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

Stewart Barberà, Ian; Sánchez Pujadas, Francisco Javier, dir. Aplicació per alumnes del servidor de correcció automàtica. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290084>

under the terms of the  license

Aplicació per alumnes del servidor de correcció automàtica

Ian Stewart Barberà

Resum— L'aplicació per alumnes del servidor de correcció automàtica del meu tutor consisteix en crear una aplicació amb interfície gràfica perquè els alumnes puguin executar correccions i lliuraments de les seves assignatures d'una manera còmoda i intuïtiva. L'aplicació inicial que utilitzaven els alumnes era simplement executar comandes a una consola per corregir i entregar treballs i saber quines notes havien obtingut. S'ha fet ús del framework WinForms en C++ per desenvolupar l'aplicació amb interfície gràfica de manera senzilla gràcies a .NET. Aquest framework ofereix unes possibilitats de desenvolupament d'aplicacions molt gratificants ja que permet un desenvolupament ràpid i eficient mitjançant eines com dissenyadors que t'ensenyen els controls que es poden mostrar a l'aplicació, com ara botons, taules o llistes d'objectes. Aquesta aplicació apart de crear el frontend pels alumnes també es capaç de vincular-se amb el backend del meu tutor per tal de realitzar les peticions necessàries al servidor, d'aquesta manera podrà realitzar les funcionalitats que ofereixen les comandes per consola.

Paraules clau— Interfície gràfica, WinForms C++, .NET, dissenyador gràfic, frontend, backend, correccions, lliuraments, notes, client, servidor, peticions.

Abstract— The application for students of my tutor's automatic correction server consists of creating an application with a graphical interface so that students can execute corrections and deliveries of their subjects in a comfortable and intuitive way. The initial application students used was simply to run commands on a console to correct and deliver exercises and know what marks they had obtained. The WinForms framework in C++ has been used to develop the application with graphical interface in a simple way thanks to .NET. This framework offers very rewarding application development possibilities as it allows rapid and efficient development through tools such as designers who teach you the controls that can be displayed in the application, such as buttons, tables or object lists. This application, apart from creating the frontend for students, is also able to link with the backend of my tutor in order to make the necessary requests to the server, in this way it can perform the functionalities offered by console orders.

Index Terms— Graphical interface, WinForms C++, .NET, graphical designer, frontend, backend, corrections, deliveries, marks, client, server, requests.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

La carrera d'enginyeria informàtica requereix provar i testear el que es programa per tal d'aconseguir el resultat esperat o, fins i tot, optimitzar el producte funcional final per obtenir millors de rendiment i perfeccionar-lo.

Aquest procés el pot dur a terme diàriament cada alumne en diverses assignatures alhora, dificultant als professors realitzar una correcció correcta i ràpida dels exercicis i pràctiques que proposen. Això provoca que els alumnes triguin a saber si el que han programat és correcte o no perjudicant el seu aprenentatge fent-lo lent i ineficient.

El meu tutor va crear un servidor de correcció automàtica d'exercicis i pràctiques de les assignatures de la menció de computació Anàlisi i Disseny d'Algorismes, Tècniques de Disseny d'Algorismes i Compiladors. D'aquesta manera

els alumnes poden optimitzar el seu rendiment corregint de forma automàtica sense l'espera d'un professor mitjançant una consola i unes comandes concretes, així podran fer les correccions que necessitin.

Els alumnes poden portar a cap correccions concretes amb comandes per consola com `CorrectionClient KRUSKAL`, `CorrectionClient PRIM`. Pels lliuraments fan servir comandes com `CorrectionClient DELIVERY SPANNING_TREE_KRUSKAL` o `CorrectionClient DELIVERY SPANNING_TREE_PRIM`. També poden veure les notes dels lliuraments amb la comanda `CorrectionClient DELIVERIES`.

Aquest projecte busca implementar una interfície gràfica en forma d'aplicació que faciliti als alumnes de computació realitzar les funcionalitats mencionades anteriorment. En comptes d'utilitzar una consola i aprendre quines comandes calen per fer les correccions i lliuraments els alumnes podran visualitzar de manera intuïtiva les diferents possibilitats que ofereix l'aplicació, a més de navegar-hi de forma còmoda i rebent respostes visualment agradables i entenedores.

-
- E-mail de contacte: ian.stewart@autonoma.cat
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Javier Sanchez Pujadas (Ciències de la Computació)
 - Curs 2023/24

2 ESTAT DE L'ART

Com s'ha explicat a la introducció l'aplicació per consola del corrector la va crear el meu tutor i és el que s'ha fet servir com a base del projecte. Existeixen altres correctors de problemes de programació similars que han servit d'inspiració a l'hora de crear-lo i són força interessants, com per exemple el campus virtual de la UAB, proporcionant exercicis tipus test en línia als alumnes que un cop acabats els diu la nota que han tret, on han fallat i el perquè.

Un de molt important és el Judge.org[1], un corrector educatiu en línia de la Universitat Politècnica de Catalunya on els alumnes d'assignatures com Anàlisi i Disseny d'Algorismes i Estructures de Dades poden resoldre més de 800 problemes utilitzant més de 22 llenguatges de programació diferents. Està destinat tant a alumnes com a professors, ja que els professors poden tenir un seguiment dels problemes que realitzen els alumnes de les seves assignatures mitjançant el corrector, inclús afegir problemes nous, crear reptes, exàmens, lliuraments i penjar documents. El corrector consta de 6 servidors on només un actua de frontend proporcionant una interfície agradable pels seus usuaris, mentre que la resta de servidors estan destinats al backend, és a dir, a la gestió de les dades i a donar resposta a les diferents peticions que s'enviïn. Podeu trobar més informació al següent enllaç[2].

També està el corrector UVA Online Judge[3], un corrector en línia de la Universitat de Valladolid que es fa servir per practicar i participar en concursos de programació. La dificultat mitjana dels problemes és considerablement més elevada que els del Judge.org, ja que estan orientats principalment als concursos. Existeix la web UVa Hunting[4] on pots introduir el teu nom d'usuari i t'ensenya els problemes que encara no has resolt en ordre del nombre de persones que ho han aconseguit, indicant-te una idea de la dificultat dels problemes que et queden. Els veredictes donen poca informació, però hi ha posts on la gent discuteix sobre els resultats.

En definitiva estan sorgint cada cop més correctors en línia d'aquest estil que ajuden a millorar l'experiència educativa dels alumnes de carreres on s'ensenyi programació o anàlisi d'algorismes. Aquestes eines ajuden tant a alumnes fent-los aprendre de manera més divertida i eficient com als professors ajudant-los a portar un seguiment de cada alumne més personalitzat i dinàmic, en comptes d'haver de corregir els problemes dels alumnes un per un poden dedicar el temps a crear nous problemes desafiants mentre els alumnes són avaluats amb la mateixa eficiència o més que si fossin avaluats pels professors directament.

3 OBJECTIUS

L'objectiu principal és realitzar una aplicació que implementi la interfície gràfica del corrector principal del tutor. L'aplicació es desenvoluparà amb el framework WinForms C++[5] que forma part del framework .NET[6], ja que

permet crear interfícies gràfiques a partir de codi en C++, és a dir, es podrà implementar aprofitant el codi en C++ del backend del tutor.

Per aconseguir aquest objectiu, es van determinar uns subobjectius durant la planificació del projecte. Primer es van determinar els subobjectius crítics que representen els objectius necessaris i crítics per implementar les funcionalitats ja existents a l'aplicació per consola, oferint una interfície agradable. Després es van determinar uns altres subobjectius per mostrar més informació rellevant dels correctors i lliuraments de cada assignatura als alumnes, però deixant clar que només es durien a terme si existia suficient temps com per implementar-los correctament.

La llista dels objectius és la següent:

-Objectiu 1 [O1] [Crític]: Permetre a l'alumne la capacitat d'iniciar sessió amb una contrasenya nova enviada a través del correu electrònic.

-Objectiu 2 [O2] [Crític]: Permetre a l'alumne que pugui fer totes les correccions i lliuraments que té disponibles.

-Objectiu 3 [O3] [Crític]: Mostrar el llistat dels lliuraments i correccions que ha fet l'alumne com a historial amb informació com les notes que ha tret.

-Objectiu 4 [O4]: Poder descarregar-se les últimes correccions realitzades per l'alumne.

-Objectiu 5 [O5]: Mostrar estadístiques personals de l'alumne, com per exemple mostrar el nombre de vegades que ha executat un corrector.

-Objectiu 6 [O6]: Mostrar totes les correccions de tots els alumnes per comparar resultats, sigui en forma de gràfica amb un diagrama de barres amb totes les notes de tots els alumnes indicant on es troba l'alumne concret o de manera numèrica.

-Objectiu 7 [O7]: Oferir a l'usuari una interfície còmoda d'utilitzar i intuïtiva de manera que faciliti l'ús de la nova aplicació i representi una millora respecte l'aplicació per consola inicial.

4 PLANIFICACIÓ

La planificació del projecte mostra les tasques realitzades durant el desenvolupament del projecte de manera que satisfacin els objectius plantejats anteriorment de manera que els resultats obtinguts siguin de bona qualitat, a més de la seva durada.

Cal destacar que alguns objectius que s'havien de dur a terme més endavant s'han realitzat abans que objectius que a priori s'havien de dur a terme primer. Per exemple, el cinquè objectiu es va implementar abans que el quart ja que s'havia d'implementar conjuntament amb el tercer

objectiu i en aquell moment encara no sabia com enfocar el quart objectiu. Objectius com la interfície d'usuari s'ha dut a terme durant tot el desenvolupament de l'aplicació en termes de durada.

Per més informació, es pot consultar la planificació del projecte a l'apèndix A1. Planificació del projecte.

5 METODOLOGIA

A continuació s'explicarà quina metodologia s'ha seguit pel correcte desenvolupament del projecte.

5.1 Familiarització amb el projecte inicial

Durant la fase de familiarització del projecte inicial es va estudiar el codi proporcionat pel professor per tal d'entendre el seu funcionament. D'aquesta manera es va saber quins arxius eren els necessaris per a implementar l'aplicació gràfica.

El tutor va proporcionar una solució de Visual Studio 2022[7] anomenada correctors on hi havia un projecte anomenat CorrectorLib que contenia els fitxers amb les funcions necessàries per aplicar la lògica a la nova aplicació. Amb aquesta informació vaig haver d'aprendre a vincular dues solucions per aconseguir que a través de la interfície gràfica es poguessin llençar peticions al servidor amb el que es connectava el projecte CorrectorLib.

També hi havia un projecte anomenat CorrectionClient on es veia el que podien fer els alumnes amb l'aplicació inicial per consola des de la banda del client, és a dir, quines comandes podien realitzar i què executaven exactament. Amb aquest projecte em podia fer una idea de les funcionalitats que necessitaria la meua aplicació i quins resultats proporcionaven un cop executades les comandes.

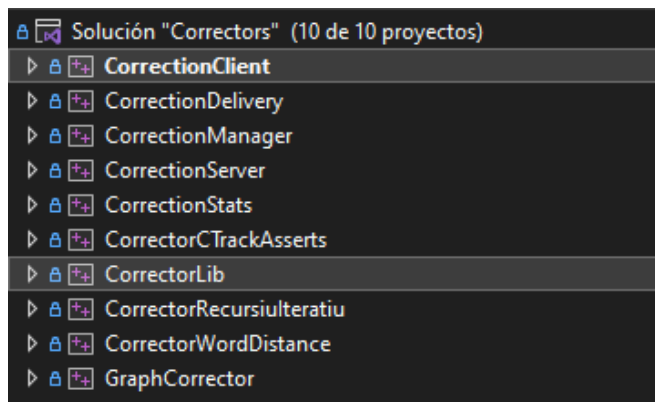


Fig. 1 Projectes proporcionats pel tutor.

5.2 Escollir tecnologies a utilitzar

Durant aquesta fase es van escollir les tecnologies a utilitzar contrastant informació a internet per a obtenir la millor resposta per a aquest projecte en concret.

L'IDE escollit pel desenvolupament de l'aplicació va ser Visual Studio 2022 perquè es pot utilitzar de manera gratuïta i es va poder utilitzar el projecte inicial del professor, també realitzat en Visual Studio 2022, i vincular-lo a l'aplicació gràfica.

El primer IDE escollit per desenvolupar l'aplicació va ser QtCreator[8]. Es va fer un canvi de framework a WinForms C++ ja que l'IDE del llenguatge QT, és a dir, QtCreator, era un IDE que havia d'aprendre a utilitzar des de zero a més de ser bastant complex a l'hora de compilar i estructurar el codi. Aquest IDE també deixava de respondre quan intentava executar el codi i era incapaç de llegir fitxers que tinguessin un espai en el nom, dificultant l'estructuració de l'aplicació gràfica junt amb els fitxers existents del projecte inicial del professor.

Per tal de tenir un control de versions sobre el projecte s'ha utilitzat un repositori de GitHub[9] on s'anava pujant el codi que s'anava desenvolupant a més de les diferents versions dels informes. Es van crear branques en funció de les funcionalitats que s'havien d'implementar i un cop la funcionalitat estava testejada es fusionava amb la branca main indicant que la funcionalitat estava acabada.

Finalment, per tenir un control sobre la planificació del projecte s'ha utilitzat la plataforma JIRA[10] dividint la feina en sprints. Es va seguir el model de planificació de projectes SCRUM ja que em permetia dividir la feina en diferents sprints bisetmanals i comprovar el meu progrés en funció del temps per saber com encaminar el projecte cada vegada que el desenvolupament es desviava de la planificació inicial. La duració dels sprints va variar en funció dels problemes que m'anaven sorgint però no va ser molt diferent que la duració que es va pensar des d'un principi a la planificació. És una bona eina per implementar una metodologia àgil al projecte i veure si la planificació és correcta o és necessari implementar canvis.

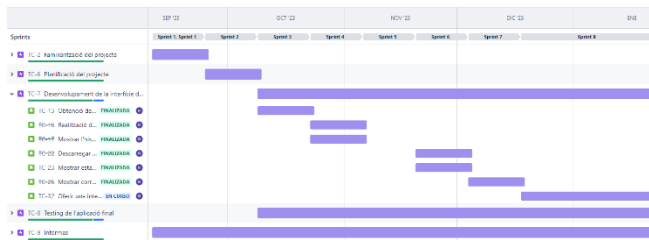


Fig. 2 Planificació del projecte a l'eina JIRA.

6 Desenvolupament

En aquest apartat s'explicarà detalladament el desenvolupament del projecte per fases i com s'han acabat assolint o ometent els objectius.

L'aplicació mostra un TabControl que permet crear una sistema de pestanyes dins l'aplicació. S'ha dividit la interfície en tres pestanyes, una per l'execució de correctors,

una altra per l'execució de lliuraments i una última per mostrar informació específica de l'alumne respecte els correctors i lliuraments que té disponibles.

Les correccions que realitzen els alumnes les executen moltes vegades al dia ja que cada cop que fan una petita modificació al codi del treball volen saber si ha canviat el resultat. La pestanya dels correctors ofereix l'avantatge de poder executar els correctors amb un sol clic les vegades que necessiti l'alumne, incrementant la comoditat a l'hora de corregir treballs en comparació a haver d'introduir una comanda concreta a una consola cada cop que es vol comprovar un petit canvi en el codi.

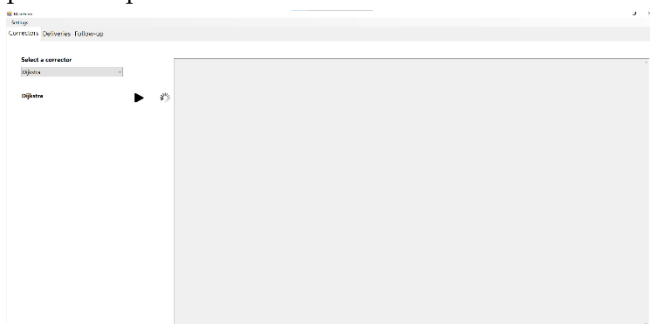


Fig. 3 Imatge de la pestanya de correctors.

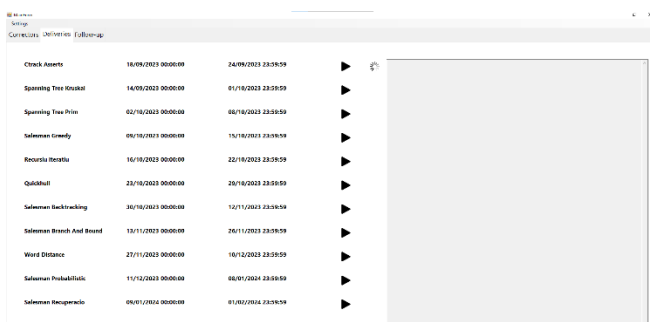


Fig. 4 Imatge de la pestanya de lliuraments.

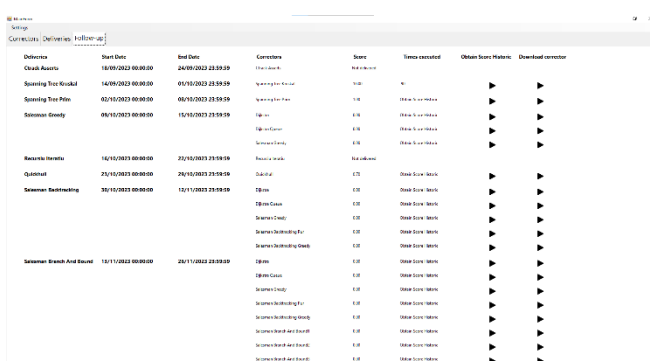


Fig. 5 Imatge de la pestanya de seguiment.

També consta d'un menú on pot triar diferents opcions per modificar el NIU de l'usuari i la seva contrasenya, a més d'afegir més alumnes a l'aplicació per si han d'entregar un treball en grup, actualitzant les notes dels treballs per a cada alumne existent a l'aplicació.

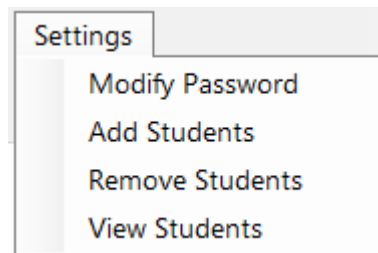


Fig. 6 Imatge del menú dels estudiants.

6.1 Obtenció de contrasenya per l'alumne

Aquesta fase tracta de realitzar el primer objectiu crític de l'aplicació gràfica, permetre a l'usuari obtenir una contrasenya nova per tal de poder realitzar correccions i lliuraments. Per fer-ho l'aplicació envia una petició al servidor mitjançant una funció del projecte del tutor, ConnectLogin, que proporciona a l'alumne una nova contrasenya al correu electrònic. Per més informació, es pot consultar el contingut de la funció ConnectLogin i la seva crida a l'apèndix A2. Funció ConnectLogin per obtenir contrasenya.

El fitxer CorrectionClient.ini és un fitxer de configuració que tenen tots els alumnes i conté la informació de l'alumne que fa servir l'aplicació, a més dels correctors disponibles, i es comunica amb el servidor per identificar l'alumne de manera correcta.

Dins el menú mencionat anteriorment, una de les opcions permet a l'alumne realitzar aquesta funcionalitat. Si clica l'opció s'obrirà un formulari que li demanarà triar un alumne gràcies a un control anomenat ComboBox. Aquest control desplega una llista d'opcions amb els NIUs del fitxer de configuració, permetent a l'alumne escollir de quin alumne vol obtenir una nova contrasenya. Un cop l'alumne esculli un NIU apareixerà un botó per executar la funció ConnectLogin i enviar la nova contrasenya al correu del NIU seleccionat. Per últim apareixerà un TextBox on es podrà escriure la nova contrasenya i un altre botó que aplicarà els canvis a l'arxiu de configuració i així poder seguir executant correctors i lliuraments.

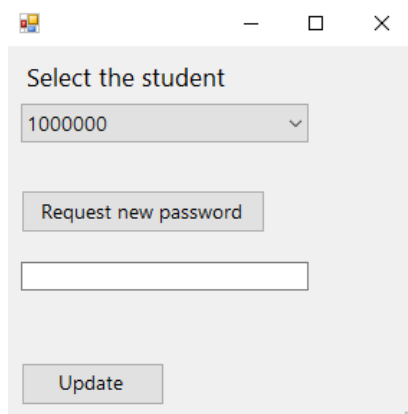


Fig. 7 Formulari per modificar la contrasenya.

6.2 Execució de correctors i lliuraments

En aquesta fase s'ha implementat el segon objectiu crític que consisteix en permetre als alumnes executar els correctors i lliuraments que tenen disponibles a les seves assignatures.

6.2.1 Execució de correctors

Per tal d'obtenir els correctors disponibles de cada alumne, dins el fitxer de configuració es poden trobar tots amb la informació de quins fitxers se li enviaren al servidor per tal de poder realitzar la correcció correctament, a més dels fitxers de sortida que genera el servidor un cop s'ha realitzat la correcció.

A l'aplicació s'ha llegit el fitxer per obtenir la llista dels correctors disponibles i crear un ComboBox, igual que el ComboBox per obtenir contrasenya, perquè l'usuari pugui seleccionar quin corrector vol executar en aquell moment. D'aquesta manera, per molt que l'alumne tingui molts correctors disponibles sempre els podrà seleccionar des d'una llista ordenada i sense sobrecarregar d'informació l'aplicació.

Select a corrector

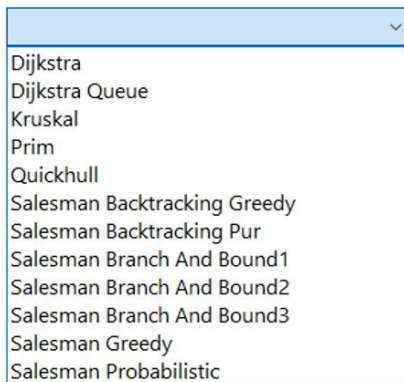


Fig. 8 ComboBox mostrant la llista de correctors disponibles de l'alumne.

Un cop l'alumne selecciona un corrector li apareixerà una taula amb el nom del corrector seleccionat i un botó amb una imatge de botó d'execució per fer-li entendre que amb aquest botó podrà executar aquest corrector en concret.

Quan l'alumne executa un corrector l'aplicació gràfica executa una funció del projecte del tutor anomenada Run que crida altre cop a la funció ConnectLogin amb uns paràmetres diferents, indicant aquest cop que necessita executar un corrector concret i no obtenir una contrasenya nova. Per més informació, es pot consultar la funció Run a l'apèndix A3. Funció Run.

Mentre s'executa un corrector el servidor va generant una sèrie de couts que van donant informació a l'alumne

de l'estat de la correcció. Aquests missatges utilitzaven un buffer per mostrar-se en una consola, per tant, es perdien. A la meua aplicació, però, s'ha aconseguit redirigir el buffer de sortida dels missatges a un TextBox per tal de mostrar en temps real l'estat de la correcció a l'alumne, d'aquesta manera estarà més atent a l'aplicació en moments on només ha d'esperar que acabi una execució.

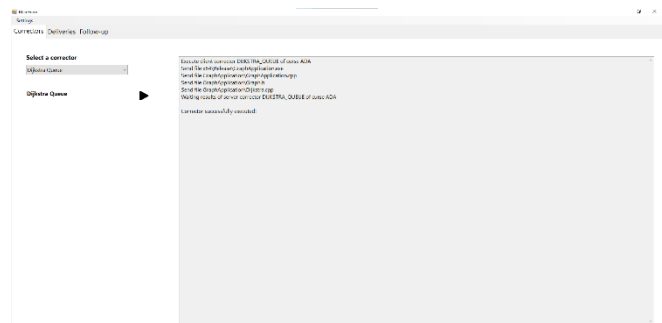


Fig. 9 TextBox amb els missatges en temps real de l'execució d'un corrector concret.

Un cop l'execució ha estat exitosa l'aplicació mostra un informe de correcció generat pel servidor dins la pròpia interfície. L'alumne pot navegar per aquest informe ja que és un fitxer html que mostra la nota que ha tret l'alumne en aquest intent junt amb les proves que ha fet el servidor per realitzar la correcció.

S'ha considerat mostrar només la informació del corrector actual que vol corregir l'alumne i no mostrar informació de cap altre corrector ja que els alumnes executaran un mateix corrector moltes vegades abans que treguin bona nota, per tant, és una bona manera de concentrar l'atenció de l'alumne només en el corrector actual fins que l'aprovi o tregui la màxima nota possible.

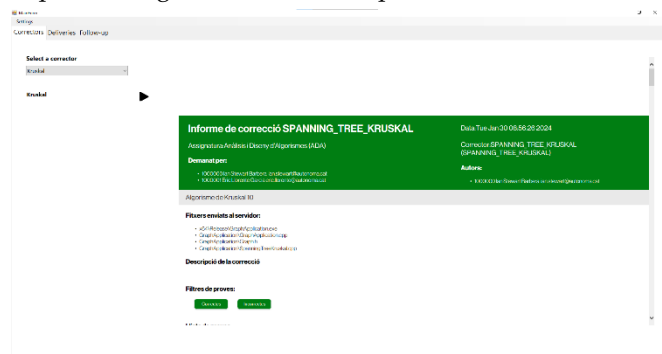


Fig. 10 Informe de correcció un cop a finalitzat l'execució d'un corrector.

6.2.2 Execució de lliuraments

Per obtenir la informació dels lliuraments disponibles es fa una crida a la funció ConnectLogin amb els paràmetres adequats. Per més informació sobre quins paràmetres es fan servir, podeu consultar l'apèndix A4. Funció ConnectLogin per obtenir lliuraments.

Una vegada s’ha obtingut la informació dels lliuraments disponibles per l’alumne s’ha creat una taula amb el nom de cada lliurament, la seva data d’inici i final d’entrega i un botó d’execució igual que el que s’ha utilitzat per executar els correctors.

Els lliuraments executen correctors, per tant, en poden executar més d’un i tenir un temps d’execució més elevat. Degut a això l’alumne disposarà de menys lliuraments que correctors, per tant, s’ha mostrat tots els lliuraments disponibles a la taula perquè l’alumne pugui veure en tot moment quina és la data límit d’entrega de cadascun d’ells.

Si es passa la data d’entrega final d’un dels lliuraments l’alumne encara podrà entregar-lo tot i que el professor serà conscient que el lliurament ha estat entregat fora de termini. En canvi, els lliuraments que tenen una data d’inici que encara no ha arribat no es mostraran ja que l’alumne no ha de poder entregar un lliurament que encara no hauria d’estar disponible.

Per tal d’executar els lliuraments s’ha utilitzat la funció `DeliveryClientRun` que també executa la funció `ConnectLogin` amb uns altres paràmetres. Per més informació consulta l’apèndix A5. Funció `ConnectLogin` de la funció `DeliveryClientRun`.

En aquest cas també s’ha mostrat en un `TextBox` els missatges que genera el servidor mentre s’executa un lliurament, fent-li saber l’estat de l’execució a l’alumne en temps real. Com que els lliuraments tenen una duració més elevada que els correctors, es mostren una quantitat de missatges més gran i durant uns intervals de temps relativament curts, ajudant a l’alumne a mantenir-se atent a l’aplicació durant l’execució d’un lliurament.

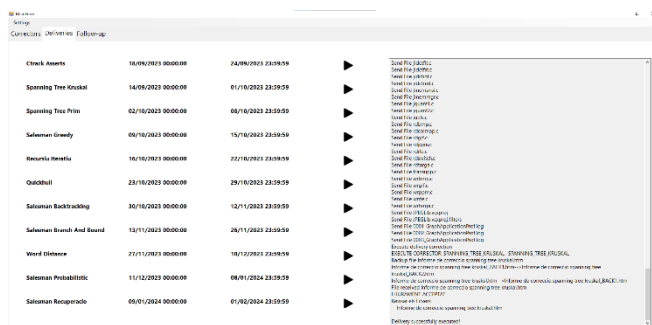


Fig. 11 `TextBox` amb els missatges generats pel servidor durant l’execució d’un lliurament.

En el cas que el nombre de lliuraments disponibles no càpiguen en pantalla apareixerà una scrollbar per permetre a l’alumne pujar i baixar a través de la taula i poder veure els lliuraments que ell desitja.

6.3 Mostrar informació de l’alumne

Aquesta fase integra el tercer objectiu crític, visualitzar la informació de l’alumne, com ara les notes dels correctors executats per l’alumne ja que les notes dels lliuraments les posa el professor en base a les notes dels correctors i l’estat dels fitxers de codi entregats al servidor durant l’execució del lliurament.

S’ha creat una taula amb tots els lliuraments disponibles per l’alumne en aquell moment en funció del servidor. Per cada lliurament hi ha una “subtaula” que mostra la informació de cada corrector que executa un lliurament en concret. D’aquesta manera s’ha aconseguit mostrar el nom de cada lliurament, les dates d’inici i finalització de les entregues de cada lliurament, el nom de cada corrector que executa cada lliurament i les notes de cada corrector.

Per poder obtenir la informació de les notes de cada corrector executat per cada alumne es crida a la funció `ConnectLogin` amb els mateixos paràmetres que s’han utilitzat per obtenir la informació dels lliuraments. Gràcies a aquesta funció s’obté tota la informació respecte els lliuraments de cada alumne i, després, a la pròpia aplicació gràfica es pot escollir quina informació mostrar.

En el cas que no s’hagi entregat algun dels lliuraments s’indicarà a l’apartat de notes dels correctors que no ha estat lliurat. Cal destacar que la nota que apareix a la taula de seguiment serà sempre la última nota obtinguda per l’última execució d’aquell corrector, és a dir, l’alumne ha de ser conscient en quin moment ha tret la seva millor nota ja que no apareixerà, això seria un objectiu extraordinari a tenir en compte si es vol ampliar el projecte.

Aquesta taula li pot servir a l’alumne per poder visualitzar el seu progrés a l’assignatura, tenint un seguiment sobre les notes que va traient al llarg del curs i saber a quins lliuraments li hauria de dedicar més hores per tal d’obtenir millors notes o simplement aprovar-los.

Settings				
Correctors Deliveries Follow-up				
Deliveries	Start Date	End Date	Correctors	Score
Crack Assets	18/09/2023 00:00:00	24/09/2023 23:59:59	Crack Assets	Not delivered
Spanning Tree Kruskal	14/09/2023 00:00:00	01/10/2023 23:59:59	Spanning Tree Kruskal	10.00
Spanning Tree Prim	02/10/2023 00:00:00	08/10/2023 23:59:59	Spanning Tree Prim	1.00
Salesman Greedy	09/10/2023 00:00:00	15/10/2023 23:59:59	Dijkstra	0.00
			Dijkstra Queue	0.00
			Salesman Greedy	0.00
Rekursiu Iteratiu	16/10/2023 00:00:00	22/10/2023 23:59:59	Rekursiu Iteratiu	Not delivered
Quickhull	23/10/2023 00:00:00	29/10/2023 23:59:59	Quickhull	0.70
Salesman Backtracking	30/10/2023 00:00:00	12/11/2023 23:59:59	Dijkstra	0.00
			Dijkstra Queue	0.00
			Salesman Greedy	0.00
			Salesman Backtracking Fur	0.00
			Salesman Backtracking Greedy	0.00

Fig. 12 Taula de seguiment amb els lliuraments i les notes de cada corrector.

6.4 Descarregar correccions

En aquesta fase s'ha implementat el primer objectiu no crític, és a dir, el quart objectiu de la llista d'objectius plantejats inicialment. Aquest objectiu consisteix en poder donar-li la opció a l'alumne de descarregar-se els fitxers utilitzats en l'última correcció realitzada per a cada corrector existent.

Per aconseguir-ho s'ha utilitzat una funció d'un dels projectes del tutor anomenat `CorrectionStats`. En aquest projecte hi havia creada la lògica per poder obtenir els fitxers enviats al servidor de l'última correcció realitzada. Aquesta funció també crida a la funció `ConnectLogin` amb uns paràmetres diferents per poder realitzar aquesta funcionalitat en concret.

El meu projecte només està vinculat amb el projecte `CorrectorLib` ja que s'utilitzen moltes de les seves funcions, en canvi, no ha estat necessari vincular el meu projecte amb el projecte `CorrectionStats` ja que només necessitava aquesta funció per complir aquest objectiu. Es podria tenir en compte en un futur per si es vol implementar alguna de les funcionalitats que ofereix aquest projecte ja que són força interessants. De fet, he hagut d'utilitzar una d'aquestes funcionalitats per realitzar l'objectiu 5, per tant, només m'han calgut dues funcions d'aquest projecte.

Com que no era necessari vincular el meu projecte amb el projecte `CorrectionStats` he implementat les dues funcions que necessitava al codi del meu projecte directament. La funció per obtenir els fitxers utilitzats en l'última correcció realitzada s'anomena `DownloadCorrectorFiles`. Per més informació sobre aquesta funció, es pot consultar l'apèndix A6. Funció `DownloadCorrectorFiles`.

Un cop s'ha obtingut la manera descarregar els fitxers corresponents s'ha aprofitat la taula de seguiment de l'anterior objectiu per afegir una columna addicional a cada corrector existent. La columna conté un botó d'execució tal com els botons d'execució dels correctors i lliuraments.

6.5 Afegir informació addicional de l'alumne

Aquesta fase assoleix el cinquè objectiu, mostrar informació d'interès a l'alumne apart de les notes dels correctors.

En aquest objectiu també s'ha fet ús d'una de les funcions del projecte `CorrectionStats` com s'ha mencionat anteriorment. Gràcies a aquesta funció es genera un arxiu html al directori de treball de l'alumne que mostra informació interessant sobre els correctors, com ara el nombre de vegades que l'alumne ha executat un corrector concret o les proves que realitza el servidor per fer una correcció. Per aconseguir això també crida la funció `ConnectLogin` amb els paràmetres correctes per a aquesta funcionalitat. Per més informació podeu consultar l'apèndix A7. Funció `GetLastCorrectorScore`.

Aprofitant la mateixa taula de seguiment del tercer objectiu s'han afegit dues columnes noves, una per obtenir aquest fitxer d'informació i l'altre per mostrar el nombre de vegades que l'alumne ha executat cada corrector. El problema de fer-ho d'aquesta manera és que l'aplicació depèn de l'existència d'aquest fitxer per poder mostrar la informació a la columna de correccions realitzades.

Es va intentar obtenir aquesta informació en forma de variables per no dependre d'un fitxer generat en funció de l'alumne i poder mostrar la informació sempre que es vulgui. Es va considerar un excés de feina ja que havia de rebre aquestes variables des del servidor i es volia mostrar altra informació com les proves que realitza el servidor quan corregeix un corrector. Degut a la falta de temps es va decidir mostrar el nombre de vegades que l'alumne va executar cada corrector i es va prioritzar millorar la interfície d'usuari.

A la columna de vegades corregides, si l'alumne no disposa del fitxer per saber el nombre de vegades apareixerà un label de text indicant-li que obtingui el fitxer mitjançant l'altra columna nova. La columna d'obtenció del fitxer mostra el mateix botó d'execució que amb els correctors, lliuraments i descàrrega de fitxers.

Correction	Start Date	End Date	Correctors	Score	Times executed	Obtain Score Matrix	Download corrector
Check Assort	16/09/2022 00:00:00	24/09/2022 23:59:59	Check Assort	0.00	0		
Spending Test Result	14/09/2022 00:00:00	01/10/2022 23:59:59	Spending Test Result	0.00	0		
Spending Test Plan	01/10/2022 00:00:00	06/10/2022 23:59:59	Spending Test Plan	0.00	0	Obtain Score Matrix	
Seamless Entry	06/10/2022 00:00:00	15/10/2022 23:59:59	Seamless Entry	0.00	0	Obtain Score Matrix	
Resource Transfer	16/10/2022 00:00:00	23/10/2022 23:59:59	Resource Transfer	0.00	0	Obtain Score Matrix	
Quarant	23/10/2022 00:00:00	29/10/2022 23:59:59	Quarant	0.00	0	Obtain Score Matrix	
Seamless Switching	30/10/2022 00:00:00	12/11/2022 23:59:59	Seamless Switching	0.00	0	Obtain Score Matrix	
Seamless Branch And Bound	13/11/2022 00:00:00	20/11/2022 23:59:59	Seamless Branch And Bound	0.00	0	Obtain Score Matrix	

Fig. 13 Columnes addicionals per mostrar el nombre de vegades que s'ha executat un corrector i descarregar l'última execució.

6.6 Últim objectiu i interfície d'usuari.

En aquesta fase es tenia previst realitzar el sisè objectiu que consistia en mostrar a l'usuari uns gràfics on es veiessin les notes de la classe de manera anònima, mostrant a l'alumne on es trobava respecte la resta dels seus companys.

Degut a la falta de temps i a la prioritització d'objectius més importants com la interfície d'usuari i el testing de l'aplicació sencera no s'ha pogut dur a terme. Tot i així, és un objectiu interessant a tenir en compte en el cas que es vulgui ampliar aquest projecte i continuar-lo en un altre treball de fi de grau.

Tot i així, s'ha pogut acabar implementant el menú d'opcions que permet a l'estudiant fer quatre coses diferents.

La primera opció permet a l'alumne modificar la seva contrasenya per poder actualitzar-la i així executar correctors, lliuraments i veure el seu seguiment, tal com s'ha explicat a l'apartat de desenvolupament del primer objectiu crític.

La segona opció permet a l'alumne afegir els estudiants que vulgui a l'aplicació, és a dir, ha d'introduir en un nou formulari el NIU i contrasenya d'un alumne nou en el cas que necessiti formar un grup de treball i que les notes dels correctors i lliuraments s'apliquin a cada alumne existent en el fitxer de configuració. Apareixerà un TextBox tant pel NIU com per la contrasenya i un botó per aplicar els canvis que modificarà el fitxer de configuració amb la nova informació.

Fig. 14: Formulari per afegir alumnes a l'aplicació i crear així grups de treball. El formulari té el títol "Add a new student". A sota hi ha dos camps de text: "NIU" i "Password". A la part inferior hi ha un botó "Add Student".

Fig. 14 Formulari per afegir alumnes a l'aplicació i crear així grups de treball.

La tercera opció consisteix en un altre formulari que permet a l'alumne seleccionar un dels alumnes que hi ha al fitxer de configuració, mitjançant una llista amb el control ComboBox. Un cop l'alumne selecciona un dels NIUs podrà clicar un botó que modificarà el fitxer de configuració eliminant l'alumne seleccionat. D'aquesta manera es pot canviar els integrants del grup de treball afegint o eliminant alumnes, inclús es poden eliminar tots els alumnes perquè només en quedi un en el cas que hi hagi correctors o lliuraments que s'hagin d'entregar de manera individual.

Fig. 15: Formulari per eliminar alumnes a l'aplicació i modificar els grups de treball. El formulari té el títol "Select the student". A sota hi ha un control ComboBox amb el valor "1000000" seleccionat. A la part inferior hi ha un botó "Remove".

Fig. 15 Formulari per eliminar alumnes a l'aplicació i modificar els grups de treball.

L'última opció tracta d'un altre formulari que fa aparèixer els alumnes actuals que hi ha a l'aplicació. Serveix perquè l'alumne que estigui utilitzant l'aplicació en aquell moment pugui recordar quins alumnes formen part del grup de pràctiques sense haver d'obrir el fitxer de configuració. D'aquesta manera sabrà si necessita modificar-lo o ja està bé com està.

Fig. 16: Formulari per visualitzar els alumnes actuals que hi ha a l'aplicació. El formulari té el títol "Students group". A sota hi ha una llista amb dos elements: "1000000" i "1000001".

Fig. 16 Formulari per visualitzar els alumnes actuals que hi ha a l'aplicació.

Les últimes setmanes es dedicaran a optimitzar la interfície d'usuari i al testing de l'aplicació. Es comprovarà que el comportament de l'aplicació sigui l'esperat sota diferents contexts, com ara que el servidor estigui apagat o la informació del fitxer CorrectionClient.ini sigui incorrecta, indicant a l'usuari quin és el motiu pel qual no pot realitzar certa acció en comptes que l'aplicació deixi de funcionar per complet per culpa d'excepcions no controlades.

7 TESTING DE L'APLICACIÓ

El testing de l'aplicació s'ha dut a terme al mateix temps que el desenvolupament, és a dir, cada cop que s'acabava una funcionalitat es provaven els diferents camins que l'alumne pot executar.

En totes les aplicacions existeix el *happy path* que consisteix en el camí desitjat per l'usuari respecte una funcionalitat, per exemple, el *happy path* de modificar la contrasenya de l'alumne seria que l'usuari obtingués la nova contrasenya al correu i pogués canviar-la mitjançant el nou formulari. Tot i així, és molt poc probable que l'alumne executi sempre el *happy path* de cada funcionalitat ja que existeixen molts altres camins que poden succeir i poden donar resultats inesperats a l'usuari o directament fer deixar de funcionar l'aplicació. El testing serveix per controlar totes aquestes possibilitats i corregir-les en forma de missatges per indicar a l'usuari quines accions no pot realitzar o per evitar que l'aplicació falli.

Amb el testing de l'objectiu de modificar contrasenya s'ha afegit uns MessageBox per indicar a l'usuari quin és el comportament de l'aplicació quan és necessari.

Per exemple, si l'usuari clica el botó d'obtenir una contrasenya apareixerà un MessageBox indicant-li que se li

ha enviat una nova contrasenya al seu correu electrònic. Un cop introdueixi una contrasenya nova i premi el botó de modificar contrasenya apareixerà un MessageBox dient-li que s'ha modificat correctament. Si hi ha algun problema amb la connexió amb el servidor o el fitxer de configuració `CorrectionClient.ini` està mal configurat l'alumne no podrà obtenir una nova contrasenya, per tant, l'aplicació li farà saber que existeix algun problema per realitzar la petició.

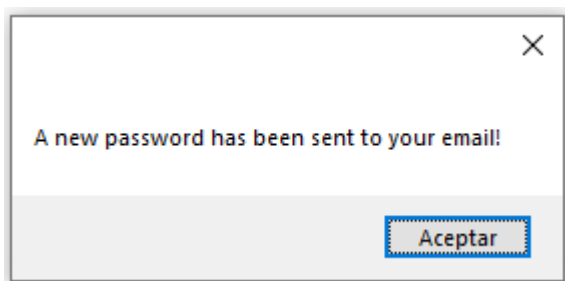


Fig. 17 MessageBox indicant que s'ha enviat una nova contrasenya al correu de l'alumne.

En el cas de l'execució de correctors, si la correcció no és exitosa se li mostrarà a l'alumne un MessageBox, igual que el mostrat anteriorment, explicant quin error ha estat el que ha provocat l'aturada de la correcció, d'aquesta manera l'alumne es podrà fer una idea de com solucionar aquest error per tornar a executar el mateix corrector de forma exitosa.

Durant l'execució dels lliuraments, si s'executa correctament també apareixerà un MessageBox indicant-li a l'alumne que l'execució ha estat exitosa. Si falla se li mostrarà un MessageBox explicant quin ha estat l'error que ha provocat la fallida. D'aquesta manera l'alumne podrà entendre quin ha estat el resultat de l'execució del lliurament que ha seleccionat.

Si el servidor no està disponible i no es pot obtenir la informació necessària per crear les taules dels lliuraments i de seguiment es mostrarà un label de text al mig de la pantalla, indicant a l'alumne que no es poden crear degut a la comunicació amb el servidor.

En el cas que es cliqui al botó de la descarrega de l'última correcció i sigui exitosa se li crearà un directori al directori de treball de l'alumne amb el nom del corrector seguit del seu NIU i la data de realització de la correcció. Així podrà guardar un historial de les correccions que més li interessin. En canvi, si falla se li mostrarà un MessageBox explicant-li perquè ha fallat.

Per últim, amb el botó per obtenir l'històric de notes s'ha testejat de la mateixa manera que el botó de descàrrega de l'última correcció realitzada, és a dir, si l'execució és exitosa es guardarà l'arxiu de l'històric al directori de treball i, en canvi, si falla mostrarà un MessageBox indicant l'error.

Gràcies a saber detalladament el que han de fer les funcionalitats s'ha pogut pensar les diferents possibilitats no desitjades que l'alumne pot arribar a trobar-se per tal de controlar-les i aconseguir que existeixi un feedback adequat en qualsevol situació de l'aplicació.

8 RESULTATS

El resultat de l'aplicació és satisfactori quant a les funcionalitats que ha de realitzar, doncs s'han assolit els objectius crítics i a més s'han pogut assolir dos objectius addicionals. És una interfície gràfica fàcil d'utilitzar i serveix com a frontend de l'aplicació del tutor i com a aplicació còmoda pels alumnes per poder cursar les assignatures d'una manera més agradable i sense haver d'introduir comandes a una consola.

Quant a interfície d'usuari s'ha escollit no sobrecarregar d'informació la pantalla per oferir una aplicació agradable als alumnes. Es podria millorar en termes de colors i d'afegir algun control més atrevit com animacions o gràfics tot i que s'ha preferit un estil minimalista.

En termes de testing es podria optimitzar l'aplicació perquè no faci aparèixer errors no tan greus però que l'alumne podria arribar a notar, com per exemple que l'última fila de la taula de seguiment aparegui una mica tallada i l'alumne hagi de baixar per la taula per visualitzar-la correctament, tot i que s'intentaran corregir de cara a la presentació final.

En definitiva és un projecte que es pot ampliar en molts sentits ja que hi ha una gran quantitat d'informació que es podria mostrar i no s'ha pogut degut a la falta de temps. Com ja he mencionat anteriorment, el sisè objectiu seria interessant per seguir amb aquest projecte ja que es fomentaria la competitivitat a les classes per veure qui aconsegueix millors resultats, provocant la obtenció de notes dels alumnes més elevades.

9 CONCLUSIÓ

El desenvolupament del projecte ha estat una mica més lent que a la planificació inicial, ja que les coses que han dificultat l'avanç diari del projecte han sigut la familiarització amb el projecte existent del professor, l'aprenentatge de la nova tecnologia de WinForms C++, la vinculació dels dos projectes i la recollida de requisits, impeding una correcta integració contínua del desenvolupament. Tot i així, l'aplicació és suficientment agradable com per ser utilitzada i facilitar la feina als alumnes.

Ha estat una experiència interessant ja que he pogut descobrir noves tecnologies que es segueixen utilitzant avui dia i em podrien servir en un futur en el món laboral. A més ha estat la primera vegada que he hagut de realitzar un projecte des de zero i sense cap company, cosa que m'ha ajudat a ser autosuficient i poder trobar la manera

d'aconseguir la informació rellevant per poder tirar endavant el projecte.

AGRAÏMENTS

En aquesta secció m'agradaria donar el merescut reconeixement al meu tutor del treball de fi de grau ja que sense la seva ajuda no hagués estat possible aconseguir la majoria dels resultats obtinguts en aquest projecte. Sempre que necessitava ajuda podia enviar-li un correu electrònic per resoldre aspectes de requisits del projecte que no tenia clars. També per les reunions que hem realitzat al llarg del semestre per mantenir un bon seguiment del projecte i evitar que m'encallés durant períodes de temps llargs fent que el desenvolupament es veiés greument afectat.

BIBLIOGRAFIA

- [1] «Judge.org», UCommons. [Online] Disponible a: <https://upcommons.upc.edu/handle/2117/91202> [Trobat: 05/10/2023]
- [2] «Judge.org», R11-8.pdf. [Online] Disponible a: <https://upcommons.upc.edu/bitstream/handle/2117/91202/R11-8.pdf?sequence=1&isAllowed=y> [Trobat: 05/10/2023]
- [3] «A new platform for the Online Judge», Online Judge. [Online] Disponible a: <https://onlinejudge.org/index.php> [Trobat: 08/10/2023]
- [4] «hunt problems that matter», uHunt. [Online] Disponible a: <https://uhunt.onlinejudge.org/> [Trobat: 08/10/2023]
- [5] «C++ Winforms Tutorial», Seminare und Bücher für Programmierer. [Online] Disponible a: <https://www.rkaiser.de/c-winforms-tutorial> [Trobat: 10/10/2023]
- [6] «Guía de escritorio (.NET para Windows Forms)», Microsoft. [Online] Disponible a: <https://learn.microsoft.com/es-es/dotnet/desktop/winforms/overview/?view=netdesktop-7.0> [Trobat: 10/10/2023]
- [7] «Github Copilot y Visual Studio 2022», Visual Studio. [Online] Disponible a: <https://visualstudio.microsoft.com/es/vs> [Trobat: 10/10/2023]
- [8] «The AI-Enabled Cross-Platform IDE», QT Group. [Online] Disponible a: <https://www.qt.io/product/development-tools> [Trobat: 25/09/2023]
- [9] «Let's build from here», GitHub. [Online] Disponible a: <https://github.com> [Trobat: 18/09/2023]
- [10] «Move fast, stay aligned, and build better-together», Jira Software, Atlassian. [Online] Disponible a: <https://www.atlassian.com/software/jira> [Trobat: 18/09/2023]

APÈNDIX

A1. PLANIFICACIÓ DEL PROJECTE

Planificació			
Fase	Descripció	Objectius	Durada aprox.
1	Familiaritzar-me amb el projecte inicial de correcció d'exercicis i pràctiques.	-	2 setmanes
2	Escollir tecnologies a utilitzar i planificar el projecte en funció dels objectius proposats.	-	2 setmanes
3	Permetre a l'alumne la capacitat d'iniciar sessió amb una contrasenya nova enviada a través del correu electrònic.	[O1] [Crític]	1,5 setmanes
4	Permetre a l'alumne que pugui fer totes les correccions i lliuraments que té disponibles.	[O2] [Crític]	2,5 setmanes
5	Mostrar el llistat dels lliuraments i correccions que ha fet l'alumne com a historial amb informació com les notes que ha tret.	[O3] [Crític]	1 setmana
6	Poder descarregar-se les últimes correccions realitzades per l'alumne.	[O4]	1 setmana
7	Mostrar estadístiques personals de l'alumne, com per exemple mostrar el nombre de vegades que ha executat un corrector.	[O5]	2 setmanes
8	Mostrar totes les correccions de tots els alumnes per comparar resultats, sigui en forma de gràfica amb un diagrama de barres amb totes les notes de tots els alumnes indicant on es troba l'alumne concret o de manera numèrica.	[O6]	2 setmanes
9	Oferir a l'usuari una interfície còmoda d'utilitzar i intuïtiva de manera que faciliti l'ús de la nova aplicació i representi una millora respecte l'aplicació per consola inicial.	[O7]	19 setmanes

A2. FUNCIO CONNECTLOGIN PER OBTENIR CONTRASENYA

CSocketAES socket;

```
CONNECTLOGIN(SOCKET, L"NEW PASSWORD V1", CLIENTCOURSEID, {}, { CLIENTSTUDENTS.BEGIN()->M_ID});
```

```
void ConnectLogin(CSocketAES& socket, const std::wstring& command, const std::wstring& courseId, const list<CClientStudent>& clientStudents, const list<std::wstring>& args)
{
    try {
        socket.ConnectServer(Server.c_str(), ServerPort.c_str());
    }
    catch (exception ex) {
        throw CorException("The values of the SERVER or the PORT field are not valid for establishing a connection. Please introduce correct values in the CorrectionClient.ini file or check the server state.");
    }
    try {
        socket.SendWString(command);
        socket.SendWString(courseId);
        socket.RecvOk();
    }
    catch (exception ex) {
        throw CorException("The COURSE field value in the CorrectionClient.ini file is not valid. Please, introduce a correct value.");
    }
    try {
        socket.SendSizeT(clientStudents.size());
    }
    catch (exception ex) {
        throw CorException("There are no students in the CorrectionClient.ini file. Please, introduce your NIU and PASSWORD.");
    }
    for (const CClientStudent& s : clientStudents) {
        try {
            socket.SendWString(s.m_Id);
            socket.SendWString(s.m_Password);
            socket.RecvOk();
        }
        catch (exception ex) {
            throw CorException("One of the NIU or PASSWORD field values in the
```

```

CorrectionClient.ini file could not be
found in the server. Please, introduce a
correct NIU or PASSWORD.");
    }
    // if (ok != 1) throw CorExcep-
    tion("El valor del NIU i la contrasenya
    d'un dels alumnes no es troba al servidor.
    Si us plau, introduïu la nova contrasenya
    enviada per correu o comproveu que els NIUs
    són correctes");
    }
    socket.SendSizeT(args.size());
    for (const wstring& arg : args)
    socket.SendWString(arg);
    socket.RecvOk(); // command accepted

    for (;;) {
        int status = socket.RecvInt();
        if (status == 0) break;
        int sz = socket.RecvInt();
        cout << "Waiting in queue position
    " << status << " of " << sz << endl;
        //throw CorException("Waiting in
    queue position " + status + sz);
    }
}

```

A3. FUNCIO RUN

```

void CClientCorrector::Run(bool backu-
pOutFiles)
{
    CSocketAES socket;
    // Borrar los ficheros de salida para
    evitar confusiones
    if (backupOutFiles) {
    }
    else {
        for (std::filesystem::path& file-
        name : m_OutFiles) {
            if (std::filesystem::exists(fi-
            lename)) {
                cout << "Delete file " <<
                ToCCP(filename) << endl;
                std::filesystem::remove(fi-
                lename);
            }
        }
    }

    ConnectLogin(socket, L"CORRECTION V3",
    m_ServerCourseId, ClientStudents, { m_Ser-
    verId });

    cout << "Execute client corrector " <<
    ToCCP(m_ClientId) << " of curse " <<
    ToCCP(m_ServerCourseId) << endl;
    // Enviar los ficheros
    socket.SendInFiles(m_InFiles);

```

```

    cout << "Waiting results of server cor-
    rector " << ToCCP(m_ServerId) << " of curse
    " << ToCCP(m_ServerCourseId) << endl;
    // Ejecutar el comando
    int result = socket.RecvInt();
    if (result == 2) {
        // Error de corrector
        socket.RecvFile("CorrectorOut-
        put.log");
        socket.RecvOk();
        throw CorException("The server has
        not been able to detect the selected cor-
        rector. Please, check the server state.");
    }
    else if (result != 1) {
        wstring message = socket.RecvWS-
        tring();
        throw CorException("The server has
        not been able to obtain the files needed to
        execute the selected corrector. Please,
        check you have the correct files or the
        CorrectionClient.ini file is configured
        correctly.");
    }
    // recibir los ficheros de salida
    //socket.RecvOutFiles(m_OutFiles);
    //closesocket(socket);
    if (!m_ViewCommand.empty()) {
        cout << "EJECUTAR : " <<
        ToCCP(m_ViewCommand) << endl;
        //SystemCMD(m_ViewCommand.c_str());
    }
}

```

A4. FUNCIO CONNECTLOGIN PER OBTENIR LLIURAMENTS

```

CSocketAES socket;

CONNECTLOGIN(socket, L"DELIVERY LIST V2", CLI-
ENTCOURSEID, CLIENTSTUDENTS, { });

```

A5. FUNCIO CONNTECTLOGIN DE LA FUNCIO DELIVERYCLIENTRUN

```

CSocketAES socket;

CONNECTLOGIN(socket, L"DELIVERY V3", COURSEID,
CLIENTSTUDENTS, { DELIVERYID });

```

A6. FUNCIO DOWNLOADCORRECTORFILES

```

private: System::Void DownloadCorrectorFi-
les(wstring correctorName) {
    wstring clientCorrectorName;
    for (CClientCorrector& c : ClientCor-
    rectors) {
        if (c.m_ServerId == corrector-
        Name) {

```

```

        clientCorrectorName =
c.m_ClientId;
    }
    try {
        CClientCorrector& corrector =
GetClientCorrector(ToIdent(clientCorrector-
Name));
        wstring studentId = ClientStu-
dents.begin()->m_Id;
        CSocketAES socket;
        ConnectLogin(socket,
L"DONWLOAD_REGISTRY V1", ClientCourseId,
ClientStudents, { corrector.m_ServerId,
studentId });
        socket.RecvOk(); // Hay regis-
tro de corrección
        std::filesystem::path destina-
tion = socket.RecvWString();
        if (!std::filesystem::cre-
ate_directory(destination)) {
            throw CorException("Er-
ror creating path %s", ToANSI(destina-
tion).c_str());
        }
        // recibir los ficheros deli-
very files
        socket.RecvOk();
        for (;;) {
            wstring name =
socket.RecvWString();
            if (name.empty()) break;
            std::filesystem::path
filename = destination / name;
            cout << "Receive file "
<< ToCCP(filename) << endl;
            socket.RecvFile(file-
name);
        }
        socket.RecvOk(); // Final
MessageBox::Show("Corrector
downloaded successfully.");
    }
    catch (exception& ex) {
        String^ exception = msclr::in-
terop::marshal_as<Sys-
tem::String^>(ToCCP(ex.what()));
        MessageBox::Show(exception);
    }
}

```

A7. FUNCIO GETLASTCORRECTORSCORE

```

void GetLastCorrectorScore(wstring correc-
torId) {
    CSocketAES socket;
    ConnectLogin(socket, L"DONWLOAD_SCORE
V1", ClientCourseId, ClientStudents, { cor-
rectorId, ClientStudents.begin()->m_Id });

```