

---

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

Sulé Carrasco, Marina; Martí Godia, Enric, dir. Sistema d'alertes per a bases de dades. 2022. (1394 Enginyeria de Dades)

---

This version is available at <https://ddd.uab.cat/record/264633>

under the terms of the  license

# Sistema d'alertes per a bases de dades

Marina Sulé Carrasco

## Resum

Aquest treball neix de la necessitat d'una empresa tecnològica alhora de garantir la qualitat de les dades rebudes a temps real d'una empresa externa amb les que, mitjançant la intel·ligència artificial, proporciona prediccions. Per poder donar resposta a aquesta necessitat, s'ha desenvolupat un sistema d'alertes que consisteix en identificar i notificar els errors de les dades recepcionades.

El sistema d'alertes s'ha dissenyat i desenvolupat en dos procediments independents. Un a través d'una aplicació externa a les dues empreses i l'altre mitjançant funcions implementades en el codi intern de l'empresa tecnològica. Els dos procediments tenen el mateix objectiu i funcionalitat (detectar errors a les dades tant de contingut com de format).

En l'article podem trobar el disseny, el desenvolupament i la verificació del correcte funcionament dels dos procediments, així com les conclusions del treball.

## Paraules clau d'aquestes

Alerta, Amalfy Analytics, Aplicació, Notificació, Qualitat i Slack.

## Abstract

This work comes from the need of a technological company to guarantee the quality of the data received in real time from an external company with, by means of artificial intelligence, provides predictions. In order to meet this need, an alert system has been developed that consists of identifying and notifying errors in the data received.

The alert system has been designed and developed in two independent procedures. One by means of an external application to the two companies and the other by means of functions implemented in the internal code of the technological company. Both procedures have the same objective and functionality (to detect errors in the data both in content and format).

In the article we can find the design, development and verification of the correct operation of both procedures, as well as the conclusions of the work.

## Index Terms

Alert, Amalfy Analytics, Application, Notification, Quality and Slack.

---

## 1 INTRODUCCIÓ - CONTEXT DEL TREBALL

Les noves tecnologies relacionades tant amb l'entorn personal com en l'empresarial, estan agilitzant, optimitant i perfeccionant la informació sobre les activitats que realitzen tots els ciutadans en el dia a dia. Com a conseqüència, el tractament de la informació és una eina en contínua evolució en els darrers anys.

Un exemple de comunicació actual, a través d'Internet, és la transmissió de missatges, imatges, vídeos i tota mena de documents des de diferents parts del món durant les 24h del dia a temps real. Això ha desplaçat definitivament l'enviament de documents per altres mitjans.

És per aquest motiu que actualment moltes empreses volen un sistema de comunicació instantani. Les empreses tecnològiques el voldran per les seves bases de dades o pel treball que en fan de les dades i les empreses del sector comercial, per exemple, ho voldran per millorar la gestió en les seves vendes.

- 
- E-mail de contacte: [marinasulecarrasco@gmail.com](mailto:marinasulecarrasco@gmail.com)
  - Treball tutoritzat per: Enric Mart Gòdia (Departament de les Ciències de la Computació)
  - Curs 2021/22

Amalfi Analytics és una empresa tecnològica que treballa amb intel·ligència artificial (IA). Actualment treballa conjuntament amb el Servei de la Salut, més concretament amb hospitals públics de Catalunya.

L'empresa Amalfi Analytics es dedica a proporcionar diferents tipus de serveis mitjançant intel·ligència artificial, ja que amb aquesta pot fer prediccions a temps real mitjançant les dades proporcionades des dels hospitals.

Mitjançant aquesta IA, proporcionen informació perquè els hospitals tinguin una orientació de quin serà el fluxe de gent que tindran, per exemple, en la següent hora. Així doncs, podran predir quina quantitat de treballadors necessitaran o quina quantitat de sales o boxes els caldrà preparar adicionalment.

A més a més, proporcionen altres serveis, com pot ser ajudar també en l'àrea de recursos humans. Ho fan mitjançant la predicció, per exemple, de baixes de treballadors a una setmana vista, tant per planta com per departament. D'aquesta manera poden saber la quantitat de personal actiu amb el que poden comptar o amb la quantitat de personal que hauran de contractar adicionalment.

Un dels principals problemes amb els que s'ha trobat l'empresa, és que basen tot el seu treball amb la qualitat de les dades que els hospitals envien. I això pot causar múltiples problemes, ja que la qualitat d'aquestes dades no recau en ells, si no en l'hospital amb el que treballen.

El fet de dependre d'una altra empresa per poder obtenir les dades no garanteix la qualitat d'aquestes. En ocasions les empreses tecnològiques han de treballar amb empreses i institucions on la seva principal preocupació i recursos no estan destinats a l'àrea informàtica.

Una mala gestió és pot veure reflectida en la qualitat de les dades obtingudes, cosa que és comprensible, fins a un cert punt, ja que en el nostre cas estem parlant del Servei de la Salut. Aquest organisme té com a principal objectiu i funció el benestar de la societat i no pas el bon emmagatzematge de les dades que obtenen.

Els problemes amb els que, fins el moment, s'ha trobat l'empresa Amalfi, han estat causats tant per errors en els continguts com per a la manca de dades amb les que treballar.

Respecte els errors de continguts es troben diferents tipus, però els més usuals són errors en les dates d'administratives. Un exemple, seria la data d'entrada i sortida d'una persona a l'hospital, és a dir, que es registri que una persona ha estat ingressada el dia 10 de maig i que hagi obtingut l'alta el 5 de maig. Com es pot veure hi ha una incongruència, ja que el temps és lineal.

A més a més, s'han trobat casos amb dades molt poc probables o irrealistes, com pot ser que l'edat introduïda d'un pacient hagi estat de 150 anys.

Tots aquests errors provoquen que l'empresa tecnològica, en aquest cas Amalfi, hagi de fer doble treball i assegurar-se que les dades obtingudes són les correctes. Això implica, invertir treball, personal i recursos addicionals en la validació de les dades abans de poder començar a treballar en intel·ligència artificial.

Un altre problema amb el que s'ha trobat Amalfi Analytics és la manca de dades. És a dir, fer prediccions sense dades a temps reals.

Això, la gran majoria dels casos, ve causat per problemes tècnics per part de l'hospital. Però implica complicacions en el servei que ofereix l'empresa.

Actualment la manera d'identificar aquest error és per la capacitat que tenen els treballadors de l'empresa per detectar-ho. Això implica treball adicional, com he comentat abans, ja que no hauria de ser responsabilitat de l'empresa que reb les dades comprovar si aquestes han arribat correctament. La responsabilitat de la qualitat del contingut de les dades hauria de recaure en l'empresa que les envia, i si fos el cas, solucionar els problemes.

Per tant, un dels principals objectius que té Amalfi a dia d'avui, és poder tenir un filtre pel qual passin les dades que reben per tal de saber si estan bé abans de treballar amb elles. O per un altre banda, si les dades amb les que han de treballar han arribat.

Actualment la majoria de sectors en el mercat treballen mitjançant l'obtenció de dades o els beneficis que obtenen treballant amb aquestes. Aquest sistema d'alertes i l'objectiu amb el que ha estat creada, fan que aquesta aplicació sigui exportable a altres empreses d'altres àrees i sectors ja que la principal funció és avisar de problemes referents a les dades d'origen.

Tenint en compte tot això, l'objectiu del projecte és dissenyar i implementar un sistema d'alertes a temps real per donar suport a l'empresa respecte a la recepció de dades massives externes, com a funcionalitats i objectius:

- La funció principal de l'aplicació de sistema d'alertes seria donar un avís a l'empresa tecnològica cada cop que les dades rebudes de l'hospital no siguessin les adequades. És a dir, si no arribessin a temps o no es corresponguessin amb el format i el contingut esperat.
- Que el sistema d'alertes doni l'avís mitjançant l'aplicació de missatgeria immediata que utilitza actualment l'empresa. En aquest cas es va fer Slack. [Ref. 1]
- Que en un futur es puguin fer servir diferents canals de comunicació. Per exemple el correu electrònic o el Telegram, ja que són aplicacions més populars.

- Que s'implementi també mitjançant un canal restringit, on només cert personal de les empreses tinguin accés. En aquest cas seria mitjançant el Zabbix. [Ref. 2]
- Implementar noves funcions en el codi de l'empresa per evitar l'ús de noves aplicacions externes. D'aquesta manera Amalfi pot modificar el tipus d'alerta sense haver de tenir coneixements del funcionament d'una aplicació externa, aconseguint poder modificar-les o crear-les directament mitjançant codi.
- Realitzar el sistema d'alertes en dos procediments diferents, el primer seria mitjançant una aplicació externa i el segon seria amb l'ús d'unes funcions en el codi de l'empresa.

En l'article podem trobar diferents apartats i subapartats on s'explica el disseny, el desenvolupament i la verificació del correcte funcionament dels dos procediments, així com les conclusions i les millores que es podrien fer del treball i els agraïments.

## 2 DESENVOLUPAMENT

El primer disseny a desenvolupar va ser mitjançant una aplicació externa, tant de l'hospital com de l'empresa. Aquesta tindria la funció de pont, la qual comunicaria les dues empreses i abans que les dades arribessin feria de filtres d'aquestes. És per aquest motiu que a l'hora d'implementar tot això s'havia de fer servir diferents eines.

La primera va ser l'aplicació Postman, ja que permetia realitzar proves API. A part, a més a més, s'havia d'utilitzar l'aplicació GitLab [Ref. 3], on es guardava tot el codi de l'empresa. La raó principal per fer servir aquesta aplicació va ser poder fusionar Postman amb l'empresa per tal de poder rebre les notificacions que detectava aquesta aplicació al trobar un problema en les dades rebudes. A més a més, es va haver d'utilitzar altres aplicacions com el Portainer [Ref. 4] o d'altres d'ús intern de l'empresa, on tenen dades pròpies dels seus servidors i dels seus serveis.

En el segon disseny, amb l'objectiu de no utilitzar aplicacions externes, es va implementar unes funcions en el mateix codi de l'empresa per crear un filtre de les dades rebudes. En aquest cas només es va utilitzar una nova aplicació ja que Gitlab s'havia utilitzat anteriorment. En aquest cas es va fer servir Visual Studio Code [Ref. 5] per poder modificar i afegir codi al ja existent, en ser un editor de codi independent.

Finalment, la metodologia que es va seguir, tant a l'hora de dissenyar i implementar l'aplicació, com les funcions en el codi de l'empresa fou la mateixa, ja que l'objectiu no canviava i només ho feia el mètode.

Aquesta metodologia va consistir en crear un event per cada alerta diferent. Un cop creada l'alerta s'havia de relacionar amb l'event esmentat. A l'alerta, s'especificaven les condicions que es volia revisar, és a dir, quin filtre volíem implementar. A continuació es va haver de crear una connexió i finalment una subscripció. Es podrà veure tot el procediment detalladament més endavant, a l'hora de explicar la implementació.

### 2.1 Arquitectura de les dues solucions

Com s'ha comentat en els objectius del treball, hi ha dos maneres d'implementar el sistema d'alertes. Per tant, en aquest cas hi ha dos dissenys a desenvolupar.

Per una banda tenim l'aplicació, que és externa a l'empresa i a l'hospital. La qual farà de pont i tindrà com a funció fer de filtre independentment a l'empresa.

Per l'altre banda tenim les funcions integrades al codi de l'empresa, que també farà de filtre però aquest serà intern.

Els dos casos tenen el mateix objectiu i seran realitzats per a la mateixa empresa, ja que és la que està interessada en filtrar les dades per aconseguir el màxim de qualitat d'aquestes i poder realitzar el seu treball adequadament.

#### 2.1.1 Solució 1

En aquest cas, tenim una aplicació que reb dades massives d'una empresa, en el nostre cas del sector sanitari, en concret d'un hospital. Tot i en un futur podria utilitzar-se en qualsevol altre sector.

En una primera instància l'aplicació ha de revisar si li ha arribat les dades, ja que l'objectiu de l'empresa és fer prediccions a temps real i és necessari rebre-les cada un cert temps. Aquest rang de temps està acordat per part de les dues institucions, ja que dependrà de la freqüència en que l'hospital vol rebre les prediccions. Per exemple, per l'àrea d'urgències potser els interessa tenir-les casa mitja hora, però en canvi per saber les baixes dels treballadors ja els hi va bé tenir les prediccions un cop a la setmana. Tot dependrà dels interessos que tingui l'hospital.

A continuació, l'aplicació notificarà a l'empresa si no ha estat així, és a dir, si no ha rebut cap document amb les dades per part de l'hospital. Si pel contrari si que li han arribat les dades, no enviarà cap alerta i l'empresa podrà començar a treballar amb elles a partir d'intel·ligència artificial.

Com es pot veure en la *Figura 1*, aquest és el disseny del sistema d'alertes fent servir una aplicació externa.

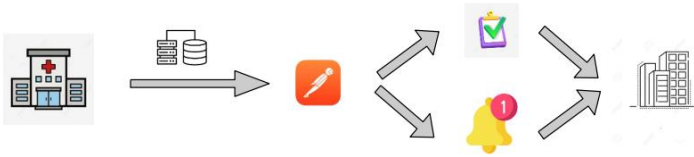


Figura 1. Disseny del procés mitjançant l'ús de l'aplicació

La Figura 1 representa tot el procés que realitzen les dades abans de poder treballar amb elles. Com es pot veure, no només es podria fer servir pel sector sanitari sinó que també es podria utilitzar en altres.

És evident que moltes empreses a dia d'avui treballen amb dades obtingudes per altres i pot ser el cas que necessitin garantir la qualitat d'aquestes dades.

### 2.1.2 Solució 2

En aquest cas, al no tenir una aplicació la qual fagi de filtre abans de que les dades arribin a l'empresa, el que es va dissenyar fou unes funcions que fessin les mateixes funcions però implementades en el codi de l'empresa. És a dir, tenir funcions en el codi propi d'Amalfi que avaluïn l'estat de les dades i l'arribada d'aquestes. També amb l'objectiu de detectar errors i notificar aquests.

A continuació, en la Figura 2 podem veure el procés mitjançant l'ús de les funcions. Aquesta mostra com les dades que reb l'empresa per part de l'hospital arriben directament a la seva base de dades i és el propi codi de l'empresa el que ha d'assegurar-se que les dades són correctes.

Això fa que s'hagi de modificar el codi propi de l'empresa amb el perill que això comporta, tot i que avui dia hi ha moltes maneres de garantir la seguretat del codi i de les dades d'una empresa com és Amalfi Analytics. Si es fa de manera externa com és en l'altre cas, no hi ha cap perill.

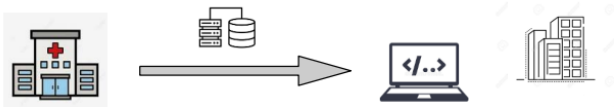


Figura 2. Disseny del procés mitjançant funcions

## 2.2 Implementació de les dues solucions

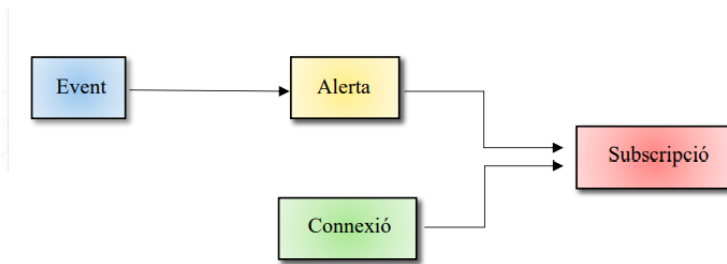


Figura 3. Implementació mitjançant un diagrama

Per tal de dur a terme la implementació de les dues solucions, s'ha seguit el diagrama mostrat anteriorment en la Figura 3.

En ell es pot veure que els mòduls més importants a tenir en compte són aquests quatre: l'Event, l'Alerta en si, la Connexió posterior que es farà de l'alerta amb l'empresa i la Subscripció d'aquesta per emmagatzemar-la.

Per tal de realitzar adequadament la implementació, tant amb l'ús de l'aplicació com amb les funcions, hem de seguir els mateixos passos, els quals són els següents.

1r) S'ha de crear un event en el qual s'ha de definir una clau i un valor per tal d'identificar-lo. En el cas de notificar la manca de dades en l'arribada de documents o d'un problema en el format o contingut de les dades.

2n) S'ha de crear una alerta, la qual s'ha de relacionar amb l'event anteriorment esmentat. És per aquest motiu que abans l'hem hagut de definir amb una clau i un valor.

A l'hora de crear aquesta alerta hem de especificar les condicions que tindrà, és a dir, programar en quin cas l'alerta s'activa i crea un avís que indica que hi ha un problema en les dades rebudes. Com a objectiu principal de l'empresa, l'alerta que s'ha creat en primer lloc ha estat la falta de dades en un període de temps.

En el cas específic de l'aplicació externa s'ha fet mitjançant la comanda Flatline que indica que no rep els documents esperants en un període de temps.

3r) Posteriorment s'ha de crear una connexió on s'ha d'indicar mitjançant quin canal vol enviar l'avís. En aquest cas, en la primera implementació s'ha especificat que el canal pel qual s'ha d'enviar l'avís ha de ser el Slack, tant a l'aplicació externa com a les funcions. Com he comentat abans aquesta és l'aplicació que fa servir l'empresa com a mitjà de comunicació.

4rt) Finalment, per poder rebre la notificació de l'alerta correctament i que així tot sigui funcional, el que s'ha de fer és crear una subscripció.

És a dir, s'ha de relacionar la connexió amb l'alerta. Això es fa mitjançant els seus corresponents id's els quals s'han obtingut a l'hora de crear anteriorment tant l'alerta com la connexió.

### 3 RESULTATS

A dia d'avui, l'aplicació ja està funcionant correctament i ja està implementada la primera alerta, la qual és la que indica que no ha arribat cap document per part de l'hospital amb les dades en un període de temps estimat.

Fer això era la part més difícil d'aconseguir ja que s'havia de connectar l'aplicació amb l'empresa, és a dir, amb el seu codi intern. Aquest però, no estava adequat per treballar conjuntament amb l'aplicació, així que un cop resolt aquest problema s'ha pogut establir la connexió de l'empresa amb l'aplicació.

Com a conseqüència, actualment només s'han de tenir uns coneixements de com funciona aquesta aplicació externa per tal de poder programar de manera ràpida i eficient noves alertes.

#### 3.1 Tests

Per tal de poder garantir el bon funcionament dels sistemes d'alertes implementats, s'han fet una sèrie de proves.

##### 3.1.1 Tests amb l'aplicació externa

En el cas de l'aplicació externa es va haver de fer diferents proves, ja que per cada procediment que es feia s'havia d'assegurar la qualitat de la feina.

És per aquest motiu que abans de poder comprovar si l'alerta responia bé a la manca de dades varem verificar que la connexió estava ben establerta. Podia ser que no s'envies cap notificació perquè les dades si que havien arribat correctament, però també perquè la connexió que s'havia fet no estava bé i per tant no es podia notificar el problema.

És per aquest motiu que les primeres proves que es van fer van ser referents a la connexió i no pas a ninguna alerta en concret. És va poder confirmar que la connexió estava ben establerta ja que en una de les aplicacions amb la qual l'empresa treballa marcava que s'havia creat una connexió nova. Això garantitzava que si no arribava cap notificació per part de l'aplicació significava que les dades estaven arribant a l'hora que es demanava.

Tot i això, el primer test que es va fer va ser si l'alerta s'havia creat bé. És a dir, si la condició d'arribada de dades s'havia implementat correctament.

Com encara no s'havia creat la connexió de l'aplicació amb l'empresa i no li passàvem manualment cap base de dades l'aplicació havia de donar l'avís.

En un primer moment se li va posar com a condició que el període fós de 2 minuts. D'aquesta manera es podia veure si les notificacions arribaven correctament i sense problemes al cap del temps.

A continuació se li va augmentar el període de temps a 15 minuts i al veure que donava l'avís ja que no se li passava cap document es va incrementar aquest rang de temps.

Finalment, es va modificar la condició a un període de 30 minuts. Era el temps que es va acordar amb l'hospital per tal de rebre les prediccions.

Un cop es van solventar les alarmes de recepció de dades, només es va haver d'activar la connexió amb l'empresa perquè des d'aquell moment, les notificacions que arribessin serien com a conseqüència de la manca de dades enviades per part de l'hospital a temps real.

A continuació, en la *Figura 4*, podreu veure una captura de pantalla de les proves que es van fer abans d'establir la connexió entre totes les parts. En ella es mostra que hi ha avisos per la manca de dades obtingudes.

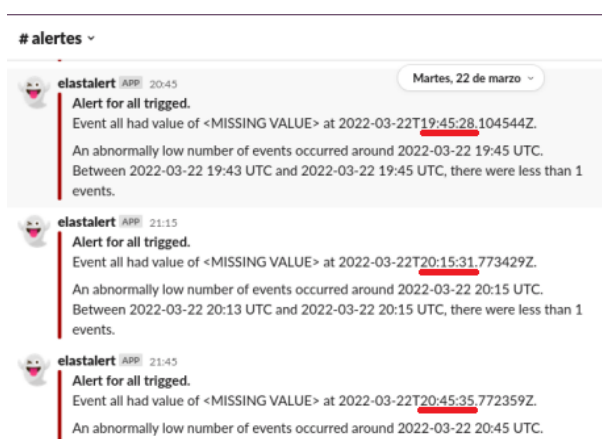


Figura 4. Prova de condicions cada 30 minuts

##### 3.1.2 Tests amb les funcions

En el cas de les funcions implementades dins del codi, es van fer unes proves i tests diferents als anteriors.

En aquest cas hi havia 6 tests per garantir el bon funcionament. Mitjançant Visual Studio Code,

es va anar fent les proves que consistien en executar i “fer correr” tots els tests.

En aquest cas era el propi terminal de l'aplicació el que mostrava si les funcions havien passat els tests de manera correcta.

En el cas que passessin les condicions d'un test, es marcava aquest amb un color verd i la paraula “Passed”. D'altra banda, en el cas que les funcions no haguessin passat els tests es notificava amb un “Failed”.

A mesura que s'anaven modificant les funcions, i per tant s'anaven millorant, hi havia més tests que es marcaven amb un “Passed”.

L'objectiu era que totes les funcions superessin els tests, d'aquesta manera es podia confirmar que les funcions implementades estaven correctament programades.

A continuació, en la *Figura 5*, es pot veure el terminal de l'aplicació Visual Studio on es mostra que les funcions han passat un test correctament.

```
unittest (testing)..... Passed
coverage (testing)..... Passed
- hook id: coverage
- duration: 0.27s
```

*Figura 5. Test de les funcions*

A continuació, com es pot observar en la *Figura 6*, en aquest cas tenim el terminal de l'aplicació on es poden veure que les funcions no han passat tots els tests. Així doncs, caldrà modificar les funcions perquè en un futur, un cop validades i testades, notifiquin els errors mitjançant les alertes.

```
flake8 (linting)..... Failed
- hook id: flake8
- exit code: 1
```

*Figura 6. Test de les funcions*

### 3.2 Situació actual

A dia d'avui, hi ha dues situacions en el desenvolupament dels sistemes d'alertes esmentats anteriorment.

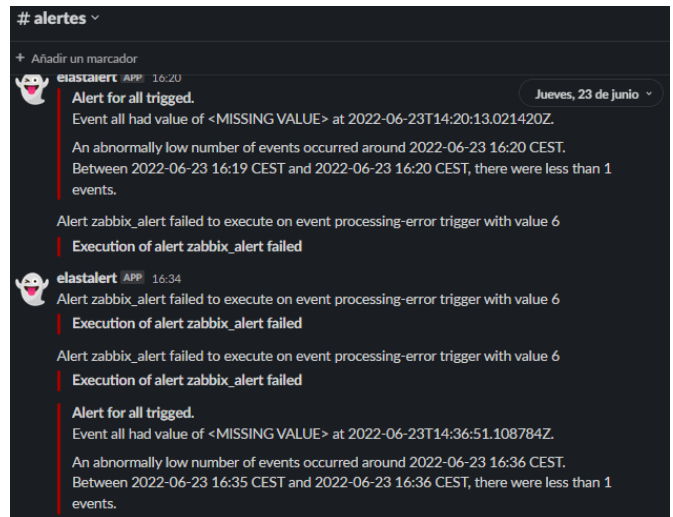
La primera situació que fa referència a l'aplicació externa podem afirmar que actualment funciona correctament.

Respecte les alertes creades a l'aplicació no hem tingut cap avís referent a la manca de dades rebudes per part de l'hospital. Això implica que al llarg del darrer

mes l'hospital ha enviat a temps real les seves dades abans dels 30 minuts que marca l'alarma.

Encara que no s'ha rebut cap notificació, s'han continuat fent proves per tal de tenir un seguiment de l'aplicació i verificar el seu correcte funcionament.

L'última prova, tal i com es pot veure a la *Figura 7*, que s'ha fet al respecte el dia 23 de juny del 2022 a les 10:45:00h.



*Figura 7. Seguiment alerta*

La segona situació que fa referència a les funcions, a dia d'avui encara està en procés de modificació. Actualment hi ha quatre tests que les funcions passen correctament, però n'hi ha dos que encara no passen els tests i que estan en procés de modificació.

## 3 CONCLUSIONS I MILLORES

La principal conclusió a la qual he arribat ha estat adonar-me que tenir un sistema d'alertes en una empresa és molt útil i necessari.

No només en una empresa tecnològica que treballa amb intel·ligència artificial a temps real, com pot ser el cas d'Amalfi Analytics, si no per a qualsevol empresa que treballi amb gran volum de dades que no depenen d'ells.

A dia d'avui es generen una gran quantitat de dades a nivell mundial les quals s'han anat emmagatzemant molts cops sense ningun tipus de criteri. A l'hora de treballar amb elles, s'ha de dedicar uns recursos addicionalment per tal d'aconseguir una bona qualitat en les dades. És per aquest motiu que trobo necessari i molt adient tenir una forma d'assabentar-se de les irregularitats, com és mitjançant un sistema d'alertes.

És veritat que actualment hi ha varies empreses que proporcionen ajuda amb les dades. És a dir, que notifiquen de certs problemes, però tot i això encara no hi ha en el mercat ninguna aplicació com la creada a Amalfi Analytics.

L'aplicació està directament dissenyada per a diferents tipus de problemes, com he comentat anteriorment, i a més pot adaptar-se a qualsevol sector tecnològic independentment del mercat on treballi.

El procés fins a obtenir el servei actual no ha estat fàcil, ja que ha sorgit certs problemes. El principal problema on he dedicat més temps, ha estat el fet de resoldre el fet d'establir la connexió entre l'empresa i l'aplicació. Com he comentat anteriorment es va haver de modificar i afegir codi en el codi actual per tal de poder fer la connexió. Un dels motius de la complexitat per resoldre aquest problema, ha estat no haver previst amb anterioritat en el codi de l'empresa futures connexions externes del tipus que ara es creaven. Això provocava bucles que no contemplaven les opcions esperades.

Considero que s'ha fet molt bon treball encara que, lògicament, hi ha punts a millorar o afegir en un futur.

- Una de les millores que personalment crec que podrien ser útils per a l'empresa és que la notificació no arribi només a ella, si no que també s'envii a l'hospital. D'aquesta manera l'empresa podria donar per notificat l'error a l'hospital en rebre l'avís.
- Un millora que jo afegiria en una nova versió seria que l'aplicació a part de notificar l'error, enviés també un missatge. En el cas de la manca de dades potser seria innecessari ja que al rebre la notificació el missatge és clar, "no s'ha enviat les dades", però en el cas d'errors administratius es podria enviar el missatge demanant la comprovació d'alguna dada en concret, per saber si es tracta d'un error o pel contrari és un cas excepcional. D'aquesta manera l'hospital podria prendre mesures per corregir els errors que genera en un futur.

## AGRAÏMENTS

En primer lloc voldria agrair aquest treball a en Martí Zamora, treballador d'Amalfi Analytics, que m'ha ajudat en tot moment amb els problemes i dubtes que he tingut. Ha estat la persona amb la que he passat més hores treballant en el període de pràctiques externes, i que m'ha servit de base per desenvolupar el TFG.

En segon lloc voldria també agrair a en Enric Martí, professor de la Universitat Autònoma de Catalunya la dedicació i interès que ha mostrat en el seguiment de les pràctiques externes, així com del TFG durant tots aquests mesos de treball.

En tercer lloc, i no per això menys important, m'agradaria agrair a Amalfi Analytics l'oportunitat d'haver estat treballant a la seva empresa durant tots aquest mesos. Aquest agraïment no és només pel fet de poder realitzar les pràctiques externes a la seva seu, si no també per la predisposició en poder fer el meu TFG fent referència al seu projecte de sistema d'alertes. Sempre estaré agraïda del tracte i la confiança donada des del primer moment.

## BIBLIOGRAFIA

A continuació es pot observar la bibliografia del treball:

[Slack]  
<https://elastalert2.readthedocs.io/en/latest/ruletypes.html#slack>, Tipus de regles i opcions de configuració. Es poden trobar exemples de diversos tipus de configuració de regles a la carpeta exemples/regles. (data últim accés: Juny 2022)

[Soil SDK]  
<https://developer.amalfianalytics.com/docs/tutorial/alerts>, Tutorials de com crear o com funcionen les alertes. (data últim accés: Juny 2022)

[WinPlus]  
<https://www.iest.com/gestion-de-alertas/>, pàgina web de l'empresa WinPlus Alertas. (data últim accés: Maig 2022)

[GitLab]  
<https://gitlab.com/amalfianalytics/deployments/dev/dev-soil/-/blob/main/alerterconfig/config.yaml>, Repositori (data últim accés: Juny 2022)

[Portainer]  
<https://portainer.yard.amalfianalytics.com/#!/4/docker/containers>, Portainer (data últim accés: Juny 2022)

[Elastic]  
[https://www.elastic.co/es/?ultron=B-Stack-Trials-EMEA-S-Exact&gambit=Stack-Core&blade=adwords-s&hulk=paid&Device=c&thor=elasticsearch&gclid=CjwKCAiAlfqOBhAe-EiwAYi43F55DmVmWzrMle1b-L5XCZL7OQs-82RAm92\\_VfuYZDOvy1VwfK6bafxoCaxEQAvD\\_BwE](https://www.elastic.co/es/?ultron=B-Stack-Trials-EMEA-S-Exact&gambit=Stack-Core&blade=adwords-s&hulk=paid&Device=c&thor=elasticsearch&gclid=CjwKCAiAlfqOBhAe-EiwAYi43F55DmVmWzrMle1b-L5XCZL7OQs-82RAm92_VfuYZDOvy1VwfK6bafxoCaxEQAvD_BwE), Elastic (data últim accés: Juny 2022)

[AmalfiAnalytics]  
<https://auth.amalfianalytics.com/password/forgot?tenantId=55a76de6-727f-40bb-b93a-3fc468a6ec7c>, Amalfi Analytics (data últim accés: Juny 2022)

[AmalfiAnalytics - web]  
<https://amalfianalytics.com/>, Web d'Amalfi Analytics (data últim accés: Juny 2022)



## APÈNDIX

### A1. REFERÈNCIES

[Ref. 1] Slack: És una aplicació de missatgeria per a empreses que connecta les persones amb la informació que necessiten. Slack transforma la manera com es comuniquen les organitzacions reunint les persones perquè treballin com un equip unificat

[Ref. 2] Zabbix: És una eina de programari lliure per a monitorització de xarxes. Permet monitoritzar i registrar l'estat de diversos serveis a una xarxa d'ordinadors, servidors o maquinari de la pròpia xarxa.

[Ref. 3] GitLab: És una aplicació basada en la web amb una interfície gràfica d'usuari que també es pot instal·lar en un servidor propi. El nucli de GitLab el formen els projectes en què es desa el codi que s'editarà en arxius digitals, els anomenats repositoris.

[Ref. 4] Portainer: És una eina lleugera i open-source de gestió de contenidors sobre Docker (o Docker Swarm). Portainer ofereix una interfície gràfica per gestionar el host Docker des de qualsevol navegador, té suport per a Raspberry Pi i es pot desplegar com un simple contenidor.

[Ref. 5] Visual Studio Code: És un editor de codi font independent que s'executa a Windows, macOS i Linux. L'elecció principal per a desenvolupadors web i JavaScript, amb una gran quantitat d'extensions per admetre gairebé qualsevol llenguatge de programació