



---

# GESTOR D'ABSÈNCIES DE SOM ENERGIA

---

Treball de Final de Grau



**CURS 2018 - 2019**

**ALUMNE: BENJAMÍ RAMOS SOLÍS**

**TUTOR: JOSEP SOLER I MASÓ**

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



## Agraïments

La realització d'aquest Treball de Final de Grau ha sigut possible gràcies a l'ajuda i el recolzament per part dels companys de feina, especialment del tutor de l'empresa, Juan Pedro Sánchez i l'expert en programació d'interfícies web David Riera. També vull agrair el recolzament de la Mireia Pérez durant la realització de la memòria.

Finalment, també vull agrair la col·laboració per part del meu tutor, Josep Soler, al llarg del desenvolupament del treball.



## Contingut:

<b>1. Introducció, motivacions, propòsits i objectius del projecte .....</b>	<b>- 9 -</b>
<b>1.1. Introducció .....</b>	<b>- 9 -</b>
<b>1.2. Som Energia .....</b>	<b>- 9 -</b>
<b>1.3. Necessitats del Projecte .....</b>	<b>- 10 -</b>
<b>1.4. Motivacions .....</b>	<b>- 10 -</b>
<b>1.5. Objectius del projecte .....</b>	<b>- 11 -</b>
<b>2. Estudi de la viabilitat del projecte.....</b>	<b>- 13 -</b>
<b>2.1. Viabilitat en relació als Recursos Humans .....</b>	<b>- 13 -</b>
<b>2.2. Viabilitat tecnològica .....</b>	<b>- 13 -</b>
<b>2.3. Valoració global de l'estudi de viabilitat .....</b>	<b>- 13 -</b>
<b>3. Marc de treball i conceptes previs. ....</b>	<b>- 15 -</b>
<b>3.1. Participants en l'elaboració d'aquest Treball de Final de Grau .....</b>	<b>- 15 -</b>
<b>3.2. Llenguatges de programació, frameworks i tecnologies .....</b>	<b>- 16 -</b>
3.2.1. Python.....	- 16 -
3.2.2. Flask .....	- 17 -
3.2.3. Django .....	- 18 -
3.2.4. PostgreSQL.....	- 19 -
3.2.5. API REST.....	- 20 -
3.2.6. Django Rest Framework.....	- 21 -
3.2.7. JSON Web Token .....	- 21 -
3.2.8. JavaScript .....	- 22 -
3.2.9. Mithril.....	- 23 -
<b>3.3. Metodologies de programació .....</b>	<b>- 23 -</b>
3.3.1. Docker .....	- 23 -
3.3.2. Unit testing .....	- 25 -
3.3.3. Test-driven Development .....	- 26 -
<b>4. Metodologia.....</b>	<b>- 27 -</b>
<b>4.1. Metodologia de planificació .....</b>	<b>- 27 -</b>
<b>4.2. Control de versions .....</b>	<b>- 27 -</b>
<b>5. Requeriments del sistema.....</b>	<b>- 29 -</b>
<b>5.1. Actors .....</b>	<b>- 29 -</b>
<b>5.2. Requisits Funcionals .....</b>	<b>- 29 -</b>
5.2.1. Requisits Funcionals del Mòdul d'Absències .....	- 29 -
5.2.2. Requisits Funcionals del Mòdul d'Integració.....	- 30 -

5.2.3.	Requisits Funcionals del Mòdul de Gestió.....	- 30 -
<b>5.3.</b>	<b>Requisits No Funcionals.....</b>	<b>- 31 -</b>
<b>5.4.</b>	<b>Restriccions per part de l'empresa.....</b>	<b>- 32 -</b>
<b>5.5.</b>	<b>Requeriments de Llicència.....</b>	<b>- 32 -</b>
<b>5.6.</b>	<b>Matriu de dependències .....</b>	<b>- 32 -</b>
<b>6.</b>	<b><i>Planificació .....</i></b>	<b>- 37 -</b>
<b>6.1.</b>	<b><i>Paquets de treball.....</i></b>	<b>- 37 -</b>
6.1.1.	Diccionari de paquets de treball.....	- 38 -
<b>6.2.</b>	<b>Matriu de traçabilitat .....</b>	<b>- 45 -</b>
<b>6.3.</b>	<b>Planificació dels paquets de treball .....</b>	<b>- 46 -</b>
6.3.1.	Pressupost per Paquets de Treball.....	- 48 -
<b>7.</b>	<b><i>Estudis i decisions.....</i></b>	<b>- 51 -</b>
<b>7.1.</b>	<b>Decisions generals sobre el projecte .....</b>	<b>- 51 -</b>
7.1.1.	La possibilitat d'utilitzar els Templates de Django o una API REST amb Django REST Framework.....	- 51 -
7.1.2.	Reaprofitament d'eines o dades de l'empresa .....	- 52 -
7.1.3.	La utilització de Github.....	- 53 -
7.1.4.	Motius de la utilització de Docker .....	- 53 -
<b>7.2.</b>	<b>Decisions sobre el desenvolupament del Backend .....</b>	<b>- 53 -</b>
7.2.1.	La utilització de Django .....	- 54 -
7.2.2.	La no utilització de la llibreria Django-swingtime .....	- 54 -
7.2.3.	Autenticació a l'API amb la llibreria Django-rest-framework-simplejwt .....	- 55 -
7.2.4.	Llenguatge utilitzat per la documentació de l'API .....	- 55 -
7.2.5.	PostgreSQL.....	- 57 -
<b>7.3.</b>	<b>Decisions sobre el desenvolupament del Frontend .....</b>	<b>- 57 -</b>
7.3.1.	La utilització de Mithril.....	- 57 -
<b>8.</b>	<b><i>Anàlisi i disseny del sistema .....</i></b>	<b>- 59 -</b>
<b>8.1.</b>	<b>Anàlisi del sistema .....</b>	<b>- 59 -</b>
8.1.1.	Anàlisi dels mòduls de gestió d'usuaris .....	- 59 -
8.1.2.	Anàlisi dels mòduls de gestió absències .....	- 60 -
<b>8.2.</b>	<b>Disseny del sistema .....</b>	<b>- 61 -</b>
8.2.1.	Disseny del model de Dades.....	- 61 -
8.2.2.	Disseny de processos de l'aplicació .....	- 70 -
8.2.3.	Disseny de casos d'us complexes .....	- 71 -
8.2.4.	Disseny de les interfícies del Frontend .....	- 73 -
<b>9.</b>	<b><i>Implementació i proves .....</i></b>	<b>- 79 -</b>

<b>9.1.</b>	<b>Preparació de l'entorn del Backend</b> .....	<b>- 79 -</b>
9.1.1.	Instal·lació de Docker .....	- 79 -
9.1.2.	Creació d'un contenidor .....	- 80 -
9.1.3.	Creació dels contenidors necessaris per desenvolupar el Backend .....	- 83 -
9.1.4.	La persistència de la base de dades i la sincronització del codi .....	- 84 -
9.1.5.	Instal·lació de Django .....	- 85 -
9.1.6.	Creació d'un projecte amb Django .....	- 88 -
9.1.7.	Implementació dels models amb Django .....	- 90 -
9.1.8.	Integració amb Django REST Framework i primeres crides .....	- 92 -
<b>9.2.</b>	<b>Desenvolupament del Backend</b> .....	<b>- 94 -</b>
9.2.1.	Problemes a l'hora d'escollir la llibreria Django-swingtime per les absències .....	- 94 -
9.2.2.	Corba d'aprenentatge de Django juntament amb Docker .....	- 95 -
<b>9.3.</b>	<b>Preparació de l'entorn de Frontend</b> .....	<b>- 98 -</b>
9.3.1.	Creació d'un component de Mithril .....	- 99 -
<b>9.4.</b>	<b>Proves</b> .....	<b>- 100 -</b>
9.4.1.	Peticions testejadades en el Backend .....	- 100 -
9.4.2.	Test de rendiment al Backend .....	- 104 -
<b>10.</b>	<b><i>Implantació i resultats</i></b> .....	<b>- 107 -</b>
10.1.	<b>Resultat d'interfície</b> .....	<b>- 107 -</b>
10.2.	<b>Resultat final del projecte</b> .....	<b>- 122 -</b>
10.3.	<b>Implantació del projecte a l'entorn de testing</b> .....	<b>- 125 -</b>
10.4.	<b>Test de concurrència i rendiment en l'entorn de testing</b> .....	<b>- 128 -</b>
<b>11.</b>	<b><i>Conclusions</i></b> .....	<b>- 133 -</b>
11.1.	<b>Valoració dels objectius assolits</b> .....	<b>- 133 -</b>
11.2.	<b>Desviació de la planificació</b> .....	<b>- 133 -</b>
11.3.	<b>Aprenentatge assolit amb la realització del projecte</b> .....	<b>- 134 -</b>
<b>12.</b>	<b><i>Treball futur</i></b> .....	<b>- 137 -</b>
<b>13.</b>	<b><i>Webgrafia</i></b> .....	<b>- 139 -</b>
13.1.	<b><i>Webgrafia utilitzada pel Backend</i></b> .....	<b>- 139 -</b>
13.2.	<b><i>Webgrafia utilitzada pel Frontend</i></b> .....	<b>- 143 -</b>
<b>14.</b>	<b><i>Annexos</i></b> .....	<b>- 145 -</b>
14.1.	<b>Planificació final detallada</b> .....	<b>- 145 -</b>
14.1.1.	Planificació Ronda [R-00] Beta-tester .....	- 146 -
14.1.2.	Planificació Ronda [R-01] No sin mi equipo .....	- 148 -
14.1.3.	Planificació Ronda [R-02] No es necesacio reinventar la rueda .....	- 150 -
14.1.4.	Planificació Ronda [R-03] No todos los dies son iguales .....	- 151 -

14.1.5.	Planificació Ronda [R-04] El final de un ciclo y el inicio del siguiente.....	- 152 -
14.1.6.	Planificació Ronda [R-05] Lo importante es el interior.....	- 154 -
14.1.7.	Planificació Ronda [R-06] Acuestate y suda.....	- 156 -
14.1.8.	Planificació Ronda [R-07] El orgullo del camino recorrido .....	- 158 -
<b>14.2.</b>	<b>Documentació de l'API .....</b>	<b>- 162 -</b>



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

## 1.1. Introducció

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

En un primer moment comptava amb pocs treballadors, ja que la majoria de feina la realitzaven voluntaris, tot i així degut a l'augment de la feina, en els darrers mesos, ha vist la necessitat d'ampliar el seu equip de treball.

Tot i que Som Energia compta amb un departament d'IT, delega part dels desenvolupaments a empreses externes per tal de poder satisfer l'elevada demanda, fent així que IT realitzi certes modificacions als aplicatius ja existents, desenvolupi algun projecte i gestioni els partners tecnològics.

## 1.2. Som Energia

Som Energia és una cooperativa sense ànim de lucre que 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.

Tot i que es va crear a finals del 2010 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, també va ser aquell any quan es van iniciar els primers projectes de generació d'energia renovable.

La seu de Som Energia sempre ha estat ubicada en el Parc Científic i Tecnològic de la Universitat de Girona, però degut al seu creixement ha anat canviant de situació física però sempre dins del propi Parc. Actualment, i des de l'estiu del 2018 està ubicada a l'edifici Airbone.

Degut a la seva estructura cooperativista ofereix moltes activitats perquè les persones sòcies puguin estar informades, participar i prendre consciència del model energètic i del funcionament tant de la cooperativa com del mercat elèctric.

Amb els anys, Som Energia ha anat evolucionant amb idees innovadores com el Generation kWh, una manera de permetre l'autoconsum compartit sense tenir les plaques solars a casa, o contribuint en actes per lluitar contra la desigualtat o afavorir el reciclatge.

L'estructura empresarial de Som Energia està compost per quatre entitats principals:

- El Consell Rector, format per persones sòcies, de manera voluntària i no remunerada que vetllen per la implementació de les directrius d'actuació de la cooperativa.
- Gerència, consistent en un conjunt de persones de l'equip de treball que gestionen la relació entre el Consell Rector i el dia a dia de l'Equip de Treball.
- Equip de treball, integrat per treballadors assalariats de la cooperativa que realitzen les tasques del dia a dia. Actualment hi ha 69 treballadors.
- Persones sòcies, que mitjançant votacions escullen les directrius d'actuació de la cooperativa. A més, cal destacar la figura dels grups locals, consistents en grups de persones sòcies que, segons el seu lloc geogràfic realitzen activitats i xerrades sobre la temàtica del model energètic actual, energies renovables, compromís social, i moltes d'altres temàtiques en concordança amb la cooperativa.

### 1.3.Necessitats del Projecte

Degut a l'augment en les persones que formen l'Equip de Treball, molts mecanismes i plataformes que anteriorment utilitzaven han esdevingut molt difícils de gestionar, com és el cas del Gestor d'Absències. Fins al moment, les diferents absències dels treballadors, ja sigui per baixes laborals, vacances, etc, es gestionaven amb un document compartit al Google Drive on tots els treballadors tenien permisos d'edició, cosa que els facilitava poder seleccionar les seves absències.

Per aquest motiu, Som Energia necessita un aplicatiu capaç de gestionar qualsevol tipus d'absència a l'oficina, com per exemple les vacances, i que permeti consultar de en tot moment, de forma ràpida, senzilla i eficient així com, sense temps de càrrega.

### 1.4.Motivacions

Tot i que part dels aplicatius de l'empresa estan desenvolupats per empreses externes, des del departament d'IT s'intenta agafar cada cop més el control d'aquests programes. Per això, el fet de crear un aplicatiu des de zero utilitzant uns frameworks que actualment s'estan usant a l'empresa, però que són gestionats majoritàriament per empreses externes, és una gran motivació per tal d'empoderar al departament d'IT de les tecnologies amb les que estan fetes les seves plataformes.

A més a més, el fet de dur a terme la totalitat del desenvolupament d'un projecte, des de recollir els requisits fins a implantar l'aplicatiu resultant, passant per la seva programació, tant del Backend com del Frontend, és la millor ocasió per aprendre a desenvolupar software, cosa que és el que més m'ha agradat de tot el Grau en Enginyeria Informàtica.

A partir d'ara, en aquest treball es parlarà de Backend per fer referència a la part del servidor i Frontend per fer al·lusió a la part del client web.

D'aquesta manera, també espero poder posar en pràctica tots els coneixements adquirits durant el grau, especialment en les assignatures d'Enginyeria del Software i Projecte de programació. De la mateixa manera, poder aprofitar el Treball de Final de Grau per aprendre noves tecnologies i practicar metodologies molt utilitzades en el món laboral considero que és la millor manera de posar fi a aquests estudis.

## 1.5.Objectius del projecte

L'objectiu d'aquest Treball Final de Grau és desenvolupar l'anàlisi, disseny i implementació d'un aplicatiu per la gestió de les absències dels treballadors de Som Energia per tal de substituir l'actual gestió feixuga de les absències per una aplicació web més robusta i intuïtiva.

Aquest sistema també haurà de facilitar la visió general de les persones que es troben treballant a l'oficina amb un calendari molt més visual i ràpid en comparació amb l'antic mètode usat per Som Energia.


*Antic Gestor d'Absències*

D'aquesta manera, un cop l'aplicació es trobi en funcionament, aquest haurà de permetre:

1. Seleccionar dies d'absència.
2. Configurar els tipus d'absències.
3. Definir calendaris amb els seus dies laborals i festius.
4. Que els treballadors puguin afegir-se a un o més equips.
5. Quantificar/modificar els dies laborals i els de vacances dels treballadors.

6. Que l'aplicatiu es pugui integrar en el programari existent a l'empresa, per això es disposarà d'un mòdul d'integració amb el que ha de poder comunicar quins dies treballa cadascun dels treballadors a la resta de Sistemes d'Informació de l'empresa.
7. Crear i eliminar usuaris i grups de treball.
8. Establir mínims de treball i en cas de no compliment que es notifiqui.
9. Veure les absències pròpies i les dels altres membres de l'Equip de Treball.

## 2. Estudi de la viabilitat del projecte

Per poder justificar la viabilitat d'aquest projecte és important acreditar la seva implementació en relació a l'impacte econòmic que pot suposar, els recursos humans que es poden requerir, així com la complexitat que pot comportar el projecte.

### 2.1. Viabilitat en relació als Recursos Humans

En referència a la viabilitat d'aquest projecte pel que requereix de recursos humans, al departament d'IT de Som Energia hi ha molts projectes que actualment no es poden realitzar per falta d'informàtics, un exemple d'aquest era el Gestor d'Absències, per aquest motiu, des d'IT es va proposar aquest Treball de Final de Grau, d'aquesta manera es podrà implementar l'aplicatiu sense que el desenvolupament d'altres projectes es vegi afectat.

Cal destacar també el fet que el mentoratge i les hores de dedicació que tindran altres treballadors de la cooperativa serà ínfim i no repercutirà en les seves tasques diàries.

### 2.2. Viabilitat tecnològica

Pel que fa a la complexitat, el Gestor d'Absències es pot assolir com a projecte en l'àmbit de les Tecnologies de la Informació ja que té un pressupost acotat i un àmbit d'actuació correctament definit i delimitat. Tot i que el desenvolupament d'aquest projecte pot comportar moltes hores de feina, aquest es pot desgranar en diferents tasques unitàries i independents entre sí que, o bé s'implementen en el marc d'aquest Treball de Final de Grau, o s'implementen en el dia a dia de la cooperativa.

També cal destacar que la totalitat del software i hardware necessari per desenvolupar aquesta aplicació no ha de suposar cap problema ja que s'utilitzarà l'ordinador que normalment utilitza a la feina, juntament amb el fet que el software necessari per desenvolupar és lliure i es pot disposar d'ell sense cap tipus de problema.

### 2.3. Valoració global de l'estudi de viabilitat

El factor més decisiu per valorar la viabilitat d'aquest projecte és el fet que la cooperativa està molt interessada en alguna aplicació per modernitzar la seva gestió d'absències. Tot i que el pressupost no s'ha calculat des d'un inici, s'estima que no serà gaire elevat. Aquest fet, juntament

amb la possibilitat d'assolir els costos en recursos humans i la viabilitat del projecte amb les tecnologies que sol treballar la cooperativa, Som Energia veu viable que el departament d'IT reculli els requeriments i encamini la implementació d'aquest projecte.

### 3. Marc de treball i conceptes previs.

Tot i que part de la formació en les tecnologies i l'anàlisi d'aquestes s'ha realitzat de forma prèvia, a mesura que han sorgit dubtes durant la programació aquests s'han anat resolent.

Cal tenir en compte que, al tractar-se d'un projecte realitzat en el marc de treball d'una empresa, el llenguatge de programació en que es desenvoluparà serà Python i JavaScript, ja es tenen clars molt abans de començar a encarar el projecte, degut a que són els llenguatges amb que treballa en el departament d'IT de Som Energia. Altres decisions com els frameworks que s'utilitzaran i les metodologies amb les que es durà a terme el desenvolupament s'han analitzat per valorar els avantatges i els inconvenients que poden suposar.

Per elaborar aquest anàlisi es tindran en compte tres vessants, l'entorn dins de Som Energia, els llenguatges de programació, frameworks i tecnologies que s'utilitzaran pel desenvolupament del projecte i la metodologia amb la que es realitzarà.

#### 3.1. Participants en l'elaboració d'aquest Treball de Final de Grau

Tot i que l'anàlisi de requisits, la planificació i el desenvolupament del projecte els realitzaré jo mateix, per la correcta creació i l'assoliment dels objectius plantejats, hauré de relacionar-me amb diferents persones de l'Equip de Treball de Som Energia. Aquestes persones són:

- **Juan Pedro Sánchez**, que realitzarà tasques d'assessorament des del punt de vista d'un desenvolupador sènior amb experiència en la recollida de requeriments, planificació i en la utilització de Django.
- **Anna Rodón**, treballadora del departament de recursos humans i encarregada de gestionar l'antiga plataforma per gestionar les absències. Serà la figura de client per aquest projecte i la persona amb la que identificaré i analitzaré els requisits del sistema.
- **David Riera**, que Realitzarà tasques d'assessorament des del punt de vista d'un desenvolupador sènior amb experiència en el desenvolupament d'interfícies web.

També cal tenir en compte que al tractar-se d'un projecte que es realitzarà dins d'una empresa, tot el departament d'IT tindrà una relació indirecta amb el projecte, ja que en el moment de planificar les diferents tasques de l'empresa, s'haurà de destinar una part de la meva jornada de treball per desenvolupar el Gestor d'Absències i una altre part de la jornada per incidències o al desenvolupament d'altres projectes de l'empresa.

## 3.2.Llenguatges de programació, frameworks i tecnologies

A continuació es descriuran els entorns de desenvolupament amb els que es realitzarà aquest Treball de Final de Grau, en concret Python, Django, PostgreSQL, API REST, Django REST Framework, JSON Web Token, JavaScript i Mithril.

### 3.2.1. Python

A principis de l'any 1994 va sorgir Python, un llenguatge interpretat amb especial èmfasi en elements que ajudin la comprensió i llegibilitat del codi, com ara la tabulació. Tanta importància té això per aquest llenguatge que existeixen diferents guies d'estil de codi, sent PEP 8 la més popular.

```
a = 2
if a < 2:
    print('less than 2')
else:
    print("more than 2")

print('hello')

# Sample block 3
for i in range(5):
    print(i)
```

*Sintaxi del Python*

Actualment, Python té llibreries i frameworks per multitud de finalitats, sent dels llenguatges més utilitzat per anàlisis de dades, Machine learning o fins i tot en entorns web. A més a més, la comunitat és molt oberta i participativa, aspecte que facilita molt trobar solucions, ajudes o llibreries que solucionin gran part dels nostres problemes de programació. També cal destacar que és una comunitat força vinculada amb el software Open Source, la qual cosa ajuda encara més a trobar llibreries o utilitats a l'hora de programar.

Python és un llenguatge multiparadigma, és a dir, permet elaborar codi segons diferents paradigmes, com ara la programació orientada a objectes, la programació imperativa o la funcional.

Una de les característiques més destacables de Python són els seus tipus dinàmics, aspecte que provoca que el tipus d'objecte d'una variable canviï en temps d'execució.



Així mateix, Python disposa d'un intèrpret donat per defecte al instal·lar-ho, però es poden utilitzar d'altres com l'ipython que aporta una major funcionalitat, a més que visualment és més fàcil d'interpretar.

Una de les principals problemàtiques que ens podem trobar a l'hora de desenvolupar diferents aplicatius amb Python és la necessitat d'utilitzar diferents versions d'una mateixa llibreria, aspecte que no és possible. Per resoldre aquesta dificultat van sorgir els virtualenv, una manera d'encapsular el binari de Python i un conjunt de llibreries en una carpeta. D'aquesta manera, a l'hora de desenvolupar tindriem un virtualenv per cada projecte, fent així que cada un tinguéssim uns requeriments i no entressin en conflicte entre ells.

Una eina essencial per utilitzar Python és Pip, el gestor de paquets que permet llistar, buscar, instal·lar i desinstal·lar paquets i llibreries que puguem necessitar per la nostra aplicació escrita amb Python.

### 3.2.2. Flask

Flask és un framework de codi lliure pel desenvolupament d'aplicacions web. És un framework conegut per ser molt simple i permetre, amb poca complexitat, aixecar un Backend que escolti peticions. El seu logotip és el següent:



*Logotip de Flask*

Flask permet molta llibertat al desenvolupador que l'utilitza, aquest fet pot provocar que, si no s'utilitza correctament, un projecte desenvolupat amb aquesta tecnologia pugui arribar a ser caòtic. Per tal que això no succeeixi, Flask marca unes pautes per estructurar el codi per tal que qualsevol desenvolupador pugui entendre el codi del Backend.

Una de les principals formes d'estructurar el codi que té Flask són els Blueprints, aquests es poden definir en qualsevol fitxer per tal d'estructurar els diferents mòduls de l'aplicació.

Tot i que per aplicacions molt simples o reduïdes pot ser molt útil, Flask no està pensat per projectes molts grans ni que requereixin molta optimització.

### 3.2.3. Django

Django és un framework de codi lliure per al desenvolupament d'aplicatius web. Va ser escrit en Python i va ser alliberat l'any 2005. El seu logotip és el següent:



*Logotip de Django*

Django implementa el patró Model - Vista – Template, permetent separar molt bé la part lògica del projecte, la part del model de dades i la part mostrada als usuaris. Això provoca que Django faciliti molt el desenvolupament d'aplicacions web molt complexes.

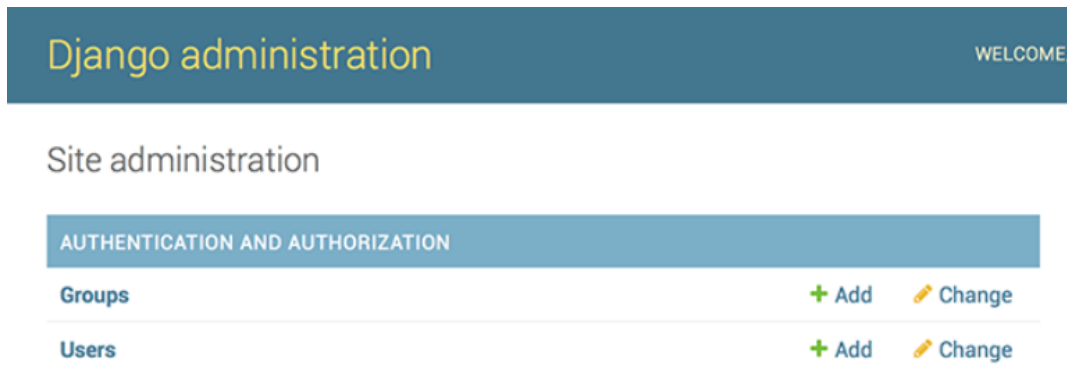
Un dels principals fonaments de Django és la reusabilitat del codi i la connexió entre les diferents parts del projecte, posant així especial èmfasi en el principi DRY, Don't Repeat Yourself. També cal destacar que és un framework amb el que es pot desenvolupar molt ràpidament aplicacions web degut a la immensa capacitat de configuració i de feina ja feta amb anterioritat, tot i que la quantitat de codi elaborat i la complexitat de trobar la solució que ens escau dins de les possibilitats que ens ofereixen, pot comportar un augment en la dificultat de l'aprenentatge.

En comptes de parlar de llibreries o mòduls, en entorns de Django parlarem de les pròpies llibreries de Django o els mòduls de dins d'un projecte com aplicacions, ja que per Django, una aplicació és tot allò desenvolupat amb Django que es pot instal·lar i aprofitar en el nostre programa.

Django, juntament amb Python, té molta comunitat de codi lliure, aspecte que facilita el fet de trobar aplicacions per poder reutilitzar, frameworks dins del propi Django o projectes sencers.

Dos dels principals punts forts a l'hora de desenvolupar un projecte amb aquest framework són que:

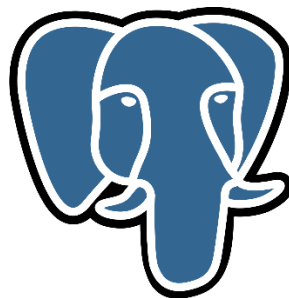
- Conté un intèrpret propi configurable de la manera que ens sigui més còmode. Amb aquest intèrpret podrem utilitzar el codi que s'està executant com si es tractés d'un ipython en un servidor en marxa o be modificar els nostres models deixant persistència en la base de dades sense necessitat d'interrompre el servei, cosa que és molt útil per investigar incidències.
- Django permet aixecar un client web amb una interfície bàsica i molt visual prèviament definida pel propi framework per tal de veure els models i modificar les entitats dels nostres programes. Al igual que el seu intèrpret, es pot modificar les dades sense necessitat d'aturar el servei. Aquesta utilitat s'anomena Admin, i com el seu propi nom indica, és molt utilitzada pels administradors de les plataformes per controlar el contingut.



*Interfície de l'Admin de Django*

### 3.2.4. PostgreSQL

PostgreSQL és un sistema de gestió de bases de dades relacional orientada a objectes originat al voltant de 1990. És de codi lliure i està dirigit per la comunitat. El seu logotip és el següent:



*Logotip de PostgreSQL*

Algunes de les principals característiques de PostgreSQL són el fet que suporta una gran concurrència d'usuaris, encara que utilitzin la mateixa taula, a més d'utilitzar la sintaxi estàndard SQL.

Actualment PostgreSQL és un dels quatre sistemes gestors de bases de dades relacionals més utilitzats. La principal eina que es sol utilitzar, juntament amb PostgreSQL és PgAdmin, un client per administrar les seves bases de dades.

### 3.2.5. API REST

API REST és una tecnologia de transferència d'informació entre diferents Sistemes d'Informació que va sorgir l'any 2000, tot i que abans ja existien diferents API, REST va guanyar i desbancar SOAP, l'estàndard en aquella època. Tot i que actualment té un fort competidor, GraphQL, REST segueix dominant la majoria d'API que es poden trobar avui en dia.

REST permet aprofitar un únic endpoint amb diferents mètodes de peticions HTTP, d'aquesta manera es poden realitzar diferents crides com podria ser el cas de crides GET, POST, PUT o DELETE al servidor per obtenir, crear, modificar o eliminar els recursos de l'aplicació.

Un endpoint és un punt d'entrada a l'API compost per una URL, un mètode HTTP, podent ser GET, PUT, etc i els headers o el body corresponent per la petició. D'ara en endavant s'utilitzarà el terme endpoint.

En el cas de la nostra aplicació, acostumarem a utilitzar els següents codis per retornar informació del servidor:

- 200: Aquest resultat indica al client de la petició que aquesta ha anat bé i que la resposta pot contenir informació en el cos del missatge.
- 201: Aquest resultat indica al client de la petició que el recurs s'ha creat correctament.
- 204: Aquest resultat indica al client de la petició que aquesta ha anat bé i no hi ha cos en el missatge.
- 400: Quan el servidor retorna aquest codi ens està comunicant que la petició feta és incorrecte.
- 401: Aquest codi indica al client que no té permisos per realitzar l'acció que ha intentat fer.
- 403: Aquest codi indica al client que no pot accedir al recurs que ha intentat accedir.
- 404: Aquest codi indica al client que el recurs al que està intentant accedir no existeix.
- 500: Aquest codi indica al client que hi ha hagut un error al processar la informació que li ha facilitat.

### 3.2.6. Django Rest Framework

Django Rest Framework és un dels molts frameworks de Django que ens permet ampliar les possibilitats que ens ofereix de per sí Django. Aquest framework permet interaccionar de manera fàcil i directa amb els models de dades. El seu logotip és el següent:



*Logotip Django Rest Framework*

Django Rest Framework substitueix la manera de retornar les dades, suprimint així la necessitat de tenir templates d'HTML per renderitzar la representació dels models i oferir a canvi la figura dels serialitzers. Els serialitzers són la capa que s'encarrega de definir quines dades s'ofereixen i es reben del client així com el format que han de tenir algunes de les validacions que han de complir o altres limitacions que es poden configurar de manera molt ràpida.

Aquest framework, juntament amb Django, està pensat per poder desenvolupar API REST molt ràpidament ja que, amb experiència i les idees del desenvolupament clares, la majoria de lògica i filtres ja estan implementats. Algun exemple d'aquesta lògica és la validació de credencials i permisos o el format d'entrada de les dades.

Al igual que Django presenta multitud de llibreries de tercers que es poden aprofitar per tal de fer un desenvolupament encara més ràpid.

### 3.2.7. JSON Web Token

JSON Web Token és un estàndard per la creació de tokens d'accés, aquests tokens d'accés permeten identificar un usuari. El token és un hash generat pel servidor utilitzant un xifrat SHA-256, aspecte que provoca un major grau de seguretat en vers la transmissió de credencials de l'usuari.

En moltes plataformes és utilitzat per tal de demanar les credencials a l'usuari un únic cop, ja que quan el servidor l'hagi identificat li proporcionarà un token que el client podrà fer servir per

acreditar-se. En la majoria d'ocasions, el token té un temps d'expiració després del qual serà necessari tornar a generar un de nou.

El token està compost de diferents parts:

- **Header:** Indica la codificació que s'ha utilitzat per generar el Token.
- **Payload:** Indica per qui ha estat creat el Token i pot contenir informació com el temps d'expiració per facilitar al servidor el coneixement del moment en què cal refrescar la clau d'accés.
- **Signature:** Es genera amb els primers dos camps i una clau encriptant-ho amb base64.

Les tres parts del Token estan encadenades utilitzant punts, un exemple de Token és:

```
"access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9  
.eyJleHAiOiJlMDU4MzE0MDYsInVzZXI6ImNybWVkbWluIiwiaXV0aG9yaXRpZXMiOiJ1Uk9MRV9VU0VSIiwiaUk9MRV9BRE1JTjJdLCJqdG  
kiOiI1NDE1N2IwNC1iN2IzLTQ0NTktYTMyMC05ZTNiMTQ1ZDU5OTUiLCJjbG11bnRfaWQiOiJjcm1DbG11bnQxIiwic2NvcGUiOiJicmVhZCI6Indya  
XR1IiwidHJ1c3QiXX0.gYbGmsmyM_XQAiMvKw00QAGB2H9L7Yr0AM7ihGvcPM",
```

*Exemple de JSON Web Token extret gràcies al Postman*

### 3.2.8. JavaScript

JavaScript és un llenguatge de programació interpretat, orientat a objectes i dèbilment tipat que va sorgir l'any 1997. Actualment es pot utilitzar per desenvolupaments de servidors o clients web, sent NodeJS el principal framework per desenvolupar servidors i Vue, React o Angular els principals per del desenvolupament de clients web. El seu logotip és el següent:



*Logotip de JavaScript*

En els seus orígens, el JavaScript s'utilitzava en pàgines estàtiques HTML per dotar-les de contingut dinàmic, actualment també es pot utilitzar per fer crides al servidor juntament amb tecnologies com AJAX.

### 3.2.9. Mithril

Mithril és un framework de JavaScript, que tot i que actualment és poc conegut, destaca per la seva rapidesa i per la seva mida reduïda a l'hora de descarregar els arxius. El seu logotip és el següent:



*Logotip de Mithril*

Tot i que actualment els frameworks més populars d'aquest llenguatge són Vue i React, Mithril té molts avantatges que el fan una molt bona alternativa. Alguns d'aquests avantatges són que:

- Els temps de càrrega són molt ràpids ja que utilitza un DOM virtual. El DOM virtual consisteix en una estructura en forma d'arbre que conté el DOM real i cada cop que aquest canvi no torna a renderitzar-lo sencer sinó que aplica les diferències.
- Té un pes molt reduït pel que fa a la descàrrega.

El gran inconvenient de Mithril és la dificultat per trobar components ja elaborats, cosa que la majoria de vegades es fa impossible i cal crear-los des de zero, fent que la maquetació del client web pugui arribar a ser més complicada.

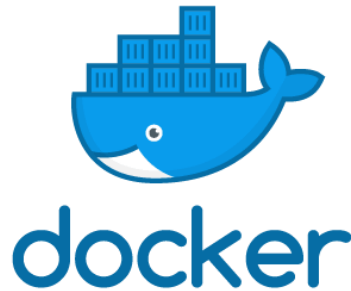
## 3.3. Metodologies de programació

En aquests apartats es descriuran les diferents metodologies de programació que s'han utilitzat per la implementació d'aquest projecte.

### 3.3.1. Docker

Docker és un software que permet automatitzar el desplegament d'aplicacions dins de contenidors, el que possibilita la creació d'entorns de desenvolupament, testing o integració continua de manera molt ràpida. Tot i que el desenvolupament ja fa que Docker sigui una

tecnologia molt a tenir en compte, es pot utilitzar també en projectes funcionals que s'han pujat a productiu gràcies als orquestadors. El seu logotip és el següent:



*Logotip de Docker*

Al parlar de Docker cal tenir present els diferents conceptes que envolten aquesta tecnologia:

- **Imatge:** Arxiu únicament de lectura que Docker utilitza per crear els contenidors. És un concepte semblant a les imatges ISO.
- **Contenedor:** Conjunt d'aplicacions i dependències empaquetades en un entorn aïllat que es poden executar de manera ràpida i fiable.
- **Dockerfile:** Fitxer de text que serveix per crear una imatge, en ell s'estipula en què es basa la imatge i es realitzen accions com afegir fitxers o modificar els ja existents.
- **docker-compose:** Eina per treballar amb múltiples contenidors simultàniament i que permet la seva creació, el seu aixecament o la seva destrucció. Per crear-los i aixecar-los utilitza un fitxer de text anomenat `docker-compose.yml` que estipula les condicions que tindran els contenidors a crear.

Els contenidors que aixeca Docker es podrien arribar a comparar amb les màquines virtuals treballades durant el grau però tenen grans diferències:

- Docker utilitza parts del Sistema Operatiu que fa de host provocant que el temps de càrrega sigui molt menor que el de les màquines virtuals, aquest fet també provoca que els contenidors siguin molt més lleugers que les màquines virtuals.
- A l'hora de creació de les màquines virtuals cal especificar els recursos als que tindrà accés, aspecte que no succeeix en el cas dels contenidors ja que poden utilitzar els recursos de la màquina host. Tot i així, existeixen maneres de limitar els recursos que poden utilitzar.



### 3.3.2. Unit testing

El unit testing és un mètode de testeig de software que consisteix en programar un seguit de tests per mètodes concrets o agrupacions de codi concretes per tal de poder controlar els canvis que es fan en aquella porció de codi. D'aquesta manera, si tenim un codi ben testejat, a l'hora de fer modificacions per afegir alguna funcionalitat podrem veure clarament si hem trencat alguna altra funcionalitat.

És una pràctica molt útil per tot tipus de desenvolupament ja que afavoreix que el software que s'utilitzi sigui de qualitat i es tinguin molt ben controlat les seves actuacions.

Tot i que alguns desenvolupadors creuen que el fet de programar els tests és una pèrdua de temps, ja que no són línies "efectives" de codi, actualment s'ha demostrat que els desenvolupaments amb codi testejat correctament progressen de manera més estable que els no testeats.

Alhora de testejar cal tenir en compte alguns conceptes com ara:

- **Test coverage:** És la cobertura de tests que té el codi, el 100% indica que totes les línies de codi estan cobertes, és a dir, que al menys 1 test passa per cada línia de codi. Tot i que l'ideal fos el 100%, a partir del 60% ja es considera un codi testejat.
- **Mock objects:** A l'hora de testejar, per exemple una funció, no només cal tenir en compte els paràmetres que aquesta agafa d'entrada sinó que també poden haver-hi altres fonts d'entrada com una base de dades o dades que es poden extreure del Sistema Operatiu com la data actual. Degut a que testejar una funció requereix tenir totes les dades d'entrada molt ben controlades van sorgir els Mocks, entitats del codi capaces d'emular les funcionalitats anteriorment esmentades.
- **Integració contínua:** Pràctica que es basa en l'execució dels tests desenvolupats en les diferents branques del repositori remot, fent així saltar una alerta en cas que els tests no passin, d'aquesta manera sempre s'intenta tenir una versió del codi estable en remot. Aquesta pràctica és molt utilitzada en desenvolupaments àgils.

Alhora de pensar un test, cal que aquest contingui, com a mínim, 3 etapes:

- **Arrange:** En aquesta etapa s'inicialitza les variables i les entrades que tindrà el fragment de codi que volem testejar.
- **Act:** En aquesta etapa es crida la funció o codi que es vol testejar.
- **Assert:** En aquest etapa es compara el resultat que ha tingut el codi respecte les variables que li hem donat i es comprova que el codi ha actuat correctament, com per exemple escriure a la base de dades.

### 3.3.3. Test-driven Development

El test-driven Development (TDD) és una pràctica en el desenvolupament del software que consisteix en escriure primer els testos i després fer el programa. Per tal d'aconseguir-hò es fa en 3 passos:

1. Escriure el test sense haver implementat la funció que es vol fer. D'aquesta manera fallarà.
2. Implementar la funció, fent així que el test passi correctament.
3. Un cop tenint la funció desenvolupada, apliquem millores o refactoris per tal de fer el codi més simple i elegant.

D'aquesta manera, en el TDD s'intenta traduir els requeriments a testos per tal que, quan els testos passin satisfactòriament, voldrà dir que l'aplicatiu s'ha programat correctament.

## 4. Metodologia

Aquest apartat té com a objectiu l'explicació de la metodologia que s'utilitzarà durant la creació del Gestor d'Absències, dividint-ho en la pròpia metodologia amb la que es programarà i com s'utilitzarà el control de versions.

### 4.1. Metodologia de planificació

Pel que fa a la planificació, el desenvolupament del Treball de Final de Grau, es realitzarà seguint una metodologia basada en Scrum Agile en la que es portaran a terme certes modificacions degut a la metodologia de desenvolupament del departament on s'elaborarà aquest projecte.

En comptes de Sprints, els períodes de temps en els que es programaran les diferents tasques es diran rondes, les quals tindran una durada de quasi 2 setmanes, començant sempre els dilluns i acabant els divendres.

La persona externa interessada en l'aplicatiu o Stakeholder serà una treballadora del departament intern de recursos humans. En comptes d'haver-hi un equip de desenvolupadors al càrrec del projecte, les tasques i la planificació d'aquestes les realitzaré jo mateix.

El tutor del projecte per part de l'empresa tindrà el rol de Scrum Master, aconsellant sobre com elaborar la planificació i aportant una visió més amplia dels requeriments de la cooperativa.

D'aquesta manera, aproximadament cada dues setmanes, amb l'inici de la ronda, ens reunirem tots tres per arribar a un acord de les tasques que es realitzaran durant la ronda. En acabar la ronda ens tornarem a reunir per fer una demostració de les tasques que s'han fet fins al moment i es valorarà la complexitat o problemes que hagin pogut sorgir durant la ronda, així com la seva resolució.

S'intentarà que cada tasca sigui independent de les altres i estigui ben definida per determinar el seu inici i la seva fi. Per tal d'organitzar les diferents tasques i estats en les que poden trobar-se s'utilitzarà el Trello en el que cada tasca es representarà com una targeta i cada columna com un estat, amb Not started, Doing i Done com els estats més freqüents.

### 4.2. Control de versions

Durant la programació, s'utilitzarà un sistema de control de versions, en aquest cas Github, en el que es treballarà amb diferents branques. Cada branca tindrà un nom adaptat a la tasca que es vol aconseguir amb ella. Els commits han d'intentar ser canvis unitaris i independents entre sí, amb

un títol descriptiu per tal d'afavorir la comprensió del codi i el seguiment del repositori per part de possibles contribuents.

Al repositori del Backend hi hauran dues branques principals: master, en la que només hi haurà el codi estable i des d'on sortiran totes les branques que es vagin generant i review, on es faran les Pull Request que l'encarregat, en aquest cas el tutor de l'empresa, revisarà i acceptarà el codi.

En el cas del repositori del Frontend, en canvi, s'efectuarà tot el desenvolupament en una única branca i no es realitzarà cap Pull Request.

El codi realitzat amb Python seguirà l'estil PEP 8 per tal que la comunitat Open Source pugui entendre fàcilment el codi, utilitzar-lo i realitzar aportacions.

Tot i que en el departament d'IT s'intenta treballar utilitzant la tècnica de Pair Programming, en el cas d'aquest projecte no es realitzarà ja que només el desenvoluparé jo.

El desenvolupament del Backend del projecte s'intentarà realitzar, sempre que sigui possible, amb test-driven development, cosa que aportarà molta seguretat al programador per fer refactoris i altres tipus de canvi en el codi. Tot i que també es podrien realitzar proves i test per la part Frontend, Som Energia no té prou experiència en testejar les interfícies web, pel que només es testejarà el Backend.

Degut a que aquest projecte es realitzarà dins d'un equip d'IT, diàriament es realitzarà una Stand-up Meeting amb els altres integrants del departament per abordar problemes que s'han trobat o oferir ajuda amb qualsevol de les tasques d'IT.

## 5. Requeriments del sistema

Aquest apartat té com a objectiu descriure els requisits que ha de complir el projecte, distingint-los entre requisits funcionals i no funcionals. També es veuran els actors que interaccionaran amb el sistema.

### 5.1. Actors

Els actors que interaccionaran amb l'aplicatiu seran els treballadors de Som Energia, podent haver-hi dos rols diferents: l'Administrador i el Treballador.

Tot i que l'Administrador serà el que tindrà els permisos per mantenir la plataforma i autoritzar certes accions, serà també un treballador, podent realitzar qualsevol acció que puguin fer els treballadors com escollir els seus dies de vacances.

### 5.2. Requisits Funcionals

A l'hora d'especificar els diferents requisits funcionals, es farà segons els diferents mòduls:

#### 5.2.1. Requisits Funcionals del Mòdul d'Absències

El Mòdul d'Absències haurà de permetre als usuaris de la plataforma veure i administrar qualsevol tipus d'absència. Els requisits per fer això són:

- **RF-00** Qualsevol usuari s'haurà de poder autenticar utilitzant un nom d'usuari i una contrasenya.
- **RF-01** Qualsevol usuari podrà veure el seu perfil.
- **RF-02** Qualsevol usuari podrà veure el perfil d'altres persones usuaris.
- **RF-03** Qualsevol usuari podrà veure el calendari d'absències de tots els usuaris.
- **RF-04** Qualsevol usuari podrà veure el calendari d'absències d'un treballador concret.
- **RF-05** Qualsevol usuari podrà veure el calendari d'absències d'un determinat Equip de Treball.
- **RF-06** Qualsevol usuari podrà seleccionar un o diversos dies de permís o absència.
- **RF-07** L'usuari administrador podrà crear nous tipus d'absència.
- **RF-08** L'usuari administrador podrà eliminar qualsevol tipus d'absència.
- **RF-09** L'usuari administrador podrà modificar qualsevol tipus d'absència.

- **RF-10** Es definiran els calendaris amb els seus dies laborals i festius.
- **RF-11** Qualsevol usuari podrà afegir-se a qualsevol Equip de Treball.
- **RF-12** Qualsevol usuari podrà deixar de formar part de qualsevol Equip de Treball.
- **RF-13** Qualsevol usuari podrà veure els dies festius que li queden per gaudir i els que ja ha gaudit.
- **RF-14** L'usuari administrador podrà assignar més dies de vacances als usuaris.
- **RF-15** L'usuari administrador podrà validar determinades absències dels usuaris.

Quan fem referència a cada un d'aquests requisits utilitzarem aquesta notació, afegint .1 si es tracta de la part Frontend i .2 si es tracta del Backend. D'aquesta manera quan fem referència a RF-05.1 voldrà dir la part Frontend del requisit funcional 5 i quan es parli de RF-05.2 voldrà dir que fem referència a la part Backend del requisit funcional 5.

### 5.2.2. Requisits Funcionals del Mòdul d'Integració.

El Mòdul d'Integració hauran de permetre la correcta interacció de l'aplicatiu amb els altres Sistemes d'Informació de Som Energia. Tot i que en un principi aquest mòdul no tindrà gaires requisits, tots els futurs requisits de comunicació amb altres Sistemes d'Informació que puguin sorgir hauran d'anar dins d'aquest mòdul.

L'únic requisit que ha de tenir aquest mòdul és el de poder comunicar quins treballadors estaran a l'oficina durant les següents dues setmanes al generador de graelles telefòniques, el Tomàtic.

### 5.2.3. Requisits Funcionals del Mòdul de Gestió.

El Mòdul de Gestió haurà de permetre als administradors de la plataforma gestionar el contingut de la plataforma. Els requisits per fer això són:

- **RF-30** Crear usuaris.
- **RF-31** Eliminar qualsevol usuari.
- **RF-32** Modificar les dades de qualsevol usuari.
- **RF-33** Crear un Equip de Treball.
- **RF-34** Eliminar qualsevol Equip de Treball.
- **RF-35** Modificar les dades de qualsevol Equip de Treball.
- **RF-36** Establir mínims de treball pels diferents Equips de Treball.
- **RF-37** Rebre notificacions en cas que els mínims establerts pels Equips de Treball no es compleixin.

- **RF-38** Aprovar les peticions dels Treballadors que siguin necessàries.
- **RF-39** Crear el calendari laboral de l'oficina.
- **RF-40** Modificar el calendari laboral de l'oficina.
- **RF-41** Eliminar un calendari laboral de l'oficina.
- **RF-42** Modificar els dies de vacances dels Treballadors.

Al igual que amb els requisits funcionals del mòdul d'absències, quan fem referència als requisits d'aquest mòdul utilitzarem aquesta notació, afegint .1 si es tracta de la part Frontend i .2 si es tracta del Backend. L'únic requisit que és unitari i no està separat en Frontend i Backend d'aquest mòdul és el RF-37.

### 5.3.Requisits No Funcionals

Tot i que la majoria de funcionalitats que l'usuari demanarà només es representaran amb requeriments funcionals, per tal de desenvolupar un aplicatiu de qualitat, amb la filosofia de Som Energia, i donar un bon servei als usuaris, s'hauran d'especificar i definir un conjunt de requisits no funcionals.

A part dels requisits funcionals de la plataforma, també s'han d'afegir requisits que assegurin la bona experiència dels usuaris. Aquests requeriments són que:

- Les interfícies seran desenvolupades per poder-se veure correctament en els navegadors Firefox i Google, Chrome amb la possibilitat de ser responsius.
- En cas d'error en algun paràmetre es mostraran missatges d'advertència descriptius.
- Es verificaran els límits pels camps que tinguin una relació directa amb els models de dades.
- S'implementaran mecanismes que assegurin la integritat de les dades.
- S'assegurarà la disponibilitat del Sistema 24 x 7 x 365.
- En cas d'errada s'arxivaran els logs per poder-se analitzar posteriorment.
- S'espera que el temps de resposta del Sistema en cas de prémer un botó no sigui superior a 5 segons.
- S'espera poder mantenir l'escalabilitat del sistema en relació a la concurrència dels usuaris fins a 40 simultàniament.
- S'espera poder mantenir l'escalabilitat del sistema en relació als usuaris que emmagatzema fins a 100.
- Les pantalles de la interfície no contindran faltes ortogràfiques.
- Es faran servir interfícies estandarditzades.

- Una persona que no conegui l'aplicatiu podrà fer-la servir i aprendre totes les funcionalitats en 2 hores.
- L'aplicatiu serà Open Source i estarà publicat a Github perquè qualsevol altra organització el pugui aprofitar.
- El Sistema ha de garantir la confidencialitat de les dades personals dels integrants de l'Equip de Treball.

## 5.4. Restriccions per part de l'empresa

Com que aquesta plataforma es desenvoluparà en una empresa, el desenvolupament haurà de complir unes restriccions per adaptar-se al marc de treball de l'equip d'IT de Som Energia. Aquestes restriccions són les següents:

- El Backend de l'aplicatiu haurà d'estar programat amb la versió 3 Python, utilitzant el framework Django i Django REST Framework.
- La gran part del codi haurà d'estar cobert per testos seguint la metodologia de unittest.
- El desenvolupament s'haurà de fer seguint la metodologia de programació Test-driven Development.
- El Frontend de l'aplicatiu haurà d'estar programat amb JavaScript, utilitzant el framework Mithril ja que és el que utilitza Som Energia en els seus desenvolupaments.
- L'aplicatiu s'haurà d'hostatjar en una màquina virtual.
- Durant el desenvolupament de l'aplicatiu s'utilitzarà el control de versions de Github.

## 5.5. Requeriments de Llicència.

Tota la documentació i el codi necessari pel desenvolupament d'aquest aplicatiu estarà sota la GNU Affero General Public License v3.0.

## 5.6. Matriu de dependències

Per tal de poder determinar la prioritat dels diferents requisits és necessària una matriu de dependències, la qual ens ajudarà a realitzar la planificació de les tasques. Per tal de minimitzar les dependències entre les diferents funcionalitats s'implementarà un MockServer, el qual ens permetrà desenvolupar el Backend i el Frontend simultàniament.



Totes les interfícies del Frontend es podran desenvolupar en paral·lel gràcies al MockServer, d'aquesta manera, un cop acabat el desenvolupament de interfícies i el servidor, serà necessari integrar el Backend amb el Frontend i les diferents interfícies del Frontend.

Tot i que els requisits RF-00.2 i RF 30.2 tinguin una dependència circular, en el moment del desenvolupament es podrà resoldre gràcies a la possibilitat de crear un usuari administrador des del shell de Django.

D'aquesta manera, i per tal de facilitar la comprensió de les dependències entre els diferents requeriments, s'ha representat en forma de taula on es pot comprovar per cada requisit, quines dependències té.

Dependències

	RF-001	RF-002	RF-011	RF-012	RF-021	RF-022	RF-031	RF-032	RF-041	RF-042	RF-051	RF-052	RF-061	RF-062	RF-071	RF-072	RF-081	RF-082	RF-091	RF-092	RF-101	RF-102	RF-111	RF-112	RF-121		
RF-001																											
RF-002																											
RF-011																											
RF-012																											
RF-021																											
RF-022																											
RF-031																											
RF-032																											
RF-041																											
RF-042																											
RF-051																											
RF-052																											
RF-061																											
RF-062																											
RF-071																											
RF-072																											
RF-081																											
RF-082																											
RF-091																											
RF-092																											
RF-101																											
RF-102																											
RF-111																											
RF-112																											
RF-121																											
RF-301																											
RF-302																											
RF-311																											
RF-312																											
RF-321																											
RF-322																											
RF-331																											
RF-332																											
RF-341																											
RF-342																											
RF-351																											
RF-352																											
RF-361																											
RF-362																											
RF-37																											
RF-381																											
RF-382																											
RF-391																											
RF-401																											
RF-402																											
RF-411																											
RF-412																											
RF-421																											
RF-422																											

Requisit

Matriu de dependències dels requisits

**Dependències**

	RF-12.2	RF-13.1	RF-13.2	RF-14.1	RF-14.2	RF-15.1	RF-15.2	RF-20	RF-30.1	RF-30.2	RF-31.1	RF-31.2	RF-32.1	RF-32.2	RF-33.1	RF-33.2	RF-34.1	RF-34.2	RF-35.1	RF-35.2	RF-36.1	RF-36.2	RF-37	RF-38.1	RF-38.2
RF-00.1																									
RF-00.2																									
RF-01.1																									
RF-01.2																									
RF-02.1																									
RF-02.2																									
RF-03.1																									
RF-03.2																									
RF-04.1																									
RF-04.2																									
RF-05.1																									
RF-05.2																									
RF-06.1																									
RF-06.2																									
RF-07.1																									
RF-07.2																									
RF-08.1																									
RF-08.2																									
RF-09.1																									
RF-09.2																									
RF-10.1																									
RF-10.2																									
RF-11.1																									
RF-11.2																									
RF-12.1																									
RF-12.2																									
RF-13.1																									
RF-13.2																									
RF-14.1																									
RF-14.2																									
RF-15.1																									
RF-15.2																									
RF-20																									
RF-30.1																									
RF-30.2																									
RF-31.1																									
RF-31.2																									
RF-32.1																									
RF-32.2																									
RF-33.1																									
RF-33.2																									
RF-34.1																									
RF-34.2																									
RF-35.1																									
RF-35.2																									
RF-36.1																									
RF-36.2																									
RF-37																									
RF-38.1																									
RF-38.2																									

**Requisit**

*Matriu de dependències dels requisits*

		Requisit									
		RF-39.1	RF-39.2	RF-40.1	RF-40.2	RF-41.1	RF-41.2	RF-42.1	RF-42.2		
Dependències	RF-00.1										
	RF-00.2										
	RF-01.1										
	RF-01.2										
	RF-02.1										
	RF-02.2										
	RF-03.1										
	RF-03.2										
	RF-04.1										
	RF-04.2										
	RF-05.1										
	RF-05.2										
	RF-06.1										
	RF-06.2										
	RF-07.1										
	RF-07.2										
	RF-08.1										
	RF-08.2										
	RF-09.1										
	RF-09.2										
	RF-10.1										
	RF-10.2										
	RF-11.1										
	RF-11.2										
RF-12.1											
RF-12.2											
RF-13.1											
RF-13.2											
RF-14.1											
RF-14.2											
RF-15.1											
RF-15.2											
RF-20											
RF-30.1											
RF-30.2											
RF-31.1											
RF-31.2											
RF-32.1											
RF-32.2											
RF-33.1											
RF-33.2											
RF-34.1											
RF-34.2											
RF-35.1											
RF-35.2											
RF-36.1											
RF-36.2											
RF-37											
RF-38.1											
RF-38.2											
RF-39.1											
RF-39.2											
RF-40.1											
RF-40.2											
RF-41.1											
RF-41.2											
RF-42.1											
RF-42.2											

Matriu de dependències dels requisits

En la taula, les caselles vermelles representen un requisit (eix de les x) amb el conjunt de dependències que té (eix de les y).

## 6. Planificació

Per realitzar la planificació del projecte, s'ha separat per paquets de treballs i s'ha especificat el període de temps es destinaria cada paquet. Per tal de visualitzar-ho correctament s'ha utilitzat un diagrama de Gantt.

### 6.1. Paquets de treball

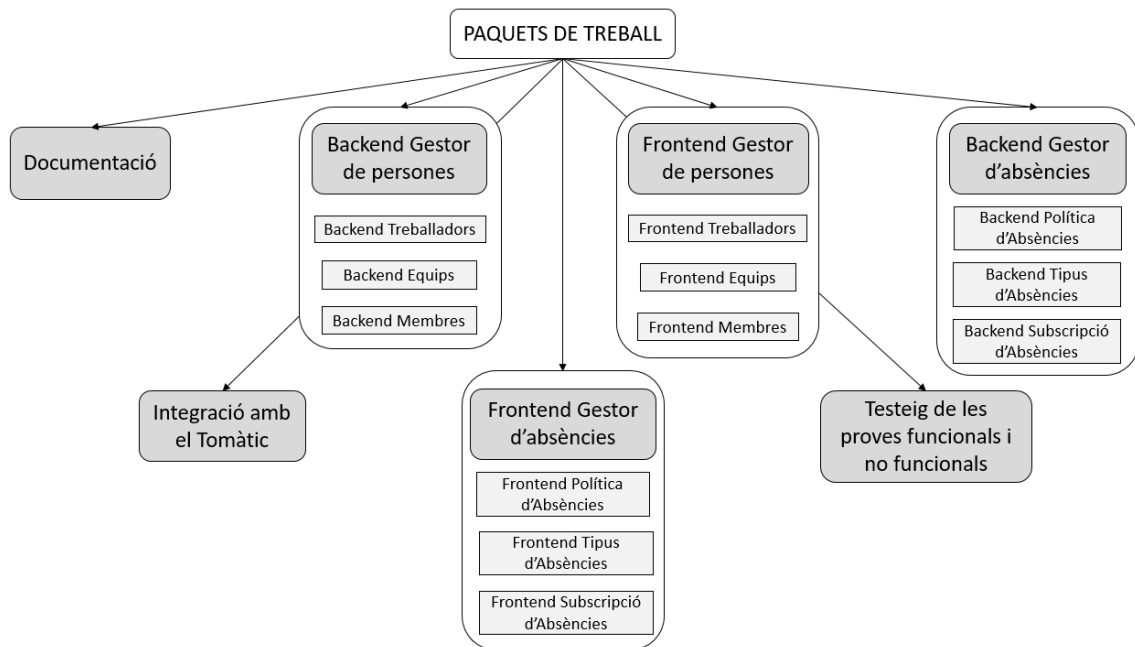
Per desglossar el desenvolupament de l'aplicatiu s'ha decidit agrupar la feina a realitzar en diferents paquets de treball. Cada paquet de treball estarà compost per un seguit de tasques a realitzar que tindran relació directa amb els requisits de l'aplicatiu.

Tot i que la manera més natural de puntuar el cost de les diferents tasques és en hores, a Som Energia utilitzem Mitjos dies (4 hores naturals) per calcular el cost de les tasques de manera més efectiva.

Encara que només es pugui veure la implementació en la descripció dels paquets de treball, aquests paquets també inclouen els testos que cobriran les tasques del paquet, ja que es programarà seguir la metodologia Test-driven Development.

Per tal de desgranar la implementació del projecte s'han separat 7 grans paquets, els quals s'han dividit en diferents subpaquets. Els paquets més generals són:

- Documentació
- Backend gestor de persones
- Frontend Gestor de persones
- Backend Gestor d'absències
- Frontend Gestor d'absències
- Integració amb el Tomàtic
- Testeig de les proves funcional i no funcionals



*Estructuració dels 7 grans paquets de treball i els petits que els componen*

### 6.1.1. Diccionari de paquets de treball

Per poder entendre millor l'abast dels diferents paquets de treball, s'explicaran segons la seva descripció, la mostra entregable al final de la realització del paquet i el temps esperat per completar-lo.

Els paquets que hi haurà per l'aplicatiu són:

#### **PAQUET 1: DOCUMENTACIÓ**

- **Documentació [Doc]**
  - **Descripció:** documentació de les tasques a realitzar, així com de la memòria del treball.
  - **Tasques:**
    - Redacció de la memòria
    - Anàlisi del projecte
  - **Lliurables:** Memòria del treball
  - **Temps:** 10 mitjos dies

## **PAQUET 2: BACKEND GESTOR DE PERSONES**

- **Backend Treballadors [BT]**

- **Descripció:** Anàlisi, disseny i implementació dels models de la Base de dades i la lògica necessària per poder veure, crear, modificar i eliminar Treballadors. També inclou la lògica d'inici de sessió.
- **Tasques:**
  - Aprenentatge i anàlisi dels components necessaris per a la realització del paquet de treball.
  - Anàlisi i disseny dels models necessaris per la representació del Treballador.
  - Anàlisi i disseny de la lògica per les diferents crides del Treballador.
  - Implementar el model de Treballador.
  - Implementar la lògica de la creació de Treballadors.
  - Implementar la lògica per la visualització de Treballadors.
  - Implementar la lògica per la modificació de Treballadors.
  - Implementar la lògica per l'eliminació de Treballadors
  - Implementar la lògica d'inici de sessió.
- **Lliurables:** Codi font totalment funcional i la documentació de les crides de l'API per utilitzar aquestes funcionalitats.
- **Temps:** 25 mitjos dies.

- **Backends Equips [BE]**

- **Descripció:** Anàlisi, disseny i implementació dels models de la Base de dades i la lògica necessària per poder veure, crear, modificar i eliminar Equips.
- **Tasques:**
  - Anàlisi i disseny dels models necessaris per la representació de l'Equip.
  - Anàlisi i disseny de la lògica per les diferents crides de l'Equip.
  - Implementar el model de Equip.
  - Implementar la lògica per la creació de Equips.
  - Implementar la lògica per la visualització de Equips.
  - Implementar la lògica per la modificació de Equips.
  - Implementar la lògica per l'eliminació de Equips.
- **Lliurables:** Codi font totalment funcional i documentació de les crides de l'API per utilitzar aquestes funcionalitats.
- **Temps:** 10 mitjos dies.

- **Backends Membres d'Equips [BM]**

- **Descripció:** Anàlisi, disseny i implementació dels models de la Base de dades i la lògica necessària per poder veure, crear, modificar i eliminar Membres (relació entre Treballadors i Equips).
- **Tasques:**
  - Anàlisi i disseny dels models necessaris per la representació dels Membres d'Equips.
  - Anàlisi i disseny de la lògica per les diferents crides dels Membres d'Equips.
  - Implementar el model de Membres d'Equips.
  - Implementar la lògica per la creació de Membres d'Equips.
  - Implementar la lògica per la visualització de Membres d'Equips.
  - Implementar la lògica per la modificació de Membres d'Equips
  - Implementar la lògica per l'eliminació de Membres d'Equips.
- **Lliurables:** Codi font totalment funcional i la documentació de les crides de l'API per utilitzar aquestes funcionalitats.
- **Temps:** 10 mitjos dies.

### **PAQUET 3: FRONTEND GESTOR DE PERSONES**

- **Frontend Treballadors [FT]**

- **Descripció:** Anàlisi, disseny i implementació de les interfícies necessàries per cobrir totes les funcionalitats de Treballadors del Backend.
- **Tasques:**
  - Aprenentatge i anàlisi dels components necessaris per a la realització del paquet de treball.
  - Anàlisi i disseny de les interfícies d'aquest paquet.
  - Implementar la interfície d'inici de sessió.
  - Implementar la interfície de visualització de perfil.
  - Implementar la interfície de creació de Treballadors.
  - Integrar el Frontend amb el Backend i les interfícies d'aquest paquet amb les altres interfícies.
- **Lliurables:** Codi font totalment funcional.
- **Temps:** 30 mitjos dies.



- **Frontend Equips [FE]**
  - **Descripció:** Anàlisi, disseny i implementació de les interfícies necessàries per cobrir totes les funcionalitats de Equips del Backend.
  - **Tasques:**
    - Anàlisi i disseny de les interfícies d'aquest paquet.
    - Implementar la interfície de visualització de Equips.
    - Implementar la interfície de modificació de Equips.
    - Implementar la interfície de creació de Equips.
    - Integar el Frontend amb el Backend i les interfícies d'aquest paquet amb les altres interfícies.
  - **Lliurables:** Codi font totalment funcional.
  - **Temps:** 20 mitjos dies.
- **Frontend Membres d'Equips [FM]**
  - **Descripció:** Anàlisi, disseny i implementació de les interfícies necessàries per cobrir totes les funcionalitats dels Membres d'Equips del Backend.
  - **Tasques:**
    - Anàlisi i disseny de les interfícies d'aquest paquet.
    - Implementar la interfície d'afegir un Treballador a un Equip.
    - Implementar la interfície d'eliminar la relació entre Treballadors i Equips.
    - Implementar la interfície per modificar la relació entre Treballadors i Equips.
    - Integar el Frontend amb el Backend i les interfícies d'aquest paquet amb les altres interfícies.
  - **Lliurables:** Codi font totalment funcional.
  - **Temps:** 10 mitjos dies.

#### **PAQUET 4: BACKEND GESTOR D'ABSÈNCIES**

- **Backend Política d'Absències [BPA]**
  - **Descripció:** Anàlisi, disseny i implementació dels models de la Base de dades i la lògica necessària per poder veure, crear, modificar i eliminar Polítiques d'Absència.
  - **Tasques:**
    - Anàlisi i disseny dels models necessaris per la representació de les Polítiques d'Absència.
    - Anàlisi i disseny de la lògica per les diferents crides de les Polítiques d'Absència.
    - Implementar el model de Polítiques d'Absència.

- Implementar la lògica per la creació de Polítiques d'Absència.
- Implementar la lògica per la visualització de Polítiques d'Absència.
- Implementar la lògica per la modificació de Polítiques d'Absència.
- Implementar la lògica per l'eliminació de Polítiques d'Absència.
- **Lliurables:** Codi font totalment funcional i la documentació de les crides de l'API per utilitzar aquestes funcionalitats.
- **Temps:** 10 mitjos dies.
- **Backend Tipus d'Absències [BTA]**
  - Descripció: Anàlisi, disseny i implementació dels models de la Base de dades i la lògica necessària per poder veure, crear, modificar i eliminar Tipus d'Absència.
  - Tasques:
    - Anàlisi i disseny dels models necessaris per la representació dels Tipus d'Absència.
    - Anàlisi i disseny de la lògica per les diferents crides dels Tipus d'Absència.
    - Implementar el model del Tipus d'Absència.
    - Implementar la lògica per la creació de Tipus d'Absència.
    - Implementar la lògica per la visualització de Tipus d'Absència.
    - Implementar la lògica per la modificació de Tipus d'Absència.
    - Implementar la lògica per l'eliminació de Tipus d'Absència.
  - **Lliurables:** Codi font totalment funcional i la documentació de les crides de l'API per utilitzar aquestes funcionalitats.
  - **Temps:** 20 mitjos dies.
- **Backend Subscripció d'Absències [BSA]**
  - **Descripció:** Anàlisi, disseny i implementació dels models de la Base de dades i la lògica necessària per poder veure, crear, modificar i eliminar Subscripcions d'Absències (relació entre Treballadors i Tipus d'Absències).
  - **Tasques:**
    - Anàlisi i disseny dels models necessaris per la representació de les Subscripcions d'Absències.
    - Anàlisi i disseny de la lògica per les diferents crides de les Subscripcions d'Absències.
    - Implementar el model de Subscripció d'Absències.
    - Implementar la lògica per la creació de les Subscripcions d'Absències.
    - Implementar la lògica per la visualització de les Subscripcions d'Absències.
    - Implementar la lògica per la modificació de les Subscripcions d'Absències.
    - Implementar la lògica per l'eliminació de les Subscripcions d'Absències.

- **Lliurables:** Codi font totalment funcional i la documentació de les crides de l'API per utilitzar aquestes funcionalitats.
- **Temps:** 40 mitjos dies.

## **PAQUET 5: FRONTEND GESTOR D'ABSÈNCIES**

### • **Frontend Política d'Absències [FPA]**

- **Descripció:** Anàlisi, disseny i implementació de les interfícies necessàries per cobrir totes les funcionalitats de Polítiques d'Absències del Backend.
- **Tasques:**
  - Anàlisi i disseny de les interfícies d'aquest paquet.
  - Implementar la interfície de visualització de Política d'Absència.
  - Implementar la interfície de modificació de Polítiques d'Absència.
  - Implementar la interfície de creació de Polítiques d'Absència.
  - Integrar el Frontend amb el Backend i les interfícies d'aquest paquet amb les altres interfícies.
- **Lliurables:** Codi font totalment funcional.
- **Temps:** 5 mitjos dies.

### • **Frontend Tipus d'Absències [FTA]**

- **Descripció:** Anàlisi, disseny i implementació de les interfícies necessàries per cobrir totes les funcionalitats de Tipus d'Absències del Backend.
- **Tasques:**
  - Anàlisi i disseny de les interfícies d'aquest paquet.
  - Implementar la interfície de visualització de Tipus d'Absències.
  - Implementar la interfície de modificació de Tipus d'Absències.
  - Implementar la interfície de creació de Tipus d'Absències.
  - Integrar el Frontend amb el Backend i les interfícies d'aquest paquet amb les altres interfícies.
- **Lliurables:** Codi font totalment funcional.
- **Temps:** 15 mitjos dies.

### • **Frontend Subscripció d'Absències [FSA]**

- **Descripció:** Anàlisi, disseny i implementació de les interfícies necessàries per cobrir totes les funcionalitats de Subscripció d'Absències del Backend.
- **Tasques:**
  - Anàlisi i disseny de les interfícies d'aquest paquet.

- Implementar la interfície de visualització de Subscripció d'Absències en forma de calendari.
  - Implementar la interfície de modificació de Subscripció d'Absències.
  - Implementar la interfície de creació de Subscripció d'Absències.
  - Integrar el Frontend amb el Backend i les interfícies d'aquest paquet amb les altres interfícies.
- **Lliurables:** Codi font totalment funcional
  - **Temps:** 30 mitjos dies.

### **PAQUET 6: INTEGRACIÓ AMB EL TOMÀTIC**

- **Integració amb el Tomàtic [ITO]**
  - **Descripció:** Anàlisi, disseny i implementar de les crides necessàries per poder satisfer les necessitats d'informació del Tomàtic.
  - **Tasques:**
    - Anàlisi i disseny de la visualització dels Treballadors i les seves Subscripcions d'Absències.
    - Implementació de la visualització dels Treballadors i les seves Subscripcions d'Absències.
  - **Lliurables:** Codi font totalment funcional i la documentació de les crides de l'API per utilitzar aquestes funcionalitats.
  - **Temps:** 10 mitjos dies.

### **PAQUET 7: TESTEIG DE LES PROVES FUNCIONALS I NO FUNCIONALS**

- **Testeig de les proves funcionals i no funcionals [Test]**
  - **Descripció:** Demostració del compliment dels requisits funcionals i no funcionals.
  - **Tasques:**
    - Anàlisi de la demostració dels requisits no funcionals.
    - Demostració dels requisits funcionals.
    - Demostració dels requisits no funcionals.
  - **Lliurables:** Informe de les proves a tots els requisits.
  - **Temps:** 3 mitjos dies.

## 6.2.Matriu de traçabilitat

La matriu de traçabilitat permetrà veure la relació directa entre els diferents requisits de l'aplicatiu i els seus mòduls a desenvolupar.

	B	B	B	F	F	F	B	B	BS	FP	F	FS	IT
<b>RF-00.1</b>				x									
<b>RF-00.2</b>	x												
<b>RF-01.1</b>				x									
<b>RF-01.2</b>	x												
<b>RF-02.1</b>				x									
<b>RF-02.2</b>	x												
<b>RF-03.1</b>												x	
<b>RF-03.2</b>									x				
<b>RF-04.1</b>												x	
<b>RF-04.2</b>									x				
<b>RF-05.1</b>												x	
<b>RF-05.2</b>									x				
<b>RF-06.1</b>												x	
<b>RF-06.2</b>									x				
<b>RF-07.1</b>											x		
<b>RF-07.2</b>								x					
<b>RF-08.1</b>											x		
<b>RF-08.2</b>								x					
<b>RF-09.1</b>											x		
<b>RF-09.2</b>								x					
<b>RF-10.1</b>												x	
<b>RF-10.2</b>								x					
<b>RF-11.1</b>						x							
<b>RF-11.2</b>			x										
<b>RF-12.1</b>						x							
<b>RF-12.2</b>			x										
<b>RF-13.1</b>				x									
<b>RF-13.2</b>	x												
<b>RF-14.1</b>				x									
<b>RF-14.2</b>	x												
<b>RF-15.1</b>												x	
<b>RF-15.2</b>									x				
<b>RF-20</b>													

<b>RF-30.1</b>				x									
<b>RF-30.2</b>	x												
<b>RF-31.1</b>				x									
<b>RF-31.2</b>	x												
<b>RF-32.1</b>				x									
<b>RF-32.2</b>	x												
<b>RF-33.1</b>					x								
<b>RF-33.2</b>		x											
<b>RF-34.1</b>					x								
<b>RF-34.2</b>		x											
<b>RF-35.1</b>					x								
<b>RF-35.2</b>		x											
<b>RF-36.1</b>					x								
<b>RF-36.2</b>		x											
<b>RF-37</b>		x											
<b>RF-38.1</b>												x	
<b>RF-38.2</b>								x					
<b>RF-39.1</b>									x				
<b>RF-39.2</b>							x						
<b>RF-40.1</b>										x			
<b>RF-40.2</b>							x						
<b>RF-41.1</b>										x			
<b>RF-41.2</b>							x						
<b>RF-42.1</b>												x	
<b>RF-42.2</b>									x				

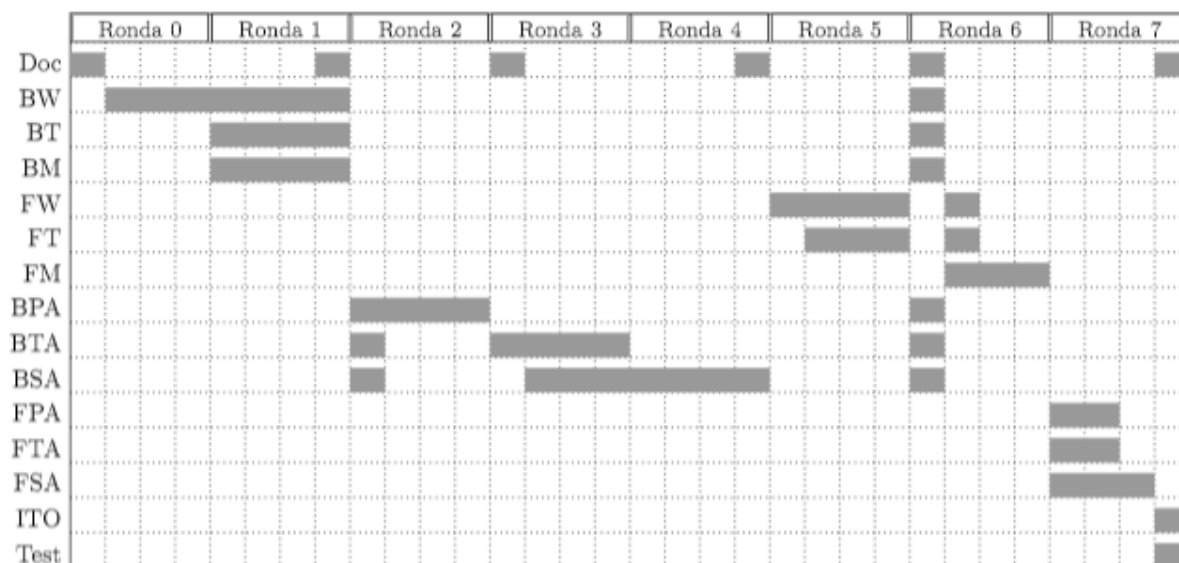
### 6.3. Planificació dels paquets de treball

Tot i que el temps previst per la implementació dels paquets de treball és massa elevat com per encabir-se en 4 mesos, s'ha planificat el projecte per tenir una primera versió per oferir al departament de recursos humans de Som Energia, d'aquesta manera el poden provar i fer suggeriments per anar millorant l'aplicatiu.

La primera ronda es començarà a principis de Febrer i l'última acabarà a finals de Maig. D'aquesta manera, durant les diferents rondes es realitzarà:

- **Ronda 0:**
  - En aquesta ronda es començarà a desenvolupar l'anàlisi i el disseny del projecte, més concretament l'API que el servidor haurà d'implementar. També es començarà la formació de Django, Django Rest Framework amb l'inici de la implementació del mòdul Backend de Treballadors.
- **Ronda 1:**
  - En aquesta ronda es seguirà amb la implementació del mòdul Backend de Treballadors i es començarà i s'acabaran els mòduls Backend de Equips i Membres d'Equips. Al final de la ronda es farà un anàlisi de com abordar els mòduls de Backend de Política d'Absències, Tipus d'Absències i Subscripció d'Absències.
- **Ronda 2:**
  - En aquesta ronda es començarà i s'acabarà el desenvolupament del mòdul de Backend de Política d'Absències i es començarà el del mòdul de Tipus d'Absències i Subscripció d'Absències.
- **Ronda 3:**
  - En aquesta ronda es seguirà amb la documentació i actualització de l'API i del codi realitzat en les anteriors rondes. També s'acabarà d'implementar el mòdul Backend de Tipus d'Absències i es seguirà amb la del mòdul de Subscripció d'Absències.
- **Ronda 4:**
  - En aquesta ronda s'acabarà la implementació del mòdul Backend de Subscripció d'Absències i s'analitzarà els requeriments d'interfície gràfica del Frontend fent així els mocks d'interfície que posteriorment s'implementaran.
- **Ronda 5:**
  - En aquesta ronda es farà la formació de les tecnologies amb les que es realitzarà part del Frontend dels mòduls Frontend de Treballadors i Equips.
- **Ronda 6:**
  - En aquesta ronda es començarà i s'acabarà la implementació del mòdul Frontend de Membres d'Equips i es documentarà tot el codi realitzat anteriorment. Serà també en aquesta ronda quan s'integrarà el Backend i el Frontend.
- **Ronda 7:**
  - Durant aquesta ronda s'implementaran els mòduls Frontend de Política d'Absències, Tipus d'Absències i la Subscripció a les Absències, al igual que es realitzaran els tests de rendiment de i s'acabarà la memòria.

Per tal de poder veure el temps de dedicació previst pels diferents paquets de treball s'ha realitzat el següent diagrama de Gantt:



Per tal de calcular el pressupost de la implementació de l'aplicatiu s'ha decidit utilitzar dos mètodes, el pressupost per punts de funció i el pressupost per paquets de treball.

En l'Annex 1 es detalla la planificació real que es va dur a terme i que difereix una mica d'aquesta planificació inicial.

### 6.3.1. Pressupost per Paquets de Treball

Per calcular el pressupost per paquets de treball cal que utilitzem la previsió d'hores de cada un dels paquets i fem els càlculs del cost segons el sou d'un programador junior.

Paquets de treball	Hores	Dies	Mesos	Salari persona/mes	Cost
<b>BT</b>	100	20	1	1500	1500
<b>BE</b>	40	8	0.4	1500	600
<b>BM</b>	40	8	0.4	1500	600
<b>FT</b>	120	24	1.2	1500	1800
<b>FE</b>	80	16	0.8	1500	1200
<b>FM</b>	40	8	0.4	1500	600
<b>BPA</b>	40	8	0.4	1500	600
<b>BTA</b>	80	16	0.8	1500	1200



<b>BSA</b>	160	32	1.6	1500	2400
<b>FPA</b>	20	4	0.2	1500	300
<b>FTA</b>	40	8	0.4	1500	600
<b>FSA</b>	120	24	1.2	1500	1800
<b>ITO</b>	40	8	0.4	1500	600
<b>TOTAL = 13800</b>					

Per tal de calcular el cost del desenvolupament de la plataforma s'ha transformat mitjos mitjos dies en hores i un cop sabem les hores només caldrà saber el cost de contractació del desenvolupador, que en aquest cas, al tractar-se d'un desenvolupador junior sense experiència laboral s'ha estipulat amb 1500 euros mensuals.

Cal tenir en compte que per fer els càlculs s'ha considerat que de cada 8 hores de jornada, només 5 són productives.

Aquest és un pressupost aproximat realitzat abans de començar a implementar el Gestor d'Absències i des de la cooperativa s'ha cregut que el cost és acceptable.



## 7. Estudis i decisions

En aquest apartat es parlarà sobre les diferents decisions que s'han pres abans de començar el desenvolupament i durant aquest. Concretament es parlarà sobre els aspectes dels frameworks utilitzats i la metodologia utilitzada durant el desenvolupament. Es parlarà també sobre els aspectes decidits per l'empresa, els motius de l'elecció i el que pot aportar al propi desenvolupament o al projecte.

### 7.1. Decisions generals sobre el projecte

En aquest subapartat es tractaran les decisions sobre l'estructura del projecte i de les metodologies que s'han utilitzat durant el desenvolupament.

#### 7.1.1. La possibilitat d'utilitzar els Templates de Django o una API REST amb Django REST Framework

Abans d'iniciar el projecte es van plantejar dues possibles estructures per l'aplicatiu:

- Només fer el Backend amb Django i poder-lo visualitzar mitjançant els Templates.
- Desenvolupar un Backend i un Frontend per separats intercanviat informació entre ells mitjançant una API.

La primera opció és la més senzilla de desenvolupar ja que només cal implementar les plantilles per renderitzar el contingut dels models, en canvi, per la segona opció cal dissenyar una API i tota la interfície web, el que donaria el doble de feina.

Tot i així, el fet de desenvolupar el Backend i el Frontend per separat aportaria molta més flexibilitat a l'aplicatiu i la possibilitat d'integrar altres funcionalitats, mòduls o comunicacions amb altres Sistemes Informàtics de la cooperativa.

El fet de desenvolupar-ho separat també aportaria molta més precisió alhora de fer els tests, ja que es podrien testear cada una de les crides per un millor control del Backend. Amb la segona opció també s'aconseguirien mòduls molt més aprofitables per altres processos de la cooperativa i, en cas de canviar els requisits del projecte, es podria modificar els diferents components de manera més fàcil que si estigués tot integrat amb un únic procés.

Per tots aquest motius s'ha decidit estructurar el projecte amb un Backend i un Frontend separats i comunicats mitjançant una API.

Pel que fa al disseny de l'API, el departament d'IT actualment només treballa amb serveis REST i per això aquest projecte també es dissenyarà una API REST. D'aquesta manera, ens podrem beneficiar de moltes característiques de REST com:

- Permet dissenyar una API amb moltes funcionalitats i amb un nombre d'endpoints molt reduït.
- Permet escalar molt bé en cas que s'augmentés molt la càrrega.
- No consumeix pràcticament recursos del servidor.

Tot i que Django no permet atendre peticions REST, existeix Django REST Framework, una llibreria de Django que en ho permetrà fer. S'ha escollit aquesta llibreria per ser la principal que té Django per aquesta classe de propòsits, aquesta llibreria, a més, ens permetrà aprofitar tots els avantatges d'una API REST.

### 7.1.2. Reaprofitament d'eines o dades de l'empresa

Des dels inicis de la cooperativa s'han utilitzat sistemes informàtics per obtenir, guardar i mostrar les dades que la cooperativa necessita per realitzar les seves activitats comercials i socials. Degut a això, actualment posseeix dades en diferents plataformes i bases de dades, cosa que es vol analitzar per la seva unificació.

En referència a les credencials dels usuaris de la plataforma, es plantejaven dues possibilitats:

- La utilització dels usuaris existents a l'ERP de la cooperativa.
- No utilitzar cap informació d'altres bases de dades i crear usuaris només per aquesta plataforma.

Un dels principals motius pel que s'haurien de crear usuaris propis del Gestor d'Absències és el fet que no tots els treballadors de la cooperativa tenen accés a l'ERP, cosa que ens obligaria a donar-los d'alta en la base de dades de l'ERP i fer modificacions de tots els permisos d'aquests usuaris per tal que no poguessin dur a terme cap acció. Tot i així, ens trobaríem amb el problema de que els usuaris creats a l'ERP tenen uns permisos mínims que no es poden treure, com per exemple el de visualitzacions d'alguns menús, fent així que si realment volguéssim restringir l'accés total de l'usuari a l'ERP, haguéssim de modificar el codi d'aquest, aspecte que comportaria un augment considerable del volum de feina. D'aquesta manera, l'opció quedaria descartada per la complexitat que podria arribar a comportar.

Per altra banda, l'opció de crear els usuaris propis pel Gestor d'Absències podria semblar que comporta massa feina, però Django ja té el model d'usuari implementat, pel que no caldria que ens preocupéssim per res.

D'aquesta manera, degut a la gran versatilitat que ens aporta una API REST elaborada amb Django i la possibilitat de poder aprofitar el mòdul d'usuaris per altres aplicacions, provoca que aquest disseny sigui molt útil per la unificació d'usuaris.

### 7.1.3. La utilització de Github

Tot i que en un primer moment aquest projecte es va pensar per ser desenvolupat per una única persona des d'un únic ordinador, tenir un control de versions en remot és una característica clau en qualsevol projecte que abordi la cooperativa.

Github ens permetrà satisfer el requisit referent a que el nostre Gestor d'Absències sigui codi lliure, facilitant també la possibilitat de tornar a versions anteriors del codi en cas de cometre algun error en el desenvolupament.

També s'ha decidit treballar amb branques i utilitzant funcionalitats d'aquest control de versions com les Pull Request per obtenir més experiència com a programador i poder desenvolupar diferents requeriments alhora en diferents branques.

### 7.1.4. Motius de la utilització de Docker

S'ha decidit dockeritzar el Backend de l'aplicació per la necessitat d'aprendre una tecnologia molt necessària en el departament d'IT i per tots els avantatges que aporta al desenvolupament del projecte; ja que:

- Permetrà que, durant el desenvolupament del Backend, tots els processos necessaris per la seva execució estiguin aïllats entre sí, així com de la resta de processos de l'ordinador.
- Permetrà a la comunitat provar el codi de manera més senzilla, ja que pel seu funcionament només caldrà clonar el repositori i aixecar els contenidors de Docker.

## 7.2. Decisions sobre el desenvolupament del Backend

En aquest subapartat es tracten decisions concretes preses sobre el desenvolupament de la part Backend de l'aplicació com els frameworks o les llibreries utilitzades.

### 7.2.1. La utilització de Django

Actualment, a la cooperativa, s'utilitzen tres frameworks diferents per desenvolupar servidors: Sanic, Flask i Django. D'aquests, el més utilitzat per la cooperativa és Flask per la seva simplicitat alhora de programar, tanmateix, el departament d'IT està intentant adquirir formació de Django per començar projectes més complexos. Pel desenvolupament d'aquest projecte s'ha preferit utilitzar Django per poder aprendre més sobre aquest framework i agafar experiència per poder seguir amb el desenvolupament d'altres plataformes que té amb el mateix framework. Així mateix, Django permetrà estructurar el codi molt millor i obtenir temps de resposta molt més baixos per part de l'aplicació.

Cal destacar que Django és el millor framework de Python pel desenvolupament d'aplicacions web complexes ja que aporta:

- Una relació directa a la base de dades gràcies als models que aporta.
- Una possibilitat de dividir una aplicació web en diferent mòduls aïllats de manera molt senzilla.
- Moltes llibreries i paquets que aporten milers de funcionalitats, aspecte que aporta versatilitat.
- Capes de seguretat per protegir la nostra base de dades i el nostre servei.
- Molta feina feta per la comunitat i es pot aprofitar sense cap complicació, com per exemple els usuaris, les autenticacions dels usuaris i un client web per administrar els models de la plataforma.

### 7.2.2. La no utilització de la llibreria Django-swingtime

Django-swingtime és una llibreria de Django que afegeix els models necessaris per definir tipus d'esdeveniments i que els usuaris puguin registrar ocurrències d'aquests en el calendari. Al tractar-se d'una de les utilitats que havia de tenir el gestor d'Absències, es va estudiar per tal d'integrar-ho en el gestor.

Aquesta llibreria facilita la creació d'un horari, amb la possibilitat de definir l'hora del dia que s'inicia l'horari, l'hora de finalització i la unitat de temps amb la que es poden afegir les tasques, per exemple, una hora.

També permet assignar documents a les ocurrències d'esdeveniment, aspecte que es pot aprofitar en cas d'absències que requereixin una justificació.

Tot i que durant unes setmanes va estar integrat en el Backend, va acabar donant problemes d'integració amb models propis del gestor pel que va haver de suprimir-se.

### 7.2.3. Autenticació a l'API amb la llibreria Django-rest-framework-simplejwt

El fet de decidir dissenyar una API per comunicar el servidor i el client web, que es tracti de peticions amb dades de caràcter personal i sigui necessari la conformitat de l'usuari implicat per realitzar accions en nom seu, com per exemple demanar vacances, feia indispensable dissenyar l'API amb autenticació.

Django-REST-Framework ofereix una llibreria per generar tokens pels usuaris que incorpora un token després de la creació d'un usuari que només es pot canviar si s'implementa la crida manualment.

Aprofitant que la manera de generar el token no semblava del tot correcta es va investigar llibreries de tercers amb compatibilitat amb Django REST Framework. Amb aquesta cerca es van trobar dos llibreries amb molta continuïtat en la comunitat:

- **Django-oauth-toolkit:** Tot i que està molt ben documentada i és molt complerta, aquesta llibreria es va descartar per tractar-se d'un mètode d'autenticació mitjançant OAuth, fet que entra en conflicte amb el principis del departament d'IT. L'OAuth és un estàndard amb el qual els usuaris es poden autenticar utilitzant les mateixes credencials que tenen en plataformes conegudes com Google o Facebook.
- **Django-rest-framework-simplejwt:** Aquesta llibreria es basa en la creació de tokens seguint l'estàndard JSON Web Token. La generació d'aquest tokens es pot configurar modificant el temps d'expiració, canviant l'algoritme de hash que els genera entre altres possibilitats. Al tractar-se de tokens més segurs, amb més capacitat de configuració i degut a la facilitat d'integració que té amb Django REST Framework s'ha decidit utilitzar aquesta llibreria.

### 7.2.4. Llenguatge utilitzat per la documentació de l'API

Com és necessari en tota API, a mesura que es va analitzar els requisits i el disseny d'aquesta, es va generar la documentació per poder veure les crides que s'havien d'implementar, els paràmetres que rebrien o els que retornarien.

Actualment, els dos estàndards més coneguts són Blueprint, que destaca per la simple sintaxi en format markdown i OpenApi, basat en la sintaxi en yaml.

```
## Question [/questions/{question_id}]
A question resource.

### View a Questions Detail **test** [GET]

+ Response 200 (application/json)

{
  "question": "Favourite programming language?",
  "published_at": "2014-11-11T08:40:51.620Z",
  "url": "/questions/1",
  "choices": [
    {
      "choice": "Swift",
      "url": "/questions/1/choices/1",
      "votes": 2048
    }, {
      "choice": "Python",
      "url": "/questions/1/choices/2",
      "votes": 1024
    }
  ]
}
```

*Exemple de sintaxi de Blueprint*

```
paths:
  /hello:
    get:
      description: Returns 'Hello' to the caller
      parameters:
        - name: name
          in: query
          description: The name of the person to whom to say hello
          required: false
          type: string
      responses:
        200:
          description: OK
```

*Exemple de sintaxi d'OpenApi*

Tot i que durant la carrera només s'ha utilitzat Blueprint, per documentar l'API del Gestor d'Absències s'ha decidit utilitzar OpenApi pels següent motius:

- El repositori d'OpenApi és molt més actiu que el de Blueprint, com demostra l'últim commit de Blueprint que és data del 2017, mentre que els commits d'OpenApi són setmanals. També cal destacar que OpenApi té moltes més estrelles, Pull Requests i Issues que Blueprint. Tot això indica que Blueprint és un estàndard que està congelat i no evolucionarà o que possiblement desapareixerà.
- Moltes empreses destacades en el món del desenvolupament són membres de la iniciativa d'OpenApi, alguns exemples són Google, Atlassian, IBM, Microsoft o Apiary que va ser el creador i precursor de l'estàndard Blueprint. Aquest fet, juntament amb la poca actualització de Blueprint aporta indicis sobre la possible desaparició d'aquest estàndard.



Tot i que actualment la versió més utilitzada és la 2.0 d'OpenApi, per documentar l'API s'utilitzarà l'última versió, la 3.0 per la seva novetat, preparant així la documentació per no haver-se de migrar en molt de temps.

#### 7.2.5. PostgreSQL

El motiu pel qual s'utilitzarà aquest sistema gestor de bases de dades és que Som Energia només treballa amb aquesta tecnologia per les bases de dades relacionals. Tot i que en la cooperativa també s'utilitzin Mongo, aquesta tecnologia està més reservada per fitxers o bases de dades amb un gran volum de dades.

Tot i que la decisió hagi vingut donada des de la cooperativa, utilitzar el sistema gestor de bases de dades PostgreSQL ens permetrà aprofitar totes les seves avantatges com ara la facilitat d'aprenentatge, ja que té una sintaxi molt similar a la resta de bases de dades SQL, la facilitat amb la que escala o la quantitat de connexions simultànies que pot atendre. Cal destacar que aquestes dues últimes característiques probablement no marcaran una diferència degut a que el volum de dades i usuaris que gestionarà la plataforma no serà molt elevat.

### 7.3. Decisions sobre el desenvolupament del Frontend

En aquest subapartat es poden observar diferents decisions presents en l'elaboració del Frontend, en concret de les tecnologies que han marcat el seu desenvolupament.

#### 7.3.1. La utilització de Mithril

Actualment, al departament d'IT de Som Energia s'està realitzant un procés de migració de projectes que utilitzen el framework d'Angular per Mithril, d'aquesta manera, aquest desenvolupament s'haurà de dur a terme amb la nova tecnologia que utilitza el departament. Gràcies a la utilització de Mithril, el projecte es podrà beneficiar de les característiques anteriorment esmentades en el marc de treball i conceptes previs.

Degut a que la migració de framework encara no ha acabat, cosa que provoca que el codi estigui en constant canvi, s'ha decidit no reaprofitar elements del codi o components genèrics, pel que es dissenyaran components molt senzills per aquesta aplicació.



## 8. Anàlisi i disseny del sistema

Aquest apartat té com a objectiu analitzar els requeriments de l'aplicació per poder dissenyar l'estructura general dels diferents processos i obtenir la informació que serà necessària guardar.

### 8.1. Anàlisis del sistema

Amb l'anàlisi del sistema d'informació es pretén estudiar de manera detallada les seves necessitats. D'aquesta manera, s'analitzaran els aspectes que requereix el sistema en relació als usuaris que tindrà i a les entitats d'aquest, com els equips, les polítiques de vacances, la tipologia d'absències i les ocurrencies d'aquestes.

#### 8.1.1. Anàlisi dels mòduls de gestió d'usuaris

De l'anàlisi dels requisits funcionals extraïem que un usuari haurà de tenir un cert nombre de dies de vacances cada any i un nombre de dies de vacances restants, fent així que els dies restants de vacances augmenti al principi de cada any.

Per petició del departament de recursos humans, cal poder saber la categoria laboral dels treballadors, el sexe d'aquests i la data que van començar a treballar a la cooperativa.

Cal resoldre també la problemàtica referent als permisos que pot tenir cada usuari dins de l'aplicació, així com tenir un control dels usuaris tenint coneixement sobre qui els ha creat, qui els ha modificat i quan ho ha fet. D'aquesta manera, en cas de tenir incidències en la plataforma es podrà investigar l'origen amb més facilitat.

No tothom hauria de tenir accés per editar, crear o suprimir les entitats de la plataforma, sinó que només els administradors de la plataforma ho hauria de poder fer. En canvi, tothom hauria de poder visualitzar als altres usuaris i les seves característiques.

En quan als equips, també ens caldrà saber qui i quan l'ha creat i qui i quan l'ha modificat per últim cop per poder resoldre problemes o incidències i tenir un major control de les dades que es gestionaran en aquesta plataforma.

Dins de cada equip a Som Energia hi ha dues persones amb un rol molt ben definit, el representant IT i el referent. El representant IT té la funció de recollir idees perquè el seu departament les pugui desenvolupar, mentre que el referent és la persona que gestiona les comunicacions amb la resta de la cooperativa, revisa les tasques a llarg termini i fa altres feines de gestió de l'equip.

Tot i que aquests dos rols no puguin influir en les absències dels treballadors, al visualitzar les dades d'un equip, es podria veure per tal de tenir-ho documentat.

### 8.1.2. Anàlisi dels mòduls de gestió absències

De l'anàlisi dels requisits funcionals, s'ha pogut concloure que existeixen moltes tipologies d'absències, existeixen absències que no resten dies de vacances, com seria el cas de les baixes laborals, absències que resten dies de vacances i, a Som Energia, existeixen també un tipus d'absències que, en cas de caure en cap de setmana, augmentarien les nostres vacances, aquestes són activitats que fa la cooperativa i que els seus treballadors poden assistir com organitzadors.

Tot i que, a Som Energia, qualsevol treballador pot agafar qualsevol tipus d'absència, l'administrador de la plataforma ha de poder decidir quin tipus d'absència requereix notificar als propis administradors quan un treballador se l'assigna. D'aquesta manera l'administrador podrà tenir un major control sobre certs tipus d'absències sense que els treballadors perdin la llibertat d'autogestionar-se. Aquest fet, demanat per departament de recursos humans, entra en conflicte amb alguns requisits funcionals, els quals s'han desestimat.

Al igual que la majoria d'entitats de la plataforma, els tipus d'absències i les ocurrències d'aquestes necessitaran un sistema per guarda la informació referent a les modificacions (qui i quan ho ha modificat) així com referent a la seva creació (qui i quan ho ha creat).

Tot i que no s'ha plasmat en un requisit funcional, des del departament de recursos humans es considera necessari extreure estadístiques sobre els treballadors i el tipus d'absències més freqüents, és per això que es desenvoluparà una entitat entre els treballadors i els tipus d'absències en la qual es pugui implementar aquests requeriments posteriorment.

Tot i que ens als requisits funcionals es va plantejar que el Gestor d'Absències havia de tenir un mòdul d'integració amb el Tomàtic, un cop analitzat el sistema he pogut observar que en lloc de dissenyar tot un mòdul perquè la plataforma comunicui les ocurrències per cada dia de la setmana, es podrà aprofitar un únic endpoint perquè el client web obtingui les ocurrències i perquè els altres Sistemes d'Informació de Som Energia també ho facin.

Encara que actualment a la cooperativa només existeixi una política de vacances, no es descarta que en un futur pugui haver-n'hi més, pel que la nova plataforma hauria de permetre canviar fàcilment de política d'absències als seus treballadors.

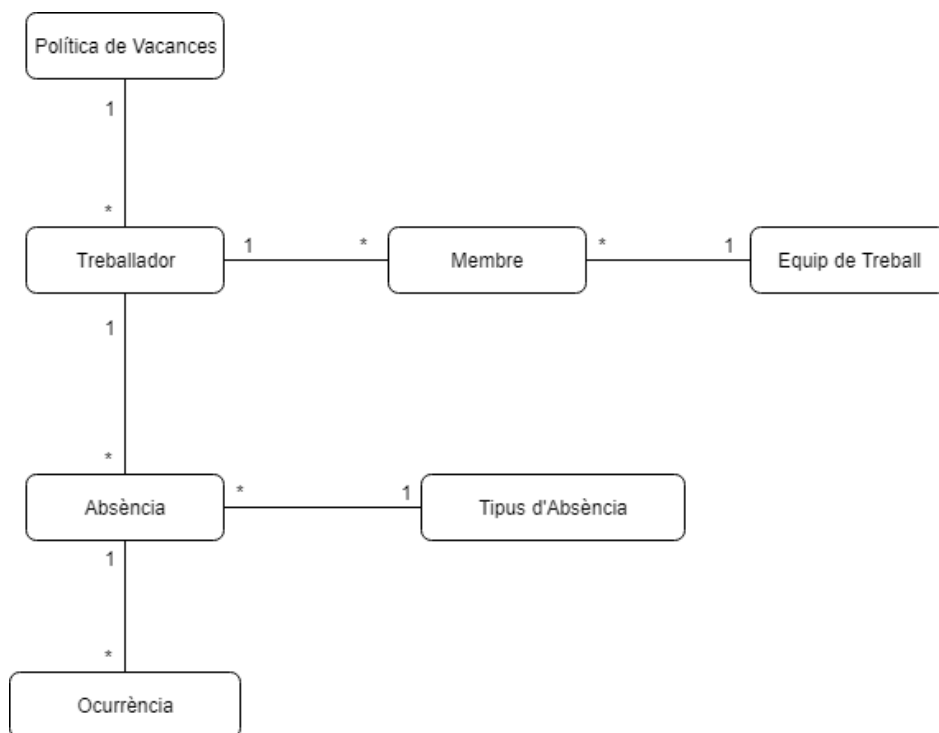
## 8.2. Disseny del sistema

Analitzades les necessitats del sistema informàtic que es vol crear, cal trobar la manera d'implementar l'aplicació que les satisfaci. El disseny del sistema té com a objectiu trobar les solucions que ens ho permetin fer. A l'hora de parlar del disseny del sistema, separarem entre el disseny de les dades, els processos que conformaran l'aplicació i certs casos complexos.

### 8.2.1. Disseny del model de Dades

Al tractar-se d'una aplicació en que totes les dades es gestionaran mitjançant l'ORM de Django i els seus models, en un primer moment es dissenyarà el sistema segons els models.

D'aquesta manera, el diagrama que s'utilitzarà pel desenvolupament del Gestor d'Absències serà el següent:



*Diagrama dels models de l'ORM*

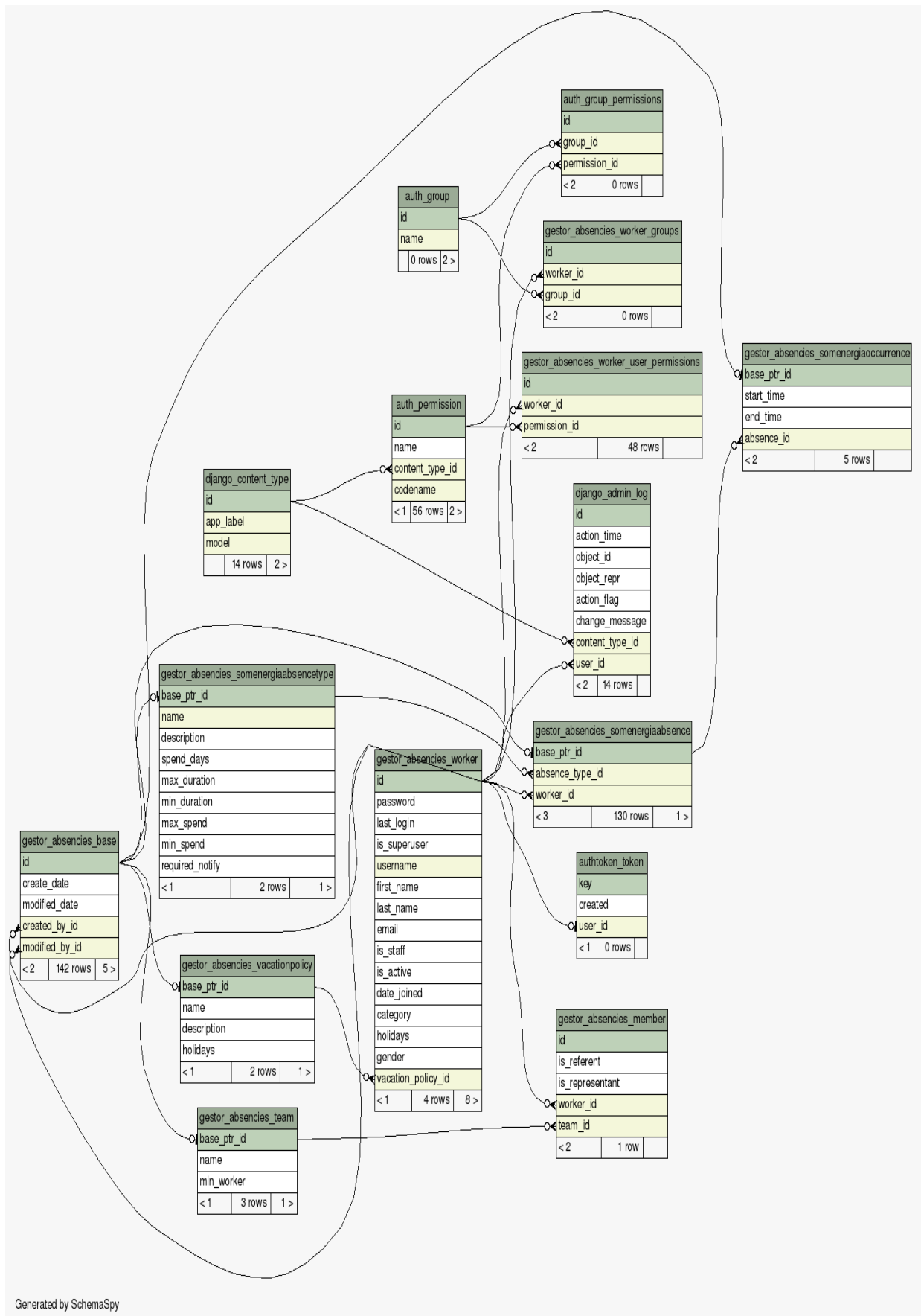
Aquest diagrama del model només ens ensenya les dependències que tindran les diferents entitats del sistema entre sí. Els models dels que s'hereten atributs, com la data de creació o l'usuari que ho ha fet, no s'han posat en aquest diagrama per poder-lo representar de manera més senzilla.

Com es pot observar al diagrama, s'ha creat una entitat pròpia per la política de vacances ja que en un futur immediat és molt possible que existeixin més d'una i abstreure-ho quan el projecte ja

estigui implantat en la cooperativa seria complicat. D'aquesta manera, pel moment, s'ha creat l'entitat Política de Vacances en la qual només hi ha el nom, la descripció, el número de dies de vacances a l'any que aporta al treballador i les dades de control necessàries per tenir un seguiment dels canvis que es realitzen en l'entitat.

En el cas de les absències succeeix d'igual forma, tot i que per la primera versió del Gestor d'Absències no és necessari poder elaborar informes de les absències més demanades i de les que ha realitzat cada treballador, s'ha considerat convenient deixar preparada l'entitat Absències que podrà fer aquestes tasques de manera molt senzilla, així doncs, amb petits ajustament en el disseny del models ens estalviarem molta feina i problemes per desenvolupaments futurs.

A l'hora de representar aquest model amb taules de la base de dades obtindrem el següent disseny:



*Disseny obtingut de les taules de la base de dades*

Degut a que el codi de la plataforma estarà escrit principalment en anglès els noms de les taules s'han traduït seguin l'exemple de com es farà en el codi per tal de representar amb major veracitat com serà el disseny de la base de dades.

És important distingir les pròpies taules que utilitza Django per funcionar o les que es generaran de la representació dels models del Gestor d'Absències. Les taules que Django requereix per funcionar o que gestiona internament són:

<b>Django_content_type</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>App_label</b>	Varchar2
<b>Model</b>	Varchar2

Aquest taula permetrà que Django instal·li diferents models en diversos mòduls al Backend. Per cada mòdul instal·lat, Django tindrà una fila en aquesta taula.

<b>Auth_group</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Name</b>	Varchar2

Aquesta taula serveix per emmagatzemar els diferents grups d'usuaris. En el nucli de Django, existeixen alguns mòduls de gestor d'usuaris els quals faciliten la base d'usuari, grups d'usuaris i permisos, tant pels usuaris com pels grups d'usuaris. Tot i que s'utilitzarà la base de gestió d'usuaris sobreescrivint la classe amb atributs propis, la gestió de grups d'usuaris no s'utilitzarà ja que en el Gestor d'Absències només hi ha dos tipus d'usuaris, els Treballadors i els Administradors representats en Django amb un booleà a la pròpia taula de l'usuari i sense gestionar-se com a cap grup d'usuaris.



<b>Auth_group_permissions</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Group_id</b>	Foreign Key
<b>Permission_id</b>	Foreign Key

Tot i que la podríem utilitzar, aquesta taula servirà per mapejar els permisos que tenen els diferents grups de l'aplicatiu. Aquesta taula ve donada pels mòduls d'usuaris del core de Django i no s'utilitzarà de la mateixa manera que tampoc s'han utilitzat els grups d'usuaris.

<b>Gestor_absencies_worker_groups</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Worker_id</b>	Foreign Key
<b>Group_id</b>	Foreign Key

Aquesta taula ens permetrà atribuir usuaris a diferents grups d'usuaris. Igual que succeïa amb les anteriors taules de gestió de grups d'usuaris, no s'utilitzarà per aquest desenvolupament.

<b>Auth_permission</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Name</b>	Varchar2
<b>Content_type_id</b>	Foreign Key
<b>Codename</b>	Varchar2

Aquesta taula, també proporcionada per Django, guardarà tots els permisos que es poden tenir sobre les diferents entitats de la plataforma, un exemple és que per l'entitat Treballador, hi ha el permís de treballador\_view, treballador\_add, treballador\_change i treballador\_remove, permeten així el CRUD típic de qualsevol aplicació. També cal destacar que es poden definir nous tipus de permisos en cas que fos necessari.

<b>Gestor_absencies_worker_user_permissions</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Worker_id</b>	Foreign Key
<b>Permission_id</b>	Foreign Key

Aquesta taula ens permetrà veure l'assignació de permisos atribuïts a cada treballador. Tot i que amb el mòdul d'usuaris bàsic que ens ofereix Django ja vinguin definits els usuaris, en el Gestor d'Absències es substituirà per un propi anomenat Treballador, motiu pel qual apareix la paraula Worker en el nom de les taules que fan referència als usuaris de la plataforma.

<b>Django_admin_log</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Action_time</b>	Timestamp
<b>Object_id</b>	Numeric
<b>Object_repr</b>	Varchar2
<b>Action_flag</b>	Varchar2
<b>Change_message</b>	Varchar2
<b>Content_type_id</b>	Foreign Key
<b>User_id</b>	Foreign Key

Aquesta taula és proporcionada pel propi Django i ens permetrà emmagatzemar les diferents accions que es realitzen des de l'admin de Django. A més, podran tenir coneixement sobre quines accions s'han realitzat, en quin moment i quin és l'objecte afectat. Pel que fa a les taules que són representacions directes dels mòduls del Gestor d'Absències o de les seves importacions són:

<b>Gestor_absencies_somenergiaoccurrence</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Base_ptr_id</b>	Foreign Key
<b>Start_time</b>	Timestamp
<b>End_time</b>	Timestamp
<b>Absence_id</b>	Foreign Key

Aquesta taula representa un període de temps en el que es manifesta una absència laboral, d'aquesta manera, cada ocurrència tindrà el seu moment d'inici i de finalització, així com el tipus d'absència de que es tracta.

<b>Gestor_absencies_somenergiaabsencetype</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Base_ptr_id</b>	Foreign Key
<b>Name</b>	Varchar2
<b>Description</b>	Varchar2
<b>Spend_days</b>	Numeric
<b>Max_duration</b>	Numeric
<b>Min_duration</b>	Numeric
<b>Max_spend</b>	Numeric
<b>Min_spend</b>	Numeric
<b>Required_notify</b>	Boolean

Aquesta taula permetrà guardar tots els tipus d'absències que presentarà el Gestor d'Absències, com la informació bàsica formada pel nom i la descripció o bé la duració en dies que podrà arribar a tenir, així com la computació de dies de vacances que tindrà.

<b>Gestor_absencies_base</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Create_date</b>	Timestamp
<b>Modified_date</b>	Timestamp
<b>Created_by_id</b>	Foreign Key
<b>Modified_by_id</b>	Foreign Key

Aquesta taula representa l'entitat de control bàsica de l'aplicació, permeten saber, en tot moment, diferents aspectes de cada entitat del nostre Backend. Aquests aspectes són la data de creació de l'entitat, la data de modificació i l'usuari que ho ha fet per tal de resoldre possibles incidències en un futur.

<b>Gestor_absencies_worker</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Password</b>	Varchar2
<b>Last_login</b>	Timestamp
<b>Is_superuser</b>	Boolean
<b>Username</b>	Varchar2
<b>First_name</b>	Varchar2
<b>Last_name</b>	Varchar2
<b>Email</b>	Varchar2
<b>Is_staff</b>	Boolean
<b>Is_active</b>	Boolean
<b>Date_joined</b>	Timestamp
<b>Category</b>	Varchar2
<b>Holidays</b>	Numeric
<b>Vacation_policy_id</b>	Foreign_key

Aquest taula possibilita l'emmagatzematge de la informació de cada un dels usuaris de la plataforma ja que, al haver sobreescrit l'usuari per defecte que Django ens proporciona, el Treballador s'ha convertit en l'entitat d'usuari, heretant així atributs com l'email, el first\_name o l'is\_staff que venen per defecte amb el model d'usuaris de Django. Cal destacar que el camp password es troba encriptat seguint un sistema SHA segons la clau secreta marcada en la configuració del projecte de Django.

<b>Gestor_absencies_somenergiaabsence</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Base_ptr_id</b>	Foreign Key
<b>Absence_type_id</b>	Foreign Key
<b>Worker_id</b>	Foreign Key

Aquesta taula ens permetrà emmagatzemar la relació que té cada treballador amb cada un dels tipus d'absències.

<b>Authtoken_token</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Key</b>	Primary Key
<b>Created</b>	Timestamp
<b>User_id</b>	Foreign Key

Aquesta taula, obtinguda per defecte amb el mòdul de gestió d'usuaris, possibilita guardar tokens a cada usuari i guardar el moment en que s'ha creat per, en cas de polítiques de caducitat del token, aquest es suprimeixi.

<b>Gestor_absencies_vacationpolicy</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Base_ptr_id</b>	Foreign Key
<b>Name</b>	Varchar2
<b>Description</b>	Varchar2
<b>Holidays</b>	Numeric

Aquesta taula ens permetrà emmagatzemar la informació necessària de les polítiques de vacances de Som Energia. Per aquest motiu han de contenir el nom i la descripció, juntament amb el nombre de dies per any que concedeix als treballadors que la tinguin. Cal destacar que, al igual que la majoria de taules generades dels models propis del Gestor d'Absències, contindrà una clau forana a la informació de control bàsica per poder tenir un bon seguiment de les entitats en cas d'incidència.

<b>Gestor_absencies_team</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Base_ptr_id</b>	Foreign Key
<b>Name</b>	Varchar2
<b>Min_worker</b>	Numeric

Aquesta taula tindrà per objectiu guardar la informació dels diferents equips de treball de la cooperativa. Cal destacar que, tot i que en la durada d'aquest treball de Final de Grau possiblement

no es pugui implementar la lògica, la taula igualment contindrà l'atribut de treballadors mínims per cobrir el server, `min_worker`.

<b>Gestor_absencies_member</b>	
<b>Columna</b>	<b>Tipus</b>
<b>Id</b>	Primary Key
<b>Is_referent</b>	Boolean
<b>Is_representant</b>	Boolean
<b>Worker_id</b>	Foreign Key
<b>Team_id</b>	Foreign Key

Per últim, aquesta taula ens permetrà emmagatzemar i connectar els usuaris de la plataforma amb els diferents equips de treball, afegint atributs com si és classificat com a referent o representant.

### 8.2.2. Disseny de processos de l'aplicació

Pel que fa als processos cal distingir dos grans blocs que tindrà l'aplicació, els processos del Backend i els del Frontend.

Processos del Backend:

- Durant el desenvolupament i la posada a producció del projecte, existirà el procés Python que executarà el servidor de Django. Aquest procés s'encarregarà d'escollir les peticions que li servirà l'Wsgi, aplicarà les modificacions i els càlculs necessaris i s'encarregarà de fer la persistència a la base de dades de PostgreSQL.
- Durant el desenvolupament i la posada a producció del projecte també es tindrà un Sistema Gestor de Base de Dades encarregat de manipular els diferents fitxers de la base de dades. En aquest cas, aquest sistema serà PostgreSQL i rebrà les consultes de l'ORM de Django.
- També serà necessari que durant el desenvolupament i la posterior posada en producció del projecte hi hagi un procés encarregat de comunicar les peticions que arriben al servidor Django, en aquest cas, aquest procés serà Wsgi.
- Quan el projecte estigui llest per posar-se a producció, ja sigui en una màquina virtual única pella o en una compartida, haurà de tenir un servei Nginx que pugui fer filtratge de peticions, redireccions o simplement servir el contingut de l'API.

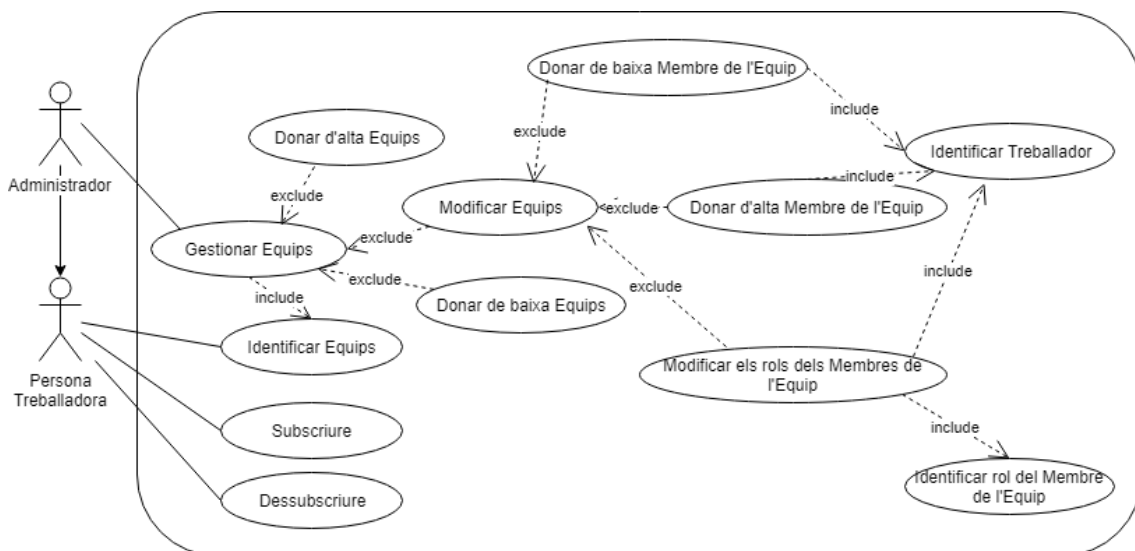
- Tot i que durant el desenvolupament s'utilitzarà Docker per aixecar els contenidors que contendran aquest processos, quan l'aplicació estigui desenvolupada i s'hagi d'implantar al servidor de productiu es farà sense Docker.

En quan als processos necessaris per tenir el Frontent operatiu, tenim que son:

- Un procés per generar els fitxers estàtics que posteriorment es serviran. Tot i que aquest procés només sigui necessari en el moment de fer el deploy, és important tenir-lo en compte ja que sense aquest no es podria aixecar el Frontend.
- Al igual que pel Backend, serà necessari un servei Nginx que serveixi el contingut estàtic al client web i afegir així una capa més per fallada de riscos o redireccions en cas de necessitar-ho.

### 8.2.3. Disseny de casos d'us complexos

Tot i que la gestió d'algunes de les entitats que la plataforma ha de tenir no són complicades de dissenyar com és el cas dels treballadors, d'altres com les ocurrències de les absències poden ser difícils d'implementar i necessiten d'un disseny previ més exhaustiu.



*Cas d'us de gestionar Equips de Treball*

S'ha afegit un diagrama pel cas d'us de gestionar equips per il·lustrar de manera ràpida i senzilla com es gestionaran, podent donar d'alta o de baixa els equips, modificar-los i identificar-los. En el cas de voler-los modificar, es podrà escollir entre donar de baixa els membres de l'equip o be

donar d'alta a nous membres, aspecte que requerirà identificar el treballador que es vol afegir o eliminar del equip. També serà possible canviar de rol als membres existents, procés que requerirà de la identificació, així com el seu rol.

Cal destacar que el treballador únicament podrà identificar els equips, donar-se d'alta ell mateix en l'equip o donar-se de baixa.

Pel que fa al resultat del disseny i l'assignació d'ocurrències, cal tenir en compte que:

- Només es podran demanar ocurrències que comencin el dia actual, sigui l'hora que sigui, o que comencin en qualsevol dia posterior a l'actual.
- Al demanar una ocurrència que descompti dies de vacances, abans de crear-se, s'haurà de verificar que l'usuari de la plataforma tingui suficients dies de vacances. En el cas de les ocurrències que generin dies de vacances, o que no es restin, no caldrà fer aquesta comprovació.
- Només es podran anular ocurrències d'absències que encara no hagin succeït, sent el límit de temps el dia actual.
- En cas d'anular una ocurrència que generava dies de vacances, s'haurà de verificar que el treballador té suficients dies de vacances per poder-se anular correctament. En el cas de les ocurrències que resten dies de vacances o no tenen cap efecte sobre aquests, no serà necessària cap comprovació prèvia.
- Un treballador només podrà demanar les seves pròpies absències, tot i que la persona encarregada d'administrar la plataforma podrà assignar dies d'absència a qualsevol treballador.
- Mai podran coincidir dues o més ocurrències en un mateix moment per una mateixa persona.
- En cas de demanar una ocurrència d'absència per un dia determinat que el mateix treballador ja té d'un altre tipus d'absència assignada, els dies coincidents seran sobreescrits per la nova ocurrència fent que es recuperin els dies de l'ocurrència que queda sobreescrita. D'aquesta manera, en cas de coincidir just en la meitat d'una ocurrència de major durada, aquesta quedarà partida en dues o si coincideix al final o al principi, aquesta acabarà abans o començarà després segons el cas que es doni.
- Pel cas de les ocurrències que hagin de descomptar dies de vacances, no contaràn els dies del cap de setmana.
- Les ocurrències poden tenir una durada màxima i una mínima establerta pel tipus d'absències que es tracti. En cas que l'ocurrència que es demani no estigui permesa per aquests valors, no serà possible crear-la.



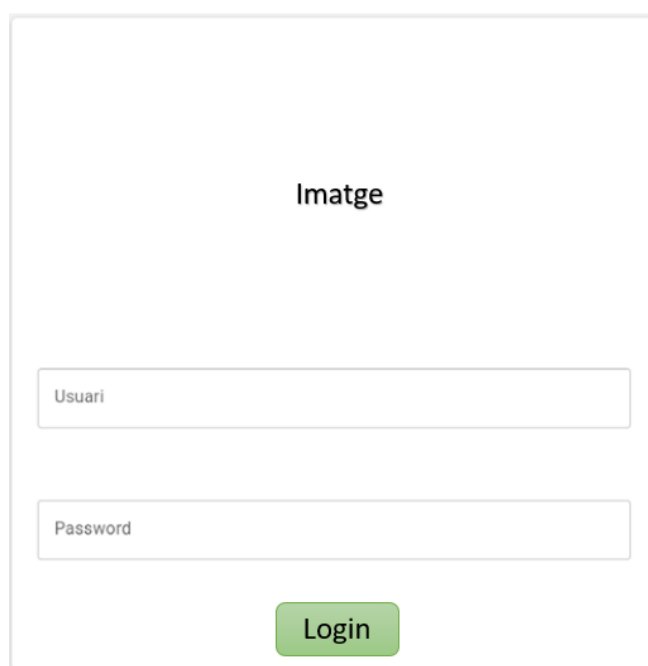
- Les ocurrències, si el tipus d'absència de la que es tracta ho permet, podran arribar a tenir una durada mínima de mig dia i una durada màxima indefinida. Un exemple de durada indefinida són les vacances, ja que els treballadors podran agafar tants dies com vulguin sempre i quan tingui suficients dies de vacances.

#### 8.2.4. Disseny de les interfícies del Frontend

Des del departament de recursos humans no s'han especificat requisits d'interfície pel que, he volgut pensar i dissenyar les interfícies principals de la plataforma. D'aquesta manera, quan calgui implementar-les es sabrà l'objectiu que es pretén, aspecte que reduirà el temps d'implementació.

Tot i que en un principi no es sabia si aprofitar les interfícies de l'Admin de Django perquè l'administrador de la plataforma pogués gestionar les absències, els tipus d'absències, els treballadors i els equips de treball, o si implementar certes funcionalitats en el Frontend perquè es poguessin gestionar aquestes entitats des de la mateixa interfície amb la que els treballadors interaccionen, s'ha escollit afegir les funcionalitats necessàries en les pròpies interfícies del Frontend per, d'aquesta manera, permetre que les persones encarregades d'administrar la plataforma no hagin de connectar-se a dos clients web diferents.

Cal destacar que la majoria de interfícies s'han dissenyat amb molta similitud, pel que no s'han dibuixat les interfícies que són molt semblants a les que es mostraran.



Imatge


Usuari

Password

Login

*Disseny de la part principal de la interfície d'iniciar sessió*

En referència a l'inici de sessió, he pensat en implementar la freqüentment utilitzada Card amb una imatge de la cooperativa o del projecte en sí acompanyat de les dades que l'usuari ha d'entrar i el botó de log in. Un cop l'usuari s'hagi identificat, accedirà a la pàgina del seu perfil, aquesta és:



*Disseny de la interfície dels treballadors*

La interfície “El meu perfil” i la de “Treballador” utilitzaran el mateix format, amb les dades que es mostraran en una Card juntament amb un botó per poder editar els diferents camps. Tot i que en un primer moment els diferents camps estaran bloquejats, al prémer el botó de la part inferior dreta, aquests es desbloquejaran i es podran editar, un cop acabada l'edició s'haurà de guardar prement el mateix botó que, aquest cop, en comptes de ser el llapis d'edició serà un disquet de guardat.

A la part esquerra de la majoria de interfícies apareixerà un menú amb el que els usuaris podran navegar per la plataforma, aquest menú contindrà les opcions següents:

- El meu perfil
- Absències
- Calendari
- ET
- Som Energia
- Log out

Aquest menú serà visible en totes les interfícies exceptuant els formularis per la creació de les noves entitats, de la mateixa manera que no serà accessible en la interfície “Log in”.

En totes les interfícies de la plataforma exceptuant els formularis i la interfície de “Log in”, l'usuari podrà veure on es troba ja que, a la part superior de la Card, apareixerà la secció on es troba.

Una altre interfície molt important és la que permetrà al treballador poder veure el còmput global de vacances que ha reservat per l'any actual i les podrà veure en un calendari anual.

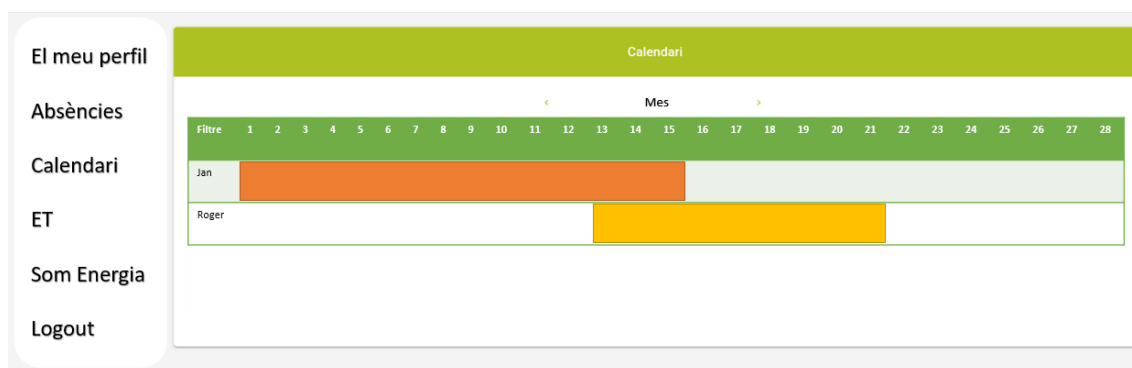


*Disseny de la interfície de "Absències del treballador"*

La interfície "Absències" contindrà el llistat de totes les absències que el treballador ha realitzat o realitzarà durant l'any indicant-ne el moment d'inici i de final. També ho haurà d'indicar en el calendari de la part dreta de la Card.

A la part inferior de la dreta de la interfície es podrà veure un botó per afegir ocurrències, el qual ens redirigirà al seu formulari. També cal destacar que es podrà navegar en els diferents anys gràcies a les fletxes que hi haurà al costat de l'any. D'aquesta manera es podran veure molt fàcilment les absències realitzades durant l'any anterior.

Per observar el conjunt d'ocurrències que tenen els diferents treballadors de la plataforma, s'implementarà una interfície que mostrarà les vacances o altres absències en forma de taula.



*Disseny de la interfície “Calendari”*

La interfície “Calendari”, al igual que la majoria de la plataforma estarà formada pel menú del Gestor d'Absències a la part esquerra, juntament amb una Card que contindrà el Calendari de l'oficina en forma de taula.

La interfície estarà formada pel mes que es visualitza a la taula, juntament amb unes fletxes per poder visualitzar mesos anteriors i posteriors, així com amb la pròpia taula de les absències que representarà el calendari del mes que s'estigui visualitzant. Una part molt important de la taula serà el filtre, que ens haurà de permetre seleccionar tant per treballadors com per equips de treball.

Un tret importat en el calendari serà que cada tipus d'absència tingui un color diferent, podent identificar el tipus d'absència de que es tracta al passar el ratolí per sobre.

Per últim, també s'ha dissenyat les interfícies dels llistats de les entitats, les entitats que es mostraran en format de llista són les següents:

- Treballadors
- Equips de treball
- Polítiques de vacances
- Tipus d'absències



### *Disseny de les interfícies de llistat*

Les interfícies de llistat, al igual que la majoria de les altres interfícies, contindran el menú de la plataforma al lateral esquerre de la interfície. Al centre es trobarà un Card que permetrà veure la secció en la que es troba l'usuari de la plataforma, en aquest cas Treballadors o Equips de Treball. Aquestes dues seccions seran subinterfícies de "ET", al igual que les interfícies de llistats de "Tipus d'Absències" i "Polítiques de Vacances" seran subinterfícies de "Som Energia". Es podrà canviar de subinterfície amb un clic a la capçalera de la Card.

A la part central de la Card es trobarà el llistat d'entitats, en aquest cas de Treballadors, juntament amb un filtre per poder torbar ràpidament el treballador que necessitem. Cada entitat de la llista serà un enllaç a la pròpia entitat, pel que podrem accedir a aquesta per veure-la, modificar-la o eliminar-la.

A la part inferior dreta de la interfície podrem trobar un botó que ens permetrà, en aquest cas, afegir nous Treballadors a la plataforma. Cal destacar que aquest botó només serà visible pel treballador que tingui permisos d'administrador de la plataforma ja que en requisits funcionals del projecte es va estipular que només podrien crear noves entitats els administradors.



## 9. Implementació i proves

Aquest apartat té com a objectiu exposar els problemes afrontats durant el desenvolupament de l'aplicació, així com la seva solució. Les principals problemàtiques trobades durant la implementació han sigut causades pel desconeixement de les tecnologies que s'han utilitzat com ara Docker, Django o Mithril. També s'ensenya com s'han preparat els entorns de desenvolupament i algun exemple pràctic de les tecnologies utilitzades.

### 9.1. Preparació de l'entorn del Backend

Per preparar l'entorn del Backend ha sigut necessari aprendre el funcionament bàsic de Docker i Django.

#### 9.1.1. Instal·lació de Docker

En un primer moment, per instal·lar Docker, es va intentar seguir els passos de la documentació oficial però aquesta és molt complexa i al haver-hi diferents versions de Docker, juntament amb el fet que és un software completament diferent a qualsevol utilitzat durant el grau, va dificultar la seva instal·lació.

Per tal d'instal·lar Docker, finalment es va fer com una llibreria més d'Ubuntu amb el següent resultat:

```
benjami@benjami-VirtualBox:~$ sudo apt install docker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  docker
0 upgraded, 1 newly installed, 0 to remove and 550 not upgraded.
Need to get 12,9 kB of archives.
After this operation, 45,1 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 docker amd64 1.5-1build1 [12,9 kB]
Fetched 12,9 kB in 0s (120 kB/s)
Selecting previously unselected package docker.
(Reading database ... 126097 files and directories currently installed.)
Preparing to unpack .../docker_1.5-1build1_amd64.deb ...
Unpacking docker (1.5-1build1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up docker (1.5-1build1) ...
```

*Instal·lació Docker*

```
benjami@benjami-VirtualBox:~$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount docker.io git git-man golang-docker-credential-helpers liberror-perl libpython-st
dlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib pigz python python-asn1crypto
  python-backports.ssl-match-hostname python-cached-property python-certifi python-cffi-backend python-chardet
python-cryptography python-docker python-dockerpty python-dockerpycreds python-docopt
  python-enum34 python-funcsigs python-functools32 python-idna python-ipaddress python-jsonschema python-minima
l python-mock python-openssl python-pbr python-pkg-resources python-requests python-six
  python-texttable python-urllib3 python-websocket python-yaml python2.7 python2.7-minimal ubuntu-fan
Suggested packages:
  aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit
git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn python-doc
  python-tk python-cryptography-doc python-cryptography-vectors python-enum34-doc python-funcsigs-doc python-mo
ck-doc python-openssl-doc python-openssl-dbg python-setuptools python-socks python-ntlm
python2.7-doc binfmt-support
```

*Instal·lació docker-compose*

Docker i docker-compose son els dos CLI que s'utilitzen per aixecar contenidors, tot i que amb Docker ja es podrien aixecar, amb docker-compose obtindrem una capa d'abstracció més, facilitant la implementació del disseny de contenidors.

Un cop feta la instal·lació podem executar les comandes de Docker però ens cal permisos de root per fer-ho. Per tal d'evitar això podem afegir el nostre usuari al grup de Docker, d'aquesta manera ja podrem utilitzar qualsevol funcionalitat de Docker sense necessitat de fer-ho com a root.

```
benjami@benjami-VirtualBox:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
benjami@benjami-VirtualBox:~$ sudo docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
```

*Comanda per veure els contenidors actius*

```
benjami@benjami-VirtualBox:~$ sudo usermod -a -G docker benjami
[sudo] password for benjami:
benjami@benjami-VirtualBox:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
```

*Comanda per afegir un usuari en un grup i la confirmació que ha funcionat*

9.1.2. Creació d'un contenidor

Un cop vam instal·lar Docker vam intentar crear els contenidors necessaris el Backend. Al tractar-se de la primera vegada que ho feia, i al no haver-hi ningú amb experiència en aquest camp, em vaig trobar amb molts problemes fins que després d'analitzar la documentació oficial vaig decidir crear primer un contenidor simple per aprendre el seu funcionament i després crear tots els necessaris per la implementació del Backend.



El primer contenidor que vaig crear per aprendre el seu funcionament va ser amb el següent Dockerfile:

```
FROM python:3
WORKDIR /opt
ADD hola.sh hola_intern_contenidor.sh
```

*Dockerfile*

Aquest Dockerfile és capaç de generar una imatge de Docker basada en una imatge de Python en el que defineix un directori de treball i copia un arxiu de la màquina host al contenidor. Tot i que, amb la simplicitat del primer contenidor no hagués fet falta, també vaig crear el fitxer docker-compose.yml per aprendre el seu funcionament obtenint el següent resultat:

```
version: "2"
services:
  exemple:
    build: .
    command: ['bash', 'hola_intern_contenidor.sh']
```

*Docker-compose.yml*

En la imatge anterior observem que:

- S'utilitza la versió 2 del llenguatge de docker-compose.
- Es defineix un servei anomenat "exmemple".
- Amb la instrucció "build: ." s'especifica que la imatge amb la que s'aixecarà el contenidor serà la que es troba en el propi directori, concretament la que es generarà amb l'arxiu Dockerfile.
- Un cop aixecat el contenidor s'executarà la comanda "bash hola\_intern\_contenidor.sh", el que farà que s'executi el següent script:

```
echo "Hola a tothom i tothoma"
sleep 60
```

*hola\_intern\_contenidor.sh*

El motiu pel qual s'ha forçat a esperar 60 segons és perquè els contenidors de Docker, si no estan configurats de manera especial. Un cop aixecats, criden la comanda indicada que generarà un o varis processos dins el contenidor. Un cop aquest processos acaben els contenidors també ho fan.

D'aquesta manera, es pot aprofitar l'sleep del procés que crida al contenidor quan s'aixeca per mantenir-lo actiu i accedir mitjançant les comandes típiques de Docker com per exemple:

```
benjami@benjami-VirtualBox:~/Projectes/demostracio_docker$ docker-compose build
Building exmemple
Step 1/3 : FROM python:3
--> 954987809e63
Step 2/3 : WORKDIR /opt
--> Using cache
--> d337e928a3d5
Step 3/3 : ADD hola.sh hola_intern_contenidor.sh
--> 8dd9abba8449
Successfully built 8dd9abba8449
Successfully tagged demostraciodocker_exmemple:latest
benjami@benjami-VirtualBox:~/Projectes/demostracio_docker$ docker-compose up
Recreating demostraciodocker_exmemple_1 ...
Recreating demostraciodocker_exmemple_1 ... done
Attaching to demostraciodocker_exmemple_1
exmemple_1 | Hola a tothom i tothoma
```

*Aixecament del contenidor*

En aquest sentit, l'opció "exec" executa una comanda dins del contenidor indicat, juntament amb els flags "-it" que obriran un canal entre la Shell del contenidor i la de la màquina host permetent-nos així entrar en el terminal del contenidor.

```
benjami@benjami-VirtualBox:~/Projectes/demostracio_docker$ docker exec -it demostraciodocker_exmemple_1 bash
root@ffee959eb1f:/opt# ls
hola_intern_contenidor.sh
root@ffee959eb1f:/opt# pwd
/opt
root@ffee959eb1f:/opt#
```

*Comanda per entrar dins el contenidor i el contingut que té aquest*

### 9.1.3. Creació dels contenidors necessaris per desenvolupar el Backend

A l'hora de crear els contenidors del Backend vaig haver de tenir en compte que són serveis que han de tenir connectivitat entre ells i amb l'exterior, estipulant així un port perquè es puguin comunicar entre ells per la xarxa de Docker, i un altre port perquè es poguessin comunicar amb l'exterior dels contenidors, fent així possible que des de fora dels contenidors es pugui accedir als seus serveis.

Els contenidors que conformen el Backend són:

- Un contenidor Django amb un port obert a l'exterior per poder efectuar crides des de la màquina host.
- Un contenidor amb PostgreSQL amb un altre port obert a l'exterior per poder-nos connectar amb el pgAdmin i veure, crear o eliminar registres de la base de dades.

Els principals problemes trobats a l'hora de crear aquests contenidors eren el fet de que Django, en el moment d'aixecar el servidor dins del contenidor, necessita que PostgreSQL estigui aixecat ja que sinó salta una excepció de connexió amb la base de dades i s'acaba l'execució.

El contenidor de PostgreSQL necessita tenir unes variables d'entorn per tal de crear la base de dades en cas de que no existeixi, aquestes variables són:

- Hostname
- Port
- Usuari propietari de la base de dades (per defecte és l'usuari postgres)
- La contrasenya de l'usuari
- El nom de la base de dades

El primer problema és un dels més freqüents que es solen trobar els desenvolupadors que utilitzen Docker ja que fins la versió 2 de docker-compose es podia determinar que un contenidor esperés que un determinat servei estigues actiu abans d'aixecar-se. Tot i que la majoria de desenvolupadors solucionen aquest problema utilitzant la versió antiga de docker-compose, vaig voler investigar com es podia tenir aquesta utilitat amb la versió més nova de docker-compose.

Per sort després d'investigar vaig trobar un script en bash que els desenvolupadors que utilitzen l'última versió de docker-compose han desenvolupat amb el que es recupera aquesta funcionalitat. D'aquesta manera, només cal executar aquest script passant com a paràmetres, el que es vol esperar que, en el nostre cas que el port 5432, port per defecte de PostgreSQL, estigui aixecat i després executar el nostre servidor.

Respecte al segon problema, en la pròpia documentació de Docker, s'ensenya com passar unes determinades variables d'entorn o ve amb l'etiqueta "entorn", en la qual s'han d'escriure

totes les variables que tindrà l’entorn del contenidor, o be amb l’etiquetes “env\_file” en la que en un fitxer .env es poden agrupar totes les variables fent així que el docker-compose.yml sigui més senzill.

#### 9.1.4. La persistència de la base de dades i la sincronització del codi

Un cop vaig tenir els contenidors preparats per aixecar el Backend vaig donar-me compta que cada cop que modificava el codi havia de tornar a crear-los i aixecar-los de nou, cosa que feia que el desenvolupament anés més lent del que desitjava.

Per tal d’arreglar-ho, investigant en la documentació de Docker vaig trobar els Bind mounts i els Volumes, maneres de persistir dades dins dels contenidors i sincronitzar-les amb la màquina host.

D’aquesta manera vaig poder crear un volum de dades mitjançant la carpeta arrel del codi i inserir-lo en un directori del contenidor per tal de tenir el codi constantment sincronitzat i no haver de crear i aixecar els contenidors per cada modificació de codi.

Va ser en aquell moment quan vaig aprofitar per fer que la base de dades de l’aplicació també fos persistent. Aquesta persistència és més complexa ja que el volum que es crea conté tots els fitxers lògics de la base de dades i els diposita en el directori per defecte que PostgreSQL guarda els fitxers de la base de dades. Així doncs, cada cop que s’aixeca el contenidor de PostgreSQL es posen els fitxers de la base de dades a “/var/lib/postgres/data” d’on el sistema gestor de base de dades els agafa per tal que, quan es parés el contenidor, els fitxers i directoris que es troben en aquest path es copiessin en una carpeta de la màquina host.

Al fer que la base de dades sigui persistent ja no es crea automàticament cada cop que s’aixeca el contenidor, fent així innecessari el pas de les variables d’entorn que s’ha comentat en l’anterior apartat.

```
FROM python:3
ENV PYTHONUNBUFFERED 1
WORKDIR /gestor_absencies
```

*Dockerfile final del projecte*

```
version: "3"

services:
  db:
    image: postgres:11.2-alpine
    hostname: db
    ports:
      - "5433:5432"
    env_file:
      - db-variables.env
    volumes:
      - ./postgres-data:/var/lib/postgresql/data
  web:
    build: .
    command: ["bash", "wait-for-it.sh", "db:5432", "--", "bash", "start-server.sh"]
    volumes:
      - ./gestor_absencies
    ports:
      - "8000:8000"
    depends_on:
      - db
```

*docker-compose.yml final del projecte*

En aquestes imatges es pot veure la implementació final del Dockerfile i la del docker-compose.yml en la que consten totes les modificacions que s'han esmentat anteriorment.

### 9.1.5. Instal·lació de Django

Tot i que pel desenvolupament d'aquest projecte s'hagi utilitzat Django dins d'un contenidor facilitant així la seva instal·lació, per instal·lar Django en una màquina com podria ser la de productiu i per tant caldria tenir un sistema de gestió de virtualenv o gestors de paquets de python.

Tot i que hi hagi diferents softwares per administrar els virtualenv, el més bàsic i senzill d'instal·lar és la pròpia llibreria de virtualenv que es pot instal·lar de la següent manera:

```
benjami@benjami-VirtualBox:~/Projectes/Django-demo$ sudo apt install virtualenv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-distutils python3-lib2to3 python3-virtualenv
The following NEW packages will be installed:
  python3-distutils python3-lib2to3 python3-virtualenv virtualenv
0 upgraded, 4 newly installed, 0 to remove and 79 not upgraded.
Need to get 265 kB of archives.
After this operation, 2.286 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-lib2to3 all 3.6.7-1-18.04 [76,5 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-distutils all 3.6.7-1-18.04 [141 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 python3-virtualenv all 15.1.0+ds-1.1 [43,4 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 virtualenv all 15.1.0+ds-1.1 [4.476 B]
Fetched 265 kB in 1s (254 kB/s)
Selecting previously unselected package python3-lib2to3.
(Reading database ... 176419 files and directories currently installed.)
Preparing to unpack .../python3-lib2to3_3.6.7-1-18.04_all.deb ...
Unpacking python3-lib2to3 (3.6.7-1-18.04) ...
Selecting previously unselected package python3-distutils.
Preparing to unpack .../python3-distutils_3.6.7-1-18.04_all.deb ...
Unpacking python3-distutils (3.6.7-1-18.04) ...
Selecting previously unselected package python3-virtualenv.
Preparing to unpack .../python3-virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking python3-virtualenv (15.1.0+ds-1.1) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking virtualenv (15.1.0+ds-1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up python3-lib2to3 (3.6.7-1-18.04) ...
Setting up python3-distutils (3.6.7-1-18.04) ...
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
```

*Instal·lació de la llibreria virtualenv*

Un cop instal·lada, podem executar la comanda virtualenv que genera un entorn virtual de python en el directori on s'executi aquesta comanda. Un cop creat, per tal d'utilitzar-lo, cal executar la següent comanda:

```
benjami@benjami-VirtualBox:~/Projectes/Django-demo$ source gestor_absencies/bin/activate
```

*Comanda per activar el virtualenv*

Fet això, per tal de tenir l'entorn de desenvolupament operatiu, és necessari instal·lar la llibreria pip de Python. Pip és un gestor de paquets de python que permet instal·lar, eliminar o llistar els paquets que tenim instal·lat en l'entorn de Python que s'està utilitzant en aquell moment.

```
benjami@benjami-VirtualBox:~/Projectes/Django-deno$ sudo apt install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  build-essential dpkg-dev fakeroot g++ g++-7 gcc gcc-7 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1 libc-dev-bin libc6-dev libc6-rtls libexpat1-dev
  libfakeroot libgcc-7-dev libitm1 liblsan0 libmpx2 libpython-all-dev libpython-dev libpython2.7-dev libquadmath0 libstdc++7-dev libstdc++7 libubsan0 libubsan0 linux-libc-dev make manpages-dev python-all
  python-all-dev python-crypto python-dbus python-dev python-gi python-keyring python-keyrings.alt python-pip-whl python-secretstorage python-setuptools python-wheel python-xdg python2.7-dev
Suggested packages:
  debconf debconf-i18n gcc-7-doc libstdc++6-7-dbg gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-7-locale libgcc1-dbg libgomp1-dbg libitm1-dbg
  libatomic1-dbg libasan4-dbg liblsan0-dbg libubsan0-dbg libquadmath0-dbg libstdc++6-7-dbg libstdc++7-doc make-doc python-crypto-doc python-dbus-dbg python-dbus-doc
  python-gi-cairo libkf5wallet-bin gir1.2-gnomekeyring-1.0 python-fs python-gdata python-keyczar python-secretstorage-doc python-setuptools-doc python-setuputils-doc
The following NEW packages will be installed:
  build-essential dpkg-dev fakeroot g++ g++-7 gcc gcc-7 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1 libc-dev-bin libc6-dev libc6-rtls libexpat1-dev
  libfakeroot libgcc-7-dev libitm1 liblsan0 libmpx2 libpython-all-dev libpython-dev libpython2.7-dev libquadmath0 libstdc++7-dev libstdc++7 libubsan0 libubsan0 linux-libc-dev make manpages-dev python-all
  python-all-dev python-crypto python-dbus python-dev python-gi python-keyring python-keyrings.alt python-pip python-pip-whl python-secretstorage python-setuptools python-wheel python-xdg python2.7-dev
0 upgraded, 46 newly installed, 0 to remove and 79 not upgraded.
Need to get 1.007 kB/50.3 MB of archives.
After this operation, 107 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 linux-libc-dev amd64 4.15.0-50.54 [1.007 kB]
Fetched 1.007 kB in 1s (802 kB/s)
Extracting templates from packages: 100%
```

## Instal·lació del gestor de paquets pip

Un cop fet això i estan dins de l'entorn virtual de python, es pot instal·lar el paquet de Django. Per tal d'instal·lar-lo s'utilitza pip i, en aquest cas, es fa amb una de les principals metodologies a l'hora de documentar els requeriments d'un projecte, el fitxer requirements.txt, el qual conté les llibreries de les que depèn el projecte.

```
(gestor absencies) benjami@benjami-VirtualBox:~/Projectes/Django-deno$ pip install -r requirements.txt
Collecting Django (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/b1/1d/2476110614367adf07999bc718621f9fc8351e9214e1750cae1832d4090/Django-2.2.1-py3-none-any.whl (7.4MB)
    |#####| 7.5MB 214kB/s
Collecting pytz (from Django->r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/3d/73/fe30c2daaaa0713420d0382b16fbb761409f532c56bdcc514bf7b6262bb6/pytz-2019.1-py2.py3-none-any.whl
Collecting sqlparse (from Django->r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/ef/53/900f7d2a54557c6a37886585a91336520e5539e3ae2423ff1102daf4f3a7/sqlparse-0.3.0-py2.py3-none-any.whl
Installing collected packages: pytz, sqlparse, Django
Successfully installed Django-2.2.1 pytz-2019.1 sqlparse-0.3.0
```

## Instal·lació de paquets utilitzant pip i el requirements.txt

Com es pot observar a la imatge, tot i que només li hem demanat que instal·li la llibreria de Django, també ha instal·lat les llibreries pytz i sqlparse ja que Django depèn d'aquestes dues llibreries per funcionar.

Tot i que amb el que s'ha pogut veure en aquest apartat seria suficient per començar un projecte amb Django, freqüentment, els desenvolupadors que utilitzen aquest framework acostumen a utilitzar eines que ofereix Django com per exemple django-admin, llibreria que conté scripts que automatitzen la creació de l'estructura d'un projecte Django o la creació d'una aplicació o mòdul de Django.

### 9.1.6. Creació d'un projecte amb Django

Com he destacat en l'anterior punt, per crear el projecte amb Django s'ha utilitzat l'eina django-admin. D'aquesta manera, amb una sola comanda genera els directoris i fitxers necessaris per desenvolupar un projecte.

```
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ django-admin startproject --pythonpath gestor_absencies/bin/python3 gestor_absencies_project .
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ ls
gestor_absencies  gestor_absencies_project  manage.py  requirements.txt
```

#### *Creació de l'estructura general d'un projecte*

Com es pot veure a la imatge, la utilització d'aquesta eina genera una carpeta on es guarda la configuració del projecte, en aquest cas gestor\_absencies\_project i el fitxer manage.py que és l'utilitzat per aixecar el servidor, aplicar els nous models o obrir l'interpret del propi framework.

Aquesta comanda també ens genera les migracions de la base de dades perquè el projecte pugui funcionar, com per exemple els permisos, usuaris o administradors. El fitxer de configuració es genera de manera predeterminada però apunta a una base de dades amb un sistema gestor sqlite3, aspecte que no ens és útil per aquest projecte.

Per tal de fer que aquest projecte utilitzi una base de dades PostgreSQL és necessari editar aquest fitxer estipulant quin sistema utilitza, la localització d'aquest i les credencials amb les que Django es podrà connectar a ell.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

*Configuració de connexió a la base de dades generada per django-admin*



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'gestor_absencies_db',
        'USER': 'gestor_absencies',
        'PASSWORD': '1234',
        'HOST': 'localhost',
        'PORT': '5432'
    }
}
```

*Configuració de connexió a la base de dades perquè utilitzi PostgreSQL*

Un cop fet això, cal aplicar les migracions que té per defecte Django i que s'han generat amb django-admin. Per tal de fer-ho s'utilitza l'opció migrate del manage.py.

```
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
```

*Aplicació de les migracions que venen per defecte amb la creació del projecte*

A continuació ja es pot executar el projecte. Per fer-ho, s'utilitza el manage.py amb l'opció runserver que provoca que s'aixequi el servidor.

```
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 18, 2019 - 11:27:55
Django version 2.2.1, using settings 'gestor_absencies_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

*Execució del servei Django*

Com es pot veure en la imatge, el servidor s'ha inicialitzat correctament i escolta les sol·licituds en el port 8000, tot i així, per tal de desenvolupar un Backend amb Django és necessari afegir apps pròpies, ja que aquestes són els mòduls que tindrà el servidor. Per tal de fer-ho, es pot utilitzar l'eina django-admin o el propi manage.py.

```
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py startapp gestor_recursos_humans
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ ls
db.sqlite3 gestor_absencies gestor_absencies_project gestor_recursos_humans manage.py requirements.txt
```

### *Creació de l'estructura general d'una app amb el manage.py*

D'aquesta manera es crea el directori on està l'esquema de la nova app, la qual podrem afegir al nostre projecte. Per fer-ho serà necessari que modifiquem el fitxer de configuració del projecte.

Un cop fet tot això, el projecte està llest per tal de poder desenvolupar en ell una aplicació de Django.

#### 9.1.7. Implementació dels models amb Django

Una de les principals dificultats que em vaig trobar durant el projecte va ser la creació dels models que necessitava l'aplicació, ja que al tractar-se d'un framework que anteriorment mai havia utilitzat i degut a la corba d'aprenentatge que té al inici vaig haver de llegir-me tota la documentació oficial de Django.

Un cop vaig entendre el funcionament dels models i com es transformen a les diferents taules de la base de dades vaig poder desenvolupar tot el que anteriorment havia dissenyat.

```
from django.db import models
class Persona(models.Model):
    nom = models.CharField(max_length=50)
```

### *Exemple d'un model de Django*

El model descrit en la imatge genera una taula en la base de dades anomenada "Persona", en la qual afegeix dos columnes: un identificador únic i un nom amb una mida màxima de 50 caràcters.

El motiu pel qual afegeix un identificador únic, que a més serà autoincremental i clau primària, és que el model “Persona” hereta del Model general que Django ofereix, el qual ja té aquests aspectes implementats.

Tot i que s’hagi escrit el codi i s’hagi reiniciat el servidor, en aquest encara no hi figura aquesta taula. Per tal de poder utilitzar el model, i que aquest tingui persistència en la base de dades, és necessari crear unes migracions amb la comanda `manage.py makemigrations` i aplicar-les a la base de dades com hem fet anteriorment amb les migracions bàsiques de Django.

```
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py makemigrations
Migrations for 'gestor_recursos_humans':
  gestor_recursos_humans/migrations/0001_initial.py
  - Create model Persona
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, gestor_recursos_humans, sessions
Running migrations:
  Applying gestor_recursos_humans.0001_initial... OK
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 18, 2019 - 11:34:13
Django version 2.2.1, using settings 'gestor_absencies_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

*Generació de les migracions, l'aplicació d'aquestes i l'execució del servidor*

Com es pot veure en la imatge, al tractar-se d'un model de l'app `gestor_recursos_humans`, Django genera l'arxiu de migració en el propi mòdul, fet que després, al aplicar aquestes migracions, es detecti que no s'han aplicat anteriorment i procedeixi a fer-ho.

Un cop aplicades aquestes migracions, quan el servidor s'executi ja es podrà utilitzar el model “Persona”.

```
(gestor_absencies) benjami@benjami-VirtualBox:~/Projectes/Django-demo$ python manage.py shell -i ipython
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from gestor_recursos_humans.models import Persona

In [2]: manel = Persona(nom='Manel')

In [3]: manel.save()
```

*Utilització del model “Persona” amb l'interpret de Django*

### 9.1.8. Integració amb Django REST Framework i primeres crides

Un cop el projecte ja ha obtingut els models que utilitzarà, cal que es pugui comunicar amb el client web. Com be s'ha exposat en anteriors apartats. Aquesta comunicació es realitzà amb REST utilitzant la llibreria de Django REST Framework.

Tot i que, un cop es coneix en profunditat com funcionen els mòduls o apps de Django pot ser senzill instal·lar i integrar Django REST Framework a Django, durant la preparació de l'entorn necessari per desenvolupar el projecte va resultar complexa.

Per instal·lar-ho i integrar-ho només es necessita afegir-ho als requeriments i a les aplicacions instal·lades dins de la configuració del projecte.

```
Django
psycopg2>=2.7,<3.0
djangorestframework
```

*Fitxer requirements.txt amb la dependència a Django REST Framework*

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'gestor_recurso_humans',
    'rest_framework'
]
```

*Apartat d'aplicacions instal·lades en el fitxer de configuració del projecte*

Com es pot veure, en el fitxer de configuració, hi ha moltes aplicacions de Django instal·lades, venint algunes per defecte amb el projecte de Django, com per exemple l'admin, també estan aquí les aplicacions que nosaltres creem com el gestor\_recurso\_humans o les aplicacions de tercers com Django REST Framework, que rep el nom de rest\_framework.

Un cop integrat, es poden crear els fitxers de vistes i serialitzadors per tal de poder interaccionar amb els models des d'una API REST.

```
from gestor_recursos_humans.models import Persona
from rest_framework import serializers

class PersonaSerializer(serializers.HyperlinkedModelSerializer):
    class Meta:
        model = Persona
        fields = ('id', 'nom')
```

*Fitxer views.py*

```
from gestor_recursos_humans.models import Persona
from rest_framework import viewsets
from gestor_recursos_humans.serializers import PersonaSerializer

class PersonaViewSet(viewsets.ModelViewSet):
    queryset = Persona.objects.all()
    serializer_class = PersonaSerializer
```

*Fitxer serialitzer.py*

```
from django.contrib import admin
from django.urls import path
from gestor_recursos_humans.views import PersonaViewSet

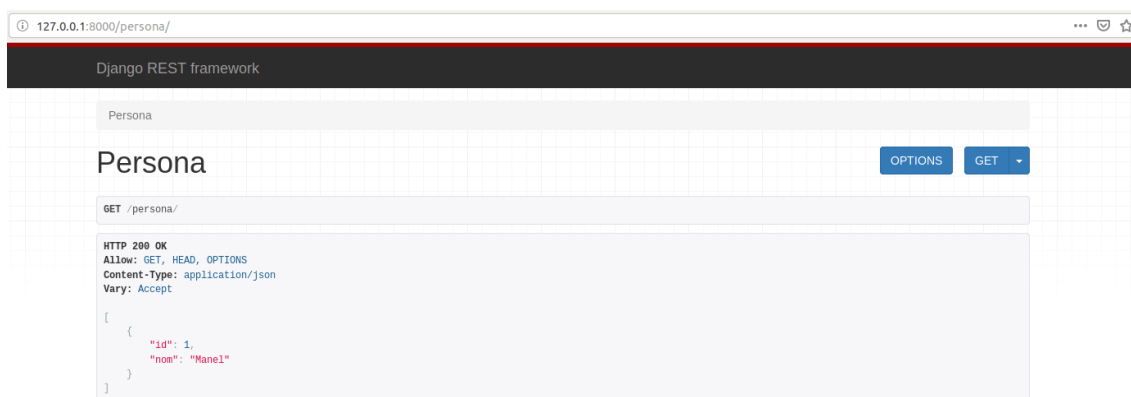
urlpatterns = [
    path('admin/', admin.site.urls),
    path('persona/', PersonaViewSet.as_view({'get': 'list'})),
]
```

*Fitxer urls.py*

Com es pot veure en aquestes imatges, el serialitzador defineix com es mostrarà l'entitat "Persona" i amb quins atributs ho farà mentre que el fitxer views.py descriu com interaccionarà l'entitat amb l'API. Un exemple d'això és el fet de que, si modifiquéssim el queryset del "PersonaViewSet", només es mostrarien les Persones que nosaltres volguéssim i, per tant, si modifiquéssim els fields de la classe "PersonaSerializer" i només deixéssim el nom, únicament es mostraria el nom en les peticions i no es mostraria el seu identificador.

En el fitxer urls.py es defineixen els endpoints amb les que el client podrà interaccionar amb l'aplicació. En aquest cas només es pot utilitzar el mètode GET en l'endpoint "persona/" i aquest mostrarà una llista.

A continuació es pot veure el resultat d'aquest tres fitxers:



*Crida GET realitzada des del navegador*

## 9.2. Desenvolupament del Backend

En aquest apartat es pretén explicar els problemes sorgits durant el desenvolupament del Backend i com s'han solucionat, a més de fer una valoració global de l'aprenentatge obtingut durant el desenvolupament d'aquest.

### 9.2.1. Problemes a l'hora d'escollir la llibreria Django-swingtime per les absències

Durant la fase de disseny del Gestor d'Absències es va veure la possibilitat d'aprofitar una llibreria que ja implementava les tipologies d'esdeveniments i les ocurrències d'aquests. Tot i que en un primer moment vaig voler aprofitar aquesta llibreria, durant el desenvolupament del Backend vaig veure que no era exactament el que necessitava i que el seu aprofitament em podria portar més problemes que solucions, en aquest apartat es parlarà sobre aquests problemes, així com la seva solució.

Primerament, tot i que no fos un problema, cal destacar que de tota la llibreria, composta per models, formularis, templates i views només s'utilitzaven els atributs dels models, ja que la resta de la llibreria estava pensada per utilitzar-se amb els templates de Django i no com una API REST.

El model per determinar els diferents tipus d'absències tenia definit un atribut únic amb la funció de clau primària, aquest atribut era de tipus "string" i podia tenir una llargada màxima de 4 caràcters. Això suposava un problema ja que, els valors únics i amb sentit que es poden generar

amb només 4 caràcters és molt reduït, a més de la complicació que genera utilitzar un “string” com a clau primària.

L'altre principal problema que va sorgir que va ser que no es podia aprofitar el model que vaig implementar, que tenia com a finalitat tenir un control de qui i quan creava l'objecte i el modificava per últim cop. Aquest fet es devia a que en heretar els models de la llibreria de `swingtime` i alhora fer-ho dels meus es produïa un conflicte amb els atributs.

Per aquestes raons i, com s'ha dit anteriorment, aprofitant que només utilitzava els atributs dels models, vaig decidir suprimir la dependència d'aquesta llibreria i implementar jo mateix els models des de zero de tal manera que s'adaptessin millor al Gestor d'Absències.

### 9.2.2. Corba d'aprenentatge de Django juntament amb Docker

Tot i que als punts anteriors s'hagi mostrat de manera molt simple com preparar l'entorn de desenvolupament i fer les primeres implementacions de codi, la corba d'aprenentatge de Docker i sobretot la de Django poden ser molt complexes en un inici.

En el cas de Django, la dificultat resideix en la quantitat de codi ja implementat i la necessitat d'aprofitar-lo. Un exemple d'això s'observa amb els models, ja que la seva possibilitat de configuració requereix d'experiència i coneixement del framework.

Una altre dificultat a l'hora d'implementar el Backend del Gestor d'Absències és Django REST Framework, ja que té tanta complexitat com el propi Django a més de que, la documentació no és tant extensa.

Un exemple dels problemes sorgits durant el desenvolupament és que, per aprofitar els permisos per defecte que ens ofereix Django, els qual són de visualització, modificació, creació i eliminació, hagi sigut necessari investigar al propi codi font del framework per esbrinar que els permisos que aprofita de Django no estan complerts, aspecte que es deu a que aquesta part del framework es va escriure amb anteriors versions de Django i encara no s'ha actualitzat.

Per solucionar aquest problema vaig haver de sobreescrivre el mapeig de permisos que Django REST Framework utilitza dels models de Django afegint-hi el permís del mètode GET.

```
class DjangoModelPermissions(BasePermission):
    """
    The request is authenticated using `django.contrib.auth` permissions.
    See: https://docs.djangoproject.com/en/dev/topics/auth/#permissions

    It ensures that the user is authenticated, and has the appropriate
    `add`/`change`/`delete` permissions on the model.

    This permission can only be applied against view classes that
    provide a `.queryset` attribute.
    """

    # Map methods into required permission codes.
    # Override this if you need to also provide 'view' permissions,
    # or if you want to provide custom permission codes.
    perms_map = {
        'GET': [],
        'OPTIONS': [],
        'HEAD': [],
        'POST': ['%(app_label)s.add_%(model_name)s'],
        'PUT': ['%(app_label)s.change_%(model_name)s'],
        'PATCH': ['%(app_label)s.change_%(model_name)s'],
        'DELETE': ['%(app_label)s.delete_%(model_name)s'],
    }
```

*Implementació de permisos de Django REST Framework  
obtinguda del seu repositori de GitHub*

```
from rest_framework.permissions import DjangoModelPermissions

class GestorAbsenciesPermissions(DjangoModelPermissions):

    perms_map = {
        'GET': ['%(app_label)s.view_%(model_name)s'],
        'OPTIONS': [],
        'HEAD': [],
        'POST': ['%(app_label)s.add_%(model_name)s'],
        'PUT': ['%(app_label)s.change_%(model_name)s'],
        'PATCH': ['%(app_label)s.change_%(model_name)s'],
        'DELETE': ['%(app_label)s.delete_%(model_name)s'],
    }
```

*Modificació de la utilització dels permisos de Django REST Framework pel Gestor d'Absències*

Un altre problema sorgit durant el desenvolupament del Backend ha sigut la necessitat d'aplicar modificacions en les respostes que generen els frameworks, com per exemple la resposta d'autenticació que es retorna.

Per tal de canviar la resposta que retorna al autenticar-se ha sigut necessari analitzar amb deteniment el codi del propi framework. D'aquesta manera, s'ha vist que el mètode POST a l'hora



de generar el body de la resposta ho fa mitjançant un mètode determinat per la pròpia configuració del projecte.

```
def post(self, request, *args, **kwargs):
    serializer = self.get_serializer(data=request.data)

    if serializer.is_valid():
        user = serializer.object.get('user') or request.user
        token = serializer.object.get('token')
        response_data = jwt_response_payload_handler(token, user, request)
        response = Response(response_data)
        if api_settings.JWT_AUTH_COOKIE:
            expiration = (datetime.utcnow() +
                          api_settings.JWT_EXPIRATION_DELTA)
            response.set_cookie(api_settings.JWT_AUTH_COOKIE,
                               token,
                               expires=expiration,
                               httponly=True)

        return response

    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

*Mètode POST de l'autenticació del framework JSON Web Token de Django REST Framework*

```
jwt_response_payload_handler = api_settings.JWT_RESPONSE_PAYLOAD_HANDLER
```

*Punt en el que agafa el mètode que estigui estipulat a la configuració del projecte*

```
def jwt_response_payload_handler(token, user=None, request=None):
    """
    Returns the response data for both the login and refresh views.
    Override to return a custom response such as including the
    serialized representation of the User.

    Example:

    def jwt_response_payload_handler(token, user=None, request=None):
        return {
            'token': token,
            'user': UserSerializer(user, context={'request': request}).data
        }

    """
    return {
        'token': token
    }
```

*Mètode per defecte de la llibreria per generar el body de la resposta d'autenticació*

Com es pot veure en el codi de la llibreria, tot i que aquest mètode rep el token, l'usuari que ha fet la petició i aquesta en sí, únicament retornen el token. En el propi mètode també s'explica molt detalladament com modificar la resposta. D'aquesta manera, en la configuració del Gestor d'Absències, s'ha modificat el mètode per defecte i s'ha fet que senyali a un propi que he implementat perquè retorni el que es va concretar en les especificacions de l'API.

```
'JWT_RESPONSE_PAYLOAD_HANDLER': 'gestor_absencies.common.utils.jwt_response_token_user_id'
```

*Configuració del Gestor d'Absències en la que es defineix un mètode propi per generar el body de la resposta d'autenticació*

```
def jwt_response_token_user_id(token, user=None, request=None):
    user = Worker.objects.filter(username=request.data['username']).first()
    return {
        'user_id': user.id,
        'is_admin': user.is_superuser,
        'token': token,
    }
```

*Mètode implementat per generar el body de la resposta d'implementació*

Com es pot veure, la resposta que retornarà el Backend del Gestor d'Absències un cop s'hagi autenticat correctament, està composta de l'identificador de l'usuari, el propi token i un booleà que indiqui si l'usuari que s'ha autenticat té permisos d'administrador.

### 9.3.Preparació de l'entorn de Frontend

Aquest apartat té com a objectiu explicar les principals dificultats amb les que m'he trobat durant la preparació de l'entorn per programar el Frontend.

La principal dificultat en relació a la implementació de la totalitat del client web ha sigut el poc coneixement del llenguatge JavaScript, acompanyat de que el framework utilitzat és molt poc conegut dins de la comunitat, fet que dificulta la cerca d'informació per resoldre els problemes més freqüents.

### 9.3.1. Creació d'un component de Mithril

Com s'ha exposat en anteriors apartats, un dels principals desavantatges que té Mithril és la necessitat d'implementar els propis components del client web com poden ser els botons o llistats entre altres. Tot i que s'han aprofitat alguns components d'altres projectes de la cooperativa, la majoria s'ha hagut d'implementar des de zero.

En el cas del Frontend del Gestor d'Absències s'ha seguit l'estil de Material Design, aspecte que ha provocat que s'hagin d'implementar els components:

- Button
- Floating action button
- Cards
- Dialogs
- List
- Navigation drawer
- Selectmenu
- Checkbox
- Drawer
- Textfield

A part d'aquests components, s'han hagut d'implementar o integrar d'altres per tal de complir amb els requeriments d'interfície.

Un exemple dels components que s'han creat ha sigut el Floating Action Button o FAB, el qual s'ha implementat de la següent manera:

```
import m from 'mithril';
import '@material/fab/dist/mdc.fab.css';

const MCWFab = {
  view: (vn) =>
    m('button.mdc-fab', vn.attrs, [
      m('span.material-icons mdc-fab__icon', vn.attrs.value),
    ])
};

export default MCWFab
```

*Implementació del FAB del Gestor d'Absències*

Aquest ha sigut un dels components més senzills d'implementar, ja que no ha sigut necessari revisar el valor de cap variable. Per fer-ho només ha sigut essencial consultar la documentació dels components de Material Design i estructurar el components propis com a la documentació s'indica.

## 9.4. Proves

En aquest apartat es pretén explicar el tipus de proves que s'han realitzat al Gestor d'Absències de forma prèvia a la posada a productiu per tal d'assegurar la bona qualitat del software. Distingirem entre els testos funcionals a l'API i un test de càrrega al Backend per assegurar la seva disponibilitat en cas d'augment de la demanda.

### 9.4.1. Peticions testejadades en el Backend

Tot i que es va plantejar fer el desenvolupament d'aquest projecte amb Test-driven development amb testos unitaris, Django i Django REST Framework són dos frameworks que permeten canviar el comportament del servidor modificant un paràmetre, dificultant així testejar el comportament de funcions internes del framework que depenen d'aquestes variables. És per això que s'ha optat per testejar les crides REST que el servidor rep.

D'aquesta manera, s'han estructurat els testos segons l'entitat que testeja, i, en cas de requerir molts testos, en el mètode HTTP que s'utilitza per interaccionar. Els casos que s'han testejat segons les diferents entitats del Backend són:

- Tipus d'Absències:
  - Que l'administrador tingui permisos per veure el conjunt de Tipus d'Absències.
  - Que l'administrador tingui permisos per veure els detalls d'un Tipus d'Absència.
  - Que l'administrador pugui crear un Tipus d'Absència correctament.
  - Que l'administrador pugui modificar un Tipus d'Absència correctament.
  - Que l'administrador pugui eliminar un Tipus d'Absència correctament.
  - Que els treballadors tinguin permisos per veure el conjunt de Tipus d'Absències.
  - Que els treballadors tinguin permisos per veure els detalls d'un Tipus d'Absència.
  - Que els treballadors no puguin crear un Tipus d'Absència correctament.
  - Que els treballadors no puguin modificar un Tipus d'Absència correctament.
  - Que els treballadors no puguin eliminar un Tipus d'Absència correctament.
  - Que al crear-se un Tipus d'Absències es creï totes les Absències corresponents.

- Membres d'Equip:
  - Que es pugui veure correctament la llista de Membres d'Equip.
  - Que es pugui filtrar per Equip la llista de Membres d'Equip.
  - Que es pugui crear un Membre d'Equip correctament.
  - Que es pugui eliminar un Membre d'Equip correctament.
  - Que es pugui modificar un Membre d'Equip correctament.
- Ocurrences d'Absències:
  - Que es pugui veure correctament la llista d'Ocurrences.
  - Que es pugui obtenir la llista d'Ocurrences d'un treballador.
  - Que es pugui obtenir la llista d'Ocurrences dels treballadors d'un equip.
  - Que es pugui obtenir la llista d'Ocurrences entre dues dates.
  - Que es puguin crear Ocurrences simples.
  - Que no es puguin crear Ocurrences de dies passats.
  - Que es puguin crear Ocurrences en el mateix dia.
  - Que al crear una Ocurrence, si ha de generar dies de vacances que ho faci correctament.
  - Que al crear una Ocurrence que resti dies de vacances ho faci correctament.
  - Que al crear una Ocurrence que resti dies de vacances, en el cas de que el treballador que la sol·licita no tingui suficients dies de vacances, que no es pugui crear.
  - Que al crear una Ocurrence que generés dies, si el treballador no li quedaven més dies de vacances es generessin els que li correspongués.
  - Que es pot crear una Ocurrence sense límit de duració si la seva tipologia així ho permet.
  - Que es pot crear una Ocurrence de mig dia si la seva tipologia ho permet.
  - Que al crear una Ocurrence i aquesta coincideixi amb una més gran, la més gran es partirà en dos.
  - Que no es pot crear una Ocurrence que no sigui continua.
  - Que no es pot crear una Ocurrence d'un dia però que no es fa ni al matí ni a la tarda, comprovant així els paràmetres d'entrada.
  - Que al crear una Ocurrence, si coincidís amb l'inici d'una altre, aquesta altre es reduiria fins al final de la que es crea.
  - Que al crear una Ocurrence, si coincidís amb el final d'una altre, aquesta altre es reduiria fins al principi de la que es crea.
  - Que al crear una Ocurrence, si coincidís amb l'inici d'una altre i la que es crea té una durada de mig dia únicament, l'altre es reduiria fins al final de la que es crea.

- Que al crear una Ocurrencia, si coincidia amb el final d'una altre i la que es crea té una durada de mig dia únicament, l'altre es reduiria fins al principi de la que es crea.
- Que un treballador no pot crear Ocurrencies d'una altre treballador.
- Que al crear una Ocurrencia que engloba els dies d'una altre Ocurrencia, aquest altre desapareix i es crea la nova Ocurrencia.
- Que al crear una Ocurrencia que engloba part dels dies d'una altre Ocurrencia, els dies que coincideix amb l'altre Ocurrencia són sobreescrits per la nova Ocurrencia.
- Que es pot eliminar una simple Ocurrencia.
- Que en cas d'eliminar una Ocurrencia que gastava dies de vacances, els dies de vacances es recuperen.
- Que en cas d'eliminar una Ocurrencia que generava dies de vacances, els dies de vacances es perden.
- Que no es pot eliminar una Ocurrencia ja començada.
- Que un treballador pot eliminar les seves pròpies Ocurrencies.
- Que un treballador no pot eliminar les Ocurrencies d'un altre treballador.
- Equips:
  - Que l'administrador tingui permisos per veure el conjunt d'Equips.
  - Que l'administrador tingui permisos per veure els detalls d'un Equip.
  - Que l'administrador pugui crear un Equip correctament.
  - Que l'administrador pugui modificar un Equip correctament.
  - Que l'administrador pugui eliminar un Equip correctament.
  - Que els treballadors tinguin permisos per veure el conjunt d'Equips.
  - Que els treballadors tinguin permisos per veure els detalls d'un Equip.
  - Que els treballadors no puguin crear un Equip correctament.
  - Que els treballadors no puguin modificar un Equip correctament.
  - Que els treballadors no puguin eliminar un Equip correctament.
- Polítiques de Vacances:
  - Que l'administrador tingui permisos per veure el conjunt de Polítiques de Vacances.
  - Que l'administrador tingui permisos per veure els detalls d'una Política de Vacances.
  - Que l'administrador pugui crear una Política de Vacances correctament.
  - Que l'administrador pugui modificar una Política de Vacances correctament.
  - Que l'administrador pugui eliminar una Política de Vacances correctament.

- Que els treballadors tinguin permisos per veure el conjunt de Polítiques de Vacances.
- Que els treballadors tinguin permisos per veure els detalls d'una Política de Vacances.
- Que els treballadors no puguin crear una Política de Vacances correctament.
- Que els treballadors no puguin modificar una Política de Vacances correctament.
- Que els treballadors no puguin eliminar una Política de Vacances correctament.
- Treballadors:
  - Que un Treballador es pugui autenticar correctament.
  - Que l'administrador tingui permisos per veure el conjunt de Treballadors.
  - Que l'administrador tingui permisos per veure els detalls d'un Treballador.
  - Que l'administrador pugui crear un Treballador correctament.
  - Que l'administrador pugui modificar un Treballador correctament.
  - Que l'administrador pugui eliminar un Treballador correctament.
  - Que els treballadors tinguin permisos per veure el conjunt de Treballadors.
  - Que els treballadors tinguin permisos per veure els detalls d'un Treballadors.
  - Que els treballadors no puguin crear un Treballador correctament.
  - Que els treballadors no puguin modificar un Treballador correctament.
  - Que els treballadors no puguin eliminar un Treballador correctament.
  - Que els treballadors puguin modificar el seu propi perfil.
  - Que per crear un Treballador sigui necessari donar-li una Política de Vacances.
  - Que al crear-se un Treballador es creïn les seves Absències.
  - Que al autenticar-se malament, qualsevol treballador obtingui un codi de resposta 400.

Cada un d'aquest testos s'ha implementat abans d'implementar el codi que cobreix, seguint així el principi de Test-driven development. Per la seva implementació s'han utilitzat les tècniques habituals com els Setups o teardowns, funcions o mètodes que s'executen abans de cada test i després de cada un per crear les entitats necessàries i eliminar-se posteriorment perquè no interfereixin en altres testos.

També s'ha utilitzat la llibreria TestCase que ofereix Django, que proporciona una bateria d'Assertions i eines tan útils com el client per fer peticions HTTP al Backend per testejar-lo.

#### 9.4.2. Test de rendiment al Backend

Un cop desenvolupat el Backend del Gestor d'Absències i degut a que un dels requisits funcionals és que pogués suportar una càrrega concurrent de 40 usuaris i tenir un temps de resposta màxim de 5 segons, es va portar a terme una prova en la que es van realitzar 3000 peticions i es va analitzar el temps de resposta del servidor.

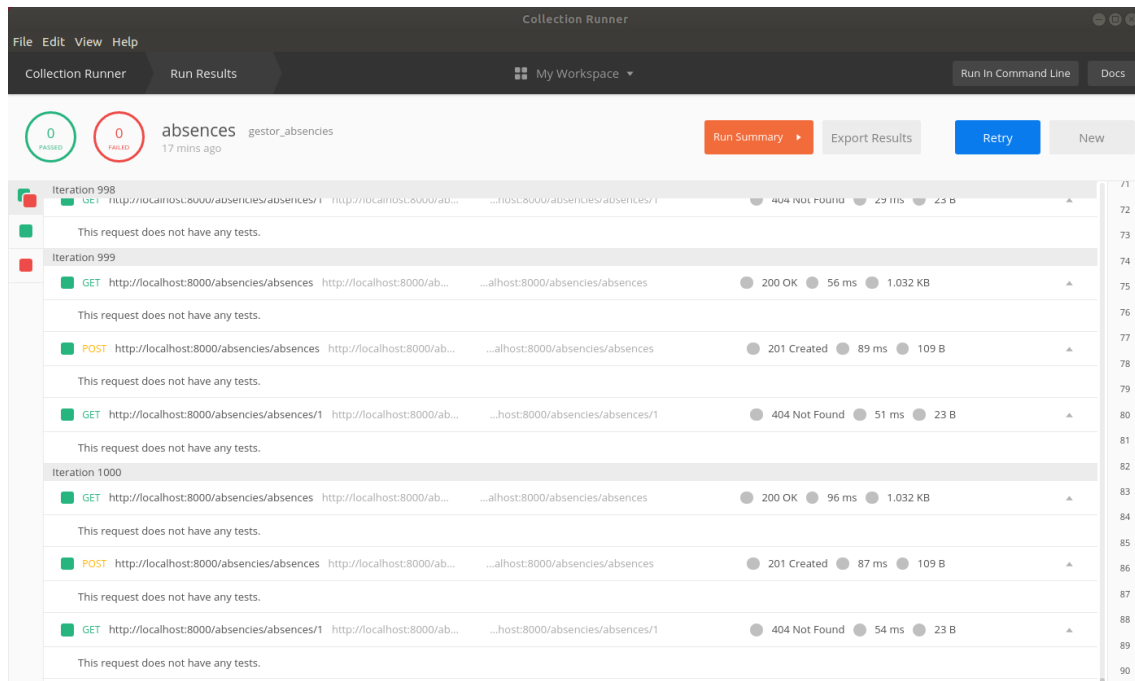
D'aquestes 3.000 peticions, 1.000 van ser peticions GET de llistes d'ocurrències amb un filtre de l'usuari, 1.000 van ser peticions POST de creació d'una nova ocurrència i les altres 1.000 van ser crides GET a una ocurrència en concret, pel que cal destacar que, dels 3 tipus de peticions que es van fer en el test, hi ha les crides que tenen una major càrrega pel Backend degut a la lògica que comporten. A més vaig escollir aquests 3 tipus de peticions ja que inclouen lectura de la base de dades, escriptura de la base de dades, filtratge i molts altres factors que aporten complexitat a les peticions.

Per realitzar aquesta prova de rendiment es va utilitzar Postman, eina molt utilitzada per fer peticions HTTP en els servidors, ja que permet configurar multitud de paràmetres o, en aquest cas, ens permet llençar una ràfega de peticions per analitzar els valors retornats i el temps de resposta.

S'ha decidit utilitzar Postman ja que, per fer petites proves a l'API, tenia tota la col·lecció de peticions que el Backend podia processar. Pel que, en el moment de realitzar el test de rendiment, s'ha decidit la col·lecció que es volia executar. En aquest cas la que computacionalment és més costosa, el número d'iteracions que s'han de realitzar per cada petició de la col·lecció i el delay entre cada petició. En aquest cas, al voler realitzar un test de rendiment, es va decidir no posar delay entre les diferents peticions i dur a terme 1.000 iteracions, d'aquesta manera, de tots els temps de resposta, es podrà obtenir el temps que necessita el servidor en respondre i el temps mínim i màxim de resposta.

Cal tenir en compte que aquesta prova es va realitzar en l'entorn de desenvolupament per tenir una primera impressió del rendiment del Backend, pel que, les especificacions de la màquina no són les mateixes que a productiu. També cal tenir en compte que l'estructura de processos no és la mateixa que a productiu.





*Imatge de Postman un cop realitzada la prova de rendiment*

En el test es va elaborar amb l'objectiu de que la petició més senzilla retornes el codi d'error 404 que Django ens permet especificar, d'aquesta manera, no només vam demostrar el temps d'espera per les peticions realitzades correctament sinó que també vam incloure una petició mal realitzada des del client.

Per analitzar les dades es van exportar en format JSON i posteriorment, amb un intèrpret de Python es van buscar el valor mínim, el màxim i el mitjà.

```
In [12]: response_times = list_response_times + post_response_times + get_response_times
In [13]: len(response_times)
Out[13]: 3000
In [14]: statistics.mean(response_times)
Out[14]: 80.82
In [15]: min(response_times)
Out[15]: 10
In [16]: max(response_times)
Out[16]: 149
```

*Tractament de les dades del temps de resposta del Backend*

Com es pot veure a la imatge, es van unir els temps de resposta de les diferents crides per tractar-les conjuntament. Es va utilitzar la llibreria “statistics” per calcular el valor mitjà i obtenir així els següents resultats:

<b>Temps d'esperat de la petició</b>	<b>Temps (ms)</b>
<b>Mínim</b>	10
<b>Màxim</b>	149
<b>Mitjà</b>	80.82

D'aquesta manera vam observar que el Backend responia molt bé atenent les 3.000 peticions que es van realitzar en aproximadament 4 minuts.

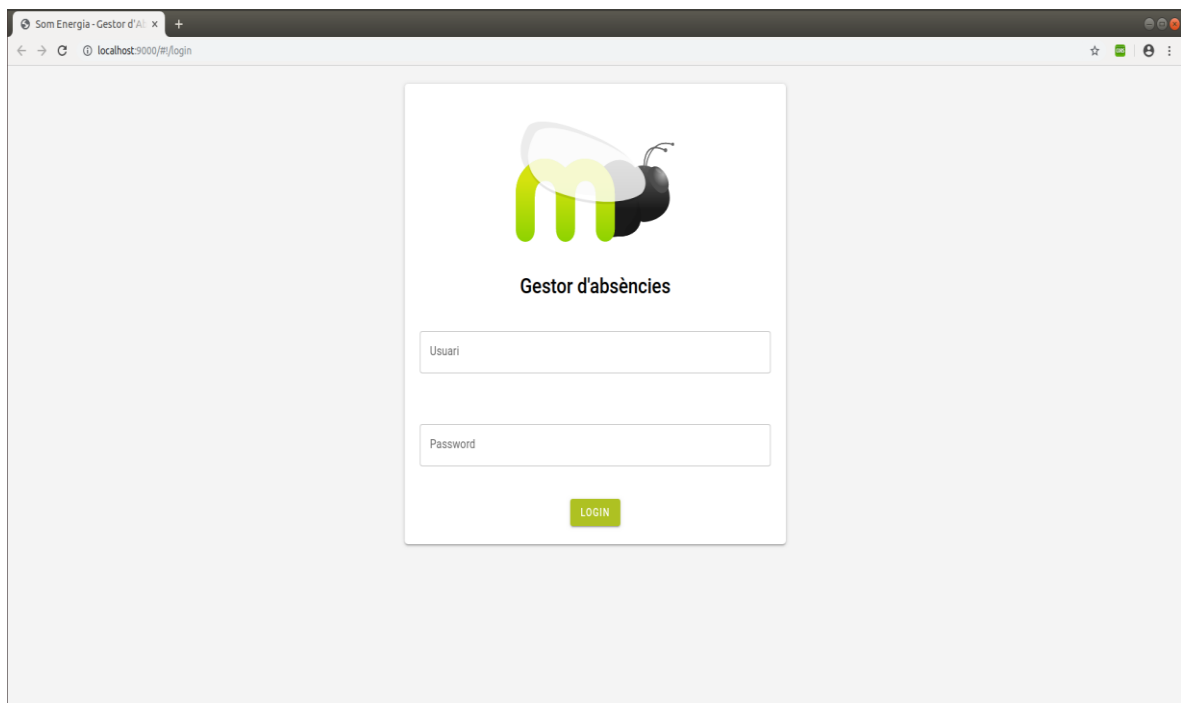
## 10. Implantació i resultats

Aquest apartat té com a objectiu exposar el resultat obtingut de l'anàlisi, el disseny i la implementació d'aquest projecte.

### 10.1. Resultat d'interfície

En aquest apartat es veuran les interfícies desenvolupades perquè els usuaris de la plataforma puguin interactuar amb aquesta. Cal destacar que totes les desviacions de les interfícies respecte als dissenys originals han sigut aprovades pel departament de recursos humans ja que han sigut realitzades per facilitar la utilització de la plataforma.

Al entrar a la plataforma es mostra la interfície de “iniciar sessió”, que és tal i com es va dissenyar durant el procés de disseny de l'aplicatiu:



*Interfície d'inici de sessió del Gestor d'Absències*

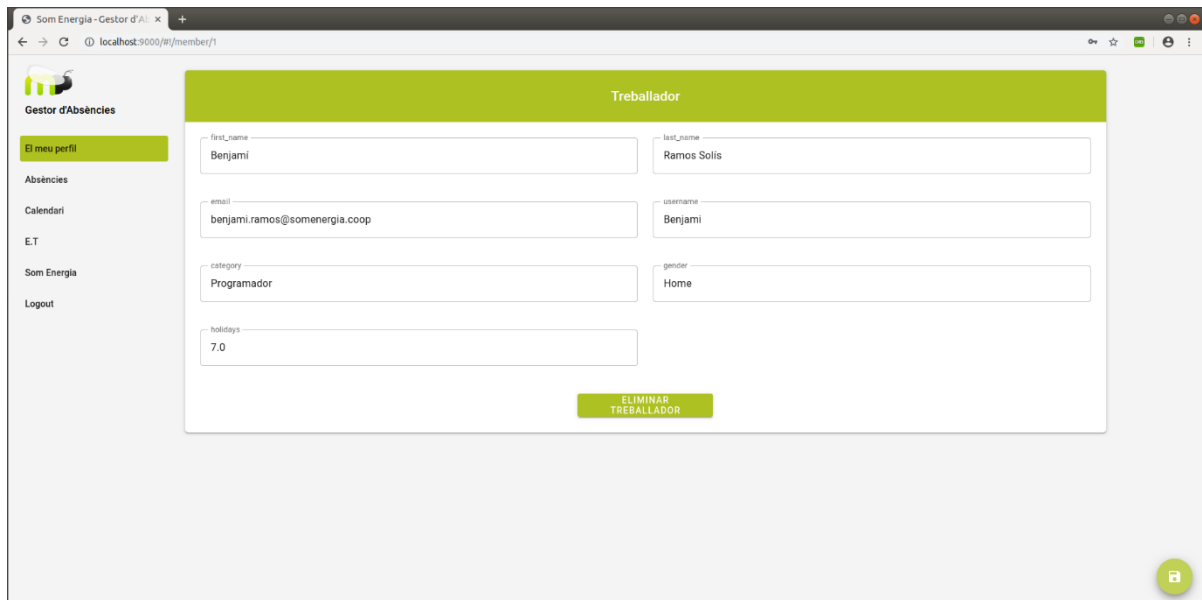
Tot i que en un principi es va dissenyar aquesta interfície perquè només tingués una imatge de la cooperativa, he decidit afegir-hi el títol “Gestor d'absències” per situar als usuaris en el moment d'entrar a la plataforma.

Un cop iniciada la sessió correctament s'accedeix al menú del propi usuari que s'ha registrat, on es troba tota la informació bàsica del treballador:

Treballador	
first_name	Benjami
last_name	Ramos Solis
email	benjami.ramos@somenergia.coop
username	Benjami
category	Programador
gender	Home
holidays	7.0

*Interfície de "El meu perfil"*

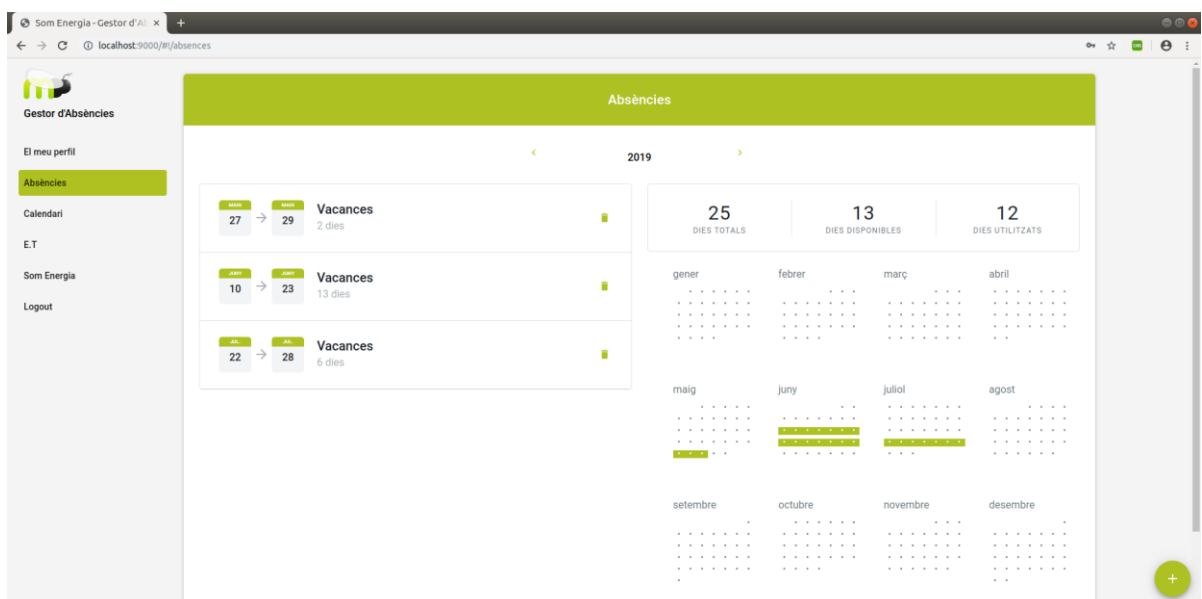
En aquesta interfície, a més de poder veure la informació pròpia, es pot editar al clicant el botó d'edició que hi ha a la part inferior de la dreta. Un cop clicat el botó es desbloquejaran els camps i es podran modificar. Cal destacar que el camp que indica el número de dies de vacances restants que té el treballador no podrà ser modificat pel propi usuari a no ser que sigui l'administrador de la plataforma, que en aquest cas podrà modificar qualsevol camp del seu perfil incloent el número de dies de vacances que té tal com es va concretar en els requisits funcionals del projecte.



*Interfície d'edició de "El meu perfil" amb permisos d'administració de la plataforma*

Com es va dissenyar, el botó per guardar els canvis realitzats és un disquet. En aquesta interfície també es troba l'opció d'eliminar el treballador de la plataforma.

Una altra interfície important és la de "Absències", aquesta interfície, tal i com es va dissenyar, ha de permetre als usuaris veure les absències que ha realitzat i les que té assignades durant l'any.

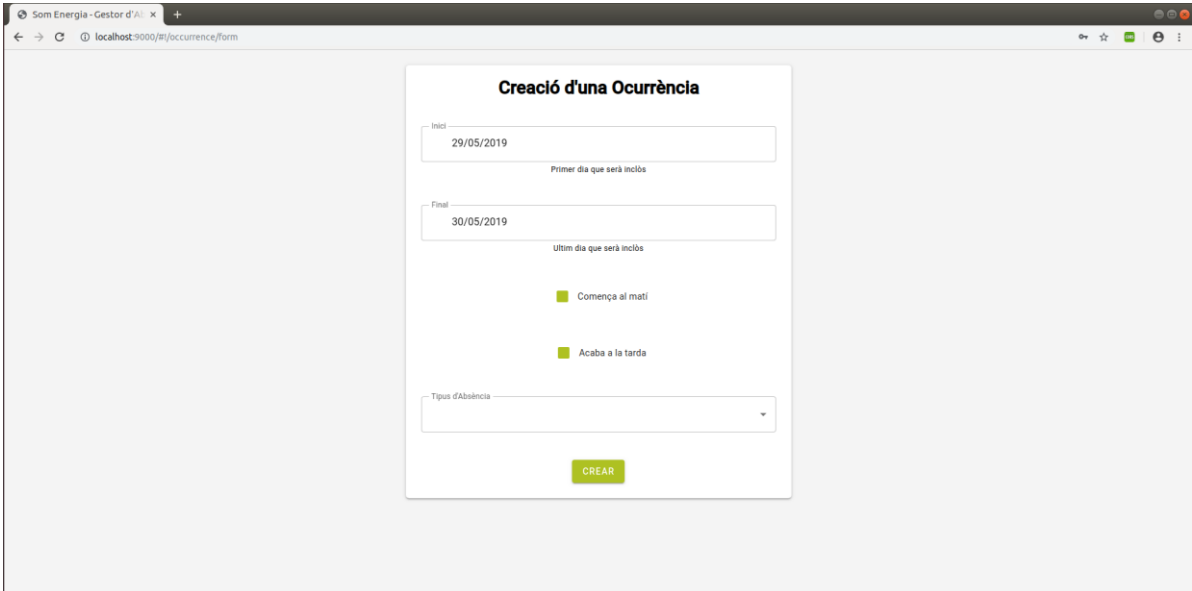


*Interfície "Absències"*

S'ha complert el disseny que es va realitzar i s'han afegit certes modificacions visuals com és el cas del llistat de les absències, amb els dies que ha durat, el seu inici i la seva finalització. També marca molt bé els dies de vacances que s'han demanat durant l'any, el número de vacances que pertoca cada any segons la política de vacances i els dies de vacances que l'usuari encara pot demanar.

Al llistat de les absències, a part del dia que comencen, el dia que acaben, la tipologia d'absència de que es tracte i la durada que té, també hi ha un botó per suprimir-la. A més pel fet de tenir aquest botó tant accessible s'ha implementat una confirmació per esborrar l'absència.

També es pot veure com les vacances que l'usuari ha demanat queden marcades al calendari i com a la part inferior dreta de la interfície hi ha un botó per afegir noves absències. Al clicar aquest botó accedim a un formulari en que consta del dia que s'inicia l'absència, el dia que conclou, si comença o acaba amb mig dia i el tipus d'absència que es tracta.

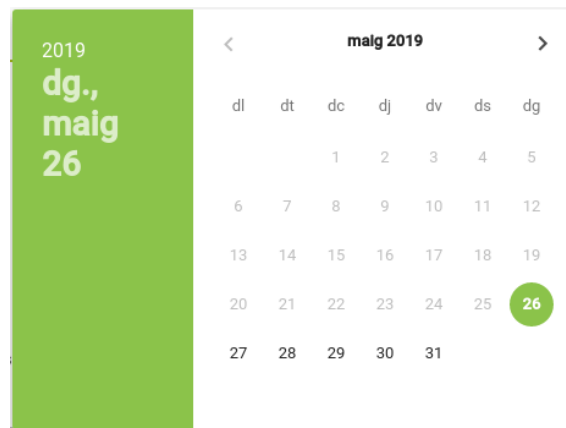


The image shows a web browser window with the URL 'localhost:9000/#/occurrence/form'. The main content is a form titled 'Creació d'una Ocurrencia'. The form has the following elements:

- An 'Inici' field containing '29/05/2019' with the text 'Primer dia que serà inclòs' below it.
- A 'Final' field containing '30/05/2019' with the text 'Ultim dia que serà inclòs' below it.
- Two radio buttons: 'Comença al matí' (selected) and 'Acaba a la tarda'.
- A dropdown menu labeled 'Tipus d'Absència'.
- A green button labeled 'CREAR' at the bottom.

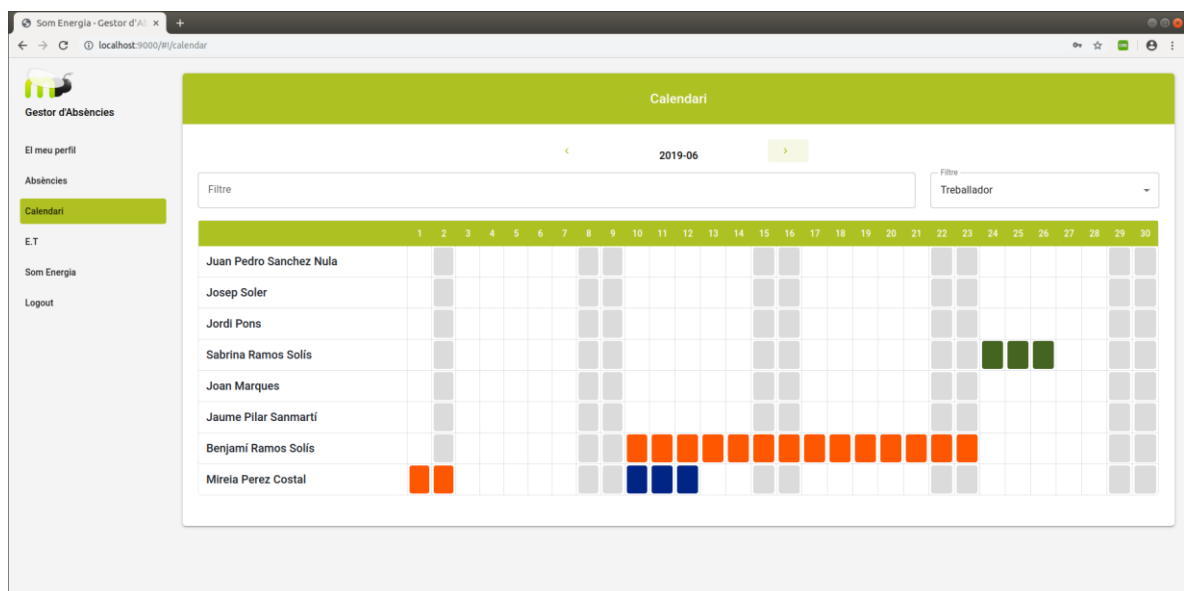
*Interfície del formulari de “creació d’una Ocurrencia”*

Per poder seleccionar el dia que comença i acaba de manera pràctica i còmode s'ha utilitzat un component específic per seleccionar el dia.



*Selector de dia*

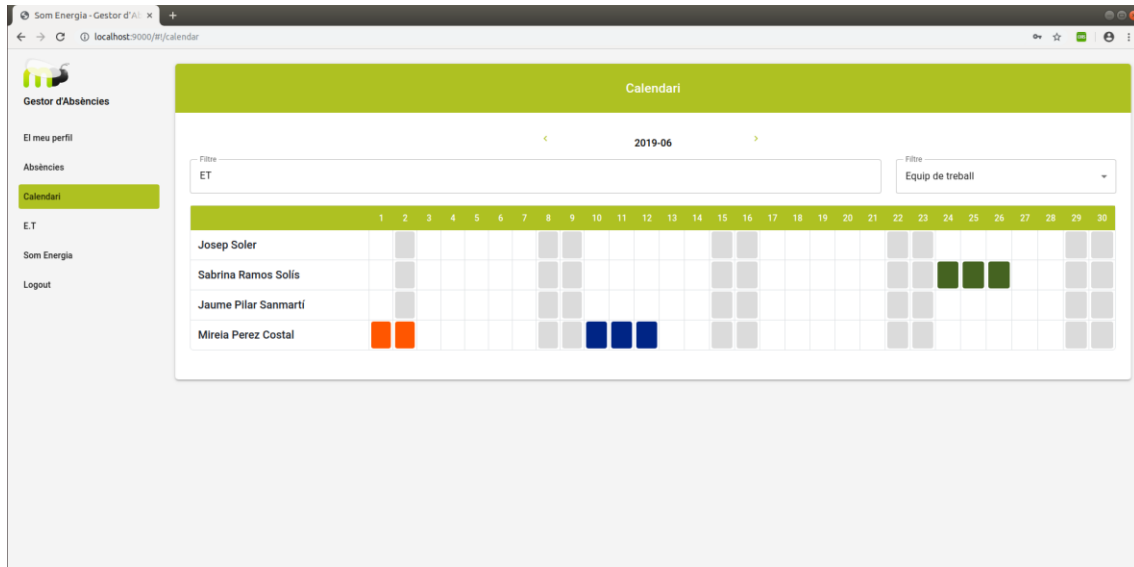
Pel que fa a la interfície de “Calendari”, s’ha estructurat de la mateixa manera que es va plantejar en el disseny de les interfícies. Consta d’un filtre, que permet escollir si filtrar per treballadors o equips de treball i un calendari en forma de taula que engloba als treballadors.



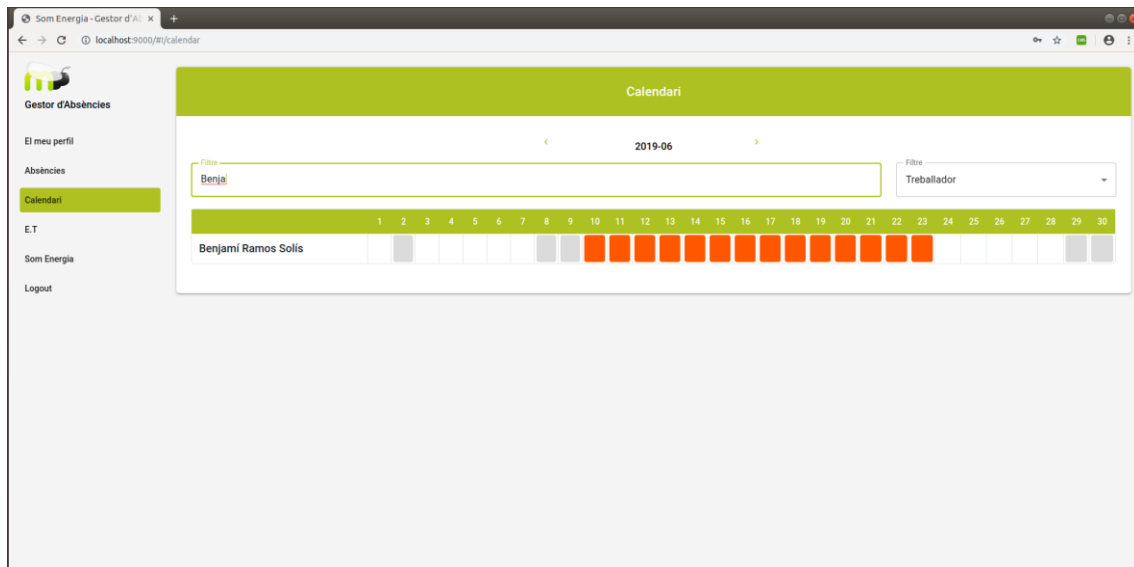
*Interfície de “Calendari”*

Com es pot observar, s’ha utilitzat el mateix disseny de Card que es va dissenyar i s’ha separat el filtre de la taula per aportar major visibilitat. Al calendari, els dies laborals es representen en blanc, mentre que els caps de setmana es poden veure de color gris. La resta de colors representen les absències que tindran els diferents treballadors. Cada tipus d’absència té un color diferent per

poder-los diferenciar entre sí i el nom del tipus d'absència que és es mostra al passar el ratolí per sobre.



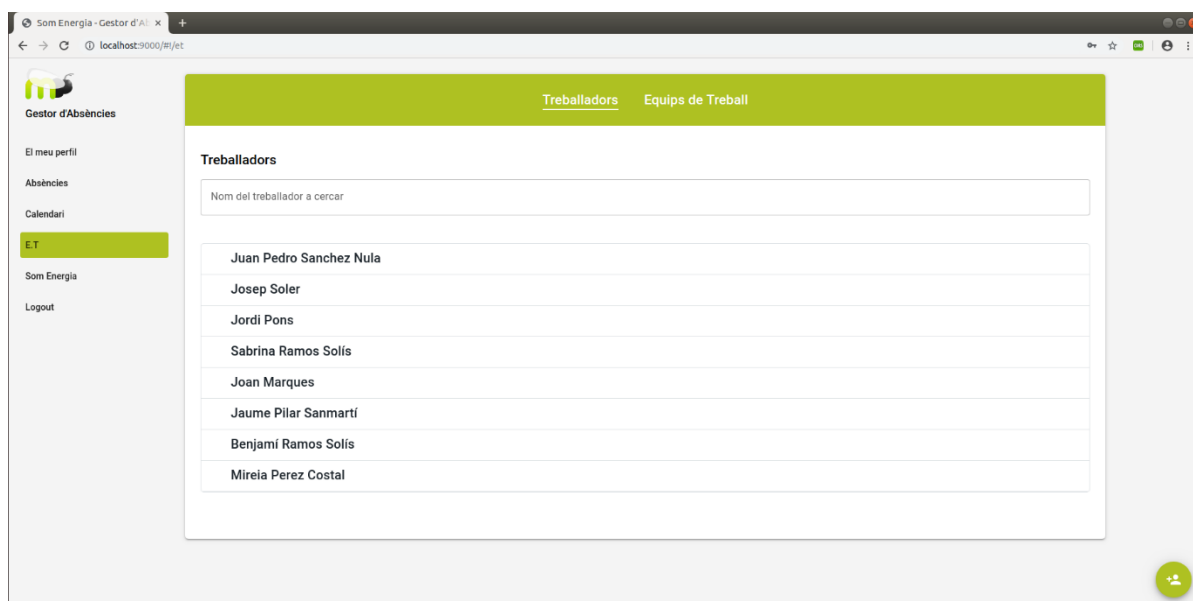
*Interfície “Calendari” en la que s’ha utilitzat el filtre “equips de treball”*



*Interfície “Calendari” en la que s’ha utilitzat el filtre “treballadors”*

Pel que fa a les interfícies de llistat de treballadors i llistat d'equips de treball, com s'ha decidit en el disseny, s'han ajuntat amb un únic grup, ET, però alhora s'han separat amb diferents subinterfícies, la de “Treballadors” i la de “Equips de treball”. Aquestes dues subinterfícies es poden alternar prement la subinterfície a la que es vulgui anar dins de la capçalera de la Card de la interfície de l'ET.

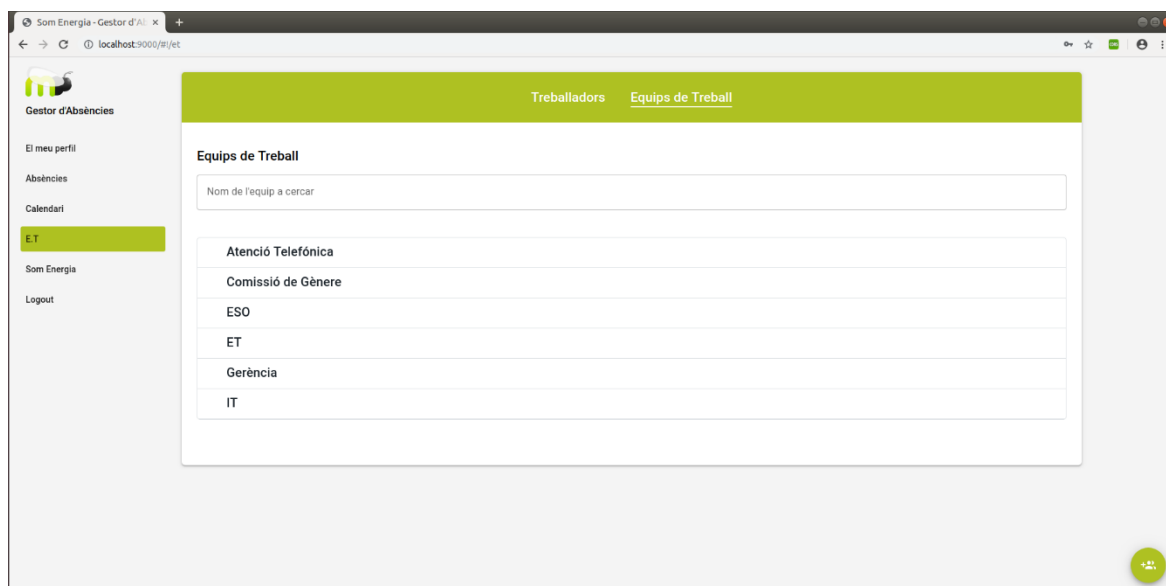




*Subinterfície “Treballadors” dins de la interfície “ET”*

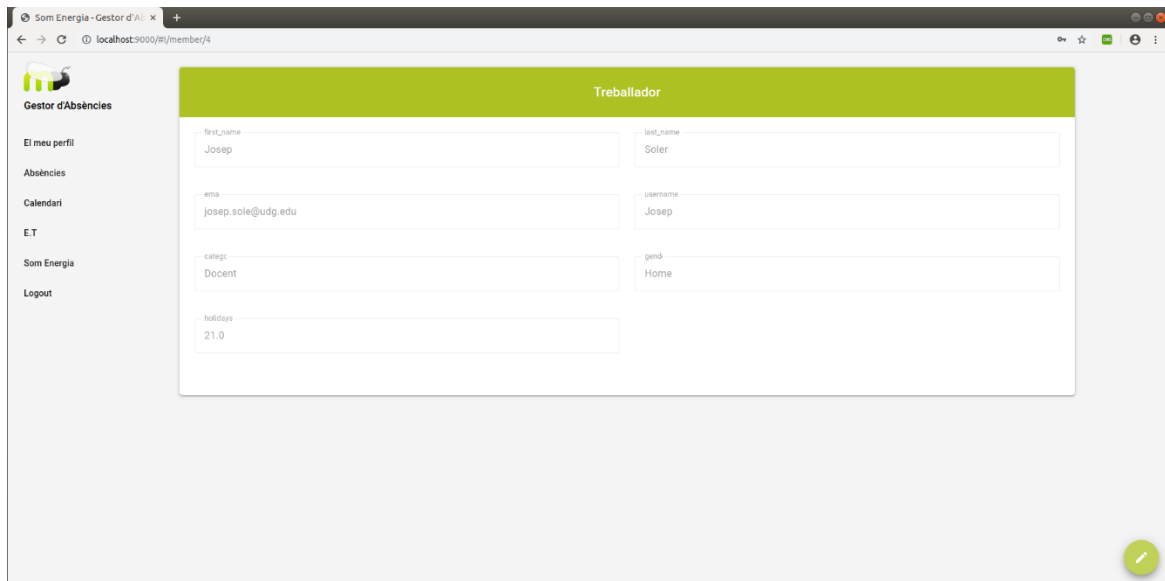
Aquesta interfície és un llistat de tots els treballadors que té la plataforma juntament amb un filtre que permet filtrar pel nom complet de la persona que vulguem trobar. Tal i com s’ha dissenyat, cada fila de la llista ens pot redirigir a la informació de l’usuari facilitant així la navegació.

Per tal d’afegir un treballador nou, l’administrador pot prémer el botó de la part inferior de la dreta de la interfície, fet que obrirà un formulari per la creació d’un usuari. Aquest botó només és visible per l’administrador de la plataforma.



*Subinterfície “Equips de treball” dins de la interfície “ET”*

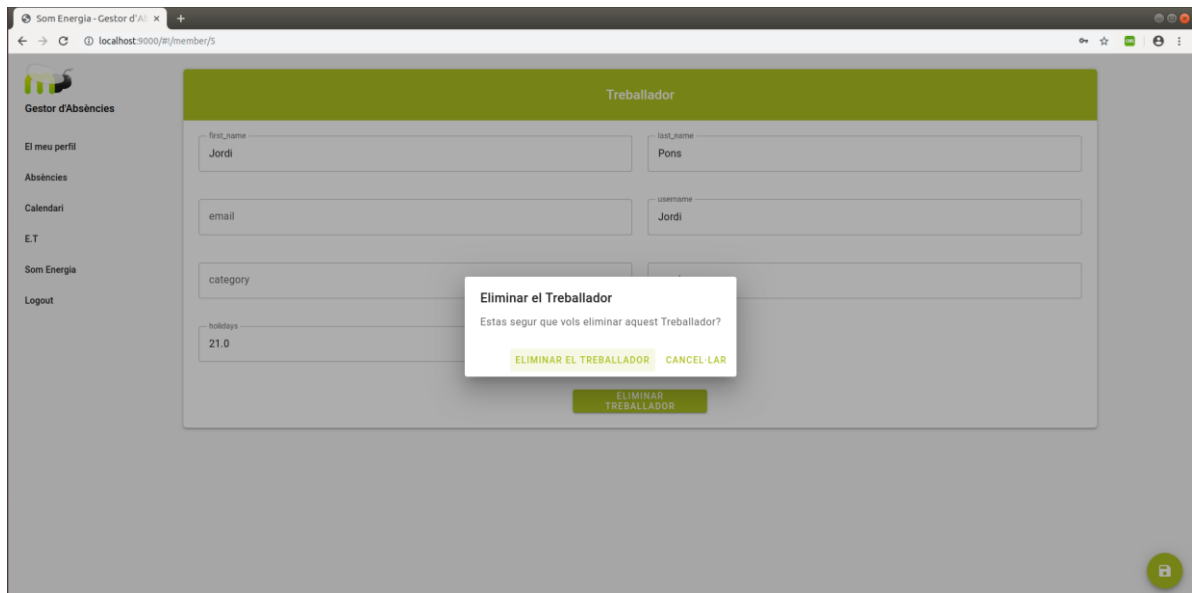
En aquest cas ens trobem amb el mateix que la subinterfície de “Treballadors”, amb l'única diferència de que el botó per crear equips de treball té un símbol diferent. Cada un dels equips de treball també redirigeix a la seva pròpia informació.



*Interfície de “Treballador”*

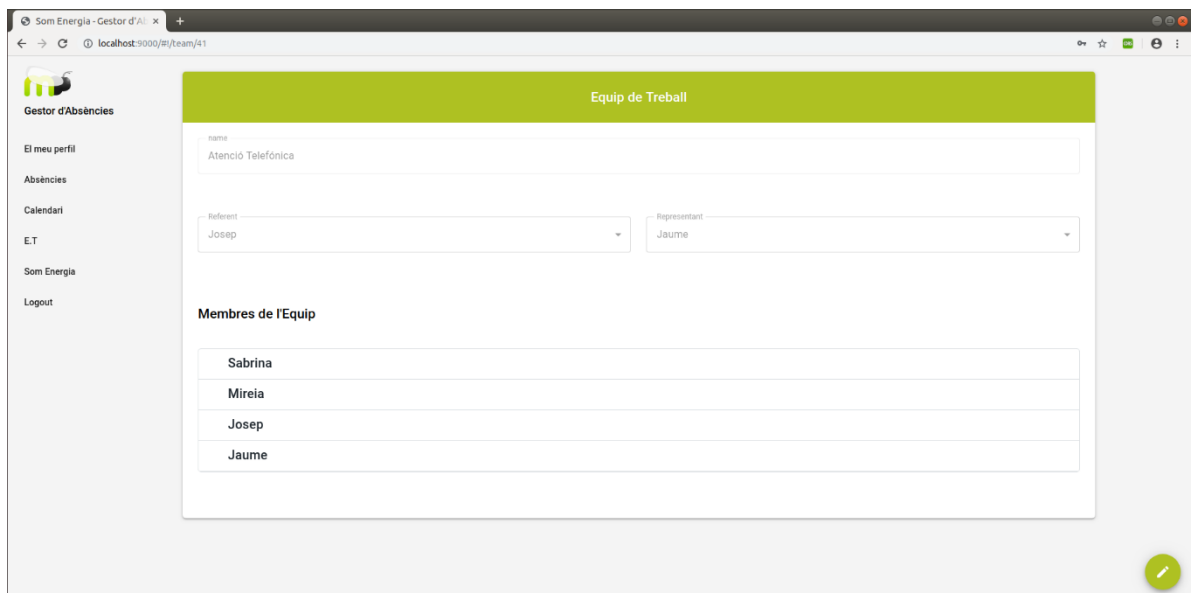
Aquesta interfície presenta el mateix disseny que la interfície “El meu perfil” ja que mostren la mateixa informació però de l'usuari al que s'hagi anat a consultar el perfil. Cal destacar que només l'administrador de la plataforma podrà modificar el perfil d'un altre treballador, tal i com es va especificar als requisits funcionals del projecte.

Com a mesura de seguretat, encara que l'administrador premi el botó d'edició i el de l'eliminació del treballador, abans de fer-ho es demanarà una confirmació explícita de que es vol eliminar al treballador.



*Confirmació preliminar un treballador*

Pel que fa a la visualització d'un equip, s'han realitzat canvis respecte el disseny original ja que en un principi no es comptava amb la complexitat que hi havia per presentar aquesta entitat correctament al client web.

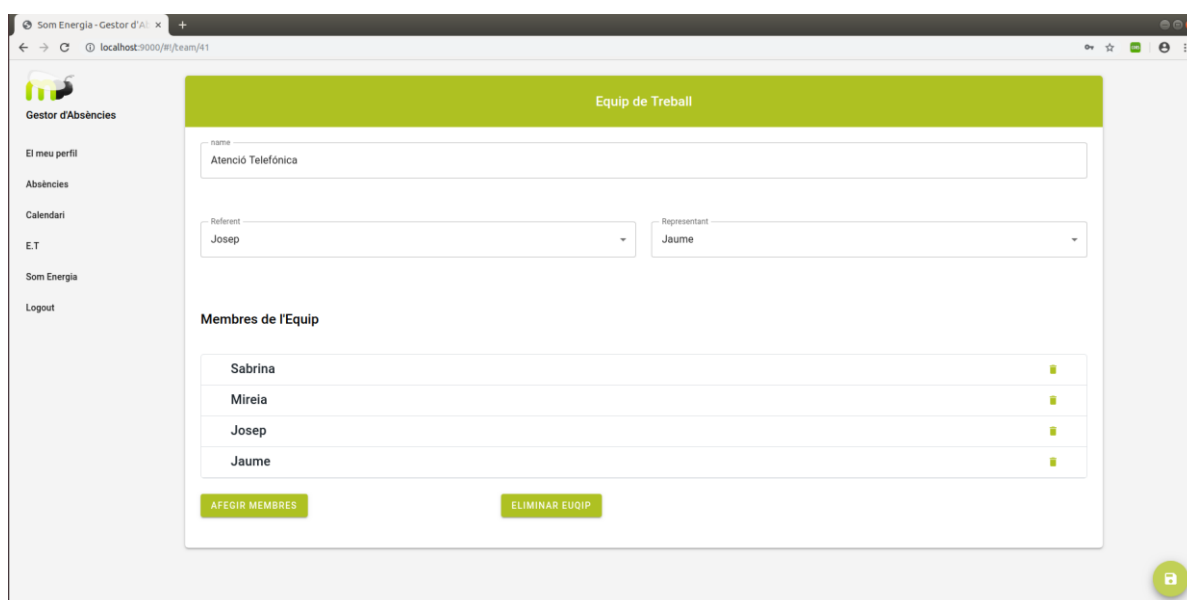


*Interfície "Equip de treball"*

En aquesta interfície es pot consultar la informació bàsica de l'equip com ara el seu nom o el seu representant o referent. A la part inferior de la interfície es pot veure als treballadors que formen

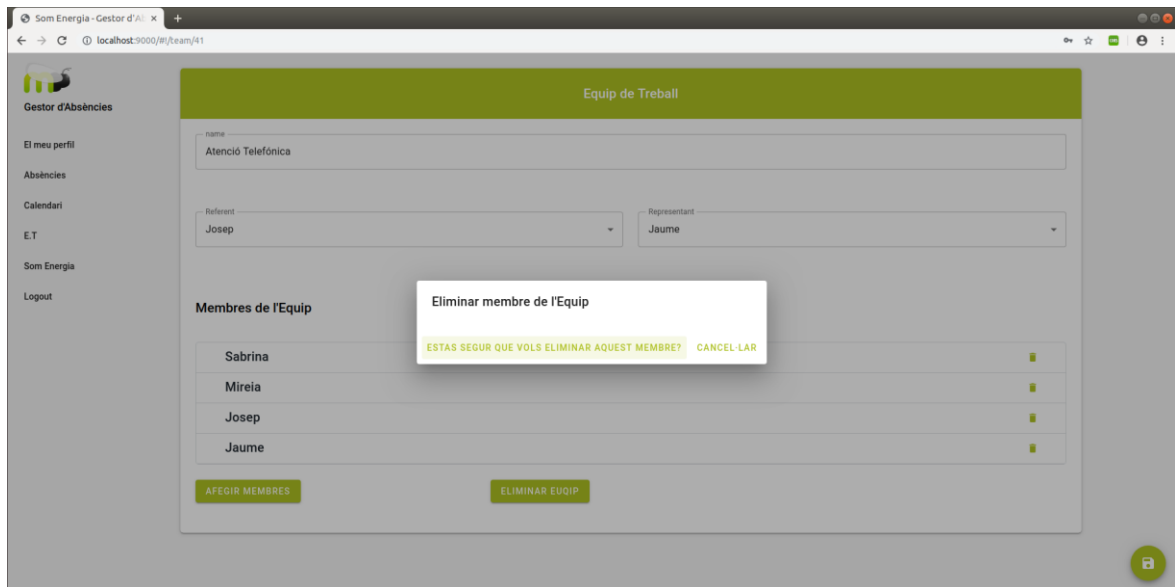
part d'aquest equip de treball i, al igual que amb el llistat de treballadors, contenen un enllaç a les dades de cada treballador.

Tal i com es va destacar als requeriments que havia de complir el projecte, únicament l'administrador de la plataforma pot modificar les dades de l'equip, afegir altres treballadors, assignar els rols de representant i referent de l'equip, eliminar els diferents treballadors de l'equip o bé l'equip sencer. Per tal de poder fer tot això, l'administrador haurà d'accedir al menú d'edició del equip de treball.

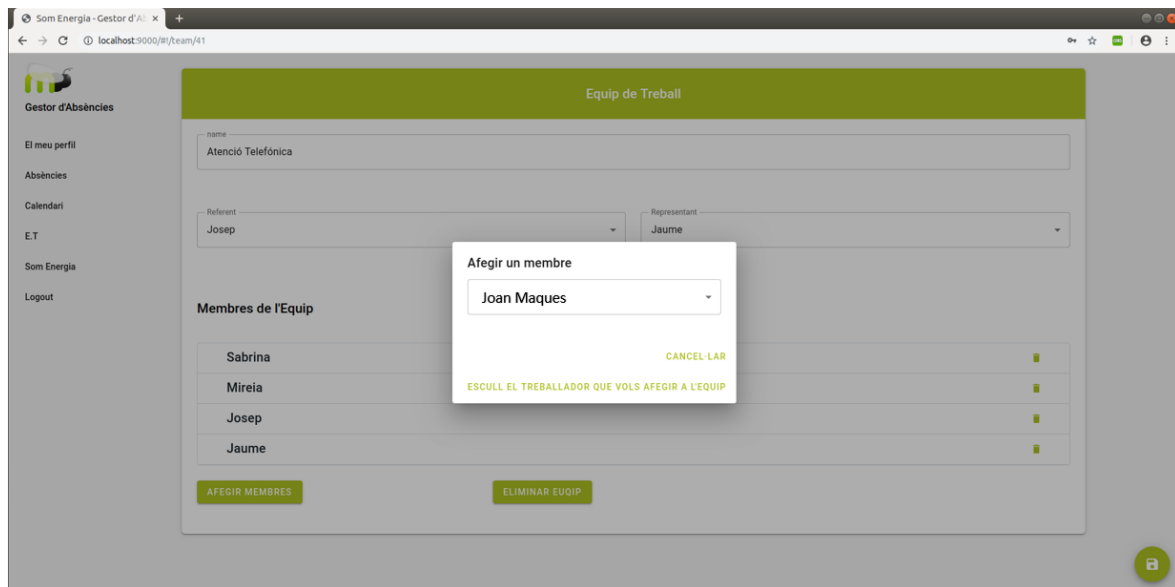


*Interfície d'edició de "Equip de treball"*

Per tal d'eliminar l'equip o afegir nous membres caldria prémer els respectius botons, els quals farien saltar un pop-up per, o bé escollir el treballador que s'afegirà al equip, o bé confirmar que es vol eliminar l'equip. Al igual que amb les absències, a cada element de la llista hi haurà un cubell d'escombraries perquè l'administrador pugui eliminar als treballadors de l'equip que, igual que succeeix amb els altres botons nomenats, requerirà d'una confirmació per fer-ho. Aquestes tres modificacions no requeriran guardar ja que es farà automàticament, mentre que el canvi de referent, representant, o fins i tot de nom, requerirà prémer el botó de guardar.



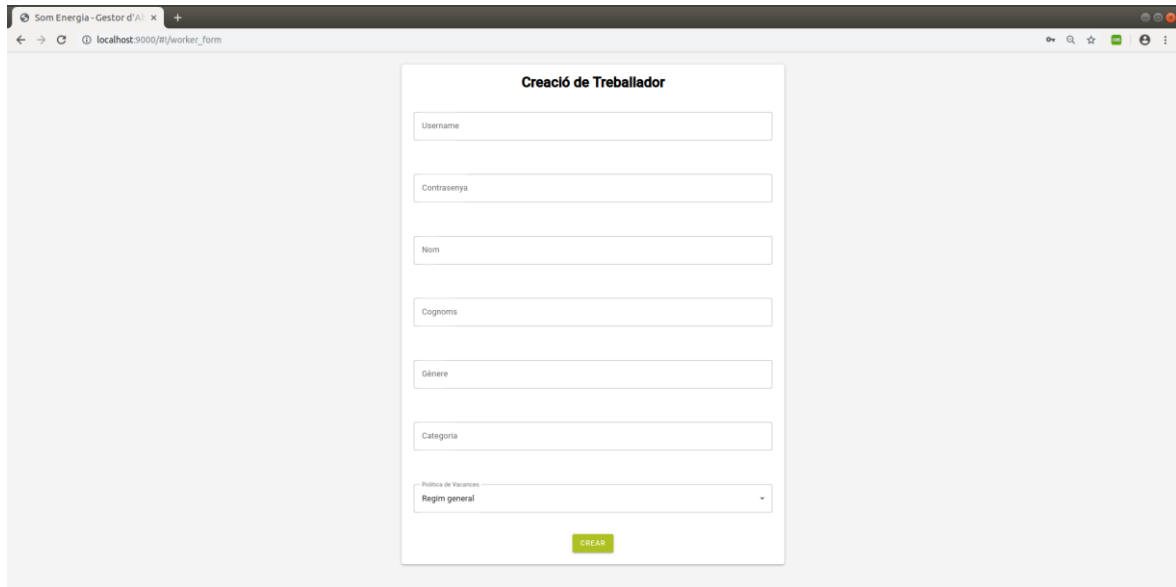
*Confirmació preliminar un membre de l'equip*



*Pop-up per escollir quin membre afegir*

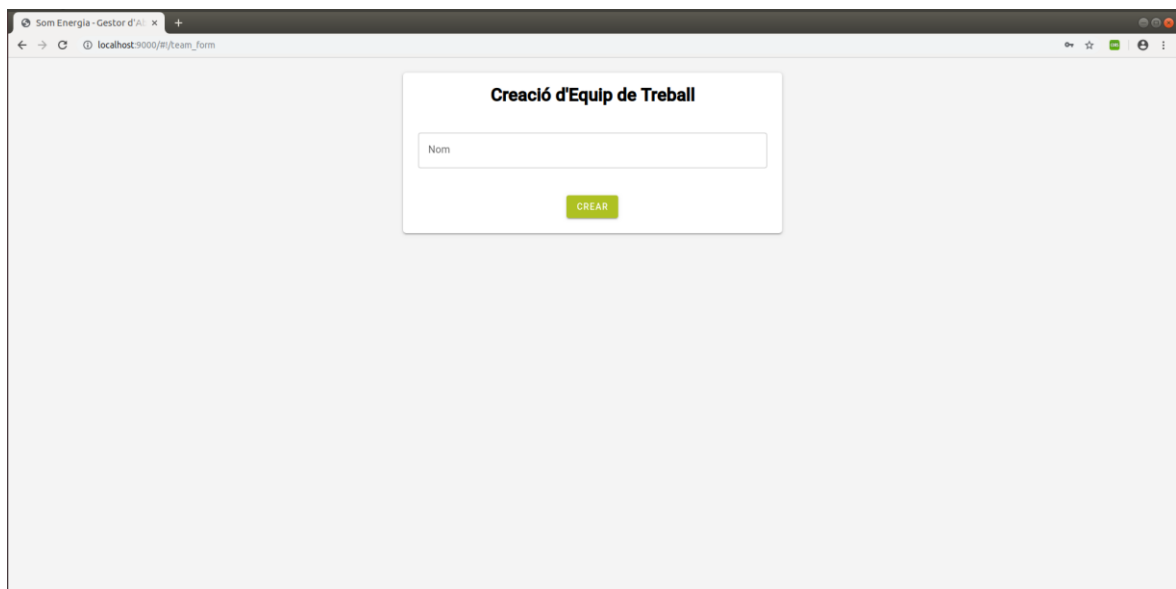
Pel que fa als formularis de creació, no contenen el menú de la dreta de la plataforma sinó que únicament contenen un Card, de la mateixa amplada que la interfície “Log in” en la que es poden veure els camps per donar valor als atributs de l’element que es vulgui crear.

Alguns formularis també contenen un selector per tal d’escollir una de les opcions vàlides.



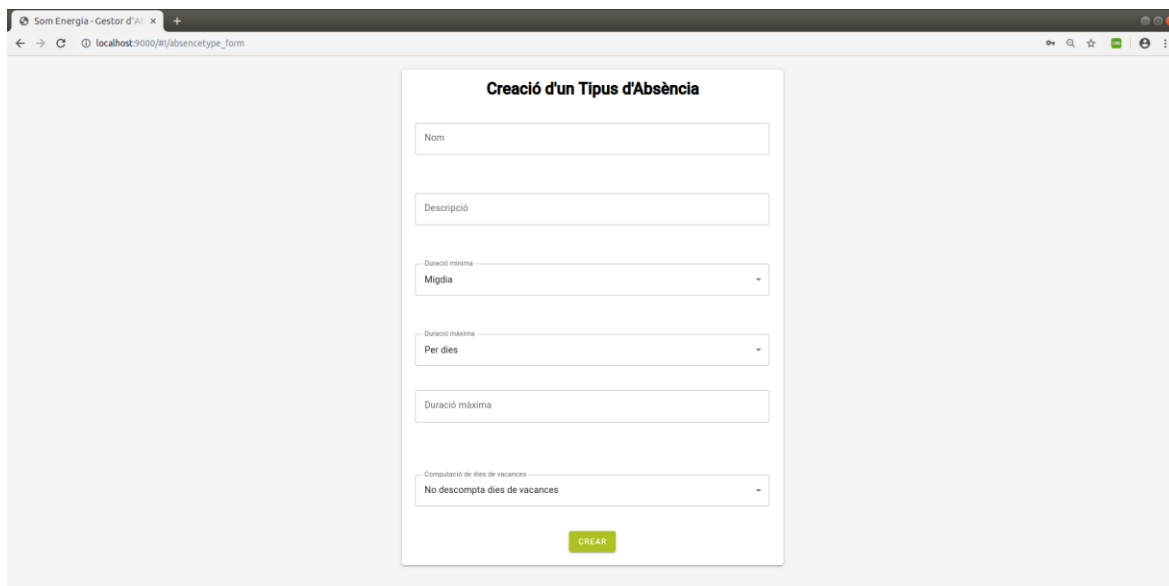
The screenshot shows a web browser window with the address bar displaying 'localhost:9000/#/worker\_form'. The main content area features a form titled 'Creació de Treballador'. The form contains the following fields: 'Username', 'Contrasenya', 'Nom', 'Cognome', 'Gènere', 'Categoria', and 'Política de Vacances' (with a dropdown menu currently showing 'Regim general'). A green 'CREAR' button is positioned at the bottom center of the form.

*Formulari de “creació d’un Treballador”*



The screenshot shows a web browser window with the address bar displaying 'localhost:9000/#/team\_form'. The main content area features a form titled 'Creació d'Equip de Treball'. The form contains a single 'Nom' field and a green 'CREAR' button at the bottom center.

*Formulari de “creació d’un Equip de treball”*



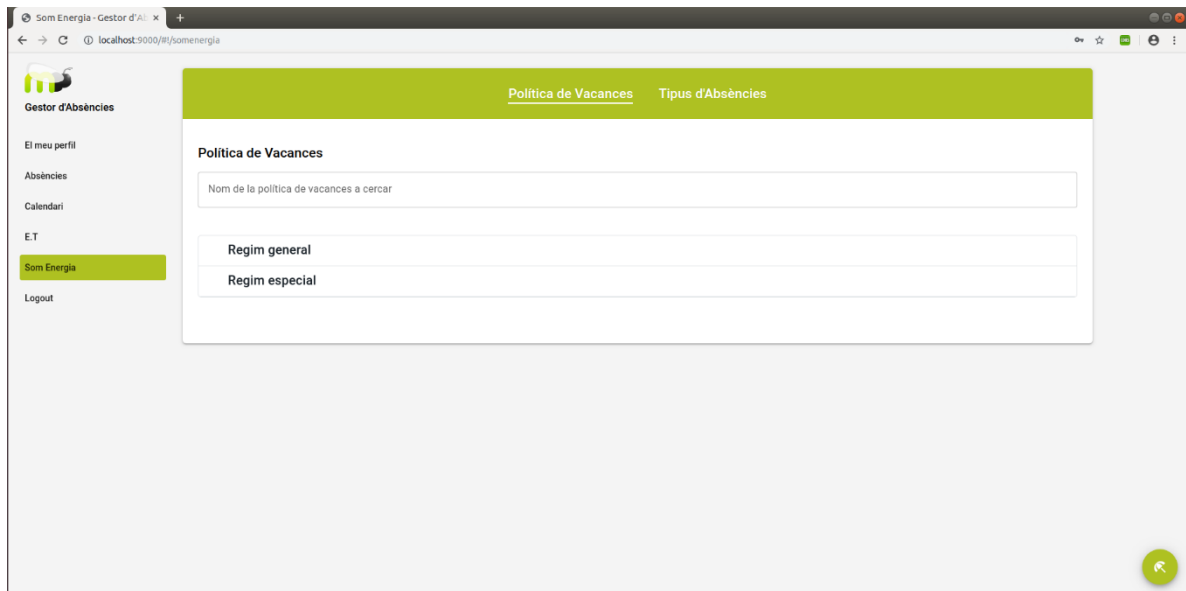
*Formulari de “creació d'un Tipus d'absència”*

Pel que fa als formularis, no s'ha seguit cap disseny concret ja que es va acordar elaborar formularis senzills en els que només apareguessin els camps a omplir i el botó per crear la nova entitat. Cal destacar que, en el cas del tipus d'absència, s'ha facilitat la comprensió del funcionament del model de dades fent que simplement hagin de seleccionar, en quan a la durada, si és mig dia, pot ser indefinit com per exemple les vacances, que el límit no el marcarà el tipus d'absència sinó les vacances que té per gaudir el treballador o en el cas de quin efecte té amb els dies de vacances. S'ha descrit breument el seu efecte per facilitar la comprensió per part de l'administrador de la plataforma.

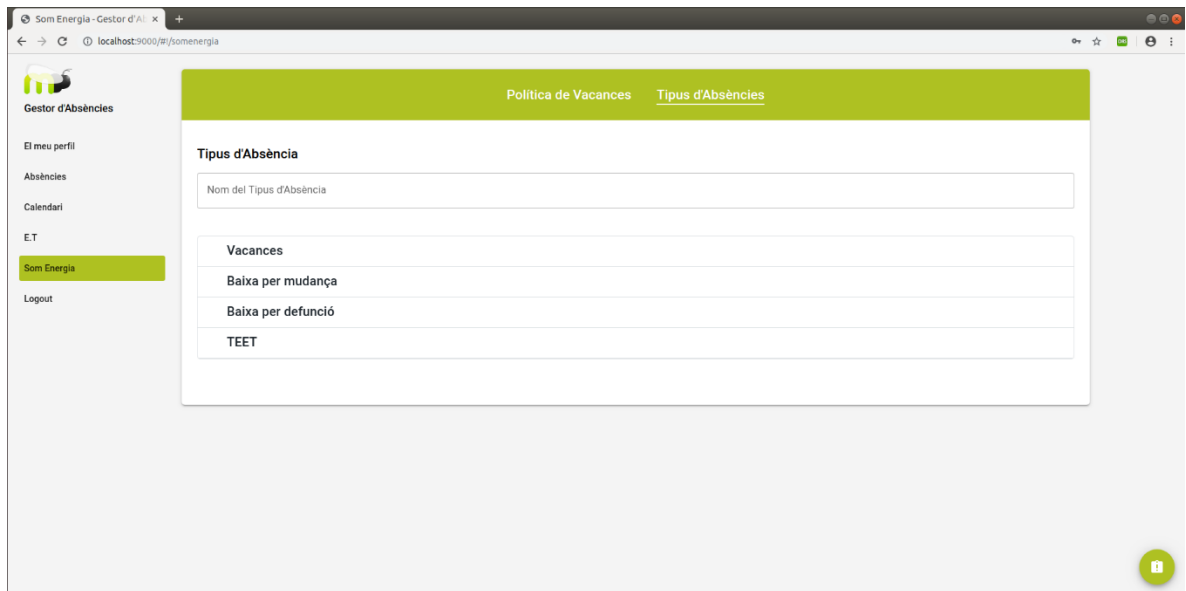
En el cas del formulari de creació d'un equip de treball s'ha preferint únicament assignar el nom de l'equip per així poder afegir usuaris o definir el representant i el referent en la pròpia interfície d'edició de l'entitat.

En el cas de la interfície de Som Energia s'ha optat per seguir l'estipulat en el disseny, de la mateixa forma que s'ha fet amb la interfície “ET”, creant així dues subinterfícies, una pels tipus d'absència i l'altre per les polítiques de vacances.

Al igual que les subinterfícies de “Treballadors” i “Equips de treball”, tenen un filtre per escollir les entitats, cada element de la llista conté un enllaç per poder veure els seus detalls i a la part inferior de la dreta de la interfície hi ha un botó per crear noves entitats.



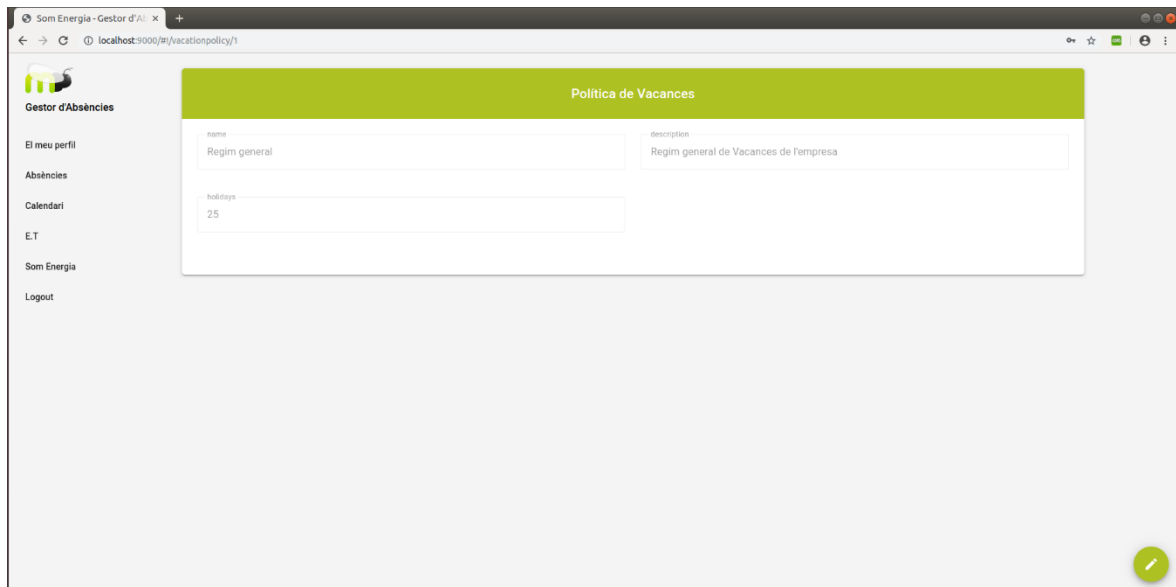
*Subinterfície “Política de Vacances” dins de la interfície “Som Energia”*



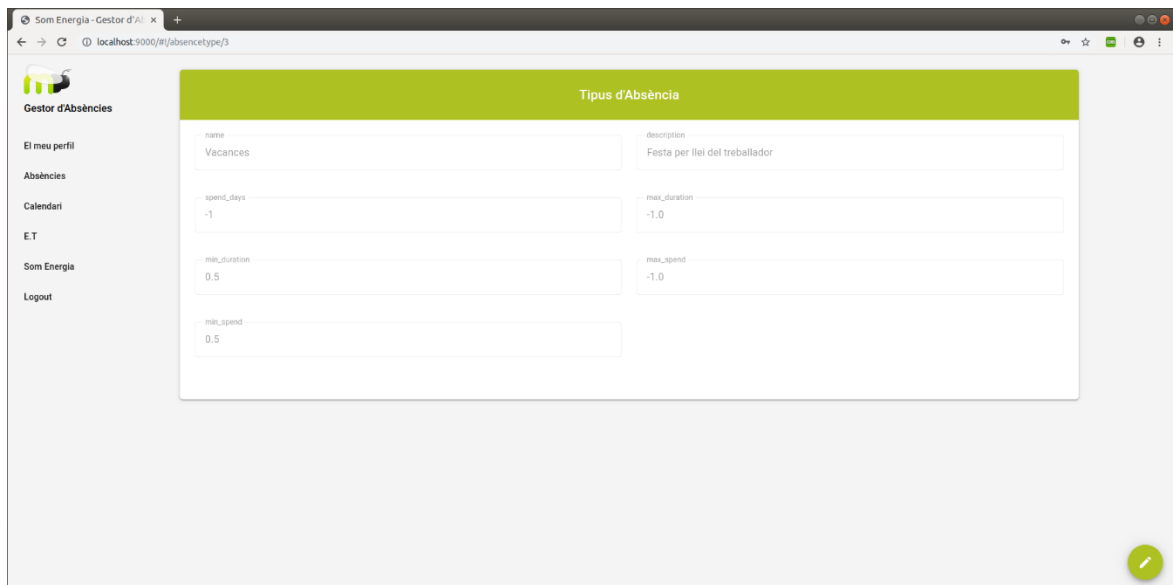
*Subinterfície “Política de Vacances” dins de la interfície “Som Energia”*

Respecte a les interfícies “Tipus d’absències” i “Política de vacances” s’ha seguit el disseny de les interfícies de les entitats que es va elaborar, mostrant els camps de l’entitat en una Card amb la capçalera de color verd i a la part inferior dreta un botó per editar l’entitat que només podran veure els administradors del Gestor d’Absències.





*Interfície “Política de Vacances”*

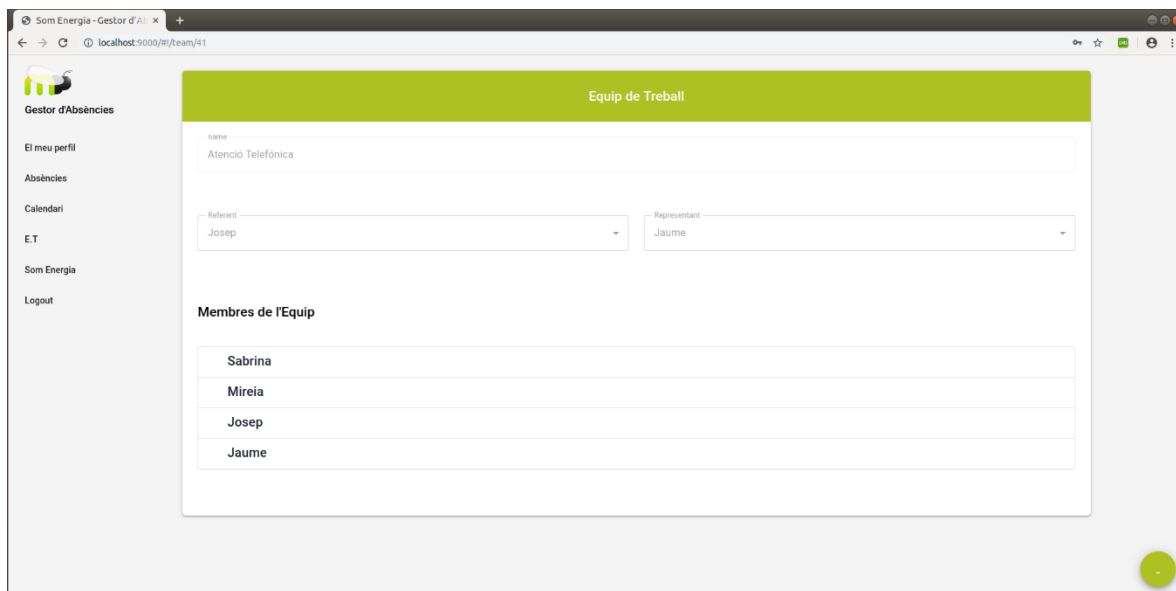


*Interfície “Tipus d'Absència”*

En la interfície “Tipus d'Absència”, la informació dels camps es mostra com a la base de dades, aspecte que no serà un problema per l'administrador de la plataforma ja que ho coneixerà però, per tal de facilitar la comprensió de tots els usuaris, en un futur es mostrarà de la mateixa manera que es fa en el formulari de creació.

Cal recordar, que, tot i que únicament els administradors podran editar aquestes entitats, tots els usuaris de la plataforma les podran visualitzar.

Per últim, tot i que les interfícies que s'han mostrat s'han visualitzat des del perfil de l'administrador, la resta de treballadors tindran petites diferències com l'absència dels botons de creació o edició de les entitats. La interfície que varia més respecte la visualització que podria tenir l'administrador de la plataforma és la interfície "Team", en la qual ja no es mostra el botó d'edició sinó que es pot veure un botó per subscriure's a l'equip de treball o eliminar la subscripció en cas de ja estar-ho.



*Interfície "Team" des del punt de vista de qualsevol usuari de la plataforma que no sigui l'administrador*

En aquest cas, es tracta de l'usuari "Mireia" la qual ja està subscripta a aquest equip de treball i es mostra el botó per eliminar la subscripció a la part inferior dreta de la interfície.

## 10.2. Resultat final del projecte

Cal tenir en compte que, degut a la durada que requereix el desenvolupament del projecte i el temps que s'ha tingut, no s'ha pogut implementar la versió final sinó que s'ha hagut d'implementar una primera versió estable perquè l'equip de recursos humans pugui provar-lo.

Per tal de mostrar visualment el que s'ha realitzat dels 58 requisits que es van plantejar en un principi es farà en forma de taula on, per cada requisit, s'especificarà si s'ha complert o no.

<b>Requisit funcional</b>	<b>Nivell d'assoliment</b>
<b>RF-00.1</b>	Implementat completament
<b>RF-00.2</b>	Implementat completament
<b>RF-01.1</b>	Implementat completament
<b>RF-01.2</b>	Implementat completament
<b>RF-02.1</b>	Implementat completament
<b>RF-02.2</b>	Implementat completament
<b>RF-03.1</b>	Implementat completament
<b>RF-03.2</b>	Implementat completament
<b>RF-04.1</b>	Implementat completament
<b>RF-04.2</b>	Implementat completament
<b>RF-05.1</b>	Implementat completament
<b>RF-05.2</b>	Implementat completament
<b>RF-06.1</b>	Implementat completament
<b>RF-06.2</b>	Implementat completament
<b>RF-07.1</b>	Implementat completament
<b>RF-07.2</b>	Implementat completament
<b>RF-08.1</b>	Implementat completament
<b>RF-08.2</b>	Implementat completament
<b>RF-09.1</b>	Implementat completament
<b>RF-09.2</b>	Implementat completament
<b>RF-10.1</b>	No implementat
<b>RF-10.2</b>	No implementat
<b>RF-11.1</b>	Implementat completament
<b>RF-11.2</b>	Implementat completament
<b>RF-12.1</b>	Implementat completament
<b>RF-12.2</b>	Implementat completament
<b>RF-13.1</b>	Implementat completament
<b>RF-13.2</b>	Implementat completament
<b>RF-14.1</b>	Implementat completament
<b>RF-14.2</b>	Implementat completament
<b>RF-15.1</b>	Requisit Desestimat
<b>RF-15.2</b>	Requisit Desestimat
<b>RF-20</b>	Implementat completament
<b>RF-30.1</b>	Implementat completament

<b>RF-30.2</b>	Implementat completament
<b>RF-31.1</b>	Implementat completament
<b>RF-31.2</b>	Implementat completament
<b>RF-32.1</b>	Implementat completament
<b>RF-32.2</b>	Implementat completament
<b>RF-33.1</b>	Implementat completament
<b>RF-33.2</b>	Implementat completament
<b>RF-34.1</b>	Implementat completament
<b>RF-34.2</b>	Implementat completament
<b>RF-35.1</b>	Implementat completament
<b>RF-35.2</b>	Implementat completament
<b>RF-36.1</b>	No implementat
<b>RF-36.2</b>	No implementat
<b>RF-37</b>	No implementat
<b>RF-38.1</b>	Requisit Desestimat
<b>RF-38.2</b>	Requisit Desestimat
<b>RF-39.1</b>	No implementat
<b>RF-39.2</b>	No implementat
<b>RF-40.1</b>	No implementat
<b>RF-40.2</b>	No implementat
<b>RF-41.1</b>	No implementat
<b>RF-41.2</b>	No implementat
<b>RF-42.1</b>	No implementat
<b>RF-42-2</b>	Implementat completament

D'aquesta manera, es pot veure com, dels 58 requisits funcionals que es van demanar en el moment de la presa de requisits, s'han desenvolupat 42 completament, 4 s'han desestimat des del propi departament de recursos humans degut a la filosofia i mentalitat d'autogestió de la cooperativa i 12 no s'han implementat.

A més dels propis requisits funcionals, durant el desenvolupament, han sorgit requisits com l'atribució de diferents rols en les relacions entre els treballadors i els equips de treball.

Tot i que els requisits no funcionals no siguin tant importants com els funcionals, cal destacar que s'han pogut complir pràcticament tots. Els requisits no funcionals complerts són:

- Les interfícies es poden veure correctament amb Firefox i Google Chrome.

- Es verifiquen el límit de tots els camps que s'han de guardar a la base de dades.
- El Sistema està disponible tots els dies.
- Els logs de l'aplicació es guarden al servidor on està instal·lat.
- El temps de resposta de l'aplicació és sempre inferior a 5 segons.
- Poden utilitzar la plataforma més de 40 persones simultàniament.
- El Sistema pot guardar fins a 100 persones sense tenir repercussions en el seu rendiment.
- Les interfícies del Frontend no contenen errors ortogràfics.
- Les interfícies són estàndards.
- La simplicitat de la plataforma fa que es pugui saber utilitzar en menys de 2 hores.
- El Gestor d'Absències és Open Source i té dos repositoris de Github, un pel Frontend i l'altre pel Backend.
- Les contrasenyes estan encriptades en la base de dades.

Alguns dels requisits no funcionals, com implementar mecanismes que assegurin la integritat de les dades, no s'han realitzat ja que s'hauria de fer amb backups en un entorn de productiu, ja que en un entorn de testing no tindria sentit.

### 10.3. Implantació del projecte a l'entorn de testing

En aquest apartat es parlarà sobre com es va implantar el projecte a l'entorn de testing. Prèviament, cal tenir en compte que la màquina de test és una màquina virtual amb les mateixes característiques, de hardware i software que tindrà la màquina en la que s'implantarà l'aplicatiu finalment, assegurant així que el funcionament i el rendiment és el mateix.

D'aquesta manera, per implantar aquest projecte necessitem els processos que es van descriure durant les decisions del projecte i que, a títol recordatori, són:

- Django
- PostgreSQL
- Wsgi
- Nginx

Tot i que no sigui necessari pel seu funcionament, per tal de gestió els processos del servidor es va configurar el supervisor.

Per sort, la configuració del PostgreSQL no ha sigut necessària ja que només es requeria el procés amb una base de dades i un usuari que tingui permisos d'edició sobre aquesta base de dades.

Cal tenir en compte que al haver parametritzat els paràmetres de l'aplicatiu, s'haurà d'introduir la informació de les credencials de la base de dades en un fitxer de configuració dins del propi projecte.

```
databases:
  develop:
    name: name
    host: host
    port: 5432
    user: user
    password: password
```

*Fitxer d'exemple de configuració del Backend*

Com es pot veure en la imatge, cal establir l'entorn on s'aixeca el servei, en aquest cas testing, juntament amb totes les dades que necessita el procés de Django per connectar-se.

El procés de Django i Wsgi van molt lligats en entorn de producció ja que Wsgi és el servei encarregat d'aixecar la instància de Python que processarà les peticions. D'aquesta manera, hem configurat l'Wsgi per tal que aixequi una instància del Backend amb 4 threads i la configuració que Django ha de tenir en l'entorn de productiu.

```
uwsgi
# Django-related settings
# the base directory (full path)
chdir      = /opt/gestor-absencies
# Django's wsgi file
module     = config.wsgi
# the virtualenv (full path)
home       = /home/somenergia/.local/share/virtualenvs/gestor-absencies-8x0zBd3o

# process-related settings
## master
master     = true
## maximum number of worker processes
processes  = 1
## maximum number of threads
threads    = 4
## the socket (use the full path to be safe)
socket     = /var/run/gestor-absencies/gestor-absencies.sock
## ... with appropriate permissions - may be needed
chmod-socket = 666

# Environment
env        = DJANGO_SETTINGS_MODULE=config.settings.production
# clear environment on exit
vacuum     = true
```

*Fitxer de configuració del Wsgi*

Com es pot veure en la imatge, s'aixeca el servidor amb un entorn virtual per tal de tenir els requisits de llibreries aïllats d'altres possibles projectes i es modifica una variable d'entorn perquè valgui el path de les configuracions de productiu. Aquest fet fa que a l'hora d'aixecar el procés de Python, aquest utilitzi la configuració de l'entorn de productiu.

Com es pot veure en la imatge configuració del Wsgi, en comptes d'escoltar en un port concret rebirà les peticions per un socket, en aquest cas en la ruta “/var/run/gestor-absencies/gestor-absencies.sock”. S'ha decidit fer-ho així ja que augmenta lleugerament el rendiment en comparació amb escoltar un port.

L'últim procés important per tal que arribin les peticions al Backend és l'Nginix, que serà l'encarregat de fer les redireccions de tràfic i de posar un filtre a les peticions que ens arriben.

```
upstream gestor-absencies {
    server unix:/var/run/gestor-absencies/gestor-absencies.sock;
}

server {
    listen 80;
    server_name gestorabsencies-demo.somenergia.local;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name gestorabsencies-demo.somenergia.local;

    large_client_header_buffers 4 32k;
    client_max_body_size 51M;
    charset utf-8;

    index index.html;

    # Frontend
    location / {
        root /var/www/gestor-absencies-ui/current;
        try_files $uri $uri/ /index.html;
    }

    # Backend
    location /absencies {
        include uwsgi_params;
        uwsgi_pass gestor-absencies;
    }

    location /login {
        include uwsgi_params;
        uwsgi_pass gestor-absencies;
    }

    location /admin {
        include uwsgi_params;
        uwsgi_pass gestor-absencies;
    }

    location /static/admin {
        autoindex off;
        alias /home/somenergia/.virtualenvs/gestor-absencies-8x0zBd3o/lib/python3.6/site-packages/django/contrib/admin/static/admin;
    }

    ssl on;

    access_log /var/log/nginx/gestor-absencies-ui_access.log;
    error_log /var/log/nginx/gestor-absencies-ui_error.log;
}
```

### *Fitxer de configuració del Nginx*

Com es pot veure al fitxer de configuració, totes les peticions que es rebien en el port 80 es redireccionaràn automàticament al 443 aprofitant el certificat. De tota la configuració cal destacar

que només entraran les peticions que vagin dirigides a “gestorabsencies-demo.somenergia.local” amb els endpoints en els ports 80 i 443. En el cas de les peticions que vagi adreçades al port 80 seran redirigint-se al 443, en el qual només entraran:

- Peticions GET per obtenir els fitxers statics del Frontend amb l’endpoint “/”.
- Peticions per accedir a l’Admin de Django amb l’endpoint “/admin” i “/static/admin”
- Peticions que es redirigiran a l’API, que comencin amb l’endpoint “/login” per la petició d’iniciar sessió i “/absencies” per tota la resta de peticions.

Cal destacar que aquestes regles només exigeixen el començament de les peticions, pel que la petició “gestorabsencies-demo.somenergia.local/absències/absences/workers” accedirà a l’API, més concretament al llistat de treballadors que tingui la plataforma guardat.

En el propi fitxer de configuració de l’Nginx s’especifiquen els fitxers de logs pels accessos.

Per ultim, es va configurar el supervisor per facilitar la gestió i el control del Backend.

```
[program:gestor_absencies]
directory=/opt/gestor-absencies
command=pipenv run uwsgi --ini config/gestor-absencies.ini
process_name=%(program_name)s
user=somenergia
autorestart=true

# logging config
stdout_logfile=/var/log/gestor_absencies/access.log
redirect_stderr=true
stdout_logfile_maxbytes=1MB
stdout_logfile_backups=15
environment=DJANGO_SETTINGS_MODULE="config.settings.production"
```

*Fitxer de configuració del supervisor*

En aquest fitxer de configuració es pot veure com s’executarà Wsgi que alhora aixecarà el procés de Python. També es pot observar l’usuari amb el que s’executarà i el fitxer on es guardaran els logs de l’aplicació.

#### 10.4. Test de concurrència i rendiment en l’entorn de testing

Per elaborar aquest test s’ha utilitzat la mateixa col·lecció que en l’anterior test, fent així 3 tipus de peticions diferents al Backend. En comptes de retornar l’error 404, en aquest cas, la crida més simple s’ha configurat perquè retorni el valor corresponent.



Degut a l'alt consum de l'aplicació de Postman, s'ha recorregut a una eina anomenada Newman que permet fer les peticions de la mateixa manera que el Postman però amb un consum de recursos inferior ja que es crida des del terminal i el resultat que genera es mostra un cop executades totes les requests.

En voler provar el rendiment de l'aplicatiu, des del Nginx fins el Backend, es va plantejar llançar processos concurrentment on cada un realitzés un seguit de peticions a la plataforma. Per tal d'assegurar el requisit no funcional que la plataforma havia de permetre a 40 usuaris veure les seves ocurrències concurrentment. Es va decidir crear un script en bash que executés concurrentment 40 proves de rendiment. Cada prova de rendiment constava de 300 peticions, 100 de cada tipus com s'ha descrit anteriorment.

A més, un dels ordinadors que compta amb un hardware més potent, va poder realitzar peticions amb el Postman tal i com s'ha descrit anteriorment generant així encara més càrrega al servidor.

Per tal de provar el límit de la plataforma, es va decidir executar aquest script en 4 ordinadors diferents, simulant així 160 usuaris concurrents.

Els resultats de les proves es van redirigir en un fitxer per tal de poder-ho tractar i abstrure les dades que requerísim.

```
#!/bin/bash
for i in $(seq 1 1 40)
do
  newman run https://www.getpostman.com/collections/2cb49d94c3039adb0f4f --folder absences -k -n 100 &
done
```

*Script creat per fer la prova de concurrència i rendiment*

Com es pot veure, aquest script crea 40 processos de Newman en la que s'executa una carpeta concreta de la col·lecció amb la que hem treballat en aquest projecte. Perquè funcioni correctament, prèviament s'ha necessitat exportar la col·lecció per URL públic. El paràmetre -n ens permet dir el número de cops que volem fer de cada petició.

Tot i que es va executar l'script en 4 ordinadors diferents només 2 van redirigir la sortida, aspecte que provoca que, de les casi 50.000 peticions, només s'hagin pogut analitzar 24.000.

Per cada prova de rendiment s'ha generat una taula que mostra les peticions que s'han realitzat, el temps que s'ha tardat en realitzar el test i el temps que ha necessitat el servidor per respondre.

```
somenergia@noti:~$ grep "31/May/2019:13:[0-1]" /var/log/nginx/gestor-absencies-ui_access.log | wc -l
49781
```

*Nombre de peticions rebudes durant la prova de concurrència i rendiment*

Com es pot veure en l'anterior imatge, s'ha revisat que el nombre de peticions concordés amb les que s'havien realitzat des dels diferents ordinadors. La duració total del test ha sigut aproximadament 20 minuts, des de les 13:00 fins les 13:19, d'aquesta manera, el filtratge de logs s'ha dut a terme amb una expressió regular que ha permès veure el número de peticions que el Backend ha respost en aquest 20 minuts.

	executed	failed
iterations	100	0
requests	300	0
test-scripts	0	0
prerequest-scripts	0	0
assertions	0	0
total run duration: 5m 17.7s		
total data received: 67.02KB (approx)		
average response time: 1027ms [min: 4ms, max: 1902ms, s.d.: 373ms]		

*Taula resum d'una prova realitzada amb Newman*

Totes aquestes dades s'han processat amb comandes de bash per tal d'extreure el temps d'espera mínim, màxim i mitjà de les 24.000 peticions.

```
benjami@benjami-VirtualBox:~/Gestor_absencies_memoria$ count=0; total=0; for i in $(grep -rI average stress_result_* | cut -d':' -f3 | cut -d' ' -f2 | cut -d'm' -f1); do ((total = total+$i)); ((count++)); done; echo 'Testos analitzats' $count; echo Total del temps mitjà en rebre resposta $((total/count))
Testos analitzats 80
Total del temps mitjà en rebre resposta 1080
benjami@benjami-VirtualBox:~/Gestor_absencies_memoria$ max=0; for i in $(grep -rI average stress_result_* | cut -d':' -f5 | cut -d'm' -f1 | cut -d's' -f1 | cut -d'.' -f1); do if ((max < $i)); then max=$i; fi; done; echo La request que ha trigat més en ser atesa ha trigat $max ms
La request que ha trigat més en ser atesa ha trigat 1997 ms
```

*Tractament de les dades obtingudes de les 80 proves*

En conèixer que l'estructura del resum de la prova és la mateixa sempre, s'ha utilitzat la comanda grep per buscar el patró "average" dels dos fitxers que contenien el resultat de les proves. Un cop

obtingudes totes les línies en les que es trobava el valor que buscava s'ha anat partint la cadena fins a obtenir el valor màxim o el valor mitjà i processar-ho.

El resultat general de les proves ha sigut:

<b>Temps d'esperat de la petició</b>	<b>Temps (ms)</b>
<b>Màxim</b>	1997
<b>Mitjà</b>	1080

Amb aquestes dades podem confirmar que la plataforma pot gestionar més de 40 usuaris concurrentment i que, al mateix temps, el temps de resposta no supera els 5 segons establerts en els requisits no funcionals.



## 11. Conclusions

Aquest apartat té com a finalitat valorar els objectius que s'han assolit durant aquest Treball de Final de Grau, al igual que la desviació que ha sofert durant el desenvolupament i l'aprenentatge que he obtingut per tal d'analitzar les conclusions del treball.

### 11.1. Valoració dels objectius assolits

Com s'ha vist en l'apartat anterior, s'han pogut complir la majoria de requisits que es van demanar des del departament de recursos humans. Aquest fet provoca que ja es pugui utilitzar la nova plataforma per demanar les vacances, motiu pel qual està en el servidor de testing.

Encara que només quedin per implementar 12 requisits dels 58 inicials, durant el desenvolupament del projecte i especialment durant la planificació de les diferents rondes, des del departament de recursos humans han sorgit nous requisits que podrien tenir més importància que els 12 que resten. D'aquesta manera, abans de seguir completant els requisits que han quedat per fer caldria tancar la primera versió del Gestor d'Absències i planificar la següent versió amb els nous requisits.

### 11.2. Desviació de la planificació

Com s'ha destacat en l'anterior apartat, durant el desenvolupament del projecte han sorgit petits requeriments com per exemple:

- Que una Ocurrencia pugui durar mig dia.
- Que en les relacions entre Treballadors i Equips puguin existir els rols de Representant IT i Referent.
- Que en la interfície web es pugin filtrar el Equips, el Tipus d'Absències i les Polítiques de Vacances a part dels Treballadors pel seu nom.

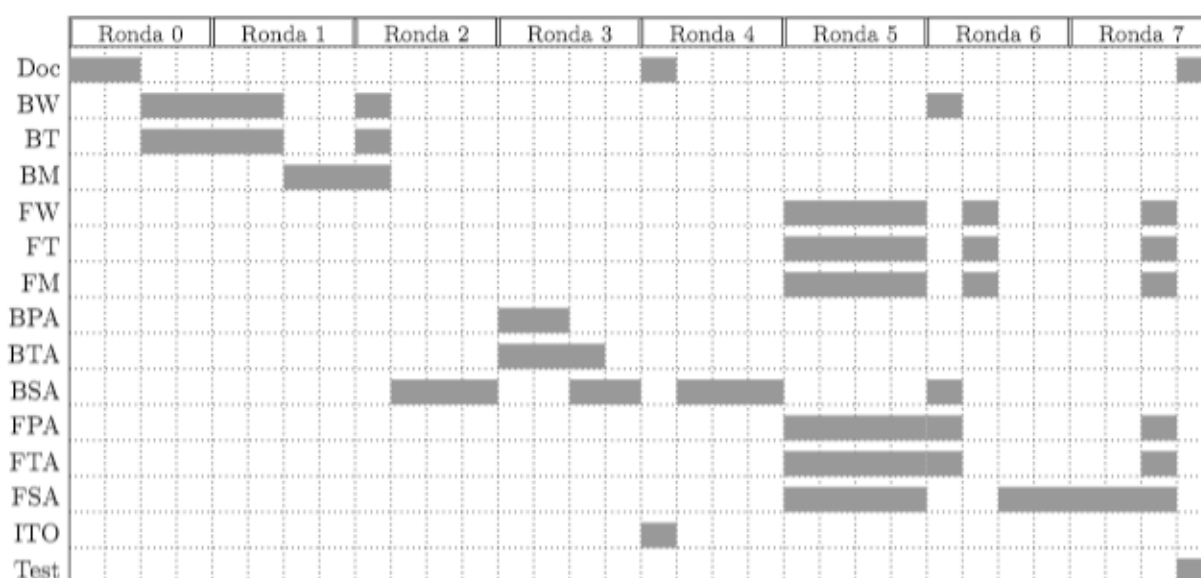
Aquests requisits s'han obtingut durant el desenvolupament del projecte i des del departament de recursos humans se'ls hi ha donat més prioritat que d'altres, és per això que han passat per davant dels requisits recollits a l'inici del projecte.

Tot i que es va planificar el desenvolupament d'aquest projectes en diferents paquets de treballs realitzats en diferents rondes, a l'hora d'implementar el projecte van sorgir petites modificacions en aquesta planificació.

El motiu pel qual es va desviar de la planificació original és principalment que tot i que una planificació ajustada i representada amb un diagrama de Gantt ens pot ajudar a visualitzar i organitzar l'elaboració dels paquets de treball en el temps, aquesta planificació no és més que una valoració del cost en temps que tenen aquests paquets de treball en setmanes o mesos abans de la seva realització, el que provoca que pugui divergir molt de la realitat al trobar-se amb imprevistos o modificació de requisits.

Cal destacar que la complexitat a l'hora de demanar ocurrencies d'absències quan aquestes coincidien amb d'altres ha causat que s'allargués aquesta part del desenvolupament.

El resultat de la planificació final ha sigut el següent:



*Diagrama de Gantt de la planificació final*

On es pot veure la desviació més important és en el fet que la majoria de interfícies del client web s'han pogut desenvolupar al mateix temps degut a l'aprofitament de la implementació d'aquestes per aprendre el llenguatge de JavaScript i el framework de Mithril.

### 11.3. Aprenentatge assolit amb la realització del projecte

El desenvolupament d'aquest projecte m'ha permès assolir molts coneixements i consolidar aspectes vistos durant el Grau en Enginyeria Informàtica. Els coneixements que he adquirit amb aquest desenvolupament han sigut:

- Aprenentatge des de zero de la utilització de Django.

- Aprenentatge des de zero de les llibreries Django REST Framework i JSON Web Token, dues de les llibreries més famoses de Django.
- Consolidar coneixements de Python, entorns virtuals de Python i dependències entre les llibreries.
- Aprenentatge des de zero de Docker.
- Consolidar coneixements de PostgreSQL com les consultes i l'administració d'una base de dades.
- Consolidar la implementació de testos per cobrir els casos d'us que s'han generat dels requisits funcionals.
- Aprenentatge des de zero de la utilització del gestor de paquets npm.
- Aprenentatge des de zero de JavaScript.
- Consolidar els coneixements d'HTML que vaig aprendre durant el Grau.
- Aprenentatge des de zero de Mithril.
- Desenvolupar una aplicació web completament des de zero.
- Aprenentatge de Material Design.
- Aprenentatge des de zero a planificar, analitzar, dissenyar i organitzar un projecte real.
- Consolidació de metodologies de programació Agile.
- Consolidació de l'ús de Github amb diferents branques i Pull Requests.
- Aprenentatge des de zero en la recollida de requeriments i la interacció amb la figura de l'Stakeholder durant el desenvolupament seguit amb metodologia Agile.
- Consolidació de coneixements bàsics en administració de sistemes per muntar una aplicació.
- Consolidació dels coneixements per dissenyar una API REST.

La conclusió final de la realització d'aquest Treball de Final de Grau ha estat molt positiva ja que he pogut aprendre i consolidar molts conceptes que durant el Grau he après i molts d'altres que no. També ha resultat molt profitosa per mi i per l'empresa ja que durant la realització del projecte, m'he format en tecnologies necessàries i desconegudes per IT fet que, juntament amb l'augment de jornada, potenciarà l'equip per realitzar futurs desenvolupaments amb aquestes tecnologies.





## 12. Treball futur

Tot i que el projecte ja pugui ser provat pel departament de recursos humans, les següents passes a seguir són:

1. Poder definir el calendari de l'any, amb totes les vacances anuals per tots els treballadors.
2. Pujar l'aplicatiu a productiu amb una direcció IP pública perquè tots els treballadors puguin accedir al seu calendari de vacances encara que no estiguin a la feina.
3. Fer més fàcil d'entendre la interfície de visualització d'un tipus d'absència.
4. Mostrar missatges informatius en el Frontend en cas d'error.
5. Establir els dies de la setmana que cada treballador està a l'oficina.
6. Permetre que l'usuari canviï la contrasenya del seu perfil des de la interfície web.
7. Establir la pàgina d'Absències com la pàgina d'aterratge a la plataforma.

En el cas de la primera tasca, tot i que actualment amb l'API que està implementada ja es podria implementar en el Frontend un bucle que per cada treballador creï una l'ocurrència determinada, és preferible que existeixi una única crida per realitzar aquest acció des del Backend i que sigui tractada com una transacció, facilitant així el possible tractament d'errors en cas que en un treballador no pogués crear l'ocurrència.

Pel que fa a la segona tasca, al tenir un administrador de sistemes a la cooperativa, segurament la realitzaria ell.

La tercera tasca, tot i que no sigui la més prioritària, cal realitzar-la abans de posar a productiu l'aplicatiu ja que, tot i que en el departament de recursos humans puguin entendre les dades mostrades en la interfície, cal fer-ho comprensible per tots els usuaris de la plataforma. Per tal de fer-ho s'agafarà com a referència el formulari de creació d'un tipus d'absència, mostrant així camps de text en els que s'expliqui la durada que pot tenir l'absència i l'efecte que té en els dies de vacances.

La quarta tasca serà una de les primeres que es realitzarà per tal que el departament del recursos pugui entendre els errors que es donen en la introducció de paràmetres.

Tot i que aquestes són les principals tasques a realitzar, hi ha altres tasques que han sorgit durant el desenvolupament d'aquest projecte que s'haurien d'implementar, aquestes tasques són:

1. Al haver entrat en vigor una llei que obliga a tenir un control horari dels treballadors de l'empresa, a Som Energia s'aprofitarà per implementar aquesta nova eina de gestió d'horaris dins del propi Gestor d'Absències, connectant també així el tipus de jornada o dies setmanals treballats amb les vacances.
2. Poder importar el calendari d'un treballador a altres plataformes com google calendar.

3. Poder afegir la càrrega que cada treballador té de telèfon setmanal perquè el Tomàtic, el generador de graelles de telèfon, pugui extreure aquesta informació juntament amb la de quins treballadors estan a l'oficina per balancejar la càrrega de telèfon.
4. Afegir una interfície en el Frontend perquè un usuari pugui canviar la seva contrasenya.
5. Afegir la data d'incorporació a Som Energia en el perfil d'un usuari.
6. Fer que l'administrador de la plataforma pugui escollir el color que tindrà un tipus d'absència en el calendari.

Totes aquestes tasques, en comptes de realitzar-se directament, hauran de ser prioritzades pel departament de recursos humans, ja que al finalitzar aquest Treball de Final de Grau deixaré de dedicar la meitat de la meva jornada a aquest projecte i el Gestor d'Absències haurà de competir amb els altres projectes que s'estan desenvolupant dins la cooperativa.

Una altra tasca, en aquest cas promoguda per IT serà refactoritzar constantment el codi de manera cíclica per tal de fer-lo cada cop més comprensible, parametrizable i mantenible. A més es vol realitzar abans de concloure la durada d'aquest Treball de Final de Grau la parametrització dels testos del Backend.

## 13. Webgrafia

Aquest apartat recull totes les fonts d'informació que s'han utilitzat per fer el projecte o la memòria d'aquest.

### 13.1. Webgrafia utilitzada pel Backend

Django Software Foundation. 2019. Models. Recuperat el dia 15/03/2019 a <https://docs.djangoproject.com/en/2.2/topics/db/models/>

Django Software Foundation. 2019. Why Django?. Recuperat el dia 15/03/2019 a <https://www.djangoproject.com/start/overview/>

Django Software Foundation. 2019. Model field reference. Recuperat el dia 15/03/2019 a <https://docs.djangoproject.com/en/2.2/ref/models/fields/>

Django Software Foundation. 2019. Making quèries. Recuperat el dia 15/03/2019 a <https://docs.djangoproject.com/en/2.2/topics/db/queries/>

Django Software Foundation. 2019. QuerySet API reference. Recuperat el dia 15/03/2019 a <https://docs.djangoproject.com/en/2.2/ref/models/querysets/>

Django Software Foundation. 2019. Migrations. Recuperat el dia 15/03/2019 a <https://docs.djangoproject.com/en/2.2/topics/migrations/>

Django Software Foundation. 2019. The Django admin site. Recuperat el dia 15/03/2019 a <https://docs.djangoproject.com/en/2.2/ref/contrib/admin/>

Encode juntament amb contribuïdors individuals. 2019. Class-based Views. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/views/>

Encode juntament amb contribuïdors individuals. 2019. ViewSets. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/viewsets/>

Encode juntament amb contribuïdors individuals. 2019. Routers. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/routers/>

Encode juntament amb contribuïdors individuals. 2019. Serializers. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/serializers/>

Encode juntament amb contribuïdors individuals. 2019. Serializer fields. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/fields/>

Encode juntament amb contribuïdors individuals. 2019. Validators. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/validators/>

Encode juntament amb contribuïdors individuals. 2019. Authentication. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/authentication/>

Encode juntament amb contribuïdors individuals. 2019. Permissions. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/permissions/>

Encode juntament amb contribuïdors individuals. 2019. Filtering. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/filtering/>

Encode juntament amb contribuïdors individuals. 2019. Testing. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/api-guide/testing/>

Encode juntament amb contribuïdors individuals. 2019. Documenting your API. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/topics/documenting-your-api/>

Encode juntament amb contribuïdors individuals. 2019. Quickstart. Recuperat el dia 15/03/2019 a <https://www.django-rest-framework.org/tutorial/quickstart/>

Blimp juntament amb contribuïdors individuals. 2017. REST framework JWT Auth. Recuperat el dia 15/03/2019 a <https://getblimp.github.io/django-rest-framework-jwt/>

Docker Inc. 2019. What is a Container?. Recuperat el dia 15/03/2019 a <https://www.docker.com/resources/what-container>

Docker Inc. 2019. Get Started. Recuperat el dia 15/03/2019 a <https://docs.docker.com/get-started/>

Docker Inc. 2019. Use volumes. Recuperat el dia 15/03/2019 a <https://docs.docker.com/storage/volumes/>

Docker Inc. 2019. Quickstart: Compose and Django. Recuperat el dia 15/03/2019 a <https://docs.docker.com/compose/django/>

Docker Inc. 2019. Postgres. Recuperat el dia 15/03/2019 a <https://docs.docker.com/samples/library/postgres/>

Docker Inc. 2019. Python. Recuperat el dia 15/03/2019 a <https://docs.docker.com/samples/library/python/>

Django Software Foundation. 2019. Django. Recuperat el dia 15/03/2019 a <https://github.com/django/django>

Encode juntament amb contribuïdors individuals. 2019. Django-rest-framework. Recuperat el dia 15/03/2019 a <https://github.com/encode/django-rest-framework/tree/master>

Blimp juntament amb contribuïdors individuals. 2017. Django-rest-framework-jwt. Recuperat el dia 15/03/2019 a <https://github.com/GetBlimp/django-rest-framework-jwt>

Django girls. 2018. Instalacion de Django. Recuperat el dia 15/03/2019 a [https://tutorial.djangogirls.org/es/django\\_installation/](https://tutorial.djangogirls.org/es/django_installation/)

Django girls. 2018. ¡Tu primer proyecto en Djago!. Recuperat el dia 15/03/2019 a [https://tutorial.djangogirls.org/es/django\\_start\\_project/](https://tutorial.djangogirls.org/es/django_start_project/)

OpenAPI Initiative. 2019. OpenAPI. Recuperat el dia 16/03/2019 a <https://www.openapis.org/>

OpenAPI Initiative juntament amb contribuïdors individuals. 2019. OpenAPI-Specification. Recuperat el dia 15/03/2019 a <https://github.com/OAI/OpenAPI-Specification>

Postman Inc. 2019. Command line integration with Newman. Recuperat el dia 30/05/2019 a [https://learning.getpostman.com/docs/postman/collection\\_runs/command\\_line\\_integration\\_with\\_newman/](https://learning.getpostman.com/docs/postman/collection_runs/command_line_integration_with_newman/)

Adam Wiggins. 2017. The Twelve-Factor App. Recuperat el dia 15/03/2019 a <https://12factor.net/es/>

## 13.2. Webgrafia utilitzada pel Frontend

Leo Horie. 2019. Mithril Introduction. Recuperat el dia 15/03/2019 a <https://mithril.js.org/>

Leo Horie. 2019. Mithril Installation. Recuperat el dia 15/03/2019 a <https://mithril.js.org/installation.html>

Leo Horie. 2019. Virtual DOM nodes. Recuperat el dia 15/03/2019 a <https://mithril.js.org/vnodes.html>

Leo Horie. 2019. Components. Recuperat el dia 15/03/2019 a <https://mithril.js.org/components.html>

Leo Horis. 2019. Lifecycle methods. Recuperat el dia 15/03/2019 a <https://mithril.js.org/lifecycle-methods.html>

Som Energia. 2019. Webforms-mithril. Recuperat el dia 15/03/2019 a <https://github.com/Som-Energia/webforms-mithril>

Google. 2019. Material Design Introduction. Recuperat el dia 20/02/2019 a <https://material.io/design/introduction/#>

Google. 2019. Buttons in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/buttons/>

Google. 2019. Cards in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/cards/>

Google. 2019. Dialogs in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/dialogs/>

Google. 2019. Lists in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/lists/>

Google. 2019. Drawers in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/drawers/>

Google. 2019. Text fields in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/input-controls/text-field/>

Google. 2019. Cards in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/design/components/cards.html>

Google. 2019. DOM in Material Design Web Components. Recuperat el dia 20/02/2019 a <https://material.io/develop/web/components/dom/>

Google. 2019. Checkboxes in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/input-controls/checkboxes/>

Google. 2019. Select Menus in Material Design Web Components. Recuperat el dia 20/05/2019 a <https://material.io/develop/web/components/input-controls/select-menus/>



## 14. Annexos

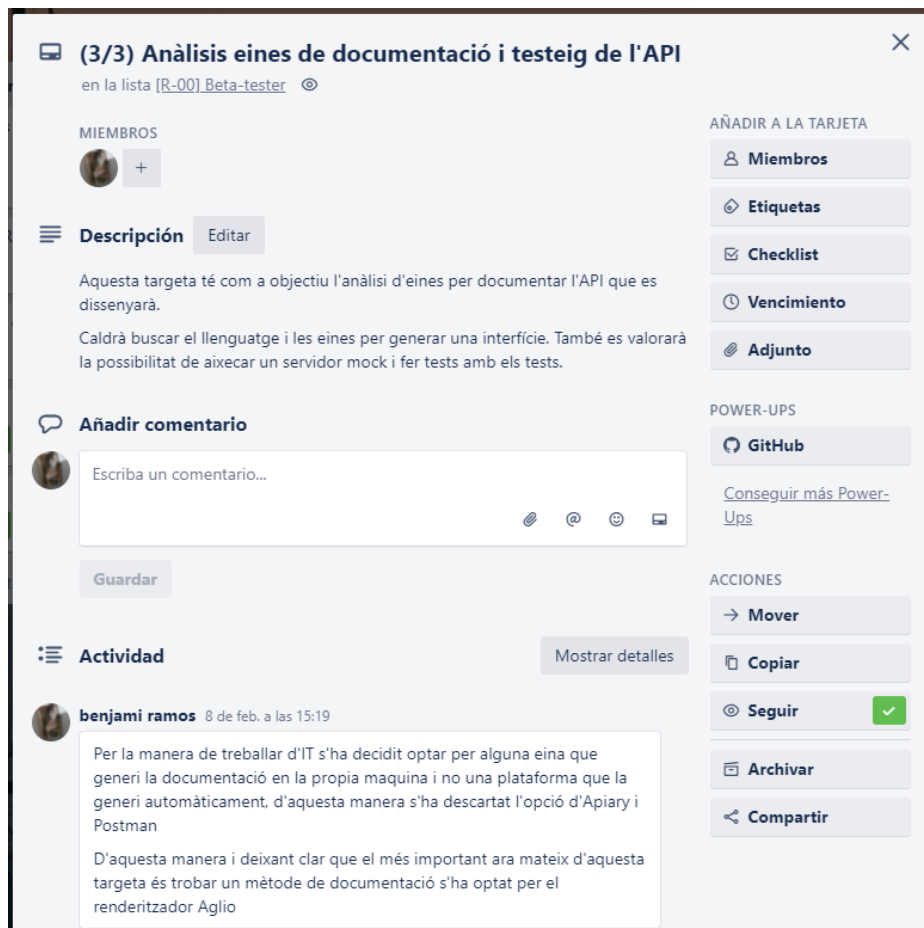
Aquest apartat té com a objectiu incloure tota aquella informació relacionada amb el Treball de Final de Grau realitzat com la planificació final detallada o la documentació de l'API que s'ha seguit.

### 14.1. Planificació final detallada

A continuació es detalla, amb més exactitud, la planificació que s'ha dut a terme durant el desenvolupament del Gestor d'Absències.

La planificació s'ha fet amb l'ajuda de Trello, un gestor de taulers que permet definir diferents columnes per representar diferents estats que poden tenir les tasques i targetes per representar cada una de les tasques. Com s'ha dit durant el treball, en el departament d'IT de Som Energia es treballa amb Mitjos Dies com a unitat de mesura del temps per les tasques, fent que cada Mig Dia equivalgui a 4 hores.

Les targetes tenen el següent aspecte:



*Targeta del Trello que representa una tasca*

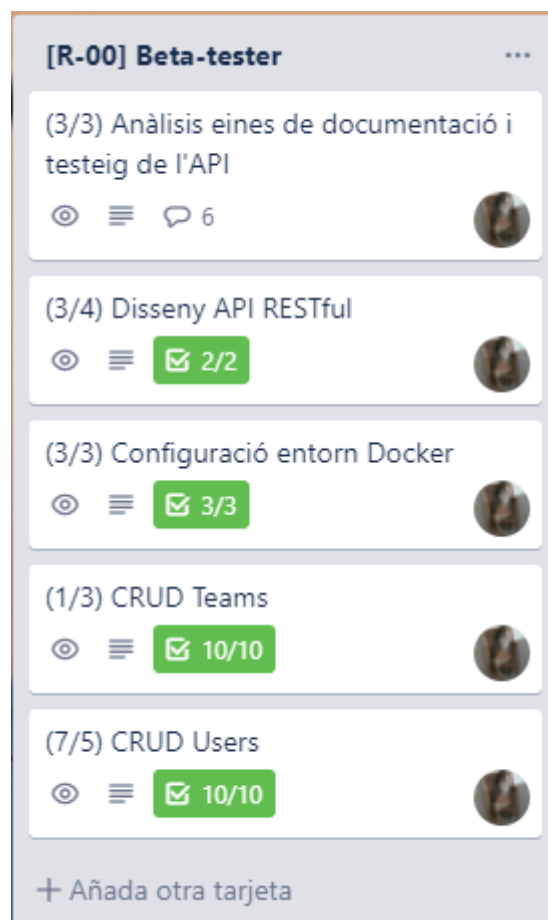
Cada targeta del Trello té un títol, el qual ha de ser breu i descriptiu, de la tasca que pretén representar juntament amb el valor del cost que ha tingut i el valor que es creia que tindria. Aquests valors es troben abans del títol i tenen el format (S/E), on S representa els Mitjos Dies que ha costat realitzar aquesta tasca i E els Mitjos Dies que es creia que costaria realitzar-la. Aquesta manera de treballar s'utilitza ja que gràcies a l'extensió Trello Plus es poden analitzar aquests números de manera automàtica.

Per cada targeta també es pot veure la persona que la realitzarà, en aquest cas només hi sóc jo i la descripció del que s'haurà de fer amb aquesta tasca. Cada targeta també pot contenir comentaris, Checklists, enllaços a GitHub entre altres.

Cada ronda, a més, té un nom que pot ser descriptiu o que pot servir als desenvolupadors per saber el que es va realitzar.

#### 14.1.1. Planificació Ronda [R-00] Beta-tester

Les tasques realitzades durant aquesta ronda van ser:



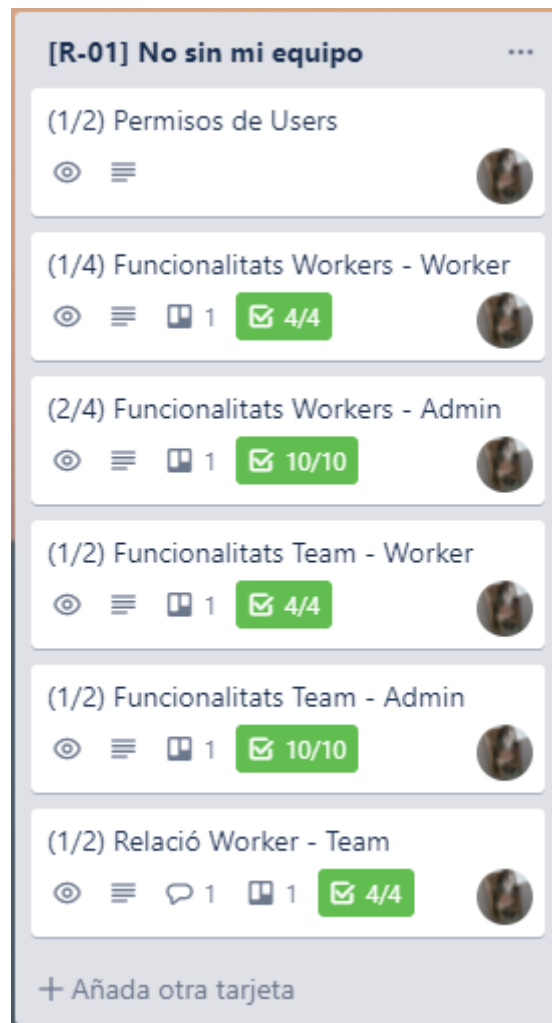
*Tasques realitzades durant [R-00] Beta-tester*

A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(3/3) Anàlisis eines de documentació i testeig de l'API**
  - **Descripció:** Aquesta targeta té com a objectiu l'anàlisi d'eines per documentar l'API que es dissenyarà. Caldrà buscar el llenguatge i les eines per generar una interfície. També es valorarà la possibilitat de aixecar un servidor mock i fer tests amb els tests.
- **(3/4) Disseny API RESTful**
  - **Descripció:** Aquesta targeta té com a objectiu analitzar i dissenyar l'API del Gestor d'absències.
- **(3/3) Configuració entorn Docker**
  - **Descripció:** Aquesta targeta consisteix en la creació dels contenidors de docker (postgres i django).
- **(1/3) CRUD Teams**
  - **Descripció:** Crear l'estructura necessària per poder fer les crides CRUD de Teams.
  - **Checklist:**
    - Create
    - Read
    - Update
    - Delete
    - List
- **(7/5) CRUD Users**
  - **Descripció:** Crear l'estructura necessària per poder fer les crides CRUD de Users.
  - **Checklist:**
    - Create
    - Read
    - Update
    - Delete
    - List

### 14.1.2. Planificació Ronda [R-01] No sin mi equipo

Les tasques realitzades durant aquesta ronda van ser:



*Tasques realitzades durant [R-01] No sin mi equipo*

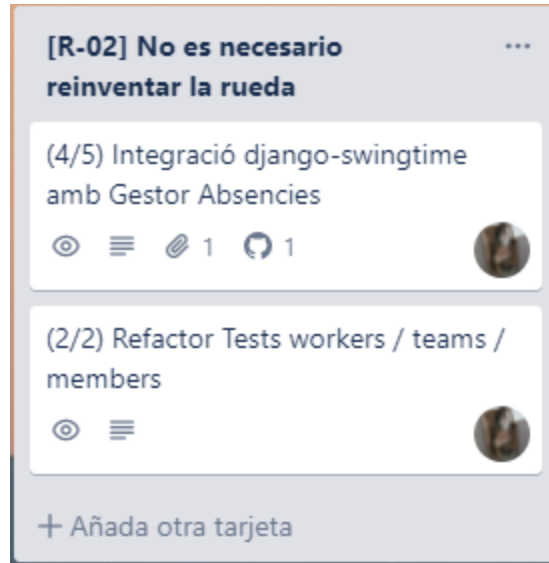
A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(1/2) Permisos de Users**
  - **Descripció:** Aquest targeta consisteix en la anàlisi de com gestionar els permisos amb Django, DRF i l'autenticació amb JWT i la seva implementació segons el document que vam acordar.
- **(1/4) Funcionalitats Workers - Worker**
  - **Descripció:** Desenvolupar les funcionalitats dels Workers des del punt de vista dels Workers.

- **Els Workers poden:**
  - Veure altres Workers
- **(2/4) Funcionalitats Workers - Admin**
  - **Descripció:** Desenvolupar les funcionalitats dels Workers des del punt de vista dels Admins.
  - **Els Admins poden:**
    - Veure altres Workers
  - Crear un Worker
  - Eliminar un Worker
  - Modificar un Worker
- **(1/2) Funcionalitats Team - Worker**
  - **Descripció:** Desenvolupar les funcionalitats dels Teams des del punt de vista dels Workers.
  - **Els Workers poden:**
    - Veure els Teams
- **(1/2) Funcionalitats Team - Admin**
  - **Descripció:** Desenvolupar les funcionalitats dels Teams des del punt de vista dels Admins.
  - **Els Admins poden:**
    - Veure la llista de Teams
    - Veure un Team
    - Crear un Team
    - Eliminar un Team
    - Modificar un Team
- **(1/2) Relació Worker - Team**
  - **Descripció:** Aquesta targeta consisteix en crear la relació entre Worker i Team (relació MN que contindrà si es referent o és representant).

### 14.1.3. Planificació Ronda [R-02] No es necesacio reinventar la rueda

Les tasques realitzades durant aquesta ronda van ser:



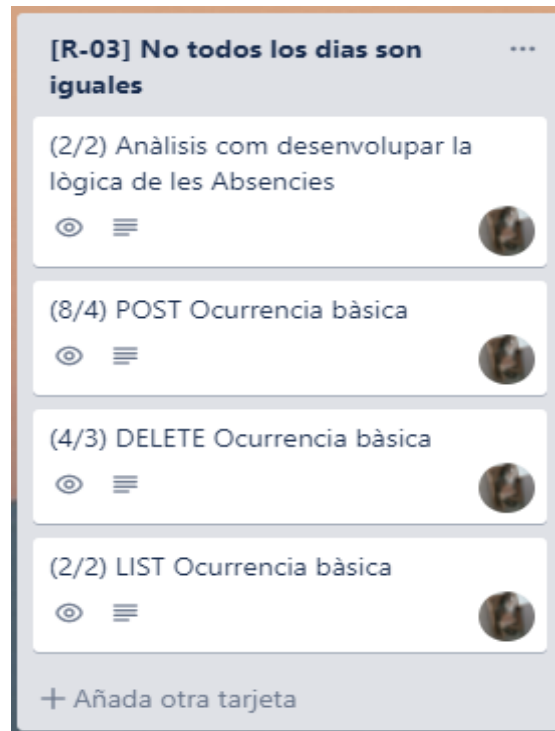
*Tasques realitzades durant [R-02] No es necesacio reinventar la rueda*

A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(4/5) Integració django-swingtime amb Gestor Absències**
  - **Descripció:** Aquesta targeta consisteix en l'estudi, anàlisi i integració del mòdul django-swingtime al gestor d'absències.
- **(2/2) Refactor Tests workers / teams / members**
  - **Descripció:** Aquesta targeta té com a objectiu aplicar un refactor als tests creats a la R-01 per tal de no replicar codi.

#### 14.1.4. Planificació Ronda [R-03] No todos los dias son iguales

Les tasques realitzades durant aquesta ronda van ser:



*Tasques realitzades durant [R-03] No todos los dias son iguales*

A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(2/2) Anàlisis com desenvolupar la lògica de les Absències**
  - **Descripció:** Aquesta targeta consistirà en fer la anàlisis de com fer implementar la lògica de les Absències i com estructurar les targetes per la seva implementació.
- **(8/4) POST Ocurrencia bàsica**
  - **Descripció:** Aquesta targeta contempla una primera implementació del POST d'Ocurrencia amb les característiques lògiques bàsiques.
- **(4/3) DELETE Ocurrencia bàsica**
  - **Descripció:** Aquesta targeta contempla una primera implementació del DELETE d'Ocurrencia amb les característiques lògiques bàsiques.
- **(2/2) LIST Ocurrencia bàsica**

- **Descripció:** Aquesta targeta contempla una primera implementació del LIST d'Ocurrencia amb les característiques lògiques bàsiques.

#### 14.1.5. Planificació Ronda [R-04] El final de un ciclo y el inicio del siguiente

Les tasques realitzades durant aquesta ronda van ser:



*Tasques realitzades durant [R-04] El final de un ciclo y el inicio del siguiente*

A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(1/1) Anàlisis mock's Frontend**
  - **Descripció:** Aquesta targeta consistirà en fer la anàlisis de com estructurar les interfícies del Frontend. No inclou fer cap Mock amb cap plataforma, inclou com s'estructuraran les principals interfícies.
- **(1/1) Setup entorn Frontend**



- **Descripció:** Aquesta targeta només engloba la creació de l'entorn necessari pel desenvolupament de la part Frontend (webpack, npm...).
- **(7/4) Casos concrets i més específics d'ocurrències**
  - **Descripció:** Aquesta targeta té com a objectiu implementar casos d'us més específics per la sol·licitud d'absències com el solapament de les absències o els permisos.
    - Un Worker només pot crear i eliminar les seves ocurrències (serà tractat en la targeta de Testejar i Posada a punt).
    - Cal poder filtrar la llista d'absències per persona / equip / data ... (el que es pugui).
    - Cal tractar les vacances anuals (1 Gener, 1 Novembre, ...) com una altre SomEnergiaAbsenceType.
    - Com fer-ho quan coincideix dues Ocurrències al mateix temps.
- **(5/6) Testejar i Posada a punt de la primera versió del Backend**
  - **Descripció:** Aquesta targeta engloba totes aquelles tasques de programació que puguin sorgir com a necessitat un cop augmentat el test coverage. Aquesta targeta té com a objectiu cobrir tots els casos d'us i els "casos de no us" del Backend. Un exemple d'aquests son el que poden i no poden fer els diferents tipus d'usuaris o la creació o l'eliminació en cascada de les diferents entitats entre si.

#### 14.1.6. Planificació Ronda [R-05] Lo importante es el interior

Les tasques realitzades durant aquesta ronda van ser:



*Tasques realitzades durant [R-05] Lo importante es el interior*

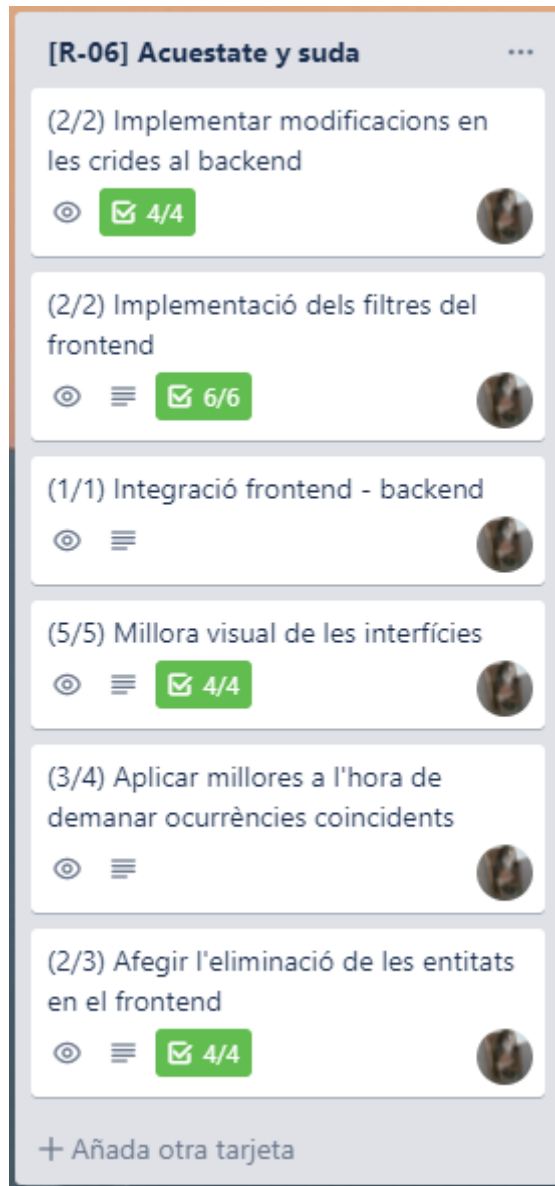
A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(6/7) Primera versió funcional de la visualització, edició i eliminació de les entitats del Frontend**
  - **Descripció:** Aquesta targeta té com a objectiu la implementació de totes les interfícies de visualització i les d'edició Frontend sense cap estil visual concret, simplement amb l'estructura bàsica per poder-ho enviar al Backend. Tot i que inclogui les crides GET, no inclou les crides PUT per tal de modificar les entitats del Backend.
    - Interfície "El meu perfil" i d'edició de "El meu perfil"
    - Interfície de "Absències"
    - Interfície de "Calendari"
    - Interfície de "Usuaris"
    - Interfície de "Equips de treball"
    - Interfície de "Política de Vacances"
    - Interfície de "Tipus d'absències"
    - Interfície de "Log in"

- **(6/5) Primera versió funcional dels formularis del Frontend**
  - **Descripció:** Aquesta targeta té com a objectiu la implementació de tots els formularis del Frontend sense cap estil visual concret, simplement amb els camps necessaris per poder-ho enviar al Backend. Tampoc inclou la integració d'aquest amb el Backend.
    - Formulari creació ocurrència
    - Formulari creació Tipus d'absència
    - Formulari creació Política de Vacances
    - Formulari creació d'usuari
    - Formulari creació Equip
  
- **(6/4) Primera versió funcional dels llistats del Frontend**
  - **Descripció:** Aquesta targeta té com a objectiu la implementació de tots els llistats d'entitats sense cap estil visual concret. En aquesta targeta s'inclou les crides al Backend genèriques, sense cap filtre aplicat per poder rebre les dades necessàries.
    - Llistat d'absències anuals del propi usuari.
    - Llistat dels treballadors de l'oficina amb les absències que tenen cada un dells.
    - Llistat dels treballadors de l'oficina.
    - Llistats dels equips de l'oficina.
    - Llistat de les polítiques de vacances de l'oficina
    - Llistat dels tipus d'absències que pot haver-hi a l'oficina

### 14.1.7. Planificació Ronda [R-06] Acuestate y suda

Les tasques realitzades durant aquesta ronda van ser:



*Tasques realitzades durant [R-06] Acuestate y suda*

A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(2/2) Implementar modificacions en les crides al Backend**

- **Descripció:** Tot i que en un principi es va dissenyar l'API amb totes les crides necessàries perquè el client web pogués obtenir tota la informació que necessites, durant la implementació del Frontend s'ha vist que es necessita retornar informació com si l'usuari és administrador o no. Aquestes petites tasques es tractaran com una targeta.
  - Modificar endpoint api-token-auth per log in.
  - Modificar la resposta de log in perquè retorni l'id de l'usuari i si és administrador de la plataforma.
  - Modificar la resposta GET de /workers/:id per tal que retorni els dies de vacances que té.
  - Afegir un filtre de dates per les ocurrències.
- **(1/1) Integració Frontend - Backend**
  - **Descripció:** Aquesta targeta té com a finalitat la integració del Frontend existent amb el Backend. Al haver-se definit molt bé l'API des de un bon inici del projecte aquesta integració hauria de ser senzilla i ràpida.
- **(5/5) Millora visual de les interfícies**
  - **Descripció:** Aquesta targeta té com a objectiu aplicar una millora visual a gran part de la plataforma.
    - Millora general de totes les interfícies.
    - Interfície Absències / apartat ocurrències definitiu.
    - Interfície Som Energia definitiva.
    - Interfície ET definitiva.
- **(3/4) Aplicar millores a l'hora de demanar ocurrències coincidents**
  - **Descripció:** Durant la ronda anterior es van trobar alguns errors al hora de crear ocurrències que coincidint en diferents dies. D'aquesta manera, en aquesta ronda es volen arreglar els errors trobats en la ronda anterior permeten demanar ocurrències en el mateix període sense que hi hagin errors.
- **(2/3) Afegir l'eliminació de les entitats en el Frontend**
  - **Descripció:** Aquesta targeta té l'objectiu d'implementar l'eliminació de les diferents entitats de la plataforma.
    - Eliminar usuaris.
    - Eliminar equips.
    - Eliminar polítiques de vacances.
    - Eliminar tipus d'absències.

### 14.1.8. Planificació Ronda [R-07] El orgullo del camino recorrido

Les tasques realitzades durant aquesta ronda van ser:

[R-07] El orgullo del camino recorrido

- (1/1) Afegir l'eliminació de les ocurrències i els membres d'equips en el frontend
- (3/3) Implementar tota la lògica de la pantalla Team
- (1/1) Verificació de permisos de l'usuari en la interfície
- (2/2) Maquetació del calendari
- (2/2) Maquetació minicalendari
- (2/2) Posada a punt de tot el client Web
- (1/1) Anàlisi integració del Gestor d'absències en l'entorn de Testing
- (1/1) Arreglar CORS
- (3/1) Implantació del Gestor d'absències a Testing
- (2/1) Proves de rendiment / càrrega del backend
- (2/2) Parametrització dels tests d'ocurrències

+ Añada otra tarjeta

*Tasques realitzades durant [R-07] El orgullo del camino recorrido*

A continuació passaré a explicar amb més detall tant la descripció de les tasques com els possibles Checklist en cas d'haver-hi:

- **(1/1) Afegir l'eliminació de les ocurrències i els membres d'equips en el Frontend**
  - **Descripció:** Degut a que durant l'anterior ronda no es va poder afegir totes les eliminacions d'entitats en el Frontend, han quedat per acabar de fer dues, ocurrències i les subscripcions del equips. Aquesta targeta té com a objectiu crear l'espai en el Frontend per tal de permetre-ho.
- **(3/3) Implementar tota la lògica de la interfície Team**
  - **Descripció:** Aquesta targeta té com a objectiu implementar la lògica de la interfície d'equips de Treball del Frontend.
    - Possibilitat d'eliminació de l'equip per part de l'Administrador.
    - Possibilitat d'afegir un treballador al equip per part de l'Administrador.
    - Possibilitat de modificar el referent i el representant per part de l'Administrador.
    - Possibilitat de treure del grup a un treballador per part de l'Administrador
    - Possibilitat d'afegir-se al Equip per part del Treballador.
    - Possibilitat de sortir del Equip per part del Treballador.
- **(1/1) Verificació de permisos de l'usuari en la interfície**
  - **Descripció:** Aquesta targeta té com a objectiu d'integrar els permisos que té l'usuari en el Backend amb les interfícies que pot visualitzar o editar en el Frontend.
- **(2/2) Maquetació del calendari**
  - **Descripció:** Aquesta targeta té com a objectiu la millora visual de la interfície Calendari amb:
    - La visualització del selector del filtre conforme l'estil establert en la plataforma.
    - La visualització dels diferents dies del mes segons la durada d'aquest.
    - La visualització dels caps de setmana en el calendari.
    - La visualització amb diferents colors de treball de les absències, en cas de mig dia caldrà poder-ho visualitzar com a mig quadre.
    - Visualitzar de manera diferent el dia actual.
    - Aplicar ajudes visuals per seguir un usuari en la taula del calendari.
    - En cas de passar el ratolí que s'indiqui el nom de l'absència. ex: Vacances.

- **(2/2) Maquetació minicalendari**
  - **Descripció:** Aquesta targeta té com a objectiu la maquetació del minicalendari de la interfície d'Absències, aquest minicalendari ha de tenir:
    - S'ha de poder veure el calendari anual sense els números, únicament amb uns punts indicatius per cada dia del mes.
    - En cas d'haver-hi absències, aquestes s'hauran de marca en el minicalendari.
    - Al passar el ratolí per sobre marcarà el dia de l'any que és i el tipus d'absència que hi ha en cas d'haver-hi.
- **(2/2) Posada a punt de tot el client Web**
  - **Descripció:** Aquesta targeta té com a objectiu la posada a punt del Frontend del Gestor d'Absències.
- **(1/1) Anàlisi integració del Gestor d'absències en l'entorn de Testing**
  - **Descripció:** Aquesta targeta té com a objectiu l'anàlisi de l'aplicatiu a l'entorn de Testing. Per fer-ho caldrà saber els processos que hauran de servir l'aplicació i on es guardaran els diferents fitxers de logs.
- **(1/1) Arreglar CORS**
  - **Descripció:** Aquesta targeta té com a objectiu habilitar el CORS del Backend per tal de poder utilitzar-lo sense necessitat de dependre d'extensions que tractin les CORS.
- **(3/1) Implantació del Gestor d'absències a Testing**
  - **Descripció:** Aquesta targeta té com a objectiu implantar el Gestor d'Absències en un entorn de testing.
    - Implantació del Django
    - Implantació del Frontend
    - Configuració de l'wSGI
    - Configuració del Nginx
    - Configuració del supervisor
- **(2/1) Proves de rendiment / càrrega del Backend**
  - **Descripció:** Aquesta targeta té com a objectiu realitzar proves de càrrega i concurrència en el Backend tant del servidor de testing com del desenvolupament.
- **(2/2) Parametrització dels tests d'ocurrències**
  - **Descripció:** Durant el desenvolupament del Backend m'he adonat que els tests son fràgils, cosa que fa que segons el dia de la setmana que els vulgui passar passin o fallin. Aquesta targeta té com a objectiu la parametrització dels tests



d'ocurrències del Backend. S'ha de permetre setejar les accions que es testejaran a partir d'una data en el futur, a ser possible en un dilluns per poder passar tots els tests amb les mateixes dates.

## 14.2. Documentació de l'API

Aquesta és la documentació amb OpenAPI 3.0:

```
openapi: 3.0.0
# Added by API Auto Mocking Plugin
servers:
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/BenjiRS94/gestor\_absencies/1.0.0
info:
  description: API del projecte del Gestor d'Absències de Som Energia
  version: "1.0.0"
  title: API per gestionar les Absències
  contact:
    email: benjami.ramos@somenergia.coop
  license:
    name: AGPL - 3.0
tags:
  - name: login
    description: Request per autenticar-se
  - name: workers
    description: Request pel tractament dels treballadors
  - name: teams
    description: Request pel tractament dels equips
  - name: members
    description: Request pel tractament dels membres d'equips
  - name: vacation_policy
    description: Request pel tractament de les polítiques de vacances
  - name: absence_type
    description: Request pel tractament dels tipus d'absències
  - name: occurrence
    description: Request pel tractament de les ocurrences d'absències
paths:
  /login:
    post:
      tags:
        - login
      summary: Autentifica a l'usuari
      description: Autentifica un usuari en el Backend
      responses:
        '200':
          description: Usuari autenticat
        '400':
          description: Error amb l'autenticació
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/login_response'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/login'
  /workers:
    get:
      security:
        - ApiKeyAuth: [JWT]
      tags:
        - workers
      summary: Retorna tots els treballadors
      operationId: get_workers
      description: |
        Retorna la llista de tots els usuaris de la plataforma
      responses:
        '200':
          description: Llista de Workers retornada
```

content:  
 application/json:  
 schema:  
 type: array  
 items:  
 \$ref: '#/components/schemas/workerList'  
 '400':  
 description: Error en el format alhora d'enviar la  
 petició  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error400'  
 '401':  
 description: Es necessita autenticació  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error401'  
 '403':  
 description: No es té permisos  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error403'  
 post:  
 security:  
 - ApiKeyAuth: [JWT]  
 tags:  
 - workers  
 summary: Crea un treballador  
 operationId: post\_worker  
 description: Afegeix un treballador a l'aplicació  
 responses:  
 '201':  
 description: Worker creat  
 '400':  
 description: Error en el format alhora d'enviar la  
 petició

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error400'  
 '401':  
 description: Es necessita autenticació  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error401'  
 '403':  
 description: No es té permisos  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error403'  
 requestBody:  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Worker'  
 /workers/{workerId}:  
 put:  
 parameters:  
 - name: workerId  
 in: path  
 description: ID del Worker  
 required: true  
 schema:  
 type: integer  
 security:  
 - ApiKeyAuth: [JWT]  
 tags:  
 - workers  
 summary: Edita un treballador  
 operationId: put\_worker  
 description: Edita un treballador de l'aplicació  
 responses:  
 '200':

description: Worker modificat  
content:  
application/json:  
schema:  
type: object  
items:  
\$ref: '#/components/schemas/Worker'  
'400':  
description: Error en el format alhora d'enviar la  
petició

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error401'

'403':  
description: No es té permisos

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error403'

requestBody:  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Worker'

get:  
parameters:  
- name: workerId  
in: path  
description: ID del Worker  
required: true  
schema:  
type: integer

security:  
- ApiKeyAuth: [JWT]  
tags:  
- workers  
summary: Retorna l'usuari que té aquest  
identificador  
operationId: get\_worker  
responses:  
'200':  
description: Usuari retornat  
content:  
application/json:  
schema:  
type: array  
items:  
\$ref: '#/components/schemas/workerList'

'400':  
description: Error en el format alhora d'enviar la  
petició

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error401'

'403':  
description: No es té permisos

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error403'

delete:  
parameters:  
- name: workerId  
in: path

description: ID del Worker  
required: true  
schema:  
  type: integer  
security:  
  - ApiKeyAuth: [JWT]  
tags:  
  - workers  
summary: Elimina l'usuari que tingui l'identificador

workerId

operationId: delete\_worker

responses:

'204':

description: Worker eliminat

'400':

description: Error en el format alhora d'enviar la

petició

content:

application/json:

schema:

\$ref: '#/components/schemas/Error400'

'401':

description: Es necessita autenticació

content:

application/json:

schema:

\$ref: '#/components/schemas/Error401'

'403':

description: No es té permisos

content:

application/json:

schema:

\$ref: '#/components/schemas/Error403'

/teams:

get:

security:

- ApiKeyAuth: [JWT]

tags:

- teams

summary: Retorna tots els equips de treball

operationId: get\_teams

responses:

'200':

description: Workers tots els equips de treball

content:

application/json:

schema:

type: array

items:

\$ref: '#/components/schemas/teamList'

'400':

description: Error en el format alhora d'enviar la

petició

content:

application/json:

schema:

\$ref: '#/components/schemas/Error400'

'401':

description: Es necessita autenticació

content:

application/json:

schema:

\$ref: '#/components/schemas/Error401'

'403':

description: No es té permisos

content:

application/json:

schema:

\$ref: '#/components/schemas/Error403'

post:

security:

- ApiKeyAuth: [JWT]

tags:

- teams

summary: Crea un equip de treball

operationId: post\_team

description: Afegeix un equip de treball a l'aplicació

responses:

'201':  
 description: Equip de treball creat

'400':  
 description: Error en el format alhora d'enviar la  
 petició

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error400'

'401':  
 description: Es necessita autenticació

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error401'

'403':  
 description: No es té permisos

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error403'

requestBody:  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Team'

/teams/{teamId}:

put:  
 parameters:  
 - name: teamId  
 in: path  
 description: ID del Team  
 required: true  
 schema:  
 type: integer

security:  
 - ApiKeyAuth: [JWT]

tags:

- teams

summary: Edita un equip de treball

operationId: put\_team

description: Edita un equip de treball de l'aplicació

responses:

'200':  
 description: Equip de treball modificat

content:  
 application/json:  
 schema:  
 type: object  
 items:  
 \$ref: '#/components/schemas/Team'

'400':  
 description: Error en el format alhora d'enviar la  
 petició

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error400'

'401':  
 description: Es necessita autenticació

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error401'

'403':  
 description: No es té permisos

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error403'

requestBody:  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Team'

get:  
 parameters:

- name: teamId \$ref: '#/components/schemas/Error403'  
in: path  
description: ID de l'Equip de treball  
required: true  
schema:  
  type: integer  
security:  
  - ApiKeyAuth: [JWT]  
tags:  
  - teams  
summary: Retorna l'equip de treball que tingui l'identificador teamId  
operationId: get\_team  
responses:  
  '200':  
    description: Equip de treball retornat  
    content:  
      application/json:  
        schema:  
          type: array  
          items:  
            \$ref: '#/components/schemas/teamList'  
  '400':  
    description: Error en el format alhora d'enviar la petició  
    content:  
      application/json:  
        schema:  
          \$ref: '#/components/schemas/Error400'  
  '401':  
    description: Es necessita autenticació  
    content:  
      application/json:  
        schema:  
          \$ref: '#/components/schemas/Error401'  
  '403':  
    description: No es té permisos  
    content:  
      application/json:  
        schema:

delete:  
parameters:  
  - name: teamId  
    in: path  
    description: ID de l'equip de treball  
    required: true  
    schema:  
      type: integer  
security:  
  - ApiKeyAuth: [JWT]  
tags:  
  - teams  
summary: Elimina l'equip de treball que tingui l'identificador teamId  
operationId: delete\_team  
responses:  
  '204':  
    description: Equip de treball eliminat  
  '400':  
    description: Error en el format alhora d'enviar la petició  
    content:  
      application/json:  
        schema:  
          \$ref: '#/components/schemas/Error400'  
  '401':  
    description: Es necessita autenticació  
    content:  
      application/json:  
        schema:  
          \$ref: '#/components/schemas/Error401'  
  '403':  
    description: No es té permisos  
    content:  
      application/json:  
        schema:  
          \$ref: '#/components/schemas/Error403'

```

/members:
  get:
    security:
      - ApiKeyAuth: [JWT]
    tags:
      - members
    summary: Retorna tots els membres d'equips
    operationId: get_members
    responses:
      '200':
        description: Membres d'equips retornats
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/memberList'
      '400':
        description: Error en el format alhora d'enviar la
        petició
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error400'
      '401':
        description: Es necessita autenticació
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error401'
      '403':
        description: No es té permisos
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error403'
  post:
    security:

```

```

      - ApiKeyAuth: [JWT]
    tags:
      - members
    summary: Crea un Membres d'equips
    operationId: post_member
    description: Afegeix un Membre d'equip a
    l'aplicació
    responses:
      '201':
        description: Membre d'equip creat
      '400':
        description: Error en el format alhora d'enviar la
        petició
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error400'
      '401':
        description: Es necessita autenticació
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error401'
      '403':
        description: No es té permisos
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error403'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Member'
/members/{memberId}:
  put:
    parameters:
      - name: memberId
        in: path

```



description: ID del Membre d'equip  
 required: true  
 schema:  
   type: integer  
 security:  
   - ApiKeyAuth: [JWT]  
 tags:  
   - members  
 summary: Edita un Membre d'equip  
 operationId: put\_member  
 description: Edita un Membre d'equip de  
 l'aplicació  
 responses:  
   '200':  
     description: Membre d'equip modificat  
     content:  
       application/json:  
         schema:  
           type: object  
           items:  
             \$ref: '#/components/schemas/Member'  
   '400':  
     description: Error en el format alhora d'enviar la  
     petició  
     content:  
       application/json:  
         schema:  
           \$ref: '#/components/schemas/Error400'  
   '401':  
     description: Es necessita autenticació  
     content:  
       application/json:  
         schema:  
           \$ref: '#/components/schemas/Error401'  
   '403':  
     description: No es té permisos  
     content:  
       application/json:  
         schema:  
           \$ref: '#/components/schemas/Error403'

requestBody:  
   content:  
     application/json:  
       schema:  
         \$ref: '#/components/schemas/Member'  
 get:  
   parameters:  
     - name: memberId  
       in: path  
       description: ID del Membre d'equip  
       required: true  
       schema:  
         type: integer  
   security:  
     - ApiKeyAuth: [JWT]  
   tags:  
     - members  
   summary: Retorna el Membre d'equip que tingui  
   l'identificador memberId  
   operationId: get\_member  
   responses:  
     '200':  
       description: Membre d'equip retornat  
       content:  
         application/json:  
           schema:  
             type: array  
             items:  
               \$ref: '#/components/schemas/memberList'  
     '400':  
       description: Error en el format alhora d'enviar la  
       petició  
       content:  
         application/json:  
           schema:  
             \$ref: '#/components/schemas/Error400'  
     '401':  
       description: Es necessita autenticació  
       content:

```

    application/json:
      schema:
        $ref: '#/components/schemas/Error401'
  '403':
    description: No es té permisos
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error403'

delete:
  parameters:
    - name: memberId
      in: path
      description: ID del Membre d'equip
      required: true
      schema:
        type: integer
  security:
    - ApiKeyAuth: [JWT]
  tags:
    - members
  summary: Elimina el Membre d'equip que tingui
  l'identificador memberId
  operationId: delete_members
  responses:
    '204':
      description: Membre d'equip eliminat
    '400':
      description: Error en el format alhora d'enviar la
      petició
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error400'
    '401':
      description: Es necessita autenticació
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error401'

/vacationpolicy:
  get:
    security:
      - ApiKeyAuth: [JWT]
    tags:
      - vacation_policy
    summary: Retorna totes les Politiques de Vacances
    operationId: get_vacation_policys
    responses:
      '200':
        description: Politiques de Vacances retornats
        content:
          application/json:
            schema:
              type: array
            items:
              $ref:
                '#/components/schemas/VacationPolicyList'
      '400':
        description: Error en el format alhora d'enviar la
        petició
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error400'
      '401':
        description: Es necessita autenticació
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error401'

```

'403':  
description: No es té permisos  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error403'

post:  
security:  
- ApiKeyAuth: [JWT]  
tags:  
- vacation\_policy  
summary: Crea una Politica de Vacances  
operationId: post\_vacation\_policy  
description: Afegeix una Politica de Vacances a l'aplicació

responses:  
'201':  
description: Politica de Vacances creada  
'400':  
description: Error en el format alhora d'enviar la petició

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error401'

'403':  
description: No es té permisos  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error403'

requestBody:  
content:

application/json:  
schema:  
\$ref: '#/components/schemas/VacationPolicy'

/vacationpolicy/{ vacationpolicyId }:  
put:  
parameters:  
- name: vacationpolicyId  
in: path  
description: ID de la Politica de Vacances  
required: true  
schema:  
type: integer  
security:  
- ApiKeyAuth: [JWT]  
tags:  
- vacation\_policy  
summary: Edita una Politica de Vacances  
operationId: put\_vacationpolicy  
description: Edita una Politica de Vacances de l'aplicació que tingui l'identificador vacationpolicyId

responses:  
'200':  
description: Politica de Vacances modificat  
content:  
application/json:  
schema:  
type: object  
items:  
\$ref: '#/components/schemas/VacationPolicy'

'400':  
description: Error en el format alhora d'enviar la petició

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació

```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/Error401'
'403':
  description: No es té permisos
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error403'
requestBody:
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/VacationPolicy'

get:
  parameters:
    - name: vacationpolicyId
      in: path
      description: ID de la Politica de Vacances
      required: true
      schema:
        type: integer
  security:
    - ApiKeyAuth: [JWT]
  tags:
    - vacation_policy
  summary: Retorna la Politica de Vacances que
tingui l'identificador vacationpolicyId
  operationId: get_vacationpolicy
  responses:
    '200':
      description: Politica de Vacances retornada
      content:
        application/json:
          schema:
            type: array
            items:

```

```

      $ref:
        '#/components/schemas/VacationPolicyList'
    '400':
      description: Error en el format alhora d'enviar la
      petició
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Error400'
    '401':
      description: Es necessita autenticació
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Error401'
    '403':
      description: No es té permisos
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Error403'

delete:
  parameters:
    - name: vacationpolicyId
      in: path
      description: ID del Worker
      required: true
      schema:
        type: integer
  security:
    - ApiKeyAuth: [JWT]
  tags:
    - vacation_policy
  summary: Elimina la Politica de Vacances la qual es
seu identificador sigui vacationpolicyId
  operationId: delete_vacation_policy
  responses:
    '204':
      description: Politica de Vacances eliminada

```

'400':  
description: Error en el format alhora d'enviar la  
petició  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error401'

'403':  
description: No es té permisos  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error403'

/absencetype:

get:  
security:  
- ApiKeyAuth: [JWT]  
tags:  
- absence\_type  
summary: Retorna tots els Tipus d'Absències  
operationId: get\_absence\_types  
responses:  
'200':  
description: Tipus d'Absències retornades  
content:  
application/json:  
schema:  
type: array  
items:  
\$ref: '#/components/schemas/AbsenceTypeList'

'#/components/schemas/AbsenceTypeList'

'400':

description: Error en el format alhora d'enviar la  
petició  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error401'

'403':  
description: No es té permisos  
content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error403'

post:

security:  
- ApiKeyAuth: [JWT]

tags:

- absence\_type

summary: Crea un Tipus d'Absència

operationId: post\_absence\_type

description: Afegeix un Tipus d'Absència a

l'aplicació

responses:

'201':  
description: Tipus d'Absència creada

'400':  
description: Error en el format alhora d'enviar la  
petició

content:  
application/json:  
schema:  
\$ref: '#/components/schemas/Error400'

'401':  
description: Es necessita autenticació

```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/Error401'
'403':
  description: No es té permisos
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error403'
requestBody:
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/AbsenceType'
/absencetype/{absencetypeId}:
  put:
    parameters:
      - name: absencetypeId
        in: path
        description: ID del Tipus d'Absència
        required: true
        schema:
          type: integer
    security:
      - ApiKeyAuth: [JWT]
    tags:
      - absence_type
    summary: Edita el Tipus d'Absència que tingui
l'identificador absencetypeId
    operationId: put_absence_type
    description: Edita un Tipus d'Absència de
l'aplicació
  responses:
    '200':
      description: Tipus d'Absència modificat
      content:
        application/json:
          schema:
            type: object
            items:
              $ref: '#/components/schemas/AbsenceType'
'400':
  description: Error en el format alhora d'enviar la
petició
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error400'
'401':
  description: Es necessita autenticació
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error401'
'403':
  description: No es té permisos
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Error403'
requestBody:
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/AbsenceType'
  get:
    parameters:
      - name: absencetypeId
        in: path
        description: ID del Tipus d'Absència
        required: true
        schema:
          type: integer
    security:
      - ApiKeyAuth: [JWT]
    tags:
      - absence_type

```

summary: Retorna el Tipus d'Absència que tingui  
l'identificador absenceTypeId

operationId: get\_absence\_type

responses:

'200':

description: Tipus d'Absència retornada

content:

application/json:

schema:

type: array

items:

\$ref:

#/components/schemas/AbsenceTypeList'

'400':

description: Error en el format alhora d'enviar la  
petició

content:

application/json:

schema:

\$ref: '#/components/schemas/Error400'

'401':

description: Es necessita autenticació

content:

application/json:

schema:

\$ref: '#/components/schemas/Error401'

'403':

description: No es té permisos

content:

application/json:

schema:

\$ref: '#/components/schemas/Error403'

delete:

parameters:

- name: absenceTypeId

in: path

description: ID del Tipus d'Absència

required: true

schema:

type: integer

security:

- ApiKeyAuth: [JWT]

tags:

- absence\_type

summary: Elimina el Tipus d'Absència que tingui  
l'identificador absenceTypeId

operationId: delete\_absence\_type

responses:

'204':

description: Tipus d'Absència eliminada

'400':

description: Error en el format alhora d'enviar la  
petició

content:

application/json:

schema:

\$ref: '#/components/schemas/Error400'

'401':

description: Es necessita autenticació

content:

application/json:

schema:

\$ref: '#/components/schemas/Error401'

'403':

description: No es té permisos

content:

application/json:

schema:

\$ref: '#/components/schemas/Error403'

/occurrence:

get:

security:

- ApiKeyAuth: [JWT]

tags:

- occurrence

summary: Retorna totes les Ocurrences

```

operationId: get_occurrences
responses:
  '200':
    description: Occurrències retornades
    content:
      application/json:
        schema:
          type: array
          items:
            $ref: '#/components/schemas/OccurrenceList'
  '400':
    description: Error en el format alhora d'enviar la
    petició
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error400'
  '401':
    description: Es necessita autenticació
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error401'
  '403':
    description: No es té permisos
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error403'
  '201':
    description: Occurrència creada
  '400':
    description: Error en el format alhora d'enviar la
    petició
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error400'
  '401':
    description: Es necessita autenticació
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error401'
  '403':
    description: No es té permisos
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Error403'
requestBody:
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Occurrence'
/occurrence/{occurrenceId}:
  get:
    parameters:
      - name: occurrenceId
        in: path
        description: ID de l'Ocurrència
        required: true
        schema:
          type: integer
    security:
      - ApiKeyAuth: [JWT]
    tags:
      - occurrence
post:
  security:
    - ApiKeyAuth: [JWT]
  tags:
    - occurrence
  summary: Crea una Occurrència
  operationId: post_occurrence
  description: Afegeix una Occurrència a l'aplicació
  responses:

```



summary: Retorna l'Ocurrencia que tingui  
 l'identificador occurrenceId  
 operationId: get\_occurrence  
 responses:  
 '200':  
 description: Ocurrencia retornada  
 content:  
 application/json:  
 schema:  
 type: array  
 items:  
 \$ref:

#/components/schemas/OccurrenceList'

'400':  
 description: Error en el format ahora d'enviar la  
 petició

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error400'

'401':  
 description: Es necessita autenticació  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error401'

'403':  
 description: No es té permisos  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error403'

delete:  
 parameters:  
 - name: occurrenceId  
 in: path  
 description: ID de l'Ocurrencia  
 required: true  
 schema:

type: integer  
 security:  
 - ApiKeyAuth: [JWT]  
 tags:  
 - occurrence  
 summary: Elimina l'Ocurrencia que tingui  
 l'identificador occurrenceId  
 operationId: delete\_occurrence  
 responses:  
 '204':  
 description: Workers retornats  
 '400':  
 description: Error en el format ahora d'enviar la  
 petició

content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error400'

'401':  
 description: Es necessita autenticació  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error401'

'403':  
 description: No es té permisos  
 content:  
 application/json:  
 schema:  
 \$ref: '#/components/schemas/Error403'

components:  
 schemas:  
 workerList:  
 type: object  
 properties:  
 results:

<p>items:</p> <p>  \$ref: '#/components/schemas/Worker'</p> <p>type: array</p> <p>Worker:</p> <p>required:</p> <ul style="list-style-type: none"> <li>- id</li> <li>- username</li> <li>- password</li> </ul> <p>properties:</p> <p>id:</p> <p>  type: integer</p> <p>  readOnly: true</p> <p>  example: 1</p> <p>username:</p> <p>  type: string</p> <p>  minLength: 0</p> <p>  maxLength: 50</p> <p>  example: benjami</p> <p>password:</p> <p>  type: string</p> <p>  minLength: 0</p> <p>  maxLength: 50</p> <p>  writeOnly: true</p> <p>first_name:</p> <p>  type: string</p> <p>  minLength: 0</p> <p>  maxLength: 50</p> <p>  example: Benjamí</p> <p>last_name:</p> <p>  type: string</p> <p>  minLength: 0</p> <p>  maxLength: 50</p> <p>  example: Ramos</p> <p>gender:</p> <p>  type: string</p> <p>  minLength: 0</p> <p>  maxLength: 50</p> <p>  example: Male</p> <p>category:</p>	<p>type: string</p> <p>minLength: 0</p> <p>maxLength: 50</p> <p>example: developer</p> <p>holidays:</p> <p>  type: integer</p> <p>  example: 25</p> <p>type: object</p> <p>teamList:</p> <p>  type: object</p> <p>  properties:</p> <p>    results:</p> <p>      items:</p> <p>        \$ref: '#/components/schemas/Team'</p> <p>        type: array</p> <p>Team:</p> <p>required:</p> <ul style="list-style-type: none"> <li>- id</li> <li>- name</li> </ul> <p>properties:</p> <p>id:</p> <p>  type: integer</p> <p>  readOnly: true</p> <p>  example: 1</p> <p>name:</p> <p>  type: string</p> <p>  minLength: 0</p> <p>  maxLength: 50</p> <p>  example: IT</p> <p>type: object</p> <p>memberList:</p> <p>  type: object</p> <p>  properties:</p> <p>    results:</p> <p>      items:</p> <p>        \$ref: '#/components/schemas/Member'</p> <p>        type: array</p>
--	--

Member:

required:

- id
- worker
- team
- is\_referent
- is\_representant

properties:

id:

- type: integer
- readOnly: true
- example: 1

worker:

- type: integer
- readOnly: true
- example: 1

team:

- type: integer
- readOnly: true
- example: 1

is\_referent:

- type: boolean
- example: true

is\_representant:

- type: boolean
- example: false

type: object

VacationPolicyList:

type: object

properties:

results:

items:

- \$ref: '#/components/schemas/VacationPolicy'

type: array

VacationPolicy:

required:

- id

- name
- description
- holidays

properties:

id:

- type: integer
- readOnly: true
- example: 1

name:

- type: string
- minLength: 0
- maxLength: 50
- example: Politica de Vacances ET

description:

- type: string
- minLength: 0
- maxLength: 250
- example: Politica de vacances utilitzada per l'ET

holidays:

- type: integer
- minimum: 0
- example: 25

type: object

AbsenceTypeList:

type: object

properties:

results:

items:

- \$ref: '#/components/schemas/AbsenceType'

type: array

AbsenceType:

required:

- id
- name
- description
- spend\_days
- max\_duration
- min\_duration

- max_spend	minimum: 0.5
- min_spend	maximum: -1
properties:	example: -1
id:	type: object
type: integer	
readOnly: true	OccurrenceList:
example: 1	type: object
name:	properties:
type: string	results:
minLength: 0	items:
maxLength: 50	\$ref: '#/components/schemas/Occurrence'
example: Baixa per maternitat	type: array
description:	Occurrence:
type: string	required:
minLength: 0	- id
maxLength: 250	- absence_type
example: Baixa que es pot demanar al tenir una	- worker
criatura	- start_time
spend_days:	- end_time
type: integer	- start_morning
minimum: -1	- start_afternoon
maximum: 1	- end_morning
example: -1	- end_afternoon
min_duration:	properties:
type: number	id:
minimum: 0.5	type: integer
maximum: -1	readOnly: true
example: 10	example: 1
max_duration:	absence_type:
type: number	type: integer
minimum: 0.5	example: 1
maximum: -1	worker:
example: -1	type: integer
min_spend:	example: 1
type: number	start_time:
minimum: 0.5	type: string
maximum: -1	format: date-time
example: 10	example: '2019-01-20T09:00:00'
max_spend:	end_time:
type: number	

type: string  
format: date-time  
example: '2019-01-23T17:00:00'  
start\_morning:  
type: boolean  
example: true  
start\_afternoon:  
type: boolean  
example: true  
end\_morning:  
type: boolean  
example: true  
end\_afternoon:  
type: boolean  
example: true  
type: object

login:  
required:  
- username  
- password  
properties:  
username:  
type: string  
example: benjami  
password:  
type: string  
readOnly: true  
type: object

login\_response:  
required:  
- worker\_id  
- is\_admin  
- token  
properties:  
worker\_id:  
type: integer  
readOnly: true  
example: 1

is\_admin:  
type: boolean  
readOnly: true  
token:  
type: string  
example:  
fn8n92ef9n9bdq9b.dfm0nf8nw38d9q289038hf08wbne8  
fnwdfndjnoiuiwenjdfs.fominmfowneoifnwefw3ehr93fE  
DFSDGwefh\_wef  
readOnly: true  
type: object

Error400:  
properties:  
message:  
type: string  
readOnly: true  
pattern: Error 400  
example: Error 400  
type: object

Error401:  
properties:  
message:  
type: string  
readOnly: true  
pattern: Error 401  
example: Error 401  
type: object

Error403:  
properties:  
message:  
type: string  
readOnly: true  
pattern: Error 403  
example: Error 403  
type: object

Error404:

properties:  
message:  
type: string  
readOnly: true

pattern: Error 404  
example: Error 404  
type: object