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

Assbaghi Asbahi, Youssef; Valveny Llobet, Ernest, dir. Aula de programació interactiva. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/257796>

under the terms of the  license

Aula de programació interactiva

Youssef Assbaghi Asbahi

Resum— Un dels majors problemes que es troben els professors de programació és la supervisió dels alumnes a l'hora de fer problemes i desenvolupar codi a classe. Aquests ho tenen complicat per rebre un feedback de manera directa, ja que no hi ha eines per les quals un professor pugui ajudar als alumnes de la classe sense exposar-se ni perdre temps anant taula per taula. A més a més aquest problema s'ha agreujat degut a la crisi sanitària, ja que la necessitat de tenir una eina que resolgui aquest problema és cada vegada més forta. Per tant hem decidit crear una aula de programació interactiva on el professor podrà publicar problemes i els estudiants els podran resoldre. La innovació en aquest projecte és que en tot moment el professor podrà veure en un llistat el codi que desenvolupen els estudiants i accedir i modificar-los, oferint així un feedback en temps real des de la comoditat del seu ordinador.

Paraules clau— Programació, C++, Python, Kanban, Problema, Classe, Covid, Alumne, Professor, MVC, Javascript, JQuery, PHP, html, css, AJAX, Aula

Abstract— One of the biggest problems teachers face in programming is supervising students when it comes to making problems and developing code in class. These make it difficult to receive feedback directly, as there are no tools by which a teacher can help students in the class without exposing themselves or wasting time going table by table. Furthermore, this problem has been exacerbated due to the health crisis, as the need to have a tool to solve this problem. So, we decided to create an interactive programming, classroom where the teacher will be able to post problems and students will be able to solve them. The innovation in this project is that at all times the teacher can see in a list the code developed by students and access and modify them, thus providing real-time feedback from the comfort of your computer.

Index Terms— Programming, C++, Python, Kanban, Problem, Class, Covid, Student, Teacher, MVC, Javascript, JQuery, PHP, html, css, AJAX, Classroom



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

A vui dia les classes de programació son molt poc dinàmiques. Es a dir que no s'optimitza el temps del professor ni dels estudiants per compartir el dubtes ni solucions. Per culpa d'aquest problema hem decidit una solució pels estudiants de programació. El problema que trobem és que a l'hora de programar moltes vegades surten problemes i el professor s'ha de desplaçar cap a l'ordinador de l'estudiant per veure el codi que ha realitzat. Aquesta és una feina ineficient i poc dinàmica, ja que es perd massa temps anant taula per taula i revisant el codi moltes vegades de manera incòmoda.

A més a més moltes vegades els alumnes no tenen les eines necessàries per poder programar de manera adequada, ja que tenen molts problemes, ja sigui perquè tenen un ordinador poc potent, o no tenen memòria per emmagatzemar els programes que volen realitzar o directament no tenen un ordinador per poder realitzar els exercicis que es demanen. Per reduir aquesta desigualtat i que els més desfavorits puguin accedir a una educació més pràctica s'ha ideat un projecte d'una aula interactiva.

Per tant per reduir aquests dos problemes i tenint en compte l'agrarbat de la crisi sanitària per mantenir una distancia de seguretat s'ha decidit crear una aula de pro-

gramació interactiva en línia, on només fa falta connexió a internet i un dispositiu sigui un ordinador o fins i tot el mateix telèfon mòbil. No fa falta que sigui un dispositiu potent, ja que les execucions de codi i l'ús de memòria serà gestionat per un servidor. A més a més el professor podrà veure el codi que estan realitzant els alumnes a classe de manera directa i això afavorirà al dinamisme de les explicacions. Ja que no fa falta que s'aixequi a anar a totes les taules a veure què estan fent els estudiants sinó que des del panel de control de l'aplicació podrà accedir al codi i poder explicar i millorar a part de modificar-lo per ajudar als alumnes a entendre.

Els principals motius per l'elecció d'aquest projecte és per disminuir la desigualtat pel que fa a recursos per alumnes. I també la millora de la seguretat a escala de salut, ja que en aquests temps és una part fonamental.

1.1 ESTAT DE L'ART

La intenció en aquest projecte és crear una web similar a leetcode (que és una plataforma de programació, on es poden comentar dubtes i té un corrector per poder corregir els problemes) referència [1] o hackerrank (és una plataforma pensada per disputar competicions de programació) referència [2] la novetat en el nostre programa serà que afegirem un sistema on es podrà compartir el codi i un professor el podrà editar com si fos un document compartir de Google Drive referència [4].

-
- E-mail de contacte: Youssef.assbaghi@gmail.com
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Ernest Valveny (Computació)
 - Curs 2021/22

2 OBJECTIUS

Volem una web que utilitzaran principalment els alumnes. L'objectiu és crear una eina perquè el professor pugui pujar problemes i que els alumnes puguin resoldre dins de la web i que tingui l'opció de poder compartir el codi amb el professor de manera senzilla per poder corregir-lo o donar feedback. Tot això volem que es desenvolupi sota els criteris apresos durant la menció d'enginyeria del software les quals estan formades per les fases de:

- Planificació
- Disseny
- Desenvolupament
- Test
- Documentació

Per tant hem de realitzar la planificació de l'abast del projecte mitjançant Sprints per poder assegurar-nos de que el projecte es viable i definir els punts més crítics a desenvolupar, tot això passa primerament per una captació de requeriments per comprendre quina és la magnitud del projecte. També hem de realitzar un disseny de la web mitjançant diagrames i prototips per tenir una base a seguir durant la fase de desenvolupament. En la fase de desenvolupament hem de posar en practica els coneixements que hem après durant tota la carrera creant una web optimitzada rapida i segura. La fase de test es una de les més importants per el qual hem d'assegurar-nos que durant la fase de desenvolupament es compleixen els criteris i requisits establerts a més de que es compleixen els mínims de seguretat. Finalment haurem de realitzar la documentació on a més de definir el funcionament de la web també haurem de crear un manual d'usuari pels usuaris futurs.

2.1 REQUISITS

Es vol crear una aplicació web que tindrà dues interfícies depenent si ets professor o alumne. L'usuari de professor podrà crear assignatures amb subgrups i definir els enunciats dels problemes i les seues descripcions a més de poder definir el temps d'execució, el llenguatge de programació que pot ser tant C++ com Python i la quantitat de memòria utilitzada. També podrà assignar professors a assignatures perquè tinguin els mateixos poders. També el professor ha de tenir l'habilitat de poder editar els problemes quan vulgui i tindrà el poder d'assignar un problema a una sessió d'un grup i activar-lo o desactivar-lo en funció de l'hora de treball a més de descarregar-los al seu directori.

L'usuari alumne en canvi les seves funcionalitats en l'aplicació seran limitades. Ja que només podrà iniciar sessió, i registrar-se com alumne. I per accedir a l'aula s'haurà d'incorporar al grup que el professor obri en aquell moment per començar a realitzar el problema.

També podran resoldre els problemes que ha creat el professor i afegir nous fixers si és necessari. Així mateix podran descarregar els problemes que han realitzat al seu directori. Per defecte, el professor haurà de poder veure el codi de tots els estudiants que estan connectats a la sessió i poder crear una connexió professor-estudiant via xat o comentaris.

Ha de tenir una base de dades on es guardaran els usuaris i els directoris a més de problemes i els noms de les assignatures i grups.

Ha de tenir un compilador que sigui capaç de compilar en Python i C++, però ha d'estar preparat per afegir nous llenguatges de manera modular. A més ha de tenir un sistema de seguretat per evitar ús de memòria i temps no desitjats. Ha de tenir un sistema per registrar-se i fer inici de sessió.

Per deixar clar els requisits que es demanen en aquest projecte s'ha fet un diagrama de casos d'ús per constatar les funcions que ha de tenir i poder realitzar cada usuari de la web.



1 Diagrama de casos d'ús

2.2 REQUISITS NO FUNCIONALS

En el que respecte a requisits funcionals, s'ha establert un ordre en funció de la demanda del projecte i del ni-vell d'ús de l'aplicació. A mesura que tingui més demanda aquest projecte, s'hauran de millorar les condicions per mantenir el rendiment desitjat. La part del servidor es codificarà en php, ja que gràcies a això podrem cridar al compilador del servidor sense grans problemes. Per altra banda, la part del frontend estarà escrita en html i dissenyada amb css i bootstrap a més a més ha de tenir javascript i AJAX per carregar les dades de manera asíncrona.

Aquest sistema funcionarà amb un servidor apache que és executat amb xampp i la base de dades de phpmyadmin. El servidor ha de tenir una capacitat de 500 GB a l'any per poder emmagatzemar les dades dels problemes de tots els alumnes. A més haurà de tenir un processador de 6 nuclis i 12 threads o més i haurà de tenir 16 GB de memòria RAM mínim per no tenir pèrdues de rendiment. El sistema operatiu on s'ha d'executar preferiblement es Linux Debian, però es pot executar també en Ubuntu o Windows sense cap problema.

3 METODOLOGIA

La metodologia que utilitzarem en aquest projecte serà una metodologia àgil gràcies al seu model iteratiu i l'alta facilitat a la gestió de canvis. Exactament serà Kanban, ja que com no formo part d'un equip gran no és necessari una metodologia com SCRUM. Mitjançant una llista de tasques amb una petita definició serà suficient per gestionar el projecte. S'utilitzarà l'aplicació Trello Referència [3], ja que té una interfície molt fàcil d'utilitzar i la gestió dels Sprints es pot fer de manera senzilla. Els esprints duraran dues setmanes en les quals s'espera desenvolupar les tasques planificades, i al final de l'esprint es farà una petita retrospectiva per avaluar els problemes trobats i la manera de solucionar-los en cas que es pugui.

4 PLANIFICACIÓ

Aquest projecte serà planificat de tal manera que es puguin implementar tots els requeriments que els professors demanen. Per fer-ho s'han establert fites de desenvolupament on en cada fita s'espera la entrega d'uns objectius ja establerts. La finalitat d'aquestes fites es la flexibilitat en la elecció de les tasques i un ajustament més optimit al temps que tenim.

Modo de	Nombre de tarea	Duración
★	Fita 1	15 días
🔧	Captació de requisits	5 días
🔧	Definició metodologia	2 días
🔧	Disseny base de dades	4 días
🔧	Definició objectius	3 días
🔧	Planificació	1 día
★	Fita 2	44 días
🔧	Implementació IDE de programació	8 días
🔧	Creació base de dades	4 días
🔧	Implementar login	2 días
🔧	Implementar registre	1 día
🔧	Implementar crear problema	3 días
🔧	Implementar edició de problemes	8 días
★	Fita 3	55 días
🔧	Implementar sistema per modificar problema	2 días
🔧	Implementar sistema per editar col·laborativament	10 días
🔧	Implementar creació d'aulas	5 días
🔧	Creació de grups de classe	4 días
🔧	Redactar i executar testing	10 días

2 Objectius Fites

A més aliat per averiguar quines eren les tasques mes importants i poder així seguir un ordre en funció de la prioritat que els professors prefereixen.

Objectius	Prioritat (Critic/Alt/Baix)
Crear base de dades per guardar problemes i usuaris	Critic
Crear interfície per usuaris amigable	Baix
Afegir un IDE que pugui acceptar múltiples llenguatges	Critic
Vincular un compilador per executar el codi i veure output	Alt
Crear sistema de login i registre per nous usuaris	Alt
Crear sistema de assignatures i classes	Baix
Crear sistema per poder modificar i visualitzar de manera interactiva el codi de l'estudiant	Critic
Crear sistema per poder emmagatzemar i importar arxius	Baix
Crear sistema per poder crear problemes i assignar-los de manera correcta	Alt
Implementar seguretat en execució	Alt
Implementar logica per executar múltiples arxius	Alt
Crear interfície per professor i estudiant	Baix

3 Taula Objectius/ Prioritats

5 DISSENY

La nostra aplicació en l'àmbit estructural tindrà un patró de disseny similar al MVC (Model - Vista - Controlador) això ens permetrà definir una lògica de control central que a la vegada podrem afegir noves funcionalitats sense gaire problema. L'estructura tindrà punts aïllats per on es guardarà les dades dels usuaris. És a dir, la carpeta de codi dels alumnes i professors i els executables que es

puguin generar quedaran fora del servidor. Això ens permetrà escalar-lo i aïllar-lo en un futur si volem utilitzar Docker o algun sistema cloud per poder compilar el codi.

La base de dades és una de les parts més susceptibles al canvi en aquest projecte. En una primera instància s'han definit diferents taules que estan vinculades entre elles:

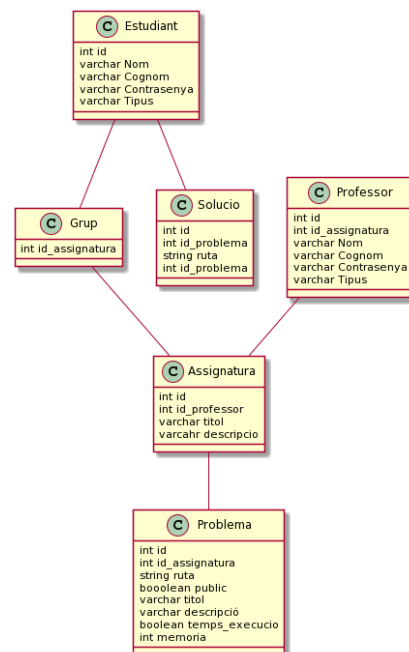
La taula estudiant serà principalment una taula senzilla que servirà perquè altres elements puguin definir-se de manera única. A causa d'això aquesta taula estarà formada per id, nom, cognom, email i una contrasenya encriptada amb hash 256.

La del professor és més o menys com la de l'estudiant, ja que es registrarà amb les mateixes dades que els estudiants. Una de les diferències és que emmagatzemarà les assignatures a les quals està assignat un professor.

La taula assignatura és una altra taula senzilla que servirà per definir més elements en un futur. És a dir que en si aquesta taula o millor dit si una assignatura desapareix, tots els elements relacionats amb aquesta assignatura quedarien eliminats. Principalment les entrades d'aquesta taula serà nom, descripció, id i curs.

Una de les taules més completes és la taula de problema. Aquesta és l'encarregada d'emmagatzemar les dades que un professor puja a un problema i les seves característiques. Les columnes d'aquesta taula són id, la ruta on s'emmagatzema el codi dins del servidor, el títol, descripció, les dades de si són públics o privats, la memòria que utilitzaran, el temps d'execució, el llenguatge de programació i l'assignatura a la qual estan assignats els problemes.

Les solucions s'emmagatzemaran en una altra taula. Aquesta serà l'encarregada d'emmagatzemar la ruta de les respostes que els estudiants generin per cada problema. Per tant també guardarà les dades del problema i assignatura a part de l'usuari que el desenvolupa.



4 Diagrama E-R

6 DESENVOLUPAMENT

En la part del desenvolupament s'ha tractat de seguir la planificació que s'ha establert al principi encara que hi ha hagut variacions com per exemple l'edició de manera asíncrona que ha hagut de deixar cap a més endavant fins que no tinguem una estructura. Aquestes variacions no afecten a la planificació inicial ja que comptàvem amb canvis. A més a més respecte al user experience i que la web sigui user friendly no s'analitzarà fins a futures versions.

6.1 CREACIÓ DE BASE DE DADES

Primerament vam implementar la base de dades que vam dissenyar l'anterior vegada. La urgència de la codificació d'aquesta era importat, ja que gràcies als dubtes que poguéssim resoldre acabàrem afegint o eliminant propietats a les taules. La creació s'ha fet manualment creant taules a phpmyadmin i afegint les columnes manualment, després les edicions que s'han fet s'han realitzat mitjançant codi perquè era la manera més senzilla. Les connexions per claus foranies també s'ha fet mitjançant la interfície de phpmyadmin, ja que era la millor solució per les poques connexions que tenia.

Una vegada creada la base de dades es va crear el codi per la connexió d'aquesta. La intenció era verificar que teníem accés des de la web per poder començar a implementar funcionalitats. Les principals taules creades són les de Estudiant, Professor, Assignatura, Problema, Solució i Tokens. Totes les taules menys la última estan interrelacionades.

6.2 REGISTRE

Una de les primeres funcionalitats que es van implementar de manera completa va ser el registre. Aquest era l'encarregat de verificar que les dades dels nous usuaris fossin dades correctes i que s'insereixin a la base de dades. Per això havíem de realitzar les primeres verificacions en JavaScript per assegurar-nos que les dades que s'estaven enviant eren compatibles amb el format que els professors demanen. Amb expressions regulars comprovem que els valors del formulari tinguin una longitud mínima i un format correcte. Una vegada tot tingui el format correcte encriptem la contrasenya mitjançant un hash de 256 bits i guardem a la base de dades el nou usuari. Totes les dades es trameten mitjançant post per evitar que les dades siguin vulnerables per un CSS (Cross site scripting).

La part d'identificar si es registra un professor o estudiant encara no està especificat. Per tant tots els registres que es fan des de la pantalla de registre són estudiants. Els professors tindran un sistema una mica diferent.

6.3 LOGIN

El inici de sessió dels usuaris va ser una altra funcionalitat que es caracteritzava per la similitud amb el registre. És a dir que vam haver de reutilitzar diverses funcions i

validacions de JavaScript per poder fer-ho, aconseguint així un desenvolupament àgil i ràpid. Igual que en el registre no s'identifica i l'usuari és un professor o estudiant la qual cosa porta al fet que de moment només els alumnes tenen l'opció d'iniciar sessió.

Una vegada s'ha validat que les dades són les correctes, s'inicialitzen les variables de sessió de la web amb el nom i email. Això ens permetrà identificar a l'usuari en totes les pàgines de la web. A més d'aquestes dades també es crea la variable de sessió per tipus d'usuari per verificar si és estudiant o professor. Pot ser que en el futur s'afegeixi un usuari admin que tingui les capacitats de fer-ho tot, d'aquesta manera podríem implementar un BackOffice i ajudar a gestionar les classes.

6.4 EDITOR

L'editor de codi és el nucli de la nostra web, ja que l'aplicació de programació interactiva no funcionarà sense un editor. Aquest mòdul està format per diferents parts que són el compilador, l'editor i la gestió d'arxius i directors.

El primer bloc és el compilador, en aquesta part ens encarreguem d'obtenir els arxius que l'usuari ha creat i compilar-los en funció del llenguatge que el professor ha definit al problema. És a dir si un problema es defineix per realitzar-lo en C++ el compilador a executar serà el GCC. A més s'ha de tenir en compte si el llenguatge ha de ser interpretat o compilat i en funció del que es demani es cridarà una funció o una altra per executar-lo.

El segon bloc és l'editor, aquest és a la vegada el bloc més complet. Per poder crear un editor que pugui mostrar els errors sintàctics que es fa per cada llenguatge hem decidit importar una llibreria de JavaScript anomenada ACE referència [8]. Aquesta llibreria ens ofereix una sèrie de funcions i comandes per poder crear un editor i que pugui identificar els errors sintàctics sense gran problema com es pot veure a la imatge adjuntada a aquest apartat. Entre altres coses també ens ofereix l'opció d'afegir comandes per teclat com CTRL+S que ens ajudarà a guardar el codi sense necessitat de prémer un botó. Cal recalcar que les funcions per guardar i actualitzar el fitxer es fan de manera asíncrona així l'usuari pot executar aquestes funcions i seguir editant el codi sense cap problema.

Una de les parts més complicades de l'editor va ser el sistema de gestió d'arxius i directors. S'ha de tenir en compte que una vegada s'obre un arxiu s'ha de crear un directori amb l'email de l'alumne i el títol del problema. Si no fem això podem perdre informació i que es sobreescriguin els problemes. Per tant aquesta part va ser una de les més delicades. Per això es van haver d'obrir diferents funcions que passaven per escaneig del directori, crear nous arxius, emmagatzemar l'arxiu i actualitzar la base de dades.

Finalment vam aconseguir un editor web que era capaç de compilar un programa que un estudiant podria redactar i editant una base que el professor hauria creat prèviament. Cosa que ens oferia una finestra amb un output de l'execució que s'ha iniciat. Que pot ser tant els errors de compilació com la solució de l'execució del codi.

Hem de tenir en compte que les funcionalitats d'aquest editor poden ser encara més com per exemple un sistema per refer el que s'ha fet o un botó per fer zoom i que es vegi bé el codi.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6 #include <sys/socket.h>
7 #include <sys/time.h>
8 #include <sys/stat.h>
9 #include <sys/mman.h>
10 #include <sys/uio.h>
11 #include <sys/wait.h>
12 #include <sys/resource.h>
13 #include <sys/epoll.h>
14 #include <sys/eventfd.h>
15 #include <sys/signalfd.h>
16 #include <sys/timerfd.h>
17 #include <sys/sem.h>
18 #include <sys/shm.h>
19 #include <sys/mount.h>
20 #include <sys/fs.h>
21 #include <sys/procfs.h>
22 #include <sys/utsname.h>
23 #include <sys/param.h>
24 #include <sys/limits.h>
25 #include <sys/unistd.h>
26 #include <sys/syscall.h>
27 #include <sys/sysinfo.h>
28 #include <sys/syslog.h>
29 #include <sys/sysmacros.h>
30 #include <sys/sysfs.h>
31 #include <sys/syslog.h>
32 #include <sys/syslog.h>
33 #include <sys/syslog.h>
34 #include <sys/syslog.h>
35 #include <sys/syslog.h>
36 #include <sys/syslog.h>
37 #include <sys/syslog.h>
38 #include <sys/syslog.h>
39 #include <sys/syslog.h>
40 #include <sys/syslog.h>
41 #include <sys/syslog.h>
42 #include <sys/syslog.h>
43 #include <sys/syslog.h>
44 #include <sys/syslog.h>
45 #include <sys/syslog.h>
46 #include <sys/syslog.h>
47 #include <sys/syslog.h>
48 #include <sys/syslog.h>
49 #include <sys/syslog.h>
50 #include <sys/syslog.h>
51 #include <sys/syslog.h>
52 #include <sys/syslog.h>
53 #include <sys/syslog.h>
54 #include <sys/syslog.h>
55 #include <sys/syslog.h>
56 #include <sys/syslog.h>
57 #include <sys/syslog.h>
58 #include <sys/syslog.h>
59 #include <sys/syslog.h>
60 #include <sys/syslog.h>
61 #include <sys/syslog.h>
62 #include <sys/syslog.h>
63 #include <sys/syslog.h>
64 #include <sys/syslog.h>
65 #include <sys/syslog.h>
66 #include <sys/syslog.h>
67 #include <sys/syslog.h>
68 #include <sys/syslog.h>
69 #include <sys/syslog.h>
70 #include <sys/syslog.h>
71 #include <sys/syslog.h>
72 #include <sys/syslog.h>
73 #include <sys/syslog.h>
74 #include <sys/syslog.h>
75 #include <sys/syslog.h>
76 #include <sys/syslog.h>
77 #include <sys/syslog.h>
78 #include <sys/syslog.h>
79 #include <sys/syslog.h>
80 #include <sys/syslog.h>
81 #include <sys/syslog.h>
82 #include <sys/syslog.h>
83 #include <sys/syslog.h>
84 #include <sys/syslog.h>
85 #include <sys/syslog.h>
86 #include <sys/syslog.h>
87 #include <sys/syslog.h>
88 #include <sys/syslog.h>
89 #include <sys/syslog.h>
90 #include <sys/syslog.h>
91 #include <sys/syslog.h>
92 #include <sys/syslog.h>
93 #include <sys/syslog.h>
94 #include <sys/syslog.h>
95 #include <sys/syslog.h>
96 #include <sys/syslog.h>
97 #include <sys/syslog.h>
98 #include <sys/syslog.h>
99 #include <sys/syslog.h>
100 #include <sys/syslog.h>

```

5 *Format del editor*

6.5 CREACIÓ DE PROBLEMES

El formulari per crear problema va ser una funcionalitat no difícil però sí completa. Ja que havíem d'emmagatzemar els arxius que el professor podria indicar com codi en una ruta adequada per poder tenir accés després des d'altres directoris. Una vegada solucionat aquest petit problema hem creat un formulari on verifiquem que l'extensió dels arxius sigui .cpp o .h per codi de C++ i .py o .python per codis de Python d'altra manera l'arxiu que s'importi no serà acceptat. Hem d'assegurar-nos que encara que no tenim una assignatura creada per si ni un sistema de creació d'assignatures, hem d'afegir a cada problema l'identificador de l'assignatura el qual pertany i així evitarem errors en un futur quan creem les assignatures.

6.6 MOSTRACIÓ D'ASSIGNATURES I PROBLEMES

El sistema per mostrar assignatures i problemes ha sigut senzill gràcies a l'arquitectura MVC que hem implementat. Ja que gràcies a això podem passar les dades d'una assignatura a un problema per realitzar la cerca sense un sistema complex de comunicació, sinó només amb peticions get.

En aquest moment ja s'està arribant a la fase on la web ja funciona correctament i el sistema funciona. Primerament, ens vam encarregar dels errors i bugs que vam trobar en la fase anterior, ja que per continuar amb la web era necessari que tot funcioni correctament.

6.7 EDITOR DE CODI INTERACTIU

Com a novetats vam començar amb el desenvolupament core de l'aplicació i per la qual s'ha creat el TFG. En aquesta part es va començar la implementació d'un sistema automàtic on el professor obtenia el control del codi de l'estudiant i el podria editar de manera interactiva, la funcionalitat a crear era que una vegada el professor editi el codi de l'estudiant aquest no té opció a editar-lo i, per tant, no hi hagi problemes d'editor de fitxer.

6.8 PREPARACIONS DEL CODI

Per crear aquest sistema s'han de fer uns preparatius abans de començar amb l'edició dinàmica. Per això primer es va crear una llista per obtenir els estudiants que havien obert el codi del professor i havien començat a editar-lo. Aquest llistat mostrarà el seu correu electrònic i quan el professor accedeixi a un estudiant l'editor del professor canviarà per l'editor de l'estudiant.

6.9 INTERACTIVITAT

Per tenir el codi interactiu amb l'estudiant s'han afegit noves funcionalitats per poder fer-ho. Primer s'han definit intervals d'un segon per guardar l'arxiu. Després s'ha modificat la taula de solucions per afegir una nova columna "editant" aquesta columna s'encarrega d'assegurar-nos si el professor està editant el codi o no. Amb això s'han afegit una Update a la base de dades una vegada el professor entra a editar el codi de l'estudiant perquè l'estudiant automàticament deixi de poder editar.

Per la part de l'estudiant en tot moment queda a l'espera que el professor editi o deixi d'editar el seu codi i en cas que el professor comenci a editar-lo l'estudiant deixarà de tenir poder en l'editor i entrarà en un mode de només lectura.

Per altra banda, el professor una vegada surti de la pàgina de l'estudiant s'ha creat un sistema que torna a modificar la base de dades i canvia a un estat on l'estudiant pot tornar a modificar el codi sense cap problema. Per donar-li més realisme al codi s'ha afegit un mètode que emmagatzema el codi editat cada vegada que s'escriu, d'aquesta manera l'estudiant tindrà una visió en directe de l'edició del professor.

6.10 CANVIS INTERFÍCIE

Una vegada la web ja funciona correctament i sense errors es van començar a fer petits canvis a la interfície d'usuari per fer-la "user friendly". Els principals canvis van ser en afegir nous estils i icones de les llibreries de "Font Awesome" Referencia [6] i "W3schools" Referencia [7] per obtenir animacions i interfícies de llistes prefabricades. A més a més se'n van canviar els formats de les llistes d'arxius en el compilador perquè es vegin una mica més alineats i es va modificar l'estat del footer perquè es quedi en tot moment en una posició fixa.

Un dels canvis més importants en l'àmbit d'interfícies va ser el sistema de redirecció i d'enllaços limitats per usuari. Això feia que en cas que un usuari tinguí un enllaç per una pàgina a la qual no té accés aquest no servirà per res. A més que s'ocultaven els botons per evitar així possibles errors de confusió.

6.11 SEGURETAT EN EL COMPILADOR

Una de les coses més importants que tenien que tenir en compte era afegir seguretat al compilador i que no es pogui afegir qualsevol mena de codi maliciós que pugui posar en compromís les dades de la base de dades o la

integritat del servidor. Per tant, s'han afegit unes premisses abans de compilar el codi per assegurar-nos del fet que no es puguin importar llibreries que tinguin accés al sistema. Inicialment, aquestes restriccions estan només pel codi de Python, però la intenció és afegir-lo per tots els compiladors.

La manera d'ús és senzilla, el fitxer que es passarà al compilador serà validat perquè no importi les llibreries de System i os que són les encarregades d'editar fitxers i modificar paràmetres del servidor, les altres llibreries com NumPy no quedaran afectades.

La fita 3 va ser l'encarregada de definir els últims detalls de l'aplicació a més de realitzar totes les proves funcionals i de rendiment de l'aplicació. Els principals objectius eren tenir la web operativa per poder publicar-la i així poder dur a terme les proves amb altres usuaris.

Per aconseguir això primerament es van haver de complir tots els requisits que es van demanar al principi de la planificació i després redactar els testos funcionals abans de poder publicar-la.

6.12 PUBLICACIÓ DE LA WEB

Ja que la web era funcional i ja es podia executar el compilador es va utilitzar el meu ordinador personal com a servidor de hosting i així poder publicar la nostra aplicació a la web. Per fer això es va descarregar el client No-IP per poder tenir un DNS i poder accedir des d'un enllaç.

Una vegada hem assignat l'enllaç a l'ordinador que anava a actuar com a servidor es va configurar el router de casa per redirigir el tràfic al port 80 de la IP pública de l'ordinador a la IP local d'aquest.

6.13 MODIFICACIO DEL PROBLEMA

Un requisit important de la web era que el professor tinguí permisos per editar i modificar el problema, a més de canviar la seva visibilitat. Per això es va afegir un menú de botons dins del llistat de problemes pel professor.

La primera funcionalitat del menú de botons era canviar la visibilitat del problema. Per aquest cas s'hauria de canviar de públic a privat i viceversa, a més a més s'havia de canviar de manera dinàmica el text per mostrar de manera visual l'estat del problema.

La funció per eliminar problema era el següent botó que es va afegir. Aquest consistia en el fet que ens donava la possibilitat d'esborrar l'exercici de la base de dades i tant professors com alumnes no podrien tornar a accedir a aquest.

El tema de l'edició de les característiques del problema es va dividir en dues parts. La primera part consistia a modificar petits fragments del codi, ja sigui per pujar una solució com per corregir petits errors que s'hagin pogut originar, a més també es pot afegir nous fitxers. D'aquesta manera quan l'alumne accedeixi al problema li sortirà una notificació a mesura que el problema ha sigut editat, i en cas que l'alumne vulgui, podrà descarregar-se la nova versió i si no es pot quedar amb la versió actual.

La segona part de l'edició era un altre botó que canviava les metadades de l'exercici. És a dir, es podia canviar el títol i la descripció, l'ús de memòria, el temps màxim d'execució a més de poder afegir nous fitxers en cas que així requereixi.

6.14 OBSERVACIO ALUMNE

La opció de poder veure lo que un alumne estava fent era un requeriment important. Ja que l'editor interactiu bloquejava l'editor de l'estudiant i per tant podria arribar a molestar a un alumne que estigui treballant. Per aquesta raó es va haver de definir un estat on el professor pugui passar a veure que etsan fent els estudiants sense afectar al seu editor de codi. D'aquesta manera a la llista d'estudiant es va afegir un botó per visualitzar als alumnes. Aquest boto redirigeix al editor del estudiant sense afectar a la seva dinàmica de treball.

7 TEST

En les primeres versions del codi el test no ha sigut una part en la que ens hem focalitzat suficientment això es degut a que molt del codi desenvolupat probablement seria substituït per un altre. Per a mesura que avancen les setmanes hem començat a desenvolupar codi que si que mantindrem en l'aplicació i aquí es on entra la part del test, en un començament hem fet proves funcionals de la aplicació ja que criem que es una bona manera de trobar errors.

7.1 PROVES FUNCIONALS

Principalment s'ha creat una fulla de càlcul de google on la intenció seria portar un seguiment del testing i l'ordre de les funcions. Aquest document (Test de la web del TFG) "Referencia [5]" consta de diferents fulles i en cada fulla s'enfoca en una funcionalitat de l'aplicació.

Cada fulla té un nom significatiu per cada funcionalitat de la aplicació i això ajuda a identificar les diferents tasques que es volen testear. En cas de múltiples versions de la aplicació s'haurà de clonar el document sencer amb totes les fulles i tornar a passar els test amb els nous de les noves funcionalitats per poder assegurar-nos de que tot funciona correctament.

Cada fulla esta formada per 5 columnes de dades. Cada columna te un significat. La primera es la columna de casos aquesta es la encarregada de oferir una breu explicació del test que es vol desenvolupar. Després tenim la columna de la descripció, aquesta en la que ens defineix les característiques del test, els inputs i outputs i la manera de executar-lo. La columna de resultat es una llista que selecciona els diferents resultats que pot tenir aquell test en concret. Els valors passen de "Tot bé", que significa que el test ha passat sense errors, "Error", que significa que el test no ha passat amb els criteris demanats, "No testejat", serà el estat en el que es troben totes les tasques

que no s'hagin testejat, i finalment la opció de "Bloquejat", aquest serà una opció per remarcar que no es pot passar el test ja que hi ha una tasca que bloqueja el desenvolupament del test. La següent columna es la anomenada Bug, aquesta es la encarregada de mostrar amb una breu explicació els bugs trobats amb petites dades que poden servir a l'hora de resoldre el problema. Finalment tenim una columna anomenada comentaris que servirà per afegir dades que creguem rellevant a la hora de desenvolupar una tasca, ja pot ser un detall per la execució o un petit indicador de per què passa algun flux d'execució.

Un exemple dels test que s'han redactat es el de login. Aquí es poden veure totes les columnes i el ús que se li dona a cada columna.

	A	B	C	D	E
1	Casos	Descripció	Resultat	Bug	Comentaris
2		Ingresar email y contraseña correctos: - verificar que redirige a la home - verificar que el el header cambia con el nombre de usuario y email	Todo Bien	-	
3	Correcto	Ingresar email incorrecto y contraseña correctos: - verificar que no redirige a la home - verificar que sale mensaje de error	Todo Bien	-	
4	Fallo email	Ingresar email correcto y contraseña incorrecta - verificar que redirige a la home - verificar que el header esta vacio	Todo Bien	-	
5	Fallo contraseña	Ingresar email correcto y contraseña corta - verificar que no redirige a la home - verificar que muestra mensaje de error	Todo Bien	-	
6	Contraseña corta	Ingresar email correcto y contraseña larga - verificar que no redirige a la home - verificar que muestra mensaje de error	Todo Bien	-	
7	Contraseña larga	Ingresar email corto y contraseña correcta - verificar que no redirige a la home - verificar que muestra mensaje de error	Todo Bien	-	
8	Email corto	Ingresar email largo y contraseña correcta - verificar que no redirige a la home - verificar que muestra mensaje de error	Todo Bien	-	
9	Email largo	Iniciar sesion con los campos vacios - verificar que no redirige a la home - verificar que no carga datos en el header	Todo Bien	-	Accedía ya que no verificamos ese campo. SOLUCIONADO
10	Campos vacios	Iniciar sesion como profesor: - verificar que salen los menus a los que puede acceder el profesor	Requerido	-	El inicio de sesion del profesor aun no esta creado y falta que se verifiquen los datos de sesion una vez logeado
11	Profesor	Iniciar sesion como estudiante: - verificar que tiene acceso a los menus de estudiante pero no tiene acceso a los menus del profesor	Requerido	-	El inicio de sesion del alumno esta creado y falta que se verifiquen los datos de sesion una vez logeado
	Estudiante				

6 Redacció tests de la web

7.2 SOLUCIO DE BUGS

Una de les funcionalitats més importants que no funcionaven era el registre i inici de sessió de professor a més del feedback de la web amb la validació del JavaScript.

Per tant, primerament es van solucionar els problemes de JavaScript, ja que era difícil registrar-se amb errors en el sistema que donava informació en cas que un paràmetre no funcionés correctament. Seguidament, vam començar amb l'inici de sessió i registre del professor.

Per aconseguir l'inici de sessió del professor vam haver de modificar la base de dades i afegir noves funcionalitats, aquestes eren perquè un professor es pugui enregistrar aquest hauria de tenir una invitació per part d'un altre professor, ja que el sistema no té un usuari administrador, els professors actuaran com tal. Per fer això es va haver de crear una nova taula a la base de dades anomenada "Token", aquesta taula serà l'encarregada de guardar els tokens que generen els professors per convidar a uns altres. Una vegada tenim la taula creada creem una funcionalitat per generar tokens aleatoris i que es guardin a la base de dades, aquests tokens només tindran poder de crear-los els professors. En un modal es mostra l'enllaç al registre amb el token generat.

Una vegada dins el registre validem que el token és vàlid i en cas que sigui vàlid es podrà registrar com a professor en cas contrari no es podrà registrar. Si està registrat, en l'inici de sessió es valida que la persona sigui estudiant o professor en cas que sigui estudiant se li assigna

una variable de sessió (1) i en cas que sigui professor se li assignarà la variable de sessió (0).

Un dels altres problemes a solucionar era el compilador d'arxius. Aquest es carregaven en una carpeta temporal i després es compilaven. La qüestió era que en executar-se en un fitxer auxiliar els altres fitxers que es creaven no funcionaven i, per tant, no es podia importar classes o llibreries que nosaltres havíem creat. Per solucionar aquest punt vam haver de modificar la ruta d'execució i obtenir la direcció absoluta del servidor per anar directament a la carpeta de la solució i executar-ho directament.

Un dels últims bugs solucionats tenien a veure amb la creació de problemes. Aquests problemes es creaven hardcodejats per una assignatura en concret i es va haver de canviar perquè la creació de problemes es pugui fer des de dins d'una assignatura i, per tant, s'assigni a aquesta.

8 CONCLUSIONS

Tal com hem pogut comprovar s'ha realitzat amb èxit la gran part del projecte. Això vol dir que les tasques de captació de requeriments, disseny i planificació es van fer amb una bona precisió. Per la part de la planificació, vam sofrir petits canvis durant les primeres fites. Aquests canvis principalment suposen una generalització de les tasques, ja que en un principi es va tractar de ser bastant precís i torbàvem errors. El treball de captació de requisits va ser una de les parts més complicades i a la vegada més útils del projecte. Perquè havíem d'investigar noves eines de desenvolupament i noves tècniques. Per això va ser un procés bastant llarg, però molt útil perquè a l'hora d'acabar aquesta fase teníem una bona base pel disseny i desenvolupament.

El desenvolupament d'aquest projecte va anar per fites. La primera fita va ser l'encarregada d'assentar les bases del projecte, és a dir, vam aconseguir totes les tasques prèvies al desenvolupament del codi. Sense aquesta primera fita hauria sigut impossible el desenvolupament del projecte.

Les següents fites van anar com estava previst a la planificació. Molts problemes que ens trobàvem en el camí els teníem ja solucionats gràcies al treball de les primeres fases del projecte. Fins a arribar a la fase de test, en aquest punt se'n van redefinir alguns objectius per la millora del sistema i que sense aquest apartat hauria sigut impossible d'identificar.

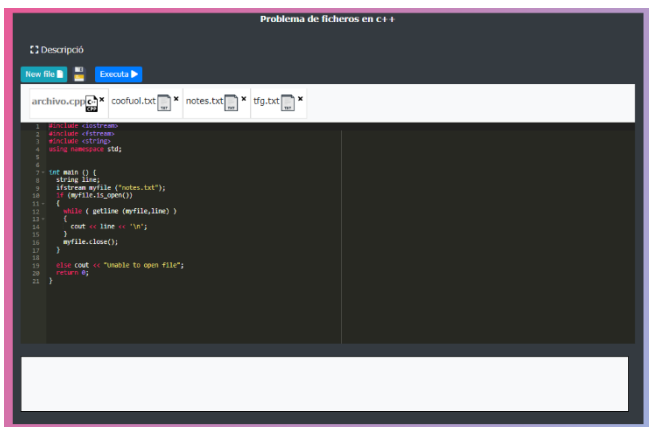
En aquest punt podem confirmar que solucionem la qüestió que es va plantejar al principi. Ara mateix els professors podran millorar la seva qualitat de les classes, ja que podran agilitzar bastant el procés d'ensenyament d'algorismes i correcció d'aquests. A més a més degut a la situació sociosanitària es podrà millorar la seguretat de les classes i els estudiants podran treballar més a gust gràcies a aquesta nova eina.

Per acabar cal esmentar que aquest projecte no s'hauria pogut realitzar en el temps d'un semestre si no hagués estudiat Enginyeria Informàtica. Gràcies a aquest grau s'han aconseguit les capacitats d'anàlisi i síntesi d'un problema i sobretot les habilitats per poder solucionar aquest problema amb una qualitat acceptable tant de codi com seguretat.

8.1 ASSOLIMENT D'OBJECTIUS

Per acabar hem de comentar que hem assolit la gran part d'objectius que vàrem planificar al principi. És veritat que encara falta una gran varietat de configuracions i adaptacions perquè funcioni correctament i es pugui escalar a alt nivell. Però per una classe grupal de 80 o 90 alumnes, ja seria operativa aquesta web.

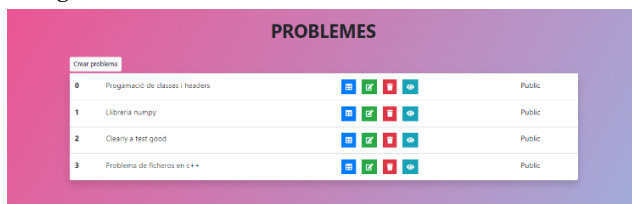
Els principals propòsits obtinguts són que tenim un editor web funcional amb un compilador, aquest agafa el codi que li passa per paràmetres i executa el directori on es troba. Com podem veure en la imatge d'avall no només és un editor, sinó que també implementa moltes funcionalitats, fins i tot algunes que no estaven planificades com salvar el codi mitjançant la comanda Ctrl+S.



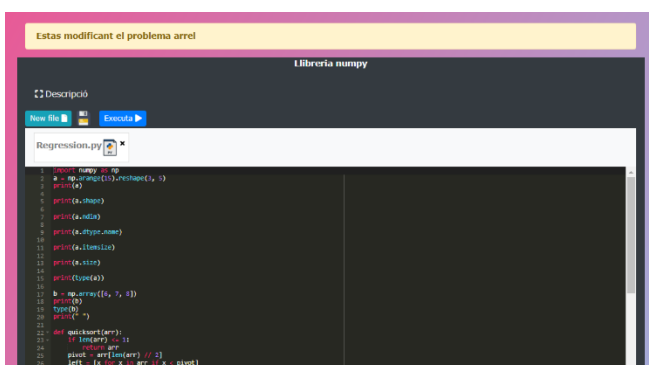
7 Editor amb compilador

També hem aconseguit poder crear i eliminar els problemes mitjançant un llistat de problemes, cal recalcar que la interfície és diferent depenent si ets alumne o professor, ja que les funcionalitats canvien.

A més a més hem afegit botons intuïtius perquè el professorat entengui correctament que ha de fer cada cosa sense gaire esforç.

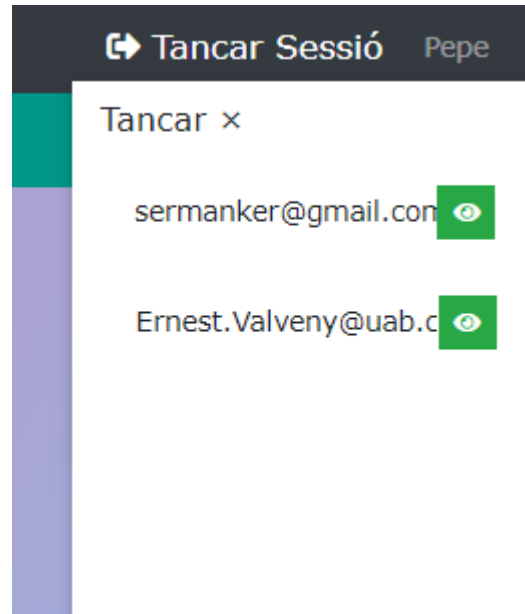


8 Llistat de problemes



9 Edició del problema

Una de les funcionalitats core que es van planificar era el tema d'editar el codi dinàmicament, i no només això sinó que també poder observar als alumnes sense afectar el codi que estan realitzant. Aquesta funcionalitat requeria múltiples funcionalitats per crear un sistema estable i sense errors. Encara que hi hagi petits errors de visualització i petites restriccions, les dues funcionalitats funcionen correctament.



11 Llistat d'alumnes per editar-los

Per acabar una de les funcionalitats més importants que es van aconseguir complir era tot el sistema de registre i inici de sessió d'alumnes i professors a més d'un sistema que convidi als professors perquè es puguin registrar.



12 Formulari de registre generic



13 Modal per convidar a nous professors.

8.2 OBJECTIUS NO ASSOLITS

Per altra banda, tampoc hem aconseguit tots els objectius que es van establir. Ja sigui per modificacions en la planificació que van fer-ho més complicat com per falta de temps. Els objectius no assolits són:

- Control de compilació d'altres fitxers. Hauríem de revisar en php que abans d'executar el codi no s'importin llibreries que puguin afectar al rendiment del servidor i sistema, per exemple llibreries (sys, os) de python.
- Control de timeout i bucles infinits. S'hauria de moure el servidor a un SO Linux per poder configurar les comandes d'execució en bash i poder definir un timeout. Quedant una comanda així (`timeout 5 /path/to/slow/ command with options`).
- Input des de el teclat. Hem de controlar els inputs per teclat per millorar l'experiència de programació.
- Control de memòria utilitzada. Hem de controlar la memòria utilitzada i poder limitar-la. Passa lo mateix que amb el sistema de timeout, amb Linux es mes senzill.
- Controlar el nom dels fitxers. A l'hora de crear els fitxers faria falta una expressió regular en JavaScript per assegurar-no de que no es crei un fitxer corrupte.
- Descarregar problemes. Falta afegir la funcionalitat per descarregar els problemes.
- Eliminar assignatures i problemes de manera definitiva. Es a dir aplicar recursivitat en els directoris per eliminar tant assignatures com problemes.

9 AGRAIMENTS

L'autor d'aquest projecte vol agrair a totes les persones que han ajudat d'alguna manera o altre a poder fer possible aquesta web, i per suposat, al tutor del TFG, Ernest Valveny per aconsellar i ajudar en tot el possible al desenvolupament del projecte i el seguiment del mateix i als meus pares que han estat allà en tot moment.

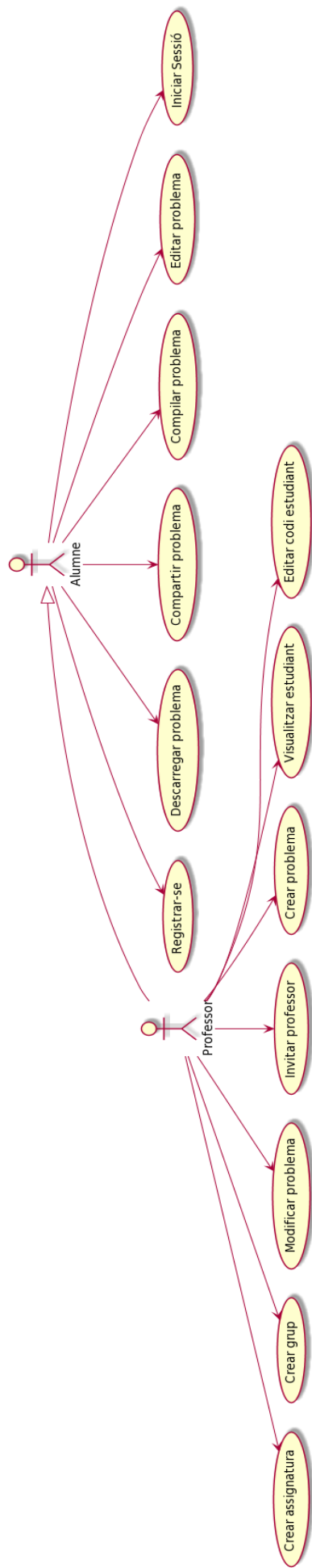
d

Bibliografia

- [1] LeetCode (2021) <https://leetcode.com/>
- [2] Hackerrank (2021) <https://www.hackerrank.com/>
- [3] Trello (2021) <https://trello.com/>
- [4] Google Drive (2021) <https://drive.google.com/drive/>
- [5] Test de la web del TFG Youssef Assbaghi (Octubre 2021) <https://docs.google.com/spreadsheets/d/1JLgbtozSzWTYMKMBUjsom2ovvzs4SYhBW1iJDyM-cp0/edit?usp=sharing>
- [6] Llibreria Font Awesome per obtenir repositori d'icones: <https://fontawesome.com/v5.15/icons?d=gallery&p=2>
- [7] W3 schools: <https://www.w3schools.com/w3css/4/w3.css>
- [8] Ace editor: <https://ace.c9.io/>
- [9]

APÈNDIX

A1. DIAGRAMA DE CASOS D'US



A2. DIAGRAMA E-R

