

Segon Informe de Progrés

Pipoyan-Paronyan, Arman

arman.pipoyan26@gmail.com
Bellaterra, 2022

1 Introducció al treball

Per aprendre a programar 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 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 Introducció al document

En aquest document s'explicarà el treball realitzat des de l'anterior informe de progrés, que va ser entregat el dia 10 d'abril del 2022. A més, es parlarà sobre els canvis que hi ha hagut en la planificació, de com s'ha planificat la nova feina i s'exposaran unes conclusions provisionals sobre el treball.

3 Desenvolupament

Des de l'última vegada s'ha estat treballant en la integració amb Jupyter, s'ha investigat la integració amb Moodle i el redisseny de la pàgina web.

3.1 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.

3.1.1 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 concluint 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



Figura 1: Notebook visualitzat a la pàgina web.

pàgina encastrada) a la vista de l'editor que substituís l'editor per defecte que hi havia.

3.1.2 Primera versió

A la primera versió es va decidir entre utilitzar un servidor extern com Binder [2] 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.

3.1.3 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 1). 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 mali-

cioses, es va haver de modificar com estava plantejat el servidor Jupyter.

3.1.4 Tercera versió

A la tercera iteració es va decidir utilitzar un contenidor de Docker que continguéss 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.

3.1.5 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 oberts 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'escollir 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 seria 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.

3.2 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.

3.3 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'especificuessin 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 provessin 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.

3.4 Redisseny de la pàgina web

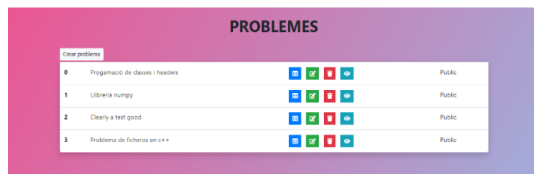
A l'anterior replanificació del projecte es va decidir afegir una tasca de redisseny de la pàgina web que es faria una vegada es tinguessin totes les tasques d'afegir noves funcionalitats (principalment les que requerien afegir o modificar vistes existents). Fent això es podria avançar ràpidament amb 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.

3.5 Tema fosc i tema clar

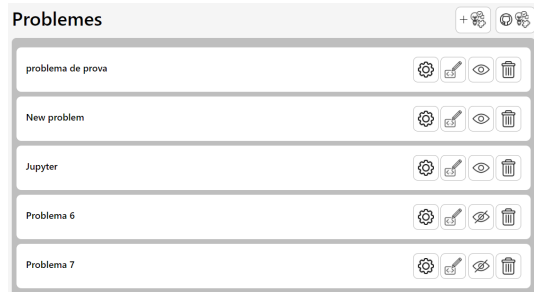
A causa de la cancel·lació d'una de les tasques (fet que es comentarà a la secció de planificació) es va decidir afegir la funcionalitat de canviar entre tema fosc i tema clar.

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) Abans del redisseny.



(b) Després del redisseny.

Figura 2: Comparació d'abans i després del redisseny.

A la imatge (b) de la figura 2 podem veure una llista redissenyada que fa servir el tema clar i a la imatge (b) de la figura 3 podem veure un formulari utilitzant el tema fosc.

3.5.1 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 2).

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).

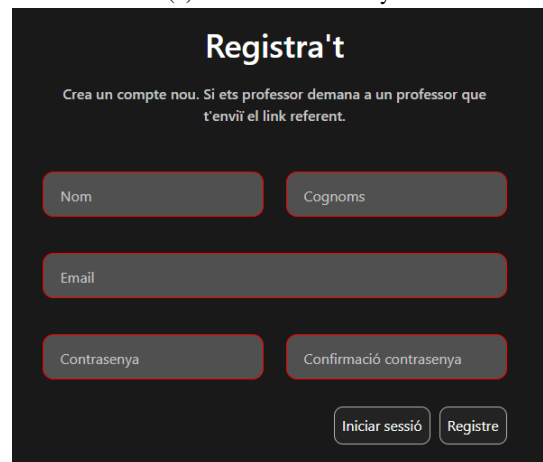
Opcionalment, es podia afegir un camp més al mapa hash de la pàgina per crear finestres emergents indicant només el títol i el contingut d'aquest (que podia ser un conjunt de camps de formulari o un text).

3.5.2 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) Abans del redisseny.



(b) Després del redisseny.

Figura 3: Comparació d'abans i després del redisseny.

A la figura 3 podem veure una comparació entre el 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.

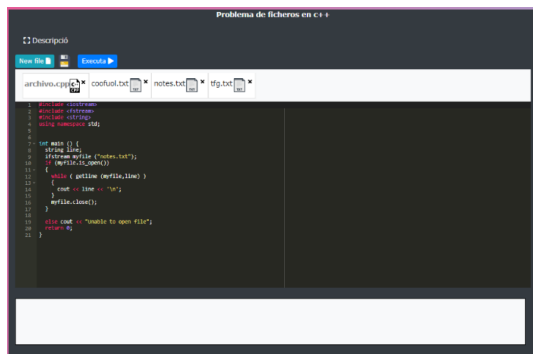
3.5.3 Editor

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

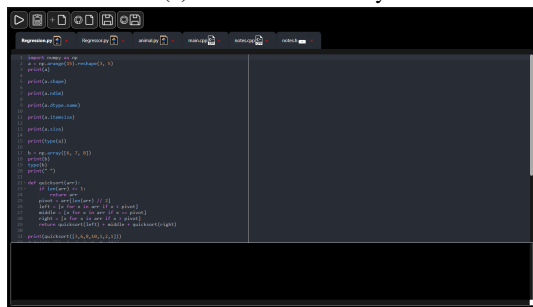
Amb el redisseny de la pàgina es va fer l'editor més gran, ja que era massa petit sent el component central d'aquesta vista. Es van modificar també els botons que apareixien a sobre de l'editor fent que aquests seguessin l'estil de les altres vistes. Com s'havia incorporat la funcionalitat de canviar entre tema fosc i tema clar es va haver de fer que l'editor es modificués com la resta de la pàgina. A la figura 4 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.

3.6 Altres tasques realitzades

Es va començar amb el testeig global de la pàgina web per comprovar que les funcionalitats noves i les antigues encara



(a) Abans del redisseny.



(b) Després del redisseny.

Figura 4: Comparació d'abans i després del redisseny.

funcionen com s'espera. Tots els resultats s'estan guardant per tal de poder incloure'ls al dossier final.

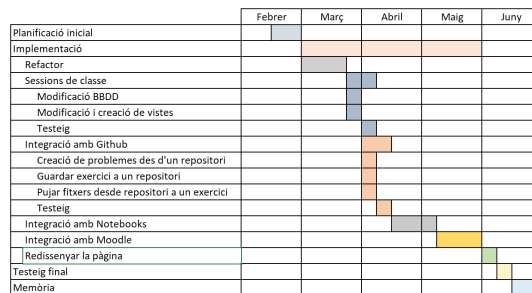
Per altra banda, s'ha parlat amb un tècnic del centre perquè es comenci a configurar una màquina on pujar l'aplicació i fer proves en un entorn real.

4 Compliment de la planificació

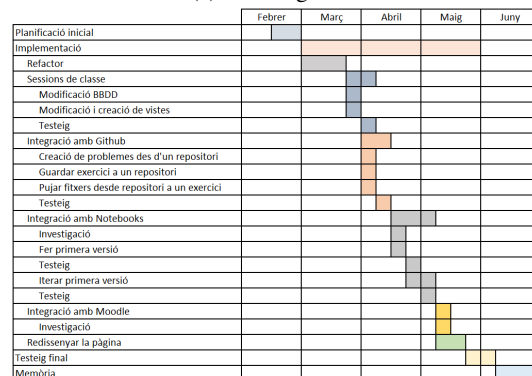
Segons la planificació que es va mostrar a l'informe anterior (imatge (a) de la figura 5), podem veure que a mitjan maig hauríem d'estar acabant la integració amb Moodle, però com aquesta tasca es va descartar perquè no era possible fer-la, la planificació es va modificar, quedant com veiem a la imatge (b) de la figura 5. Podem veure que les tasques noves estan desglossades, que el temps requerit per la tasca de la integració amb Moodle s'ha reduït dues setmanes fent que les següents tasques comencin abans del que s'havia planificat inicialment, això també ens ha permès dedicar més temps a redissenyar la pàgina i permetrà tenir més temps per fer la memòria final.

Els canvis que hi han hagut respecte a l'última actualització són els següents:

- Les noves tasques (les que s'han desenvolupat aquestes setmanes) estan desglossades.
- La duració de la tasca de la integració amb Caronte s'ha reduït a només una setmana (encara que la investigació va durar tres dies).
- Les tasques posteriors a l'última integració s'han avançat.



(a) Planning antic.



(b) Planning nou.

Figura 5: Comparació amb el planning de l'últim informe

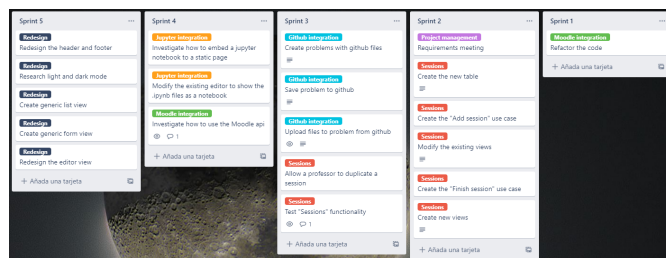


Figura 6: Sprints acabats representats a Trello.

- La cancel·lació d'una de les tasques ha permès que es pugui dedicar més temps al redisseny de la pàgina i al testeig general de la pàgina. Gràcies a això s'han afegit eines per facilitar la creació i edició de les vistes ja existents, fent que el codi sigui més fàcil de llegir i mantenir.

5 Metodologia utilitzada

La metodologia inicial que es va proposar i la que s'utilitzarà durant el desenvolupament del projecte és Agile [3].

Per tal de mantenir un ritme constant de treball s'ha utilitzat Trello [4], tal com es va dir inicialment, per planificar i tenir present quines tasques s'havien fet i quines no. Podem veure a la figura 6 els sprints que s'han completat. Cal dir que aquests sprints no sempre han durat el mateix perquè s'ha intentat que la càrrega de treball de cada sprint fos més o menys el mateix.

A més, cada setmana s'ha tingut una reunió amb el professor perquè tingués una idea general del que es va fent i perquè donés la seva opinió sobre el que s'estava fent.

6 Conclusions provisionals

Podem veure que el projecte ha anat seguint la planificació i faltant encara un mes per acabar s'han completat totes les tasques d'implementació que s'havien planificat. Veien aquestes dades podem concloure que s'ha treballat amb continuïtat i s'han aportat els resultats que s'esperaven.

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.

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.

Per altra banda, com s'ha comentat a la secció 3.6, s'ha començat a crear la infraestructura necessària per pujar el projecte a una màquina del centre per poder fer una prova amb usuaris reals principalment per detectar errors que tingui l'aplicació i per descobrir quines funcionalitats serien necessàries de cara a un futur.

Referències

- [1] Y. A. Asbahi, "Aula de programació interactiva." <https://shorturl.at/jkruG>, 2022.
- [2] "Binder." <https://jupyter.org/binder>.
- [3] "What is agile?." <https://www.atlassian.com/agile>.
- [4] "Trello." <https://trello.com/en>.