

Usama Benabdelkrim Zakan

Migració i Optimització d'Aplicacions al Núvol

TREBALL DE FI DE GRAU

Pedro García López

Grau d'Enginyeria Informàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2022

Agraïments

En aquesta ocasió, voldria transmetre el meu agraïment més gran per a totes les persones que han fet possible el desenvolupament d'aquest Treball de Fi de Grau.

En primer lloc, al meu tutor, el Doctor Pedro García López, per la seva informació, organització i motivació durant aquest projecte.

En segon lloc, a la meva família, la meva mare Zohra i als meus germans Adel, Walid i Youssef, la meva cunyada Khawla i a les meves nebodes Jannat i Wijdan. Ells s'han deixat la pell, i han dedicat la seva vida a que avui jo pugui tenir el privilegi d'escriure aquestes paraules.

En tercer lloc, a Josep Oliveras i Samitier, geògraf i catedràtic de la Universitat Rovira i Virgili, i al seu cosí Josep per a tot el que han fet per a mi.

En quart lloc, al meu amic Robert Radu i d'altres persones, que sempre m'han recolzat i han tingut paciència amb les meves desaparicions durant cada quadrimestre.

En cinquè lloc, també voldria dedicar aquest Treball de Fi de Grau als que ja no estan entre nosaltres. En concret, als meus avis i a la meva tieta Malika, difunta durant la elaboració d'aquests agraïments.

També m'agradaria expressar el meu agraïment als meus professors de la Universitat Rovira i Virgili per acollir-me, i convertir-me en millor persona. Després de finalitzar aquest projecte, escric aquestes paraules finalitzant així el meu TFG.

No tinc cap mena de dubte que ha estat el projecte que més m'ha fet créixer com a enginyer i persona. Per aquest motiu, m'agradaria fer arribar aquests agraïments a totes les persones que m'han ajudat a finalitzar aquesta etapa.

A tots ells, gràcies de tot cor.

Resum.

En aquest projecte, s'emigra un software de l'àmbit de la metabolòmica al núvol, amb l'objectiu d'aprofitar el potencial d'aquesta tecnologia, per a millorar el rendiment de l'aplicació. El prototip final es basa en una arquitectura serverless, que permet escalar els recursos en funció de la demanda que es produeixi. D'aquesta manera, el sistema s'adapta optimitzant així els recursos utilitzats.

A més, aquesta arquitectura s'ha creat de manera que permeti fàcilment la integració d'altres aplicacions a la plataforma. També s'han dut a terme tasques d'optimització de l'aplicació.

D'altre banda, s'ha creat un sistema d'autenticació d'usuaris per tal de monitoritzar l'ús que facin de la plataforma.

Per últim, s'ha dissenyat una nova interfície que permet als usuaris configurar els seus anàlisis i gestionar els seus resultats.

Resumen.

En este proyecto, se emigra un software del ámbito de la metabolómica a la nube, con el objetivo de aprovechar el potencial de esta tecnología, para mejorar el rendimiento de la aplicación. El prototipo final se basa en una arquitectura serverless, que permite escalar recursos en función de la demanda que se produzca. De este modo, el sistema se adapta optimizando así los recursos utilizados.

A demás, esta arquitectura se ha creado de modo que permita fácilmente la integración de otras aplicaciones a la plataforma. También se han llevado a cabo tareas de optimización de la aplicación.

Por otro lado, se ha creado un sistema de autenticación de usuarios con el objetivo de monitorizar el uso que hagan de la plataforma.

Por último, se ha diseñado una nueva interfaz que permita a los usuarios configurar sus análisis i gestionar sus resultados.

Abstract.

In this project, a metabolomics software is migrated to the cloud, to take advantage of the potential of this technology, to improve the performance of this application. The final prototype is based on a serverless architecture, that allows scaling resources depending on the demand that occurs. In this way, the system adapts to optimize the resources used.

In addition, this architecture has been created in such a way as to allow to allow the easy integration of other applications to the platform. Optimizations task have also been carried out on the application.

On the other hand, a user authentication has been created to monitor their use of the platform.

Finally, a new interface has been designed to allow users to configure their analysis and manage their results.

Índex

1	INTRODUCCIÓ	4
2	DESCRIPCIÓ GENERAL DEL PROJECTE	6
3	REQUISITS	7
3.1	REQUISITS FUNCIONALS	8
3.1.1	<i>Especificació textual dels casos d'ús</i>	8
3.1.2	<i>Diagrama de casos d'ús</i>	12
3.2	REQUISITS NO FUNCIONALS	12
4	ANÀLISI DELS REQUISITS FUNCIONALS	13
5	DISSENY	18
5.1	ARQUITECTURA DE L'APLICACIÓ	18
5.1.1	<i>Arquitectura inicial</i>	18
5.1.2	<i>Arquitectura final</i>	19
5.2	RHERMES REFACTORING	23
5.3	DISSENY DE LA INTERFÍCIE GRÀFICA	25
5.3.1	<i>Configuració de la tasca</i>	26
5.3.2	<i>Explorador de fitxers</i>	32
5.3.3	<i>Autenticació</i>	34
6	IMPLEMENTACIÓ	38
6.1.1	<i>Tecnologies utilitzades (Background)</i>	39
6.1.2	<i>Aspectes rellevants de la implementació</i>	45
6.1.3	<i>Optimització d'RHermes</i>	50
7	AVALUACIÓ	54
7.1	EXPERIMENT PER DEFECTE	55
7.1.1	<i>Registre</i>	55
7.1.2	<i>Autenticació</i>	55
7.1.3	<i>Configuració d'una tasca</i>	56
7.1.4	<i>Descàrrega dels resultats</i>	58
7.2	CASOS D'ERROR	59
7.2.1	<i>Petició de la tasca sense token</i>	59
7.2.2	<i>Petició de la tasca amb un token erroni</i>	59
7.2.3	<i>Petició de la tasca amb un token vàlid</i>	59
8	CONCLUSIONS	60
9	RECURSOS UTILITZATS	62
9.1	BIBLIOGRAFIA WEB	62
9.2	PROGRAMARI	64
10	ANNEXES	64
10.1	ANNEX 1: POSADA EN MARXA	64
10.2	ANNEX 3: MANTENIMENT	64

Índex de taules

TAULA 1. LLISTAT DE REQUERIMENTS NO FUNCIONALS.....	13
---	----

Índex de figures

IL·LUSTRACIÓ 1. DIAGRAMA DE CASOS D'ÚS.....	12
IL·LUSTRACIÓ 2. DIAGRAMA DE CLASSES DEL PRIMER CAS D'ÚS.....	13
IL·LUSTRACIÓ 3. DIAGRAMA DE SEQÜÈNCIES DEL PRIMER CAS D'ÚS.....	14
IL·LUSTRACIÓ 4. DIAGRAMA DE CLASSES DEL SEGON CAS D'ÚS.....	14
IL·LUSTRACIÓ 5. DIAGRAMA DE SEQÜÈNCIES DEL SEGON CAS D'ÚS.....	15
IL·LUSTRACIÓ 6. DIAGRAMA DE CLASSES DEL TERCER CAS D'ÚS.....	15
IL·LUSTRACIÓ 7. DIAGRAMA DE SEQÜÈNCIES DEL TERCER CAS D'ÚS.....	16
IL·LUSTRACIÓ 8. DIAGRAMA DE CLASSES DEL QUART CAS D'ÚS.....	16
IL·LUSTRACIÓ 9. DIAGRAMA DE SEQÜÈNCIES DEL QUART CAS D'ÚS.....	17
IL·LUSTRACIÓ 10. DIAGRAMA DE SEQÜÈNCIES DEL CINQUÈ CAS D'ÚS.....	17
IL·LUSTRACIÓ 11. DIAGRAMA DE CLASSES DEL SISÈ CAS D'ÚS.....	18
IL·LUSTRACIÓ 12. DIAGRAMA DE SEQÜÈNCIES DEL SISÈ CAS D'ÚS.....	18
IL·LUSTRACIÓ 13. DIAGRAMA DE L'ARQUITECTURA DE L'APLICACIÓ.....	20
IL·LUSTRACIÓ 14. DIAGRAMA DE L'EXECUCIÓ D'UNA TASCA SIMPLIFICAT.....	21
IL·LUSTRACIÓ 15. ESQUEMA DEL SISTEMA DE FITXERS DE LA PLATAFORMA.....	22
IL·LUSTRACIÓ 16. DIAGRAMA DEL FUNCIONAMENT DEL PROCÉS A D'HERMES. AUTOR: ROGER GINÉ BERTOMEU.....	24
IL·LUSTRACIÓ 17. DIAGRAMA DEL FUNCIONAMENT DEL PROCÉS B D'HERMES. AUTOR: ROGER GINÉ BERTOMEU.....	24
IL·LUSTRACIÓ 18. DIAGRAMA DEL FUNCIONAMENT DEL PROCÉS C D'HERMES. AUTOR: ROGER GINÉ BERTOMEU.....	24
IL·LUSTRACIÓ 19. DIAGRAMA DEL FUNCIONAMENT DEL PROCÉS D D'HERMES. AUTOR: ROGER GINÉ BERTOMEU.....	24
IL·LUSTRACIÓ 20. CAPTURA DE LA CAPÇALERA DE LA PÀGINA WEB.....	25
IL·LUSTRACIÓ 21. VISUALITZACIÓ ALTERNATIVA DE LA BARRA.....	25
IL·LUSTRACIÓ 22. BARRA DE NAVEGACIÓ DESPLEGADA.....	25
IL·LUSTRACIÓ 23. CAPTURA DE PANTALLA DE LA INTERFÍCIE GRÀFICA CREADA AMB SHINY. AUTOR: USAMA BENABDELKRIM ZAKAN.....	26
IL·LUSTRACIÓ 24. CAPTURA DE PANTALLA DELS PARÀMETRES DEL MÒDUL HERMESANALYSIS.....	27
IL·LUSTRACIÓ 25. EXEMPLE D'UNA ENTRADA NUMÈRICA.....	27
IL·LUSTRACIÓ 26. EXEMPLE DE DESPLEGABLE D'UNA ÚNICA OPCIÓ.....	28
IL·LUSTRACIÓ 27. EXEMPLE DE DESPLEGABLE DE MÚLTIPLES OPCIONS.....	28
IL·LUSTRACIÓ 28. CAPTURA DE LA ZONA D'ENTRADA DE FITXERS DE TIPUS "SAMPLE".....	28
IL·LUSTRACIÓ 29. CAPTURA DE L'ENTRADA DE FITXERS MENTRE S'HI ARROSSEGA UN.....	29
IL·LUSTRACIÓ 30. CAPTURA DE LA ZONA D'ENTRADA DE FITXERS DE TIPUS "BLANK". ES POT APRECIAR COM ES PUJA EL FITXER, I LA INFORMACIÓ D'AQUEST.....	29
IL·LUSTRACIÓ 31. CAPTURA DEL FINAL DE PÀGINA S'APRECIA EL BOTÓ QUE ENVIA LA TASCA AL "BACKEND".....	30
IL·LUSTRACIÓ 32. SOL·LICITUD DE CONFIRMACIÓ DELS PARÀMETRES PER PART DE L'USUARI.....	30
IL·LUSTRACIÓ 33. CAMP PER A INDICAR EL NOM DE L'EXPERIMENT.....	31
IL·LUSTRACIÓ 34. "POP-UP" INFORMATIU.....	31
IL·LUSTRACIÓ 35. CAPTURA DE L'ERROR AL NO SELECCIONAR CAP ELEMENT DE LA "ADDUCTS LIST".....	31
IL·LUSTRACIÓ 36. CAPTURA DELS ERRORS AL NO ADJUNTAR CAP FITXER.....	32
IL·LUSTRACIÓ 37. CAPTURA DE L'ERROR AL NO INDICAR EL NOM DE L'EXPERIMENT.....	32
IL·LUSTRACIÓ 38. CAPTURA D'UN EXEMPLE DE LES CARPETES DELS EXPERIMENTS D'UN USUARI.....	33
IL·LUSTRACIÓ 39. CAPTURA DELS RESULTATS D'UN EXPERIMENT. EN AQUEST CAS S'HA MODIFICAT EL ZOOM DEL NAVEGADOR PER A UNA VISTA COMPLETA.....	34
IL·LUSTRACIÓ 40. CAPTURA DE L'EXPLORADOR DE FITXERS EN CAS DE NO TENIR CAP ANÀLISI FINALITZAT.....	34
IL·LUSTRACIÓ 41. PÀGINA D'INICI DE SESSIÓ PER A ACCEDIR A L'APLICACIÓ WEB.....	35
IL·LUSTRACIÓ 42. PÀGINA DE REGISTRE AL SISTEMA.....	35
IL·LUSTRACIÓ 43. FORMULARI DE VERIFICACIÓ DEL COMPTE.....	36
IL·LUSTRACIÓ 44. MISSATGE D'ERROR EN CAS DE NO OMLIR ELS CAMPS MOSTRATS.....	36
IL·LUSTRACIÓ 45. MISSATGE D'ERROR EN CAS D'INTRODUIR UN NOM D'USUARI O CONTRASENYA ERRONI.....	36
IL·LUSTRACIÓ 46. MISSATGE D'ERROR EN CAS DE NO OMLIR TOTS ELS CAMPS.....	37
IL·LUSTRACIÓ 47. MISSATGE D'ERROR EN CAS D'INTRODUIR UN CORREU NO VÀLID.....	37
IL·LUSTRACIÓ 48. MISSATGE D'ERROR PER A CONTRASENYES DE MENYS DE 6 CARÀCTERS.....	38
IL·LUSTRACIÓ 49. MISSATGE D'ERROR PER A CONTRASENYES SENSE MAJÚSCULES.....	38
IL·LUSTRACIÓ 50. MISSATGE D'ERROR PER A CONTRASENYES SENSE NOMBRES.....	38
IL·LUSTRACIÓ 51. MISSATGE D'ERROR PER A CONTRASENYES SENSE SÍMBOLS.....	38
IL·LUSTRACIÓ 52. MISSATGE D'ERROR PER A INCONSISTÈNCIES EN LA CONFIRMACIÓ DE LA CONTRASENYA.....	38
IL·LUSTRACIÓ 53. CAPTURA DEL PORTAL AL INTRODUIR UN CODI DE VERIFICACIÓ ERRONI.....	38

IL·LUSTRACIÓ 54. DIAGRAMA DE FUNCIONAMENT D'AWS COGNITO. FONT: HTTPS://DOCS.AWS.AMAZON.COM/ES_ES/COGNITO/LATEST/DEVELOPERGUIDE/WHAT-IS-AMAZON-COGNITO.HTML	39
IL·LUSTRACIÓ 55. ESQUEMA D'AWS DE LES FUNCIONALITATS D'S3 I ALGUNS CASOS D'ÚS.....	40
IL·LUSTRACIÓ 56. DIAGRAMA DEL FUNCIONAMENT D'AWS API GATEWAY. FONT: HTTPS://AWS.AMAZON.COM/ES/API-GATEWAY/	41
IL·LUSTRACIÓ 57. ESQUEMA D'AWS PER A MOSTRAR UN CAS D'ÚS DE LAMBDA EN APLICACIONS WEB.....	41
IL·LUSTRACIÓ 58. ESQUEMA D'AWS QUE MOSTRA EL CICLE DE VIDA DE BATCH.....	42
IL·LUSTRACIÓ 59. IL·LUSTRACIÓ DE LA TAULA D'ESPECIFICACIONS D'AWS.....	51
IL·LUSTRACIÓ 60. GRÀFICA DE L'EXECUCIÓ AMB 1 FITXER "SAMPLE".....	51
IL·LUSTRACIÓ 61. GRÀFICA DELS COSTOS DE CADA INSTÀNCIA AMB 1 FITXER "SAMPLE".....	52
IL·LUSTRACIÓ 62. GRÀFICA DE L'EXECUCIÓ AMB 2 FITXERS "SAMPLE".....	52
IL·LUSTRACIÓ 63. GRÀFICA DELS COSTOS DE CADA INSTÀNCIA AMB 2 FITXERS "SAMPLE".....	52
IL·LUSTRACIÓ 64. GRÀFICA DE L'EXECUCIÓ AMB 4 FITXERS "SAMPLE".....	53
IL·LUSTRACIÓ 65. GRÀFICA DELS COSTOS DE CADA INSTÀNCIA AMB 4 FITXERS "SAMPLE".....	53
IL·LUSTRACIÓ 66. GRÀFICA DE L'EXECUCIÓ AMB 8 FITXERS "SAMPLE".....	53
IL·LUSTRACIÓ 67. GRÀFICA DELS COSTOS DE CADA INSTÀNCIA AMB 8 FITXERS "SAMPLE".....	54
IL·LUSTRACIÓ 68. REGISTRE D'UN NOU USUARI A LA PLATAFORMA.....	55
IL·LUSTRACIÓ 69. VERIFICACIÓ D'UN USUARI.....	55
IL·LUSTRACIÓ 70. AUTENTICACIÓ D'UN USUARI.....	55
IL·LUSTRACIÓ 71. PARÀMETRES ESCOLLITS PER A LA TASCA.....	56
IL·LUSTRACIÓ 72. FITXER "BLANK" ADJUNTAT.....	56
IL·LUSTRACIÓ 73. FITXER "SAMPLE" ADJUNTAT.....	57
IL·LUSTRACIÓ 74. CONFIRMACIÓ DELS PARÀMETRES ESCOLLITS.....	57
IL·LUSTRACIÓ 75. INTRODUCCIÓ DEL NOM DE L'ANÀLISI.....	57
IL·LUSTRACIÓ 76. S'INFORMA DE QUE LA TASCA CORRE CORRECTAMENT.....	58
IL·LUSTRACIÓ 77. CAPTURA DE LA TASCA ENVIADA A BATCH.....	58
IL·LUSTRACIÓ 78. SORTIDA VISUALITZADA AL FILEMANAGER.....	58
IL·LUSTRACIÓ 79. CAPTURA DELS RESULTATS DESCARREGATS.....	58
IL·LUSTRACIÓ 80. MISSATGE D'ERROR PER A PETICIONS SENSE EL TOKEN.....	59
IL·LUSTRACIÓ 81. CAPTURA DEL RESULTAT D'UNA PETICIÓ NO VÀLIDA.....	59
IL·LUSTRACIÓ 82. CAPTURA DE LA RESPOSTA D'UNA TASCA ENVIADA CORRECTAMENT.....	60

1 Introducció

El terme núvol o “cloud” cada cop és més freqüentat en el nostre dia a dia. La majoria de persones conviuen i utilitzen directament aquestes tecnologies sense donar-se compte, o entendre realment el que és. En un inici, com qualsevol persona desconeixia aquest món. La meua curiositat va fer-me aprendre sobre el tema cada cop més amb el pas dels anys. Però, no vaig entendre la seva magnitud fins que no vaig iniciar el projecte que es plasma en aquest Treball de Fi de Grau.

Així mateix, el meu interès per aquesta temàtica va incrementar-se per dos factors fonamentals. El primer és el creixement de l’adopció del “cloud” per part de les corporacions i usuaris finals en els últims temps. Cada cop més persones utilitzen els serveis al núvol de les empreses com ara l’iCloud o Google Photos, Google Drive, OneDrive, Dropbox, entre d’altres. D’altre banda, empreses com Netflix, McDondals, Moderna, Fórmula 1, BMW Group, o bp; s’han decantat per solucions “cloud” com AWS¹. De la mateixa manera, d’altres com IBM han proporcionat els seus serveis a empreses com Allianz, Panasonic o Vodafone.

En definitiva, tant els usuaris finals com les corporacions donen el pas cap al núvol degut a una sèrie de beneficis. Entendre en detall aquests beneficis i el funcionament tècnic és un dels motius pel qual sempre m’ha interessat aquesta branca. Però, gràcies a l’assignatura de Sistemes Distribuïts impartida pel professor Pedro García López; vaig apassionar-me realment pel cloud computing. En especial, em va sorprendre el gran potencial de les arquitectures serverless i FaaS².

Certament, durant la pandèmia del COVID-19, les grans empreses tecnològiques van patir un creixement substancial. Aquest fet, el va provocar la extremada dependència de la major part de la població mundial cap a les seves eines. D’aquesta manera, es va facilitar parcialment el seguiment de les activitats acadèmiques i professionals. Arran d’aquesta nova forma de treballar i/o estudiar, tant empreses com entitats acadèmiques han aprofitat els esforços realitzats. Cada cop trobem més teletreball i eines d’estudi remot pels alumnes.

A banda d’això, durant els anys anteriors la indústria del *cloud* ja venia d’una tendència alcista. Però, la situació sanitària i la introducció del 5G van provocar un increment espectacular [1]. Aquest fet, va desencadenar la emigració i optimització de moltes aplicacions i serveis al núvol, per a poder suportar l’agressiva demanda. Per entendre realment l’abast d’aquest mercat, esmentarem algunes dades de la seva evolució.

Convé destacar que les dades corporatives emmagatzemades al núvol, han crescut fins a un 60% l’any 2022. A més, des de l’any 2010 fins a l’actualitat s’ha produït un creixement de la capitalització de la indústria del *cloud computing* i *hosting*, d’un 535%. Altrament, durant l’any 2020, el *Desktop as a Service* (DaaS) va créixer un 95,4%. Aquesta xifra com ens podem imaginar, la va produir el confinament on moltes empreses van optar per teletreballar

¹ AWS: Amazon Web Services

² FaaS: Function as a Service

mitjançant escriptoris virtuals. Però, durant els anys posteriors la resta del mercat (IaaS³, FaaS, etc), ha seguit creixent un 18% de mitja.

Ara bé, *quina ha estat l'experiència per part d'aquestes empreses al cloud?* Doncs bé, en la mateixa font [1], s'especifica que la immensa majoria de les empreses han notat una millora en la seguretat i operativitat. El motiu és simplement que la informació és molt més difícil d'accedir per part dels ciberdelinqüents. Tot i així, gairebé la mateixa xifra d'empreses creu que la seguretat segueix sent la major preocupació. Pel que fa als costos, el 82% de les petites i mitjanes empreses han experimentat una reducció dels costos.

Adicionalment, el 67% de les infraestructures del núvol pertanyen a únicament a 4 empreses: Amazon Web Services, Azure, Google Cloud i Alibaba Cloud. La que posseeix més quota de mercat és AWS amb un 32%.

En relació a les estadístiques estudiades, considero que tots ens podem imaginar l'interès tant dels professionals, com dels acadèmics en aquesta branca de la informàtica, el *cloud computing*. Però, en articles com el *Where are we at with cloud computing?: A descriptive literature review* del 2009 [2], ja es parlava del creixent interès per la recerca del tema aquí tractat. A l'actualitat, la recerca d'aquest sector ha crescut encara més per millorar l'IoT⁴, el sistemes de *real-time* o les aplicacions mòbils; tal i com exposa l'article publicat a *ResearchGate* [3].

Personalment, considero que aquest projecte és extremadament interessant un cop sabem quina és la situació del mercat tecnològic, i la seva previsió segons els experts [4]. El repte iniciat mitjançant aquest treball, m'ha portat a emigrar una aplicació al núvol. A més, aquesta s'ha optimitzat i millorat mitjançant els serveis que proporcionen els proveïdors corresponents. Ara bé, tota l'experiència adquirida crec que pot ser molt útil per crear solucions que facilitin el pas al *cloud*. Objectiu que ja s'han proposat projectes com CloudButton [5].

Per a satisfer el propòsit d'aquest TFG, s'han plantejat diversos objectius que tractarem a continuació.

En primer lloc, s'emigrarà una aplicació web al núvol, passant així a utilitzar recursos del nostre proveïdor *cloud*, en lloc dels recursos locals en aquest cas (tant d'una màquina personal com d'un servidor web).

1. Executar l'aplicació en un contenidor Docker de manera local.
2. Reiterar el punt anterior de forma remota, mitjançant el *cloud*.
3. Escalar l'aplicació.

³ IaaS: Infrastructure as a Service

⁴ IoT: Internet of Things

En segon lloc, es farà un *refactoring* de l'aplicació derivant d'ella 4 mòduls principals. Cadascun esdevindrà en una pàgina independent de l'aplicació. En tercer lloc, s'optimitzaran les tasques més exigents de l'aplicació RHermes.

En quart i últim lloc, es crearà un sistema d'autenticació d'usuaris per permetre fer el seguiment dels recursos consumits i facturar-ho.

Finalment, mitjançant els objectius exposats, es pretén utilitzar les infraestructures del núvol per a crear una aplicació web. En aquesta, els usuaris podran desenvolupar les activitats que desitgin de manera satisfactòria. L'escalabilitat i la transparència de l'arquitectura, permetran evitar la limitació dels recursos locals utilitzats tradicionalment, entre d'altres colls d'ampolla. D'aquesta manera, es millorarà de manera important l'experiència d'usuari o UX.

2 Descripció general del projecte

En aquest projecte es presenta una aplicació anomenada RHermes. Es tracta d'un programa de metabolòmica per identificar compostos en mostres biològiques o ambientals. Concretament, es treballa amb dades de LC-MS⁵ i LC-MS/MS. Com a enginyers informàtics, evidentment desconeixem aquests conceptes en detall. Però per a tenir una visió general, cal esmentar breument del que es tracta. LC-MS es defineix com una tècnica de laboratori que separa físicament compostos, en conjunt de la seva detecció basada en masses. En concret, permet detectar quantitats de, fins i tot nano, grams de substàncies com fàrmacs, productes naturals o pesticides.

Segons els autors d'RHermes, aquesta aplicació resulta adequada per a usuaris de diferents tipus. Per a exemplificar, *big players* (farmacèutiques, fabricants d'instruments, etc), que treballen amb moltes dades i volen simplificar el seu *workflow*. Un altre tipus d'usuari, poden ser científics i personal tècnic que no estan especialitzats en l'àmbit de la bioinformàtica. Per tant, no disposen dels coneixements suficients per a tractar les seves dades, i necessiten d'ajuda externa.

Pel que fa a l'aplicació inclou dos parts: el *frontend* i el *backend*. Ambdues parts s'han programat amb el llenguatge R. Gràcies a l'extensibilitat d'R, s'ha pogut utilitzar Shiny com a *framework* per a la creació de la seva interfície gràfica.

D'altre banda, el *backend* es tracta d'un grup de scripts en R que junts donen vida al *frontend* creat. Cal dir, que aquesta aplicació permet el seu ús mitjançant la consola d'RStudio. Però, per a ser comercialitzada es requereix d'una bona interfície d'usuari, que faciliti el seu ús.

Un altre punt és que RHermes es pot qualificar com una aplicació de tipus *Batch Data Analysis*. És a dir, estem parlant de que l'usuari introdueix una sèrie de dades com a *input* i el temps de processat d'aquestes dades pot ser considerablement alt. Concretament, podem estar parlant d'hores o dies tranquil·lament. Pels informàtics, aquestes temporalitats podem

⁵ LC-MS: Liquid chromatography-mass spectrometry

creure que són extremadament altes. La realitat és que en aquests camps es tracten d'esperes molt normals, on l'usuari deixa que la màquina faci la feina, mentre realitza una altra tasca.

Per tant, l'exigència d'aquestes tasques poden fer insuficients els recursos dels computadors personals. En aquest cas, els propis autors de l'aplicació es troben amb insuficient memòria RAM⁶. En definitiva, l'escalabilitat de recursos pot ser un autèntic problema que es veu solucionat pel *cloud computing*.

Fins ara, hem parlat de l'execució per part de cada usuari en la seva màquina local (que és com ha funcionat fins ara mitjançant la seva instal·lació). Però, si es volgués fer accessible mitjançant un servidor tradicional, el problema anterior seria encara més complicat de solucionar amb un tràfic d'usuaris normal. Segons els experts, els autors d'RHermes, una aplicació d'aquestes característiques podria tenir un tràfic de 10 usuaris simultàniament.

Per satisfer aquestes necessitats, s'ha optat per AWS com a proveïdor de serveis al núvol. Es tracta del major competidor del sector, gràcies als recursos i serveis que proporciona a un preu realment competitiu.

Tal com s'ha dit als objectius de la introducció, el disseny de la nova aplicació contempla la distinció de 4 mòduls principals. Cada mòdul es tractarà d'un script RMarkdown basat en RHermes com a llibreria. Com a excepció, el quart mòdul tractarà la visualització dels resultats. En aquest projecte, s'ha treballat sobre un únic mòdul, desenvolupat per Roger Giné Bertomeu (un dels autors d'RHermes) per a poder crear tota la infraestructura sobre la que correran la resta de mòduls. A l'apartat de disseny, veurem més en detall aquesta arquitectura.

Finalment, cal esmentar que es tracta d'un projecte encara en desenvolupament que contempla la emigració d'altres programaris com Erah [6]. D'aquesta forma, es pretén acabar creant una plataforma amb diverses eines de la branca metabolòmica. Concretament, es tracta del projecte anomenat PDC2021-121799-I00. Aquest, s'està realitzant amb la col·laboració del grup d'investigació de metabolòmica liderat per Óscar Yanes, i al que pertanyo, CloudLab liderat per Pedro García López.

3 Requisits

Quant als requisits del projecte, es divideixen en funcionals i no funcionals. Per un costat, els primers defineixen els anomenats guions, que no són res més que els casos d'ús. Per l'altre, els no funcionals tracten les restriccions mitjançant atributs i mètriques.

⁶ RAM: Random-Access-Memory

3.1 Requisits funcionals

3.1.1 Especificació textual dels casos d'ús

Cd'ú 01. Iniciar sessió

- Resum de la funcionalitat: Determinar qui és l'usuari i presentar-li les funcions corresponents.
- Paràmetres d'entrada: Nom d'usuari i contrasenya.
- Paràmetres de sortida: *Token* de sessió.
- Actors: Usuari, sistema.
- Precondició: Es requereix estar registrat prèviament en el sistema.
- Postcondició: S'ha identificat satisfactòriament l'usuari, i ja té la capacitat d'utilitzar els nostres serveis.
- Procés normal principal:
 1. El sistema demana el nom d'usuari i la contrasenya de l'usuari
 2. L'usuari introdueix el nom d'usuari i la contrasenya.
 3. El sistema presenta la pàgina principal oferint les funcions assignades a l'usuari.

Alternatives de procés i excepcions:

- 3a. El nom de l'Usuari no existeix o la contrasenya no és correcte:.
 - 3a1. El sistema presenta un missatge d'error.
 - 3a2. El sistema torna al pas 1.

Cd'ú 02. Registrar l'usuari

- Resum de la funcionalitat: Crear un nou usuari al sistema per a que pugui satisfer el primer cas d'ús.
- Paràmetres d'entrada: Nom d'usuari, adreça de correu electrònic , contrasenya i codi de verificació.
- Paràmetres de sortida: Cap.
- Actors: Usuari, sistema.
- Precondició: Cap.
- Postcondició: S'ha registrat l'usuari al sistema permetent que pugui iniciar sessió.
- Procés normal principal:
 1. El sistema demana el nom d'usuari, la contrasenya i l'adreça de correu electrònic de l'usuari.
 2. L'usuari introdueix els paràmetres d'entrada esmentats.
 3. El sistema presenta les dades introduïdes per l'usuari.
 4. El sistema demana el codi de verificació enviat al correu de l'usuari.
 5. L'usuari introdueix el codi de verificació.
 6. El sistema el redirigeix al inici de sessió per a poder oferir-li les funcionalitats corresponents.

Alternatives de procés i excepcions:

- 2a. L'adreça de correu electrònic no compleix el format requerit.
 - 2a1. El sistema presenta un missatge d'error.

- 2a2. El sistema torna al pas 1.
- 2b. La contrasenya no compleix el format requerit.
 - 2b1. El sistema presenta un missatge d'error.
 - 2b2. El sistema torna al pas 1.
- 4a. L'usuari introdueix un codi de verificació erroni.
 - 4a1. El sistema presenta un missatge d'error.
 - 4a2. El sistema torna al pas 4.

Cd'ú 03. Realitzar un anàlisis

- Resum de la funcionalitat: Proporcionar les eines necessàries per a que l'usuari pugui encarregar una tasca mitjançant la nostra plataforma.
- Paràmetres d'entrada: Al disposar d'una quantitat considerable de paràmetres, els classificarem en 4 categories:
 1. "Instrument"
 2. "Sample"
 3. "Processing"
 4. Fitxers d'entrada

Adicionalment, s'indicarà el nom de l'experiment.

- Paràmetres de sortida: Un conjunt de fitxers en format .csv, un únic fitxer en format .Rds, i un fitxer en format .zip que comprimirà els arxius anteriors.
- Actors: Usuari, sistema.
- Precondició: Estar registrat i autenticat al sistema.
- Postcondició: S'ha realitzat l'experiment satisfactòriament, visualitzant la sortida en un explorador de fitxers.
- Procés normal principal:
 1. El sistema mostra diferents contenidors per seleccionar els paràmetres d'entrada
 2. L'usuari manté o modifica els paràmetres per defecte que proposa el sistema.
 3. L'usuari selecciona les opcions que desitgi del que s'anomena "Adducts list". En aquest cas, el sistema no selecciona cap opció per defecte.
 4. L'usuari introdueix un fitxer de la categoria "Blank".
 5. L'usuari introdueix un fitxer de la categoria "Sample".
 6. El sistema puja els fitxers al núvol en l'instant que l'usuari els indica.
 7. L'usuari indica que vol enviar la tasca.
 8. El sistema presenta els paràmetres seleccionats per rebre confirmació per part de l'usuari.
 9. L'usuari confirma els paràmetres.
 10. El sistema demana associar un nom a l'experiment.
 11. L'usuari introdueix el nom de l'experiment.
 12. El sistema envia la tasca.
 13. El sistema informa de que la tasca està corrent satisfactòriament.

Alternatives de procés i excepcions:

- 4a. L'usuari no selecciona cap opció de la "Adducts list".
 - 4a1. El sistema mostra un missatge d'error.
 - 4a2. El sistema torna al pas 1.
- 5a. L'usuari no introdueix cap o més d'un fitxer de tipus "Blank".
 - 5a1. El sistema mostra un missatge d'error.
 - 5a2. El sistema torna al pas 1.
- 6a. L'usuari no introdueix cap fitxer de tipus "Sample".
 - 6a1. El sistema mostra un missatge d'error.
 - 6a2. El sistema torna al pas 1.
- 9a. L'usuari tanca la finestra (tècnicament s'anomena "modal").
 - 9a1. El sistema amaga la finestra.
 - 9a2. El sistema torna al pas 1.
- 10a. L'usuari no introdueix cap nom d'experiment.
 - 10a1. El sistema mostra un missatge d'error.
 - 10a2. El sistema torna al pas 10.
- 13a. El sistema pateix un error al intentar córrer la tasca.
 - 13a1. El sistema mostra un missatge d'error.

Cd'ú 04. Consultar els resultats dels experiments

- Resum de la funcionalitat: Proporcionar un explorador de fitxers que permeti a l'usuari consultar i descarregar els resultats.
- Paràmetres d'entrada: Cap.
- Paràmetres de sortida: Cap.
- Actors: Usuari, sistema.
- Precondició: Estar autenticat a la plataforma.
- Postcondició: S'han visualitzat els diversos experiments finalitzats en format de carpetes.
- Procés normal principal:
 1. El sistema proporciona un botó per obrir el "File Manager".
 2. L'usuari accedeix a l'explorador.
 3. El sistema mostra les diferents carpetes que contenen els resultats obtinguts.

Alternatives de procés i excepcions:

- 3a. L'usuari no disposa d'experiments finalitzats.
 - 3a1. El sistema mostra un missatge informant a l'usuari sobre l'absència d'experiments solucionats.

Cd'ú 05. Descarregar els resultats de tots els experiments

- Resum de la funcionalitat: Permetre a l'usuari la descàrrega de tots els resultats dels seus experiments finalitzats.
- Paràmetres d'entrada: Cap.
- Paràmetres de sortida: Un o més fitxers en format .zip que contenen les sortides.
- Actors: Usuari, sistema.

- Precondició: Estar autenticat a la plataforma i tenir resultats disponibles.
- Postcondició: S'han descarregat diversos fitxers .zip, concretament, un per cada experiment.
- Procés normal principal:
 1. El sistema visualitza els resultats a l'explorador de fitxers de *l'app*.
 2. El sistema proporciona un botó per fer la descàrrega.
 3. L'usuari demana fer la descàrrega de tots els seus experiments mitjançant el botó.
 4. El sistema en els propers instants obrirà l'explorador de la màquina per a que l'usuari esculli la ubicació i el nom d'un fitxer.
 5. L'usuari guarda el fitxer.
 6. Es reiteren els passos 3 i 4 fins que no quedin fitxers per a descarregar.

Alternatives de procés i excepcions:

- 3a. L'usuari vol descarregar un únic experiment
 - 3a1. L'usuari escull la carpeta desitjada.
 - 3a2. El sistema mostra totes les sortides de l'experiment.
 - 3a3. L'usuari selecciona fer la descàrrega dels resultats mitjançant un botó.
 - 3a4. El sistema continua amb el pas 4 ignorant el pas 6.
- 3a2a. Com a alternativa al pas 3a2 de l'alternativa 3a, l'usuari pot escollir un fitxer concret del total de resultats mostrats.
 - 3a2a1. L'usuari selecciona el fitxer desitjat.
 - 3a2a2. El sistema continua amb el pas 3a4.

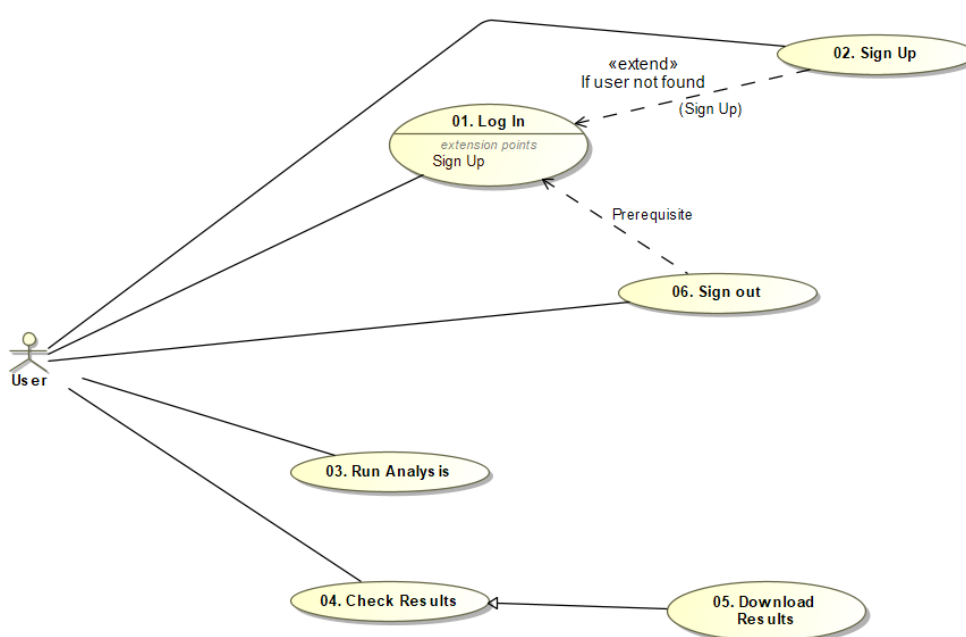
Cd'ú 06. Tancar sessió

- Resum de la funcionalitat: Es permet tancar al sessió de l'usuari finalitzant així l'ús de la plataforma.
- Paràmetres d'entrada: Cap.
- Paràmetres de sortida: Cap.
- Actors: Usuari, sistema.
- Precondició: Estar autenticat a la plataforma.
- Postcondició: S'ha tancat la sessió.
- Procés normal principal:
 1. El sistema proporciona un botó per tancar la sessió.
 2. L'usuari tanca la sessió.
 3. El sistema gestiona el tancament de la sessió eliminant les seves credencials del navegador.

3.1.2 Diagrama de casos d'ús

Amb l'objectiu de visualitzar les funcionalitats de l'aplicació, s'ha elaborat un diagrama de casos d'ús. En aquest, trobem un actor principal que inicia tots els casos especificats.

En primer lloc, trobem que a l'inici de sessió en cas de no estar registrat, l'usuari es veurà redirigit al cas d'ús 02. *Sign Up*. D'altra banda, pel cas d'ús per tancar la sessió (06. *Sign Out*) cal que l'usuari estigui autenticat al sistema. En cas contrari, no es podrà executar aquest. Finalment, la resta de casos d'ús, un cop autenticats no requereixen de cap altre condició. Però, pel que fa al cas 05. *Download Results* s'especifica una herència en conjunt al cas 04. *Check Results*, degut a que inclou alguna acció més respecte a aquest últim. En altres paraules, per executar la descàrrega de fitxers cal accedir al "File Manager".



Il·lustració 1. Diagrama de casos d'ús

3.2 Requisits no funcionals

Pel que fa als requisits no funcionals, s'especificaran les característiques del funcionament de la plataforma. Per tant, aquest apartat tractarà els atributs que caracteritzin el propi sistema.

Requeriment	Descripció
Fiabilitat	El sistema haurà de funcionar permanentment, al mateix temps que manté la integritat de les dades.
Disponibilitat	La plataforma ha d'estar operativa els 365 dies de l'any, durant les 24h del dia.
Estabilitat	L'aplicació mantindrà un nombre de fallades el més proper possible del 0.

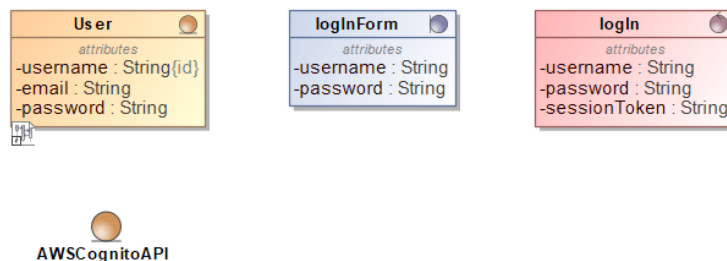
Mantenibilitat	El bon disseny i la documentació facilitaràn les tasques de manteniment.
Escalabilitat	Davant dels canvis de la infraestructura i els pics de tràfic de la plataforma, aquesta haurà de mantenir un correcte funcionament i rendiment.
Portabilitat	El sistema ha de permetre ser executant des de qualsevol plataforma o sistema operatiu.
Elasticitat	L'aplicació proporcionarà la quantitat justa de recursos per satisfer una tasca concreta.
Privacitat	La plataforma ha de vetllar per la confidencialitat de les dades que maneguin els usuaris.
Seguretat	El sistema haurà de mantenir de manera privada les credencials dels usuaris i la seva informació. També controlarà l'accés als recursos computacionals.
Integritat de dades	S'hauran de proporcionar infraestructures i mecanismes que permetin garantir la integritat de les dades. En concret, es faran revisions regularment.
Robustesa	El sistema controlarà tot tipus d'entrades errònies en qualsevol dels casos d'ús especificats.
Rendiment	El sistema mantindrà un bon rendiment de processat o <i>throughput</i> . Addicionalment, es requerirà una ràpida transmissió de dades i baixa utilització de recursos computacionals.

Taula 1. Llistat de requeriments no funcionals.

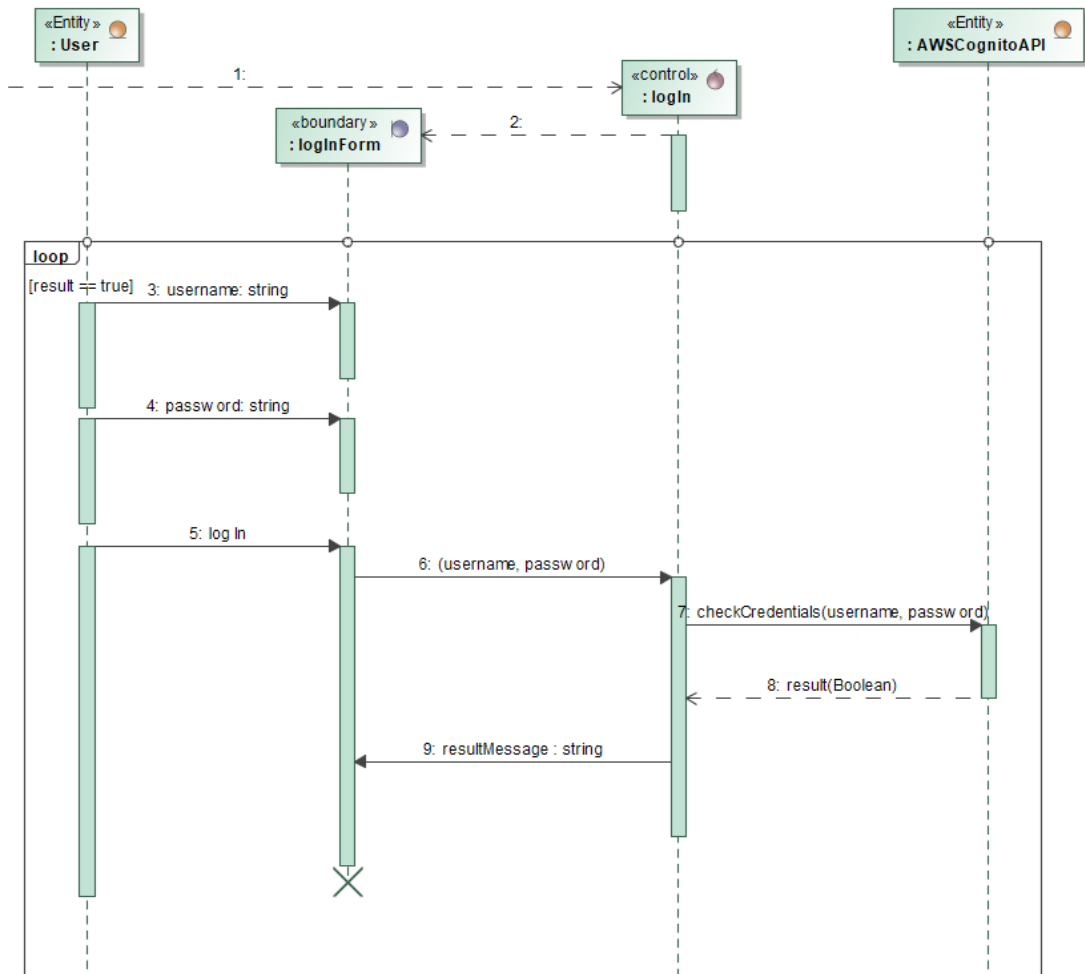
4 Anàlisi dels requisits funcionals

A continuació, es realitza un anàlisi dels requisits funcionals, mitjançant diagrames en UML. Concretament, s'ha utilitzat la eina MagicDraw utilitzada a l'assignatura d'ADA.

Cd'ú 01. Iniciar sessió

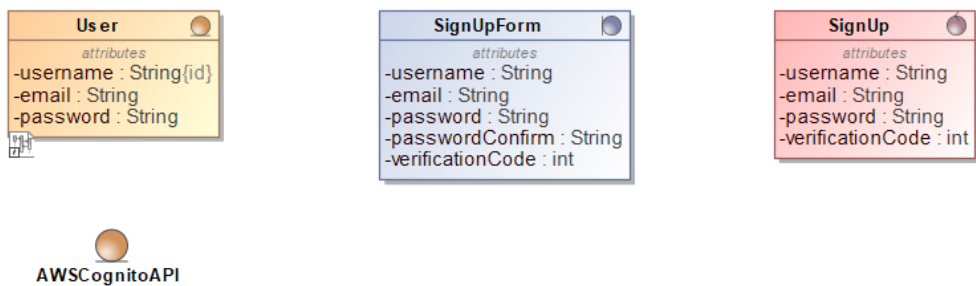


II·lustració 2. Diagrama de classes del primer cas d'ús.

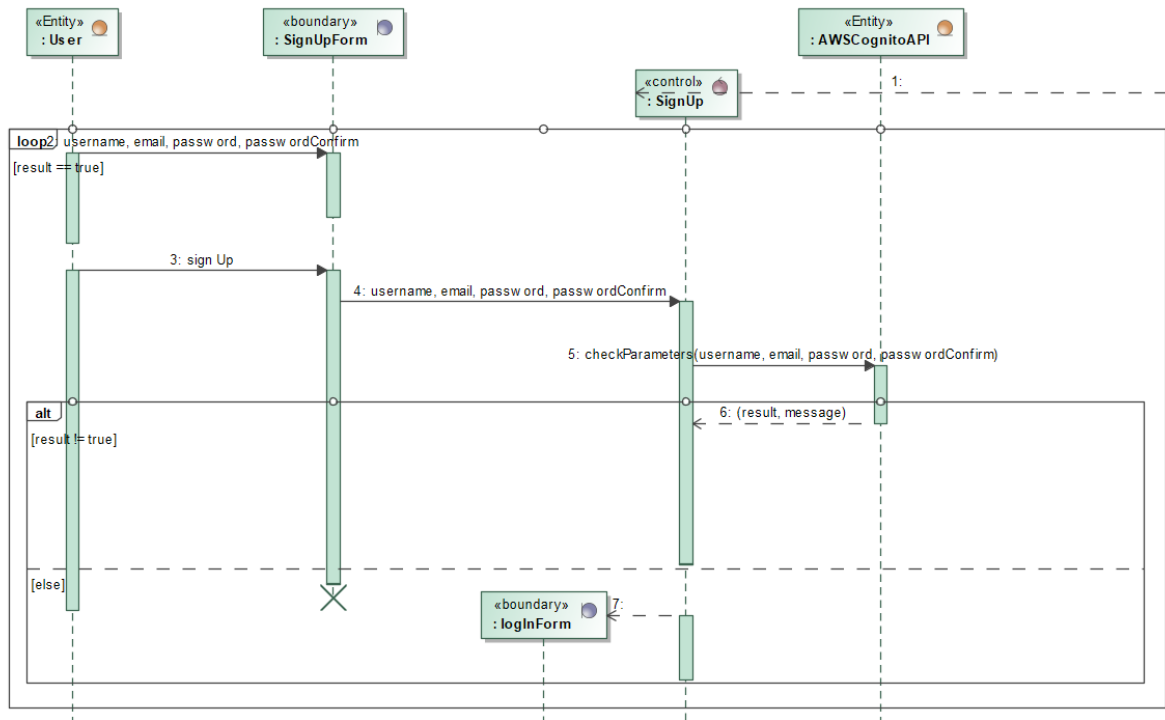


Il·lustració 3. Diagrama de seqüències del primer cas d'ús.

Cd'ú 02. Registrar l'usuari

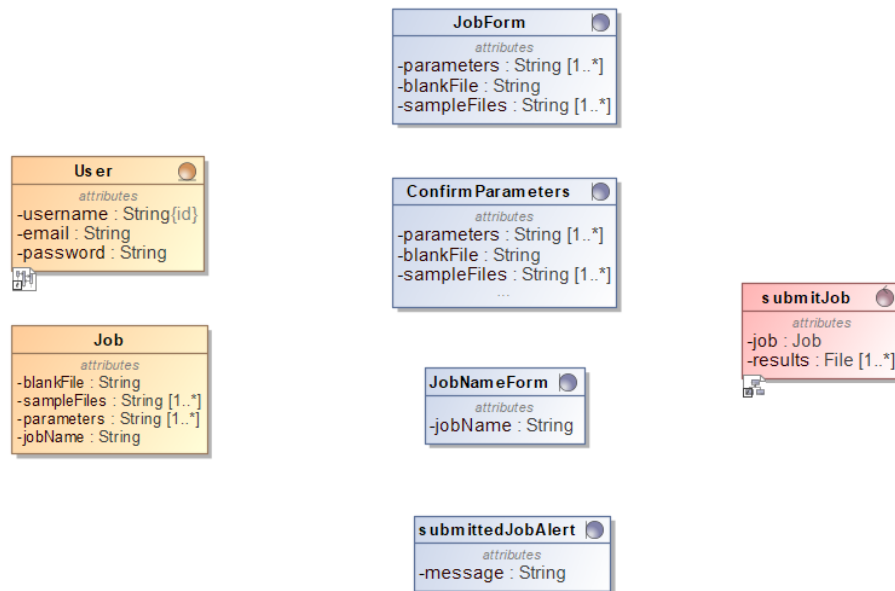


Il·lustració 4. Diagrama de classes del segon cas d'ús.

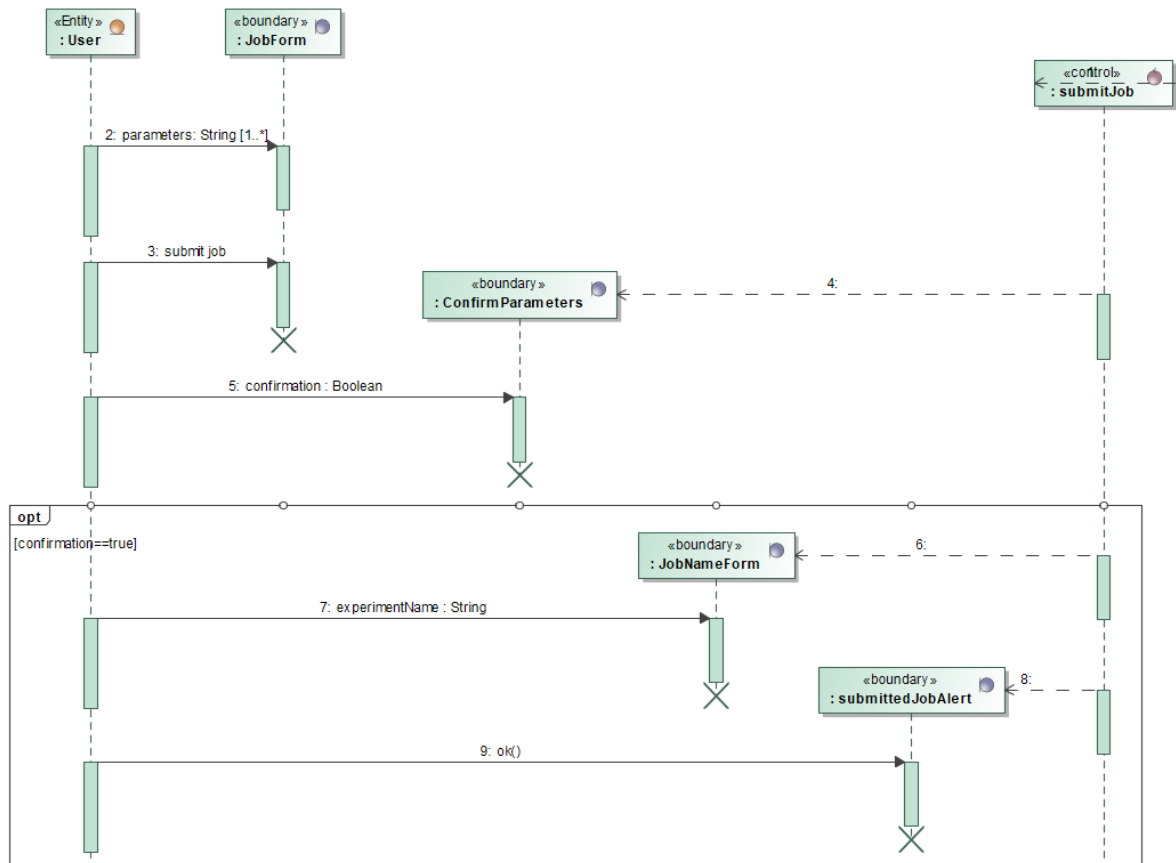


Il·lustració 5. Diagrama de seqüències del segon cas d'ús.

Cd'ú 03. Realitzar un anàlisis



Il·lustració 6. Diagrama de classes del tercer cas d'ús.

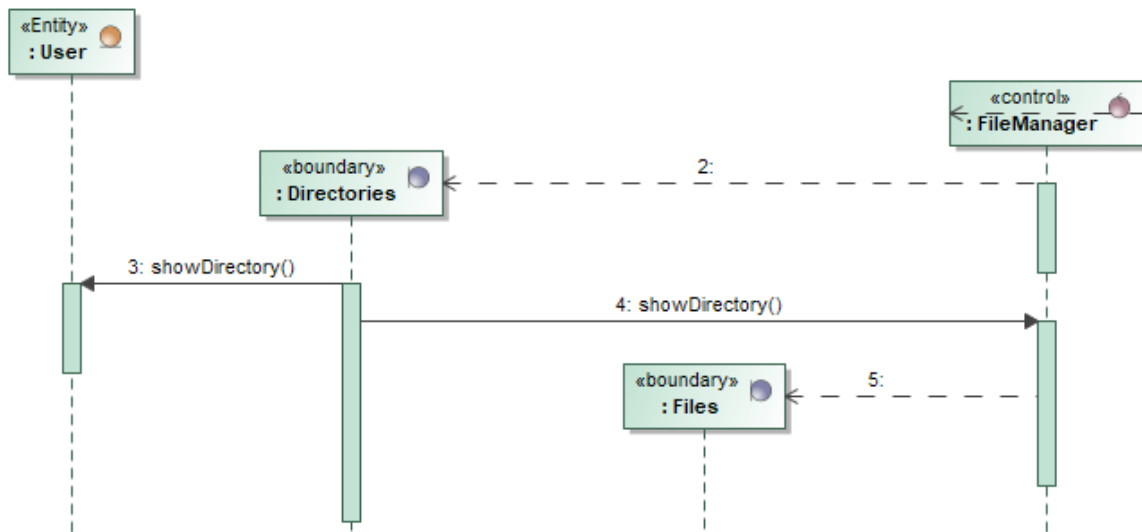


Il·lustració 7. Diagrama de seqüències del tercer cas d'ús.

Cd'ú 04. Consultar els resultats dels experiments

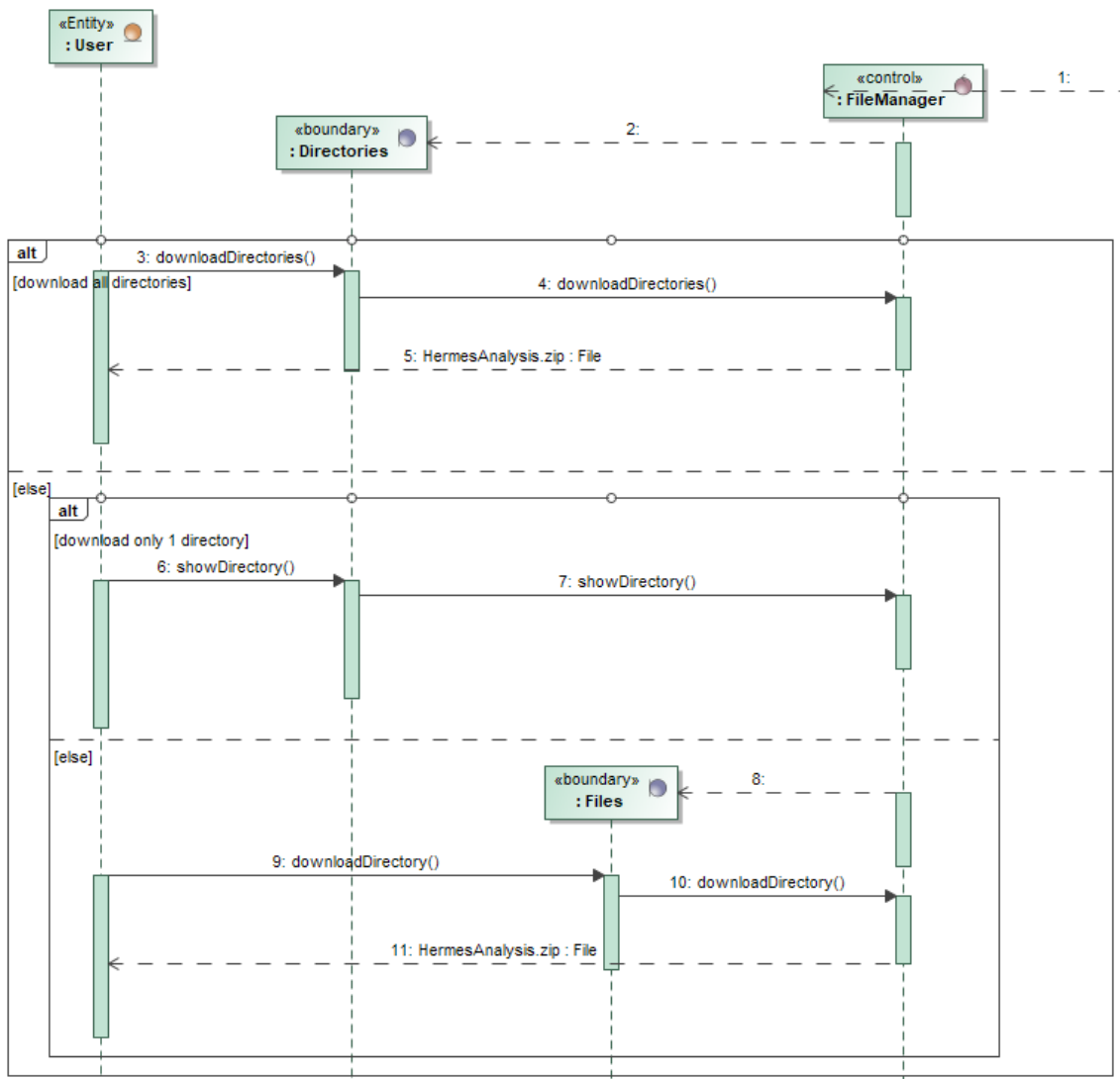


Il·lustració 8. Diagrama de classes del quart cas d'ús.



Il·lustració 9. Diagrama de seqüències del quart cas d'ús.

Cd'ú 05. Descarregar els resultats de tots els experiments

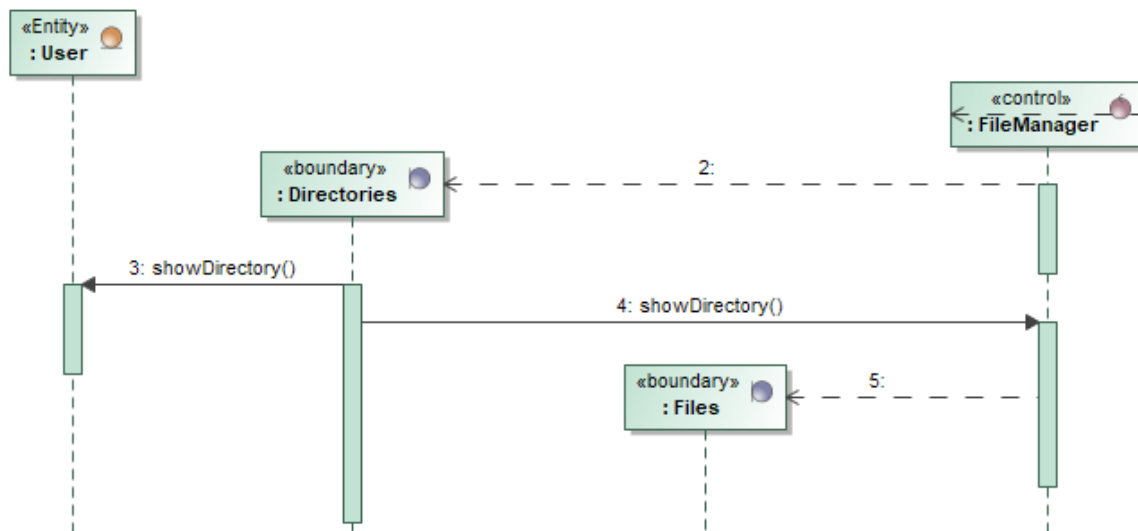


Il·lustració 10. Diagrama de seqüències del cinquè cas d'ús.

Cd'ú 06. Tancar sessió



Il·lustració 11. Diagrama de classes del sisè cas d'ús.



Il·lustració 12. Diagrama de seqüències del sisè cas d'ús.

5 Disseny

5.1 Arquitectura de l'aplicació

Des d'un primer moment, es van proposar 2 grans fases, que encapsularan l'evolució de la feina realitzada en aquest projecte. D'aquesta forma, es pretén iniciar una corba d'aprenentatge per part del desenvolupador, amb l'objectiu d'arribar a un segon disseny amb l'aplicació funcionant al núvol. Posteriorment, es procedirà en la millora d'aquesta.

5.1.1 Arquitectura inicial

En primer lloc, es crea un contenidor local que permetrà l'execució de l'aplicació en diferents entorns. Per ara, només caldrà obtenir la correcte execució d'RHermes sense cap interacció amb el núvol; en altres paraules, localment.

Un cop assolit el primer objectiu, mitjançant el servei d'EC2 d'AWS, s'instal·larà el contenidor en una instància prèviament configurada. D'aquesta manera, s'aconsegueix el funcionament del sistema en una màquina al núvol, la qual pot ser accedida mitjançant el

navegador i la URL⁷ de la màquina. La millora obtinguda fins ara, no és res més que la capacitat de poder executar un servei d'es d'un servidor. Aquest, pot tenir diferents característiques tècniques en funció de la necessitat (RAM, nombre de processadors, entre d'altres). Per a ser usada per un únic usuari pot arribar a ser suficient. Però, tal i com s'ha explicat anteriorment, es pretén crear un servei que sigui capaç de suportar múltiples usuaris simultàniament.

Per tant, aquesta arquitectura no compleix els requisits no funcionals especificats. Principalment, no es satisfan ni la escalabilitat, ni la elasticitat del sistema. Per un costat, es poden estar pagant recursos que no cal consumir per a tasques més senzilles. Altrament, per a pics de demanda no tenim cap mena de mecanisme que incrementi les màquines o recursos que utilitza la plataforma.

A més, s'haurà de mantenir engegada la màquina les 24h del dia durant tot l'any per a satisfer la disponibilitat. Durant les estones que no s'estigui utilitzant el servei, estariem pagant per una màquina que no estem necessitant.

En definitiva, fins ara no s'ha creat res més que un servidor que atén les peticions dels usuaris. Els seus recursos són limitats, i per tant, insuficients per a fluctuacions importants del tràfic. Però, això no deixen de ser passos entremitjos per a que de manera progressiva, s'arribin a satisfer aquests requisits.

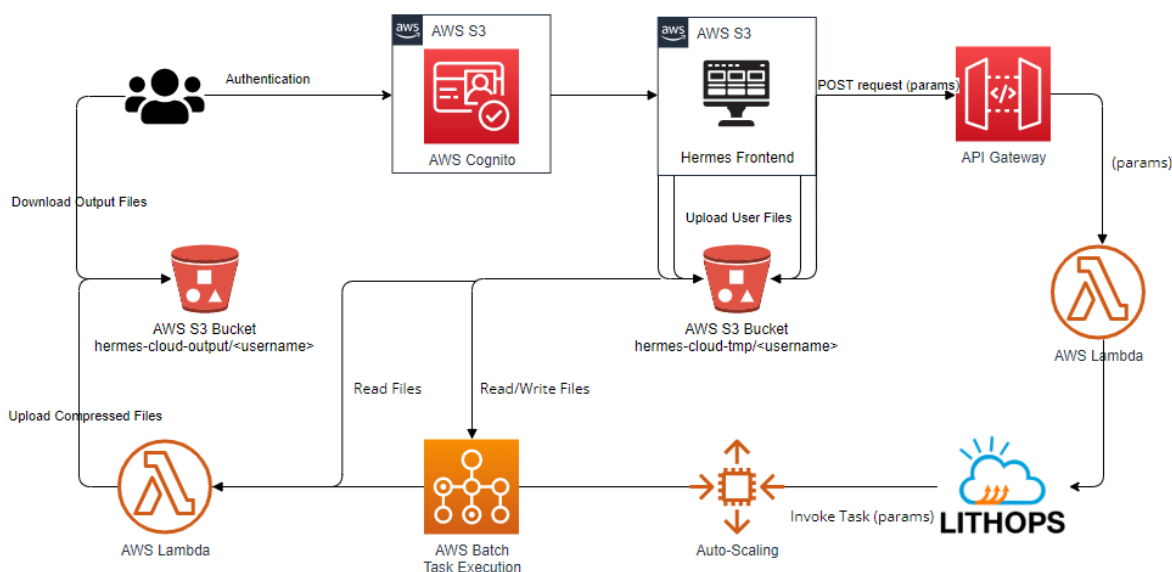
Finalment, noteu que pel moment no s'ha dissenyat cap base de dades, entre d'altres elements. És una versió inutilitzable pels nostres usuaris ja que els resultats dels anàlisis es guarden localment a la màquina (no deixa de ser una aplicació pensada per l'execució local). Per accedir a ells, s'ha d'utilitzar la CLI d'AWS i copiar-los on es desitgi, i no és una forma adequada de treballar pels clients.

Per aquests motius, un cop comprovat el correcte funcionament d'RHermes en aquest entorn, s'ha procedit a dissenyar una arquitectura final.

5.1.2 Arquitectura final

A continuació, es tractarà el segon disseny esmentat, on ja disposem d'una aplicació funcionant des d'un servidor al núvol. La intenció és obtenir un prototip el més proper possible de la versió final, mitjançant diversos increments d'aquest. Després de diferents propostes, s'ha acabat implementant la versió que es veu a la *Il·lustració 13*.

⁷ URL: Uniform Resource Locator



Il·lustració 13. Diagrama de l'arquitectura de l'aplicació

Els següents apartats tracten les diferents funcionalitats que disposa actualment Hermes al núvol, *HermesCloud*. Cadascun d'aquests es tradueix en una pàgina de l'aplicació web. Totes aquestes són allotjades en un *bucket* d'S3, el qual admet l'accés públic a cada element mitjançant un enllaç generat per AWS.

5.1.2.1 Configuració de la tasca

Qualsevol anàlisi que es realitzi en el mòdul d'HermesAnalysis, comporta 4 etapes diferents. A continuació, veurem en detall cadascuna d'aquestes.

5.1.2.1.1 Gestió dels fitxers d'entrada

Com veurem al disseny, concretament a l'apartat 5.2.2 *Configuració de la tasca*, l'usuari haurà d'inserir diversos fitxers. Aquests, es guardaran a la carpeta personal de l'usuari en un *bucket* d'S3. D'aquesta manera, aquest acabarà contenint multitud de directoris diferents, on cadascun correspondrà a un usuari concret. A cada carpeta, es trobaran els diferents fitxers generats. D'aquesta manera, el sistema permetrà al *backend* llegir els fitxers dinàmicament, accedint a aquest repositori.

D'altre banda, un tema interessant i que està fora del meu abast, és el temps de la disponibilitat d'aquests fitxers. Com és d'esperar, AWS cobra una quantia depenent del volum d'espai ocupat. Per això, cal esperar a que el director del projecte, Óscar Yanes, ens comuniqui quina temporalitat ens interessa establir, des del punt de vista econòmic. D'aquesta manera, si l'usuari volgués fer un altre anàlisi amb un fitxer utilitzat anteriorment, no caldrà tornar a pujar-lo. A més, un cop caduquin els fitxers, s'eliminaran automàticament.

5.1.2.1.2 Tramitació de la tasca

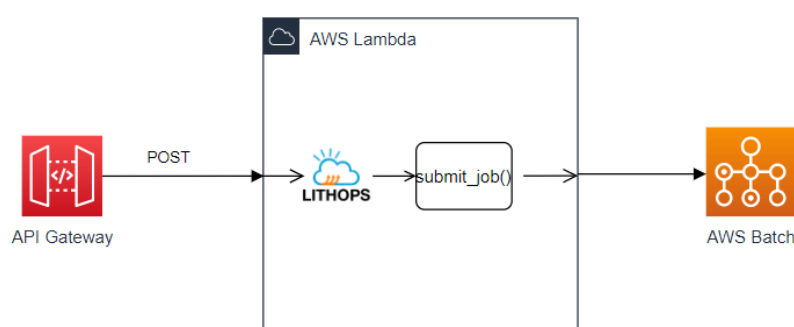
Com s'ha esmentat, aquesta es tracta d'una pàgina web estàtica. El que això implica en aquest cas, és que es passen els paràmetres en format JSON al *backend*. Un cop reunits, s'enviaran a través d'una petició *POST* al servei API Gateway d'AWS.

Un cop arriba la petició a l'API Gateway, aquest s'encarregarà d'invocar una funció d'AWS Lambda. Aquesta, rebrà la petició amb els paràmetres de l'usuari, i s'encarregarà d'encuar la tasca a AWS Batch. D'aquesta manera, des d'una web estàtica es permetrà executar un

anàlisi dinàmicament, sense tenir un servidor encès les 24h del dia. Aquest disseny pren més sentit en una aplicació com aquesta, on el nombre d'usuaris no és extremadament elevat; i per tant, estan més dispersos en el temps.

5.1.2.1.3 Execució de la tasca

Pel que fa a la funció Lambda, un cop és invocada s'encarrega d'executar Hermes mitjançant l'API de Lithops. Ho veurem més en detall a la implementació. Però, en aquest cas s'ha especificat que la tasca s'executi sobre AWS Batch. Per tant, Lambda crearà una tasca amb els paràmetres enviats a la petició. És a dir, s'executarà l'script `HermesAnalysis.Rmd` passant-li el contingut del JSON anterior. Posteriorment, la tasca s'enviarà a una cua de tasques de Batch. Un cop Batch executa la tasca, la Lambda es tancarà automàticament, de manera que la resta de la feina la farà Batch.



Il·lustració 14. Diagrama de l'execució d'una tasca simplificat.

Noteu que Batch és qui realment executa Hermes. Per tant, aquest servei d'AWS és el que permet escalar els recursos en funció de la seva demanda. La resta de components s'encarreguen de poder crear aquestes tasques basant-se en esdeveniments (provocats pels usuaris).

Així mateix, Batch acabaria formant una cua de tasques on s'executin paral·lelament les que contingui. D'aquesta manera, cada tasca disposa d'un contenidor dedicat, per a no afectar al rendiment per a molts elements que tingui a la cua.

D'altra banda, durant l'elaboració del projecte s'ha dissenyat una forma d'optimitzar els recursos que utilitza Batch per a executar Hermes. Depenent del nombre de fitxers a processar i els paràmetres especificats, l'exigència de la tasca pot variar considerablement. En altres paraules, depenent de l'anàlisi podem estar utilitzant més o menys recursos dels necessaris. Ambdós casos són negatius, ja que podríem estar pagant recursos innecessaris, o utilitzant-ne menys del compte, afectant així al temps d'espera de l'usuari. Per això, s'ha pensat en tractar dos aspectes del sistema: el nombre de vCPUs⁸, i la quantitat de RAM.

Per un costat, tal i com veurem a la implementació, Hermes s'ha optimitzat pel que fa al processat de múltiples fitxers. Per tant, es pot configurar Lithops per a que creï una tasca de

⁸ vCPU : virtual CPU

Batch, que s'executi sobre el mateix nombre de processadors que de fitxers. D'aquesta manera, cada processador s'encarregaria de processar un fitxer paral·lelament.

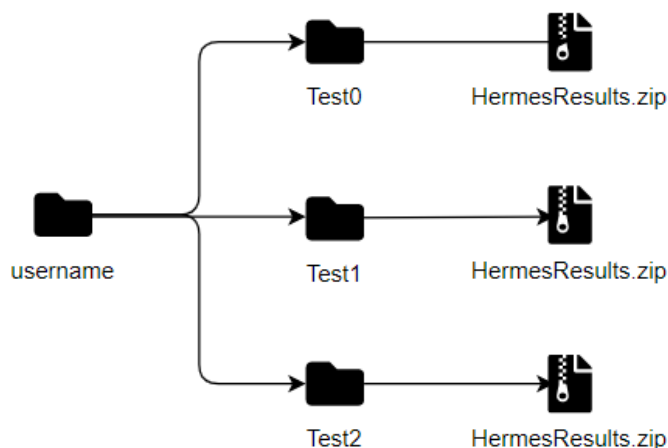
Per l'altre, la RAM es podria optimitzar de la mateixa manera que la CPU; en el sentit d'indicar la quantitat de memòria mitjançant la configuració de Lithops. Ara bé, el càlcul d'aquesta quantia és més complicada de realitzar. En primer lloc, cal esperar a la futura versió d'Hermes, on s'optimitzarà més a fons. A més, caldrà fer multitud d'experiments, i per cadascun variar els paràmetres més rellevants, pel que fa al cost computacional. El motiu és que l'exigència de la tasca no creix en funció de la mida dels fitxers, sinó que depèn dels paràmetres també. Per un únic paràmetre diferent, per exemple, podem passar d'un temps d'execució de 7 minuts a 5 hores.

En definitiva, Lithops ens permet aportar elasticitat al sistema, modificant simplement el seu fitxer de configuració (en aquest cas, ubicat a la Lambda). Però, degut a les dificultats exposades pel càlcul de la RAM, s'han prioritzat d'altres aspectes del projecte.

En resum, la web estàtica mitjançant API Gateway i Lambda, ens permet crear tasques d'Hermes sobre AWS Batch. Aquest mecanisme, permetrà que d'una manera molt econòmica, el sistema tingui la capacitat de servir els clients de la manera més òptima possible.

5.1.2.1.4 Gestió dels fitxers de sortida

Pel que fa als fitxers de sortida, cal esmentar que són generats per Hermes. Un cop finalitzi el processat d'HermesAnalysis, el mateix script Rmd⁹ pujarà els fitxers a un bucket d'S3. En aquest cas, s'anomena *hermes-cloud-output*. Al seu interior trobarem diferents carpetes, on cadascuna correspondrà a un usuari. A més, dins de cada directori trobarem una carpeta per a cada experiment realitzat per l'usuari, per a una millor organització.



Il·lustració 15. Esquema del sistema de fitxers de la plataforma.

Com que els resultats dels anàlisis poden tractar-se de al voltant de 20 fitxers, descarregar-los un per un pot ser una mala experiència d'usuari. Per a cadascun s'obriria

⁹ Rmd: RMarkdown

l'explorador de fitxer, per a escollir el seu nom i la ubicació. Per aquest motiu, es planteja la opció de comprimir-los tots en un fitxer per a ser descarregat. Per a fer-ho es presenten dos opcions: executar el codi que els comprimeix al navegador del client, o fer-ho des del *backend* per a pujar-ho a S3 i descarregar-lo.

En un principi, es va implementar la primera opció evitant així passos addicionals com pujar el fitxer comprimit a S3, a més d'estalviar els costos addicionals per a comprimir la sortida. Un cop implementada, a l'ordinador personal la funcionalitat funcionava sense problema. Però, en d'altres ordinadors s'han experimentat errors i inconsistències (múltiples factors: S.O, navegador, versió del navegador, etc).

Amb l'objectiu d'evitar aquests i d'altres possibles problemes, s'ha acabat desenvolupat la segona opció. Des de Batch, un cop finalitzada l'execució d'HermesAnalysis, s'invoca a una funció Lambda anteriorment creada. Aquesta es tracta d'un altre contenidor que executa un script en Python. Aquest codi, descarrega tots els fitxers de la ruta especificada (<username>/<experiment-name>"). Un cop finalitzades les descàrregues, es comprimirà la sortida. Per acabar, es pujarà el fitxer comprimit a S3.

5.1.2.2 Explorador de fitxers

Pel que fa a l'explorador de fitxers, permet saber si els resultats d'un anàlisi estan disponibles per a descarregar-los. Tal i com s'ha dit, Hermes es tracta un software de tipus *Batch Analysis*. L'usuari normalment no estarà pendent dels resultats, ja que el temps d'espera pot ser elevat. Per això, no cal res més que de tant en tant, el client consulti els seus fitxers. Per a cada consulta, el *FileManager* visualitzarà el contingut de la carpeta d'S3 corresponent.

5.1.2.3 Autenticació

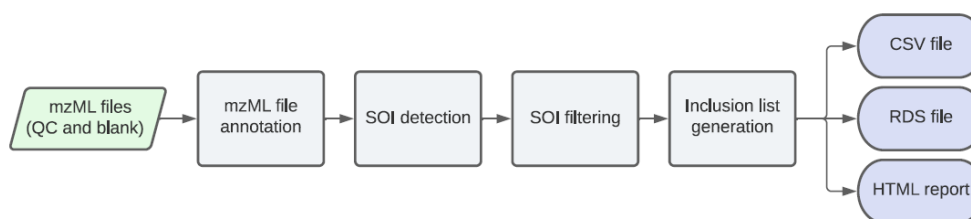
Un cop l'usuari accedeix a la nostra web, es trobarà amb un portal dedicat a la identificació dels clients. Aquest portal, presenta dos formularis principals. El primer que és el que es presenta des d'un inici, està dedicat a l'inici de sessió. L'altre, servirà pel registre d'usuaris a la nostra plataforma. Al disseny de la interfície veurem més en detall aquesta plana.

Per a fer possible aquesta funcionalitat, s'ha optat per l'ús d'AWS Cognito. Aquest servei permet crear i autenticar als usuaris. Però, l'estudiarem més a fons a les tecnologies utilitzades.

5.2 RHermes *refactoring*

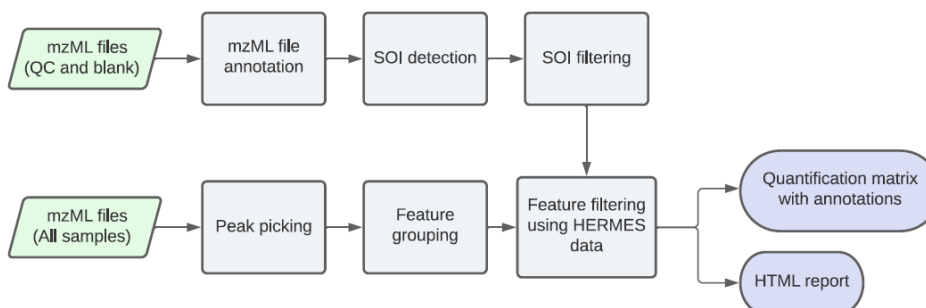
Tal com s'ha dit, durant aquest projecte s'ha redissenyat RHermes. En concret, s'ha dividit en 4 *workflows* principals. Cadascun està format per una sèrie de fases o funcions que treballen sobre una sèrie de paràmetres i fitxers d'entrada. A més, cadascun genera una sortida que se li proporcionarà al client.

Process A: MS1 Sample characterization



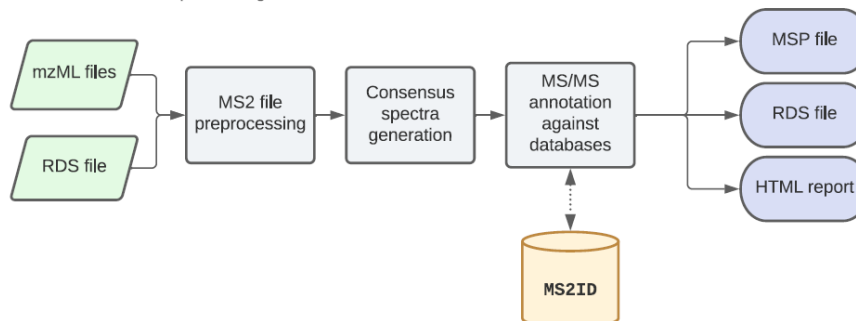
II·lustració 16. Diagrama del funcionament del procés A d'Hermes (HermesAnalysis). Autor: Roger Giné Bertomeu.

Process B: MS1 Sample characterization and quantification



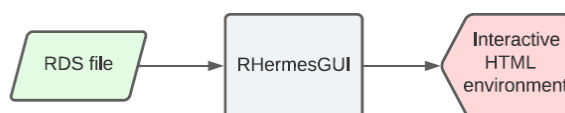
II·lustració 17. Diagrama del funcionament del procés B d'Hermes. Autor: Roger Giné Bertomeu.

Process C: MS2 Data processing and annotation



II·lustració 18. Diagrama del funcionament del procés C d'Hermes. Autor: Roger Giné Bertomeu.

Process D: Data visualization module



II·lustració 19. Diagrama del funcionament del procés D d'Hermes. Autor: Roger Giné Bertomeu

Un cop definits els 4 processos, l'autor proposa crear un disseny modular que permeti als usuaris combinar els diferents passos dels *workflows*. Per tant, això requereix d'una interfície que permeti combinar aquestes fases de manera senzilla (per exemple, mitjançant arrosseables). Però, per ara la prioritat és implementar els processos per defecte.

Per últim, cal dir que la complexitat d'aquest projecte ha estat crear tota la infraestructura que permetrà l'execució de tots els mòduls. Per això, en aquest cas s'ha treballat només sobre el procés A, *HermesAnalysis*. Pel que fa a la resta, la seva implementació serà relativament senzilla, ja que s'han dissenyat totes les eines necessàries per a executar els seus scripts amb API Gateway, Lithops, etc.

5.3 Disseny de la interfície gràfica

En primer lloc, s'ha dissenyat una capçalera per a la pàgina, que proporciona l'accés a les diferents funcionalitats. En aquesta, podem trobar el logotip d'RHermes, i un menú. Aquest, ofereix accés a la pàgina inicial, a l'explorador de resultats, una pàgina informativa d'un futur *startup*, la pàgina de contacte, i el botó per tancar la sessió. Cal dir que les *About* i *Contact* no s'han dissenyat ja que encara no s'ha començat a treballar en l'aspecte empresarial.



II·lustració 20. Captura de la capçalera de la pàgina web.

D'altra banda, si l'usuari per exemple treballa amb pantalla dividida, aquesta barrà es visualitzarà de la següent manera.



II·lustració 21. Visualització alternativa de la barra.



II·lustració 22. Barra de navegació desplegada.

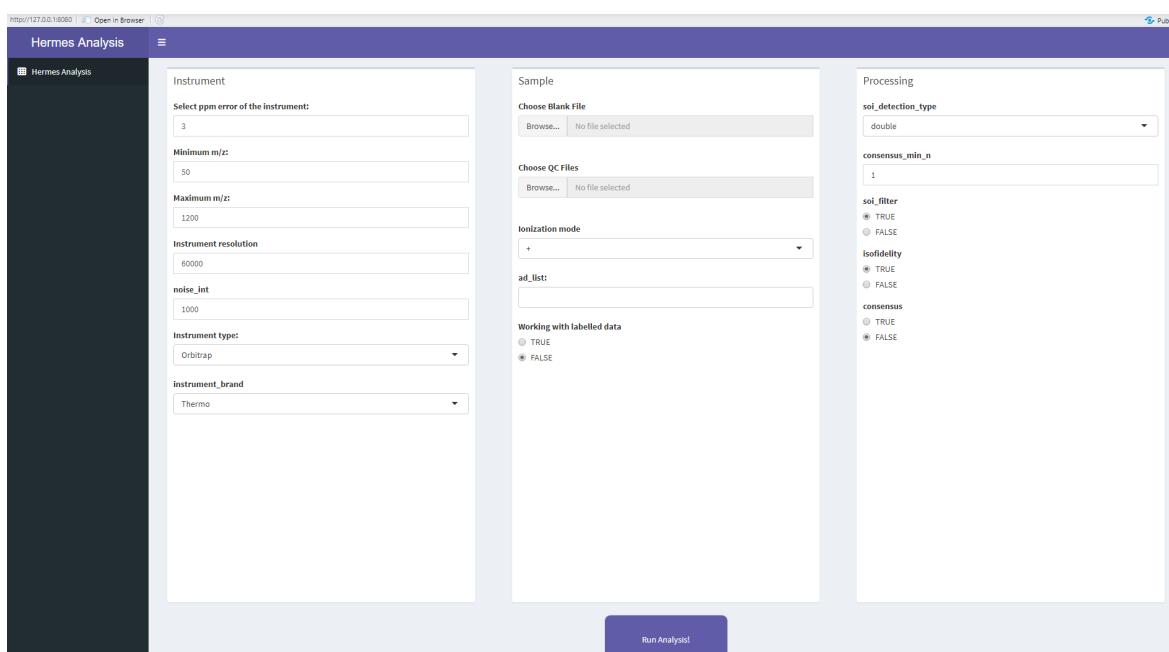
A continuació, es detallaran cadascuna de les funcionalitats implementades. Però, abans cal esmentar que quan es col·loca el ratolí sobre el botó *Sign Out*, aquest transforma les lletres de color vermell.

5.3.1 Configuració de la tasca

5.3.1.1 Prototip de la interfície gràfica en R i Shiny

Pel que fa a aquesta plana, des d'un bon inici sempre s'ha volgut un disseny senzill. D'aquesta manera, tant els usuaris avançats, com els que no, trobarien una interfície simple, guiada, i que requereixi una corba d'aprenentatge molt curta.

Com s'ha comentat, existeixen 4 mòduls d'Hermes, i cadascun ha de tenir la seva interfície. En un principi, es va acordar entre el principal creador d'RHermes, Roger Giné Bertomeu i jo, començar fent un prototip amb Shiny i R. De manera, que l'usuari disposi d'una sèrie de paràmetres que seleccionar.



II-Il·lustració 23. Captura de pantalla de la interfície gràfica creada amb Shiny. Autor: Usama Benabdelkrim Zakan

Per defecte, la web ja suggereix uns valors que l'usuari pot modificar. Després de fer-ho i seleccionar el fitxer de tipus "Blank" i "QC", es podrà executar la tasca. Un cop finalitzada, la pàgina mostrarà un missatge informant-ho a l'usuari.

5.3.1.2 Prototip final de la interfície gràfica

Com s'ha vist, fins ara només s'ha treballat amb interfícies fetes amb Shiny. Certament és un paquet per a crear interfícies molt senzilles. Com a conseqüència, s'obtenen webs simples. Però, aquest *framework* no és adequat per a crear webs avançades. Sobretot, pel que fa a la comunicació amb el *backend*. Un exemple, podria ser unificar un *frontend* amb l'SDK d'AWS, o crear algunes interaccions avançades com el *Drag & Drop* de fitxers. Per aquest motiu, a partir de l'acord fet entre Roger, el tutor d'aquest TFG (Pedro), i jo, es va procedir a posar en marxa la creació d'una web amb una tecnologia diferent.

Per la següent versió, es va optar per la combinació entre HTML, CSS i Javascript. Addicionalment, s'ha usat Bootstrap com a *framework* per a crear una interfície molt més moderna.

En primer lloc, s'ha volgut respectar la mateixa distribució vista a la *Il·lustració 23*. Partint de les 3 columnes de paràmetres (Instrument, Sample, Processing), s'han realitzat algunes modificacions. Deixant de banda alguns canvis en les terminologies suggerides pels companys d'RHermes, s'han eliminat els *inputs* dels fitxers. A més, s'ha afegit la opció d'escollir la base de dades de fórmules desitjada, en cas de no voler les que estan per defecte. Alternativament, l'usuari podrà escollir-ne una de local. També s'han modificat les parelles de botons (“true” i “false”) de tipus “radio”, per un únic botó molt més visual.

The screenshot shows a user interface for configuring parameters in the HermesAnalysis module, organized into three columns: Instrument, Sample, and Processing.

- Instrument Column:**
 - PPM error: Input field with value 3.
 - Minimum m/z: Input field with value 50.
 - Maximum m/z: Input field with value 1200.
 - Instrument resolution: Input field with value 60000.
 - Noise Threshold: Input field with value 1000.
 - Instrument type: Dropdown menu with 'Orbitrap' selected.
 - Instrument brand: Dropdown menu with 'Thermo' selected.
- Sample Column:**
 - Ionization Mode: Dropdown menu with 'Positive (+)' selected.
 - Select a formula database: Dropdown menu.
 - or submit your own:
 - Formula DB: Input field.
 - Browse: Button.
 - Adducts List: Dropdown menu.
 - Working with Labelled data: Radio button (unchecked).
- Processing Column:**
 - SOI Detection Type: Dropdown menu with 'Double' selected.
 - SOI Filter: Radio button (checked).
 - Isotopic fidelity filter: Radio button (checked).
 - Consensus: Radio button (unchecked).
 - Consensus Min. Samples: Input field with value 1.

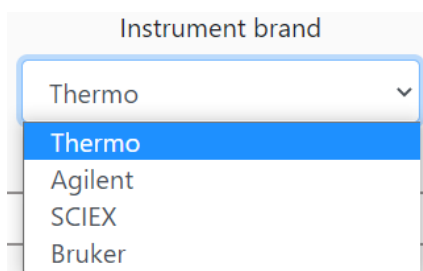
Il·lustració 24. Captura de pantalla dels paràmetres del mòdul HermesAnalysis.

Pel que fa a la interacció d'aquest camp, hi han diferents aspectes a comentar. El primer correspon a les entrades numèriques. Aquestes poden ser modificades mitjançant el teclat, per a esborrar i/o modificar el valor. De forma alternativa, també es proporcionen dos botons verticals per a fer la mateixa operació, però, amb el ratolí.

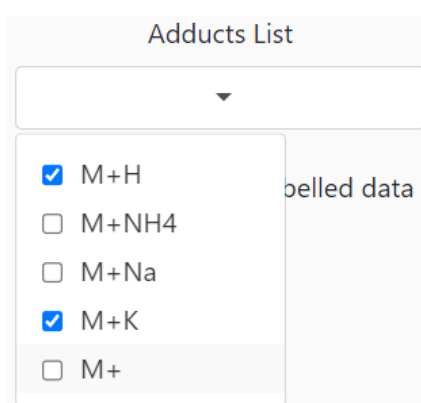
A close-up view of the 'PPM error' input field. The field contains the number '3'. To the right of the input field are two vertical arrows (up and down) for navigating through the values.

Il·lustració 25. Exemple d'una entrada numèrica.

D'altra banda, disposem de dos tipus de desplegable: els que admeten una única opció, i els que permeten seleccionar-ne de diferents. Pel que fa a aquests últims, ofereixen les opcions en conjunt a un *checkbox*, que mostra quins elements s'han seleccionat.

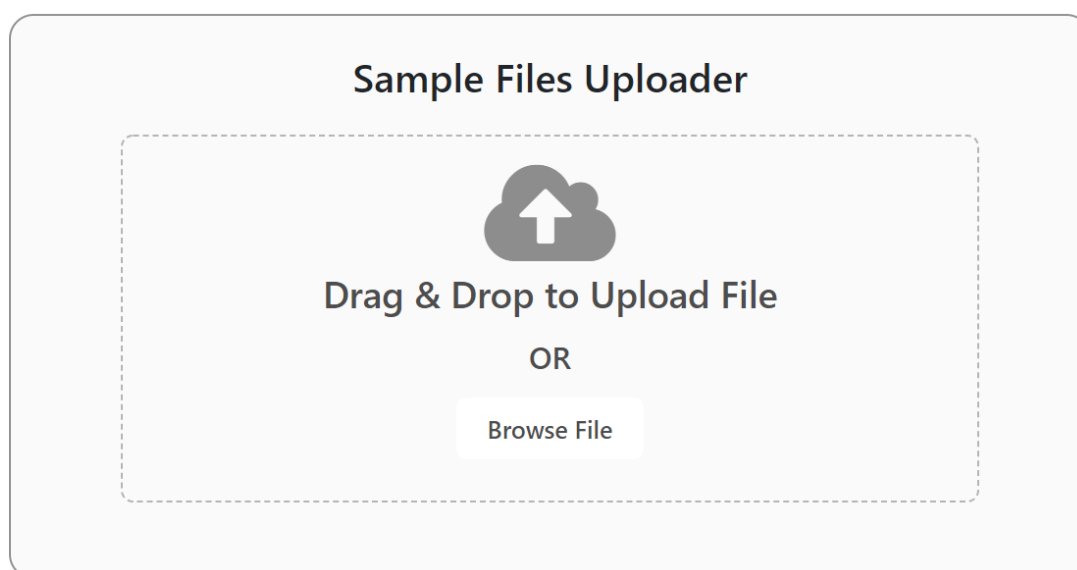


Il·lustració 26. Exemple de desplegable d'una única opció.

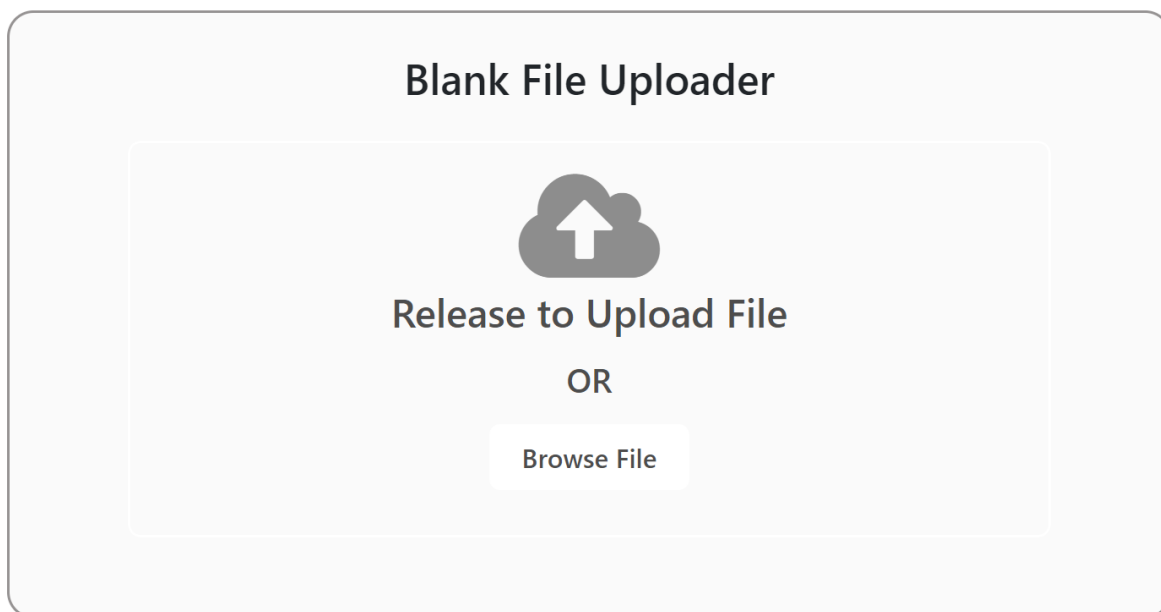


Il·lustració 27. Exemple de desplegable de múltiples opcions.

En segon lloc, les entrades del fitxers eliminades s'han desplaçat en un parell de zones on l'usuari pot adjuntar-los. En un principi disposa de dues formes: seleccionar-los amb l'explorador de fitxers del S.O, o arrossegar i deixar els fitxers (*Drag & Drop*). D'aquesta manera, l'usuari té dos opcions diferents fàcils d'utilitzar per pujar els fitxers. Des del punt de vista de la interacció amb l'usuari, el *Drag and Drop* pot resultar molt còmode per aquests.

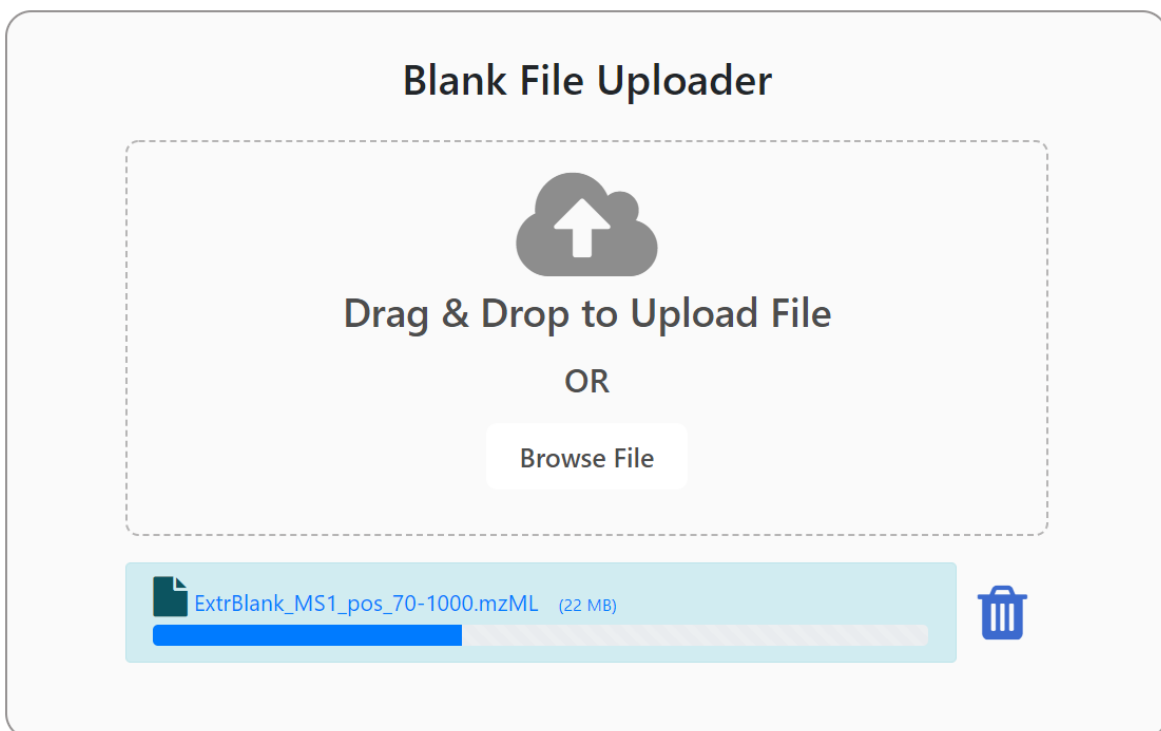


Il·lustració 28. Captura de la zona d'entrada de fitxers de tipus "Sample".



Il·lustració 29. Captura de l'entrada de fitxers mentre s'hi arrossega un.

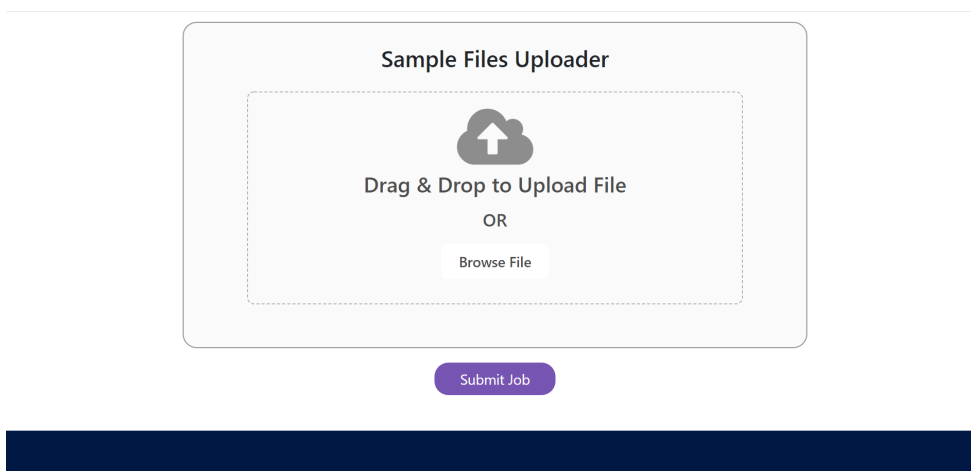
Quan un usuari arrossega un fitxer, el sistema substitueix la línia discontinua, per una de continua i blanca. A més, reemplaça el text “Drag & Drop to Upload File” per “Release to Upload File”, per a indicar que la pàgina està disposada a rebre les entrades. Un cop es deixa anar el fitxer, torna al seu estat inicial. D'aquesta manera, l'usuari rep *feedback* per a una millor experiència d'usuari.



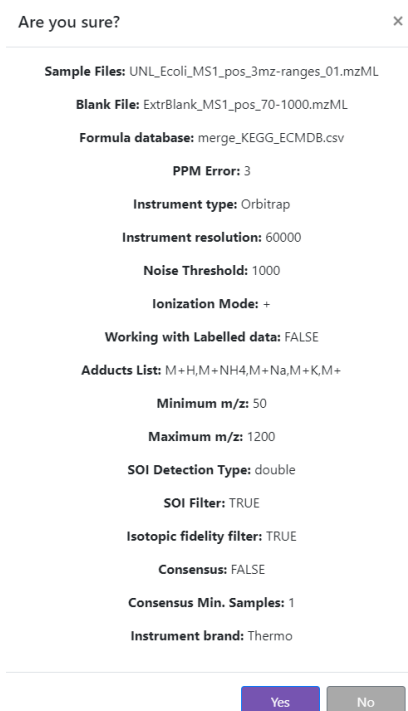
Il·lustració 30. Captura de la zona d'entrada de fitxers de tipus "Blank". Es pot apreciar com es puja el fitxer, i la informació d'aquest.

Tal i com es pot apreciar, un cop adjuntat el fitxer, el sistema el pujarà a la carpeta personal de l'usuari d'S3. Aquest pas, es durà a terme justament després d'arrossegar cada element. Per informar d'aquest procés, es desplegarà un camp on es mostra el procés de la pujada, el nom, i la mida del fitxer. A més, es dóna la opció a l'usuari d'eliminar aquest fitxer de l'experiment. Quan es col·loca el ratolí sobre la icona per a eliminar-lo, es produeix un moviment ràpid oscil·latori sobre aquest. Un cop premut, s'elimina el fitxer satisfactòriament.

Per últim, per a enviar la tasca es proporciona un botó al final de la pàgina, el qual quan es prem, es demana la confirmació dels paràmetres.



Il·lustració 31. Captura del final de pàgina s'aprecia el botó que envia la tasca al "backend".



Il·lustració 32. Sol·licitud de confirmació dels paràmetres per part de l'usuari.

Posteriorment, l'usuari haurà d'indicar un nom assignat a l'experiment.

Il·lustració 33. Camp per a indicar el nom de l'experiment.

Il·lustració 34. "Pop-up" informatiu.

Per acabar, la pàgina mostrarà un “pop-up” per a indicar de l'estat de l'execució, i del temps aproximat d'espera. Com s'ha explicat anteriorment, aquest temps no representa una prioritat pel projecte en aquests moments.

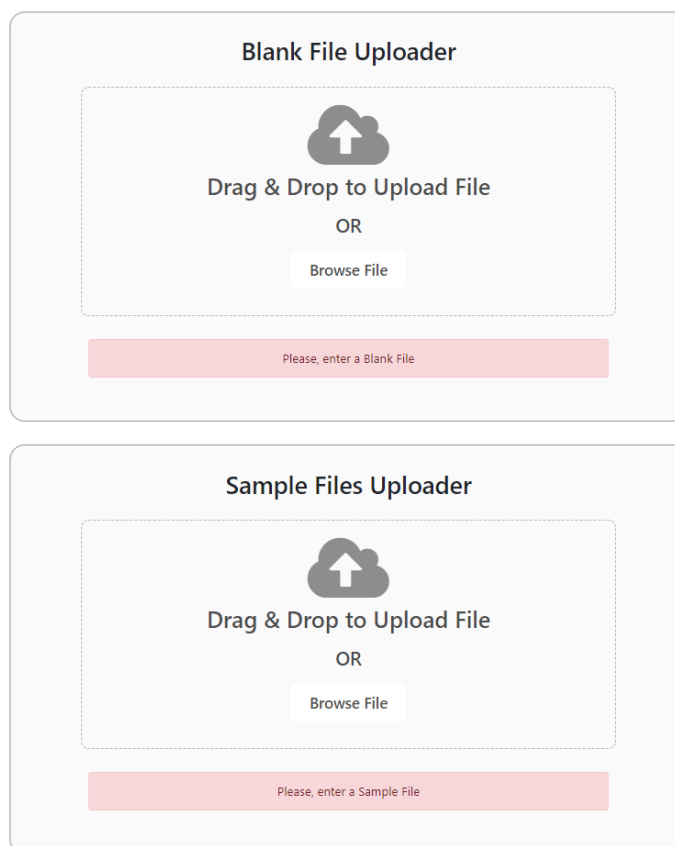
5.3.1.2.1 Control d'excepcions

Com és evident, els usuaris poden cometre certs errors, que puguin provocar el mal funcionament de l'aplicació. Per això, s'han dissenyat diferents controls d'excepcions per a evitar possibles anomalies.

En primer lloc, si l'usuari introdueix els fitxers i no selecciona cap element de la “Adducts list”, el sistema ho informarà al enviar la tasca.

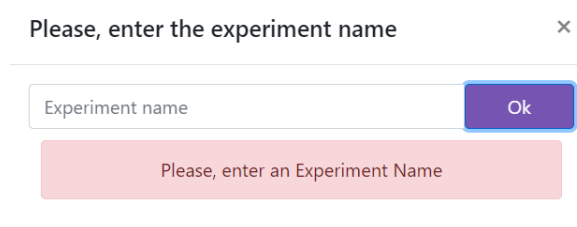
Il·lustració 35. Captura de l'error al no seleccionar cap element de la “Adducts list”.

En segon lloc, si l'usuari no introdueix cap fitxer “Blank” i/o “QC”, es mostrarà el següent missatge d'error.



Il·lustració 36. Captura dels errors al no adjuntar cap fitxer.

Adicionalment, de la manera que s'ha dissenyat l'arquitectura, és necessari assignar un nom per a cada experiment. Per tant, en cas de no fer-ho el sistema llançarà una excepció.



Il·lustració 37. Captura de l'error al no indicar el nom de l'experiment.

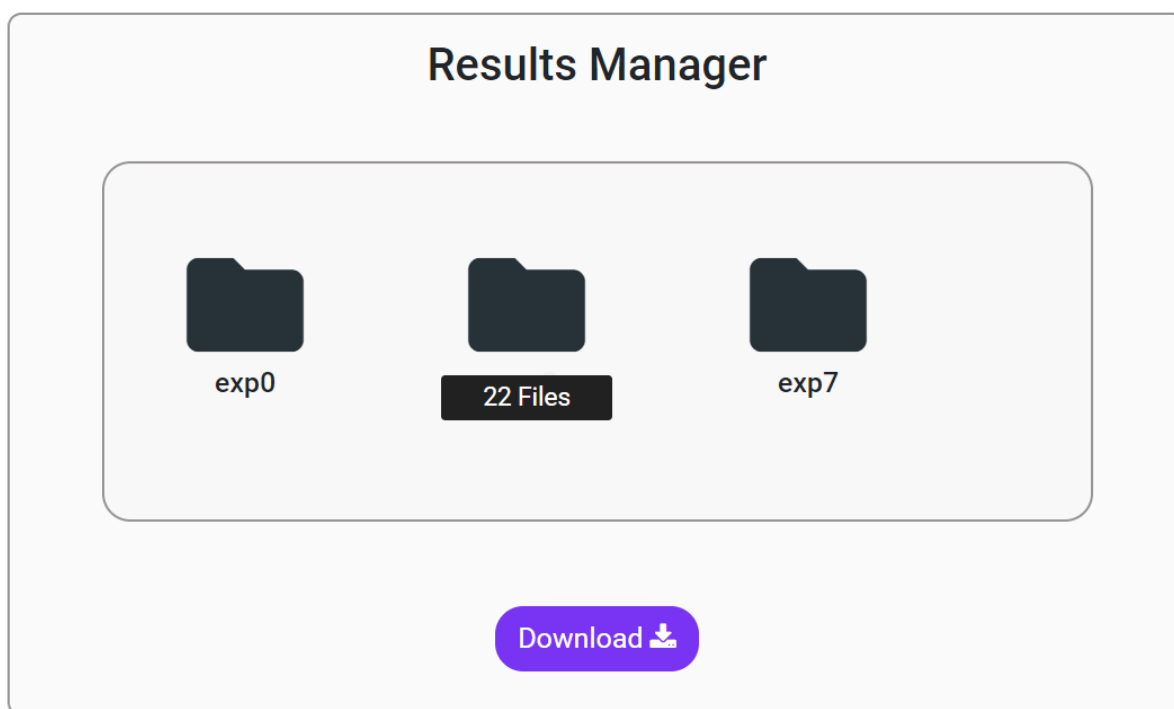
Per últim, si algun usuari intenta accedir aquesta plana sense estar autenticat, abans de carregar aquesta pàgina, el sistema el redirigirà a l'inici de sessió. Però, aquesta no és la solució adequada per a evitar que s'utilitzin els nostres serveis de manera fraudulenta. Més endavant, a l'apartat 5.4.2.5 *Seguretat d'API Gateway*, veurem com s'evita aquest perill.

5.3.2 Explorador de fitxers

Pel que fa a l'explorador de fitxers, permet als usuaris consultar els resultats de tots els seus experiments.

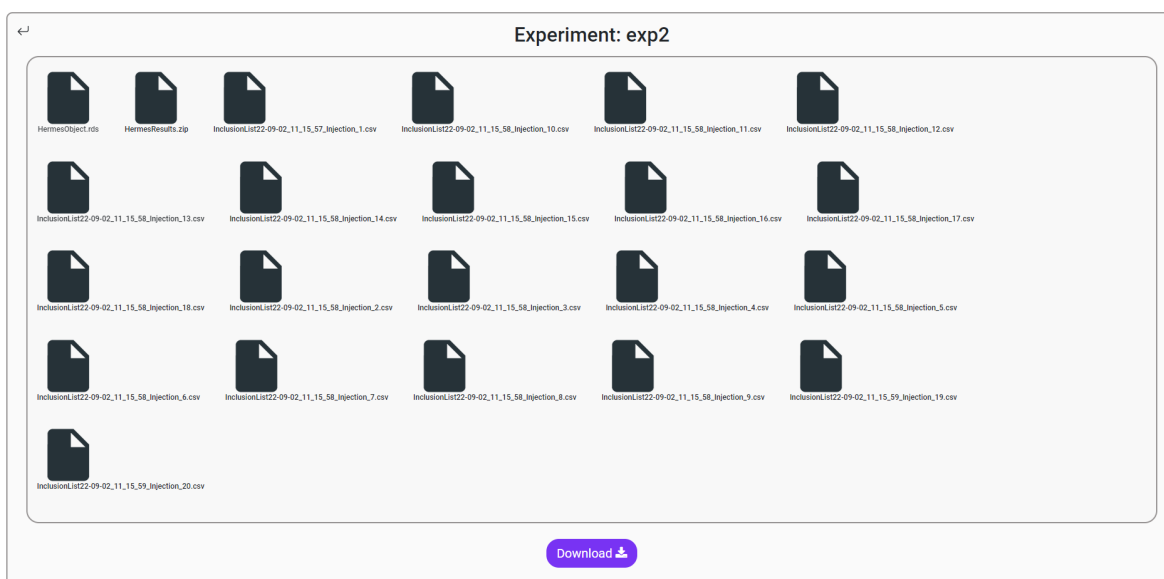
En primer lloc, un cop obert l'explorador, es mostra/en una o múltiples carpetes, en funció de cada usuari. Cada carpeta correspon als resultats d'un experiment. Si l'usuari col·loca el cursor sobre una de les carpetes, es mostra el nombre de fitxers que conté. Tanmateix, si prem sobre aquesta, es mostraran els múltiples fitxers de la sortida. La immensa majoria d'ells es tracten de fitxers en format .csv. A més, trobarem un fitxer en format “.Rds” i el fitxer “.zip” que comprimeix els anteriors.

En cas de que l'usuari hagi accedit a una carpeta i vulgui tornar enrere, la interfície disposa d'un botó dedicat a aquesta funcionalitat.



Il·lustració 38. Captura d'un exemple de les carpetes dels experiments d'un usuari

En segon lloc, existeixen diferents opcions per a descarregar els resultats. En la plana en la qual es mostren les carpetes, es disposa d'un botó per a descarregar els resultats (fitxers .zip) de tots els experiments. En cas d'accedir a una de les carpetes, l'usuari també disposa d'un botó dedicat a la descàrrega d'aquest únic directori. Per acabar, des de dins d'una carpeta, si l'usuari vol descarregar un únic fitxer, ho pot fer fent “click” sobre el que desitja.



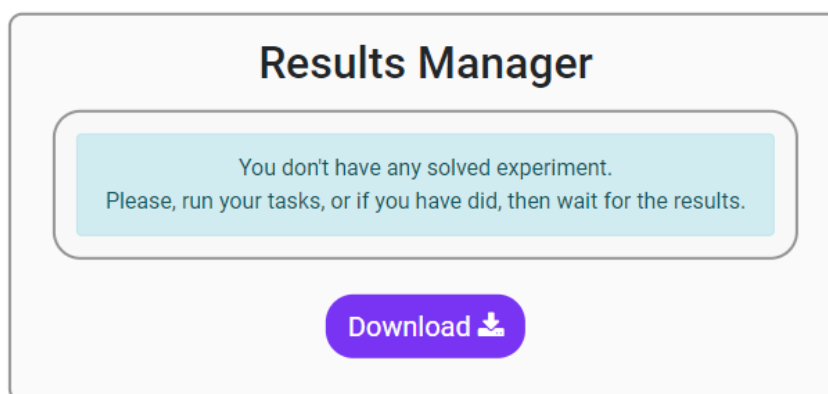
Il·lustració 39. Captura dels resultats d'un experiment. En aquest cas s'ha modificat el zoom del navegador per a una vista completa.

Finalment, per a qualsevol de les anteriors descàrregues, s'obrirà l'explorador de fitxers local per a escollir la ubicació d'aquest. En cas de voler descarregar tots els experiments, aquest pas es repetirà tantes vegades com experiments disposi l'usuari.

5.3.2.1 Control d'excepcions

En aquest cas, també es verifica que l'usuari estigui autenticat. En cas contrari, es redirigeix la web a l'inici de sessió.

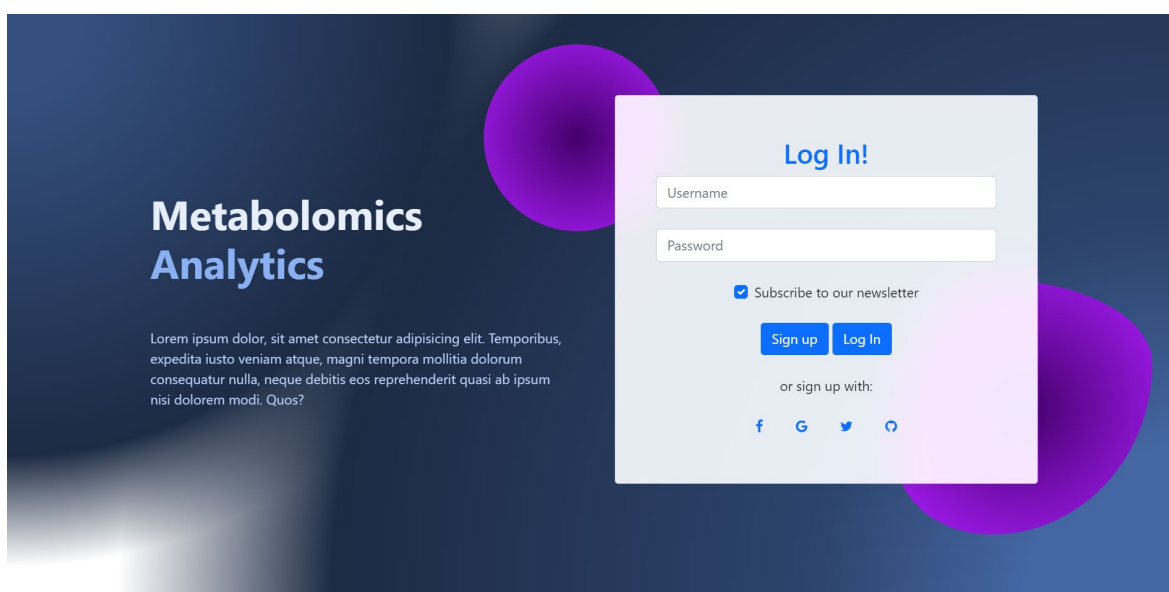
A més, es controla que l'usuari tingui algun experiment fet. Si no es així, es mostra un missatge informatiu.



Il·lustració 40. Captura de l'explorador de fitxers en cas de no tenir cap anàlisi finalitzat.

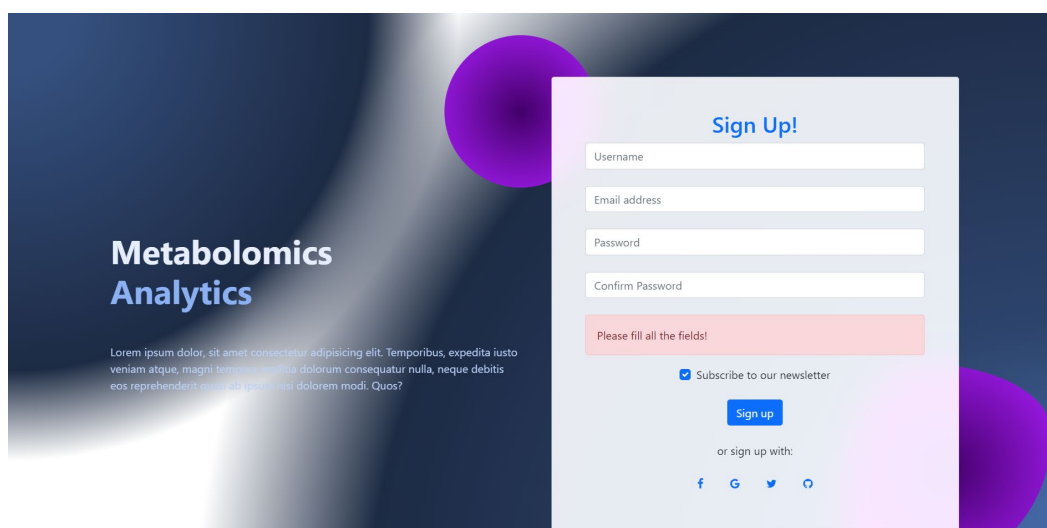
5.3.3 Autenticació

La interfície de l'autenticació ofereix en un primer lloc un formulari d'inici de sessió. Es demana el nom d'usuari i la contrasenya. Si l'usuari prem el botó *Log In*, el sistema s'encarregarà d'autenticar-lo. En el cas de que les credencials proporcionades siguin vàlides, es mostrarà un missatge d'èxit, i es redirigirà a la plana de la configuració de la tasca. En cas contrari, s'informarà amb un missatge d'error.



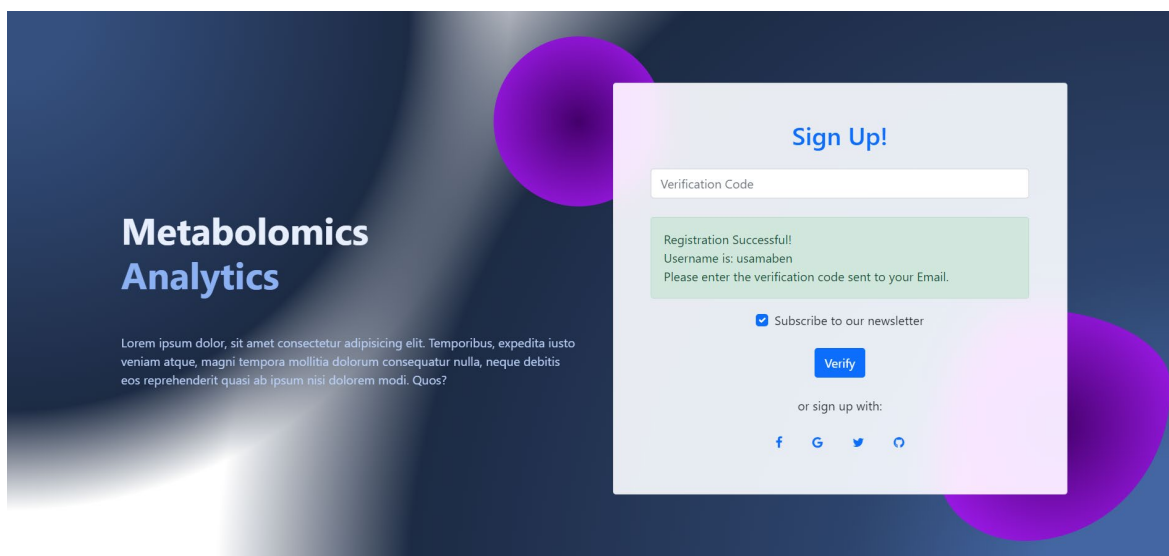
Il·lustració 41. Pàgina d'inici de sessió per a accedir a l'aplicació web.

Altrament, si l'usuari prem el botó *Sign Up*, la pàgina substituirà el formulari d'inici de sessió per el de registre. En aquest es troben diferents camps: nom d'usuari, correu electrònic, contrasenya i la confirmació de la contrasenya. En cas d'introduir algun camp que no compleixi les restriccions corresponents, es mostrarà un missatge d'error. En cas contrari, es mostraria un missatge amb les dades anteriorment demanades (exceptuant la contrasenya); i un camp on l'usuari ha d'introduir el codi de verificació enviat al seu correu electrònic.



Il·lustració 42 Pàgina de registre al sistema.

Per acabar, un cop l'usuari introdueixi el codi de verificació correctament, es tornarà a mostrar el formulari d'inici de sessió.



Il·lustració 43. Formulari de verificació del compte.

5.3.3.1 Control d' excepcions

En primer lloc, ens podem trobar amb el cas on l'usuari introdueixi només el nom d'usuari o la contrasenya. En aquest cas, es mostraria el següent missatge.

Il·lustració 44. Missatge d'error en cas de no omplir els camps mostrats.

En segon lloc, l'usuari pot proporcionar els dos camps però poden ser incorrectes.

Il·lustració 45. Missatge d'error en cas d'introduir un nom d'usuari o contrasenya erroni.

Pel que fa al registre, si l'usuari deixa camps sense emplenar, es mostrarà un altre missatge d'error.

The image shows a 'Sign Up!' form with four input fields: 'usamaben', 'Email address', 'Password', and 'Confirm Password'. Below the fields is a red error message box that says 'Please fill all the fields!'.

Il·lustració 46. Missatge d'error en cas de no omplir tots els camps.

En cas de que el correu no compleixi el format adequat, s'informarà a l'usuari de l'error.

The image shows the same 'Sign Up!' form, but with the 'Email address' field filled with 'usama.benabdelkrim@'. The password fields are filled with dots. A red error message box at the bottom says 'Invalid email address format.'.

Il·lustració 47. Missatge d'error en cas d'introduir un correu no vàlid.

Adicionalment, la contrasenya disposa de diferents restriccions: es demana una longitud major a 6 caràcters, i que contingui majúscules, caràcters especials, i nombres. A més, les dues contrasenyes han de coincidir.

Expected string length >= 6, but found 3 for params.Password

Il·lustració 48. Missatge d'error per a contrasenyes de menys de 6 caràcters.

Password did not conform with policy: Password must have uppercase characters

Il·lustració 49. Missatge d'error per a contrasenyes sense majúscules.

Password did not conform with policy: Password must have numeric characters

Il·lustració 50. Missatge d'error per a contrasenyes sense nombres.

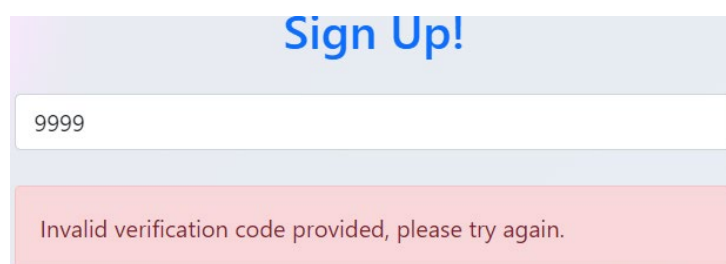
Password did not conform with policy: Password must have symbol characters

Il·lustració 51. Missatge d'error per a contrasenyes sense símbols.

Confirm password failed!

Il·lustració 52. Missatge d'error per a inconsistències en la confirmació de la contrasenya.

Per últim, si s'ha fet el registre satisfactòriament, el sistema demanarà el codi de verificació enviat. Si el que s'introdueix és erroni, el portal informarà a l'usuari de l'excepció.



The image shows a web form titled "Sign Up!". It has a text input field containing "9999". Below the input field is a red error message box that reads: "Invalid verification code provided, please try again."

Il·lustració 53. Captura del portal al introduir un codi de verificació erroni.

6 Implementació

En primer lloc, estudiarem els principals serveis d'AWS que s'han utilitzat per crear l'aplicació web que s'ha explicat. Posteriorment, esmentarem els llenguatges de programació utilitzats, *frameworks*, entre d'altres.

Per últim, es detallaran els aspectes més rellevants de la implementació, com per exemple, com s'ha utilitzat Lithops al projecte, o codi a destacar. A més, es dedicarà un apartat pel que fa a la optimització que s'ha fet d'RHermes.

6.1.1 Tecnologies utilitzades (Background)

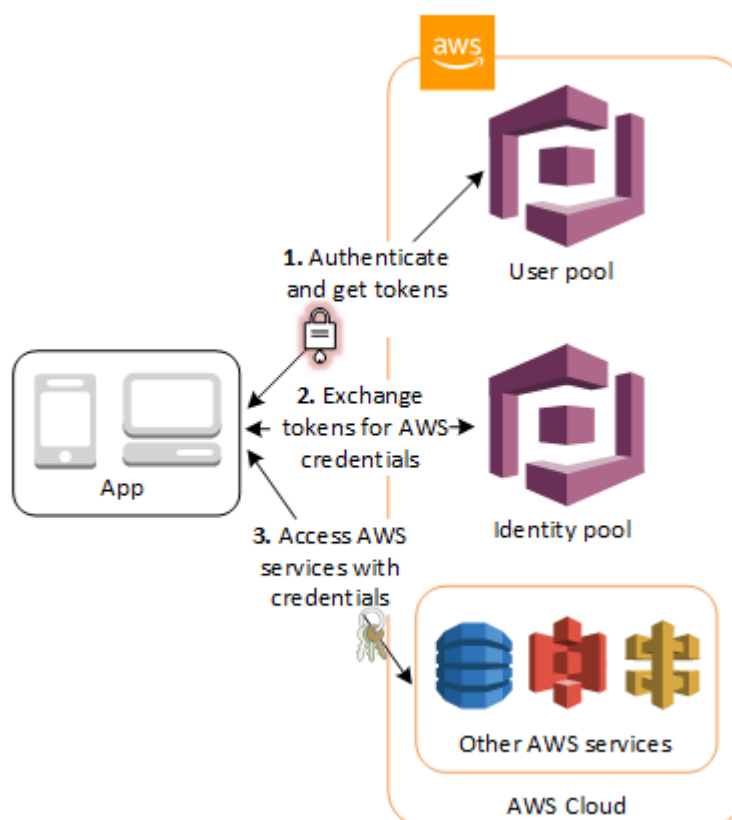
6.1.1.1 Amazon Web Services

6.1.1.1.1 AWS Cognito

Fins ara, s'ha explicat com funciona de cara a l'usuari l'inici de sessió i el registre. Però, pel darrere cal tenir un sistema que gestioni els nostres clients. En aquest cas, s'ha utilitzat AWS Cognito.

Cognito es tracta d'un servei que ofereix el nostre proveïdor del núvol per autenticar, autoritzar i administrar usuaris. Els grups d'usuaris i grups d'identitats permeten gestionar aquests. Pel que fa al primer, proporciona la capacitat d'administrar opcions d'inici de sessió i registre als usuaris, obtenint així un directori d'usuaris amb accés a la nostra *app*. A diferència d'aquest, el grup d'identitats administrà els permisos dels usuaris a d'altres serveis d'AWS, com per exemple S3.

A continuació, estudiarem un exemple que proposa la pàgina oficial d'AWS: autenticar un usuari i concedir-li accés als serveis d'AWS.



Il·lustració 54. Diagrama de funcionament d'AWS Cognito.

Font: https://docs.aws.amazon.com/es_es/cognito/latest/developerguide/what-is-amazon-cognito.html

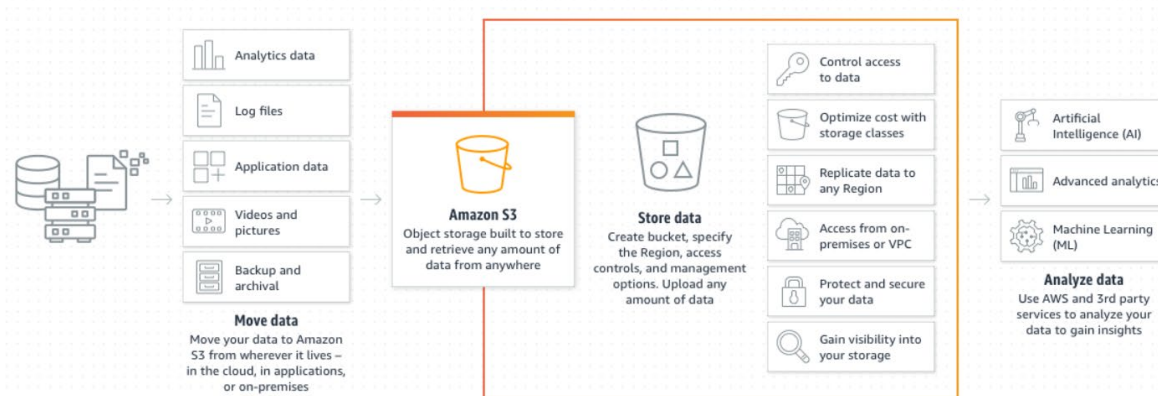
En primer lloc, l'usuari introduirà els camps requerits per a iniciar sessió (per exemple, el nom d'usuari i contrasenya). Si el grup d'usuaris valida aquest intent d'inici de sessió, l'usuari rebrà una sèrie de *tokens*. Posteriorment, l'aplicació obtindrà les credencials

dels clients intercanviant aquests *tokens*, a través del grup d'identitats. Per acabar, l'usuari podrà accedir a serveis com S3, DynamoDB, entre d'altres.

En conclusió, Cognito compta amb diverses avantatges de cara a la nostra aplicació. El grup d'usuaris ens permet fer l'inici de sessió, i el grup d'identitats permet accedir a serveis d'AWS. Però, també ens permet autenticar-nos mitjançant xarxes socials com Facebook, Google, entre d'altres mètodes.

6.1.1.1.2 AWS S3

AWS S3 és un servei d'emmagatzemament d'objectes. Aquest, ofereix escalabilitat, disponibilitat, seguretat, un alt rendiment la possibilitat de guardar-hi qualsevol volum de dades. A més, la seva arquitectura permet optimitzar els costos, i configurar les polítiques d'accés i seguretat de les dades.



Il·lustració 55. Esquema d'AWS de les funcionalitats d'S3 i alguns casos d'ús.

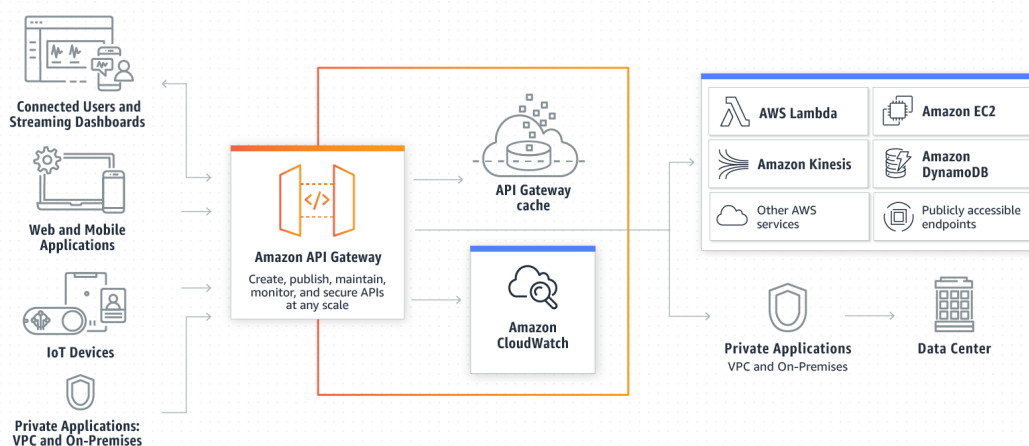
En concret, resulta ideal per a gairebé qualsevol cas d'ús. Alguns exemples poden ser aplicacions d'anàlisi de Big Data, IA, o "Machine Learning".

Per a aquest projecte, S3 s'ha utilitzat tant pels fitxers d'entrada als experiments, com pels seus resultats. A més, Lithops també ha usat aquest servei per a guardar metadades de les tasques encarregades a Batch. Per últim, s'ha utilitzat com *hosting* per al *frontend*.

6.1.1.1.3 API Gateway

AWS API Gateway es tracta d'un servei desenvolupat per a crear APIs, que permeten accedir a les dades o funcions del *backend* d'una aplicació.

AWS ofereix dos tipus d'APIs: les API RESTful i API WebSocket. En aquesta arquitectura, s'ha utilitzat la primera, que permet treballar amb arquitectures serverless i *backends* de tipus HTTP.



II·lustració 56. Diagrama del funcionament d'AWS API Gateway. Font: <https://aws.amazon.com/es/api-gateway/>

En particular, API REST proporciona un conjunt de recursos i mètodes que són integrats amb funcions Lambda, DynamoDB o d'altres serveis d'AWS. Es basa en un model petició/resposta, per tant, els usuaris enviaran una petició que serà resolta de manera síncrona. En el nostre cas, l'API està integrada en conjunt amb AWS Lambda, que d'una manera molt ràpida, podrem posar en marxa el nostre *backend*.

Pel que fa a les avantatges, API Gateway proporciona una latència molt reduïda per a les peticions i respostes. A més, permet controlar el tràfic, evitant així deteriorar el rendiment en els pics més alts. Addicionalment, permet gestionar la seguretat mitjançant IAM i Cognito. També es cuida de monitorar diferents dades i mètriques (mitjançant AWS CloudWatch), i proporciona preus molt competitius en funció del seu ús.

6.1.1.1.4 AWS Lambda

AWS Lambda es tracta d'un servei d'Amazon que permet la creació d'arquitectures serverless, basant-se en esdeveniments. Aquests, desencadenen l'execució del codi prèviament programat d'una manera molt ràpida. Aquestes funcions poden ser cridades des de multitud de serveis o aplicacions. Per tant, la seva combinació amb d'altres components, poden formar aplicacions on es processin dades a gran escala o executi *backends* interactius.

En aquest cas, s'ha utilitzat per a la creació d'una aplicació basada en esdeveniments. A més, tot i que el seu cost és elevat en comparació a d'altres opcions, té un potencial enorme que el converteixen en un servei molt atractiu. Cal dir que només es paga pels recursos utilitzats.



II·lustració 57. Esquema d'AWS per a mostrar un cas d'ús de Lambda en aplicacions web.

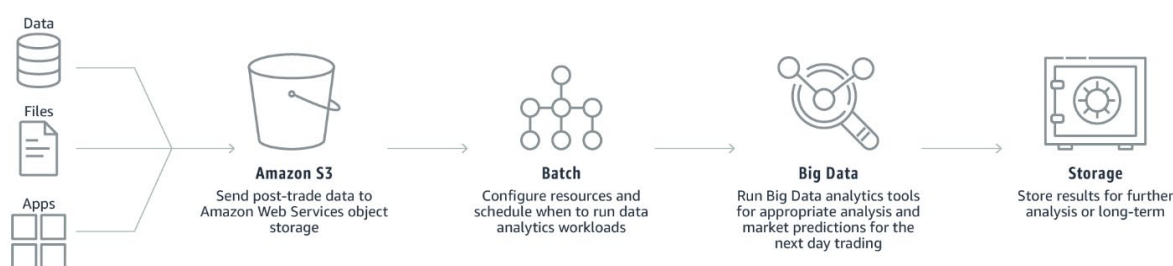
Pel que fa a les avantatges, es proporciona una escalabilitat òptima i automàtica, sense haver d'administrar servidors. També cal remarcar que proporciona un rendiment estable, que amb el hardware que ofereix es pot executar codi en qüestió de milisegons. A més, no s'afegeix cap cost addicional per la simple execució del codi. Però, depenent del cas, el seu alt cost pot convertir-se en motiu de no utilitzar aquest servei.

Finalment, en el nostre cas l'ús de és molt apropiat ja que ens permet evitar tenir servidors engegats tot el dia. A més, el cost econòmic és molt reduït ja que el temps d'ús ronda pels 4 segons, aproximadament.

6.1.1.1.5 AWS Batch

AWS Batch és un servei d'AWS que permet l'execució de milers de tasques sobre la seva infraestructura. S'ajusta als recursos requerits, i optimitza les instàncies per al tipus de recurs desitjat. A més, s'encarrega de gestionar i planificar les tasques, de manera que la resta de la feina es redueix a la lògica del codi que es vol implementar.

Com en el cas de Lambda, Amazon no afegeix cap cost per l'ús d'aquest servei. Per tant, el que realment es paga és per l'ús dels recursos.



Il·lustració 58. Esquema d'AWS que mostra el cicle de vida de Batch.

En definitiva, Batch és una eina potent pel que fa a la gestió i execució de tasques programades. Com es veu a la imatge superior, el servei permet rebre dades d'S3, planifica les feines, les executa, i finalment guarda els resultats. A grans trets, no deixa de ser una abstracció del nostre cas d'ús.

6.1.1.2 Llenguatges de programació

6.1.1.2.1 HTML i CSS

En primer lloc, tot i que HTML i CSS no són considerats llenguatges de programació, s'ha volgut incloure en aquest apartat per a simplicitat de la documentació.

Pel que fa a HTML, està dissenyat per a crear l'estructura o l'esquelet dels textos. La seva clau està en la capacitat de relacionar aquests, mitjançant "hyperlinks". Degut a la seva estandardització en la indústria dels navegadors, resulta una eina ideal per a crear webs com la que s'ha creat en aquest treball.

D'altra banda, CSS es tracta d'un llenguatge de fulls d'estil que complementa els documents de marques. En concret, es cuida de la semàntica d'aquests elements, de manera que permet estilitzar-los al nostre gust.

En conclusió, es tracten de tecnologies molt conegudes i utilitzades. A més, la possibilitat de combinar-les amb Bootstrap facilita el desenvolupament de webs modernitzades.

6.1.1.2.2 *Javascript*

JavaScript és un llenguatge de programació interpretat que permet treballar sobre HTML, per a crear interactivitat en la web. Permet crear gairebé tot: des de respostes al prémer un botó, fins a gràfics 3D.

Bona part del seu èxit, es deu al desenvolupament d'un munt de paquets que permeten implementar funcionalitats molt útils. Alguns exemples poden ser algunes APIs que permeten generar dinàmicament text en HTML i estilitzar-lo en CSS. També permet importar funcionalitats d'APIs de tercers com Facebook.

En definitiva, JavaScript és ideal per complementar el codi HTML i CSS. Permet afegir funcionalitats realment complexes.

Per acabar, en el nostre cas s'ha utilitzat per proporcionar interactivitat a la web. A més, AWS proporciona l'SDK en Javascript, i per tant s'ha utilitzat complementant-ho amb funcions de JavaScript que actuen d'interfície amb l'API d'AWS. Sobretot ha estat essencial per treballar amb AWS Cognito i AWS S3.

6.1.1.2.3 *R*

R és un llenguatge de programació i un entorn de desenvolupament de programari enfocat en l'estadística i l'anàlisi de dades. També permet la creació de gràfics basats en aquestes dades. A més, suporta la complementació de multitud de paquets com el que s'ha s'explicarà més endavant, Shiny.

6.1.1.2.4 *Python*

Python és un dels llenguatges més coneguts del món. Es tracta d'un llenguatge d'alt nivell que suporta la programació estructurada, orientada a objectes i funcional. Compta amb infinitat de llibreries, que fan de Python un llenguatge ideal per *machine learning*, o anàlisi de dades. A més, el fet de permetre crear codi molt llegible, el fan atractiu pels desenvolupadors.

6.1.1.3 Frameworks

6.1.1.3.1 *Shiny*

Shiny es tracta d'un paquet d'R que facilita la creació d'aplicacions web interactives. Proporciona interactivitat i permet integrar-hi codi Javascript, CSS i htmlwidgets. És un "framework" bastant útil i senzill per a prototips o aplicacions web d'una certa complexitat.

En realitat, resulta ideal per a la visualització de dades, gràfiques i estadístiques. Per tant, es converteix un bon aliat d'R. Però, a partir de cert punt el seu rendiment comença a ser un coll d'ampolla.

En definitiva, resulta un bon paquet per a afegir una interfície gràfica per a que interactiu amb R. Ara bé, pel nostre cas no acaba de ser el més ideal degut a la pròpia

arquitectura de l'aplicació. A més, limita molt certs aspectes ja que està dissenyat per utilitzar-lo sense tenir coneixements de desenvolupament web.

6.1.1.3.2 *Bootstrap*

Bootstrap és un *framework* de codi obert fet en HTML, Javascript i CSS, que permet dissenyar aplicacions responsives. Mitjançant la seva sintaxi, permet crear planes modernes sense dedicar massa temps en fer-ho des de zero. Aquestes, s'adapten a la mida de la pantalla automàticament. D'aquesta manera, es garanteix el correcte funcionament en diferents dispositius.

6.1.1.3.3 *Lithops*

Lithops és una eina informàtica que fou desenvolupada per un grup d'investigació de la URV per al projecte CloudButton. Cal remarcar que el coordinador del grup és Pedro García López, tutor d'aquest TFG.

Es tracta d'un *framework* de computació distribuïda *multicloud* en Python. En altres paraules, és un paquet que ens permet executar codi de manera distribuïda en diverses plataformes al núvol. En aquest cas, s'admet IBM Cloud, AWS, Azure, Google Cloud, entre d'altres. Addicionalment, facilita la gestió de l'emmagatzemament orientat a objectes com AWS S3 o IBM Cloud Object Storage.

Pel que fa al seu ús, es pot aplicar a aplicacions que treballin amb Big Data, i tasques paral·leles exigents. En concret, és preferent usar-lo en processos que no requereixin de comunicació. Exemples d'ús més concrets poden ser *deep learning*, *machine learning*, computació en metabolòmica, o anàlisi geoespacial.

En el nostre cas, Lithops actua d'interfície per a encuar tasques metabolòmiques al backend (AWS Batch). Per últim, es recomana consultar la referència [7] per a més informació.

6.1.1.4 Docker

Docker és una plataforma que permet instal·lar software en contenidors que contenen tot el necessari per a una execució exitosa. Això implica que un contenidor ha de tenir un sistema operatiu, llibreries i el codi de l'aplicació, com a exemples. Docker resulta ideal per a aplicacions distribuïdes.

Una forma de construir imatges, és mitjançant el que s'anomena Dockerfile. Un Dockerfile no és res més que un fitxer que disposa de les comandes necessàries per a la personalització de la imatge. En concret, s'ha d'indicar la imatge base sobre la que es vol treballar (una simple imatge base d'un sistema operatiu, o imatges que a més disposen de cert programari). Posteriorment, s'indicaran totes les llibreries que es volen instal·lar, i comandes que es vulguin executar cada cop que s'inicia el contenidor. En realitat, es poden afegir més tipus de comandes.

Per acabar, la plataforma disposa de certes comandes per a la gestió de contenidors. Uns exemples poden ser les comandes per a construir la imatge, i iniciar, aturar o eliminar un contenidor. En aquest cas, Lithops és qui construeix i publica la imatge a AWS per a ser utilitzada per Batch.

6.1.2 Aspectes rellevants de la implementació

6.1.2.1 Execució d'Hermes sobre AWS Batch

Com s'ha esmentat anteriorment, el mòdul d'HermesAnalysis es tracta d'un script fet en RMarkdown que rep una sèrie de paràmetres. També s'ha explicat que Lithops és qui permet l'execució d'aquest codi a Batch, mitjançant la funció “call_async()”. D'una banda, Lithops treballa amb Python. De l'altra, HermesAnalysis ho fa amb R o Rmd. Per tant, es presenta la necessitat de buscar la forma d'executar codi Rmd a través de Python.

La primera solució proposada, fou la llibreria de Python “rpy2”. Aquest paquet es tracta d'una interfície que permet executar codi en R, a través de Python. En concret, els objectes d'R es converteixen en classes de Python. Aquestes, contenen certes funcions que es corresponen amb les mètodes que treballen amb aquests objectes.

En un principi es van fer algunes proves, però, es van experimentar deficiències en el rendiment. En particular, els bucles d'R es converteixen en un veritable coll d'ampolla. En altres paraules, en les funcions més exigents d'RHermes, com per exemple “findSOI()”, l'execució ha tardat fins a tal punt que mai he sigut capaç de superar l'execució d'aquesta. A més, en alguns fòrums d'Internet es poden trobar experiències d'aquest estil, on l'execució del codi es multiplica al voltant de 5 vegades.

Al intentar buscar solucions a aquest problema, es va pensar en una forma molt eficient i simple d'executar aquest codi. Consisteix en el següent.

A la imatge Docker que s'executa sobre Batch, se li pot copiar l'script HermesAnalysis.Rmd. Per a executar-lo, no ha calgut cap esforç més. El motiu és que anteriorment es va dissenyar una imatge que executa RHermes sobre un servidor EC2. Com ja es comptava amb aquest Dockerfile, només ha calgut afegir-hi l'script Rmd.

Per tant, ja es disposa de la infraestructura necessària per a córrer el codi d'Hermes.

Pel que fa a l'execució, s'ha desenvolupat un codi que és el que s'executa dins de la funció “call_async()” de Lithops. Aquest, consisteix en crear a Batch un script R que contingui la crida al fitxer HermesAnalysis.Rmd. A més, se li passaran els paràmetres escollits per l'usuari. Un cop creat l'script d'R, es procedeix a executar-lo, de tal manera que s'executarà HermesAnalysis.Rmd.

A continuació s'adjunta el codi per a més detalls:

```
def render_analysis(args):
```

```

with open("script.R", "w") as f:          # Crea l'script d'R

    f.write('''markdown::render(input = 'Hermes_Analysis.Rmd', params
= list(qc_files = c('' + str(args[0]).replace(',','\','\') +
'''),blank_file = '' + args[1] + '',met_db = '' + args[2] +
'',ppm_error = '' + args[3] + '',instrument = '' + args[4] +
'',resolution = '' + args[5] + '',noise_int = '' + args[6] +
'',ionization = '' + args[7] + '',labelled = '' + args[8] +
'',ad_list = c('' + args[9].replace(',','\','\') + ''),min_max_mz =
c('' + args[10] + '', '' + args[11] + ''), soi_detection_type = '' +
args[12] + '',soi_filter = '' + args[13] + '',isofidelity = '' +
args[14] + '',consensus = '' + args[15] + '',consensus_min_n = '' +
args[16] + '',instrument_brand = '' + args[17] + '',aws_output_dir =
'' + args[18] + '')''')          # Escriu la crida al script Rmd amb
tots els paràmetres rebuts.

    subprocess.run(["Rscript", "script.R"])          # Executa l'script d'R
    subprocess.run(["python3", "/lithops/zip.py", args[18]]) # Codi que
comprimeix els resultats. S'estudiarà més en detall.

#handler() s'executa a Lambda

def handler(event, context):
    fexec = lithops.ServerlessExecutor(runtime="hermes_analysis_img:01")
    fexec.call_async(render_analysis, data=event['params'].split(' '))
#codi que s'executa a Batch
    return {
        'statusCode': 200,
        'headers': {
            'Access-Control-Allow-Headers': 'Origin,Content-Type,X-Amz-
Date,Authorization,X-API-Key,X-Amz-Security-Token',
            'Access-Control-Allow-Origin': '*',
            'Access-Control-Allow-Methods': 'OPTIONS,POST'
        },
        'body': json.dumps('Hello from Lambda!')
    }

```

Codi 1. Codi de la funció Lambda.

En resum, es tracta d'una forma òptima d'executar els processos d'Hermes, ja que simplement es crea un script dinàmicament. Per tant, qualsevol dels altres mòduls d'Hermes, es podran executar de la mateixa forma.

Per acabar, cal recordar que HermesAnalysis és qui puja els fitxers a S3 també. Per a fer-ho a la ruta indicada (<username>/<experiment_name>), l'últim paràmetre que es passa correspon a aquesta.

6.1.2.2 Compressió dels resultats a Lambda

Tal i com s’ha esmentat, l’script de Python que comprimeix aquests fitxers es troba dins d’un altre contenidor que l’executa. Aquest contenidor s’executa mitjançant Lambda. Per a fer-ho, es podria haver utilitzat Lithops, però, en aquest cas no era del tot necessari degut a que només es necessita invocar aquesta Lambda. Per tant, la opció escollida ha estat la llibreria “boto3” que permet interactuar amb l’API d’AWS.

```
def zipResults(id):
    client = boto3.client('lambda',
        aws_access_key_id='aws_access_key_id ',
        aws_secret_access_key= aws_secret_access_key ',
        region_name='us-east-1')
    client.invoke(
        FunctionName='zipResults',
        InvocationType='Event',
        Payload=b'"{id}": \'+ id.encode() + b'"\}'')
)

zipResults(sys.argv[1])
```

Codi 2. Script que executa la funció Lambda.

Un cop configurat el client i indicat el directori a comprimir, s’executarà la compressió. Aquest script, s’anomena “zip.py” que correspon al que es crida quan finalitza l’execució d’HermesAnalysis.Rmd (última línia de la funció “render_analysis()” de l’apartat anterior). Per tant, un cop es disposen dels resultats a S3, es procedeix a comprimir-los.

6.1.2.3 Lithops

Pel que fa a l’ús d’aquest *framework* cal remarcar sobretot un dels requisits principals, el fitxer de configuració. Tal i com indica el manual oficial [8], s’han d’indicar una sèrie de paràmetres. Alguns són totalment necessaris, i d’altres són opcionals.

En aquest cas, s’han configurat els serveis AWS S3 i AWS Batch.

Donat que la configuració d’S3 no té massa misteri, es procedirà amb Batch. Aquesta requereix d’una mica més d’elaboració. A continuació, veurem els paràmetres més importants.

En primer lloc, s’ha d’especificar la memòria RAM i el nombre de CPUs virtuals desitjats per a cada contenidor. Podem veure cada contenidor com a una tasca de la cua independent. A més, s’especifiquen el nombre de CPUs virtuals totals que calen.

Un altre paràmetre és el *timeout*, que indicarà el temps màxim que s’esperarà Lithops per a fer la tasca. Si es supera aquest temps, la tasca serà avortada. Aquest temps compta amb un límit, que si es supera es treballa sobre el límit de Lithops. En el cas d’Hermes, en alguns casos es requereix de més temps. Per aquest motiu, s’ha demanat un canvi de fins a dos hores per als anàlisis més exigents.

Per acabar pel que fa a paràmetres, s’ha especificat que el tipus d’instàncies d’AWS utilitzades siguin “SPOT”. Aquest tipus de hardware es correspon a totes aquelles instàncies

que no estan sent utilitzades al núvol. Per tant, s'ofereixen a un preu molt més reduït que les instàncies "On-Demand". Concretament, "SPOT" costa fins a un 90% menys que l'altre alternativa.

Finalment, cal esmentar que la elasticitat de Lithops que s'ha explicat es produeix mitjançant aquest fitxer. Només cal que mitjançant un script es substituïxin els valors de la RAM i de les vCPUs. Però com s'ha dit, per ara aquesta funcionalitat s'ha desactivat fins que s'hagin realitzat els algorismes que prediran el cost en RAM. Per ara, Batch està configurat amb 4 vCPUs per cada contenidor, i 32GB de RAM.

6.1.2.4 Dockerfile per a AWS Batch

Com s'ha explicat anteriorment, Docker disposa de la opció de crear Dockerfiles per a configurar les imatges. El més rellevant en aquest cas, ha estat la extremada dificultat per a configurar un entorn per a l'execució d'RHermes.

En un inici, la dificultat principal va ser solucionar tots els errors que provoquen els paquets d'R. Un exemple pot ser que alguns paquets per defecte estan configurats per a ser executats en arquitectures de 32 bits. La solució va consistir en reunir una sèrie de llibreries de Linux, per a corregir aquesta anomalia.

D'altra banda, amb el temps van sorgir alguns problemes al intentar tornar a construir la imatge. Resulta que el paquet d'R "devtools", va començar a provocar masses problemes per a instal·lar RHermes. Segons la comunitat, es tracta d'un problema comú. En aquest cas, es va procedir a canviar de paquet pel "pak", que realitza la mateixa funció. La realitat és que s'han patit alguns errors més endavant, però s'han solucionat, i funciona correctament.

En resum, tal i com es pot apreciar, periòdicament poden sorgir certs desperfectes entre alguns paquets, que cal anar mantenint.

6.1.2.5 Seguretat d'API Gateway

Pel que fa a API Gateway, com s'ha vist s'encarrega de fer arribar les peticions dels usuaris a Lambda. Però gràcies a les opcions de configuració que disposa, s'ha pogut solucionar un problema de seguretat important: que qualsevol persona pugui encarregar tasques.

Anteriorment s'ha comentat que si s'accedeix a la web sense estar registrat, aquesta ens redirigirà automàticament a l'inici de sessió. Però, no es una solució ja amb eines com Postman es poden enviar peticions a l'API.

La solució ha estat especificar mitjançant la consola d'AWS, que només es permetin les sol·licituds que compleixin un requisit concret. Aquest es tracta de que els usuaris estiguin registrats a AWS Cognito.

Per a fer-ho, només cal enviar un *token* d'identificació amb la petició, per a que API Gateway pugui verificar que existeix l'usuari que s'ha indicat. En cas de que no ho estigui, es retorna l'error "401 Unauthorized".

6.1.2.6 Seguretat d'S3

Un altre aspecte important de seguretat és el d'accedir a fitxers que no pertanyen a un usuari concret. Per tant, aquest podria intentar fer-ho de manera fraudulenta, per a robar dades que poden ser d'alta confidencialitat. Un exemple poden ser dades tècniques sobre un medicament que encara no s'ha patentat.

Per evitar aquests perills, s'ha creat una política d'IAM que restringeix l'accés als següents "buckets". El primer és "hermes-cloud-tmp" que guarda els fitxers d'entrada. El segon és "hermes-cloud-output" que emmagatzema els resultats. Ambdós estan estructurats de la mateixa manera, cada usuari té una carpeta personal.

Per tant, la solució és relativament senzilla. Només cal no permetre l'accés a directoris que no corresponguin als de cada usuari. Per això, s'ha indicat que s'apliqui la següent lògica als usuaris amb un nom d'usuari registrat a AWS Cognito. A continuació, s'ha especificat que puguin llegir i escriure d'aquests "buckets" només en la ruta següent: "`/<username>/*`". D'aquesta manera, IAM no els permetrà intentar realitzar operacions en rutes que no els correspongui.

6.1.2.7 Vue

Aquest apartat tracta d'una decisió que s'ha pres recentment en el projecte. Com que encara no està del tot acabada, es tractarà de manera breu ja que és interessant des del punt de vista teòric.

Actualment, el desenvolupament de la versió estudiada en aquest projecte s'ha aturat per a emigrar l'aplicació web a Vue.js. Es tracta d'un *framework* de JavaScript que permet de manera senzilla desenvolupar interfícies professionals i SPA (single-page application). Combina alguns aspectes d'Angular i React, i es basa en HTML, CSS i JavaScript.

A més, proporciona un model de programació declaratiu que es basa en l'ús de components. D'aquesta manera, permet crear codi reutilitzable per a d'altres ocasions.

En aquest cas resulta ideal ja que el que s'ha fet per exemple, és crear un component que representa l'àrea per a pujar fitxers. Per tant, es pot reutilitzar aquest component tantes vegades com faci falta.

Doncs bé, la realitat és que l'aplicació en Vue està prou enllestida. La major part de la feina ha consistit en canviar la forma en que s'utilitza l'SDK d'AWS. Però, encara falten aspectes importants com per exemple, canviar les vistes en funció de certs esdeveniments. Un exemple pot ser amagar la pàgina de l'autenticació per a mostrar la pàgina principal de *l'app*.

En resum, s'ha escollit l'ús de Vue degut a la reutilització del codi. A més, ofereix un rendiment excel·lent, i més comparat amb la competència.

6.1.3 Optimització d'RHermes

Una de les fases principals que s'han dut a terme en aquest projecte, és la optimització d'RHermes. La millor manera de fer-ho, és mitjançant l'execució massiva de funcions Lambda paral·leles. Això comportaria un canvi molt important en el codi, i dificultaria el seu manteniment. Com que en conjunt amb el Roger s'ha decidit optimitzar més a fons el paquet, s'ha procedit d'una altra manera.

En aquest cas, s'ha intentat mantenir el mateix paquet d'RHermes, però sobreescrivint algunes funcions a HermesAnalysis.Rmd. Aquestes contenen algunes optimitzacions que veurem a continuació.

En primer lloc, s'ha realitzat un estudi del codi per a detectar les parts del codi més exigents des del punt de vista computacional. Per fer-ho, s'ha utilitzat la llibreria “profvis” que ens permet analitzar el codi més en detall. En el cas d'RHermes, s'han detectat que les funcions que més temps requereixen són processMS1, findSOI i filterSOI. De les tres anteriors, la que resulta més costosa és findSOI per diferència. En concret, la meitat del temps d'execució correspon a aquest procediment. Cal remarcar que totes aquestes s'executen de manera seqüencial perquè cadascuna depèn del resultat de l'anterior.

En segon lloc, s'ha procedit a paral·lelitzar el codi intern de les funcions anteriors. Pel que fa a processMS1, s'ha procedit amb les funcions internes que utilitza (preprocessing(), IsoCalc(), IonicForm() i import_and_filter()). La funció filterSOI se li han paral·lelitzat els dos bucles més exigents. Per últim, findSOI s'ha optimitzat cadascuna de les funcions.

Fins a aquest punt, s'han experimentat millores considerables que es detallaran més endavant.

Cal dir, que cada procés que s'executa processa un dels fitxers de tipus “Sample”. Per tant, si s'introdueix un únic fitxer el temps d'execució del script serà el mateix que sense optimitzar. Per solucionar-ho, s'ha realitzat un *profiling* del codi intern de cada una d'aquestes funcions. A continuació, s'ha procedit a paral·lelitzar les parts més costoses. La realitat és que el sobrecost de la paral·lelització en R, ha resultat excessivament gran per a optimitzar el processat de cada fitxer. Concretament, podem estar parlant de que aquesta versió del codi ha resultat un 40% menys eficient.

Com aquesta versió del codi s'ha executat cada fil en un procés (multiprocessing), s'ha intentat mitjançant fils d'execució (multithreading). Però, no s'han notat millores.

Després de molta investigació a Internet, s'ha arribat a la conclusió que R és un llenguatge bastant ineficient. Sobretot intentar paral·lelitzar codi d'R, resulta ser poc beneficiós en alguns casos. El motiu és que utilitza masses recursos per cadascun dels processos. Però, la RAM és el recurs que més pateix en aquest cas. Per a alguns experiments, el sistema operatiu pot arribar a, fins i tot, a quedar paral·litzat.

Per tant, la decisió final ha consistit en mantenir la versió inicial en la que la paral·lelització es produeix a nivell de fitxer.

Per últim, en R es paral·lelitzava mitjançant *clusters*. En aquest cal especificar el nombre de processadors locals que es dedicaran. Per tant, s'ha optat per detectar dinàmicament el nombre de processadors de la màquina local. D'aquest nombre, s'assignen tots els processadors menys un per a no saturar el sistema.

6.1.3.1 Benchmarks

A continuació, es poden veure les proves que s'han d'aquesta versió d'RHermes, en diferents instàncies d'AWS. En concret, cada cas s'acompanya amb una gràfica del temps d'execució i els costos d'execució.

Pel que fa a les especificacions de les instàncies utilitzades es poden trobar a la següent taula.

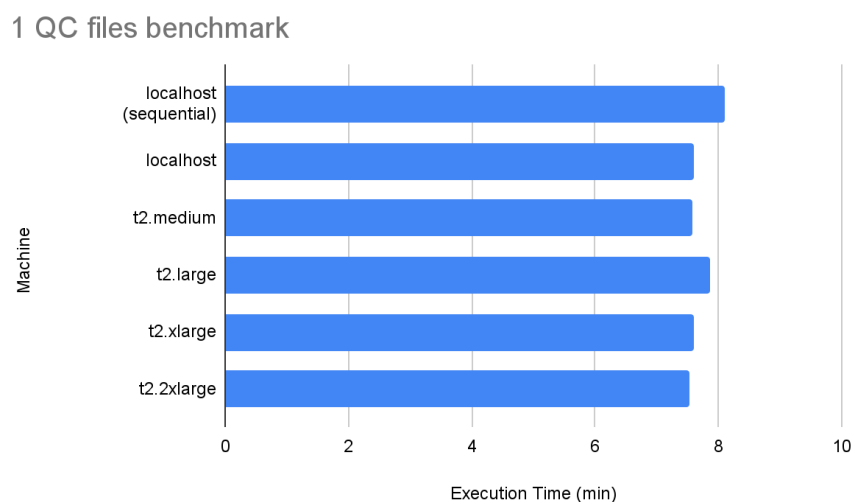
	Nombre	vCPU	RAM (GiB)
t2.nano	1	1	0,5
t2.micro	1	1	1,0
t2.small	1	1	2,0
t2.medium	2	2	4,0
t2.large	2	2	8,0
t2.xlarge	4	4	16,0
t2.2xlarge	8	8	32,0

Il·lustració 59. Il·lustració de la taula d'especificacions d'AWS.

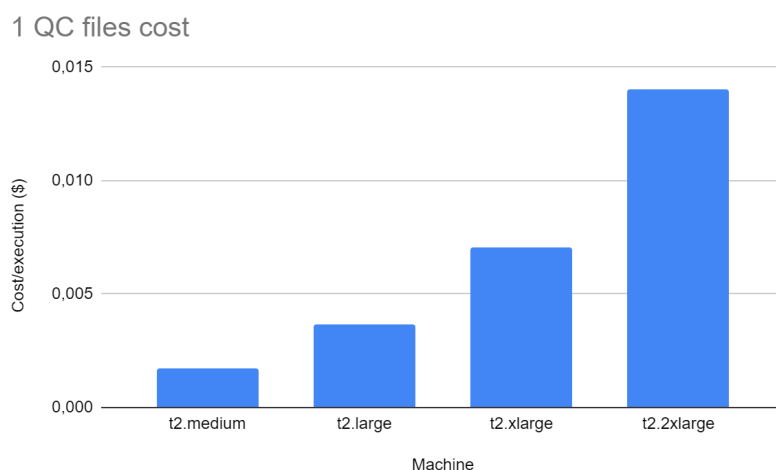
Cal remarcar que els fitxers de tipus “Sample” i “QC” són els mateixos.

6.1.3.1.1 Benchmark 1

Per al primer cas, s'indica el temps d'execució de la versió seqüencial al meu ordinador local (disposa de 4 CPUs i 8GB de RAM). Pel que fa a la resta, indiquen el temps d'execució de la versió paral·lelitzada.

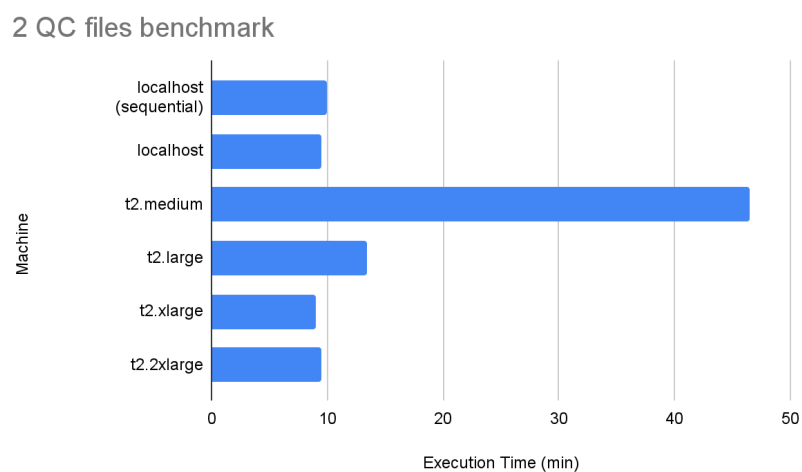


Il·lustració 60. Gràfica de l'execució amb 1 fitxer "Sample".

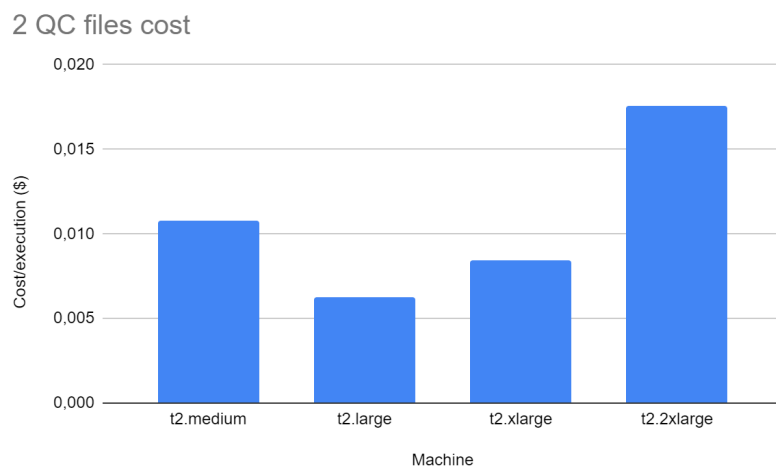


Il·lustració 61. Gràfica dels costos de cada instància amb 1 fitxer "Sample".

6.1.3.1.2 Benchmark 2



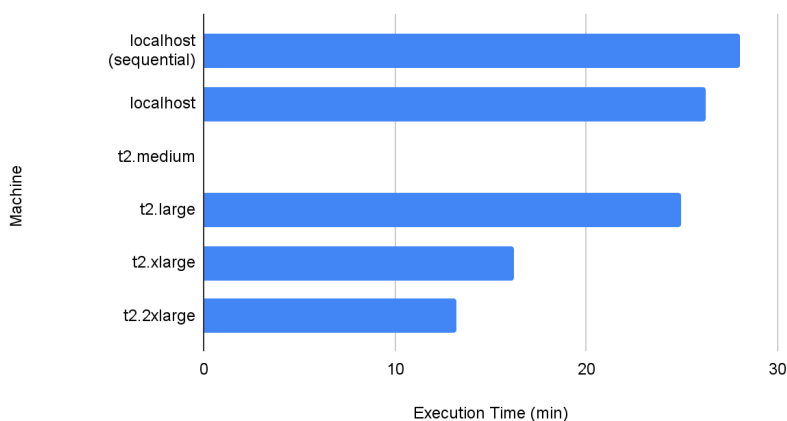
Il·lustració 62. Gràfica de l'execució amb 2 fitxers "Sample".



Il·lustració 63. Gràfica dels costos de cada instància amb 2 fitxers "Sample".

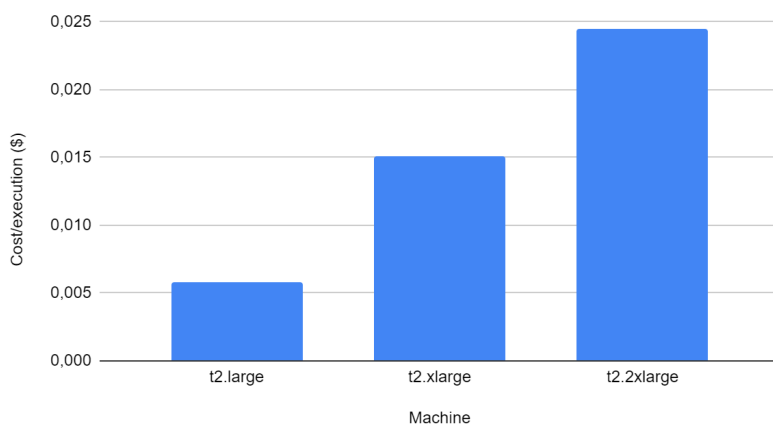
6.1.3.1.3 Benchmark 3

4 QC files benchmark



II·lustració 64. Gràfica de l'execució amb 4 fitxers "Sample".

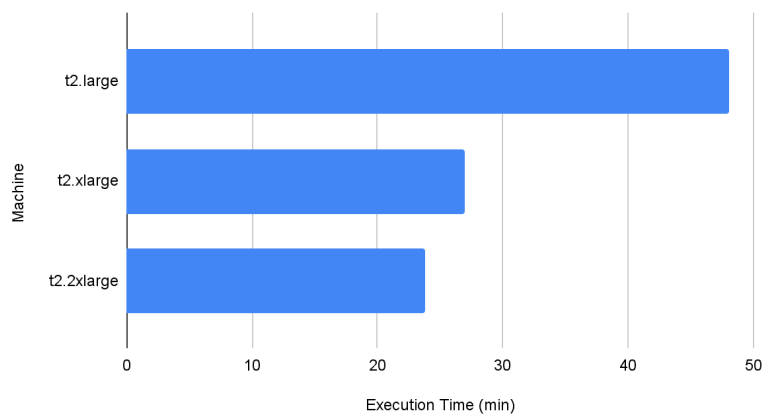
4 QC files cost



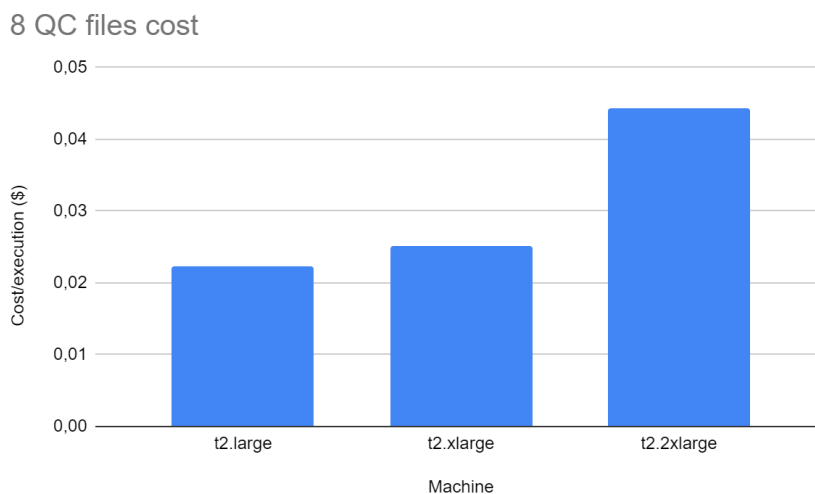
II·lustració 65. Gràfica dels costos de cada instància amb 4 fitxers "Sample".

6.1.3.1.4 Benchmark 4

8 QC files benchmark



II·lustració 66. Gràfica de l'execució amb 8 fitxers "Sample".



Il·lustració 67. Gràfica dels costos de cada instància amb 8 fitxers "Sample".

6.1.3.1.5 Conclusions

Per concloure, en els dos primers casos no s'aprecia cap millora rellevant. És més, pel cas de la t2.medium del segon cas es pot apreciar com es dispara el temps. A més, en algunes proves no s'han inclòs algunes màquines degut a la seva saturació. Però, l'important és que la resta han millorat al voltant d'un 50% en la màquina més potent. Concretament, aquesta disposa de 8 vCPUs i 32GB de RAM.

7 Avaluació

Per a avaluar la feina feta, es farà un experiment des del primer pas fins a l'últim. D'aquesta manera, es mostrarà un exemple d'ús de l'aplicació.

En concret, s'ha fet l'experiment per defecte amb el que s'han fet els *benchmarks* anteriors.

7.1 Experiment per defecte

7.1.1 Registre

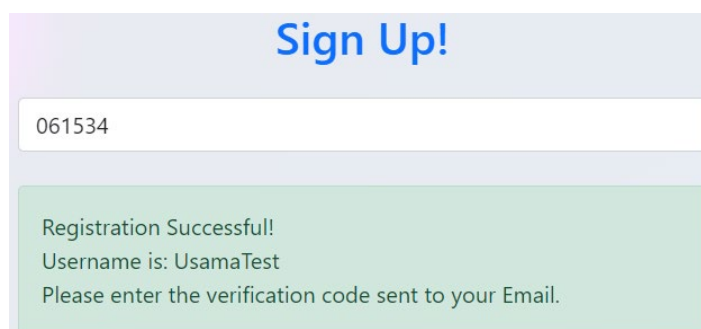


The screenshot shows a 'Sign Up!' form with the following fields and content:

- Username field: UsamaTest
- Email field: usama.benabdelkrim@estudiants.urv.cat
- Password field: (masked)
- Confirm Password field: (masked)
- Message box: Please fill all the fields!

Il·lustració 68. Registre d'un nou usuari a la plataforma

7.1.1.1 Verificació

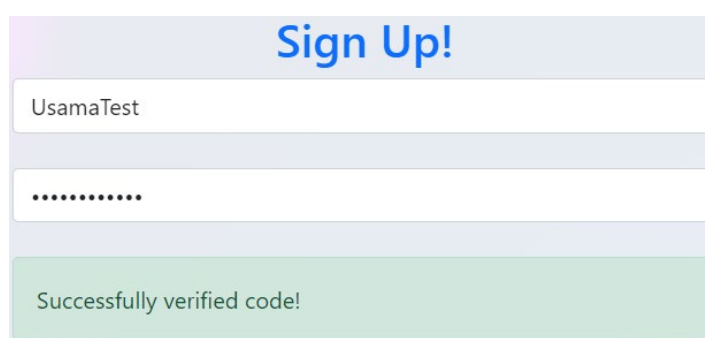


The screenshot shows a 'Sign Up!' form with the following fields and content:

- Verification code field: 061534
- Message box: Registration Successful!
Username is: UsamaTest
Please enter the verification code sent to your Email.

Il·lustració 69. Verificació d'un usuari.

7.1.2 Autenticació



The screenshot shows a 'Sign Up!' form with the following fields and content:

- Username field: UsamaTest
- Password field: (masked)
- Message box: Successfully verified code!


Il·lustració 70. Autenticació d'un usuari.

7.1.3 Configuració d'una tasca

Instrument	Sample	Processing
PPM error 3	Ionization Mode Positive (+)	SOI Detection Type Double
Minimum m/z 50	Select a formula database: [Dropdown]	<input checked="" type="checkbox"/> SOI Filter
Maximum m/z 1200	or submit your own: Formula DB [Browse]	<input checked="" type="checkbox"/> Isotopic fidelity filter
Instrument resolution 60000	Adducts List [Dropdown]	<input type="checkbox"/> Consensus
Noise Threshold 1000	<input checked="" type="checkbox"/> M+H <input checked="" type="checkbox"/> M+NH4 <input checked="" type="checkbox"/> M+Na <input checked="" type="checkbox"/> M+K <input checked="" type="checkbox"/> M+	Consensus Min. Samples 1
Instrument type Orbitrap		
Instrument brand Thermo		

Il·lustració 71. Paràmetres escollits per a la tasca.



Blank File Uploader



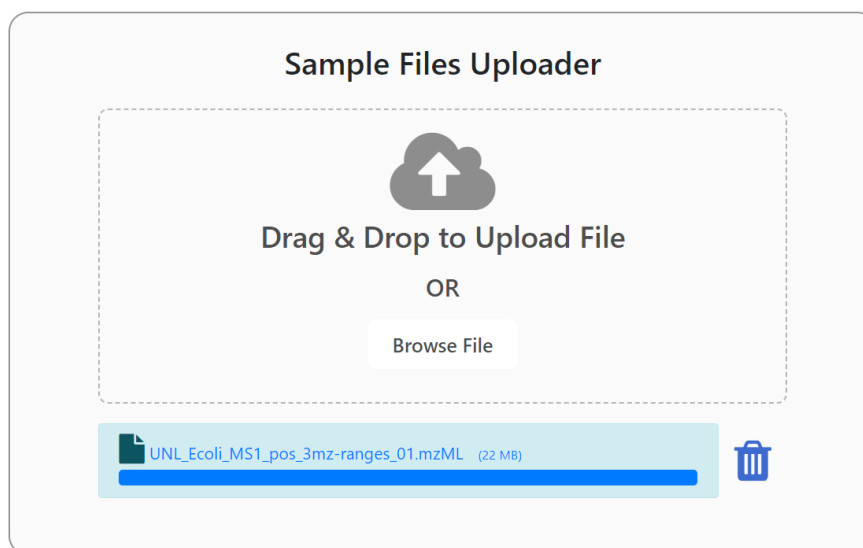
Drag & Drop to Upload File

OR

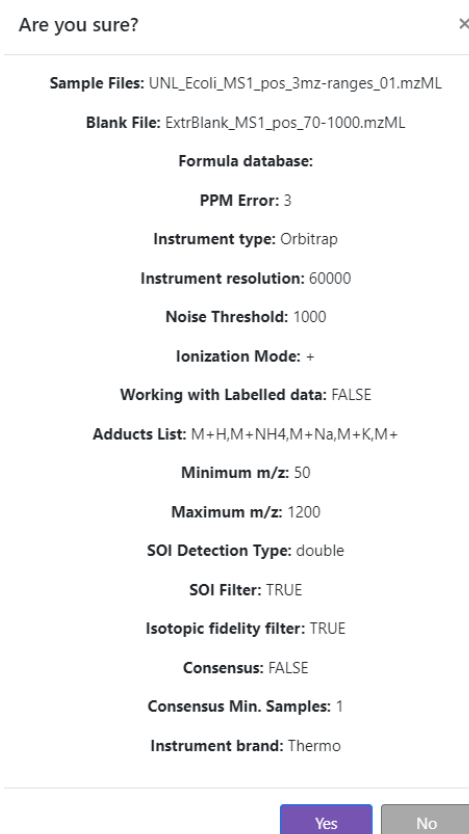
[Browse File](#)


ExtrBlank_MS1_pos_70-1000.mzML (22 MB)

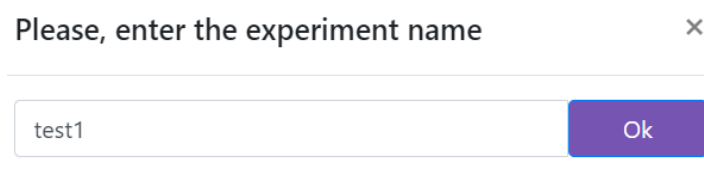
Il·lustració 72. Fitxer "Blank" adjuntat.



II·lustració 73. Fitxer "Sample" adjuntat.



II·lustració 74. Confirmació dels paràmetres escollits.



II·lustració 75. Introducció del nom de l'anàlisi.

Running Job...



Test1 job will take X min approximately...



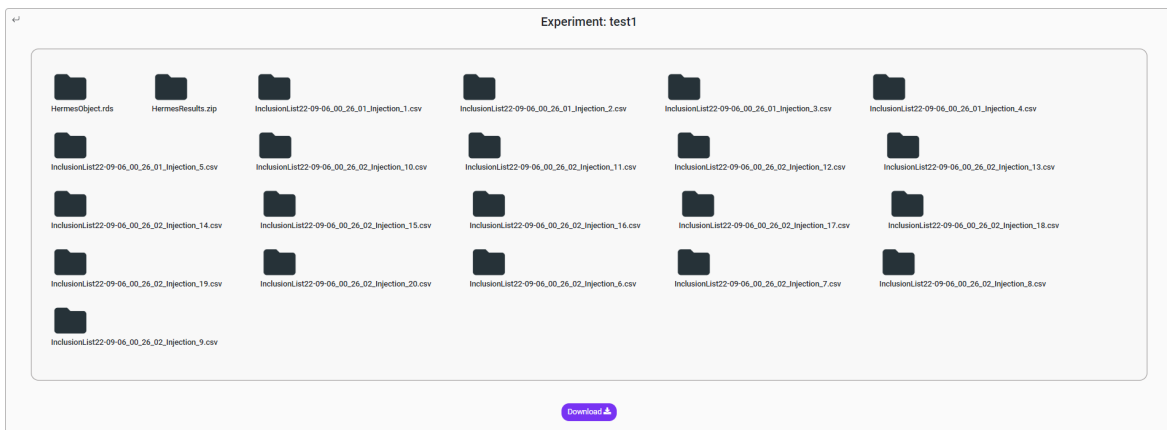
Ok

II·lustració 76. S'informa de que la tasca corre correctament.

aws-batch_lithops_v2-7-1-dev0_FPQ4-SPOT-hermes_analysis_img-01--30720mb_6f60c2-0-A000	368d3ba4-4bc3-4657-a73a-055053675b7f	container	--	Aug 22 2022 18:15:30	Aug 22 2022 18:19:10	Aug 22 2022 18:27:29	0 días 00:08:19	SUCCEEDED
---	--------------------------------------	-----------	----	----------------------	----------------------	----------------------	-----------------	-----------

II·lustració 77. Captura de la tasca enviada a Batch.

7.1.4 Descàrrega dels resultats



II·lustració 78. Sortida visualitzada al FileManager.

Nombre	Tipo	Tamaño comprimido
HermesObject.rds	Archivo RDS	30.258 KB
InclusionList22-09-06_00_26_01_Injection_1.csv	Archivo de origen Comma Separ...	4 KB
InclusionList22-09-06_00_26_01_Injection_2.csv	Archivo de origen Comma Separ...	4 KB
InclusionList22-09-06_00_26_01_Injection_3.csv	Archivo de origen Comma Separ...	4 KB
InclusionList22-09-06_00_26_01_Injection_4.csv	Archivo de origen Comma Separ...	4 KB
InclusionList22-09-06_00_26_01_Injection_5.csv	Archivo de origen Comma Separ...	3 KB
InclusionList22-09-06_00_26_02_Injection_6.csv	Archivo de origen Comma Separ...	3 KB
InclusionList22-09-06_00_26_02_Injection_7.csv	Archivo de origen Comma Separ...	3 KB
InclusionList22-09-06_00_26_02_Injection_8.csv	Archivo de origen Comma Separ...	2 KB
InclusionList22-09-06_00_26_02_Injection_9.csv	Archivo de origen Comma Separ...	2 KB
InclusionList22-09-06_00_26_02_Injection_10.csv	Archivo de origen Comma Separ...	3 KB
InclusionList22-09-06_00_26_02_Injection_11.csv	Archivo de origen Comma Separ...	3 KB
InclusionList22-09-06_00_26_02_Injection_12.csv	Archivo de origen Comma Separ...	2 KB
InclusionList22-09-06_00_26_02_Injection_13.csv	Archivo de origen Comma Separ...	2 KB
InclusionList22-09-06_00_26_02_Injection_14.csv	Archivo de origen Comma Separ...	2 KB
InclusionList22-09-06_00_26_02_Injection_15.csv	Archivo de origen Comma Separ...	1 KB
InclusionList22-09-06_00_26_02_Injection_16.csv	Archivo de origen Comma Separ...	1 KB
InclusionList22-09-06_00_26_02_Injection_17.csv	Archivo de origen Comma Separ...	1 KB
InclusionList22-09-06_00_26_02_Injection_18.csv	Archivo de origen Comma Separ...	1 KB

II·lustració 79. Captura dels resultats descarregats.

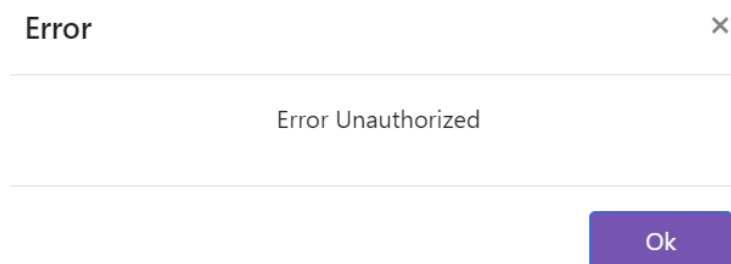
7.2 Casos d'error

Per a no allargar aquesta memòria, en aquest cas no s'inclouran els casos d'error que s'han vist al disseny.

En tot cas, tractarem els que no s'han vist de manera visual. Per tant, els que no s'han demostrat.

7.2.1 Petició de la tasca sense token

Per a reproduir aquest cas, només cal eliminar el camp *Authorization de la petició*.

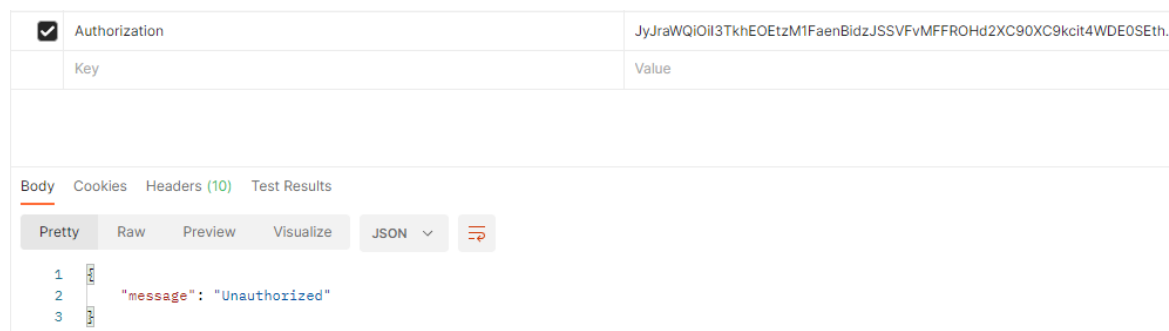


Il·lustració 80. Missatge d'error per a peticions sense el token.

7.2.2 Petició de la tasca amb un token erroni

En aquest cas, el que es vol demostrar és que no es puguin enviar tasques per part d'usuaris no registrats. Aquests poden intentar enviar qualsevol token. Per això, per facilitar les proves s'ha utilitzat Postman.

A continuació, veurem la petició amb un token erroni:



Il·lustració 81. Captura del resultat d'una petició no vàlida.

7.2.3 Petició de la tasca amb un token vàlid

En aquest cas, es tracta una situació normal d'un usuari que encarrega un experiment.

<input checked="" type="checkbox"/>	Authorization	eyJraWQlOIl3TkhEOEtzM1FaenBidzJSSVfVnMFFROhd2XC90XC9kcit4WDE0SEth...
	Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "statusCode": 200,
3    "headers": {
4      "Access-Control-Allow-Headers": "Origin,Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token",
5      "Access-Control-Allow-Origin": "*",
6      "Access-Control-Allow-Methods": "OPTIONS,POST"
7    },
8    "body": "\"Hello from Lambda!\""
9  }

```

Il·lustració 82. Captura de la resposta d'una tasca enviada correctament.

8 Conclusions

Per recapitular, en aquest projecte s'ha emigrat una aplicació redissenyada al núvol. Per a fer-ho realitat, ha calgut crear una nova interfície per a interactuar amb l'arquitectura desenvolupada. A més, s'ha tancat el cicle de vida per a una plataforma sense servidor. Aquesta, representa tota la infraestructura sobre la que s'executaran tots els possibles mòduls d'Hermes, i de futures aplicacions. Addicionalment, s'ha hagut de tractar codi en R per a optimitzar certs processos del procés implementat en aquest projecte.

A més, s'ha dissenyat un sistema d'autenticació que permet monitoritzar les despeses del sistema. Com el diner és un recurs limitat, també ha calgut tractar aspectes de seguretat que podrien produir un ús fraudulent del sistema.

Tot seguit, m'agradaria posar en volar el coneixement adquirit i les tecnologies que s'han hagut d'aprendre en aquest TFG.

Per un costat, tot i haver treballat el *cloud computing* durant els estudis superiors, en aquest projecte s'ha treballat a un altre nivell. A partir de les diferents necessitats s'han hagut de cercar les eines pel *cloud* més adequades, i aprendre a utilitzar-les. Val la pena dir que una de les dificultats més gran, es tracta del nivell d'abstracció amb el que es treballa. Es tracten de tecnologies molt complexes d'entendre en un principi. Per això, cal dedicar una fase d'estudi, i una altre d'experimentació per a familiaritzar-se amb la seva implementació. Un exemple d'aquestes eines, pot ser AWS Batch, Lambda o Lithops. Realment requereixen d'un esforç per a entendre el seu funcionament i les seves arquitectures. A més, un cop s'ha entès en detall, caldrà dissenyar una forma d'aplicar-les al projecte corresponent, i implementar-la.

Per l'altre, cal remarcar també la interacció amb el món de la metabolòmica. Es tracta d'un camp de la química, que com és d'esperar, desconexem com a enginyers informàtics. És per aquest motiu, que per a començar amb aquest projecte s'ha necessitat una fase d'anàlisi de codi i investigació. En tot cas, com a informàtics només podem entendre el que es fa a nivell computacional (estructura del codi, bucles, lògica, etc).

A més a més, cal recordar que s'ha creat un sistema d'usuaris els quals podran tenir accés a la plataforma, de manera que es monitoritzi el seu ús. Però, no s'ha dissenyat simplement per a un prototip d'un treball acadèmic. La realitat es que es disposaran d'usuaris reals (científics, professionals de la química o farmacèutiques), que podran contractar un servei que els faciliti la seva feina. Per això, actualment s'està treballant en la creació d'un *startup*, que mitjançant un model de negoci, es pugui oferir un servei totalment funcional.

Per últim, tal com s'ha demostrat a la introducció, la computació al núvol creix constantment, i és un camp amb investigacions molt importants. A més, innumerables empreses estan fent inversions importants per a emigrar les seves plataformes al núvol. En aquest Treball de Fi de Grau, s'ha treballat en un d'aquests casos. D'aquesta manera, aquest projecte es converteix per a mi en una font molt enriquidora d'experiència, que serà molt útil per a la meua futura carrera professional.

9 Recursos utilitzats

9.1 Bibliografia web

- 1] J. Flynn, «Zippia,» 23 03 2022. [En línia]. Available: https://www.zippia.com/advice/cloud-adoption-statistics/#Cloud_Adoption_Statistics_by_Clespacio.
- 2] H. Yang i Mary Tate, «ResearchGate,» 01 2009. [En línia]. Available: https://www.researchgate.net/publication/228709740_Where_are_we_at_with_cloud_computing_A_descriptive_literature_review.
- 3] H. Li, Zhang Chendong, Ti Yang i Wang Ce, «ResearchGate,» 08 2021. [En línia]. Available: https://www.researchgate.net/publication/353634653_Analysis_the_current_state_of_the_cloud_computing_development.
- 4] Insights, Fortune Business, «GlobeNewswire,» 15 06 22. [En línia]. Available: <https://www.globenewswire.com/en/news-release/2022/06/15/2462842/0/en/With-17-9-CAGR-Global-Cloud-Computing-Market-Size-Worth-USD-791-48-Billion-in-2028.html>.
- 5] CloudButton, «CloudButton,» [En línia]. Available: <https://cloudbutton.eu/>.
- 6] J. Finch i X. Domingo-Almenara, «Github,» 27 05 2021. [En línia]. Available: <https://github.com/xdomingoal/erah-devel>.
- 7] Lithops Team, [En línia]. Available: <https://lithops-cloud.github.io/docs/index.html>.
- 8] Lithops Team, «Github,» [En línia]. Available: https://github.com/lithops-cloud/lithops/blob/master/docs/source/compute_config/aws_batch.md.
- 9] R. Giné Bertomeu, J. M. Badia i J. Capellades Tomas, «Github,» 01 11 2021. [En línia]. Available: <https://github.com/RogerGinBer/RHermes>.
- 10] Rstudio, [En línia]. Available: <https://shiny.rstudio.com/>.
- 11] AWS, [En línia]. Available: <https://aws.amazon.com/es/s3/faqs/>.
- 12] D. S. Ruenes. [En línia]. Available: https://campusvirtual.urv.cat/pluginfile.php/3769299/mod_resource/content/11/Tema%20IV-2020.pdf.
- 13] «Wikipedia,» 10 08 2022. [En línia]. Available: [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)).

- 14] AWS, [En línia]. Available: https://docs.aws.amazon.com/es_es/cognito/latest/developerguide/what-is-amazon-cognito.html.
- 15] T. Bacinger, «Toptal,» [En línia]. Available: <https://www.toptal.com/frontend/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>.
- 16] AWS, [En línia]. Available: <https://aws.amazon.com/es/api-gateway/>.
- 17] AWS, [En línia]. Available: <https://aws.amazon.com/es/batch/>.
- 18] AWS, [En línia]. Available: https://aws.amazon.com/es/s3/?trk=d0993ae4-4193-4d67-b2a9-e83cdb563369&sc_channel=ps&s_kwid=AL!4422!3!588732081318!e!!g!!aws%20s3&ef_id=CjwKCAjw9suYBhBIEiwA7iMhNMpeccQrVFbdcn76BK6TG3xmc1CMh7iVgGDB5rL9OSrk1tMt_QQ-8hoC91oQAvD_BwE:G:s&s_kwid=AL!4422!3!5887.
- 19] MDN, [En línia]. Available: <https://developer.mozilla.org/es/docs/Learn/JavaScript>.
- 20] MDN, [En línia]. Available: <https://developer.mozilla.org/es/docs/Web/CSS>.
- 21] MDN, [En línia]. Available: <https://developer.mozilla.org/es/docs/Web/HTML>.
- 22] «Wikipedia,» 27 07 2022. [En línia]. Available: [https://ca.wikipedia.org/wiki/R_\(llenguatge_de_programaci%C3%B3\)](https://ca.wikipedia.org/wiki/R_(llenguatge_de_programaci%C3%B3)).
- 23] AWS, [En línia]. Available: <https://aws.amazon.com/es/docker/>.
- 24] rpy2, «Github,» [En línia]. Available: <https://rpy2.github.io/doc/latest/html/index.html>.
- 25] Vue.js, [En línia]. Available: https://vuejs.org.translate.google.com/guide/introduction.html?_x_tr_sl=en&_x_tr_tl=ca&_x_tr_hl=ca&_x_tr_pto=sc.
- 26] altexsoft, 11 09 2019. [En línia]. Available: <https://www.altexsoft.com/blog/engineering-pros-and-cons-of-vue-js/>.
- 27] S. Kailasam, «News Courier,» 17 05 2021. [En línia]. Available: <https://www.news-courier.com/tn/articles/lc-ms-what-is-lc-ms-lc-ms-analysis-and-lc-msms-348238#D3>.

9.2 Programari

- 1) AWS Cognito
- 2) AWS S3
- 3) AWS API Gateway
- 4) AWS Lambda
- 5) AWS Batch
- 6) HTML i CSS
- 7) JavaScript
- 8) R
- 9) Python
- 10) Shiny
- 11) Bootstrap
- 12) Lithops

10 Annexes

10.1 Annex 1: Posada en marxa

Per a accedir a la web, caldrà introduir el següent enllaç al navegador.

Enllaç: <https://hermes-cloud-cognito.s3.amazonaws.com/index.html/index.html>

Pel que fa a l'ús d'Hermes al núvol, no es requereix cap mena d'instal·lació. Tal com s'ha dit, és una aplicació web i per tant, es necessita únicament un navegador amb connexió a internet.

10.2 Annex 3: Manteniment

Pel que fa al manteniment d'Hermes, només cal tronar a construir la imatge que conté els seus scripts. Aquesta es correspon a la mateixa que Batch. Per tant, per a qualsevol actualització, caldrà executar *lithops build* amb els canvis localment.