
This is the **published version** of the bachelor thesis:

Solanich Faja, Marc; Marchán Hernández Hernández, Juan Fernando, dir. Local Rainfall Analysis : Integració del submòdul Python d'anàlisi de pluges de la plataforma HEIMDALL a l'aplicació de QGIS. 2020. (1373 Màster Universitari en Geoinformació)

This version is available at <https://ddd.uab.cat/record/231422>

under the terms of the  license

Marc Solanich Faja

Local Rainfall Analysis:

Integració del submòdul Python d'anàlisi de pluges de la plataforma HEIMDALL a l'aplicació de QGIS

Projecte de final de màster



Universitat Autònoma de Barcelona

Institut Cartogràfic i Geològic de Catalunya



Bellaterra, 2020

Resum

La pluja és el desencadenant més generalitzat i freqüent d'esllavissades al llarg del món i per tant els llindars de pluja són una eina essencial per a: 1) la predicció a escala local i/o regional de riscos naturals desencadenats per la pluja, i 2) la implementació de sistemes d'alerta primerenca. D'aquesta manera caracteritzar efectivament els patrons de pluja amb intensitat i/o duració suficient per a desencadenar esllavissades és molt important. Tot i això, establir llindars crítics d'aquestes pluges és complicat degut a la gran variabilitat de la pluja depenent l'estacionalitat, geografia i climatologia i la relació amb altres factors, sobretot antropogènics. Conseqüentment, l'anàlisi dels paràmetres de les pluges amb potencial de generar esllavissades és un cas d'estudi fonamental tant en la detecció de riscos geològics derivats d'esllavissades i per l'eficient gestió dels plans d'emergència per enfrontar les crisis derivades d'aquests riscos, com ara per la comprensió i definició de nous valors llindars per ser comparats amb els episodis de pluja i identificar-los com a potencials o no de generar esllavissades.

El desenvolupament del plugin Local Rainfall Analysis a partir de l'algoritme de l'anàlisi de pluges del mòdul de simulació d'esllavissades de la plataforma HEIMDALL desenvolupat per l'Institut Cartogràfic i Geològic de Catalunya (ICGC) permet la descripció i parametrització d'un o diversos episodis de pluja concrets en l'espai i el temps, per tal de determinar si aquesta ha set la causa i/o desencadenant d'una esllavissada. Aquesta anàlisi ajuda a la presa de decisions de l'equip del ICGC i proporciona informació sobre valors de pluja llindars a escala local per augmentar la precisió dels coneixements de valors llindars d'episodis de pluja amb potencial de desencadenar un episodi d'esllavissades i d'aquesta manera estudiar amb més coneixement les possibles situacions de risc i/o de crisi.

Índex

Resum	1
1. Introducció	3
1.1. Inestabilitats de vessant i moviments en massa	3
1.2. Condicions crítiques de pluja i definició de llindars	4
1.3. Projecte Heimdall	5
1.4. Institut Cartogràfic i Geològic de Catalunya	6
2. Objectius	7
3. Desenvolupament de l'aplicació	8
3.1. Context i necessitats actuals	8
3.1.1. Entorn Python en els Sistemes d'informació Geogràfica i QGIS	8
3.1.2. Plugins de Python com a eines de QGIS	8
3.1.3. Arquitectura de la plataforma HEIMDALL – Modelització i Simulació	9
3.2. Anàlisi de requisits	14
3.2.1. requisits funcionals	14
3.3. Disseny funcional	15
3.4. Solució metodològica	15
3.4.1. L'ecosistema format per Python i Qgis	16
3.4.2. Disseny de l'entorn de desenvolupament de Python	17
3.4.3. Anàlisi del codi existent	19
3.5. Implementació	23
3.5.1. Modificacions al codi	23
3.5.2. Creació del plugin	42
4. Prova de funcionament en un entorn real	51
4.1. Primer cas d'anàlisi: colada de fang a la carretera de Begues (Begues, Baix Llobregat)	51
4.2. Segon cas d'anàlisi: camí cap a Can Cadafalc, Mas Canyelles (Castellar del Vallès, Vallès Occidental)	57
5. Conclusions	65
6. Bibliografia	66
7. Annex	69
7.1. Anàlisi de requeriments	69
7.2. Anàlisi del codi existent	75
7.2.1. Submòdul meteocat_MTIG.py	75
7.2.2. Submòdul localrainfallanalysis_MTIG.py	80

1. Introducció

1.1. Inestabilitats de vessant i moviments en massa

Els moviments en massa o de vessant, també coneguts amb altres termes genèrics com ara inestabilitats gravitatòries, esllavissades o ensorraments, entre d'altres (en aquest projecte s'utilitza el terme general d'esllavissades), es categoritzen com a fenòmens associats a processos i dinàmiques externes a escala de conca. Aquests fenòmens geològics fan referència a un conjunt de factors complexos generalment associats amb un desencadenant natural o antropogènic (Malamud *et al.*, 2004; Corominas 2004) que poden manifestar-se en multitud de formes incloent caigudes de roques, corrents d'esbaldregalls (*debris flows*), colades de fang (*mud flows*), reptacions del sòl (Hansen, 1984). Els moviments de vessant han estat extensament estudiats pel que fa referència a: els seus desencadenants, factors de control, evolució, caracterització i classificació (Terzaghi 1950; Skempton i Hutchinson 1969; Diaku *et al.*, 1996 Vaunat i Leroueil 2002; Hungr, 1014).

Conseqüentment, els moviments en massa es caracteritzen per tenir lloc a gairebé cada país del món, essent una causa de possible mortalitat i representat un gran impacte per a les infraestructures cada any (Petley, 20011; Kirschbaum *et al.*, 2015). Aquests es desencadenen en una gran varietat de litologies, morfologies, configuracions hidrològiques, cobertores de sols i zones climàtiques, tot aquest conjunt de característiques defineixen també les zones amb més o menys susceptibilitat a generar un moviment en massa. Un dels factors més comuns de desencadenament són les pluges intenses i prolongades (Malamud *et al.*, 2004; Kirschbaum i Stanley 2018). Tot i això, molts d'altres factors també es caracteritzen pel potencial de desencadenament d'esllavissades, com ara l'activitat sísmica, els canvis elevats de temperatura, o activitats antropogèniques com la mineria, la construcció, un drenatge incorrecte, canvis d'usos de sòl i desforestació (Larsen i Santiago-Román, 2001; Glade, 2003; Guzzetti *et al.*, 2009; Keefer, 1994).

Com a processos complexos que són, les esllavissades són difícils de predir en termes de localització, moment de fractura, velocitat i abast. Mentre que les fractures reactivades poden ser observades i monitoritzades, les primeres son difícils de preveure amb precisió (Glade, 2003).

La pluja és el desencadenant més generalitzat i freqüent d'esllavissades al llarg del món (Dunning *et al.*, 2007); per tant, caracteritzar efectivament els patrons de pluja amb intensitat i/o duració suficient per a desencadenar esllavissades és molt important. Tot i això, establir llindars crítics d'aquestes pluges és complicat degut a la gran variabilitat de la pluja depenent l'estacionalitat, geografia i climatologia (Guzzetti *et al.*, 2008) i la relació amb altres factors, sobretot antropogènics.

1.2. Condicions crítiques de pluja i definició de llindars

La pluja es destaca com un dels principals factors desencadenants d'esllavissades. La identificació dels paràmetres pluviomètrics o meteorològics relacionats amb una esllavissada es relaciona directament amb altres factors com ara la composició del sòl i les condicions hidrològiques que al ser sobrepassades, són capaces de generar esllavissades (Guzzetti, 2007).

D'aquesta manera, els llindars de pluja són una eina essencial per a: 1) la predicció a escala local i/o regional de riscos naturals desencadenats per la pluja, i 2) la implementació de sistemes d'alerta primerenca (Jakob *et al.*, 2006; Badoux *et al.*, 2009; Tiranti and Rabuffeti, 2010; Papa *et al.*, 2013; Berenguer *et al.*, 2015; Stähli *et al.*, 2015).

L'estudi d'aquests comporta diverses dificultats:

- a) Obtenir mesures directes de pluja en les àrees font properes dins les conques amb potencial de fluxos torrencials és complicat i per això, la majoria de llindars pluviomètrics publicats han estat establerts utilitzant dades mesurades a partir dels pluviòmetres més propers, el què pot significar una distància de diversos kilòmetres entre l'estació de mesura i l'inici del flux torrencial (Brunetti *et al.*, 2010),
- b) Determinar precisament la pluja real que desencadena l'esdeveniment és difícil degut a l'intens comportament convectiu i la variabilitat temporal i espacial de les pluges desencadenants (Abancó *et al.*, 2012; Nikolopoulos *et al.*, 2014; Restrepo *et al.*, 2008),
- c) A causa de l'important efecte de l'escala, els llindars determinats a escales regionals, locals i/o a escala d'àrea de captació haurien de ser analitzats per separat a causa de l'efecte important de la dependència dels llindars amb el lloc específic d'origen i/o mesura (Segoni *et al.*, 2014a),
- d) La dificultat d'establir criteris clars i objectius per tal de definir la duració d'un episodi de pluja en concret (Abancó *et al.*, 2016 defineix la duració d'un episodi amb la condició de no haver existit precipitació durant una hora abans i després de l'episodi),
- e) La distribució temporal dels episodis de pluja desencadenants de fluxos torrencials, els quals la majoria es donen en moments d'estiu, i
- f) La fusió de la neu en la primavera influencia dràsticament els episodis durant la primavera amb un input excessiu d'aigua.

A partir de l'anàlisi de diversos fluxos torrencials generats per esdeveniments pluviomètrics crítics conjuntament amb la caracterització d'aquests per tal de ser contrastats amb altres esdeveniments pluviomètrics que no han desencadenat cap flux torrencial, Abancó *et al.*, (2016) apunta que:

- 1) Els fluxos torrencials són desencadenats per pluges curtes i intenses, on la majoria es donen en períodes d'estiu i amb períodes de retorn de menys de dos anys.
- 2) Els episodis de pluja analitzats indiquen que la intensitat de la pluja és el paràmetre clau per la discriminació entre episodis amb potencial desencadenant de fluxos torrencials i episodis sense potencial desencadenant. Finalment, Abancó *et al.*, (2016) utilitza aquests paràmetres com a variables per a definir els valors llindars de les pluges crítiques. Els llindars són calculats a partir de càlculs gràfics entre la intensitat mitja i la durada dels episodis de pluja. Aquest és el mètode estàndard de calcular gràficament episodis catastròfics possibles en l'anàlisi de riscos d'esllavissades.

1.3. Projecte Heimdall

En els darrers anys, s'ha incrementat el nombre de desastres naturals, amb més o menys complexitat i/o severitat, que han afectat a Europa. Episodis extrems d'incendis forestals (p.ex. Portugal l'any 2003, 2005 i 20017, Grècia l'any 2007, Espanya l'any 2015 i França l'any 2016) i d'inundacions a escala regional (p.ex. Sèrbia i Croàcia l'any 2014, Regne unit l'any 2014, Àustria i República Txeca l'any 2013 i Alemanya els anys 2002, 2006 i 2013), amb milers de persones afectades, resultant en la destrucció de propietats i significatius impactes en l'economia i en els ecosistemes al llarg de diverses regions i països (ECHO, 2017). Les interdependències del risc i els efectes en cascada amplifiquen la intensitat i l'impacte. Així doncs, aquestes situacions complexes remarquen la necessitat de millorar les estratègies de planificació col·lectives a llarga escala al llarg de tots els factors involucrats en la planificació i resposta del risc (HEIMDALL, 2017).

La gestió de situacions de crisi complexes, essent naturals, accidentals o d'origen intencionat, generalment requereix la participació i coordinació de múltiples equips de primera intervenció de diferents organitzacions, incloent, però sense limitar-se a: unitats de bombers, departaments policials, serveis d'emergències mèdiques, unitats de protecció civil i centres de comandament i control (HEIMDALL, 2017).

Amb l'objectiu de millorar la gestió de la complexitat d'aquest tipus de crisi es va desenvolupar el projecte HEIMDALL finançat per la Unió Europea a través de la línia del Programa Horizon 2020 (*Innovation Action. SEC-01-DRS-2016 – Secure societies – Integrated tools for response Planning and scenario Building*). L'objectiu del projecte és millorar la resposta a situacions d'emergències naturals complexes mitjançant l'ús d'eines informàtiques integrades en una plataforma que doni suport a la planificació de la resposta. La solució es basa en la construcció d'escenaris realistes multi-temàtics que reproduïxien els fenòmens naturals.

La plataforma també permet valorar el risc i l'impacte derivats de les accions de resposta mitjançant les observacions de camp i les simulacions. D'aquesta manera la informació aportada per la plataforma del servei HEIMDALL dota els usuaris d'una visió global de l'emergència que integra diferents aspectes i que ajuda a la presa de decisions, alhora que es pot fer servir com a element d'aprenentatge ja que permet crear situacions fictícies o reproduir situacions històriques (Friedemann *et al.*, 2018). El desenvolupament del programari compta amb l'estreta participació de les autoritats locals com proteccions civils, bombers, etc.

Per a la realització del projecte, l'Institut Cartogràfic i Geològic de Catalunya (ICGC) ha desenvolupat dos mòduls: el simulador d'esllavissades i el mòdul de sensors *in situ*. El primer té com a objectiu l'elaboració de mapes de susceptibilitat per esllavissades (lliscaments rotacionals, caigudes de roques i fluxos d'esbaldregalls) d'una àrea d'estudi, que generalment serà un lloc on hi ha hagut un incident per esllavissades. Aquest mòdul inclou una eina o submòdul que obté les pluges anteriors a l'incident, la previsió pels dies següents i per comparació amb les corbes de Guzzetti *et al.*, (2007) per tal de preveure si el perill per esllavissades augmenta disminueix en els següents dies.

El mòdul de sensors *in situ* està concebut per que en esllavissades amb una velocitat prou lenta per a poder gestionar el risc, es pugui instal·lar sensors (fissuròmetres, inclinòmetres, piezòmetres, etc.) que permeten fer un seguiment de l'evolució del moviment i d'aquesta manera modular les accions a fer per evitar el risc per a les persones i infraestructures. El mòdul consisteix en una interfície gràfica que permet observar i fer un seguiment de les dades obtingudes pels sensors (Friedemann *et al.*, 2018; Battiston *et al.*, 2019).

Aquests dos mòduls, sigui conjuntament o per separat, es consideren molt útils per part dels usuaris finals del projecte HEIMDALL (bombers, policia, protecció civil, etc.), ja que permeten determinar els llocs segurs per instal·lar el centre de comandament avançat, si el perill pot augmentar o no en els dies posteriors i les àrees a evacuar en els casos que sigui necessari (Battiston *et al.*, 2019)..

1.4. Institut Cartogràfic i Geològic de Catalunya

En el Diari Oficial de la Generalitat de Catalunya (DOGC), núm. 6551, del 30 de gener de 2014, es publica la Llei 2/2014, de creació de l'Institut Cartogràfic i Geològic de Catalunya (ICGC) i la supressió de l'Institut Cartogràfic de Catalunya (ICC) i de l'Institut Geològic de Catalunya (IGC). Aquesta nova institució entra en funcionament l'1 de febrer de 2014. Les seves funcionalitats són les relacionades amb l'exercici de les competències sobre la geodèsia i la cartografia, i sobre la infraestructura de dades espacials de Catalunya, i també les d'impulsar i dur a terme les actuacions relatives al coneixement, prospecció i la informació sobre el sòl i el subsòl, en els termes establerts per la Llei 16/2005, del 27 de desembre de la informació geogràfica i de l'Institut Cartogràfic de Catalunya, i per la Llei 19/2005, del 27 de desembre, de l'Institut Geològic de Catalunya (Institut Cartogràfic i Geològic de Catalunya (2019)).

L'Institut Cartogràfic i Geològic de Catalunya participa en el projecte HEIMDALL (*multi-Hazard cooperative Management tool for Data Exchange, response Planning and scenario building*) finançat per la Unió Europea a través de la línia del Programa Horizon 2020: Innovation Action. SEC-01-DRS-2016 – Secure societies – Integrated tools for response Planning and scenario Building (Institut Cartogràfic i Geològic de Catalunya (ICGC), 6 de juny de 2017).

El projecte es desenvoluparà en base a escenaris realistes en àrees geogràfiques específiques (Catalunya, Dinamarca i Escòcia) amb l'estreta participació de les autoritats locals com protecció civils, bombers, etc. Les tasques de l'ICGC, conjuntament amb el Centre Tecnològic de Comunicacions de Catalunya, se centraran en el desenvolupament de les eines i dels escenaris de moviments del terreny. El treball principal de l'ICGC en aquest projecte és el desenvolupament de les eines i dels escenaris de moviments del terreny per tal de donar resposta adequada en casos d'emergència (Institut Cartogràfic i Geològic de Catalunya (ICGC), 6 de juny de 2017).

L'Institut ja desenvolupa actualment tasques d'identificació, cartografia i zonificació de riscos geològics per moviments de terreny, com ara plans cartogràfics (Geotrell VI), suport a la Direcció General d'Ordenació del Territori i Urbanisme (estudis d'identificació de riscos geològics), actuacions en problemàtiques locals (Sallent) i és informant dels plans d'emergència (INUNCAT) (Institut Cartogràfic i Geològic de Catalunya (ICGC), 6 de juny de 2017).

2. Objectius

L'objectiu d'aquest projecte és incloure i adaptar el submòdul contingut dins del simulador d'esllavissades del projecte HEIMDALL d'anàlisi de pluges, *Rainfall Analysis*, a l'entorn SIG de QGIS a través de la creació d'un plugin. Aquest procés ha de possibilitar al usuari final de l'ICGC realitzar anàlisis, seguint els criteris de Abancó *et al.*, (2016) sobre possibles episodis de pluja desencadenants de fluxos torrencials i esllavissades, per tal de caracteritzar els paràmetres d'aquestes i entendre la seva variabilitat espacial i temporal com un factor clau en l'anàlisi de riscos geològics, a partir de:

1. Anàlisi de requisits del plugin a crear en l'entorn de QGIS, considerant les necessitats de l'usuari final de l'eina, i disseny d'una interfície del plugin que les compleixi.
2. Revisió detallada dels conceptes teòrics i pràctics del llenguatge de programació Python incloent-hi els marcs conceptuals i pràctics del llenguatge en l'entorn de QGIS.
3. Definir i construir un entorn Python (instal·lació de la IDE i llibreries de Python necessàries) que faciliti l'anàlisi del codi Python existent per tal d'entendre l'arquitectura i funcionalitats dels mòduls *METEOCAT* i *LOCALRAINFALANALYSIS*.
4. Utilitzar l'entorn Python creat per albergar les modificacions indicades al codi i afegir-hi les característiques necessàries per integrar-se a l'entorn de QGIS.
5. Aplicar el marc de treball establert (requeriments, disseny i modificacions del codi) per la integració dels mòduls *meteocat_MTIG* i *localrainfallanalysis_MTIG* del projecte Heimdall en l'entorn SIG de QGIS a través de la creació d'un plugin.

3. Desenvolupament de l'aplicació

3.1. Context i necessitats actuals

3.1.1. Entorn Python en els Sistemes d'informació Geogràfica i QGIS

QGIS és molt més que un software de Sistemes d'Informació Geogràfica (SIG); és, a més a més, un ecosistema de programació geoespacial, el qual permet als usuaris construir aplicacions geoespacionals utilitzant Python (Oxoli, 2017). QGIS (formalment coneguda com a Quantum GIS) és una de les aplicacions SIG professionals més populars, seguint el paradigma de desenvolupament gratuït i de codi lliure (*Free and Open Source Software, FOSS*) llicenciat sota GNU – GPL. QGIS és un projecte oficial d'*Open Source Geospatial Foundation (OSGeo)* executable en Linux, Unix, Mac OSX, Windows i Android i suporta nombrosos formats i funcionalitats de dades vectorials, dades ràster i bases de dades (QGIS Development Team, 2016).

Mentre que QGIS està desenvolupat mitjançant C++, aquest inclou un suport extensiu per la programació amb Python. QGIS incorpora un intèrpret de Python que pot ser utilitzat interactivament a través de la consola de Python de QGIS o per executar plugins escrits amb Python. A més a més QGIS estableix una API de fàcil comprensió per interactuar i controlar QGIS utilitzant codi Python. QGIS estableix tres possibilitats per utilitzar Python per treballar amb el sistema de QGIS: (1) a través de la consola de Python, (2) mitjançant plugins i (3) a partir d'aplicacions externes (Westra, 2014). En aquest projecte s'encamina a la utilització d'un plugin de Python per utilitzar codi Python dins l'ecosistema de QGIS.

PyQGIS, és la referència a les llibreries de Python dins QGIS, és la barreja de Python i QGIS per tal d'estendre i millorar la caixa d'eines de codi obert del SIG proporcionant una interfície completament programable del sistema de QGIS. PyQGIS permet escriure scripts i plugins per tal d'implementar noves funcionalitats i automatitzar tasques (Sherman, 2014).

3.1.2. Plugins de Python com a eines de QGIS

QGIS posseeix una potent interfície de programació que permet estendre tant les funcionalitats del software del nucli del programa com ara escriure scripts que permetin automatitzar les feines. D'aquesta manera, és possible estendre les funcionalitats de QGIS a través de l'ús de plugins. Aquests poden ser escrits amb Python, on es posicionen per davant dels plugins escrits amb C++ degut a la seva fàcil distribució i desenvolupament (Dobias, 2011).

Els plugins de QGIS es categoritzen en plugins del nucli o plugins de contribució. Els plugins del nucli són part de la distribució del QGIS i s'inclouen en la instal·lació d'aquest. Aquests poden estar escrits tant amb C++ o Python. Els plugins de contribució estan escrits amb Python i són desenvolupats generalment per la comunitat d'usuaris (Sherman, 2014).

En funció del plugin que es pretengui desenvolupar o del tipus de funcionalitat que es pretengui afegir a QGIS, aquest ofereix la possibilitat de generar:

- 1) un algorisme de processament (o un conjunt d'aquests) que proporciona una millor integració amb el programa, ja que pot executar-se dins del marc de processament amb altres components de QGIS (Dobias, 2011), o ara bé un

- 2) plugin amb interfície gràfica d'usuari (*Graphical User Interface, GUI*) amb el consegüent disseny personalitzat de la lògica per processar les dades. Tot i la simplicitat de funcionament

dels algorismes de processament (requereixen només la definició de dades d'entrada i sortida), els plugins amb GUI permeten a l'usuari una millor personalització i interacció amb el plugin.

3.1.3. Arquitectura de la plataforma HEIMDALL – Modelització i Simulació

La Fig. 1 representa l'arquitectura general de la plataforma HEIMDALL on es destaca el mòdul de Modelització i Simulació HEIMDALL (*simulators*) com a "caixa negra" dins de l'arquitectura en general. Aquest fet es deu al fet que aquest mòdul executa totes les comunicacions i interaccions amb altres components de HEIMDALL a través de la plataforma de serveis (*Service Platform, SP*). El conjunt de mòduls de simulació són controlats i executats a través de la SP que proporciona les entrades i paràmetres necessaris per executar i alhora gestiona els resultats necessaris per enviar als altres components del sistema, també s'encarrega d'alimentar a la interfície gràfica d'usuari (GUI) o enviar la informació a altre mòdul.

Els simuladors són activats per la SP que proveeix els paràmetres d'entrada per a fer les simulacions i obtenir resultats, els quals son emmagatzemats localment per a cada simulador, i són enviats una altra vegada a la SP segons demanda, que s'encarrega de representar els resultats a la interfície gràfica d'usuari (GUI) o per seguir processant aquestes dades en la resta de components de HEIMDALL (Battiston, 2016).

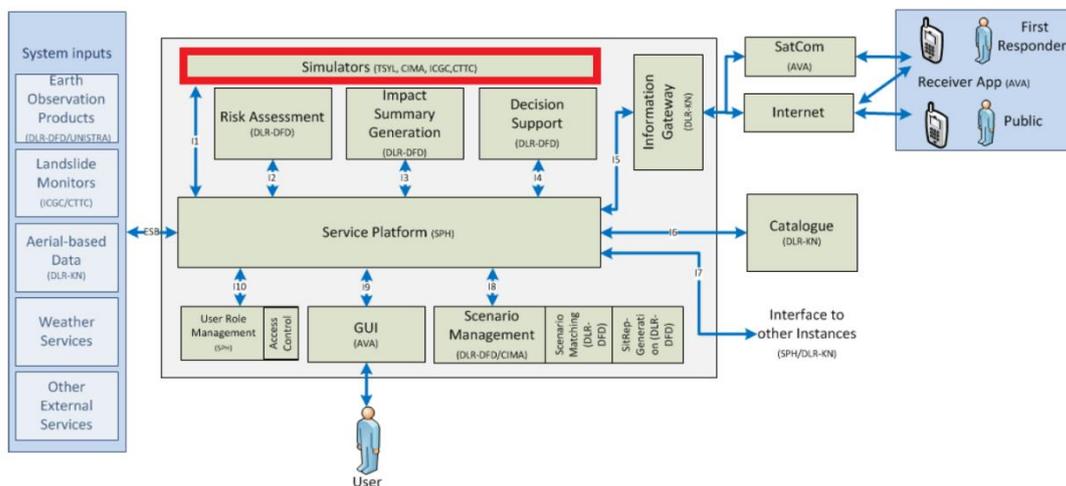


Figura 1 Ubicació del mòdul de Modelització i Simulació dins de l'arquitectura de la plataforma HEIMDALL. Modificat de (Battiston, 2016).

Cada simulador es comunica directament amb la SP, d'aquesta manera reben sol·licituds del SP per executar diversos tipus d'operacions, rebre dades i paràmetres per executar processos i simulacions per proveir resultats encarregats per la SP i compartir-la amb altres components. A la Fig. 2 es representen tots els components amb els quals els simuladors interactuen mitjançant la SP (Battiston, 2016).

- *Sistem inputs*: majorment dades provenint de monitors i sensors aeris i/o meteorològics
- *GUI*: projecta els resultats de simulacions i rep paràmetres d'entrada pels usuaris.
- *Risk Assessment*: rep informació de les extensions de risc i la intensitat del risc dels simuladors i proporciona l'exposició de les dades al FFS.
- *Scenario management*: els resultats generats pels simuladors son associats a un escenari en concret.

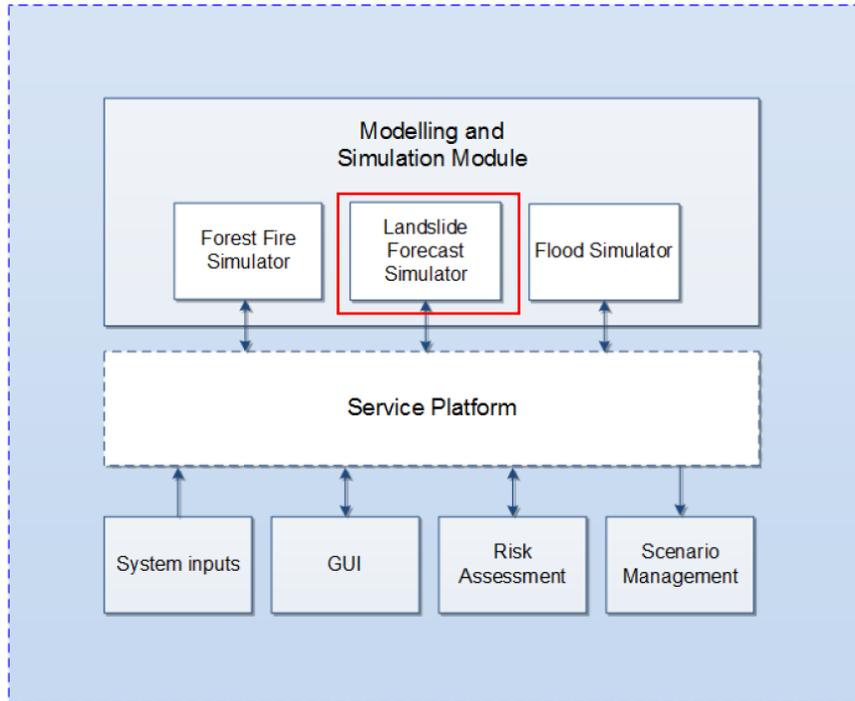


Figura 2 Arquitectura general del mòdul de Modelització i Simulació. Modificat de (Battiston, 2016).

Dins del mòdul de modelització i simulació, es troba el submòdul de simulació d'esllavissades desenvolupat pel ICGC, el qual és l'objectiu d'aquest treball. El simulador d'esllavissades de la plataforma HEIMDALL va ser desenvolupat per obtenir informació sobre (Fig.3):

- Cartografiar les àrees potencialment afectades pel trencament i propagació d'esllavissades (Fig. 4).
- Analitzar les condicions de pluja que desencadenen les esllavissades i com les condicions climàtiques post-esdeveniment poden afectar a l'estabilitat.
- Entrenament i preparació dels serveis d'emergència.

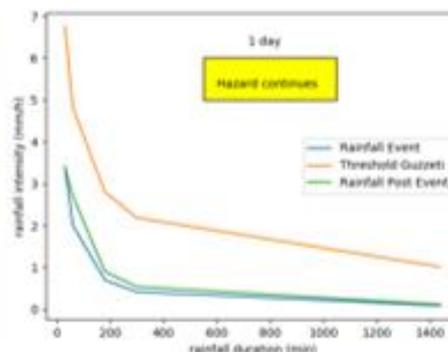
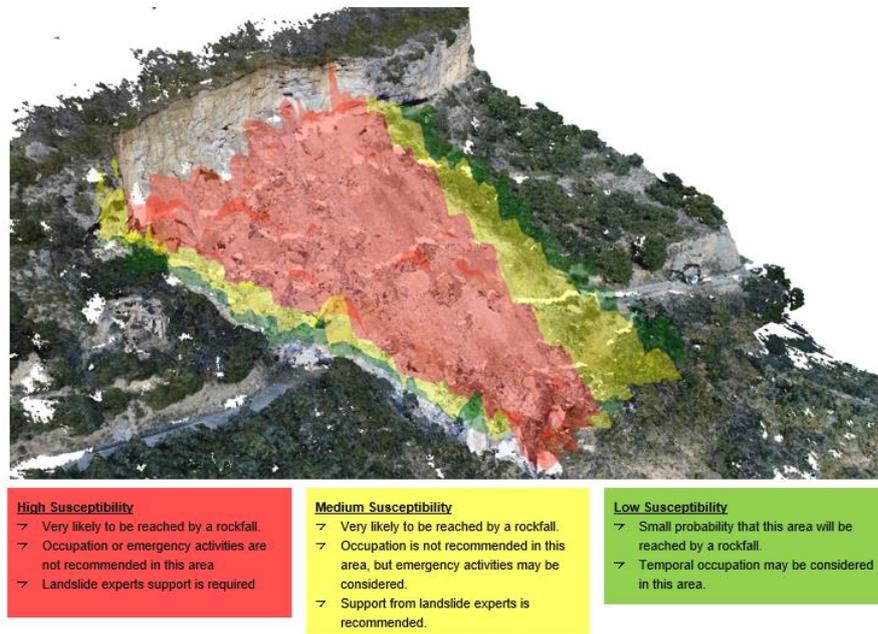


Figura 3 Representació dels resultats desitjats que complementen el simulador d'esllavissades de la plataforma HEIMDALL. A l'esquerra: cartografia d'àrees susceptibles o amb potencial d'esllavissament, al mig: comparació de la corba d'intensitat d'un episodi de pluges amb els llindars de pluges crítiques i a la dreta: planificació del sistema dels serveis d'emergència.

Landslide simulator



Multi-hazard Cooperative Management Tool for Data Exchange, Response Planning and Scenario Building

Figura 4 Cas d'exemple d'una àrea afectada per una esllavissada. A partir del submòdul de simulació i modelització d'esllavissades es poden cartografiar les àrees amb més susceptibilitat a desencadenar una esllavissada.

Aquest submòdul de simulació d'esllavissades es construeix a partir de dos submòduls (FIG. 5a) els quals han set dissenyats i implementats per l'Institut Cartogràfic i Geològic de Catalunya (ICGC). Aquests són: 1) Un model de susceptibilitat d'esllavissades per a caigudes de roques, corrents d'esbaldregalls (*debris flows*) i esllavissades rotacionals, i 2) un algoritme que automàticament analitza dades meteorològiques (de la xarxa d'estacions meteorològiques de Meteocat) pre- i post-esdeveniment amb l'objectiu d'enregistrar la intensitat de la pluja desencadenant i preveure si el risc incrementarà o es reduirà els dies posteriors a l'esdeveniment. En aquest cas no s'inclou el mòdul de sensors *in situ*.

El segon producte del simulador d'esllavissades proporciona les tendències de les condicions desencadenants de moviments de terreny. La pluja és el desencadenant més comú d'esllavissades i l'anàlisi de les condicions crítiques de pluja és una feina de recerca fonamental. Els llindars de pluja han estat definits en tot el món. El simulador d'esllavissades conté el submòdul que recull dades de pluja de l'estació meteorològica més propera (incloent les dades passades que van desencadenar l'esllavissada i les dades de predicció futura, amb l'objectiu d'analitzar com les condicions d'estabilitat evolucionaran en el futur) i son comparades amb els valors llindars de pluja predefinits.

El propòsit de l'anàlisi de pluja consisteix en: 1) Ajudar a establir llindars d'escala regional de pluja desencadenant d'esllavissades, 2) utilitzar aquests llindars definits per la comparació dels valors de la duració i de la intensitat de les pluges enregistrades i previstes per tal d'identificar si poden ser crítiques i 3) alimentar la base de dades dels esdeveniments de pluja desencadenants d'esllavissades amb l'objectiu de disposar d'aquestes per utilitzar-les en possibles escenaris futurs.

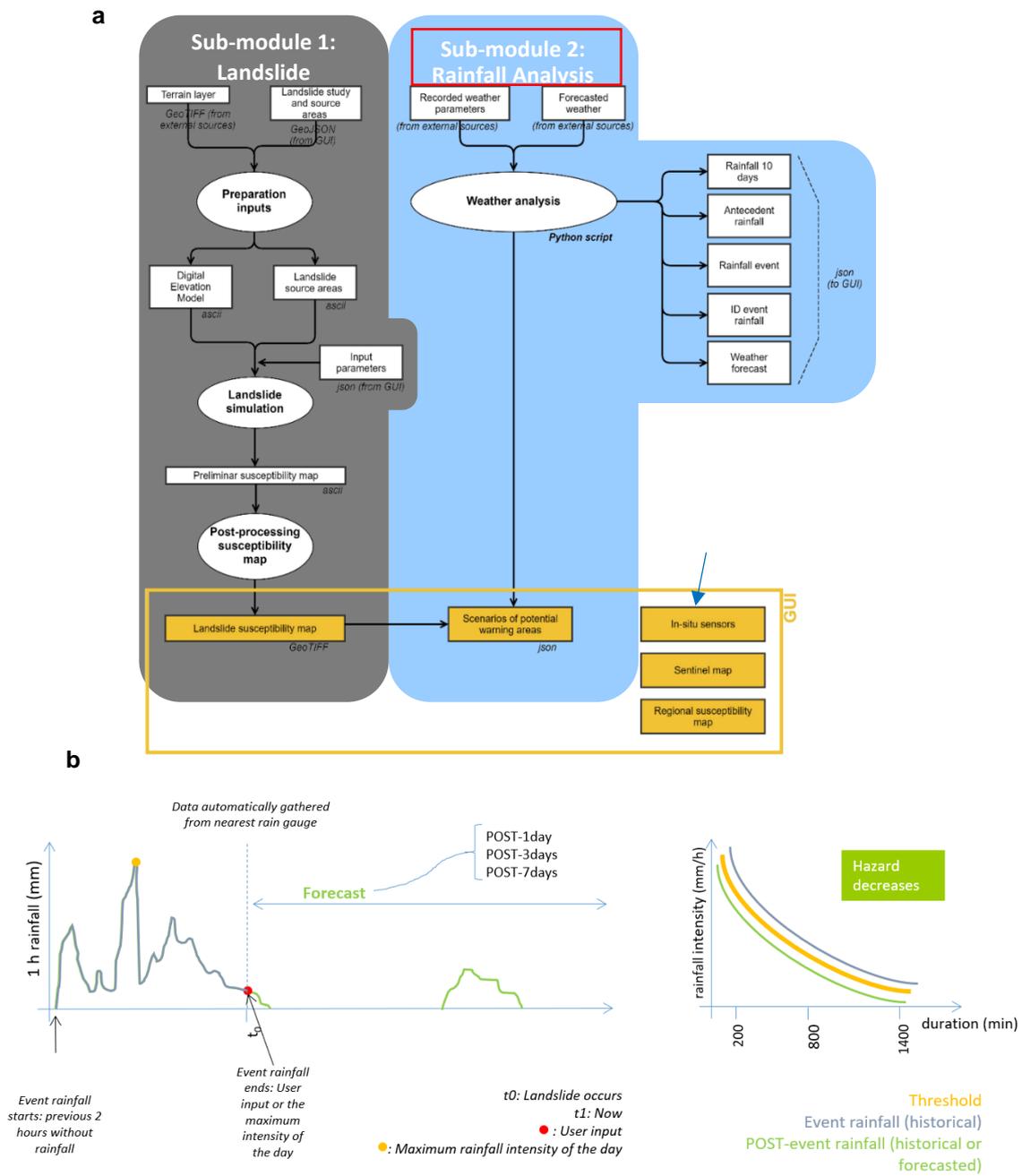
L'algoritme de l'anàlisi de pluges està construït entorn de la feina de Abancó *et al.* (2016) en l'estudi i caracterització de les condicions pluviomètriques crítiques que poden desencadenar fluxos torrencials durant episodis de pluja. Abancó *et al.* (2016) subratlla la pluja com el factor desencadenant més comú de fluxos en massa i torrencials. A partir de l'estudi de dades d'esdeveniments passats, Abancó *et al.* (2016) defineix empíricament a partir d'aquests el conjunt de valors llindars per a pluges desencadenants de fluxos d'esbaldregalls (*debris flows*) i, fluxos torrencials (*hiperconcentrated flows*).

L'anàlisi de pluges consisteix a analitzar per un període de temps, les condicions meteorològiques entorn la data introduïda per l'usuari (Fig. 5b). En el cas que durant aquest període de temps hi hagi pluja (es considera pluja si un conjunt de valors de precipitació supera les 2 hores de continuïtat de pluja), l'algoritme d'anàlisi de pluges genera els càlculs necessaris per comparar les intensitats de la pluja durant el període analitzat amb els valors llindars per determinar la perillositat de la pluja. L'anàlisi de pluges té una part de dades històriques (estudi de l'episodi de pluges que podria desencadenar una esllavissada) i una part de predicció futura per tal d'analitzar l'estabilitat i com aquesta pot ser alterada (Fig. 5b).

Les dades provenen de l'estació meteorològica més propera al punt introduït per l'usuari. Automàticament el programa calcula la diferència mínima de distància entre el punt de l'esdeveniment i l'estació més propera.

Aquest projecte es centra en el submòdul d'anàlisi de pluges (*Rainfall Analysis*) dins el mòdul de simulació i predicció d'esllavissades (Fig. 5 b) el qual és contribució directa del Institut Cartogràfic i Geològic de Catalunya (ICGC). En aquest cas de projecte, l'ICGC pretén incorporar, en la seva metodologia de treball en el sector de la prevenció de riscos geològics, l'anàlisi de pluges com a plugin de QGIS3, com a concepte d'integració de plataformes i d'informació. Aquest plugin, com a exemple de paquets dissenyats per ser executats dins l'ecosistema de QGIS, són una de les eines més efectives per estendre les funcionalitats i interactuar amb QGIS.

L'adaptació del submòdul d'anàlisi de pluges com a plugin al sistema de QGIS pretén agilitzar la mesura, identificació i presa de decisions a l'equip del ICGC que treballa amb la prevenció de riscos d'esllavissades, tal i com ha de permetre servir de base de dades per albergar els paràmetres i factors pluviomètrics que s'estudiïn per ajudar o aportar nous coneixements en la determinació de llindars pluviomètrics i climàtics en zones locals, i d'aquesta manera reduir la falta de precisió que poden generar alguns dels llindars d'escala regional.



3.2. Anàlisi de requisits

L'anàlisi de requisits s'orienta en reunir en un sol document escrit els requisits de tot el projecte o sistema, en aquest cas del futur plugin a l'entorn SIG de QGIS, així com també a definir el marc de desenvolupament d'aquest, contextualitzar-lo depenent dels objectius o necessitats de l'usuari final i definir sintèticament però de forma clara i ordenada l'abast i restriccions que es presenten a l'hora d'implementar-lo (veure annex 6.1).

L'especificació de requeriments, a part de ser la peça fonamental en un projecte de desenvolupament de software, s'estableixen com un dels principals mecanismes de control per a l'etapa de desenvolupament. A més a més, l'especificació d'aquests es la base que permet verificar si s'aconsegueixen o no els objectius establerts en el projecte (Chaves, 2005).

3.2.1. requisits funcionals

La funcionalitat principal de l'eina de QGIS es permetre l'anàlisi de condicions meteorològiques prèvies i posterior al possible desencadenament d'esllavissades en zones potencialment susceptibles. Tot i que la participació de l'Institut Cartogràfic i Geològic de Catalunya (ICGC) al projecte HEIMDALL va ser contribuir en el disseny i construcció de l'aplicació sota els requeriments dels cossos d'emergències que participen en el projecte europeu, en aquest cas d'estudi, l'aplicació que es pretén construir és una modificació o actualització de l'aplicació HEIMDALL a una versió adaptada al QGIS que està dirigida als propis usuaris locals del ICGC per tal de facilitar les tasques de treball en l'entorn de prevenció i anàlisi de riscos geològics, desenvolupant la modelització i els diferents escenaris de risc per moviments del terreny per tal de donar informació sobre els fets que poden generar, o han generat casos de risc.

Concretament, l'usuari accedeix a la informació a partir de la interfície del plugin, situada dintre del conjunt de QGIS, específicament al menú principal de complements (plugins). La interfície gràfica d'usuari que ofereix el plugin permet a l'usuari introduir els paràmetres necessaris per executar l'anàlisi segons les condicions que es presentin. La funcionalitat de l'anàlisi generat pel plugin es veu condicionada per dos paràmetres principals: Les (1) coordenades de l'esdeveniment potencial o existent a analitzar i la (2) data concreta en què s'ha detectat una perillositat elevada o s'ha produït l'esdeveniment. A partir de la introducció dels paràmetres i l'execució de l'anàlisi el plugin entrega a l'usuari les dades meteorològiques per a la ubicació i data concreta a través de dades gràfiques i informes .csv. El conjunt de gràfiques que contenen informació meteorològica relacionada amb el moment i ubicació concreta de l'esdeveniment proporcionen a l'usuari una perspectiva de la variació dels paràmetres pluviomètrics de la zona, amb un rang de dades de 10 dies endavant i enrere de la data establerta per l'usuari, que poden haver generat situacions de risc. Això permet un estudi de la pluja acumulada al moment del possible esdeveniment tant com la intensitat i la posició de l'esdeveniment comparant-se les seves dades de pluviometria entre diversos llocs per tal d'avaluar-ne el risc.

3.3. Disseny funcional

El plugin Local Rainfall Analysis ha de permetre als usuaris i usuàries de l'Institut Cartogràfic i Geològic de Catalunya la descripció i parametrització d'un o varis episodis de pluja concrets en l'espai i el temps, per tal de determinar si aquesta ha set la causa i/o desencadenant d'una esllavissada (Fig. 6). A partir d'una interfície gràfica d'usuari els usuaris han de poder detallar els paràmetres coneguts de l'esdeveniment els quals consisteixen en informació sobre la pluja, tals com la data i la localització de l'esdeveniment, i també han de poder navegar i seleccionar altres paràmetres per complementar la parametrització de l'anàlisi pluviomètric. Aquests altres paràmetres són: l'estació meteorològica amb la qual es volen sol·licitar les dades de pluja als servidors de MeteoCat, selecció del llinar climàtic en funció de l'àrea a estudiar de Catalunya entre d'altres.

El conjunt del plugin i els paràmetres que caracteritzen cada anàlisi en concret han de generar un resultat, en forma de gràfics i informes, que permetin a l'usuari obtenir la informació necessària per a caracteritzar i descriure les condicions pluviomètriques que es van donar durant l'esdeveniment, i com a més important, discriminar si la pluja en el moment de l'esdeveniment va superar el llinar establert i per tant tenir potencial per a desencadenar tal esllavissada.

A diferència del submòdul d'anàlisi de pluges incorporat a la plataforma HEIMDALL, el plugin Local Rainfall Analysis s'utilitzarà únicament per usuàries del ICGC i amb l'objectiu de recopilar dades històriques d'esdeveniment passats per analitzar i descriure els factors desencadenants d'aquests, es a dir, no s'utilitzarà com a eina de gestió de risc per prevenir, detectar i anticipar-se a nous riscos. Aquest plugin ha d'ajudar a aconseguir un extens registre de paràmetres de pluges crítiques per a obtenir-ne un conjunt considerable per a poder-se utilitzar per a la possible definició de valors llinars crítics d'àmbit local.

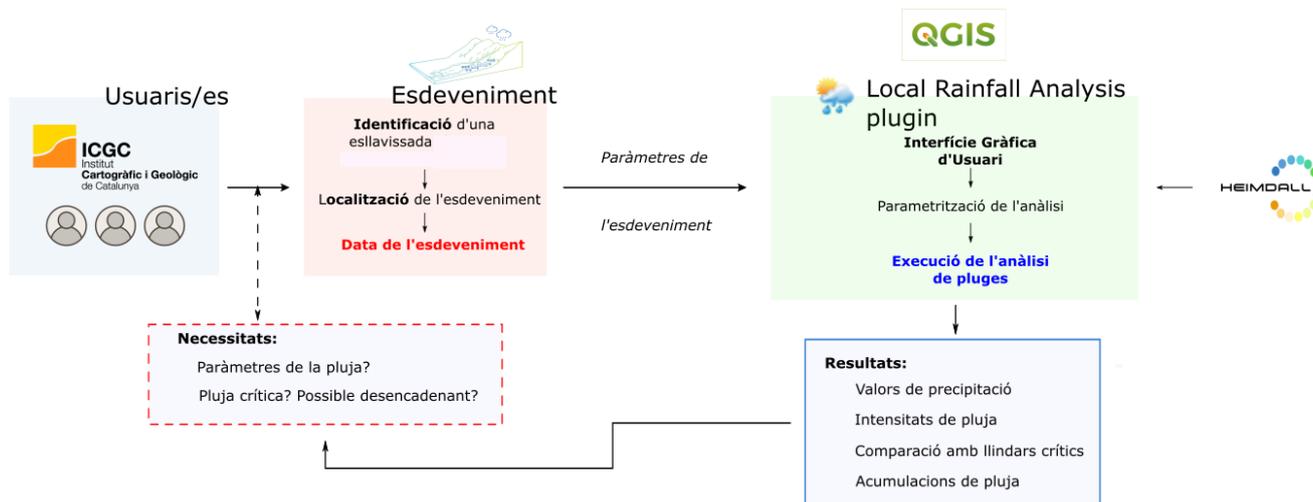


Figura 6 Diagrama de flux del disseny funcional del plugin Local Rainfall Analysis amb els seus components i les relacions entre ells. Elaboració pròpia.

3.4.1. L'ecosistema format per Python i Qgis

Com prèviament s'ha esmentat, el sistema de QGIS esta escrit amb C++, i conté un conjunt propi d'APIs que també estan escrites amb C++. Les APIs de Python s'implementen com a embolcalls de les APIs de C++. Per exemple, hi ha una classe de Python anomenada *QgisInterface* que actua com a embolcall de la mateixa classe de C++ amb el mateix nom. Això significa que tots els mètodes, i classes de C++ són accessibles a través dels embolcalls Python. Significa que, quan accedeixes a les APIs de Python al QGIS, no significa que s'accedeix a l'API de QGIS directament, sinó que, l'embolcall Python connecta el codi proporcionat amb els objectes i mètodes de C++ subjacents. Per tant, la feina de les APIs de Python dins del QGIS es facilitar l'accés al codi C++ amb el qual treballa QGIS (Westra, 2014).

3.4.1.1. Arquitectura de les llibreries PyQGIS

Actualment, QGIS ha arribat al punt en la seva evolució contenint més de 1.200 classes que formen l'aplicació, el nucli i les bases de la interfície gràfica del usuari (*graphical user interface*, GUI), l'extensa majoria de les quals estan habilitades a la interfície de Python, facilitant la utilització d'aquestes a través d'un fàcil accés als seus mètodes i atributs (Sherman, 2014). Operacions bàsiques com ara carregar orígens de dades a mapes, manipular i exportar mapes, etc., estan disponibles directament a PyQGIS el qual és el paquet principal de Python inclòs en la instal·lació per defecte de QGIS (Oxoli, 2017).

Les llibreries de PyQGIS estan estructurades a partir del paquet anomenat *qgis*. Les llibreries són subpaquets dintre d'aquest paquet principal; els següents cinc paquets són els que construeixen la llibreria PyQGIS (Westra, 2014):

qgis.core	Proporciona accés a la funcionalitat SIG principal que s'utilitza a tot el QGIS.
qgis.gui	Defineix una gamma de <i>widgets</i> GUI que poden incloure als propis programes creats per usuaris.
qgis.analysis	Proporciona eines d'anàlisi espacial per analitzar dades de format vectorial i ràster.
qgis.networkanalysis	Proporciona eines per crear i analitzar topologies
qgis.utils	Implementa funcions diverses que permeten treballar amb l'aplicació QGIS mitjançant Python.

Per tal d'utilitzar una de les classes de QGIS dins d'un mòdul o subpaquet, només cal importar-la a la interfície de Python (a la consola de Python que conte QGIS o dins l'arquitectura d'un plugin), i després cridar els mètodes desitjats i/o accedir als seus atributs (Sherman, 2014). Gràcies a aquest fet es possible des de l'entorn Python treballar amb les eines de QGIS, proporcionant l'accés a les propietats dels elements de QGIS des de la consola de Python o a través del plugin.

3.4.1.2. Llibreries PyQt

Qt és un conjunt de llibreries multi-plataforma que implementen APIs d'alt nivell per a poder accedir a molts aspectes dels sistemes d'escriptori moderns. Els plugins de QGIS comparteixen una estructura en comú basada en les llibreries de Python en el marc del Qt, en aquest cas, PyQt (Oxoli, 2017). PyQt és la interfície API Python a Qt, el marc de C++ amb el qual QGIS esta basat

(Sherman, 2014). Degut a que QGIS està construït a partir del marc de Qt, s'utilitza per a aquest projecte PyQt per tal de crear un plugin per tal de proveir tots els elements de la GUI que puguin ser necessaris.

3.4.2. Disseny de l'entorn de desenvolupament de Python

Aquest apartat presenta una descripció detallada dels components necessaris per a construir un ambient local de desenvolupament (utilitzant un ambient virtual) necessari i favorable per al llenguatge Python. Aquest ambient es tradueix a la preparació, instal·lació i optimització dels diversos programes i llibreries a la màquina local per Python, i que aquesta permeti desenvolupar i utilitzar els scripts i plugins creats a partir de PyQGIS.

Pel que fa en l'àmbit d'aquest treball, l'ambient de desenvolupament objectiu consta de:

- a. Python 3.7
- b. Spider 3.3.6 com a Entorn de Desenvolupament Integrat (*Integrated Development Environment*, IDE).
- c. Anaconda 1.9.2 com a plataforma de distribució de Python.
- d. PyQT5
- e. Qt Creator i Qt Designer

A partir d'aquests components, és possible crear l'entorn Python necessari per tal de procedir amb l'anàlisi del codi existent i realitzar les modificacions necessàries prèvies a la implementació i creació del plugin amb QGIS. Posteriorment s'esmentaran els requeriments de QGIS i mòduls utilitzats per la creació del plugin. En aquest apartat, es limitarà a descriure detalladament la funcionalitat i propietats de cada component per tal de descriure i donar una fàcil interpretació al concepte d'ambient de desenvolupament de Python.

a. Python

Depenent del possible sistema operatiu que es pretengui treballar, pot ser que Python 3 estigui ja instal·lat per defecte. Per defecte, el sistema operatiu de Mac OS X incorpora la versió de Python 3. En el cas de Windows, també incorpora la versió de Python 3 accessible a través de la seva aplicació de comandament "PowerShell". En aquest cas d'estudi, s'utilitza Python 3.7 en el sistema operatiu de Windows 10, prèviament instal·lat. En el cas d'utilitzar Windows, la instal·lació de QGIS utilitzant l'instal·lador d'OSGeo4W aporta tot l'ambient necessari per utilitzar PyGIS, el que demostra ser un gran avantatge.

OSGeo4W és un repositori binari que conte el conjunt de software geoespacial del projecte OSGeo per a entorns Windows. QGIS i totes les seves dependències (software de suport), Python i GRASS, SAGA GIS i GDAL estan inclosos en el repositori (Ahmed *et al.*, 2018). Per aquest projecte s'utilitza la versió 3.7 de Python del repositori de Python (www.python.org).

b. Spider com a Entorn de Desenvolupament Integrat

Quan es tracti d'escriure i executar codis amb Python, concretament PyQGIS, un ampli debat pot ser obert a l'hora d'escollir un software o editor de text per interpretar el codi. Tot i haver generat molta controvèrsia alhora d'escollir, al final es resumeix com una elecció segons les preferències personals (Sherman, 2014).

Spyder es un potent ambient científic escrit en Python originalment pensat i destinat per científics, enginyers i analistes de dades amb interès de treballar amb Python. En aquest projecte es selecciona Spider, concretament Spyder 3.3.6 per la versalitat i combinació de funcionalitats, a destacar, a part de que proporcionin subratllar la sintaxi i identacions són: la fàcil i interactiva exploració del codi a través del explorador de variabls i les eines de debugging que inclouen les built-in característiques, i per la facilitat d'estendre les capacitats a través del seu sistema de plugin i API, permeten la utilització de la llibreria extesa de PyQT5 (SPYDER, 2018).

c. Anaconda com a plataforma de distribució de Python

Anaconda s'utilitza coma plataforma de distribució de Python, en aquest cas, per facilitar la creació i optimització de l'ambient de desenvolupament. Anaconda 2020.02 permet i ofereix mecanismes per construir, distribuir, instal·lar i actualitzar de manera cross-platform, el conjunt d'ambients i els paquets de treball a partir del seu repositori (Anaconda, 2020). Aquests son necessaris per estendre les funcionalitats del codi Python, i per tal de poder treballar amb els mòduls de HEIMDALL es necessari preparar l'ambient de treball amb els paquets necessaris que demani per tal de poder utilitzar-lo.

d. PyQT5

PyQT és la interfície Python de la API de Qt, el marc de C++ amb el qual QGIS esta basat. Degut a que QGIS esta construït mitjançant el marc de Qt, en aquest pojecte s'utilitza PyQT per tal de proporcionar al plugin tots els elements de la GUI que són necessaris.

Depenent del sistema operatiu que s'utilitzi, pot ser que PyQT ja estigui instal·lat. En aquest cas, si s'ha utilitzat l'instal·lador de OSGeo4W al Windows, la instal·lació de Qt i PyQt ja s'instal·len per defecte com a part de la instal·lació de QGIS (Sherman, 2014).

e. Qt creator

Qt Creator 4.11.2 proporciona les eines i marc de treball necessàries per tal a crear la interfície de la GUI. Qt Creator és una IDE destinada a desenvolupadors d'aplicacions multi-plataforma. Aquesta permet treballar tots els components que configuren la interfície d'usuari del plugin.

3.4.2.1. Instal·lació del programari i implementació de l'entorn de treball

Degut a que la majoria de programes que s'utilitzen en aquest projecte són d'accés gratuït i/o programari lliure, es descarreguen des de les pàgines webs oficials corresponents. Tot i això, depenent del Sistema Operatiu que s'utilitzi pot ser que algun programari estigui ja instal·lat en la versió de sèrie (p.ex: Python).

La importància de la implementació d'un entorn de treball recau en la possibilitat d'accedir a una API (QGIS/PyQGIS, Python i Qt/PyQt) fora de QGIS, Qt Creator o qualsevol altre aplicació amb una API pròpia. La implementació de l'entorn de treball també és necessària si es vol integrar la IDE (Spyder) amb el QGIS.

3.4.3. Anàlisi del codi existent

Els primers passos d'aquest projecte, un cop establert l'entorn de treball Python, han consistit en l'anàlisi del codi existent del submòdul d'anàlisi de pluges pertanyent al mòdul de simulació d'esllavissades de HEIMDALL. L'objectiu d'aquest apartat és analitzar l'estructura Python que dona forma i funció a aquest submòdul analític amb l'objectiu de donar una visió integrada i sintètica de l'arquitectura i funcionalitat d'aquest i els seus constituents i així ressaltar el seu interès i implicació en la subseqüent elaboració del plugin. Consegüentment, aquest anàlisi permetrà entendre els requisits del codi Python que el construeix; com ara els paràmetres i valors d'entrada, el conjunt de llibreries i mòduls externs Python necessaris i els resultats que genera. Per tant, aquesta fase ha de servir per entendre la lògica del codi Python integrat en el submòdul de *Local Rainfall Analysis* per posteriorment planificar l'adaptació i modificacions necessàries per l'entorn de QGIS.

L'execució d'aquest anàlisi s'ha dut a terme a partir de la IDE Spyder la qual permet visualitzar, estructurar i executar el codi Python provinent del mòdul d'anàlisi de pluges (*Local Rainfall Analysis*). S'utilitza una plataforma IDE degut al conjunt de funcions integrades que conté les quals faciliten la interpretació del codi Python. D'aquesta manera es possible dissenyar la posterior fase de modificacions i/o adaptació del codi Python per al desenvolupament del plugin.

3.4.3.1. Aspectes generals

Com bé s'ha expressat en capítols anteriors, l'objectiu del submòdul d'anàlisi de pluges és analitzar dades meteorològiques (de la xarxa d'estacions meteorològiques de Meteocat) pre- i post-esdeveniment amb l'objectiu d'enregistrar els paràmetres que caracteritzin un o més episodis de pluja amb possibilitat de desencadenar condicions crítiques i preveure si el risc incrementarà o es reduirà els dies posteriors a l'esdeveniment. El submòdul està constituït per dos arxius i/o mòduls Python (.py): *meteocat_MTIG.py* i *localrainfallanalysis_MTIG.py*.

Aquests arxius .py són els dos submòduls Python que construeixen el submòdul de Simulació i Modelització d'esllavissades del projecte HEIMDALL. Ambdós són dos mòduls format Python (.py) que malgrat la segregació són dependents l'un de l'altre. El submòdul *meteocat_MTIG.py* (Fig. 7) genera l'accés i obtenció a les dades dels servidors de Meteocat i d'aquesta manera poder obtenir la precipitació corresponent al període d'anàlisi especificat per l'usuari. Aquest mòdul no només recupera les dades a partir d'unes condicions de cerca concretes (data de l'esdeveniment a estudiar i coordenades de la localització de l'esdeveniment) sinó que elabora un procés de tractament de les dades obtingudes per prepara-les pel mòdul *localrainfallanalysis_MTIG.py*. A més a més, si es considera necessari, el submòdul genera un arxiu format .csv o .json amb el conjunt de dades per ser exportat. El sub-mòdul *localrainfallanalysis_MTIG.py* recupera les dades prèviament tractades i estableix els càlculs necessaris per a genera el conjunt d'anàlisi de pluja (intensitat, acumulació, etc.) i generar la representació gràfica necessària per presentar els resultats.

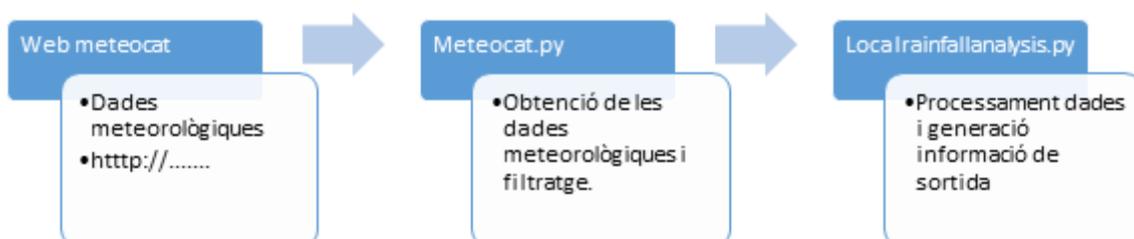


Figura 7 Model esquemàtic i sintètic del flux de treball dels submòduls de l'anàlisi de pluges.

Per tal de visualitzar ràpidament i/o obtenir una visió general de l'estructura del codi, un dels conceptes més importants en aquesta fase és entendre la dependència del conjunt de funcions que construeixen el codi Python (entenent la naturalesa del llenguatge i les seves característiques) que criden a altres funcions i/o classes d'objectes que són els responsables de l'elaboració i confecció del conjunt de dades de les quals s'esperen els resultats.

3.4.3.2. Submòdul *meteocat_MTIG.py*

El primer submòdul, *meteocat_MTIG.py* (Fig. 8 i annex 6.2.1) consisteix en un mòdul Python amb el codi necessari per a obtenir i processar les dades dels servidors de *MeteoCat*, per a un dia i ubicació concrets, per a una determinada estació meteorològica. La Fig. 9 esquematitza la dependència de les funcions i reflexa l'arquitectura funcional del primer submòdul.

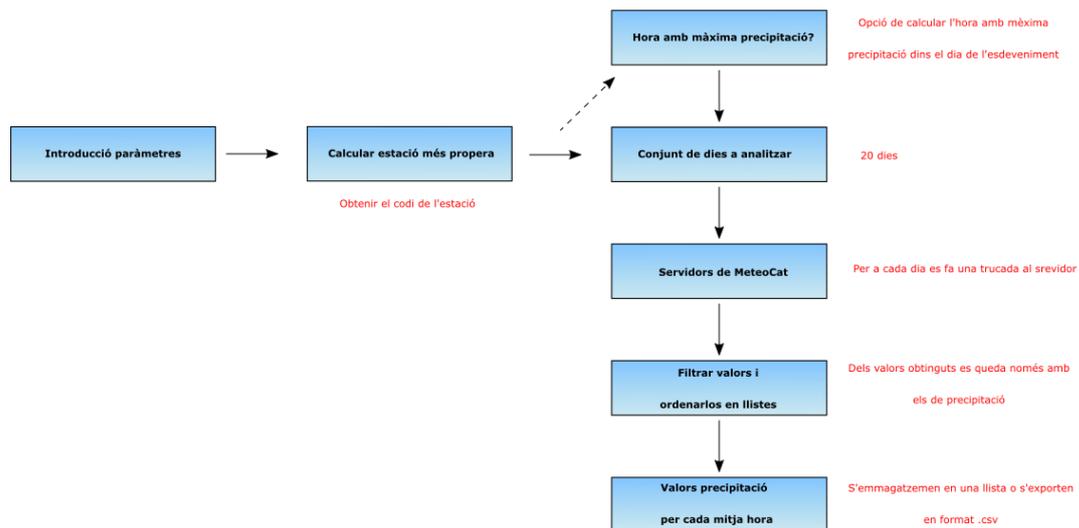


Figura 8 Diagrama de flux del primer submòdul *meteocat_MTIG.py* del submòdul d'anàlisi de pluges de la plataforma HEIMDALL. Elaboració pròpia.

Diagrama de crides de les funcions del mòdul `meteocat_MTIG.py`

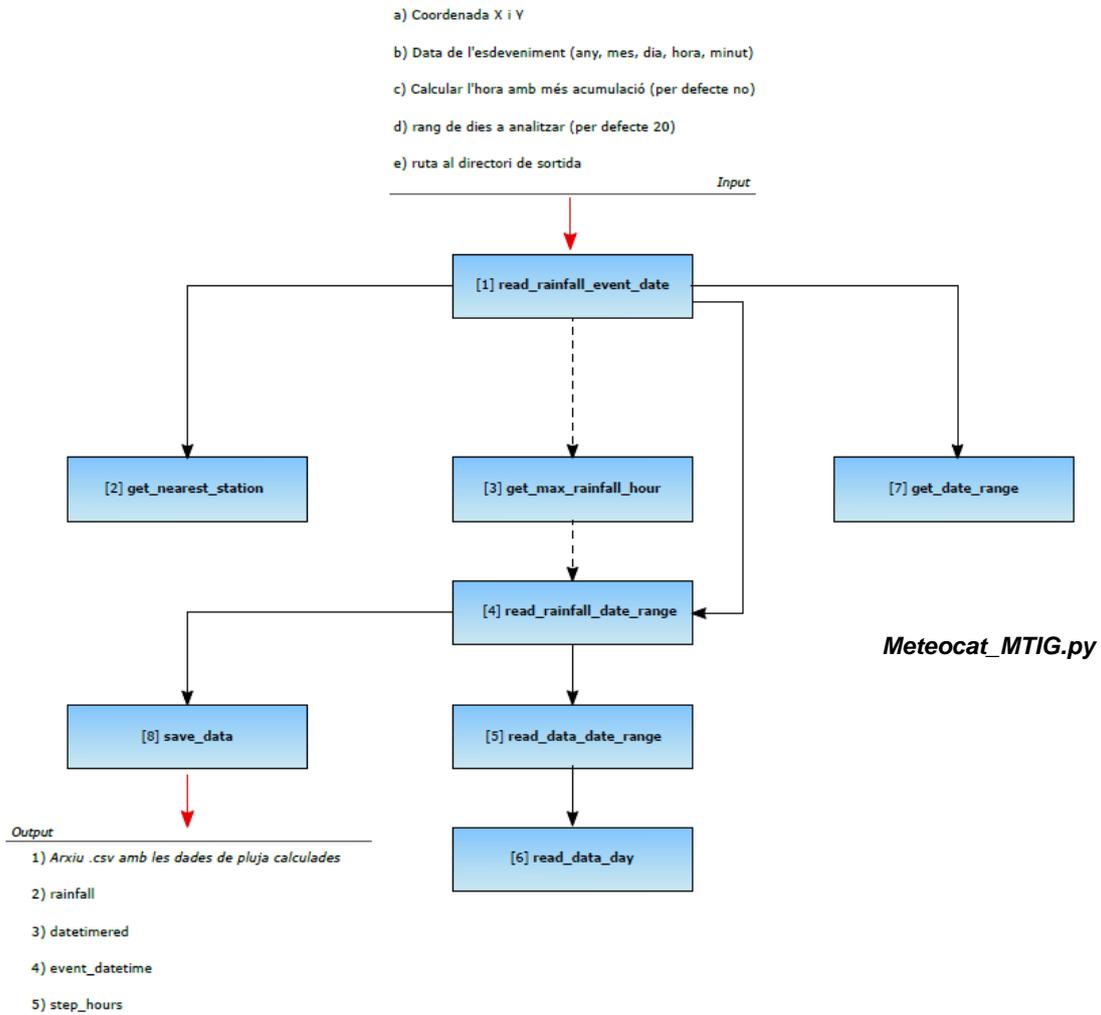


Figura 9 Diagrama de crides de les funcions del primer submòdul `meteocat_MTIG.py` del submòdul d'anàlisi de pluges de la plataforma HEIMDALL. En aquest diagrama s'especifica la dependència entre funcions del mòdul Python i els seus paràmetres d'entrada i de sortida. Elaboració pròpia.

3.4.3.3. Submòdul localrainfallanalysis_MTIG.py

Localrainfallanalysis_MTIG.py, com a submòdul complementari al meteocat_MTIG.py es caracteritza per una complexitat més elevada en termes d'estructura i interdependència de crides a les funcions. Com bé meteocat_MTIG.py presentava una estructura més sintètica i lineal, aquest nou submòdul està segregat en 4 subgrups de funcions (veure annex 6.2.2). El submòdul s'estructura i les seves funcions s'agrupen en (Fig. 10): a) funcions principals que realitzen les crides a la resta, b) lectura de paràmetres i dades, c) càlculs i d) gràfics. Aquesta segregació facilita la comprensió del codi i l'estructuració d'aquest.

L'apartat o subgrup de funcions principals a) es caracteritza, com en el submòdul anterior, per ser les funcions on s'introdueixen els paràmetres d'entrada i desencadenen les crides a la resta de funcions organitzades en els altres 3 subgrups. El subgrup b) es caracteritza per la crida al submòdul meteocat_MTIG.py. Si bé el primer submòdul pot executar-se independentment al segon, amb els valors d'entrada i sortida definits anteriorment, el segon submòdul permet entrar de nou els valors d'entrada necessaris (inclosos els de meteocat_MTIG.py) i el propi submòdul localrainfallanalysis_MTIG.py fa la crida i execució d'ell i recupera els seus valors de sortida, els quals són els valors d'entrada clau (event_datetime, rainfall, datetimered i step_hours) per l'execució del segon mòdul.

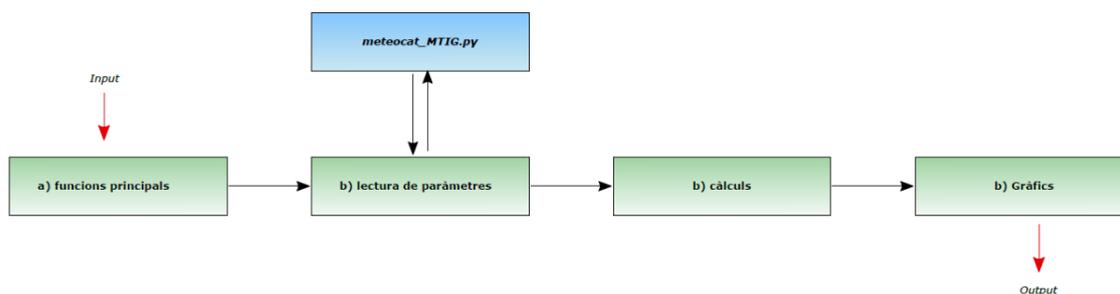


Figura 10 Diagrama esquemàtic dels quatre blocs de funcions del segon submòdul Localrainfallanalysis_MTIG.py i la seva dependència i/o relacions entre ells. Elaboració pròpia.

Tot i això, si bé l'apartat b) de lectura de paràmetres permet executar el mòdul meteocat_MTIG.py per recuperar les dades de pluja, l'apartat b) de localrainfallanalysis_MTIG.py està preparat per llegir les dades de pluja provinents de formats externs de la plataforma HEIMDALL, format .json, i/o .csv. En aquest cas, i per l'interès de les fases futures de construcció del plugin, es conserva la lectura de paràmetres a partir de la crida al primer submòdul i per tant la resta s'obviarà en aquest apartat per alleugerir la documentació de l'estructura del submòdul. Tot i això, en termes de treball futur, és possible que es recuperi la funcionalitat de llegir les dades en format .csv, per tal que el plugin permeti introduir dades d'una base de dades. A més a més, i com en el cas del primer submòdul, aquest també permet generar resultats en format d'arxius .json, aquests també s'han obviat, ja que no es un resultat desitjat a produir amb el plugin (es pretén obtenir tots els resultats en forma de taules de dades format .csv i gràfics en format .png)

3.5. Implementació

Tot i existir diverses metodologies de desenvolupament, pel que fa a aquest projecte, el flux de treball i la metodologia de desenvolupament tant de les fases inicials de modificacions i adaptació del codi com a la creació del plugin i la seva instal·lació s'han donat a partir de la següent metodologia documentada a la Fig. 11:

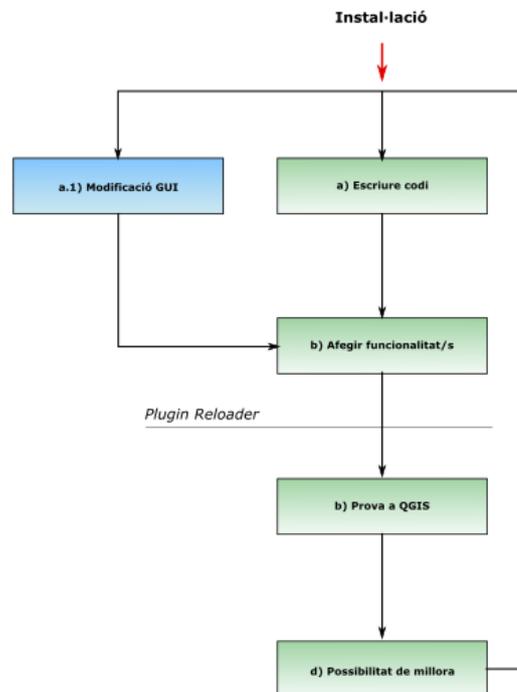


Figura 11 Esquema de la metodologia de desenvolupament seguida en aquest projecte.

3.5.1. Modificacions al codi

En aquesta fase del projecte s'explica detalladament totes les intervencions als dos submòduls (meteocat_MTIG.py i localrainfallanalysis_MTIG.py). Es pretén donar una explicació detallada i estructurada que permeti documentar les modificacions aplicades i el motiu i la lògica de cada intervenció en el codi. El conjunt de modificacions i intervencions en els mòduls Python s'han fet per adaptar els mòduls a l'estructura d'un plugin de QGIS, i d'aquesta manera, crear l'aplicació amb els objectius de funcionalitat detallats en la fase de requisits funcionals d'aquesta memòria.

Tot i que el procés d'adaptació dels submòduls Python a formar part d'un context més específic i aplicat com és un plugin per la plataforma QGIS, aquest ha involucrat moltes fases de modificació i de prova, en aquesta memòria es sintetitzen en un format que expliqui de forma sintètica el motiu de la modificació, entenent el motiu com a la necessitat d'afegir una funcionalitat nova prèviament no existent en el submòdul o ara bé, la modificació d'una estructura preexistent per adaptar-la al format del plugin; i la lògica de la qual es parteix per tal d'aplicar la modificació

corresponent a l'objectiu plantejat inicialment. Tot aquest conjunt de modificacions han estat ideades i aplicades un cop ha finalitzat la fase d'anàlisi del codi existent.

Prèviament a estructurar el conjunt de modificacions cal introduir la teoria que fa referència a l'estructura interna d'un plugin, concretament en el cas dels plugins creats explícitament pel QGIS, ja que, dins el context estructural i lògic del plugin entren en joc altres elements a part del codi dels sub-mòdus.

3.5.1.1. Arquitectura del plugin de QGIS:

En aquest apartat s'aprofundirà en l'estructura física bàsica d'un plugin de QGIS concentrant-se en l'arquitectura d'aquest i com es reparteixen els elements estructurals i la seva funció dins del plugin i l'entorn de QGIS.

Els plugins de QGIS estan comprimits dins d'un arxiu *.zip*. el qual consisteix en un directori de nivell superior el qual conte els arxius del plugin i cada un dels subdirectoris. Per a un plugin bàsic, una llista del directori principal és la següent (FIG!!!):

```
PYTHON_PLUGINS_PATH/
MyPlugin/
  __init__.py --> *requerit*
  mainPlugin.py --> *requerit*
  mainPlugindialog.py
  mainPlugin_dialog_base.ui
  metadata.txt --> *requerit*
  resources.qrc --> *probablement útil*
  resources.py --> *versió compilada, probablement útil*
  form.ui --> * probablement útil *
  form.py --> * versió compilada, probablement útil *
```

MyPlugin/

Directori de nivell superior que conte el conjunt de subdirectoris del plugin (FIG!!)

__init__.py

Mòdul Python que conté un sol mètode anomenat *ClassFactory()* que inicialitza la classe del plugin i informa a QGIS de l'activació del plugin.

Les classes proporcionen una via per agrupar dades i funcionalitats juntes. Una classe defineix un objecte, el qual permet crear noves instàncies per aquest. Cada instància de classe pot tenir atributs i mètodes (definites per la seva classe) adjunts per modificar l'estat de l'objecte.

L'arxiu *__init__.py* es requereix pel sistema d'importació de Python, i QGIS explícitament requereix que aquest arxiu inclogui el mètode *classfactory()* el qual és cridat quan el plugin es carregat al QGIS. Un cop cridat, l'arxiu *__init__.py* ha de fer referència al arxiu principal del plugin (*local_rainfall_analysis.py*) i la classe definida pel plugin:

```
def classFactory(iface): # pylint: disable=invalid-name
    """Load LocalRainfallAnalysis_MTIG class from file LocalRainfallAnalysis_MTIG.

    :param iface: A QGIS interface instance.
    :type iface: QgsInterface
    """
    #
    from .local_rainfall_analysis_mtig import LocalRainfallAnalysis_MTIG
    return LocalRainfallAnalysis_MTIG(iface)
```

metadata.txt

Arxiu format *.txt* que conté informació sobre el plugin, incloent el nom del plugin, la descripció, la versió del plugin, la icona i la versió del QGIS mínima requerida.

```
# This file contains metadata for your plugin.
```

```
# This file should be included when you package your plugin.# Mandatory items:
```

```
[general]
name=Local Rainfall Analysis MTIG
qgisMinimumVersion=3.0
description=This plugin creates a report from a day
rainfall-event from the nearest metereological station inside Catalonia
version=0.1
author=Marc Solanich Faja
email=marcsolanichfaja@gmail.com
```

resources.qrc

Descriu els recursos utilitzats pel plugin i el conjunt gràfic de la Interfície Gràfica d'Usuari (GUI).

resources_rc.py

Arxiu Python generat per el mòdul *resources.py* pel compilador de recursos *pyrcc5* (veure capítol següent).

mainPlugin.py

Arxiu Python amb la implementació principal del plugin que suporta la instal·lació i la desinstal·lació al repositori de plugins instal·lats al QGIS, i l'execució de les funcionalitats del plugin.

Aquest arxiu està implícitament lligat en l'evolució del projecte, ja que albergarà el codi dels submòduls *meteocat_MTIG.py* i *localrainfallanalysis_MTIG.py* i totes les modificacions, incloent-hi les funcionalitats de la GUI i els elements que apareixen en ella. El conjunt de classes, objectes i mètodes tant de les APIs de PyQGIS i PyQt es criden des d'aquest arxiu igual que totes les llibreries Python necessàries per a l'execució del conjunt de funcions.

mainPlugindialog.py

És l'arxiu que executa la comunicació entre el plugin i la seva GUI. Aquest mòdul carrega l'arxiu *.ui* i d'aquesta manera PyQt pot afegir els elements a la GUI provinents del Qt Designer (veure en el següent capítol).

```
import os

from qgis.PyQt import uic
from qgis.PyQt import QtWidgets

# This loads your .ui file so that PyQt can populate your plugin with the elements from Qt Designer
FORM_CLASS, _ = uic.loadUiType(os.path.join(
    os.path.dirname(__file__), 'local_rainfall_analysis_mtig_dialog_base.ui'))

class LocalRainfallAnalysisDialog(QtWidgets.QDialog, FORM_CLASS):
    def __init__(self, parent=None):
        """Constructor."""
        super(LocalRainfallAnalysisDialog, self).__init__(parent)
        # Set up the user interface from Designer through FORM_CLASS.
        # After self.setupUi() you can access any designer object by doing
        # self.<objectname>, and you can use autoconnect slots - see
        # http://qt-project.org/doc/qt-4.8/designer-using-a-ui-file.html
        # #widgets-and-dialogs-with-auto-connect
        self.setupUi(self)
```

mainPlugin_dialog_base.ui

Arxiu creat per Qt Designer amb la informació pertanyent a la interfície de la GUI. Conté la definició i posició de tots els elements PyQt que formaran la interfície gràfica del plugin al QGIS.

La creació i instal·lació d'aquest conjunt estructural que constitueix físicament el plugin s'explicarà en el capítol 2.5. En canvi, les modificacions aplicades al codi dels submòduls *meteocat_MTIG.py* i *localrainfallanalysis_MTIG* es poden ja documentar, ja que s'estableix en quin entorn de l'estructura física del plugin es troben i com interaccionen amb el conjunt estructural del plugin.

3.5.1.2. Documentació de les modificacions al codi

En aquest capítol es documenta detalladament tant les modificacions al codi com l'addició d'elements nous al codi Python del submòdul *Local Rainfall Analysis* de HEIMDALL amb l'objectiu de preparar el codi per la plataforma de QGIS.

En el capítol 2.5.2, es parlarà de la fase de construcció de l'estructura del plugin i els seus directoris i arxius bases a partir de la creació d'una plantilla de plugin o *template*, com de les parts posteriors de la construcció del plugin que són les fases de prova, la personalització del plugin i la creació de la GUI mitjançant el Qt Designer i la posterior compilació, i instal·lació dels arxius del plugin al QGIS.

Les modificacions del codi existent, com bé podria donar-se el cas d'escriure el codi de nou, es un apartat clau dins de la construcció d'un plugin. Aquest es un pas avançat en la creació d'un plugin, on els passos anteriors a aquest són: primer la identificació d'una necessitat o funcionalitat que no es té i que a través de la creació d'un plugin es pretén cobrir; en aquest cas, incloure els submòduls de HEIMDALL per integrar-los en el sistema QGIS i poder utilitzar els elements i eines geoespacionals conjuntament amb l'anàlisi de pluges del codi. Un cop es planteja la idea la qual el plugin ha de cobrir, es genera l'estructura inicial que representarà físicament el plugin i alhora contindrà la seva informació (capítol anterior). Un cop l'estructura està construïda

sòlidament, s'introdueix el codi Python nou, o com es el cas d'aquest projecte, una adaptació del codi Python existent:

Com s'ha comentat anteriorment, l'arxiu `local_rainfall_analysis.py` es el principal element estructural del plugin (Fig. 12) i el responsable d'executar la totalitat del codi corresponent a l'anàlisi de pluges generat pels dos mòduls. Per tal d'entendre la funcionalitat, dependència de les funcions contingudes en ell i l'arquitectura, es important subratllar les parts que el componen (Fig. 12). Les úniques funcions imprescindibles son, a) `__init__` la qual dona accés a la interfície de QGIS, b) `initGui()` es crida quan el plugin es carregat, i c) `unload()` que es crida quan es fa la desinstal·lació el plugin.

Per tal d'estructurar bé aquesta part de modificacions del codi, es parlarà de les a) modificacions intrínseques al codi dels dos submòduls, per tal d'adaptar-lo a l'entorn de QGIS, b) creació de les funcions per afegir valors i recuperar-ne de la GUI i 3) altres funcionalitats noves. L'algoritme que es construeix per l'anàlisi de pluges a partir dels dos submòduls es resumeix a la Fig. 13 i es detalla en l'apartat següent.

El conjunt de modificacions del codi existent i la creació de noves funcionalitats formaran part de d) `run()` la qual, un cop validada, desencadena tota la funcionalitat del codi, i per tant l'anàlisi de pluges que generen els submòduls de HEIMDALL. Tot i això, hi ha funcionalitats noves que no s'inclouran dins cap de les funcions ja definides sinó que formaran part de la classe creada pel propi plugin, com a mètodes d'aquesta.

```

from PyQt4.QtCore import *
from PyQt4.QtGui import *
from qgis.core import *

# initialize Qt resources from file resources.py
import resources

class TestPlugin:

    def __init__(self, iface):
        # save reference to the QGIS interface
        self.iface = iface

    def initGui(self):
        # create action that will start plugin configuration
        self.action = QAction(QIcon(":/plugins/testplug/icon.png"), "Test plugin", self.iface.mainWindow())
        self.action.setObjectName("testAction")
        self.action.setWhatsThis("Configuration for test plugin")
        self.action.setStatusTip("This is status tip")
        QObject.connect(self.action, SIGNAL("triggered()"), self.run)

        # add toolbar button and menu item
        self.iface.addToolBarIcon(self.action)
        self.iface.addPluginToMenu("&Test plugins", self.action)

        # connect to signal renderComplete which is emitted when canvas
        # rendering is done
        QObject.connect(self.iface.mapCanvas(), SIGNAL("renderComplete(QPainter *)"), self.renderTest)

    def unload(self):
        # remove the plugin menu item and icon
        self.iface.removePluginMenu("&Test plugins", self.action)
        self.iface.removeToolBarIcon(self.action)

        # disconnect from signal of the canvas
        QObject.disconnect(self.iface.mapCanvas(), SIGNAL("renderComplete(QPainter *)"), self.renderTest)

    def run(self):
        # create and show a configuration dialog or something similar
        print "TestPlugin: run called!"

    def renderTest(self, painter):
        # use painter for drawing to map canvas
        print "TestPlugin: renderTest called!"

```

Figura 12 Model bàsic o plantilla de l'estructura de l'arxiu Python principal del plugin. L'arxiu conté la definició de la classe pròpia del plugin i els seus mètodes complementaris.

Diagrama de flux de l'algoritme de l'anàlisi de pluges del plugin Local Rainfall Analysis

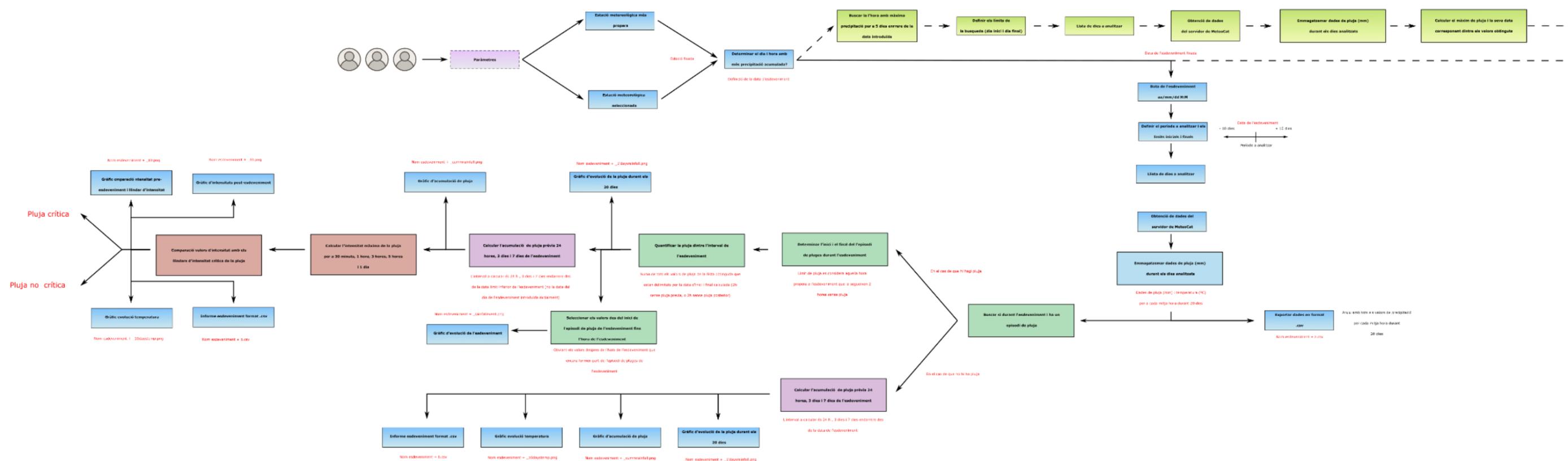


Figura 13 Diagrama de flux del conjunt estructural del plugin Local Rainfall Analysis i el seu arxiu Python principal local_rainfall_analysis.py. En l'esquema es detalla l'ordre i l'arquitectura del algoritme de l'anàlisi de pluges. Elaboració pròpia.

a) Modificacions intrínseques al codi de meteocat_MTIG i localrainfallanalysis_MTIG.py

Les modificacions intrínseques al codi fan referència al conjunt d'intervencions a l'estructura inicial del codi Python per afegir-hi o modificar elements funcionals (objectes, mètodes i/o funcions) per tal d'adaptar-les al marc operacional de la plataforma QGIS. També han de servir per a compatibilitzar la presència de la interfície gràfica d'usuari i/o l'adició de funcionalitats noves des de l'entorn QGIS.

1. Concatenació dels submòduls de HEIMDALL

L'arxiu `local_rainfall_analysis.py` es el principal element estructural del plugin i el responsable d'executar la totalitat del codi corresponent a l'anàlisi de pluges generat pels dos mòduls. La primera modificació intrínseca als submòduls és la concatenació dels dos submòduls en un.

Ja que l'execució del submòdul `localrainfallanalysis_MTIG.py` te implícita la crida i execució del primer submòdul per obtenir els valors de dates i pluges a analitzar, la concatenació de mòduls es fa efectiva posicionant la crida a la funció principal del segon submodul [3] prescindint de [1] i [2] de `localrainfallanalysis_MTIG` com a retorn de la funció principal del primer submòdul `meteocat_MTIG` [1]. Per tant, un cop el plugin fa la crida a la funció `run()`, aquesta crida a:

```
[1] read_rainfall_event_date(event_name, event_x, event_y, event_datetime,
threshold_climatic_area, auto_determine_hour=False, days_range=20, epsg=25831,
out_file=None, out_10days_file=None, out_cumulated_file=None, out_event_file=None,
out_intensity_file=None, out_resume_json_file=None, out_resume_lite_json_file=None)
```

Obté dades de pluja del Meteocat a partir d'una coordenada i un rang de dates

Retorna:

(1) data hora de l'event

(2) una llista de dates

(3) una llista de pluges per cada data

(4) interval en hores de cada valor de pluja

```
(5) local_rainfall_analysis_from_data(station_code, event_name, datetimered, rainfall,
step_hours, threshold_climatic_area, event_datetime, min_distance_str, out_10days_file,
out_10daystemp, out_cumulated_file, out_event_file, out_intensity_file, out_file_2)
```

Per tant aquesta és la dependència de les dues principals funcions del dos submòduls un cop concatenats dins la funció de `run()` del plugin. A partir d'aquí el plugin es capaç d'executar el conjunt de funcions dels dos submòduls per obtenir els paràmetres de Meteocat i generar els l'anàlisi de pluges.

2. Selecció d'estació meteorològica

En el submòdul `meteocat_MTIG` l'estació meteorològica amb la qual es pretén executar l'anàlisi per un esdeveniment en concret es calcula a partir de la diferència mínima de distància entre el punt de l'esdeveniment i l'estació més propera. És interessant per al plugin que l'usuari pugui seleccionar manualment la estació amb interès per a un esdeveniment en concret. Per tant, com a paràmetre nou a [1] el qual es recuperarà de la GUI és el codi de l'estació meteorològica que l'usuari prefereixi (Fig. 14). Aquest fet te una lògica pel que fa a una distribució irregular de les precipitacions independentment de les estacions meteorològiques. Es a dir, hi ha la possibilitat que un esdeveniment quedi en una posició mitjana entre dos o més estacions, però físicament el punt que representa l'esdeveniment esta lleugerament més proper a una que les altres

estacions, tot i això, les altres estacions menys properes poden haver enregistrat dades de precipitació molt més elevades que l'estació més propera, fet que pot interessar per l'analítica.

PyQGIS es capaç d'utilitzar la seva API per permetre a Python interaccionar amb les classes i objectes espacials de QGIS. D'aquesta manera, s'incorpora l'arxiu *.shp* que conté les estacions dins el directori general del plugin i les funcions poden accedir a les característiques espacials de les estacions meteorològiques. Per tant, en aquest cas de poder seleccionar automàticament

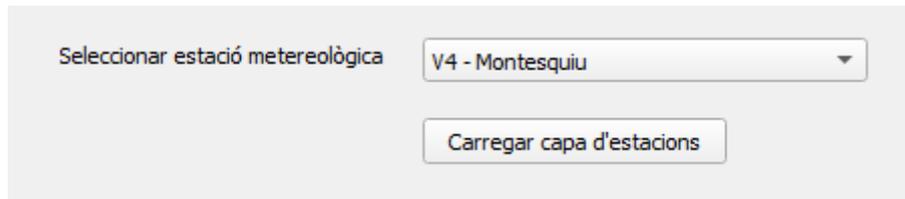


Figura 14 Desplegable de la GUI que permet a l'usuari seleccionar manualment una estació meteorològica en concret o utilitzar la funció de seleccionar l'estació més propera.

l'estació més propera o introduir la estació d'interès, es recupera de l'estació meteorològica en qüestió: a) codi d'estació, b) nom de l'estació, c) distància mínima amb l'esdeveniment i d) diferència altimètrica entre l'esdeveniment i l'estació.

3. Adició del nom de l'esdeveniment, codi d'estació i nom de l'estació

S'afegeix a [1] del mòdul *MeteoCat_MTIG.py* les variables que inclouran el nom de l'esdeveniment, el codi d'estació i el nom de l'estació de la xarxa d'estacions de *MeteoCat*. Els dos primers paràmetres son introduïts des de la GUI (el codi d'estació en el cas que manualment l'usuari en seleccioni una, sinó s'activarà per defecte calcular la més propera). El nom de l'estació es recupera un cop es selecciona una estació, indiferentment de si es selecciona una a partir del seu codi o es calcula automàticament la més propera.

4. Detectar l'hora de màxima precipitació automàticament

Originalment, el submòdul *meteocat_MTIG.py* tenia per defecte el paràmetre d'autodeterminar l'hora amb màxima precipitació del dia de l'esdeveniment (00:00 h a les 23:30h) per tal de re adaptar la data de l'esdeveniment en funció a l'hora amb la màxima acumulació. Tot i això, la GUI del plugin inclou la opció de tria on l'usuari pot escollir si mantenir la data que ha introduït o si que prefereix que [3] del primer submòdul seleccioni una nova data (Fig. 15). Una segona modificació en aquesta funcionalitat de calcular l'hora amb més precipitació es allargar el període de consulta. Es a dir, anteriorment es fa per un període d'inici i final del dia natural que conté la data de l'esdeveniment, i per incrementar la possibilitat que automàticament seleccioni el pic de pluja més elevat, de manera que facilita a l'usuari reduir les incerteses en el cas que no conegui el moment exacte de l'esdeveniment i el pugui associar al primer descriptor responsable de causar una esllavissada, el plugin buscarà per un període de 5 dies naturals des del dia que conte la data de l'esdeveniment. No es farà una cerca de dies endavant ja que, un cop detectada una esllavissada, independentment del coneixement de la possible pluja desencadenant, no es pot atribuir a pluges post-esdeveniment.

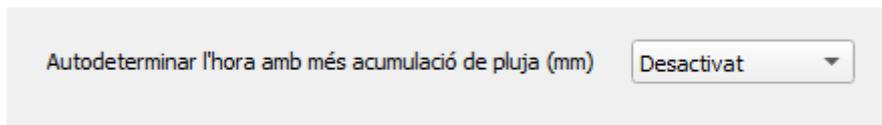


Figura 15 Desplegable inclòs a la GUI que permet a l'usuari activar o desactivar la funció de calcular l'hora amb més precipitació en un interval de 5 dies enrere de la data de l'esdeveniment. Si s'activa aquesta funció, l'algoritme del plugin utilitzarà aquesta nova data per a realitzar l'anàlisi.

5. Evitar error del rang de dates a obtenir si se sobrepassa la data actual

Com que els servidors de MeteoCat no posseeixen les dades de predicció futura de la precipitació el codi del primer submòdul genera un error si es pretén seleccionar una data per l'esdeveniment tan propera a l'actual que no hi ha 10 dies de marge entre les dues dates (el mòdul sol·licita valors per 10 dies abans i després de l'esdeveniment). Per tal d'evitar aquest problema i facilitar a l'usuari del plugin poder escollir dates pròximes a l'actualitat que suposi una diferència de dies inferiors a 10, es determina el diferencial de dies entre l'actualitat i l'esdeveniment mitjançant objectes *datetime*. En el cas que la diferència sigui menor a 10, es redefeixen els valors llindars de les dates que condicionen l'anàlisi i la data del valor llindar de final de període es torna a calcular per a la diferència de dies que hi ha entre l'esdeveniment passat i l'actual. Tot i això, per evitar errors de precisió dins un mateix dia, és a dir, no sempre es calcula per les 00:00 o les 23:30 h sinó que pot ser qualsevol hora entre el rang d'un dia, la data més propera a l'actualitat que es pot seleccionar per dur a terme l'anàlisi és el dia abans del present.

6. Obtenció de dades de temperatura

Aquesta modificació consisteix a obtenir les dades de temperatura per a cada dia del període d'anàlisi aplicant la mateixa metodologia d'obtenció de dades pluviomètriques als servidors de MeteoCat.

L'objectiu d'afegir dades de temperatura mitjana (°C) a l'anàlisi de pluges és disposar a l'usuari d'un segon descriptor, el qual permet discriminar els possibles desencadenants d'esllavissades en períodes on no hi ha precipitació. En el cas l'anàlisi resulti en el fet que no hi ha pluja durant l'esdeveniment, ni en els dies pròxims, es pot associar el factor desencadenant a un pic màxim o mínim de temperatura.

Per tal d'obtenir les dades de temperatura es realitza un procés de captura i tractament de les dades igual que per les dades de precipitació. En [4], [5] i [6] de `meteocat_MTIG.py` es modifiquen els criteris de filtratge de dades del diccionari de dades (`data_dict`) que genera [6] amb les dades crues obtingudes dels servidors de MeteoCat ordenades ja per dates (cada mitja hora durant el període d'anàlisi) (Fig. 16a). Es genera un gràfic per sintetitzar els resultats igual que en el cas de la precipitació (Fig. 16b):

```
a
try:
    temp = [value for (_event_date, _hours, value) in data_dict['TM']]
except KeyError:
    temp = "No hi ha dades de temperatura per l'estacions seleccionada"
```

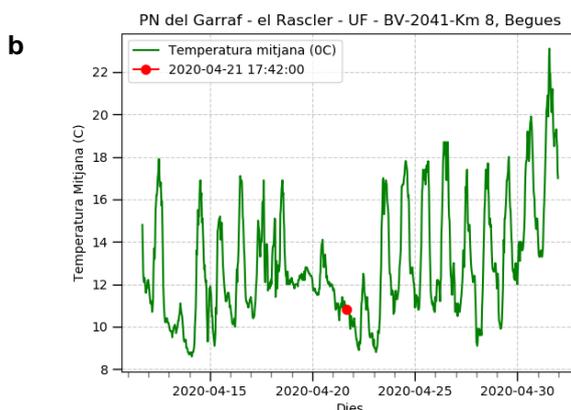


Figura 16 a: Criteri de filtratge de les dades de temperatura que s'obtenen en el procés d'obtenció de dades de precipitació als servidors de MeteoCat. **b:** Gràfic resultant de l'evolució de la temperatura pel període d'anàlisi.

S'ha de contemplar la possibilitat de no obtenir les dades pel fet que certes estacions puntuals de la xarxa de MeteoCat no tenen registre de la temperatura.

7. Càlcul del valor d'acumulació de pluja per dia (mm)

Tot i que el segon mòdul `localrainfallanalysis_MTIG.py` genera un informe de la pluja acumulada 24 hores, 3 dies i 7 dies, és interessant tenir un registre individual de la precipitació acumulada per cada 24 hores durant el període d'anàlisi. Degut al fet que [5] genera una iteració per a cada dia que d'anàlisi registrat en la llista de dies (`day_date`) per tal d'unir els valors de precipitació de `data_day_dict` (segregats per cada dia d'anàlisi) en una llista corresponent al període sencer (1006 entrades). S'afegeix dintre de la iteració la suma dels valors de la llista de `data_day_dict` (valors corresponents per a cada dia individualment) per obtenir el total de precipitació acumulada per a cada dia d'anàlisi.

8. Modificació de l'estil dels gràfics

El submòdul `localrainfallanalysis_MTIG.py` genera un total de 7 gràfics. Les modificacions que en aquest cas s'han aplicat han set per millorar la seva representació i facilitar-ne la comprensió del conjunt de dades que exposen.

- a) Tots els gràfics incorporen el nom de l'esdeveniment que l'usuari estableix.
- b) S'aplica una graella a cada gràfic per facilitar la visualització de les dades.
- c) Traducció al Català del text present als eixos dels gràfics.
- d) Addició de llegenda a tots els gràfics amb la informació corresponent a cada un.

- e) Modificació dels intervals de l'eix X (dates) per evitar que es solapin les dates i no es puguin comprendre bé.

9. Càlcul dels paràmetres de les corbes d'intensitat

Una de les principals funcionalitats que ha d'incorporar la creació del plugin és poder realitzar de forma ràpida i iterativa anàlisis pluviomètrics a tot Catalunya. D'aquesta manera, el plugin ha de servir d'eina de captació de dades de la pluja per tal de ser utilitzades en un futur. Es a dir, interessa que el plugin pugui generar un informe descrivint els paràmetres de les equacions de les intensitats de la pluja durant l'esdeveniment (Fig. 17). A mesura que s'emmagatzemin aquestes dades, podran contribuir en la definició de nous valors llindars de pluges crítiques d'àmbit local, ajudant a reduir l'ambigüitat dels actuals valors llindars d'escala regional.

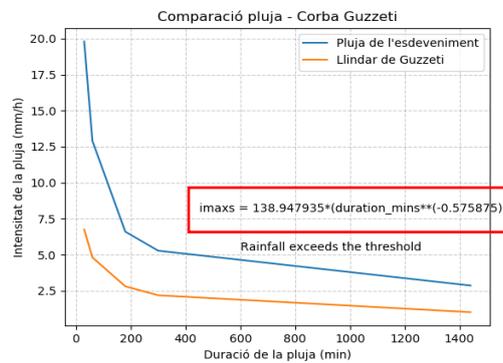


Figura 17 Paràmetres de la corba de l'esdeveniment i l'equació d'aquesta en el gràfic d'intensitat de l'esdeveniment.

10. Generació d'informe de resum de l'esdeveniment

Tot essent el principal resultat de l'anàlisi executat pel plugin els 7 gràfics, és necessari generar un resultat descriptiu del període de l'esdeveniment i els seus paràmetres. Per tant, es genera un arxiu format .csv on es resumeixen de forma sintètica els paràmetres que caracteritzen l'esdeveniment tals com:

- a) Nom de l'esdeveniment
- b) Data de l'esdeveniment i període analitzat
- c) Codi i nom de l'estació
- d) Distància entre l'esdeveniment i l'estació
- e) Precipitació total acumulada durant el període analitzat (mm)
- f) Precipitació total acumulada durant l'esdeveniment (mm)
- g) Intensitats màximes prèvies a l'esdeveniment (mm/h) per a (0,5 i 24 hores , 3 dies i 7 dies)
- h) Equació i paràmetres de la corba d'intensitat de l'esdeveniment.
- i) Pluja acumulada per dia.

Aquest informe ha de facilitar la comprensió i documentació de l'esdeveniment per l'usuari, ja que el primer informe generat pel meteocat_MTIG.py és de difícil manipulació a causa de la seva llargada i contingut de dades.

b) Addició d'elements a l'arxiu localrainfallanalysis.py del plugin

L'addició d'elements a l'arxiu *localrainfallanalysis.py* i el seu codi fan referència al conjunt d'intervencions a l'estructura de l'arxiu principal del plugin que, de la mateixa manera que alberga el codi provinent del submòdul *Local Rainfall Analysis* també és objectiu d'emmagatzemar i executar el conjunt d'elements (objectes, mètodes i/o funcions) que es creen expressament per l'adaptació del mòdul al QGIS i l'ampliació de les seves funcionalitats.

1. Importació de mòduls i llibreries

Les llibreries de Python són una col·lecció de funcions i mètodes que permeten realitzar un gran conjunt d'accions sense haver d'escriure el codi (Fig. 18). Per la viabilitat d'execució del codi del plugin (intrínsec i extrínsec als submòduls anteriors), l'arxiu principal *localrainfallanalysis.py* del plugin importa el següent de llibreries externes:

```

from qgis.PyQt.QtCore import QSettings, QTranslator, QCoreApplication
from qgis.PyQt.QtGui import QIcon
from qgis.PyQt.QtWidgets import QAction, QFileDialog, QMessageBox
from qgis.core import QgsProject, Qgs, QgsVectorLayer, QgsExpression, QgsFeatureRequest, QgsCoordinateReferenceSystem
from qgis.gui import QgsMapToolEmitPoint
from qgis.PyQt import QtGui, QtCore
from qgis.utils import iface
# Initialize Qt resources from file resources.py
from .resources import *
# Import the code for the dialog
from .local_rainfall_analysis_dialog import LocalRainfallAnalysisDialog

import os.path
import os
import datetime
import pytz
import ogr
import urllib
import urllib.error
import urllib.request
import html
import re
import json
import osp
from enum import Enum
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.ticker as ticker
from matplotlib.ticker import MultipleLocator, FormatStrFormatter, AutoMinorLocator
from decimal import Decimal
import numpy
import pylab
import matplotlib.dates as mdates
from scipy import optimize
    
```

Figura 18 Secció del mòdul principal del plugin on es realitzen les incorporacions d'altres llibreries de Python, PyQt i PyQGIS.

A part de la importació de llibreries externes de Python, PyQt i PyQGIS, s'importa l'arxiu de recursos (*resources.qrc*) i el codi necessari per carregar i inicialitzar la GUI del plugin (*LocalRainfallAnalysisDialog*).

2. Invocar funcions a través de la GUI

En qualsevol disseny de la interfície gràfica, el botó d'ordres és el botó més important i el que s'utilitza més sovint. Els botons amb *Guardar*, *Obrir*, *Acceptar*, *Si*, *No*, i *Cancel·lar* i altres, som a subtítol són familiars per a qualsevol usuari d'ordinador. A l'API de PyQT l'objecte de la classe *QPushButton* presenta un botó que quan es fa clic, es pot programar per invocar una determinada funció (Fig. 19):

```
# MSF: Crides de cada pushButton de la GUI a les seves corresponents funcions
self.dlg.pushButton.clicked.connect(self.select_output_file)
self.dlg.pushButton_2.clicked.connect(self.load_layer)
self.dlg.pushButton_3.clicked.connect(self.activate_pointTool)
self.dlg.pushButton_4.clicked.connect(self.desactivate_pointTool)
```

Figura 19 Connexió dels clics en el llenç del QGIS amb el plugin.

self. és una variable interna que fa referència a la instància a la mateixa classe del plugin.

Dins la funció *run()* del arxIU principal *LocalRainfallAnalysis.py* del plugin conté la crida per invocar certes funcions quan l'usuari realitza un clic en la casella corresponent des de la GUI:

- a) Seleccionar una ruta al directori de sortida
- b) Carregar la capa d'estacions meteorològiques
- c) Activar el captador de coordenades
- d) Desactivar el captador de coordenades

El mètode *run(self)* és cridat quan és clicada la icona del plugin des del menú d'eines de QGIS o des del menú de plugIns instal·lats a QGIS.

3. Seleccionar una ruta al directori de sortida

La classe *QFileDialog* de PyQt (Fig. 20) proporciona un diàleg que permet als usuaris seleccionar fitxers o directoris. En aquest cas, l'usuari podrà buscar el directori on vulgui guardar els resultats de l'anàlisi a partir d'un navegador de fitxers que s'emplaçarà a la GUI. La ruta destí generada es construirà a partir de la concatenació de la ruta al fitxer destí seleccionat i el nom de l'esdeveniment. Es a dir, que l'arxiu generat, en aquest cas un nou directori que contindrà els arxius de sortida (veure més endavant) s'anomenarà com el nom de l'esdeveniment. Aquest fet es produirà automàticament quan la GUI recuperi el nom de l'esdeveniment estipulat per l'usuari i l'envii a la classe *QFileDialog*.

```
# MSF: Definir una ruta destí al directori que l'usuari definèixi
def select_output_file(self):
    event_name = self.dlg.lineEdit.text()
    global filename
    filename, _filter = QFileDialog.getSaveFileName(
        self.dlg, "Select output file ", event_name)
    self.dlg.lineEdit_9.setText(filename)
    check_filename = os.path.isfile(filename + "_10daysrainfall.png")

    if check_filename == True:
        QMessageBox.critical(None, "Nom seleccionat està en ús", "Introduir un nom de sortida vàlid")
```

Figura 20 Codi que permet generar una ruta de sortida que l'usuari especifica des de la GUI.

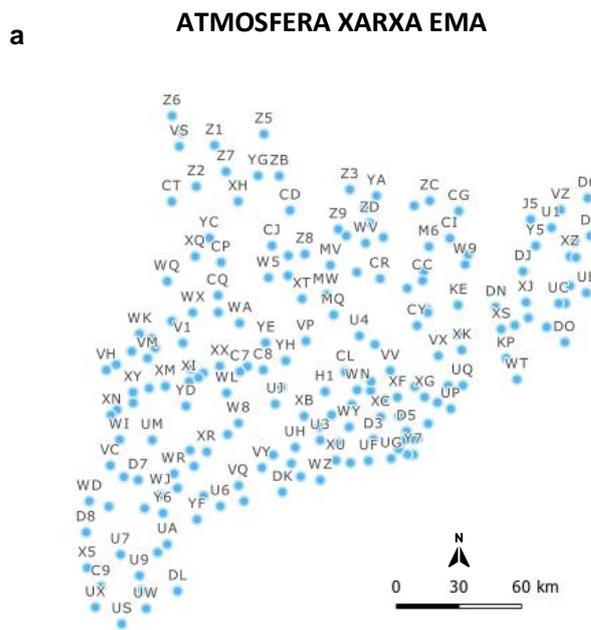
En el cas que el directori seleccionat de sortida contingui un fitxer amb el mateix nom amb el qual es pretén guardar els resultats de l'anàlisi, el plugin retornarà un missatge d'error per a que aquest sigui modificat.

4. Generar un directori nou per a cada anàlisi

Per tal de no sobrecarregar el possible directori seleccionat per l'usuari per a emmagatzemar els resultats d'anàlisis consecutius, al seleccionar una ruta destí per a executar cada vegada l'anàlisi, el plugin genera un directori individual que conté tots els gràfics i informes a partir de la classe *os.makedirs()* del mòdul OS. El mòdul OS proporciona funcions per interactuar amb el sistema operatiu.

5. Càrrega de la capa d'estacions meteorològiques

Es defineix una funció nova dintre de la classe *LocalRainfallAnalysis* que permet carregar la capa format *shapefile* de les estacions meteorològiques al projecte actual de QGIS (Fig. 21a), d'aquesta manera, facilita l'usuari el reconeixement i ubicació espacial de la xarxa d'estacions meteorològiques. PyQGIS es capaç de crear un objecte vectorial de la classe *QgsVectorLayer* que permet obtenir informació de la capa i a través dels seus atributs representar-la al llenç del mapa. A més a més s'afegeix un arxiu d'estil de capa de QGIS (format *.qml*) per donar més integritat als valors de la capa d'estacions meteorològiques (Fig. 21b). Per tant, la si l'usuari prefereix afegir la capa d'estacions meteorològiques al projecte, la GUI fa una crida a [a] (Fig. 19b) i mitjançant la classe *QgsProject*. Aquesta classe encapsula el projecte QGIS, incloent-hi el conjunt de mapes, capes, estils i anotacions. D'aquesta manera *QgsProject* dóna accés a la referència canònica del projecte oberta dins l'aplicació principal de QGIS:



b

```
# NSF: Carregar la capa d'estacions al projecte de QGIS en el cas que l'usuari clicki al pushbutton corresponent
def load_layer(self, state):
    layer_1 = QgsVectorLayer(ruta_capa, "ATMOSFERA_XARXA_EMA", "ogr")
    if not layer_1.isValid():
        print("Layer failed to Load!")
    if layer_1.isValid():
        crs = layer_1.crs()
        crs.createFromId(25831)
        layer_1.setCrs(crs)
        QgsProject.instance().addMapLayer(layer_1)
        layer_1.loadNamedStyle(r"C:\Users\marc\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\Local_rainfall_analysis\ATMOSFERA_XARXA_EMA.qml")
        layer_1.triggerRepaint()
```

Figura 21 a: Capa d'estacions meteorològiques de la xarxa MeteoCat que l'usuari pot carregar des de la GUI. b: codi del mòdul principal del plugin que permet realitzar tal acció des de la GUI.

6. Activar el captador de coordenades

El captador de coordenades es una eina que inclou la GUI que permet a l'usuari definir les coordenades de l'esdeveniment a analitzar d'una manera fàcil i ràpida (Fig. 22). Aquesta eina depèn d'un o més clics al llenç del mapa (*map canvas*) per realitzar la seva funcionalitat, es necessari configurar-lo com una eina de mapa (*map tool*). Com s'ha explicat abans pels botons d'invocació a funcions, la GUI conté dos botons relacionats amb l'eina de coordenades: a) Iniciar captura i b) Finalitzar captura.



Figura 22 Capturador de coordenades disponible a la GUI que permet a l'usuari canviar d'eina de mapa al captador, que un cop es realitza un clic en el mapa, imprimeix les coordenades en les caselles corresponents de la GUI.

Si l'usuari prem el botó d'iniciar captura, invoca a la funció de la pròpia classe del plugin (*self.activate_pointTool*). L'eina de capturar coordenades es converteix en l'eina de mapa activa al QGIS (FIG. 23) i es responsabilitza de suportar tots els clics al llenç del mapa. Aquesta eina romandrà activa fins que l'usuari premi el botó b) o activi una altre eina del mapa.

```
# MSF: Activar el captador de coordenades modificant la eina del canvas de QGIS
def activate_pointTool (self):
    # MSF: Store reference to the map canvas:
    self.canvas = self iface.mapCanvas()
    # MSF: Create the map tool using the canvas reference
    self.pointTool = QgsMapToolEmitPoint(self.canvas)
    # MSF: Set our pointTool to be the current map tool
    self.canvas.setMapTool(self.pointTool)
    # MSF: Connect signal that the canvas was clicked
    self.pointTool.canvasClicked.connect(self.display_point)
```

Figura 23 Codi del mòdul principal del plugin que permet canviar l'eina del llenç de QGIS a un captador de coordenades, que a partir del clic realitzat per l'usuari genera un punt que permet extreure'n les coordenades.

La classe *QgsMapToolEmitPoint* de PyQGIS implementa una eina de mapa que quan és clicada emet un punt que conté les coordenades d'aquest sobre el mapa. En aquestes instàncies s'utilitza la variable interna *self* per a definir el conjunt de funcions a ser emmagatzemades com atributs de la pròpia classe del plugin.

A partir de la creació de l'eina de mapa per seleccionar les coordenades proporcionant-li la referència al llenç del mapa, es vincula cada clic al mapa, es a dit, cada senyal en forma de punt emesa per la classe *QgsMapToolEmitPoint* amb el mètode amb el qual es pretén processar la informació d'aquest senyal (coordenada X i coordenada Y) (Fig. 24).

Les coordenades enregistrades per la classe anterior es materialitzen en forma d'objecte *QgsPointX*. Aquest objecte representa un punt en 2-D amb una posició caracteritzada per una coordenada X i una coordenada Y. El mètode *self.display_point* és responsable de rebre l'objecte *QgsPointXY* contenint les coordenades del clic al mapa. Amb aquesta informació genera dos

cadena de text amb els valors de les coordenades utilitzant els mètodes x() i y(). Finalment, aquests valors són enviats a la GUI per ser utilitzats de paràmetre per l'anàlisi de pluges.

```
# MSF: Obtenir les coordenades del punt generat amb el click i imprimir-les a la GUI
def display_point(self, point, button):
    #self.dlg.hide()
    coordX = float("{}").format(point.x())
    coordY = float("{}").format(point.y())
    X = round(coordX)
    Y = round(coordY)
    self.dlg.lineEdit_2.setText(str(X))
    self.dlg.lineEdit_3.setText(str(Y))
    #self.dlg.show()
```

Figura 24 Obtenició de les coordenades del punt generat per l'eina del captador i impressió d'aquestes a la GUI, aquesta funció també es troba en el codi del mòdul principal del plugin.

7. Comprovació d'utilització del SCR corresponent

El plugin per defecte utilitza el Sistema de Coordenades de Referència EPSG: 25831 – ETRS89/UTM zone 31N. En el cas que l'usuari posseeixi un projecte obert amb un SCR diferent del requerit pel plugin, aquest envia un senyal d'error al iniciar-lo. Tot i això, en el cas que l'usuari no posseeixi cap projecte carregat, i que per tant no hi hagi cap capa o arxiu carregat que defineixi l'SCR del projecte, a l'obrir el plugin, aquest força al canvi de SRC al compatible amb aquest (Fig.25).

```
# MSF: Comprobar que s'utilitza el SCR corresponent (EPSG: 25831)
# MSF: Si no hi ha cap capa carregada, s'estableix el crs EPSG: 25831 per defecte
if iface.activeLayer() is None:
    my_crs = QgsCoordinateReferenceSystem(25831, QgsCoordinateReferenceSystem.EpsgCrsId)
    QgsProject.instance().setCrs(my_crs)
# MSF: En el cas que si hi hagi un projecte anterior amb un crs diferent es dona l'avis:
else:
    vlayer = iface.activeLayer()
    vlayer_dp=vlayer.dataProvider()
    vlayer_crs=vlayer_dp.crs()
    vlayer_crs_str=vlayer_crs.authid()
    vlayer_EPSG_int=int(vlayer_crs_str[5:])
    print("SRC actual:",vlayer_EPSG_int)
    if vlayer_EPSG_int != 25831:
        iface.messageBar().pushMessage("Sistema de Coordenades de Referència incorrecte", "Canviar a EPSG: 25831 - ETRS89/UTM zone 31N ", level=Qgis.Critical)
```

Figura 25 Codi del mòdul principal del plugin que permet identificar si s'utilitza el SRC corresponent al funcionament del plugin, en el cas que no, envia un missatge d'informació perquè aquest es canviï.

8. Addició d'elements a la GUI

Alguns dels paràmetres a introduir a la GUI no poden ser introduïts manualment per l'usuari a partir d'un camp de text sinó que es facilita un desplegable (Fig. 26) que permeti a l'usuari navegar entre les possibles opcions. Aquest és el cas de: a) determinar l'hora automàticament, b) seleccionar estació meteorològica i c) seleccionar el llinar d'àrea climàtica:

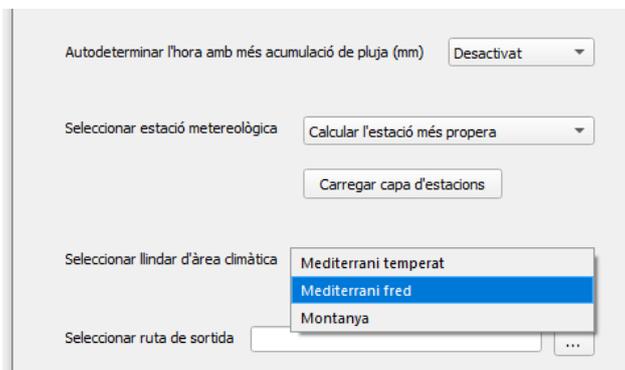


Figura 26 Exemple d'adició d'elements a la GUI en aquest cas, els desplegables que requereixen afegir-hi valors perquè l'usuari pugui seleccionar certs paràmetres per executar l'anàlisi.

Per a aquests desplegables s'ha d'afegir la lògica al plugin per tal de poblar els desplegables amb la informació continguda a la capa *shapefile* d'estacions meteorològiques (Fig. 27).

```
# MSF: Insertar al ComboBox els codis d'estacions de cada feature (estació) del shapefile
iterator_estacions = layer.getFeatures()
estacions_llista = list(iterator_estacions)

self.dlg.comboBox.addItem(["Calcular l'estació més propera"])
#self.dlg.comboBox.addItem([f["CODI_ESTAC"] for f in estacions])
self.dlg.comboBox.addItem(sorted([f["CODI_ESTAC"] + " - " + f["NOM_ESTACI"] for f in estacions_llista]))

# MSF: Insertar al ComboBox els Thresholds climàtics
self.dlg.comboBox_2.addItem(["Mediterrani temperat", "Mediterrani fred", "Montanya"])

# MSF: Insertar al ComboBox la possibilitat d'activar o no la funció auto_determinate_hour al posterior analisis
self.dlg.comboBox_3.addItem(["Desactivat", "Activat"])
```

Figura 27 Codi del mòdul principal de plugin que permet omplir els desplegables de la GUI amb els paràmetres per realitzar l'anàlisi.

9. Obtenció dels valors introduïts a la GUI

La GUI és el mitjà amb el qual el plugin es comunica amb l'usuari. D'aquesta manera, l'usuari pot establir i/o modificar els paràmetres d'entrada del plugin per a executar consecutivament l'anàlisi de pluges modificant-hi les variables depenent de les seves necessitats (Fig. 28). Un cop l'usuari ha introduït els valors dels paràmetres amb els quals pretén realitzar l'anàlisi, el plugin recupera aquests valors prèviament a realitzar l'anàlisi per enviar-los a les funcions del mòdul principal del plugin localrainfallanalysis.py. En el cas que no hi hagués una GUI per aquest plugin, si l'usuari volgués modificar els paràmetres de l'anàlisi hauria d'obrir l'arxiu Python contingut dins el directori del plugin i modificar-los manualment.

```
def threshold_value(Threshold_climatic_area):

    if Threshold_climatic_area == "Mediterrani temperat":
        threshold = 1
    elif Threshold_climatic_area == "Mediterrani fred":
        threshold = 2
    else:
        threshold = 3
    print("Selected threshold climatic area", Threshold_climatic_area, threshold)
    return threshold

threshold = threshold_value(Threshold_climatic_area)
threshold_number = int(threshold)
```

```
# MSF: Un cop recuperats els paràmetres introduïts per l'usuari a la GUI, fem la crida a la funció principal que executa tot l'anàlisi
read_rainfall_event_date(event_name, station_ID, event_x, event_y, datetime.datetime(year, month, day, hour, minutes), threshold_number, auto, 20, 25831,
os.path.normpath(filename + "/" + event_name + ".csv"),
os.path.normpath(filename + "/" + event_name + "_10daysrainfall.png"),
os.path.normpath(filename + "/" + event_name + "_10daystemp.png"),
os.path.normpath(filename + "/" + event_name + "_cummrainfall.png"),
os.path.normpath(filename + "/" + event_name + "_rainfalLevent.png"),
os.path.normpath(filename + "/" + event_name + "_ID.png"),
os.path.normpath(filename + "/" + event_name + ".b.csv"))
```

Figura 28 Recuperació de paràmetres que l'usuari ha especificat a la GUI i introducció d'aquests a la funció principal del mòdul Local Rainfall Analysis que desencadena les crides a les funcions per executar l'anàlisi de pluges.

3.5.2. Creació del plugin

En aquest apartat del projecte es documenten els passos realitzats per a la creació del plugin, es a dir, el conjunt d'accions per al desenvolupament físic del directori emmagatzemant els arxius necessaris (explicats anteriorment) que defineixen el plugin i que són el material que possibilita la instal·lació i execució a l'aplicació de QGIS. En aquest apartat també es documentarà la creació de la Interfície Gràfica d'Usuari (GUI).

3.5.2.1. Creació de la plantilla a través del Plugin Builder

Generar una plantilla a través del *Plugin Builder* permet, de forma ràpida i efectiva, generar l'estructura bàsica comuna per a tots els plugins. Com s'ha parlat en l'apartat anterior, tots els plugins estan compostos per una arquitectura física bàsica en format de directori de nivell superior i un conjunt d'arxius que defineixen la forma, funcions etc., del plugin. El *Plugin Builder* genera el directori de nivell superior amb el conjunt d'arxius necessaris per a començar a construir el plugin.

El plugin de *Plugin Builder* (Fig. 29) es per si mateix un plugin Python de QGIS que incorpora un conjunt de dades de l'usuari (sobre el plugin que pretén generar) i genera tot el conjunt de dades necessàries per a un plugin nou. Posteriorment depèn de l'usuari de personalitzar les funcionalitats d'aquest i afegir-hi el codi Python que realment contribuirà al fet que el plugin realitzi certes tasques.

Els passos que requerits (Fig. 30) per a la creació de la plantilla inicial del plugin son:

1. Instal·lació del plugin *Plugin Builder* a l'aplicació de QGIS.
2. Omplir tots els camps requerits del formulari que proporciona el plugin.
3. Seleccionar el tipus de plantilla a crear (en aquest cas un plugin amb interfície d'usuari)
4. Seleccionar on es pretén que QGIS emmagatzemi el plugin un cop instal·lat a l'aplicació (en aquest cas en el menú de plugins).
5. Seleccionar on es vol emmagatzemar la plantilla un cop *Plugin Builder* la generi (directori que contindrà els arxius base).



Figura 29 Característiques del plugin Plugin Builder 3 accessible des del Plugin Manager de QGIS.

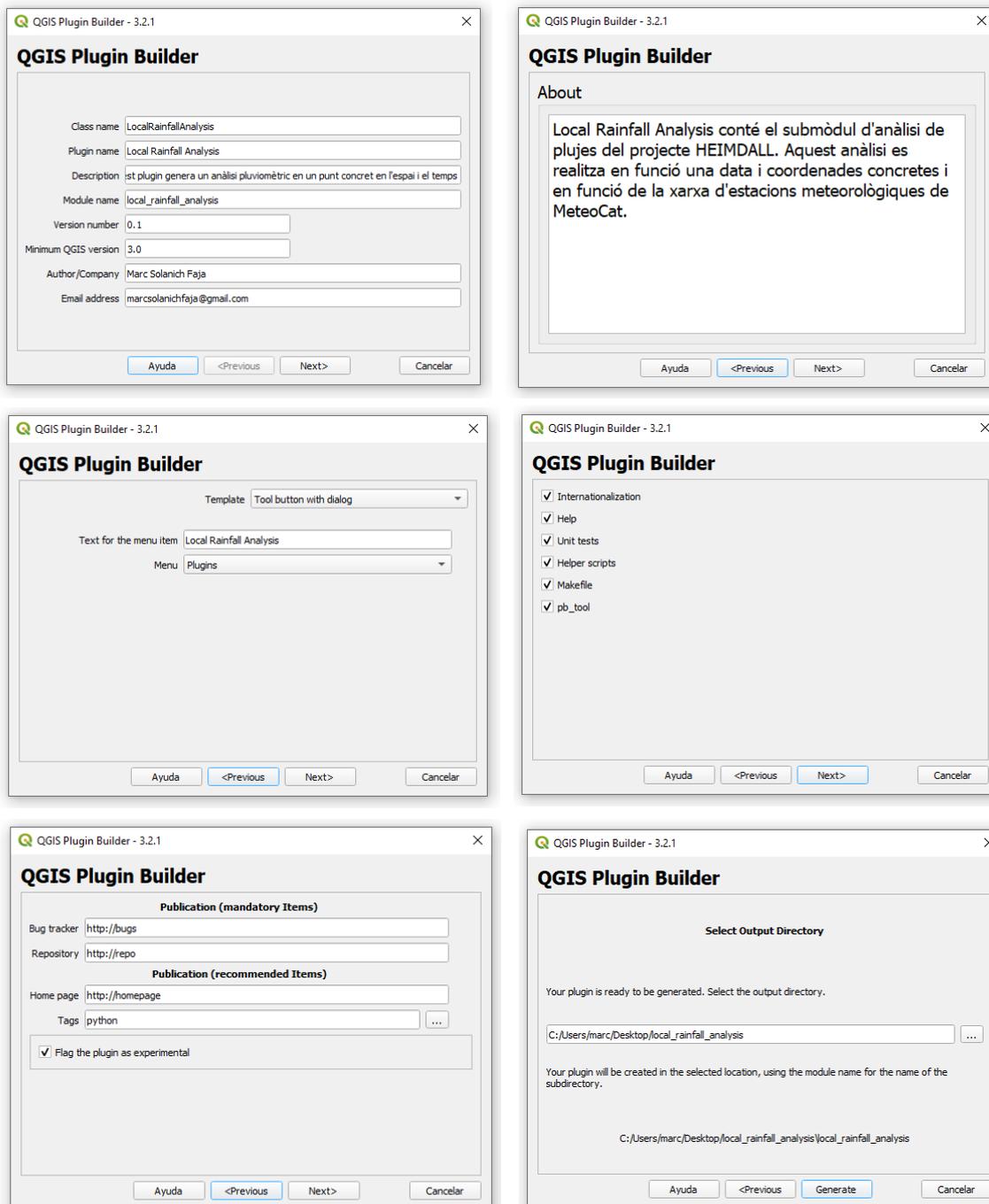


Figura 30 Conjunt de passos a seguir a la interfície del Plugin Builder 3 que permeten generar la plantilla del plugin d'aquest projecte.

El quadre de diàleg resultant de generar el plugin amb *Plugin Builder* (Fig. 31) conté informació útil per iniciar el procés de personalització del plugin:

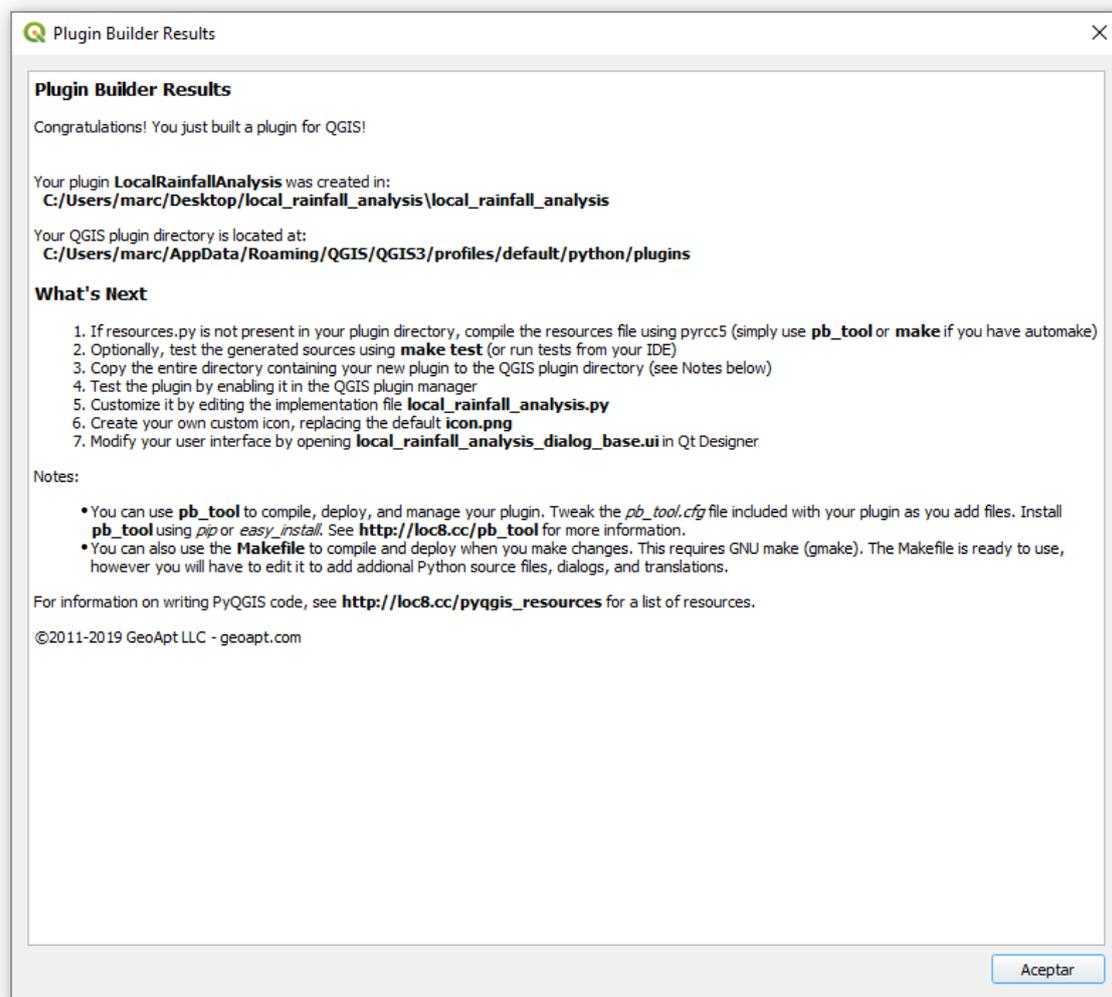


Figura 31 Quadre de diàleg resultant amb l'informe sobre la creació de la plantilla del plugin.

1. On s'ha guardat el plugin generat.
2. La ubicació del directori de QGIS on aquest emmagatzema els plugins.
3. Instruccions de com instal·lar el plugin.
4. Instruccions de com compilar l'arxiu de recursos (*resources.qrc*) i l'arxiu de la interfície gràfica (*local_rainfall_analysis_dialog_base.ui*).
5. Com personalitzar el plugin per ampliar-ne les funcionalitats.

En el cas que la compilació automàtica falli durant el procés de generació del plugin, s'haurà de fer manualment utilitzant pyrcc5. Prèviament a la utilització del plugin creat, es necessària la compilació de l'arxiu *resources.qrc* que el propi *Plugin Builder* ha generat. Aquest arxiu es part del *Qt Resource System*, o bé el Sistema de Recursos de Qt, el qual referencia tots els arxius binaris utilitzats en el plugin. En el cas del plugin d'aquest projecte, aquest arxiu contindrà informació sobre l'icona seleccionada del plugin i el conjunt gràfic de la GUI. A partir de la compilació d'aquests arxius es genera un codi d'aplicació que permet utilitzar-los en el plugin independentment de la plataforma amb la qual el plugin s'estigui utilitzant.

Pyrcc5 utilitza un arxiu de recursos de Qt (*.qrc*) en aquest cas el generat per *Plugin Builder* i el converteix en un mòdul Python el qual pot ser importat a una aplicació compatible amb PyQt.

Tot i haver-hi diverses metodologies compatibles amb el Sistema Operatiu de Windows per a compilar arxius, en aquest cas es crea un arxiu `.bat` que emmagatzemarà la línia de comandament amb `pyrcc5` (FIG. 32a) i s'afegeix al directori del plugin creat (Fig. 32b). Cada cop que s'efectuï un canvi en l'arxiu de la GUI o es canviï la icona del plugin, s'ha de compilar de nou els arxius per actualitzar els canvis.

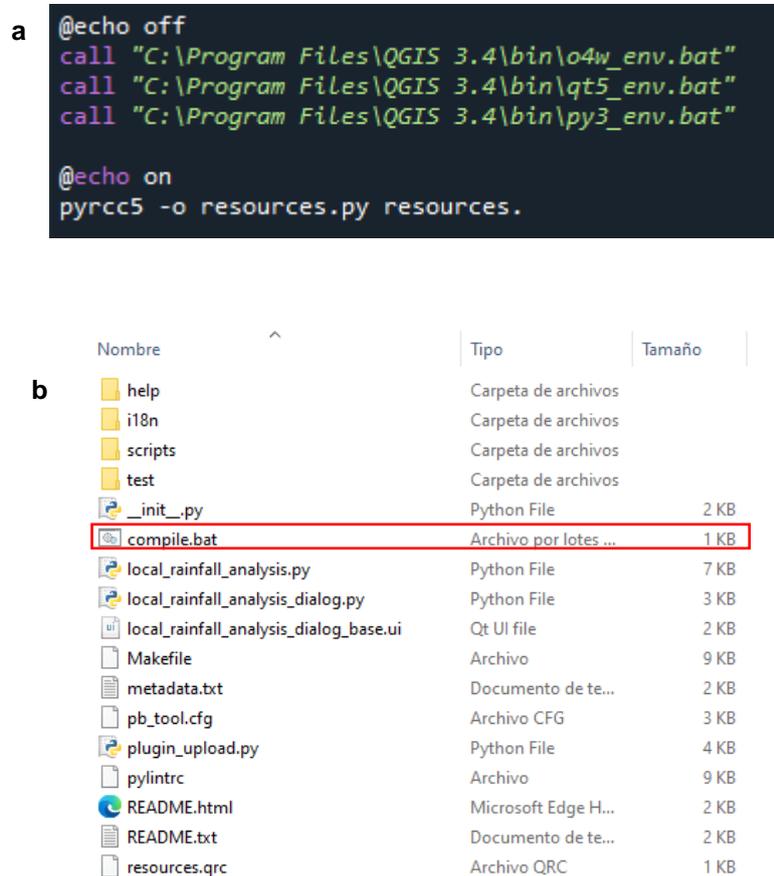


Figura 32 a: Compilador dels arxius del plugin `compile.bat`. b: localització del mòdul `compile.bat` dins el directori principal del plugin i l'arxiu creat a partir de la compilació: `resources.qrc`.

A partir de la plantilla creada i compilada a través de `compile.bat` el plugin es ja funcional en certs aspectes:

1. Es pot carregar a QGIS, afegint-se al menú de plugins i a la barra d'eines.
2. Cridar al plugin (clicant a la icona) executa un quadre de diàleg (GUI) buit.
3. Clicant en OK emet un senyal d'`accept()` i tanca la GUI (un cop el plugin posseeixi el codi per executar una funcionalitat aquest botó la desencadenarà).
4. Clicant en Cancel emet una senyal de `reject()` i tanca la GUI.

A partir d'aquest punt entra en funció l'aplicació de tot el codi modificat en l'apartat anterior per a complementar l'arxiu `local_rainfall_analysis` generat a través del *Plugin Builder* i que per ara no genera cap funcionalitat (Fig 33). La GUI és editada i personalitzada modificant l'arxiu `local_rainfall_analysis_dialog_base.py`.

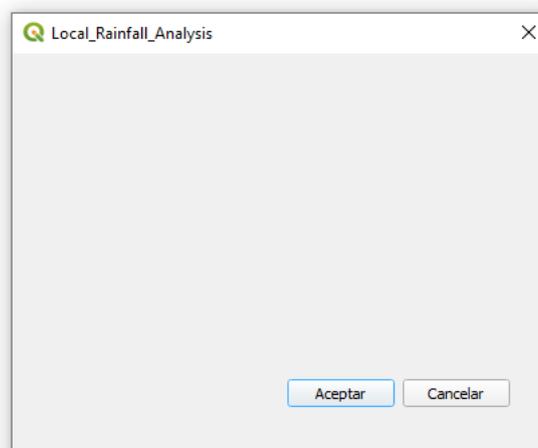
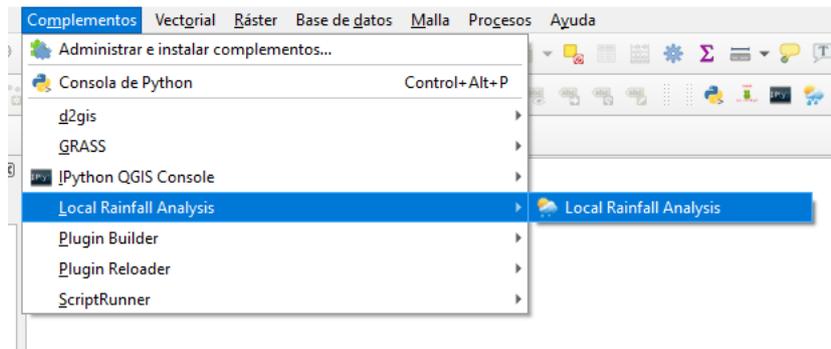


Figura 33 Localització del plugin creat (plantilla) en el menú de plugins de QGIS i aspecte de la plantilla inicial del plugin un cop es selecciona a través del menú de QGIS.

3.5.2.2. Creació de la Interfície gràfica d'usuari (GUI)

Per tal de modificar i personalitzar la GUI (Fig. 34) tal i com el plugin requereix, s'utilitza la mateixa eina que han utilitzat els desenvolupadors C++ amb el QGIS: Qt Creator. Aquest, com a eina de disseny visual d'aplicacions, permet crear caixes de diàleg i finestres a partir de *widgets* (ginys o *gadgets*) i especificant les seves propietats. En l'apartat anterior de disseny de l'aplicació, es defineixen les entrades de paràmetres que ha de tenir la GUI per garantir l'execució correcta del plugin.

L'objectiu de la GUI (Fig. 35) es permetre a l'usuari comunicar-se amb el plugin. D'aquesta manera, a través d'aquesta, l'usuari ha de poder introduir els paràmetres que caracteritzaran l'anàlisi de pluges a executar, els quals poden ser diferents per a cada anàlisi realitzat. Per tant, la GUI ha de presentar de forma entenedora i àgil, les diverses opcions perquè l'usuari afegeixi en aquesta els valors dels paràmetres en qüestió.

Per personalitzar la GUI a través del Qt Designer es necessari obrir l'arxiu local_rainfall_analysis_dialog_base.ui al Qt Creator. En la figura (Fig. 33) es pot veure el quadre de diàleg generat per l'arxiu resultant de *Plugin Builder*, el qual només té funcions per executar el plugin o tancar la finestra de diàleg.

Qt Creator permet arrastrar els *widgets* necessaris des de les seves caixes d'eines al quadre de diàleg que representa la GUI del plugin. Cada *widget* arrastrat i incorporat a la GUI es representat per un objecte i la seva corresponent classe ambdós de Qt i PyQt (Fig. 34) Qt Creator incorpora les propietats de cada *widget* de Qt i per tant cada objecte associat que s'incorpora al quadre de diàleg de la GUI. D'aquesta manera, es possible vincular els objectes creats i afegits a l'arxiu de la GUI del plugin, local_rainfall_analysis_dialog_base.ui, al arxiu principal del plugin local_rainfall_analysis.py per tal d'interactuar amb ells, amb l'objectiu d'afegir i poblar els camps de la GUI o recuperar-ne els valors introduïts per l'usuari.

Cada modificació realitzada a l'arxiu de la GUI des del Qt Creator requereix executar l'arxiu *compile.bat* afegit al directori principal del plugin per tal de compilar els canvis i poder ser visibles des dels mòduls Python del plugin.

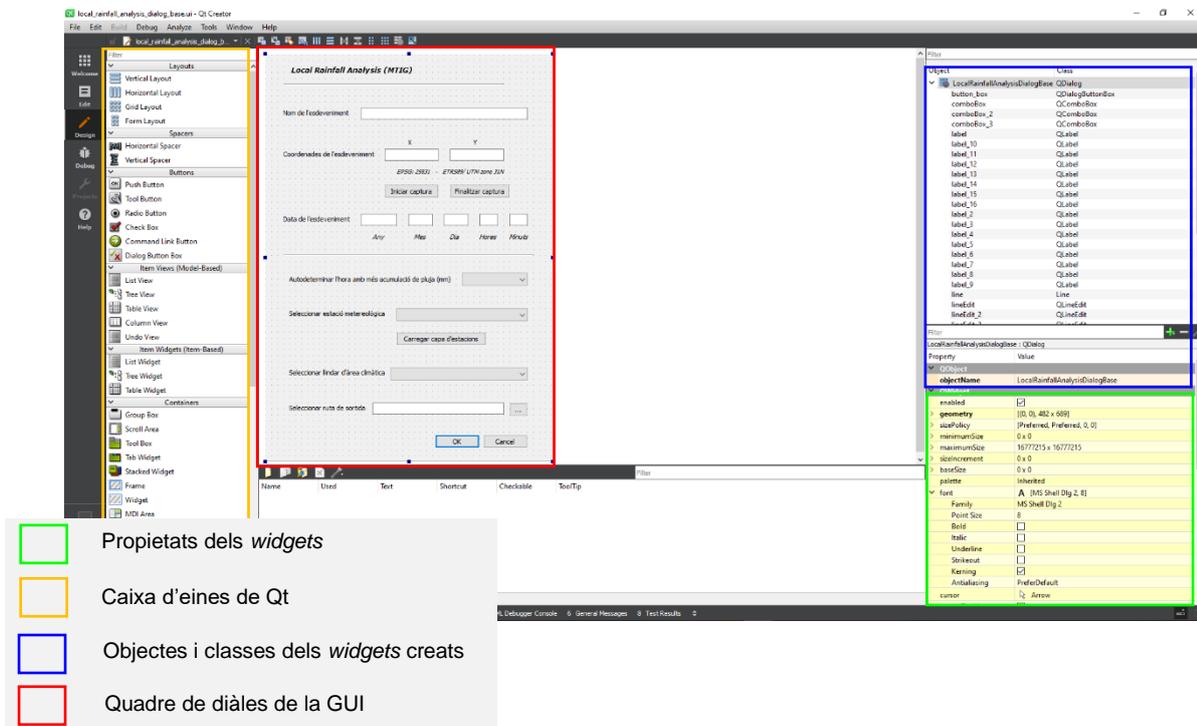


Figura 34 Parametrització de la GUI i disseny d'aquesta a través del Qt Creator.

Local Rainfall Analysis (MTIG)

Nom de l'esdeveniment

Coordenades de l'esdeveniment X Y
EPSG: 25831 - ETRS89/ UTM zone 31N

Data de l'esdeveniment
Any Mes Dia Hores Minuts

Autodeterminar l'hora amb més acumulació de pluja (mm)

Seleccionar estació metereològica

Seleccionar líndar d'àrea climàtica

Seleccionar ruta de sortida

Figura 35 GUI resultant del procés de disseny del plugin, aquesta permet a l'usuari comunicar-se amb el plugin.

3.5.2.3. Instal·lació del plugin

Els plugins són instal·lats al QGIS utilitzant el *Plugin Manager*. Aquest es l'administrador dels plugins existents i instal·lats en el QGIS i de connectar-se en els repositoris oficials de plugins de QGIS per buscar i descarregar-ne en el cas que sigui necessari. Cada cop que s'obre el *Plugin Manager*, els repositoris configurats són consultats i es mostra una llista dels plugins actualitzada i classificada per categories: a) Tots, b) Instal·lats, c) No instal·lats i d) Nous.

Per instal·lar un plugin com el que s'ha realitzat en aquest projecte, per tant un plugin classificat com a experimental i que per tant que no es pot trobar en els repositoris oficials de QGIS, es pot fer de diverses maneres:

1. Situar-se en la carpeta del perfil actiu de QGIS on és possible navegar fins el directori de python que conté els directoris de cada plugin i copiar el directori del plugin a instal·lar, en aquest cas el generat en aquest projecte (Fig. 36).

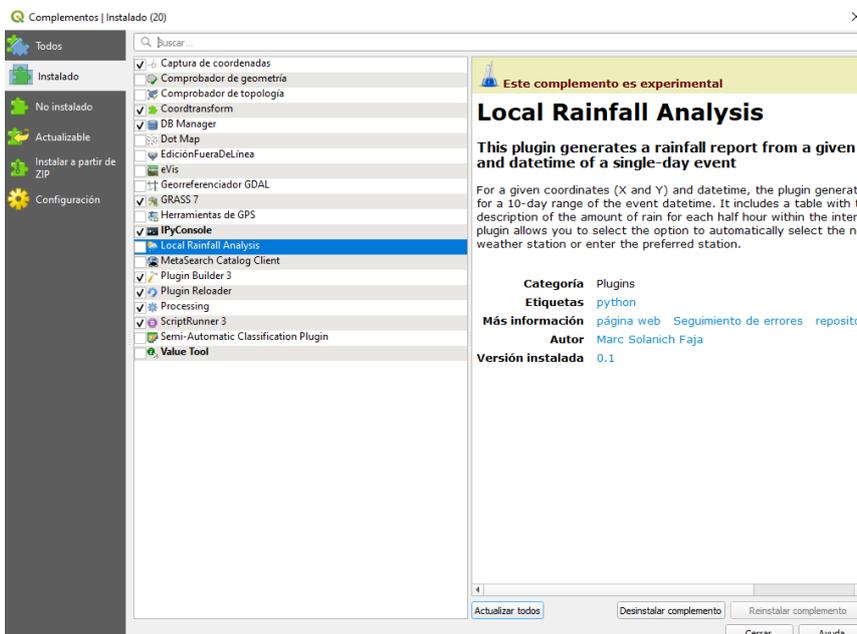
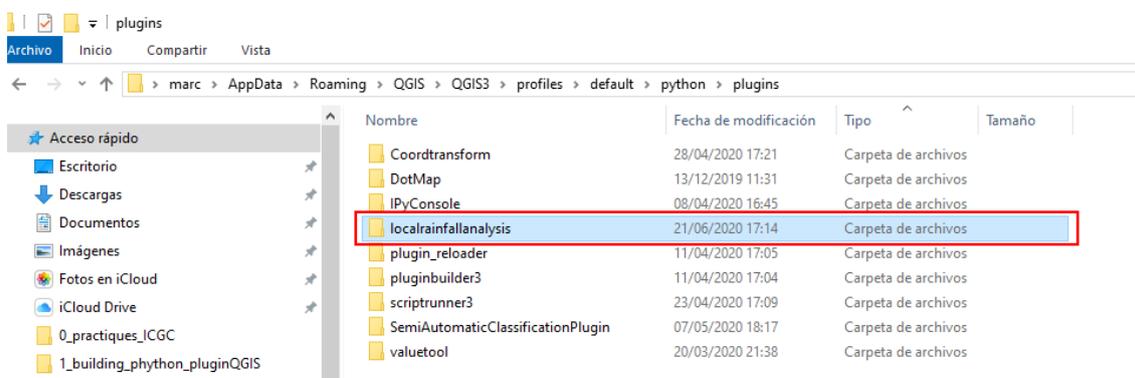


Figura 36 Instal·lació manual del plugin al directori actiu de QGIS.

Un cop l'arxiu és afegit al directori python del directori del perfil actiu de QGIS, des del *Plugin Manager* ja es accessible l'activació d'aquest des de la categoria c).

2. Instal·lació a partir d'un arxiu *.zip* (comprimir el directori principal del plugin en un arxiu *.zip* amb tot el seu contingut), des del *Plugin Manager* (Fig. 37).

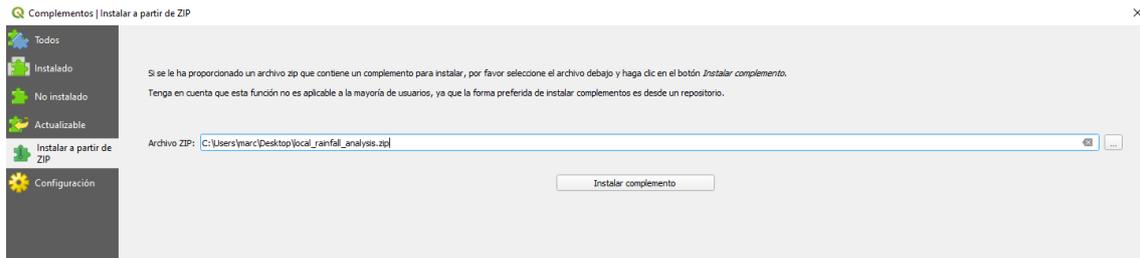


Figura 37 Instal·lació manual del plugin a través d'un arxiu *.zip*.

La instal·lació del plugin és un pas previ al inici del període prova-error que concentra la major part dels canvis i desenvolupament del plugin. Com a eina essencial de la metodologia de desenvolupament establerta per aquest projecte, *Plugin Reloader*, es instal·lat al mateix moment en que s'instal·la el *Plugin Builder*.

Plugin Reloader és un plugin destinat a desenvolupadors d'aplicacions de QGIS. Aquest permet actualitzar el plugin després de cada fase de canvis al codi o addició de funcionalitats, sense haver de reiniciar el QGIS durant les fases de desenvolupament.

4. Prova de funcionament en un entorn real

En aquest apartat del projecte es presenten, de forma sintètica, les funcionalitats i resultats del plugin *Local Rainfall Analysis* en dos casos d'estudi reals d'esllavissades vinculats en períodes de pluja intensos. Aquests casos formen part del dia a dia dels tècnics del ICGC i per tant permeten reproduir l'entorn real d'aplicació del plugin. D'aquesta manera es pot comprovar la utilitat de l'eina desenvolupada i que serà utilitzada per l'equip corresponent dins del ICGC, millorant l'anàlisi d'esllavissades que es donen recurrentment a Catalunya quan hi ha episodis de pluges intenses.

Els dos casos estudiats són una esllavissada en una zona rural (Mas Canyelles, Castellar del Vallès) i en una zona de carretera transitada (kilòmetre 8 de la carretera BV-2041, Begues – Gavà). Per a cada cas s'han definit els paràmetres de la manera indicada depenent de les condicions i característiques que es coneixen de cada esdeveniment.

4.1. Primer cas d'anàlisi: colada de fang a la carretera de Begues (Begues, Baix Llobregat)

A les 17:42 h del dia 21 d'abril d'aquest any, la policia local de Begues (Barcelona) informa via Twitter d'un tall al kilòmetre 8 de la carretera a Gavà (BV-2041) per una colada de fang interceptant la carretera (Fig. 38 i 39).



Figura 38 Imatges capturades a la carretera a Gavà el dia 21 d'abril per les autoritats locals, en el moment que la pluja va desencadenar una colada de fang.

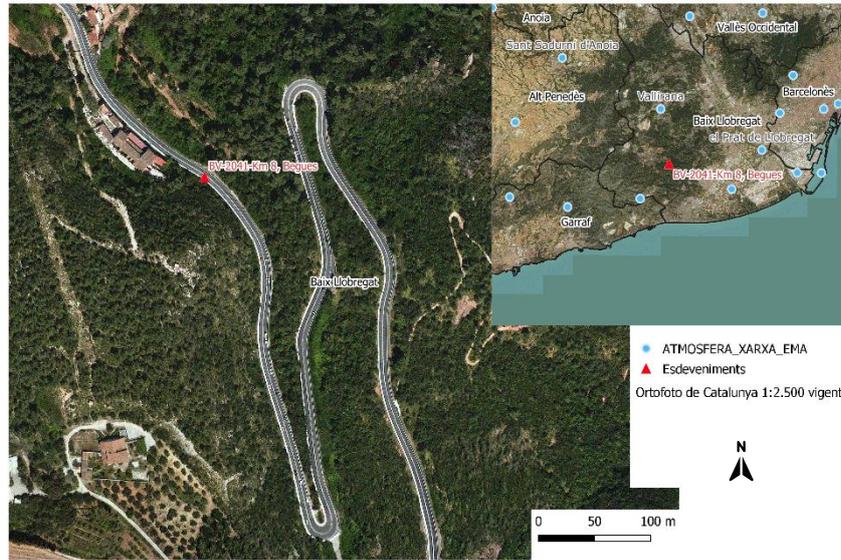


Figura 39 Localització del punt de l'esdeveniment, el qual s'utilitza per realitzar l'anàlisi de pluges. Carretera de Gavà BV-2041 kilòmetre 8.

Degut a l'esposició de l'esdeveniment en una via pública de fàcil excés i per tant de fàcil detecció, és fàcil considerar l'hora d'avís del perill a la carretera per un corrent d'esbaldregalls com a l'hora inicial (entenenent que no es pot determinar com a un punt concret d'inici en el temps, sinó com un lapse de temps més o menys ampli d'hores). En aquest cas, els paràmetres d'entrada de l'anàlisi (Fig. 40), coneixent ja la localització exacte de l'esdeveniment, s'establiran a partir de la data considerada d'inici (21/04/2020 – 17:42 h), i no es buscarà el màxim d'acumulació de pluja per a l'interval de dies prèvis a l'esdeveniment, des del dia 16 d'abril fins al mateix dia de l'esdeveniment.

Figura 40 Introducció de paràmetres a la GUI per realitzar l'anàlisi de pluges per el cas de la carretera a Gavà.

L'anàlisi s'executa a partir de l'estació meteorològica: PN del Garraf – el Rascler de la xarxa del Servei Meteorològic de Catalunya, la qual és la més propera al quilòmetre 8 de la carretera BV-2041, situada aproximadament a uns 5 kilòmetres del punt de l'esdeveniment. Aquesta enregistra un episodi de pluges intenses durant els dies analitzats. Els resultats de l'anàlisi de pluges, amb unes dades pluviomètriques de la zona obtingudes a partir de l'anàlisi de 20 dies consecutius (Fig. 41a), des de l'11 d'abril a l'1 de maig, registra un total de 255,5 mm d'acumulació durant aquest període, el 88% d'aquesta acumulació (224,8 mm) són acumulats durant l'esdeveniment (Fig. 42b), el qual s'associa a una pluja amb una duració de més de 4 dies (de les 17 h del dia 18 d'abril a les 4:30 h del 22 d'abril).

El resultat de l'anàlisi subratlla la presència d'un pic de pluja molt més elevat, dins el període de pluja de l'esdeveniment, que el pic on es situa la data de l'esdeveniment. En el cas que s'utilitzés l'opció de determinar automàticament l'hora i el dia amb màxima acumulació de pluja, la data de l'esdeveniment automàticament s'hagués posicionat en el màxim del pic del dia 19 d'abril. En aquest cas d'anàlisi, es considera que la data d'origen utilitzada és representativa i que permet, juntament amb les dades d'acumulació total i intensitat que complementen la interpretació de les condicions crítiques de l'esdeveniment, caracteritzar i discriminar si la pluja del dia 21 d'abril és potencialment responsable del desencadenament del corrent d'esbaldregalls a la carretera de Begues. A més a més, l'anàlisi d'intensitats màximes de la pluja destaca el pic del dia 21 d'abril per sobre del pic del dia 19 d'abril pel que fa a valors més alts d'intensitat.

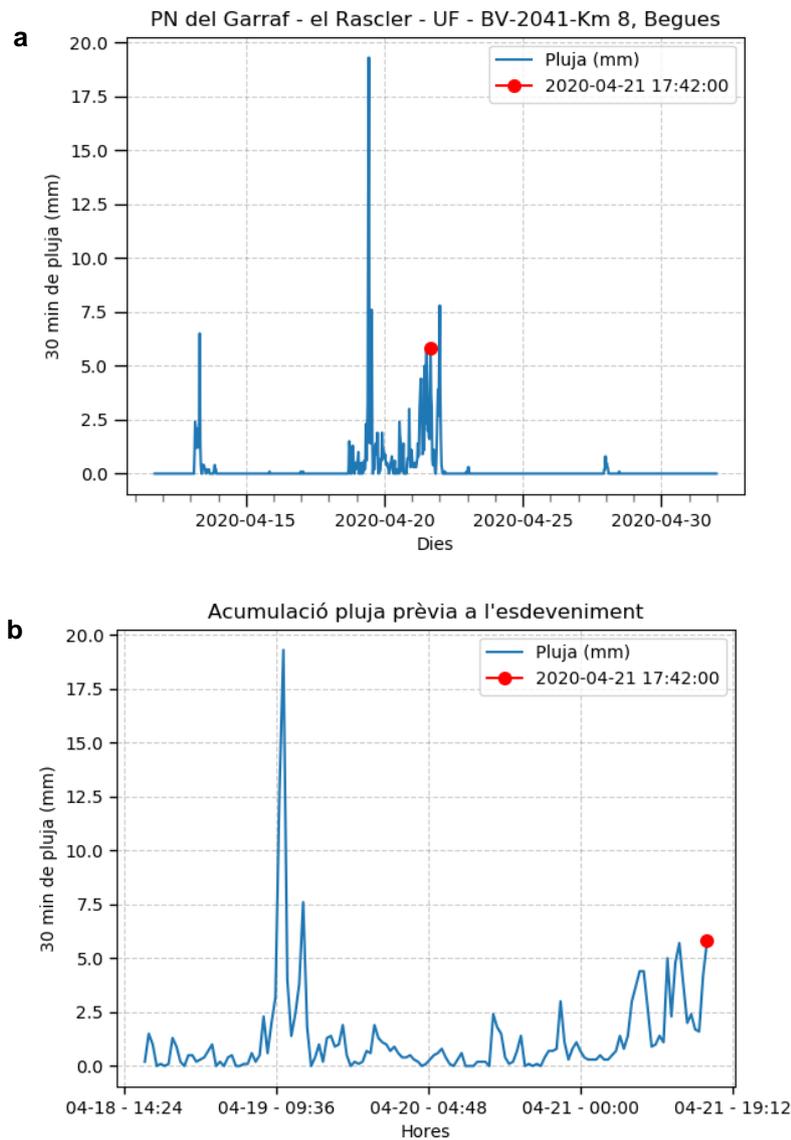


Figura 41 a: evolució de la pluja al llarg dels dies de l'esdeveniment (20 dies analitzats) on es situa l'hora concreta de l'esdeveniment. b: evolució de la pluja durant el període inicial de l'esdeveniment (des de l'inici de les pluges de l'esdeveniment fins al moment concret de l'esdeveniment).

Les dades d'acumulació total prèvies a l'inici del període de l'esdeveniment (Fig. 42) a les 17 h del dia 18 d'abril, indiquen que aquest pot estar directament influenciat per l'acumulació prèvia de pluja. Aquesta es caracteritza per la presència de valors elevats en l'acumulació total que s'han acumulat durant 7 dies previs (del 9 d'abril al 18 d'abril). Les dades d'acumulació per als períodes de 3 dies i 24 hores prèvies a l'esdeveniment són quasibé nul·les, per tant, es pot apuntar que la màxima acumulació total es genera dins l'interval de 4 a 7 dies prèvies al període de l'esdeveniment (corresponents als dies 9 i 13 d'abril).

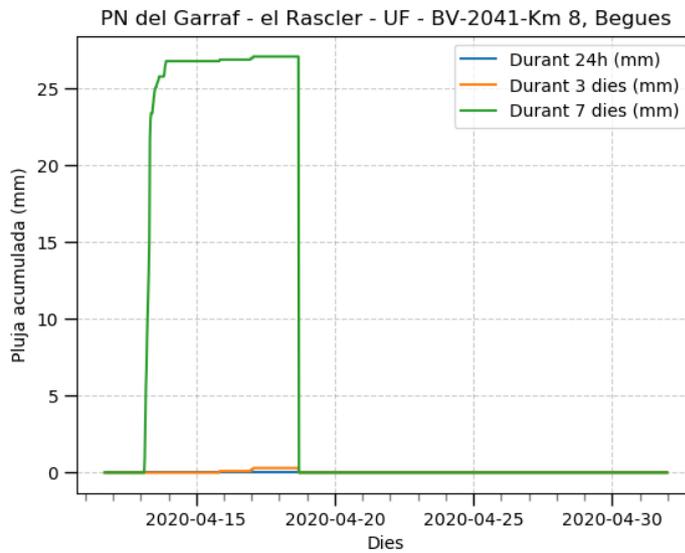


Figura 42 Acumulació prèvia a l'esdeveniment per períodes de 24 hores, 3 i 7 dies.

Pel que fa a l'anàlisi de la intensitat de la pluja acumulada per a diferents períodes de mesura (0,5 hores, 24 hores, 3 dies i 7 dies previs a l'esdeveniment, Fig.43) es desmarquen uns valors molt més elevats d'intensitat per als moments pròxims a l'esdeveniment (0,5 hores i 24 hores) on els valors d'intensitat màxima són: 38,6 i 32 mm/h respectivament. Pel que fa a les intensitats mesurades pels períodes de 3 i 7 dies aquestes són menors: 14,5 i 11,6 mm/h respectivament. Aquest increment en la intensitat en moments propers a l'esdeveniment situen la corba d'intensitat que caracteritza la pluja de l'esdeveniment per sobre del llindar crític escollit per l'anàlisi. Això significa que la pluja enregistrada durant el dia 21 d'abril es considerada una pluja crítica amb potencial desencadenant d'una colada de fang.

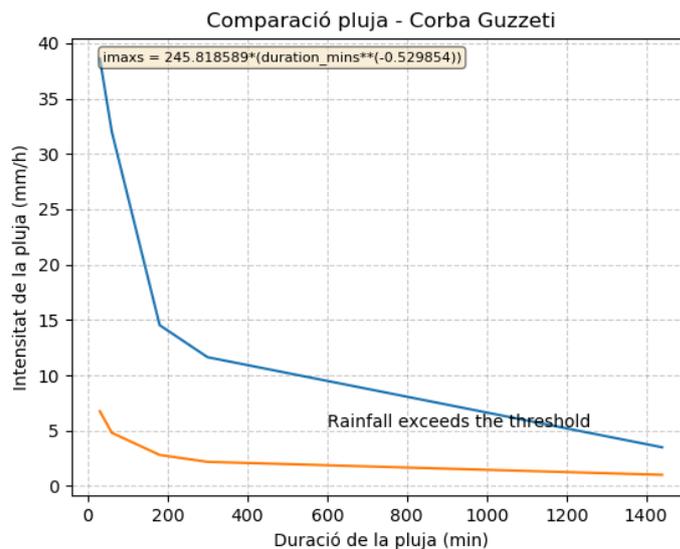


Figura 43 Comparació corba d'intensitat de l'esdeveniment amb la corba de Guzzeti et al., (2008) que defineix els valors llindars d'intensitat. En aquest cas, la pluja que caracteritza l'esdeveniment sobrepassa els llindars establerts.

El gràfic amb l'evolució de la temperatura mitjana (°C) durant el període d'anàlisi (Fig. 44), com a element complementari en el cas que no hi hagues presència d'un episodi de pluges, situa el dia 21 d'abril en un moment d'inestabilitat, amb un descens generalitzat de les temperatures que dura entre 3 i 4 dies. El resultat de l'anàlisi també incorpora els 3 gràfics de previsió futura del possible risc (Fig.45) generat per la pluja que caracteritza l'esdeveniment, per a períodes de 24 hores, 3 dies i 7 dies. Tot i no ser del tot necessaris amb l'objectiu amb el qual els usuaris realitzen l'anàlisi (compilació de dades, no anàlisi de risc) poden aportar informació sobre com evoluciona l'impacte de la pluja crítica.

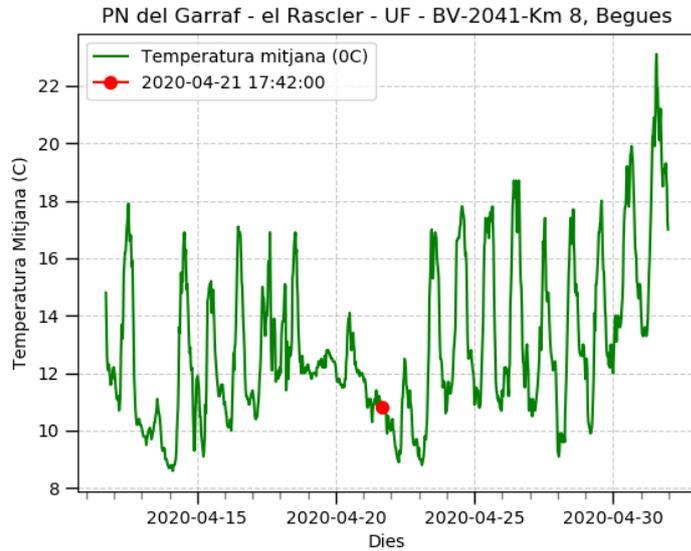


Figura 44 Evolució de la temperatura durant el període d'anàlisi (20 dies).

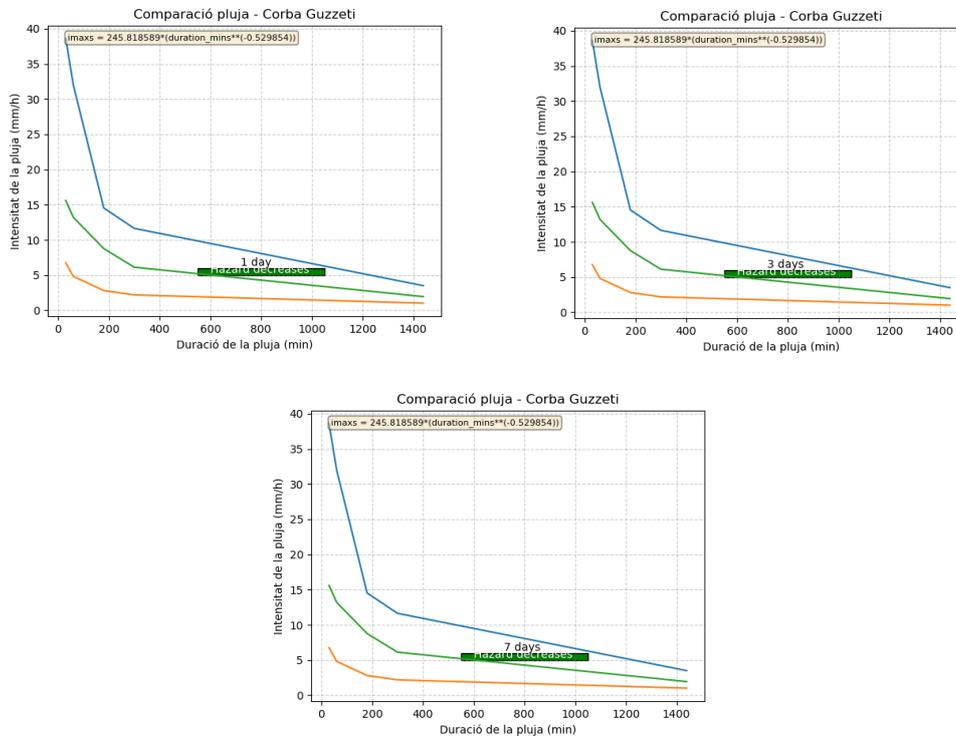


Figura 45 Evolució de la intensitat de la pluja en els gràfics post-esdeveniment per a 24h 3 i 7 dies. La intensitat de l'esdeveniment segueix indicant períodes de perillositat en períodes posteriors.

4.2. Segon cas d'anàlisi: camí cap a Can Cadafalc, Mas Canyelles (Castellar del Vallès, Vallès Occidental)

El dia 23 d'abril s'informa a l'Institut Cartogràfic i Geològic de Catalunya (ICGC) amb relació a unes esllavissades que afectaven el camí de Can Cadafalc a l'altura del Mas Canyelles, al municipi de Castellar del Vallès (Fig. 46). Es desconèix el moment concret en el qual es produeix l'esdeveniment, tot i això s'associa a un període de pluges del 18 al 24 d'abril (equivalent a l'episodi de pluges del cas anterior).



Figura 46 Imatges capturades el dia 23 d'abril al Camí de Can Cadafalc per les autoritats locals, en un moment posterior on s'apunta que la pluja va desencadenar una esllavissada, conjuntament amb la ubicació precisa del punt de l'esdeveniment.

Al no conèixer el desencadenant de l'esllavissada ni el moment exacte, el plugin te la funcionalitat de proporcionar les dades de pluja durant aquest període per tal de discriminar si la pluja es pot relacionar amb el desencadenament de l'esllavissada si es categoritzada com a pluja crítica. Es considera com a data d'entrada per a l'anàlisi el dia 23 d'abril, el qual es va detectar l'esllavissada (Fig. 47).

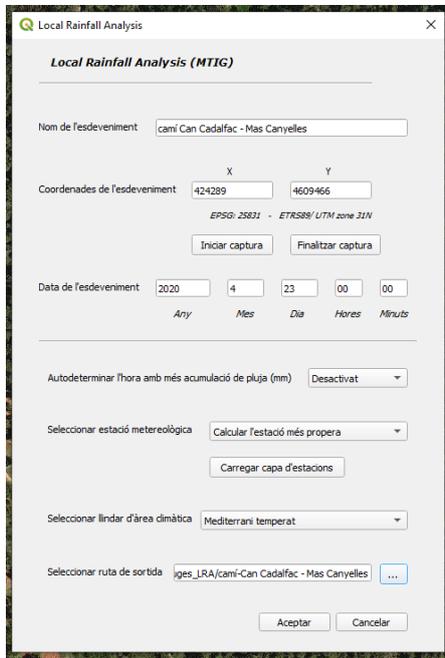


Figura 47 Introducció de paràmetres a la GUI per realitzar l'anàlisi de pluges per al cas del Camí de Can Cadafac.

L'anàlisi s'executa a partir de l'estació meteorològica de Caldes de Montbui de la xarxa del Servei Meteorològic de Catalunya, la qual és la més propera al camí de Can Cadafac al punt del Mas Canyelles, situada aproximadament a 7 kilòmetres del punt de l'esdeveniment. A diferència del cas anterior, aquesta no enregistre cap episodi de pluges intenses concretament durant el dia 23 d'abril quan en teoria es produeix l'esllavissada però, si que enregistre un període intens de pluja durant els dies del 18 al 22 d'abril. A part dels gràfics que il·lustren i documenten el resultat de l'anàlisi, aquest es complementa amb els informes .csv que sintetitzen, tot el conjunt de valors estudiats (dates i valors de precipitació per cada mitja hora de tot el període d'anàlisi) i un resum de les característiques de l'esdeveniment (Fig. 48).

Informe descriptiu		Llistat de valors de precipitació	
Nom de l'esdeveniment: camí - Can Cadafac - Mas Canyelles		Data,Hora,Precipitació (mm)	
Data de l'esdeveniment: 2020-04-23 00:00:00		13/04/2020,00:00 - 00:30,0,0	
Durada de l'esdeveniment: 2020-04-22 22:00:00+00:00 - 2020-04-22 22:00:00+00:00		13/04/2020,00:30 - 01:00,0,0	
Període analitzat: 2020-04-13 00:00:00+00:00 - 2020-05-03 23:30:00+00:00		13/04/2020,01:00 - 01:30,0,0	
Codi d'estació: X9		13/04/2020,01:30 - 02:00,0,0	
Nom de l'estació: Caldes de Montbui		13/04/2020,02:00 - 02:30,0,0	
Distància entre l'esdeveniment i l'estació (m): 6840,4		13/04/2020,02:30 - 03:00,0,0	
Precipitació total acumulada (mm): 157.79999999999995		13/04/2020,03:00 - 03:30,0,0	
Precipitació total acumulada durant l'esdeveniment (mm): 0,0		13/04/2020,03:30 - 04:00,0,0	
Intensitats màximes previes a l'esdeveniment (mm/h):		13/04/2020,04:00 - 04:30,0,0	
0,5 h previes: No hi ha acumulació previa de pluja		13/04/2020,04:30 - 05:00,0,6	
24 h previes: No hi ha acumulació previa de pluja		13/04/2020,05:00 - 05:30,2,5	
3 dies previes: No hi ha acumulació previa de pluja		13/04/2020,05:30 - 06:00,1,1	
7 dies previes: No hi ha acumulació previa de pluja		13/04/2020,06:00 - 06:30,0,4	
		13/04/2020,06:30 - 07:00,0,7	
		13/04/2020,07:00 - 07:30,0,4	
		13/04/2020,07:30 - 08:00,1,1	
		13/04/2020,08:00 - 08:30,1,4	
		13/04/2020,08:30 - 09:00,1,9	
		13/04/2020,09:00 - 09:30,1,7	
		13/04/2020,09:30 - 10:00,3,6	

Figura 48 Informes descriptius del resultat de l'anàlisi de pluges. A l'esquerra, l'informe descriptiu amb les característiques de l'esdeveniment. A la dreta, el llistat de valors de precipitació per a cada mitja hora durant el període analitzat.

Els resultats de l'anàlisi de pluges, amb dades pluviomètriques de la zona obtingudes a partir de l'anàlisi de 20 dies consecutius, des del 13 d'Abril al 3 de maig, registra un total de 157 mm d'acumulació durant aquest període, dels quals un 0% es situa dins la franga definida per l'esdeveniment del dia 23. El 100% d'aquesta acumulació es acumula durant els dies anteriors (Fig. 49 a i b).

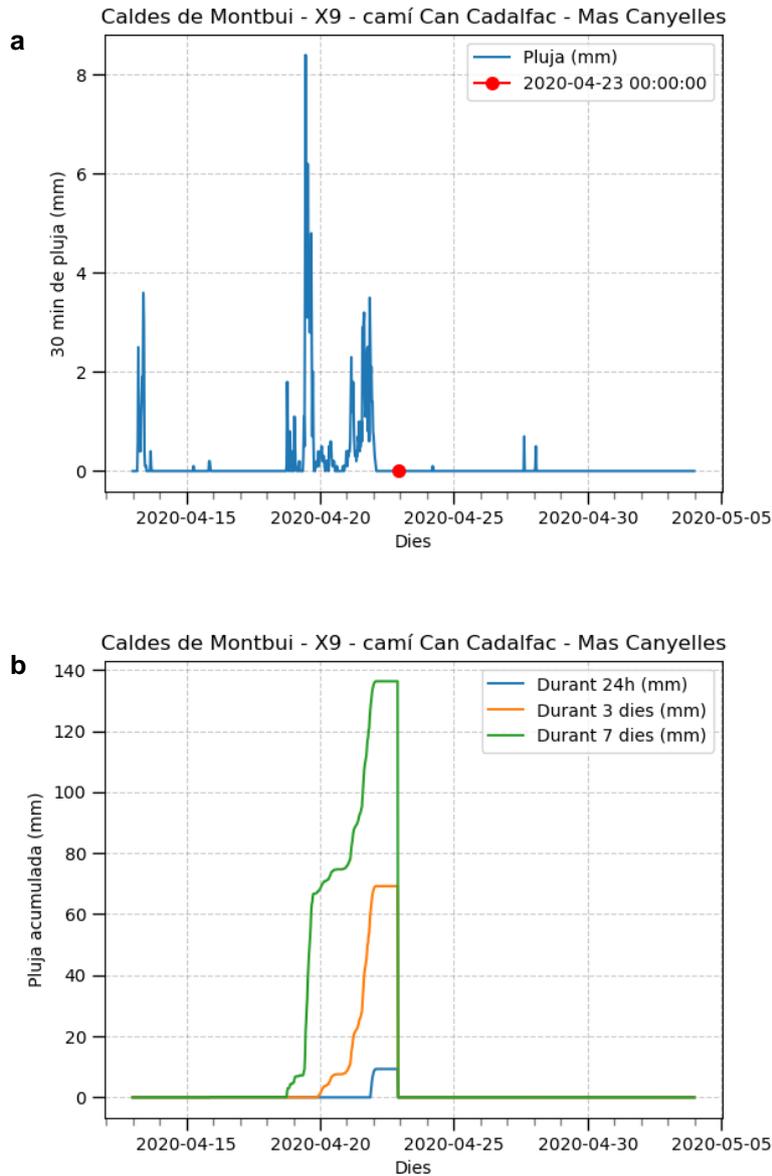


Figura 49 a: evolució de la pluja al llarg dels dies de l'esdeveniment (20 dies analitzats) on es situa l'hora concreta de l'esdeveniment. **b:** Acumulació prèvia a l'esdeveniment per períodes de 24 hores, 3 i 7 dies.

Els resultats de l'anàlisi subratlla la presència d'uns pic de pluja molt més elevat amb una morfologia semblant en el cas anterior que es situen dies enrere. Com que es considera que la data introduïda del moment de l'esllavissada no cau dins de cap episodi de pluja, no es construeix cap període d'esdeveniment amb la seva pluja associada. Degut a això, no es realitza tota la part d'anàlisi que comporta l'estudi de la pluja que caracteritza l'esdeveniment.

En aquest cas, no es pot dir que el dia 23 hi hagi un desencadenant que sigui la pluja. És interessant doncs ara, i en els casos on no se sap amb certesa el moment en el qual s'ha donat

l'esdeveniment i/o si ha hagut un episodi de pluges al voltant dels dies on s'ha detectat una esllavissada, utilitzar la funcionalitat del plugin d'analitzar, en un període de 5 dies anteriors a la data introduïda, on es situa el pic de pluja amb el màxim de precipitació (Fig. 50). D'aquesta manera, l'usuari fàcilment pot identificar, en el cas de no saber-ho anteriorment, on es situen els episodis de pluja més intensa, i concretar la data on la pluja (depenent de les característiques) hagi pogut desencadenar l'esllavissada.

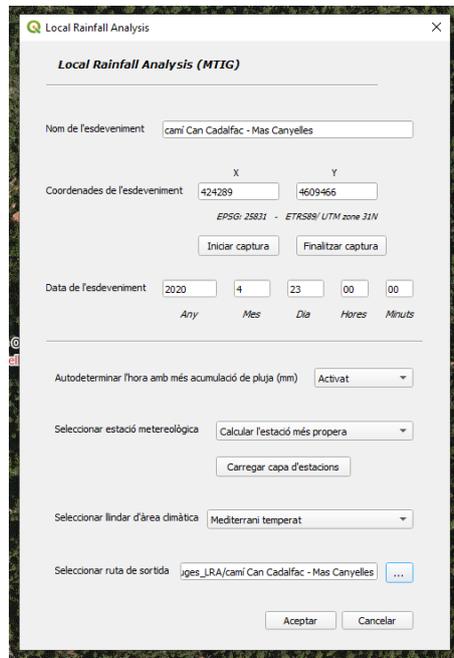


Figura 50 Introducció de nou dels paràmetres pel cas d'anàlisi concret activant la funció de calcular l'hora amb més precipitació per un període de 5 dies anteriors a l'esdeveniment.

Els resultats de l'anàlisi amb la condició de calcular i ajustar-se al dia amb màxima acumulació, implica que la data de l'esdeveniment es trasllada al 19 d'abril, 4 dies abans, on es troba el pic màxim de pluja dintre l'episodi de pluges d'entre els dies 18 i 24 d'abril (Fig. 51 a i b). Tot i haver-hi presència d'altres pics elevats de pluja dintre de l'interval d'anàlisi, el l'algoritme del plugin sempre escollirà el que tingui el valor més alt. Això pot suposar obviar pics de pluja més propers al dia original (23 d'abril) on es considerava inicialment el dia on es va produir l'esllavissada. Tot i això, com que es un cas d'incertesa on es relaciona el desencadenant a un episodi de pluja que dura més de 3 dies, el més probable es assignar el pic de pluja màxima al possible desencadenant de l'esllavissada (depenent de l'intensitat crítica).

La Fig. 49c enregistra l'episodi de pluja concret que caracteritza l'esdeveniment del dia 19 d'abril. La pluja que representa el pic escollit té una durada de les 9:00 h del dia 4 d'abril fins les 3:00 h del mateix dia (tot i que la figura trunca l'episodi de l'esdeveniment al registre d'hores prèvies fins l'esdeveniment). Aquesta caracterització de la pluja (Fig. 51c) significa que, durant les 4 hores d'aquest episodi la pluja es considera continuada per cada mitja hora dins l'interval. L'algoritme salta de mitja hora en mitja hora analitzant si, 2 hores abans i després hi ha valors de pluja, i per tant, en el moment on detecta que no es compleix tal premissa, s'estableix el principi i/o el final de l'episodi de pluges que caracteritza l'esdeveniment per a la data seleccionada (Fig. 52).

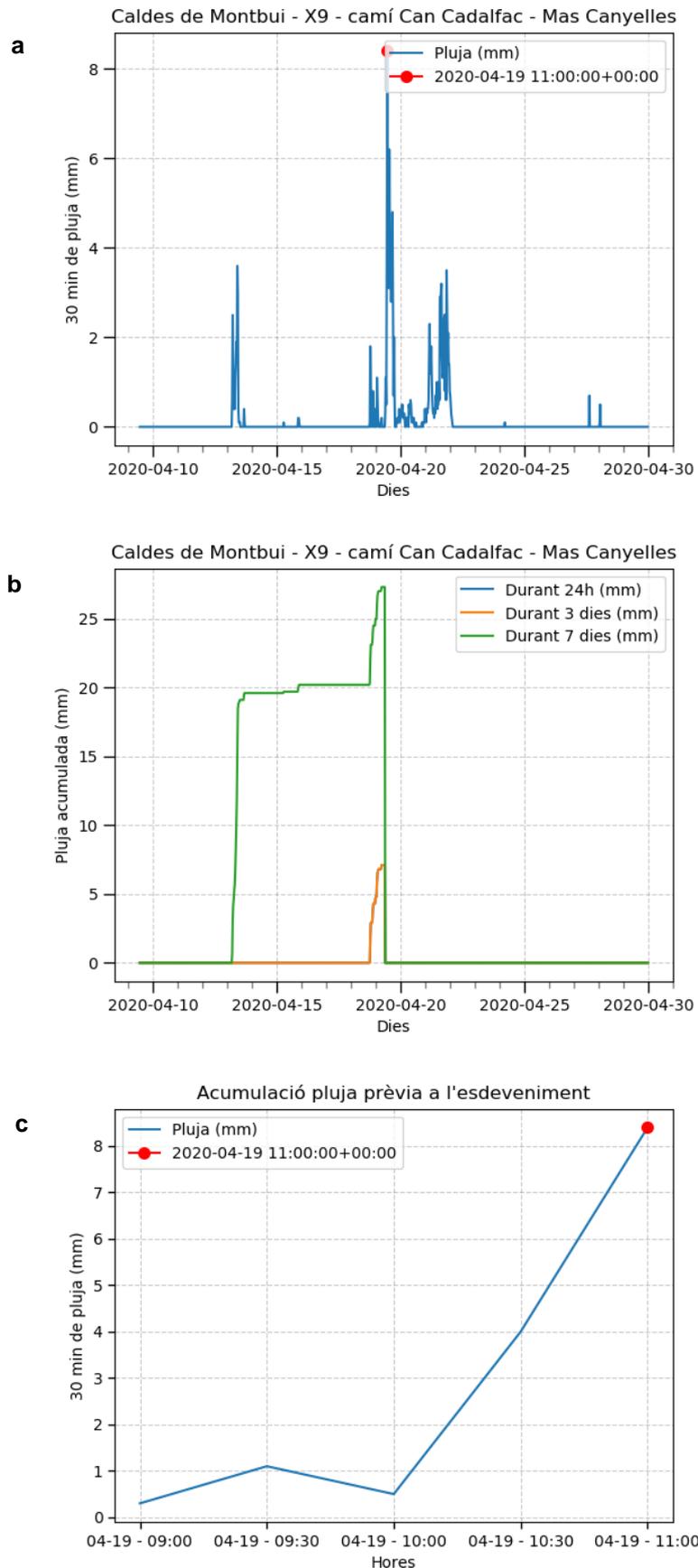


Figura 51 a: evolució de la pluja al llarg dels dies de l'esdeveniment (20 dies analitzats). b: Acumulació prèvia a l'esdeveniment. c: evolució de la pluja durant el període inicial de l'esdeveniment (des de l'inici de les pluges de l'esdeveniment fins al moment concret de l'esdeveniment).

Data de l'esdeveniment:	2020-04-19 11:00:00+00:00
Durada de l'esdeveniment:	2020-04-19 09:00:00+00:00 - 2020-04-20 15:00:00+00:00
Període analitzat:	2020-04-09 11:00:00+00:00 - 2020-04-29 23:30:00+00:00
Codi d'estació:	X9
Nom de l'estació:	Caldes de Montbui
Distància entre l'esdeveniment i l'estació (m):	6840.4
Precipitació total acumulada (mm):	157.79999999999995
Precipitació total acumulada durant l'esdeveniment (mm):	67.49999999999999
Intensitats màximes previes a l'esdeveniment (mm/h):	
0,5 h previes:	16.8
24 h previes:	12.4
3 dies previes:	4.766666666666667
7 dies previes:	2.8600000000000003
Equació de l'esdeveniment:	
imaxs =	182.248664*(duration_mins**(-0.689203))
Parametres de l'esdeveniment':	
	[182.24866421 -0.68920298]
Pluja acumulada per dia (data, precipitació acumulada (mm)):	
2020-04-09 00:00:00,	0.0
2020-04-10 00:00:00,	0.0
2020-04-11 00:00:00,	0.0
2020-04-12 00:00:00,	0.0
2020-04-13 00:00:00,	19.6
2020-04-14 00:00:00,	0.0
2020-04-15 00:00:00,	0.6
2020-04-16 00:00:00,	0.0
2020-04-17 00:00:00,	0.0
2020-04-18 00:00:00,	4.700000000000001
2020-04-19 00:00:00,	63.20000000000002
2020-04-20 00:00:00,	7.599999999999998
2020-04-21 00:00:00,	58.9
2020-04-22 00:00:00,	1.8
2020-04-23 00:00:00,	0.0
2020-04-24 00:00:00,	0.1
2020-04-25 00:00:00,	0.0
2020-04-26 00:00:00,	0.0
2020-04-27 00:00:00,	0.7
2020-04-28 00:00:00,	0.6
2020-04-29 00:00:00,	0.0

Figura 52 Informe descriptiu de l'esdeveniment.

Com que en aquest cas que s'utilitza l'hora amb més precipitació com a hora de l'esdeveniment, i a més a més es determina que sí que hi ha un episodi de pluja, es possible generar la resta de resultats i càlculs corresponent a les intensitats de pluja.

Pel què fa a l'anàlisi de la intensitat de la pluja acumulada per a diferents períodes de mesura (0,5 hores, 24 hores, 3 dies i 7 dies previs a l'esdeveniment, Fig. 53) es desmarquen uns valors molt més elevats d'intensitat per als moments pròxims a l'esdeveniment (0,5 hores i 24 hores) on els valors d'intensitat màxima són: 16.8 i 12.4 mm/h respectivament. Pel que fa a les intensitats mesurades pels períodes de 3 i 7 dies aquestes són menors: 4.7 i 2.8 mm/h respectivament. Aquest increment en la intensitat en moments propers a l'esdeveniment situen la corba d'intensitat que caracteritza la pluja de l'esdeveniment per sobre del llindar crític escollit per l'anàlisi. Això significa que la pluja enregistrada durant el dia 19 d'abril es considerada una pluja crítica amb potencial desencadenant d'una colada de fang.

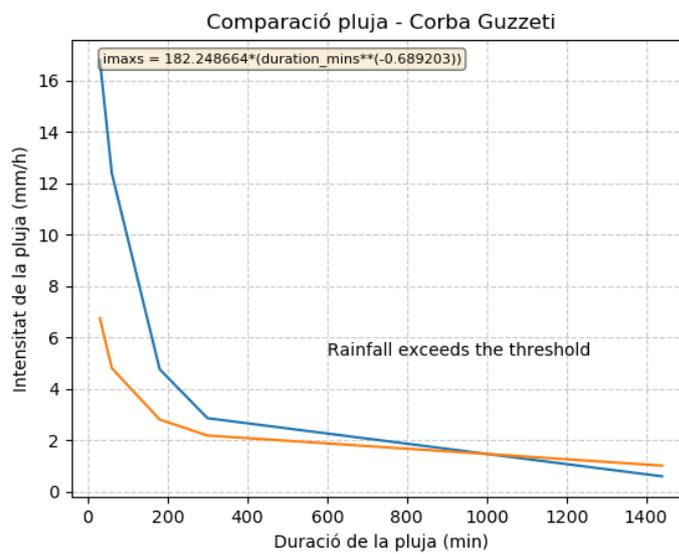


Figura 53 Comparació corba d'intensitat de l'esdeveniment amb la corba de Guzzeti et al., (2008) que defineix els valors llindars d'intensitat. En aquest cas, la pluja que caracteritza l'esdeveniment sobrepassa els llindars establerts.

Complementàriament, el gràfic amb l'evolució de la temperatura mitjana (°C) durant el període d'anàlisi (Fig. 54), situa el dia 19 d'abril en un moment d'inestabilitat, amb un descens generalitzat de les temperatures que dura entre 3 i 4 dies. El resultat de l'anàlisi també incorpora els 3 gràfics de previsió futura del possible risc (Fig. 55) generat per la pluja que caracteritza l'esdeveniment, per a períodes de 24 hores, 3 dies i 7 dies. Tot i no ser del tot necessaris amb l'objectiu amb el qual els usuaris realitzen l'anàlisi (compilació de dades, no anàlisi de risc) poden aportar informació sobre com evoluciona l'impacte de la pluja crítica.

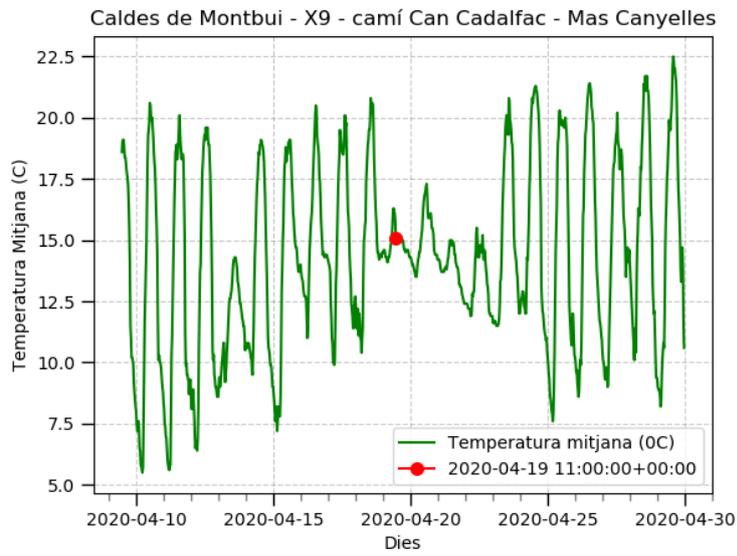


Figura 54 Evolució de la temperatura durant el període d'anàlisi (20 dies).

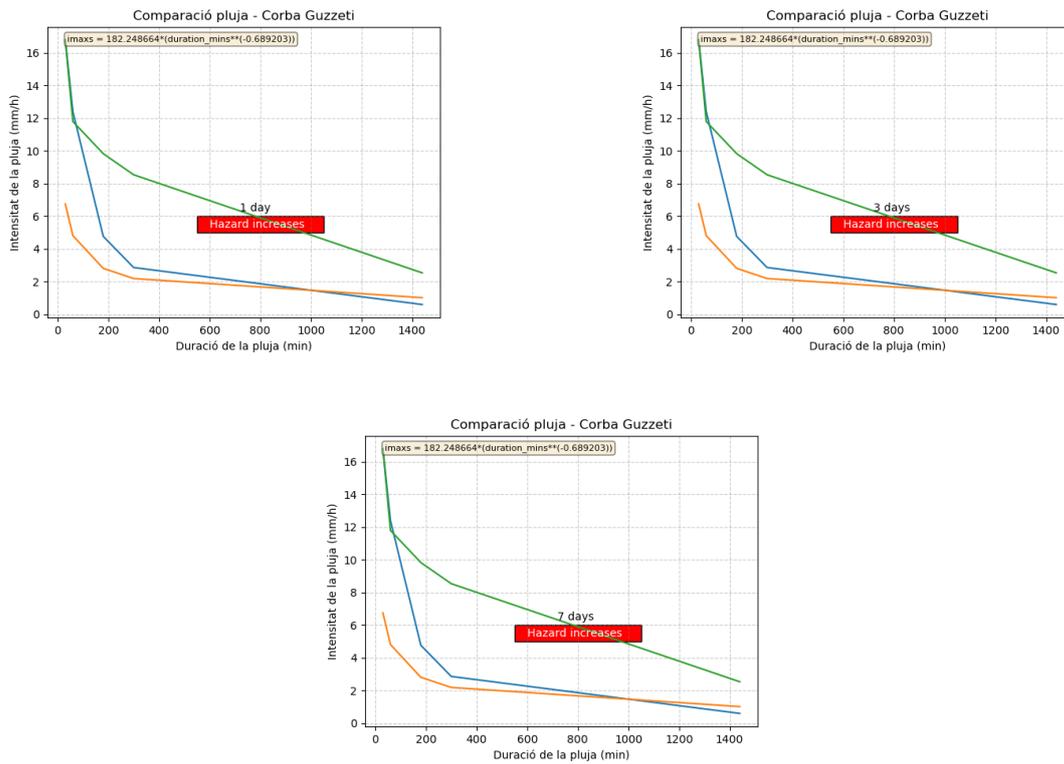


Figura 55 Evolució de la intensitat de la pluja en els gràfics post-esdeveniment per a 24h 3 i 7 dies. La intensitat de l'esdeveniment segueix indicant períodes de perillositat en períodes posteriors.

5. Conclusions

- El nou plugin Local Rainfall Analysis integra el submòdul d'anàlisi de pluges de la plataforma HEIMDALL i posa a disposició dels usuaris finals la possibilitat d'executar l'anàlisi pluviomètric a través de la interfície gràfica d'usuari del plugin.
- A través de les eines Python que ofereix l'API de QGIS, i Qt és possible afegir de manera fàcil, noves funcionalitats al QGIS a través de la consola o escrivint scripts emmagatzemats dins dels plugins.
- El plugin Local Rainfall Analysis permet de manera fàcil i sintètica caracteritzar els episodis de pluja que poden haver desencadenat una esllavissada o un altre moviment en massa, a través de la mateixa interfície del programa, el que permet agilitzar i optimitzar l'entorn de treball de l'usuari destí.
- El plugin, a diferència del simulador d'esllavissades de la plataforma HEIMDALL i el submòdul d'anàlisi de pluges, ha de permetre un anàlisi de les condicions històriques d'episodis de pluja intensos que hagin generat un procés d'esllavissament. En aquest cas, no s'utilitzarà el plugin com una eina de gestió de riscos geològics actuals.
- La intensitat de la pluja en el moment desencadenant d'una possible esllavissada queda subratllat com el paràmetre més característic d'aquesta a l'hora de generar situacions crítiques que afavoreixin al desencadenament d'aquests moments catastròfics.
- L'estudi dels valors llindars d'intensitats crítiques ha estat realitzat per diversos autors i té una gran importància en l'estudi de la iniciació d'esllavissades i fluxos torrencials induïts per pluges intenses. Aquest plugin ha d'aportar informació sobre els paràmetres pluviomètrics de les pluges estudiades relacionades amb el desencadenament d'esllavissades i fluxos torrencials per proporcionar noves dades en l'estudi de valors llindars de pluja de caràcter local.
- Amb els resultats obtinguts a les fases de prova en un entorn real d'esllavissades vinculades a períodes de pluja intenses, els quals analitzen els tècnics del ICGC, es demostra que el plugin *Local Rainfall Analysis* permet millorant l'anàlisi d'esllavissades que es donen recurrentment a Catalunya quan hi ha episodis de pluges intenses.
- Tot i que aquest plugin genera un informe detallat de les condicions meteorològiques que es donen en el moment en el qual es produeix una esllavissada o un altre moviment en massa del terreny i que per tant podrien estar relacionades, la resta de valoracions finals o consideracions sobre la caracterització i perillositat de la pluja en un punt del terreny que puguin desencadenar una esllavissada o un altre tipus de moviment en massa, entenent que entren en joc criteris hidrològics i geològics, queden fora de l'abast d'aquest projecte.

6. Bibliografia

- Abancó, C., Hürlimann, M., Moya, J., & Berenguer, M. (2016). Critical rainfall conditions for the initiation of torrential flows. Results from the Rebaixader catchment (Central Pyrenees). *Journal of hydrology*, 541, 218-229.
- Ahmed, Z., Krupnik, T. J., & Kamal, M. (2018). Introduction to basic GIS and spatial analysis using QGIS: Applications in Bangladesh.
- Anaconda inc. (2020) ANACONDA. www.anaconda.com
- Battiston, S., & Clandillon, S. Instrument Collaborative Project Call/Topic H2020-SEC-2016-2017/H2020-SEC-2016-2017-1 Project Title Multi-Hazard Cooperative Management Tool for Data Exchange, Response Planning and Scenario Building Project Number 740689.
- Battiston, S., Friedemann, M., Gascón, D., Viseras, A., Cardil, A., Mendes, M. A. I., ... & Clandillon, S. (2019). HEIMDALL: A technological solution for multi-hazard management Support including wildfires.
- Bitelli, G., Dubbini, M., & Zanutta, A. (2004). Terrestrial laser scanning and digital photogrammetry techniques to monitor landslide bodies. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(B5), 246-251.
- Chaves, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, 6(10), 1-13.
- Chowdhury, R., & Flentje, P. (2002). Uncertainties in rainfall-induced landslide hazard. *Quarterly Journal of Engineering Geology and Hydrogeology*, 35(1), 61-69.
- Corominas, J., & de Caminos, I. (2004). Tipos de rotura en laderas y taludes. *Ingeniería del Terreno Ingeniería 4*. Ed: López Jimeno, Carlos. Universidad Politécnica de Madrid, 191-213.
- Dikau, R., Brunnsden, D., Schrott, L., & Ibsen, M. L. (1996). Landslide recognition: identification, movement and causes, International Association of Geomorphologists, Publication No. 5, John Wiley and Sons, Chichester, 1996, 251 p.
- Dobias, T. (2011). PyQGIS Developer Cookbook. Retrieved on 13th July.
- Dunning, S. A., Mitchell, W. A., Rosser, N. J., & Petley, D. N. (2007). The Hattian Bala rock avalanche and associated landslides triggered by the Kashmir Earthquake of 8 October 2005. *Engineering Geology*, 93(3-4), 130-144.
- ECHO (Directorate-General for European Civil Protection and Humanitarian Aid Operations; European Commission) (2017): Overview of natural and man-made
- Friedemann, M., Barth, B., Vendrell, J., Mühlbauer, M., & Riedlinger, T. (2018). Conceptual scenario model for collaborative disaster response planning. *Environmental Informatics: Techniques and Trends*, 119-125.
- Gandhi, U. (2018). QGIS Tutorials and Tips. Last updated on Apr, 30.
- Glade, T. (2003). Vulnerability assessment in landslide risk analysis. *Erde*, 134(2), 123-146.
- Graser, A., & Olaya, V. (2015). Processing: A python framework for the seamless integration of geoprocessing tools in QGIS. *ISPRS International Journal of Geo-Information*, 4(4), 2219-2245.
- Guzzetti, F., Ardizzone, F., Cardinali, M., Rossi, M., & Valigi, D. (2009). Landslide volumes and landslide mobilization rates in Umbria, central Italy. *Earth and Planetary Science Letters*, 279(3-4), 222-229.

Guzzetti, F., Peruccacci, S., Rossi, M., & Stark, C. P. (2008). The rainfall intensity–duration control of shallow landslides and debris flows: an update. *Landslides*, 5(1), 3-17.

Hansen, A. (1984). Landslide hazard analysis. *Slope instability*, 523-602.

HEIMDALL Project Homepage (2017): <http://heimdall-h2020.eu/> (20.05.2018).

Hervás, J., Barredo, J. I., Rosin, P. L., Pasuto, A., Mantovani, F., & Silvano, S. (2003). Monitoring landslides from optical remotely sensed imagery: the case history of Tessina landslide, Italy. *Geomorphology*, 54(1-2), 63-75.

Hungr, O., Leroueil, S., & Picarelli, L. (2014). The Varnes classification of landslide types, an update. *Landslides*, 11(2), 167-194.

Institut Cartogràfic i Geològic de Catalunya (2019). Institut Cartogràfic i Geològic de Catalunya. <https://www.icgc.cat/>

Institut Cartogràfic i Geològic de Catalunya (ICGC). (6 de juny de 2017). Notícia 189 – L'ICGC participa en el projecte HEIMDALL. Recuperat de <https://icgc.cat/en/The-ICGC/News3/Noticia-189-L-ICGC-participa-en-el-projecte-HEIMDALL>.

Keefer, D. K. (1994). The importance of earthquake-induced landslides to long-term slope erosion and slope-failure hazards in seismically active regions. In *Geomorphology and Natural Hazards* (pp. 265-284). Elsevier.

Kirschbaum, D., & Stanley, T. (2018). Satellite-based assessment of rainfall-triggered landslide hazard for situational awareness. *Earth's future*, 6(3), 505-523.

Kirschbaum, D., & Stanley, T. (2018). Satellite-based assessment of rainfall-triggered landslide hazard for situational awareness. *Earth's future*, 6(3), 505-523.

Larsen, M. C., & Santiago-Román, A. (2001). Mass wasting and sediment storage in a small montane watershed: an extreme case of anthropogenic disturbance in the humid tropics. *Geomorphic Processes and Riverine Habitat*, 4, 119-138.

Malamud, B. D., Turcotte, D. L., Guzzetti, F., & Reichenbach, P. (2004). Landslide inventories and their statistical properties. *Earth Surface Processes and Landforms*, 29(6), 687-711.

Oxoli, D., Prestifilippo, G., & Bertocchi, D. (2017). Enabling spatial autocorrelation mapping in QGIS: The hotspot analysis Plugin.

Polemio, M., & Petrucci, O. (2000). Rainfall as a landslide triggering factor an overview of recent international research. In *Landslides in research, theory and practice*. Thomas Telford Ltd..

QGIS Development Team. (2016). QGIS geographic information system. *Open source geospatial Foundation project*.

Sendra, J. B., & García, R. C. (2000, January). El uso de los sistemas de Información Geográfica en la planificación territorial. In *Anales de Geografía de la Universidad complutense* (Vol. 20, p. 49).

Sherman, G. (2014). *The PyQGIS Programmer's Guide*. Locate Press.

Skempton, A. W., & Hutchinson, J. (1969). Stability of natural slopes and embankment foundations. In *Soil Mech & Fdn Eng Conf Proc/Mexico/*.

Spyder Website Contributors. (2018). SPYDER. <https://www.spyder-ide.org/>

Tarchi, D., Casagli, N., Fanti, R., Leva, D. D., Luzi, G., Pasuto, A., ... & Silvano, S. (2003). Landslide monitoring by using ground-based SAR interferometry: an example of application to the Tessina landslide in Italy. *Engineering geology*, 68(1-2), 15-30.

Terzaghi, K. (1950). Mechanism of landslides. *Application of geology to engineering practice, Geol. Soc. Am*, 83-123.

Vaunat, J., & Leroueil, S. (2002). Analysis of post-failure slope movements within the framework of hazard and risk analysis. *Natural Hazards*, 26(1), 81-107.

Westra, E. (2014). *Building mapping applications with QGIS*. Packt Publishing Ltd.

Zambelli, P., Gebbert, S., & Ciolli, M. (2013). Pygrass: An object oriented python application programming interface (API) for geographic resources analysis support system (GRASS) geographic information system (GIS). *ISPRS International Journal of Geo-Information*, 2(1), 201-219.

7. Annex

7.1. Anàlisi de requeriments

Ubicació del plugin al QGIS

Codi: REQ_FUN_001	Ubicació del plugin
El plugin ha de ser accessible sempre des de la pestanya de complements de QGIS 3, i si és necessari o l'usuari ho prefereix, també ha de ser accessible des de les barres d'eines modificables per l'usuari.	
Requeriments pare:	

Codi: REQ_FUN_001_001	Icona del plugin
El plugin ha d'estar representat amb una icona visible i fàcil de distingir, per facilitar l'usuari la cerca del plugin a l'apartat de complements del QGIS.	
Requeriments pare: REQ_FUN_001	

Codi: REQ_FUN_001_002	Nom del plugin
El plugin ha d'estar representat sota un nom sintètic i fàcil d'entendre , relacionar amb la funcionalitat i objectiu d'aquest, i possiblement fàcil de recordar.	
Requeriments pare: REQ_FUN_001	

Codi: REQ_FUN_002	Access al plugin
-------------------	------------------

<p>La icona i nom del plugin establerts (representants del plugin en la pestanya de complements o barra d'eines de QGIS) han de ser els desencadenants d'accés a la interfície del plugin i al seu conjunt a través d'un clic d'activació.</p>
<p>Requeriments pare:</p>

<p>Codi: REQ_FUN_002_001</p>	<p>Capturador de coordenades</p>
<p>L'activació i inici del plugin (a través del clic de l'usuari) ha de facilitar a l'usuari capturar les coordenades del esdeveniment a analitzar a través d'un sol clic al llenç del mapa (utilitzant la capa que vulgui, no te que ser la capa d'estacions la capa activa).</p>	
<p>Requeriments pare: REQ_FUN_002</p>	

Interfície Gràfica d'Usuari (GUI)

<p>Codi: REQ_FUN_003</p>	<p>Inici de la GUI</p>
<p>La GUI s'activarà i per tant apareixerà a al llenç de l'aplicació del QGIS quan l'usuari seleccioni la icona del plugin.</p>	
<p>Requeriments pare:</p>	

<p>Codi: REQ_FUN_003_001</p>	<p>Introducció de paràmetres a la GUI</p>
<p>La GUI ha de permetre a l'usuari introduir el conjunt de paràmetres necessaris per a la següent execució de l'anàlisi. La inserció dels paràmetres ha de ser de forma ordenada i sintètica que faciliti la comprensió del que requereix el programa per a funcionar. La GUI mostrarà camps editables de text o desplegable que l'usuari podrà omplir o seleccionar les diverses variables</p>	
<p>Requeriments pare: REQ_FUN_003</p>	

Codi: REQ_FUN_003_001_001	Introducció del nom de l'esdeveniment
<p>El primer paràmetre que la GUI ha de permetre introduir és el nom de l'esdeveniment que l'usuari cregui corresponent. Aquest nom acompanyarà el conjunt de paràmetres i resultats al llarg de la anàlisi com a nom representant el conjunt, per tal de fer fàcil i comprensible el conjunt de processos i resultats que corresponen a un sol anàlisi.</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	

Codi: REQ_FUN_003_001_002	Introducció de les coordenades
<p>El segon paràmetre que la GUI ha de permetre introduir son les coordenades de l'esdeveniment (X i Y). Per introduir les coordenades l'usuari ha de poder escriure-les manualment o utilitzar un captador de coordenades que a través d'un clic al llenç del mapa siguin capturades i introduïdes com a paràmetre. Hi ha d'haver-hi els botons corresponents per activar tal eina a la GUI.</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	

Codi: REQ_FUN_003_001_003	Introducció de la data de l'esdeveniment
<p>El tercer paràmetre a introduir és la data de l'esdeveniment. La GUI ha de permetre a l'usuari introduir manualment l'any, mes, dia i hora.</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	

Codi: REQ_FUN_003_001_003_001	Auto determinar hora
<p>La GUI ha de permetre la opció de determinar i utilitzar la hora amb la màxima acumulació de pluja del dia introduït per l'usuari. A la GUI aquesta opció es representarà com un desplegable amb les opcions Activar i Desactivar, on l'usuari podrà determinar si vol utilitzar la data amb l'hora introduïda manualment o vol que el programa automàticament calculi l'hora del dia de l'esdeveniment amb màxim d'acumulació de pluja i utilitzi aquesta darrera hora.</p>	
<p>Requeriments pare: REQ_FUN_003_001_003</p>	

Codi: REQ_FUN_003_001_004	Selecció de l'estació meteorològica
<p>El cinquè paràmetre que ha de permetre la GUI d'introduir es el codi d'estació amb el qual l'usuari vol analitzar les dades. EN aquest cas la GUI mostrarà un desplegable amb el total de codis i noms d'estacions disponibles a més de la opció de calcular la estació mes propera segons les coordenades (en aquest cas es calcula automàticament). Per defecte la GUI oferirà la opció de calcular l'estació més propera. En el cas que l'usuari prefereixi una estació en concret podrà navegar el desplegable i escollir la que necessiti.</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	

Codi: REQ_FUN_003_001_005	Shapefile amb les estacions meteorològiques
<p>La GUI ha de permetre a l'usuari l'opció de carregar una capa en format <i>shapefile</i> al projecte de QGIS que contingui la ubicació i simbolització del conjunt d'estacions meteorològiques de la xarxa MeteoCat amb la qual treballa el plugin.</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	

Codi: REQ_FUN_003_001_006	Selecció del valor del llindar climàtic
<p>La GUI ha de permetre a l'usuari l'opció de seleccionar el llindar climàtic que prefereixi per realitzar l'anàlisi a partir d'un desplegable. Els tres valors llindars els quals han de ser extensibles per l'usuari són: 1) Mediterrani fred, 2) Mediterrani temperat i 3) Muntanya. Per defecte el valor del llindar és mediterrani temperat.</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	
Codi: REQ_FUN_003_001_007	Selecció del directori destí per els resultats
<p>El sisè i últim paràmetre que la GUI ha de permetre introduir es la ruta al directori que l'usuari vulgui per guardar els resultats. Es crearà un navegador de carpetes per tal de que l'usuari pugui buscar-la i escollir-la. Per defecte, el nom dels arxius resultants serà el mateix que el nom de l'esdeveniment (prèviament escollit) i un complement que automàticament s'afegirà per distingir els resultats. El navegador de directoris ja introduirà el nom de l'esdeveniment introduït a la GUI i l'usuari no cal que el modifiqui (a no ser que vulgui).</p>	
<p>Requeriments pare: REQ_FUN_003_001</p>	

Codi: REQ_FUN_003_001_006	Execució del plugin
La GUI ha de contenir dos botons que permetin (1) cancel·lar la GUI i sortir del plugin o bé (2) acceptar els paràmetres introduïts i executar l'anàlisi utilitzant-los.	
Requeriments pare: REQ_FUN_003_001	

Resultats

Codi: REQ_FUN_003	Resultats
Un cop l'usuari ha executat el plugin a través de la GUI, el plugin ha de generar els resultats desitjats per l'usuari, aquests, es presentaran en format gràfic <i>.png</i> i amb arxius format <i>.csv</i>	
Requeriments pare:	

Codi: REQ_FUN_003_001	Informe esdeveniment
L'informe de l'esdeveniment ha de contenir la informació corresponent (en format <i>.csv</i>) a les característiques del esdeveniment en qüestió tals com un resum dels paràmetres d'entrada i un resum sintètic dels resultats obtinguts amb l'anàlisi.	
Requeriments pare: REQ_FUN_003	

Codi: REQ_FUN_003_002	Taula de dades de pluja
La taula de dades de pluja ha de ser l'arxiu generat que el mòdul inicial Meteocat_MTIG ja incorpora, en aquest cas en format <i>.csv</i> , on hi ha el valor de precipitació per a cada unitat de temps durant el període d'anàlisi	
Requeriments pare: REQ_FUN_003	

Codi: REQ_FUN_003_003	Gràfics
<p>El conjunt de gràfics que genera el segon mòdul LocalRainfallAnalysis_MTIG s'ha de conservar però haurà d'albergar totes les modificacions necessàries que sol·liciti l'usuari, tant com l'eliminació, modificació o adició d'elements en ells.</p>	
<p>Requeriments pare: REQ_FUN_003</p>	

7.2. Anàlisi del codi existent

7.2.1. Submòdul meteocat_MTIG.py

Inici del mòdul

Aquest submòdul esta format per un total de vuit funcions les quals són cridades i desencadenades a partir de la funció principal del submòdul, [1], a la qual s'hi introdueixen els paràmetres externs (*input*) per a l'execució de la primera part de l'anàlisi de pluges:

[1] `read_rainfall_event_date(event_x, event_y, event_datetime, auto_determine_hour=False, days_range=20, epsg=25831, out_file=None)`

Obté dades de pluja del MeteoCat a partir d'una coordenada i un rang de dates.

Retorna:

(1) `event_datetime`: data hora de l'event.

(2) `datetimered`: una llista de dates.

```
[datetime.datetime(2020, 1, 13, 3, 0, tzinfo=<UTC>),
datetime.datetime(2020, 1, 13, 3, 30, tzinfo=<UTC>),
datetime.datetime(2020, 1, 13, 4, 0, tzinfo=<UTC>),
datetime.datetime(2020, 1, 13, 4, 30, tzinfo=<UTC>),
datetime.datetime(2020, 1, 13, 5, 0, tzinfo=<UTC>),
datetime.datetime(2020, 1, 13, 5, 30, tzinfo=<UTC>),
```

(3) `rainfall`: una llista de pluges per cada data.

```
0.8, 1.8, 2.1, 2.4, 2.9, 3.5, 2.2, 3.3, 2.5, 1.5, 0.5, 0.8,
0.3, 0.4, 0.6, 0.5, 0.2, 0.0, 0.1, 0.0, 0.0, 0.0, 0.1, 1.6,
2.2, 3.4, 6.0, 6.3, 6.0, 5.7, 3.2, 2.4, 1.7, 0.3, 0.0, 0.0,
0.0, 0.0, 0.1, 0.1, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.5, 1.2, 0.9, 2.1, 1.1, 0.2, 1.2, 0.1, 0.2,
```

(4) `step_hours`: interval en hores de cada valor de pluja.

Es a dir, aquesta funció actua com a receptora de paràmetres de l'usuari, fa les crides al subseqüent conjunt de funcions que completen el mòdul i la funció d'obtenir les dades de MeteoCat i retorna a l'usuari el conjunt de valors esperats.

Els paràmetres d'entrada o *input* de la funció principal [1] són: les coordenades x i y de l'esdeveniment en qüestió (`event_x` i `event_y`), la data concreta de l'esdeveniment (`event_datetime`), condició d'activar l'anàlisi de l'hora del dia de l'esdeveniment amb més precipitació, *True* o *False* (`auto_determine_hour`), el conjunt de dies a analitzar (`days_range`) normalment per defecte, el sistema de coordenades de referència per defecte (`epsg`) el qual és ETRS89/UTM zone 31N i el nom de l'arxiu de sortida si es requereix (`out_file`). Tot aquest conjunt de paràmetres clau per a l'anàlisi s'especificaran al llarg de la documentació del mòdul.

Els valors de les quatre variables que retorna la funció principal del submòdul representen el resultat esperat del submòdul (*output*): (1) la data concreta de l'esdeveniment (any, mes, dia, hores i minuts) en format objecte `datetime.datetime` (`event_datetime`), (2) una llista amb un total

de 1002 entrades de dates (`datetimered`) calculades a partir de la data de l'esdeveniment, que representen un període de 20 dies (`days_range = 20`), en intervals de 30 minuts, entorn de la data central de l'esdeveniment dividides per a cada mitja hora. (3) Una llista amb 1002 valors de precipitació (mm) recuperats dels servidors de MeteoCat en funció una estació meteorològica en concret (`rainfall`), aquests valors de pluja s'associen a cada data en la llista de dates, per tant, per a cada mitja hora en un període de 20 dies te associat el seu valor de pluja, (4) representa el valor pel qual es divideix el conjunt de dies (`step_hours = 0,5`), en aquest cas per a cada mitja hora representats per el valor de la fracció de 0.5.

Càlcul de l'estació més propera

Per tal d'elaborar aquest resultat i retornar els conjunts de valors desitjats, [1] fa les crides corresponents al conjunt de funcions dependents que construeixen l'esquelet del submòdul. Les dades de pluja s'obtenen a partir del servei de MeteoCat, el qual alhora permet seleccionar les dades per a una estació meteorològica en concret dins la seva xarxa d'estacions amb un total de 189 estacions repartides per a tot Catalunya. Aquestes estacions i la seva localització són accessibles en format d'un arxiu *shpefile*. A partir dels paràmetres de coordenades de l'esdeveniment (`event_x` i `event_y`) que rep [1], aquesta crida a la funció [2] la qual calcula l'estació més propera respecte les coordenades introduïdes:

[2] **get_nearest_station**(`event_x`, `event_y`, `epsg=25831`)

Obté la estació meteorològica més propera a les coordenades

Retorna:

(1) `min_distance`: Distància a l'estació més propera.

(2) `station_code`: Codi d'estació

La funció permet calcular la distancia mínima entre l'esdeveniment i l'estació en concret a gràcies a la llibreria *ogr* i *osr* que permeten accedir a la capa *shpefile* i abstrure la informació espacial. A partir dels valors que retorna les variables de [2] queda determinada una referència a l'estació de MeteoCat en concret amb la qual es sol·licitaran el conjunt de dades de pluja. [2] utilitza el sistema de coordenades de referència EPSG: 25831 ETRS89 / UTM zone 31N en el cas que s'introduís SCR diferent als paràmetres de [1], [2] incorpora un transformador de coordenades.

Determinar l'hora amb més acumulació de pluja (mm)

Un cop [1] ha incorporat el codi d'estació amb la qual es reproduirà l'anàlisi, hi ha la possibilitat de redefinir l'hora del dia de l'esdeveniment (`event_datetime`) amb l'hora de major acumulació de pluja (el valor de pluja en un total de 48, en intervals de mitja hora, i que per tant, pot ser diferent a la introduïda). Aquest procés proporciona a l'usuari la possibilitat de buscar màxims de pluja en el cas que es desconeixin les condicions meteorològiques en el moment de l'esdeveniment. En el cas que es demani obtenir l'hora del dia amb major acumulació de pluja (`auto_determine_hour=True` en els paràmetres d'entrada del mòdul) [1] fa la crida a la funció [3] la qual determina l'hora del dia de l'esdeveniment amb la màxima acumulació de pluja per a una determinada estació de MeteoCat:

[3] **get_max_rainfall_hour**(`station_code`, `event_datetime`)

Obté l'hora del dia que té la pluja màxima per una determinada estació del metecat

Retorna:

(1) `event_datetime`: recalcula l'hora del dia de l'esdeveniment amb més acumulació de pluja.

Si s'indica com a necessari calcular i determinar l'hora del dia de l'esdeveniment amb més precipitació [1] fa la crida a [3], en el cas que no sigui necessari i per tant no s'hagi indicat, [1] crida directament a [4] evitant aquest pas. Per tal de determinar l'hora del dia de l'esdeveniment amb més acumulació (mm), [3] determina el valor màxim d'acumulació a partir de la creació d'una llista amb on s'hi afegeixen els valors d'acumulació de pluja per tot el dia (valors de precipitació per a cada mitja hora, per tant, 48 valors). Per tal de determinar l'inici i el final del període d'anàlisi, es defineix una data d'inici i final, essent el límit inicial en aquest cas les 00:00h del dia seleccionat i el final a les 23:30h. Per tal de definir i construir el conjunt de dates que s'utilitzaran per a capturar el conjunt de valors de precipitació, s'utilitza la classe `datetime` important la llibreria `datetime`. Per tant, [3] redefineix de nou el paràmetre d'entrada de [1] `event_datetime`, la data de l'esdeveniment definida per l'usuari, utilitzant l'hora del dia que te vinculada el màxim d'acumulació i el retorna com a nova data de l'esdeveniment. Per de poder calcular el màxim d'acumulació, [3] crida a [4] la qual retorna una llista de dates (corresponents al dia de l'esdeveniment) i una llista de valors d'acumulació (mm) per a cada data:

[4] `read_rainfall_date_range(station_code, begin_datetime, end_datetime, out_file=None)`

Obté dades de pluja del MeteoCat a partir d'una estació meteorologica i un rang de dates

Retorna:

- (1) `datetimered`: llista de dates amb intervals de mitja hora (48 entrades per dia)
- (2) `rainfall`: llista de valors d'acumulació per a cada mitja hora (mm) (48 entrades per dia)
- (3) `step_hours`: defineix el valor a dividir de les entrades a la llista.

[4] es caracteritza per crear les dues llistes necessàries per a determinar la pluja que hi ha hagut durant el període d'anàlisi que determinen les dates límit d'inici i final del període definides en [3]. Una llista correspon al conjunt de dates (que s'omple prèviament a la sol·licitud de dades als servidors de MeteoCat) i una llista amb el conjunt de valors de precipitació que corresponen a cada data (valors de precipitació per a cada mitja hora, s'obtenen a partir de les trucades als servidors de MeteoCat). El rang de dates es defineix a partir de les dates límit d'inici i final, condicionades per la data de l'esdeveniment, l'interval d'hores a dividir (`step_hours = 0,5`) i els dies a analitzar (`days_range = 20`). La llista de valors d'acumulació que [4] retorna a [1] s'omple a partir del diccionari de dades que retorna la funció [5] en funció del rang de dates (`datetimered`) i la resposta del servidor de MeteoCat. D'aquesta manera [4] crida a [5] per obtenir els valors resultants de la trucada al servidor.:

[5] `read_data_date_range(station_code, begin_datetime, end_datetime, out_file=None)`

Obté dades del MeteoCat a partir d'una estació meteorologica i un rang de dates

Retorna:

(1) `data_dict`: Diccionari de valors obtinguts a partir de la consulta als servidors de MeteoCat.


```
(datetime.datetime(2020, 2, 2, 0, 0), '09:30 - 10:00',
103.0), (datetime.datetime(2020, 2, 2, 0, 0), '10:00 -
10:30', 147.0), (datetime.datetime(2020, 2, 2, 0, 0), '10:30
```

(2) step_hours: defineix el valor a dividir de les entrades a la llista.

[5] afegeix i emmagatzema el conjunt de valors de pluja per a un sol dia (dia de l'esdeveniment, per tal de calcular l'hora amb màxima precipitació per a retornar el valor a [1]) o per a més dies (veure més endavant). [5] genera un diccionari que albergarà el conjunt de valors de precipitació per a cada data assignada. Aquest diccionari s'omple a partir del resultat de cada trucada al servidor de MeteoCat i es el conjunt "cru" de valors (data + valor de precipitació) que [4] filtrarà i segregará en les dues llistes. Per tal d'omplir el diccionari de dates, [5] crida a [6] comunicant-li el dia a sol·licitar les dades (no el conjunt de dates segregades cada mitja hora) degut a que, [6] fa la trucada als servidors utilitzant un "url" on es defineixen els criteris de cerca a partir del dia en qüestió, un exemple de trucada al servidor i dades disponibles a MeteoCat és el següent:

<http://www.meteo.cat/observacions/xema/dades?codi=VP&dia=2020-01-13T00:00Z>

Aquest url és generat per [6] amb la data comunicada per [5]. Les dades que MeteoCat ofereix per a aquesta consulta són les següents (Fig. A1):

▼ Període TU	TM °C	TX °C	TN °C	HRM %	PPT mm	VVM (6 m) km/h	DVM (6 m) graus	VVX (6 m) km/h	RS W/m²
00:00 - 00:30	-1.3	-0.8	-1.7	98	0.0	3.6	238	5.8	0
00:30 - 01:00	-1.4	-0.4	-2.3	97	0.0	1.4	209	6.8	0
01:00 - 01:30	-1.8	-1.4	-2.2	98	0.0	0.4	239	2.9	0
01:30 - 02:00	-1.7	-0.9	-2.2	98	0.0	0.0	310	1.8	0
02:00 - 02:30	-2.4	-1.6	-2.9	99	0.0	1.1	269	4.7	0
02:30 - 03:00	-2.4	-2.2	-2.8	100	0.0	0.4	197	2.9	0
03:00 - 03:30	-1.7	-1.2	-2.2	100	0.0	1.8	199	6.1	0
03:30 - 04:00	-1.9	-1.5	-2.2	100	0.0	0.4	235	2.5	0
04:00 - 04:30	-2.2	-2.0	-2.5	100	0.0	0.7	248	2.9	0
04:30 - 05:00	-2.4	-1.7	-3.2	99	0.0	4.0	214	9.0	0
05:00 - 05:30	-2.7	-2.0	-3.2	97	0.0	1.1	229	4.3	0
05:30 - 06:00	-3.1	-2.9	-3.2	98	0.0	0.0	321	1.8	0

Figura A 1 Taula de dades disponible als servidors de MeteoCat per a un dia en concret. Es ressalten els valors de precipitació (PPT) i temperatura mitjana (TM) objectius d'aquest mòdul.

El fet que el mòdul meteoCAT_MTIG.py utilitzi el conjunt de dates i valors de precipitació en intervals de mitja hora és degut a que els servidors de MeteoCat treballen amb el mateix sistema. Per a cada dia especificat en l'url que genera [6] MeteoCat respon amb la taula (Fig. A 1) que es construeix a partir de 48 valors de temperatura (mitjana, màxima i mínima: TM, TX i TN), humitat relativa (HRM), precipitació (PPT) i altres conjunts de valors:

[6] `read_data_day(station_code, event_date, out_file=None)`

Obté dades del MeteoCat a partir d'una estació meteorològica i un dia

Retorna

(1) Header_list: el capçal de la llista de valors obtinguts al MeteoCat.

(2) Data_dict: el diccionari buit de valors a omplir amb el resultat de la consulta als servidors.

[6] fa la petició a l'enllaç url especificat per als servidors de MeteoCat. Amb el qual es fa una protecció a la crida en el cas que hi hagi risc d'error en la connexió a Internet o en el cas de que la data de la petició estigui fora d'abast (més de 2 anys enrere, temps màxim en el qual els servidors de MeteoCat disposen públicament de les dades) utilitzant les llibreries *urllib* (concretament *urllib.error* i *urllib.request*). A partir dels valors del capçal del diccionari de valors (*data_dict*) cerca els valors al servidor (utilitzant puntualment la llibreria *html*), a mesura que s'obtenen es retornen a [5] la qual les filtra i afegeix al diccionari de dades. A partir d'aquest diccionari (concatenació de dia, rang de dates per cada mitja hora i el seu valor de precipitació) [4] podrà segregar en dos llistes diferents les hores d'anàlisi per el període establert i els valors de precipitació per a cada mitja hora. Un cop s'han segregat els valors en dues llistes diferents, [3] podrà calcular el màxim de precipitació dins la llista i detectar a quina hora correspon el pic d'acumulació i retornarà a [1] la nova data d'esdeveniment (*event_datetime*) amb la qual podrà realitzar l'anàlisi de pluges per al període de 20 dies.

Obtenció de dades de pluja

Si bé es fa una consulta al servidor de MeteoCat per a la consulta de quina hora té més acumulació durant el dia de l'esdeveniment (*event_datetime*) i es realitza generant un diccionari de dades que posteriorment es segrega amb dues llistes de valors, l'esquelet i motiu principal de la configuració i arquitectura del submòdul està destinada a obtenir un conjunt de valors que representin no només el moment i/o dia d'esdeveniment i la seva precipitació, sinó que es construeix un conjunt de valors d'abast pre- i post- esdeveniment, es a dir, un conjunt de valors en un rang de 20 dies on la data central és la data de l'esdeveniment, i que per tant la informació sobre les condicions de pluja per el dia determinat com a dia de l'esdeveniment (*event_datetime*) es complementaran amb la informació de la precipitació 10 dies abans i 10 dies després de l'esdeveniment.

La metodologia amb la qual el mòdul genera l'informe que conte una llista amb valors pre- i post-esdeveniment definint un abast de 20 dies entorn la data de l'esdeveniment en posició central no difereix en excés de la utilitzada per a obtenir la llista de valors per a calcular la màxima precipitació durant el dia de l'esdeveniment. Per tant, el mòdul segueix el mateix flux de treball que el cas anterior, generant un diccionari de dades que s'obtenen a partir de trucades als servidors de MeteoCat, i posteriorment es generen les dues llistes de dates i valors de precipitació. L'única diferència es que es realitzaran 20 trucades als servidors i en comptes d'unes llistes amb 48 valors, hi haurà unes llistes amb 1006 valors (20 dies en intervals de 30 min).

Indiferentment o no de si es considera necessària la determinació de la hora amb màxima acumulació o no (per tant segueix la data de l'esdeveniment definida en els paràmetres inicials de [1]), [7] defineix de nou els límits del període de temps a analitzar (en el cas que no es calculi l'hora de màxima acumulació, aquests es defineixen per primera vegada, i es defineixen pels 20 dies), en comptes de tenir un diferencial de gairebé 24h com en el cas anterior, aquests límits estan separats per 20 dies de diferència, mantenint la data de l'esdeveniment (*event_datetime*) en la posició central. [1] envia a [7] el paràmetre *days_range*, en aquest cas igual a 20, i es el paràmetre responsable d'establir l'abast de l'anàlisi. [7] retorna a [1] el valor dels límits de dates de durada de l'anàlisi:

[7] **get_date_range**(*event_datetime*, *days_range*=20)

Obté un rang de dates a partir d'una data central i un nombre de dies (repartits la meitat davant i meitat darrere)

Retorna:

(1) *begin_datetime*: limit inferior de dates a realitzar l'anàlisi.

(2) end_datetime: limit superior de dates a realitzar l'anàlisi.

Un cop definit el període de dates a obtenir els seus valors de precipitació els servidors de MeteoCat (a diferència dels passos anteriors, es sol·liciten valors corresponents a 20 dies, no a un sol dia), la funcionalitat que defineix la dependència de les funcions és igual que en el cas de determinar l'hora de màxima acumulació, sent aquesta: [1] crida a [4] (un cop [7] retorna els valors de dates límit), [4] a [5], [5] a [6], [6] a [8]. Tot i ser un procés semblant pel què fa a les trucades entre les funcions, el conjunt de valors amb el que es treballa és més extens i varia el comportament en certes funcions:

- Les llistes de valors de precipitació (rainfall) i llista de dates (datetimered) de [5] estan compostes per 1002 entrades i no per 48 entrades (representant un sol dia).
- [6] fa la crida a [7] 20 vegades, el que significa que per cada dia [7] farà la petició als servidors de MeteoCat i retornarà els valors per omplir el diccionari de dades (concatenant data, hora i precipitació).

Exportar les dades de pluja

Com a conjunt de valors a recuperar, els quals són les dates del període de temps analitzat i el seu valor de precipitació per a cada mitja hora, [5] crida a [9] on envia el diccionari de dades (dates concatenades amb els valors de precipitació) per tal de exportar-se en format .json o .csv. El format de sortida és la concatenació de: any, mes, dia i hora seguit del valor de precipitació (mm) com a cadena de text separat per comes (,):

save_data(station_code, header_list, data_dict, out_file, data_filter_list=None, step_hours=0.5)

Guarda la informació de pluja en un arxiu .csv o .json.

Les dades es poden filtrar passant una llista d'identificador de dades (del diccionari) a guardar

7.2.2. Submòdul localrainfallanalysis_MTIG.py

Inici del mòdul

La interpretació de l'estructura i de la dependència de les funcions esta basada en 11 de les 24 funcions, degut a que aquestes funcions o bé representen la possibilitat de llegir valors d'entrada que no es posseeixen o generen valors de sortida i resultats en formats no desitjats. El submòdul esta compost, igual que en el primer submòdul, per una crida a una de les funcions principals d'a), en i que desencadena les crides al conjunt de funcions del submòdul:

[1] **read_rainfall_event_date**(event_name, event_x, event_y, threshold_climatic_area, event_datetime, auto_determine_hour=False, out_10days_file=None, out_cummlated_file=None, out_event_file=None, out_intensity_file=None, out_resume_json_file=None, out_resume_lite_json_file=None, out_rainfall_file=None)

Obtenció de les dades de pluja del MeteoCat processades pel submòdul meteocat_MTIG.py

Retorna:

(1) [3] **local_rainfall_analysis_from_data**(event_name, dateandtimered, rainfall, step_hours, threshold_climatic_area, event_datetime, out_10days_file, out_cummlated_file, out_event_file, out_intensity_file, out_resume_json_file, out_resume_lite_json_file)

El conjunt de paràmetres d'entrada d'aquesta funció principal comparteixen identitat amb el mòdul anterior (event_x, event_y, event_datetime, auto_determine_hour). Tot i això en aquest cas apareixen nous paràmetres com ara el nom de l'esdeveniment (event_name), el llinar d'àrea climàtica on apareix l'esdeveniment (threshold_climatic_area) i el conjunt d'arxius de sortida (out_10days_file, out_cummlated_file, out_event_file, out_intensity_file, out_resume_json_file, out_resume_lite_json_file, out_rainfall_file). El conjunt de paràmetres s'explicaran explícitament al llarg de la documentació del mòdul.

[1] crida a [2] on bàsicament demana el conjunt de valors de sortida del primer submòdul (event_datetime, datetimered, rainfall, step_hours). Com anteriorment s'ha explicat, consisteix en fer la crida i execució del submòdul meteocat_MTIG per obtenir els seus valors de sortida. Això implica que el segon submòdul importa el primer submòdul:

[2] **read_rainfall_meteocat**(event_x, event_y, event_datetime, auto_determine_hour=False, days_range=20, epsg=25831, out_file=None)

Llegeix les dades de pluja del MeteoCat de l'estació més propera a les coordenades entre un rang de dates (per defecte 20 dies, 10 endavant i 10 enrera).

Retorna:

(1) **meteocat_MTIG.read_rainfall_event_date**(event_x, event_y, event_datetime, auto_determine_hour, days_range, epsg, out_file)

Un cop [2] recupera els valors del primer submòdul, [1] te com a retorn [3] (similar el que fa [2] amb ella mateixa). Això significa que una funció pot tornar valors al procés principal a través de la instrucció retorn, i que per tant [1] i [2] no retornen una variable o un conjunt de valors sinó que retornen la crida a una altra funció. En el moment que [1] obté els valors de dates i pluja del submòdul meteocat_MTIG crida a [3] la qual es responsable de desencadenar les crides a les funcions que s'agrupen en els blocs c) i d) de càlculs de dades i construcció de gràfics respectivament:

[3] **local_rainfall_analysis_from_data**(station_code, event_name, datetimered, rainfall, rainfall_step_hours, threshold_climatic_area, event_datetime, out_10days_file=None, out_cummlated_file=None, out_event_file=None, out_intensity_file=None, out_file_2=None)

Genera els arxius corresponents a l'anàlisi de pluges.

[3] és l'eix principal del mòdul d'anàlisi de pluges. A partir de les dades inicials obtingudes del submòdul anterior (event_datetime, rainfall, datetimered i step_hours) procedeix als càlculs estadístics (mitjançant la crida a les funcions del grup c)) que calculen, per el període analitzat (20 dies en funció de la data de l'esdeveniment: 1) evolució de la pluja durant el període d'anàlisi, 2) pluja acumulada en períodes previs a l'esdeveniment de 1 dia, 3 dies i 7 dies, 3) acumulació de pluja durant l'esdeveniment, 4) la màxima intensitat de la pluja per un interval de temps prèvi

a l'esdeveniment i 5) la intensitat de la pluja post esdeveniment en períodes de 1 dia, 3 dies i 7 dies. Un cop obtinguts aquests valors derivats de les llistes inicials de valors de precipitació (rainfall), el grup d) de funcions generen el sistema de gràfics que documentaran i seran utilitzats de valors de sortida per al submòdul, i caracteritzaran el resultat i l'expressió del submòdul d'anàlisi de pluges de HEIMDALL.

1) Evolució de la pluja durant el període d'anàlisi

Les dades que provenen de meteocat_MTIG.py corresponen a un total de 1006 entrades que representen valors de pluja durant un període de 20 dies d'anàlisi en intervals de mitja hora, on el període corresponent a l'esdeveniment, es defineix com a la data concreta indicada (event_datetime) que correspon a la posició central del període d'anàlisi. Tot i això, son valors dependents del comportament meteorològic durant el període analitzat, d'aquesta manera l'anàlisi pot dur-se a terme per un període on no hi ha hagut precipitació o bé amb una precipitació molt abundant. Per tant, els valors de la llista de precipitació que corresponen al període de l'esdeveniment poden oscil·lar entre 0, els quals indiquen que no hi ha hagut acumulació i per tant no hi ha hagut precipitació o valors positius els quals indiquen l'acumulació real de pluja. El primer condicionant que aplica [3] és definir que es considera el període de pluja de l'esdeveniment (no els 20 dies sencers) i si hi ha hagut pluja durant aquest, i en el cas de que si, quina quantitat.

Per fer tal consideració, [3] crida a [4] la qual calcula si hi ha un episodi de pluja durant la data de l'esdeveniment (event_datetime). Per tal de fer tal càlcul, el primer que fa [4] es determinar la posició de la data de l'esdeveniment (event_index) dins el llistat de valors de pluja (rainfall). Un cop determinada la posició de l'esdeveniment en el període de 20 dies, s'estableix el criteri per a determinar si en el moment de l'esdeveniment (event_datetime i event_index) hi ha hagut pluja. [4] genera dues llistes per calcular els valors d'acumulació 2 hores abans i 2 hores després de l'esdeveniment. En el cas que la llista de valors prèvia a l'esdeveniment i la següent tinguin valors positius, es considera que la data de l'esdeveniment te associat un moment de pluja (Fig. A 2). L'episodi de pluja definit d'aquesta manera te inici i final al moment que es troba un període de més de 2 hores sense pluja.

En el cas que no es consideri que hi ha hagut un període de pluja durant l'esdeveniment [4] retorna el missatge de: "No hi ha hagut pluja durant l'esdeveniment":

[4] **get_event_info**(utc_dt, datetimered, rainfall, hours_norain=2)

Calcula el rang d'influència de l'esdeveniment (2h amb pluja per davant i per darrere) que determina si hi ha hagut pluja o no durant l'esdeveniment).

Retorna:

- (1) event_index: correspon al index de la posició del valor de precipitació dins de (rainfall).
- (2) event_begin_index: posició inicial dins la llista de pluges (rainfall).
- (3) event_end_index: posició final de la llista de pluges (rainfall).
- (4) event_rainfall_accum: pluja acumulada dins el rang de l'esdeveniment.

Si la pluja acumulada en el període de 4 hores (2 hores prèvies i 2 hores posteriors a la data de l'esdeveniment) es superior a 0, considerant que hi ha pluja durant l'esdeveniment, es calcularà, per les hores posteriors i anteriors a l'esdeveniment (sense contar la precipitació ja calculada en el moment de l'esdeveniment) si hi ha pluja de forma continuada, per tant, que hi segueixen havent valors superiors a 0 consecutius i aquests es sumaran a l'acumulació en el període de l'esdeveniment fins que quedin tallats per un conjunt de valors 0 superiors al valor llindar per considerar-se pluja continua (valors de 0 superiors a un rang de 2 hores), el qual indicarà, l'inici o final d'un període sense precipitació. Aquest càlcul es fa de forma iterativa tant per la pluja prèvia o posterior a l'esdeveniment, on [3] va saltant de mitja hora en mitja hora analitzant si per la posició on esta hi ha 2 hores de pluja prèvies o posteriors (Fig. A2a). La posició de la data de l'esdeveniment (event_index) i els límits inicial i final de l'esdeveniment (event_begin_index i event_end_index) defineixen la posició del període de l'esdeveniment (per tant, la durada de l'episodi de pluja que defineix l'esdeveniment) dins la llista de valors de precipitació (rainfall).

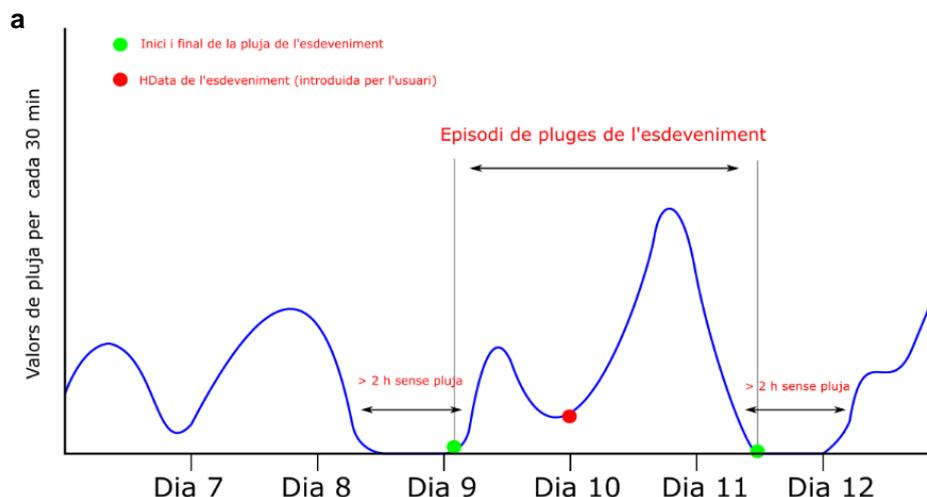
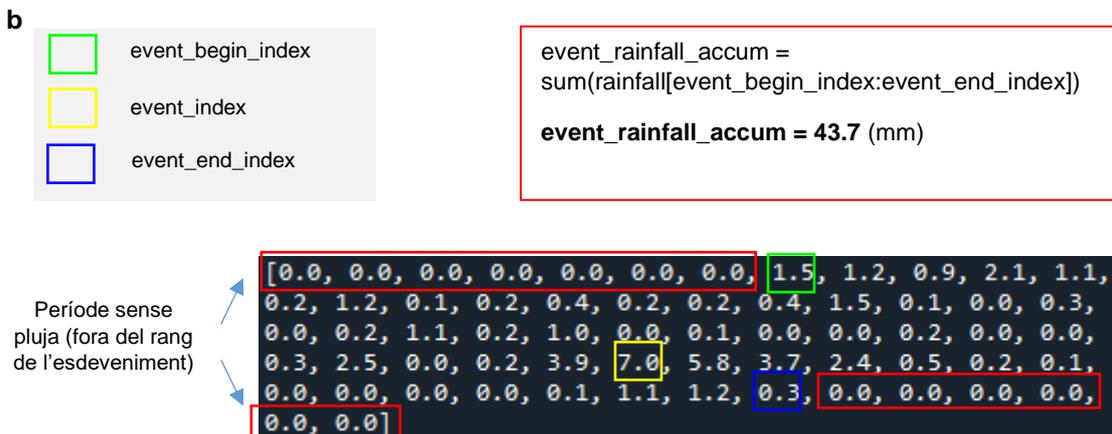


Figura A 2 Representació gràfica del concepte període de l'esdeveniment (que pot contenir o no un episodi de pluges) i el període total analitzat de dies. L'esdeveniment es caracteritza per la valoració a partir de la data d'anàlisi introduïda per l'usuari i la possibilitat que existeixi un episodi de pluja si es considera que prèviament hi ha un període de 2 hores sense pluja i dintre d'aquest interval la pluja es continua. b. Representació numèrica del període de l'esdeveniment i el possible episodi de pluges relacionat que el caracteritza. Elaboració pròpia.

Finalment, el conjunt de valors a retornar més importants és l'acumulació de pluja durant el període de l'esdeveniment (event_rainfall_accum). Aquesta pluja és calculada (Fig. A2b) a partir



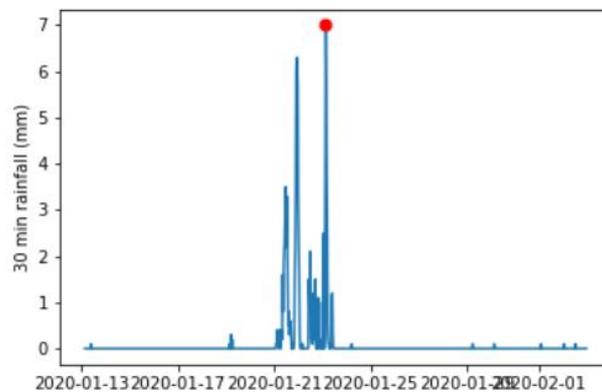
de la suma de tots els valors de pluja (mm) per a cada mitja hora dins el rang de l'esdeveniment (dins l'interval definit entre event_begin_index i event_end_index).

En el moment que [3] discrimina si durant el període de l'esdeveniment hi ha hagut pluja o no, crida a [5] que genera la representació gràfica corresponent a l'acumulació de pluja durant el període de 20 dies i l'acumulació puntual durant el període de l'esdeveniment:

[5] **create_10days_file**(event_name, event_index, datetimered, rainfall, out_10days_file)

Generació d'informe de 20 dies.

Creació de la figura de l'evolució de la pluja (acumulació/dia) i marquem amb un punt vermell el moment de l'esdeveniment.



2) Càlcul de la pluja acumulada en períodes previs a l'esdeveniment de 1, 3, i 7 dies

Si bé [5] determina si durant el moment de l'esdeveniment ha existit precipitació, el qual es significat en el context de l'anàlisi dels factors de desencadenament d'esllavissades o moviment de vessants i per atribuir realment la presència d'un episodi de pluja com a factor desencadenant, l'acumulació prèvia de pluja en un interval més llarg de temps és un altre dels factors importants. Finalitzada [5], [3] realitza la crida a [6] que calcula la pluja acumulada en un cert interval de temps (24 hores ,3 dies i 7 dies respectivament):

[6] **get_accumulated_rainfall**(event_begin_index, event_end_index, event_rainfall_accum, rainfall, rainfall_step_hours, hours)

Calcula la pluja acumulada en un cert interval de temps

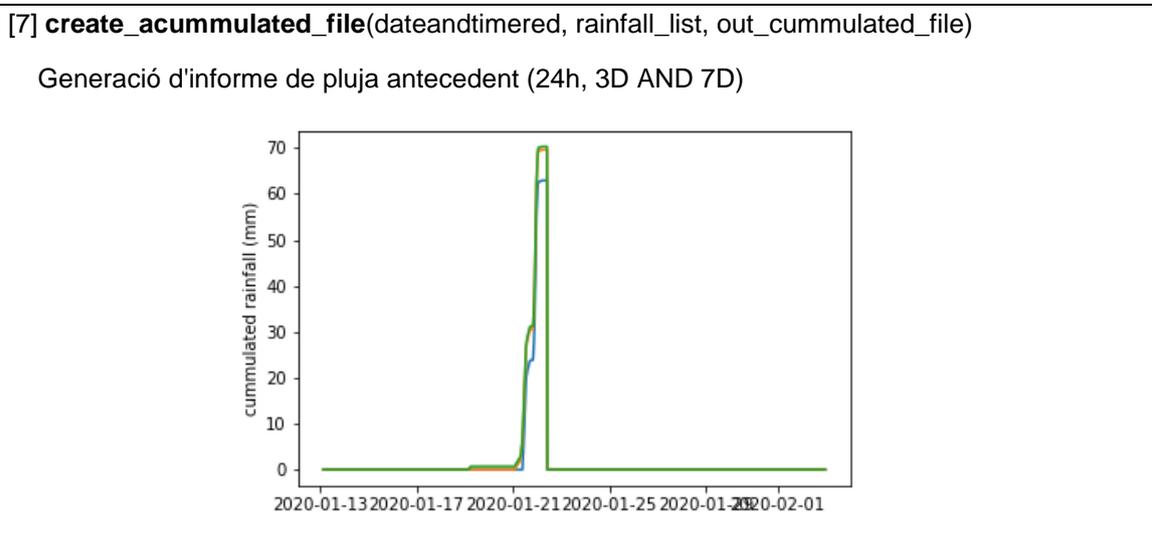
Retorna:

(1) rainfall_accum: una llista de pluja acumulada

Per tal d'aconseguir el càlcul per els 3 períodes establerts [3] fa una iteració per a cada un fent la crida a [6], on es considera que, tant si hi ha hagut pluja o no durant el període de l'esdeveniment, es suma la pluja per cada iteració depenent del període a analitzar (24 h, 3 dies i 7 dies) on es calculen, a partir de la llista de valors de pluja (rainfall) i segons la posició dels índex de l'esdeveniment (event_begin_index i event_end_index) tots aquells índexs de valors (posició dins la llista de valors de pluja) que apareixen entre el principi del període i l'esdeveniment (p.ex: seleccionar els valors de pluja dins la llista en l'interval de la posició de 7 dies abans de l'esdeveniment fins la data concreta de l'esdeveniment).

En el cas que no hi hagi suficient dades obtingudes els servidors de MeteoCat que pugui fer servir [6] per a calcular la pluja acumulada dins els períodes i intervals de temps establerts, aquesta retorna un missatge d'alerta.

En el moment que [6] ha calculat l'acumulació prèvia de pluja durant els 3 períodes, crida a [7] que genera la representació gràfica corresponent a l'acumulació de pluja durant els 3 períodes previs d'acumulació abans de l'esdeveniment:



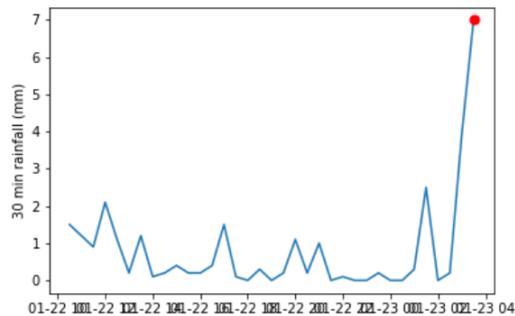
3) Acumulació de pluja durant l'esdeveniment

Un cop determinada la duració del període de l'esdeveniment (valors de pluja continus associats a les hores consecutives a l'hora central de l'esdeveniment), el que significa que hi ha un episodi de pluja durant l'esdeveniment, [3] calcula l'evolució de pluja dintre del interval definit per l'inici del període de l'esdeveniment (després de 2 hores sense pluja) i la data de l'esdeveniment (event_datetime i event_index) i s'obvien els valors post esdeveniment dins del període de l'esdeveniment.

La quantitat de valors de precipitació que caracteritza la durada de l'esdeveniment, en el cas que existeixi un episodi de precipitació, [3] crida a [8] i genera la representació gràfica del període de l'esdeveniment:

[8] `create_event_file`(event_index, dateandtimered, rainfall, event_dateandtimered, event_rainfall, out_event_file)

Creació d'informe de l'event



4) Intensitat màxima de la pluja 24 hores prèvies a l'esdeveniment

[3] genera un càlcul d'intensitat màxima de la pluja prèvia a l'esdeveniment a partir de la qual es determina si la intensitat de la pluja supera el valor del llindar d'intensitat regional de Guzzeti (Guzzeti, 2008). Aquest procés es dona quan es determina que hi ha precipitació durant el període de l'esdeveïment (event_begin_index, event_end_index, event_datetime i event_rainfall_accum). Per calcular la possibilitat que la intensitat de la pluja prèvia a l'esdeveniment pugui superar el llindar, [3] crida a [9] que calcula el conjunt de valors llindars per a un període de temps concret (el qual serà el mateix que la durada de l'esdeveniment):

[9] **get_duration_threshold**(duration_hours, threshold_climatic_area)

Calcula el threshold d'un període de temps (h)

Retorna:

(1) threshold: paràmetres i definició de l'equació que generara un valor segons la durada del període de pluja

$$\text{threshold} = 8.70 * ((\text{duration_hours}) ** (-0.66))$$

Un cop calculada l'equació que definirà el conjunt de valors llindars els quals determinaran si la intensitat d'una pluja per un esdeveniment concret pot ser categoritzada com a crítica i per tant amb potencial de desencadenar una esllavissada, [3] procedeix a calcular els valors d'intensitat prèvies, cridant a [10], a l'esdeveniment. El conjunt de valors d'intensitat provenen de la iteració de cada un dels valors de la llista de pluja durant l'esdeveniment (event_rainfall) on es calcula per a cada mesura de pluja (valor per a cada mitja hora) el coeficient d'aquest mes el valor que el precedeix entre l'interval de mesura (cada mitja hora). D'aquesta manera [10] retorna a [3] una llista d'igual llargada que la de pluja prèvia a l'esdeveniment amb valors de mesura segons els períodes de 0,5 h, 1 h, 3h, 5h i 24 h. La intensitat màxima es representarà a partir del valor d'intensitat més alt de la llista que retorna [10]:

[10] **get_max_intensity_rainfall**(event_rainfall, rainfall_step_hours, duration_hours)

Calcula la intensitat màxima de pluja acumulada en un període de temps (h)

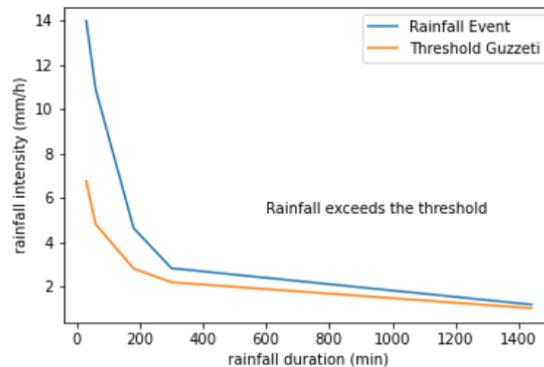
Retorna:

(1) max_intensiy: valor més alt de la llista d'intensitats corresponents al període de l'esdeveniment.

Un cop calculats els valors d'intensitat previs a l'esdeveniment i els valors llindars per el mateix període, en funció de l'àmbit climàtic on s'hagi donat l'esdeveniment (Mediterràni temperat, Mediterràni càlid i Montanya), [3] compara els valors per a cada punt de precipitació i informa si els valors d'intensitat de la pluja superen als del llindar. Un cop feta la comparació, [3] crida a [11] per generar la representació gràfica avisant de si el llindar és superat:

[11] **create_intensity_file**(duration_hours, imaxs, post_label, post_imaxs, threshold, rainfall_exceeds_threshold, rainfall_evolution, out_intensity_file)

Creació d'informe de duració de pluja



5) Intensitat de la pluja post esdeveniment en períodes de 1 dia, 3 dies i 7 dies

A partir del càlcul que [3] fa per la intensitat de l'esdeveniment i les hores prèvies, les quals representen les més crítiques, [3] genera el càlcul d'intensitats posteriors a l'esdeveniment. A partir de la crida a [10] reiterativament per períodes de 24 hores, 3 dies i 7 dies post esdeveniment. [3] pot calcular les intensitats posteriors (havent-hi pluja o no) a l'esdeveniment i d'aquesta manera contrastar-la amb els valors obtinguts anteriorment. En el cas que els valors d'intensitat post esdeveniment siguin més elevats que els del propi esdeveniment i les hores anteriors, [3] genera un missatge d'alerta que enviarà a [11] que representa gràficament els 3 períodes d'intensitats els quals inclouen la comparació de l'evolució de les intensitats per a cada període i la intensitat de l'esdeveniment. Aquest missatge s'interpreta com a una alerta de perill, en el cas que les intensitats segueixin augmentant post-esdeveniment:

[11] **create_intensity_file**(duration_hours, imaxs, post_label, post_imaxs, threshold, rainfall_exceeds_threshold, rainfall_evolution, out_intensity_file)

Creació d'informe de duració de pluja

