



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Facultat d'Informàtica de Barcelona
Grau en Enginyeria Informàtica
Especialitat de Computació

Integració de models de patrimoni cultural en un Sistema d'Informació Geogràfica

Treball de Fi de Grau
Document final

Xavier Valls Ruiz

Director: Oscar Argudo
Codirector: Carlos Andújar Gran
Tutor GEP: Ferran Marfil Sánchez

10 d'octubre de 2022

Índex

1. Introducció i Contextualització	5
1.1. Context.....	5
1.2. Definició de conceptes.....	6
1.2.1. Sistema d'Informació Geogràfica (SIG)	6
1.2.2. QGIS.....	6
1.2.3. Model 3D.....	7
1.2.4. Plugin.....	7
1.2.5. Format JPG/JPEG	7
1.3. Descripció del problema	8
1.4. Actors implicats	9
1.6. Justificació de la tria	9
2. Abast del Projecte	11
2.1. Objectius	11
2.2. Funcionalitats	12
2.3 Requeriments	12
2.3.1. Requeriments funcionals.....	12
2.3.2. Requeriments no funcionals.....	12
2.4. Obstacles i Riscos	13
2.5. Metodologia.....	13
2.5.1. Mètode AGILE	13
2.5.2. Eines.....	14
2.5.3. Mètode de validació	14
3. Planificació temporal	14
3.1. Recursos necessaris	14
3.1.1. Recursos humans	14
3.1.2. Recursos materials	15
3.2. Descripció de tasques	15
3.2.1. GP - Gestió del projecte.....	15
3.2.2. TP – Treball previ.....	16
3.2.3. DP – Desenvolupament del plugin	17
3.2.4. VU – Validació dels usuaris.....	18
3.3. Diagrama de Gantt	19
3.4. Gestió de riscos.....	20
3.5. Diagrama de Gantt actualitzat	20

4. Gestió econòmica	22
4.1. Costos de personal.....	22
4.2. Costos genèrics.....	23
4.2.1. Amortitzacions	23
4.2.2. Espai de treball	24
4.2.3. Consum elèctric i Internet.....	24
4.3. Imprevistos	24
4.4. Contingències.....	25
4.5. Cost total	25
4.6. Control de gestió	25
5. Sostenibilitat	26
5.1. Dimensió ambiental	26
5.2. Dimensió econòmica	26
5.3. Dimensió social	27
6. Bases Teòriques	27
6.1 Conceptes	27
6.1.1 Sistema de Referència de Coordenades.....	27
6.1.2. Dades SIG / Capes	29
6.2. QGIS	33
6.2.1 Característiques principals	33
6.2.2. Interfície	35
7. Implementació tècnica	39
7.1. Disseny.....	39
7.1.1 Plugin Builder	39
7.1.2. Disseny	42
7.1.3. Interfície	43
7.2. Importar fotografies	44
7.2.1. Fotografies georeferenciades.....	47
7.2.2. Fotografies no georeferenciades.....	47
7.2.3. Dificultats	48
7.3. Importar Models 3D	48
7.4. Exportar a la web.....	50
7.4.1. Fitxer “script”	54
7.5. Visualització a la web	56
8. Conclusions	58
8.1. Respecte la resolució del problema	58

8.2. Respecte la planificació	59
8.3. Feina futura	59
Bibliografia	60

Índex de figures

Figura 1: Fotografia de l'església i monument de Sant Quirze de Pedret [1]	5
Figura 2: Part del model 3D de l'església de Sant Quirze de Pedret [5]	6
Figura 3: Model 3D d'e la escultura: Bellesa dominant la força [10]	7
Figura 4: Església de Sant Quirze de Pedret vist des de un mapa topogràfic [13].....	8
Figura 5: Model 3D de l'Església de Sant Quirze de Pedret al segle X [14].....	8
Figura 6: Model 3D de l'Església de Sant Quirze de Pedret al segle XIII [15].....	9
Figura 7: Consulta de cerques a la web de gvSIG i QGIS. [20]	10
Figura 8: Model 3D d'una ciutat ajuntat amb el mapa ortofotogràfic. [22]	11
Figura 9: Esquema de metodologia AGILE [24]	13
Figura 10: Diagrama de Gantt [25].....	19
Figura 11: Diagrama de Gantt actualitzat [26]	21
Figura 12: la Terra dividida en longitud i latitud. [30]	28
Figura 13: Projeccions de la terra en forma cilíndrica, cònica i en pla. [31]	29
Figura 14: Dades vectorials vistes des de QGIS [32]	30
Figura 15: Caselles d'una capa raster [33].....	31
Figura 16: Capa raster utilitzada com a mapa base. [34]	31
Figura 17: Capa raster DEM convertida a model 3D. [35]	32
Figura 18: Capa raster utilitzada com a mapa temàtic. [36].....	32
Figura 19: Interfície de QGIS. Elaboració pròpia.	35
Figura 20: Interfície de QGIS. Barra d'eines de menú. Elaboració pròpia.....	36
Figura 21: Interfície de QGIS. Barra d'eines d'ajuda. Elaboració pròpia.	36
Figura 22: Interfície de QGIS. Barra d'estat i localitzador. Elaboració pròpia.....	36
Figura 23 : Interfície de QGIS. Passos per obrir la vista 3D. Elaboració pròpia.	37
Figura 24: Interfície de QGIS. Vista 3D i vista del mapa. Elaboració pròpia.	37
Figura 25: Interfície de QGIS. Vista de plugins. Elaboració pròpia.	38
Figura 26: Interfície de QGIS. Passos per instal·lar un plugin. Elaboració pròpia.	38
Figura 27: Interfície inicial de Plugin Builder.....	40
Figura 28: Interfície inicial de Plugin Builder.....	40
Figura 29: Carpeta creada per Plugin Builder.	40
Figura 30: Codi del fitxer integració_models.py.....	41
Figura 31: Codi del fitxer integració_models_provider.py.	41
Figura 32: Codi del fitxer integració_models_algorithm.py.	42
Figura 33: Interfície de la "toolbox" a QGIS.....	43
Figura 34: Exemple de interfície d'un algoritme.....	44
Figura 35: Interfície del algoritme Importar Fotografies.	45
Figura 36: Codi de la funció get_geo_infos_from_photo().	46
Figura 37: Interfície de QGIS després d'executar l'algoritme Importar Fotos i carregar una capa raster de base.	46
Figura 38: Part del codi d'Importar fotografies.....	47
Figura 39: Part del codi d'Importar fotografies.....	47

Figura 40: Interfície del algoritme Importar Models 3D.	49
Figura 41: Interfície de QGIS després d'executar l'algoritme Importar Models 3D i carregar una capa raster de base.	50
Figura 42: Interfície de QGIS després d'executar l'algoritme Importar Models 3D i carregar una capa raster de base.	51
Figura 43: Codi de la funció asc2ply().	52
Figura 44: Explicació dels triangles de l'objecte a partir de les cel·les.	52
Figura 45: Carpeta generada per l'algoritme d'Exportar.	53
Figura 46: Codi per calcular les coordenades de textura.	55
Figura 47: Exemple de la descripció dels sol a través de l'azimut i l'altura.[49].	56
Figura 48: Visualització web del contingut exportat.	57
Figura 49: Visualització web del contingut exportat amb les Fotografies mostrades.	57
Figura 50: Visualització web del contingut exportat. Azimut = 200, Elevació = 0.	57
Figura 51: Visualització web del contingut exportat. Azimut = 200, Elevació = 90.	58

Índex de taules

Taula 1: Resum detallat de les tasques que formen el projecte. Elaboració pròpia.	18
Taula 2: Taula amb els preus per hora dels perfils de treballador. Elaboració pròpia.	22
Taula 3: Taula resum de costos de personal. Elaboració pròpia.	23
Taula 4: Consum elèctric per dispositiu. Elaboració pròpia.	24
Taula 5: Taula del sobre cost per possibles imprevistos. Elaboració pròpia.	24
Taula 6: Taula de contingència del 15 % per tipus de cost. Elaboració pròpia.	25
Taula 7: Pressupost del cost total del projecte. Elaboració pròpia.	25

1. Introducció i Contextualització

Durant la història, els éssers humans hem tingut el desig d'entendre el món que ens envolta. Aquesta voluntat de conèixer ens ha impulsat com a espècie i ens ha fet progressar fins a arribar a l'actualitat. Ara, de tota la història, ens queda el patrimoni cultural, la principal font d'informació del passat que forma part dia a dia de les societats modernes.

Com a éssers humans tenim el deure de preservar i conservar el patrimoni cultural. Amb les tecnologies actuals s'obre un ventall de possibilitats noves per poder preservar el patrimoni de manera molt diferent de la utilitzada durant molts anys.



Figura 1: Fotografia de l'església i monument de Sant Quirze de Pedret [1]

1.1. Context

La facultat d'Informàtica de Barcelona, imparteix el Grau en Enginyeria Informàtica, el qual té diferents mencions. Aquest treball està enfocat en la menció de la computació, en l'àmbit de la computació gràfica. El projecte es realitza dins del marc de desenvolupament del grup d'investigació ViRVIG [2] que està especialitzat en la visualització gràfica, realitat virtual i interacció gràfica.

Un dels projectes en els quals treballa aquest grup d'investigació, és el projecte EHEM [3], projecte guanyador de la Iniciativa de Programació Conjunta sobre la conservació i protecció del Patrimoni Cultural (JPI CH) [4]. Com es pot veure a la Figura 2, el projecte EHEM proporciona als historiadors i investigadors reconstruccions virtuals del patrimoni artístic medieval (sobretot aquells amb murals) que s'aproximen a l'original en diferents èpoques, tot incorporant el coneixement històric-artístic i la perspectiva diacrònica. En aquest treball es pretén ampliar les solucions existents per facilitar als historiadors interactuar amb l'entorn de les reconstruccions virtuals.



Figura 2: Part del model 3D de l'església de Sant Quirze de Pedret [5]

1.2. Definició de conceptes

A continuació, es defineixen alguns conceptes necessaris per facilitar la comprensió d'aquest projecte.

1.2.1. Sistema d'Informació Geogràfica (SIG)

Un Sistema d'Informació Geogràfica [6] és una eina per poder treballar amb informació georeferenciada, és a dir, informació que ve acompanyada d'una posició geogràfica.

Les funcions que generalment ofereix un SIG són:

- Llegir, editar, guardar i gestionar informació espacial.
- Analitzar dades geogràfiques, o bé la temàtica o bé la component espacial.
- Generació de resultats com mapes, gràfics, informes...

1.2.2. QGIS

Quantum GIS, o normalment conegut com QGIS [7] és un Sistema d'Informació Geogràfica, el qual és capaç de visualitzar, editar i analitzar dades geogràfiques com s'ha explicat al punt anterior.

Un dels grans avantatges que aporta QGIS és que és de codi lliure i qualsevol persona és capaç d'utilitzar-lo en diversos sistemes operatius. També ofereix la possibilitat de modificar el codi per tal d'afegir noves funcions.

Gary Sherman, el seu fundador, va començar el desenvolupament de QGIS l'any 2002. Però no és fins al 2009 quan finalment alliberen la versió 1.0 kore.

1.2.3. Model 3D

Un model 3D [8] és la representació matemàtica de qualsevol objecte tridimensional en un entorn software 3D. L'objecte pot ser real o imaginari. Per a la realització d'un model 3D és necessari tenir en compte les propietats físiques, com poden ser les textures, grandària, vèrtexs... Generalment, aquests models 3D es guarden en fitxers de tipus .obj [9].



Figura 3: Model 3D d'e la escultura: Belleza dominant la força [10]

1.2.4. Plugin

Els plugins són programes complementaris que amplien les funcions de les aplicacions o de pàgines web. A l'instal·lar-los s'afegeixen noves funcionalitats al programa. El programa i el plugin es poden comunicar a través de l'API [11] (Application Programming Interface) que es pot traduir com interfície de programació d'aplicacions.

Aquest API permet als programadors externs utilitzar determinades funcions del programa original sense canviar el seu codi font. Per tant, com fa servir part del programa original, els plugins són extensions de funcionalitats que no requereixen augmentar significativament la grandària de l'aplicació principal.

1.2.5. Format JPG/JPEG

El format JPEG [12] ve de l'acrònim de Joint Photographic Experts Group, que és un algorisme dissenyat per comprimir imatges.

Normalment, aquest format és conegut com JPG ja que als sistemes operatius l'extensió dels fitxers no pot ser superior a les 3 lletres. Aquest grup es va iniciar l'any 1982, i no és fins a l'any 1992 que es va publicar l'article que el va fer popular.

1.3. Descripció del problema

Els historiadors, sovint es dediquen a investigar monuments de patrimoni cultural, no sempre tenen la necessitat o la facilitat d'anar personalment i fer les investigacions. Per això, és convenient escanejar els monuments i crear models 3D. Amb aquests models els historiadors tenen la facilitat de treballar des de casa amb una rèplica. El projecte EHEM permet crear aquests models com es pot veure a la Figura 2.

Però, a vegades el que volen investigar no només es troba al monument, sinó que volen explorar el seu context, és a dir, els interessa saber el terreny que l'envolta, els desnivells que té, quin és el riu més proper, quina població és la més propera, que hi ha en direcció a la porta...

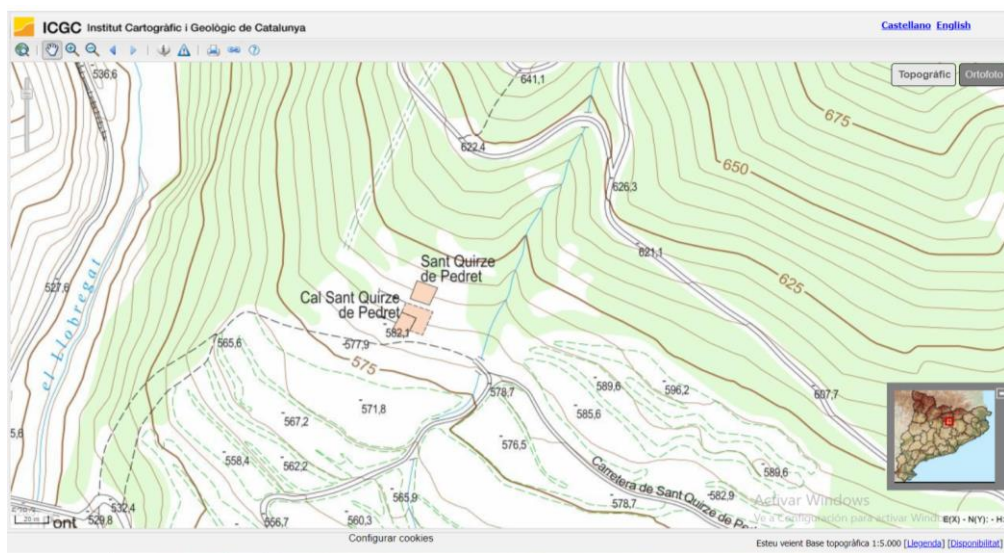


Figura 4: Església de Sant Quirze de Pedret vist des de un mapa topogràfic [13]

De vegades, també pot donar-se el cas que l'historiador tingui els plànols/informació d'un edifici en diferents èpoques. Amb aquests plànols també es poden arribar a crear els models 3D, però requereixen alguna eina on fer la comparació amb els models 3D que sigui més senzilla. A les figures 5 i 6 es pot veure l'església de Sant Quirze de Pedret en dues èpoques diferents



Figura 5: Model 3D de l'Església de Sant Quirze de Pedret al segle X [14]



Figura 6: Model 3D de l'Església de Sant Quirze de Pedret al segle XIII [15]

1.4. Actors implicats

És ben clar que el projecte va especialment dirigit als investigadors d'humanitats (demografia, geografia, història, sociologia...), tot i que també es podria arribar a utilitzar en altres casos menys enfocats a professionals. Però com l'objectiu del projecte és facilitar la investigació als historiadors, són ells qui principalment faran ús d'aquesta nova eina que els permet explorar no només el monument sinó també el seu entorn. Per tant, el projecte intenta que l'aprenentatge dels historiadors per poder fer servir aquesta nova funció sigui mínima.

Un altre grup implicat és el mateix ViRVIG, que ha estat l'encarregat de la comunicació amb els historiadors per facilitar l'enteniment de l'eina que necessitaven. D'aquest equip forma part el director del projecte Oscar Argudo i el codirector Carlos Andújar.

Per últim, jo Xavier Valls Ruiz també estic implicat i soc l'autor en aquest treball de fi de grau.

1.6. Justificació de la tria

Per dur a terme la realització del projecte hi ha diverses coses que s'han de decidir abans de començar. Primerament, tot i que es podria crear un sistema d'informació geogràfica des de zero utilitzant python, és molt més convenient fer ús d'algun ja existent, ja que seran els historiadors els que faran ús del plugin desenvolupat. Per tant, hem de facilitar al màxim la utilització d'aquest, doncs els historiadors no tenen per què tenir coneixements a l'àmbit de la informàtica. Això s'aconseguirà fent ús un SIG ja creat.

Els sistemes d'informació geogràfica més usats actualment són el QGIS [16], gvSIG [17] i ARCGIS [18]. Les característiques principals de cada SIG són:

- QGIS: SIG de codi lliure. Disposa d'actualitzacions cícliques. És ràpid i robust, amb una interfície senzilla. Té més de 900 plugins complementaris. Basat amb Python. Suporta nombroses funcionalitats i formats vector, raster i base de dades.

- gvSIG: Programa de codi lliure. És multi plataforma, multi idioma i basat en Java. Conegut per la seva interfície amigable i per ser capaç d'accedir a formats vectorial i raster més utilitzats actualment.
- ARCGIS: Sistema d'Informació Geogràfica de pagament. Interfície moderna basada en cintes, té vistes 2D i 3D. És capaç de processar formats vectorials i raster.

La primera diferència clara és que els dos primers són de codi obert i el tercer SIG és de pagament. Per aquesta raó ARGIS no s'utilitzarà, ja que l'objectiu del projecte és que el pugui utilitzar qui vulgui.

Entre els dos SIG restants es decideix fer ús del primer, QGIS, ja que és una mica més senzill i permetrà que sigui més fàcil emprar el plugin. També, disposa de possibilitat d'utilitzar QGis2ThreeJS [19] un altre plugin ja creat que facilita la visualització de models 3D a navegadors web. A més, com es pot veure ala figura 7, és el SIG més utilitzat entre els usuaris.

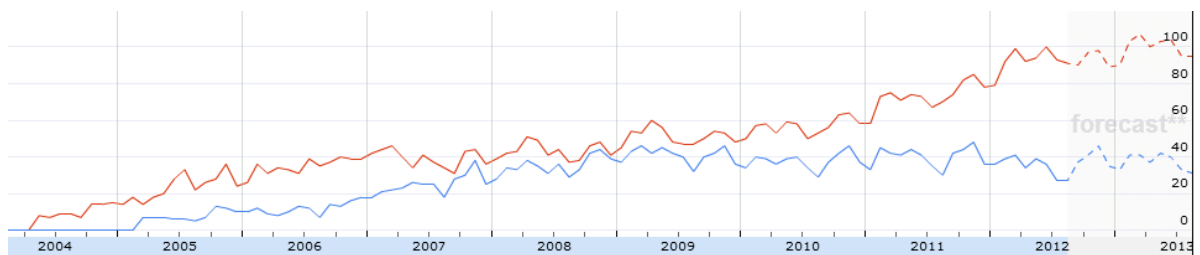


Figura 7: Consulta de cerques a la web de gvSIG i QGIS. [20]

Finalment, també s'ha de decidir quines dades triar per referenciar el model 3D al SIG. Es podria escanejar el terreny amb drons o amb satèl·lit, però òbviament seria molt costós. Per tant, el projecte farà ús de les dades aportades per l'Institut Cartogràfic i Geològic de Catalunya [21].

2. Abast del Projecte

Com tot projecte, un cop contextualitzat el problema que tracta. És essencial definir l'abast que tindrà la resolució. A aquest apartat es definiran els objectius genèrics, funcionalitats i requeriments. Així com els obstacles i riscos que hi puguin haver.

2.1. Objectius

L'objectiu principal del projecte és desenvolupar una eina que permeti integrar models 3D de patrimoni cultural en Sistemes d'Informació Geogràfica, per tal de facilitar la inspecció d'aquests models, juntament amb altres capes, en el seu context (terreny que l'envolta).

L'entrada que tindrà el plugin creat durant el projecte seran els models 3D i un conjunt de fotos del model. Aquesta informació ja està creada i només s'haurà d'importar. Els models 3D i les fotografies estan acompanyades d'un fitxer amb les seves coordenades

La sortida del programa constarà de dues parts:

- Un projecte de QGIS (carpeta) amb les diferents capes (ortofotos, model del terreny, topogràfic...).
- El model 3D amb el terreny, juntament amb l'aplicació per visualització web.

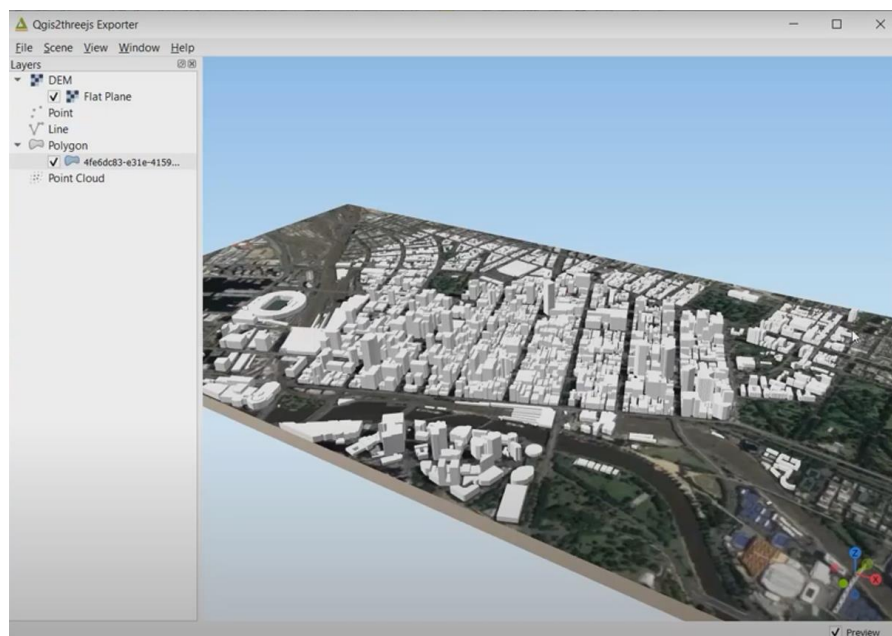


Figura 8: Model 3D d'una ciutat ajuntat amb el mapa ortofotogràfic. [22]

2.2. Funcionalitats

Les funcionalitats principals que ha de tenir el programa són:

- Importar el model 3D (format OBJ).
- Importar les fotografies (format JPG).
- Importar la informació d'enregistrament de les fotografies respecte del model 3D (format Bundler).
- Facilitar la georeferenciació del model 3D i les fotos no enregistrades.
- Crear capes amb les entitats adjacents per representar les càmeres corresponents a les fotografies.
- Permetre a l'usuari determinar la regió d'interès (per exemple, 10 km x 10 km al voltant del model 3D).
- Permetre a l'usuari triar les capes a exportar.
- Exportar capes, model 3D i fotos a un projecte ThreeJS (QGIS2ThreeJS).

2.3 Requeriments

2.3.1. Requeriments funcionals

Tot i que la majoria de requeriments funcionals del programa es defineixen a l'explicació de l'objectiu a continuació es defineixen altres requisits pel funcionament de l'aplicació:

1. Integració a l'aplicació inicial - Tal com hem dit al punt 2.1, el treball parteix de l'aplicació QGIS, per tant, aquest nou plugin ha de ser compatible.
2. Retorna SIG - El programa, un cop utilitzat ha de ser capaç de retornar-nos un Sistema d'Informació Geogràfica de la grandària especificada.

2.3.2. Requeriments no funcionals

A continuació es defineixen els requisits no funcionals del plugin, és a dir, aquells requeriments que no estan estrictament relacionats del funcionament del sistema, però que s'han de tenir en compte:

1. Usabilitat - Com que els historiadors són els que faran servir en gran part aquest programa, i generalment no tenen un alt nivell en informàtica, el programa ha de ser el més senzill possible.
2. Eficàcia - El temps és important, així com qualsevol programa s'ha d'intentar programar de la manera més eficient.
3. Reusabilitat - Per facilitar la integració del programa a altres SIG, és interessant que les tècniques implementades es puguin aplicar a altres projectes.

2.4. Obstacles i Riscos

Tot projecte té les seves dificultats i sempre poden sorgir inconvenients. Per tal de minimitzar els impactes a aquesta secció analitzarem els possibles obstacles que ens podem trobar.

Un altre possible obstacle sorgeix a l'hora de programar, ja que sempre poden aparèixer errors d'implementació, són els errors que apareixen amb major freqüència i s'ha d'anar fent proves per no arrossegar l'error.

Sempre està el risc que es puguin perdre dades per qualsevol tipus de causa, sigui interna o externa al mateix treball. Per tal d'evitar-ho es faran còpies i es guardaran externament.

Finalment, a causa dels obstacles i riscos ja comentats, està el risc de no poder complir els terminis especificats. Així doncs, caldrà una bona planificació i compliment d'aquesta. S'anirà seguint un control per evitar al màxim aquest risc.

2.5. Metodologia

Pel fet que el temps disponible per la realització del treball no és il·limitada, en aquest apartat es justifica la metodologia emprada.

2.5.1. Mètode AGILE

La metodologia que s'utilitzarà per al projecte serà AGILE [23], que com el seu nom indica és una metodologia àgil. Es tracta d'una manera de treballar focalitzada a anar fent esprints de feina, en les diferents tasques del projecte. Com que el temps és limitat, la metodologia encaixa perfectament, ja que així la sensació d'anar progressant serà major.

Per altra banda, permet afrontar el risc d'arrossegar un error durant un temps amb major facilitat. Per què fins que no es completa una tasca no es passa a la següent. A més permet una adaptabilitat i flexibilitat davant dels possibles canvis que pugui experimentar el projecte.

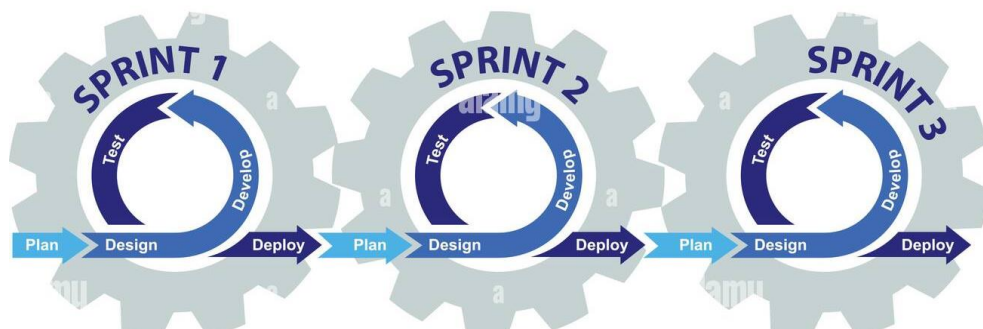


Figura 9: Esquema de metodologia AGILE [24]

El projecte passa per diverses fases de treball. Aquestes fases es poden veure com els sprints que es faran durant la implementació del projecte:

1. Estudi de la interfície de QGIS.
2. Estudi de formes d'expandir QGIS per tal d'afegir el plugin.
3. Estudiar quins formats poden funcionar bé amb el projecte.
4. Escriure script/plugin per QGIS.
5. Avaluació/Validació amb els usuaris finals.

2.5.2. Eines

Per poder seguir la metodologia AGILE, es farà ús d'una eina bàsica que facilitarà el flux del treball.

Aquesta eina que s'utilitzarà durant el projecte és el git/GitHub [25], que és el VCS (Version Control System) escollit. És l'opció més popular i senzilla. Es crearà un repositori on s'anirà afegint el codi i s'anirà actualitzant a mesura que es vagin completant els sprints de la metodologia AGILE.

2.5.3. Mètode de validació

La validació del projecte es farà setmanalment a través de reunions amb el director i codirector per tal d'assegurar que el treball avança adequadament. També s'anirà revisant si s'han assolit les tasques programades per la setmana.

3. Planificació temporal

La durada aproximada del projecte és d'aproximadament 4 mesos. El projecte que va començar oficialment el 21 de febrer de 2022 té previst acabar a finals de juny del mateix any. S'espera una dedicació continuada, d'unes 4 hores diàries, comptant amb algunes variacions ocasionals que puguin sorgir durant la realització del projecte. D'aquest treball de fi de grau en total s'esperen entorn a les 450 hores de dedicació que s'especificaran a continuació.

3.1. Recursos necessaris

Tots els projectes necessiten certs recursos bàsics per a poder desenvolupar-se correctament. En aquest apartat es pot veure els recursos que es necessiten per la realització del projecte.

3.1.1. Recursos humans

En aquest projecte es necessiten 4 perfils diferents. Es requereix un cap de projecte, un investigador, un programador i un tester. No obstant això, com que aquest treball el realitza una sola persona, serà aquesta l'encarregada d'assumir tots els rols.

1. CP - Cap del projecte: Encarregat de la planificació del projecte, escriure la documentació i organitzar les reunions amb l'equip.

2. I - Investigador: El responsable de la investigació del projecte. Duu a terme el disseny i s'encarrega de crear les proves per als usuaris, després avalua els resultats i contribueix amb la memòria.
3. P - Programador: És el responsable d'implementar el plugin.
4. T - Tester: És l'encarregat de fer les proves de validació. Presenta un informe amb els resultats obtinguts.

3.1.2. Recursos materials

En aquesta secció s'exposen els recursos materials necessaris pel correcte desenvolupament del projecte. Aquests recursos seran utilitzats durant totes les tasques del projecte com es pot veure en la figura 10.

- H - Hardware: Únicament es necessita un ordinador portàtil amb connexió a internet. Té un baix consum i proporciona gran flexibilitat.
- S - Software: Només es preveu utilitzar programari gratuït, com ara QGIS on es desenvolupa el plugin, un editor de codi, Google Drive per compartir arxius i Google meet per les reunions.

3.2. Descripció de tasques

Les tasques a realitzar durant el desenvolupament del projecte es poden separar en quatre blocs.

3.2.1. GP - Gestió del projecte

Aquest bloc del projecte agrupa les tasques no tècniques, les quals són fonamentals per una realització exitosa de treball. S'estima que aquestes tasques es podran fer en 141 hores. Com ja s'ha exposat, el cap del projecte serà l'encarregat de la majoria de tasques d'aquest bloc.

GP.1 – Context i abast

Abans de començar amb el treball és imprescindible definir el context en el qual es treballarà, definint l'abast i el propòsit final del projecte. També s'expliquen les metodologies de treball. El tems requerit per aquesta tasca són 25 hores.

GP.2 – Planificació

Per poder complir amb els objectius especificats fa falta una bona planificació. En aquesta tasca es farà una planificació temporal i de recursos a més de buscar alternatives a possibles inconvenients. Aquesta fase ha durat unes 10 hores.

GP.3 – Pressupost

Aquesta tasca consisteix en la creació d'un pressupost per quantificar el cost del projecte. Per cada tasca es farà una estimació del cost, tenint en compte el personal i l'equip necessari. Aquesta tasca s'estima una dedicació de 8 hores.

GP.4 – Informe de sostenibilitat

En aquesta fase es tindran en compte els possibles impactes mediambientals, econòmics i socials del projecte. El temps estimat per aquesta tasca és de 8 hores.

GP.5 – Reunions

La validació del treball s'anirà fent a través de reunions setmanals amb el director i codirector. En aquestes s'aclareixen els dubtes i es va mostrant el desenvolupament. Les reunions són setmanals i d'aproximadament 1 hora de duració, en total són 20 setmanes, per tant, la dedicació total serà d'unes 20 hores.

GP.6 – Documentació

Una part important del treball és la seva memòria, ja que és on s'explica el procés i resultats. La documentació s'anirà fent de forma paral·lela al projecte, d'aquesta manera s'anirà incorporant les seccions que es treballin. És un llarg que s'estima en 60 hores de treball.

GP.7 – Presentació

Consisteix en la preparació de la defensa davant del tribunal que avaluarà el projecte. Consistirà en la preparació del material per la presentació. Aproximadament requerirà 10 hores.

3.2.2. TP – Treball previ

En aquesta secció, s'especifiquen les tasques a realitzar abans del desenvolupament, és a dir, les tasques de preparació i estudi previ. Com és feina de preparació es pot anar fent a la vegada que les tasques de gestió del projecte. La duració estimada és de 35 hores.

TP.1 – Familiarització amb QGIS

QGIS és l'eina principal amb la qual es desenvoluparà el projecte, per tant, és essencial tenir unes nocions bàsiques de com utilitzar-lo. Aquesta tasca se centra en l'aprenentatge de la interfície i de les funcions principals. S'estima que l'aprenentatge serà de 10 hores.

TP.2 – Familiarització amb plugins QGIS

L'objectiu del projecte és acabar creant un plugin, però hi ha diverses maneres de fer-ho. En aquesta fase es valorarà quina opció és la més recomanable en aquest treball i s'estudiarà a fons per poder aplicar-ho. La duració estimada és de 15 hores.

TP.3 – Estudi formats

Hi ha diversos formats que utilitzen els Sistemes d'Informació Geogràfica, en aquesta tasca s'analitzen i s'estudien els més complets. Aproximadament 10 hores de duració.

3.2.3. DP – Desenvolupament del plugin

Un cop fet l'aprenentatge és moment de passar a l'acció. En aquestes tasques es comença a realitzar el plugin, tot seguint un ordre de realització. En total les tasques de desenvolupament s'estima que durin 240 hores.

DP.1 - Disseny

La fase de desenvolupament comença amb el disseny. Fase molt important, ja que decideixen les bases del plugin. Fen ús de l'estudi de la fase anterior s'analitza la millor manera de desenvolupar el plugin. S'estima una dedicació d'entorn les 25 hores.

DP.2 – Importar informació

La primera funció que s'ha d'implementar al plugin és la capacitat d'importar i comunicar els models 3D, les fotografies i els mapes amb el plugin. Aproximadament una dedicació de 22 hores.

DP.3 – Ajuntar model 3D i mapa

Un cop la informació està carregada s'ha de crear una nova capa o conjunt de capes que ajunti el mapa amb el model 3D i les fotografies. Aquesta funció es preveu poder implementar-la en 25 hores.

DP.4 – Representació de les càmeres

Les fotografies també són una part rellevant del patrimoni cultural, ja que ensenyen exactament com són els monuments. Poder mostrar aquestes fotografies de la millor forma possible és molt important. S'estima una dedicació de 28 hores.

DP.5 – Visualitzar en 3D

El model 3D també és crucial, per tant, s'ha d'incorporar una funció que permet la interacció amb el model 3D. La duració aproximada de la tasca és de 20 hores.

DP.6 – Interfície del plugin

En aquesta tasca s'implementa la interfície del plugin. Ha de ser el més senzill possible per facilitar l'ús al màxim. S'estima que la tasca pot arribar a les 22 hores.

DP.7 – Georeferenciació automàtica

La possibilitat de tenir alguna fotografia del monument sense georeferenciació existeix. Així doncs, aquesta tasca crearà la funció de fer-ho automàticament. Amb aproximadament 25 hores.

DP.8 – Exportar per threeJS

Un cop ajuntat el mapa amb el model 3D i les fotografies s'ha de poder exportar el conjunt a threeJS. Aproximadament seran necessàries 20 hores.

DP.9 – Editar threeJs a la web

L'última tasca de la fase de desenvolupament és la d'editar threeJS per modificar la interfície a la web i facilitar al màxim la usabilitat del resultat del plugin. Aproximadament es requereixen 30 hores.

3.2.4. VU – Validació dels usuaris

Un cop desenvolupat el plugin, es realitzen les proves amb els historiadors per estudiar el grau de satisfacció. El total d'hores per la realització de la validació serà de 35 hores.

VU.1 – Preparació

En aquesta fase es prepara la validació del projecte dels investigadors. Aproximadament en 5 hores.

VU.2 – Estudi

Un cop preparat tot, els historiadors faran ús del plugin i es recolliran les dades. La duració aproximada serà de 15 hores.

VU.3 – Avaluació

Amb les dades dels investigadors s'avaluarà el resultat i si cal es farà les correccions corresponents. S'estima que la duració serà de 15 hores.

Codi	Tasca	Temps	Dependència	Rol	Recursos
GP	Gestió del projecte	141 h	-	-	-
GP.1	Context i abast	25 h	-	CP	H, S
GP.2	Planificació	10 h	GP.1	CP	H, S
GP.3	Pressupost	8 h	GP.1 GP.2	CP	H, S
GP.4	Informe de sostenibilitat	8 h	GP.1 GP.2	CP	H, S
GP.5	Reunions	20 h	-	CP, I, P, T	H, S
GP.6	Documentació	60 h	-	CP, I	H, S
GP.7	Presentació	10 h	GP.6	CP	H, S
TP	Treball previ	35 h	-	-	-
TP.1	Familiarització amb QGIS	10 h	-	I, P	H, S
TP.2	Familiarització plugins QGIS	15 h	TP.1	I, P	H, S
TP.3	Estudi formats	10 h	TP.1	I, P	H, S
DP	Desenvolupament del plugin	240 h	TP	-	-
DP.1	Disseny	25 h	-	I	H, S
DP.2	Importar informació	22 h	DP.1	P	H, S
DP.3	Ajuntar model 3D i mapa	25 h	DP.2	P	H, S
DP.4	Representació de les càmeres	28 h	DP.3	P	H, S
DP.5	Visualitzar en 3D	20 h	DP.3	P	H, S
DP.6	Interfície del plugin	22 h	DP.1	P	H, S
DP.7	Georeferenciació automàtica	25 h	DP.4	P	H, S
DP.8	Exportar per threeJS	20 h	DP.3 DP.4 DP.5	P	H, S
DP.9	Editar threeJs a la web	30 h	DP.8	P	H, S
VU	Validació dels usuaris	35 h	DP	-	-
VU.1	Preparació	5 h	-	I, T	H, S
VU.2	Estudi	15 h	VU.1	T	H, S
VU.3	Avaluació	15 h	VU.2	I	H, S

Taula 1: Resum detallat de les tasques que formen el projecte. Elaboració pròpia.

3.3. Diagrama de Gantt

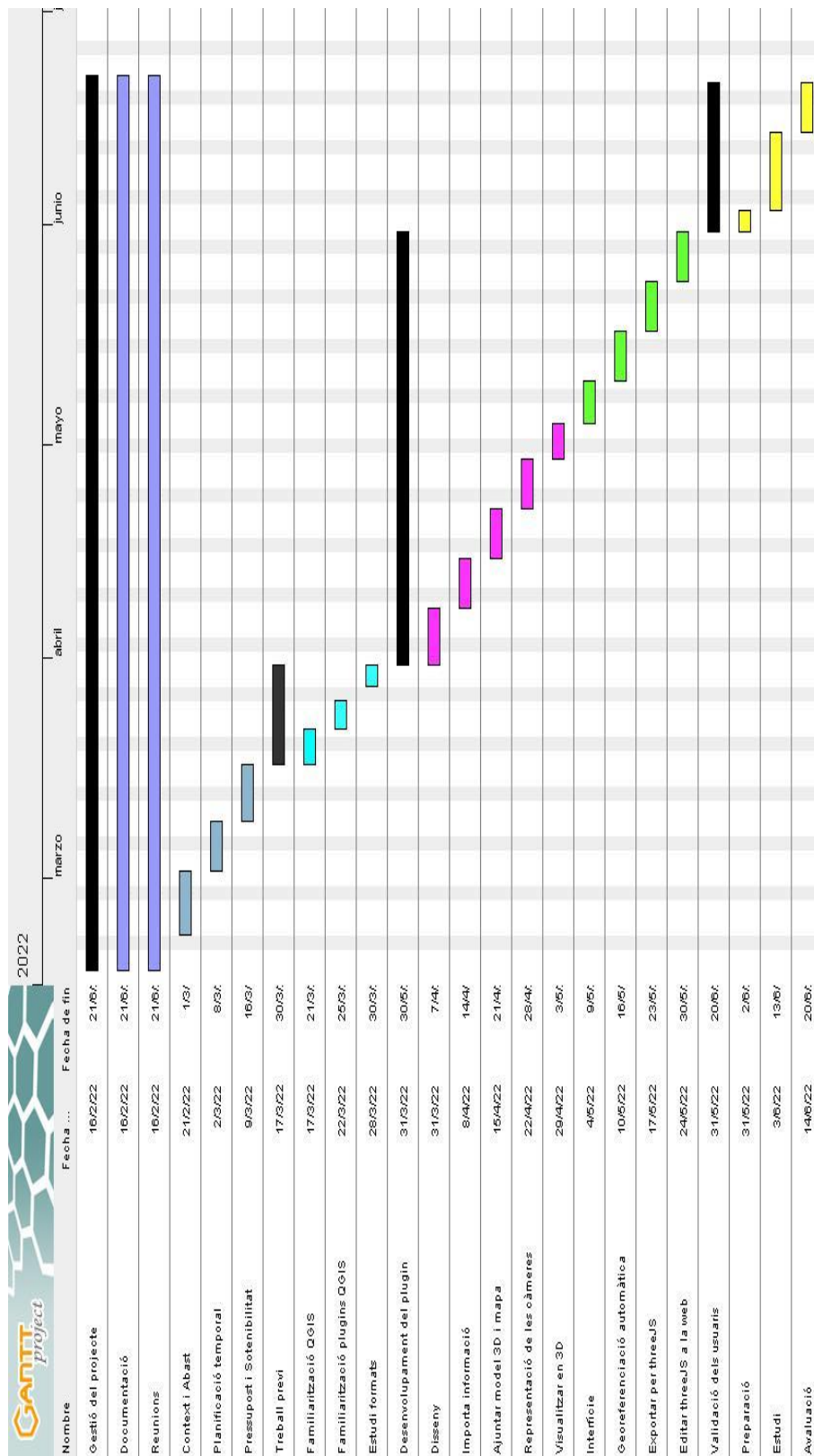


Figura 10: Diagrama de Gantt [25]

A la figura 10 es pot veure el diagrama de Gantt que es seguirà durant la realització del projecte. Cada canvi de color indica un canvi de esprint de la metodologia AGILE.

3.4. Gestió de riscos

Tot i que hi ha reunions setmanals amb el director i codirector per assegurar que el treball sigui un èxit, és una bona idea tenir algun pla B per als possibles inconvenients que puguin passar com ja es va explicar a l'apartat de riscos i inconvenients.

Un dels principals obstacles del qual vàrem parlar va ser el no compliment dels terminis especificats. Per tal d'evitar que pugui a passar, s'ha sobreestimat el cost de les principals tasques de la implementació.

Com que els problemes de perduda d'informació o tenir errors a la implementació són errors que hem d'evitar com sigui, es farà el possible perquè no s'arribin a donar. Per tal d'evitar la perduda d'informació es farà una còpia al núvol cada dia de la feina feta. I per evitar errors d'implementació s'estudiarà d'una manera profunda les eines necessàries.

Finalment, s'ha deixat lliure l'última setmana del projecte. Aquesta setmana servirà de marge en el cas que no es puguin complir els terminis per qualsevol dels imprevistos que puguin sorgir. És un "salvavides" pel pitjor dels casos.

3.5. Diagrama de Gantt actualitzat

Com ja s'ha comentat a l'apartat anterior, és possible que un dels riscos del projecte és el no compliment de les dades especificades. Degut a la poca experiència en projectes tan grans com aquest, la planificació temporal inicial s'ha vist alterada ja que era impossible poder seguir els terminis especificats. Per tant, s'ha decidit augmentar el temps de realització del projecte fins al 10 d'octubre.

Aquesta poca experiència planificant, sumada a problemes personals de l'autor del projecte, ha fet que es tingui que crear un nou diagrama de Gantt actualitzat amb les dades correctes. A més hi ha algunes tasques que s'han hagut d'eliminar ja que tot i que s'ha modificat la planificació no s'ha vist possible la realització d'aquestes.

A la següent figura 11 es pot veure com queda finalment el diagrama de Gantt actualitzat. Però, aquest diagrama no inclou juliol ni agost ja que l'autor del projecte no ha estat disponible per poder realitzar el projecte.

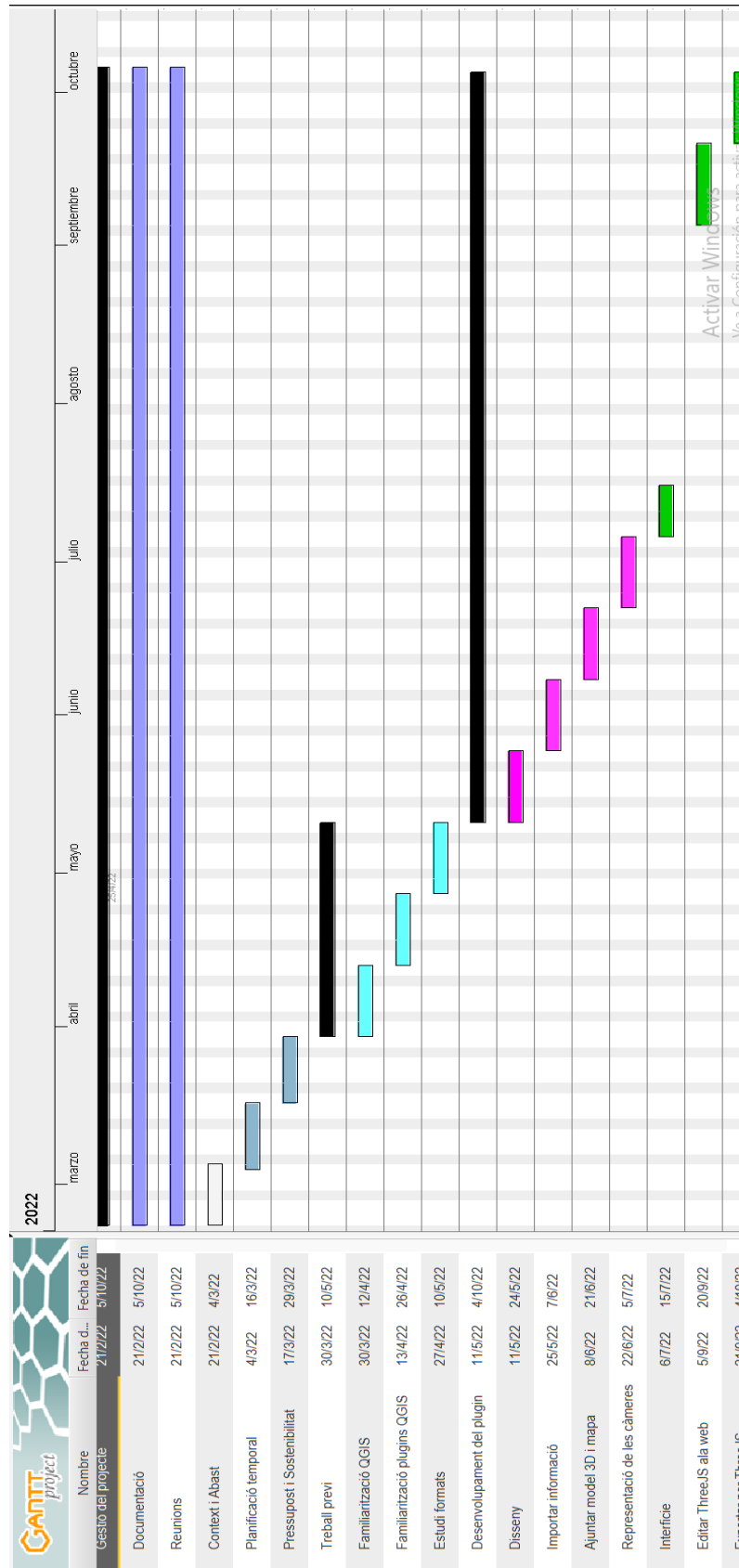


Figura 11: Diagrama de Gantt actualitzat [26]

4. Gestió econòmica

Un cop realitzada la planificació temporal del projecte, s'estimen els costos necessaris pel desenvolupament. Hi ha diversos tipus de costos, com els costos de personal, el de les eines de treball, o l'espai de treball entre d'altres. A més, per poder superar possibles obstacles ja exposats anteriorment es crea un pla de contingència, una partida d'imprevistos i s'exposen els mecanismes per controlar el pressupost.

4.1. Costos de personal

Per començar a fer el pressupost serà necessari saber el cost per hora de cada rol de treballador, juntament amb la planificació s'obtindrà el cost del personal per la realització del projecte.

Segons un estudi de la plataforma PagePersonnel [27], a la taula 2, es pot veure una taula on s'indiquen les retribucions que li correspon a cada perfil de treballador en euros per hora. Aquests valors han estat ajustats per incloure l'aportació establerta a la Seguretat Social.

Rol	Sou net	SS	Sou brut
Cap de projecte	20€/h	6€/h	26€/h
Investigador	17€/h	4.10€/h	22.10€/h
Programador	13€/h	3.90€/h	16.90€/h
Tester	13€/h	3.90€/h	16.90€/h

Taula 2: Taula amb els preus per hora dels perfils de treballador. Elaboració pròpia.

Un cop definits els preus per hora dels perfils de treballadors ja es pot calcular el CPA (cost per activitat). Això es fa a través de la multiplicació de les hores per tasca definida en l'apartat 4.2 i el preu per hora.

Codi	Tasca	Temps	Rol	Cost/h	Cost
GP.1	Context i abast	25 h	CP	26€	650€
GP.2	Planificació	10 h	CP	26€	260€
GP.3	Pressupost	8 h	CP	26€	208€
GP.4	Informe de sostenibilitat	8 h	CP	26€	208€
GP.5	Reunions	20 h x4	CP, I, P, T	26€, 22.1€ 16.9€, 16.9€	520€, 442€ 338€ 338€
GP.6	Documentació	60 h	CP 40h, I 20h	26€, 22.1€	1040€, 442€
GP.7	Presentació	10 h	CP	26€	260€
TP.1	Familiarització amb QGIS	10 h	P	16.90€	169€
TP.2	Familiarització plugins QGIS	15 h	P	16.90€	253.5€
TP.3	Estudi formats	10 h	P	16.90€	169€
DP.1	Disseny	25 h	I	22.10€	552.5€
DP.2	Importar informació	22 h	P	16.90€	371.8€

DP.3	Ajuntar model 3D i mapa	25 h	P	16.90€	422.5€
DP.4	Representació de les càmeres	28 h	P	16.90€	473.2€
DP.5	Visualitzar en 3D	20 h	P	16.90€	338€
DP.6	Interfície del plugin	22 h	P	16.90€	371.8€
DP.7	Georeferenciació automàtica	25 h	P	16.90€	422.5€
DP.8	Exportar per threeJS	20 h	P	16.90€	338€
DP.9	Editar threeJs a la web	30 h	P	16.90€	507€

VU.1	Preparació	5 h	I, T	22.1€, 16.90€	110.5€, 84.5€
VU.2	Estudi	15 h	T	16.90€	253.5€
VU.3	Avaluació	15 h	I	22.10€	331.5€

Total					9874.80€

Taula 3: Taula resum de costos de personal per tasca. Elaboració pròpia.

Rol	Sou brut	Hores	Cost
Cap de projecte	26€/h	121 h	3146 €
Investigador	22.10€/h	85 h	1878.50 €
Programador	16.90€/h	247 h	4174.30 €
Tester	16.90€/h	40 h	676 €
Total	-	473 h	9874.80 €

Taula 4: Taula resum de costos de personal. Elaboració pròpia.

4.2. Costos genèrics

Aquests costos es calculen de manera genèrica, ja que no depenen de les tasques, perquè intervenen en totes les tasques. Aquests costos seran les amortitzacions, l'espai de treball i el consum elèctric i d'Internet.

4.2.1. Amortitzacions

Durant el projecte només es necessitarà un ordinador portàtil de gamma mitjana d'un cost aproximat de 600 €. Suposant que la mitjana de vida útil d'aquest ordinador és de 60 mesos (5 anys) i la duració del projecte és de 450 hores, si els treballadors dediquessin 40 hores setmanals requeriria d'onze setmanes, que equival a 2 mesos i mig. L'amortització de l'ordinador és de $2.5/60 * 600 = 25€$.

La part software d'aquest projecte com ja s'ha exposat anteriorment és totalment gratuïta, per tant, l'amortització és de **0€**.

4.2.2. Espai de treball

L'espai on es desenvoluparia el projecte seria una oficina, la qual hauria de tenir la capacitat de poder mantenir les quatre persones de l'equip durant tot el projecte.

A Internet s'ha trobat una oficina de 50 m² per 550€ al mes, amb l'immobiliari necessari per començar a treballar. Com la realització del projecte requereix aproximadament de 450 hores, que equival a 2 mesos i mig. El cost de l'espai de treball seria de 550x 2.5 = **1375 €**.

4.2.3. Consum elèctric i Internet

Comparant ofertes entre diferents companyies d'Internet, s'ha trobat una oferta on es paguen 20€ [28] mensuals d'Internet. Com s'acaba d'exposar la duració del projecte és de 2 mesos i mig. Per tant, el cost d'Internet durant el treball serà de **50€**.

El preu de l'electricitat actualment és de 0.42844€/kWh [29], podem estimar el cost de mantenir en funcionament del portàtil i la llum, com es pot veure a la taula 5.

Dispositiu	Potència	Hores	Consum	Cost
Portàtil	30 W	450 h	13.5kWh	5.79€
Bombeta	2 W	450 h x 6	5.4kWh	2.31€
Total				8.10€

Taula 5: Consum elèctric per dispositiu. Elaboració pròpia.

4.3. Imprevistos

Com ja s'ha comentat a l'apartat 2.4, existeixen possibles riscos i imprevistos que poden fer que el plantejament temporal i pressupost canviïn. Per això s'ha quantificat el risc que hi ha i s'ha preparat un cost addicional perquè si arriba a passar, el projecte estigui preparat. A la taula 6 es detalla el cost.

El principal risc que es va exposar és el de no poder complir el termini a temps. Però, per tal d'evitar-ho, s'ha augmentat el temps de desenvolupament del plugin. Al programador se li afegiran 30 hores, les quals tenen un cost de 507€. El risc que aquestes hores siguin necessàries s'estima en un 25%.

L'altre risc que s'ha de tenir en compte és la possibilitat que el dispositiu falli. En aquest cas serà necessari comprar un de nou. S'estima que la probabilitat de què el dispositiu falli és del 5%.

Imprevist	Cost	Risc	Cost total
Augment temps	507 €	25 %	126.75 €
Portàtil	600 €	5 %	30 €
Total	1107 €	-	156.75 €

Taula 6: Taula del sobre cost per possibles imprevistos. Elaboració pròpia.

4.4. Contingències

Com tot projecte, és important afegir un sobrecost per cobrir possibles obstacles i imprevistos. En aquest cas, al tractar-se d'un treball amb informàtic, la probabilitat de trobar problemes durant el desenvolupament no és 0, en concret s'estima un 15% de sobre cost. A la taula 7 es detalla la contingència en cada cas.

Tipus	Cost	Contingència
Personal	9874.80 €	1481.22 €
Espai	1375 €	206.25 €
Software	0 €	0 €
Hardware	25 €	3.75 €
Llum i internet	58.10 €	8.75 €
Total	12882.90 €	1699.97 €

Taula 7: Taula de contingència del 15 % per tipus de cost. Elaboració pròpia.

4.5. Cost total

Un cop presentats tots els costos del projecte, a la taula 8 es presenta el pressupost final del treball. El qual inclou tots els costos mencionats anteriorment. El cost total del projecte és de 14972.08 €.

Concepte	Cost
Personal	9874.80 €
Genèrics	1458.10 €
Imprevistos	156.75 €
Contingències	1699.97 €
TOTAL	13189.62 €

Taula 8: Pressupost del cost total del projecte. Elaboració pròpia.

4.6. Control de gestió

Durant el transcurs del projecte es faran revisions de costos per tal d'assegurar que la desviació d'aquests amb els costos estimats anteriorment és la mínima possible.

La forma més convenient de realitzar aquestes revisions serà a través de les reunions setmanals, ja que no hi ha cap necessitat d'utilitzar un mecanisme paral·lel per dur a terme aquesta funció. Altrament, d'aquesta manera, es podrà corregir de forma proactiva i eficaç les possibles desviacions que es detectin durant el transcurs del projecte.

S'utilitzaran les mètriques següents:

$$\text{Desviació de cost} = (CE - CR) \times CHR$$
$$\text{Desviació del consum} = (CHE - CHR) \times CE$$

On:

- **CE:** Cost Estimat
- **CR:** Cost Real

- **CHR:** Consum d'Hores Real
- **CHE:** Consum d'Hores Estimat

Gràcies a aquest control, es podrà detectar qualsevol desviació i saber a quina tasca i quin tipus de desviació. Si les desviacions detectades són degudes a imprevistos ja mencionats, es farà ús de la partida d'imprevistos per cobrir els costos. En altre cas, s'utilitzarà la partida de contingència explicada al punt 5.4. En el pitjor dels casos, si les desviacions són excessivament grans es plantejarà l'opció d'ajustar l'abast del projecte per no sobrepassar el pressupost disponible.

5. Sostenibilitat

La informàtica com qualsevol altre sector, té implicacions positives i negatives en la vida de les persones, sigui aplicades a condicions ambientals o de la societat en general. Per això s'ha de ser conscient de les possibles implicacions d'aquest projecte. En aquest apartat s'exposarà una anàlisi de sostenibilitat del projecte tenint en compte aquestes tres dimensions: l'ambiental, l'econòmica i la social.

5.1. Dimensió ambiental

El desenvolupament del plugin d'aquest projecte no requereix cap hardware específic, únicament es requereix un ordinador, el qual s'ha optat per un portàtil, ja que facilita la flexibilitat i el consum energètic és inferior a un de fix. Pel que fa a l'ús del paper, es limitarà la utilització d'aquest, i únicament es farà servir per a la impressió de la memòria final.

Aquest projecte es construeix utilitzant models 3D, els quals els obtenim utilitzant una eina ja creada i, per tant, evitem la despesa de recursos i energia per crear-la. D'igual forma succeeix amb el nostre TFG, ja que si els investigadors haguessin d'aprendre a fer servir QGIS profundament per poder fer la incorporació del model 3D ells mateixos, requeririen recursos i d'energia extra per poder fer-ho. En canvi, usant el plugin creat a aquest projecte serà molt més ràpid i, per tant, el consum elèctric serà menor.

Una altra opció que permetria als historiadors investigar sobre el terreny on es troba el monument seria la de desplaçar-se personalment, fet que requeriria algun tipus de vehicle i combustible, per tant, poder fer-ho des d'un ordinador redueix la contaminació ambiental.

5.2. Dimensió econòmica

El cost del projecte exposat amb detall al capítol anterior, està desglossat apropiadament per tal de distingir els costos associats als recursos naturals, els associats a recursos humans o els associats a imprevistos.

Com ja s'ha exposat, el software emprat és de codi obert, i el temps és el més ajustat possible, per tant, sembla difícil reduir encara més el cost del projecte, ja que no hi ha espai per on poder retallar la despesa.

A la dimensió ambiental s'ha exposat el fet que els investigadors no requereixen d'aprendre a utilitzar QGIS profundament, fet que repercuteix directament amb la dimensió econòmica, ja que les hores que s'estalvien costarien diners.

Aquest projecte, al tractar-se d'un treball amb caire de facilitar la recerca, no és viable econòmicament, pel fet que no s'utilitzarà per comercialitzar. El producte es posarà a l'abast de tothom que ho requereixi de forma pública.

5.3. Dimensió social

Al contrari que en la dimensió econòmica, el projecte és molt viable a la dimensió social. Això és degut al fet que la finalitat d'aquest projecte és facilitar la preservació del patrimoni cultural i la de facilitar als investigadors aprofundir en les estructures de patrimoni cultural.

Aquest tipus de preservació del patrimoni cultural és molt innovador, ja que és diferent dels usats normalment com pot ser el paper. A part, a l'utilitzar una rèplica exacta del monument, si en algun cas s'arribés a perdre el monument original en tindríem una reconstrucció digital exacta.

L'eina creada durant el projecte, posteriorment es podrà fer servir per crear una de nova més enfocada en la població, permetent visitar digitalment alguns dels monuments de patrimoni cultural.

6. Bases Teòriques

Per facilitar la comprensió del setè capítol, el qual tracta el desenvolupament del projecte. Es requereix d'algunes nocions més avançades sobre els SIG i el programa on és desenvolupa el projecte, QGIS.

Per tant, en aquest capítol s'introduiran els conceptes relacionats amb els Sistemes d'Informació Geogràfica, més àmpliament que a l'apartat 1.2 , i també s'aprofundirà en les característiques més rellevants de QGIS.

6.1 Conceptes

6.1.1 Sistema de Referència de Coordenades

Un Sistema de Referència de Coordenades (SRC) és el sistema que s'utilitza per representar qualsevol punt de la terra. El punt està definit per tres números denominats coordenades.

Existeixen molts Sistemes de Referència de Coordenades, tots aquests sistemes estan dividits en dues classes: els sistemes de coordenades geogràfiques i els sistemes de coordenades projectades.

6.1.1.1. Sistema de Coordenades Geogràfiques

Els Sistemes de Referència de Coordenades Geogràfiques són molt comuns, i són els que la majoria de gent coneix. Aquests sistemes són els que utilitzen com a coordenades el graus de latitud, longitud i altitud.

Per poder representar els graus de latitud, existeixen les línies de latitud. Aquestes línies separen la terra en 180 seccions de nord a sud, sent la línia que passa per l'Equador la central, i totes les demás són paral·leles a aquesta. Cada secció s'identifica digitalment amb un grau de latitud que va del -90 al 90. La línia del Equador és la número 0, a l'hemisferi nord són els valors de latitud positius i a l'hemisferi sud els valors negatius. Totes les seccions tenen la mateixa distància entre línies.

Per altre banda, les línies de longitud, conegudes com meridians, són perpendiculars a l'Equador i convergeixen als pols. Aquestes es divideixen en 360 seccions que s'identifiquen digitalment del -180 al 180. En les línies de longitud, la línia central es la que passa per Greenwich, Anglaterra.

Com que la terra és aproximadament rodona, la distància que separa cada línia de longitud a l'Equador no és la mateixa que als pols. Únicament a l'Equador la distància entre seccions és la mateixa que la distància entre seccions de latitud. Per altre banda, a un punt en els pols, on convergeixen la distància és 0.

Un cop traçades les línies, com es pot veure a la figura 12, l'àrea que cobreixen les seccions properes a l'Equador és aproximadament de 12363 km². Amb una secció tan gran és difícil determinar la posició amb una coordenada exacta amb només els graus. Per aquesta raó, cada grau es divideix també en minuts i segons. Cada grau es compon de 60 minuts i cada minut de 60 segons.

Per últim està el grau d'altitud, el qual és l'encarregat de representar l'altura d'un punt. En aquest cas, la línia central es troba al nivell del mar. I la distància entre graus és com la del Equador, 111,12 km.

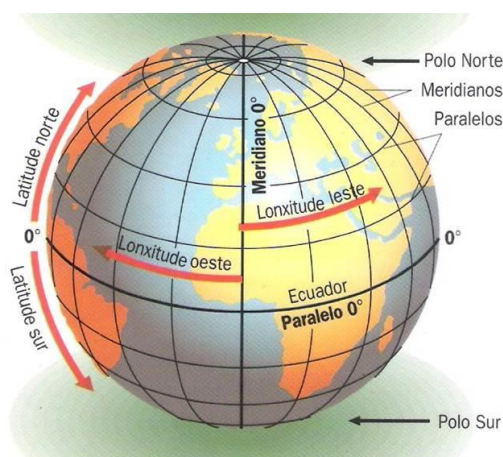


Figura 12: la Terra dividida en longitud i latitud. [30]

6.1.1.2. Sistema de Referencia de Coordenades projectades

Les projeccions es poden imaginar com si es col·loqués una llum dins d'una esfera transparent representant la Terra amb les zones terrestres opaques. Les imatge projectades en un pla bidimensional es poden col·locar de diferents formes. A la figura 13, es pot veure com es representaria en forma cilíndrica, cònica o en pla.

La projecció cartogràfica és la funció matemàtica de transformació de les coordenades esfèriques de longitud i latitud que indiquen posicions sobre la superfície aproximadament el·lipsoidal de la Terra a coordenades cartesianes que indiquen posicions en el pla.

La part negativa de les projeccions és que no poden representar amb absoluta precisió la Terra. Això és degut a que tots els sistemes de projecció intenten amb funcions matemàtiques que no es vegin distorsionats en la conformitat angular, l'àrea o la distància, però no existeix cap sistema de projecció capaç de no distorsionar-ne almenys un, tot i que aconsegueixen resultats molt semblants a la realitat.

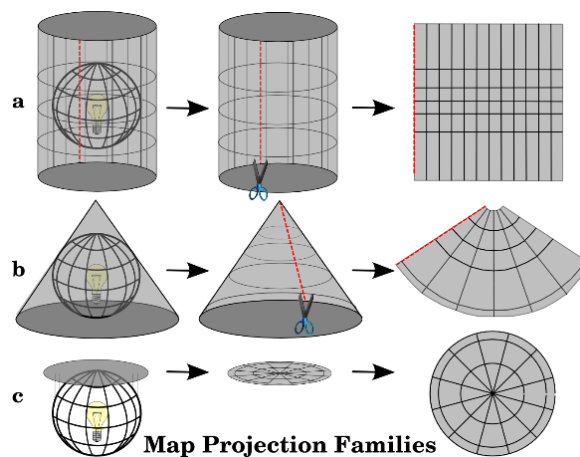


Figura 13: Projeccions de la terra en forma cilíndrica, cònica i en pla. [31]

6.1.2. Dades SIG / Capes

La informació que normalment s'utilitza als Sistemes d'Informació Geogràfica té forma geogràfica, però també s'utilitza informació temàtica. Una de les característiques principals dels SIG és la facilitat d'associar informació no geogràfica amb informació geogràfica. De fet, gràcies a la digitalització és possible guardar molta informació diferent d'un mateix lloc, una cosa molt difícil als mapes tradicionals.

Les capes són el mètode d'organització de la informació espacial segons les seves característiques, podent agrupar per temes. Per exemple hi ha capes on es descriuen carreteres, desnivells, ciutats...

A continuació s'exposen els dos tipus de dades més utilitzats normalment.

6.1.2.1. Dades vectorials

Les dades vectorials són una manera de representar objectes del món que ens envolta dins de un ambient SIG. Un objecte espacial pot ser qualsevol cosa, com per exemple un arbre, un riu, un bosc...

Les dades vectorials es representen com un conjunt de coordenades que indiquen la posició i uns certs atributs, els quals poden ser informació numèrica o textos que descriuen els objectes espacials.

Les dades vectorials s'utilitzen generalment per representar punts, línies i àrees en un context de SIG. A la figura 14 es pot veure un exemple.

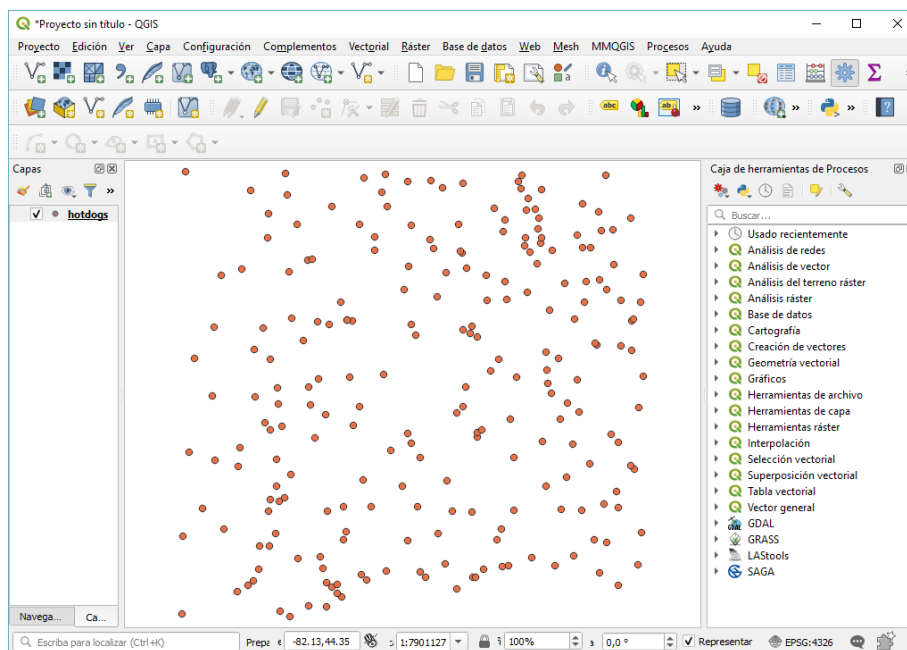


Figura 14: Dades vectorials vistes des de QGIS [32]

6.1.2.2. Dades raster

Al contrari que les capes vectorials, que es representen amb geometria, les dades raster es componen d'una matriu de píxels, també conegudes com caselles, com es pot veure a la figura 15 . Cada un d'aquests píxels té un valor que representa la zona coberta del món real.

La part negativa d'aquest tipus de dades es que si t'apropes molt, es començaran a veure en blocs els píxels, tot depenent de la resolució de la capa.

10.44	10.53	10.62	10.69	10.85	10.95	11.02	11.08	11.09	11.14				
10.56	10.68	10.64	10.91	10.82	10.61	10.58	10.58	10.71	10.7				
10.64	10.87	10.91	10.88	10.89	10.73	10.59	10.58	10.63	10.7				
10.8	10.84	11.08	10.99										
11.19	11.23	10.96	11.08	10.44	10.53	10.62	10.69	10.85	10.95	11.02	11.08	11.09	11.14
11.23	11.42	11.31	10.91	10.56	10.68	10.64	10.91	10.82	10.61	10.58	10.58	10.71	10.7
11.36	11.42	11.32	11.08	10.64	10.87	10.91	10.88	10.89	10.73	10.59	10.58	10.63	10.7
11.26	11.34	11.25	11.07	10.8	10.84	11.08	10.99	10.96	10.66	10.5	10.43	10.56	10.68
11.17	11.16	11	10.97	11.19	11.23	10.96	11.08	10.97	10.89	10.59	10.64	10.77	10.47
10.7	10.78	10.81	10.82	11.23	11.42	11.31	10.91	11	10.83	10.78	10.76	10.52	10.3
				11.36	11.42	11.32	11.06	10.81	10.89	10.75	10.45	10.44	10.32
				11.26	11.34	11.25	11.07	10.78	10.65	10.74	10.65	10.43	10.38
				11.17	11.16	11	10.97	10.68	10.7	10.52	10.57	10.5	10.36
				10.7	10.78	10.81	10.82	10.78	10.54	10.59	10.39	10.46	10.57

Figura 15: Caselles d'una capa raster [33]

Normalment aquest tipus de dades s'utilitza quan la informació que es vol mostrar és continua i no pot ser fàcilment dividida en entitats vectorials. Una de les utilitzats més rellevants d'aquest tipus de dades és la capacitat de representar les fotografies que fan els satèl·lits que orbiten la terra, la representació de fenòmens com la temperatura o la elevació del terreny.

Però, també pot ser útil per representar dades temàtiques (discretes) com l'ús de la superfície de la terra.

A continuació s'expliquen les quatre categories principals on s'utilitzen les capes raster.

- Mapa base

Aquesta funcionalitat és de les més utilitzades quan s'utilitzen capes raster, s'utilitza com a capa complementària en la visualització d'altres capes d'entitats. Per exemple, poden ser fotografies que hagin captats els satèl·lits complementat una altra capa que sigui de un mapa de carreteres com es pot veure a la figura 16. En aquest projecte s'utilitzarà aquesta funcionalitat.

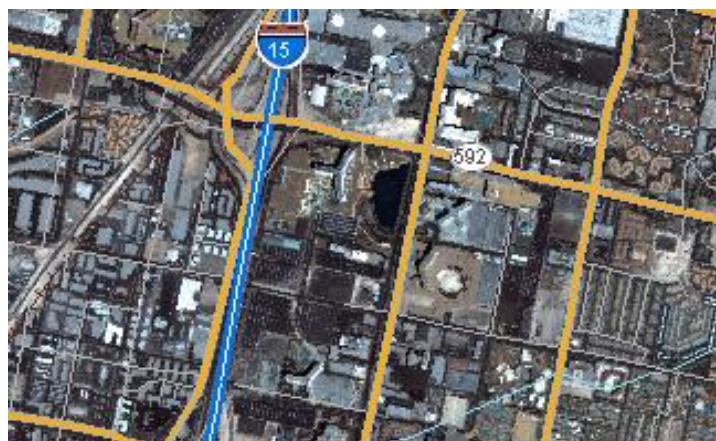


Figura 16: Capa raster utilitzada com a mapa base. [34]

- Mapa de superfície

Les capes raster són bones per representar dades continuades de la superfície i per representar dades que canvien contínuament. Per això, són la millor opció per representar temperatura, precipitacions, densitat de població... Però la utilització més comú dels mapes de superfície és per representar les elevacions del terreny, coneguts com models digitals de elevació o DEM. A la figura 17 es pot veure un fragment d'una capa DEM amb el seu model 3D. Aquestes capes DEM seran de gran utilitat durant el projecte per poder representar el terreny en 3D.

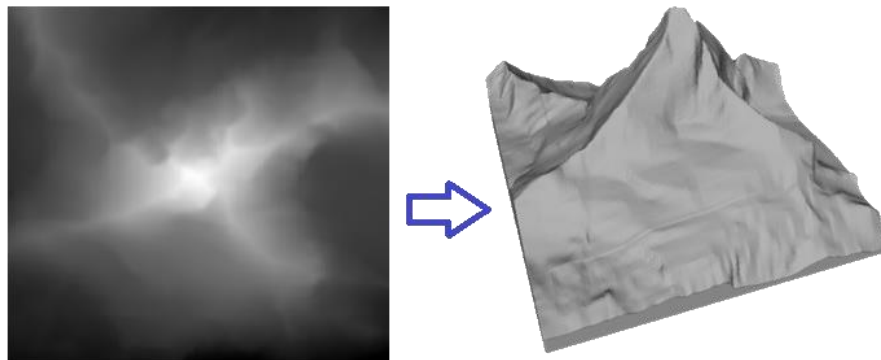


Figura 17: Capa raster DEM convertida a model 3D. [35]

- Mapa temàtic

Els rasters que representen mapes temàtics es poden derivar ja que analitzen altres dades, però, unes de les aplicacions més comunes és la de classificar els objectes de la superfície d'una imatge de satèl·lit, és a dir, agrupar per classes com tipus de vegetació, o aigua i assignar-li un valor categòric. La figura 18 es pot veure un exemple. Altres utilitzats d'aquesta categoria de capa raster seria a partir d'operacions de geoprocessament que combini varies fonts per una activitat específica, com poden ser els mapes de distribució de la població.

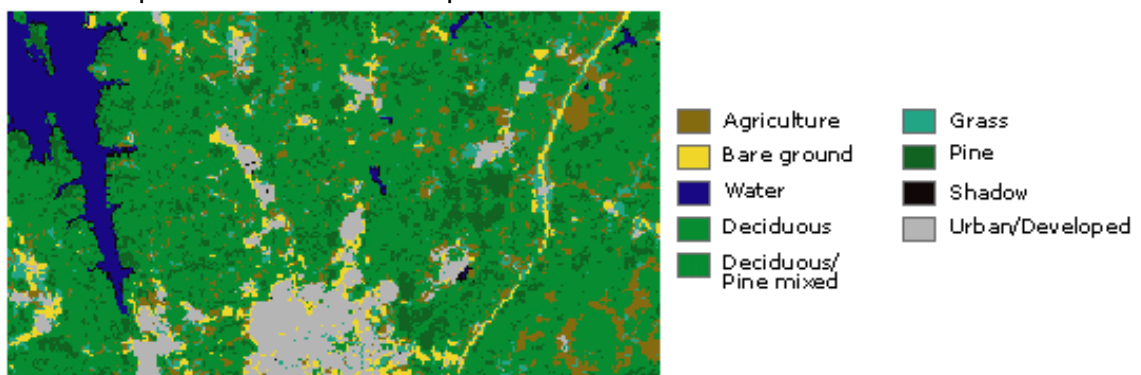


Figura 18: Capa raster utilitzada com a mapa temàtic. [36]

- Atributs d'una entitat

Aquest tipus de raster es poden tractar com fotografies digitals, documents o dibuixos relacionats amb un objecte o ubicació geogràfica. Poden ser moltes les utilitats d'aquesta categoria. Per exemple, unes fotografies de les entrades d'una cova associada a entitats de punts.

6.2. QGIS

Quantum GIS (QGIS) és un Sistema d'Informació Geogràfica de codi obert. Des del començament al maig de 2002, l'objectiu de QGIS ha estat competir amb els altres SIG però aquest fent-ho de forma gratuïta, per això al mateix 2002 es va unir a Source Forge [37], un repositori de projectes de codi obert.

Actualment, QGIS funciona en diferents sistemes operatius com Unix, Windows i Mac. Gràcies a la seva interfície amigable, és un dels SIG més coneguts i més utilitzats, arribant a competir amb altres SIG de pagament.

Aquest software, està desenvolupat utilitzant Qt [38] i C++ [39] i la seva llicència és pública GNU (GPL General Public License). Aquesta llicència permet revisar i modificar el codi font. Per tant, com a usuaris, és possible modificar o afegir noves funcions.

6.2.1 Característiques principals

El Sistema d'Informació Geogràfica QGIS és capaç de realitzar les funcionalitats bàsiques de qualsevol SIG. Ja sigui gràcies a les seves característiques integrades o a les aportades pels usuaris als plugins. En aquest apartat s'explicaran breument les característiques d'aquest programari de codi obert.

6.2.1.1. Visualització de dades

A QGIS es poden visualitzar els dos tipus de dades esmentats abans, dades vectorials i dades raster. Aquestes dades es poden visualitzar en format 2D o en format 3D utilitzant diferents tipus de projecció. QGIS a l'hora de visualitzar-les les converteix automàticament al format que haguem seleccionat com a sistema de projecció del visor.

Les dades vectorials les accepta en diferents formats, sent aquests els més utilitzats globalment. El mateix succeeix amb les dades raster, que accepta tots els formats dins la llibreria GDAL [40] (Geospatial Data Abstraction Library).

Per últim és capaç de tractar dades espacials en línia a través de serveis web OGC [41], com, per exemple, els que faciliten organismes cartogràfics oficials [42] o OpenStreetMap [43].

6.2.1.2. Explorar dades i crear mapes

Amb la interfície amigable de QGIS és senzill crear i explorar dades espacials. Les eines més útils disponibles a la interfície, de la qual parlarem en els pròxims apartats, són les següents:

- Navegador
- Reprojecció

- Gestor de base de dades
- Panell de vista general
- Anotacions
- Editar/Veure/Buscar atributs

6.2.1.3. Crear, editar, gestionar i exportar dades

QGIS ofereix moltes eines per poder crear, administrar, editar i exportar capes vectorials i rasters en diferents formats. Aquests són alguns exemples:

- Eines de digitalització per formats OGR [43] i capes vectorials GRASS [44]
- Permet crear i editar múltiples formats de fitxers
- Eines GPS [44] per importar i exportar formats GPX [46]
- Possibilitat de crear bases de dades a partir de fitxers
- Eines per poder gestionar les taules d'atributs vectorials
- Opció per guardar captures pantalla com imatges georeferenciades

6.2.1.4. Analitzar dades

La possibilitat d'analitzar dades és una altre de les funcionalitats que QGIS ofereix als seus usuaris. Les eines disponibles per fer-ho són d'anàlisi de vectors, mostreig, geoprocessament i gestió tant de geometria com de bases de dades. També es pot utilitzar qualsevol eina GRASS integrada que inclou totes les funcionalitats de GRASS. Per últim, es poden utilitzar molts plugins creats per tercers.

6.2.1.5. Plugins

Com ja s'ha comentat a apartats anteriors, la facilitat d'incorporar nous plugins creats per usuaris és una de les particularitats que fa que destaquí com a SIG. Actualment hi ha més de 1600 plugins disponibles.

La gran majoria dels plugins estan creats amb el llenguatge de programació de Python, i es poden trobar al repositori oficial de QGIS [47] on es pot veure documentació detallada sobre el plugin, la versió de QGIS necessària, autors i altre informació rellevant. La instal·lació d'aquests plugins es molt senzilla a través de la interfície.

Hi ha d'altres plugins però, que no es troben a la web oficial de QGIS i que estan autogestionats per els propis autors, aquests no són tan senzills d'instal·lar a QGIS però tampoc requereixen de grans complicacions. Aquest és el tipus de plugin que es realitza durant el projecte.

6.2.1.6. Publicar mapes a Internet

Com ja hem comentat QGIS és capaç d'utilitzar mapes a través d'Internet. A més a més també ofereix l'opció de publicar els mapes a Internet utilitzant un webserver amb UMN MapServer [48].

6.2.1.7. Consola de Python

QGIS proporciona una consola de Python integrada on poder fer scripts. Aquesta finestra s'obre com no-modal. Per poder interaccionar amb QGIS existeix la variable `qgis.utils iface` que és una instància de `QgisInterface`. Amb aquesta interfície es pot accedir als mapes, menús, eines... Per provar l'script únicament l'has d'escriure a la consola i s'executarà de forma automàtica.

6.2.2. Interfície

En aquest apartat s'expliquen les característiques principals de la interfície amigable de QGIS, la qual ja s'ha anat esmentant en apartats anteriors. En la següent figura estan senyalitzats els fonaments principals de la estructura bàsica de la interfície.

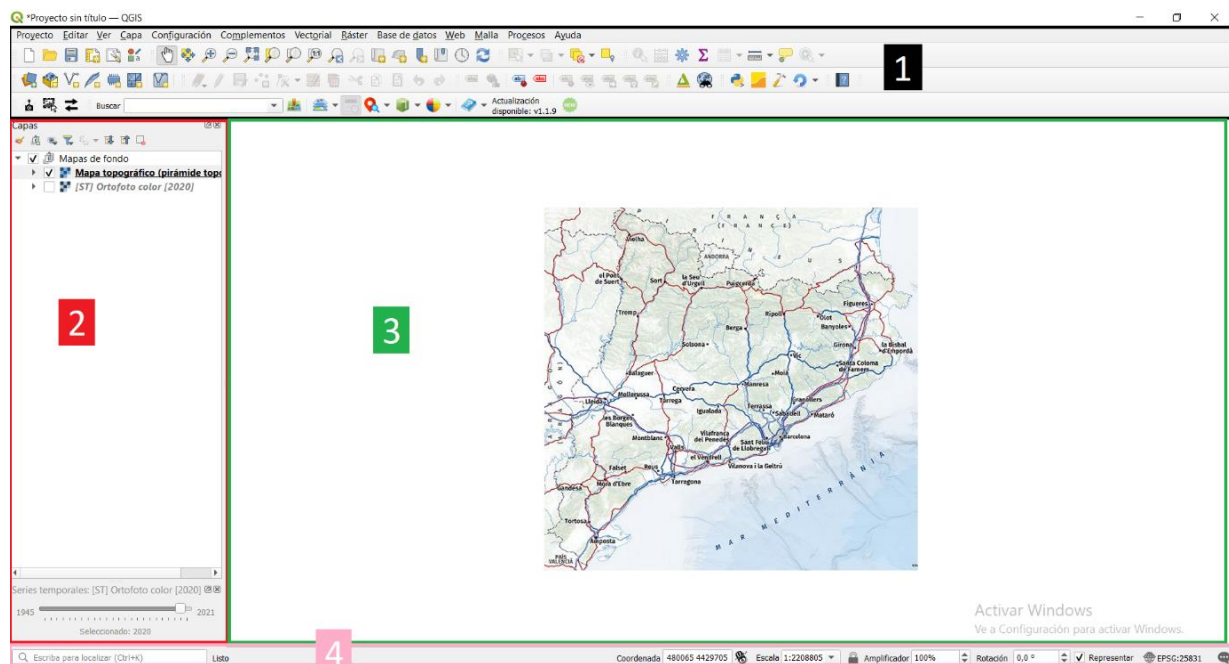


Figura 19: Interfície de QGIS. Elaboració pròpia.

La figura anterior es una captura de pantalla del que es veu quan s'inicia l'aplicació de QGIS després de carregar un mapa. Els elements identificats són els següents:

1. Barra d'eines
2. Llista de capes
3. Vista del mapa
4. Barra d'estat i localització

6.2.2.1. Barra d'eines

La barra d'eines a QGIS està dividida en dues parts, la barra d'eines de menú i la barra d'eines d'ajuda.

La barra d'eines de menú, que es pot veure a la figura 20, es representa mitjançant un menú jeràrquic estàndard com sol estar a la majoria d'aplicacions. Amb aquesta barra es pot realitzar qualsevol acció que QGIS permeti a la aplicació.

Figura 20: Interfície de QGIS. Barra d'eines de menú. Elaboració pròpia.

Per altre banda, la barra d'eines d'ajuda és més visual i representativa. Aquesta barra, que es pot veure a la figura 21 proporciona moltes de les funcions de la barra de menú i eines addicionals per interactuar amb el mapa.

Aquesta barra es personalitzable per a que es tinguin només les eines que s'utilitzen amb més freqüència. A més, cada element d'aquesta barra té una ajuda emergent disponible, per visualitzar la ajuda s'ha de mantenir el ratolí sobre de l'element.

Per últim, aquesta barra també inclou eines dels plugins que tinguem seleccionats. Per exemple, a la figura 21, els elements subratllats són plugins descarregats per l'usuari.



Figura 21: Interfície de QGIS. Barra d'eines d'ajuda. Elaboració pròpia.

6.2.2.2. Llista de capes

En aquesta part de la interfície es mostra una llista amb les capes disponibles. Al passar el ratolí per sobre de les capes es veu una informació bàsica com és el nom, la geometria, el sistema de referència de coordenades i la ruta de la ubicació al dispositiu. Fer clic dret sobre la capa fa que s'obri un nou menú amb opcions extra.

L'ordre de les capes a la llista significa que les capes col·locades a la part superior es dibuixen a la vista del mapa per sobre de les capes mostrades més avall a la llista.

6.2.2.3. Vista del mapa

L'objectiu de QGIS es poder representar el mapa, i això és el que es pot veure en aquesta secció de la interfície. Aquí es mostra les capes seleccionades a la llista de capes.

A més, a la vista del mapa es pot interactuar amb el mapa representat. Algunes de les accions que es poder fer són: ampliar, allunyar, desplaçar, seleccionar entitats...

6.2.2.4 Barra d'estat i localització

Aquesta barra es troba a la part inferior de la interfície i s'utilitza per dues funcionalitats diferents:

- Un localitzador que és capaç d'accedir a tots els objectes de QGIS
- Una barra d'estat que proporciona informació sobre el mapa actual, a més de ser capaç d'ajustar l'escala i la rotació del mapa i visualitzar les coordenades del cursor

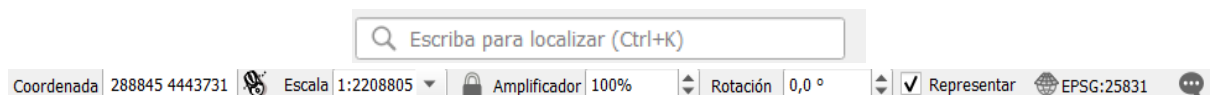


Figura 22: Interfície de QGIS. Barra d'estat i localitzador. Elaboració pròpia.

6.2.2.5. Vista 3D

La vista 3D és una altre funcionalitat que proporciona QGIS. Aquesta però, no s'inicialitza per defecte al iniciar QGIS. Per la realització del projecte és una vista important ja que es la que s'utilitza per visualitzar si el model 3D s'ha carregat correctament.

Per iniciar-la, es pot fer mitjançant la combinació de tecles Control+Alt+M o bé es pot clicar a "Ver" a la barra d'eines de menú i després clicar a "Nueva vista de mapa 3D". A la figura 23 es pot veure un exemple.

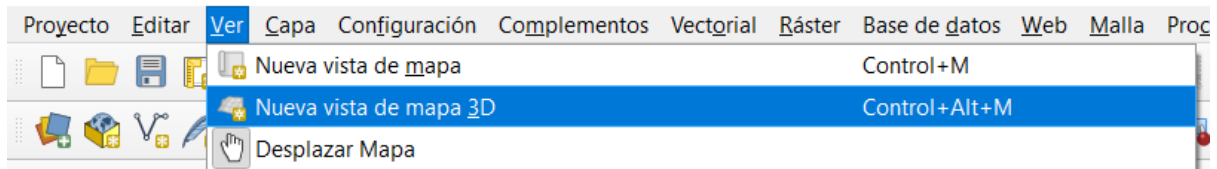


Figura 23 : Interfície de QGIS. Passos per obrir la vista 3D. Elaboració pròpia.

Al iniciar la vista 3D s'obre una nova finestra petita que no s'aconsegueix veure res. Aquesta finestra es pot arrossegar i posicionar-la amunt de la vista del mapa. Com es veu a la figura 24, es pot veure el mateix mapa però en format 3D. De igual manera que a la vista del mapa es pot interaccionar amb els elements.

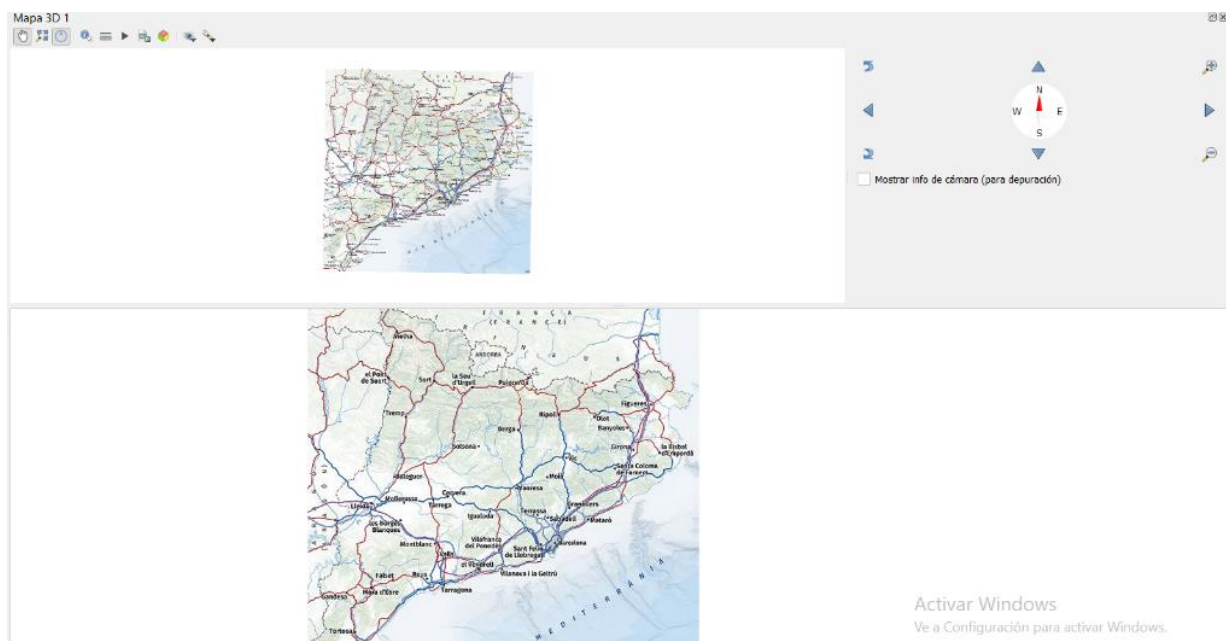


Figura 24: Interfície de QGIS. Vista 3D i vista del mapa. Elaboració pròpia.

6.2.2.6. Plugins

Els plugins, són una eina imprescindible a QGIS com ja s'ha esmentat abans. A la figura 21 s'ha pogut veure com apareixen els plugins a la interfície i com es poden utilitzar. En aquest apartat, un cop ja explicada la interfície, es pot veure com es poden instal·lar els plugins ja siguin validats oficialment o no.

6.2.2.6.1. Plugins validats oficialment

Aquest tipus de plugins, al haver estat validats són més senzills d'instal·lar. Per fer-ho, simplement s'ha de clicar a la part del menú on hi posa "Complementos" i fer clic a la primera opció "Administrar e instal·lar complementos". Això obrirà una nova pestanya on es poden buscar nous plugins, veure quins plugins es tenen instal·lats, actualitzar... A la figura 25 es pot veure la pestanya que s'obre.

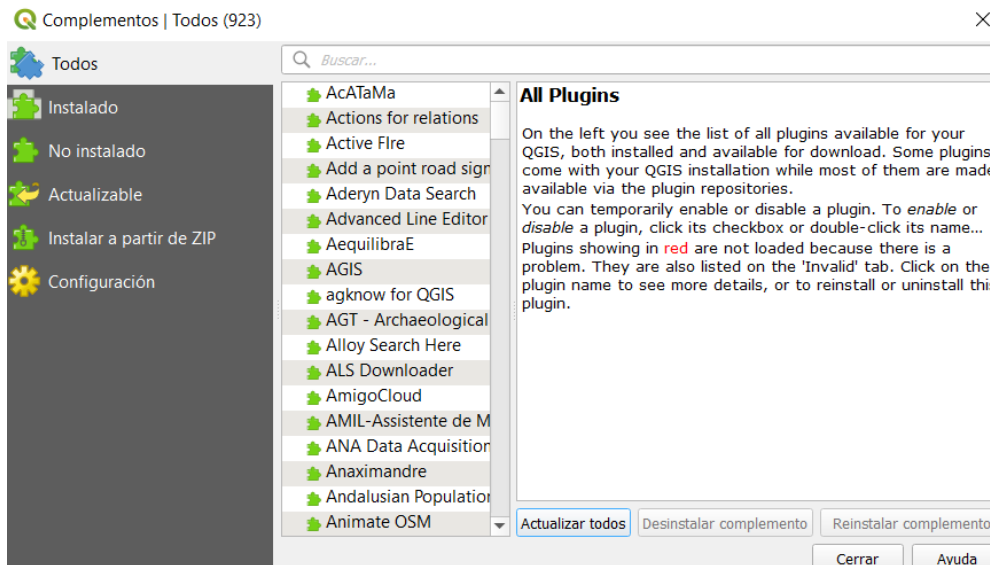


Figura 25: Interfície de QGIS. Vista de plugins. Elaboració pròpia.

6.2.2.6..2. Plugins no validats

Per altra banda, els plugins que no han estat validats oficialment requereixen de passos extrems per a la seva instal·lació. Aquest serà el mètode que es seguirà per instal·lar el plugin creat durant aquest projecte. Tot i que tots els plugins es poden instal·lar d'aquesta manera, si el plugin ja ha estat validat és més senzill fer-ho de la altre manera.

Primerament, per instal·lar manualment un plugin no validat requerim del seu codi font. Si es té el codi del plugin, el que s'ha de fer, com es pot veure a la figura 26, es clicar a la pestanya "configuració" del menú, desplegar un altre menú al clicar la primera opció "Perfiles de usuario" i clicar a la opció d' "Abrir carpeta del perfil activo". Això, farà que s'obri una nova carpeta de sistema ubicada al perfil actiu de QGIS.

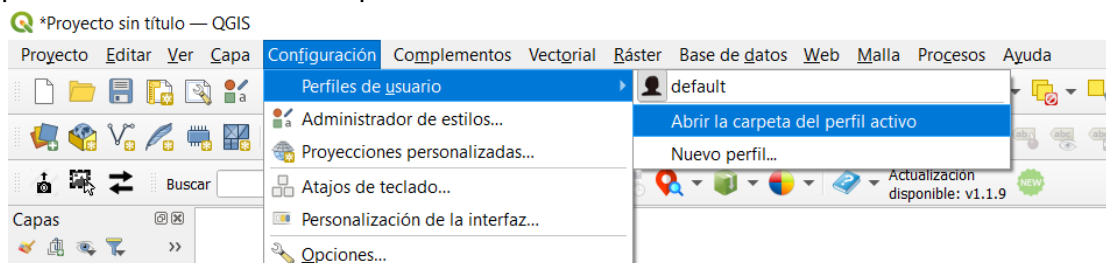


Figura 26: Interfície de QGIS. Passos per instal·lar un plugin. Elaboració pròpia.

Un cop la carpeta de fitxers s'hagi obert, s'ha de buscar dins la carpeta anomenada "python" i obrir-la. Un cop dins la carpeta de "python", s'ha de accedir a la següent carpeta amb nom "plugins". En aquesta carpeta "plugins" apareixen tots els plugins que l'usuari tingui instal·lats, i és aquí on s'haurà de copiar el codi font del plugin que es vulgui instal·lar.

7. Implementació tècnica

Aquest capítol conté l'explicació tècnica del treball. S'introdueix breument les tecnologies i eines emprades, i s'explica per separat com s'han desenvolupat cadascuna de les funcions que componen la totalitat del plugin.

7.1. Disseny

7.1.1 Plugin Builder

Abans de començar a fer un estudi del disseny, s'ha de saber com es poden crear nous plugins per a QGIS. Després d'una cerca a la web per veure les possibilitats que hi ha, s'observa que QGIS ofereix un plugin que serveix per crear nous plugins.

Aquest plugin anomenat Plugin Builder, ens proporciona dues opcions:

- Crear un plugin des de zero. Aquesta opció requereix crear la comunicació entre fitxers que componen el plugin i crear una interfície per interaccionar amb l'usuari mitjançant Qt.
- Crear un "processing plugin". Aquesta opció et genera els fitxers bàsics necessaris per a la comunicació entre fitxers i a més disposa de paràmetres que es poden afegir a la interfície sense haver de crear-los.

Els mateixos creadors de QGIS recomanen la utilització del segon mètode, ja que és més senzill d'incorporar. A més, en disposar de "widgets" ja creats, és més fàcil per als usuaris entendre del funcionament del nou plugin. Ja que el programador indica els paràmetres i el tipus, i el Processing genera la interfície.

Amb les dues opcions és possible de realitzar la finalitat d'aquest projecte, que és poder processar dades com les fotografies i els models 3D. Però, un dels requeriments principals del projecte, és la fàcil adaptació dels historiadors a aquesta nova eina. I les interfícies que genera el "processing plugin" són les mateixes que les de molts altres menús i diàlegs de QGIS, les quals ja estan acostumats a fer-les servir. Per tant, la segona opció serà l'escollida per crear el nou plugin.

Per a crear el plugin, Plugin Builder necessita d'alguns paràmetres bàsics com es pot veure a la figura 27. Aquests paràmetres són: el nom del plugin que es vol crear, el nom de la classe, una descripció del plugin, el nom del mòdul, el número de versió, la versió de QGIS necessària, en nom de l'autor i el seu correu.

Un cop omplert tots els camps necessaris, es clica a següent. S'obre una nova pestanya com la de la figura 28. Aquesta pestanya demana la informació bàsica de l'algorisme que utilitza el plugin. La informació requerida és: el seu nom, el grup al qual pertany, el proveïdor i una descripció.

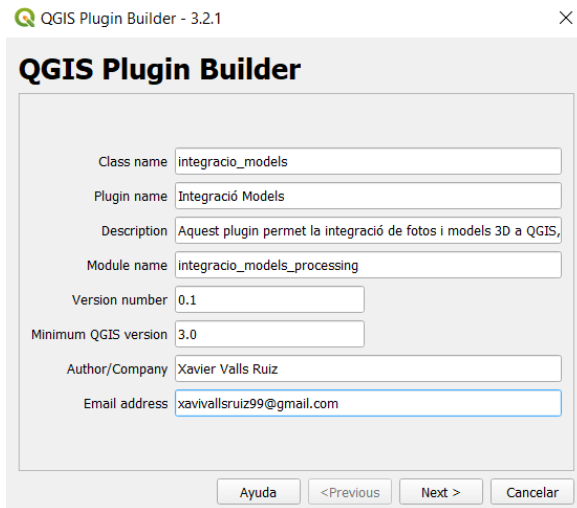


Figura 27: Interfície inicial de Plugin Builder.

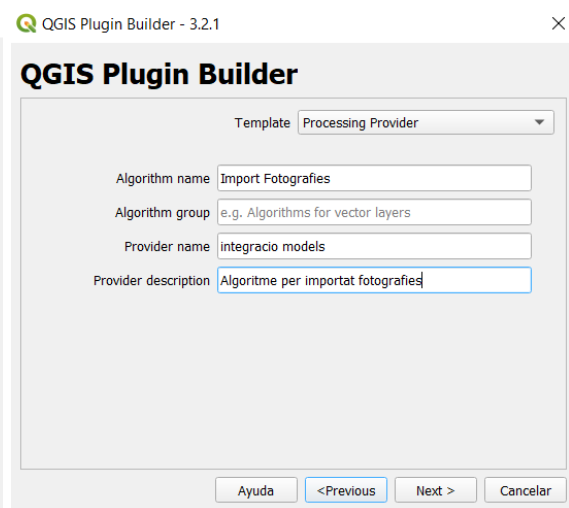


Figura 28: Interfície inicial de Plugin Builder.

Per acabar la creació del plugin, Plugin Builder et pregunta per alguns directoris. Els desenvolupadors de QGIS recomanen deixar les opcions per defecte i marcar l'opció que descriu que és un plugin experimental.

A la figura 29 es pot veure la carpeta que genera Plugin Builder. Aquesta inclou diversos fitxers entre els quals destaquen els fitxers python.

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
help	✘	06/10/2022 17:35	Carpeta de archivos	
i18n	✘	06/04/2022 17:00	Carpeta de archivos	
scripts	✘	06/04/2022 17:00	Carpeta de archivos	
test	✘	06/04/2022 17:00	Carpeta de archivos	
__init__	✘	06/10/2022 17:40	Archivo de origen ...	2 KB
integracio_models	✘	06/10/2022 17:35	Archivo de origen ...	3 KB
integracio_models_algorithm	✘	06/10/2022 17:35	Archivo de origen ...	7 KB
integracio_models_provider	✘	06/10/2022 17:35	Archivo de origen ...	4 KB
Makefile	✘	06/10/2022 17:35	Archivo	9 KB
metadata	✘	06/10/2022 17:35	Documento de tex...	2 KB
pb_tool	✘	06/10/2022 17:35	Archivo de origen ...	3 KB
plugin_upload	✘	06/04/2022 17:00	Archivo de origen ...	4 KB
pylintrc	✘	06/04/2022 17:00	Archivo	9 KB
README	✘	06/10/2022 17:35	Chrome HTML Do...	2 KB
README	✘	06/10/2022 17:35	Documento de tex...	1 KB

Figura 29: Carpeta creada per Plugin Builder.

El primer fitxer python que es troba a la carpeta és el `__init__.py`. Aquest fitxer només crida a una classe que es troba a `integració_models.py`.

A la figura 30 es pot veure el codi de `Integració_models.py`. Aquest fitxer té una classe `integració_modelsPlugin()` que conté diverses funcions per inicialitzar el plugin a QGIS.

Per inicialitzar el plugin crida a la classe `integració_modelsProvider()` ubicada al fitxer `integració_models_provider.py`. Aquesta classe és la que proveeix l'algoritme creat a la figura 28 amb Plugin Builder. Es pot veure a la figura 31 el codi d'`integració_models_provider.py`.

```
import os
import sys
import inspect

from qgis.core import QgsProcessingAlgorithm, QgsApplication
from .integració_models_provider import integració_modelsProvider

cmd_folder = os.path.split(inspect.getfile(inspect.currentframe()))[0]

if cmd_folder not in sys.path:
    sys.path.insert(0, cmd_folder)

class integració_modelsPlugin(object):

    def __init__(self):
        self.provider = None

    def initProcessing(self):
        """Init Processing provider for QGIS >= 3.8."""
        self.provider = integració_modelsProvider()
        QgsApplication.processingRegistry().addProvider(self.provider)

    def initGui(self):
        self.initProcessing()

    def unload(self):
        QgsApplication.processingRegistry().removeProvider(self.provider)
```

Figura 30: Codi del fitxer `integració_models.py`.

```
from qgis.core import QgsProcessingProvider
from .integració_models_algorithm import integració_modelsAlgorithm

class integració_modelsProvider(QgsProcessingProvider):

    def __init__(self):
        """
        Default constructor.
        """
        QgsProcessingProvider.__init__(self)

    def unload(self):
        """
        Unloads the provider. Any tear-down steps required by the provider
        should be implemented here.
        """
        pass

    def loadAlgorithms(self):
        """
        Loads all algorithms belonging to this provider.
        """
        self.addAlgorithm(integració_modelsAlgorithm())
        # add additional algorithms here
        # self.addAlgorithm(MyOtherAlgorithm())

    def id(self):
        """
        Returns the unique provider id, used for identifying the provider. The
        string should be a unique, short, character only string, eg "qgis" or
        "gdal". This string should not be localised.
        """
        return 'integració models'
```

Figura 31: Codi del fitxer `integració_models_provider.py`.

En últim lloc, Plugin Builder crea `integració_models_algorithm.py`, que es pot veure a la figura 32. Aquest fitxer té una classe `integració_modelsAlgorithm` on hi ha varies funcions.

La primera funció anomenada `initAlgorithm()` és l'encarregada de carregar els paràmetres necessaris per a l'algoritme. La segona funció `processAlgorithm()` és l'encarregada de tractar els paràmetres i fer que l'algoritme funcioni.

Inicialment, aquest fitxer ja disposa d'un algoritme com a exemple del que pot fer aquest tipus de plugins.

```

class integracio_modelsAlgorithm(QgsProcessingAlgorithm):
    OUTPUT = 'OUTPUT'
    INPUT = 'INPUT'

    def initAlgorithm(self, config):
        self.addParameter(
            QgsProcessingParameterFeatureSource(
                self.INPUT,
                self.tr('Input layer'),
                [QgsProcessing.TypeVectorAnyGeometry]
            )
        )
        self.addParameter(
            QgsProcessingParameterFeatureSink(
                self.OUTPUT,
                self.tr('Output layer')
            )
        )

    def processAlgorithm(self, parameters, context, feedback):
        source = self.parameterAsSource(parameters, self.INPUT, context)
        (sink, dest_id) = self.parameterAsSink(parameters, self.OUTPUT,
            context, source.fields(), source.wkbType(), source.sourceCrs())
        total = 100.0 / source.featureCount() if source.featureCount() else 0
        features = source.getFeatures()

        for current, feature in enumerate(features):
            if feedback.isCanceled():
                break
            sink.addFeature(feature, QgsFeatureSink.FastInsert)

            # Update the progress bar
            feedback.setProgress(int(current * total))
        return {self.OUTPUT: dest_id}

```

Figura 32: Codi del fitxer `integració_models_algorithm.py`.

7.1.2. Disseny

Un cop els fitxers han estat generats per Plugin Builder és hora de modificar-los per tal d'aconseguir un bon disseny per realitzar les necessitats del projecte.

Aquest projecte es divideix en tres funcionalitats a implementar, les quals són:

- Importar les fotografies
- Importar els models 3D
- Exportar-los a la web

Gràcies a Plugin Builder es pot tenir diferents algoritmes que utilitzen diferents paràmetres. Per tant, seria ideal poder fer tres algoritmes diferents capaços de fer les diferents funcionalitats.

Per fer-ho d'aquesta manera, el fitxer `integració_modelsProvider` ha de ser capaç de proveir al plugin de més d'un algoritme. Això es pot assolir creant dos nous fitxers d'algoritme i fer que el proveïdor els afegixi al plugin com a la figura 31.

Per a simplificar els noms, al fitxer d'algoritme creat per Plugin Builder, se li canvia el nom a `integracio_models_algorithm_import_fotos`. Aquest algoritme s'encarrega d'importar les fotografies a QGIS.

Per altra banda, els dos nous fitxers d'algoritme, s'anomenen `integracio_models_algorithm_import_models` i `integracio_models_algorithm_exportar`, són els encarregats d'importar els models 3D i d'exportar-los a la web respectivament.

7.1.3. Interfície

Pel que respecte a la interfície, el plugin creat amb Plugin Builder ja s'encarrega de mostrar els algorismes a la "toolbox" de la interfície de QGIS.

A la figura 33, es pot veure el nou plugin a la interfície de QGIS, que consta dels tres algorismes descrits recentment.

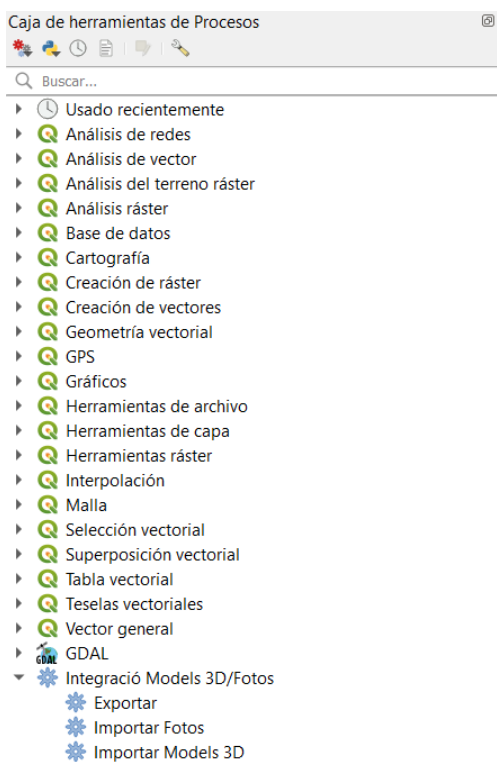


Figura 33: Interfície de la "toolbox" a QGIS.

Per altra banda, la interfície de cada algoritme depèn dels paràmetres utilitzats en l'algoritme, però, aquests paràmetres sempre s'estructuren de la mateixa manera. A la figura 34 es pot veure un exemple d'una interfície d'algoritme. Aquesta interfície té una selecció de camp per triar la capa d'entrada i un camp per triar on desar el resultat.

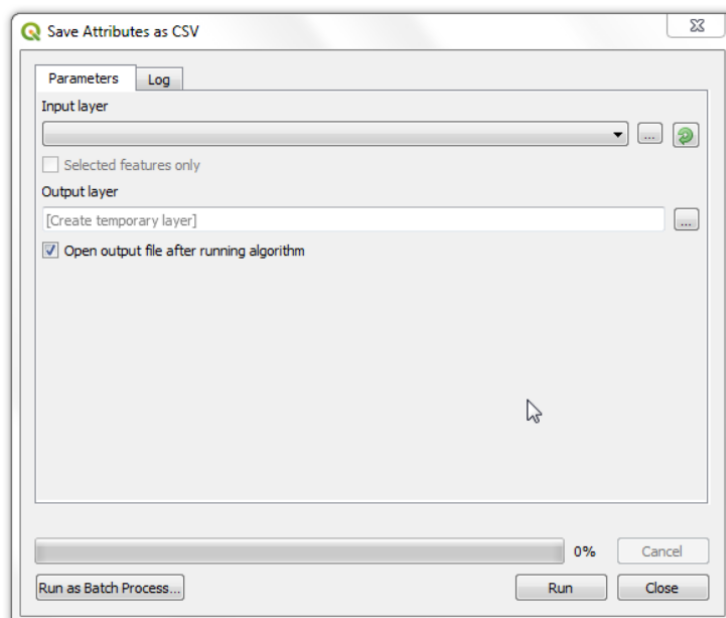


Figura 34: Exemple de interfície d'un algoritme.

7.2. Importar fotografies

La funció per importar fotografies a QGIS és la primera que es crea en aquest projecte. El primer a pensar són els paràmetres que necessita l'algoritme per poder importar les fotografies.

El primer paràmetre que necessita són les mateixes fotografies que es vol importar. QGIS Processing ofereix la possibilitat d'incorporar la selecció d'elements individuals a qualsevol carpeta o la possibilitat de seleccionar una carpeta sencera. Es tria la segona opció per tal de poder seleccionar més d'una fotografia alhora. Aquest paràmetre es pot configurar per només permetre seleccionar els arxius .jpg de la carpeta.

El segon paràmetre necessari, ja que hi ha fotografies sense ubicació GPS, serà la ubicació al mapa de la fotografia. QGIS Processing ofereix un paràmetre per seleccionar un punt a un mapa.

L'últim paràmetre que necessita aquest algoritme és la carpeta de destí on es crea l'arxiu de la capa vectorial que s'ha de crear amb les fotografies.

A la figura 35 es mostra com es veu la interfície de l'algoritme per importar fotografies a QGIS.

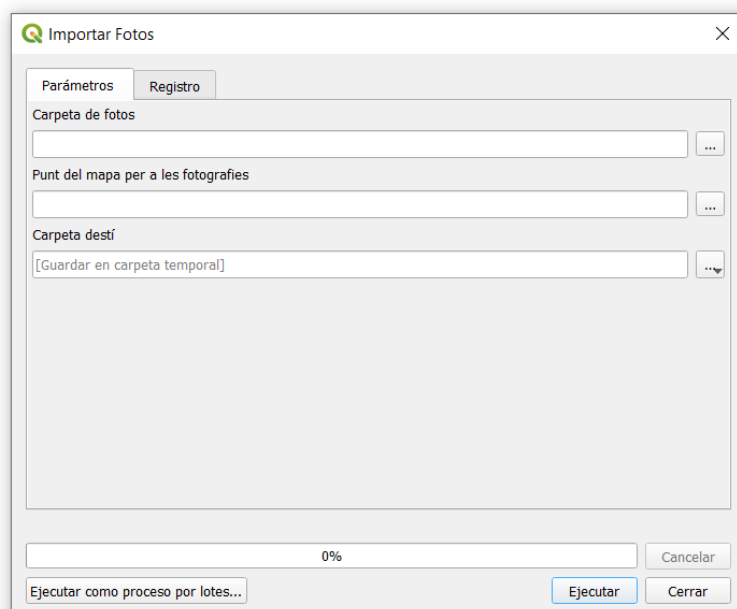


Figura 35: Interfície del algoritme Importar Fotografies.

Un cop els paràmetres ja han estat programats a la funció `initAlgorithm()`, és hora de programar la funció `procesAlgorithm()`, la funció encarregada d'importar les fotografies.

Primerament, s'han de tractar els paràmetres i ubicar-los en variables de la funció correctament inicialitzats. El punt de les fotografies és el paràmetre que més tractament necessita per a convertir-se correctament a un string.

Un cop ja estan els paràmetres tractats, es crea una llista dels paths de les fotografies que es vol importar. També es crea una capa vectorial on s'afegiran les fotografies.

Tot seguit es recorre la llista de fotografies per afegir-les a la capa vectorial. Aquesta tasca es pot trobar amb dos tipus de fotografies: les georeferenciades i les no georeferenciades.

Per saber si una fotografia està georeferenciada o no, es crida a una funció que accedeix a les propietats de la fotografia. Aquesta funció, que es pot veure a la figura 36, retorna un parell d'elements:

- Un booleà que indica si la fotografia té georeferenciació o no
- Un diccionari amb tres elements:
 1. Tipus: `Feature`. Serveix per identificar que el diccionari és una característica.
 2. Geometria: un diccionari que indica que és un punt i les seves coordenades.
 3. Propietats: un diccionari amb un id per identificar el punt, el nom de la fotografia, les coordenades de latitud i de longitud, el path i un `imagesrc`.

Aquestes propietats i geometria s'extrauen de les propietats de la fotografia.

```

def get_geo_infos_from_photo(self, img_path, count):
    imagesSrc = '<img src = "' + img_path + '" width="300" height="225"/>'
    with open(img_path, 'rb') as src:
        img = Image(src)
    if img.has_exif:
        lat = img.gps_latitude[0] + img.gps_latitude[1] / 60 + img.gps_latitude[2] / 3600
        lon = img.gps_longitude[0] + img.gps_longitude[1] / 60 + img.gps_longitude[2] / 3600
        if img.gps_longitude_ref == 'W':
            lon = -lon
        if img.gps_latitude_ref == 'S':
            lat = -lat
        geo_info = {"type": "Feature",
                    "geometry": {"type": "Point", "coordinates": [lon, lat]},
                    "properties": {'id': count, 'Name': os.path.basename(img_path),
                                   'Lon': lon, 'Lat': lat, 'Path': img_path, 'Images': imagesSrc}}
        return (True, geo_info)
    else:
        return (False, 0)

```

Figura 36: Codi de la funció `get_geo_infos_from_photo()`.

Un cop ja s'ha cridat a la funció per saber si la fotografia està georeferenciada, als apartats 7.2.1. i 7.2.2. s'explica com afegir les fotografies a la capa vectorial depenent si tenen geoinformació o no.

Finalment, quan les fotografies ja han estat afegides a la capa vectorial, aquesta capa s'escriu en un fitxer "ESRI shapefile" ubicat a la carpeta destí especificada als paràmetres.

Per acabar, es carrega aquesta capa a la llista de capes de la interfície de l'usuari. A la figura 37 es mostra un exemple de com es veu aquesta capa i els seus punts al mapa després d'executar l'algoritme.

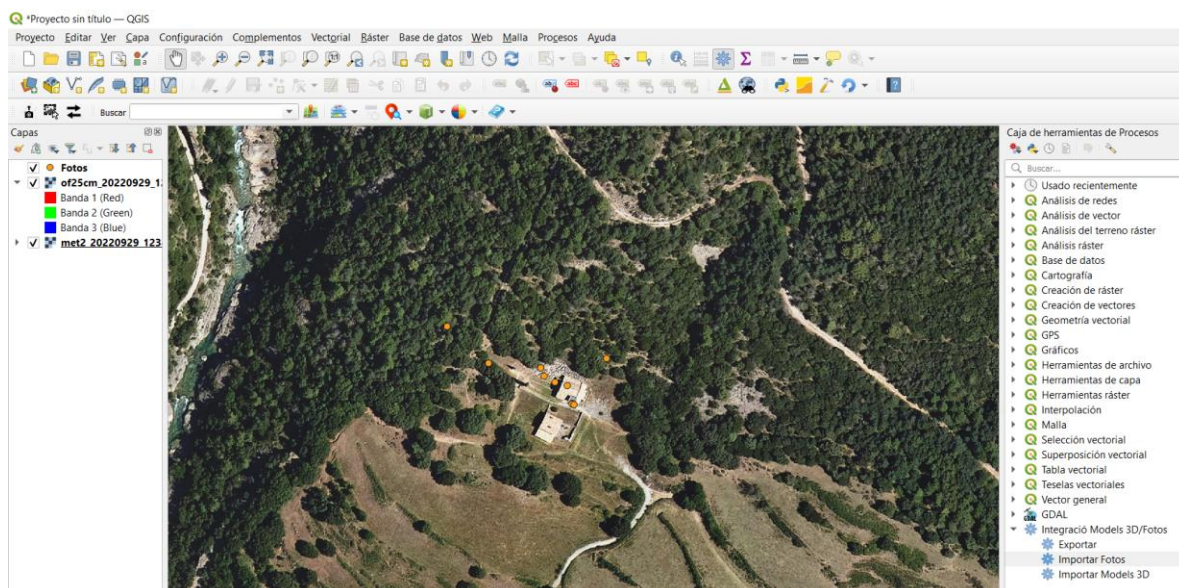


Figura 37: Interfície de QGIS després d'executar l'algoritme `Importar Fotos` i carregar una capa raster de base.

7.2.1. Fotografies georeferenciades

Un cop ja s'ha cridat a la funció per saber si la fotografia està georeferenciada, si ho està, s'aboca el diccionari retornat per la funció a una altra funció json que el transforma en un text format json. Amb aquest json, es crida a una funció de QGIS per convertir aquest json en una variable amb els camps ("fields") per a la nova capa vectorial.

El següent pas és comprovar si a la capa vectorial ja es té assignats els camps o si encara no perquè és la primera fotografia. Si no s'havien assignat els camps se'ls assignen.

Tot seguit, de la mateixa manera que amb els camps, a través d'una funció de QGIS, es crea una variable amb les característiques dels camps de la fotografia. Aquestes característiques s'afegeixen a la capa vectorial i aquesta s'actualitza.

A la següent figura es pot veure el codi de l'algoritme quan la fotografia tractada no té geoinformació.

```
geo_info[0] == True:
    geo_info2 = json.dumps(geo_info[1])
    fields = QgsJsonUtils.stringToFields(geo_info2)
    if not fields_ready:
        pr.addAttributes(fields.toList())
        v1.updateFields()
        fields_ready = True
    feature = QgsJsonUtils.stringToFeatureList(geo_info2, fields)[0]
    pr.addFeature(feature)
    v1.updateExtents()
```

Figura 38: Part del codi d'Importar fotografies

7.2.2. Fotografies no georeferenciades

Per altra banda, si la fotografia no disposa de geoinformació, com que la funció per a comprovar-ho només retorna un False i un 0, si encara no han estat creats els camps de la capa vectorial, es creen d'un en un i s'afegeixen.

Un cop els camps ja han estat creats, es comprova que les coordenades introduïdes com a paràmetre estan representades amb la mateixa projecció i s'afegeix a través de funcions de QGIS la geometria i característiques de la fotografia.

Els camps i les característiques introduïdes són les mateixes que a les fotografies georeferenciades.

```
#si la foto no te geoinformacio
else:
    if not fields_ready:
        pr.addAttributes([QgsField('id', QVariant.Int),QgsField('Name', QVariant.String),
            QgsField('Lon', QVariant.String), QgsField('Lat', QVariant.String),
            QgsField('Path', QVariant.String), QgsField('Imagesrc', QVariant.String)])
        v1.updateFields()
        fields_ready = True
    imagesrc = '<img src = "" + photo_path + "" width="300" height="225"/>'
    inProj = Proj(espj)
    outProj = Proj('epsg:4326')
    x2,y2 = transform(inProj,outProj, x_point,y_point)
    feat = QgsFeature()
    feat.setGeometry(QgsGeometry.fromPointXY(QgsPointXY(y2, x2)))
    feat.setAttributes([count, os.path.basename(photo_path), str(y2), str(x2), photo_path, imagesrc])
    pr.addFeature(feat)
    v1.updateExtents()
```

Figura 39: Part del codi d'Importar fotografies.

7.2.3. Dificultats

En aquest apartat es mostren algunes de les dificultats que s'han trobat en la creació de l'algoritme d'importar fotografies.

Les primeres dificultats a aparèixer van ser les d'escriure el fitxer shapefile de la capa vectorial i la de fer-lo aparèixer a la llista de capes. No hi havia manera de realitzar les dues. O bé s'escrivía el fitxer o bé apareixia a la llista de capes. A més, la informació trobada estava desactualitzada i, per tant, s'han estat fent coses que ja no era possible de fer per la versió.

Un altre error que es va cometre durant el desenvolupament de l'algoritme, és el d'intercanviar la longitud i latitud. A causa d'aquest error, tot i que la part tècnica estava bé, no es podia visualitzar el resultat a QGIS. Es va buscar una solució a la web i va trobar la possibilitat d'afegir els camps i característiques a través de la funció de QGIS de json.

Com l'error seguia, ja que era degut a la latitud i longitud, gràcies a les reunions amb el director es va poder solucionar i per això les fotografies georeferenciades s'afegeixen d'una forma i les no georeferenciades d'una altra.

7.3. Importar Models 3D

L'algoritme per importar models 3D a QGIS és el segon que es crea en aquest projecte. Com a l'algoritme anterior, el primer a pensar són els paràmetres que es volen utilitzar.

El primer paràmetre necessari són els models 3D, com a l'algoritme anterior es tria l'opció d'usar el paràmetre de seleccionar una carpeta per poder seleccionar més d'un model 3D alhora.

Un altre paràmetre que es necessita, de la mateixa manera que per importar les fotografies és la ubicació al mapa del model 3D.

L'últim paràmetre necessari per a aquest algoritme és la carpeta destí de la capa vectorial que s'ha de crear per cada model 3D.

A la figura 40 es mostra com es veu la interfície de l'algoritme per importar models 3D a QGIS.

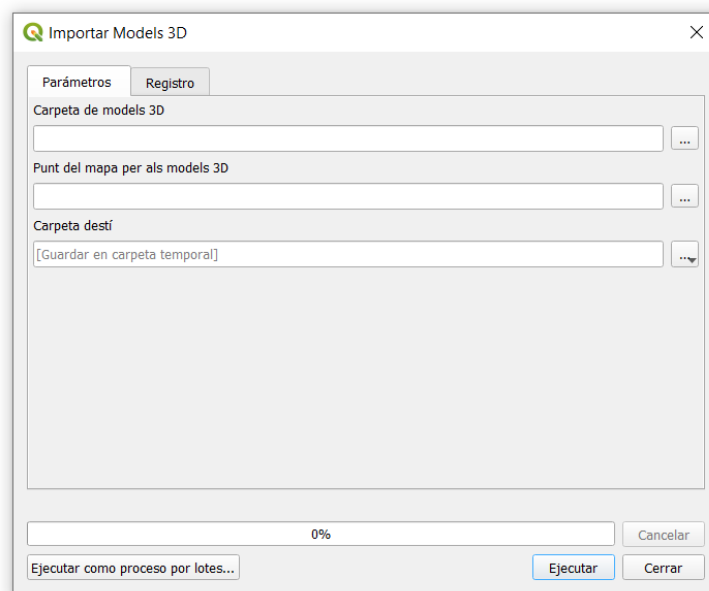


Figura 40: Interficie del algoritme Importar Modelos 3D.

Un cop els paràmetres ja han estat establerts a la funció `initAlgorithm()`, és hora de programar la funció `procesAlgorithm()`, l'encarregada d'importar els models 3D.

A l'inici de la funció `procesAlgorithm()` s'ha de fer el mateix que a la de l'altre algoritme. És a dir, primerament s'han de tractar els paràmetres i ubicar-los en variables de la funció correctament inicialitzats. Un cop ja estan els paràmetres tractats, s'ha de crear una llista dels paths dels models 3D i de les textures que es vol importar.

Tot seguit, per cada model 3D que es vol importar a QGIS es crea una capa vectorial diferent per tal de poder activar o desactivar de la llista de capes els models que es vulgui en el moment. A aquesta capa vectorial, se li assigna els camps: id, path del model 3D i path de la textura. Als camps se li dona valor a través de les característiques del model 3D. Un cop la capa ja té assignats les característiques s'escriu el fitxer "shapefile" a la carpeta destí indicada.

El següent pas és la creació d'un material per a la vista 3D del model 3D, ja que aquesta vista no permet afegir la textura del model. També es crea un símbol, al qual se li assigna que té la forma indicada al path del model 3D. Aquest pas va donar molts problemes, ja que a QGIS es veia que s'afegia el path, però no es veia l'objecte a la vista 3D. Al final, amb l'ajuda del director del projecte vam descobrir que era necessari especificar al diccionari: `overwriteMaterial = True`. Altrament, no se li assignava un material correcte a l'objecte i no es renderitzava al visor 3D

```
symbol.setShapeProperties({'model':objecte[0], 'overwriteMaterial': True})
```

Finalment, s'assigna aquest símbol al renderitzat 3D de la vista 3D. I per tal que aparegui aquest símbol a la vista 3D al punt indicat s'assigna aquest renderitzat a la capa vectorial que defineix el model 3D.

A la figura 41 es mostra un exemple de com es veuen les capes creades després d'executar l'algoritme. A la part inferior es veu el punt on es troba el model 3D i a la part superior es veu el model 3D a la vista 3D.

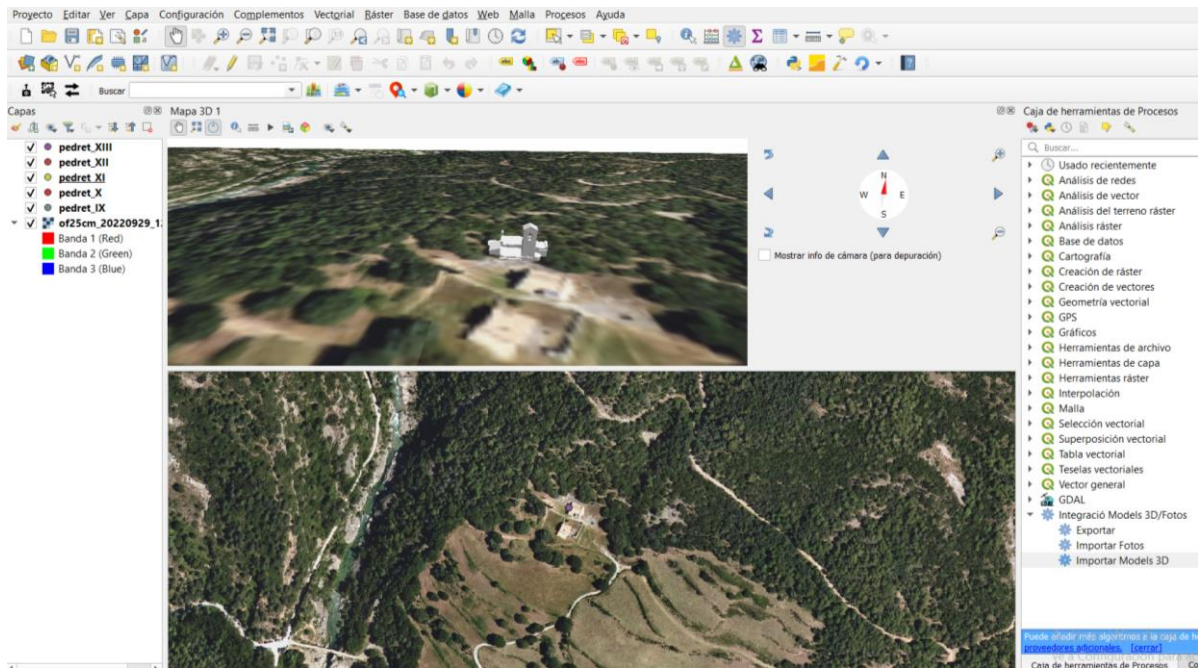


Figura 41: Interfície de QGIS després d'executar l'algoritme Importar Modelos 3D i carregar una capa raster de base.

7.4. Exportar a la web

L'últim algoritme és el que s'utilitza per exportar a un visualitzador web les fotografies i models 3D importats anteriorment, juntament amb els mapes. Com als algoritmes anteriors, els paràmetres necessaris són la primera cosa a pensar.

Els primers paràmetres necessaris són les capes amb els models 3D i la capa amb les fotografies. A diferència dels algoritmes anteriors, aquests paràmetres no s'afegeixen a la funció `initAlgorithm()`. Com pot haver-hi diferent quantitat de capes que es vol exportar a cada execució de l'algoritme, s'exportaran aquelles capes que, a la llista de capes del QGIS (veure Figura 19 requadre 2) estiguin marcades en aquest moment com a visibles. Això simplifica la interfície del plugin, ja que no cal replicar un selector de capes quan QGIS ja en té un, i resulta natural ja que només s'exporta allò que l'usuari té visible actualment

Per altra banda, la capa DEM i la capa raster que indica la textura del terreny sempre son necessàries per a poder exportar, ja que volem mostrar al visor web el model del terreny texturat. Per tant, aquestes dues capes s'afegeixen com a paràmetres, ja que QGIS permet l'opció d'utilitzar com a paràmetres les capes raster. A més serà necessari un paràmetre numèric addicional que faci referència a l'escala de la capa DEM.

Un altre paràmetre necessari és l'àrea que es vol exportar, perquè no es pot exportar tot Catalunya ja que seria molt costós i trigaria molt. Juntament amb aquesta àrea, un punt central del mapa anirà bé més endavant.

Per últim, serà necessària una carpeta de destí per a poder construir tots els fitxers que requereix la visualització web. A més en aquest cas, es crearà una nova carpeta a la carpeta destí per tal d'organitzar el arxius. El nom de la carpeta serà un paràmetre de la funció.

A la figura 42 es mostra com es veu la interfície de l'algoritme per exportar els models 3D i les fotografies a la web.

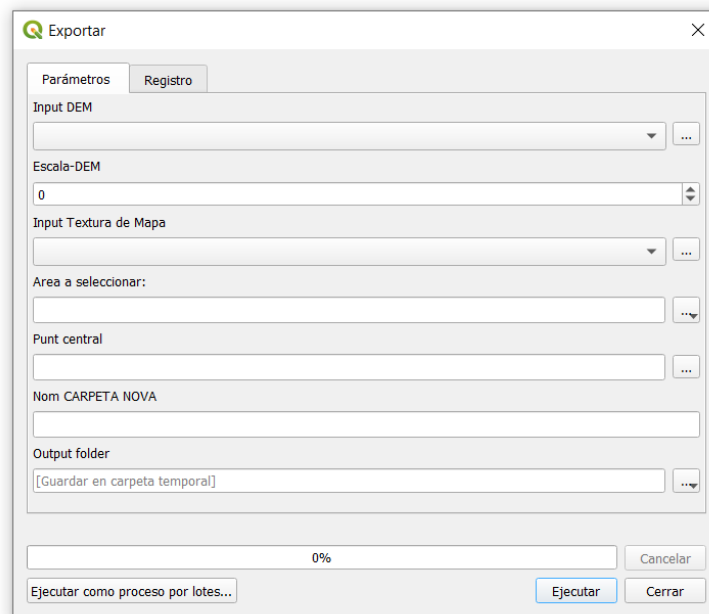


Figura 42: Interfície de QGIS després d'executar l'algoritme Importar Models 3D i carregar una capa raster de base.

Un cop els paràmetres ja han estat establerts a la funció `initAlgorithm()`, és hora de programar la funció `procesAlgorithm()` que és la funció encarregada d'exportar les fotografies i models 3D.

A l'inici de la funció `procesAlgorithm()` s'ha de fer el mateix que a la dels altres algoritmes. És a dir, primerament s'han de tractar els paràmetres i ubicar-los en variables de la funció correctament inicialitzades. S'ha de crear strings per a tots el paths necessaris de l'algoritme, com els de les carpetes noves i fitxers nous. També s'ha de tractar el paràmetre de l'àrea i del punt per ubicar-los correctament a dues llistes amb coordenades x, y i z.

Just després del tractament de paràmetres es crea la nova carpeta i es copien alguns fitxers i carpetes necessaris per a la visualització.

Tot seguit, es retallen les capes DEM i de textura del mapa. Això s'aconsegueix a través de la funció `gdal.translate()`. Aquesta, rep per paràmetres els fitxers d'entrada i sortida, unes coordenades les quals indiquen les puntes de l'àrea i el tipus de format de sortida. La nova capa DEM retallada es guarda en el tipus d'arxiu ASC, i la capa de textura del mapa es guarda en format PNG.

Un cop ja es disposa de la capa DEM retallada en format ASC s'ha de convertir en un tipus de fitxer 3D. L'opció triada per fer-ho i que es pugui visualitzar a la web és convertir-ho a format PLY.

Per convertir el fitxer ASC a PLY es crida a la funció `asc2ply`. Aquesta funció s'ha programat per tal de generar els triangles necessaris per crear un objecte 3D utilitzant les altures de la capa DEM. A la figura 43 es pot veure el codi de `asc2ply()`.

```
def asc2ply(ascFile, outFile, scale):
    H, nodata, resolution, points, pmin = readAscFile(ascFile)
    nrows, ncols = points

    fply = open(outFile if outFile.endswith('.ply') else outFile + '.ply', 'wb')
    fply.write(bytes('ply\n', encoding='ascii'))
    fply.write(bytes('format binary_little_endian 1.0\n', encoding='ascii'))
    fply.write(bytes('element vertex %d\n' % (nrows*ncols), encoding='ascii'))
    fply.write(bytes('property float x\n', encoding='ascii'))
    fply.write(bytes('property float y\n', encoding='ascii'))
    fply.write(bytes('property float z\n', encoding='ascii'))
    fply.write(bytes('element face %d\n' % ((nrows-1)*(ncols-1)*2), encoding='ascii'))
    fply.write(bytes('property list uint8 int32 vertex_index\n', encoding='ascii'))
    fply.write(bytes('end_header\n', encoding='ascii'))
    for i in range(nrows):
        for j in range(ncols):
            fply.write(struct.pack('<fff', pmin[0] + j*resolution, pmin[1] + (nrows-1-i)*resolution, H[i][j]*scale))

    for i in range(nrows-1):
        for j in range(ncols-1):
            fply.write(struct.pack('<Biii', 3, i*ncols + j, (i+1)*ncols + j, i*ncols + j + 1))
            fply.write(struct.pack('<Biii', 3, (i+1)*ncols + j, (i+1)*ncols + j + 1, i*ncols + j + 1))
    fply.close()
```

Figura 43: Codi de la funció `asc2ply()`.

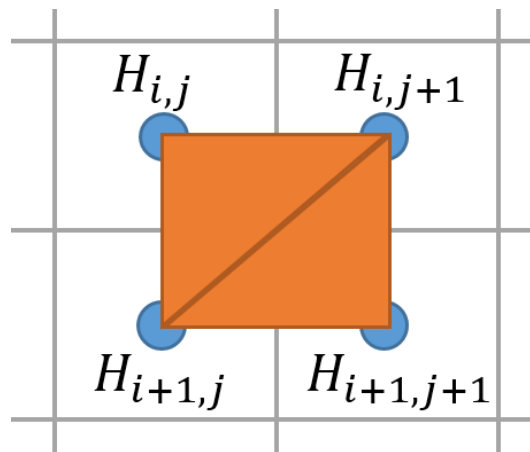


Figura 44: Explicació dels triangles de l'objecte a partir de les cel·les.

Aquesta funció primer crea el fitxer de sortida ply, com a fitxer binari, i escriu la capçalera del format PLY, el nombre de vèrtexs (el nombre total de cel·les del DEM), els camps dels vèrtexs del model (les coordenades X,Y,Z com a float), el nombre de cares del model (cada bloc de 2x2 cel·les crearà dos triangles), i el format d'una cara donat com a llista d'enters.

Tot seguit, generem un vèrtex per cada valor d'alçada del DEM. Les coordenades X,Y es poden calcular automàticament sabent on és la cantonada superior esquerra del DEM i la resolució d'aquest (distància en metres entre dos valors). Cal tenir en compte, però, que les

files van de nord a sud, així que invertim l'ordre del recorregut per orientar-les amb l'eix Y cartesià que utilitzarà el visor 3D.

Finalment, escrivim la llista de cares del model. Una cara és un triangle, i s'expressa com tres enters. Com hem escrit els vèrtexs iterant files i columnes del DEM, és trivial calcular l'identificador o índex lineal de cada vèrtex. Per una correcta orientació dels triangles, cal tenir en compte que els tres vèrtexs vinguin donats en sentit antihorari.

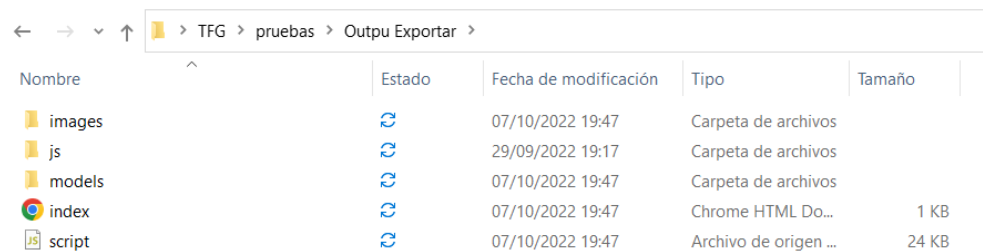
El següent pas per a realitzar l'exportació a la web és visualitzar quines capes hi ha actives per saber que s'ha d'exportar.

Pel que respecte a les fotografies, per cada una d'elles es guarda la ubicació on es troba amb la seva altura i el nom. A més, per limitacions de la web, es copien totes les fotografies a la carpeta creada.

Pel que respecte als models 3D, de la mateixa manera que a les fotografies es copia el model 3D i la textura a la nova carpeta. A més es guarda el seu nom i el nom de la textura.

Per acabar, es crida a la funció `creaScript()`, la qual requereix paràmetres com el fitxer DEM PLY, el fitxer raster de la textura del mapa, el punt central del mapa i les imatges i els models que s'acaba de guardar. Aquesta funció genera l'script necessari per poder visualitzar l'escena a la web.

Aquest algoritme crea una carpeta com la que es pot veure a la figura 45.



The screenshot shows a file explorer window with the path 'TFG > pruebas > Output Exportar'. The contents are listed in a table with columns for Nombre, Estado, Fecha de modificación, Tipo, and Tamaño.

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
images	🔄	07/10/2022 19:47	Carpeta de archivos	
js	🔄	29/09/2022 19:17	Carpeta de archivos	
models	🔄	07/10/2022 19:47	Carpeta de archivos	
index	🔄	07/10/2022 19:47	Chrome HTML Do...	1 KB
script	🔄	07/10/2022 19:47	Archivo de origen ...	24 KB

Figura 45: Carpeta generada per l'algoritme d'Exportar.

A la carpeta "images" es troben totes les fotografies que s'han exportat. A la carpeta "js", hi ha tot de fitxers necessaris per a poder visualitzar-ho a la web. A la carpeta "models" hi ha els models 3D, textures d'aquests i les capes DEM i de textura de mapa retallades. El fitxer index és de format html per visualitzar a la web. L'últim fitxer "script", és on està definida l'escena de la visualització web. Al següent apartat es veu com està implementat el fitxer "script".

7.4.1. Fitxer “script”

Aquest fitxer és el cos del fitxer html, per tant, és el fitxer que crea l'escena que es visualitza a la web.

El primer que es troba a aquest fitxer són les declaracions globals de variables bàsiques com l'escena, la càmera, les llums, el renderitzador i els objectes.

El següent element visible al codi és la variable de capes, la qual està formada per una capa per al mapa, una per a les fotografies i una per a cada model 3D. Aquestes capes serveixen per visualitzar a la web el contingut de la capa seleccionada.

Les dues següents variables globals del codi són “movimentmodels” i “movimentllum”. La variable “movimentmodels” està formada per 4 floats, un representant l'angle de rotació del model i els altres la translació en els eixos x, y i z. La variable “movimentllum” està formada per un azimuth i una altura. Aquestes variables es fan servir per moure els models i la llum per l'escena, com indica el seu nom.

La següent part de l'script és una funció que serveix per inicialitzar l'escena i els seus elements. Aquesta funció comença amb la creació d'una escena i posant-li el color gris clar de fons. Tot seguit, es crea la càmera i s'ubica a la posició (0, 50, 50). També es crea la llum, que consta d'una llum d'ambient i una llum direccional blanca a la posició (-8e4, 1e5, 4e4). Addicionalment, s'afegeixen els eixos de coordenades per ajudar a la visualització.

Després de crear els elements bàsics d'una escena, a la mateixa funció d'inicialització es creen els objectes dels models 3D. Aquest procés es repeteix per cada model 3D. La creació dels objectes comença amb la creació d'un material que carrega la textura del model 3D, tot seguit es crea un objecte al qual se li assigna aquest material i la forma del model 3D.

Un cop creat l'objecte amb el seu material, s'afegeix aquest objecte a la variable global d'objectes, es calcula el seu centre i s'ubica el model a la posició (0, 0, 0), s'utilitza una funció per configurar el material perquè es vegi correctament, se li assigna una capa de l'escena i finalment s'afegeix l'objecte a l'escena.

Encara a la funció d'inicialització, es crea l'objecte que representa el voltant dels models 3D. Aquest objecte és el fitxer PLY creat anteriorment. La creació d'aquest objecte, comença amb la creació del material al qual es carrega la capa raster que representa la textura. Tot seguit es crea l'objecte amb el fitxer PLY i se li assigna la textura.

Però, la textura assignada a l'objecte PLY no té coordenades de textura i s'ha de crear unes. A la figura 46 es veu la funció per a crear les coordenades.

```

var box = new THREE.Box3().setFromObject( mesh );
var uvs = [];
var posAttribute = geometry.getAttribute('position');
for ( var i = 0; i < posAttribute.count; i ++ ) {
    var x = posAttribute.getX(i);
    var y = posAttribute.getY(i);
    uvs.push((x - box.min.x)/(box.max.x - box.min.x));
    uvs.push((y - box.min.y)/(box.max.y - box.min.y));
}
geometry.setAttribute( 'uv', new THREE.BufferAttribute( new Float32Array(uvs), 2 ) );

```

Figura 46: Codi per calcular les coordenades de textura.

L'objecte del mapa, es gira -90 graus i es transporta al centre de l'escena. Això és pel fet que per la funció `asc2ply`, l'objecte PLY està projectat sobre el pla x,y en comptes del pla x,z. A més l'objecte està ubicat a les coordenades reals de la projecció, per tant, s'ha de transportar al centre de l'escena. Finalment, se li assigna una capa de l'escena i s'afegeix a l'escena.

A la mateixa funció d'inicialització també s'afegeixen les fotografies. Aquestes es creen amb una geometria rectangular i una textura que és la mateixa imatge. Després s'ubica a la posició corresponent i s'afegeix a la capa de fotografies. També s'afegeix a l'escena.

En últim lloc, a la funció d'inicialització, es crea un renderitzador de l'escena, s'afegeixen els `OrbitControls` per a poder desplaçar-se per l'escena, es crea un oient per si es reajusta la grandària de la finestra i es crida a una funció per inicialitzar la interfície amb la qual l'usuari pot interaccionar amb l'escena.

Aquesta interfície, amb la qual l'usuari pot interaccionar, afegeix uns botons per seleccionar la capa que es vol visualitzar. També afegeix uns controls lliscants per definir els valors de les variables globals "movimentmodels" i "movimentllum", ja mencionats anteriorment.

Aquests moviments tenen lloc a la funció `animate()`, la qual s'activa amb cada frame. Aquesta funció utilitza els valors de la variable global "movimentmodels" per tornar a calcular la rotació dels models 3D i la translació en els eixos x, y i z.

La funció `animate()` també crida a la funció per actualitzar la posició de la llum. Aquesta funció s'ha programat per representar la llum com el sol a la figura 47. Com es pot veure, els sol, pot girar al voltant del model 3D amb l'azimut, i pot estar a una altura de 0 a 90 graus. Inicialment, el valor de l'azimut és 45 graus i el de l'altura és 60 graus.

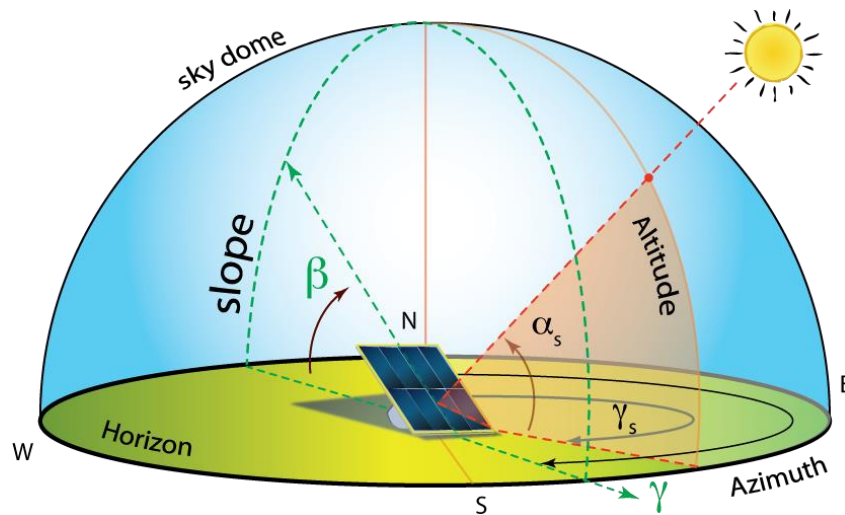


Figura 47: Exemple de la descripció dels sol a través de l'azimut i l'altura.[49].

7.5. Visualització a la web

Primerament, per poder visualitzar el contingut de l'exportació a la web, hem de saber que no es pot simplement fer doble clic sobre l'arxiu index. Això no és possible, ja que un servei web no pot accedir als arxius de l'usuari al sistema.

Per a poder visualitzar el contingut a la web onilne, caldria disposar d'un servidor de webs estàtiques, al qual s'hauria de carregar tota la carpeta generada pel plugin. D'aquesta forma seria accessible en línia.

Per altra banda, per a poder visualitzar el contingut a la web en local, és necessari accedir través de la consola a la carpeta creada per l'algoritme. Un cop ubicats a la carpeta s'ha de cridar la comanda:

```
python3 -m http.server
```

Aquesta comanda crea un servidor local que serveix com a amfitrió (host) per poder carregar els arxius del sistema. Per visualitzar el contingut a la web, s'ha d'escriure al buscador localhost:8000. A les següents figures es poden veure exemples del resultat.



Figura 48: Visualització web del contingut exportat.

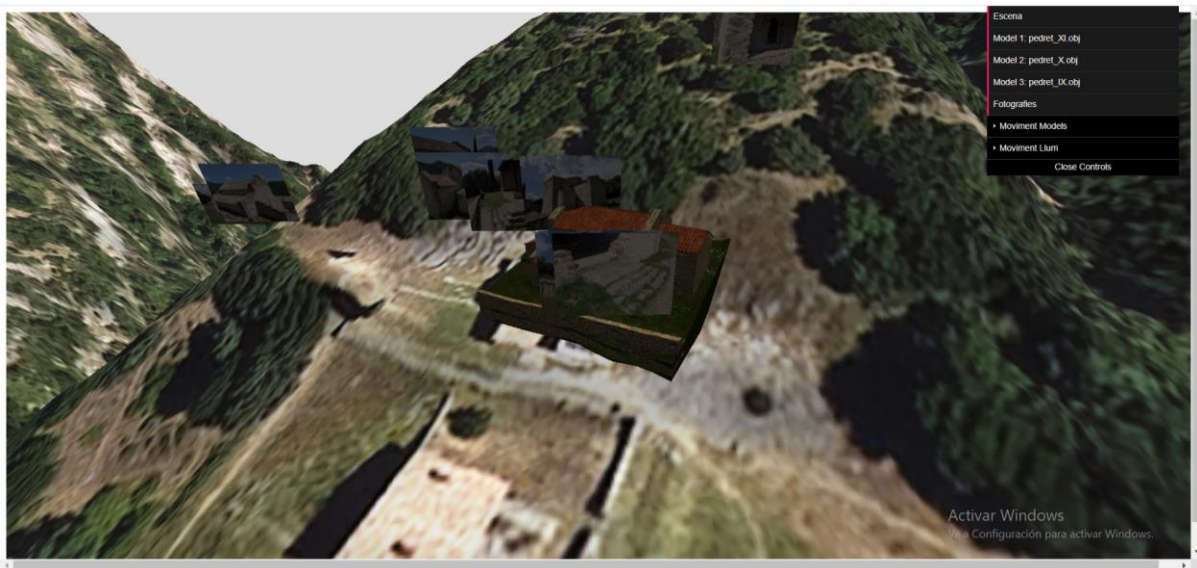


Figura 49: Visualització web del contingut exportat amb les Fotografies mostrades.

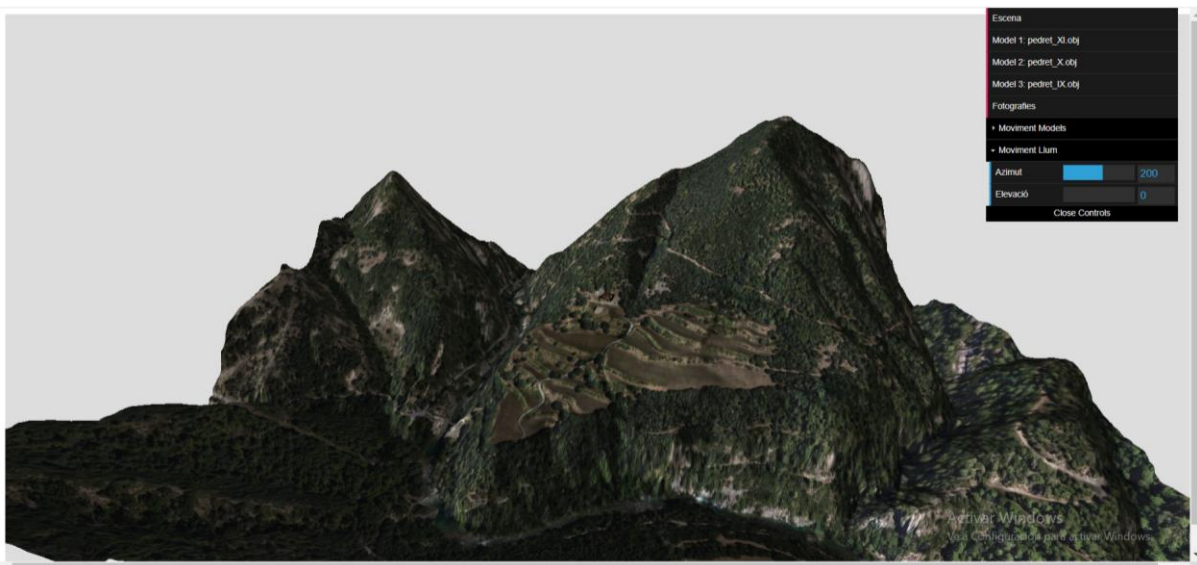


Figura 50: Visualització web del contingut exportat. Azimut = 200, Elevació = 0.

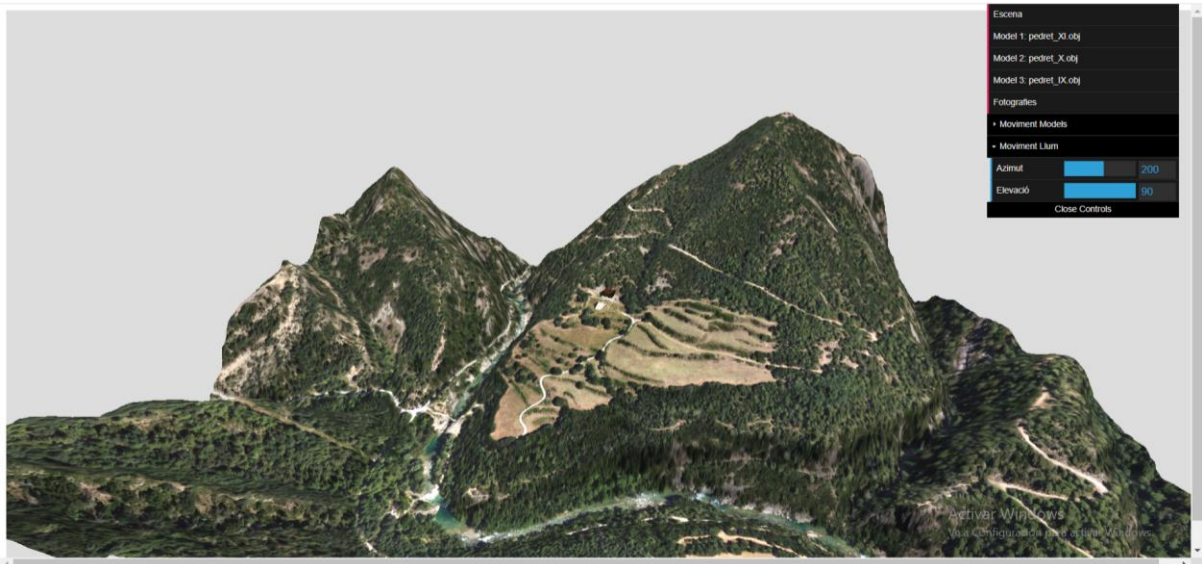


Figura 51: Visualització web del contingut exportat. Azimut = 200, Elevació = 90.

8. Conclusions

Aquest últim capítol presenta les reflexions i conclusions del projecte dut a terme. Ja que per poder concloure correctament un projecte, s'ha d'elaborar una anàlisi del treball realitzat als últims mesos, deixant constància dels objectius complerts, les decisions preses i s'ha de tenir en compte les possibles ampliacions que pot requerir el projecte en un futur.

Per facilitar la lectura d'aquest últim capítol s'ha dividit en tres apartats: conclusions respecte a la resolució del problema, conclusions respecte a la planificació i feina futura.

8.1. Respecte la resolució del problema

L'objectiu del projecte era integrar un model 3D a un Sistema d'Informació Geogràfica. Addicionalment, el projecte havia de permetre la integració de fotografies i la possible exportació a una visualització web. Aquestes característiques del plugin han estat desenvolupades completament tal com estava plantejat inicialment. És possible importar les fotografies i models 3D a QGIS. I és possible exportar-los a la web per a la seva visualització.

La solució construïda permet un manteniment fàcil i simple a l'haver separat cada funció. La utilització del plugin és senzilla d'entendre i els historiadors podran fer ús del plugin sense cap complicació. Tot i que no s'ha pogut realitzar la validació dels usuaris com estava plantejat inicialment.

En definitiva, es pot afirmar que la solució que es presenta resol el problema i compleix la majoria dels objectius plantejats a l'inici.

8.2. Respecte la planificació

Tal com ja s'ha explicat, no s'ha pogut complir els objectius proposats a les iteracions. Això ha estat degut a la poca experiència planificant projectes i el poc coneixement que es tenia de QGIS.

Per a poder dur a terme el projecte, s'ha hagut d'ampliar el termini de finalització. Amb aquesta ampliació sí que s'ha pogut acabar la part tècnica del projecte, però, igualment ha faltat una mica de temps per a poder realitzar la validació d'usuaris. Tot i que el director i el codirector del projecte han ajudat a validar la feina per tal que els usuaris la puguin utilitzar correctament.

Finalment, pel que fa a la planificació econòmica, contrastant les dades de gestió econòmica pressupostades inicialment amb les dades finals, es pot concloure que es va elaborar un bon pressupost. Tot i que finalment sí que ha estat necessari fer ús del fons de contingència per la falta de temps com ja s'havia plantejat a la secció d'imprevistos.

8.3. Feina futura

Tot i que el projecte finalment compleix amb les funcionalitats plantejades inicialment, es poden afegir d'altres que no s'havien plantejat. Però, la part de la validació d'usuaris quedaria pendent, i en futur estaria bé fer-la per acabar de validar el projecte.

Hi ha altres funcionalitats que es podrien implementar al plugin, per exemple la capacitat de visualitzar les fotografies a QGIS de forma senzilla, o la capacitat d'afegir una textura al model 3D a QGIS. També es podria afegir una altra interfície, per tal que el plugin aparegui a la barra superior com la resta de plugins.

Bibliografia

[1] Wikipedia – Sant Quirze de Pedret. https://ca.wikipedia.org/wiki/Sant_Quirze_de_Pedret [en línia] (consulta: 27/02/2022).

[2] ViRVIG. <https://www.virvig.eu/> [en línia] (consulta: 27/02/2022).

[3] EHEM Project. <http://www.ehemproject.eu/> [en línia] (consulta: 28/02/2022).

[4] Heritage Research Hub. <https://www.heritageresearch-hub.eu/> [en línia] (consulta: 28/02/2022).

[5] EHEM Project. 3D Model *screenshot*.

[6] Geoinnova. https://geoinnova.org/blog-territorio/que-es-un-sig-gis-o-sistema-de-informacion-geografica/?gclid=CjwKCAiAyPyQBhB6EiwAFUuaks-bRE-kQAC62FvGF7wrv1YFfhfPckuvbSf2rQClqxLF_hXQbR7o7xoCt0UQAvD_BwE [en línia] (consulta: 01/03/2022).

[7] QGIS. <https://www.qgis.org/es/site/> [en línia] (consulta: 28/02/2022).

[8] Niixer. <https://niixer.com/index.php/2020/11/10/modelos-3d/> [en línia] (consulta: 10/03/2022).

[9] Wikipedia - .obj file. https://en.wikipedia.org/wiki/Wavefront_.obj_file [en línia] (consulta: 10/03/2022).

[10] Sketchfab - Model 3D *screenshot*. <https://sketchfab.com/3d-models/la-bellesa-dominant-la-forca-bmnb-75413a5559214dfa918c060b94c487cc> [en línia] (consulta: 10/03/2022).

[11] Wikipedia – Interfície de Programació d'Aplicacions. https://ca.wikipedia.org/wiki/Interf%C3%ADcie_de_programaci%C3%B3_d%27aplicacions [en línia] (consulta: 10/03/2022)

[12] Wikipedia – JPEG. <https://ca.wikipedia.org/wiki/JPEG> [en línia] (consulta: 10/03/2022)

[13] Institut Cartogràfic i Geològic de Catalunya *screenshot*. <http://www.icc.cat/vissir3/> [en línia] (consulta: 02/03/2022).

[14] EHEM Project. 3D Model *screenshot*. Munoz-Pandiella, I., Argudo, O., Otzet, I. L., Comas, J. F., Casademont, G. A., Pueyo, X., & Andujar, C. (2022). Digital Reintegration of Distributed Mural Paintings at Different Architectural Phases: the Case of St. Quirze de Pedret. EG GCH 2022, Delf

[15] EHEM Project. 3D Model *screenshot*. Munoz-Pandiella, I., Argudo, O., Otzet, I. L., Comas, J. F., Casademont, G. A., Pueyo, X., & Endollar, C. (2022). Digital Reintegration of

Distributed Mural Paintings at Different Architectural Phases: the Case of St. Quirze de Pedret. EG GCH 2022, Delf

[16] QGIS. <https://www.qgis.org/es/site/> [en línia] (consulta: 28/02/2022).

[17] GVSIG. <http://www.gvsig.com/es> [en línia] (consulta: 01/03/2022).

[18] ARGIS. <https://www.arcgis.com/index.html> [en línia] (consulta: 01/03/2022).

[19] QGIS Plugins. <https://plugins.qgis.org/plugins/Qgis2threejs/> [en línia] (consulta: 02/03/2022).

[20] Mapping GIS. <https://mappinggis.com/2012/08/cinco-razones-para-comenzar-a-usar-gvsig/> [en línia] (consulta: 17/03/2022).

[21] Institut Cartogràfic i Geològic de Catalunya. <https://www.icgc.cat/> [en línia] (consulta: 02/03/2022).

[22] Youtube – How to create 3D Building map view *screenshot*. <https://www.youtube.com/watch?v=b-k0LEw3t7I> [en línia] (consulta: 03/03/2022).

[23] AGILE- Manifesto. <https://agilemanifesto.org/principles.html> [en línia] (consulta: 02/03/2022).

[24] Alamy. <https://www.alamy.com/agile-software-development-methodology-image226028076.html> [en línia] (consulta: 03/03/2022).

[25] Elaboració pròpia amb GanttProject.

[26] Elaboració pròpia amb GanttProject.

[27] Page Personnel. <https://www.pagepersonnel.es/> [en línia] (consulta: 05/03/2022).

[28] Mas Mòvil – Fibra òptica. https://www.tufibramasmovil.com/fibra-optica/?u=0101010507&gclid=Cj0KCQjwz7uRBhDRARIsAFqjulkUYLFRgEmRoCTtcwbpiW4_ttCH8oVPoKtbSW7cr2I7hUMsLDqxrWAaAvEXEALw_wcB [en línia] (consulta: 10/03/2022).

[29] Luz Ahora – Tarifa. <https://tarifaluzhora.es/> [en línia] (consulta: 10/03/2022).

[30] WordPress - <https://miguelangelsuarez.files.wordpress.com/2012/07/logitud-y-latitud1.jpg> [en línia] (consulta: 04/06/2022).

[31] Doc QGIS – Sistema de coordenades de referència. https://docs.qgis.org/2.14/es/docs/gentle_gis_introduction/coordinate_reference_systems.html [en línia] (consulta: 04/06/2022).

- [32] MappinGIS - .Como modificar la estructura de la tabla de atributos de una capa. <https://mappinggis.com/2016/10/modificar-la-estructura-la-tabla-atributos-una-capa/> [en línia] (consulta: 05/06/2022).
- [33] CursosGis <https://www.cursosgis.com/como-extraer-valores-de-archivos-raster-a-entidades-de-punto-con-arcgis/> [en línia] (consulta: 05/06/2022).
- [34] ArcGis – Que son los datos raster?. <https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm> [en línia] (consulta: 11/06/2022).
- [35] DEM to 3D – Impresión 3D de modelos de terreno. <https://demto3d.com/> [en línia] (consulta: 11/06/2022).
- [36] ArcGis – Que son los datos raster?. <https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm> [en línia] (consulta: 11/06/2022).
- [37] SourceForge. <https://sourceforge.net/> [en línia] (consulta: 17/06/2022).
- [38] Qt <https://www.qt.io/> [en línia] (consulta: 17/06/2022).
- [39] Wikipedia – C++. <https://es.wikipedia.org/wiki/C%2B%2B> [en línia] (consulta: 17/06/2022).
- [40] GDAL – Documentation. <https://gdal.org/> [en línia] (consulta: 17/06/2022).
- [41] Open Geospatial Consortium. <https://www.ogc.org/> [en línia] (consulta: 17/06/2022).
- [42] Institut Cartogràfic i Geològic de Catalunya. <http://www.icc.cat/vissir3/> [en línia] (consulta: 02/10/2022).
- [43] Open Street Map. <https://www.openstreetmap.org/#map=6/40.007/-2.488> [en línia] (consulta: 02/10/2022).
- [43] OsGeoLive – GDAL/OGR. https://live.osgeo.org/es/overview/gdal_overview.html [en línia] (consulta: 19/06/2022).
- [44] Grass GIS. <https://grass.osgeo.org/> [en línia] (consulta: 19/06/2022).
- [45] Wikipedia – GPS. <https://es.wikipedia.org/wiki/GPS> [en línia] (consulta: 19/06/2022).
- [46] Wikipedia – GPX. <https://es.wikipedia.org/wiki/GPX> [en línia] (consulta: 19/06/2022).
- [47] QGIS - Plugins. <https://plugins.qgis.org/plugins/> [en línia] (consulta: 19/06/2022).
- [48] Map Server. <https://mapserver.org/> [en línia] (consulta: 19/06/2022).

[49] E-education – Solar Resource Assesement and Economics. <https://www.e-education.psu.edu/eme810/node/576> [en l nia] (consulta: 06/10/2022).