

```
int z = trunc(dDJ);
double F = dDJ-trunc(dDJ);
int iAlfa= trunc((z-1867216.25)/36524.25);
int iA = ((z < 2299161) ? (z) : (z + 1 + iAlfa - trunc(iAlfa/4)));
int iB = iA + 1524;
int iC = trunc((iB-122.1)/365.25);
int iD = trunc(365.25*iC);
int iE = trunc((iB-iD)/30.6001);
```

```
double dDia = iB - iD - trunc(30.6001*iE) + F;
int iMes = ((iE<14) ? (iE-1) : (iE-13));
int iAny = ((iMes>2) ? (iC-4716) : (iC-4715));
```

```
QString sDia = QString::number(dDia);
QString sMes = mesos[iMes-1];
QString sAny = QString::number(iAny);
QString sData = (sDia+" "+sMes+" "+sAny);
```

```
return(sData);
}
```

```
double RCalculadora::intervalDies(QString sData1,QString sData2)
```

```
{
    QStringList listData1 = sData1.split(" ");
    QStringList listData2 = sData2.split(" ");
```

```
double dDia1 = listData1[0].toDouble();
QString sMes1 = listData1[1];
int iAny1 = listData1[2].toInt();
double dDiaJulia1 = diaJulia(dDia1,sMes1,iAny1);
```

```
double dDia2 = listData2[0].toDouble();
QString sMes2 = listData2[1];
int iAny2 = listData2[2].toInt();
```

MATEMÀTIQUES I ASTRONOMIA

Pseudònim: A.Lovelace

Desenvolupament d'una
simulació per calcular l'hora i el
lloc de la sortida i la posta de Sol
en un perfil topogràfic local

RESUM

En aquest treball, partint de l'interès per l'astronomia s'ha iniciat la recerca dels càlculs relacionats amb efemèrides solars per poder trobar els instants de sortida i posta del Sol en un perfil topogràfic local.

Després de buscar informació sobre l'observació dels astres des de la Terra, s'ha procedit a fer els càlculs necessaris per trobar les hores de sortida, posta i posició del Sol en qualsevol moment i des de qualsevol lloc de la Terra.

Un cop comprovats els resultats, s'han transformat les fórmules en algoritmes informàtics i s'ha creat una simulació per representar les dades de forma gràfica. Per completar la recerca s'ha mesurat un horitzó local i s'ha inclòs a la simulació. Finalment, s'han comparat els resultats amb un programa de simulacions astrals ja existent per confirmar les dades obtingudes.

ABSTRACT

In this document, based on the interest in astronomy the search for the calculations related to solar ephemeris started to be able to find the instants of sunrise and sunset in a local topographic profile.

After searching information about star observations from Earth, the calculations to find the hour of sunrise, sunset, and position of the Sun at any moment at any given place on Earth have been carried out.

Once checked the results, they have been transformed into informatic algorithms to create a simulation to represent the obtained data graphically. To complete the research a local horizon has been measured and included in the simulation. Finally, through an already existent star simulation program the results have been compared to prove the obtained data.

Índex

AGRAÏMENTS.....	3
INTRODUCCIÓ.....	4
PART TEÒRICA: RECERCA I CÀLCULS.....	6
1. PREVI A LA RECERCA.....	7
2. L'ESFERA CELESTE.....	9
2.1. SISTEMES DE COORDENADES.....	10
2.2. TRANSFORMACIÓ DE COORDENADES.....	11
3. LA DECLINACIÓ SOLAR.....	14
3.1. FRANGES HORÀRIES, DATES I CALENDARIS.....	15
3.1.1. Nombre de dies que té l'any.....	15
3.1.2. El dia julià.....	16
3.1.3. Solsticis i equinoccis.....	20
3.1.4. Nombre de dies a l'any.....	21
4. L'ANGLE HORARI.....	22
4.1. EL TEMPS SIDERAL MITJÀ DE GREENWICH (GMST o θ).....	23
4.2. ASCENSIÓ RECTA.....	24
5. CÀLCUL COMPLET DE LA POSICIÓ DEL SOL EN UN INSTANT CONCRET.....	25
PART PRÀCTICA: DESENVOLUPAMENT DE LA SIMULACIÓ.....	27
1. PANTALLA.....	28
2. FUNCIONS.....	31
2.1. FUNCIONS GENERALS.....	31
2.2. FUNCIONS DE CÀLCULS MATEMÀTICS.....	33
2.3. FUNCIONS PER INCLOURE UN PERFIL TOPOGRÀFIC LOCAL.....	34
2.3.1. Interpolació lineal.....	36
2.3.2. Mesura d'un relleu real.....	37
2.3.3. Comprovacions.....	38
CONCLUSIONS.....	44
FONTS DOCUMENTALS.....	45
ANNEX I – CODI FONT.....	47
ANNEX II – CONTINGUT DELS ARXIUS AMB LES DADES DEL RELLEU.....	77
ANNEX III – TAULES PER AL CÀLCUL DE SOLSTICIS I EQUINOCCIS.....	78

AGRAÏMENTS

Primer de tot m'agradaria agrair tot el suport del tutor per guiar-me en la realització del treball i per la seva ajuda a l'hora de resoldre tots els dubtes que he tingut.

Seguidament vull donar les gràcies a en Josep Maria Cors Iglesias pel temps que em va dedicar i la informació que em va proporcionar.

També a l'Agrupació Astronòmica d'Osona, i concretament a en Pep Pujols Puigdesens, per la seva ajuda, col·laboració i guia a l'hora de realitzar el treball.

I, finalment, als meus pares pel constant suport durant els mesos que he dedicat a desenvolupar el Treball de Recerca.

INTRODUCCIÓ

La determinació del tema del Treball de Recerca la vaig dur a terme després de contemplar dos interessos personals: les matemàtiques i l'astronomia. Són dues ciències que es relacionen entre elles de forma directa: a partir de càlculs amb nombres s'obtenen dades sobre l'univers. Com que aquestes dues disciplines em cridaven l'atenció vaig decidir orientar-hi el treball.

Vaig reduir l'àmbit de recerca a una branca de l'astronomia que s'anomena astrometria, més concretament a la determinació de les efemèrides solars. La idea em va sorgir després d'observar com s'amagava el Sol darrere les muntanyes ja que em va fer preguntar si es poden saber les hores concretes de sortida i posta del Sol encara que variïn per culpa de la posició des d'on s'observen i del relleu de l'horitzó. L'objectiu principal del treball és poder trobar les dades de la sortida i posta del Sol tenint en compte un perfil topogràfic local existent.

Després d'una breu recerca en el tema, vaig arribar a la conclusió que la millor forma d'expressar els resultats seria desenvolupant una simulació informàtica. Ja existeixen programes que calculen les efemèrides solars però ho fan al nivell del mar, és a dir, només són exactament correctes si s'observen des d'una platja i el Sol surt o s'amaga darrere l'horitzó de l'oceà. Per tant, un objectiu secundari va ser trobar les dades de la sortida i posta del Sol, però a través d'un programa informàtic que també inclogués una opció per incloure un perfil topogràfic concret. Per assolir aquests objectius vaig decidir dividir el treball en una part de recerca de càlculs i l'altra de desenvolupament del programa informàtic (o simulació).

D'aquesta manera, el treball queda compost de dos blocs. El primer és la part teòrica i el vaig dur a terme utilitzant diferents fonts. Primer em vaig posar en contacte amb l'Agrupació Astronòmica d'Osona que em va facilitar un llibre de Jean Meeus titulat *Astronomical algorithms* (1991). Es tracta d'un recull de fórmules i mètodes per calcular efemèrides. A més a més, vaig realitzar una videoconferència amb en Josep Maria Cors, doctor en matemàtiques i professor de Matemàtica Aplicada a la UPC, per buscar el millor procés per assolir els objectius del treball. A través de la cerca per internet vaig acabar de complementar la part teòrica.

El segon bloc del treball és la part pràctica. Aquesta consisteix en tot el desenvolupament de la simulació i també en una part de treball de camp relacionada amb la mesura d'un perfil topogràfic local. Per a la creació del programa informàtic vaig comptar amb l'assistència i ajuda d'en Pep Pujols Puigdesens, analista programador i membre de l'Agrupació Astronòmica d'Osona.

Per acabar, la part que completa el Treball de Recerca és la comprovació dels resultats obtinguts.

Durant tot el procés de recerca em van sorgir complicacions puntuals en l'elaboració de càlculs i sobretot a l'hora de realitzar la simulació. De totes maneres, no em van perjudicar per assolir els objectius. Una idea que vaig haver d'abandonar degut a la falta de temps va ser la de crear una simulació en format 3D. Pel que fa a complicacions materials, a causa de la pandèmia la trobada amb en Josep Maria Cors no va ser presencial però amb una videoconferència vam poder fer una sessió explicativa sense complicacions.

Finalment, cal destacar que per poder desenvolupar la simulació vaig haver d'aprendre a treballar amb un programari anomenat Qt i també a programar amb dos llenguatges diferents: C++ i QML. Ja en tenia coneixements previs però els vaig haver d'ampliar per poder assolir el nivell de programació que exigia la simulació.

PART TEÒRICA: RECERCA I CÀLCULS

1. PREVI A LA RECERCA

En aquest primer punt del treball s'explica la situació física dels astres involucrats en la posta i la sortida del Sol.

Primerament, s'ha de saber que la Terra és un planeta del sistema solar, és a dir, que es troba en un sistema heliocèntric, en què els vuit planetes pertanyents al sistema solar orbiten al voltant d'una estrella: el Sol. En la seva òrbita, la Terra es veu afectada simultàniament per quatre moviments: la translació, la rotació, la precessió i la nutació. Els dos primers són, respectivament, el moviment de la Terra al voltant del Sol i el moviment de gir de la Terra sobre el seu propi eix¹ (que queda inclinat respecte de l'òrbita).

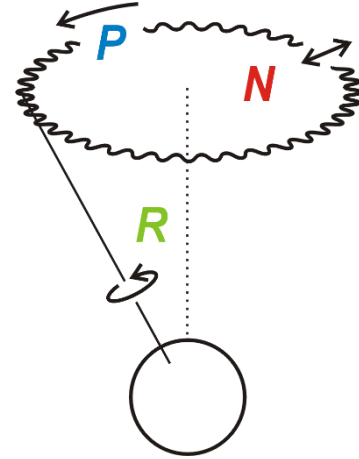


Figura 1. Representació dels moviments de rotació (R), precessió (P) i nutació (N).

El moviment de precessió va associat al gir de l'eix de rotació de la Terra. A causa de la lleugera inclinació d'aquest eix es produeix un moviment harmònic (fig. 1). Aquest fenomen té lloc perquè la Terra és un cos sòlid no esfèric que està sotmès a forces exteriors. És el mateix que li passa a una baldusa quan el seu eix de rotació no és perpendicular a la superfície de rotació. Al moviment de precessió s'hi suma també la variació de l'obliqüitat, que és l'angle que forma l'eix de rotació de la Terra amb la perpendicular del pla de la seva òrbita al voltant del Sol.

La nutació és un moviment que consisteix en una petita oscil·lació de l'eix de rotació de la Terra causada per la influència gravitatòria de la Lluna, el satèl·lit que orbita la Terra (fig. 1).

De la mateixa forma que l'eix de rotació inclinat (aproximadament uns 23,5°) provoca els moviments de precessió i nutació, també és el principal causant de les estacions. Conseqüentment, la superfície de la Terra és il·luminada de forma desigual pel Sol. Això fa variar les condicions climàtiques de cada lloc del planeta durant el període d'un any. Les estacions queden establertes pels solsticis i els

¹ L'eix de rotació de la Terra es defineix com la línia imaginària que passa pels pols terrestres i sobre el qual gira el planeta.

equinoccis. Els solsticis marquen els instants de màxima i mínima inclinació de la Terra respecte al Sol i determinen els inicis de l'estiu i l'hivern. Els equinoccis són els dos instants en què la inclinació és la mateixa per als dos hemisferis² de la Terra. Això vol dir que el dia i la nit tenen la mateixa durada en els dos hemisferis.

D'altra banda, quan es volen calcular els instants de sortida i posta de Sol també s'han de tenir en compte els nodes de la Lluna, que són els punts en què el satèl·lit terrestre travessa l'eclíptica³ del Sol. Aquest fenomen influeix en la visió de la trajectòria de l'estrella i per tant haurà de ser contemplat en els càlculs. Amb relació a la Lluna, també s'hauran de tenir en compte els conceptes de longitud mitjana de la Lluna i també del Sol. Les longituds mitjanes indiquen la posició que tindrien els astres si la seva òrbita fos circular i sense perturbacions. En darrer lloc, també és necessari tenir en compte la longitud vertadera del Sol, que és on es trobaria l'estrella si la inclinació de la seva eclíptica al voltant de la Terra tingués una inclinació nul·la.

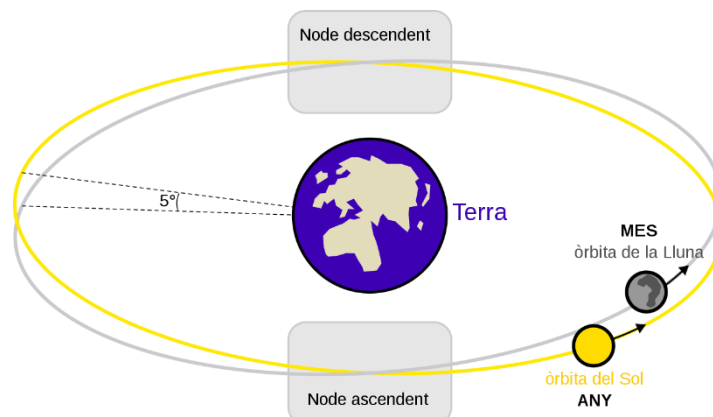


Figura 2. Nodes de la Lluna i eclíptica del Sol.
Imatge: SuperManu <<https://commons.wikimedia.org/w/index.php?curid=2054454>>

Finalment, amb els càlculs explicats en el treball s'obtenen les dades sobre la posició geomètrica del Sol (en la sortida i la posta) des del punt de vista d'un observador que pot estar situat en un punt qualsevol de la superfície de la Terra. A més a més, també cal tenir en compte la refracció atmosfèrica per l'efecte que té sobre l'hora de sortida i posta del Sol. És a dir, que el veiem sortir i pondre's més tard del que indica el càlcul i per això s'han d'aplicar correccions a les fórmules. La

² La línia de l'equador divideix la Terra en els hemisferis nord i sud.

³ L'eclíptica és la corba del moviment aparent del Sol vist des de la Terra.

refracció atmosfèrica ve donada per tres factors: la temperatura, la pressió i l'alçada en què es troba l'observador. Com ja s'explicarà més endavant, en aquest treball es tindrà en compte l'alçada però no els errors causats per la temperatura i la pressió.

2. L'ESFERA CELESTE

Com ja s'ha esmentat, l'heliocentrisme és el model astronòmic que situa el Sol al centre del sistema solar amb tots els planetes orbitant al seu voltant. Aquesta idea ha estat vigent des de fa més de dos mil anys. Tot i així, en certes situacions és convenient canviar de perspectiva i considerar que la Terra és un punt fix en l'espai i que la resta d'astres es mouen al seu entorn. Pel que fa a l'observació celeste des de la superfície terrestre, el principal interès és poder posicionar les estrelles i planetes en un mapa estel·lar des del punt de vista d'un observador.

L'*esfera celeste* és el concepte que ajuda a visualitzar punts al voltant de la Terra. Es pot entendre com una lona negra que envolta el planeta amb punts de llum que indiquen on hi ha estrelles i planetes. Observant el cel amb aquest enfocament es pot apreciar com l'esfera i totes les estrelles es mouen rotant al voltant d'un punt estàtic. Aquest moviment aparent és degut al moviment de rotació de la Terra. Per comoditat, el radi de l'esfera celeste és arbitrari perquè s'hi puguin representar tots els astres, independentment de la distància on es trobin.

Un observador situat al pol Nord veurà totes les estrelles rotar al voltant del *zenit*.⁴ Aquest punt és un dels

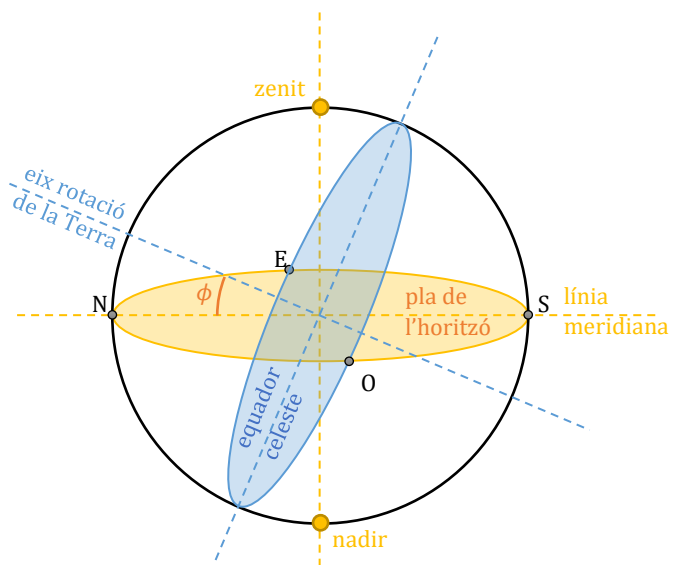


Figura 3. Esfera celeste amb els seus elements representats.

⁴ El zenit està situat sobre el cap de l'observador, en un punt de la recta que passa per la posició de l'observador mateix i pel centre de la Terra (fig. 3).

pols celestes. Com més a prop de l'equador terrestre es troba l'observador, més petita és la distància del pol a l'horitzó. A la figura 3 es poden veure els elements principals de l'esfera celeste: el nadir és el punt més baix a la mateixa vertical del zenit; la línia meridiana transcorre de nord (N) a sud (S) passant per la posició de l'observador (centre de l'esfera celeste); el pla de l'horitzó és perpendicular a la vertical zenit-observador-nadir i divideix l'esfera celeste en dos hemisferis (el visible i el no visible per a l'observador); l'eix de rotació de la Terra,⁵ i el pla de l'equador celeste que és perpendicular a l'eix de rotació terrestre i marca l'est (E) i l'oest (O) en els punts d'intersecció amb el pla de l'horitzó.

2.1. SISTEMES DE COORDENADES

Com en qualsevol mapa, en l'esfera celeste s'utilitzen coordenades per situar els astres. Des del punt de vista de l'observador, el sistema de coordenades més pràctic és el de *coordenades esfèriques* ja que facilita el posicionament de les estrelles. La referència és el pla de l'horitzó i compta amb dues coordenades: l'azimut (a) i l'altura (h), representades a la figura 4. L'azimut és l'angle de la posició d'un astre mesurat sobre l'horitzó celeste des del punt cardinal sud i pot tenir valors d'entre 0° i 360° .⁶ La segona coordenada, l'altura, és la distància angular des de l'astre fins a l'horitzó i pot prendre valors d'entre 90° (zenit) i -90° (nadir). Quan el valor és negatiu vol dir que l'astre està per sota del pla de l'horitzó i que, per tant, no és visible per a l'observador.

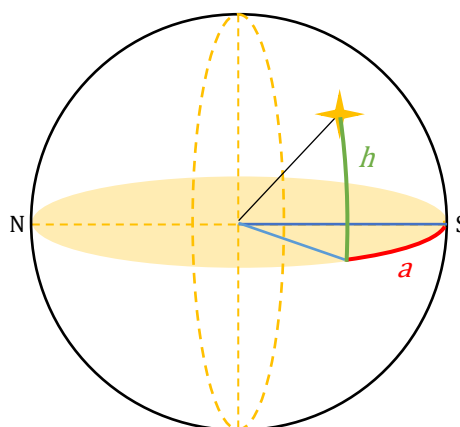


Figura 4. Representació de les coordenades esfèriques.

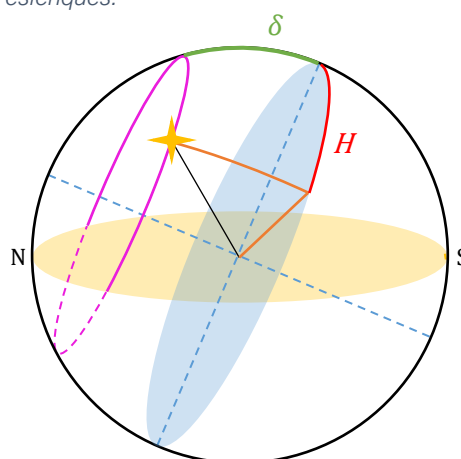


Figura 5. Representació de les coordenades horàries.

⁵ La inclinació d'aquest eix ve donada per la latitud (ϕ) de l'observador.

⁶ S'ha de tenir en compte, però, que el punt a 0° és el mateix que a 360° .

Com ja s'ha comentat anteriorment, el sistema de coordenades esfèriques és molt eficaç i còmode per a l'observador. De totes maneres, no és pràctic per posicionar els astres en l'espai de forma general, ja que depenen completament de la posició de l'observador.

Existeix, per tant, un segon sistema de coordenades: les *coordenades horàries* (fig. 5). En aquest sistema de coordenades la referència és el pla de l'equador celeste. En conseqüència, la posició de l'astre dependrà de la latitud⁷ on es trobi l'observador. També consta de dues coordenades: la declinació (δ) i l'angle horari (H) i en certa forma es poden considerar equivalents de l'altura i l'azimut respectivament. La declinació indica l'angle entre l'astre i l'equador terrestre. Pot agafar valors d'entre 90° i -90° , però en el cas de la Terra (cas d'interès per a aquest treball), només va dels $23,45^\circ$ als $-23,45^\circ$ a causa de l'angle d'inclinació de l'eix de la Terra. L'angle horari representa la distància angular que recorre l'astre des del punt d'intersecció de l'equador amb el meridià de l'observador fins a la posició de l'astre en sentit horari. Tot i que es pot mesurar en graus, generalment la seva unitat de mesura és l'hora⁸ i al llarg d'un dia augmenta en 24 h.

A la figura 5 es pot observar en color magenta el recorregut que faria l'astre dibuixat en un dia. Les línies discontinües indiquen la zona de pas en què l'astre no és visible per a l'observador ($h < 0$). En astronomia, l'*orto* es defineix com el moment en què l'observador comença a veure l'astre i l'*ocàs*, com el moment en què l'astre es pon. Depenent de la posició de l'observador i de l'astre que es tingui en compte, aquest serà visible durant tot el dia o en cap moment. Als astres que segueixen un recorregut com el primer se'ls anomena circumpolars.

2.2. TRANSFORMACIÓ DE COORDENADES

Des del punt de vista de l'observador i de forma teòrica, si es coneixen l'altura i l'azimut del Sol en cada moment del dia es podria realitzar un gràfic del seu recorregut al llarg d'un dia similar al de la figura 6, on es representa el recorregut

⁷ La latitud i la longitud conformen el sistema de coordenades geogràfiques. La latitud indica la distància angular sobre un meridià entre un punt de la superfície terrestre i l'equador.

⁸ Una hora equival a 15° , un arc de minut (') a $\frac{1}{60}$ de grau (°) i un arc segon (") a $\frac{1}{3600}$ de grau (°).

del Sol (no real) en els equinoccis i solsticis (per mostrar en quins casos és més llarga o més curta la trajectòria del Sol). La variació en el recorregut vist des d'un mateix lloc però en diferents moments de l'any és deguda a la declinació. Aquesta té el seu valor màxim al juny, durant el solstici d'estiu, i és $\delta = 23,45^\circ$. Els equinoccis de març i setembre són els únics moments de l'any en què el Sol surt exactament per l'est i es pon exactament per l'oest. Això és perquè la declinació és $\delta = 0^\circ$ i per tant el recorregut del Sol coincideix amb la línia del pla de l'equador terrestre. El mínim és al desembre, durant el solstici d'hivern, quan la declinació és $\delta = -23,45^\circ$.

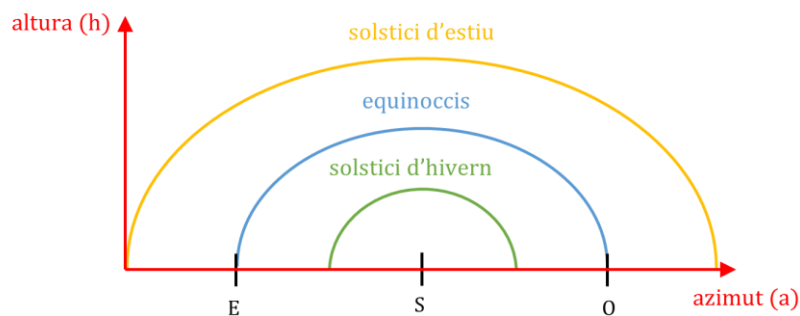


Figura 6. Representació gràfica (no real) del recorregut del Sol des del punt de vista de l'observador en els solsticis i equinoccis.

Queda clar, doncs, que el que es necessita per poder fer el gràfic en un dia de l'any són l'altura i l'azimut. Una forma manual d'aconseguir les dades és mesurar els angles manualment, però la poca efectivitat d'aquest càlcul és evident i no seria vàlida per poder saber la trajectòria del Sol en un dia futur. No hi ha cap més alternativa que trobar aquestes dades a partir d'unes altres i aplicar diversos càlculs matemàtics.

Tenint en compte tots els conceptes que es coneixen, es recorre a la relació que existeix entre les coordenades esfèriques i les horàries. A la figura 7 es poden veure representats els dos sistemes de coordenades. La relació més evident que s'hi troba és que l'eix y (del sistema de coordenades esfèriques,

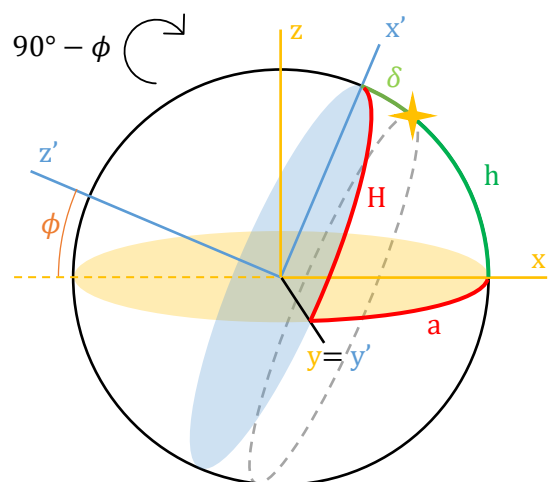


Figura 7. Representació de dos tipus de coordenades: les esfèriques i les horàries.

en groc) i l'eix y' (del sistema de coordenades horàries, en blau) estan en la mateixa recta. Si mitjançant aquest eix y es fan rotar en sentit horari un angle de $90^\circ - \phi$ els eixos x' i z' , llavors aquests coincideixen amb els eixos x i z .

El que això indica és que si es calcula matemàticament aquest gir de rotació entre els dos sistemes de coordenades s'establirà una relació directa entre ells que permetrà calcular l'altura i l'azimut a partir de la declinació i l'angle horari i viceversa.

El procediment descrit s'anomena transformació de coordenades. Per a aquest treball s'utilitzen les fórmules següents:

Equació 1. Transformació de coordenades en l'eix x.

$$\cos h \cdot \cos a = \sin \phi \cdot \cos \delta \cdot \cos H - \cos \phi \cdot \sin \delta$$

Equació 2. Transformació de coordenades en l'eix y.

$$\cos h \cdot \sin a = \cos \delta \cdot \sin H$$

Equació 3. Transformació de coordenades en l'eix z.

$$\sin h = \cos \phi \cdot \cos \delta \cdot \cos H + \sin \phi \cdot \sin \delta$$

On h = altura, a = azimut, ϕ = latitud, δ = declinació, H = angle horari.

El procés d'obtenció d'aquestes fórmules comença per les coordenades cartesianes, on es troba el valor de les coordenades x , y i z de la posició d'un astre (fig. 8). Per a això es necessiten tres dades: l'angle de l'altura (h), l'angle de l'azimut (a) i el valor del radi de l'esfera celeste r , que és la distància que hi ha de l'observador a l'astre. De totes maneres, com que el radi de l'esfera celeste és arbitrari, en els càlculs següents es considera que el seu valor és 1 i així se simplifiquen de forma considerable.

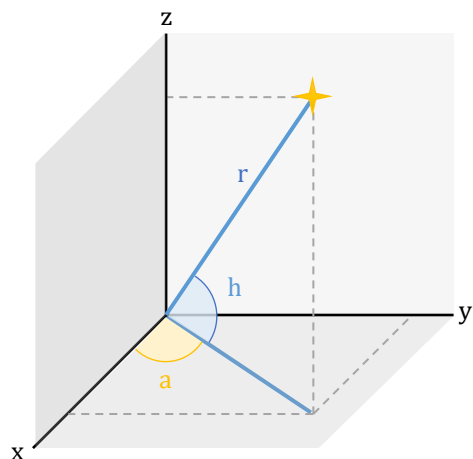


Figura 8. Coordenades de la posició d'un astre en el sistema de coordenades esfèriques.

D'acord amb la figura 8, els valors de les coordenades són:

$$x = r \cdot \cos h \cdot \cos a$$

$$y = r \cdot \cos h \cdot \sin a$$

$$z = r \cdot \sin h$$

El següent i últim pas és aplicar una matriu de rotació⁹ a les equacions anteriors per reproduir el gir de $90^\circ - \phi$ graus en els eixos i relacionar així els dos sistemes de coordenades. Després del càlcul, s'obtenen les equacions 1, 2 i 3.

La finalitat de la transformació de coordenades és poder trobar l'altura i l'azimut del Sol per a cada moment del dia i poder dibuixar un gràfic semblant al de la figura 6. És necessari, però, tenir els valors de la latitud, la declinació i l'angle horari per substituir-los a les equacions i trobar les incògnites.

3. LA DECLINACIÓ SOLAR

Com s'ha comentat prèviament, la declinació varia segons l'estació de l'any degut a la translació al voltant del Sol i al moviment de rotació. Si l'eix de la Terra no fos inclinat, la declinació sempre seria de 0° . Però només agafa aquest valor durant els equinoccis de tardor i primavera.

Existeix una fórmula que calcula la declinació de forma general per a diferents astres i per a diferents aplicacions, però n'hi ha una de més complexa que només s'aplica per trobar la declinació del Sol:

Equació 4. Declinació solar.

$$\delta = \phi \cdot \cos\left[C \cdot (d - d_r) / d_y\right]$$

On ϕ és l'angle d'inclinació de l'eix de la Terra ($23,45^\circ$), C són 360° complets (o 2π radians), d és el dia julià corresponent a la data, d_r és el dia julià del solstici d'estiu i d_y és el nombre de dies que té l'any (que dependrà de si es tracta d'un any de traspàs o no).

⁹ En aquest treball no s'explica el càlcul de la matriu de rotació ja que només és necessari el resultat i el procés d'obtenció depassa els objectius d'aquest treball.

3.1. FRANGES HORÀRIES, DATES I CALENDARIS

En tots els càlculs que apareixen en els apartats previs i posteriors del treball, les dades que s'utilitzen estan sempre vinculades al temps universal coordinat (UTC). Aquest és un sistema utilitzat per mesurar el temps i a partir del qual el món regula els rellotges i el temps. Totes les hores dels diferents llocs del món es calculen a partir de la zona horària de referència.¹⁰ A la figura 9 hi ha un mapamundi que indica les actuals franges horàries. Es pot observar com en molts casos les hores afegides o restades a l'UTC de cada lloc s'adapten a les fronteres polítiques i no coincideixen amb les franges geogràfiques. Per exemple, si es vol saber quina serà la declinació solar el dia 5 de novembre de l'any 2021 a les 12 h del migdia a Catalunya (+01:00 UTC, com indica el mapa de la pàgina següent), abans de fer els càlculs necessaris cal convertir el temps local de Catalunya a UTC. En aquest cas, per tant, és necessari restar una hora a les 12 h i fer els càlculs pertinents amb la dada obtinguda. Això implica que el resultat també serà en UTC i, per tant, si llavors es vol saber l'hora local del resultat s'hauran de sumar o restar les hores corresponents.

3.1.1. Nombre de dies que té l'any

Una de les dades necessàries per poder calcular la declinació solar és el nombre de dies que té l'any en qüestió. Per poder-ho saber es necessita conèixer l'organització dels calendaris.

Es parla de calendaris en plural perquè tot i que el calendari referent actual és el gregorià, prèviament s'emprava el calendari julià. Aquest va ser proposat per Juli Cèsar l'any 46 aC i va entrar en vigor l'1 de gener de l'any 45 aC. Es tracta d'una reforma del calendari romà que va ser dissenyada pels matemàtics i astrònoms grecs. Aquest calendari constava de dos tipus d'any distribuïts en un cicle de 3 anys de 365 dies i 1 any de 366, coincidint amb els anys divisibles per 4.

¹⁰ La zona horària de referència és la franja corresponent al meridià de Greenwich.

Això suposa un any mitjà de 365,25 dies i com que el valor real de l'any solar¹¹ és de 365,24219 dies, el calendari julià guanya un dia cada 128 anys. Aquest calendari va ser predominant a l'Imperi romà i a Occident durant uns 1.600 anys però el papa Gregori XIII va fer una modificació per reduir l'any mitjà a 365,2425 dies per aproximar-se al valor de l'any solar.

D'aquesta forma, durant els segles següents hi va haver una adaptació generalitzada al calendari gregorià, amb l'excepció d'algunes zones on predominaven l'Església ortodoxa i els berbers. El calendari gregorià distingeix entre tres tipus d'any: els comuns (de 365 dies), els de traspàs (de 366 dies) i els seculars (tots els anys acabats en "00"). Això comporta cicles de 400 anys on:

- En 400 anys hi ha $\frac{400}{4} - (4 \text{ anys seculars}) = 96$ anys de traspàs.
- Cada 4 anys seculars un és de traspàs, per tant, en realitat té 97 anys de traspàs.
- La resta, $400 - 97 = 303$ són anys comuns.

Un any serà de traspàs si és divisible per 4 amb les excepcions següents:

- No són anys de traspàs els anys seculars que no siguin divisibles per 400 (per exemple els anys 1700, 1800, 1900 i 2100 són anys comuns tot i ser divisibles per quatre).
- Els anys seculars divisibles per 400, però, sí que són de traspàs (per exemple 1600, 2000 i 2400).

3.1.2. El dia julià

A més a més del nombre de dies que té l'any, per la declinació també és necessari saber a quin dia de l'any es correspon una data donada però en dies julians. El dia julià es correspon al nombre de dies solars comptats des de l'1 de gener de l'any 4713 aC a les 12 h del migdia fins a la data desitjada. Per exemple, el dia 5 de novembre de 2021 (a les 00:00 h) és el dia julià 2459523,5.

¹¹ L'any solar és el temps que el Sol triga a tornar a la mateixa posició vist des de la Terra.

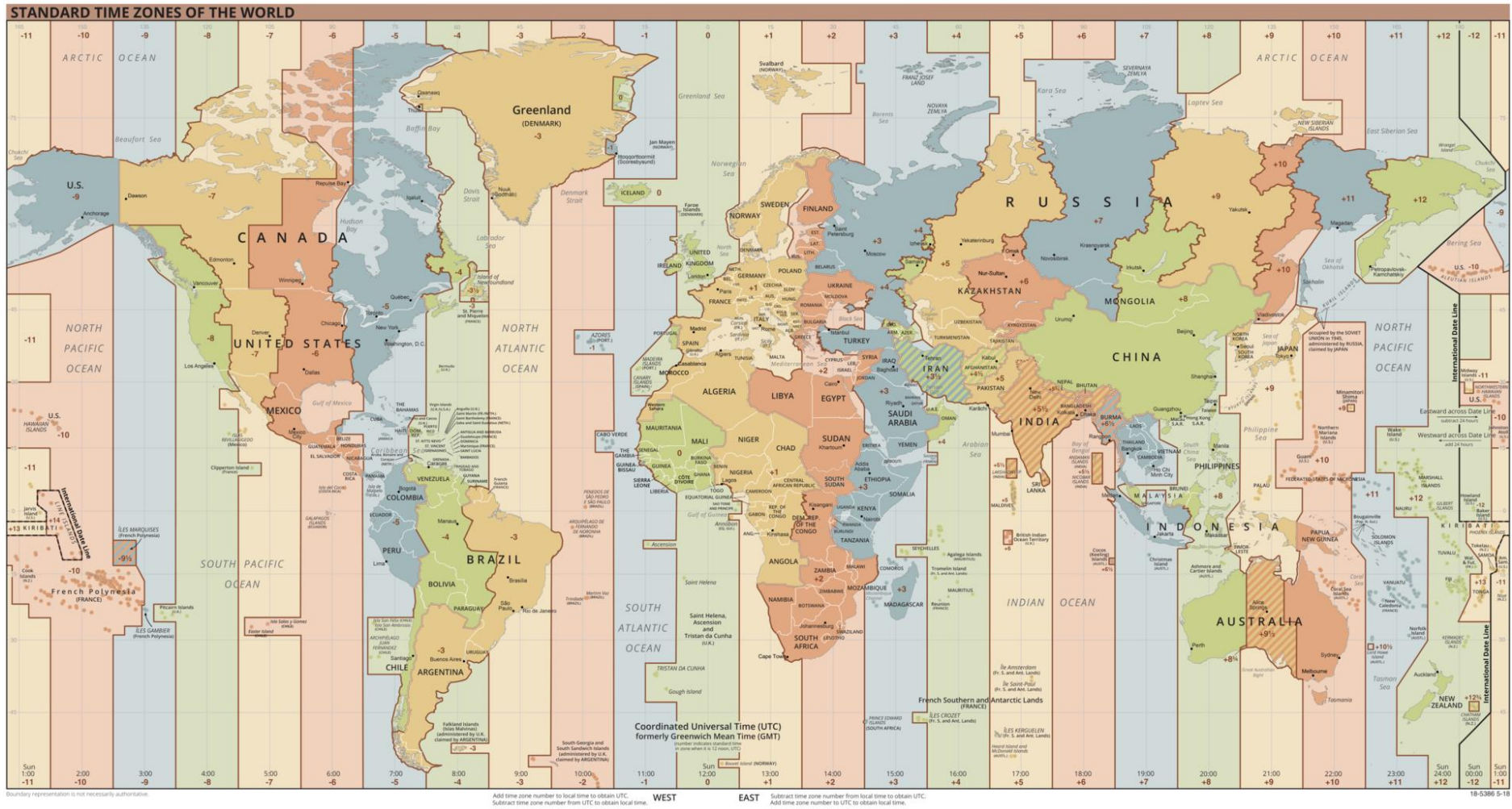


Figura 9. Font: TimeZonesBoy - US Central Intelligence Agency <<https://commons.wikimedia.org/w/index.php?curid=22556731>>

Aquest càlcul el poden fer moltes calculadores en línia però Jean Meeus n'explica el càlcul en el seu llibre *Astronomical Algorithms* (1991).

Per poder realitzar el càlcul segons Jean Meeus, és necessari tenir dos fets en compte. El primer és que l'any precedent a l'1 dC és l'1 aC; dit d'una altra forma, no existeix l'any 0. Malgrat aquest fet, per facilitar els càlculs es tindrà en compte l'any 0 i també s'utilitzarà notació d'anys negatius per als previs. Seguint aquest criteri, els anys 1, 5, 9 i 13 aC passen a ser els anys 0, -4, -8 i -12 respectivament a l'hora de calcular el dia julià corresponent d'una data pertanyent a aquests anys. A més a més, el procediment per calcular el dia julià varia segons si la data forma part del calendari gregorià (després del 24 de febrer de 1582) o del calendari julià (abans de la reforma).

El procediment de Jean Meeus (orientat ja al càlcul per a ordinador) és el següent: donada una data en el format "4,81 octubre 1957" el dia (D) és "4,81", el mes (M) és el nombre del mes corresponent (en aquest cas $M = 10$) i l'any (Y) és "1957". Llavors:

- Si la data és al gener o al febrer ($M = 1$ o $M = 2$), es considera que el mes és el 13è o 14è de l'any anterior. Per això, es resta un any a Y i se sumen 12 mesos a M . En cas contrari, Y i M no varien.
- Es calculen els paràmetres A i B tenint en compte el calendari al qual correspon la data: $A = INT\left(\frac{Y}{100}\right)^{12}$ i si la data forma part del calendari julià $B = 0$, sinó $B = 2 - A + INT\left(\frac{A}{4}\right)$. Llavors, el dia julià (DJ) serà:

Equació 5. Dia julià.

$$DJ = INT[365,25 \cdot (Y + 4716)] + INT[30,6001 \cdot (M + 1)] + D + B - 1524,5$$

Si provem el càlcul de la data "4,81 octubre 1957", com que forma part del calendari gregorià, $D = 4,81$, $M = 10$ i $Y = 1957$. Successivament trobem:

$$A = INT\left(\frac{1957}{100}\right) = INT(19,57) = 19$$

¹² $INT(a)$ és una funció que utilitza l'ordinador per truncar a , és a dir, agafa la part entera de a .

$$B = 2 - 19 + INT\left(\frac{19}{4}\right) = 2 - 19 + 4 = -13$$

$$DJ = INT(365,25 \cdot 6673) + INT(30,6001 \cdot 11) + 4.81 - 13 - 1524,5$$

$$DJ = 2436116,31$$

Exemple 1. Calcular el dia julià d'una data.

Hi ha dues aplicacions interessants del dia julià; la primera és que quan es busca un interval de dies entre dues dates (pot ser entre dos fenòmens astronòmics) aquest és senzillament la diferència entre els corresponents dies julians, i la segona és que es pot trobar fàcilment el dia de la setmana dividint el dia julià entre set. El residu indica el dia que és següent la taula següent:

0	1	2	3	4	5	6
Diumenge	Dilluns	Dimarts	Dimecres	Dijous	Divendres	Dissabte

En el mateix llibre, Jean Meeus també especifica un mètode per trobar la data corresponent a un dia julià, és a dir, realitzar el procés invers a l'anterior:

- Primerament se suma 0,5 al dia julià i seguidament se separa la part entera (z) de la part decimal (F). Si $z < 2299161$, llavors $A = z$. En cas contrari es calcula:

$$\alpha = INT\left(\frac{z-1867216.25}{36524.25}\right) \quad A = z + 1 + \alpha - INT\left(\frac{\alpha}{4}\right)$$

I els paràmetres B , C , D i E :

$$B = A + 1524 \quad C = INT\left(\frac{B-122,1}{365,25}\right)$$

$$D = INT(365,25 \cdot C) \quad E = INT\left(\frac{B-D}{30,6001}\right)$$

- El dia del mes (*dia*) amb decimals, el nombre del mes (M) i l'any (Y) seran:

$$dia = B - D - INT(30,6001 \cdot E) + F$$

$$M = \begin{cases} E - 1 & \text{si } E < 14 \\ E - 13 & \text{si } E = 14 \end{cases} \quad Y = \begin{cases} C - 4716 & \text{si } M > 2 \\ C - 4715 & \text{si } M = 1 \text{ o } M = 2 \end{cases}$$

Per poder comprovar els càlculs, s'ha aplicat aquest mètode al dia julià 2436116,31 (resultat del càlcul del dia julià utilitzat com a exemple prèviament) i, efectivament, s'ha obtingut $dia = 4.81$, $M = 10$ (*octubre*) i $Y = 1957$, la data utilitzada.

3.1.3. Solsticis i equinoccis

Fixant de nou l'atenció en l'equació 4, amb el que s'ha explicat fins a aquest punt ja es pot saber l'angle d'inclinació de la Terra ($23,45^\circ$) i el dia julià de la data (d). Per trobar d_r (el dia julià del dia del solstici d'estiu de l'any en qüestió) s'explicarà el mètode que proposa Jean Meeus en el llibre esmentat abans. Realment, els solsticis i els equinoccis són tan sols un instant, només en un moment concret del dia la declinació solar serà màxima, mínima o nul·la. Tot i que el mètode té una precisió de 51 segons¹³ sobre l'instant en el qual tindrà lloc el solstici o l'equinocci, aquesta és irrellevant ja que només és necessari trobar el dia del fenomen.

El primer pas del mètode de Jean Meeus és trobar l'instant de l'equinocci o solstici mitjà (DJE_0) a partir d'un polinomi de quart grau seguint les taules A i B de l'annex III. Hi ha vuit expressions, dues per a cada solstici i equinocci, de les quals només se n'utilitza una segons el rang d'anys al qual pertanyi la data (entre l'any 1000 aC i el 1000 dC o entre l'any 1000 dC i el 3000 dC). S'obté un dia julià 'provisional' ja que seguidament s'aplica la següent correcció on DJ és el dia julià final de l'equinocci o solstici:

$$DJ = DJE_0 + \frac{0,00001 \cdot S}{\Delta\lambda}$$

On S és el sumatori de 24 termes periòdics (taula C de l'annex III) segons la fórmula indicada: $S = \sum A \cdot \cos(B + CT)$.¹⁴ Després es troba el valor dels paràmetres T i W i es calcula $\Delta\lambda$:

$$T = \frac{DJE_0 - 2451545,0}{36525} \quad W = 35999,373^\circ \cdot T - 2,47^\circ$$

$$\Delta\lambda = 1 + 0,0334 \cdot \cos(W) + 0,0007 \cdot \cos(2W)$$

Com ja s'ha indicat, el resultat dels càlculs és el dia julià de l'instant en què té lloc el solstici o equinocci. Per poder obtenir el dia del mes que tindrà lloc el fenomen, sense especificar l'hora concreta, només fa falta passar el dia julià calculat

¹³ Cal concretar que aquesta precisió és només entre els anys 1951 i 2050. Com més allunyada és la data, menys precisió tindrà el càlcul.

¹⁴ Les funcions trigonomètriques amb la calculadora de l'ordinador es fan per defecte en radians; per tant, com que els termes B i C estan en graus, abans de calcular el cosinus s'han de passar a radians.

a la seva data corresponent i truncar el valor del dia obtingut (per si aparegués una part decimal).

Si per exemple es vol saber quin dia serà el solstici d'estiu de l'any 1957, segons les taules A i B de l'annex III, l'instant de l'equinocci mitjà es calcula amb el polinomi de quart grau de B.2. on $Y = \frac{1957-2000}{1000} = -0,043$ i, per tant:

$$JDE_0 = 2451\,716,56767 + 365\,241,62603 Y + 0,00325 (-0,043)^2 + 0,00888 (-0,043)^3 - 0,00030 (-0,043)^4 = 2436011,178$$

Afegint-hi la correcció:

$$T = \frac{2436011,178 - 2451545,0}{36525} = -0,4252928746$$

$$W = 35999,373^\circ \cdot (-0,4252928746) - 2,47^\circ = -15312,74683$$

$$\Delta\lambda = 1 + 0,0334 \cdot \cos(-15312,74683) + 0,0007 \cdot \cos(2 \cdot (-15312,74683)) = 0,9680550013$$

$$S = \sum A \cdot \cos(B + CT) = +359$$

$$DJ = 2436011,178 + \frac{0,00001 \cdot 359}{0,9680550013} = 2436011,182$$

Si passem el dia julià obtingut a la seva data corresponent trobem que l'any 1957 l'equinocci d'estiu tindrà lloc el 21,682 de juny, és a dir, el 21 de juny a les 16:22 h.

Exemple 2. Dia de l'equinocci.

3.1.4. Nombre de dies a l'any

Finalment, l'única dada que falta saber per poder calcular la declinació (equació 4) és saber el nombre de dies que té un any. Això dependrà de si és un any de traspàs o no, ja que tindrà 366 o 365 dies respectivament. Seguint les característiques dels diferents tipus d'any que existeixen en el calendari gregorià, es pot calcular si un any és de traspàs o no mitjançant la funció¹⁵ *anyTraspàs()* (fig. 10) escrita en C++, pertanyent al codi de la simulació (vegeu l'annex I). La funció rep l'any com a dada i

```

111 int RCalculadora::anyTraspàs(int iAny)
112 {
113     if (iAny%4==0)
114     {
115         if (iAny%100==0)
116         {
117             if (iAny%400==0)
118                 return(1);
119             else
120                 return(2);
121         }
122         else
123             return(1);
124     }
125     else
126         return(2);
127 }

```

Figura 10. Funció *anyTraspàs()*.

¹⁵ Una funció d'un codi font realitza uns càlculs o accions a partir d'unes dades que se li poden donar (o no) i pot retornar valors.

el que fa és mirar si és divisible per 4, 100 i 400 i segons els diferents casos retorna un 1 (si és un any comú) o un 2 (si és un any de traspàs).

Si es vol saber la declinació solar del dia 4,81 d'octubre de 1957, primer trobarem: $d = 2436116,31$ (exemple 1), $d_r = 2436011,182$ (exemple 2) i $d_y = 365$ (el 1957 no és un any de traspàs). Llavors:

$$\delta = 23,45^\circ \cdot \cos\left(360^\circ \cdot \frac{2436116,31 - 2436011,182}{365}\right) = -5,54^\circ$$

Exemple 3. Calcular la declinació solar.

Si es realitzés el mateix càlcul que a l'exemple 3 però amb la data d'un equinocci qualsevol, es veuria com el valor de la declinació seria pràcticament 0° . El signe positiu o negatiu del valor indica si és tardor o hivern (declinació negativa) o primavera o estiu (declinació positiva).

4. L'ANGLE HORARI

Si la latitud de l'observador i la declinació són conegudes, en l'equació 3 es podria obtenir una altura per a cada valor de l'angle horari. Jean Meeus explica una fórmula senzilla per trobar aquest angle que és la següent:

Equació 6. Angle horari.

$$H = \theta - L - \alpha$$

Utilitzar aquesta fórmula no seria necessari ja que es podrien trobar l'altura i l'azimut del Sol amb les equacions 2 i 3 substituint l'angle horari per valors de 0 a 24 hores i s'obtindrien les dades de la posició del Sol en el període d'un dia. Però quan l'angle horari val 0 h les dades obtingudes no seran necessàriament corresponents a les 00:00 h del dia pel qual s'han realitzat els càlculs. Per poder relacionar la posició del Sol directament amb cada hora del dia és més senzill calcular primerament l'angle horari en funció de l'hora i llavors substituir-lo a les fórmules de transformació de coordenades per trobar l'altura i l'azimut en aquella hora concreta.

A l'equació 6 θ és el temps sideri mitjà a Greenwich, L és la longitud de l'observador i α és l'ascensió recta. A partir de θ és com es relaciona l'angle horari amb l'hora del dia concreta.

4.1. EL TEMPS SIDERAL MITJÀ DE GREENWICH (GMST o θ)

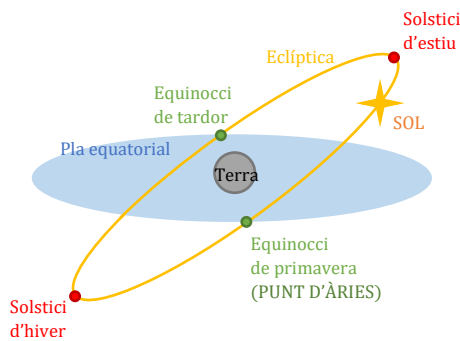


Figura 11. Punt d'Àries i l'eclíptica.

El temps sideri és la mesura de la rotació terrestre respecte a objectes celestes distants. L'any sideri compta amb un dia més que l'any solar. Quan es parla de temps sideral local es fa referència a l'angle horari que forma el punt d'Àries amb el meridià de l'observador. El punt d'Àries es troba a la recta d'intersecció del pla de l'equador celeste amb el de l'eclíptica. A la figura 11 es pot apreciar com aquest punt és el pas del Sol de l'hemisferi sud a l'hemisferi nord i coincideix amb l'equinocci de primavera. Per tant, en el temps sideral mitjà de Greenwich l'angle horari és el que es forma entre el punt d'Àries i el meridià de Greenwich. Per calcular-lo primer es troba T , per a la qual es necessita saber el dia julià (de l'instant en el qual es vol trobar la posició del Sol) i, per tant, trobem aquesta relació directa entre l'angle horari i l'hora del dia:

$$T = \frac{DJ - 2451545,0}{36525} \quad (DJ = \text{dia julià de l'instant})$$

Llavors, el temps sideral mitjà de Greenwich s'obté amb l'equació 7, a la qual s'aplica una correcció per trobar el temps sideral aparent a Greenwich (a l'equació 8):

Equació 7. Temps sideral mitjà de Greenwich.

$$\theta_0 = 280,460\,618\,37 + 360,985\,647\,366\,29 \cdot (JD - 2\,451\,545,0) + 0,000\,387\,933 \cdot T^2 - \frac{T^3}{38710000}$$

Equació 8. Temps sideral aparent a Greenwich.

$$\theta = \theta_0 + \frac{\Delta\psi \cdot \cos \varepsilon}{15} \quad [\text{en segons}]$$

On es corregeixen els fenòmens de $\Delta\psi$ (nutació en longitud) i ε (obliqüitat) de l'eclíptica i es calculen de la forma següent:

Equació 9. Obliqüitat.

$$\boxed{\varepsilon = 23,69930 - 0,00013 \cdot Y} \quad (Y = any)$$

$$\Delta\psi = -17,20'' \cdot \sin \Omega - 1,32'' \cdot \sin 2L - 0,23'' \cdot \sin 2L' + 0,21'' \cdot \sin 2\Omega$$

A la vegada, la nutació es veu afectada per Ω , els nodes de la Lluna; L i L' , longitud mitjana del Sol i de la Lluna respectivament i es poden trobar segons:

$$\Omega = 125,044\ 52 - 1\ 934,136\ 261 \cdot T + 0,0020708 \cdot T^2 + \frac{T^3}{450000}$$

$$L = 280,466\ 5^\circ + 36\ 000,769\ 8 \cdot T \quad L' = 218,316\ 5^\circ + 481\ 267,881\ 3^\circ \cdot T$$

4.2. ASCENSIÓ RECTA

L'ascensió recta és una coordenada equatorial equivalent a la longitud terrestre en coordenades geogràfiques. Es mesura en hores, minuts i segons d'arc. Es troba segons l'equació:

Equació 10. Ascensió recta.

$$\boxed{\alpha = ATN2(\cos \varepsilon \cdot \sin \Theta, \cos \Theta)}^{16}$$

On ε és l'angle de l'obliqüitat (equació 9) i Θ és la longitud vertadera del Sol, que es troba de la forma següent:

Equació 11. Longitud verdadera del Sol.

$$\boxed{\Theta = L_0 + C}$$

Es calculen L_0 i C segons:

$$L_0 = 280,46645^\circ + 36000,76983^\circ \cdot T + 0,0003032^\circ \cdot T^2$$

$$C = (1,914699^\circ - 0,004817^\circ \cdot T - 0,000014^\circ \cdot T^2) \cdot \sin M \\ + (0,019993^\circ - 0,000101^\circ \cdot T) \cdot \sin 2M + 0,000290^\circ \cdot \sin 3M$$

$$M = 357,52910^\circ + 35999,05030^\circ \cdot T - 0,0001559^\circ \cdot T^2 - 0,00000048^\circ \cdot T^3$$

Finalment, podent calcular el GMST i l'ascensió recta, ja es pot trobar l'angle horari del Sol en un instant determinat d'un dia concret.

¹⁶ La funció $ATN2(\alpha)$ calcula un angle α en el quadrant correcte ja que es donen les seves dues components (separades per una coma).

5. CÀLCUL COMPLET DE LA POSICIÓ DEL SOL EN UN INSTANT CONCRET

Per comprovar efectivament que tots els càlculs donen la posició correcta del Sol en un instant concret s'ha fet la comprovació intentant trobar l'altura i l'azimut del Sol el 5 de novembre de 2021 a les 14 h i 30 min vist des de les coordenades d'un punt de Vic (2,241609737°E 41,933318377°N).

Primer trobem el dia julià corresponent a aquest instant passat a UTC (com que a Vic es porten dues hores d'avançament, es resten 2 h a l'instant escollit abans de calcular el dia julià) i seguidament calculem successivament totes les dades per trobar la declinació i l'angle horari:

$DJ = 2\,459\,524,020\,83$	$\Omega = -5,19191\ rad$
$T = 0,2184536845995844$	$\Delta\psi = -16,62915''$
$d = DJ$	$\varepsilon = 23,43657^\circ$
$d_r = 2459386,648$	$\theta_0 = -0,00424^\circ$
$d_y = 365$	$\theta = 232,45607^\circ$
$\delta = -16,72314^\circ$	$\Theta = 223,3205^\circ$
$L' = 1838,75775\ rad$	$\alpha = -139,13264^\circ$
$L = 142,15650\ rad$	$H = 13,83129^\circ$

L'únic que falta és aplicar la transformació de coordenades. Per trobar l'altura, s'utilitza l'equació 3:

$$h = \arcsin (\cos 41,93332^\circ \cdot \cos (-16,72314)^\circ \cdot \cos 13,83129^\circ + \sin 41,93332^\circ \cdot \sin (-16,72314)^\circ) = 29,96757^\circ = 29^\circ 58' 3,252''$$

En aquest cas, com que les funcions trigonomètriques donen valors del segon i quart quadrant, segur que s'obté el valor correcte de l'altura ja que aquesta pot prendre valors d'entre 90 i -90° . Un cop obtinguda, si se substitueix a l'equació 2, obtindrem el valor de l'azimut:

$$a = \arcsin \left(\frac{\cos (-16,72314)^\circ \cdot \sin 13,83129^\circ}{\cos 29,96757^\circ} \right) = 15,32451^\circ$$

Al contrari que amb l'altura, l'azimut ha de tenir valors de 0° a 360° i és necessari saber a quin quadrant pertany l'angle. En aquest cas en particular, l'azimut correcte és de 195,32451° (s'hauria de sumar 180° al resultat), però a través del llibre de Jean Meeus, s'ha trobat la següent fórmula que també s'utilitza per a la transformació de coordenades:

$$\tan a = \frac{\sin H}{\cos H \cdot \sin \phi - \tan \delta \cdot \cos \phi} \text{ que utilitzant la funció } ATAN2(\beta) \text{ és equivalent a}$$

Equació 12. Azimut.

$$a = ATAN2(\sin H, \cos H \cdot \sin \phi - \tan(\delta) \cdot \cos \phi)$$

Aquesta equació, utilitzant la funció $ATAN2(\beta)$, que es pot utilitzar en codi font (per a més endavant a la simulació), dona els angles amb valors d'entre -180° i 180° (angles en quatre quadrants) ja que calcula l'angle a partir de les seves dues components. Tot i així, cal sumar 180° al resultat per obtenir sempre un valor positiu de l'azimut que anirà de 0° a 360°. És a dir:

$$\begin{aligned} a &= ATAN2(\sin 13,83129^\circ, \cos 13,83129^\circ \cdot \sin 41,93332^\circ \\ &\quad - \tan(-16.72314^\circ) \cdot \cos 41,93332^\circ) = 15,32451^\circ + 180^\circ \\ &= 195,32451^\circ = 195^\circ 19' 28,236'' \end{aligned}$$

Per tant, els resultats obtinguts són que $h = 29^\circ 58' 3,252''$ i $a = 195^\circ 19' 28,236''$.¹⁷

Un cop ja conegut el procés per arribar a trobar la posició del Sol vist per un observador en un instant determinat, el pas següent és aplicar-lo a un programa informàtic. L'objectiu és aconseguir desenvolupar una simulació capaç de fer tots aquests càlculs, que els representi i que siguin el màxim d'exactes possible. La part pràctica del treball explica les bases de la simulació i el codi font i com s'ha aconseguit crear-la.

¹⁷ Tots els càlculs d'aquest apartat s'han obtingut utilitzant un ordinador, amb més precisió que una calculadora física.

PART PRÀCTICA: DESENVOLUPAMENT DE LA SIMULACIÓ

1. PANTALLA

Per crear una simulació va ser necessari trobar un entorn de treball apte i eficaç amb el qual es pogués programar amb un llenguatge de programació familiar. Amb coneixements previs de C++, es va escollir treballar amb el Qt, que és un programari lliure amb llicències comercials. El Qt permet treballar en un entorn multiplataforma i amb interfície gràfica d'usuari¹⁸ i compta amb les dues eines bàsiques: un editor de text (per escriure el codi) i un compilador (que analitza i executa el programa).

La simulació es basa en dos grans blocs que s'han programat amb llenguatges informàtics diferents: QML (basat en JavaScript) per a la part gràfica i d'interacció amb l'usuari (pantalla) i C++ per a la part tècnica de càlculs matemàtics més complexos (funcions). D'aquesta forma, el programa queda organitzat i és senzill de crear i de millorar.

El procés general s'ha basat en quatre parts:

1. Obtenció d'informació.
2. Disseny.
3. Codificació.
4. Comprovació.

Abans de començar a programar, la primera part ja s'ha finalitzat. Consta de tota la informació necessària relacionada, en aquest cas, els càlculs matemàtics explicats a la part teòrica. Seguidament té lloc el procés de disseny.

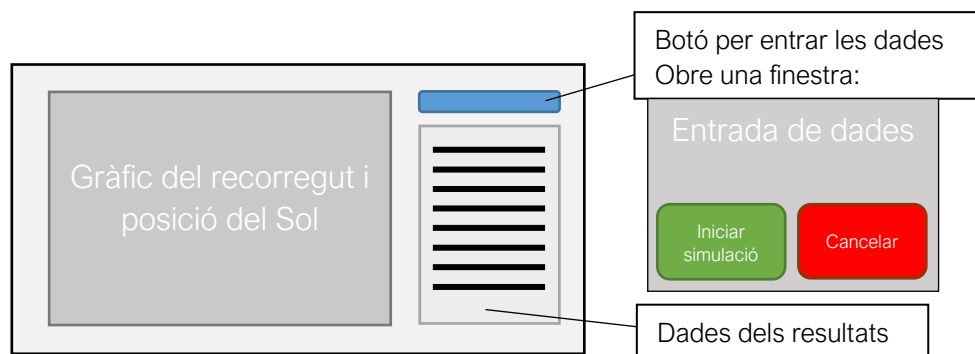


Figura 12. Esbós del disseny de la pantalla del programa.

¹⁸ Bàsicament, un programa creat amb Qt és capaç d'executar-se tant en Linux com en Windows com en Mac i compta amb una forma bastant intuïtiva per treballar en un programa informàtic.

A la figura 12 es pot veure el primer disseny bàsic del programa. No és gaire diferent del final, ja que l'estructura és la mateixa, però canvia la distribució dels elements. El procés de definició del disseny tant inclou el que es visualitza a la pantalla com l'estructuració del codi. Llavors, de les dues parts bàsiques del codi (la pantalla i les funcions),¹⁹ la part gràfica ha quedat organitzada amb elements tal i com es veu a la imatge de la figura 13.

Cada element té unes propietats per defecte que permeten visualitzar informació, canviar paràmetres o obrir i tancar finestres. El programador és qui regula aquestes propietats. El programa mostra un *canvas*²⁰ buit, amb un *button*²¹ a la part inferior de la pantalla (amb el text "Entrar dades") i que en ser clicat obre un rectangle al centre de la pantalla amb una *grid*²² i tots els elements necessaris on s'entren les dades que caldran per fer els càlculs. Seguidament, un cop clicat el *button* "Iniciar Simulació", el programa té indicat amb algoritmes tot el que ha de fer amb les dades introduïdes i retorna els valors calculats dels angles de l'altura i l'azimut. Cal apuntar que quan s'entren les coordenades geogràfiques de l'observador el programa té en compte si són de l'est o de l'oest, ja que si resulta que la longitud és pertanyent a l'est aquesta dada s'ha de passar a negativa perquè els càlculs tinguin sentit (com explica Jean Meeus al seu llibre).

El programa calcula per a cada segon del dia l'altura i l'azimut del Sol i les dades es retornen en forma de punt on les abscisses són l'azimut i les ordenades són l'altura, per exemple P(230°, 23,4°). Llavors, en el *canvas* es dibuixa cada un dels punts i tots s'uneixen mitjançant línies rectes.

Per poder representar cada punt, el programa necessita saber a quin píxel correspon i els angles no li proporcionen aquesta informació. Per tant, s'han de trobar els píxels corresponents a cada punt fent una regla de proporcionalitat entre l'angle màxim i el total de píxels en l'amplada o l'alçada del *canvas*. Per exemple, si el *canvas* té una amplada de 900 px (píxels), com que l'azimut va de 0° a 360° i es

¹⁹ Les funcions en un programa informàtic són algoritmes que permeten utilitzar-se repetidament. Com quan s'ha de fer un càlcul diversos cops per a dades diferents.

²⁰ El *canvas*, entre d'altres funcions, permet dibuixar línies rectes o corbes, figures simples o complexes, gràfics i imatges, o incloure textos.

²¹ Els *buttons* són elements que en ser clicats funcionen com un comandament per realitzar accions.

²² Les *grids* són graelles que permeten disposar elements en línies i columnes.

representa a l'eix d'abscisses, es trobarà el píxel corresponent als 230° segons: $px_a = \frac{230^\circ \cdot 900 \text{ px}}{360^\circ} = 575$ (el píxel 575 es correspon a 230°). També s'ha de tenir en compte que l'origen de coordenades d'un *canvas* no està situat a baix i a l'esquerra com en un eix de coordenades cartesianes, sinó que està a dalt a l'esquerra. Per aquest motiu, el valor de l'altura s'ha d'invertir per tal de ser representada correctament.

Si es vol representar l'altura de $23,4^\circ$ primer s'ha de trobar el píxel corresponent a aquest angle. Suposant que l'altura del *canvas* és de 300 px, com que l'angle de l'altura va de 90° a -90° , $+23,4^\circ$ en un rang de 0° - 180° seran $h = 90^\circ + 23,4^\circ = 113,4^\circ$. Llavors, el píxel corresponent seria $px_h = \frac{113,4^\circ \cdot 300 \text{ px}}{180^\circ} = 189$. Però com que l'origen de coordenades del *canvas* és a la cantonada superior esquerra i no a la inferior (com a les coordenades cartesianes), perquè dibuixi el punt a l'altura que toca en realitat el píxel ha de ser: $px_h = 300 - 189 = 111$. Després de trobar l'equivalència de tots els angles de tots els punts en píxels, es dibuixen, s'uneixen amb línies i s'obté la línia groga que indica el recorregut del Sol durant el dia en qüestió.

Seguidament, si es vol dibuixar el Sol a una hora concreta, només s'han d'entrar les dades i el programa calcularà l'azimut i l'altura del Sol en l'instant concret. La imatge es col·loca als píxels corresponents utilitzant el mètode explicat anteriorment. Si quan s'entren totes les dades no s'especifica una hora concreta i s'inicia la simulació, llavors el programa directament no col·locarà la imatge del Sol en el *canvas*.

En el moment de clicar el *button* "Iniciar Simulació" també s'actualitza la *grid* de la part inferior amb les dades obtingudes dels càlculs. Per tant, surten per pantalla les dades entrades de la localització, el dia i l'hora (si s'ha especificat) més les hores de sortida i posta del Sol amb les coordenades de la seva posició que el programa ha calculat.

És interessant interpretar la línia recta horitzontal que es dibuixa i que uneix el primer i l'últim punt del recorregut del Sol. Aquesta línia simplement apareix perquè l'últim azimut amb un valor de 360° o aproximadament, segueix amb l'azimut de 0°

a l'altra banda del *canvas* i la línia que els uneix travessa el *canvas* d'una banda a l'altra. Per tant, el que indica és que el Sol fa un recorregut circular. L'únic cas en què no passaria és quan els azimuts de 0° i 360° coincidissin amb les 00:00 h i les 24:00 h del dia respectivament; llavors el programa ja no dibuixaria aquesta última línia.

Tots els elements que es veuen a la figura 13 es poden veure escrits dins el codi font a l'annex I.

2. FUNCIONS

Després del procés de disseny comença la codificació. És necessari afegir que aquest és el procés on apareixen tots els inconvenients i on, moltes vegades, s'han de canviar les estratègies. Per exemple, moltes de les característiques que tenen relació amb el *canvas* de l'apartat anterior no s'havien tingut en compte en fer el disseny però es van haver de resoldre durant la codificació.

Les funcions (vegeu annex I) es van crear totes de forma independent en C++ ja que és més ràpid i senzill de processar per al compilador de l'ordinador. Segons les accions o càlculs que realitzen es poden distingir en tres grans blocs: les de caràcter general, les de càlculs matemàtics i les que s'utilitzen per incloure un perfil topogràfic local.

2.1. FUNCIONS GENERALS

Les funcions que es consideren generals són aquelles que s'utilitzen en casos molt concrets però que apareixen i es fan servir en altres funcions per fer canvis d'unitat o processos mecànics que s'haurien d'escriure en cada cas repetint codi. Les següents són les que apareixen al codi font:

- **numMes():** a partir del nom d'un mes troba el seu número corresponent (gener: 1; febrer: 2; març: 3...).

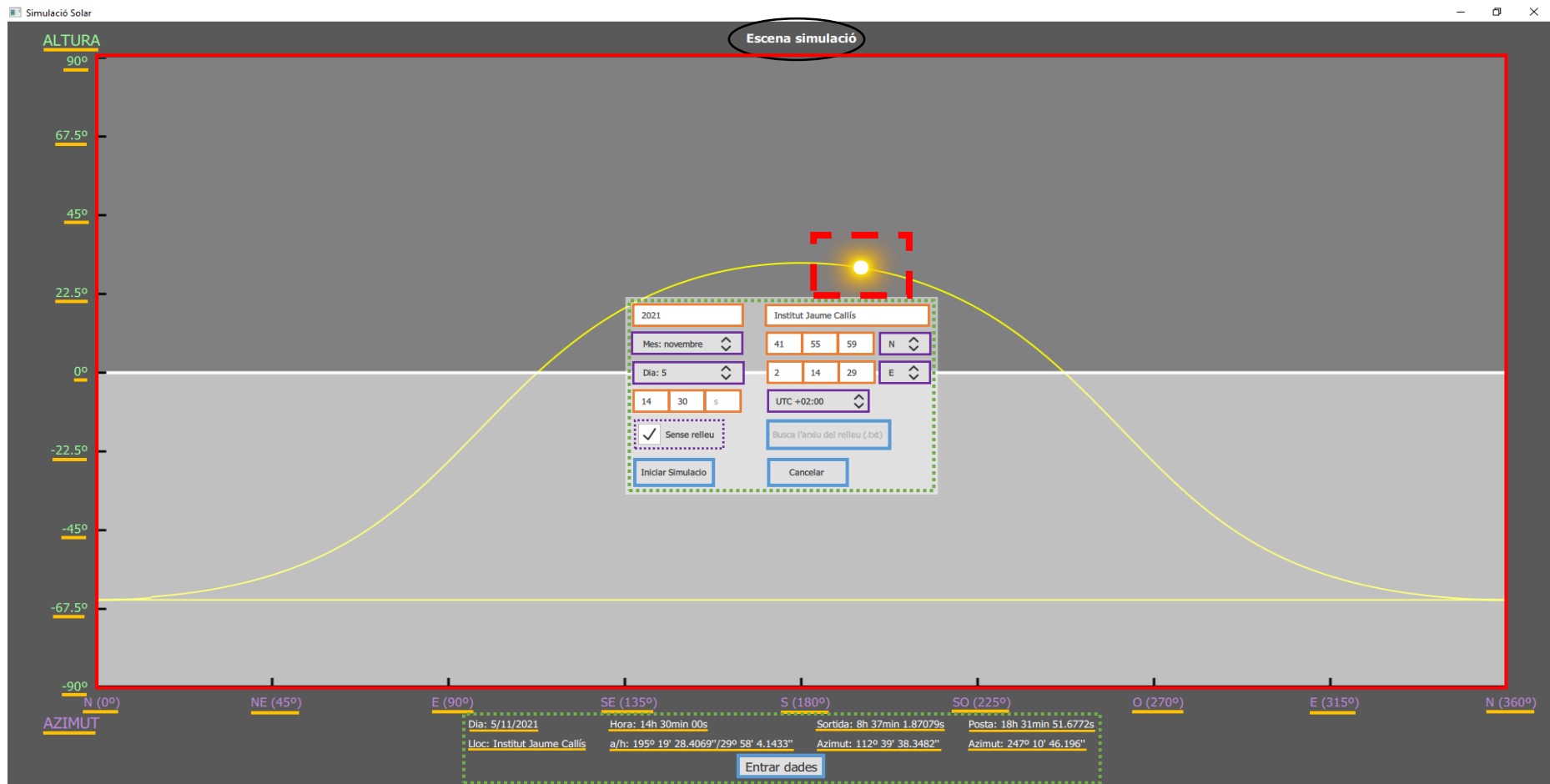


Figura 13. Disseny de la pantalla de la simulació.

Image	Text	Canvas	Label	Grid	Button	Text Field	Combo Box	Check Box

- **passarADecimals()** i **passarASeparats()**: a partir d'un valor de graus o hores separats en graus/hores, minuts i segons els passa tots a graus o hores en decimals o viceversa (195,32451°: 195° 19' 28,236").
- **treureVoltes()**: fa el procés de reduir un angle qualsevol a un rang de 0°-360° (1 326°: 246°).
- **horesAGraus()** i **grausAHores()**: converteix graus a hores o viceversa (15° equivalen a una hora).
- **guardarArxiu()** i **llegirArxiu()**: són dues funcions que el que fan és guardar les dades entrades en cada moment a la simulació dins la memòria de l'ordinador per no haver-les d'entrar repetidament (és pràctic si només es volen canviar algunes dades concretes i no totes).

2.2. FUNCIONS DE CÀLCULS MATEMÀTICS

És en aquestes funcions on s'han aplicat les fórmules i mètodes per acabar calculant l'altura i l'azimut i les dades que apareixen per pantalla un cop s'inicia la simulació. Les funcions són:

- **diaJulia()** i **dataJulia()**: troben el dia julià d'una data i la data a partir d'un dia julià, respectivament.
- **anyTraspas()**: indica si un any és de traspàs o no (retorna un 1 si l'any entrat és de traspàs o un 2 si l'any és comú).
- **solsticisEquinoccis()**: troba la data concreta del solstici o equinocci indicat (any i mes).
- **declinacioSolar()**, **tT()**, **o0()**, **obliquitat()**, **ascensioRecta()**: **trobarGMST()**, **trobarGMST()**, **angleHorari()**: realitzen els càlculs pertinents de la declinació solar, variables com T , la longitud vertadera del Sol, l'obliquïtat, l'ascensió recta, el temps sideral mitjà de Greenwich i l'angle horari.
- **alturaAzimut()**: és la funció principal ja que utilitza totes les anteriors (de forma directa o indirecta) per arribar a trobar l'altura i l'azimut del Sol. Troba per a cada segon del dia, l'altura i l'azimut corresponents i els adjudica a un punt (del gràfic).

L'última funció també és l'encarregada de trobar les dades de la sortida i la posta de Sol. És important el fet que aquestes dades no es calculen expressament a partir d'una fórmula, sinó que simplement es fan comparacions amb la resta de dades. És a dir, per poder trobar la posició de la sortida del Sol, s'ha de buscar de tots els punts el primer en què l'altura passa de negativa a positiva (de sota a sobre l'horitzó) i a partir d'aquesta dada es troben l'azimut d'aquell punt per saber el lloc on sortirà i també el dia julià exacte en el qual el Sol està en aquella posició. Segons aquesta última dada, si es troba la data corresponent a aquell dia julià i es passen els decimals del dia a hores, aquell serà l'instant exacte de la sortida. El mateix passa amb la posta, però llavors s'ha de trobar el punt en el qual l'altura passa de ser positiva a negativa.

Un cas especial és el d'algunes zones del món on el Sol o no surt durant una part de l'any o mai no es pon (zones més extremes dels pols). En aquests casos, es detecta que l'altura no passa en cap moment de negativa a positiva i que no hi ha ni sortida ni posta del Sol.

El darrer pas per al desenvolupament de la simulació és la comprovació. Un cop implementades totes les funcions, el programa és capaç de realitzar els càlculs necessaris i retornar els valors correctes. Això s'ha comprovat comparant els resultats del programa amb els mateixos càlculs realitzats a mà.

El programa tarda uns segons a poder trobar totes les dades i a representar-les al *canvas*. Això passa perquè ha de calcular per a tots els segons del dia (86.400 segons) l'azimut i l'altura del Sol i, a més a més, comprovar si cada un d'ells és el punt de sortida o posta del Sol. Per accelerar la comprovació de càlculs concrets es va crear un *button* temporal a través del qual es feien les comprovacions del correcte funcionament de les funcions sense haver de realitzar tots els càlculs, cosa que alentiria el procés de comprovació.

2.3. FUNCIONS PER INCLOURE UN PERFIL TOPOGRÀFIC LOCAL

Un cop comprovat el bon funcionament de la simulació en aquell punt, es va procedir a intentar incorporar un element que altres programes que realitzen càlculs

similars no tenen: la possibilitat d'afegir-hi un perfil topogràfic local. És evident que la sortida i la posta de Sol no tindran lloc en el mateix moment vistes des de la platja (a nivell del mar on el Sol es veu a partir d'alçades més grans que 0) que, per exemple, en una vall, on les muntanyes o altres formes de relleu a l'horitzó condicionen la visibilitat del Sol i els altres astres.

El fet d'afegir-hi un relleu només suposa una elevació de l'horitzó. A conseqüència, la sortida i la posta no tindran lloc en els punts on l'altura passa de ser positiva a negativa o viceversa, sinó que serà en el moment que l'altura del Sol sigui superior (en la sortida) o inferior (en la posta) a l'altura del relleu en el mateix azimut. De totes maneres, s'han de tenir en compte més factors ja que el programa pot detectar més d'una sortida i una posta. Per exemple, si el Sol en un determinat relleu s'amaga darrere una muntanya després d'haver sortit i es torna a veure una estona després un cop travessada la muntanya, el programa interpretarà que hi ha més d'una sortida o posta ja que l'altura passarà a ser més gran que la del relleu en més d'una ocasió. Per poder fer entendre aquests casos al programa, un punt es correspondrà amb la sortida si: *a*) el Sol estava prèviament amagat ($\text{altura} < \text{altura relleu}$); *b*) si l'altura passa a ser més gran que la del relleu, i *c*) si abans no hi ha hagut cap altre punt en què passés el mateix (d'aquesta forma es troba només el primer cop que surt). Per a la posta, les condicions són contràries: el Sol ha d'estar per sobre del relleu quan aquest s'amaga ($\text{altura} > \text{altura relleu}$) i ja hi ha hagut una sortida prèviament.

Un cop estudiat de quina manera la implementació del relleu afectarà la simulació, cal saber l'altura del relleu. La forma més senzilla i exacta d'aconseguir-ho és mesurant l'angle de visió del relleu respecte de l'observador. Utilitzant aquest mètode, no importa la distància a la qual es trobi una muntanya ni la seva alçada ja que l'angle mateix és l'altura del relleu que el Sol ha de superar per ser visible per l'observador.

Abans d'intentar mesurar un relleu real, però, es va utilitzar un relleu inventat com el de la figura 14. En aquesta es pot veure l'altura de l'angle de visió del relleu per cada 10° de l'azimut. Amb aquestes dades es va crear un fitxer d'extensió '.txt' amb l'azimut i l'altura de cada punt col·locats l'un sota l'altre (el format es pot veure a l'annex II). El programa el que fa és demanar un arxiu amb el format esmentat

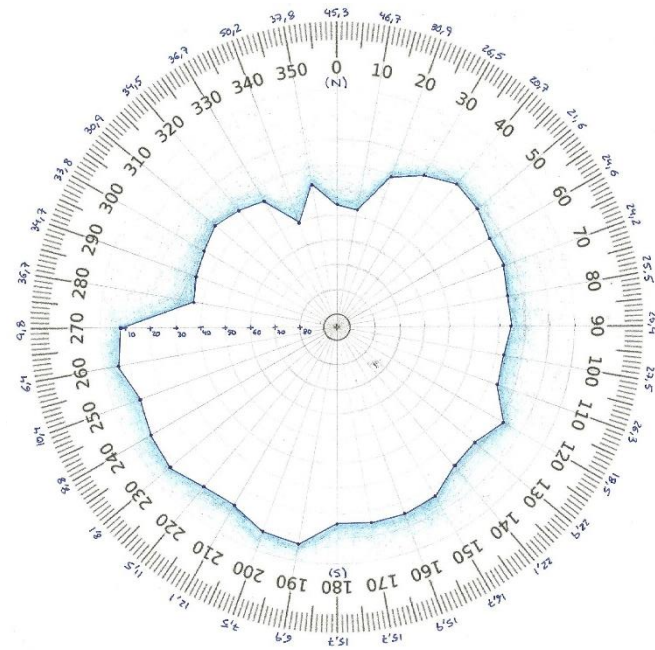


Figura 14. Representació d'un relleu no real.

i crea i dibuixa els punts en el *canvas* per a finalment unir-los entre ells amb línies. La funció encarregada d'interpretar l'arxiu en el codi font (annex I) s'anomena *llegirArxiuRelleu()*.

2.3.1. Interpolació lineal

La simulació amb el relleu dibuixat, tot i tenir les dades de punts concrets, no pot saber tots els punts que conformen tota la silueta del relleu pel fet que ha estat dibuixat de forma automàtica unint cada punt amb el següent. Però per poder comparar l'altura del Sol amb la del relleu en cada segon del dia és necessari tenir les dades de tots els punts. Per tant, s'utilitza el mètode de la interpolació lineal per poder trobar les altures desconegudes de tot el recorregut de l'azimut.

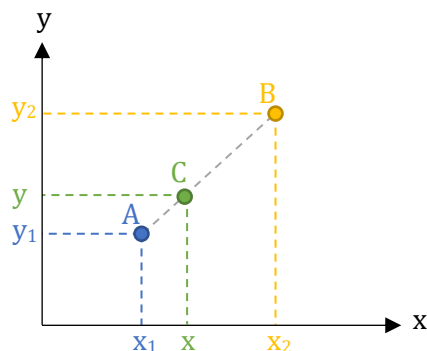


Figura 15. Gràfic d'interpolació lineal.

A la figura 15, són coneguts els punts $A(x_1, y_1)$ i $B(x_2, y_2)$ però es vol conèixer el valor de la y del punt $C(x, y)$. Tenint en compte que els punts pertanyen tots a la mateixa recta, el pendent entre els punts A i C serà el mateix que entre A i B , per tant, $\frac{y-y_1}{x-x_1} = \frac{y_2-y_1}{x_2-x_1}$.

En aquesta fórmula, si s'aïlla y , es troba el seu valor a partir de les coordenades dels punts A i B i de la coordenada x desitjada (entre aquests dos punts A i B).

En el cas del relleu, l'altura h en un azimuth a , serà interpolada a partir dels dos punts coneguts del relleu més propers [$P_1(a_1, h_1)$ i $P_2(a_2, h_2)$]:

Equació 13. Interpolació lineal de l'altura en un azimuth desconegut.

$$h = h_1 + \left(\frac{h_2 - h_1}{a_2 - a_1} \right) (a - a_1)$$

A la funció *trobarAlturaRelleu()* s'utilitza la fórmula anterior per trobar a cada punt del relleu l'altura corresponent per a després comparar-la amb la del Sol a la funció *alturaAzimut()* i saber si en aquell punt tenen lloc la sortida o la posta.

2.3.2. Mesura d'un relleu real

Després de comprovar que amb relleu o sense les hores de sortida i posta variaven, es va procedir a intentar mesurar un relleu real. El procediment va ser el següent:

- 1) Primer es va fer una fotografia en format panoràmica del relleu que es volia mesurar (s'ha de tenir en compte fer-la a l'altura de la vista):



Figura 16. Panoràmica del relleu.

- 2) El segon pas va ser mesurar amb una aplicació de mòbil l'angle entre dos punts visibles en la panoràmica:



Figura 17. Mesura d'un angle en el relleu.

3) Seguidament es van situar i marcar en la panoràmica amb l'ajuda d'una brúixola els punts cardinals (en groc) i els punts del relleu més destacats (en vermell):



Figura 18. Panoràmica del relleu amb els punts cardinals i els punts destacats del relleu marcats.

4) Finalment, utilitzant la relació entre l'angle mesurat i la distància entre els dos punts que el delimiten en el paper (en cm), es va obtenir la relació de proporcionalitat entre centímetres i angles i es van calcular la resta de valors de cada punt. Aquests valors es van posar en un fitxer '.txt' el resultat del qual es troba a l'annex II (en el fitxer «RelleuReal.txt»).



Figura 19. Proporció graus-centímetres.

2.3.3. Comprovacions

Finalment, l'últim procés després de comprovar que el programa detectés bé les hores de sortida i de posta amb un relleu (tant el real com el fals), es va fer una comprovació en temps real de les dades obtingudes a través del programa. El dia 12 d'octubre de 2021 es va fer una observació des del lloc on s'havia mesurat el relleu per veure si realment coincidien l'hora de sortida i de posta del programa amb les reals tenint en compte el perfil topogràfic mesurat. Aquests van ser els resultats:

Hora en el programa:

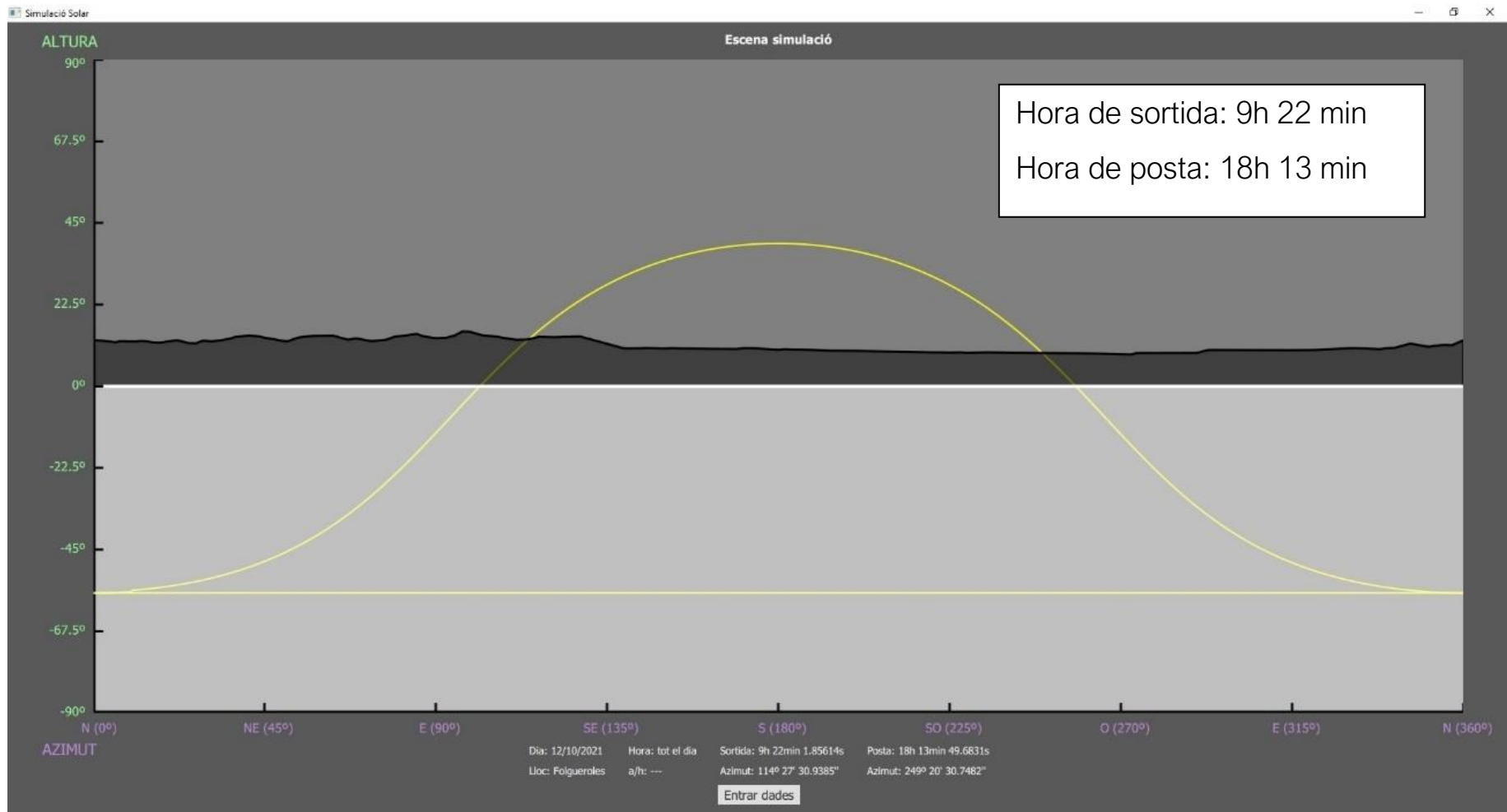


Figura 20. Captura de pantalla dels resultats del programa el 12 d'octubre de 2021.

Hora real:

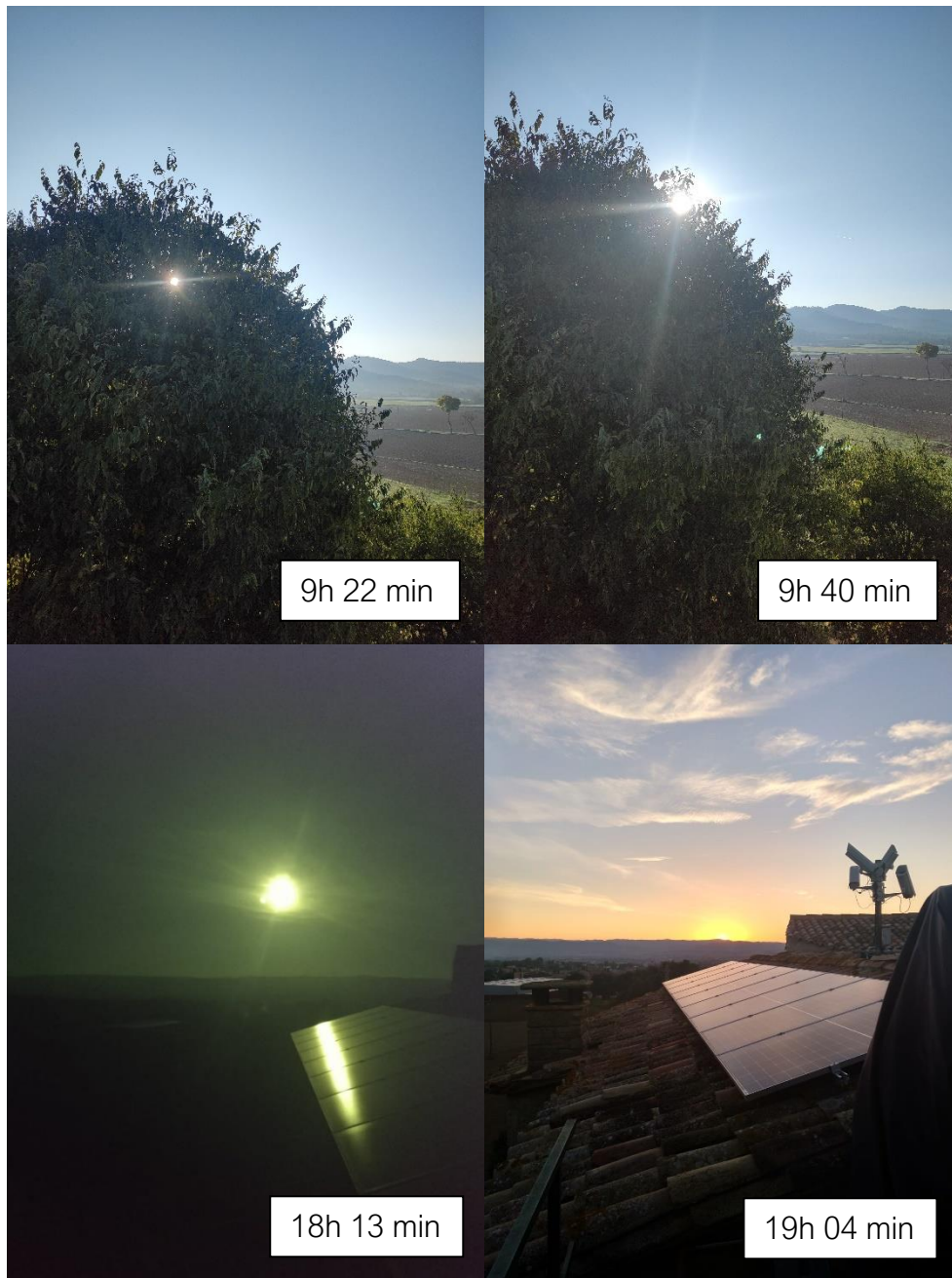


Figura 21. Imatges de la sortida i posta de Sol a l'hora teòrica i la real.

Com es pot observar, a la sortida hi va haver una diferència de 18 minuts entre l'hora teòrica i la real, però a la posta aquesta diferència va ser de pràcticament d'una hora. Per tant, de les hores calculades és més aproximada la de la sortida que la de la posta.

Com es pot veure, la foto de les 18:13 h té unes tonalitats diferents ja que va ser feta a través d'un filtre de soldador per neutralitzar els rajos del Sol i fer la imatge més nítida.

Reunint totes les dades obtingudes del programa, es pot dir que el resultat ha estat considerablement bo. L'error en les hores calculades respecte de les reals mostra que el programa no és prou precís. De totes maneres, es va decidir comparar els càlculs del programa amb els de l'Stellarium, un programari lliure també disponible en format web que realitza càlculs relacionats amb les efemèrides d'estrelles i planetes. La comparació s'ha de fer sense tenir en compte el relleu de la localització de l'observador ja que l'Stellarium no permet contemplar-lo. Tot i així, amb les dades de la localització (latitud i longitud) i l'alçada a la qual es va realitzar l'observació (el 12 d'octubre de 2021) es van obtenir les dades següents:

Amb el programa creat:

Dia: 12/10/2021	Hora: tot el dia	Sortida: 8h 8min 31.8801s	Posta: 19h 5min 10.6664s
Lloc: Folgueroles	a/h: ---	Azimut: 101° 37' 14.2211"	Azimut: 258° 8' 53.9753"

[Entrar dades](#)

Figura 22. Dades (sense relleu) obtingudes del dia 12 d'octubre del 2021 amb el programa creat.

Amb l'Stellarium:

Sol

Tipus: **estel**
Magnitud: **-26.75** (reduit a **-26.97** per **2.88** Masses d'aire)
Magnitud Absoluta: 4.83
AR/Dec (J2000.0): 13h10m55.18s/-7°31'21.1"
AR/Dec (en data): 13h12m02.30s/-7°38'11.7"
AH/Dec: 3h35m13.19s/-7°36'08.9" (aparent)
Az./Alt.: +238°32'00.2"/+20°18'34.7" (aparent)
Long./lat. gal.: -48°37'23.8"/+55°01'46.1"
Long./lat. Supergal.: +124°46'14.0"/+1°45'38.5"
Long./lat. ecl. (J2000.0): +199°12'44.7"/-0°00'04.3"
Long./lat. ecl. (en data): +199°30'43.9"/-0°00'08.4"
Obliquïtat de l'eclíptica (en data): +23°26'15.6"
Temps Sideral Mitjà: 16h47m23.4s
Temps Sideral Aparent: 16h47m22.4s
Sortida: 7h 59m
Trànsit: 13h 37m
Posta: 19h 15m
Dia: 11n10m
Angle paral·làctic: +39°48'27.9"
Constel·lació UAI: Vir
Moviment cada hora: +0°02'27" cap a 112.2°
Moviment cada hora: d α =+0°02'17" d δ =-0°00'56"
Distància: 0,998 UA (149,287 M km)
Temps de llum: 0h08m18.0s
Diàmetre aparent: +0°32'02.44"
Diàmetre: 1391400.0 km
Dia sideral: 609h07m11.6s
Velocitat de rotació equatorial: 1.993 km/s

Data i hora: 2021 - 10 - 12 Dia Julià: 17 : 12 : 41

Sortida: 7h 59 min
Posta: 19h 15 min

Figura 23. Dades (sense relleu) obtingudes del dia 12 d'octubre de 2021 amb l'Stellarium.

Com es pot comprovar a les figures 22 i 23, al final l'error dels càlculs és només de 10 minuts. En definitiva, són més aproximats. El més probable és que aquest error sigui degut a no haver tingut en compte algun factor extern que afecta la posició del Sol vist des de la Terra; de la mateixa manera que no es tenen en compte la pressió ni la temperatura del lloc on és l'observador. Pel que fa a l'error quan es fan els càlculs incloent-hi el relleu (i a la diferència d'error entre la sortida i la posta), el més probable és que sigui degut a una mala mesura del relleu i no pas a un càlcul erroni. Els resultats són bastant aproximats (com s'ha comprovat amb l'Stellarium) i haurien de tenir un marge d'error més o menys similar quan s'hi afegeix l'alçada del relleu (si aquest s'ha mesurat bé).

El que sí que s'ha pogut apreciar a través del programa ha estat la variació del recorregut del Sol al llarg de l'any i la seva trajectòria vista des de la Terra. A la figura 24 es pot apreciar aquest canvi del seu recorregut. I a la figura 25, el canvi de la trajectòria en un mateix dia (5 de novembre de 2021) en diferents punts de la Terra: a Reykjavík, Islàndia (a prop del pol Nord) i a Nairobi, Kenya (pràcticament a la línia de l'equador terrestre).

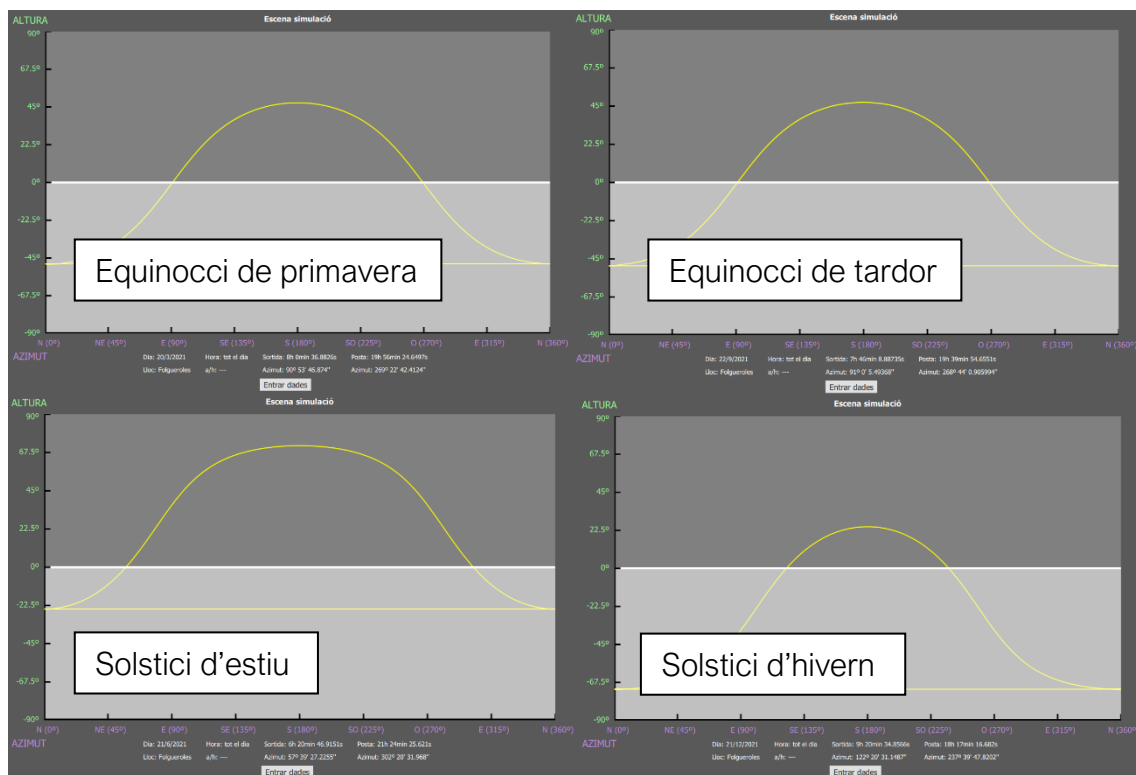


Figura 24. Recorregut del Sol als solsticis i equinoccis de l'any 2021.

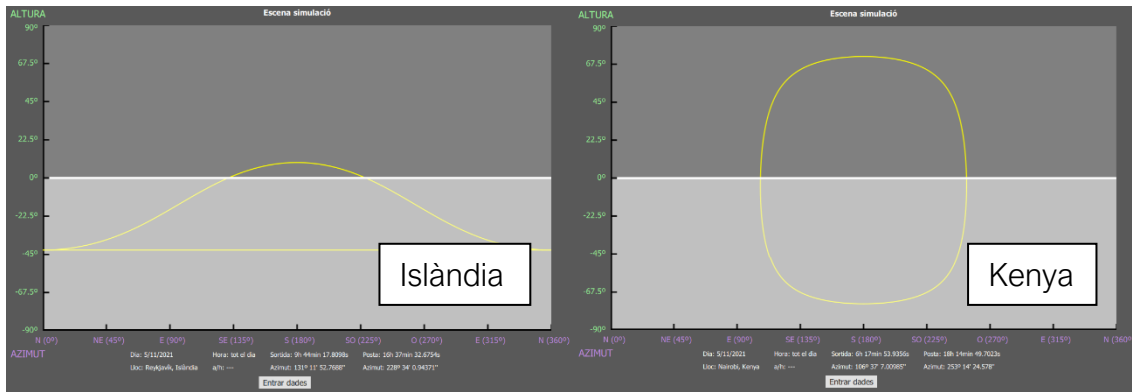


Figura 25. Recorregut del Sol el 5 de novembre de 2021 vist des de Reykjavík (Islàndia) i Nairobi (Kenya).

Aquestes comprovacions es poden fer amb el programa descarregant-lo a través del següent enllaç:

[https://drive.google.com/file/d/1r2GeDlDa8FtzPS9SHypKO1r0PDFbbQO0/vi
ew?usp=sharing](https://drive.google.com/file/d/1r2GeDlDa8FtzPS9SHypKO1r0PDFbbQO0/vi
ew?usp=sharing)

CONCLUSIONS

Després de realitzar totes les comprovacions i comparacions esmentades, puc dir que estic orgullosa del resultat ja que la simulació funciona i inclou tots els elements que m'havia plantejat en un principi. La implementació dels càlculs, la representació gràfica dels resultats i l'opció de tenir en compte el perfil topogràfic local. Un dels fets que he trobat més interessant és com uns càlculs relativament senzills són capaços de trobar concretament l'instant en què surt o es pon el Sol. De totes maneres també he pogut comprovar de primera mà les complicacions que comporta aplicar-los a la realitat i més després de mesurar el relleu local i veure que les dades no eren prou exactes. Amb això vull dir que la teoria és molt diferent de la pràctica i normalment suposa un seguit de complicacions que s'han de saber resoldre de forma intuïtiva i no basant-se en càlculs ni algoritmes establerts. És un punt de millora aconseguir que els càlculs siguin més exactes segons l'observació a temps real de la sortida i la posta del Sol. Pel que fa al meu treball seria interessant tornar a fer una mesura del relleu local per comparar amb la primera mesura. D'aquesta manera es podria confirmar si l'error va ser donat perquè vaig mesurar el relleu incorrectament o perquè algun càlcul era erroni.

A tot això s'ha d'afegir que he adquirit molts coneixements relacionats amb la programació, sobretot en l'àmbit de disseny i d'interacció amb la pantalla. Fins al moment no havia programat res semblant i el treball m'ha permès aprofundir molt en la programació tant en C++ com en QML.

Puc concloure el treball amb l'afirmació que el resultat final ha estat un èxit, tenint en compte que, amb més o menys precisió, he pogut assolit tots els objectius.

FONTS DOCUMENTALS

MEEUS, Jean. *Astronomical algorithms*. Richmond, Virginia: Willmann-Bell, Inc., 1991.

DE ROOIJ, Dricus. *Declination Angle*. <<https://sinovoltaics.com/learning-center/basics/declination-angle/>> [Última consulta: 26 setembre 2021].

ScienceDirect. *Solar Declination*. <<https://www.sciencedirect.com/topics/engineering/solar-declination>> [Última consulta: 31 juliol 2021].

TANNER, Jay. *Obliquity of the Ecliptic*. <https://www.neoprogrammics.com/obliquity_of_the_ecliptic/Obliquity_Of_The_Ecliptic_Calculator.php> [Última consulta: 29 setembre 2021].

POLLOCK, Robert. *Dates of equinoxes and solstices, 1997 – 2030 AD*. <<https://stonesofwonder.com/stones7.htm>> [Última consulta: 26 setembre 2021].

The Qt Company. *Qt Documentation*. <<https://doc.qt.io/>> [Última consulta: 25 setembre 2021].

CHÉREAU, Fabien i Guillaume. *Stellarium Web*. <<https://stellarium-web.org/>> [Última consulta: 26 setembre 2021].

ZOTTI, Georg i Wolf, Alexander. *Stellarium 0.21.1 User Guide* [Arxiu PDF]. <https://www.researchgate.net/publication/350449204_Stellarium_0210_User_Guide> [Última consulta: 26 setembre 2021]

KARNEY, Kevin. (Setembre de 2014). *Basic Solar Positional Astronomy* [Arxiu PDF]. <<http://www.precisedirections.co.uk/Sundials/Basic%20Solar%20Positional%20Astronomy.pdf>> [Última consulta: 23 agost 2021].

JoshTheEngineer. *Explained: Linear Interpolation [Math]*. <https://www.youtube.com/watch?v=Cvc-XaIN_kk> [Última consulta: 13 octubre 2021].

Smart Tool Factory. *Angle Meter* (1.3.0) [Aplicació mòbil].

<<https://play.google.com/store/apps/details?id=com.stfactory.anglemeter&gl=ES>>

[Última consulta: 28 octubre 2021].

Viquipèdia. <<https://ca.wikipedia.org/wiki/Portada>> [Última consulta: 27 octubre 2021].

ANNEX I – CODI FONT

RCalculadora.h

```
#ifndef RCALCULADORA_H
#define RCALCULADORA_H
#include <QString>
#include <QObject>
#include <QVariantList>

class RCalculadora : public QObject
{
    Q_OBJECT
private:
    int iError;
    QString sMissatgeError;
    bool bErrorSortida;

public:
    RCalculadora(QObject * parent = nullptr);

    Q_PROPERTY(int error READ error WRITE setError NOTIFY
errorChanged)
    int error(){return iError;}
    void setError(int iPErrer){iError=iPErrer; emit
errorChanged(iPErrer);}
    Q_PROPERTY(QString missatgeError READ missatgeError WRITE
setMissatgeError NOTIFY missatgeErrorChanged)
    QString missatgeError(){return sMissatgeError;}
    void setMissatgeError(QString
sPMissatgeError){sMissatgeError=sPMissatgeError; emit
missatgeErrorChanged(sPMissatgeError);}
    Q_PROPERTY(bool errorSortida MEMBER bErrorSortida)

    QVariantList datesRelleu;

    Q_INVOKABLE QString numMes (QString mes);
    Q_INVOKABLE double passarADecimals(QString graus, QString minuts,
QString segons);
    Q_INVOKABLE QString passarASeparats (bool graus, double decimals,
double horaUTC = 0.);
    Q_INVOKABLE double treureVoltes(double dGraus);
    Q_INVOKABLE double horesAGraus(double hores, double minuts, double
segons);
    Q_INVOKABLE double grausAHores(QString sGraus, double dMinuts,
double dSegons);
    Q_INVOKABLE double diaJulia(double dia, QString mes, int any, int
hora = 0, int minut = 0, double segon = 0.);
    Q_INVOKABLE QString dataJulia(double dDJ);
    Q_INVOKABLE int anyTraspas(int iAny); // 1: any de traspàs; 2: any
comú.
    Q_INVOKABLE QString solsticisEquinoccis(int iAny, QString sMes);
    Q_INVOKABLE double declinacioSolar(QString sData);
    Q_INVOKABLE double tT(double dDJ);
```



```

    Q_INVOKABLE double o0(double dT);
    Q_INVOKABLE double obliquitat(int iAny);
    Q_INVOKABLE double ascensioRecta(double obliquitat, double d0);
    Q_INVOKABLE double trobarErrorGMST(double T, double obliquitat);
    Q_INVOKABLE double trobarGMST(double dDJ, double dT, double
errorGMST);
    Q_INVOKABLE double angleHorari(double dGMST, double dLongitud,
double dAscensioRecta);
    Q_INVOKABLE QVariantList alturaAzimut(double dDJ, QString hora,
int any, double dLatitud, double dLongitud, bool bTrobarPosicio, bool
calcularRelleu);
    Q_INVOKABLE void guardarArxiu(QString any, QString mes, QString
dia, QString localitzacio, QString latGraus, QString latMinuts,
QString latSegons, QString lonGraus, QString lonMinuts, QString
lonSegons, QString ns, QString eo, QString hora, QString minut,
QString segon, int indexUTC);
    Q_INVOKABLE QVariantList llegirArxiu();
    Q_INVOKABLE QVariantList llegirArxiuRelleu(QString arxiuRelleu);
    Q_INVOKABLE double trobarAlturaRelleu(double azimut);

signals:
    void errorChanged(int iPErrer);
    void missatgeErrorChanged(QString sPMissatgeError);

};

#endif // RCALCULADORA_H

```

```

#include "RCalculadora.h"
#include <cmath>
#include <QProcessEnvironment>
#include <QFile>
#include <QDir>
#include <QSettings>
#include <QUrl>
#include <QTextStream>
#include <QDebug>

RCalculadora::RCalculadora(QObject * parent):QObject(parent)
{

}

QString RCalculadora::numMes(QString mes)
{
    QStringList mesos = {"gener", "febrer", "març", "abril", "maig",
"juny", "juliol", "agost", "setembre", "octubre", "novembre",
"desembre"};
    QString sMes = QString::number(mesos.indexOf(mes)+1);
    return (sMes);
}

double RCalculadora::passarADecimals(QString grausHores, QString
minuts, QString segons)
{
    double g = grausHores.toInt();
    double min = minuts.toInt();
    double s = segons.toDouble();
    return(g+min/60+s/3600);
}

QString RCalculadora::passarASeparats(bool graus, double decimals,
double horaUTC)
{
    int num1 = trunc(decimals);
    double n1 = (decimals-num1)*60.;
    int num2 = trunc(n1);
    double num3 = (n1-num2)*60;

    double hora = num1+horaUTC;
    if (hora<0)
        hora += 24;
    if (hora>24)
        hora -= 24;

    if (graus)
        return (QString::number(num1)+"° "+QString::number(num2<0?-
num2:num2)+"' "+QString::number(num3<0?-num3:num3)+"'");
    else
        return (QString::number(hora)+"h "+QString::number(num2<0?-
num2:num2)+"min "+QString::number(num3<0?-num3:num3)+"s");
}

```

```

}

double RCalculadora::treureVoltes(double dGraus)
{
    dGraus -= 360*(trunc(dGraus/360));
    if (dGraus<0)
        dGraus+=360;
    return (dGraus);
}

double RCalculadora::horesAGraus(double hores, double minuts, double
segons)
{
    return((hores+(minuts/60)+(segons/3600))*15);
}

double RCalculadora::grausAHores(QString sGraus, double dMinuts,
double dSegons)
{
    double dGraus;
    if (sGraus[0]=='-')
    {
        dMinuts=-abs(dMinuts);
        dSegons=-abs(dSegons);
    }
    dGraus = sGraus.toDouble();

    return(dGraus+(dMinuts/60)+(dSegons/3600));
}

double RCalculadora::diaJulia(double dia, QString mes, int any, int
hora, int minut, double segon)
{
    double dDia = dia;
    int iMes;
    int iAny = any;
    int iA;
    int iB;
    QString calendari = ((iAny>1582 || (iAny==1582 && dDia>23 &&
mes=="febrer"))?"gregorià":"julià");
    QStringList mesos = {"gener", "febrer", "març" , "abril", "maig",
"juny", "juliol", "agost", "setembre", "octubre", "novembre",
"desembre"};
    iMes = mesos.indexOf(mes)+1;
    if (iMes==1 || iMes==2)
    {
        --iAny;
        iMes+=12;
    }
    iA=trunc(iAny/100);
    iB = (calendari=="gregorià" ? 2-iA+trunc(iA/4):0);

    return((trunc(365.25*(iAny+4716))+trunc(30.6001*(iMes+1))+dDia+iB-
1524.5)+hora/24.+minut/1440.+segon/86400.);
}

```

```

QString RCalculadora::dataJulia(double dDJ)
{
    dDJ += 0.5;
    QStringList mesos = {"gener", "febrer", "març", "abril", "maig",
"juny", "juliol", "agost", "setembre", "octubre", "novembre",
"desembre"};

    int z = trunc(dDJ);
    double F = dDJ-trunc(dDJ);
    int iAlfa= trunc((z-1867216.25)/36524.25);
    int iA = ((z < 2299161) ? (z) : (z + 1 + iAlfa - trunc(iAlfa/4)));
    int iB = iA + 1524;
    int iC = trunc((iB-122.1)/365.25);
    int iD = trunc(365.25*iC);
    int iE = trunc((iB-iD)/30.6001);

    double dDia = iB - iD - trunc(30.6001*iE) + F;
    int iMes = ((iE<14) ? (iE-1) : (iE-13));
    int iAny = ((iMes>2) ? (iC-4716) : (iC-4715));

    QString sDia = QString::number(dDia);
    QString sMes = mesos[iMes-1];
    QString sAny = QString::number(iAny);
    QString sData = (sDia+" "+sMes+" "+sAny);

    return(sData);
}

int RCalculadora::anyTraspas(int iAny)
{
    if (iAny%4==0)
    {
        if (iAny%100==0)
        {
            if (iAny%400==0)
                return(1);
            else
                return(2);
        }
        else
            return(1);
    }
    else
        return(2);
}

QString RCalculadora::solsticisEquinoccis(int iAny, QString sMes)
{
    QStringList mesos = {"gener", "febrer", "març", "abril", "maig",
"juny", "juliol", "agost", "setembre", "octubre", "novembre",
"desembre"};
    int A[] = {485, 203, 199, 182, 156, 136, 77, 74, 70, 58, 52, 50,
45, 44, 29, 18, 17, 16, 14, 12, 12, 12, 9, 8};

```

```

    double B[] = {324.96, 337.23, 342.08, 27.85, 73.14, 171.52,
222.54, 296.72, 243.58, 119.81, 297.17, 21.02, 247.54, 325.15, 60.93,
155.12, 288.79, 198.04, 199.76, 95.39, 287.11, 320.81, 227.73, 15.45};
    double C[] = {1934.136, 32964.467, 20.186, 445267.112, 45036.886,
22518.443, 65928.934, 3034.906, 9037.513, 33718.147, 150.678,
2281.226, 29929.562, 31555.956, 4443.417, 67555.328, 4562.452,
62894.029, 31436.921, 14577.848, 31931.756, 34777.259, 1222.114,
16859.074};

    int iMes = mesos.indexOf(sMes)+1;

    double dY;
    double tempJD;

    if (iAny>=-1000 && iAny<1000)
    {
        dY = iAny/1000.;

        if (iMes==3)
            tempJD = 1721139.29189 + 365242.13740*dY +
0.06134*(pow(dY,2.0)) + 0.00111*(pow(dY,3.0)) - 0.00071*(pow(dY,4.0));
        else if (iMes==6)
            tempJD = 1721233.25401 + 365241.72562*dY -
0.05323*(pow(dY,2.0)) + 0.00907*(pow(dY,3.0)) + 0.00025*(pow(dY,4.0));
        else if (iMes==9)
            tempJD = 1721325.70455 + 365242.49558*dY -
0.11677*(pow(dY,2.0)) - 0.00297*(pow(dY,3.0)) + 0.00074*(pow(dY,4.0));
        else if (iMes==12)
            tempJD = 1721414.39987 + 365242.88257*dY -
0.00769*(pow(dY,2.0)) - 0.00933*(pow(dY,3.0)) - 0.00006*(pow(dY,4.0));
        else
            return ("En aquest mes no té lloc cap solstici ni
equinocci.");
    }
    else if (iAny>=1000 && iAny<=3000)
    {
        dY = (iAny-2000.)/1000.;

        if (iMes==3)
            tempJD = 2451623.80984 + 365242.37404*dY +
0.05169*(pow(dY,2.0)) - 0.00411*(pow(dY,3.0)) - 0.00057*(pow(dY,4.0));
        else if (iMes==6)
            tempJD = 2451716.56767 + 365241.62603*dY +
0.00325*(pow(dY,2.0)) + 0.00888*(pow(dY,3.0)) - 0.00030*(pow(dY,4.0));
        else if (iMes==9)
            tempJD = 2451810.21715 + 365242.01767*dY -
0.11575*(pow(dY,2.0)) + 0.00337*(pow(dY,3.0)) + 0.00078*(pow(dY,4.0));
        else if (iMes==12)
            tempJD = 2451900.05952 + 365242.74049*dY -
0.06223*(pow(dY,2.0)) - 0.00823*(pow(dY,3.0)) + 0.00032*(pow(dY,4.0));
        else
            return ("En aquest mes no té lloc cap solstici ni
equinocci.");
    }
}

```

```

else
    if (iMes==3)
        return("20.5 març"+QString::number(iAny));
    else if (iMes==6)
        return("20.5 juny"+QString::number(iAny));
    else if (iMes==9)
        return("22.5 setembre"+QString::number(iAny));
    else if (iMes==12)
        return("21.5 desembre"+QString::number(iAny));
    else
        return(" ");

double T = (tempJD-2451545.0)/36525;
double W = (35999.373*T - 2.47)*(M_PI/180);
double K = 1 + 0.0334*cos(W) + 0.0007*cos(2*W);
double S = 0;

for (int i=0; i<24; i++)
{
    double s = A[i]*cos((B[i] + C[i]*T)*(M_PI/180));
    S+=s;
}

double JD = tempJD + ((0.00001*S)/K);
return(dataJulia(JD));
}

double RCalculadora::declinacioSolar(QString sData)
{
    QStringList listData1 = sData.split(" ");
    QStringList listData2 =
(solsticisEquinoccis( (listData1[2].toInt()), "juny")).split(" ");
    double DJ1 = diaJulia( (listData1[0].toDouble()), listData1[1],
(listData1[2].toInt()));
    double DJ2 = diaJulia( (listData2[0].toDouble()), listData2[1],
(listData1[2].toInt()));
    int nDies = (anyTraspas((listData1[2].toInt()))==1 ? 366 : 365);

    double dDeclinacio = (469*M_PI/3600)*cos(2*M_PI*((DJ1-
DJ2)/nDies));
    return (dDeclinacio);
}

double RCalculadora::tT(double dDJ)
{
    return((dDJ-2451545.)/36525);
}

double RCalculadora::o0(double dT)
{
    double L = treureVoltes((280.46645 + 36000.76983*dT +
0.0003032*pow(dT,2.0)));
    double M = treureVoltes((357.52910 + 35999.05030*dT -
0.0001559*pow(dT,2.0) - 0.00000048*pow(dT,3.0)));
    double Mrad = M*(M_PI/180);
}

```

```

    double C = (1.914600 - 0.004817*dT -
0.000014*pow(dT,2.0))*sin(Mrad) + (0.019993 - 0.000101*dT)*sin(2*Mrad)
+ 0.000290*sin(3*Mrad);
    return((L + C)*(M_PI/180));
}

double RCalculadora::obliquitat(int iAny)
{
    return((23.69930-0.00013*(iAny))*(M_PI/180));
}

double RCalculadora::ascensioRecta(double obliquitat, double d0)
{
    return(atan2(cos(obliquitat)*sin(d0),cos(d0)));
}

double RCalculadora::trobarErrorGMST(double T, double obliquitat)
{
    double longitudLluna = (218.3165 + 481267.8813*T)*(M_PI/180);
    double longitudSol = (280.4665 + 36000.7698*T)*(M_PI/180);
    double omega = (125.04452 - 1934.136261*T + 0.0020708*pow(T,2.0) +
pow(T,3.0)/450000.)*(M_PI/180);
    double psi = -17.20*sin(omega) - 1.32*sin(2*longitudSol) -
0.23*sin(2*longitudLluna) + 0.21*sin(2*omega);
    return((psi*cos(obliquitat)/15)*15/3600);
}

double RCalculadora::trobarGMST(double dDJ, double dT, double
errorGMST)
{
    return(treureVoltes(((280.46061837 + 360.98564736629*(dDJ-
2451545.0) + 0.000387933*pow(dT,2.0) - pow(dT,3.0)/38710000) +
errorGMST)));
}

double RCalculadora::angleHorari(double dGMST, double dLongitud,
double dAscensioRecta)
{
    return(treureVoltes(((dGMST*(M_PI/180) - dLongitud -
dAscensioRecta)*(180/M_PI))));
}

QVariantList RCalculadora::alturaAzimut(double dDJ, QString hora, int
any, double dLatitud, double dLongitud, bool bTrobarPosicio, bool
calcularRelleu)
{
    QVariantList altures = {};
    QStringList horaMinutSegon = hora.split(" ");
    dLatitud = dLatitud*(M_PI/180);
    dLongitud = dLongitud*(M_PI/180);
    double dDJinicial = dDJ;
    double dDJfinal = dDJinicial+1;
    double dInterval = 1.;
    double dIntervalDies = dInterval/86400; //Per calcular la posició
del sol en cada segon del dia.
}

```

```

    if (bTrobarPosicio)
    {
        if (horaMinutSegon[0]=="" && horaMinutSegon[1]=="" &&
horaMinutSegon[2]=="")
            return (altures);

dDJinicial+=horaMinutSegon[0].toDouble()/24.+horaMinutSegon[1].toDoubl
e()/1440.+horaMinutSegon[2].toDouble()/86400;
    }

double azimutSortida = 0.;
double horaSortida = 0.;
double alturaTransit = 0.;
double horaTransit = 0.;
double azimutPosta = 0.;
double horaPosta = 0.;

double hMaxim = -91;
bool bSolAmagat = false;
bool bUltimSolAmagat = true;
bool bElSolHaSortit = false;
bool bPrimerSortida = false;
bErrorSortida = false;

for (double DJ = dDJinicial; DJ<=dDJfinal; DJ+=dIntervalDies)
{
    double T = tT(DJ);
    double dDeclinacio = declinacioSolar(dataJulia(DJ));
    double d0 = o0(T);
    double dObliquitat = obliquitat(any);
    double dAscensioRecta = ascensioRecta(dObliquitat, d0);
    double errorGMST = trobarErrorGMST(T, dObliquitat);
    double dGMST = trobarGMST(DJ, T, errorGMST);
    double dAngleHorari = angleHorari(dGMST, dLongitud,
dAscensioRecta);
    double dAngleHorariRad = dAngleHorari*(M_PI/180);
    double h =
(asin(cos(dLatitud)*cos(dDeclinacio)*cos(dAngleHorariRad) +
sin(dLatitud)*sin(dDeclinacio)))*(180/M_PI);
    double a =
((atan2(sin(dAngleHorariRad), (cos(dAngleHorariRad)*sin(dLatitud) -
tan(dDeclinacio)*cos(dLatitud))))*(180/M_PI))+180; // Sumem 180 perquè
no retorni els angles entre -180 i 180 sinó entre 0 i 360.
    altures << a << h;
    if (bTrobarPosicio)
        break;
    double hRelleu = (calcularRelleu?trobarAlturaRelleu(a):0);
    if (h<0)
        bSolAmagat=true;
    if (h>0)
        bElSolHaSortit=true;

    // Sortida.
    if (bSolAmagat && h>hRelleu && !bPrimerSortida)

```



```

    {
        azimuthSortida = a;
        horaSortida = DJ;
        bPrimerSortida = true;
    }

    // Posta.
    if (bElSolHaSortit && !bUltimSolAmagat && h<=hRelleu)
    {
        azimuthPosta = a;
        horaPosta = DJ;
    }

    // Trànsit (es fa el càlcul però no es mostra per pantalla).
    if (h>hMaxim)
    {
        alturaTransit=h;
        horaTransit=DJ;
        hMaxim = h;
    }
    bUltimSolAmagat = h<=hRelleu;
}

// Si no hi ha sortida:
if (!bPrimerSortida)
{
    bErrorSortida = true;
}

    altures << azimuthSortida << horaSortida-0.5 << alturaTransit <<
horaTransit-0.5 << azimuthPosta << horaPosta-0.5; // -0.5 perquè s'ha
de tenir en compte que són dies julians i aquests comencen a 0.5.
    return (altures);
}

void RCalculadora::guardarArxiu(QString any, QString mes, QString dia,
QString localitzacio, QString latGraus, QString latMinuts, QString
latSegons, QString lonGraus, QString lonMinuts, QString lonSegons,
QString ns, QString eo, QString hora, QString minut, QString segon,
int indexUTC)
{
    // Guardar dades seleccionades.
    QString sDirAppData=
QProcessEnvironment::systemEnvironment().value("APPDATA");
    QString sCarpeta=sDirAppData+"/BTS";
    if (!QDir(sCarpeta).exists())
        QDir().mkpath(sCarpeta);
    QString sArxiuIni=sCarpeta+"/config.ini";
    QSettings settings(sArxiuIni, QSettings::IniFormat);
    settings.setValue("POSICIO/LOCALITZACIO", localitzacio);
    settings.setValue("POSICIO/LATGRAUS", latGraus);
    settings.setValue("POSICIO/LATMINUTS", latMinuts);
    settings.setValue("POSICIO/LATSEGONS", latSegons);
    settings.setValue("POSICIO/LONGRAUS", lonGraus);
    settings.setValue("POSICIO/LONMINUTS", lonMinuts);
}

```

```

settings.setValue("POSICIO/LONSEGONS", lonSegons);
settings.setValue("POSICIO/LATGRAUS", latGraus);
settings.setValue("POSICIO/NS", ns);
settings.setValue("POSICIO/E0", eo);
settings.setValue("TEMPS/ANY", any);
settings.setValue("TEMPS/MES", mes);
settings.setValue("TEMPS/DIA", dia);
settings.setValue("TEMPS/HORA", hora);
settings.setValue("TEMPS/MINUT", minut);
settings.setValue("TEMPS/SEGON", segon);
settings.setValue("TEMPS/INDEXUTC", indexUTC);
}

QVariantList RCalculadora::llegirArxiu()
{
    // Llegir paràmetres de l'arxiu.
    QString sDirAppData=
QProcessEnvironment::systemEnvironment().value("APPDATA");
    QString sCarpeta=sDirAppData+"/BTS";
    if (!QDir(sCarpeta).exists())
        QDir().mkpath(sCarpeta);
    QString sArxiuIni=sCarpeta+"/config.ini";
    QSettings settings(sArxiuIni, QSettings::IniFormat);
    QString localitzacio =
settings.value("POSICIO/LOCALITZACIO").toString();
    QString latGraus = settings.value("POSICIO/LATGRAUS").toString();
    QString latMinuts =
settings.value("POSICIO/LATMINUTS").toString();
    QString latSegons =
settings.value("POSICIO/LATSEGONS").toString();
    QString lonGraus = settings.value("POSICIO/LONGRAUS").toString();
    QString lonMinuts =
settings.value("POSICIO/LONMINUTS").toString();
    QString lonSegons =
settings.value("POSICIO/LONSEGONS").toString();
    QString ns = settings.value("POSICIO/NS").toString();
    QString eo = settings.value("POSICIO/E0").toString();
    QString any = settings.value("TEMPS/ANY").toString();
    QString mes = settings.value("TEMPS/MES").toString();
    QString dia = settings.value("TEMPS/DIA").toString();
    QString hora = settings.value("TEMPS/HORA").toString();
    QString minut = settings.value("TEMPS/MINUT").toString();
    QString segon = settings.value("TEMPS/SEGON").toString();
    int indexUTC = settings.value("TEMPS/INDEXUTC").toInt();

    QVariantList variantListValors;
    variantListValors << localitzacio << latGraus << latMinuts <<
latSegons << lonGraus << lonMinuts << lonSegons << ns << eo << any <<
mes << dia << hora << minut << segon << indexUTC;
    return (variantListValors);
}

QVariantList RCalculadora::llegirArxiuRelleu(QString arxiuRelleu)
{
    QVariantList relleu;

```

```

QFile inputFile(QUrl(arxiuRelleu).toLocalFile());
if (inputFile.open(QIODevice::ReadOnly))
{
    QTextStream in(&inputFile);
    while (!in.atEnd())
    {
        QString sLinia = in.readLine();
        QStringList azimuthAltura = sLinia.split(";");
        relleu << azimuthAltura[0] << azimuthAltura[1];
    }
}
inputFile.close();
dadesRelleu = relleu;
return(dadesRelleu);
}

double RCalculadora::trobarAlturaRelleu(double azimuth)
{
    QVariantList relleu = dadesRelleu;
    relleu << relleu[0].toDouble() << relleu[1].toDouble();
    int numPunts = relleu.size()/2.-1;
    double h = -1;

    for (int punt = 0; punt<numPunts; punt++)
    {
        if (azimuth >= relleu[punt*2].toDouble() && azimuth <=
relleu[2*punt+2].toDouble())
        {
            double a1 = relleu[punt*2].toDouble();
            double h1 = relleu[(punt*2)+1].toDouble();
            double a2 = relleu[2*punt+2].toDouble();
            double h2 = relleu[2*punt+3].toDouble();
            h = h1 + ((h2-h1)/(a2-a1)) * (azimuth-a1);
            break;
        }
    }

    return(h);
}
}

```

```

import QtQuick 2.0
import QtQuick.Controls 2.5
import QtQuick.Window 2.2
import QtQuick.Dialogs 1.2

Item {
    id: container

    property var calculs;
    property var calculsSol;
    property var azimuthPosicio;
    property var alturaPosicio;
    property var relleu;
    property bool permetreDibuixar: false;
    property string arxiuRelleu: "";

    function omplirValorsDefecte () {
        var valors = calculadora.llegirArxiu();
        textFieldNomLocalitzacio.text=valors[0];
        textFieldGrausLatitud.text=valors[1];
        textFieldMinutsLatitud.text=valors[2];
        textFieldSegonsLatitud.text=valors[3];
        textFieldGrausLongitud.text=valors[4];
        textFieldMinutsLongitud.text=valors[5];
        textFieldSegonsLongitud.text=valors[6];
        comboBoxLatitud.currentIndex=(valors[7]=="S"?1:0);
        comboBoxLongitud.currentIndex=(valors[8]=="0"?1:0);
        textFieldAny.text=valors[9];
        comboMesos.currentIndex=calculadora.numMes(valors[10]);
        comboDies.currentIndex=valors[11];
        textFieldHora.text=valors[12];
        textFieldMinuts.text=valors[13];
        textFieldSegons.text=valors[14];
        comboBoxUTC.currentIndex=valors[15];
    }

    Component.onCompleted: omplirValorsDefecte()

    Image {
        id: imatgeSol
        source: "Imatges/SOL.png"
        visible: false
    }

   .GroupBox {
        id: escenaIControls
        anchors.fill: parent
        anchors.leftMargin: Screen.width/85
        anchors.bottomMargin: 25
        anchors.rightMargin: Screen.height/50
        background: Rectangle {color: "transparent"; border.color:
"#595959"}
    }

```

```

Text {
    id: textTitol
    font.pointSize: 12
    font.bold: true
    text: "Escena simulació"
    anchors.horizontalCenter: canvas.horizontalCenter
    color: "#ffffff"
    height: 30;
}

Canvas {
    id: canvas
    anchors.left: parent.left
    anchors.right: parent.right
    anchors.top: parent.top
    anchors.bottom: parent.bottom
    anchors.topMargin: 36
    anchors.bottomMargin:
gridDades.implicitHeight+botonsControl.height+30
    anchors.rightMargin: 25
    anchors.leftMargin: 75
    antialiasing: true

    onPaint: {

        var ctx = canvas.getContext('2d');

        // Netejar canvas:
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        ctx.globalAlpha = canvas.alpha;
        ctx.globalCompositeOperation = "source-over";

        // Omplir de color gris:
        ctx.fillStyle=Qt.rgb(0.5, 0.5, 0.5, 1);
        ctx.fillRect(0, 0, canvas.width, canvas.height)

        // Línies escala:
        ctx.beginPath();
        ctx.strokeStyle=Qt.rgb(0,0,0,1);
        ctx.lineWidth=6;
        ctx.moveTo(0, 0);
        ctx.lineTo(0, canvas.height);
        ctx.lineTo(canvas.width, canvas.height);
        ctx.stroke();
        ctx.closePath();

        if (!permetreDibuixar)
            return;

        // Vèrtex:
        var numVertex = (container.calculs.length-6)/2;

        ctx.beginPath();
        ctx.strokeStyle=ctx.createPattern("yellow",
Qt.SolidPattern);

```

```

    ctx.lineWidth=2;
    var ultimAzimut = -1;
    var azimut;
    var altura;
    var azimutPX;
    var alturaPX;

    for (var vertex = 1; vertex<=numVertex; vertex++)
    {
        azimut = container.calculs[(vertex-1)*2];
        altura = container.calculs[2*vertex-1];

        azimutPX = azimut*canvas.width/360;
        alturaPX = -((altura-90)/180*canvas.height);

        if (vertex===1)
            ctx.moveTo(azimutPX,alturaPX)
        else
            ctx.lineTo(azimutPX,alturaPX)
    }

    ctx.stroke();
    ctx.closePath();

    numVertex = (container.calculsSol.length-6)/2;
    if (numVertex===1)
    {
        azimut = container.calculsSol[0];
        altura = container.calculsSol[1];

        container.azimutPosicio =
calculadora.passarASeparats(true,azimut);
        container.alturaPosicio =
calculadora.passarASeparats(true,altura);

        azimutPX = azimut*canvas.width/360;
        alturaPX = -((altura-90)/180*canvas.height);

        var escalatImatge = 2;
        var ampladaImatge =
imatgeSol.sourceSize.width/escalatImatge;
        var alturaImatge =
imatgeSol.sourceSize.height/escalatImatge;

        ctx.drawImage(imatgeSol,azimutPX-
ampladaImatge/2,alturaPX-alturaImatge/2, ampladaImatge, alturaImatge);
    }

    // Horitzó amb relleu:
    if (!checkBoxSenseRelleu.checked)
    {
        var numPunts =
Math.trunc((container.relleu.length/2));
        ctx.beginPath();
        ctx.strokeStyle=Qt.rgba(0,0,0,1);

```

```

ctx.fillStyle=Qt.rgb(0,0,0,0.5);
ctx.lineWidth=3;
for (var punt = 1; punt<=numPunts; punt++)
{
    azimuth = container.relleu[(punt-1)*2];
    altura = container.relleu[2*punt-1];

    azimuthPX = azimuth*canvas.width/360;
    alturaPX = -((altura-90)/180*canvas.height);

    if (vertex===1)
        ctx.moveTo(azimuthPX,alturaPX)
    else
        ctx.lineTo(azimuthPX,alturaPX)
}
ctx.lineTo(canvas.width,-((container.relleu[1]-
90)/180*canvas.height));
ctx.lineTo(canvas.width,canvas.height/2);
ctx.lineTo(0,canvas.height/2);
ctx.closePath();
ctx.fill();
ctx.stroke();
}

// Horitzó pla:
var yRefHoritzo = canvas.height/2;
ctx.beginPath();
ctx.strokeStyle=Qt.rgb(1, 1, 1, 1);
ctx.lineWidth=4;
ctx.moveTo(13, yRefHoritzo);
ctx.lineTo(canvas.width, yRefHoritzo);
ctx.stroke();
ctx.closePath();

//Terra:
ctx.beginPath();
ctx.fillStyle=Qt.rgb(1,1,1,0.5);
ctx.fillRect(3, yRefHoritzo, canvas.width,
yRefHoritzo-3);
ctx.fill();
ctx.closePath();

// Línies números altura:
ctx.beginPath();
ctx.strokeStyle=Qt.rgb(0,0,0,1);
ctx.lineWidth=3;
var numLinies1 = 0

for (var n1 = 0; numLinies1<=9; n1++, numLinies1++)
{
    ctx.moveTo(0, canvas.height/8*n1)
    ctx.lineTo(13, canvas.height/8*n1)
}
ctx.stroke();
ctx.closePath();

```

```

        // Línies números azimuth:
        ctx.beginPath();
        ctx.strokeStyle=Qt.rgba(0,0,0,1);
        ctx.lineWidth=3;
        var numLinies2 = 0

        for (var n2 = 0; numLinies2<=9; n2++, numLinies2++)
        {
            ctx.moveTo(canvas.width/8*n2, canvas.height)
            ctx.lineTo(canvas.width/8*n2, canvas.height-13)
        }
        ctx.stroke();
        ctx.closePath();
    }
}

// Número de graus altures:
Label {
    id: labelALTURES
    width: 50
    color: "#98ff98"
    text: qsTr("ALTURA")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 15
    anchors.leftMargin: 20
}

Label {
    id: label90
    width: 40
    color: "#98ff98"
    text: qsTr("90°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: 30
    anchors.leftMargin: 25
}

Label {
    id: label675
    width: 40
    color: "#98ff98"
    text: qsTr("67.5°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+canvas.height/8)
    anchors.leftMargin: 25
}

```



```

}

Label {
    id: label45
    width: 40
    color: "#98ff98"
    text: qsTr("45°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*2)
    anchors.leftMargin: 25
}

Label {
    id: label225
    width: 40
    color: "#98ff98"
    text: qsTr("22.5°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*3)
    anchors.leftMargin: 25
}

Label {
    id: label0
    width: 40
    color: "#98ff98"
    text: qsTr("0°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*4)
    anchors.leftMargin: 25
}

Label {
    id: label_225
    width: 40
    color: "#98ff98"
    text: qsTr("-22.5°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*5)
    anchors.leftMargin: 25
}

Label {

```

```

    id: label_45
    width: 40
    color: "#98ff98"
    text: qsTr("-45°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*6)
    anchors.leftMargin: 25
}

```

```

Label {
    id: label_675
    width: 40
    color: "#98ff98"
    text: qsTr("-67.5°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*7)
    anchors.leftMargin: 25
}

```

```

Label {
    id: label_90
    width: 40
    color: "#98ff98"
    text: qsTr("-90°")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignRight
    font.pointSize: 12
    anchors.topMargin: (24+(canvas.height/8)*8)
    anchors.leftMargin: 25
}

```

```

//Número de graus azimuts:
Label {
    id: labelAZIMUTS
    width: 50
    color: "#c38be6"
    text: qsTr("AZIMUT")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 15
    anchors.topMargin: (70+canvas.height)
    anchors.leftMargin: 20
}

```

```

Label {
    id: labela0
    width: 60
}

```

```

        color: "#c38be6"
        text: qsTr("N (0°)")
        anchors.left: parent.left
        anchors.top: parent.top
        horizontalAlignment: Text.AlignLeft
        font.pointSize: 12
        anchors.topMargin: (45+canvas.height)
        anchors.leftMargin: 60
    }

    Label {
        id: labela45
        width: 40
        color: "#c38be6"
        text: qsTr("NE (45°)")
        anchors.left: parent.left
        anchors.top: parent.top
        horizontalAlignment: Text.AlignHCenter
        font.pointSize: 12
        anchors.topMargin: (45+canvas.height)
        anchors.leftMargin: (60+(canvas.width/8))
    }

    Label {
        id: labela90
        width: 40
        color: "#c38be6"
        text: qsTr("E (90°)")
        anchors.left: parent.left
        anchors.top: parent.top
        horizontalAlignment: Text.AlignHCenter
        font.pointSize: 12
        anchors.topMargin: (45+canvas.height)
        anchors.leftMargin: (60+(canvas.width/8)*2)
    }

    Label {
        id: labela135
        width: 40
        color: "#c38be6"
        text: qsTr("SE (135°)")
        anchors.left: parent.left
        anchors.top: parent.top
        horizontalAlignment: Text.AlignHCenter
        font.pointSize: 12
        anchors.topMargin: (45+canvas.height)
        anchors.leftMargin: (60+(canvas.width/8)*3)
    }

    Label {
        id: labela180
        width: 40
        color: "#c38be6"
        text: qsTr("S (180°)")
        anchors.left: parent.left

```

```
anchors.top: parent.top
horizontalAlignment: Text.AlignHCenter
font.pointSize: 12
anchors.topMargin: (45+canvas.height)
anchors.leftMargin: (60+(canvas.width/8)*4)
}
```

```
Label {
    id: labela225
    width: 40
    color: "#c38be6"
    text: qsTr("S0 (225°)")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 12
    anchors.topMargin: (45+canvas.height)
    anchors.leftMargin: (60+(canvas.width/8)*5)
}
```

```
Label {
    id: labela270
    width: 40
    color: "#c38be6"
    text: qsTr("O (270°)")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 12
    anchors.topMargin: (45+canvas.height)
    anchors.leftMargin: (60+(canvas.width/8)*6)
}
```

```
Label {
    id: labela315
    width: 40
    color: "#c38be6"
    text: qsTr("E (315°)")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 12
    anchors.topMargin: (45+canvas.height)
    anchors.leftMargin: (60+(canvas.width/8)*7)
}
```

```
Label {
    id: labela360
    width: 40
    color: "#c38be6"
    text: qsTr("N (360°)")
    anchors.left: parent.left
    anchors.top: parent.top
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 12
}
```

```

        anchors.topMargin: (45+canvas.height)
        anchors.leftMargin: (60+(canvas.width/8)*8)
    }

    Column {
        id: controlIDades
        anchors.bottom: parent.bottom
        anchors.bottomMargin: -20
        anchors.horizontalCenter: parent.horizontalCenter
        spacing: 10

        Grid {
            id: gridDades
            rowSpacing: 10
            columnSpacing: 30
            rows: 2
            columns: 4
            visible: false
            anchors.horizontalCenter: parent.horizontalCenter

            Label {
                id: dia
                font.pointSize: 10
                color: "white"
                text: "Dia: " +
                    ((comboDies.currentText=="D"?"" : comboDies.currentText) + "/" +
                    (comboMesos.currentText=="M"?"" : calculadora.numMes(comboMesos.current
                    Text)) + "/" + textFieldAny.text
            }

            Label {
                id: hora
                font.pointSize: 10
                color: "white"
                text: "Hora: " +
                    ((textFieldHora.text=="&&textFieldMinuts.text=="&&textFieldSegons.
                    text=="")?"tot el
                    dia":((textFieldHora.text=="||textFieldHora.text=="0"? "00":textFiel
                    dHora.text)+"h
                    "+(textFieldMinuts.text=="||textFieldMinuts.text=="0"? "00":textFiel
                    dMinuts.text)+"min
                    "+(textFieldSegons.text=="||textFieldSegons.text=="0"? "00":textFiel
                    dSegons.text))+ "s ")
            }

            Label {
                id: horaSortida
                font.pointSize: 10
                color: "White"
                text: ""
            }

            Label {
                id: horaPosta
                font.pointSize: 10

```

```

        color: "white"
        text: "Posta: "
    }

    Label {
        id: lloc
        font.pointSize: 10
        color: "white"
        text: "Lloc: " + textFieldNomLocalitzacio.text
    }

    Label {
        id: posicio
        font.pointSize: 10
        color: "white"
        text: "a/h: " +
((textFieldHora.text=="&&textFieldMinuts.text=="&&textFieldSegons.
text=="")?"---
":(container.azimutPosicio+"/"+container.alturaPosicio))
    }

    Label {
        id: posicioSortida
        font.pointSize: 10
        color: "white"
        text: ""
    }

    Label {
        id: posicioPosta
        font.pointSize: 10
        color: "white"
        text: ""
    }
}

Row {
    id: botonsControl
    anchors.horizontalCenter: parent.horizontalCenter
    spacing: 10

    Button {
        id: botonIniciarSimulacio
        text: "Entrar dades"
        implicitHeight: 25
        font.pointSize: 12
        onClicked: rectangleIniciarSimulacio.visible=true
    }

    Button {
        id: botonCalculadora
        font.pointSize: 12
        implicitHeight: 25
        text: "Càlculs bàsics."
        visible: false
    }
}

```

```

        Connections {
            target: buttonCalculadora
            onClicked: dialogCalculadora.open()
        }
    }
}

Rectangle {
    id: rectangleIniciarSimulacio
    height: gridDadesIniciarSimulacio.height+20
    width: gridDadesIniciarSimulacio.width+20
    visible: false
    color: Qt.rgba(1,1,1,0.5)
    anchors.centerIn: parent

    Grid {
        id: gridDadesIniciarSimulacio
        anchors.centerIn: parent
        columnSpacing: 30
        rowSpacing: 10
        rows: 6
        columns: 2

        TextField {
            id: textFieldAny
            width: 135
            height: 28
            placeholderText: qsTr("Any:")
        }

        TextField {
            id: textFieldNomLocalitzacio
            width: 203
            height: 28
            placeholderText: qsTr("Nom de la localització:")
        }

        ComboBox {
            id: comboMesos
            displayText: "Mes: " + currentText
            width: 135
            height: 28
            model:
["M", "gener", "febrer", "març", "abril", "maig", "juny", "juliol", "agost", "s
etembre", "octubre", "novembre", "desembre"]
            enabled: (textFieldAny.text==="")?false:true
        }

        Row {
            id: latitud

            TextField {
                id: textFieldGrausLatitud

```

```

width: 45
height: 28
placeholderText: qsTr("°")

TextField {
    id: textFieldMinutsLatitud
    width: 45
    height: 28
    anchors.left: parent.left
    anchors.leftMargin: 45
    placeholderText: qsTr("''")
}

TextField {
    id: textFieldSegonsLatitud
    width: 45
    height: 28
    anchors.left: parent.left
    anchors.leftMargin: 90
    placeholderText: qsTr("''")
}

ComboBox {
    id: comboBoxLatitud
    width: 63
    height: 28
    anchors.left: parent.left
    anchors.leftMargin: 140
    model: ["N", "S"]
}
}

ComboBox {
    id: comboDies
    displayText: "Dia: " + currentText
    width: 135
    height: 28
    model: if
(comboMesos.currentText=== "abril" || comboMesos.currentText=== "juny" | | co
mboMesos.currentText=== "setembre" | | comboMesos.currentText=== "novembre"
)

["D", "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"]

    else if (comboMesos.currentText=== "febrer")
    {
        if
(calculadora.anyTraspas(textFieldAny.text)===1)

["D", "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"
]

        else

```



```

["D","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"]
    }
    else

["D","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","31"]
        enabled:
(comboMesos.currentText=="M"||comboMesos.enabled!=true)?false:true
    }

    Row {
        id: longitud

        TextField {
            id: textFieldGrausLongitud
            width: 45
            height: 28
            placeholderText: qsTr("°")

            TextField {
                id: textFieldMinutsLongitud
                width: 45
                height: 28
                anchors.left: parent.left
                anchors.leftMargin: 45
                placeholderText: qsTr("'")
            }

            TextField {
                id: textFieldSegonsLongitud
                width: 45
                height: 28
                anchors.left: parent.left
                anchors.leftMargin: 90
                placeholderText: qsTr("'")
            }
        }

        ComboBox {
            id: comboBoxLongitud
            width: 63
            height: 28
            anchors.left: parent.left
            anchors.leftMargin: 140
            model: ["E","O"]
        }
    }
}

    Row {
        id: entrarHora

        TextField {

```

```

        id: textFieldHora
        width: 45
        height: 28
        placeholderText: "h"

        TextField {
            id: textFieldMinuts
            width: 45
            height: 28
            anchors.left: parent.left
            anchors.leftMargin: 45
            placeholderText: "min"
        }

        TextField {
            id: textFieldSegons
            width: 45
            height: 28
            anchors.left: parent.left
            anchors.leftMargin: 90
            placeholderText: "s"
        }
    }

    ComboBox {
        id: comboBoxUTC
        displayText: "UTC " + currentText
        width: 135
        height: 28
        model: ["±00:00", "-12:00", "-11:00", "-10:00", "-09:00", "-08:00", "-07:00", "-06:00", "-05:00", "-04:00", "-03:00", "-02:00", "-01:00", "+01:00", "+02:00", "+03:00", "+04:00", "+05:00", "+06:00", "+07:00", "+08:00", "+09:00", "+10:00", "+11:00", "+12:00", "+13:00"]
    }

    CheckBox {
        id: checkBoxSenseRelleu
        text: "Sense relleu"
    }

    Button {
        id: buttonIncerirArxiuRelleu
        text: "Busca l'arxiu del relleu (.txt)"
        enabled: (checkBoxSenseRelleu.checked?false:true)
        onClicked: {
            fileDialogIncerirArxiu.open()
        }
    }

    Button {
        id: buttonIniciSimulacio

```

```

        text: "Iniciar Simulacio"
        enabled:
            (textFieldAny.text==="" || comboMesos.currentText=== "M" || comboDies.curre
            ntText=== "D" || (!checkBoxSenseRelleu.checked &&
            arxiuRelleu=== "")) ? false : true
        onClicked: {
            permetreDibuixar=true;

            var grausLatitud =
            (textFieldGrausLatitud.text=== "") ? "0.0" : textFieldGrausLatitud.text
            var minutsLatitud =
            (textFieldMinutsLatitud.text=== "") ? "0.0" :
            textFieldMinutsLatitud.text
            var segonsLatitud =
            (textFieldSegonsLatitud.text=== "") ? "0.0" :
            textFieldSegonsLatitud.text.replace(",",".")
            var grausLongitud =
            (textFieldGrausLongitud.text=== "") ? "0.0" :
            textFieldGrausLongitud.text
            var minutsLongitud =
            (textFieldMinutsLongitud.text=== "") ? "0.0" :
            textFieldMinutsLongitud.text
            var segonsLongitud =
            (textFieldSegonsLongitud.text=== "") ? "0.0" :
            textFieldSegonsLongitud.text.replace(",",".")

            var latitud =
            calculadora.passarADecimals(grausLatitud,minutsLatitud,segonsLatitud)
            var longitud =
            (comboBoxLongitud.currentText=== "E") ? -
            (calculadora.passarADecimals(grausLongitud,minutsLongitud,segonsLongit
            ud)) :
            calculadora.passarADecimals(grausLongitud,minutsLongitud,segonsLongitu
            d)

            // Es calcula el dia julià de l'hora en UTC per poder fer els càlculs:
            var horaUTC =
            parseInt((comboBoxUTC.currentText[0]=== "+" ||
            comboBoxUTC.currentText[0]=== "±" ? "+" : "-") +
            comboBoxUTC.currentText[1] + comboBoxUTC.currentText[2])
            var DJ =
            calculadora.diaJulia(comboDies.currentText, comboMesos.currentText,
            textFieldAny.text)

            var hora = parseInt(textFieldHora.text)-
            horaUTC

            if (hora<0)
            {
                hora += 24
                DJ -= 1
            }
            if (hora>24)
            {
                hora -= 24
                DJ += 1
            }
        }
    }
}

```

```

        var horaMinutSegon = hora.toString() + " " +
textFieldMinuts.text + " " + textFieldSegons.text
        var dataUTC =
            calculadora.dataJulia(DJ).split(" ")
        var any = dataUTC[2]

        relleu =
calculadora.llegirArxiuRelleu(arxiuRelleu);
        calculs = calculadora.alturaAzimut(DJ,
horaMinutSegon, any, latitud, longitud,
false, !checkBoxSenseRelleu.checked);
        var bErrorSortida = calculadora.errorSortida;
        calculsSol = calculadora.alturaAzimut(DJ,
horaMinutSegon, any, latitud, longitud,
true, !checkBoxSenseRelleu.checked);

        // Cas em què no hi hagi ni sortida ni posta:
        if (bErrorSortida)
        {
            horaSortida.text = "Sortida: no hi ha
sortida"

            posicioSortida.text = "Azimut: ---"
            horaPosta.text = "Posta: no hi ha posta"
            posicioPosta.text = "Azimut: ---"
        }
        else
        {
            horaSortida.text = "Sortida: " +
(calculadora.passarASeparats(false, (container.calculs[container.calcul
s.length-5]-Math.trunc(container.calculs[container.calculs.length-
5]))*24, horaUTC));
            posicioSortida.text = "Azimut: " +
(calculadora.passarASeparats(true, container.calculs[container.calculs.
length-6]));
            horaPosta.text = "Posta: " +
(calculadora.passarASeparats(false, (container.calculs[container.calcul
s.length-1]-Math.trunc(container.calculs[container.calculs.length-
1]))*24, horaUTC));
            posicioPosta.text = "Azimut: " +
(calculadora.passarASeparats(true, container.calculs[container.calculs.
length-2]));
        }

        canvas.requestPaint();//
        text: "Iniciar Simulació"
        rectangleIniciarSimulacio.visible=false
        gridDades.visible=true
        calculadora.guardarArxiu(textFieldAny.text,
comboMesos.currentText, comboDies.currentText,
textFieldNomLocalitzacio.text, textFieldGrausLatitud.text,
textFieldMinutsLatitud.text, textFieldSegonsLatitud.text,
textFieldGrausLongitud.text, textFieldMinutsLongitud.text,
textFieldSegonsLongitud.text, comboBoxLatitud.currentText,
comboBoxLongitud.currentText, textFieldHora.text,
textFieldMinuts.text, textFieldSegons.text, comboBoxUTC.currentIndex)

```

```

    }
  }
  Button {
    id: buttonCancelIniciSimulacio
    text: "Cancelar"
    onClicked: {
      rectangleIniciarSimulacio.visible=false
    }
  }
}
}
}

FileDialog {
  id: fileDialogIncerirArxiu
  title: "Incereix l'arxiu amb les dades del relleu."
  onAccepted: {
    arxiuRelleu = fileUrl;
    buttonIncerirArxiuRelleu.text = "Arxiu seleccionat."
  }
}
}
}

```

ANNEX II – CONTINGUT DELS ARXIUS AMB LES DADES DEL RELLEU

RelleuReal.txt		RelleuFals.txt	
azimut(°);altura(°)			
0;12.7	99;15.0	228;9.35	0;45.3
2;12.6	102;14.1	229;9.25	10;46.7
6;12.1	106;13.65	235;9.35	20;30.9
7;12.4	111;12.8	263;9.0	30;26.5
11;12.3	114;12.85	270;8.8	40;20.7
13;12.5	117;13.6	273;8.75	50;21.6
16;12.05	121;13.5	274;9.15	60;24.6
18;12.05	128;13.7	290;9.2	70;24.2
22;12.7	135;11.7	293;10.0	80;25.5
25;11.9	139;10.55	315;9.95	90;25.4
27;11.85	140;10.4	321;10.0	100;27.5
29;12.6	144;10.45	324;10.15	110;26.3
31;12.35	145;10.55	331;10.5	120;18.5
34;12.75	147;10.5	334;10.4	130;22.9
36;13.1	150;10.35	338;10.2	140;22.1
37;13.5	152;10.5	340;10.5	150;16.7
41;14.0	154;10.4	342;10.6	160;15.9
43;13.8	162;10.3	346;11.75	180;15.7
45;13.4	169;10.25	351;10.85	190;15.7
49;12.65	171;10.5	352;11.1	200;6.9
51;12.35	174;10.45	355;11.4	210;7.5
54;13.45	178;10.15	357;11.3	220;12.1
58;13.9	180;10.05		230;11.5
63;13.95	182;10.2		240;8.1
67;12.8	183;10.1		250;8.8
69;13.2	185;10.1		260;10.4
73;12.45	188;10.05		270;6.4
77;12.8	192;9.9		280;9.8
79;13.6	194;9.8		290;36.7
85;14.4	200;9.75		300;34.7
87;13.7	209;9.55		310;33.8
90;13.25	218;9.4		320;30.9
93;13.4	220;9.35		330;34.5
95;14.0	222;9.35		340;36.7
97;15.1	225;9.3		350;50.2

ANNEX III – TAULES PER AL CÀLCUL DE SOLSTICIS I EQUINOCCIS

Taula A	Per als anys -1000 fins +1000	$Y = \frac{any}{1000}$
A.1. Equinocci de març (principi de la primavera astronòmica): $JDE_0 = 1721\ 139.29189 + 365\ 242.13740 Y + 0.06134 Y^2 + 0.00111 Y^3 - 0.00071 Y^4$		
A.2. Solstici de juny (principi de l'estiu astronòmic): $JDE_0 = 1721\ 233.25401 + 365\ 241.72562 Y - 0.05323 Y^2 + 0.00907 Y^3 + 0.00025 Y^4$		
A.3. Equinocci de setembre (principi de la tardor astronòmica): $JDE_0 = 1721\ 325.70455 + 365\ 242.49558 Y - 0.11677 Y^2 - 0.00297 Y^3 + 0.00074 Y^4$		
A.4. Solstici de desembre (principi de l'hivern astronòmic): $JDE_0 = 1721\ 414.39987 + 365\ 242.88257 Y - 0.00769 Y^2 - 0.00933 Y^3 - 0.00006 Y^4$		

Taula B	Per als anys +1000 fins +3000	$Y = \frac{any-2000}{1000}$
B.1. Equinocci de març (principi de la primavera astronòmica): $JDE_0 = 2451\ 623.80984 + 365\ 242.37404 Y + 0.05169 Y^2 - 0.00411 Y^3 - 0.00057 Y^4$		
B.2. Solstici de juny (principi de l'estiu astronòmic): $JDE_0 = 2451\ 716.56767 + 365\ 241.62603 Y + 0.00325 Y^2 + 0.00888 Y^3 - 0.00030 Y^4$		
B.3. Equinocci de setembre (principi de la tardor astronòmica): $JDE_0 = 2451\ 810.21715 + 365\ 242.01767 Y - 0.11575 Y^2 + 0.00337 Y^3 + 0.00078 Y^4$		
B.4. Solstici de desembre (principi de l'hivern astronòmic): $JDE_0 = 2451\ 900.05952 + 365\ 242.74049 Y - 0.06223 Y^2 - 0.00823 Y^3 + 0.00032 Y^4$		

Taula C					
$S = \sum A \cdot \cos(B + CT)$			* B i C en graus		
A	B	C	A	B	C
485	324.96	1934.136	45	247.54	29929.562
203	337.23	32964.467	44	325.15	31555.956
199	342.08	20.186	29	60.93	4443.417
182	27.85	445267.112	18	155.12	67555.328
156	73.14	45036.886	17	288.79	4562.452
136	171.52	22518.443	16	198.04	62894.029
77	222.54	65928.934	14	199.76	31436.921
74	296.72		12	95.39	14577.848
70	243.58		12	287.11	31931.756
58	119.81		12	320.81	34777.259
52	297.17		9	227.73	1222.114
50	21.02		8	15.45	16859.074