

Aula de programació interactiva

Arman Pipoyan Paronyan

Resum– L'objectiu d'aquest projecte és millorar una eina de suport docent afegint noves funcionalitats i refinant algunes que ja en té.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean facilisis mi ut sapien pulvinar, eget aliquet metus gravida. In at dapibus nibh. Interdum et malesuada fames ac ante ipsum primis in faucibus. Integer fringilla, lectus quis fermentum rutrum, libero metus rhoncus mi, sit amet consectetur orci massa eu risus. Nam odio massa, malesuada sit amet nisi sit amet, pellentesque maximus purus. Etiam ullamcorper, eros eget sollicitudin mollis, mauris ipsum congue enim, in ultrices orci sapien et turpis. Aliquam aliquam venenatis nibh et interdum. Praesent efficitur mauris diam, ut vestibulum purus luctus sit amet. Vestibulum in diam non dui facilisis facilisis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

Paraules clau– Eina de suport docent, programació on-line, docència on-line, desenvolupament de software

Abstract– The objective of this project is to improve a teaching support tool by adding new functionalities and refining some of the existing ones.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean facilisis mi ut sapien pulvinar, eget aliquet metus gravida. In at dapibus nibh. Interdum et malesuada fames ac ante ipsum primis in faucibus. Integer fringilla, lectus quis fermentum rutrum, libero metus rhoncus mi, sit amet consectetur orci massa eu risus. Nam odio massa, malesuada sit amet nisi sit amet, pellentesque maximus purus. Etiam ullamcorper, eros eget sollicitudin mollis, mauris ipsum congue enim, in ultrices orci sapien et turpis. Aliquam aliquam venenatis nibh et interdum. Praesent efficitur mauris diam, ut vestibulum purus luctus sit amet. Vestibulum in diam non dui facilisis facilisis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

Keywords– Teaching support tool, online programming, online teaching, software development



quantitat d'estudiants és molt gran i es disposa d'un temps limitat.

1 INTRODUCCIÓ

PER aprendre a programa s'ha de programar, això és el que diuen tots, però també és important equivocar-se i aprendre dels errors que es cometien. Una bona manera de fer-ho, i més si s'està a una classe, és preguntant a algú que sàpiga molt del tema, que a les aules acostuma a ser el professor. Gran part d'aquests errors són comuns, per tant, tenir la possibilitat d'explicar la solució d'un error a tot el grup tenint un exemple real al que assenyalar resulta molt útil, ja que alguns conceptes són molt difícils de deixar clar sense pràctica. Tot i ser una cosa tan important és una tasca que es dificulta quan la

Un altra problema que ens trobem és que no tots tenen les eines necessàries per poder programar, ja que alguns programes tenen uns requeriments mínims una mica elevats i no totes les màquines les poden executar. Per tant, poder programar en un navegador i que el codi s'executi a una màquina remota fa que tots els usuaris tinguin les mateixes condicions de treball.

Per aquests motius es va crear aquesta eina de suport a la docència que permet als estudiants escriure codi que pot ser revisat pel professor que està present a l'aula sense la necessitat de desplaçar-se a la seva taula i, a més, executar-ho remotament per poder veure el resultat de l'execució.

Cal tenir en compte que la primera versió d'aquest

- E-mail de contacte: arman.pipoyan26@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Ernest Valveny (Computació)
- Curs 2021/22

projecte es va desenvolupar durant la pandèmia del COVID-19 i que un dels objectius que es perseguia era el de disminuir el contacte entre persones i ajudar a que, si s'estava a l'aula les distàncies mínimes de seguretat es mantinguessin. Com actualment les restriccions han variat aquests objectius ja no tenen tant de pes com abans, però han sigut importants per fer que el projecte sigui com és.

El que s'explicarà en aquesta document són els canvis introduïts tant a nivell de funcionalitats com a nivell gràfic i el procés que s'ha seguit per tal de poder-los implementar.

2 ESTAT INICIAL DEL PROJECTE

La versió del projecte que s'havia de modificar [1] permetia als professors afegir assignatures i dins d'aquestes afegir, editar, amagar i eliminar problemes. També tenien la possibilitat de mirar, editar i executar l'última solució guardada de qualsevol problema dels alumnes.

Els alumnes tenien menys permisos, només podien resoldre problemes eliminant, afegit o modificant els fitxers que el professor establia en crear el problema i, a més, podien executar el codi per veure el resultat.

En quant al registre de nous usuari, els alumnes podien crear comptes nous sense cap problema, però els professors necessitaven una invitació per part d'altres membres de l'equip docent per tal de tenir permisos especials en crear-se el compte.

3 OBJECTIUS

Es volien introduir canvis a la web que ajudessin tant als professors com als alumnes a tenir més facilitats al utilitzar l'aplicació.

Cal dir que aquests canvis no modificaven l'objectiu principal de la primer versió que s'havia desenvolupat de la web, que era la creació d'una eina perquè els professors poguessin pujar problemes i que els alumnes els poguessin resoldre dins de la web, tenint sempre l'opció de poder compartir el codi amb el professor de manera senzilla perquè aquest ho pogués corregir i donar feedback.

3.1 Sessions de classe

L'objectiu d'aquesta funcionalitat era permetre als professors crear agrupacions de problemes que s'havien de comentar en una classe presencial. Durant les classes els alumnes haurien d'accedir a la sessió creada pel seu professor per tal de poder tenir la seva supervisió mentre resolien els problemes que s'ha proposat, ja que el professor només seria capaç de veure les respostes dels estudiants de la seva sessió.

Per satisfer aquest objectiu s'havia de modificar la base de dades afegint les taules 'session' i 'session_problems'. La primera taula serviria per relacionar una sessió amb un professor i assignatura mentre que la segona taula serviria per relacionar una sessió amb els problemes que s'han de

resoldre durant aquesta.

A continuació tenim una breu descripció dels casos d'ús d'aquesta funcionalitat que es van haver de completar per considerar-la acabada.

- **Iniciar sessions de classe:** Un professor havia de ser capaç d'iniciar una sessió de classe indicant un nou i un conjunt de problemes que havien de pertànyer a la mateixa assignatura que la sessió.

Per permetre que es poguessin obrir sessions, es va afegir un botó a cada ítem del llistat d'assignatures que portava al formulari de creació de sessions. Aquest formulari comptava només amb dos camps: el camp del nom i un selector que permetrà indicar quins exercicis s'havien de realitzar durant la sessió.

- **Veure sessions actives:** Tant professors com alumnes havien de poder veure les sessions actives d'una assignatura.

A cada ítem del llistat d'assignatures es va afegir un botó que portava al llistat de sessions actives. A aquesta llista només es mostrarà el nom de la sessió.

- **Accedir a sessions:** Els usuaris havien de poder accedir a una sessió i resoldre els problemes d'aquesta comptant amb l'ajuda del professor.

Es va decidir no implementar cap mesura de prevenció per evitar que els alumnes entressin a sessions que no els pertocava, ja que es va assumir que no estarien interessats en fer-ho.

- **Veure estudiants de la sessió:** Els professors havien de tenir un llistat amb tots els membres de la seva sessió i, a més, havien de poder accedir a les seves propostes de solució i editar-les si era necessari.

Poder veure els alumnes que estan resolent un problema i veure els canvis que feien eren funcionalitats ja existents que es van haver de modificar. El que es va fer va ser acotar els alumnes que un professor veia, és a dir, en comptes de veure a tots els alumnes que havien accedit a un problema només veien als que estaven a la seva sessió.

- **Tancar sessions de classe:** Els professors havien de ser capaços d'esborrar les sessions que consideraven ja acabades.

Es va haver d'afegir un botó a cada ítem del llistat de sessions l'opció de tancar la sessió, fent que s'esborrés definitivament de la BDD però deixant les propostes de solució dels problemes de la sessió perquè els alumnes poguessin accedir posteriorment.

3.2 Integració amb Github

L'objectiu d'aquesta integració era permetre al professor crear problemes utilitzant directament carpetes del seu repositori privat i als estudiants guardar el treball que fan durant les sessions per tal de poder accedir a aquests posteriorment sense accedir a la pàgina web.

Per implementar aquesta funcionalitat es va utilitzar

la REST API de Github [2], concretament el wrapper escrit amb PHP [3], que va facilitar la utilització de l'API i el tractament d'errors, ja que no s'havia de mirar la resposta es rebia, sino que en cas d'error saltaven excepcions, que eren més fàcils de tractar.

3.3 Integració amb Moodle

L'objectiu d'aquesta integració era permetre al professor vincular exercicis avaluable de Caronte amb problemes de l'eina, permetent als alumnes fer entregues des de la pàgina web sense la necessitat d'haver de descarregar el codi i tornar a pujar-ho al Moodle.

Per implementar aquesta funcionalitat s'havia d'analitzar com es feien les entregues d'exercicis a VPL, el plug-in que s'utilitza a Caronte per crear exercicis de programació.

3.4 Integració amb Jupyter Notebooks

Es volia donar la possibilitat de poder treballar amb Notebooks de Jupyter perquè al no ser fitxers de text pla, l'editor de la primera versió de l'eina no els visualitzava correctament.

Per implementar aquesta funcionalitat s'havia d'executar un servidor de Jupyter, ja fos en local o utilitzant un servidor remot, que servís els fitxers que demanava l'usuari i mostrar-los utilitzant un iframe d'HTML.

3.5 Redisseny de la pàgina web

L'objectiu principal d'aquesta tasca era unificar el disseny del que ja estava fet de la pàgina amb els canvis nous, però es va acabar fent un redisseny de tota la pàgina web.

- **Crear problemes amb fitxers de Github:** El professor havia de poder crear un problema indicant en comptes de fitxers que es troben a la seva màquina el link a una carpeta de Github. Els fitxers d'aquesta carpeta s'afegirien al problema quan es creés i, si es troba un fitxer amb la extensió ".md", el contingut havia de ser utilitzat com descripció del problema.

S'havia d'afegir al llistat de problemes un botó que permetés crear problemes utilitzant Github. A aquest formulari es demanava principalment el nom del problema i el link a la carpeta contenidora.

- **Pujar fitxers de Github a problemes:** L'estudiant havia de poder pujar fitxers a un problema que estigués resolent des d'un repositori de Github.

S'havia d'afegir a la vista de l'editor l'opció de pujar fitxers des de Github.

- **Guardar problema a Github:** L'estudiant havia de poder guardar tots els fitxers d'un exercici a un repositori.

S'havia d'afegir a la vista de l'editor l'opció de guardar els fitxers a Github. Quan es premés el botó s'havia de demanar el link a un repositori. Dins d'aquest repositori s'havia de crear una carpeta amb el nom del problema i guardar els fitxers endins.

4 METODOLOGIA

Per desenvolupar el projecte es va decidir utilitzar una metodologia Agile [4], utilitzant Trello [5] per poder organitzar les tasques i tenir una visió general del que s'ha fet i el que falta per fer. La durada dels sprints es va marcar que seria d'una setmana, intentant durant aquesta franja de temps tenir una porció del mòdul que s'estigui implementat finalitzat, assegurant així que es treballa amb constància.

Per tal de tenir sempre una versió estable del codi es va decidir utilitzar un programari de control de versions, concretament Github. Es tindrien dues branques actives: la principal, on estaria la versió estable del codi i la branca de desenvolupament, on s'implementarien les noves funcionalitats. Cada vegada que un mòdul estigués acabat i testejat els canvis es pujarien a la branca principal i es continuaria treballant a la segona branca.

5 PLANIFICACIÓ

El projecte es va dividir en 5 fases, com podem veure a la figura 1.

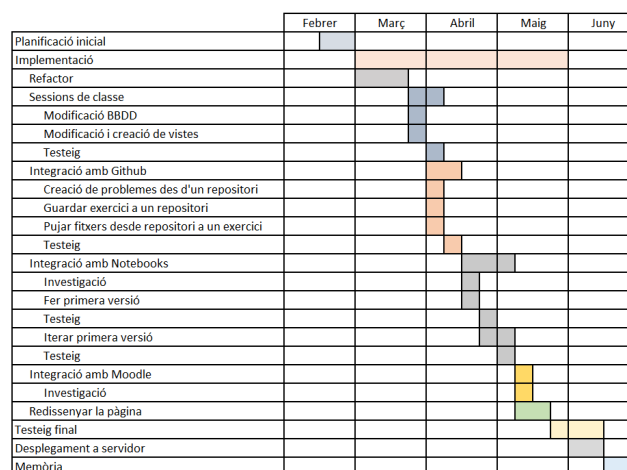


Fig. 1: Planificació del projecte

La primera d'aquestes fases era la de planificació, que consistia en analitzar els requeriments donats per part del tutor i decidir si eren viables i les subtasques que aquestes contenien.

La següent fase era la d'implementació, que tenia tants blocs com funcionalitats noves i canvis que s'han comentat a la secció 3.

La tercera fase era el testeig general de la web. Tot i que cada fase de desenvolupament contemplava també el temps necessari per provar que funcionava correctament, es va decidir dedicar aquest temps en comprovar que el conjunt de l'eina funcionava tal i com s'esperava.

La quarta fase consistia en fer una release del projecte a una de les màquines del centre docent per tal de poder-la provar en un entorn real per detectar possibles errors que tingués la web i també per explorar noves funcionalitats que calien afegir en un futur.

Durant la cinquena es redactaria la memòria final i es començaria a preparar la presentació.

6 DESENVOLUPAMENT

6.1 Refactorització

Aquesta tasca s'ha fet amb la intenció de familiaritzar-se amb el codi i el funcionament general de la web i per resoldre problemes d'estructura que tenia el codi.

Principalment, el que s'ha fet és esborrar mètodes i variables que no s'utilitzaven; traduir tot el codi a un únic idioma (l'anglès), ja que s'utilitzava tant l'anglès, com el castellà com el català per definir mètodes o variables; afegir comentaris al codi; introduir constants per facilitar la lectura del codi; modificar i estandarditzar el nom de variables, mètodes o fitxers que no eren autoexplicatius i que no sempre seguien el mateix format; dividir els fitxers molt grans en fitxers petits i finalment fer una anàlisi estàtica del codi utilitzant SonarLint [6].

S'han modificat fragments de codi difícils d'entendre com el que es pot veure a la primera imatge de la figura 2. En aquest cas, el que s'ha fet és moure el codi que es trobava a un fitxer '.php' a un fitxer '.js' per tal de no barrejar scripts amb HTML i PHP, a més, s'han afegit comentaris i s'ha modificat el nom de les variables per ajudar a entendre el codi, el resultat del refactor ho veiem a la segona imatge de la figura 2.

```
//Estudiante nunca podrá estar en modo reiterativo por ende no se puede hacer así
if (isset($_GET["reiteratiu"])) {
    if ($_GET["reiteratiu"]==1) {
        if ($_SESSION['idpp']==0) {?>
            <script>
                //editor.setReadOnly(true);
                var myVar = setInterval(save, 4000);
            </script>
        <?php }
    }elseif ($_GET["reiteratiu"]==2) { ?>
        <script>
            editor.setReadOnly(true);
        </script>
    }
    <?php }
}>
```

(a) Codi sense refactoritzar.

```
// Set the auto-check options depending on the user and his actions
userType = document.getElementById( elementId: "user_type").innerText;
// User type 1 is student and 0 professor
if (userType === "1") {
    setInterval(checkChanges, timeout: 3000);
} else if (userType === "0") {
    let viewMode = document.getElementById( elementId: "view_mode");
    // View mode 1 is edit mode and 2 read only
    if (viewMode) {
        let viewModeValue = viewMode.innerText;
        if (viewModeValue === "1") {
            setInterval(save, timeout: 4000);
        } else if (viewModeValue === "2") {
            editor.setReadOnly(true);
        }
    }
}
}
```

(b) Codi refactoritzat.

Fig. 2: Comparació abans i després del refactor.

6.2 Implementació de les sessions de classe

6.2.1 Modificació de la BDD

Tal com es va planificar, la nova estructura de la BDD incorpora la taula 'session' i 'session_problems'.

A més, s'han creat les següents relacions per integrar els canvis amb les taules existents:

- 'sessions' té una clau forana de 'professor_id' i 'subject_id' per indicar que aquest està creat per un professor i que pertany a una única assignatura.
- 'student' té una clau forana de 'session_id' per indicar que només pot estar connectat a una sessió.
- 'session_problems' conté claus foranes de 'session_id' i 'problem_id' per tal de representar que una sessió pot tenir molts problemes i que un problema pot estar present en més d'una sessió a l'hora.

A la figura 3 podem veure com queda la nova estructura.

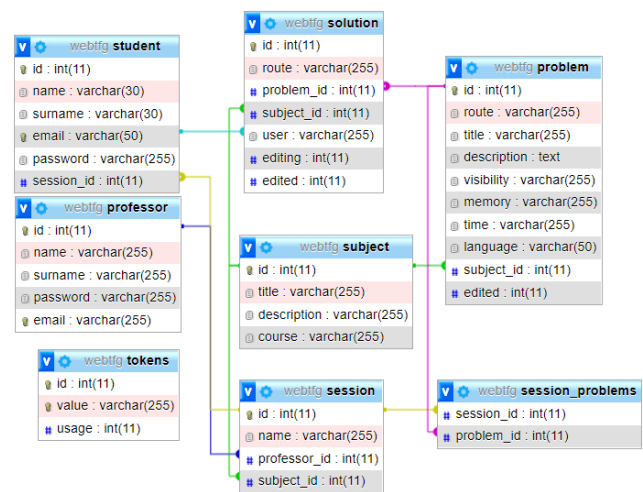


Fig. 3: Nova estructura de la base de dades.

6.2.2 Actualització de les vistes existents

Just com s'havia plantejat anteriorment, s'han afegit nous botons als ítems que apareixen al llistat d'assignatures, permetent accedir al llistat de sessions actives (si existeix alguna) i, només als professors, crear sessions (veure figura 4).

Per altra banda, el llistat d'estudiants que un professor pot veure a un problema no mostra a tots els alumnes, sinó només els que estan a la seva sessió.

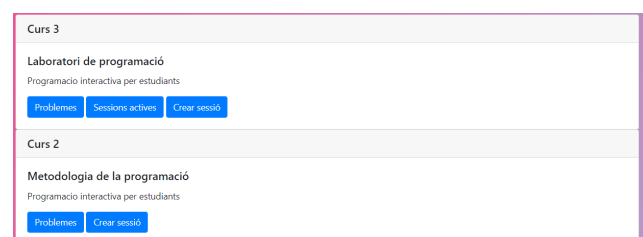


Fig. 4: Ítems del llistat d'assignatures.

6.2.3 Creació de vistes noves

S'han hagut de crear dues vistes noves, la llista de sessions i el formulari per crear una sessió (veure figura 5). Cal insistir en que no s'ha volgut treballar molt en el disseny perquè es farà durant les últimes setmanes.

Fig. 5: Formulari per crear una nova sessió.

6.3 Integració amb Jupyter Notebooks

L'objectiu d'aquesta integració és permetre al professor crear problemes que tinguin fitxers '.ipynb' i que aquests es mostrin als estudiants com els mostraria Jupyter.

6.3.1 Idea inicial

Durant els primers dies es va estar fent una tasca d'investigació per decidir si era viable o no i es va acabar conclouent que sí que era viable.

La idea que es tenia abans de començar, i la que es va acabar utilitzant, era crear un iframe d'HTML (una pàgina encastada) a la vista de l'editor que substituís l'editor per defecte que hi havia.

6.3.2 Primera versió

A la primera versió es va decidir entre utilitzar un servidor extern com Binder [7] o un servidor Jupyter que servís els fitxers '.ipynb' per tal de crear el contingut de l'iframe. Es va decidir fer servir el servidor extern, ja que aquest oferia un entorn totalment aïllat a cada estudiant on aquest podria executar qualsevol comanda disponible de Python (com per exemple intentar esborrar tots els fitxers de l'entorn). Desafortunadament, aquesta idea no es va poder dur a terme perquè Binder, a dia 20 de maig de 2022, no permet encastar els entorns que genera en altres pàgines web.

6.3.3 Segona versió

Per la segona versió es va crear un servidor de Jupyter que servia els fitxers accedint a un URL local. Aquest servidor s'executava a la carpeta on estaven totes les solucions dels alumnes fent que amb l'URL no es pogués accedir als fitxers del sistema.

Tenint el servidor en local els estudiants podien visualitzar i interactuar amb el Notebook com si ho tinguessin a la seva pròpia màquina (vegeu figura 6). Però això comportava un problema: tenien accés il·limitat a qualsevol fitxer

de la màquina on s'executava el servidor si executaven comandes de Python.

Com no es va trobar la manera de crear un pre-execute hook (una acció que s'executés abans de que el codi de l'estudiant fos interpretat) per evitar que s'executessin línies de codi malicioses, es va haver de modificar com estava plantejat el servidor Jupyter.

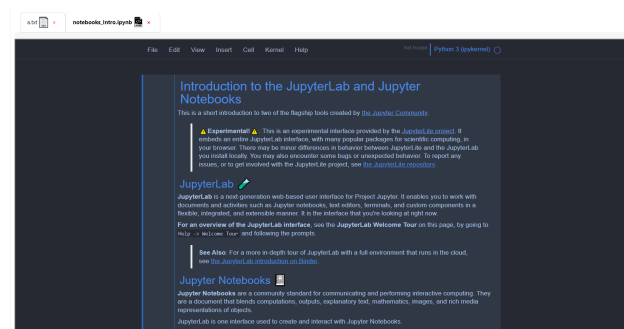


Fig. 6: Notebook visualitzat a la pàgina web.

6.3.4 Tercera versió

A la tercera iteració es va decidir utilitzar un contenidor de Docker que contingués el servidor de Jupyter. Això permetia aïllar el servidor de Jupyter de la màquina física on s'estava executant el codi.

Aquest contenidor tenia enllaçada la carpeta de solucions de la màquina contenidora amb la seva carpeta 'home' i se servien els fitxers d'aquesta carpeta. Això feia que amb l'URL només es pogués accedir a les solucions i en cas que un estudiant intentés executar codi maliciós no afectaria la màquina real.

Amb aquest canvi els estudiants només tindrien accés al contenidor i la màquina física no podria ser modificada. Però això encara no evitava que els alumnes poguessin accedir als fitxers dels altres fent servir comandes Python.

6.3.5 Quarta versió

A la quarta i última versió el que es va decidir fer va ser el següent:

1. Quan un estudiant entrés a un problema de Jupyter es crearia un contenidor per ell que només contindria les solucions dels problemes que ell havia fet. Aquest contenidor s'assignaria a un port de la màquina física i, tant el port com l'identificador del contenidor es guardarien a la sessió de l'usuari.
2. Quan l'estudiant sortís de la pàgina s'utilitzaria l'identificador que s'havia guardat i s'esborrarien tant el contenidor com les dades de la sessió relacionades amb el contenidor.

Això permetia als alumnes utilitzar qualsevol comanda al Notebook i assegurar que la màquina física estaria protegida contra estudiants que volguessin executar codi maliciós.

Es van fer algunes consideracions més:

- Es va tenir en compte que un estudiant podria tenir més d'una finestra activa que requerís un servidor de Jupyter. Per tant, es va fer que el contenidor fos un singleton. És a dir, a la sessió es guardaria quants problemes de Jupyter l'usuari té obert i fins que el comptador no arribés a 0, aquest no s'esborraria.
- Es va tenir en compte que a cada contenidor se l'havia d'escollar per un port diferent. Com els ports d'usuari van del 1024 al 49151 i el màxim d'alumnes que estarien utilitzant alhora el sistema seia 40, el que es faria era escollir un port aleatori i mirar si hi havia algun contenidor utilitzant aquell port, en cas de col·lisió es tornaria a escollir un port aleatori i tornar a fer la comprovació. Com el nombre de ports disponibles era molt superior als que es podrien arribar a ocupar simultàniament no es va trobar necessària la utilització de cap sistema complex de gestió de ports.

6.4 Integració amb Caronte

L'objectiu d'integrar l'aplicació amb Caronte era permetre al professor enllaçar els exercicis del Moodle amb els de la pàgina web. Això permetria als estudiants entregar els problemes que el professor marqués com exercicis avaluats amb un clic.

6.5 Investigació

Es va estar investigant si era possible realitzar la integració i el temps que es tardaria.

Després d'analitzar la informació disponible de l'API de Moodle, del plugin de 'Virtual Programming Lab' i analitzar el codi d'aquest per trobar coses que no s'especifiquessin a la seva documentació es va concloure que era impossible pels següents motius:

- S'hauria de tenir accés al Moodle de la universitat per poder configurar els serveis necessaris per poder fer consultes a aquesta des de la nostra aplicació. Una mala utilització d'aquests permisos podria representar una bretxa de seguretat si s'activaven serveis sense configurar-los apropiadament i es proveïssin amb rigor. Aquesta tasca requeriria utilitzar molt temps del projecte en tasques administratives que no formen part d'aquest.
- Tot i poder haver activat i configurat els serveis correctament hi havia un problema més. Revisant el codi disponible del plugin de VPL es va concloure que aquest és un mòdul autocontingut on els enviaments de fitxers es feien cridant a funcions que no estaven publicades a cap API.

6.6 Redisseny de la pàgina web

A l'anterior replanificació del projecte es va decidir afegir una tasca de redisseny de la pàgina web que es faria una vegada es tinguessin totes les tasques d'afegir noves funcionalitats (principalment les que requerien afegir o modificar

vistes existents). Fent això es podria avançar ràpidament amb les tasques sense haver de pensar massa a deixar la pàgina web molt presentable i, a més, com totes les modificacions de disseny es farien alhora l'estil seria més uniforme.

6.7 Tema fosc i tema clar

A causa de la cancel·lació d'una de les tasques (fet que es comentarà a la secció de planificació) es va decidir afegir la funcionalitat de canviar entre tema fosc i tema clar.

Aquesta és una funcionalitat que molts IDE tenen i que agrada molt als desenvolupadors perquè no tots tenen la mateixa preferència respecte al tema a utilitzar.

La preferència seleccionada per l'usuari es guarda en la seva sessió, és a dir, que quan aquest tanqui sessió el tema tornarà a ser el per defecte (que és el clar). Com es volia donar la possibilitat als nous usuaris de modificar el tema es va fer que si no es tenia una sessió iniciada les preferències es guardessin al navegador. És molt importat que es guardi perquè si no, en canviar de vista es tornaria al tema clar.

A la imatge (b) de la figura 7 podem veure una llista redissenyada que fa servir el tema clar i a la imatge (b) de la figura 8 podem veure un formulari utilitzant el tema fosc.

6.7.1 Llistats

Es va crear una vista genèrica parametrizable d'un llistat per tal de poder mantenir una consistència entre les vistes i facilitar i mecanitzar la creació de noves vistes d'aquest tipus (vegeu figura 7).

El funcionament d'aquesta vista era el següent:

1. Quan l'usuari accedia a una vista, es cridava al controlador d'aquesta.
2. El controlador s'encarregava de fer consultes a la BDD per obtenir els objectes que s'havien de mostrar.
3. Amb les dades dels objectes es generava un mapa hash que contenia: el títol de la pàgina, una llista de mapes hash addicional que contenia les accions visibles només pel professor (com el botó per crear objectes d'aquella vista) i, per últim, una llista de mapes hash més on estava la informació de cada ítem (principalment títol, descripció i llista d'accions de l'element).

Opcionalment, es podia afegir un camp més al mapa hash de la pàgina per crear finestres emergents indicant només el títol i el contingut d'aquest (que podia ser un conjunt de camps de formulari o un text).

6.7.2 Formularis

Pels formularis es va seguir la mateixa idea: es va crear una vista parametrizable per crear formularis de manera ràpida i fàcil. Amb aquest sistema afegir, treure i modificar camps es pot fer tocant només 1 línia del controlador.

A la figura 8 podem veure una comparació entre el



(a) Abans del redisseny.



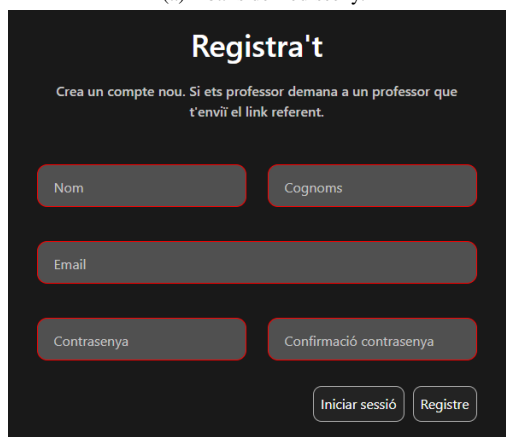
(b) Després del redisseny.

Fig. 7: Comparació d'abans i després del redisseny.

disseny anterior del formulari i el nou. L'estructura dels camps del formulari de registre s'ha mantingut, però s'han modificat els camps i els botons d'aquest.



(a) Abans del redisseny.



(b) Després del redisseny.

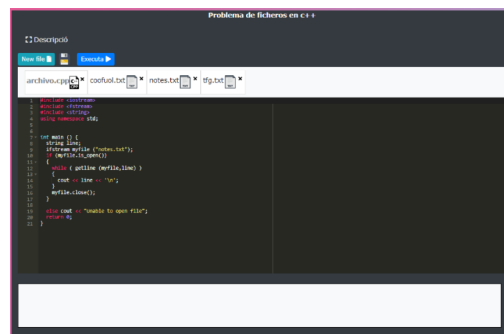
Fig. 8: Comparació d'abans i després del redisseny.

6.7.3 Editor

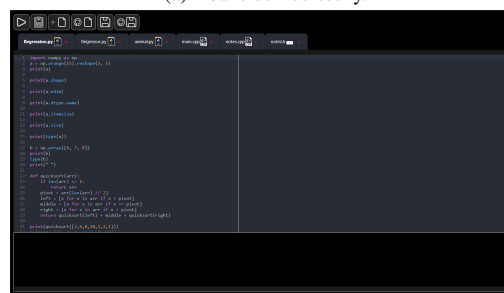
Per la vista de l'editor no es va fer servir cap vista genèrica ja que aquesta no es reutilitza.

Amb el redisseny de la pàgina es va fer l'editor més

gran, ja que era massa petit sent el component central d'aquesta vista. Es van modificar també els botons que apareixien a sobre de l'editor fent que aquests seguessin l'estil de les altres vistes. Com s'havia incorporat la funcionalitat de canviar entre tema fosc i tema clar es va haver de fer que l'editor es modifiqués com la resta de la pàgina. A la figura 9 podem veure la diferència entre l'editor anterior (imatge (a)) i el nou (imatge (b)) que està utilitzant el tema fosc com la resta de la pàgina.



(a) Abans del redisseny.



(b) Després del redisseny.

Fig. 9: Comparació d'abans i després del redisseny.

7 TESTEIG

TODO

8 CONCLUSIONS

Durant aquestes setmanes de recerca i desenvolupament s'ha après a treballar correctament amb PHP, HTML i JS separant clarament on s'utilitza cadascun. Com són llenguatges molt flexibles permeten fer moltes barreges que dificulten la lectura i mantenibilitat del codi. Durant la primera tasca que es va fer, la de refactorització, es va entendre perquè barrejar aquests llenguatges no era una bona idea i des d'aquell moment s'ha intentat seguir amb capes ben definides, tot i que a vegades no ha sigut possible.

8.1 Objectius assolits

El resultat ha sigut el que s'esperava i el disseny de la pàgina ha quedat millor del que s'esperava. És una aplicació que, amb les funcionalitats que ofereix, pot arribar a ser molt útil per als docents i s'espera que així sigui.

8.2 Objectius no assolits

TODO

REFERÈNCIES

- [1] Y. A. Asbahi, "Aula de programació interactiva." <https://shorturl.at/jkruG>, 2022.
- [2] "Github rest api." <https://docs.github.com/en/rest>.
- [3] "Github rest api wrapper for php." <https://github.com/KnpLabs/php-github-api>.
- [4] "What is agile?." <https://www.atlassian.com/agile>.
- [5] "Trello." <https://trello.com/en>.
- [6] "Sonarlint." <https://www.sonarlint.org/>.
- [7] "Binder." <https://jupyter.org/binder>.

APÈNDIX

A.1 Diagrama de casos d'ús

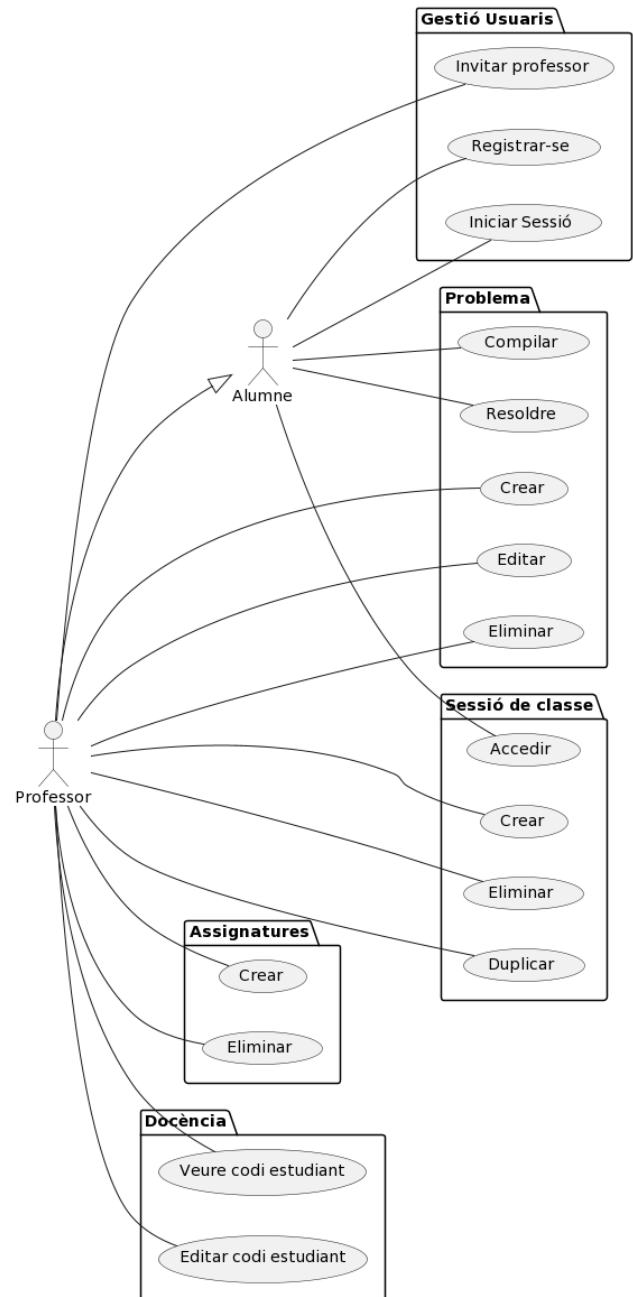


Fig. 10: Diagrama de casos d'ús de la web

A.2 Diagrama d'entitats i relacions

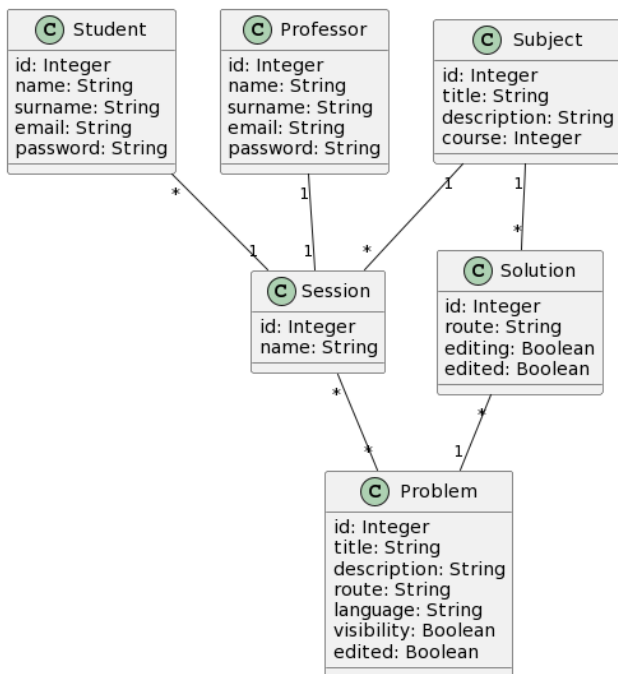


Fig. 11: Diagrama d'entitat-relació de la web