Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

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

Sarqui Altamirano, Paula Fiorella; Serra-Sagristà, Joan, dir. Dades obertes, bona ciència i programari lliure. 2021. (958 Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/248460

under the terms of the BY-NC-ND license

Dades obertes, bona ciència i programari lliure

Paula Fiorella Sarqui Altamirano

Resum– En la societat en la que vivim és de gran importància, i cada vegada més, que la informació derivada de les investigacions es comparteixi lliurement, però no tothom té l'avantatge d'accedir a totes les eines disponibles de forma gratuïta a l'Internet per aquest fi. En molts casos, és necessari tenir un coneixement previ de programació a l'hora de poder-ne fer us. Amb aquest projecte es busca facilitar l'ús d'una eina en desenvolupament als usuaris que o bé no tenen coneixements de programació o entenen les bases.

Paraules clau– dades obertes, Experiment Notebook, enb, Python, plantilla, Jinja, projecte, gestió

Abstract– In today's society, it is of great and increasing importance that knowledge deriving from scientific research is shared openly. Despite this, not everyone has the advantage of being able to use every tool in the field that is freely available on the Internet for this end, as, in many cases, a prior knowledge of programming is required. This project aims to improve the ease of use of a tool currently in development so users with limited or no prior knowledge of programming can better benefit from it.

Keywords– open data, Experiment Notebook, enb, Python, template, Jinja, project, project management

1 INTRODUCCIÓ

LES dades porten a l'enteniment i l'enteniment porta a l'avenç. En el camp de la ciència, el disposar del suficient volum d'informació com per extreure conclusions, fa que els resultats siguin més fiables i significatius. Però, què passa si no disposem de dades suficients? És aquí on entra en joc el concepte d'*Open Data*.

Què és *Open Data*? Segons la Fundació de Coneixement Obert (*Open Knowledge Fundation*[1]), per definició, "les dades i continguts oberts poden ser utilitzats de forma lliure, modificats i redistribuïts sense restriccions"[2]. Si ens endinsem més al concepte, però, podem trobar una llista de requisits a complir perquè una llicència sobre un contingut sigui considerada lliure[3].

A grans trets, segons l'*Open Data Handbook*[4], podem dir que els trets principals per definir el concepte d'*Open Data* es resumeixen en els següents punts:

- **Disponibilitat i accès**: les dades han d'estar disponibles en un format adient, de forma completa, preferiblement a *Internet* i han de poder ser modificables.
- **Re-usabilitat i redistribució**: les dades han de ser publicades d'una forma que permeti la seva reutilització així com la seva re-publicació.
- **Participació Universal**: tothom ha de ser capaç d'accedir a les dades, reutilitzar-les i re-distribuir-les sense discriminació de cap mena cap a la persona que ho faci.

Ara bé, sovint el procés d'explotació de les dades implica una sèrie de passos els quals no tothom està preparat per portar a terme. La programació i/o modificació d'algorismes no és fàcil per qui no ha programat abans.

En aquest context, precisament, es centra el nostre projecte.

Experiment Notebook és una llibreria, desenvolupada en *Python* que té com un dels seus objectius facilitar el processament de dades de caire principalment numèric. Actualment, per poder fer ús d'aquesta llibreria, s'ha d'integrar mitjançant l'elaboració d'un script de *Python*.

Aquest treball té dos propòsits principals. Per una banda, tenim l'elaboració, de principi a fi, d'un projecte, des de la

• E-mail de contacte: pfsarqui@gmail.com
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: nom i cognoms del tutor (departament)
• Curs 2020/21

seva planificació a la seva implementació i posterior integració. Per altra banda, tenim la contribució a un repositori de programari lliure per tal de fer que la utilització de la llibreria sigui més senzilla pels usuaris que, o bé puguin tenir pocs coneixements de programació, o bé no en tinguin cap.

En aquest article presentem la motivació que ha fet que en un primer moment es plantegés aquest projecte. Seguidament, fem una presentació de l'estat de l'art. A continuació, ampliem l'explicació dels objectius del treball, fem menció a la metodologia utilitzada en el desenvolupament i a la planificació plantejada. expliquem les diferents parts en les que s'ha dividit el desenvolupament del projecte i exposem els resultats, així com un anàlisi breu d'aquests mateixos. Finalment, presentem les conclusions extretes.

2 MOTIVACIÓ

Per definir la motivació que ha portat a la realització d'aquest projecte, cal abans tindre clar el concepte d'*Open Data*, definit a l'apartat anterior. Una vegada entenem el que significa, podem començar a entendre el concepte d'*Open Science*. Segons la **UNESCO**[5], encoratjar la ciència oberta per centrar-se a les necessitats de la societat actual, incentivant la comunicació entre centres d'investigació i cercant la igualtat d'oportunitats, té el poder d'accelerar l'avenç de la ciència, tal i com ha sigut demostrat durant el passat any 2020 amb la pandèmia de la COVID-19 i la creació de les vacunes. El treball constant cap endavant, basant-nos en l'experiència de les dades recollides, permet que al present s'estableixin les bases del futur. Però, què passa quan no tenim prou informació de casos passats? Què passa si un equip d'investigació té la capacitat de processament de dades però no l'oportunitat de la recol·lecció de mostres suficients sobre un cas de prova concret?

Com ja hem comentat a la introducció, l'avenç de la ciència i la tecnologia implica la necessitat i/o producció de grans volums de dades. Sovint, les investigacions en l'àmbit científic-tecnològic, o bé precisen de grans volums de dades, o bé en generen. Per tal que aquestes dades puguin ser representades de forma llegible per l'ésser humà s'han de processar. Aquestes transformacions poden significar càlculs, i sovint es necessiten representacions gràfiques per tal d'observar l'espectre complert de la informació.

Com exemple més recent tenim l'article *Expert Insights on the Impacts of, and Potential for, Agricultural Big Data*[6] de 2021, que analitza els beneficis potencials de la compartició de dades en l'àmbit de l'agricultura.

Amb la compartició de dades podem crear una font de coneixement comú amb la qual impulsar la investigació de noves tecnologies, nous coneixements a la medicina, la biologia, la microbiologia, etc. Per exemple, en el cas del *software*, aquest concepte ha sigut fonamental per la creació d'eines com **LibreOffice**, **VLC Media Player** i **Linux**.

Donat el context de la necessitat de compartició de dades, aquest projecte ha intentat cercar una forma de facilitar l'ús de la llibreria **Experiment-Notebook** per aquells usuaris menys experimentats en l'àmbit de la programació.

3 ESTAT DE L'ART

Actualment, ja hi ha projectes en desenvolupament a nivell europeu per la compartició de dades de caire científic per tal d'incentivar la col·laboració entre institucions.

"El projecte *European Open Science Cloud* (EOSC) es va iniciar al 2015 amb l'objectiu principal de construir un entorn fiable, virtual i federat que traspassi fronteres, tant polítiques com institucionals per guardar, compartir, processar i reutilitzar informació digital." [7]

Aquesta iniciativa segueix els cada vegada més estesos principis FAIR. Aquests principis estableixen una guia a seguir per tal d'incrementar l'adequació de les dades al processament per ordinador, ja que cada vegada més depenem de la capacitat de còmput d'aquests per tal de tractar les dades.

Les sigles FAIR (*Findablility, Accessibility, Interoperability, and Reusability*[8]) indiquen que la informació publicada ha de ser fàcil de trobar tant per ordinadors com per persones. Ha de comptar també amb metadades llegibles per les màquines per facilitar el descobriment automàtic de conjunts de dades. Ha de ser clar la manera amb la que accedir a les dades. Ha de poder ser emprada per interactuar amb altres dades i/o aplicacions. Per últim, ha de poder ser reutilitzable.

Basat en el projecte abans esmentat, també tenim el projecte ExPaNDS que es centra en fer accessible la informació generada per les fonts de fotons i neutrons d'Europa.

Pel que fa al projecte 'Experiment Notebook', actualment compta amb funcions que permeten la realització d'experiments de compressió d'imatges amb diferents paràmetres. També aporta funcionalitats que permeten la realització de gràfics donat un conjunt de dades o *dataset*. Per major referència podem consultar la pàgina web oficial de la documentació de la llebreria [9]

4 OBJECTIUS

Aquest projecte s'ha basat en la col·laboració a una llibreria de codi obert, anomenada *Experiment Notebook*[10], en endavant *enb*, que compta com a desenvolupador principal amb el Miguel Hernández-Cabronero, encara que actualment disposa de diversos contribuïdors actius.

La llibreria en qüestió està desenvolupada en *Python* i té com funcions l'anàlisi de dades per tal de crear de forma senzilla gràfics i la compressió d'imatges amb i sense pèrdues. És fàcil d'instal·lar per línia de comandes i disposa d'una documentació en anglès útil per aprendre a utilitzar-la [9].

Actualment, el projecte del que s'ha partit no disposa d'una línia de comandes, sinó que les funcionalitats s'han d'integrar a un *script* de creació pròpia. Aquest treball pretenia dotar a la llibreria de funcionalitats que evitessin o minimitzessin la necessitat d'haver de programar algorismes a mà per tal d'utilitzar la llibreria.

A més, s'ha cercat en tot moment el beneficiar-se de l'experiència que comporta la realització d'un projecte com aquest.

Així doncs, els objectius principals queden definits en els següents punts:

- Exercitar les etapes de planificació i desenvolupament d'un projecte per tal de beneficiar-se de l'experiència.
- Contribuir a un projecte de codi obert que cerca poder facilitar el processament de les dades en format digital al camp de la investigació en l'àmbit científic-tecnològic.

Als següents sub-apartats procedim a explicar amb més profunditat l'espectre dels objectius.

4.1 Procés formal de desenvolupament

Un dels objectius principals del treball ha sigut el de poder fer l'exercici de la realització de les etapes de planificació i desenvolupament d'un projecte. Per portar a terme un procés formal de desenvolupament s'ha dividit el projecte en dues fases principals:

- Fase de planificació.
- Fase de desenvolupament.

A la primera fase, s'ha establert un pla d'acció a portar a terme que consistia en l'elucidació de requisits i la definició de tasques. Per l'elucidació de requisits s'ha portat a terme una entrevista.

Pel que fa al desenvolupament, s'ha dividit en tres subfases: implementació, realització de proves i documentació.

Aquesta experiència, tot i que no ha comportat tants beneficis com en un primer moment s'esperava, ha aportat un enriquiment personal al que avaluar la magnitud, dificultat, duració i viabilitat d'un projecte es refereix.

Més endavant, a l'apartat de resultats i anàlisi dels mateixos, podrem veure l'estat del projecte.

4.2 Contribució a la comunitat

El segon objectiu principal del treball era el de poder contribuir a la comunitat científica-tecnològica amb una eina oberta, d'utilitat i totalment gratuïta amb la qual aportar a l'avenç del camp. Amb aquesta fita en ment, s'ha decidit optar per facilitar la utilització de la llibreria preexistent *enb*. Per fer-ho, s'havien establert els objectius d'elaborar una línia de comandes que permetés la creació, modificació i utilització de plantilles, inclusió de *plugins* i execució d'anàlisis. Addicionalment, s'havia plantejat la fita d'addició de tipus de gràfics diferents. Aquesta última meta s'havia de desenvolupar en funció dels resultats d'una enquesta a la comunitat científica per tal d'extreure els tipus de gràfics més utilitzats.

4.3 Per futures iteracions de la llibreria *enb*

En qüestió a futures iteracions del projecte, es suggereix que s'incorpori una interfície gràfica per tal de fer-la una eina més intuïtiva i *user-friendly*. Això faria que un major nombre d'usuaris puguin beneficiar-se de les seves característiques.

5 Metodologia i Planificació

Al desenvolupament d'un projecte podem distingir dues fases principals, la de planificació i la de desenvolupament d'un projecte. Pel que a aquest projecte respecta, inicialment, es va fer l'estimació de que la primera fase comprendria les següents activitats:

- Estudi de projectes similars.
- Elucidació de requisits.
- Definició de tasques.

Tot i que en primera instància era una fulla de ruta clara, en dependre l'elucidació de requisits de l'enquesta a realitzar, es va veure que, en la pràctica, aquesta activitat hauria de conviure en paral·lel amb la fase de desenvolupament, motiu pel qual la planificació es va haver d'adaptar.

A la fase de desenvolupament, doncs, havíem d'englobar la implementació del codi del projecte, l'elucidació de requisits, la realització de proves i la documentació.

Pel que fa a la metodologia seguida, no ha variat en gran mesura respecte a l'exposada a informes anteriors. Tal i com vam fer a l'anterior informe, a continuació dividirem les metodologies segons les àrees de funció.

5.0.1 Gestió

Per dur a terme la planificació del projecte s'ha utilitat la metodologia "Kanban"[11], juntament amb la plataforma "Jira"[12] com eina de gestió. Tot i que a la fase corresponent a l'informe de progrés I es va establir unes tasques a realitzar, al final d'aquella mateixa fase es va veure que la complexitat real del projecte era major que l'estimada prèviament i, per tant, calia redefinir les tasques. A la figura 11 podem observar el diagrama de Gantt resultant de les modificacions que ha calgut realitzar a les tasques.

5.1 Implementació

Al que a la implementació respecta, l'únic canvi ha estat que, en comptes de pujar els canvis efectuats al repositori de GitHub del projecte[10], s'han registrat en un repositori local i, posteriorment, s'han pujat també a un *fork* del projecte original[**?**], des d'on es podran reintegrar al repositori del projecte un cop acabi el desenvolupament si així es desitgés. Pel que fa a l'IDE emprada, segueix sent Pycharm[**?**] conjuntament amb la consola bash nativa d'Ubuntu, que ha ajudat a sol·lucionar inconvenients en la gestió d'entorns virtuals a l'IDE.

6 Desenvolupament del projecte

6.1 Entrevista

En començar el projecte es va realitzar una entrevista a l'Alberto Zurita Carpio, personal d'investigació en formació. L'Alberto és un home de trenta anys, format a la UAB en ciències biomèdiques i doctorand des de 2016. Encara que és un usuari avançat de Windows, pel que respecta a la programació, té alguns conceptes superficials però no programa.

Degut a aspectes de confidencialitat relacionats amb la seva feina, l'Alberto ens ha demanat que es censurés una de les imatges aportades, raó per la que la figura 1 es trobarà amb etiquetes en lloc de noms.

L'entrevista va ser breu i es va realitzar per missatges de text que posteriorment es van comentar de forma presencial en una reunió informal.

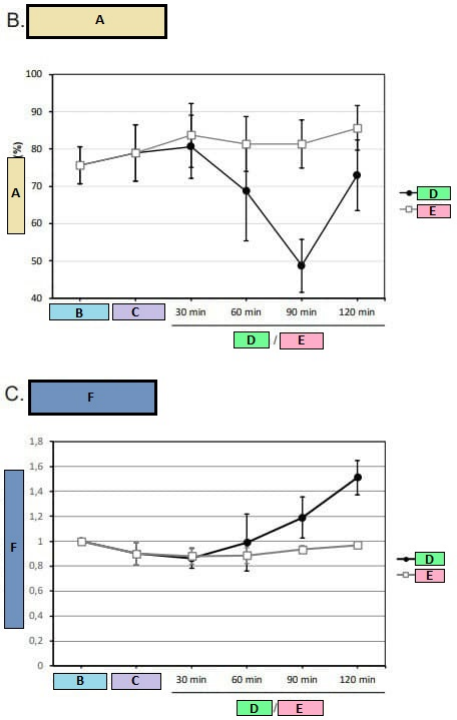

Fig. 1: Exemple de gràfiques més utilitzades.

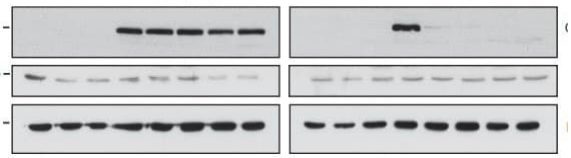

Fig. 2: Exemple d'experiment.

Paula Com tractes les dades que reculles? Quines eines utilitzes, alguna vegada has programat per processar les dades?

Alberto El segon és fàcil de respondre: no; però sé del tema a grans trets. Sobre el primer sí que depèn molt del tipus de dada que estem parlant, et puc posar exemples del que faig.

Paula Els exemples em venen genial. De moment, què tipus de representacions gràfiques utilitzes més?

Alberto Depèn, al meu camp sempre se sol tractar i comparacions. Presència/absència de 'X' marca/*spot*/proteïna, aquest tipus de coses, acompanyat de taules descriptives i gràfiques amb estadístiques bàsiques. En essència per demostrar que el que estàs posant a la figura es representatiu de la població d'estudi. Aquí el tipus de gràfica depèn del tipus de dada, si és recompte, histogrames, si va per percentatges poden ser línies, corbes, etc.

*(L'**Alberto** adjunta la figura 1)*

Alberto Les figures s'acompanyen de l'experiment en sí mateix, clar, això ve de la mateixa figura, per exemple.

*(L'**Alberto** adjunta la figura 2)*

Paula Què utilitzes per fer aquestes gràfiques?

Alberto Depèn, per estadística jo utilitzo Excel i/o Graphpad Prism, segons si tan sols haig de fer la figura o si haig de fer algun estadístic decent.

Paula Canviaries d'aquestes eines alguna cosa que creguis que et faria estalviar tems?

Alberto No especialment, ja tinc les meves *macros* muntades per moltes casuístiques.

6.1.1 Conclusions

En una primera instancia, tenint en compte el context de l'entrevistat, podem veure que és poc probable que una persona amb el seu perfil, o similar, utilitzi la llibreria en el seu estat de desenvolupament actual, ja que no disposa d'una eina gràfica.

6.2 Enquesta

A l'enquesta realitzada es van fer un total de vuit preguntes i es van rebre un total de set respostes.

A la pregunta "Quin és el teu camp d'investigació?" (*"What's your investigation field?"*), com podem veure a la figura 3 tenim que el 28.6% investiguen al camp de l'intel·ligència artificial, un 14.3% en el camp de la física i l'altre 57.1% al camp de l'enginyera.

A la pregunta "Quina es la teva categoria professional" (*"What's your professional cateory?"*), segons la figura 4, un 28.6% de la gent són professors agregats, en anglés *professor associate*, que compta amb un 14.3% més, pel que tenim que el 42.9% són professors agregats. Un 28.6% dels que han respost són estudiants de doctorat, un 14.3% es compta entre els investigadors tècnics i l'altre 14.3% es correspon a estudiants de grau.

A la tercera pregunta, que podem trobar a la figura 5, "En termes de temps, quant diries que trigues a processar les dades d'un experiment?" (*"In time terms, how much would you say that you spend in processing the outcome data of the experiments?"*), podem veure que la majoria es debat entre el temps just i massa.

Amb la quarta pregunta, figura 6, podem veure que una mica més de la meitat de les persones utilitzen eines de línia de comandes per processar les seves dades.

La cinquena pregunta, present a la figura 7, ens diu que tots han hagut de programar algun cop les eines amb les que processen les dades.

A la sisena pregunta, figura 8, podem observar que els problemes amb els que es troben son de diversa índole. Dues de les respostes, "Portar a terme múltiples experiments" (*"Perform multiple experiments"*) i "A lo millor organitzar les dades en tables per ser processades" (*"Maybe orgaanizing the data into tables to be processed"*) destaquen en relació a la llibreria *Experiment Notebook*, ja que a priori si que podrien ser resoltes per la llibreria.

A la setena pregunta, corresponent amb la figura 9, podem veure que la majoria han programat en algun llenguatge de programació semblant a *Python*.

Finalment, a la última pregunta, figura 10, podem veure les respostes varien, tot i que les dues amb més seleccions són les de diagrames de dispersió (*"Scatter plots"*) i diagrames de capses (*"Box plots"*).

6.2.1 Conclusions

Donat que la mostra a la que es va fer l'enquesta és reduïda, no podem concloure que els resultats ens donin un espectre clar del panorama actual per l'àrea d'estudi del treball. Si comparem els resultats de l'enquesta amb el resultat de l'entrevista, sense entrar en gaire detall, podem veure que els perfils provenen d'àmbits d'investigació diferents, sent la majoria d'ells del camp científic tecnològic.

6.3 Addicions

En aquest apartat presentarem les modificacions produïdes al codi. En un principi, per tal d'afegir un intèrpret de comandes que, donat un conjunt d'instruccions, generi, modifiqui o executi una plantilla d'un experiment, vam dividir el problema en dues parts:

- **Parsers (intèrprets):** com a terme general, un *parser* és una eina que ens permet la "traducció" d'una comanda en una acció desitjada. En el nostre cas concret, el que es buscava era que el *parser* de la comanda introduïda generés un objecte JSON que servís per, posteriorment, generar un *script* de codi executable.
- **Generació d'*scripts* o arxius de codi:** algorisme que té com a objectiu, donat un objecte JSON, crear un arxiu de codi executable que permetés l'execució d'un experiment.

Pel que respecta a la primera part, si ens fixem en la classe *Singleton_cli*[13], de la versió funcional actual del projecte *Experimental Notebook*[10], ja disposa d'una estructura que permet la creació d'arguments no posicionals de forma única a tot el projecte, això només existeixi una instància de la classe. Dit això, hi ha diversos factors a tindre en compte. Primerament hem hagut de decidir quines accions havíem de considerar a l'hora d'afegir o modificar una plantilla o *template*.

Inicialment, es va considerar que les operacions bàsiques a portar a terme únicament en la modificació de plantilles, ja que considerem la creació una operació prèvia a qualsevol modificació, encara que es realitzes a la mateixa comanda. Les operacions a realitzar a qualsevol modificació serien les següents:

- **Afegir**: dins d'aquesta operació es va considerar que el que es podia afegir a una *template* era una de tres possibles opcions:
  - ***Experiment***: al context del projecte *enb* i al d'aquest projecte, per raons pràctiques, considerem experiment a tota classe que hereti de la classe *Experiment* present a l'*script*[14]. Més endavant explicarem la raó que ha portat a decantar-se per aquest criteri.
  - ***Anàlisi***: com al cas de l'opció anterior, per raons de simplificació, considerem un anàlisi a tota classe que hereti de la classe *Analyzer*, present a l'*script*[15].
  - ***Step***: aquesta opció va ser inclosa com a concepte inicialment a aquesta part del desenvolupament amb la idea de que l'usuari pogués decidir quan voldria que s'executés què a l'*script* posteriorment generat, podent escollir tant l'arxiu d'entrada com de sortida, ja sigui d'un anàlisi o experiment. Però aquesta noció es va descartar degut a la complexitat programàtica que aportava respecte el poc o nul benefici que en podria treure l'usuari, ja que quantes és comandes hagués d'executar, més complexitat tindria la utilització de la línia de comandes.

  Més endavant també es va considerar afegir l'opció d'afegir *plugins* o extensions, considerant-se *plugin* a tota classe que hereti de la classe *AbstractCodec*[16]. Però, com va passar amb l'opció d'*step*, també es va descartar. Als apartats següents explicarem el perquè.
- **Eliminar**: també aquí haguéssim tingut tres opcions:
  - ***Experiment***.
  - ***Anàlisi***.
  - ***Step***.

Pel que fa a la **generació d'*scripts***, s'ha partit des de la suposició de l'existència d'un objecte *JSON* i, amb l'ajuda de la llibreria **Jinja2**, s'ha creat un *template* per la generació d'*scripts*. En aquest mateix apartat, aquesta part del projecte disposa d'un sub-apartat que explica el procés en més profunditat.

Encara que no s'ha pogut arribar a assolir tots els objectius marcats en un principi, sí que s'ha pogut elaborar una base per tal de poder, en un futur, incorporar una línia de comandes funcional. A continuació entrem en més detall sobre la implementació.

6.3.1 Parsers

Per parlar d'aquesta part del desenvolupament, primer hem de parlar del mòdul de *Python*, **argparse**. Què és *argparse*? Aquest mòdul permet que, de forma senzilla, es puguin escriure línies de comandes *user-friendly*. Concretament, permet la creació d'intèrprets i sub-intèrprets de comandes, així com l'addició d'arguments definits als paràmetres per tal que s'adeqüin al que necessitem.

De tipus d'arguments bàsics, n'hi ha dos:

- **Posicionals**: arguments en base al quals l'intèrpret actua depenent de la posició en la que es trobin a la comanda.
- **Opcionals**: els arguments que més comunament podem trobar a qualsevol intèrpret de comandes de *Linux* i que ens permeten definir una opció concreta a qualsevol punt d'una comanda tenint en compte que cada opció que utilitzem ha d'estar dins del marge d'operabilitat del seu *parser*.

Una vegada definits els tipus bàsic d'arguments, hem planificat una estructura inicial per la creació dels arguments que ha anat canviat conforme avançàvem al projecte.

Inicialment, l'estructura a aconseguir va ser la següent:

- **enb**: *parser* ja present al codi, que actualment admet una sèrie d'opcions definides a l'*script* de configuració *"config"*[17].
  - ***−−template/−t***: argument opcional que a continuació esperava un *string* o cadena de caràcters que seria el nom de la *template* a crear o modificar.
  - ***−−working_dir/−w***: argument opcional que, donat un directori existent, el guardaria. En el cas d'afegir una *template* nova, seria al directori al que crearíem una carpeta dins de la qual es trobaria la plantilla a crear. En el cas d'una modificació de plantilla, seria al directori al que es buscaria la plantilla de la qual hauríem d'haver proveït el nom.
  - ***−−add/−a***: argument opcional que esperava una de tres cadenes de caràcters:
    - "experimet"
    - "analysis"
    - "step"
  - ***−−experiment_type/−x***: argument opcional que esperava una cadena de caràcters a comparar amb la llista d'experiments disponibles. una vegada validat, hagués fixat el tipus d'experiment a afegir.
  - ***−−input/−i***: argument opcional que, com l'anterior, esperava una cadena de caràcters que compararia amb la llista d'experiments disponibles.
  - ***−−erase/−e***: argument opcional per defecte a *false* que, de ser present a l comanda, es guarda a *true*. Aquest argument, de ser utilitzat a la comanda, s'encarregaria d'esborrar la *template* indicada.
  - ***−−create_new_template/−c***: argument opcional per defecte a *false* que, de ser present a la comanda, es guarda a *true*. En utilitzar-se aquest argument, estaríem indicant que volem crear la plantilla de la qual indiquem el nom i el directori.
  - ***−−list_experiments***: argument opcional per defecte a *false* que, de ser utilitzat es guardaria com *true* i executaria una funció que llistaria els experiments disponibles.
  - ***−−list_analysis***: argument opcional per defecte a *false* que, de ser utilitzat es guardaria com *true* i executaria una funció que llistaria els anàlisis disponibles.

Cal dir que aquesta és una estructura bàsica, ja que, a la pràctica, s'ha de tenir en compte els paràmetres necessaris per cada experiment i anàlisis.

Aquesta estructura pot crear-se amb el codi ja existent, que permet la creació d'arguments opcionals amb la definició de mètodes abstractes a l'arxiu *config*, sent el nom del mètode el nom estès de la funció i la primera cadena de caràcters l'abreviació, deixant els paràmetres consecutius als paràmetres de configuració del propi argument, propis d'*argparse*. Després d'haver creat aquesta estructura, va arribar el moment de "parsejar" els paràmetres.

Problema: resulta una estructura confosa de processar amb codi. Tal i com s'extreuen els paràmetres a la versió original d'*enb*, es retorna un diccionari amb tots els valors per defecte o donats als paràmetres esmentats a la comanda. Si es vol seleccionar només un grup de paràmetres, s'ha de controlar la seva presència o bé afegint sempre valors per defecte, ja que sinó no són afegits al diccionari, o bé vigilant que existeixen al mateix. Vam provar de descartar les operacions que hauríem de processar amb la cadena de caràcters que composa la comanda en sí mateixa però es va descartar ja que comportava altres problemes no previstos, com per exemple, que un *template* es digués "experiment", fet que ocasionaria un error ja que "experiment" es troba dins de les opcions de "add".

Solució: després de trobar-nos amb aquest problema, es va considerar que la solució seria afegir *subparsers* respecte al *parser* ja existent *enb* ja que es va creure que, al penjar del *parser* principal, sería més senzill extreure la informació dels paràmetres de forma organitzada. Com veurem en breus instants, això no va ser així.

Després de descartar l'estructura anterior com solució a aquesta part del projecte, vam definir l'estructura següent:

- **enb**: *parser* ja present al codi, que actualment admet una sèrie d'opcions definides a l'*script* de configuració *"config"*[17].
  - **template**: *subparser* d'*enb* que gestionava les comandes pertinents a la creació i modificació de *templates*.
    - ***template_name***: argument posicional, és a dir, obligatori, que tot just després de "template" esperava un *string* o cadena de caràcters que seria el nom de la *template* a crear o modificar.
    - ***−−working_dir/−w***: argument opcional que, donat un directori existent, el guardaria. En el cas d'afegir una *template* nova, seria al directori al que crearíem una carpeta dins de la qual es trobaria la plantilla a crear. En el cas d'una modificació de plantilla, seria al directori al que es buscaria la plantilla de la qual hauríem d'haver proveït el nom.
    - ***−−create_new_template/−c***: argument opcional per defecte a *false* que, de ser present a la comanda, es guarda a *true*. D'utilitzar-se aquest argument, estaríem indicant que volem crear la plantilla de la qual indiquem el nom i el directori.
    - **add**: *subparser* del *parser template* que gestionava les comandes relatives a l'addició d'elements a una plantilla.
      - **experiment**: *subparser* d'*add* que hauria gestionat els paràmetres relatius als experiments. Addicionalment, a aquest **parser** li havia estat assignat els paràmetres opcionals ***−−experiment_type/−x***, ***−−alias/−a*** (nom que es donaria a la variable encarregada de guardar l'experiment executat) i ***−−list_experiments−l*** que,

com el nom indica, s'encarregaria de llistar els experiments.

- **analysis**: d'igual manera que al *subparser "experiment"*, aquest *parser* comptava amb les mateixes aplicades a l'àmbit d'operacions del anàlisis.
- **step**: en aquest moment del desenvolupament, encara que es va partir amb la idea clara de que aquesta opció comptaria amb un nom d'*input*, un nom *output*, i algun mecanisme d'identificació que permetés crear una seqüència d'*steps*, es va deixar de veure la utilitat i no es va continuar projectant.

* **erase**: *subparser* del *parser template* que s'encarregaria d'esborrar una plantilla. Més tard es va decidir que aquest *parser* no feia falta i es va prescindir d'ell.

Problema: per falta de comprensió del codi del que es disposava, en un principi es van duplicar i triplicar els mètodes que permetien l'addició de paràmetres al *enb*, modificant-los perquè, de forma senzilla, poguéssim afegir paràmetres als nostres *subparsers*. Però aquesta solució no era ni molt menys elegant i, no vam trigar en adonar-nos de que no era fàcilment escalable. Tampoc el manteniment era assequible.

Solució: es va crear un sol mètode a la classe ***Singleton_cli*** dit ***parsers_builder***, apart del mètode ***property***, que contemplés la necessitat de crear *parsers* que pengin d'altres així com l'addició de paràmetres a aquests mateixos intèrprets de comandes.

Problema: això no va resoldre el problema comentat anteriorment. La informació seguia sent complicada de recollir i seguia depenent de l'anàlisi de la cadena de caràcters que conforma la comanda.

Solució: procurant no canviar la forma original en la que es recullen els valors dels paràmetres, es va modificar de forma que els paràmetres quedessin inserits al diccionari d'opcions en diccionaris niuats segons el *parser* del que formessin part. S'ha de mencionar que, en el moment de la redacció d'aquest informe, aquesta versió no es troba pujada al repositori.

Problema: per causes que encara no es coneixen, l'ordre en el que es declaren els paràmetres en funció del seu *parser* i dels altres paràmetres del mateix importa, fet que, en el moment en el que es consulta l'ajuda de cada *parser* no és aparent però si ho resulta quan es pretén executar una comanda.

A banda, en el moment d'implementar les funcions que elaborarien el llistat d'experiments i anàlisis de forma dinàmica, es va decidir fer servir el mòdul de *Python "inspect"*.

Problema: en el moment en el que es carrega la classe *Singleton_cli* al projecte, tots els altres mòduls no estan encara carregats, pel que no es poden inspeccionar. De fer-se un *local*, s'incorre en un *import* circular, fent impossible que aquesta operació s'executi des d'aquesta classe.

Solució: es va provar a externalitzar aquesta operació. Des d'una classe externa a *Singleton_cli*, importada des de l'*script* "__main__" funciona. Actualment la classe que s'encarrega del processament de les opcions de les comandes està en fase de desenvolupament degut a que els problemes trobats han consumit molt temps i es va decidir deixar a una banda aquesta part del projecte i centrar−se a la part de generació d'*scripts*.

6.3.2 Generació d'*scripts*

Aquesta part del projecte, ha estat desenvolupada amb el mòdul **Jinja**, que es tracta d'una eina de creació de plantilles de text amb una sintaxis semblant a la que utilitza *django* per la creació de *templates*. Encara que en un principi es va plantejar com un mòdul part d'*enb*, el funcionament propi de *Jinja* va complicar aquesta tasca, pel que al final es va crear un mòdul extern dit ***template_genertor***[18].

El *template* que s'ha utilitzat per la generació d'*scripts* es troba a l'enllaç[19]. Pel seu format característic hem decidit no adjuntar-la com apèndix a aquest article, ja que, de voler utilitzar-la com plantilla generadora, tindríem problemes a l'hora de fer-la servir, és important que els espais, longituds de línia i salts es conservin tal i com es mostren ja que, segons on s'alteri, pot generar un *script* que no funcionaria.

7 RESULTATS

Si, desde l'*script* "__main__" importem els mòduls "atemplate"[20] i "template_generator" se la següent forma:

```
import enb.atemplate as atemplate
import template_generator
```

I, a més incloem les línies:

```
template = atemplate.ATemplate()
template.load_template(
    "/elMeuDirectori/"
    + "/elMeuTemplate/", "elMeuTemplate")
template_generator
    .template_scripts
    .script_generator(template)
```

Sent "elMeuTemplate" l'arxiu Json generador de la figura 13, després de reinstal·lar de, forma local la llibreria*enb*, en executar la comanda "enb"per consola, hauríem d'obtenir l'*script* anomenat "elMeuTemplate.py" al directori especificat. En el peu cas, l'generat ha estat el que es mostra a la figura 12.

Si executem aquest script podem executar un experiment equivalent a l'exemple de *lossless compression*[21], proveït per la pròpia llibreria *Experiment Notebook*.

7.1 Anàlisi de resultats

Si tenim en consideració els objectius a aconseguir, a nivell de programació, sent directes, podem dir que aquest projecte no ha estat un èxit. Si ho mirem exclusivament des d'el punt de vista de l'objectiu d'aconseguir un contacte amb una experiència real de gestió i desenvolupament d'un projecte, aleshores, sí, s'ha assolit l'objectiu.

Segons l'article[22], dues de les causes principals per les que un projecte fracassa són la falta de comunicació i les estimacions poc acurades. I, encara que això es veritat, la realitat és que, sigui de l'àmbit que sigui, un projecte està composat de moltes variables a considerat, fent de les estimacions i planificació un projecte complexe en sí mateix.

8 CONCLUSIONS

La inexperiència en la gestió de projectes ha fet que, des del primer moment, s'elaboressin unes estimacions excessivament optimistes, fet que ha provocat que la contribució al projecte no acabés sent la que s'havia planificat.

Si bé el projecte no ha sigut finalitzat, cal re-visitar els conceptes apressos a l'execució del mateix. L'aprenentatge ha estat extens a tots els àmbits. No sols s'ha hagut de posar en pràctica els coneixements apressos a diferents assignatures del grau, sinó que també s'ha hagut d'exercitar la comunicació, tot i que, al final, ha estat el factor que més ha mancat.

AGRAÏMENTS

En primer lloc, agraeixo al Joan Serra Sagristà, el tutor d'aquest TFG i al Miguel Hernández-Cabronero, el desenvolupador principal de la llibreria *Experiment Norebook* per haver-me guiat en aquest per aquest procés i haver estat, en tot moment, disponibles per qualsevol dubte i haver-se mostrat comprensius inclús quan fallava.

En segon lloc, agraeixo al Daniel Sànchez Gil el temps que ha invertit en ajudar-me a treure endavant el projecte.

Agraeixo a Alberto, a qui jo vaig conèixer per 'Zurita' ara fa ja més de sis anys, que m'hagi ajudat a aquesta tasca.

Així mateix, agraeixo a les persones més properes a mi per haver estat comprensives, inclòs quan els posava dels nervis.

REFERÈNCIES

[1] Open Knowledge Foundation. Air free and open future, . URL https://okfn.org/.

[2] Open Knowledge Foundation. Open definition - the open definition, . URL https://opendefinition.org/.

[3] Open Knowledge Foundation. Open definition, . URL https://opendefinition.org/od/2.1/en/.

[4] Open Knowledge Foundation. Open data handbook - what is open data?, . URL https://opendatahandbook.org/guide/en/what-is-open-data/.

[5] UNESCO. Open science. URL https://en.unesco.org/science-sustainable-future/open-science.

[6] Rim Lassoued, Diego M. Macall, Stuart J. Smyth, Peter W. B. Phillips, and Hayley Hesseln. Expert insights on the impacts of, and potential for, agricultural big data. *Sustainability*, 13(5), 2021. ISSN 2071-1050. doi: 10.3390/su13052521. URL https://www.mdpi.com/2071-1050/13/5/2521.

[7] European open science cloud (eosc). URL https://ec.europa.eu/info/research-and-innovation/strategy/goals-research-and-innovation-policy/open-science/european-open-science-cloud-eosc_en.

[8] M. Wilkinson, M. Dumontier, and I Aalbersberg. The fair guiding principles for scientific data management and stewardship. *Sci Data*, (3), 2021. ISSN 2071-1050. doi: 10.1038/sdata.2016.18. URL https://doi.org/10.1038/sdata.2016.18.

[9] Et. al. Miguel Hernández-Cabronero. Welcome to the experiment notebook, . URL https://miguelinux314.github.io/experiment-notebook/.

[10] Et. al. Miguel Hernández-Cabronero. Experiment notebook, . URL https://github.com/miguelinux314/experiment-Notebook.

[11] Qué es kanban: Definición, características y ventajas. URL https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban.

[12] Jira software. URL https://www.atlassian.com/es/software/jira.

[13] Miguel Hernández-Cabronero, . URL https://github.com/miguelinux314/experiment-notebook/blob/master/enb/singleton_cli.py.

[14] Miguel Hernández-Cabronero, . URL https://github.com/miguelinux314/experiment-notebook/blob/master/enb/experiment.py.

[15] Miguel Hernández-Cabronero, . URL https://github.com/miguelinux314/experiment-notebook/blob/master/enb/aanalysis.py.

[16] Miguel Hernández-Cabronero, . URL https://github.com/miguelinux314/experiment-notebook/blob/master/enb/icompression.py.

[17] Miguel Hernández-Cabronero, . URL https://github.com/miguelinux314/experiment-notebook/blob/master/enb/config.py.

[18] P. Sarqui, . URL https://github.com/AlysH/experiment-notebook/tree/dev_paula/template_generator.

[19] P. Sarqui, . URL https://github.com/AlysH/experiment-notebook/blob/dev_paula/template_generator/templates/experiment_template.py.tlp.

[20] P. Sarqui, . URL https://github.com/AlysH/experiment-notebook/blob/dev_paula/enb/atemplate.py.

[21] Miguel Hernández-Cabronero, . URL https://github.com/miguelinux314/experiment-notebook/blob/master/templates/lossless_compression_example/lossless_compression_experiment_example.py.

[22]

APÈNDIX

A.1 Secció d'Apèndix

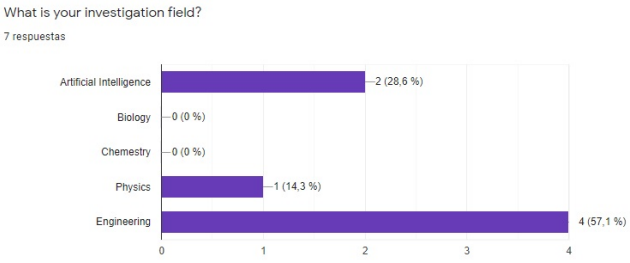

Fig. 3: Exemple de gràfiques més utilitzades.

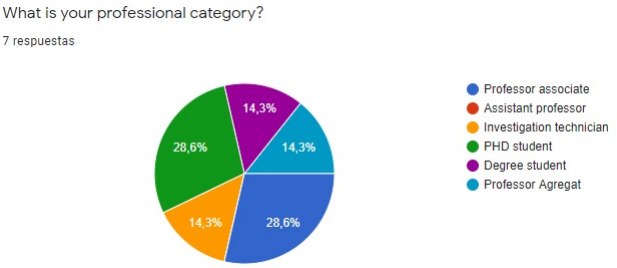

Fig. 4: Exemple de gràfiques més utilitzades.

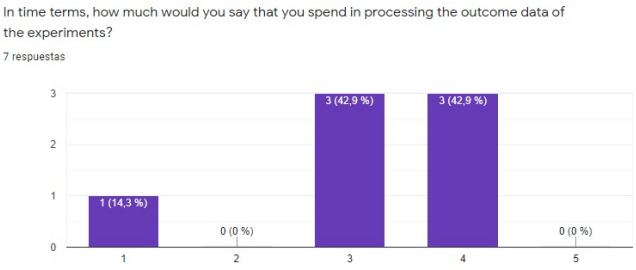

Fig. 5: Exemple de gràfiques més utilitzades.

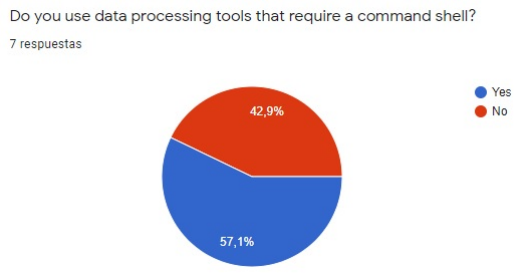

Fig. 6: Exemple de gràfiques més utilitzades.

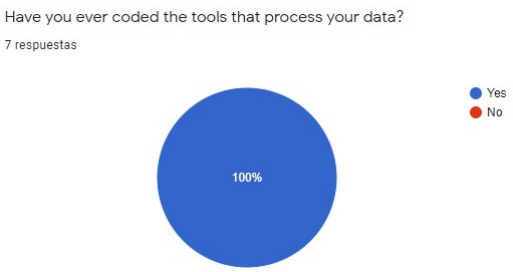

Fig. 7: Exemple de gràfiques més utilitzades.

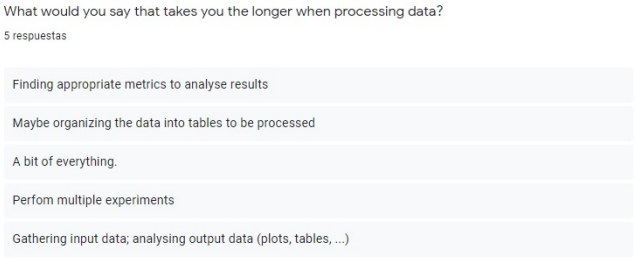

Fig. 8: Exemple de gràfiques més utilitzades.

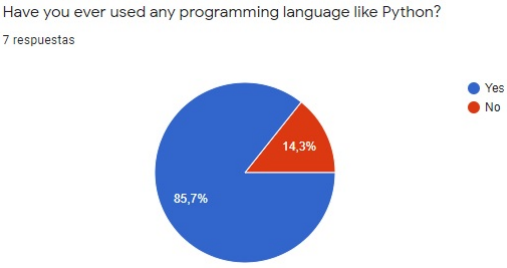

Fig. 9: Exemple de gràfiques més utilitzades.

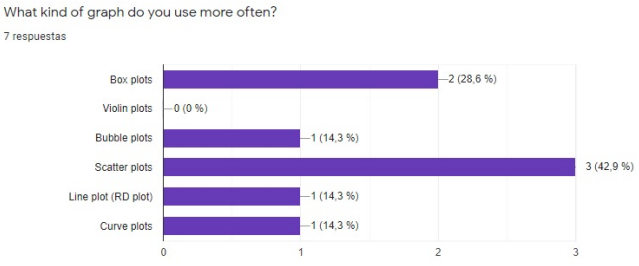

Fig. 10: Exemple de gràfiques més utilitzades.

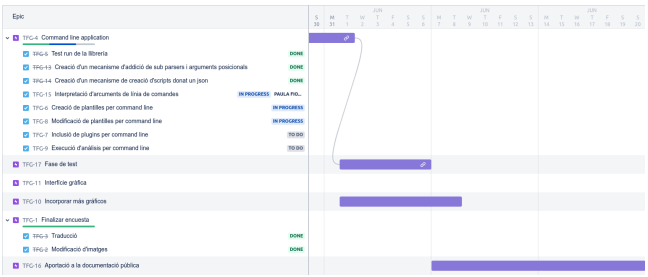

Fig. 11: Diagrama de Gantt resultat de la planificació.

```python
import os
from enb.config import get_options

options = get_options(from_main=False)

from enb import icompression, aanalysis
import plugins

if __name__ == '__main__':
    options.base_dataset_dir = os.path.join("/home/alys/Documents/UAB/2021/TFG/desperation_tests/-
MyThingy/", "data")

    codecs_myExperiment = []
    codecs_myExperiment.append(plugins.plugin_jpeg.jpeg_codecs.JPEG_LS(max_error=0))
    exp_myExperiment = icompression.LosslessCompressionExperiment(codecs=codecs_myExperiment)

    df_myExperiment = exp_myExperiment.get_df()


    analyzer_myAnalysis = aanalysis.ScalarDistributionAnalyzer()
    analyzer_myAnalysis.analyze_df(
        full_df=df_myExperiment,
        target_columns=['compression_ratio_dr', 'bpppc', 'compression_time_seconds'],
        output_csv_file="./analysis/lossless_compression_analysis.csv",
        column_to_properties=exp_myExperiment.joined_column_to_properties,
        group_by="task_label"
    )
```

Fig. 12: Script generat pel generador de templates.

home > alys > Documents > UAB > 2021 > TFG > tests > MyThingy > {} MyThingy.json > ...

```json
{
    "template_name": "myTemplate",
    "template_workdir": "/home/alys/Documents/UAB/2021/TFG/desperation_tests/MyThingy/",
    "plugins": {
        "plugins_dir": null,
        "plugins_dict": {}
    },
    "experiments": [
        {
            "plugins": [
                {
                    "plugin": "plugin_jpeg",
                    "codec": "jpeg_codecs",
                    "class": "JPEG_LS",
                    "parameters": {"max_error": 0}
                }
            ],
            "parameters": {
            },
            "experiment_name": "myExperiment",
            "experiment_type": "LosslessCompressionExperiment",
            "input": "someFile",
            "output": "someInfo"
        }
    ],
    "analysis": [
        {
            "analysis_name": "myAnalysis",
            "analysis_type": "ScalarDistributionAnalyzer",
            "parameters": {
                "full_df": "myExperiment",
                "target_columns": ["compression_ratio_dr", "bpppc", "compression_time_seconds"],
                "output_csv_file": "./analysis/lossless_compression_analysis.csv",
                "column_to_properties": "joined_column_to_properties",
                "group_by":"task_label"
            }
        }
    ]
}
```

Fig. 13: Objecte JSON generador de l'script.