

Primer Informe de Progrés

Pipoyan-Paronyan, Arman

arman.pipoyan26@gmail.com
Bellatera, 2022

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

Per aquest motiu es va crear aquesta aplicació 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 [1].

2 Estat inicial del projecte

Al començament, el projecte permetia als professors afegir assignatures i dins d'aquestes afegir, editar, amagar i eliminar problemes, que tenien un títol, descripció, fitxers inicials i algunes dades que no s'utilitzaven, com per exemple el temps màxim d'execució i la memòria que pot utilitzar el programa mentre s'executa. Els professors també tenien la possibilitat de mirar, editar i executar l'última solució guardada dels alumnes de qualsevol problema.

Els alumnes tenien menys permisos, només podien resoldre problemes eliminant, afegit o modificant els fitxers que el professor estableix 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.

Van haver objectius que no es van poder aconseguir van ser, que es llisten a continuació:

- Controlar que no es puguin importar llibreries que afectin el rendiment del servidor o sistema.

- Controlar els timeouts i bucles infinits fent servir la variable de temps màxim d'execució d'un problema.
- Controlar la memòria que utilitza el programa utilitzant la variable que es defineix en crear un problema.
- Controlar els inputs per teclat per millorar l'experiència de programació.
- Permetre descarregar els problemes.
- Eliminar assignatures i problemes definitivament, és a dir, esborrar les carpetes del servidor.

3 Objectius inicials

Els objectius inicials que es van plantejar van ser:

- Refactorització del codi amb l'objectiu d'uniformitzar l'estil, arreglar errors existents al codi i fer neteja de mètodes i variables no utilitzades.
- Millorar el sistema de control d'accés a les activitats d'una assignatura per iniciar sessions interactives només si el professor vol.
- Analitzar la viabilitat d'integrar l'aplicació amb Caronte [2] utilitzant la api de Moodle [3] i realitzar la integració si és viable.
- Integrar l'aplicació amb git per tal de poder pujar i descarregar fitxers de Github [4] utilitzant la api que proporcionen [5].
- Analitzar la viabilitat d'integrar notebooks interactius de jupyter [6] a l'aplicació.

Integrar l'aplicació amb el moodle i jupyter podia ser una tasca molt llarga, per tant, es va pensar en dedicar un temps per analitzar la viabilitat i només realitzar-les si es tenia temps suficient. Si alguna d'aquestes integracions resultava ser molt difícil de fer en un temps acotat, s'analitzaran noves funcionalitats per afegir a l'aplicació.

4 Metodologia

Per desenvolupar el projecte es va decidir utilitzar una metodologia Agile [7], utilitzant Trello [8] per poder organitzar les tasques i tenir una visió general del que s'ha fet i el que

falta per fer. La durada dels esprints 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 inicial

La planificació que es va plantejar inicialment és la que podem veure a la figura 1.

	Febrer	Març	Abril	Maig	Juny
Planificació inicial					
Implementació					
Refactor					
Control d'accessos					
Integració amb Github					
Integració amb Notebooks					
Integració amb Moodle					
Testeig final					
Memòria					

Figura 1: Planificació inicial del projecte

La primera de les tasques consistiria en refactoritzar el codi sencer per tal de poder arreglar errors ja existents i per dividir el codi en mòduls més petits per tenir més facilitat per afegir els canvis planificats.

Es va planificar una fase final de testeig per tal de provar la aplicació sencera de forma manual, tot i que cada fase acabés tenint la seva pròpia fase de testeig.

6 Avenços

Durant aquestes setmanes s'ha estat treballant en la refactorització del codi i en la implementació de les sessions de classe. Per tal de decidir els requeriments d'aquesta funcionalitat i els de la integració amb Github, es va fer una captura de requeriments, els resultats del qual estan detallats a continuació.

6.1 Especificació de requeriments

Durant la reunió es va decidir que les funcionalitats prioritàries eren les que s'han comentat anteriorment. Cal destacar que aquesta decisió no modifica els objectius del projecte sinó que només defineix a quines tasques es donarà prioritat i, que durant la reunió no es va parlar del disseny de la pàgina web, per tant, és un tema al que encara no s'ha donat molta prioritat.

Els casos d'ús que es van decidir implementar per cada funcionalitat són els que veiem a la figura 2.

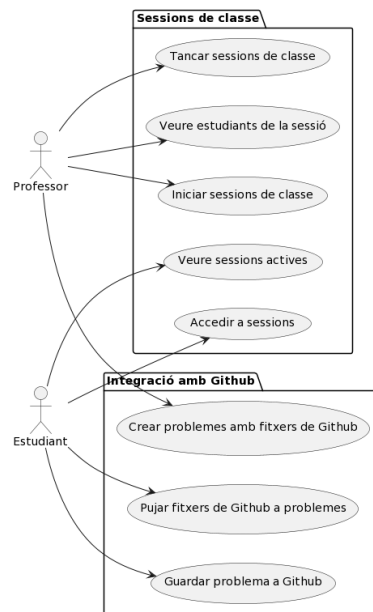


Figura 2: Diagrama de casos d'ús resultant de la reunió.

6.1.1 Sessions de classe

L'objectiu d'aquesta funcionalitat és permetre als professors crear agrupacions de problemes que s'han de comentar en una classe presencial. Durant les classes els alumnes hauran d'accedir a la sessió que hagi creat el seu professor per tal de poder tenir la seva supervisió mentre estan resolent els problemes que s'ha proposat, ja que el professor només podrà veure les respostes dels membres de la seva sessió.

Per satisfer aquest objectiu s'haurà de modificar la base de dades afegint les taules 'session' i 'session_problems'. La primera taula relacionarà a la sessió amb un únic professor i assignatura, per tant, haurà de tenir 4 camps: l'identificador únic, el nom de la sessió i els identificadors del propietari i l'assignatura a la que pertany. La segona taula s'utilitzarà per relacionar un llistat de problemes i una sessió, per tant, tindrà només dos camps: l'identificador de la sessió i l'identificador del problema. A la figura 3 tenim representades aquestes taules i les relacions amb les taules ja existents.

A continuació tenim una breu descripció dels casos d'ús d'aquesta funcionalitat i les tasques que s'han de completar per considerar-la acabada.

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

Per permetre que es puguin obrir sessions, s'afegirà un botó a cada ítem del llistat d'assignatures que portarà al formulari de creació de sessions. Aquest formulari només tindrà el camp del nom i un selector que permetrà seleccionar quins exercicis conformaran la sessió.

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

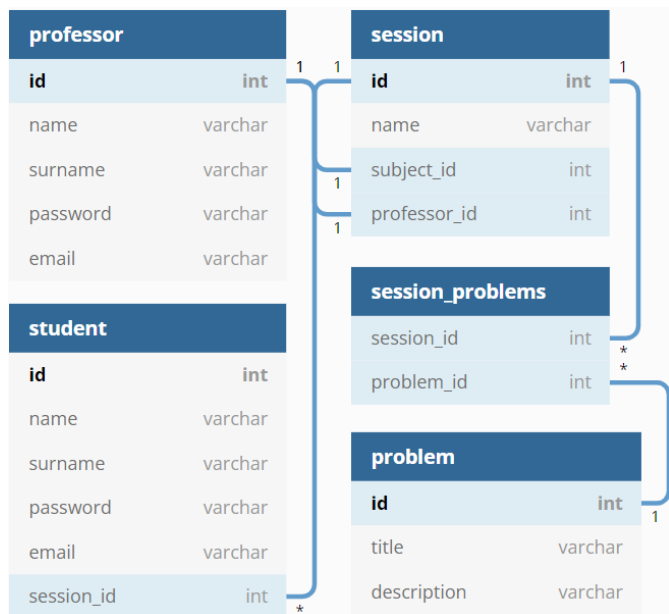


Figura 3: Diagrama de com es relacionaran les noves taules amb les ja existents.

Al llistat d'assignatures s'afegirà un botó que portarà al llistat de sessions actives. A aquesta llista només es mostrarà el nom de la sessió.

- **Accedir a sessions:** Els usuaris tindran la possibilitat d'accedir a una sessió i resoldre els problemes que aquest té amb el benefici de que el professor podrà veure el seu progrés. No s'implementarà cap mesura de prevenció per evitar que els alumnes entrin a sessions que no els pertoca, ja que s'assumeix que no estaran interessats en fer-ho.

- **Veure estudiants de la sessió:** Els propietaris de les sessions tindran una llista amb tots els membres de la seva sessió i, a més, podran veure els canvis que aquests estan realitzant.

Poder veure els alumnes que estan resolent un problema i veure els canvis que han fet eren funcionalitats ja existents que es modificaran. El que es farà és acotar els alumnes que un professor veu, és a dir, en comptes de veure a tots els alumnes que han accedit a un problema només es veurà als que a més estan a la seva sessió.

- **Tancar sessions de classe:** Un cop que el professor consideri que la sessió ha acabat podrà eliminar-la, esborrant totes les dades de la base de dades.

El professor tindrà al llistat de sessions, l'opció de tancar la sessió, fent que ja no sigui visible.

6.1.2 Integració amb Github

L'objectiu d'aquesta integració és permetre al professor crear problemes utilitzant directament carpetes que estan al seu repositori privat i als estudiants guardar el treball que fan durant les sessions per tal de poder accedir a aquests quan la sessió es tanqui.

Per implementar aquesta funcionalitat s'utilitzarà la REST API de Github [5], concretament el wrapper escrit amb PHP [9], que facilitarà la utilització de l'API i el tractament d'errors.

- **Crear problemes amb fitxers de Github:** El professor haurà 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'afegiran al problema quan es creï i, si es troba un fitxer amb la extensió ".md", el contingut serà utilitzat com descripció del problema. S'haurà d'afegir al llistat de problemes un botó que permetrà crear problemes utilitzant Github. A aquest formulari es demanarà el nom del problema i el link a la carpeta contenidora.

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

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

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

S'haurà d'afegir a la vista de l'editor l'opció de guardar els fitxers a Github. Quan es premi el botó es demanarà el link a un repositori, dins d'aquest repositori es crearà una carpeta amb el nom del problema on es guardaran els fitxers.

6.2 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 estandaritzar 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 [10].

S'han modificat fragments de codi difícils d'entendre com el que es pot veure a la primera imatge de la figura 4. 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 4.

6.3 Implementació de les sessions de classe

6.3.1 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:

```
//Estudiant no pot estar en mode reiteratiu per que no se queda hacer así
if (isset($_GET["reiteratiu"])) {
    if ($_GET["reiteratiu"]==1) {
        if ($_SESSION['tipo']==0) {>
            <script>
                //editor.setReadOnly(true);
                var myVar = setInterval(save, 4000);
            </script>
        <?php }
    }elseif ($_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.

Figura 4: Comparació abans i després del refactor.

- ‘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 5 podem veure com queda la nova estructura.

6.3.2 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 6).

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.

6.3.3 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 7). Cal insistir en

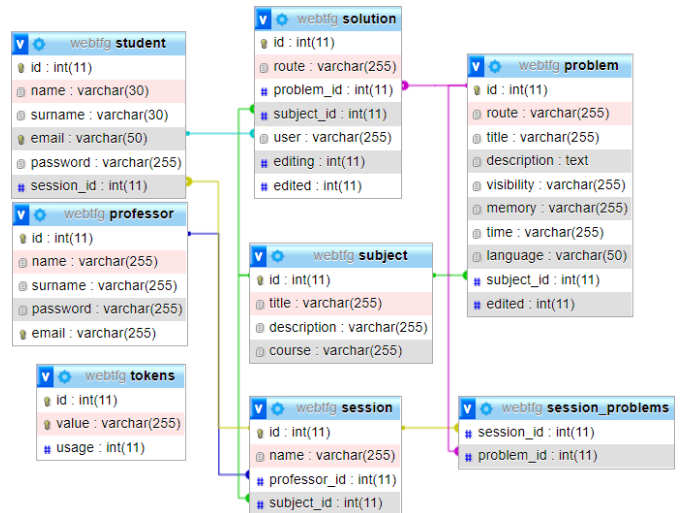


Figura 5: Nova estructura de la base de dades.

Figura 6: Ítems del llistat d’assignatures.

que no s’ha volgut treballar molt en el disseny perquè es farà durant les últimes setmanes.

6.4 Actualització de la planificació

Segons la planificació inicial, a mitjan abril la integració amb Github hauria d’estar acabada. Per ara no es porta cap retard, ja que s’han començat les tasques d’aquesta funcionalitat i s’estima que no es tardarà molt a tenir-ho acabat.

Tot i tenir anar bé de temps s’ha hagut de fer una petita re-planificació per no haver estimat correctament el que es tardaria en fer el re-factor i també degut a que no es va poder dedicar

Figura 7: Formulari per crear una nova sessió.

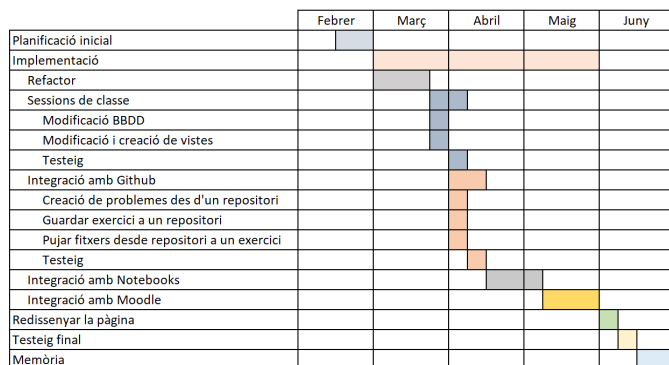


Figura 8: Planificació actual del projecte.

tant de temps com s'havia previst inicialment, principalment per altres entregues o exàmens.

Per altra banda, s'ha decidit incorporar una tasca nova per redissenyar la pàgina per fer que tot tingui un estil uniforme i sigui més atractiu a la vista. Aquesta tasca es farà durant la primera setmana de Juny, fent que la tasca de testeig final duri una menys de la que s'havia previst, però això no hauria de suposar cap problema ja que cada funcionalitat que s'implementa es testeja abans de pujar-ho al software de control de versions.

La nova planificació (veure figura 8) deixa constància del que s'ha comentat.

7 Metodologia utilitzada

La metodologia inicial que es va proposar i la que s'utilitzarà durant el desenvolupament del projecte és Agile [7].

Durant les primeres setmanes no s'ha notat la utilització d'aquesta metodologia ja que al estar fent una refactorització, on no s'implementa cap funcionalitat nova, no havia necessitat/possibilitat de mostrar canvis al client (en aquest cas el professor). Però un cop acabada la primera funcionalitat es va planificar una reunió per tal de verificar que el treball que es porta s'adequa a les expectatives.

Referències

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