberta i cares del mateix edifici.

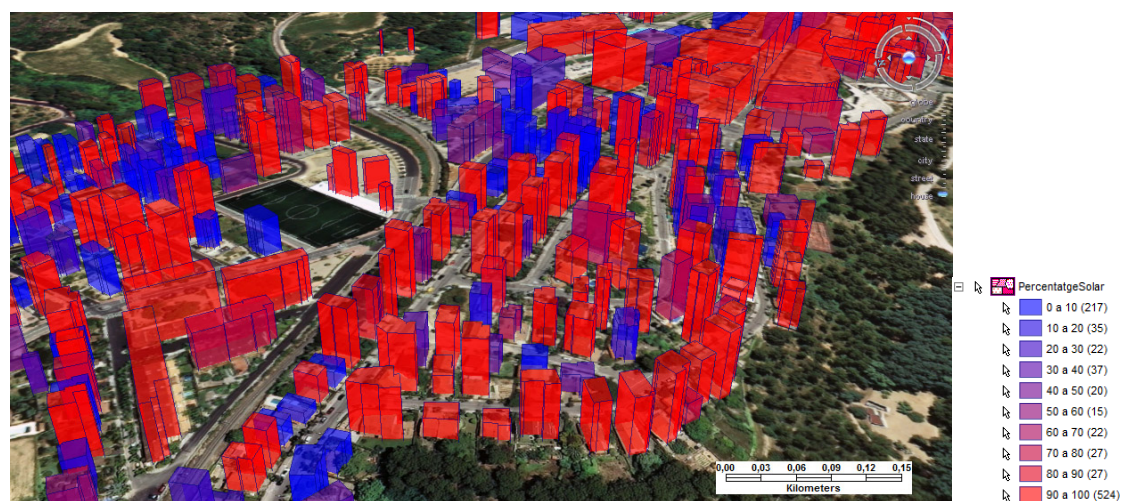
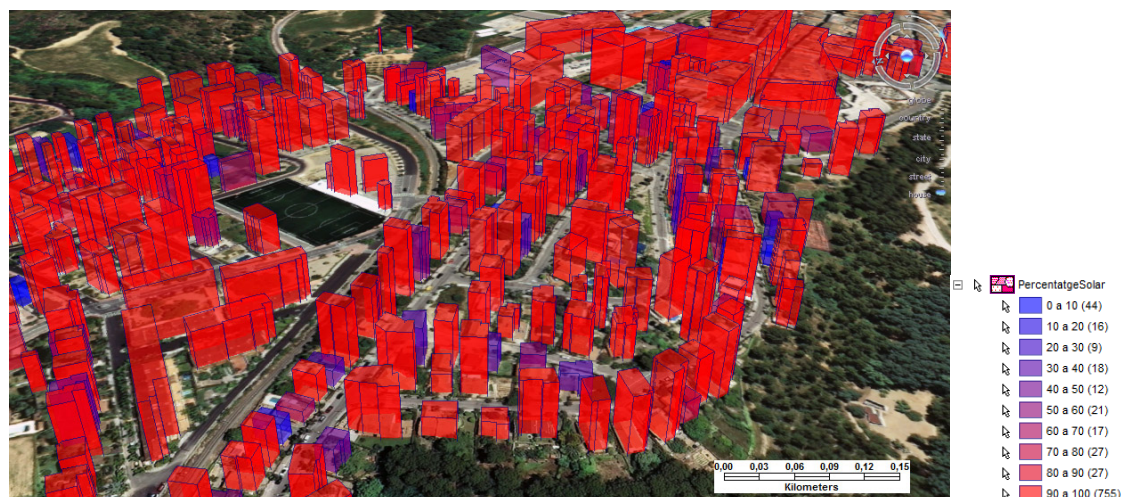


Figura 27. Percentatge de llum solar sobre la coberta al municipi de Sant Esteve Sesrovires a les 19:00 hores locals el dia 1 de juny de 2013. Calculat amb (amunt) *ray tracing* i (avall) proporció de triangles. La direcció d'observació de la càmera correspon aproximadament a la del sol.

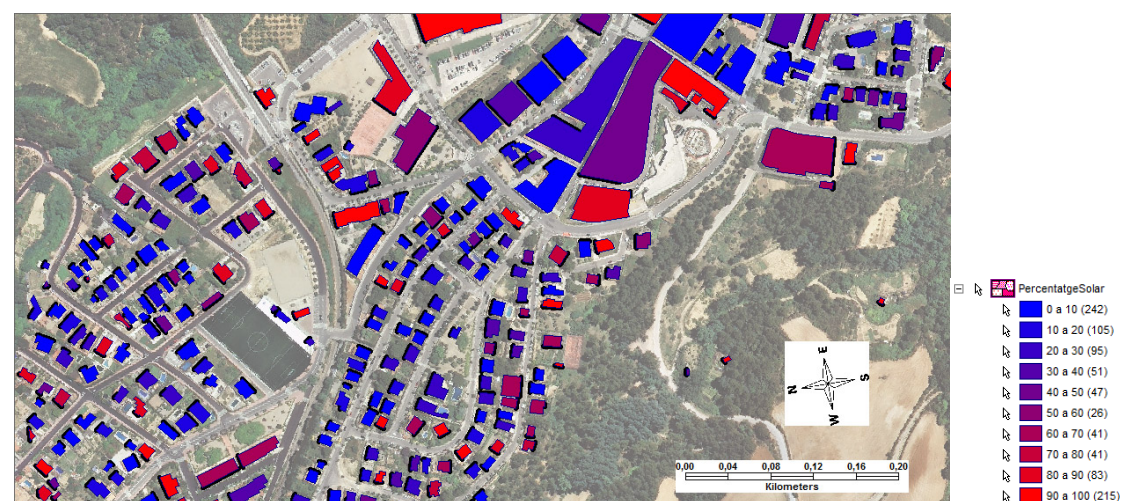


Figura 28. Percentatge de llum solar sobre la façana dels edificis al municipi de Sant Esteve Sesrovires a les 19:00 hores locals el dia 1 de juny de 2013. Els rajos de sol venen aproximadament de la part inferior. Càlcul realitzat amb *ray tracing*. Les façanes de cada edifici estan indicades amb una línia negra sòlida. L'ortofoto es mostra a sota amb transparència.

5.3 Hores totals

L'última funcionalitat implementada és el càlcul de les hores totals a la coberta i als diferents pisos per a un període donat. Com a mode d'exemple s'ha escollit de nou el dia 1 de juny de 2013, on el número d'hores de sol son aproximadament 15, i s'ha fet servir un radi de cerca d'edificis de 100 metres i un interval de temps de 60 minuts. Com ja s'ha esmentat, la Figura 26 mostra la direcció dels rajos de sol per a l'ortus i l'ocàs. Els algorismes han trigat 78 minuts i 55 minuts per a *ray tracing* i proporció de triangles, respectivament. La Figura 29 mostra els dos resultats, on de nou es pot veure que proporció de triangles és més pessimista que *ray tracing*.

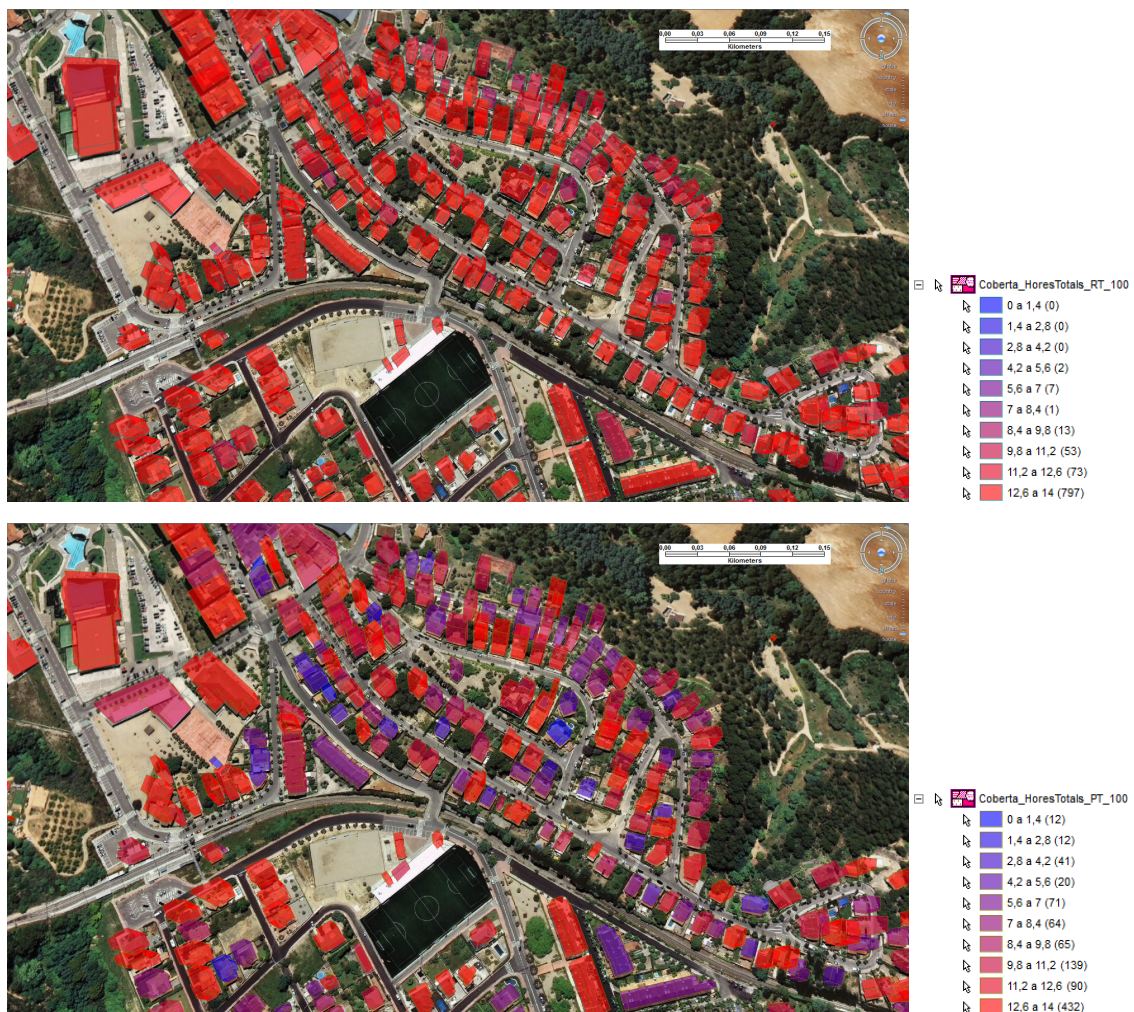


Figura 29. Número d'hores totals sobre la coberta pel dia 1 de juny de 2013 al municipi de Sant Esteve Sesrovires. Calculat amb (amunt) *ray tracing* i (avall) proporció de triangles. Càmera orientada de nord a sud (nord cap avall).

La Figura 30 mostra les hores totals de la planta baixa i de la tercera planta (edificis de com a mínim 12 metres) per al cas de *ray tracing*. Es poden apreciar diferències respecte a la coberta. La figura també inclou una captura amb els polígons corresponents amb les façanes per a poder interpretar millor els resultats, per exemple, les façanes orientades al nord tenen un número significativament més petit d'hores de sol.

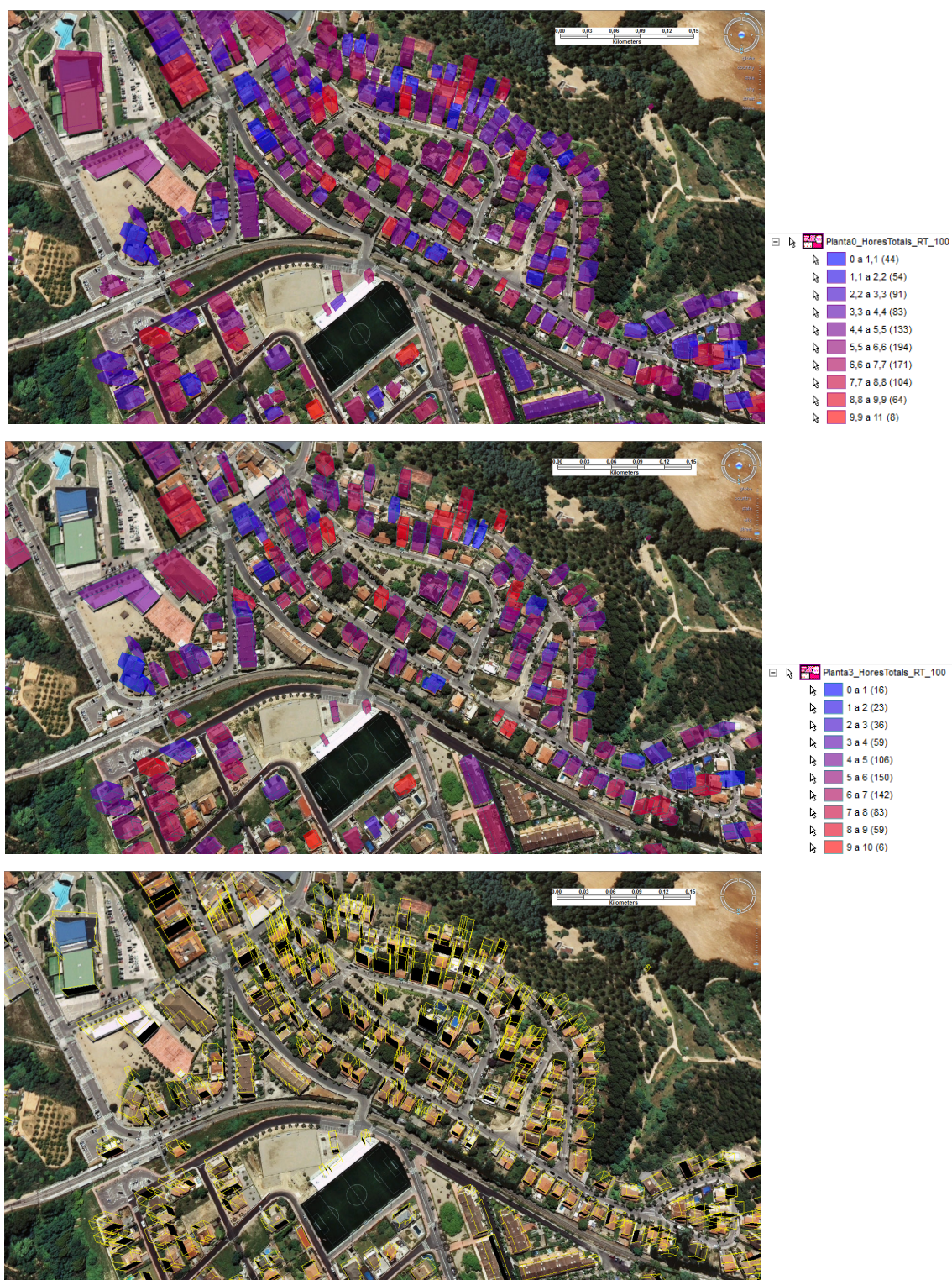


Figura 30. Número d'hores totals sobre (amunt) planta baixa i (mig) tercera planta pel dia 1 de juny de 2013 al municipi de Sant Esteve Sesrovires. (Avall) Representació de les façanes (polígons negres).
Calcul realitzat amb *ray tracing*.

5.4 Comparativa de Temps

A continuació es presenta una comparativa dels temps de càlcul dels dos algorismes utilitzats pel cas del percentatge solar per a un dia i hora donats. El càlcul es va fer de nou amb el municipi de Sant Esteve Sesrovires contenint un total de 946 edificis. La data seleccionada és l'1 de juny de 2013, a les 17:00 UT (19:00 locals). La Taula 2 mostra el resultat dels temps de càlcul modificant els dos paràmetres possibles dels algorismes, a saber, el radi màxim dins del qual els edificis s'han d'avaluar, i el número de punts per costat a l'hora de definir els punts a sobre la coberta o la façana. La darrera columna mostra el resultat normalitzat respecte al resultat més ràpid, és a dir, pel cas d'un radi de 50 m i 4 punts per costat, on la normalització és fa per a cada mètode independentment. Aquest factor, juntament amb el temps mig per edifici, permet fer una estimació del temps en funció dels paràmetres i del número d'edificis per a un escenari determinat. Els càlculs s'han fet amb un portàtil ASUS de la sèrie R500V amb processador IntelCore i5-3210 @ 2.5 GHz i 8 GB de memòria RAM.

Algorisme	Radi [metres]	Punts/Costat	Temps	Temps/edifici	Factor
<i>Ray tracing</i>	50	4	5 min 17 s	0.33 s	1
<i>Ray tracing</i>	100	4	9 min 20 s	0.59 s	1.76
<i>Ray tracing</i>	50	8	10 min 51 s	0.68 s	2.05
<i>Ray tracing</i>	100	8	24 min 7 s	1.52 s	4.56
Proporció triangles	50	4	4 min 3 s	0.25 s	1
Proporció triangles	100	4	5 min 16 s	0.33 s	1.30
Proporció triangles	50	8	6 min 45 s	0.42 s	1.66
Proporció triangles	100	8	10 min 15 s	0.65 s	2.53

Taula 2. Comparativa del cost computacional pels dos algorismes de càlcul de llum solar implementats amb diferents opcions de configuració.

La comparativa de temps de càlcul deixa clar que l'algorisme de *ray tracing* té un cost computacional més alt que el de proporció de triangles, especialment quan s'augmenta el número de punts per costat. Això és raonable degut a que *ray tracing* avalua totes les cares dels edificis interferents, mentre que proporció de triangles només avalua la coberta. Tot i així, no és un factor cinc més lent per aquest fet. D'altra banda, també cal considerar que el percentatge de llum sobre la façana sempre es calcula fent servir *ray tracing*, i això afegeix al càlcul un temps addicional no negligible.

Considerant ara cada algorisme per separat, seria d'esperar que un augment del número de punts per un factor 2 resultés en un factor 4 en el temps de càlcul, ja que en principi s'estan avaluant quatre cops més punts per polígon (recordar que el valor que introdueix l'usuari és el número de punts per costat). Això indica que hi ha una càrrega computacional important a més de la dels càlculs dels algorismes de *ray tracing* i proporció de triangles, i que per tant és independent dels paràmetres indicats. Un exemple seria l'avaluació dels edificis (funció *EdificiValid*), que sempre tindrà el mateix cost. De cara a una optimització seria, doncs, desitjable caracteritzar millor cada part de l'aplicació.

Finalment, només afegir que a part del municipi de Sant Esteve Sesrovires també s'ha analitzat una part de l'Eixample de Barcelona. Donat que en aquest segon cas la distribució dels edificis és força diferent, amb edificis molt més grans, el paràmetre del radi es va ampliar fins als 300 metres. En qualsevol cas, els resultats eren comparables als mostrats aquí.

6. CONCLUSIONS I LÍNIES FUTURES

En aquest treball fi de carrera s'ha desenvolupat una aplicació informàtica dins de la temàtica dels sistemes d'informació geogràfica que consisteix en el càlcul de la quantitat de la llum solar rebuda pels edificis d'una ciutat. L'aplicació s'ha desenvolupat sota l'entorn de GeoMedia, una plataforma comercial per a l'avaluació d'eines SIG, i ha estat programada en Visual Basic fent servir l'entorn Visual Studio 2012.

Les funcionalitats que s'han implementat són, en primer lloc, la generació dels polígons 3D dels edificis a partir d'una alçada donada i dels polígons 2D dels mateixos. La segona funcionalitat consisteix en el càlcul del percentatge de llum solar sobre la coberta i la façana d'un edifici. Finalment, l'última funcionalitat és el càlcul d'hores de llum sobre la coberta i els diferents pisos d'un edifici d'un període de temps donat.

Cada funcionalitat ha passat per les etapes típiques de l'enginyeria de programari: una primera etapa d'estudi, seguida d'una de disseny, i finalment una darrera d'implementació i depuració. Donada la complexitat del problema a resoldre s'ha posat especial èmfasi a l'etapa d'estudi, on s'han desenvolupat els algorismes que s'han fet servir per a resoldre la geometria del problema, la qual consisteix principalment en saber si un punt d'un edifici rep o no la llum directa del sol, on els edificis circumdants són els que s'han d'avaluar per a veure si fan o no ombra. Els algorismes que normalment solucionen aquest tipus de geometria són els anomenats de traçat de rajos, o *ray tracing* en anglès. Donades les geometries presents, que consisteixen en polígons senzills, s'ha optat en implementar un algorisme simple de *ray tracing* basat en el càlcul de la intersecció entre una recta i un pla. A més, en base a un suggeriment del director del projecte, també s'ha implementat un segon algorisme que explota la tècnica anomenada proporció de triangles.

Adicionalment a la part d'enginyeria del projecte, el segon aspecte important on s'ha invertit una gran part de l'esforç ha estat en conèixer GeoMedia i GeoMedia 3D, i sobre tot en familiaritzar-se amb la programació sobre GeoMedia. Aquest darrer punt ha requerit força temps donada la informació limitada disponible. En qualsevol cas, és evident que es tracta d'una qüestió de temps, ja que sempre s'ha pogut trobar la solució adient, i si alguns aspectes no s'han acabat de polir, més que per manca de coneixements, ha estat per aquesta manca d'hores al final del dia.

L'aplicació desenvolupada s'ha validat amb el municipi de Sant Esteve Sesrovires, situat a la comarca del Baix Llobregat. Durant el desenvolupament i implementació dels algorismes s'han considerat geometries senzilles que han permès optimitzar i depurar el codi. L'avaluació visual dels resultats amb les geometries reals també ha servit com a criteri per a depurar els algorismes. En aquest sentit, s'han desenvolupat i implementat satisfactòriament dos algorismes gairebé equivalents per a calcular si un punt rep la llum directa del sol. L'algorisme de *ray tracing* és més exacte que el de proporció de triangles, però amb un cost computacional més elevat, amb un factor tres en el pitjor dels casos.

D'altra banda, també s'ha fet un esforç en el disseny de l'aplicació per a maximitzar la reutilització de codi per les dues aplicacions del càlcul solar: percentatge solar i hores totals. En primer lloc, totes corren dins el mateix bucle principal, de manera que cada

classe s'ha definit com una sub-classe de la mateixa classe. L'ús del polimorfisme dins la programació orientada a objectes ha permès una implementació eficient adaptada a cada funcionalitat. D'igual manera, el disseny dels algorismes de *ray tracing* i proporció de triangles s'ha fet explotant polimorfisme, per d'aquesta manera també facilitar l'ampliació de l'aplicació en un futur amb d'altres algorismes similars.

D'acord amb tot l'exposat es pot concloure, doncs, que els objectius plantejats al començament del treball fi de carrera s'han assolit satisfactòriament. Primer, s'ha realitzat un projecte real de programació passant per totes les etapes d'enginyeria de programari (estudi, disseny, implementació); segon, s'han sabut plantejar, analitzar i solucionar correctament una sèrie de problemes relacionats amb la temàtica SIG; finalment, s'ha realitzat la implementació sobre un programari nou (GeoMedia), fet que implica que s'han assimilat una sèrie de competències i coneixements que no es tenien en començar el TFC.

Arribat aquest punt, però, cal pensar en les possibles extensions del treball realitzat. L'eina desenvolupada es pot millorar en certs aspectes. D'una banda tenim la generació dels polígons 3D, on caldria integrar dins l'aplicació un accés a una font de dades amb informació referent a l'alçada real dels edificis i la localització de la façana. Quant a la interacció per part de l'usuari, és evident que certs aspectes es podrien automatitzar a partir les interfícies que GeoMedia ofereix, per exemple, possibilitar la selecció d'edificis als quals aplicar l'eina, inserció automàtica de la informació *metadata*, inserció automàtica de la llegenda, etc. Respecte a la part algorísmica, hi ha potencial per reduir el temps de càlcul. En aquest sentit, seria força directe la paral·lelització dels algorismes amb múltiples *threads*. Però on probablement es podria guanyar fins a un ordre de magnitud o més en el temps de càlcul seria a través de l'ús de targetes gràfiques o GPUs (*graphics processor unit* en les seves sigles en anglès). Aquest maquinari està precisament dissenyat per resoldre el tipus de geometries que s'han plantejat en aquest TFC de manera eficient. Actualment existeixen diverses llibreries de lliure distribució com CUDA [15][16]¹⁴ o OpenCL [17][18]¹⁵ que es podrien integrar dins l'aplicació desenvolupada.

Finalment, cal pensar en l'objectiu original d'aquest projecte. Com ja s'ha plantejat en la introducció, aquesta eina SIG té l'objectiu d'ajudar en el disseny i planificació de noves construccions per minimitzar el consum de recursos energètics. Així, per exemple, de les dades obtingudes es podrien derivar les localitzacions òptimes per posar panells solars a sobre els edificis i, fins i tot, quina inclinació haurien de tenir per maximitzar-ne l'eficiència.

En definitiva, els recursos naturals de la Terra són limitats, i, per tant, és el nostre deure com a habitants seus no només respectar-la, sinó també cuidar-la i preservar-la. Optimitzar els recursos energètics és un pas necessari. Aquest TFC és un *petit* pas en la direcció correcta.

¹⁴ <http://es.wikipedia.org/wiki/CUDA>

¹⁵ <http://ca.wikipedia.org/wiki/OpenCL>

BIBLIOGRAFIA

- [1] **Bataller Díaz, A.** (2007) "Gestió i desenvolupament de projectes" FUOC
- [2] <http://geospatial.intergraph.com/products/GeoMedia/Details.aspx> (última revisió : 01/06/2013)
- [3] <http://www.microsoft.com/visualstudio> (última revisió : 01/06/2013)
- [4] <http://www.nomagic.com/products/magicdraw.html> (última revisió : 01/06/2013)
- [5] <http://office.microsoft.com> (última revisió : 01/06/2013)
- [6] Institut Cartogràfic de Catalunya <http://www.icc.cat> (última revisió : 01/06/2013)
- [7] **Institut Cartogràfic de Catalunya** (2011) Especificacions Tècniques de la base topogràfica de Catalunya 1:5000 (BT - 5M) v2.0 ICC
- [8] <http://www.intergraph-forum.de/forum/posts/list/141.page> (última revisió: 01/06/2013)
- [9] http://en.wikipedia.org/wiki/Spherical_coordinate_system (última revisió: 01/06/2013)
- [10] http://en.wikipedia.org/wiki/Line-plane_intersection (última revisió: 01/06/2013)
- [11] <http://www2.arnes.si/~gljsentvid10/sun.html> (última revisió: 01/06/2013)
- [12] <http://asa.usno.navy.mil/> (última revisió: 01/06/2013)
- [13] http://en.wikipedia.org/wiki/Horizontal_coordinate_system (última revisió: 01/06/2013)
- [14] <http://suncalc.net> (última revisió: 01/06/2013)
- [15] http://www.nvidia.com/object/cuda_home_new.html (última revisió: 01/06/2013)
- [16] <http://es.wikipedia.org/wiki/CUDA> (última revisió: 01/06/2010)
- [17] <https://developer.nvidia.com/opencl> (última revisió: 01/06/2010)
- [18] <http://ca.wikipedia.org/wiki/OpenCL> (última revisió: 01/06/2010)
- [19] **Rodríguez Lloret, J.; Olivella, R.** (2007) "Introducció als sistemes d'informació geogràfica. Conceptes i operacions fonamentals" FUOC