



**Universitat Autònoma  
de Barcelona**

**Treball de fi de grau  
Curs 2021-2022**

## **WEB DE SEGUIMENT DE TAQUES A LA PELL**

**Informe de progrés I**

Autor/a: **Daniel Calvo Ramos**

NIU autor/a: **1494116**

Tutor/a: **Helena Boltà Torrell**

Grau: **Grau en enginyeria informàtica**

Menció: **Enginyeria del Software**

Data entrega: **10/4/2022**

## Índex

<b>Introducció</b> .....	3
<b>Treball realitzat</b> .....	4
<b>Requeriments del sistema</b> .....	4
<b>Planificació</b> .....	5
<b>Disseny</b> .....	6
Llenguatge i arquitectura Software.....	6
Model relacional .....	6
<b>Metodologia</b> .....	7
Metodologia de desenvolupament del projecte.....	7
Control de versions .....	7
<b>Bibliografia</b> .....	8

## Introducció

En aquest primer informe de progrés, s'explicarà com han estat els primers avenços del TFG des de l'entrega de l'informe inicial del dia 6 de març.

S'explicarà el que s'ha avançat durant aquest més a nivell de programació del projecte, així com explicar si s'han seguit les bases fixades a l'informe inicial com la planificació, la metodologia establerta o si els requeriments inicials del sistema eren els correctes.

## Treball realitzat

Durant el temps comprés entre el lliurament de l'informe inicial i el primer informe de seguiment s'han realitzat les següents activitats:

- **Definició de l'entorn de treball i sistema de control de versions:** Primer de tot, s'ha hagut de fer una cerca sobre quin podria ser el IDE que millor es pogués adaptar al projecte. Finalment es va seleccionar Visual Studio Code, degut a la gran quantitat de extensions sobre Laravel que té i per la seva facilitat a l'hora d'utilitzar el control de versions. Posteriorment, es va crear el projecte base amb Laravel i es va enllaçar el projecte en local amb un repositori de Github, el qual també es va crear. Finalment, es va definir un servidor Apache per a poder visualitzar el resultat de la programació realitzada i una base de dades MySQL.
- **Estructura de dades i relacions BD:** Definició de l'esquelet relacional de la web. Per realitzar aquesta estructura de dades, s'ha de fer mitjançant migracions, que és la forma que té el *framework* Laravel per a crear entitats dins una base de dades. Per a cada entitat o taula s'han de definir els atributs corresponents. Un cop creades les taules de forma local, es fa una migració de totes juntes per a tenir-les disponibles al servidor Apache.
- **Login + registre pacients:** La primera peça del projecte és realitzar un sistema de login i autenticació per a que els usuaris puguin iniciar sessió en el sistema. Cal a dir que s'ha de poder permetre la distinció de rols. És a dir, depenent del rol que tingui un usuari, s'ha de mostrar una cosa o una altra. Addicionalment, s'ha realitzat el formulari de creació de pacients, per a que els metges puguin crear pacients quan sigui requerit.
- **Disseny front-end:** Disseny de mitja fidelitat de tot el front-end que tindrà la web. D'aquesta forma es pot tenir una idea molt més clara i precisa de totes les finestres i transicions que tindrà la web.
- **Crear les rutes web:** Crear totes les URL necessàries per a que la web pugui funcionar correctament. És la creació de les rutes privades de cada tipus d'usuari.

## Requeriments del sistema

Pel que fa als requeriments del sistema, s'han afegit tres noves funcionalitats en el sistema: per una banda s'ha afegit un nou tipus d'usuari admin el qual pugui gestionar la creació de metges al sistema. Aquesta serà la seva única funcionalitat.

Per altra banda, també s'ha d'afegir una nova funcionalitat als pacients per a que puguin canviar la seva contrasenya de inici de sessió, la qual vindrà definida per defecte.

Finalment, el sistema també a de poder marcar el contorn de una taca.

Així, la llista de requeriments del sistema queda de la següent manera (en negreta els nous requeriments):

- El sistema ha de poder quantificar la gravetat d'una taca a partir de l'anàlisi d'una imatge.
- El sistema ha de permetre la generació d'un històric de la evolució d'una taca d'un pacient.

- El sistema ha de permetre al pacient pujar imatges de taques.
- El sistema ha de permetre que els metges puguin modificar els camps d'una imatge.
- El sistema ha de proveir als metges la capacitat de crear nous pacients.
- El sistema ha de poder mostrar outputs diferents de dades segons el tipus d'usuari.
- El sistema ha de permetre al metge poder comparar les fotos d'un mateix pacient de forma clara.
- El sistema ha de permetre obtenir dades d'una API externa i processar-les.
- El sistema ha de poder enviar fotos a la API externa.
- El sistema ha de permetre als metges navegar entre els diferents pacients.
- El sistema haurà de permetre mostrar l'evolució de les taques.
- El sistema ha de proveir als metges la capacitat de canviar el metge associat a un dels seus pacients.
- **El sistema ha de proveir als pacients la capacitat de canviar la seva contrasenya un cop han iniciat sessió.**
- **El sistema ha de tenir tres tipus d'usuari: pacient, metge i admin.**
- **El sistema ha de permetre als usuaris de tipus admin la capacitat de poder gestionar la base de dades del sistema.**
- **El sistema ha de ser capaç de marcar el contorn d'una taca a partir de una imatge.**
- **El sistema ha de permetre l'accés al sistema només al usuaris que estiguin registrats.**

## Planificació

En quant a la planificació del projecte, no ha petit cap canvi fins al moment, tot i que és possible que en un futur hi hagi una reestructuració d'algunes activitats.

Així la planificació d'aquestes 5 setmanes ha estat la següent:

FASE	Activitat	Inici activitat	Final activitat
Implementació	Definició entorn de treball + control de versions	07/03/2022	13/03/2022
Implementació	Estructura de dades i relacions bd	14/03/2022	20/03/2022
Implementació	Login + registre	21/03/2022	27/03/2022
Implementació	Disseny frontend	28/03/2022	03/04/2022
Implementació	Definir API local	04/04/2022	10/04/2022

**Figura 1. Taula de la planificació**

## Disseny

En aquest apartat s'explica l'evolució pel que respecta a temes de disseny i model relacional del projecte.

### Llenguatge i arquitectura Software

Pel que fa a llenguatge i arquitectura Software del projecte, es segueixen les mateixes directrius dictaminades a l'informe inicial: El projecte es farà amb el *framework* Laravel seguint una arquitectura de Software MVC o Model-Vista-Controlador.

### Model relacional

El model relacional del sistema és la part que més canvis ha patit.

Inicialment, el model relacional estava format per 3 taules: metge, pacient i imatge. Doncs actualment tenim 4 taules o entitats: les 3 que teníem anteriorment, una nova taula admin i una altra usuari.

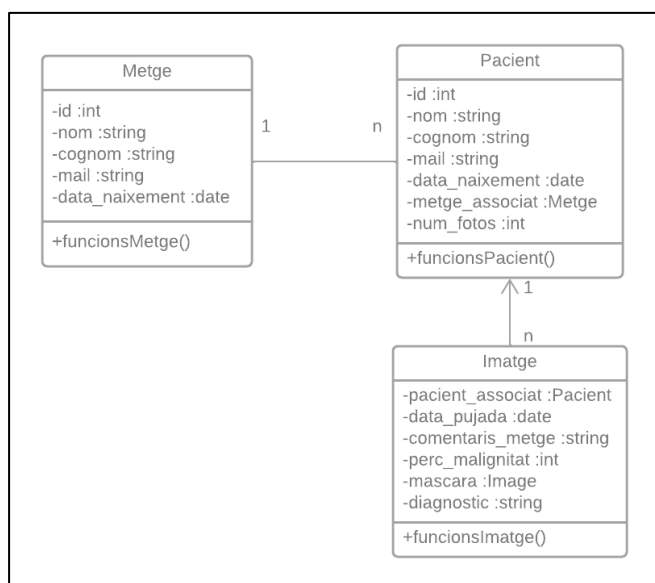
El motiu d'incloure una nova taula usuari ve de la necessitat de poder diferenciar rols d'usuaris per a facilitar les tasques d'autenticació i de separació de funcionalitats depenent del tipus d'usuari. D'aquesta forma tindrem la taula usuari que estarà formada per les taules metge, pacient i admin.

Per altra banda, s'ha afegit un nou tipus d'usuari: admin. La funció d'aquest tipus d'usuari serà únicament la de gestionar la base de dades del sistema (principalment crearà metges), per això aquesta taula no tindrà cap atribut extra.

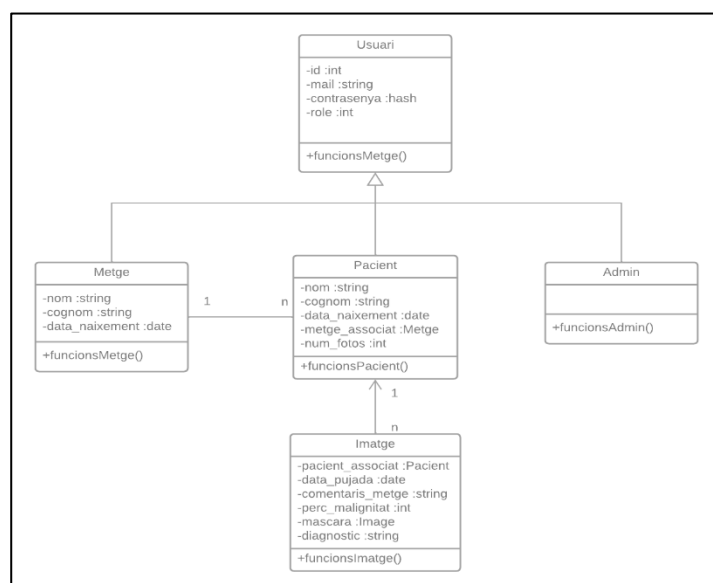
Finalment, també ha hagut canvis en els atributs de algunes taules, ja que sinó hi havia funcionalitats del sistema que no es podrien implementar.

Concretament, s'ha hagut d'afegir un camp contrasenya a les taules metge i pacient.

Per a què es pugui apreciar la diferència de forma visual, a continuació podem apreciar la diferència entre el model relacional definit a l'informe inicial (figura 2), i l'actual (figura 3).



**Figura 2. Model relacional inicial**



**Figura 3. Model relacional actual**

## Metodologia

En aquest apartat es parlarà dels canvis que ha patit la metodologia proposada inicialment per a la realització del projecte.

### Metodologia de desenvolupament del projecte

Tal i com es va planificar a la fase inicial, la metodologia que s'ha seguit no ha patit cap variació.

Per a la realització del projecte s'ha seguit una metodologia iterativa o incremental basada en petits *Sprints* de una setmana de duració.

Així, es planificaven petites parts del projecte amb una duració d'una setmana, juntament amb una reunió setmanal o *Sprint review* amb el client per a comprovar que el treball realitzat durant el *Sprint* compleix els objectius inicials i s'adapta a les necessitats del client.

El fet de estar desenvolupant el projecte seguint aquesta metodologia, fa que es puguin detectar de forma ràpida desviacions en el projecte o requisits no complerts o no adequats. I al realitzar una reunió setmanal d'uns 20 minuts amb el client, també ajuda a que el client estigui implicat i assabentat dels canvis que es van realitzant en el projecte i poder seguir d'una forma més propera l'evolució del mateix.

### Control de versions

Pel que fa al control de versions, durant aquestes primeres setmanes, com l'objectiu principal era establir les bases del projecte, hi ha hagut poca activitat pel que fa a actualitzacions al repositori de Github.

Tot i això, seguirà sent la font d'intercanvi de codi i documentació amb el client i professora tutora.

El link del Github és el següent: **<https://github.com/DanielCalvoRamos/WEB-TAQUES>**

## Bibliografia

- [1] LucidChart. (2008). *Software de Diagramas Online*. Recuperat de: <https://lucid.app/>
- [2] Xavier Albaladejo. (2019). *Execució de una iteració (Sprint)*. Recuperat de: <https://proyectosagiles.org/ejecucion-iteracion-sprint>
- [3] Mercado Ramos, Victor Hugo. *Revista de Investigación, Desarrollo e Innovación*. Editorial UPTC, 2015. Trobat a: <https://revistas.uptc.edu.co/>
- [4] Concepción, Pedro. *Planificación de proyectos de Software*. Recuperat de: <http://www.cs.umss.edu.bo/>
- [5] Monreal, Carlos. (2018). *Ciclos de vida iterativo e incremental, ¿Qué son?* Recuperat de: <https://www.cursodireccionproyectos.com/>
- [6] (2020). *Cicle de vida del Software: Tot el que necessites saber*. Recuperat de: <https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>