

## Treball final de grau

**Estudi:** Grau en Enginyeria Informàtica

**Títol:** Kalinfo: eina per a l'atenció telefònica de Som Energia

**Document:** Memòria

**Alumne:** Marta Fernández Vergés

**Tutor:** Josep Soler i Masó

**Departament:** Departament d'Informàtica

**Àrea:** Matemàtica Aplicada i Estadística

**Convocatòria (mes/any):** 09/2020

# **Kalinfo:** eina per a l'atenció telefònica de **Som Energia**

---

**Treball de Final de Grau**

**Curs 2019-2020**

**Alumne:** Marta Fernández Vergés

**Tutor:** Josep Soler i Masó

**Grau en Enginyeria informàtica – Universitat de Girona**

# Contingut

1. Introducció, motivacions, propòsits i objectius del projecte .....	6
1.1. Som Energia .....	6
1.2. Motivacions .....	7
1.3. Necessitat del projecte .....	7
1.4. Objectius: .....	10
2. Estudi de la viabilitat .....	11
2.1. Viabilitat tecnològica .....	11
2.2. Viabilitat en Recursos Humans .....	11
3. Metodologia .....	12
3.1. Metodologia de planificació .....	12
3.2. Metodologia de desenvolupament .....	13
4. Planificació .....	14
4.1. Paquets de treball .....	14
4.1.1. Documentació .....	14
4.1.2. Frontend .....	14
4.1.3. Comunicació .....	15
4.1.4. API .....	15
4.1.5. Centralita .....	15
4.1.6. Gestió .....	16
4.1.7. Backend .....	16
4.2. Comparativa esforços .....	16
4.3 Evolució .....	17
5. Marc de treball i conceptes previs .....	19
5.1 Persones que col·laboren .....	19
5.2 TDD i Unit testing .....	19
5.3 Llenguatges de programació i frameworks .....	20
5.3.1 Python .....	20
5.3.2 JavaScript .....	21
5.3.3 Flask .....	22
5.3.4 Mithril .....	22
6. Requeriments del sistema .....	23
6.1 Context .....	23
6.2 Requisits funcionals .....	24
6.3 Requisits no funcionals .....	26
6.4 Requisits per part de l'empresa .....	26

6.5 Requisits de llicència.....	27
7. Estudis i decisions .....	28
7.1 Websockets .....	28
7.2 Polythene.....	28
8. Anàlisi i disseny del sistema .....	30
8.1 Anàlisi .....	30
8.1.1 Actors del sistema .....	30
8.1.2 Diagrames de casos d'ús .....	30
8.1.3 Fitxes de casos d'ús.....	33
8.2 Disseny del sistema .....	38
8.2.1 Situació eina .....	38
8.2.2 Fluxe del procés .....	39
8.2.3 Model de dades.....	41
8.2.4 Disseny de interfície.....	45
9. Implementació i proves .....	48
9.1 Connexions websockets .....	48
9.2 Extreure la informació.....	49
9.3 Creació dels casos .....	54
9.3.1 Creació de casos trucada telefònica.....	54
9.3.2. Creació de casos d'atenció client .....	54
9.4 Script per creació de casos .....	56
9.5 API .....	57
9.6 Desenvolupament del projecte .....	72
9.7 Proves realitzades .....	72
10. Implantació i resultats .....	73
10.1 Implantació.....	73
10.2 Resultats .....	73
11. Conclusions .....	83
12. Treball futur .....	84
13. Webgrafia.....	85
14. Annexos.....	86
14.1 Planificació detallada .....	86
Ronda 0 (inici a 25-03-2019): .....	86
Ronda 1 (26-03-2019 a 12-04-2019):.....	88
Ronda 2 (13-04-2019 a 07-05-2019):.....	90
Ronda 3 (08-05-2019 a 28-05-2019):.....	92
Ronda 4 (03-06-2019 a 24-06-2019):.....	98

Ronda 5 (27-06-2019 a 09-07-2019):.....	107
Ronda 6 (17-02-2020 a 28-02-2020):.....	113
Ronda 7 (02-03-2020 a 27-03-2020):.....	116
Ronda 8 (30-03-2020 a 20-04-2020):.....	120
Ronda 9 (21-04-2020 a 01-05-2020):.....	125
Ronda 10 (06-05-2020 a 05-06-2020):.....	127
Ronda 11 (06-06-2020 a 26-06-2020):.....	132

# 1. Introducció, motivacions, propòsits i objectius del projecte

Aquest Treball de Final de Grau s'ha realitzat a la cooperativa Som Energia, actualment situada al Parc Científic i Tecnològic de la Universitat de Girona.

Per a la cooperativa, l'atenció telefònica és una tasca molt important. Requereix visualitzar informació provinent de diferents entorns i això provoca problemes de rendiment i alenteix l'atenció al client. Així doncs, poder agrupar tota aquesta informació per tal de poder-la visualitzar de manera ràpida des d'una única font, va originar la idea del que acabaria essent el *Kalinfo*.

## 1.1. Som Energia

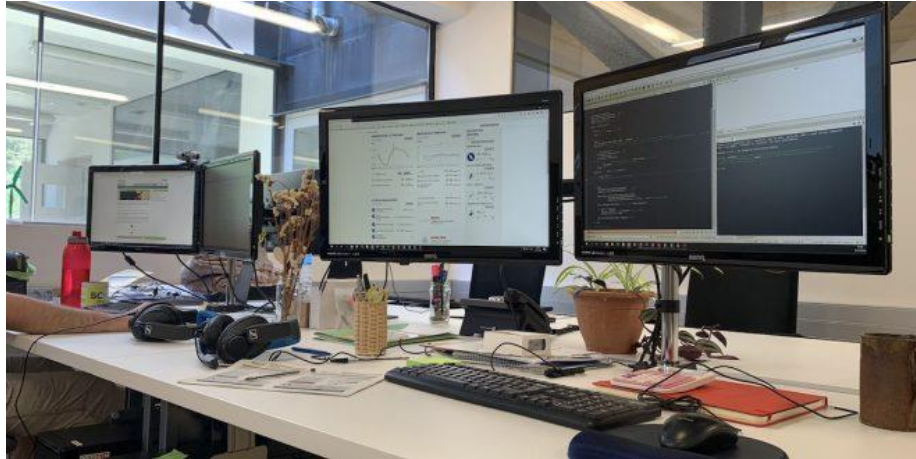
Som Energia SCCL (Societat Cooperativa Catalana Limitada) és una cooperativa catalana de producció i consum d'energies renovables amb seu al Parc Científic i Tecnològic de la Universitat de Girona. És una organització sense ànim de lucre que va néixer l'11 de desembre de 2010 a Girona.

Té com a objectiu principal el consum i la producció d'energia a partir de fonts 100% renovables per tal de modificar el model energètic actual i aconseguir un model completament renovable. Al començament, va iniciar com un projecte de participació ciutadana per canviar el model energètic. Durant l'any 2011 va començar a comercialitzar electricitat, sempre amb certificats de procedències renovables, i es van iniciar els primers projectes de generació d'energia renovable.

Amb els anys, Som Energia ha impulsat idees innovadores com el Generation kWh, que permet l'autoconsum compartit sense la necessitat d'instal·lar plaques solars a casa.

Degut al seu creixement, s'estan experimentant diversos canvis pel que fa a la organització i assignació de rols a la cooperativa, però l'estructura de Som Energia està principalment compost per quatre entitats:

- Consell Rector, escollit pels socis. S'encarrega d'implementar les directrius marcades per l'assemblea tot respectant els valors recollits en els estatuts de la cooperativa.
- Gerència, consistent en dues persones que gestionen la relació entre el Consell Rector i el dia a dia de l'Equip de Treball.
- Equip de treball, que s'encarrega de l'activitat empresarial de la cooperativa. Actualment hi ha 88 treballadors.
- Persones sòcies, que mitjançant votacions escullen les directrius d'actuació de la cooperativa. Cal destacar la figura dels grups locals, persones sòcies que, segons el seu lloc geogràfic realitzen activitats i xerrades sobre la temàtica del model energètic actual, energies renovables, compromís social i altres temàtiques en concordança amb la cooperativa.



*Oficines de Som Energia*

## **1.2. Motivacions**

El desenvolupament d'aquesta eina és una bona oportunitat per ampliar el meu ventall de coneixement. Formo part de l'equip de desenvolupament web i, per altra banda, no he treballat amb massa profunditat pel que fa al desenvolupament de mòduls per a l'ERP de la cooperativa.

La presa de requeriments, disseny de l'aplicació, desenvolupament Backend i Frontend permetran que pugui treballar amb tecnologies que fins ara no havia tingut oportunitat.

A més, com he mencionat anteriorment, l'atenció telefònica té un pes molt important per a la cooperativa. És molt valorada per a les sòcies degut a que, a diferència d'altres empreses, la interacció sempre és humana, a excepció del contestador quan totes les línies esta ocupades. Poder agilitzar aquesta tasca que es realitza cada dia de manera que faciliti a les persones treballadores la informació necessària per poder atendre bé les trucades, és una tasca important que no s'ha pogut prioritzar fins ara.

No hi ha un equip fixe dedicat a la atenció de telèfon. És una tasca que fan cada dia de 9.00 h a 14.00 h diferents persones de diferents equips i de diferents àmbits. Per aquest motiu, quan els equips tenen massa feina, es dificulten tasques que venen implícites amb l'atenció telefònica: enregistrar les trucades rebudes amb els motius d'aquestes i obrir casos d'atenció al client al ERP de la cooperativa quan la trucada ha estat una reclamació.

Aquesta feina que a vegades suposaria una sobrecàrrega es pot veure força alleujada si es pot fer tot de manera quasi automàtica des de la pàgina que utilitzen per a la gestió dels torns de telèfon: el *Tomàtic*.

## **1.3. Necessitat del projecte**

A Som Energia, actualment, hi ha una pàgina interna que anomenem *Tomàtic* que permet gestionar els torns de telèfon. Hi ha un total de quatre pestanyes, tres d'elles actives:

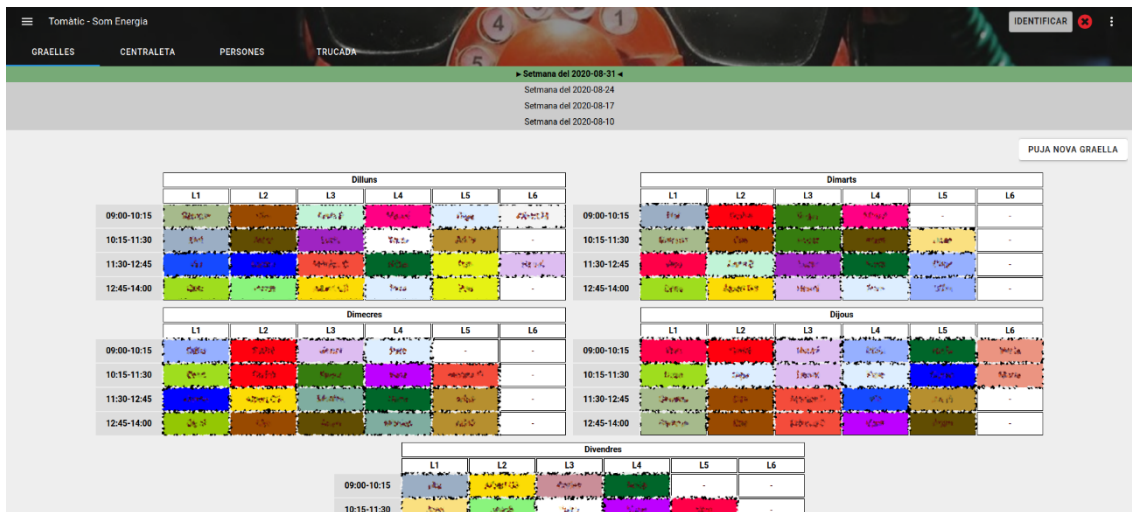
GRAELLES: Es poden veure les graelles amb els horaris dels torns d'atenció telefònica (que varien cada setmana)

LÍNIA: La línia activa que està atenent trucades (es poden fer pauses i afegir-hi gent)

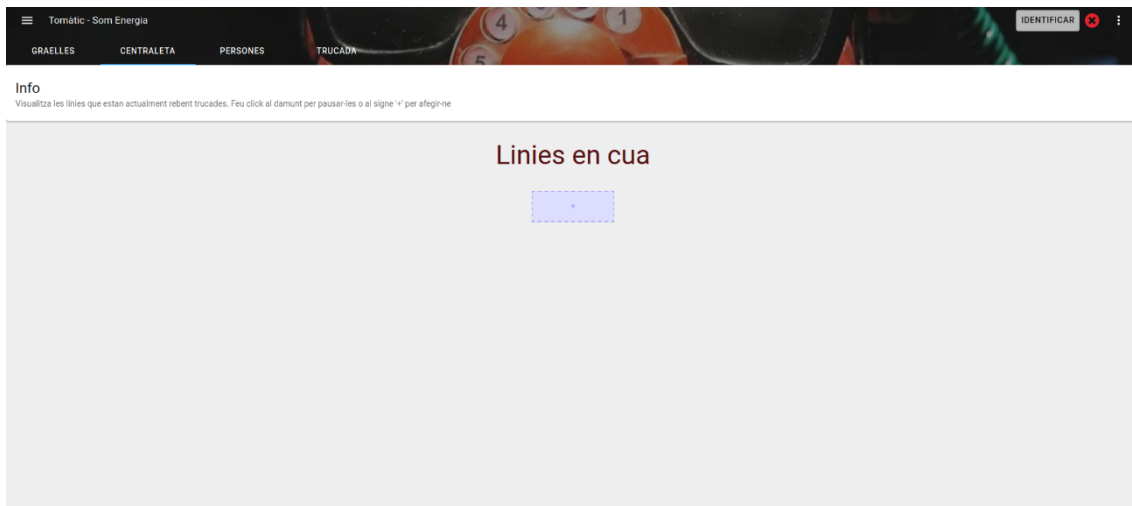
PERSONES: Les persones que poden fer atenció telefònica amb la seva corresponent informació (número d'extensió, color...)

TRUCADA: Pàgina en blanc.

**Els responsables de Som Energia han considerat que es necessita una eina que permeti fer un tipus d'atenció més personalitzada i eficient, de manera que es pugui consultar tota la informació des d'un mateix lloc.** Aquesta nova eina, a la que anomenem KALINFO, és la que es desenvoluparà en aquest TFG i s'hi podrà accedir a través de la pestanya 'Trucada' del Tomàtic.

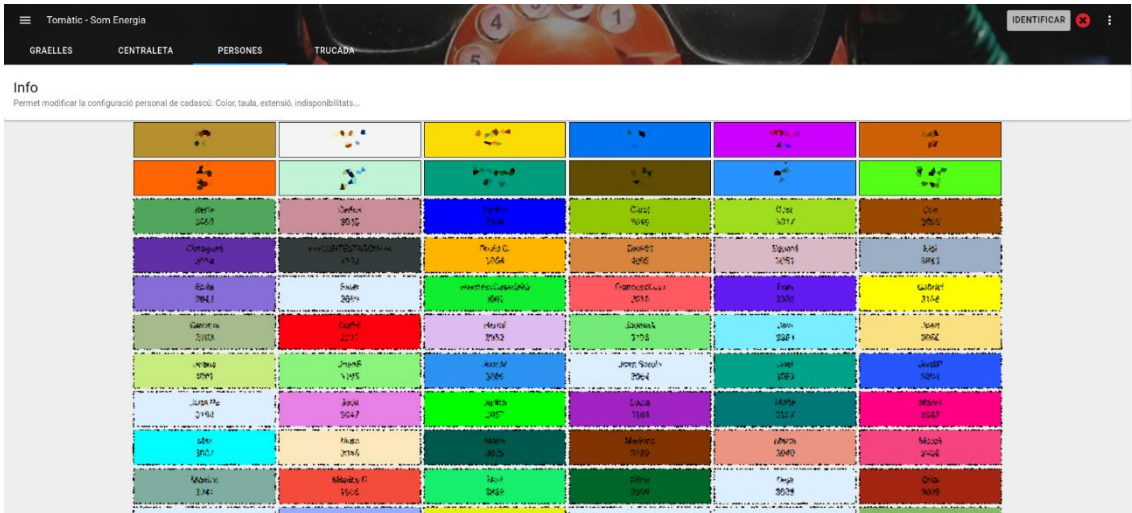


*Graelles del Tomàtic*

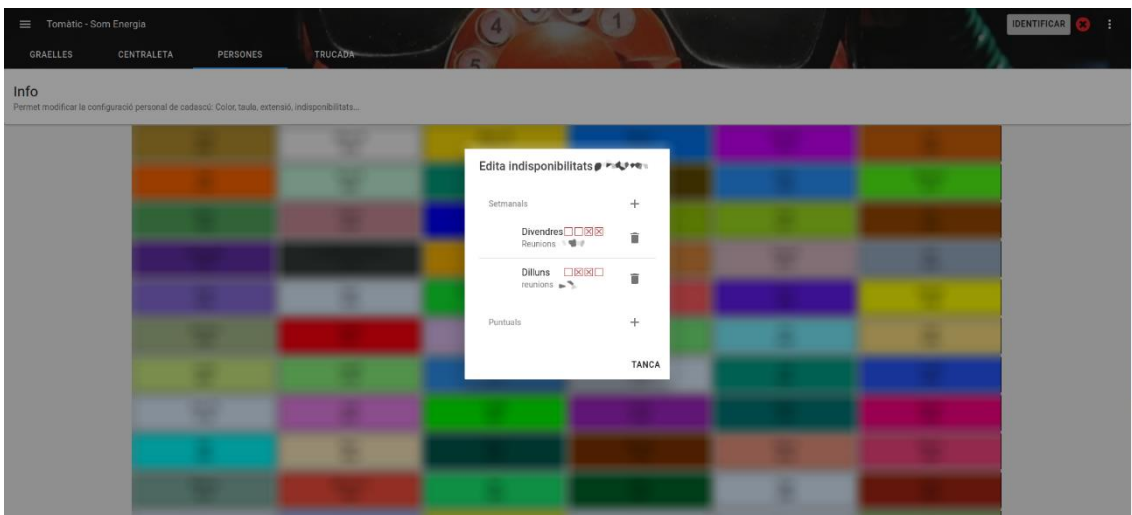


*Línies de cua actives del Tomàtic*

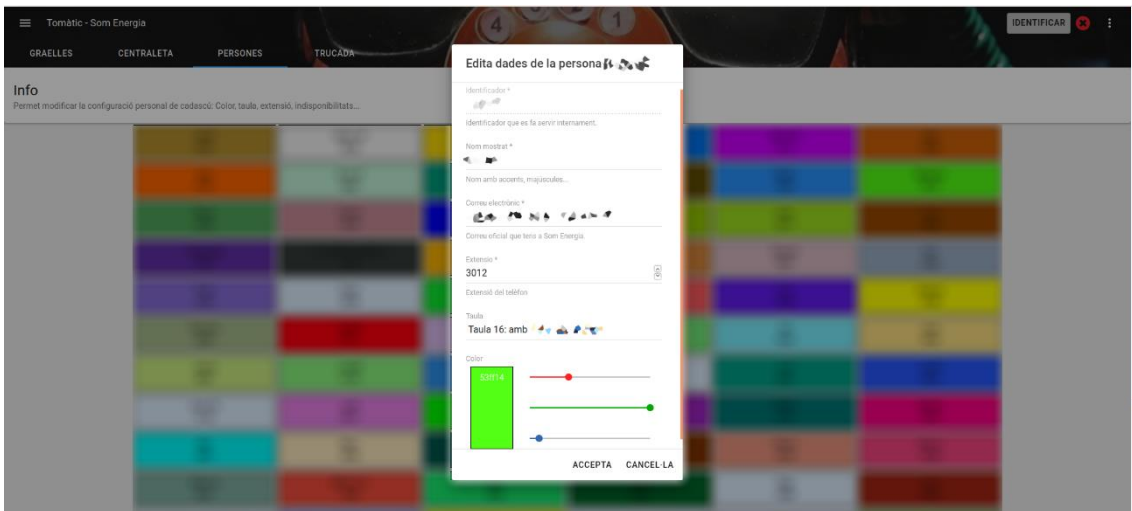




*Persones del Tomàtic*



*Indisponibilitats de persones del Tomàtic*



*Dades de persona del Tomàtic*

## 1.4. Objectius:

**L'objectiu principal d'aquest TFG consistirà en desenvolupar l'anàlisi, disseny, implementació i posada a producció d'aquesta eina per l'atenció telefònica al client de Som Energia.**

Aquesta eina permetrà:

1. Visualitzar la informació dels clients i els seus respectius contractes a través del refresc automàtic de la pàgina quan la persona que està fent atenció telefònica rebí la trucada.
2. Visualitzar la informació dels clients i els seus respectius contractes a través de cerques realitzades amb un cercador.
3. Respondre un qüestionari sobre el motiu de la trucada.
4. Respondre un qüestionari específic quan la trucada ha estat una reclamació.
5. Portar un registre de les trucades rebudes i el motiu d'aquestes.
6. Portar un registre de les reclamacions rebudes i/o gestionades a través de trucades.
7. Visualitzar el registre de trucades ateses per a la persona que està realitzant atenció telefònica.
8. Visualitzar el registre de trucades realitzades per al número que està trucant.
9. Ensenyar les dades i estadístiques que es considerin oportunes a partir del registre de trucades i la informació que es pugui extreure.



*Tomàtic del TV3*

## **2. Estudi de la viabilitat**

Per tal de desenvolupar aquest projecte, hi haurà impacte econòmic tant per els recursos humans que es necessitaran com per la complexitat tecnològica que pot suposar. Com que l'equip IT de Som Energia està intentant externalitzar el mínim els serveis informàtics, no es farà un estudi de viabilitat econòmica comparant amb alternatives d'altres softwares del mercat. A més, els responsables de la cooperativa tenen molt clara la necessitat d'aquesta eina i per això s'ha procedit al seu desenvolupament

### **2.1. Viabilitat tecnològica**

El projecte, haurà de complir unes funcions específiques. Està ben delimitat i té un objectiu clar. Desenvolupar aquesta eina internament pot permetre que sigui més personalitzat i s'adeqüi a les necessitats que l'equip que fa atenció telefònica pensa que necessita. A més, ofereix la oportunitat d'anar-se ampliant i canviant en funció del que es vagi observant. Un cop es tingui una primera versió estable, mitjançant el sistema de treball de integració contínua, es pot anar modificant i afegint/traient les funcionalitats que es creguin convenients.

A Som Energia som partidaris del Software Lliure tant pel que fa al nostre propi desenvolupament com quan hem d'escollir agents externs. Degut a aquest motiu, no hi haurà costos de desenvolupament software ni es requerirà hardware nou. Tot es desenvoluparà amb les mateixes eines amb les quals desenvolupem els altres projectes.

### **2.2. Viabilitat en Recursos Humans**

El temps de dedicació dels altres treballadors i del tutor, serà mínim. Estarà contemplat com a temps d'organització i no repercutirà en el seu flux de treball natural.

Aquest, és un projecte que interessava desenvolupar però no podien dedicar-li el temps necessari degut a altres prioritats i la mida de l'equip IT. Poder desenvolupar eines d'aquest estil com a Treballs de Final de Grau, interessa a la cooperativa, ja que permet impulsar altres eines que necessiten treballar amb una base inicial més gran de manera que els altres projectes no es vegin tan afectats i, alhora, milloren i aconseguen impulsar el desenvolupament d'altres eines que no es poden prioritzar (degut a que no hi ha un equip concret que les prioritzi, la competitivitat entre prioritats, etc).

## 3. Metodologia

La metodologia de desenvolupament determina l'estructura del projecte. En aquest apartat s'explicarà la metodologia utilitzada durant la creació del Kalinfo.

### 3.1. Metodologia de planificació

El sistema de planificació que s'utilitza a Som Energia està inspirat amb la metodologia *Scrum*, amb petites variacions i matisos que permeten una millor organització amb la resta de l'equip.

*Scrum* és una metodologia en la que s'apliquen un conjunt de bones pràctiques per tal de treballar en equip i obtenir el millor resultat d'un projecte. És una tecnologia àgil, així que permet obtenir resultats ràpid i anar treballant sobre una base funcional a través de les versions, millorant-les i corregint-les. D'aquesta manera s'augmenta la productivitat i permet obtenir millors resultats quan els requisits són canviants o poc definits, permetent molta flexibilitat.

La idea és realitzar entregues parcials i regulars del producte final, prioritzades per el benefici que aporten al receptor del projecte. Per aquest motiu, s'eliminen els casos on el client està esperant massa per la entrega d'un producte final acabat.

En *Scrum*, un projecte s'executa en cicles de durada fixe anomenats *Sprints*. Per al desenvolupament del meu projecte els anomenem *Rondes* (igual que a la metodologia de planificació general de l'equip IT de la cooperativa) i tindran una durada de 2 o 3 setmanes.

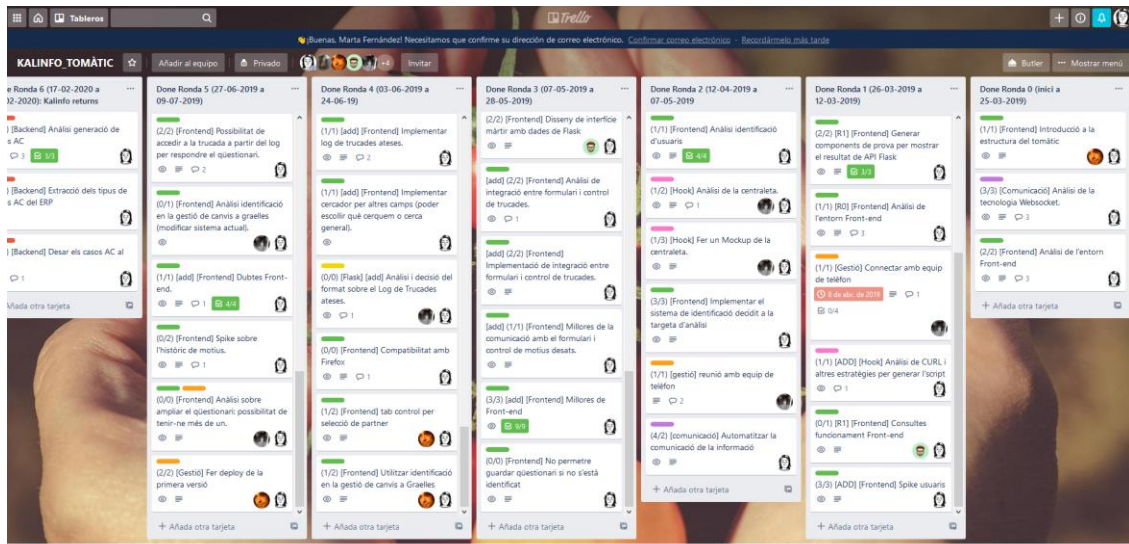
El procés inicia amb una llista d'objectius/requisits prioritzats que serviran per planificar el projecte. Aquesta llista estarà en una columna específica que anomenarem *Rebost* en un tauler del *Trello* del projecte, pàgina que utilitzarem per visualitzar les *Targetes* (objectius/requisits), i altra informació sobre el resultat i procés d'aquesta metodologia de treball.

La idea és que el representant IT (client) prioritzi les *Targetes* (objectius) balancejant el valor que aporten respecte el seu cost, per tal de determinar quines entraran a *Ronda* i quines quedaran al *Rebost*.

La manera de planificar cada *Ronda*, es farà mitjançant una reunió l'últim dia d'aquesta. En aquesta reunió, el client presenta la llista de requisits prioritzada. L'equip pregunta els dubtes i selecciona el requisits més prioritaris que preveu que podrà completar aquella *Ronda*.

Seguidament, l'equip elabora la llista de *Targetes* de la *Ronda*, necessàries per desenvolupar els requisits seleccionats. Es fa la estimació del temps que es necessitarà i s'organitza com es treballaran.

Durant la *Ronda*, cada dia es fa la *Standup*, una reunió curta amb tot l'equip IT on s'informa en què s'està treballant i possibles dependències entre tasques, per poder fer les adaptacions necessàries i seguir treballant. Es respon a què s'ha fet des de la última *Standup*, a què es farà a partir d'aquest moment i amb quins impediments em puc trobar.



Trello del Kalinfo

### 3.2. Metodologia de desenvolupament

La metodologia que s'utilitzarà per desenvolupar el projecte es basa en el control de versions amb GitHub.

La idea inicial era treballar cada desenvolupament amb la seva branca específica, que tindria el nom dels canvis concrets que s'hi realitzarien. Això permetia anar fent petites *releases* aplicant aquestes petites actualitzacions al final de cada *Ronda*. Finalment s'han acabat fent dues releases importants:

- La primera s'ha treballat en una mateixa branca separada de la branca principal (*master*) anomenada `NEW_CallInfo`. Aquesta va permetre que s'anés treballant paral·lelament amb *master* de manera que, quan es tingués una versió que permetés presentar-la a l'equip tècnic de Som Energia mostrant un mínim de les funcionalitats que es requerien, només calgués fer un *merge* amb *master*.
- La segona, contenia un *re-factor* important d'aquesta primera versió i alguns canvis en quant a la idea inicial del funcionament. Més endavant s'explicarà més detalladament el per què d'aquests canvis i en què consisteixen. Aquí s'ha treballat en dues branques. Una s'anomenava `NEW_claims_integration_with_erp` que va acabant fent *merge* amb la segona branca anomenada `WIP_kalinfo_refactor`. Sobre aquesta, es van acabar de fer els canvis del que seria la segona versió del Kalinfo. La idea és mantenir-la actualitzada amb *master* i, tot i que es posi a productiu, no fer la crida a l'equip fins que inclogui totes les funcionalitats.

Es va acabar adoptant aquesta metodologia de treball degut a que hi ha altres equips realitzant tasques sobre el mateix producte. Els representants IT del projecte, no volien presentar l'eina amb petites releases a l'equip. Com que un cop estigués finalitzada, canviaria notablement el flux de treball que s'utilitza regularment, no es va voler presentar-ho parcialment perquè des d'un inici l'equip fos conscient de la essència i el benefici de l'eina completa. Si l'eina no es presentava d'aquesta manera, el problema d'haver de consultar diferents programes per realitzar l'atenció telefònica persistia, així que no aportava massa benefici.

## 4. Planificació

Separació en paquets de treball amb el període de temps que s'hi destinarà.

### 4.1. Paquets de treball

Cada paquet de treball inclourà el temps que se li dedicarà i les tasques que inclou. A Som Energia s'utilitza la unitat de temps *Mitjos dies (MD)*. Un *Mig dia* correspondria a 4h Aquesta serà la unitat de temps que s'utilitzarà per mesurar el temps que es preveu que s'hagi de dedicar a cada paquet de treball.

Aquest paquets de treball inclouen també el temps dedicat a la realització dels tests ja que el desenvolupament serà amb *Test-Driven-Development*.

Els paquets de treball més generals seran:

- Documentació
- Frontend
- API
- Comunicació
- Centraleta
- Gestió
- Backend

#### 4.1.1. Documentació

**Descripció:** Documentació del treball realitzat per el Treball de Final de Grau i la memòria.

**Tasques:** Redacció de la memòria i anàlisi del projecte.

**A lliurar:** Memòria del Treball de Final de Grau.

**Temps:** 15 mitjos dies.

#### 4.1.2. Frontend

**Descripció:** Tot el que estigui relacionat amb la UI i la part Frontend de l'eina.

**Tasques:**

- Situar al projecte i anàlisi: 4 MD
- Cercador: 4 MD
- Components per mostrar la informació: 30 MD
- Identificació d'usuaris: 11 MD
- Qüestionari: 8 MD
- Refresc automàtic: 4 MD
- Log de trucades: 10 MD
- Casos AC: 4 MD
- Testeig de la interfície: 5 MD

**A lliurar:** Codi font funcional disponible al repositori de GitHub.

**Temps:** 80 mitjos dies.

### 4.1.3. Comunicació

**Descripció:** Anàlisi del funcionament dels websockets i implementació de la connexió entre el Tomàtic i la Centraleta per al refresc automàtic.

**Tasques:**

- Anàlisi i ús de Websockets: 3 MD
- Comunicació de la informació: 2 MD

**A lliurar:** Codi font funcional disponible al repositori de GitHub.

**Temps:** 5 mitjos dies.

### 4.1.4. API

**Descripció:** Anàlisi, disseny i implementació del codi necessari per tal de que es creïn els *endpoints* i es pugui comunicar amb l'ERP i els models de dades corresponents.

**Tasques:**

- Cercador: 7 MD
- Trucades: 2 MD
- Informació-ERP: 12 MD
- Log trucades: 1 MD
- ERP-AC: 3 MD

**A lliurar:** Codi font funcional disponible al repositori de GitHub.

**Temps:** 25 mitjos dies.

### 4.1.5. Centraleta

**Descripció:** Anàlisi de la Centraleta, Asterisk i el servidor.

**Tasques:**

- Anàlisi Centraleta: 3 MD
- Mockup Centraleta: 3 MD

**A lliurar:** Codi font funcional disponible al repositori de GitHub.

**Temps:** 6 mitjos dies.

#### 4.1.6. Gestió

**Descripció:** Temps dedicat a les reunions amb els representants IT, a les diferents presentacions del producte que s'han fet, a la creació d'un entorn de testing per al Tomàtic i al deploy a productiu.

**Tasques:**

- Reunions: 2 MD
- Presentació producte: 2 MD
- Deploy: 4 MD
- Creació entorn testing: 2 MD

**Temps:** 10 mitjos dies.

#### 4.1.7. Backend

**Descripció:** Anàlisi, disseny i implementació del codi necessari per tal de desenvolupar els scripts, crons i funcions necessàries per a que l'eina pugui complir la seva funció.

**Tasques:**

- Registre al full de càlcul: 3 MD
- Crear casos AC: 15 MD
- Extreure tipus de casos AC de manera dinàmica: 1 MD
- Testeig per transaccions: 3 MD

**A lliurar:** Codi font funcional disponible al repositori de GitHub.

**Temps:** 22 mitjos dies.

### 4.2. Comparativa esforços

A l'apartat número 10 d'aquesta documentació, es pot veure en detall les Targetes que han entrat per ronda, les tasques que s'han realitzat, l'esforç esperat i l'invertit.

Amb aquesta informació, descriuré una taula on es pugui veure de manera fàcil quin era l'esforç esperat i quin s'hi ha acabat dedicant. D'aquesta manera, es veurà l'estimació del temps a dedicar a cada tasca i es podrà comparar amb la dedicació real. La taula correspondrà a l'esforç que es pensava dedicar a cada ronda i el dedicat, finalment.

Podem veure que l'esforç total suposa 158 MD. Això seria un total de 632 hores. Si el preu per hora fos de 20€ el preu del producte final resultaria a 12.640€. Abans però, veiem l'esforç que realment ha entrat a Ronda:



	<b>Esforç esperat</b>	<b>Esforç dedicat</b>
R0	6	6
R1	12	12
R2	12	11
R3	15	24
R4	17	22
R5	17	17
R6	5	5
R7	7	10
R8	7	18
R9	10	15
R10	16	20
R11	7	8
<b>Total</b>	<b>131</b>	<b>168</b>

L'esforç que ha entrat fins a data d'avui equival a 131 MD. Això serien 524 hores. Si el preu a l'hora és de 20 €, el preu final al qual sortiria el producte serien 10.480€.

Per altra banda, veiem que l'esforç dedicat és major: un total de 168 mitjos dies. Equivaldrien a 672 hores. El preu final del producte seria 13.440€.

### 4.3 Evolució

Per tal de que es pugui visualitzar de manera general l'evolució del treball i com s'han anat dividint les tasques, s'utilitzarà un Diagrama de Gantt.

Com que la feina s'ha realitzat per Rondes, es podrà visualitzar el temps que s'ha dedicat a cada Ronda. Les columnes de *Inici* i *Final* mostren les dates exactes mentre que els requadres contemplen els mesos en general. Si una ronda ha començat a mig mes, s'ha pintat el requadre sencer.

La feina realitzada a cada Ronda dels diferents paquets de treball es veurà en els següents apartats.

Ronda	Inici	Final	03-2019	04-2019	05-2019	06-2019	07-2019	08-2019	09-2019	10-2019
R0	01-03-2019	25-03-2019								
R1	26-03-2019	12-04-2019								
R2	13-04-2019	07-05-2019								
R3	08-05-2019	28-05-2019								
R4	03-06-2019	24-06-2019								
R5	27-06-2019	09-07-2019								
R6	17-02-2020	28-02-2020								
R7	02-03-2020	27-03-2020								
R8	30-03-2020	20-04-2020								
R9	21-04-2020	01-05-2020								
R10	06-05-2020	05-06-2020								
R11	06-06-2020	26-06-2020								

*Primera part de l'evolució del treball.*

Ronda	Inici	Final	11-2019	12-2019	01-2020	02-2020	03-2020	04-2020	05-2020	06-2020
R0	01-03-2019	25-03-2019								
R1	26-03-2019	12-04-2019								
R2	13-04-2019	07-05-2019								
R3	08-05-2019	28-05-2019								
R4	03-06-2019	24-06-2019								
R5	27-06-2019	09-07-2019								
R6	17-02-2020	28-02-2020								
R7	02-03-2020	27-03-2020								
R8	30-03-2020	20-04-2020								
R9	21-04-2020	01-05-2020								
R10	06-05-2020	05-06-2020								
R11	06-06-2020	26-06-2020								

*Segona part de l'evolució del treball.*

## 5. Marc de treball i conceptes previs

Pel que fa al marc de treball, primerament mencionaré a les persones que han col·laborat d'una manera més directa. Per últim, passaré a mencionar els llenguatges de programació i frameworks utilitzats i que m'ha calgut aprendre per desenvolupar aquest projecte.

Des de que vaig entrar a la cooperativa, vaig iniciar amb el desenvolupament d'aquest projecte, així que vaig anar aprenent a mida d'anar desenvolupant. També vaig anar entenent moltes idees i conceptes que, prèviament no tenia molt clars. Això va afectar al desenvolupament del projecte de manera que, en algun punt, vaig dedicar part del temps que tenia per dedicar a fer algun refactor del codi o millorar certs problemes de rendiment.

### 5.1 Persones que col·laboren

- ❖ **Francisco Cubero**, desenvolupador sènior que em va introduir al projecte, al funcionament i manera de organitzar-se de Som Energia i em va explicar quin anàlisi previ s'hi havia fet i com s'estava enfocant.
- ❖ **Victor Carreño i Carol Coll**, primers *representants IT* que vaig tenir del projecte que em van donar els primers requeriments.
- ❖ **Benjamí Ramos**, tutor del projecte que m'ha ajudat amb la organització i assessorat des del punt de vista d'un desenvolupador amb experiència en la recollida de requeriments i en el procés de desenvolupar un TFG a la empresa.
- ❖ **Judith Frigolé i Joan Utges**, actuals *representants IT* que han ajudat amb la segona part del treball i han donat els nous requeriments del projecte.

Altres persones del departament de IT han anat treballant amb el projecte Tomàtic i ens hem hagut de coordinar, sobretot pel que fa a la presa de decisions de coses que afectaven al desenvolupament general però, des de l'inici, l'eina que hem anomenat *Kalinfo*, ha estat realitzada per mi.

### 5.2 TDD i Unit testing

S'ha realitzat la part backend del codi amb **test-driven development**. És una pràctica de desenvolupament de software que consisteix en escriure primer els tests i, a partir d'aquests, anar desenvolupant el programa. Es tracta d'anar generant vermells i verds (errors i correccions), afinant cada cop més el codi perquè cobreixi tots els casos que es detectin. A part d'assegurar el funcionament del codi, també permet certa garantia de que, si s'han de fer modificacions, res deixi de funcionar (passant els tests).

Pel que fa al frontend, també es podria haver implementat amb TDD, però a l'inici del projecte, es va fer un anàlisi i no es va trobar cap eina que satisfés totes les necessitats. Actualment s'està treballant amb *cypress* i, quan s'hagi fet la migració del framework del *Tomàtic* a React, es podrà plantejar fer-ho mitjançant aquesta pràctica.

Pel que fa al mètode de testeig de software, s'ha utilitzat el que es coneix com a **unit testing**.

Es programen els tests per mètodes concrets per tal de que tot sigui el més atòmic possible. Tenir cada part del codi testejada, aporta el beneficis mencionats anteriorment. Si les funcions realitzen accions concretes i estan ben definides, garanteixen que es puguin realitzar tests concrets i específics per a garantir un funcionament més absolut.

S'han de tenir en compte conceptes com el test coverage que té (línies de codi cobertes), els mock objects (entitats capaces d'emular funcionalitats/dades per tal de poder realitzar tests que tenen requeriments externs) i la integració contínua (que es basa en la execució dels tests quan s'ha de fer un nou desenvolupament per garantir-ne la estabilitat).

## 5.3 Llenguatges de programació i frameworks

Els llenguatges que s'utilitzaran són Python i Javascript i els frameworks Mithril i Flask.

Aquests són els que s'utilitzen a Som Energia i, a més, amb els que està desenvolupat el Tomàtic. No han estat escollits però en el seu moment es va fer un anàlisi i es va arribar a la conclusió que són els més convenients per a desenvolupar aquests i tots els projectes de Som.

### 5.3.1 Python

La millor manera d'entendre en què es diferencia aquest llenguatge dels altres, penso que és llegir el que es coneix com El Zen de Python, descrit per el desenvolupador Tim Peters, que serveix per indicar què python és un codi:

- *Bonic és millor que lleig.*
- *Explícit és millor que implícit.*
- *Simple és millor que complex.*
- *Pla és millor que niat.*
- *Dispers és millor que dens.*
- *La llegibilitat compte.*
- *Els casos especials no són tan especials com per trencar les normes.*
- *Lo pràctic guanya lo pur.*
- *Els errors mai haurien de deixar-se passar silenciosament.*
- *A menys que hagin estat silenciats explícitament.*
- *Davant la ambigüitat, rebutja la temptació a endevinar.*
- *Hauria d'haver-hi una -i preferiblement només una- manera obvia de fer-ho.*
- *Encara que aquesta manera no sigui obvia al principi a no ser que siguis holandès.*
- *Ara és millor que mai.*
- *Encara que mai és normalment millor que ara mateix.*
- *Si la implementació és difícil de explicar, és una mala idea.*
- *Si la implementació és fàcil d'explicar, pot ser que sigui una bona idea.*
- *Els "espais de noms" (namespaces) són una gran idea. Fem-ne més!*

Com es pot veure, fa incís en la llegibilitat del codi. És tracta d'un llenguatge de programació multiparadigma que suporta la orientació a objectes, programació imperativa i, en menor mesura, programació funcional. És un llenguatge interpretat, dinàmic i multiplataforma.

Actualment, hi ha moltes llibreries i frameworks i és un dels llenguatges més utilitzat per a l'anàlisi de dades i Machine learning. La comunitat és gran, oberta i participativa, fet que ajuda a

que sigui fàcil consultar i trobar solucions a diferents problemes que puguin aparèixer, al desenvolupament de diferents llibreries per a funcionalitats molt variades, etc.

Podem trobar diferents problemes com l'interpret per defecte que s'instal·la amb el python (molt simple, en alguns aspectes) i problemes de compatibilitat de versions de llibreries per a diferents projectes. Tot i així, es pot utilitzar l'interpret ipython, que aporta una major funcionalitat i, pel que fa al tema de les incompatibilitats de les llibreries, hi ha els virtualenvs. Aquests, són una manera d'encapsular el binari del Python i un conjunt de llibreries en un directori de manera que es pot tenir un virtualenv per cada projecte específic i, dins d'aquest, les versions de les llibreries que es necessiten. Per tal d'instal·lar aquestes llibreries, es fa mitjançant el pip, el gestor de paquets.



### 5.3.2 JavaScript

JavaScript és un llenguatge de programació interpretat, dialecte de l'estàndard ECMAScript. Es defineix com orientat a objectes basat en prototips, imperatiu i dèbilment tipat i dinàmic.

S'utilitza sobretot per a la part Frontend, per a la part web i interfície d'usuari. JavaScript es va dissenyar amb una sintaxi similar al C tot i que adopta convencions del llenguatge de programació Java.

Tots els navegadors moderns interpreten el codi JavaScript integrat a les pàgines web. Per interactuar amb una pàgina web es proveeix al llenguatge una implementació del DOM (Document Object Model).

El seu origen ve de utilitzar-se a pàgines web HTML per a fer operacions concretes únicament en el marc de la aplicació client, sense poder accedir a funcions del servidor.

En els projectes amb JavaScript, es poden utilitzar gestors de paquets com npm o yarn. En aquest projecte s'ha utilitzat npm, el per defecte de Node.js. Es poden especificar els paquets necessaris per al projecte en el fitxer *setup.js*, i s'instal·len al directori del projecte concret anomenat *node\_modules*.



### 5.3.3 Flask

Flask és un framework minimalista escrit en Python que permet crear aplicacions web ràpidament. Està basat en la especificació de Wsgi de Werkzeug. Té una llicència BSD. Està basat en Unicode, té una extensa documentació i disposa d'un servidor de desenvolupament i depuració.

No té capa d'abstracció de base de dades, validació de formularis ni cap altre component. Per altra banda, es poden utilitzar diverses llibreries de python per tal de complementar les funcionalitats que es trobin a faltar.

Per al projecte, el framework que s'ha utilitzar per a crear els endpoints de la API ha estat Flask.



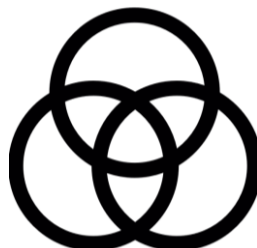
### 5.3.4 Mithril

Mithril és un framework de JavaScript que destaca per la seva rapidesa i mida reduïda a l'hora de descarregar els arxius.

En el seu moment, es va escollir utilitzar aquest framework per desenvolupar el tomàtic perquè, aleshores, s'estava utilitzant Angular. Els avantatges que es van veure respecte aquest, són el ràpid temps de càrrega que utilitza gràcies al DOM virtual. Consisteix en una estructura en forma d'arbre que conté el DOM real i que permet renderitzar només les diferències dels canvis que s'efectuïn i no tota la pàgina.

Per altra banda, la comunitat que hi ha darrera ha disminuït notablement els últims anys i cada cop hi ha més dificultat per trobar components ja elaborats. El Tomàtic n'utilitza molts de [polythene](#), desenvolupats per Arthur Clemens, amb les modificacions necessàries per tal de que s'adapti el màxim possible a les necessitats desitjades.

Per altra banda, per a decisió de la cooperativa i l'equip de desenvolupament web, s'ha decidit que s'aniran migrant els diferents projectes fets amb altres frameworks a React. Per a aquest motiu, després realitzar un gran refactor per separar els diferents components desenvolupats per al Kalinfo, no es va prioritzar millorar la part Frontend del treball.



*Logo de Mithril*

## 6. Requeriments del sistema

Seguidament es descriuran els requisits funcionals i no funcionals que ha de complir el sistema. Abans, però, es comentarà el context i els canvis que s'han anat realitzant des de la idea inicial del projecte.

Els requeriments del Kalinfo han anat canviant lleugerament amb el temps degut (sobretot) al canvi de representants IT que hi va haver enmig del desenvolupament del projecte.

### 6.1 Context

En un inici, la intenció del Kalinfo era poder mostrar informació de manera automàtica a les persones que realitzen atenció telefònica sobre la persona que realitza la trucada entrant (a partir del telèfon). A més, poder mostrar estadístiques amb informació que es considerés rellevant. Per tal de poder realitzar aquest segon punt, era necessari recopilar informació de totes les trucades que s'atenen cada dia. S'estaven utilitzant dos formularis de Google que s'havien d'emplenar al acabar les trucades; un indicava el motiu de la trucada i, l'altre, en cas de que la trucada fos per a una reclamació, el tipus de reclamació. Per tal de que tothom que estigués fent atenció telefònica entrés aquesta informació sense haver de sortir del Tomàtic, es va demanar que des del Kalinfo es pogués respondre el motiu de la trucada o el tipus de reclamació i desar-se al mateix document on anaven les respostes del formulari.

Un cop desenvolupada aquesta primera part, es van realitzar millores com tenir un registre de trucades rebudes i saber quines estan registrades o no, per tal de poder recuperar-les i respondre el qüestionari sense necessitat de que es fes justament després de la trucada.

Veient la facilitat amb la que es podia recopilar la informació de la trucada, de seguida es va detectar que es podia automatitzar un procés que es feia manualment fins ara: la generació de casos AC (reclamacions) generats a través de trucades telefòniques. És una informació que requereix la CNMC (Comissió Nacional dels Mercats i la Competència) i que costa mantenir actualitzada per al volum de informació i la manca d'eines que ho facilitin. Gràcies al Kalinfo, es va veure que es podia utilitzar la informació que s'extreia de la trucada i la resposta al qüestionari perquè, en cas de que la trucada fos una reclamació, es generés automàticament el cas AC corresponent al ERP de la cooperativa. Un cop es va veure aquesta opció, de seguida es va decidir apartar la idea inicial i intentar aprofitar la eina per tal de generar aquests casos diàriament i de manera automàtica.

Un cop posat sobre la taula el requeriment de que es registressin els casos AC automàticament al ERP, es va veure que el Kalinfo podia servir una mica de CRM: al ERP existeix un mòdul anomenat "Trucades Telefòniques" que va obrir una porta a la possibilitat de, deixar de desar la informació rebuda de les trucades entrants a una fulla de càlcul i començar a desar-la també al ERP. D'aquesta manera, també es podia automatitzar que, després de respondre el qüestionari del Kalinfo, es generessin aquests casos de "Trucades Telefòniques" i es desés tota la informació al ERP.

Això també millorava i facilitava el fet d'extreure estadístiques interessants sobre les trucades. Per aquest motiu, el que semblava que era un requeriment inicial, va acabar essent un benefici que es podia extreure d'un procés d'automatització de la generació de casos AC i trucades telefòniques a través del qüestionari del Kalinfo, sobre trucades concretes.

## 6.2 Requisits funcionals

El Kalinfo és una eina d'ús intern. El poden utilitzar només les persones treballadores de l'equip tècnic. Tothom es pot crear un usuari, però per no desar massa informació s'intenta regular i mantenir només els de les persones que realitzen atenció telefònica.

Encara que no es tingui usuari, es pot utilitzar el Tomàtic. Tot i així, per realitzar certes accions com canviar els torns de telèfon cal que la persona estigui identificada (per tant, ha de tenir usuari). Els principals actors del nostre sistema seran els usuaris identificats i els usuaris no identificats. Més endavant explicaré amb més detall els altres actors que interactuen amb el sistema (la centraleta i l'script). Els usuaris identificats podran realitzar algunes accions més que els no identificats.

Les persones que realitzen atenció telefònica, podran:

- Identificar-se (seleccionar qui és).
- Desidentificar-se (indicar que ja no és aquella persona).

Les persones no identificades, podran:

- Identificar-se quan volen canviar un torn de telèfon.
- Cercar a través d'un buscador a la pestanya "trucada" per els camps:
  - Telèfon
  - Nom i cognoms
  - Nif
  - Email
  - Tot

Les persones identificades:

- Podran canviar torns de telèfon.
- Podran realitzar les mateixes cerques que les no identificades.
- Se'ls hi refrescarà automàticament la pestanya "trucada" del Tomàtic amb la informació de la persona que truca. Aquesta informació s'obté a través del número de telèfon, si aquest figura a la nostre base de dades i serà:
  - Referent a la titular:
    - Nom i cognoms.
    - NIF.
    - Número de sòcia/partner.
    - Província i municipi.
    - Email (amb enllaç directe al helpscout).
    - Indicar si és sòcia d'energètica.
    - Indicar si ha obert l'oficina virtual.
    - Tots els contractes que té associats com a titular, sòcia, pagadora o notificadora, ordenats segons prioritat.
  - Referent a un contracte concret:
    - Número.
    - Nom de la persona titular.
    - CUPS (Codi Universal de Punt de Subministrament).
    - Adreça CUPS.
    - Potència.



- Tarifa.
  - Data última lectura facturada.
  - Estat pendent.
  - S'indica amb T/S/P/N si la persona seleccionada està relacionada amb aquell contracte com a Titular/Soci/Pagador/Notificador.
  - Data d'alta.
  - Data de baixa/Actualitat.
  - Si té facturació suspesa.
  - Els casos ATR que tingui oberts (altes de nous contractes, baixes de contractes, modificacions contractuals, etc).
  - Si és d'Energètica.
  - Si té Generation kWh.
  - Lot de facturació.
  - Compte bancari.
  - Si és no estimable.
  - Comptadors associats.
  - Últimes 12 factures.
- Referent a un comptador concret:
    - Número de comptador
    - Data
    - Lectura
    - Origen
    - Període
  - Referent a una factura concreta:
    - Número de factura
    - Data inici
    - Data fi
    - Empresa
    - Total
    - Energia facturada
    - Dies facturats
    - Data factura
    - Data venciment
    - Estat
- Se'ls hi actualitzarà el registre de trucades ateses quan rebin una trucada.
  - Podran cercar informació referent a una trucada concreta seleccionant-la al registre.
  - Podran respondre un qüestionari referent a la trucada indicant:
    - Motiu de la trucada.
    - Tipus de reclamació (en cas de que la trucada sigui una reclamació).
  - Podran bloquejar la pàgina perquè no es refresqui automàticament.
  - Obrir una nova pàgina en un nou TAB.

A més, amb aquesta informació referent a els motius de les trucades i als tipus de reclamacions, un cron s'executarà diàriament per tal d'executar els scripts necessaris per realitzar les següents tasques:

- Crear els casos de Reclamació al ERP.
- Crear els casos de Trucades Telefòniques al ERP.

## 6.3 Requisits no funcionals

Tot i que la majoria de requisits eren funcionals, s'han afegit alguns requeriments per tal d'assegurar una bona experiència per als usuaris que utilitzin l'eina:

- La interfície es podrà visualitzar correctament amb els navegadors Firefox i Google Chrome i serà tant responsive com es permeti tenint en compte el volum de informació que es voldrà visualitzar.
- Es controlaran els errors que hagi de tenir en compte l'usuari i se'ls hi comunicarà de la manera més clara i precisa possible.
- Es controlaran els errors dels scripts que crearan els casos de trucades telefòniques i els casos ATC (reclamacions) per tal de que es pugui revisar què ha fallat quan es detecti que hi ha hagut algun error.
- Es desarà la informació respecte a la trucada amb la informació necessària per tal de garantir l'èxit de la creació dels casos. Per tant, els camps obligatoris seran requerits i no es permetrà desar-los si falta algun.
- Els tests respectaran la integritat de dades.
- L'eina és d'ús exclusivament intern. Només per persones de la cooperativa podran usar-la i, per tant, visualitzar tota aquesta informació. El servidor està degudament seguritzat per garantir que ningú tingui accés a aquesta informació a través de la API des de fora.
- S'assegurarà la disponibilitat del sistema de manera regular i, més, durant els torns de telèfon.
- S'espera un temps de resposta ràpid. Es treballarà per optimitzar i millorar el rendiment de la aplicació ja que treballa amb molta informació i ha de realitzar moltes cerques.
- S'espera no tenir problemes de concurrència ni per rebre les trucades i per desar els casos.
- S'espera que cada usuari rebrà la informació de la seva trucada. S'assegurarà que en cap cas hi hagi creuament.
- No hi hauran faltes ortogràfiques.
- L'eina seguirà l'estil del Tomàtic, amb les interfícies estandarditzades.
- L'eina serà Open Source i es podrà trobar a Github.
- Es garantirà que la informació que es mostra és 100% verídica i que, en cap cas, s'està proporcionant informació que no és correcte al client per culpa de l'eina.

## 6.4 Requisits per part de l'empresa

Els únics requisits de part de Som Energia pel que fa al desenvolupament de l'eina, han estat:

- El backend serà programat amb python.
- El framework utilitzat per desenvolupar la API serà Flask.
- El codi ha d'estar cobert per tests seguint la metodologia unittest.
- El desenvolupament del backend es farà amb TDD.
- El frontend de l'aplicatiu es programarà amb JavaScript < ES6.
- El framework utilitzat per a la UI serà Mithril.
- L'eina estarà integrada dins del "Tomàtic".
- Es desenvoluparà amb branques i utilitzant el control de versions de Github.

## **6.5 Requisites de llicència**

Tota la documentació i codi necessari està sota la GNU Affero General Public License v3.0.

## 7. Estudis i decisions

Pel que fa al codi del *backend*, s'utilitza el llenguatge de programació *Python*. Per tal de que es segueixi un estil concret, el codi serà estil PEP8.

Per altra banda, el *frontend* està desenvolupat amb *JavaScript*. S'utilitzaran versions menors a JS6 (*ECMAScript 2016*) ja que degut a dependències de diferents llibreries que s'utilitzen, utilitzar JS6 causava un problema de compatibilitat amb la resta de producte. Tot i així, hi ha pendent una migració de tot el producte del *framework* actual que utilitza (*Mithril*) a *React*.

La metodologia de treball que s'utilitza normalment és *pair programming*, però com que el desenvolupament d'aquesta eina és per a un Treball de Final de Grau, només se'n va fer a l'inici per situar-me en el projecte i per tal de que pogués començar a desenvolupar per el meu compte.

El desenvolupament del *backend* s'ha fet amb TDD (*Test-Driving-Development*). Pel que fa al *frontend*, no s'ha desenvolupat amb tests. L'equip no té prou experiència encara i, tal com he mencionat, es preveu fer una migració de tot el producte i, des d'allà, establir uns criteris i començar a desenvolupar també amb TDD la part de *frontend*.

S'han utilitzat moltes llibreries durant tot el desenvolupament del projecte, però les principals són: *websockets* de *python* i *polythene* per *javascript* ja que el disseny del tomàtic utilitzava aquests components.

### 7.1 Websockets

S'ha utilitzat la llibreria de *python* [websockets](#) per a realitzar una de les parts més importants del projecte. Per tal de que cada cop que es rebés una trucada es notifiqués a la persona a la que se li estava fent la redirecció a nivell de centraleta, i només es refresqués la seva pàgina, s'ha hagut de crear un sistema de identificació i notificació a través de *websockets*. Es va fer un anàlisi a l'inici del projecte per tal de veure quina opció s'adheria millor a les necessitats i aportava un millor rendiment i, finalment, es va optar per utilitzar aquesta llibreria.

S'obre una connexió a un port concret i, quan es rep una trucada i es coneix la extensió de la persona que l'ha d'entendre, es notifica a la part servidor de la relació *sockets* client-servidor de que la persona amb la concreta extensió rebrà la trucada. A través d'una taula de relacions, es sap quina és el client amb aquesta extensió (la informa al registrar-se) i, per tant, se l'avisa de que ha de refrescar la informació del Kalinfo per una trucada del número entrant (fa la cerca i mostra la informació).

Es pot veure el funcionament de la llibreria a l'apartat 9 del treball. Allà s'explica el mètode utilitzat per fer les connexions i com es defineixen aquestes funcions.

### 7.2 Polythene

[Polythene](#) inclou els components que s'han utilitzat per construir la UI de la eina. Els components principals que s'han utilitzat són:

- Button
- Card
- Dialog layout i Full screen dialog

- Icon
- Checkbox
- Radio Button
- Text Field
- List
- List Tie
- Tabs
- Toolbar

Es poden trobar tots ben explicats a la pàgina oficial de [Github](#) sobre Mithril.

Explicaré una mica la idea de com funciona l'ús d'aquests components amb l'exemple del Button:

Primerament s'hauria d'importar: `const { Button } = polythene;`

Per utilitzar el botó, ho podem fer de la següent manera:

```
m("div", [
  m(Button, {
    label: "Accepta",
    raised,
  })
])
```

Com es pot veure, al diccionari que s'afegeix com a segon paràmetre del m, hi ha les propietats del botó que tindrà. A la següent taula es poden veure algunes de les disponibles:

Paràmetre	Requerit	Tipus	Per defecte	Descripció
border	opcional	Boolean	false	Afegeix contorn al botó. Per defecte el contorn no té color. Es pot afegir a la classe.
contained	opcional	Boolean	false	Dona al botó un aspecte amb més padding lateral.
disabled	opcional	Boolean	false	Desactiva el botó
inactive	opcional	Number		Nombre de segons que estarà inactiu després d'un click.

El "label" indica el contingut en text del botó i el "raised" crea un botó tipus Raised (elevat).

Es poden veure totes les propietats del botó a la [documentació](#) pertinent.

D'aquesta manera, utilitzant variant paràmetres i definint classes concretes per a cada objecte i component, s'ha anat creant casi tota la interfície del Kalinfo.

Un avantatge d'utilitzar aquests components, és que si es fa la prevista migració a React del projecte Tomàtic, també seran compatibles i es podran reutilitzar.

## **8. Anàlisi i disseny del sistema**

Descripció de l'anàlisi i el disseny del sistema.

### **8.1 Anàlisi**

En aquest apartat es comentaran les necessitats del sistema. Es començarà anomenant els actors del sistema que interactuaran amb l'eina, els diagrames de casos d'ús i les fitxes de casos d'ús. La finalitat és analitzar de manera detallada les necessitats del sistema i detallar el model de processos.

#### **8.1.1 Actors del sistema**

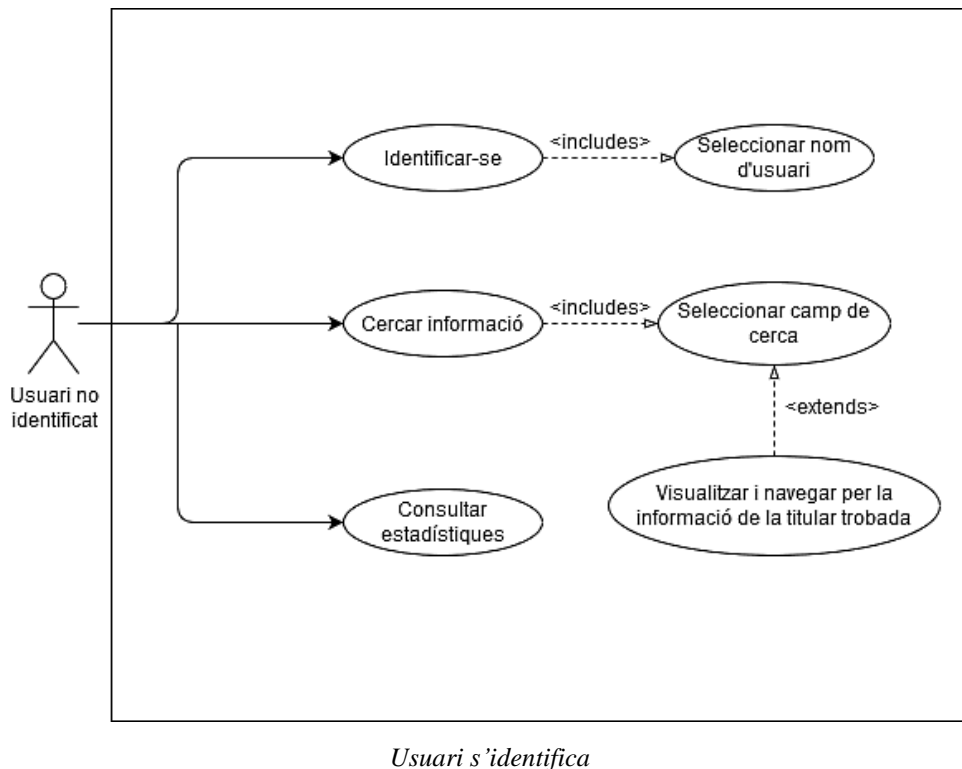
S'entén com a actor qualsevol unitat externa que interactua amb el nostre sistema i utilitza les seves funcionalitats. Els actors d'aquesta eina seran:

1. Persona que no s'ha identificat
2. Persona que s'ha identificat
3. Centraleta
4. Script de creació de casos de trucades telefòniques i atenció client.

#### **8.1.2 Diagrames de casos d'ús**

Pel que fa a l'usuari no identificat:

- Pot identificar-se i, per tant, seleccionar el seu nom d'usuari.
- També podrà cercar informació, seleccionant el camp de cerca prèviament. Si hi ha informació, podrà visualitzar i navegar per la informació trobada.
- Consultar les estadístiques sobre la informació de les trucades i sobre com estan anant els torns de telèfon.

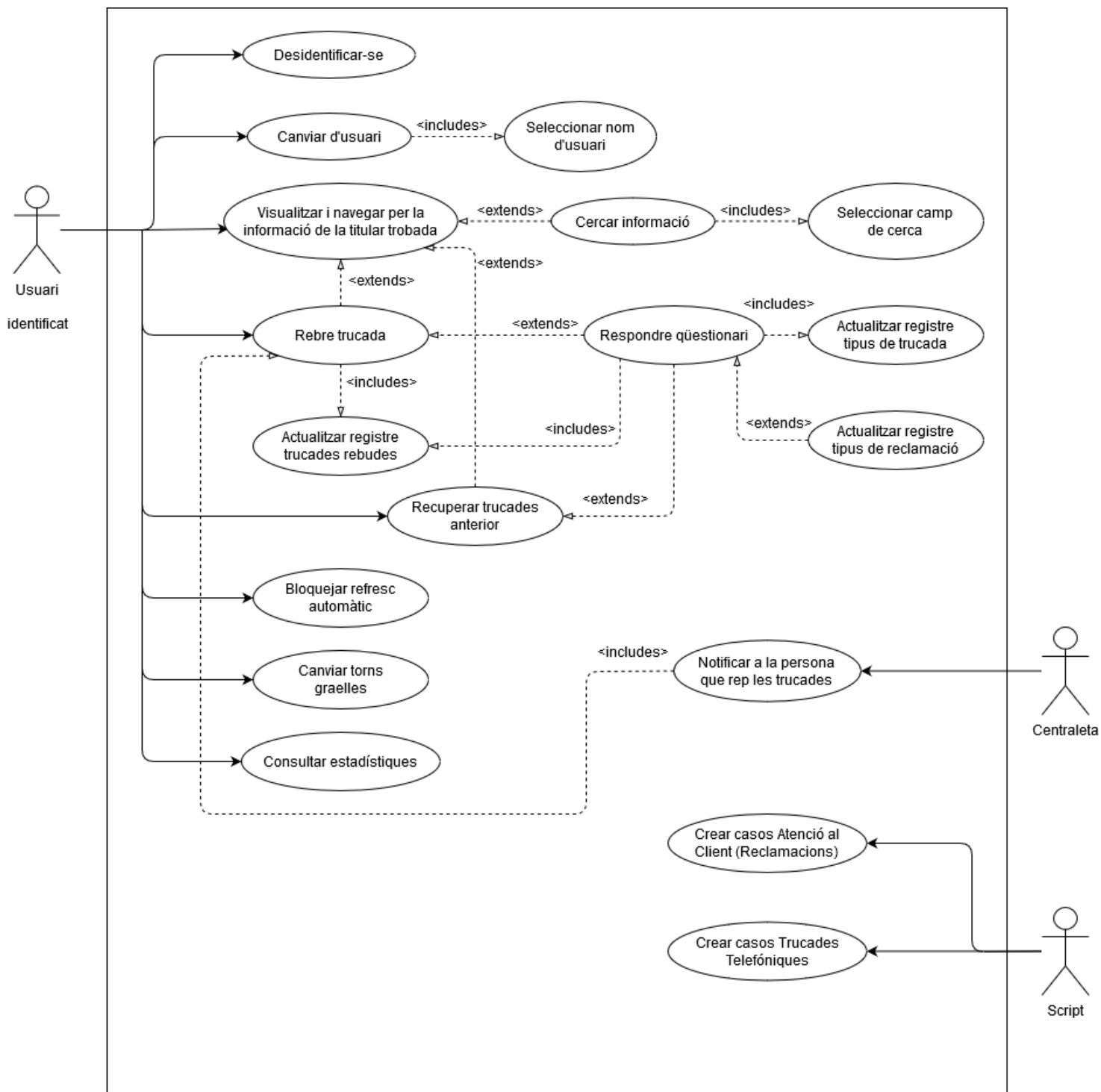


Pel que fa a l'usuari identificat:

- Pot desidentificar-se.
- Canviar d'usuari, seleccionant el nom d'usuari que pertorqui.
- Pot visualitzar la informació d'un client quan:
  - Cerca la informació i, per tant, indica un camp de cerca.
  - Rep una trucada i, per tant, s'actualitza el registre de trucades rebudes.
  - Recupera una trucada.
- Rebre una trucada i, per tant, visualitzar la informació trobada si n'hi ha. S'actualitzarà el registre de trucades i podrà respondre el qüestionari indicant informació sobre la trucada. Si respon el qüestionari:
  - S'actualitza el registre de trucades rebudes.
  - S'actualitza el registre de tipus de trucades.
  - Es podrà actualitzar el registre de tipus de reclamació (si el motiu de la trucada és una reclamació).
- Recuperar trucades anteriors. Si recupera una trucada anterior:
  - Visualitzarà la informació de la persona si n'hi ha.
  - Pot respondre el qüestionari.
- Pot bloquejar el refresc automàtic del Kalinfo.
- Pot canviar torns de telèfon a les graelles.
- Consultar estadístiques.

La Centralita notifica a la persona que rep la trucada i, implica, que l'usuari identificat rebrà la trucada.

L'Script crea casos al ERP de Trucades Telefòniques i de casos de reclamacions (casos AC).



*Usuari realitza atenció telefònica*



### 8.1.3 Fitxes de casos d'ús

A continuació veurem les principals fitxes de casos d'us

Usuari s'identifica	
<b>Descripció</b>	L'usuari pot identificar-se per indicar qui és
<b>Actor</b>	Usuari no identificat
<b>Pre-condició</b>	L'usuari està al <i>linphone</i> i s'ha creat al Tomàtic
<b>Flux principal</b>	<ol style="list-style-type: none"><li>1. L'usuari obre el Tomàtic</li><li>2. L'usuari clica al botó "Identifica-se"</li><li>3. L'usuari selecciona el seu username</li></ol>
<b>Flux alternatiu</b>	
<b>Post-condició</b>	El servidor actualitza la taula de connexions segons la extensió de l'usuari i la seva IP.

Usuari canvia d'identificació	
<b>Descripció</b>	L'usuari pot identificar-se tot i estar identificat per indicar qui és
<b>Actor</b>	Usuari identificat
<b>Pre-condició</b>	L'usuari està identificat i el seu nou usuari està al <i>linphone</i> i s'ha creat al Tomàtic
<b>Flux principal</b>	<ol style="list-style-type: none"><li>1. L'usuari està identificat a al Tomàtic</li><li>2. L'usuari clica al botó on apareix el seu username.</li><li>3. L'usuari selecciona el seu nou username.</li></ol>
<b>Flux alternatiu</b>	
<b>Post-condició</b>	El servidor actualitza la taula de connexions segons la nova extensió de l'usuari i la seva IP.

Usuari es desidentifica	
<b>Descripció</b>	L'usuari pot desidentificar-se
<b>Actor</b>	Usuari identificat
<b>Pre-condició</b>	L'usuari està identificat
<b>Flux principal</b>	<ol style="list-style-type: none"><li>1. L'usuari està identificat a al Tomàtic</li><li>2. L'usuari clica al botó amb la creueta per desidentificar-se.</li></ol>
<b>Flux alternatiu</b>	
<b>Post-condició</b>	El servidor actualitza la taula de connexions.

<b>Cerca d'informació</b>	
<b>Descripció</b>	L'usuari pot cercar informació d'una partner
<b>Actor</b>	Usuari no identificat i Usuari identificat
<b>Pre-condició</b>	
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari obre el Tomàtic</li> <li>2. L'usuari accedeix a la pestanya "Trucada"</li> <li>3. L'usuari selecciona al cercador segons quin paràmetre vol realitzar la cerca.</li> <li>4. L'usuari escriu al camp de text el paràmetre de cerca.</li> <li>5. Mentre es carrega la informació, l'usuari veurà que s'està carregant la informació.</li> <li>6. L'usuari veu la informació de les partners relacionades amb el que ha cercat.</li> </ol>
<b>Flux alternatiu</b>	<p>Si la cerca és poc específica s'indica que ha estat poc específica i que hi ha masses resultats.</p> <p>Si la cerca no té resultats, s'indica que no hi ha informació.</p>
<b>Post-condició</b>	El servidor actualitza la taula de connexions segons la extensió de l'usuari i la seva IP.

<b>Informació</b>	
<b>Descripció</b>	L'usuari visualitza la informació referent a una titular concreta.
<b>Actor</b>	Usuari identificat i Usuari no identificat.
<b>Pre-condició</b>	L'usuari ha rebut una trucada o ha realitzar una cerca i està a la pestanya "Trucada".
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari visualitza la informació del partner i pot navegar entre els diferents partners torbats.</li> <li>2. L'usuari visualitza la informació dels contractes del partner seleccionat i pot navegar entre els diferents contractes d'aquest.</li> <li>3. L'usuari visualitza la informació dels comptadors del contracte seleccionat.</li> <li>4. L'usuari visualitza la informació de les últimes factures del contracte seleccionat.</li> </ol>
<b>Flux alternatiu</b>	Si un camp concret no té informació, apareix un text alternatiu indicant que no hi ha informació o que no s'ha especificat.
<b>Post-condició</b>	L'usuari veu la informació.

<b>Usuari rep trucada</b>	
<b>Descripció</b>	L'usuari està realitzant atenció telefònica i rep una trucada.
<b>Actor</b>	Usuari identificat.

<b>Pre-condició</b>	L'usuari té el Tomàtic obert.
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. La pestanya de "Trucada" del "Tomàtic" s'actualitza</li> <li>2. L'usuari pot visualitzar la informació</li> </ol>
<b>Flux alternatiu</b>	
<b>Post-condició</b>	La pestanya "Trucada" de la persona que atén la trucada s'actualitza amb la informació trobada a través del número entrant.

<b>Usuari bloqueja refresc automàtic</b>	
<b>Descripció</b>	L'usuari bloqueja el refresc automàtic de la pàgina "Trucada".
<b>Actor</b>	Usuari identificat
<b>Pre-condició</b>	L'usuari està a la pestanya "Trucada" del "Tomàtic".
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari clica al botó per bloquejar la pàgina.</li> <li>2. No s'actualitza la pàgina automàticament quan l'usuari rep una nova trucada.</li> </ol>
<b>Flux alternatiu</b>	Si l'usuari té la pestanya bloquejada, però té oberta una altra pestanya que no ho està, es refresca només aquella pestanya.
<b>Post-condició</b>	El refresc automàtic per trucada rebuda queda desactivat per a la pestanya on s'ha clicat el botó corresponent.

<b>Usuari empena qüestionari</b>	
<b>Descripció</b>	L'usuari empena el qüestionari del Kalinfo indicant quin ha estat el motiu de la trucada.
<b>Actor</b>	Usuari identificat
<b>Pre-condició</b>	L'usuari ha rebut una trucada o n'ha recuperat una de les anteriors.
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari clica al botó per respondre el qüestionari</li> <li>2. L'usuari indica el motiu de la trucada del llistat de motius. Pot utilitzar el filtre de motius.</li> <li>3. L'usuari afegeix una descripció.</li> <li>4. L'usuari desa la resposta.</li> </ol>
<b>Flux alternatiu</b>	<p>Si la trucada és una reclamació, l'usuari també ha d'indicar:</p> <ol style="list-style-type: none"> <li>1. L'usuari responsable.</li> <li>2. Si s'ha resolt.</li> <li>3. Si s'ha resolt, ha d'indicar si el cas és <i>procedente</i>, <i>improcedente</i> o <i>no gestionable</i>.</li> <li>4. Una descripció més acurada respecte la reclamació.</li> </ol>

<b>Post-condició</b>	Es desa la resposta al registre de trucades respostes i s'actualitza el registre de trucades rebudes.
----------------------	---

<b>Usuari recupera trucada anterior</b>	
<b>Descripció</b>	L'usuari clica una de les trucades que apareixen al registre de trucades rebudes i "recupera la trucada".
<b>Actor</b>	Usuari identificat
<b>Pre-condició</b>	L'usuari ha rebut alguna trucada i no ha respost el qüestionari.
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari clica una de les trucades que apareixen al registre de trucades rebudes.</li> <li>2. La pàgina mostra la informació realitzant la cerca per al número que ha realitzat la trucada.</li> </ol>
<b>Flux alternatiu</b>	Si la informació que apareix (o no apareix) no és l'adequada per a respondre el qüestionari, l'usuari pot realitzar una cerca sobre aquella trucada (i respondre el qüestionari sobre aquell partner cercat).
<b>Post-condició</b>	Es recupera el número del registre de trucades ateses i es realitza la cerca corresponent per mostrar la informació que pertoca.

<b>Usuari canvia torn de telèfon</b>	
<b>Descripció</b>	L'usuari pot canviar un torn concret de telèfon
<b>Actor</b>	Usuari identificat
<b>Pre-condició</b>	
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari obre el tomàtic</li> <li>2. L'usuari va a la pestanya "Graelles"</li> <li>3. L'usuari clica sobre el registre que vulgui canviar i el modifica.</li> </ol>
<b>Flux alternatiu</b>	Si l'usuari no està identificat, li demana que s'identifiqui prèviament. Altrament, no pot realitzar el canvi.
<b>Post-condició</b>	L'usuari canvia el torn de telèfon i queda registrat qui ha fet el canvi.

<b>Usuari consulta estadístiques</b>	
<b>Descripció</b>	L'usuari pot consultar estadístiques de la centraleta i de les trucades rebudes.

<b>Actor</b>	Usuari identificat.
<b>Pre-condició</b>	
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari obre el dashboard corresponent.</li> <li>2. L'usuari pot visualitzar la informació respecte les estadístiques concretes.</li> </ol>
<b>Flux alternatiu</b>	
<b>Post-condició</b>	L'usuari pot visualitzar i consultar les estadístiques pertinents.

<b>Centraleta rep trucada</b>	
<b>Descripció</b>	La centraleta rep una trucada i avisa al servidor de qui la rebrà.
<b>Actor</b>	Centraleta
<b>Pre-condició</b>	
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. Centraleta rep trucada.</li> <li>2. Centraleta redirigeix trucada segons extensió disponible.</li> <li>3. Centraleta executa comanda per desar la redirecció utilitzada a el log pertinent.</li> <li>4. Centraleta executa comanda per avisar al servidor que la extensió rebrà una trucada del número entrant i comparteix aquest.</li> </ol>
<b>Flux alternatiu</b>	Si cap extensió està disponible, salta el contestador.
<b>Post-condició</b>	El servidor revisa la seva taula d'extensions identificades i notifica a la persona amb el número que l'està trucant perquè es pugui fer la cerca (automàticament).

<b>Script genera casos Trucada Telefònica</b>	
<b>Descripció</b>	Script genera els casos de Trucada Telefònica quan s'acaba el torn d'atenció telefònica.
<b>Actor</b>	Script
<b>Pre-condició</b>	El format del fitxer log on es desa la informació de les trucades rebudes és correcte.
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'script revisa tots els fitxers del directori del dia d'avui.</li> <li>2. L'script extreu la informació pertinent d'aquests fitxers amb el format que pertoca.</li> <li>3. L'script crea els casos per a cada trucada desada al registre.</li> </ol>
<b>Flux alternatiu</b>	Si hi ha qualsevol error, l'script el captura i el desa al registre perquè es pugui revisar què i on ha fallat.
<b>Post-condició</b>	S'ha creat un cas Trucada Telefònica per a cada trucada desada al registre.

<b>Script genera casos Atenció al Client</b>	
<b>Descripció</b>	Script genera es casos de Atenció al Client al ERP després del torn d'atenció telefònica.
<b>Actor</b>	Script
<b>Pre-condició</b>	El format del fitxer log on es desa la informació de les reclamacions ateses és correcte.
<b>Flux principal</b>	<ol style="list-style-type: none"> <li>1. L'script revisa tots els fitxers del directori del dia d'avui.</li> <li>2. L'script extreu la informació pertinent d'aquests fitxers amb el format que pertoca.</li> <li>3. L'script crea els casos AC per a cada reclamació desada al registre.</li> </ol>
<b>Flux alternatiu</b>	
<b>Post-condició</b>	S'ha creat un cas AC per a cada trucada desada al registre.

## 8.2 Disseny del sistema

Per entendre una mica més el disseny del sistema, primerament s'explicarà la situació de l'eina: on està i amb quins agents/actors interactua. Seguidament, es mostrarà el diagrama de fluxe que mostra el comportament que hauria de permetre l'eina Kalinfo mentre es realitza atenció telefònica.

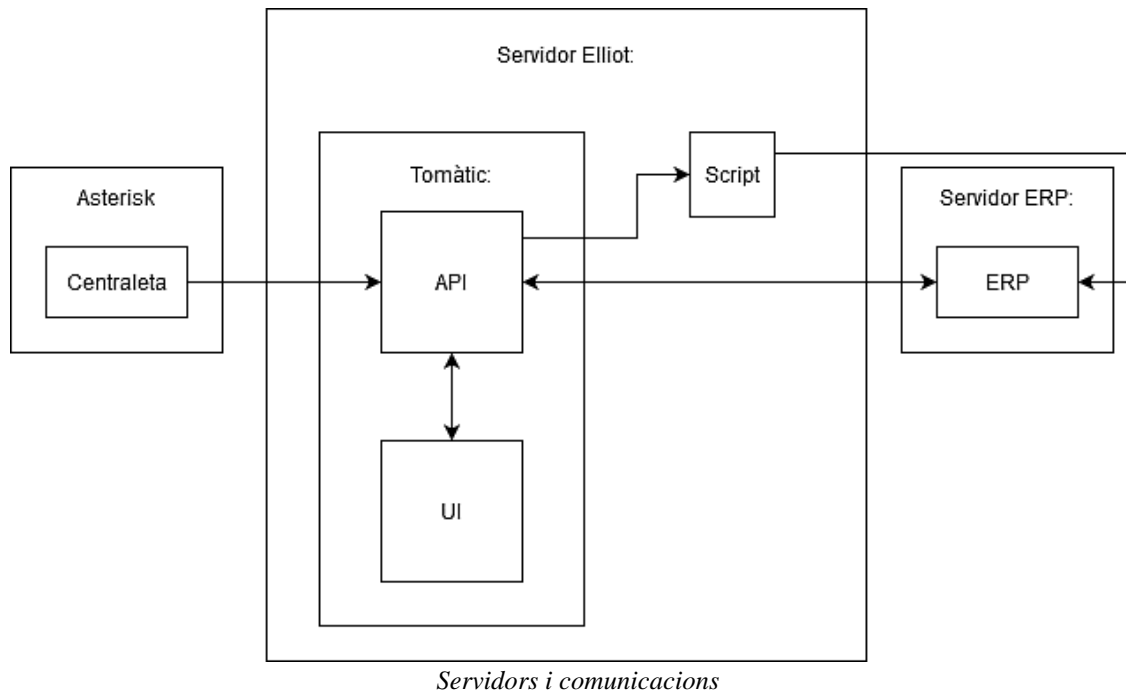
### 8.2.1 Situació eina

El servidor que té dins la Centraleta (que s'encarrega de la re-direcció de trucades telefòniques) s'anomena Asterisk. Aquest interactua amb el servidor Elliot, enviant missatges a la API del Tomàtic quan cada cop que rep alguna trucada.

Dins el servidor d'Elliot, hi ha tant la part Frontend com la part Backend del Tomàtic. El Kalinfo forma part del Tomàtic, així que comparteixen API i UI. Dins d'Elliot també hi ha l'Script que s'encarrega de la generació de casos Atenció al Client (reclamacions) al ERP. L'API i la UI es comuniquen entre elles. L'API, també es comunica amb l'Script i amb l'ERP.

L'ERP es situa al Servidor ERP i interactuen amb ell tant l'Script com l'API.

El disseny esquematitzat i simplificat seria el següent:



### 8.2.2 Fluxe del procés

El següent diagrama indica el fluxe que permetrà seguir la eina quan es realitza atenció telefònica. Es pot utilitzar en altres ocasions, però en un cas on s'estigui realitzant atenció telefònica aquesta és la idea bàsica d'ús de l'eina.

Una persona client fa una trucada. La Centraleta rep la trucada. Si no hi ha extensions disponibles, la trucada es re-dirigirà al contestador, altrament, es farà a la extensió que estigui disponible i que faci més temps que no respon cap trucada.

La Centraleta avisarà del número de telèfon i de l'extensió de qui rebrà la trucada a l'API del Tomàtic. Si la extensió no està identificada no s'actualitzarà la informació de cap persona, altrament, s'actualitzarà la pàgina Trucada amb la informació que es trobi dels clients que relacionats amb el número de telèfon entrant.

Si la informació que es visualitza no és la correcte, l'usuari pot cercar el client a través d'algun dels camps que permet el cercador. Un cop s'està visualitzant la informació correcta, l'usuari pot consultar la informació.

En cas de que no hi hagi temps per respondre el qüestionari, aquest es podrà respondre recuperant la trucada més tard. Quan es respongui, es desarà la informació sobre la trucada telefònica. A més, si la trucada és una reclamació, també es desarà el cas atenció al client.

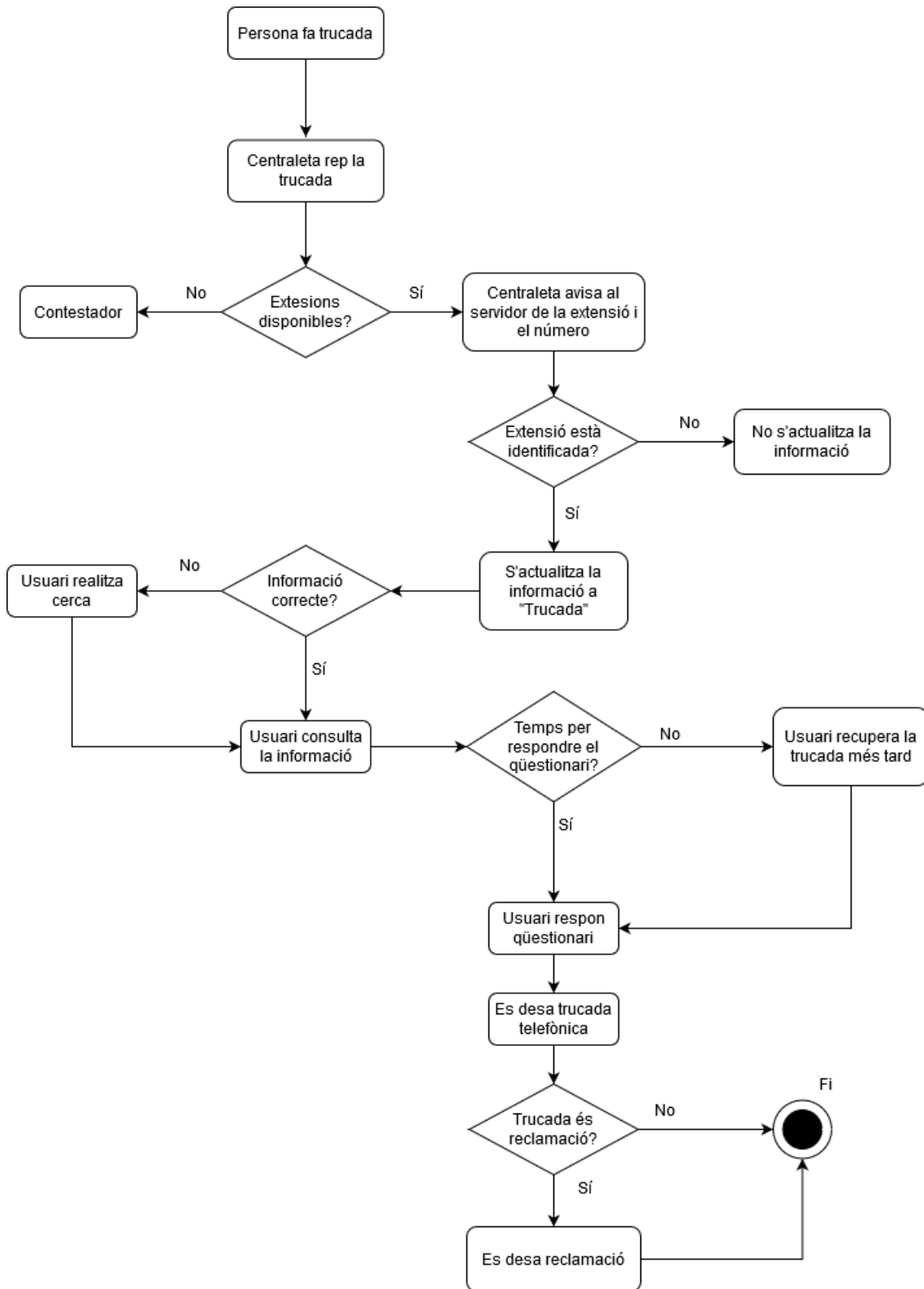


Diagrama de fluxe Kalinfo



### 8.2.3 Model de dades

La base de dades que consulta l'ERP té moltes taules, moltes dades i moltes relacions. Les taules que s'han utilitzat ja existien i no s'han hagut de modificar.

Les taules que s'han utilitzat per a consultar informació del client, es citaran i descriuran seguidament el nom de la taula, què s'hi desa i els camps més rellevants que la formen. Els camps concrets que s'utilitzen es podran veure en els següents apartats on s'explica la implementació del codi. Les claus foranes s'indiquen amb <fk> mentre que la clau de la taula amb <key>.

#### **giscedata\_polissa:**

Conté la informació sobre el contracte:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- titular <fk>
- name\_auto
- tensio <fk>
- active
- tarifa <fk>
- renovació\_auto
- cups <fk>
- name
- observacions

#### **res\_partner:**

Conté la informació de les persones titulars:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- comment
- active
- date
- lang
- customer
- credit\_limit
- user\_id <fk>
- name
- title
- website
- parent\_id <fk>
- supplier
- ref
- vat
- debit\_limit
- vat\_subjected
- payment\_type\_customer <fk>
- payment\_type\_supplier <fk>
- comercial <fk>
- bank\_inversions
- include\_in\_mod347
- ref2
- empowering\_token
- aeat\_registered
- country\_id <fk>

#### **res\_partner\_address:**

Tota la informació sobre les adreces:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- function
- fax
- street2
- phone
- street
- active
- partner\_id <fk>
- city
- zip
- title
- mobile
- country\_id <fk>
- birthdate
- state\_id <fk>
- type
- email
- id\_poblacio <fk>
- aclarador
- id\_municipi <fk>
- ref\_cadastral
- notes
- name
- apartat\_correus

### **giscedata\_cups\_ps:**

Tota la informació del CUPS:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- direcció
- active
- id\_municipi <fk>
- name
- id\_poblacio
- ref\_catastral
- distri\_id <fk>
- notes
- conany\_origen
- conany\_data
- conany\_kwh
- empowering
- empowering\_quarentine
- polissa\_polissa <fk>
- force\_potencia\_adscrita
- catas\_bq
- autoconsum\_id<fk>
- potencia\_adscrita

### **gisce\_data\_switching:**

Informació sobre els casos oberts (modificacions contractuals, contractacions, baixes...):

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- accio\_pendent
- comer\_sortint\_id <fk>
- data\_solicitud
- proces\_id <fk>
- whereiam
- versio
- proces
- conversation\_id <fk>
- step\_id <fk>
- ref\_contracte
- company\_id <fk>
- cups\_id <fk>
- seq\_solicitud
- case\_id <fk>
- obrir\_cas\_pendent
- codi\_solicitud
- validacio\_pendent
- cups\_polissa\_id <fk>
- finalitzat
- tarifa\_comer\_id <fk>
- cups\_input
- additional\_info
- documents\_solicitud\_adjunts
- actualitzar\_tarifa\_comer
- user\_observations
- last\_observation\_line
- polissa\_ref\_id <fk>
- accio\_pendent\_comerdist
- mode\_facturacio
- actualitzar\_mode\_facturacio

### **generation\_kwh\_assignment:**

Informació sobre el generation kWh:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- contract\_id <fk>
- end\_date
- priority
- member\_id <fk>

### **giscedata\_lectures\_comptador:**

Conté la informació sobre les lectures dels comptadors:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- lloguer
- name
- polissa <fk>
- data\_alta
- active
- giro
- data\_baixa
- propietat
- descripcio\_lloguer
- preu\_lloguer
- uom\_id <fk>
- empowering\_last\_measure
- meter\_type
- empowering\_last\_p5d\_measure
- empowering\_last\_f5d\_measure
- technology\_type
- cod\_pm
- contract\_import
- registrador\_id

### **giscedata\_facturacio\_factura:**

Informació sobre les factures:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- polissa\_id <fk>
- total\_potencia
- data\_inici
- polissa\_state
- tarifa\_acces\_id <fk>
- total\_energia
- date\_boe
- tipo\_facturacion
- data\_final
- invoice\_id <fk>
- facturació
- cups\_id <fk>
- llista\_preu
- potencia
- lot\_facturacio
- tipo\_factura
- ref
- tipo\_rectificadora
- estat\_impagat
- payment\_mode\_id
- total\_lloguers
- total\_reactiva
- importació\_id <fk>
- enviat\_carpeta
- enviat\_data
- enviat
- total\_atr
- per\_enviar
- energia\_kwh
- dies
- cch\_fact\_aviable
- polissa\_tg
- devolució\_id <fk>
- has\_facturae
- potencia\_kwdia
- total\_altres
- visible\_ov
- total\_exces\_potencia
- total\_generacio

### **giscedata\_lectures\_lectura:**

Informació sobre les lectures de comptadors:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- periode
- name
- observacions
- consum\_int
- incidencia\_int
- comptador
- origen\_id <fk>
- lectura <fk>
- tipus
- origen\_comer\_id <fk>
- motiu\_ajust
- ajust\_int
- lectura\_exporta
- ajust\_exporta\_int
- generacio\_int
- ajust
- ajust\_exporta
- consum
- generacio

Les que s'han utilitzat per desar els casos de reclamacions són:

### **crm\_case:**

Conté la informació sobre el que seria un cas CRM i la formen els camps:

- id <key>
- create\_uid
- create\_date
- write\_date
- write\_uid
- date\_closed
- ref2
- description
- probability
- canal\_id <fk>
- date\_action\_last
- planned\_cost
- partner\_address\_id <fk>
- som
- section\_id <fk>
- date
- active
- partner\_id <fk>
- date\_action\_next
- user\_id <fk>
- date\_deadline
- planned\_revenue
- categ\_id <fk>
- priority
- state
- email\_cc
- ref
- email\_from
- category2\_id <fk>
- duration
- partner\_name
- note
- case\_id <fk>
- partner\_name2
- partner\_mobile
- stage\_id <fk>
- partner\_phone
- name
- time\_tracking\_id <fk>
- polissa\_id <fk>

### **crm\_case\_section:**

Informació sobre la secció de cas CRM:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- user\_id <fk>
- name
- sequence
- parent\_id <fk>
- reply\_to
- allow\_unlink
- active
- code

### **crm\_case\_history:**

Conté informació sobre l'històric del cas CRM:

- id <key>
- create\_uid <fk>
- create\_date
- write\_uid <fk>
- log\_id <fk>
- description
- email

### **giscedata\_atc:**

Conté la informació referent al cas ATC (reclamació):

- id <key>
- create\_uid <fk>
- create\_date
- write\_uid <fk>
- not\_allowed\_subtipus
- provincia <fk>
- subtipus\_id <fk>
- reclamante
- total\_cups
- agent\_actual
- resultat
- crm\_id <fk>
- tancar\_cac\_al\_finalitzar\_rl

### **giscedata\_subtipus\_reclamacio:**

Informació referent al subtipus de reclamació:

- id <key>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- type
- name
- desc
- sector
- default\_section <fk>
- atr
- circular

### **res\_users:**

Conté la informació dels usuaris (persones treballadores):

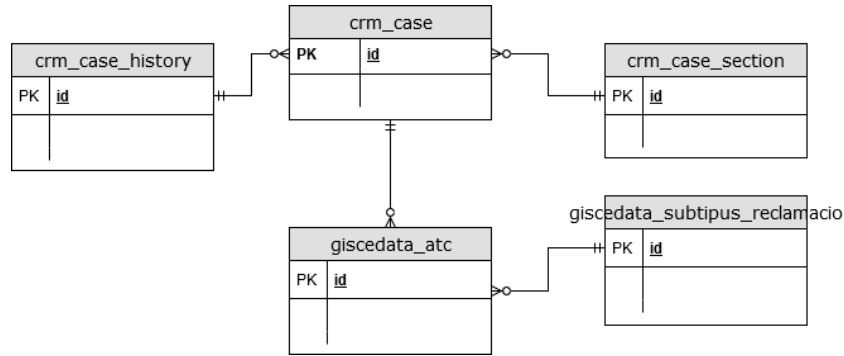
- id <key>
- name
- active
- login
- password
- context\_tz
- signature
- context\_lang
- action\_id <fk>
- create\_uid <fk>
- create\_date
- write\_date
- write\_uid <fk>
- menu\_id <fk>
- address\_id <fk>
- company\_id <fk>

Per altra banda, la taula on es desarien les Trucades Telefòniques sí que s'hauria de modificar i s'hauria de treballar més en el mòdul de l'ERP corresponent, per tal de poder desar els camps que més interessaria. Això implicaria també definir exactament quins tipus de trucades es volen definir i quins camps interessa guardar.

Pel que fa a les taules que es modifiquen, es farà un model entitat-relació simplificat per tal de veure gràficament les relacions i taules anteriorment mencionades.

La taula `crm_case` té `section_id` que és una clau forana a la taula `crm_case_section` i també té la clau forana a `crm_case_history`.

La taula `giscedata_atc` té clau forana a `giscedata_subtipus_reclamació` (`subtipus_id`) i a `crm_case` (`crm_id`).



## 8.2.4 Disseny de interfície

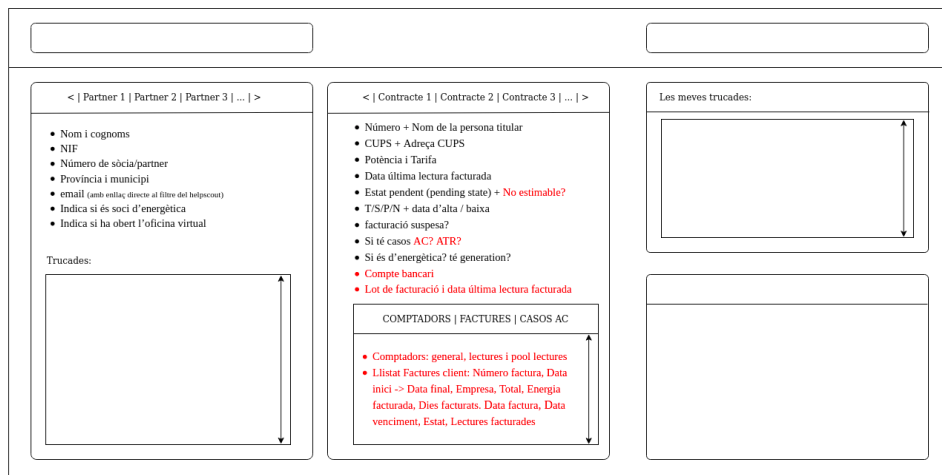
Des de l'inici, el disseny del Kalinfo ha vingut marcat per el disseny que es va escollir alhora de desenvolupar el Tomàtic. El caracteritza una interfície simple i funcional que separa en blocs concrets la informació que conté cada pàgina.

El disseny ha anat variant a mida que s'ha anat desenvolupant el projecte. Al principi estava molt focalitzat en només mostrar la informació.

En quant es va veure el potencial de respondre els qüestionaris per generar els casos de trucades telefòniques i de reclamacions, el disseny va canviar per donar molta prioritat al qüestionari.

Finalment, es va tornar a plantejar la interfície tenint en compte la funció principal del Kalinfo: mostrar informació del client que truca.

Des de llavors, l'esquema del disseny final és el següent:



*Esquema final de la interfície del Kalinfo*

Estaria format per dues parts:

- 1) **Barra de navegació:** on pots realitzar les diferents cerques, visualitzar el número entrant i on es trobarien els botons de bloquejar la pàgina, “new tab” i configuració.
- 2) **Informació:** De moment, trobaríem 3 components essencials on es visualitzaria la informació.

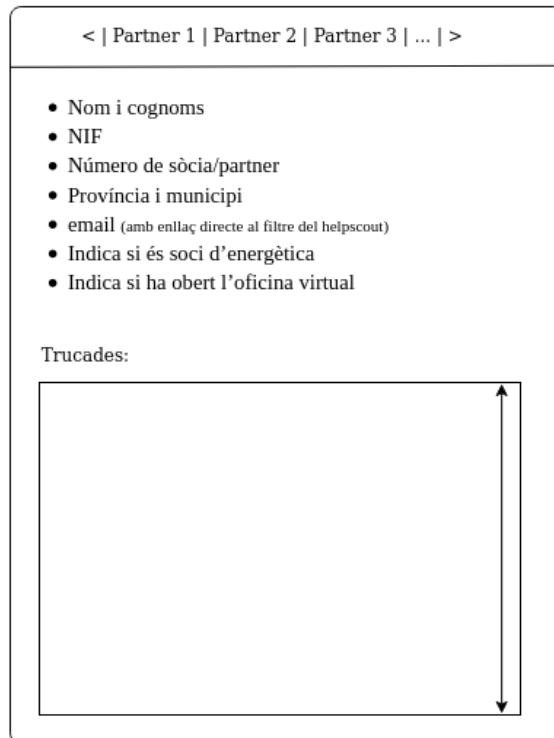
### Components:

- 1) Informació partner:

A la barra superior, es podria navegar entre els diferents partners relacionats amb el número entrant o la cerca realitzada.

Es mostraria la informació especificada.

Es veurien les trucades pròpies de cada partner registrades.



*Component de la informació partner*

- 2) Informació contracte:

A la barra superior, es podria navegar entre tots els diferents contractes relacionats amb el partner (com a titular, soci, pagador o notificador).

Es mostraria la informació especificada.

Es podria veure també la informació que es demana relativa a comptadors, factures i casos AC. Pel que fa a aquesta informació caldria limitar el nombre de factures i lectures de comptadors que es volen visualitzar. Carregar totes les factures, tots els casos i totes les lectures podria suposar una càrrega excessiva.

< | Contracte 1 | Contracte 2 | Contracte 3 | ... | >

- Número + Nom de la persona titular
- CUPS + Adreça CUPS
- Potència i Tarifa
- Data última lectura facturada
- Estat pendent (pending state) + **No estimable?**
- T/S/P/N + data d'alta / baixa
- facturació suspesa?
- Si té casos **AC? ATR?**
- Si és d'energètica? té generation?
- **Compte bancari**
- **Lot de facturació i data última lectura facturada**

COMPTADORS | FACTURES | CASOS AC

- **Comptadors: general, lectures i pool lectures**
- **Llistat Factures client: Número factura, Data inici -> Data final, Empresa, Total, Energia facturada, Dies facturats. Data factura, Data venciment, Estat, Lectures facturades**

*Component de la informació contracte*

### 3) Les meves trucades:

Per últim, es podrien visualitzar les últimes 20 trucades que la persona identificada ha atès i tornar a respondre el qüestionari de les que no he tingut temps (apareixeran en vermell).

Les meves trucades:

*Component de les trucades persona identificada*

Al capítol número 10 es podran veure els resultats. Aquest ha estat només l'esquema que s'ha definit i el resultat final presenta algunes petites variacions.

## 9. Implementació i proves

Aquest apartat es dedicarà sobretot a explicar sobre la implementació del backend: la relació de websockets, les funcions per extreure la informació que es mostrarà, les funcions per crear els casos d'atenció telefònica i l'script realitzat.

### 9.1 Connexions websockets

Es desarà la informació respecte els websockets a la API. La estructura de dades on es desaran les connexions websocket serà un diccionari: `websockets = {}`

Per tal de inicialitzar el client, primerament mira si el client concret ja estava identificat i el desidentifica. Seguidament, si no hi ha la extensió ja registrada, inicialitza l'array de connexions. Seguidament, afegeix el client a l'array de connexions de la extensió:

```
def initialize_client(client, server, extension):
    client_left(client, server)
    step("Identifying client as {}", extension)
    if extension not in websockets:
        websockets[extension] = []
    websockets[extension].append(client)
```

Es notifica que un nou usuari s'ha identificat. Per tal de fer-ho, obté totes les connexions segons la extensió. Si no n'hi ha cap, error: no es pot notificar a ningú. Per a cada client dels clients de connexions obtinguts anteriorment amb la extensió, envia un missatge especificant la identificació:

```
def say_new_user_logged(client, server, extension, iden):
    step("Saying to the page that now {} is there", iden)
    clients = websockets.get(extension, [])
    if not clients:
        error("Trying to send message to {} but has no client.", extension)
    for client in clients:
        app.wserver.send_message(client, "IDEN:" + iden)
```

Es notifica que la extensió ha canviat. Primerament, s'obtenen totes les connexions de la extensió concreta. Si no n'hi ha cap, error: no es pot enviar. Altrament, per a cada connexió, s'envia un missatge indicant la nova extensió.

```
def say_logcalls_has_changed(extension):
    clients = websockets.get(extension, [])
    if not clients:
        error("Trying to send message to {} but has no client.", extension)
    for client in clients:
        app.wserver.send_message(client, "REFRESH:" + extension)
```



Quan es rep un missatge. Es divideix segons el protocol acordat, separant per ":" i es desa al tipus de missatge que està a la primera posició. Si el tipus de missatge és de identificació, es recupera la extensió i la identificació i s'inicialitza el client i es comunica el nou usuari:

```
def on_message_recieved(client, server, message):
    divided_message = message.split(":")
    type_of_message = divided_message[0]
    if type_of_message == "IDEN":
        extension = divided_message[1]
        iden = divided_message[2]
        initialize_client(client, server, extension)
        say_new_user_logged(client, server, extension, iden)
    else:
        error("Type of message not recognized.")
```

Quan un client es desidentifica. Per a cada extensió desada al diccionari de websockets, si trobem la connexió del client, es desidentifica eliminant-lo de la taula.

```
def client_left(client, server):
    for extension in websockets:
        if client in websockets[extension]:
            step("Unidentifying client as {}", extension)
            websockets[extension].remove(client)
            break
    else:
        warn("New client")
```

Quan s'inicialitza el servidor. S'inicialitza el socket del servidor amb la informació desada al fitxer de configuració i es substitueixen les funcions per defecte amb les modificades.

```
def startCallInfoWS(app, host):
    app.wserver = WebSocketServer(
        port=CONFIG.websocket_port,
    )
    app.wserver.set_fn_message_received(on_message_recieved)
    app.wserver.set_fn_client_left(client_left)
    thread = Thread(target=app.wserver.run_forever)
    thread.start()
    return thread
```

## 9.2 Extreure la informació

Per tal d'extreure i buscar la informació concreta de partner i els seus contractes vinculats, s'han creat diferents funcions al fitxer callinfo.py. Es comentaran de manera general les més similars i les que s'aprofundirà en les que tinguin algun funcionament més complexa.

Per tal d'obtenir la informació i filtrar directament del ERP, s'han creat les diferents funcions (que segueixen la mateixa estructura):

```

def partnerByName(self, data):
    return self.O.ResPartner.search([
        ('name', 'ilike', data),
    ])

```

Tal com es pot veure es fa una cerca on es busca que el nom sigui semblant a “data”. Per el camp email, un partner pot tenir diferents adreces assignades:

```

def partnerByAddressId(self, address_ids):
    result = []
    if len(address_ids) == 0:
        return result
    return [
        address[0]
        for address
        in self.O.ResPartnerAddress.read(address_ids, 'partner_id')
        if address and address[0]
    ]

```

Primerament s’han cercat les adreces de la titular, si no en té, retorna 0. Altrament, retorna totes les adreces vinculades per a cada id.

La informació referent al partner s’obté amb la següent funció:

```

def partnerInfo(self, partner_data):
    result = ns(
        id_soci=self.anonymize(partner_data.ref) if partner_data.ref else "",
        lang=partner_data.lang,
        name=self.anonymize(partner_data.name),
        city=partner_data.www_municipi[1]['name'] if partner_data.www_municipi else "",
        email=self.anonymize(partner_data.www_email),
        state=partner_data.www_provincia[1]['name'] if partner_data.www_provincia else "",
        dni=self.anonymize(partner_data.vat),
        ov=partner_data.empowering_token != False,
        energetica=24 in partner_data.category_id,
    )
    return result

```

La funció que recupera la informació dels contractes segueix el mateix estil però ho fa per a cada contracte vinculat amb el partner.

Per tal d’obtenir els contractes relacionats:

```

def getPartnerRelatedContracts(self, partner_id):
    contract_tp_ids = self.O.GiscedataPolissa.search([
        ('titular.id', '=', partner_id),
        ('pagador.id', '=', partner_id),
    ])
    contract_titular_ids = self.O.GiscedataPolissa.search([
        ('titular.id', '=', partner_id),
    ])
    contract_pagador_ids = self.O.GiscedataPolissa.search([
        ('pagador.id', '=', partner_id),
    ])
    contract_soci_ids = self.O.GiscedataPolissa.search([
        ('soci.id', '=', partner_id),
    ])

    contracts = contract_tp_ids
    for contract_titular_id in contract_titular_ids:
        if contract_titular_id not in contracts:
            contracts.append(contract_titular_id)
    for contract_pagador_id in contract_pagador_ids:
        if contract_pagador_id not in contracts:
            contracts.append(contract_pagador_id)
    for contract_soci_id in contract_soci_ids:
        if contract_soci_id not in contracts:
            contracts.append(contract_soci_id)
    return contracts

```

Primerament es fa el filtre concret per a cada relació que hi pot haver entre partner i contracte (soci, titular, pagador i notificador). Els contractes principals són els que s'han trobat on el partner és el notificador. Seguidament els que hi està relacionat només com a titular, com a pagador i, finalment, com a soci.

També hi ha una funció anomenada `partnersInfo` per obtenir la informació del partner. Primerament s'obtenen els camps que estan dins el model `ResPanter` per cada partner id que tinguem, juntament amb la informació de tots els contractes que tingui vinculats.

Pel que fa a la informació al respecte del contracte, hi ha diferents funcions específiques per obtenir-la amb el format desitjat. Uns exemples serien les mostrades seguidament.

```

def hasOpenATR(contract_id, case):
    cases = self.O.GiscedataSwitching.search([
        ('cups_polissa_id', '=', contract_id),
        ('proces_id.name', '=', case),
        ('state', '!=', 'done')
    ])
    return len(cases) > 0

```

Ens indica si el contracte té casos ATR oberts. Per tal de saber-ho, cerca al model de GiscedataSwitching, quins casos tindrien el contracte id específic (vinculat al CUPS), amb el tipus de cas coincident amb l'especificat i l'estat del qual sigui diferent a "done" (finalitzat).

Per altra banda, també hi ha la funció que ens retornaria els casos concrets ATR que té oberts un contracte. Per tal de saber-los, utilitza el mateix model però enlloc de fer una cerca i retornar si s'han trobat resultats, busca els casos no finalitzats els quals el contracte vinculats al CUPS és coincident amb el que estem cercant. Retorna una llista amb el nom de cada un dels casos trobats.

```
def openCases(contract_id):
    cases = self.O.GiscedataSwitching.browse([
        ('cups_polissa_id', '=', contract_id),
        ('state', '!=', 'done')
    ])
    return [case.proces_id.name for case in cases]
```

Pel que fa a la resta d'informació sobre el contracte, s'obté també accedint al model de dades concret i filtrant per els camps pertinents. Un altre exemple que accedeix a un altre model seria la funció que ens retorna si un contracte té generationkWh:

```
def hasGeneration(contract_id):
    assignments = self.O.GenerationkwhAssignment.search([
        ('contract_id', '=', contract_id),
        ('end_date', '=', None),
    ])
    return len(assignments) > 0
```

També s'extreu informació sobre els comptadors i les factures:

```
def meterReadings(meter_ids):
    readings = []
    for meter_id in meter_ids:
        meter = getMeterReadings(meter_id)
        if not meter['active']:
            break
        if not meter['lectures']:
            break
        meter_readings_ids = meter['lectures']
        limited_meter_readings_ids = meter_readings_ids[
            :self.meter_readings_limit
        ]
        for reading_id in limited_meter_readings_ids:
            reading = getReading(reading_id)
            data = {
                'comptador': self.anonymize(meter['name']),
                'data': reading['name'],
                'periode': reading['periode'][1],
                'lectura': self.anonymize(str(reading['lectura'])),
                'origen': reading['origen_id'][1],
            }
            readings.append(data)
    return readings
```

Per tal d'obtenir la informació de les lectures de comptador, primerament obtenim els comptadors vinculats al contracte, ens assegurem que estigui actiu i tingui lectures i, un cop obtenim l'identificador, s'obtenen els camps que interessin de les lectures i es dona format a la informació.

Es fa el mateix per a la informació de les últimes factures:

```
def lastInvoices(contract_id):
    invoices = []
    last_invoices_ids = self.O.GiscedataFacturacioFactura.search([
        ('polissa_id', '=', contract_id),
        ('state', '!=', 'draft'),
        ('type', 'in', ['out_invoice', 'out_refund'])
    ][:self.invoices_limit])
    for invoice_id in last_invoices_ids:
        invoice = getInvoice(invoice_id)
        invoices.append({
            'number': self.anonymize(invoice['number']),
            'initial_date': invoice['data_inici'],
            'final_date': invoice['data_final'],
            'payer': self.anonymize(invoice['partner_id'][1]),
            'amount': invoice['amount_total'],
            'energy_invoiced': invoice['energia_kwh'],
            'days_invoiced': invoice['dies'],
            'invoice_date': invoice['date_invoice'],
            'due_date': invoice['date_due'],
            'state': invoice['state'],
        })
    return invoices
```

Per tal d'obtenir la informació en funció del camp de cerca, hi ha funcions específiques per a cada un dels casos:

```
def getByPhone(self, phone):
    address_ids = self.addressByPhone(phone)
    partners_ids = self.partnerByAddressId(address_ids)
    return self.getByPartnersId(partners_ids)
```

Com que el telèfon està dins el model de adreces, primer s'obté la informació de les adreces vinculades al telèfon i, seguidament, als partners en funció d'aquestes adreces. Es retorna la informació referent al partner amb el format pertinent.

Així doncs, per tal de poder filtrar per qualsevol dels camps (opció "Tot" del cercador), el que es fa a nivell de funció és cridar a cada una d'aquestes cerques per a camps específics i buscar la informació de tots els partners trobats.

```

def getByData(self, data):
    address_ids = self.addressByPhone(data)
    address_p_ids = self.partnerByAddressId(address_ids)
    email_ids = self.addressByEmail(data)
    email_p_ids = self.partnerByAddressId(email_ids)
    soci_p_ids = self.partnerBySoci(data)
    dni_p_ids = self.partnerByDni(data)
    name_p_ids = self.partnerByName(data)
    ids = address_p_ids + email_p_ids + soci_p_ids + dni_p_ids + name_p_ids
    return self.getByPartnersId(ids)

```

Es poden veure la resta de funcions concretes al [GitHub](#).

## 9.3 Creació dels casos

### 9.3.1 Creació de casos trucada telefònica

S'ha realitzat un anàlisi de l'estat del mòdul del ERP Trucades Telefòniques on es volen desar aquestes i s'ha vist que es volen desenvolupar noves funcionalitats i millorar-ne la optimització per tal d'emmagatzemar tota aquesta informació allà i poder-ne extreure amb funcions concretes les dades que es creguin oportunes.

Després d'aquest anàlisi, s'ha decidit no desar les trucades telefòniques fins que no es facin les millores pertinents per part de l'equip ERP de Som Energia o de Gisce.

### 9.3.2. Creació de casos d'atenció client

Per tal de crear els casos d'atenció telefònica, s'han creat les funcions pertinents al fitxer dins del repositori de Tomàtic anomenat claims.py. Seguidament, s'explicarà el fluxe de la funció principal:

```

def create_atc_case(self, case):
    ...
    Expected case:

    namespace(
        person:
            - date: D-M-YYYY H:M:S
            person: person
            reason: '['section.name'] 'claim.name', 'claim.desc''
            partner: partner number
            contract: contract number
            procedente: ''
            improcedente: ''
            solved: ''
            user: section.name
            cups: cups number
            observations: comments
            - ...
        ...
    )
    ...

```

Aquest és el format de case que s'espera per tal de que la funció pugui crear el cas correctament.

```
partner_id = partnerId(self.erp, case.partner)
partner_address = partnerAddress(self.erp, partner_id)
crm_section_id = crmSectionID(self.erp, case.user)
claim_section_id = claimSectionID(
    self.erp, case.reason.split('.')[1].strip()
)
```

Primerament s'obté l'identificador del partner en funció del partner del cas especificat. Es fa el mateix per l'adreça, l'identificador de secció de crm i de reclamació.

```
data_crm = {
    'section_id': crm_section_id,
    'name': sectionName(self.erp, claim_section_id),
    'canal_id': PHONE,
    'polissa_id': contractId(self.erp, case.contract),
    'partner_id': partner_id,
    'partner_address_id': partner_address.get('id'),
    'state': 'done' if case.solved else 'open',
    'user_id': ''
}
crm_id = self.erp.CrmCase.create(data_crm).id
```

Seguidament, s'emplena la informació necessària per a crear un cas CRM (necessari per crear un cas del tipus reclamació).

```
data_history = {
    'case_id': crm_id,
    'description': case.observations
}
crm_history_id = self.erp.CrmCaseHistory.create(data_history).id
```

Es desa l'identificador del cas creat a la estructura de dades que s'utilitzarà per crear un registre de cas CRM. Aquesta és una acció que actualment han de realitzar manualment generant un registre a la pantalla de casos CRM un cop l'han creat.

Finalment és crea el cas d'atenció al client (reclamació) també especificant els camps necessaris a la estructura de dades que es passarà a la funció que crearà el cas internament.

Actualment romandrà com a usuari el que ha establert la connexió amb l'ERP ja que, per tal de identificar els treballadors de tipus usuari, s'hauria de disposar del email dins la fitxa usuari de cada usuari del ERP i aquest treball no s'ha fet encara a dia d'avui. En quant es vulgui mantenir el registre de quina persona crea el cas dins de l'ERP (que seria la persona que emplena el qüestionari del Kalinfo), s'haurà d'actualitzar aquesta informació.

El codi està preparat per fer-ho, però mentre no s'actualitza roman comentat.

```

data_atc = {
    'provincia': partner_address.get('state_id')[0],
    'total_cups': 1,
    'cups_id': cupsId(self.erp, case.cups),
    'subtipus_id': claim_section_id,
    'reclamante': RECLAMANTE,
    'resultat': resultat(
        self.erp,
        case.procedente,
        case.improcedente
    ) if case.solved else "",
    'date': case.date,
    'email_from': partner_address.get('email'),
    'time_tracking_id': COMERCIALIZADORA
}
# user_id = userId(self.erp, self.emails, case.person)
# if user_id:
#     data_crm['create_uid'] = user_id
data_atc['crm_id'] = crm_id
case = self.erp.GiscedataAtc.create(data_atc)

return case.id

```

Es poden veure la resta de funcions implementades també al [GitHub](#).

## 9.4 Script per creació de casos

L'script que s'executa cada dia després de l'horari d'atenció telefònica, és l'anomenat `create_atc_cases.py`, també dins del repositori del Tomàtic.

L'script comença llegint els paràmetres i establint la configuració en funció d'aquests i els fitxers pertinents. El fluxe principal, vindria a ser:

```

def main(yaml_directory, current_date):
    erp = erppeek.Client(**dbconfig.erppeek)
    claims = Claims(erp)

    atc_yaml_file = "{}.yaml".format(current_date)
    atc_yaml_path = os.path.join(yaml_directory, atc_yaml_file)

```

Primerament, estableix la connexió amb l'ERP. Seguidament genera l'objecte importat del fitxer que crea les reclamacions amb la connexió establerta prèviament. En funció de la data, genera el nom del fitxer que haurà de llegir (seguint el protocol que es segueix per desar el fitxer).



```

try:
    atc_yaml = ns.load(atc_yaml_path)
except Exception as e:
    logging.error(" Can't load file {}: {}".format(
        atc_yaml_file,
        str(e))
    )
return

```

Intenta carregar el fitxer. Si no es pot, es desa l'error al log especificant el nom del fitxer que no s'ha pogut obrir i l'error que ha retornat. Altrament:

```

for person in atc_yaml:
    for case in atc_yaml[person]:
        try:
            case_id = claims.create_atc_case(case)
            logging.info(" Case {} created.".format(case_id))
        except Exception as e:
            logging.error(" Something went wrong in {}: {}".format(
                atc_yaml_file,
                str(e))
            )
            logging.error(" CASE: {}".format(case))

```

Per a cada persona que estigui en el fitxer, per a cada cas d'aqueta persona, crea el cas enviant la informació d'aquest a la funció de l'objecte que hem inicialitzat prèviament.

Si hi ha algun error, es desa l'error al log especificant el fitxer i el missatge d'error. També s'inclou el cas concret que ha donat l'error.

El codi complet també es pot trobar al [GitHub](#).

## 9.5 API

La manera amb la que interactuarà el frontend del Tomàtic amb l'ERP i la centraleta per reunir la informació, serà a través de l'existent API desenvolupada en Flask.

Les respostes són en format YAML. S'han afegit els següents endpoints:

@app.route('/api/info/phone/<phone>', methods=['GET'])	
<b>Nom</b>	getInfoPersonByPhone
<b>Descripció</b>	Obtenció de la informació de persona segons el camp telèfon
<b>Paràmetres</b>	<b>Query:</b> <phone>: telèfon

**Resposta**

Quan hi ha informació:

info:

info:

partners:

- id\_soci: S\_\_\_\_\_

lang: ca\_ES

name: Fernández Vergés, Marta

city: Girona

email: marta.fernandez@somenergia.cop

state: Girona

dni: \_\_\_\_\_

ov: true

energetica: false

contracts:

- start\_date: '2020-9-02'

end\_date: ''

power: 3.3

cups: \_\_\_\_\_

faré: 2.0A

state: activa

number: '\_\_\_\_\_'

last\_invoiced: '2020-09-02'

suspended\_invoicing: false

pending\_state: Correct

open\_cases: []

is\_titular: true

is\_partner: true

is\_notifier: true

is\_payer: true

cups\_address: \_\_\_\_\_

titular\_name: Fernández Vergés, Marta

energetica: false

generation: false

iban: \_\_\_\_\_

lot\_facturacio: 09/2020

no\_estimable: true

lectures\_comptador:

- comptador: '\_\_\_\_\_'

data: '2020-09-02'

periode: 2.0A (P1)

lectura '1234'

origen: Telegestió

invoices:

- number: FE\_\_\_\_\_

initial\_date: '2020-09-02'

final\_date: '2020-09-02'

payer: Fernández Vergés, Marta

amount: 20.02

energy\_invoiced: 30

days\_invoiced: 20

invoice\_date: '2020-09-02'

due\_date: '2020-09-02'

State: paid

message: ok

	<p>Quan no es troben resultats:</p> <pre>info:   info:     partners: null   message: no_info</pre> <p>Quan hi ha massa informació:</p> <pre>info:   info:     partners: Masses resultats   message: response_too_long</pre>
--	---

@app.route('/api/info/name/<name>', methods=['GET'])	
<b>Nom</b>	getInfoPersonByName
<b>Descripció</b>	Obtenció de la informació de persona segons el camp nom
<b>Paràmetres</b>	<p><b>Query:</b></p> <pre>&lt;name&gt;: nom</pre>
<b>Resposta</b>	<p>Quan hi ha informació:</p> <pre>info:   info:     partners:       - id_soci: S_____         lang: ca_ES         name: Fernández Vergés, Marta         city: Girona         email: marta.fernandez@somenergia.cop         state: Girona         dni: _____         ov: true         energetica: false         contracts:           - start_date: '2020-9-02'             end_date: ''             power: 3.3             cups: _____             faré: 2.0A             state: activa             number: '_____',             last_invoiced: '2020-09-02'             suspended_invoicing: false             pending_state: Correct             open_cases: []             is_titular: true             is_partner: true             is_notifier: true</pre>

	<pre> is_payer: true cups_address: _____ titular_name: Fernández Vergés, Marta energetica: false generation: false iban: _____ lot_facturacio: 09/2020 no_estimable: true lectures_comptador: - comptador: '_____'   data: '2020-09-02'   periode: 2.0A (P1)   lectura '1234'   origen: Telegestió invoices: - number: FE_____   initial_date: '2020-09-02'   final_date: '2020-09-02'   payer: Fernández Vergés, Marta   amount: 20.02   energy_invoiced: 30   days_invoiced: 20   invoice_date: '2020-09-02'   due_date: '2020-09-02'   State: paid message: ok </pre>
	<p>Quan no es troben resultats:</p> <pre> info:   info:     partners: null   message: no_info </pre>
	<p>Quan hi ha massa informació:</p> <pre> info:   info:     partners: Masses resultats   message: response_too_long </pre>

@app.route('/api/info/nif/<nif>', methods=['GET'])	
<b>Nom</b>	getInfoPersonByNif
<b>Descripció</b>	Obtenció de la informació de persona segons el camp nif
<b>Paràmetres</b>	<b>Query:</b> <nif>: nif

**Resposta**

Quan hi ha informació:

info:

info:

partners:

- id\_soci: S\_\_\_\_\_

lang: ca\_ES

name: Fernández Vergés, Marta

city: Girona

email: marta.fernandez@somenergia.cop

state: Girona

dni: \_\_\_\_\_

ov: true

energetica: false

contracts:

- start\_date: '2020-9-02'

end\_date: ''

power: 3.3

cups: \_\_\_\_\_

faré: 2.0A

state: activa

number: '\_\_\_\_\_'

last\_invoiced: '2020-09-02'

suspended\_invoicing: false

pending\_state: Correct

open\_cases: []

is\_titular: true

is\_partner: true

is\_notifier: true

is\_payer: true

cups\_address: \_\_\_\_\_

titular\_name: Fernández Vergés, Marta

energetica: false

generation: false

iban: \_\_\_\_\_

lot\_facturacio: 09/2020

no\_estimable: true

lectures\_comptador:

- comptador: '\_\_\_\_\_'

data: '2020-09-02'

periode: 2.0A (P1)

lectura '1234'

origen: Telegestió

invoices:

- number: FE\_\_\_\_\_

initial\_date: '2020-09-02'

final\_date: '2020-09-02'

payer: Fernández Vergés, Marta

amount: 20.02

energy\_invoiced: 30

days\_invoiced: 20

invoice\_date: '2020-09-02'

due\_date: '2020-09-02'

State: paid

message: ok

	<p>Quan no es troben resultats:</p> <pre>info:   info:     partners: null   message: no_info</pre> <p>Quan hi ha massa informació:</p> <pre>info:   info:     partners: Masses resultats   message: response_too_long</pre>
--	---

@app.route('/api/info/soci/<iden>', methods=['GET'])	
<b>Nom</b>	getInfoPersonBySoci
<b>Descripció</b>	Obtenció de la informació de persona segons el camp número de sòcia
<b>Paràmetres</b>	<p><b>Query:</b></p> <p>&lt;iden&gt;: número de sòcia</p>
<b>Resposta</b>	<p>Quan hi ha informació:</p> <pre>info:   info:     partners:       - id_soci: S_____         lang: ca_ES         name: Fernández Vergés, Marta         city: Girona         email: marta.fernandez@somenergia.cop         state: Girona         dni: _____         ov: true         energetica: false         contracts:           - start_date: '2020-9-02'             end_date: ''             power: 3.3             cups: _____             faré: 2.0A             state: activa             number: '_____',             last_invoiced: '2020-09-02'             suspended_invoicing: false             pending_state: Correct             open_cases: []             is_titular: true             is_partner: true             is_notifier: true</pre>

	<pre> is_payer: true cups_address: _____ titular_name: Fernández Vergés, Marta energetica: false generation: false iban: _____ lot_facturacio: 09/2020 no_estimable: true lectures_comptador: - comptador: '_____'   data: '2020-09-02'   periode: 2.0A (P1)   lectura '1234'   origen: Telegestió invoices: - number: FE_____   initial_date: '2020-09-02'   final_date: '2020-09-02'   payer: Fernández Vergés, Marta   amount: 20.02   energy_invoiced: 30   days_invoiced: 20   invoice_date: '2020-09-02'   due_date: '2020-09-02'   State: paid message: ok </pre>
	<pre> Quan no es troben resultats: info:   info:     partners: null   message: no_info </pre>
	<pre> Quan hi ha massa informació: info:   info:     partners: Masses resultats   message: response_too_long </pre>

@app.route('/api/info/email/<email>', methods=['GET'])	
<b>Nom</b>	getInfoPersonByEmail
<b>Descripció</b>	Obtenció de la informació de persona segons el camp email
<b>Paràmetres</b>	<b>Query:</b> <email>: email

**Resposta**

Quan hi ha informació:

info:

info:

partners:

- id\_soci: S\_\_\_\_\_

lang: ca\_ES

name: Fernández Vergés, Marta

city: Girona

email: marta.fernandez@somenergia.cop

state: Girona

dni: \_\_\_\_\_

ov: true

energetica: false

contracts:

- start\_date: '2020-9-02'

end\_date: ''

power: 3.3

cups: \_\_\_\_\_

faré: 2.0A

state: activa

number: '\_\_\_\_\_'

last\_invoiced: '2020-09-02'

suspended\_invoicing: false

pending\_state: Correct

open\_cases: []

is\_titular: true

is\_partner: true

is\_notifier: true

is\_payer: true

cups\_address: \_\_\_\_\_

titular\_name: Fernández Vergés, Marta

energetica: false

generation: false

iban: \_\_\_\_\_

lot\_facturacio: 09/2020

no\_estimable: true

lectures\_comptador:

- comptador: '\_\_\_\_\_'

data: '2020-09-02'

periode: 2.0A (P1)

lectura '1234'

origen: Telegestió

invoices:

- number: FE\_\_\_\_\_

initial\_date: '2020-09-02'

final\_date: '2020-09-02'

payer: Fernández Vergés, Marta

amount: 20.02

energy\_invoiced: 30

days\_invoiced: 20

invoice\_date: '2020-09-02'

due\_date: '2020-09-02'

State: paid

message: ok



	<p>Quan no es troben resultats:</p> <pre>info:   info:     partners: null   message: no_info</pre> <p>Quan hi ha massa informació:</p> <pre>info:   info:     partners: Masses resultats   message: response_too_long</pre>
--	---

@app.route('/api/info/all/<field>', methods=['GET'])	
<b>Nom</b>	getInfoPersonBy
<b>Descripció</b>	Obtenció de la informació de persona segons qualsevol dels camps anteriorment citats.
<b>Paràmetres</b>	<p><b>Query:</b></p> <pre>&lt;field&gt;: camp</pre>
<b>Resposta</b>	<p>Quan hi ha informació:</p> <pre>info:   info:     partners:       - id_soci: S_____         lang: ca_ES         name: Fernández Vergés, Marta         city: Girona         email: marta.fernandez@somenergia.cop         state: Girona         dni: _____         ov: true         energetica: false         contracts:           - start_date: '2020-9-02'             end_date: ''             power: 3.3             cups: _____             faré: 2.0A             state: activa             number: '_____',             last_invoiced: '2020-09-02'             suspended_invoicing: false             pending_state: Correct             open_cases: []             is_titular: true             is_partner: true             is_notifier: true</pre>

	<pre> is_payer: true cups_address: _____ titular_name: Fernández Vergés, Marta energetica: false generation: false iban: _____ lot_facturacio: 09/2020 no_estimable: true lectures_comptador: - comptador: '_____',   data: '2020-09-02'   periode: 2.0A (P1)   lectura '1234'   origen: Telegestió invoices: - number: FE_____   initial_date: '2020-09-02'   final_date: '2020-09-02'   payer: Fernández Vergés, Marta   amount: 20.02   energy_invoiced: 30   days_invoiced: 20   invoice_date: '2020-09-02'   due_date: '2020-09-02'   State: paid message: ok </pre>
	<p>Quan no es troben resultats:</p> <pre> info:   info:     partners: null   message: no_info </pre>
	<p>Quan hi ha massa informació:</p> <pre> info:   info:     partners: Masses resultats   message: response_too_long </pre>

@app.route('/api/info/ringring', methods=['POST'])	
<b>Nom</b>	callingPhone
<b>Descripció</b>	Actualitzar quin número està trucant a quina extensió.
<b>Paràmetres</b>	<p><b>Body:</b></p> <pre> {   phone: 616161616   ext: 1234 } </pre>

<b>Resposta</b>	<p>Quan es notifica a una persona:</p> <pre>info:   notified: 1   phone: '616161616'   ext: '1234'</pre>
	<p>Quan no es notifica a ningú:</p> <pre>info:   notified: 0   phone: '616161616'   ext: '1234'</pre>

<b>@app.route('/api/socketInfo', methods=['GET'])</b>	
<b>Nom</b>	getConnectionInfo
<b>Descripció</b>	Retorna l'adreça on està escoltant el socket servidor.
<b>Paràmetres</b>	
<b>Resposta</b>	<pre>info:   port_ws: 4445   message: ok</pre>

<b>@app.route('/api/callReasons/&lt;info_type&gt;', methods=['GET'])</b>	
<b>Nom</b>	getReasonsInfo
<b>Descripció</b>	Obtenció dels motius de trucada segons d els casos d'info o de les reclamacions.
<b>Paràmetres</b>	<p><b>Query:</b></p> <p>&lt;info_type&gt;: tipus d'informació que es demana</p>
<b>Estat</b>	<p>Quan s'ha pogut obtenir la informació:</p> <pre>info:   info: [[motiu 1], [ motiu 2]...]   message: ok</pre> <p>Quan no s'ha pogut obtenir la informació:</p> <pre>info:   info: []   message: error_fet_fullsheet</pre>

@app.route('/api/infoReasons', methods=['POST'])	
<b>Nom</b>	savePhoneInfosLog
<b>Descripció</b>	Actualitzar el log de trucades info amb la informació especificada al body.
<b>Paràmetres</b>	<b>Body:</b> <pre>{   date: '2020-09-02',   person: marta,   phone: 616161616,   reason: "[INFO] Motiu de la trucada",   extra: "No entén la factura" }</pre>
<b>Resposta</b>	<p>Quan tot ha anat bé:</p> <pre>info:   message: ok</pre> <p>Si hi ha algun error:</p> <pre>info:   message: 'error_add_to_las_row'</pre>

@app.route('/api/claimReasons', methods=['POST'])	
<b>Nom</b>	savePhoneClaimsLog
<b>Descripció</b>	Actualitzar llistat de reclamacions amb la informació especificada al body.
<b>Paràmetres</b>	<b>Body:</b> <pre>{   date: '2020-09-02',   person: marta,   partner: '_____',   contract: '_____',   cups: '_____',   user: FACTURA,   procedente: false,   improcedente: false,   solved: true,   phone: 616161616,   reason: "[FACTURA] Motiu de la reclamació",   observations: "No entén la factura" }</pre>

<b>Resposta</b>	Quan tot ha anat bé: info: message: ok
	Si hi ha algun error: info: message: 'error_add_to_las_row'

@app.route('/api/log/<phone>', methods=['GET'])	
<b>Nom</b>	getPhoneLog
<b>Descripció</b>	Obtenció del log de trucades realitzades per el telèfon especificat.
<b>Paràmetres</b>	<b>Query:</b> <phone>: telèfon
<b>Resposta</b>	Quan tot ha anat bé: info: message: ok info: [[date, person, phone, reason, extra], [...], ...]
	Si hi ha algun error: info: message: 'error_get_fullsheet' info: []

@app.route('/api/personlog/<ext>', methods=['GET'])	
<b>Nom</b>	getMyLog
<b>Descripció</b>	Obtenció del log de trucades ateses per l'usuari identificat.
<b>Paràmetres</b>	<b>Query:</b> <ext>: extensió de telèfon
<b>Resposta</b>	Quan tot ha anat bé: info: message: ok info: - data: '2020-09-02' telefon: 616161616 partner: '____' contracte: '____' motius: '[RECLAMACIONS] Descripcio reclamacio'

	<p>Si hi ha algun error:</p> <pre>info:   message: 'no_register_yet'   info: ""</pre>
--	---

@app.route('/api/updateslog/<ext>', methods=['POST'])	
<b>Nom</b>	updateMyLog
<b>Descripció</b>	Actualitzar el log de trucades ateses per l'usuari identificat segons extensió.
<b>Paràmetres</b>	<p><b>Query:</b>          &lt;ext&gt;: extensió de telèfon</p> <p><b>Body:</b></p> <pre>{   data: '2020-09-02'   telefon: 616161616   partner: '_____'   contracte: '_____'   motius: '[RECLAMACIONS] Descripcio reclamacio'</pre>
<b>Resposta</b>	<p>Quan tot ha anat bé:</p> <pre>info:   message: ok</pre> <p>Si hi ha algun error:</p> <pre>info:   message: 'error_update_log'</pre>

@app.route('/api/updateClaims', methods=['GET'])	
<b>Nom</b>	updateClaims
<b>Descripció</b>	Actualitza els possibles diferents motius de casos reclamacions.
<b>Paràmetres</b>	
<b>Resposta</b>	<p>Quan tot ha anat bé:</p> <pre>info:   message: ok</pre> <p>Si hi ha algun error:</p> <pre>info:   message: 'error'</pre>

@app.route('/api/getClaims', methods=['GET'])	
<b>Nom</b>	getClaims
<b>Descripció</b>	Obtenció de tots els possibles motius de reclamacions per trucada.
<b>Paràmetres</b>	
<b>Resposta</b>	<p>Quan tot ha anat bé:</p> <pre>info:   message: ok   claims: [ "Claim1", "Claim2", ... ]   dict: { "comptador": ["Claim1", "Claim4"] }</pre> <p>Si hi ha algun error:</p> <pre>info:   message: 'error'   claims: []   dict: {}</pre>

@app.route('/api/atrCase', methods=['POST'])	
<b>Nom</b>	postAtrCase
<b>Descripció</b>	Afegeix el cas de reclamació del body al registre de reclamacions.
<b>Paràmetres</b>	<p><b>Body:</b></p> <pre>{   date: '2020-09-02',   person: marta,   reason: "[FACTURA] Motiu reclamacio",   partner: '_____',   contract: '_____',   procedente: false,   improcedente: false,   solved: true,   user: "FACTURA",   cups: '_____',   observations: "Comentaris afegits" }</pre>
<b>Resposta</b>	<p>Quan tot ha anat bé:</p> <pre>info:   message: ok</pre> <p>Si hi ha algun error:</p> <pre>info:   message: 'error'</pre>

## 9.6 Desenvolupament del projecte

S'ha treballat per rondes. L'eina que s'ha utilitzat per a fer el seguiment ha estat el Trello.

Inicialment, hi havia les següents columnes:

- **Rebot:** Es desen totes les targetes que es van creant i es van ordenant segons prioritats.
- **Ronda X (data inici – data final):** Conté les targetes que han entrat per treballar-se durant aquella ronda.
- **Doing:** Targeta amb la que s'està treballant.
- **Bloquejat:** Targetes que no es poden seguir per algun motiu en concret que s'especifica a la secció de comentaris. Es vetlla perquè la targeta es desbloquegi i s'hi pugui seguir treballant el més aviat possible.
- **Pendent de testeig:** Targetes que requereixen de que es testegin.
- **Pendent de posar a productiu:** Quan la targeta requereix fer un deploy.
- **Pendent de comunicar als representants IT:** Quan s'han de comunicar els canvis realitzats i actualitzats als representants IT.
- **Done Ronda X:** Un cop s'ha finalitzat la feina de la targeta i s'ha testejat, s'ha fet el deploy si calia, i s'ha comunicat als representants, pot passar al Done.

Per a cada Ronda completada, s'ha afegit una nova columna de Done.

A cada ronda s'ha assolit una part de la feina a realitzar segons la planificació inicial. Es detallarà a l'annex 1 a trets generals, aquesta feina realitzada i quan es va fer. Aquesta informació s'ha extret del Trello del Kalinfo.

Es citaran les targetes completades a cada ronda. El format del títol de les targetes segueix l'estil:

(2/2) [API] Crear nou endpoint

(Temps real dedicat en mitjos dies/ Temps estimat de dedicació en mitjos dies) [Paquet de treball segons la tasca de la targeta] Tasca a fer

Dins la targeta s'inclou una descripció més ampla del treball a realitzar i, si cal, un checklist.

En els comentaris de la targeta s'ha anat indicant el progrés i problemes trobats. També si la targeta canvia de columna, per exemple, a bloquejat, el motiu del bloqueig i la feina que requereix per tal de que es pugui seguir treballant-hi.

## 9.7 Proves realitzades

Totes les funcions dels diferents apartats han estat testejades alhora que desenvolupades. El codi s'ha realitzat amb TDD i, per tant, s'han inclòs els tests necessaris d'unittest per tal de garantir el seu funcionament i regular els canvis i afectacions de futurs desenvolupaments.

Pel que fa a la informació sensible d'aquests tests, s'ha utilitzat una anonimització de les dades sensibles per tal de que, en els tests, només es veiessin els últims tres dígit i es comparés aquesta informació ens els tests b2b data.



## 10. Implantació i resultats

En aquest apartat es parlarà de la implantació i els resultats obtinguts finalment.

### 10.1 Implantació

El sistema s'ha pogut implementar i actualment està a productiu la part que permet visualitzar la informació i respondre el qüestionari de reclamacions.

Pel que fa a la de generació de casos, la part que permetria desar els casos AC responnent el qüestionari està finalitzada juntament amb l'script que els crea al ERP.

Per altra banda, falta la part de desar els casos de Trucada Telefònica al ERP i la d'extreure estadístiques d'aquesta informació.

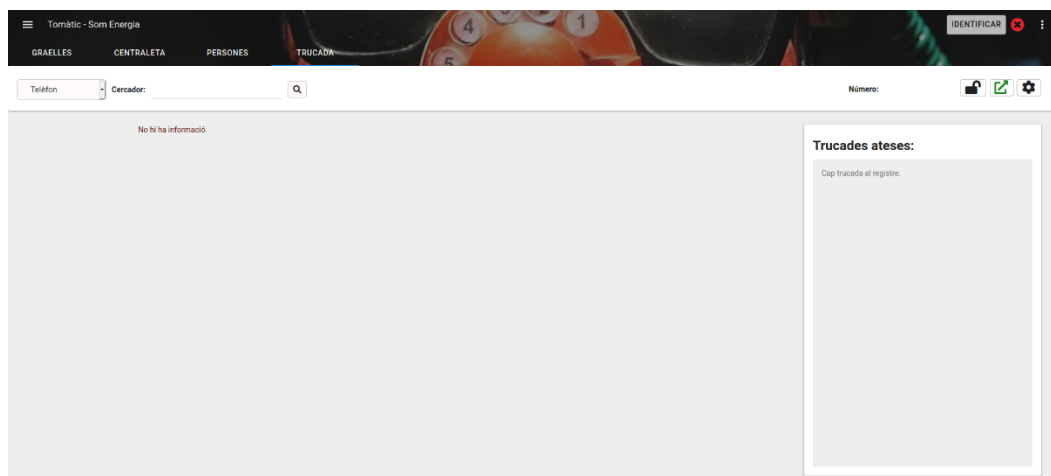
Tot i així, s'han realitzat els anàlisis corresponents i s'ha decidit que l'eina que s'utilitzarà per mostrar la informació un cop es tingui la informació dels tipus de trucades serà el [Superset](#).

Pel que fa al mòdul del ERP de Trucades Telefòniques, cal desenvolupar-lo millor abans de començar a desar-hi les trucades. S'hi treballarà properament.

### 10.2 Resultats

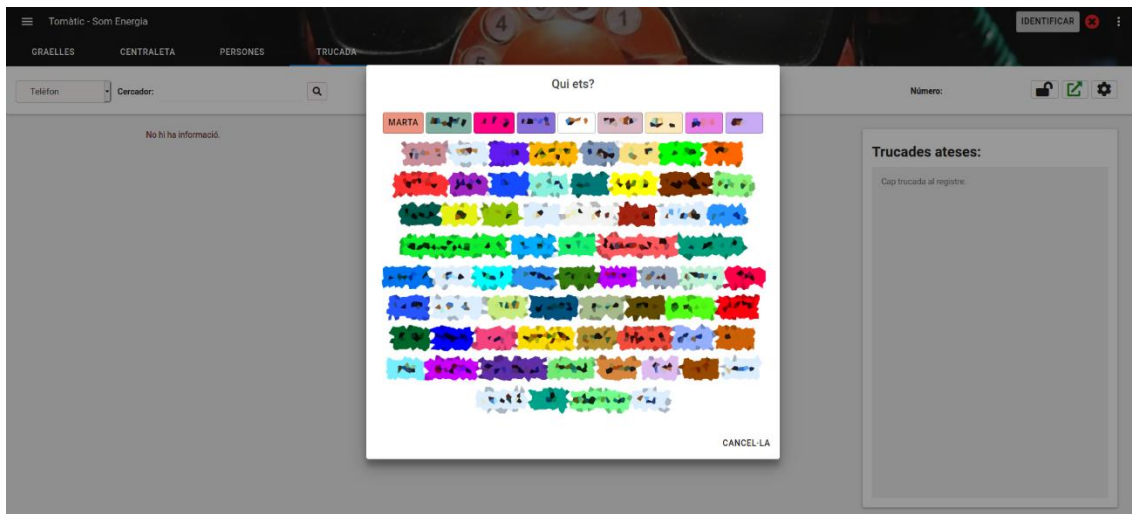
Seguidament mostraré les captures del resultat final del Kalinfo mentre explico una mica el fluxe que permet seguir l'eina. La informació sensible ha estat pixelada.

Quan s'obre el "Tomàtic" i es va a la pestanya "Trucada", es veu el següent:



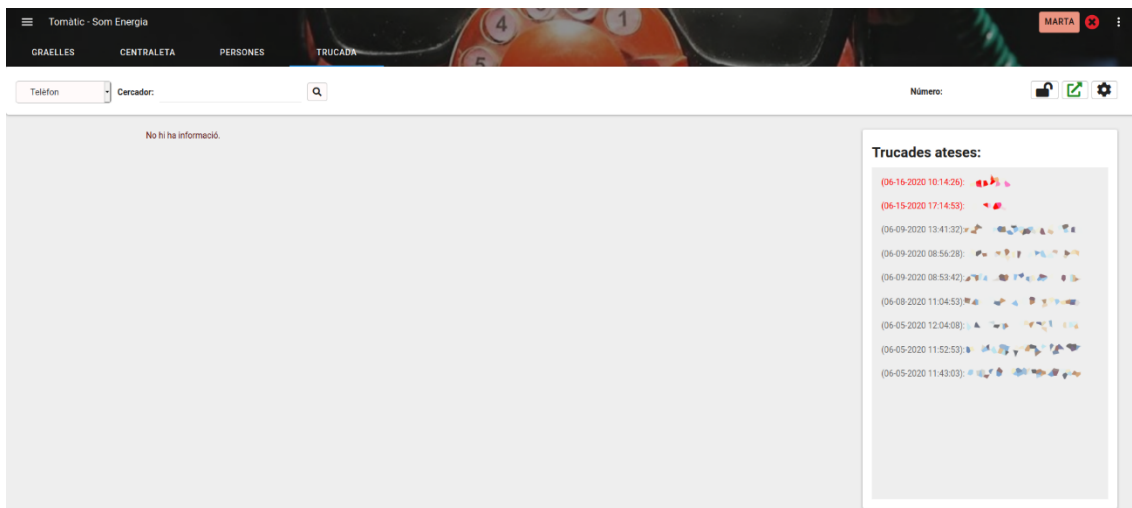
*Pantalla inicial Kalinfo*

Des d'aquí pots tant realitzar alguna cerca amb el cercador de la part superior esquerra com identificar-se. Passarem a indicar l'usuari clicant al botó de la part superior dreta on posa "Identificar". Apareixerà el següent:



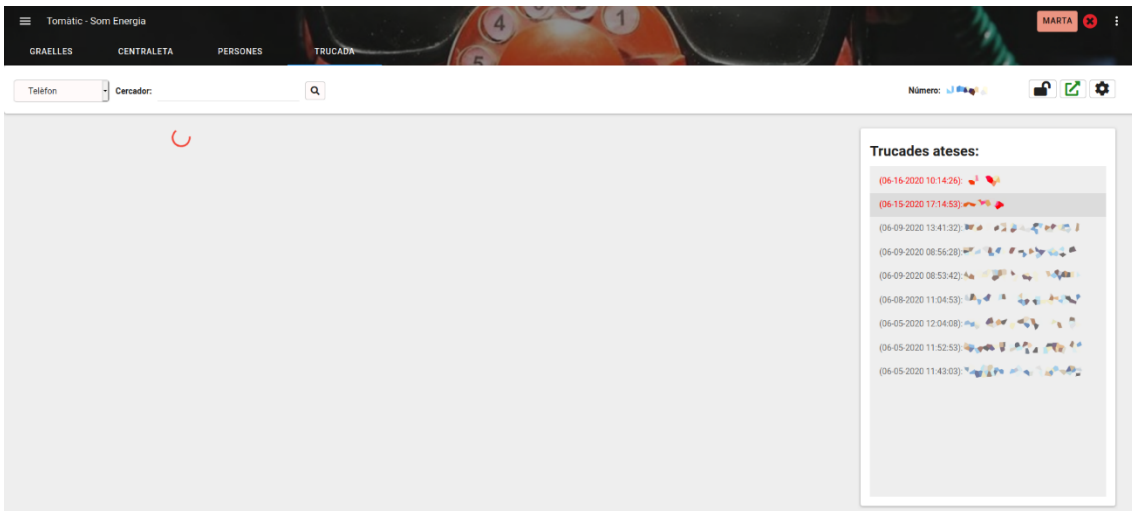
*Dialog que permet indicar el teu usuari*

Seleccionarem l'usuari "Marta" per a realitzar les següents proves. Un cop cliquem el botó del nostre usuari, es tancarà el dialog i es mostrarà altre cop la pantalla inicial però amb els següents canvis:

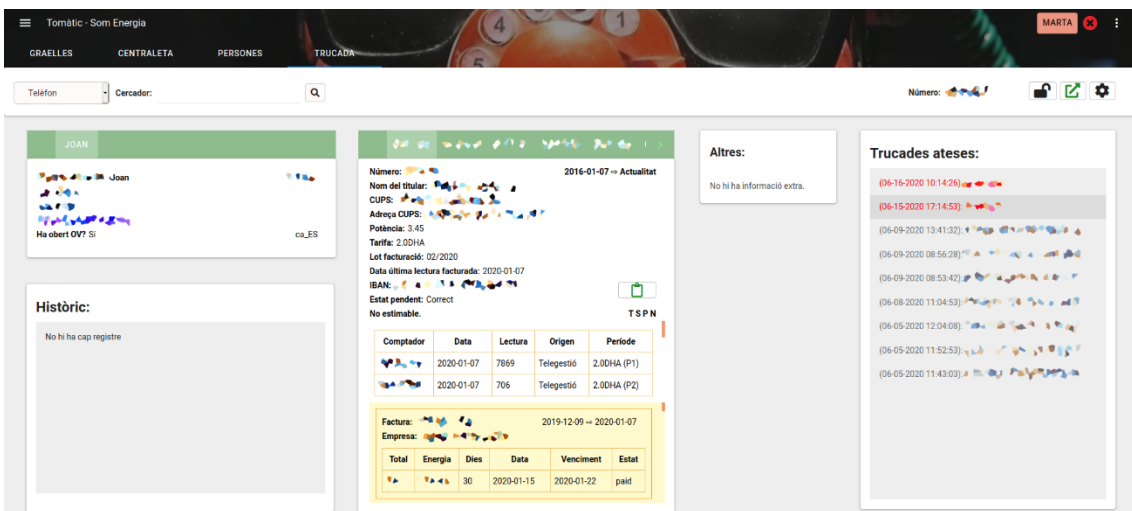


*Pantalla inicial per a usuari identificat*

Com podem observar, “Trucades ateses” ja no apareix buit. Ara conté el llistat de les últimes trucades rebudes amb la data, el número si no ha estat a través d’una cerca i el número de sòcia i de contracte sobre el qual s’ha respost. Les que estan en vermell es poden clicar per “recuperar-les”. Són les quals no s’ha emplenat el qüestionari. Si en cliquem una veurem com carrega la informació i finalment es mostra:



*Està carregant la informació*

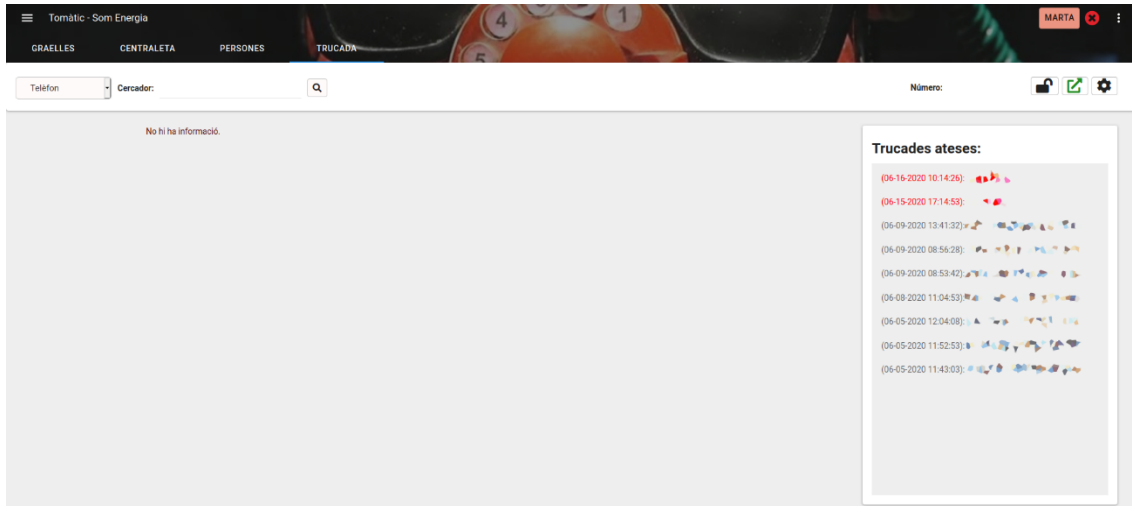


*Trucada recuperada*

Tal com podem observar, es mostra la informació del client a la part superior esquerra. A la part inferior hi ha el seu històric de trucades, actualment inactiu. Al mig, podem veure tots el contractes associats al partner que tenim seleccionat i la informació d’aquest. Al costat, hi ha un altre petit component que mostra informació extra sobre el contracte.

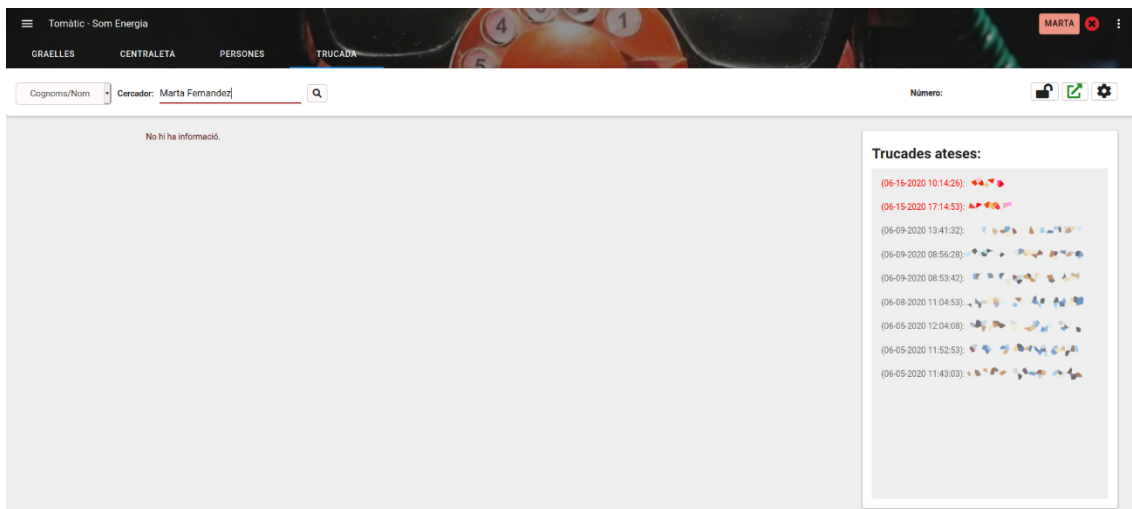
En l'anterior versió del Kalinfo, quan es desaven les respostes del Kalinfo a la fulla de càlcul, es recuperaven les respostes del partner concret i es mostraven en aquest històric. Actualment, es desaran al mòdul de trucades telefòniques i s'hauran de recuperar d'allà. Com que ja no es desen a la fulla de càlcul, s'ha desactivat aquest component.

Si deseleccionem la trucada, tornem a la imatge inicial:



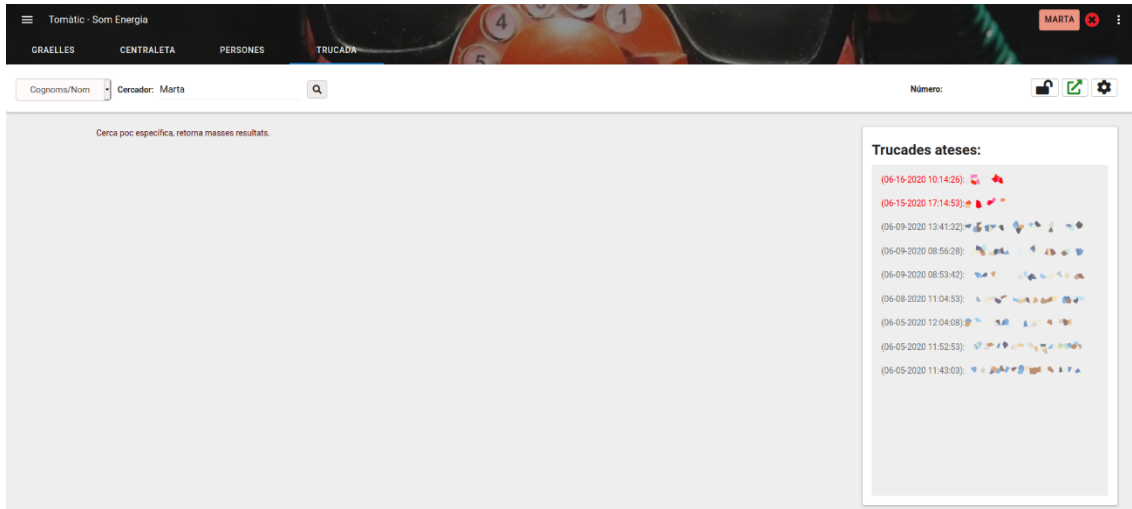
*Trucada deseleccionada*

Si realitzem alguna cerca i no hi ha informació, obtenim el següent missatge:



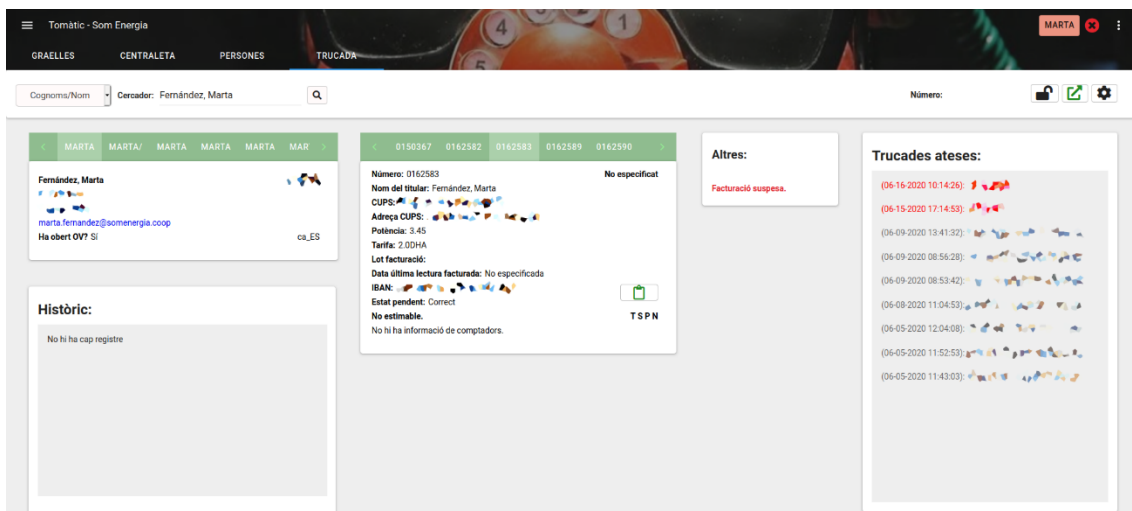
*No hi ha informació*

Si realitzem una cerca poc específica, apareix el següent missatge:



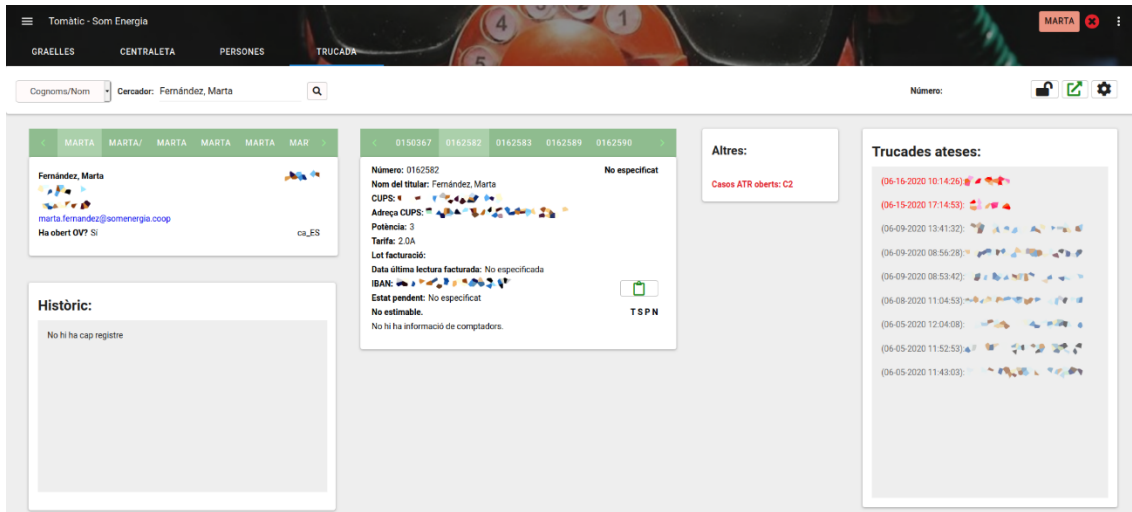
*Cerca poc específica*

Quan es realitza una cerca correcta, apareix la mateixa informació que la que hem vist quan hem recuperat una trucada. Si s'actualitzés la informació a través del refresc automàtic, també es mostraria de la mateixa manera:



*Cerca amb èxit*

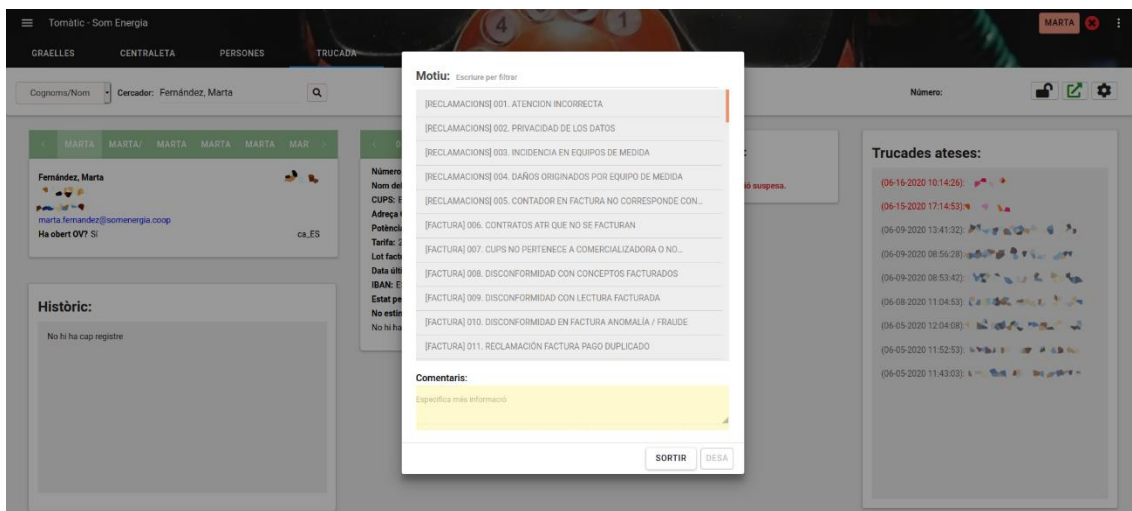
Tal com podem veure, el contracte seleccionat té la facturació suspesa i apareix al component “Altres”. També apareixeria allà si té generationkWh, si és d’energètica o si té algun cas ATR obert:



Contracte amb C obert

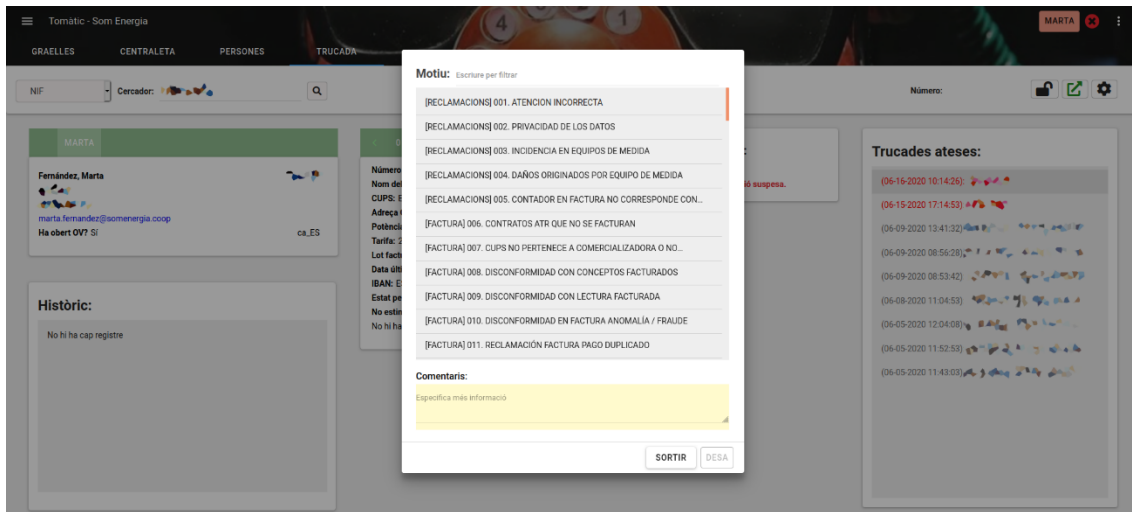
Aquests contractes han estat creats expressament a testing amb fake-data per tal de poder mostrar aquesta informació. Si es clica al correu del partner, es va a la pàgina del Helpscout amb el correu filtrat per poder visualitzar els correus enviats/rebutts per aquest email.

Si ara intentéssim respondre el qüestionari de trucades clicant al botó del contracte, visualitzaríem el següent:



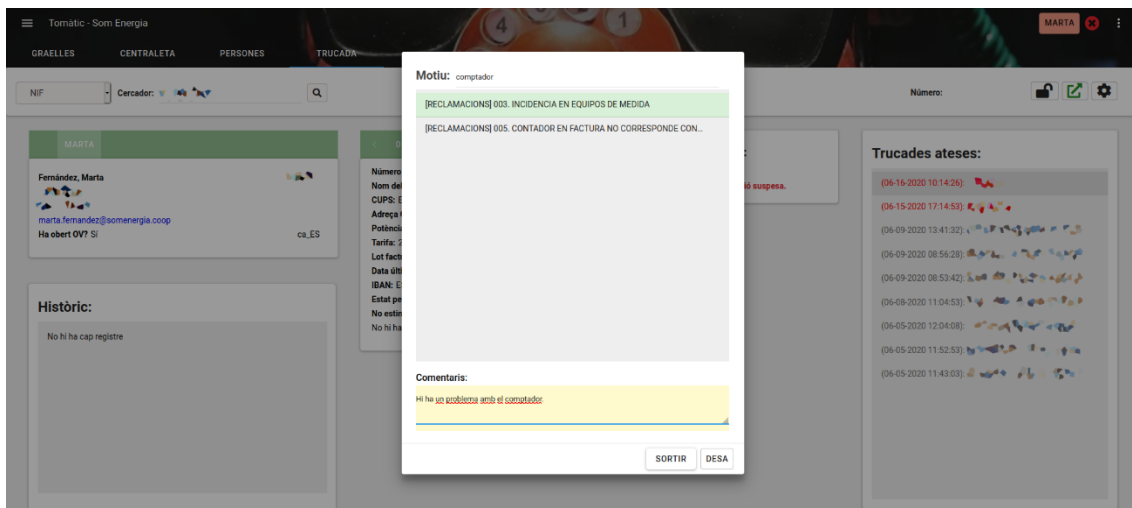
Questionari desactivat

El questionari apareixeria desactivat perquè no estem responnent respecte a cap trucada. Si, per altra banda, sortim i seleccionem una trucada de “Trucades ateses” i tornem a intentar-ho, veurem:



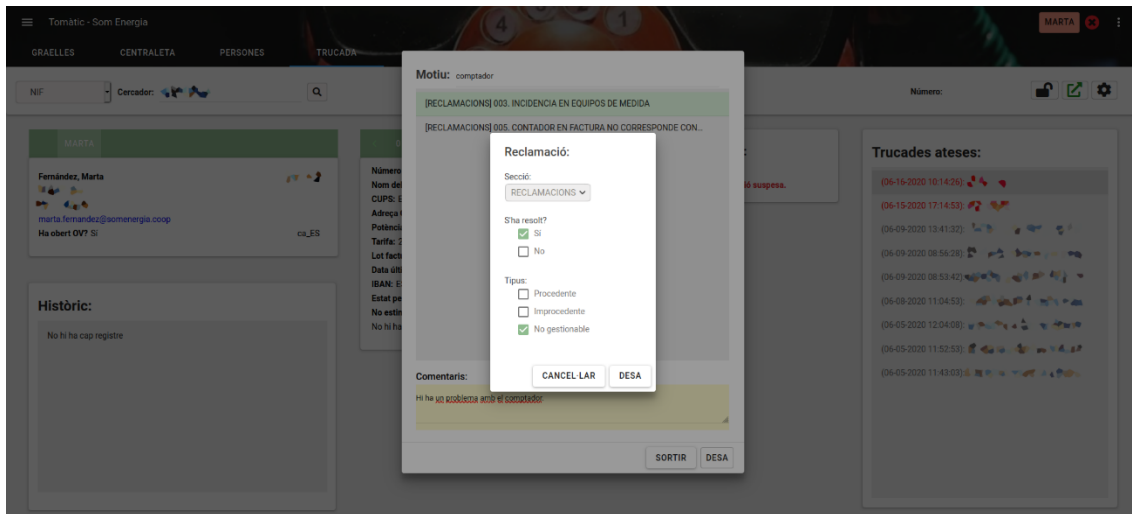
*Questionari activat*

Ara ja podríem seleccionar el motiu de la trucada, escriure un comentari i prémer desar. Per trobar el motiu de manera ràpida, es pot fer ús del filtre i de les paraules claus que aquest compren. Si, per exemple, la trucada era per parlar d'algun error del comptador, filtrariem:



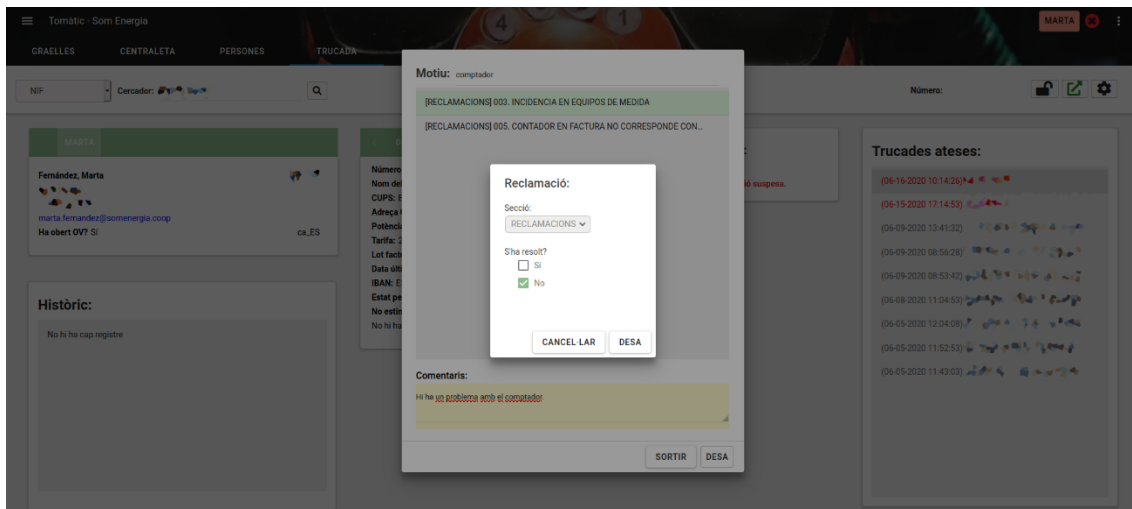
*Filtre comptador*

Podem prémer desar i, seguidament, visualitzarem:



*Dialog reclamació*

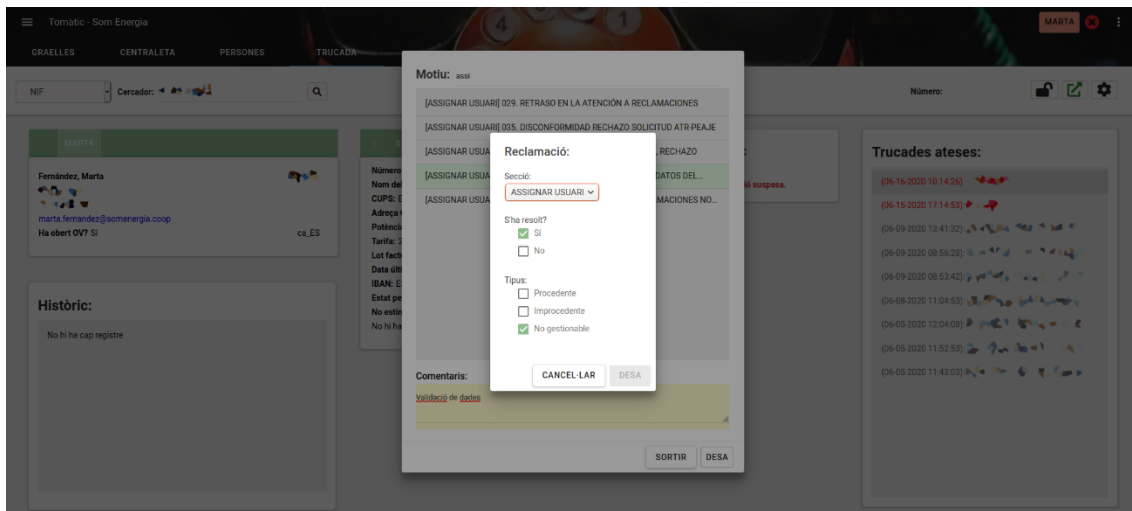
Com que és un cas de reclamació, es demana aquesta informació extra. Si s'ha resolt apareix que indiquem el tipus, altrament:



*Dialog reclamació no resolta*

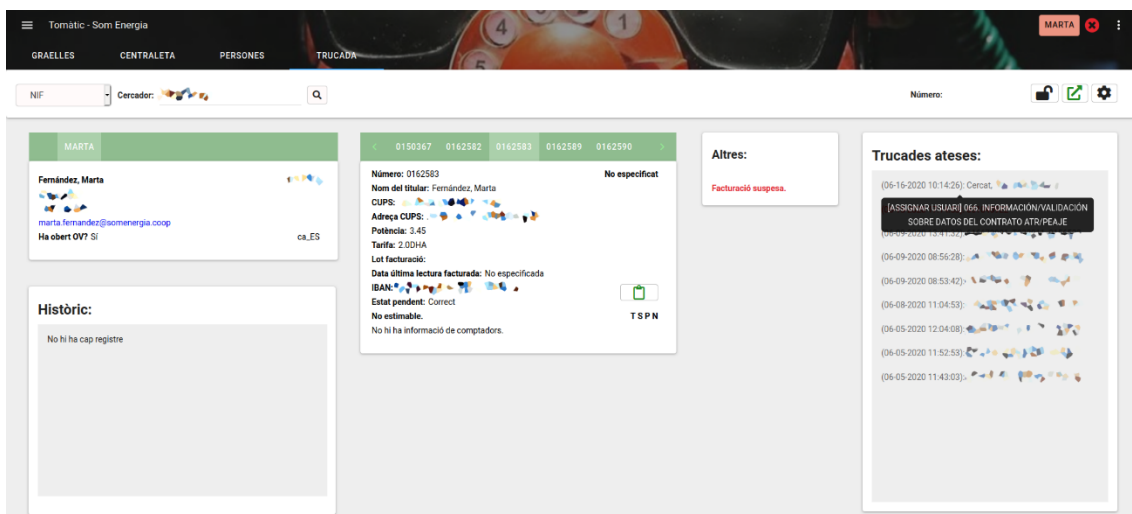


La secció ja ve seleccionada perquè el motiu té un usuari assignat. L'usuari és el que apareix entre les claus "[RECLAMACIONES]". Si, per altra banda, seleccionéssim un sense usuari assignat, la visualització seria la següent:



*Assignar usuari*

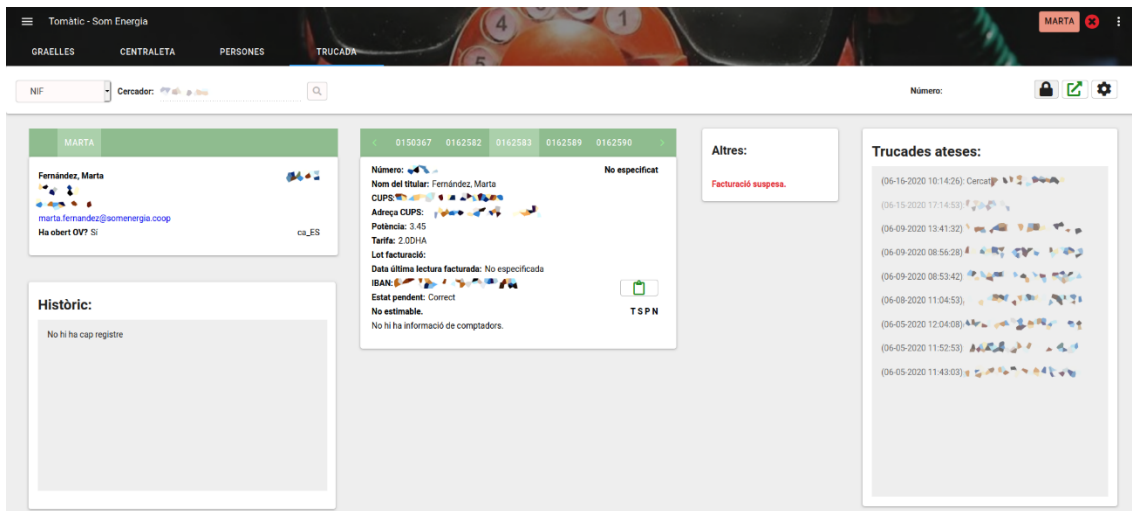
Si desem, es podran veure els canvis reflectits de la següent manera:



*Trucada desada*

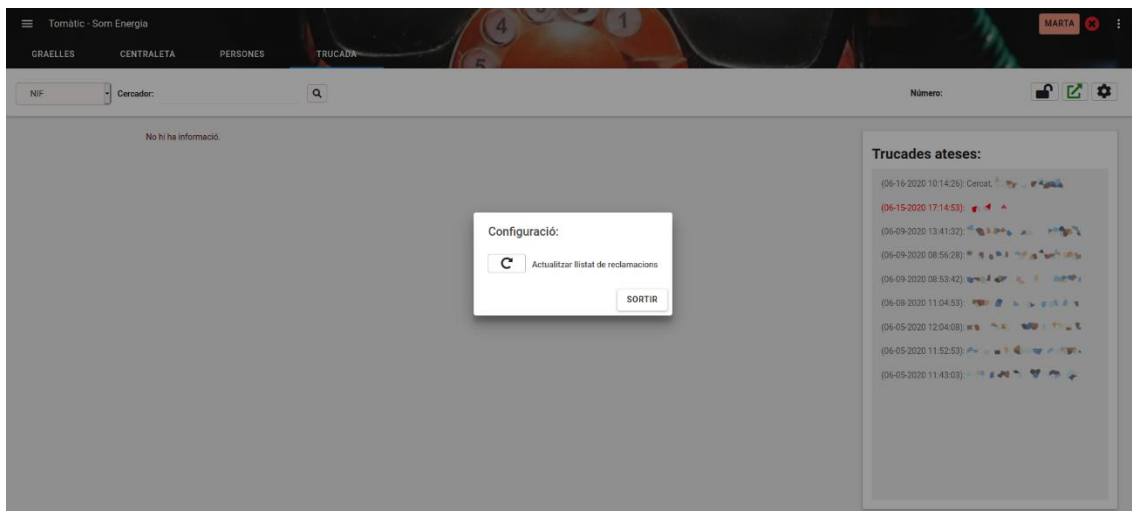
Ara que s'ha contestat el qüestionari, apareix de color gris i, si passes el ratolí per sobre, es pot observar la resposta que s'ha seleccionat.

Si es vol bloquejar el refresc automàtic, s'ha de clicar el cadenat que apareix a la part superior dreta. Si es clica queden bloquejat tant el refresc automàtic com els camps que podrien modificar la informació que s'està mostrant:



*Refresc automàtic bloquejat*

Al costat, podem observar un altre botó que serveix per obrir una altra pestanya en un nou TAB. El botó restant serveix per obrir el menú d'opcions. Actualment només conté un altre botó per actualitzar el llistat de reclamacions possibles de trucades:



*Actualitzar llistat de reclamacions*

I aquesta seria, a trets generals, la presentació de la interfície del Kalinfo.

## 11. Conclusions

A mida d'anar treballant en el projecte, he presenciat problemes de treballar mitjançant desenvolupaments àgils: és molt fàcil que hi hagi canvis en les necessitats a mitjans del desenvolupament i, tot i que s'hagin fixat uns requisits a l'inici, aniran variant.

Precisament pel fet de que el desenvolupament hagi anat variant a mesura que s'anava desenvolupant i es veien altres funcionalitats que podien resultar útils, ha acabat convertint el Kalinfo en una eina que, a part de mostrar informació dels clients, permet a l'equip generar casos de reclamacions de manera pràcticament automàtica.

Pel que fa a l'eina, m'ha agradat treballar-hi i veure-hi la utilitat en el seu dia a dia, ja que va agradar molt a l'equip quan es va fer la presentació i han vist que estalvia molta feina manual i mecànica.

Desenvolupar amb *Test Driven Development* ha estat tot un repte i ha ajudat a garantir el codi coverage i a poder desplegar els canvis que s'anaven desenvolupant a productiu amb certa garantia i seguretat.

El feedback de l'equip ha estat molt positiu: poder veure la informació dels clients que truquen sense haver-ho de buscar per diferents llocs els estalvia temps i comencen les trucades "amb bon peu". Un petit detall que és de gran ajuda, per exemple, és poder veure l'idioma que té seleccionat la persona a la seva oficina virtual. Amb aquesta dada, poden variar la salutació si ho creuen convenient.

La dificultat per desar les reclamacions que tenien com a canal d'entrada les trucades telefòniques s'ha alleujat gràcies a poder-ho tenir tot en un mateix lloc. A més, poder-ho gestionar al teu ritme (amb la facilitat de recuperar les trucades) permet que cada persona respongui el qüestionari en el moment que cregui més oportú.

Ha faltat desar la informació de trucades telefòniques. Aquesta informació es vol desar al ERP i cal fer un anàlisi del mòdul i les modificacions que es considerin necessàries ja sigui per part de l'equip ERP de Som Energia o bé de GISCE.

Un cop es disposi d'aquesta informació, es podran mostrar les estadístiques que es considerin més interessants. S'ha fet una feina d'anàlisi de diferents eines per mostrar indicadors com Grafana i Redash i de moment s'està optant per realitzar aquests tipus de gràfiques amb Superset.

El projecte ha tendit a ser un petit CRM que es preveu anar ampliant i, qui sap si fins i tot, agafar encara més entitat pròpia.

## 12. Treball futur

Els responsables IT de Som Energia tenen moltes ganes de que es segueixi desenvolupant aquest projecte. Algunes de les propostes, millores i treball futur que es proposen:

- Mostrar informació sobre els casos AC dels clients.
- Migrar tota la UI del Tomàtic a React.
- Canviar el sistema de identificació a Sessions.
- Fer els tests menys fràgils creant un cercador de casos concrets.
- Millorar rendiment del ERP i optimitzar els mòduls implicats.
- Millorar rendiment de les cerques que es realitzen per poder obtenir la informació amb menys temps.
- Desar la informació de les trucades telefòniques al ERP.
- Tenir un gran registre sobre el motiu de les trucades.
- Poder extreure dades i estadístiques reals amb aquestes dades.
- Tenir un dashboard on poder consultar tota aquesta informació.
- Poder utilitzar tècniques d'intel·ligència artificial quan es disposi de suficient informació per veure si existeix algun tipus de correlació entre diferents dades.

## 13. Webgrafia

Seguidament es citaran les referències utilitzades per desenvolupar el projecte. La major part de recerca es va realitzar a l'inici del projecte. Un cop preses les decisions la feina va ser més aviat de desenvolupament:

- Proyectos Ágiles Org. 2019. *Qué es Scrum?*. Consultat el dia 01/03/2019 a <https://proyectosagiles.org/que-es-scrum/>
- Armin Ronacher. 2012. *Welcome to Flask*. Consultat el dia 11/03/2019 a <https://flask-doc.readthedocs.io/en/latest/>
- Fundación Wikimedia Inc. 2019. *Flask*. Consultat el dia 11/03/2019 a <https://es.wikipedia.org/wiki/Flask>
- W3Schools. 2019. *JavaScript*. Consultat des del dia 15/03/2019 fins a data actual a <https://www.w3schools.com/js/default.asp>
- Aymeric Augustin and contributors. 2019. *Websockets Documentation*. Consultat el dia 15/03/2019 a <https://websockets.readthedocs.io/en/stable/>
- Fundación Wikimedia Inc. 2019. *Websocket*. Consultat el dia 15/03/2019 a <https://es.wikipedia.org/wiki/WebSocket>
- Daniel Stenberg. 2018. *Command line tool and library for transferring data with URLs*. Recuperat el dia 19/03/2019 a <https://curl.haxx.se/>
- Leo Horie. 2019. *Documentació del framework Mithril*. Consultat el dia 04/04/2019 a <https://mithril.js.org/>
- Arthur Clemens. 2019. *Polythene per a Mithril*. Consultat el dia 04/04/2019 a <https://github.com/ArthurClemens/Polythene>
- W3Schools. 2019. *HTML*. Consultat des del dia 21/04/2019 fins a data actual a <https://www.w3schools.com/html/default.asp>
- W3Schools. 2019. *CSS*. Consultat des del dia 15/05/2019 fins a data actual a <https://www.w3schools.com/css/default.asp>

## 14. Annexos

Aquest apartat inclou la informació detallada de la feina realitzada a cada ronda. Es pot veure la tasca definida a la targeta i un recompte de el total de mitjos dies que s'esperava dedicar en total i els que s'hi ha acabat dedicant.

### 14.1 Planificació detallada

#### Ronda 0 (inici a 25-03-2019):

##### (1/1) [Frontend] Introducció a la estructura del Tomàtic

###### Descripció:


Formació amb David amb temes de Front-end, JavaScript i TDD i Refactor.

##### (3/3) [Comunicació] Anàlisi de la tecnologia WebSocket

###### Descripció:

- Cercar informació sobre **Websockets**
- Analitzar els beneficis que poden aportar al projecte, on es podrien utilitzar i quines fortaleses i debilitats tenen sobre l'entorn.


###### Activitat:


 **Marta Fernández** 15 de mar. de 2019 a las 10:47

Documentació interessant i representativa sobre *Websockets*:


**Índex:**  
<https://websockets.readthedocs.io/en/stable/>


**Exemples:**  
<https://websockets.readthedocs.io/en/stable/intro.html>

 - [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)


 **Marta Fernández** 15 de mar. de 2019 a las 10:57

He creat un entorn virtual on fer proves amb la tecnologia de *Websockets*, per tal de provar el seu funcionament i realitzar un millor anàlisi sobre els possibles beneficis que podrien aportar.

 - [Editar](#) - [Eliminar](#)

 **Marta Fernández** 15 de mar. de 2019 a las 12:13

He creat un petit exemple client-servidor que es comuniquen via websocket de Python 3 (especificant @IP local i #port). Podria servir per la primera part on es comunica la *Centraleta* per enviar el número de telèfon entrant.

 - [Editar](#) - [Eliminar](#)

## (2/2) [Frontend] Anàlisi de l'entorn Frontend

### Descripció:

- Cercar informació sobre les diferents tecnologies que utilitzem a l'entorn de **Frontend**
- Analitzar els beneficis que poden aportar al projecte, on es podrien utilitzar i quines fortaleses i debilitats tenen sobre l'entorn.

### Activitat:



**Marta Fernández** ha copiado el comentario de **Marta Fernández** de la tarjeta [\(1/1\). \[R0\]. \[Frontend\]. Anàlisi de l'entorn Front-end](#) 18 de mar. de 2019 a las 12:00

He fet una pàgina local simple per simular la centraleta, on entres un número i te'l mostra per pantalla. Ara s'hauria d'ampliar perquè el comunicués via *Websocket* al *Tomàtic*.



**Marta Fernández** ha copiado el comentario de **Marta Fernández** de la tarjeta [\(1/1\). \[R0\]. \[Frontend\]. Anàlisi de l'entorn Front-end](#) 25 de mar. de 2019 a las 11:04

Anàlisi de l'estructura de DOM Virtuals, Vnodes.



**Marta Fernández** 26 de mar. de 2019 a las 11:20 (editado)

Targeta tallada, continua a [\(1/1\). \[R0\]. \[Frontend\]. Anàlisi de l'entorn Front-end](#)

- [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

Total de migdies esperats: 6

Total de migdies treballats: 6

## Ronda 1 (26-03-2019 a 12-04-2019):

### (2/2) [Frontend] Crear dins la pàgina el buscador a partir del número

#### Descripció:

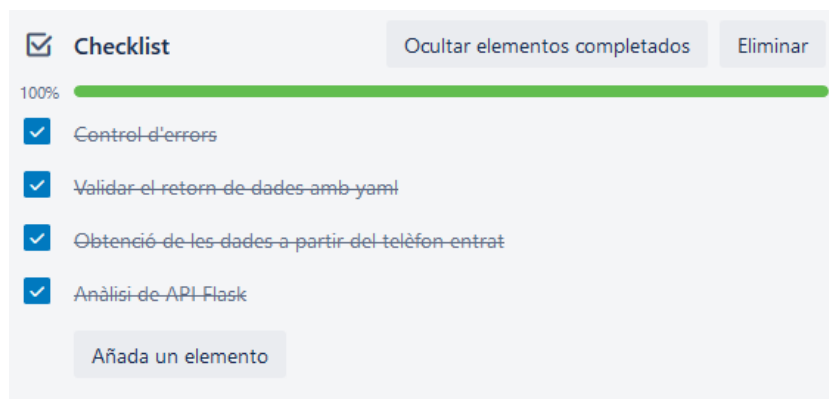
Crear dins l'apartat de **CALL INFO** del Tomàtic, un buscador on s'introduirà el número.

### (3/3) [API] Connexió entre el cercador i API de Flask

#### Descripció:

Obtenció i validació de les dades que extraurem a partir del número entrat més els tests corresponents.

#### Activitat:

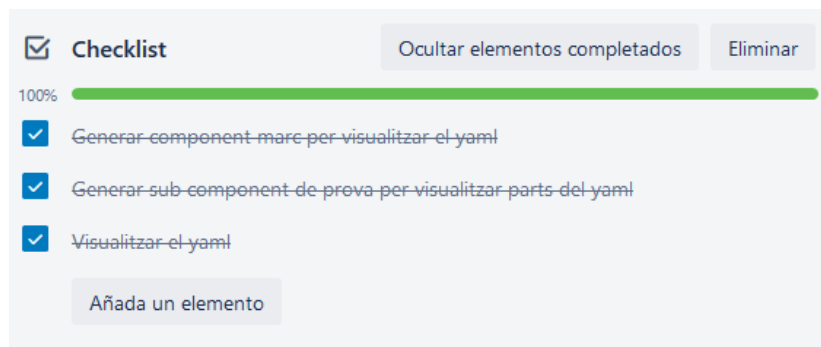


### (2/2) [Frontend] Generar components de prova per mostrar el resultat de API Flask

#### Descripció:

Llistar els diferents elements que es podran visualitzar gràcies a la extracció de dades prèviament realitzada i mostrar a la pàgina del Tomàtic el resultat de API Flask, amb la informació obtinguda a través dels components generats.

#### Activitat:






### (1/1) [Frontend] Anàlisi de l'entorn Front-end

#### Descripció:


- Cercar informació sobre les diferents tecnologies que utilitzem a l'entorn de **Frontend**
- Analitzar els beneficis que poden aportar al projecte, on es podrien utilitzar i quines fortaleses i debilitats tenen sobre l'entorn.

#### Activitat:

Finalitzar feina que quedava de la ronda anterior:


 **Marta Fernández** 26 de mar. de 2019 a las 11:20 (editado)

Targeta tallada, continua a [\(1/1\) \[R0\] \[Frontend\] Anàlisi de l'entorn Front-end](#)

 - [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

### (1/1) [ADD] [Centraleta] Anàlisi de CURL i altres estratègies per generar l'script


#### Activitat:

 **Marta Fernández** 19 de mar. de 2019 a las 12:05 (editado)

Després d'estar uns dies mirant la tecnologia dels *Websockets*, he començat a mirar alternatives:

**CURL:**  
<https://curl.haxx.se/docs/httpscripting.html>

Estic fent recerca d'altres mètodes per generar i enviar l'script.

 - [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

### (3/3) [ADD] [Frontend] Spike usuaris

#### Descripció:

Anàlisi i Spike sobre la identificació d'usuaris segons els torns de línies de cua.

Total de migdies esperats: 12

Total de migdies treballats: 12

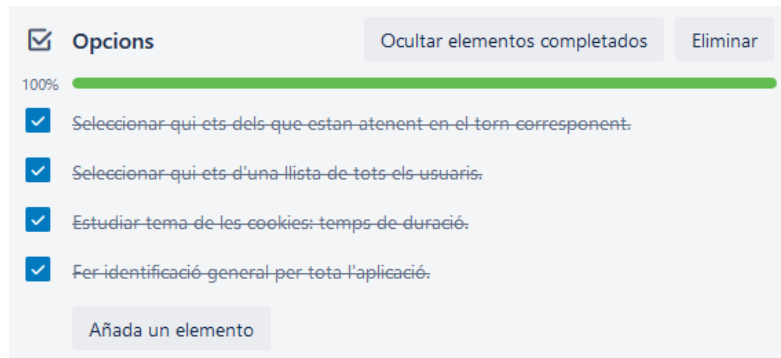
## Ronda 2 (13-04-2019 a 07-05-2019):

### (1/1) [Frontend] Anàlisi identificació d'usuaris

#### Descripció:

Anàlisi sobre el disseny basat amb l'Spike ((3/3) [ADD] [Frontend] Spike usuaris) per decidir com es farà la identificació dels usuaris.

#### Activitat:



### (1/2) [Centraleta] Anàlisi de la centraleta.

#### Descripció:

Analitzar els scripts i com funcionen els mecanismes de dins la centraleta, per saber com estan fets i com afecten al projecte.

#### Activitat:

S'ha trobat l'script que es necessita.

### (1/3) [Centraleta] Fer un Mockup de la centraleta.

#### Descripció:

Basat en el que hem extret de l'anàlisi ((1/2) [Centraleta] Anàlisi de la centraleta), per fer les proves deslligat de la centraleta real i poder simular el que passi.

#### Activitat:

Es pot executar el mode `--debug` en local i desactivar la connexió establerta amb la centraleta i línies de cua.

**(3/3) [Frontend] Implementar el sistema de identificació decidit a la targeta d'anàlisi**

**Descripció:**

Implementar el sistema de identificació decidit a la targeta d'anàlisi ((1/1) [Frontend] Anàlisi identificació d'usuaris) a la pàgina del tomàtic Call Info.

**(1/1) [Gestió] reunió amb equip de telèfon**

**Descripció:**

Llistat de dates proposades.

**Activitat:**

Primera reunió amb equip de telèfon. Reunió realitzada el dia 07-05-2019 a les 10.00 h.

**(4/2) [Comunicació] Automatitzar la comunicació de la informació**

**Descripció:**

Automatitzar els processos de refrescar informació quan es rep una trucada i dels diferents torns de telèfon de l'apartat Call Info.

Total de migdies esperats: 12

Total de migdies treballats: 11

## Ronda 3 (08-05-2019 a 28-05-2019):

### (1/1) [API] Anàlisi de com gestionar últimes trucades

#### Descripció:

Fase d'anàlisi per veure com gestionar el registre de trucades per mantenir-ho en un període de temps.

#### Activitat:



**Marta Fernández** 13 de may. de 2019 a las 8:23 (editado)

Després de realitzar l'anàlisi del disseny de la interfície gràfica a la targeta:

[\(2/2\) \[Frontend\] Disseny de interfície màrtir amb dades de Flask](#)

A partir de dos components de la interfície Front-end:

1) Història.

En aquest es podrà veure el recull de informació desat.

Tindrà el format: [[Qui va atendre] [Data]: Motiu (durada)]

2) Motiu.

Amb aquest component es podrà seleccionar i desar el motiu per el qual la persona està trucant.

Sabem el número, sabem qui està atenent al número, sabem la data, sabem el motiu (es marcaria).

D'aquesta manera es podrà desar un registre de últimes trucades amb informació prou útil i de manera relativament fàcil.

- [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

### (1/1) [ADD] (Comunicació) Arreglar bug automatització comunicació de la informació

#### Descripció:

En alguns casos, la Card amb la informació del Partner, quan s'ha d'actualitzar al haver-hi una trucada, desapareix.

#### Activitat:



**Marta Fernández** 15 de may. de 2019 a las 9:23

Després de realitzar diferents proves, he detectat el motiu:

Quan estàvem visualitzant la informació d'un partner (el 3r, per exemple), quan entrava una nova trucada, si aquesta tenia menor nombre de partners, no s'actualitzava el valor de "partner que es mostra" i intentava mostrar el que s'estava visualitzant (seguint amb l'exemple, el 3r).

S'ha arreglat actualitzant que, cada cop que entri una trucada, es mostri per defecte el primer partner.

- [Editar](#) - [Eliminar](#)

### (1/1) [ADD] [Frontend] Arreglar bug de tenir el callinfo a difrents Tabs

#### Descripció:

Al limitar que un usuari només es pugui identificar un cop, afectàvem al número de sockets que un mateix client pot tenir oberts (que interessa no limitar per si té diferents pestanyes obertes del callinfo).

#### Activitat:



**Marta Fernández** 15 de may. de 2019 a las 9:33

S'ha canviat el diccionari on es desava la informació a un diccionari de clau extensió i valor una llista de sockets.

- [Editar](#) - [Eliminar](#)



**Marta Fernández** 15 de may. de 2019 a las 9:33

S'ha adaptat el codi existent perquè funcioni amb la nova estructura de dades.

- [Editar](#) - [Eliminar](#)

### (1/1) [ADD] [Comunicació] Canviar el sistema de Websockets actual

#### Descripció:

Després d'haver d'aïllar el component de identificació del Callinfo, s'ha detectat una alta dependència entre aquests.

Segons el que s'ha decidit, el component de Login serà l'encarregat de gestionar les connexions entre websockets, avisant a Callinfo quan aquest necessiti refrescar la pàgina. D'aquesta manera, en un futur, la comunicació via websockets no estarà només limitada al Callinfo.

### (2/2) [API] Anàlisi del estat de comunicació entre Flask i ERP

#### Descripció:

Fase d'anàlisi per veure com extreure de ERP (i el seu funcionament) les dades que necessitarem posteriorment per mostrar-les.

### (4/4) [API] Implementar analisi de Flask-ERP

#### Descripció:

Fase d'implementació de l'estratègia prèviament decidida a la targeta d'anàlisi:  
((2/2) [API] Anàlisi del estat de comunicació entre Flask i ERP)

### (0/1) [Comunicació] Arreglar bug de primera identificació/login

**Descripció:**

Quan es demana la primera connexió, pot ser que encara no s'hagi obert el socket d'escolta del servidor.

En cas de que es faci "on response", al tardar a respondre s'envien varies peticions, que fa que per a una identificació es rebin 3 o 4 respostes de connexió.

Fer anàlisi de com resoldre-ho i fer-ho.

**Activitat:**

**Marta Fernández** 8 de may. de 2019 a las 8:46

Solucionat.

La petició d'obrir el WebSocket d'escolta del servidor es fa el primer cop que l'usuari s'identifica.

D'aquesta manera quan l'usuari s'ha identificat (i abans de que s'envii la petició de connexió), el WebSocket d'escolta del servidor ja haurà rebut la petició perquè s'obri.

- [Editar](#) - [Eliminar](#)

### (1/1) [Gestió] Presentar interfície màrtir i rebre feedback constructiu

**Descripció:**

Reunir-se amb els representants IT del projecte i presentar la interfície màrtir del que s'ha aconseguit fins ara i rebre el feedback i propostes de per on seguir.

**Activitat:**

Estan contents amb el que es veu. Demanen altres camps d'informació i si es pot incloure el qüestionari d'atenció telefònica.

### (1/1) [API] Implementació de gestió últimes trucades

**Descripció:**

Fase d'implementació de l'estratègia prèviament decidida a la targeta d'anàlisi:

((1/1) [API] Anàlisi de com gestionar últimes trucades)

#### (4/4) [Frontend] Implementació de interfície màrtir amb dades Flask

##### Descripció:

Després d'acordar un primer disseny:

((2/2) [Frontend] Disseny de interfície màrtir amb dades de Flask)

Implementar el que s'ha acordat per tal de mostrar la informació.

##### Activitat:



The screenshot shows a GitHub comment by Marta Fernández from May 16, 2019, at 9:44. The comment text is "Bloquejada perquè depen de:" followed by a link to the issue "(1/1) [Flask] Implementació de gestió últimes trucades". Below the comment are options to edit, add as an attachment, or delete. Below the comment is a checklist titled "Components:" with a progress bar at 100%. The checklist items are: Partners, Contractes, Història-trucades, and Log-metius, all of which are checked. There is also a button to "Añada un elemento" (Add an element).

#### (2/2) [Frontend] Disseny de interfície màrtir amb dades de Flask

##### Descripció:

En base a l'anterior reunió amb l'equip de telèfon:

((1/1) [Gestió] Reunió amb equip de telèfon)

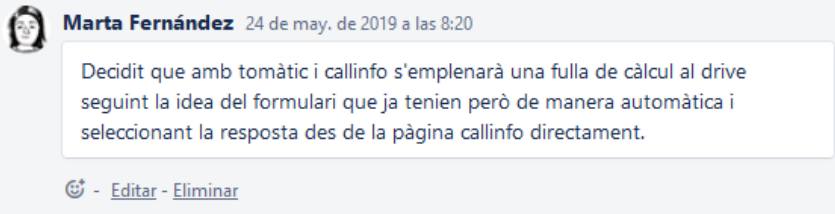
Segons les dades acordades a obtenir i mostrar, dissenyar una primera versió de interfície per a anar canviat i treballant sobre aquesta (referència).

## (2/2) [ADD] [Frontend] Anàlisi de integració entre formulari i control de trucades

### Descripció:


Després de la reunió amb els representant IT de telèfon, es farà un anàlisi de la petició de integrar els formularis d'atenció telefònica al Kalinfo.

### Activitat:



**Marta Fernández** 24 de may. de 2019 a las 8:20

Decidit que amb tomàtic i callinfo s'emplenarà una fulla de càlcul al drive seguint la idea del formulari que ja tenien però de manera automàtica i seleccionant la resposta des de la pàgina callinfo directament.

 - [Editar](#) - [Eliminar](#)

## (1/1) [ADD] [Frontend] Millores comunicació amb formulari i control de motius desats

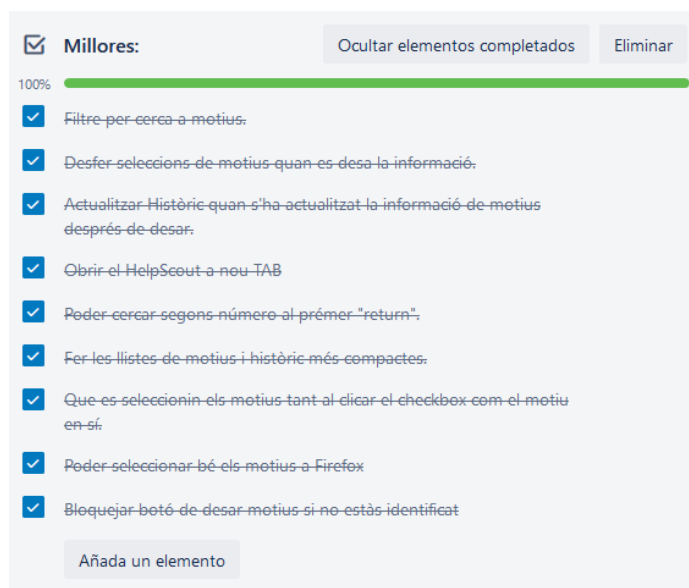
### Descripció:

Canviat el format del qüestionari perquè inclogui el correu electrònic (especificat en una pàgina del drive) i llegeixi i escrigui les dades amb un ordre diferent al que feia.

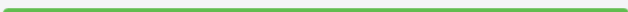
També s'ha modificat el botó de "Desar" de manera que, quan algú desi, mentre s'estiguin desant les dades, es bloquegi i canviï el text a "Desant". Un cop hagi acabat, que torni a estar activat i amb el text "Desar".

## (3/3) [ADD] [Frontend] Millores de Front-end

### Activitat:



**Millores:** Ocultar elementos completados Eliminar

100% 

- Filtre per cerca a motius.
- Desfer seleccions de motius quan es desa la informació.
- Actualitzar Històric quan s'ha actualitzat la informació de motius després de desar.
- Obrir el HelpScout a nou TAB
- Poder cercar segons número al prémer "return".
- Fer les llistes de motius i històric més compactes.
- Que es seleccionin els motius tant al clicar el checkbox com el motiu en sí.
- Poder seleccionar bé els motius a Firefox
- Bloquejar botó de desar motius si no estàs identificat



**(0/0) [Frontend] No permetre guardar qüestionari si no s'està identificat**

**Descripció:**

Evitar que algú sense identificar pugui guardar al formulari de motius de trucada

Total de migdies esperats: 15

Total de migdies treballats: 24

ADD: 6 targetes – 9 mitjos dies

## Ronda 4 (03-06-2019 a 24-06-2019):

### (0/0) [Frontend] Canvis als texts

#### Descripció:

Afegir canvis als texts on es mostren partners i contractes.

#### Activitat:

canvis partner Ocultar elementos completados Eliminar

100%

Adreça de correu → link a pestanya del helpscout

canvis contracte Ocultar elementos completados Eliminar

100%

Estat → Estat pendent

Última data facturacio → Data última lectura facturada

(0/0) [Frontend] [Bug] No convertir el número de contracte a int

### (0/0) [Frontend] reposicionar filtre de motiu

#### Descripció:

Colocar el filtre de motiu a la dreta de "moitu"

### (0/0) [Frontend] Bug: no convertir el número de contracte a int

#### Descripció:

Conservar el contracte amb els 0's davant per evitar cerques obertes al erp

Bonus: Fer-ho seleccionable amb doble click sense que agafi part de la data, parlar amb @davidgarcia93 o @davidriera6 per fer-ho amb "class" tractat a ((2/2) [Frontend] Divs amb classes i estils per els diferents camps a partner/contractes)

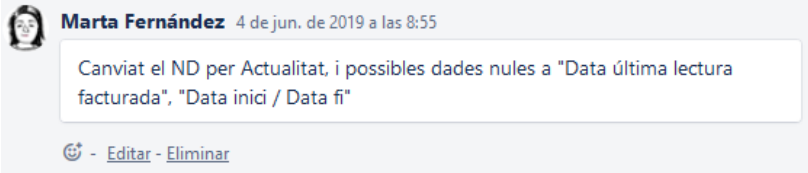
Revisar si el problema està al Flask o al Front-end

**(0/0) [Frontend] Fecha de fin: ND -> Actualidad i valors nuls/falsos**

**Descripció:**

Canviar ND per Actualitat en data final de contracte i vigilar dades nules o falses reotrnades per el Flask.

**Activitat:**



A screenshot of a GitHub commit message. At the top left is a small profile picture of a woman. To its right, the name "Marta Fernández" is displayed in bold, followed by the timestamp "4 de jun. de 2019 a las 8:55". Below this is a white text box containing the commit message: "Canviat el ND per Actualitat, i possibles dades nules a 'Data última lectura facturada', 'Data inici / Data fi'". At the bottom left of the screenshot, there are icons for a smiley face, a pencil, and a trash can, followed by the text "Editar - Eliminar".

**(0/0) [Frontend] Fer visible el número de telèfon entrant**

**Descripció:**

Fer visible el número de telèfon que esta entrant ara mateix. si es un fixe es podria extreure informació de província d'on truca...

Relacionat amb ((2/2) [Frontend] Poder bloquejar refresc de trucada entrant i visualitzar-ho)

**(2/2) [API] Ordenar botons de partners i llista de contractes**

**Descripció:**

Ordre a seguir:

- partners: Socis primer Titulars després
- contractes: Titular primers Pagadors i per últim Soci

**(1/1) [Frontend] Extreure a fitxer extern yaml dades estàtiques**

**Descripció:**

Extreure a un fitxer yaml de configuració o al dbconfig apartat tomatic totes les IP's noms de fitxers, numero de port i dades tipus "constants d'aplicació".

**(0/0) [API] Extreure a fitxer extern yaml dades estàtiques**

**Descripció:**

Extreure a un fitxer yaml de configuració o al dbconfig apartat tomatic totes les IP's noms de fitxers, numero de port i dades tipus "constants d'aplicació"

## (1/1) [API] Mes dades a Contracte

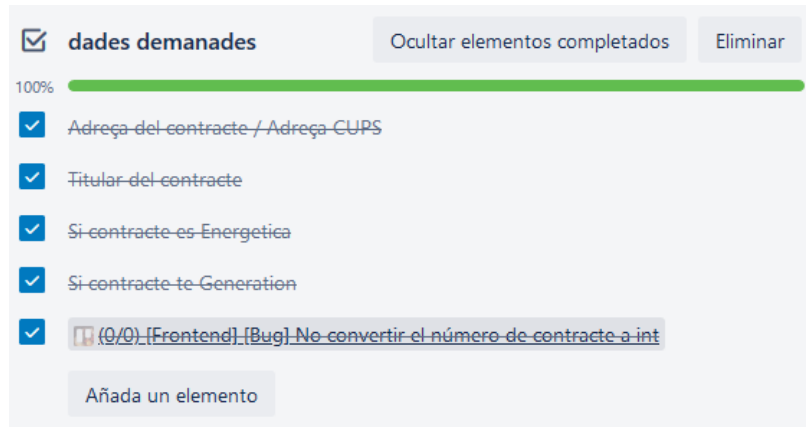
### Descripció:

Es demana afegir a contracte les següents dades.

Visualització a:

((0/0) [Frontend] Gestionar canvis en les dades d'entrada de partner i contracte)

### Activitat:



The screenshot shows a task management interface. At the top, there is a header 'dades demanades' with a checkmark icon, and two buttons: 'Ocultar elementos completados' and 'Eliminar'. Below the header is a progress bar showing 100% completion. A list of tasks follows, each with a checkmark icon and a text label:

- Adreça del contracte / Adreça CUPS
- Títular del contracte
- Si contracte es Energetica
- Si contracte te Generation
- ((0/0) [Frontend] [Bug] No convertir el número de contracte a int

At the bottom of the list is a button labeled 'Añada un elemento'.

## (1/1) [Frontend] utilitzar icones font awesome o material

### Descripció:

Utilitzar icones font awesome i material per l'aplicació:

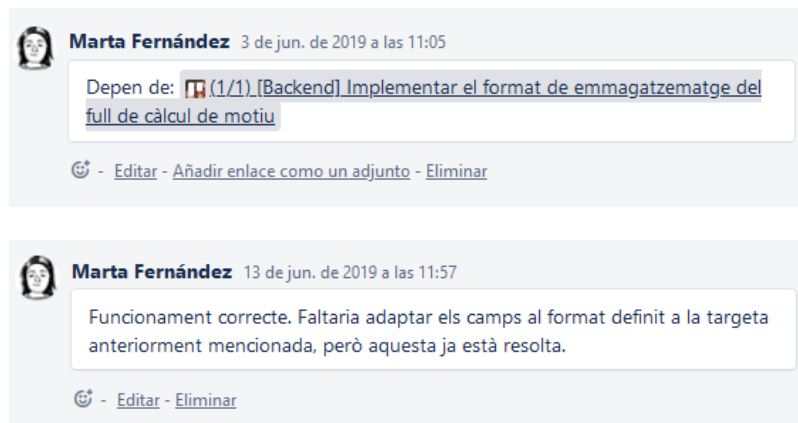
<https://fontawesome.com/icons?d=gallery>

## (1/1) [Frontend] Afegir camp observacions a formulari motiu

### Descripció:

Afegir el camp d'observacions on puguin escriure text que s'ha de guardar a full de càlcul d'emmagatzematge.

### Activitat:



The screenshot shows two comments from Marta Fernández. The first comment, dated 3 de jun. de 2019 a las 11:05, says: 'Depen de: ((1/1) [Backend] Implementar el format de emmagatzematge del full de càlcul de motiu)'. The second comment, dated 13 de jun. de 2019 a las 11:57, says: 'Funcionament correcte. Faltaria adaptar els camps al format definit a la targeta anteriorment mencionada, però aquesta ja està resolta.' Both comments have 'Editar' and 'Eliminar' buttons.

## (2/2) [Frontend] Divs amb classes i estils per les diferents camps a partners/contractes

### Descripció:

Reorganitzar la forma en que es mostra la informació de partner/contractes utilitzant divs, classes i estils

### Avantatges:

- Soluciona problemes de seleccionar amb doble click
- Html mes lleuger
- Canvi d'estils mes modular

## (1/1) [API] Mes dades a Partner

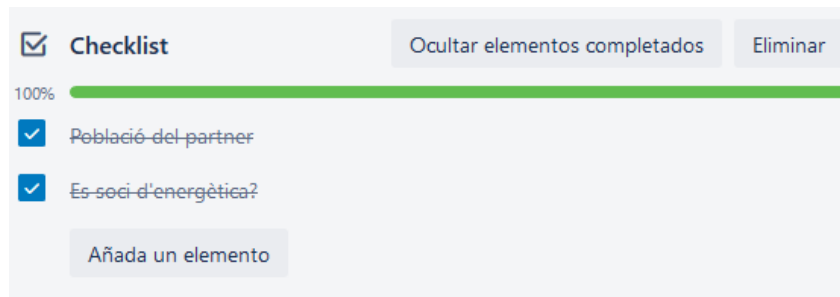
### Descripció:

Es demana afegir a partner les següents dades.

### Visualització a:

((0/0) [Frontend] Gestionar canvis en les dades d'entrada de partner i contracte)

### Activitat:



Checklist

Ocultar elementos completados Eliminar

100%

- Població del partner
- Es soci d'energètica?

Añada un elemento

## (0/0) [Frontend] Gestionar canvis en les dades d'entrada de partner i contracte

### Descripció:

Gestionar com es mostren les noves dades demanades a l'api flask a les targetes:

((1/1) [API] Més dades a Partner)

((1/1) [API] Més dades a Contracte)

### Activitat:



 **Marta Fernández** 13 de jun. de 2019 a las 8:58

Actualitzades totes les dades menys si partner és soci d'energètica (pendent d'extreure la informació).

 - [Editar](#) - [Eliminar](#)

### (0/1) [API] Bug: detector de OV no funciona correctament (sempre fals)

**Descripció:**

Es detecta que el codi que diu si s'ha accedir a la OV o no falla.

**Activitat:**

S'ha comprovat que no era un bug.

### (1/1) [API] Anàlisi de com poder cercar per altres camps

**Descripció:**

Fer un anàlisi per poder cercar per altres camps diferents de telèfon:

- Correu electrònic
- Número de soci
- Número de DNI/NIF/NIE
- Nom i cognoms

**Activitat:**

Es crearan funcions que generaran partner id's per generar un llistat conjunt a partir d'un únic camp de cerca, s'han d'utilitzar wildcards com % i.

### (0/1) [add] [Frontend] Anàlisi necessitat funció "Bloquejar pantalla" i alternatives

**Descripció:**

Si prioritzem l'atenció al client, és millor atendre la trucada i que es refresqui la informació que bloquejar-la per tal de poder seguir visualitzant l'anterior.

No contestar el telèfon, en sí, bloqueja la trucada (script s'executa quan es despenja).

**Activitat:**

Alternatives Ocultar elementos completados Eliminar

100%

- Log històric de trucades rebudes amb la informació de si s'ha emplenat el qüestionari (ràpid accés per, quan no hi hagi trucada, poder clicar i respondre el qüestionari en qüestió o visualitzar la informació de la trucada).
- Bloquejar per tab. Obrir nou tab amb callinfo desbloquejat per poder atendre la trucada entrant.

Añada un elemento



**Marta Fernández** 14 de jun. de 2019 a las 12:56

Començar amb bloqueig i new tabs. Potser ampliar al log històric de trucades a futur (a més).

- [Editar](#) - [Eliminar](#)

## (2/2) [Frontend] Poder bloquejar refresc de trucada entrant i visualitzar-ho

### Descripció:

En cas de trucada entrant poder bloquejar l'autorefresc de manera que aparegui la trucada entrant i ens permeti "agafar-la" (carregar les dades) quan nosaltres volguem. Lligat amb: ((0/0) [Frontend] Fer visible el número de telèfon entrant)

### Activitat:



**Marta Fernández** 11 de jun. de 2019 a las 8:54

Bloquejada per la dependència de la targeta afegida:

[\(0/1\) \[add\] \[Frontend\] Anàlisi de la necessitat de la funció "Bloquejar pantalla" i possibles alternatives](#)

- [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)



**Marta Fernández** 14 de jun. de 2019 a las 12:58

Canviat objectiu a: bloqueig + obrir nova finestra sense bloqueig per refrescar trucada entrant.

- [Editar](#) - [Eliminar](#)

## (1/1) [ADD] [Frontend] Adaptar cercador per buscar per altres camps

### Descripció:

Tal com estava fet el cercador, la crida a API Flask es feia a partir del número.

### Activitat:



**Canvis:**

Ocultar elementos completados

Eliminar

100%



La crida a API Flask i com es passa la informació del que hem de cercar.



Com/ón realitzar la crida a API Flask per obtenir l'històric de trucades.



Separar el que estem cercant del número (variables diferents).



Visualització del telèfon entrant.



Que cercar no afecti al refresh automàtic després de desbloquejar.

[Añadir un elemento](#)



**Marta Fernández** 18 de jun. de 2019 a las 12:42

S'ha afegit un add a [\(1/1\) \[Frontend\] Anàlisi sobre l'obtenció de l'històric de motius quan es cerca per altres camps.](#)

Per el tema de l'històric de trucades.

- [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

**(1/1) [ADD] [Frontend] Anàlisi log de trucades ateses**

**Descripció:**

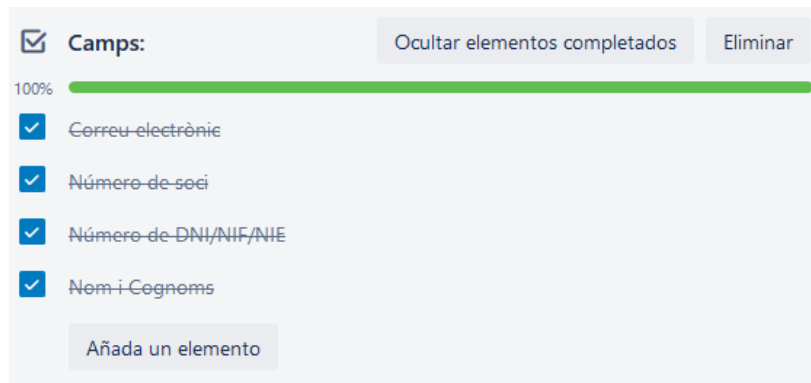
Anàlisi sobre el component que mostri el log de trucades ateses.

**(3/3) [ADD] [API] Implementar de com cercar per altres camps**

**Descripció:**

Implementar per poder cercar per altres camps diferents de telèfon.

**Activitat:**



**(0/0) [ADD] [API] Separar dades estàtiques de Fulles de Càlcul**

**Descripció:**

Separar dades estàtiques de Fulles de Càlcul a consultar a fitxer de configuració extern i constants.

**(1/1) [ADD] [Frontend] Implementar cercador per altres camps**

**Descripció:**

Implementar cercador per altres camps (poder escollir què cerquem o cerca general).



## (1/1) [add] [Frontend] Implementar log de trucades ateses.

### Descripció:

Atès a les decisions preses a:

((1/1) [Frontend] Anàlisi sobre el component que mostri el log de trucades ateses)

### Activitat:



**Marta Fernández** 20 de jun. de 2019 a las 12:15

Es pot visualitzar el log de "trucades ateses".

M'han sorgit una sèrie de dubtes:

- Vincular-ho amb poder emplenar el qüestionari si no s'ha fet?
- Separar informació de qüestionari amb trucades rebudes?
- Desar totes les trucades rebudes? O que sigui temporal per torn/dia?

Actualment, si no s'ha emplenat el qüestionari (que no hauria de passar: es desen les trucades quan emplenem el qüestionari), surten en vermell. Està preparat perquè sigui relativament fàcil canviar-ho en funció del que es decideixi de les preguntes de dalt.

- [Editar](#) - [Eliminar](#)



**Marta Fernández** 26 de jun. de 2019 a las 13:12

S'hauria de dedicar una targeta d'anàlisi per tal de veure com encarar aquest component en cas de que es vulgui utilitzar.

- [Editar](#) - [Eliminar](#)

## (0/0) [ADD] [API] Anàlisi Log de Trucades ateses

### Descripció:

Anàlisi i decisió del format sobre el Log de Trucades ateses

### Activitat:



**Marta Fernández** 26 de jun. de 2019 a las 13:16

Desar les últimes 20 trucades (per exemple) en un fitxer yaml. S'anirà actualitzant la informació.

- [Editar](#) - [Eliminar](#)

### (0/0) [Frontend] Compatibilitat amb Firefox

**Descripció:**

Al utilitzar firefox motius no es seleccionable cal anar amb compte amb la visualització a diferents navegadors.

De moment hem d'assegurar compatibilitat amb:

- Chrome-Chromium
- Firefox

Considerem navegadors poc usat o deprecats als següents:

- Internet explorer
- Opera
- Edge
- Edge/webkit

### (1/2) [Frontend] tab control per selecció de partner

**Descripció:**

Fer servir un tab control similar al de l'aplicació per mostrar els partners trobats amb el mateix número de telefon.

- Marcar quin tab esta seleccionat
- Situar el selector a dalt
- Aplicar colors per partners?

### (1/2) [Frontend] Utilitzar identificació en la gestió de canvis a Graelles

**Descripció:**

Utilitzar el sistema de identificació per gestionar els canvis de la pestanya de graelles per evitar que els canvis siguin sempre a nom de nobody

Si es possible evitar que es faci el canvi si no s'està identificat.

Total de migdies esperats: 17

Total de migdies treballats: 22

ADD: 8 targetes – 7 mitjos dies



**(0/0) [ADD] [API] BUG: Error quan es rep trucada i no s'està identificat.**

**Descripció:**

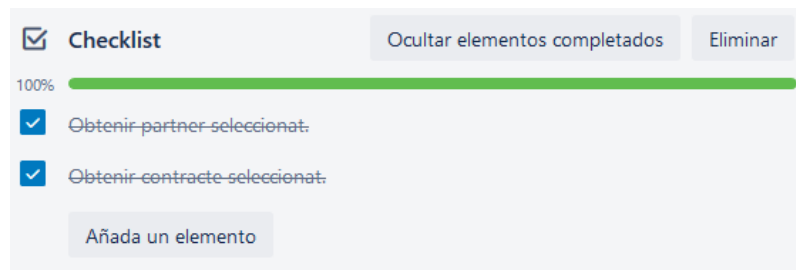
S'ha detectat un error quan es rep una trucada i no s'està identificat. Analitzar el motiu i arreglar-lo.

**(2/3) [Frontend] Seleccionar un contracte i partner per qüestionari**

**Descripció:**

Poder seleccionar un contracte i partner per relacionar-lo amb els qüestionaris.

**Activitat:**



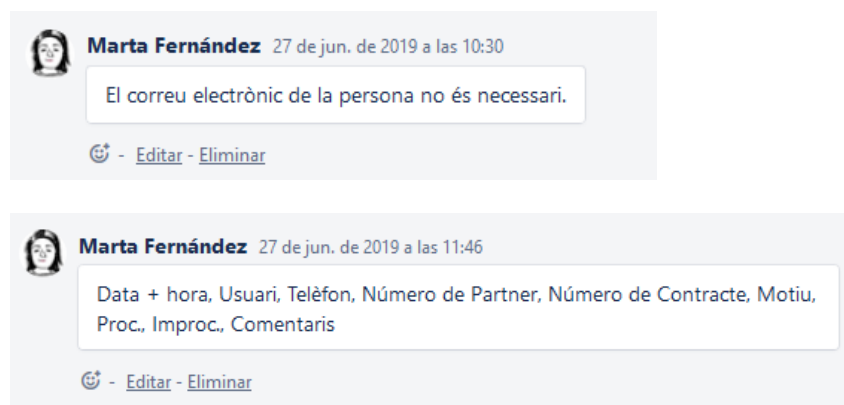
The screenshot shows a checklist interface. At the top, there is a header with a checked box icon, the word "Checklist", and two buttons: "Ocultar elementos completados" and "Eliminar". Below the header is a progress bar showing 100% completion. There are two items in the checklist, both with checked boxes: "Obtenir partner seleccionat." and "Obtenir contracte seleccionat.". At the bottom, there is a button labeled "Añada un elemento".

**(1/1) [Backend] Implementar el format de emmagatzematge del full de càlcul de motiu**

**Descripció:**

Definir un format per emmagatzemament les entrades de Motiu que sigui fàcil per fer cerques i adaptat a les necessitats del equip de telèfon per a generar estadístiques ràpidament.

**Activitat:**



The screenshot shows two chat messages from Marta Fernández, dated 27 de jun. de 2019. The first message, at 10:30, says "El correu electrònic de la persona no és necessari." and has "Editar" and "Eliminar" options. The second message, at 11:46, lists fields: "Data + hora, Usuari, Telèfon, Número de Partner, Número de Contracte, Motiu, Proc., Improc., Comentarís" and also has "Editar" and "Eliminar" options.

**(2/2) [Frontend] Procedente i Improcedente a motius**

**Descripció:**

Dos checks amb la lògica de que no es puguin marcar tots dos, però que es puguin deixar desmarcats tots dos. Que es converteixi en una columna de excel.

**(1/1) [ADD] [Gestió] Preparació i realització de presentació del CallInfo a l'equip tècnic**

**Descripció:**

Preparar la presentació a realitzar a la següent reunió mensual de tot l'ET. Realitzar també una demo de com s'utilitza l'eina.

**(1/1) [Frontend] Anàlisi obtenció de històric de motius quan es cerca per altres camps**

**Descripció:**

Quin número s'hauria d'obtenir si apareixen diferents partners?  
Com tractaríem la informació si es vol emplenar el qüestionari?  
Quina informació visualitzaríem a l'històric?

**(1/1) [ADD] [Frontend] BUG: Històric de trucades del número entrant**

**Descripció:**

Al desar els motius, s'actualitza el registre de trucades del número quan es rep la confirmació de que tot ha anat bé (són tantes crides com motius, es va refrescant per cada motiu afegit).

Si estem desant (és lent) i algú truca, es veu l'històric de trucades del nou número entrant, però quan es rebí la confirmació de l'anterior crida, es refrescarà l'històric de trucades del número actual (no mostra informació errònia, però no s'hauria de refrescar).

També mirar la possibilitat de fer-ho només amb una crida (i només esperar una confirmació per a tots els motius).

## (2/2) [ADD] [Frontend] Millores i neteja del codi.

### Descripció:

De les funcions de crides a la API Flask i de Front-end de mostrar components.

### Activitat:

Checklist Ocultar elementos completados Eliminar

100%

- Emplenar el qüestionari correctament si no hi ha informació del número entrant.
- Afegir possibilitat d'emplenar el qüestionari si es realitza una cerca.
- Que aparegui informació a l'històric només quan hi ha trucades.
- Que l'històric només es refresqui quan cerca informació del número entrant (no al desar).
- Al clicar alguna de les trucades ateses, desar si no s'ha respost el qüestionari.

[Añada un elemento](#)

## (2/2) [Frontend] Possibilitat accedir trucada a partir del log

### Descripció:

Possibilitat de accedir a la trucada a partir del log per respondre el qüestionari.

Dubtes:

Només les no contestades? O totes per si es vol afegir alguna cosa?

### Activitat:

**Marta Fernández** 2 de jul. de 2019 a las 13:57

El click sobre la trucada la qual volem emplenar el qüestionari simularà una trucada amb el número i la data especificats. La resta de dades s'emplenaran de la mateixa manera que es faria amb una trucada natural.

Per tal de desar la informació, es cercarà on està la corresponent trucada i es farà un update amb la nova informació.

- [Editar](#) - [Eliminar](#)

**Marta Fernández** 5 de jul. de 2019 a las 8:53

Revisar l'update: en alguns casos es fa correctament, en altres fa nou post.

- [Editar](#) - [Eliminar](#)

**(0/1) [Frontend] Anàlisi identificació en la gestió de canvis a graelles**

**Descripció:**

Anàlisi identificació en la gestió de canvis a graelles (modificar sistema actual).

**(1/1) [add] [Frontend] Dubtes Frontend**

**Descripció:**

Dubtes de disseny sobre coses que he fet d'una manera però m'agradaria consultar i veure alternatives.

**Activitat:**

✓ **Dubtes** Ocultar elementos completados Eliminar

100% Progress bar

- ✓ Center-dialog.
- ✓ Import icones FontAwesome.
- ✓ Center cercador a Firefox.
- ✓ Utilitzar Moment.js?

Añada un elemento



**Marta Fernández** 13 de jun. de 2019 a las 11:34

Necessari utilitzar <https://momentjs.com/> ?  
(Només ús de la funció date2str)

- [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

**(0/0) [Frontend] Anàlisi sobre ampliar el qüestionari**

**Descripció:**

Anàlisi sobre ampliar el qüestionari: possibilitat de tenir-ne més de un.

**(2/2) [Gestió] Fer deploy de la primera versió**

**Descripció:**

Fer deploy de la release de la primera versió del tomàtic visible per tot-hom  
Necessitarem en @davidgarcia93

**(0/2) [Frontend] Spike sobre l'històric de motius.**

**Descripció:**

Relacionat amb ((1/1) [Frontend] Anàlisi sobre l'obtenció de motius quan es cerca per altres camps)

Fer un prova sobre les diferents maneres de obtenir el llistat de motius a partir dels partners, telèfon...

**Activitat:**



**Marta Fernández** 9 de jul. de 2019 a las 11:23

S'ha pres la decisió de que només es mostrarà l'històric de motius quan hi hagi una trucada. Per a les cerques no específiques no hi haurà informació.

- [Editar](#) - [Eliminar](#)

Total de migdies esperats: 17

Total de migdies treballats: 17

ADD: 6 targetes – 5 mitjos dies



## Ronda 6 (17-02-2020 a 28-02-2020):


### (3/2) [Backend] Anàlisi generació de casos AC

#### Descripció:

Anàlisi sobre la creació dels casos AC

#### Activitat:

**TODO** Ocultar elementos completados Eliminar

100% 

- Com extreure els tipus de casos AC del ERP
- Com desar els casos AC al ERP
- Com assignar paraules clau als casos



**Marta Fernández** 25 de feb. a las 10:21 (editado)

#### Com extreure els tipus de casos AC del ERP:

Els casos de reclamacions estan en el model:

```
giscedata.subtipus.reclamacio
```

El `name` conté el codi de la reclamació i `desc` la descripció d'aquesta. Tenen un paràmetre anomenat `default_section` que és clau forana del model `crm.case.section`. Indica a quina secció pertanyen (pe. RECLAMACIONS, FACTURA...) amb el camp `name`. Si no té usuari assignat, `default_section` és `False`.

Amb aquesta informació, ja podriem recuperar els diferents tipus de reclamacions del ERP i mantenir el format:

CODI. [SECCIÓ] TIPUS DE RECLAMACIÓ

#### TODO:

- Preguntar si es vol mantenir la llista fixa (amb les reclamacions més usuals) i la llista extensa (que es mostra al filtrar). En cas de que interessi, pensar com fer-ho de manera adequada.

 - [Editar](#) - [Eliminar](#)



**Marta Fernández** 25 de feb. a las 14:54 (editado)

### Com desar els casos AC al ERP:

Primerament, s'ha de crear un `crm.case`. Els camps requerits per tal de crear-lo són el `section_id` i el `name`. El primer l'obtenim del Kalinfo mateix (explicat al comentari anterior). El segon, fa referència a la descripció. S'hauria de decidir quin format ha de seguir.

Un cop creat, per tal de crear un `giscedata.atc` es necessita un diccionari amb:

```
{
  'subtipus_desc': descripció de la secció ( section.name ),
  'provincia': id_provincia ,
  'cups_id': id_cups ,
  'total_cups': 1,
  'crm_id': crm_id
}
```

### TODO:

- Decidir en quin format ha de tenir la descripció del cas crm.
- A part dels camps estrictament necessaris per crear el cas atc, decidir quina altra informació es vol especificar i com.

- [Editar](#) - [Eliminar](#)



**Marta Fernández** 25 de feb. a las 15:38 (editado)

### Com assignar paraules clau als casos

Dues possibilitats (a preguntar quina es prefereix):

- Desar-ho al drive tal i com es fa ara.
- Crear un apartat al Kalinfo on es puguin veure, afegir, modificar i eliminar paraules clau.

- [Editar](#) - [Eliminar](#)

## (1/1) [Backend] Extracció dels tipus de casos AC del ERP

### Descripció:

Després de l'anàlisi, extracció dels tipus de casos AC del ERP

## (1/2) [Backend] Desar els casos AC al ERP

### Descripció:

Després de l'anàlisi corresponent, desar els casos AC al ERP

### Activitat:



**Marta Fernández** 2 de mar. a las 8:19

Feta una funció per desar els casos AC al ERP però faltaria veure com testejar correctament (per tal de fer rollback i no crear cas cada cop que es facin tests).

😊 - [Editar](#) - [Eliminar](#)

Total de migdies esperats: 5

Total de migdies treballats: 5

## Ronda 7 (02-03-2020 a 27-03-2020):

### (1/1) [Gestió] Anàlisi amb equip ERP sobre l'estat dels mòduls

#### Descripció:

Anàlisi amb equip ERP sobre l'estat dels mòduls 'Casos AC' i 'Trucades telefòniques'

#### Activitat:



**Marta Fernández** 2 de mar. a las 12:45 (editado)

Ens hem reunit amb Fran i ens ha comentat que:

- El mòdul de casos AC del ERP no està optimitzat. A causa de totes les automatitzacions quan es creen les reclamacions, es poden generar problemes de bloquejos i lentitud.
- Es podria fer un script que s'executés cada migdia a les 14.00 h (després dels torns de telèfon) que crees tots els casos AC (tant els de reclamacions com les consultes telefòniques).
- Es podria fer un anàlisi de si surt a compte separar els diferents casos en *Reclamacions / Infos* i crear-los en diferents moments del dia. Com que els casos AC de reclamacions tenen afectacions amb altres processos (com facturació), aquests s'haurien de crear al migdia. Més tard, es podrien fer la resta (casos AC de info).
- Faltaria revisar si **Trucades telefòniques** i **atc** són dos punts d'entrada al mateix model per tal de saber on s'ha de desar exactament cada trucada segons la seva tipificació.
- També hem comentat que, el problema sorgeix quan parlem de crear els casos però que, en cas de consultar la informació, no s'hauria de veure massa afectat el rendiment del ERP en general.

😊 - [Editar](#) - [Eliminar](#)



**Marta Fernández** 2 de mar. a las 12:54

S'ha creat la targeta:

(0/2). [Backend] Realitzar script que crei els casos AC amb la informació extreta del Kalinfo

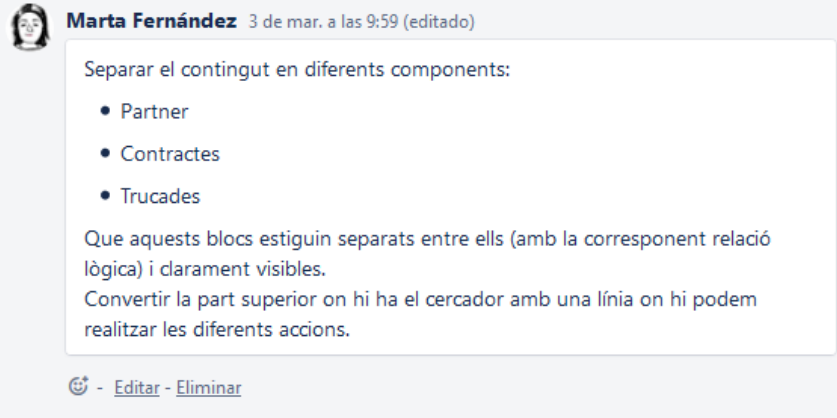
Per fer l'script necessari que crearà els casos en funció de la informació extreta del Kalinfo.

😊 - [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

### (3/1) [Frontend] Anàlisi refactor interfície

**Descripció:**

Donar més visibilitat a la informació que es mostra referent al número que truca i que el qüestionari estigui més en segon pla.

**Activitat:**

**Marta Fernández** 3 de mar. a las 9:59 (editado)

Separar el contingut en diferents components:

- Partner
- Contractes
- Trucades

Que aquests blocs estiguin separats entre ells (amb la corresponent relació lògica) i clarament visibles.

Convertir la part superior on hi ha el cercador amb una línia on hi podem realitzar les diferents accions.

😊 - [Editar](#) - [Eliminar](#)

### (1/1) [Backend] TDD, ús de transaccions i rollback

**Descripció:**

Implementar manera correcta de fer tets amb oop fent rollback per a transaccions que realitzin commit i canvis al ERP.

**Activitat:**

**Marta Fernández** 2 de mar. a las 10:01 (editado)

Afegida la dependència de `ERPPeek-WST` i creada la funció de `discarded_transaction`

😊 - [Editar](#) - [Eliminar](#)

### (0/0) [Gestió] Concretar camps necessaris per a la creació del cas AC

**Descripció:**

Concretar camps necessaris per a la creació del cas AC (i Trucades Telefòniques)

**Activitat:**

Informació de l'anàlisi documentada a un document privat de l'ET.

### (1/1) [API] Integració Frontend i Backend per la obtenció de les reclamacions del ERP

**Descripció:**

Integració Frontend i Backend per la obtenció de les reclamacions del ERP

### (1/1) [Frontend] Botó per refrescar el llistat de reclamacions

**Descripció:**

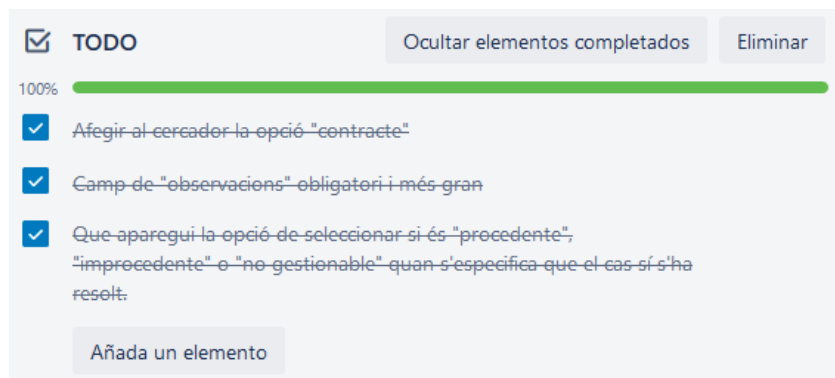
La obtenció del llistat de reclamacions es fa via la crida API específica que actualitza el que emmagatzema el servidor de manera que quan hi hagin canvis a nivell d'ERP hi haurà un botó que clicant-lo actualitzarà el llistat.

### (1/1) [Frontend] Canvis a la interfície Kalinfo

**Descripció:**

Diferents millores i canvis que es volen realitzar a la interfície

**Activitat:**



## (2/1) [Frontend] Implementar refactor segons anàlisi

### Descripció:

Targeta que depèn de ((3/1) [Frontend] Anàlisi refactor interfície).

### Activitat:

#TODO Ocultar elementos completados Eliminar

25%

- Component PARTNERS
- Component TRUCADES
- Component CONTRACTES
- Component 'Línia d'accions'

Añada un elemento



**Marta Fernández** ha copiado el comentario de **Marta Fernández** de la tarjeta [\(4/3\) \[Frontend\] Implementar refactor segons anàlisi](#) 26 de mar. a las 8:19

Bloquejat fins a ronda on s'acabarà de plantejar la interfície màrtir i decidir què es vol exactament.

Total de migdies esperats: 7

Total de migdies treballats: 10

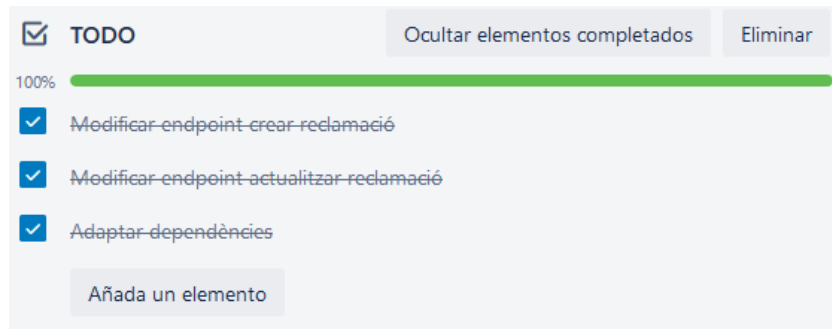
## Ronda 8 (30-03-2020 a 20-04-2020):

### (4/2) [API] Canviar on es desen les reclamacions

#### Descripció:

Del drive a un fitxer (log) local amb el que s'executarà l'script que crearà els casos AC al ERP.

#### Activitat:



### (2/2) [Backend] Anàlisi crear cas AC amb els camps especificats

#### Descripció:

En funció dels camps especificats a ((0/0) [Gestió] Concretar camps necessaris per a la creació del cas AC)

Veure els camps necessaris i com obtenir-los.

### (4/0) [ADD] [Backend] Funció crear cas AC amb els camps especificats

#### Descripció:

A partir de la informació extreta de l'anàlisi, actualitzar la funció 'create\_atc\_case' per tal de que el creï amb tots els camps + tests.



### (2/0) [ADD] [Backend] Anàlisi bug erppeek\_wst

#### Descripció:

Al utilitzar les transaccions de erppeek\_wst per a fer els tests corresponents, hem vist que en algunes ocasions dona algun error.

Aquesta targeta inclou el temps dedicat a intentar veure d'on ve i quines repercussions pot tenir.

#### Activitat:



**Marta Fernández** 15 de abr. a las 17:01

Hem vist que no té afectacions i que es poden seguir utilitzant les transaccions perquè es fa el rollback correctament.

- [Editar](#) - [Eliminar](#)

### (1/1) [Backend] Anàlisi nova manera de fer els tests

#### Descripció:

Targeta que contempla el temps de veure i pensar diferents maneres de fer els tests del Kalinfo per desfragilitzar-los i la presa de decisió.

#### Activitat:



**Marta Fernández** 20 de abr. a las 8:36

La millor opció, és **fer ús d'un cercador de casos** que permeti obtenir la informació necessària que ara està escrita de manera fixe i no revisa possibles canvis (a part de amb la fallada del test).

- [Editar](#) - [Eliminar](#)


## (1/0) [ADD] Fix de bugs Frontend

### Descripció:

Arreglar bugs trobats al Frontend.

### Activitat:

**Cases** Ocultar elementos completados Eliminar

100% 

- Trucada + Respondre
- Trucada + Cerca + Respondre
- Trucada + Trucada
- Trucada + Deseleccionar + Cerca + "Respondre"
- Trucada + Seleccionar-altre
- Seleccionar + Respondre
- Seleccionar + Cercar + Respondre
- Seleccionar + Trucada

Añada un elemento



**Marta Fernández** 15 de abr. a las 17:05 (editado)

**Fixed.** Tots els casos funcionen. S'hauria de:

- Actualitzar correctament l'estat de "DESANT".
- Bloquejar el botó de desar si es preten omplir el qüestionari sense tenir cap trucada seleccionada.

 - [Editar](#) - [Eliminar](#)


## (2/2) [ADD] [Backend] Contemplar tots els casos de possibles trucades

### Descripció:

S'ha detectat que poden haver diferents casos de trucades i que s'hauria de poder completar el qüestionari per totes elles (actualment, s'han afegit unes restriccions que feien que no es compleixi).

### Activitat:

**TODOS** Ocultar elementos completados Eliminar

100% 

Poder seleccionar i deseleccionar trucades del registre.

Poder buscar amb una trucada seleccionada i respondre sobre aquesta.



**Marta Fernández** 2 de abr. a las 10:23 (editado)


Tant per els números registrats com per els que no ho estan, els casos es redueixen a:

#### Cas 1:

Si un número registrat (o no) truca per demanar info, es podran respondre trucades de info sense especificar contracte.

#### Cas 2:


Si un número registrat (o no) truca per reclamar, s'haurà d'especificar el contracte sobre el qual s'està reclamant: o bé el que surt o s'haurà d'utilitzar el cercador per trobar el contracte corresponent i seleccionar-lo.

 - [Editar](#) - [Eliminar](#)



**Marta Fernández** 2 de abr. a las 10:25

Actualment, només s'està treballant amb els casos AC (reclamacions) per tant, s'adaptarà el que hi ha ara per tal de contemplar el **cas 2**. S'haurà de fer l'ampliació de que, dins una trucada, es pugui cercar info i respondre sobre aquella trucada amb la informació cercada.

 - [Editar](#) - [Eliminar](#)

## (2/2) [Backend] Script que crei els casos AC amb la informació extreta del Kalinfo

### Descripció:

Un cop adaptem la interfície del Kalinfo, podrem realitzar l'script que utilitzi aquesta informació per crear els casos AC del dia.

Targeta sorgeix de:


((1/1) [Gestió] Anàlisi amb equip ERP sobre l'estat dels mòduls 'Casos AC' i ' Trucades Telefòniques')

### Activitat:



**Marta Fernández** 20 de abr. a las 8:46

S'ha fet un script que accedeix als casos desats al directori especificat i executa la funció que crea el cas.

 - [Editar](#) - [Eliminar](#)

Total de migdies esperats: 7

Total de migdies treballats: 18

ADD: 4 targetes – 8 mitjos dies

## Ronda 9 (21-04-2020 a 01-05-2020):

### (4/3) [Frontend] Implementar refactor segons anàlisi

#### Descripció:

Targeta que depèn de ((3/1) [Frontend] Anàlisi refactor interfície)

#### Activitat:

#TODO Ocultar elementos completados Eliminar

80%

- Component PARTNERS
- Component TRUCADES
- Component CONTRACTES
- Component 'Línia d'accions'
- Qüestionari

Añada un elemento



**Marta Fernández** 29 de abr. a las 10:37

No s'havia contemplat el canvi del qüestionari en aquesta targeta però també s'hauria de fer. Per tant, es necessitarà més temps del previst per tal d'acabar-la.

- [Editar](#) - [Eliminar](#)

### (6/4) [Frontend] Interfície Kalinfo per mostrar més informació

#### Descripció:

Mostrar més informació.

#### Activitat:

Checklist Ocultar elementos completados Eliminar

67%

- Factures
- Comptadors
- Casos AC

Añada un elemento


## (4/2) [API] Obtenir altra informació

### Descripció:

S'ha demanat que aparegui més informació.

### Activitat:

Checklist Ocultar elementos completados Eliminar

50% 

- Factures
- Comptadors
- Casos AC
- IBAN
- Lot facturació
- ATR-cases



**Marta Fernández** 29 de abr. a las 18:48

No s'han pogut fer tests dels casos ATR W1 i A1 perquè no n'hi ha cap a testing. Pendent de crear-los.

 - [Editar](#) - [Eliminar](#)

## (1/1) [ADD] [Backend] Corregir tests

### Descripció:

No s'ha canviat el plantejament inicial dels tests. Per tal de poder seguir desenvolupant amb TDD, primerament s'havien de corregir els tests actuals.

Total de migdies esperats: 10

Total de migdies treballats: 15

## Ronda 10 (06-05-2020 a 05-06-2020):

### (10/4) [Frontend] Fix interfície

#### Descripció:

Després del refactor de la interfície, la idea és revisar que tot funciona correctament i contemplar en aquesta targeta el temps dedicat a revisar-ho i arreglar els possibles errors que es vegin.

#### Activitat:

Checklist Ocultar elementos completados Eliminar

100%

- Variable que indiqui num de comptadors i factures (per agilitzar els tests)
- Canviar a 12 comptadors i factures.
- Cercar info: Revisar que les cerques funcionen
- Revisar que el registre de trucades ateses: Pots seleccionar i deseleccionar
- Revisar que el registre de trucades ateses: Pots seleccionar i respondre sobre la trucada amb cerca (i s'actualitza ok)
- Revisar que trucades entren bé amb curl: quan hi ha info
- Revisar que trucades entren bé amb curl: quan no hi ha info
- Revisar que el registre de trucades ateses: S'actualitza bé el registre de trucades que han entrat
- Respondre qüestionari (Trucades): Revisar que la info es guarda bé a les trucades ateses
- Respondre qüestionari (Cerques): Revisar que trucada + cerca + resposta funciona i es guarda bé a les trucades ateses
- Identificació funciona
- Des-identificació funciona
- Respondre qüestionari (Trucades): Revisar que la info es guarda bé a les reclamacions
- Respondre qüestionari (Cerques): Revisar que la info es guarda bé a les reclamacions
- canviar css table encapsulat dins comptadors/factures
- Revisar settings dialog
- Bloquejar qüestionari quan no trucada.

Añada un elemento

**BUGS**
Ocultar elementos completados
Eliminar

50%

- Revisar lot facturació.
- Quan tens seleccionat un contracte número 3 i et mous a un que només en té un no es veu res
- Revisar les factures que es mostren.
- Desidentificacio chrome?

Añada un elemento

**Millores**
Ocultar elementos completados
Eliminar

25%

- Canviar com es mostra si té generation i quan és d'energetica (perquè ocupi menys espai) 🕒 👤 ⋮
- Quan no hi ha factures, que posi "No hi ha informació de factures"
- Canviar com es mostren els casos ATR oberts, si és d'energètica, si té generation o facturació suspesa.
- Cercar info: cerca per contracte

Añada un elemento

### (3/2) [Backend] Fix pujar ERP casos AC

**Descripció:**

Arreglar problemes detectats al pujar al ERP els casos AC

**Activitat:**

**Checklist**
Ocultar elementos completados
Eliminar

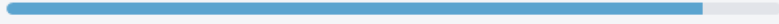
100%

- Revisar que es crea el cas bé al ERP amb la funció
- Revisar que script OK
- Revisar si error + log

Añada un elemento



**creació cas ATC** Ocultar elementos completados Eliminar


90% 

- Subtipus-reclamació
- Descripció-subtipus
- Comentaris
- Reclamant
- Empresa
- Contacte-empresa
- Pòlissa
- CUPS
- Agent-Actual
- Canal
- Imputació-de-temps
- Usuari ?
- Responsable ?
- Descripció-?
- Secció
- Email-empresa
- Status-obert,-resultat-buit
- Status-tancat,-resultat-procedente
- Status-tancat,-resultat-improcedente
- status-tancat,-resultat,-no-gestionable



**Marta Fernández** 18 de may, a las 13:40

Es deixa com a Descripció el camp `claim.desc` .  
Quedaria veure si l'assignació de l'usuari i responsable es fa bé. Com que a testing tots els correus són els mateixos i la relació es fa entre iden/correu, apuntant a testing no es pot cercar a través del correu.

 - [Editar](#) - [Eliminar](#)



**Marta Fernández** 19 de may, a las 15:34

Faltaria acabar de definir com comunicar els errors i la manera de gestionar-ho.

 - [Editar](#) - [Eliminar](#)

## (2/2) [Gestió] Crear entorn de testing de Tomàtic

### Descripció:

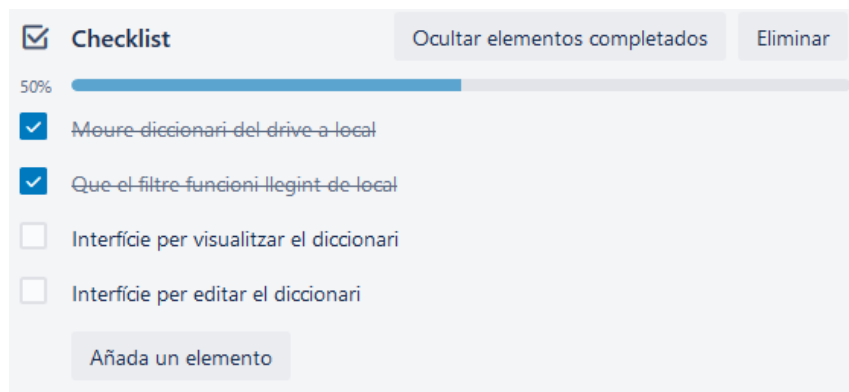
Crear entorn de Testing per al Tomàtic en un nou servidor.

## (1/2) [Frontend] Interfície per editar el diccionari de paraules clau

### Descripció:

Començar modificacions necessàries per a extreure el diccionari de paraules clau.

### Activitat:



The screenshot shows a checklist titled "Checklist" with a progress bar at 50%. The checklist items are:

- Mòdre diccionari del drive a local
- Que el filtre funcioni llegint de local
- Interfície per visualitzar el diccionari
- Interfície per editar el diccionari

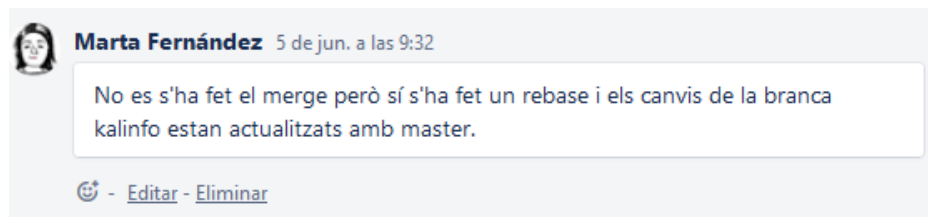
Buttons: "Ocultar elementos completados" and "Eliminar". A button "Añada un elemento" is at the bottom.

## (2/2) [Gestió] Fer merge a master

### Descripció:

Fer merge a master.

### Activitat:



The screenshot shows a commit message from Marta Fernández, dated 5 de jun. a las 9:32. The message text is:

No es s'ha fet el merge però sí s'ha fet un rebase i els canvis de la branca kalinfo estan actualitzats amb master.

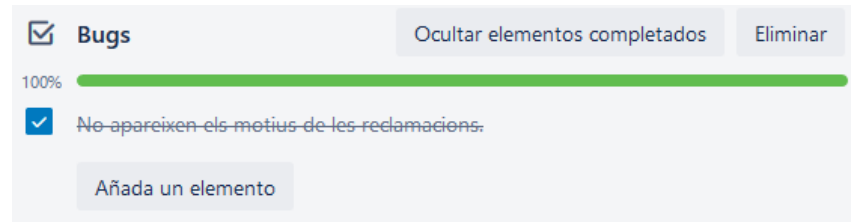
Buttons: "Editar" and "Eliminar".

## (2/4) [ADD] [Frontend] Fix bugs

### Descripció:

Targeta que inclou el temps dedicat a solucionar els bugs importants contemplats en la posada a testing abans de posar a producció.

### Activitat:



The screenshot shows a bug tracking interface. At the top, there is a header with a checkmark icon, the word "Bugs", and two buttons: "Ocultar elementos completados" and "Eliminar". Below the header, a green progress bar is shown at 100%. Underneath the progress bar, there is a list of bugs. The first bug is checked with a blue checkmark and has the text "No apareixen els motius de les reclamacions.". At the bottom of the list, there is a button labeled "Añada un elemento".

Total de migdies esperats: 16

Total de migdies treballats: 20

ADD: 1 targeta – 2 mitjos dies

## Ronda 11 (06-06-2020 a 26-06-2020):

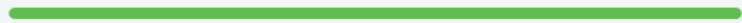
### (2/2) [Backend] Fix bugs creació casos ATC

#### Descripció:

Fix dels bugs o coses que s'haurien de canviar detectades després de la demo.

#### Activitat:

**Bugs** Ocultar elementos completados Eliminar

100% 


- que només surtin les factures client
- que no hi hagi responsable
- que la descripció aparegui directament (historize)

Añada un elemento



**Marta Fernández** 8 de jun. a las 15:25

No hi ha les adreces als usuaris de ERP i, per tant, no es poden vincular els usuaris per saber qui l'ha creat a partir del Kalinfo (discrepància noms Tomàtic i ERP).

 - [Editar](#) - [Eliminar](#)

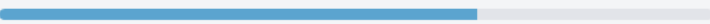
### (3/2) [Backend] Fix d'altres bugs

#### Descripció:

Aquesta targeta contempla el temps dedicat a fixar altres bugs que es vagin trobant mentre es corregeixen/canvien altres coses.

#### Activitat:

**Checklist** Ocultar elementos completados Eliminar

67% 

- identificar - desidentificar
- creació carpeta logs per a la creació de casos ATC en python 2 i 3
- ASSIGNAR USUARI: index out of range

Añada un elemento



**Marta Fernández** 8 de jun. a las 11:14

**ASSIGNAR USUARI: index out of range:** no s'ha pogut reproduir l'error, ha funcionat correctament. Les proves es feien sobre un contracte en mal estat.

 - [Editar](#) - [Eliminar](#)

## (2/2) [Frontend] Visualització correcta des de diferents mides de pantalles

### Descripció:

Que la infotmació es pugui visualitzar correctament desde diferents mides de pantalles.

### Activitat:



**Marta Fernández** 16 de jun. a las 17:05

Bones @joanutges @judithfrigole1 ! He fet un canvi a la interfície del Kalinfo:

Ara la informació de si té casos ATR oberts, si és d'energètica, si té facturació suspesa i si té generation apareix en un requadre al costat del contracte sota el títol "Altres".

Estava fent correccions perquè es visualitzés correctament i quan aquesta info (que només apareix quan, per exemple, la persona té generation) sortia, provocava que el quadre de contractes quedés una mica tallat de sota... Així que ho he posat tot al costat (en l'espai buit que quedava). Què us sembla? :D

Ho podeu veure a testing:



😊 - [Editar](#) - [Añadir enlace como un adjunto](#) - [Eliminar](#)

## (1/1) [Frontend] Anàlisi que la informació del Kalinfo sigui responsive

### Descripció:

Modificacions perquè el Frontend sigui un mínim de responsive.

Total de migdies esperats: 7

Total de migdies treballats: 8

