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

Paz Cepero, Joel; Estravis Nieto, Sara, dir. Multi-Language Integrated Web-Based IDE for Programming Contests. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/280738>

under the terms of the  license

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

Paz Cepero, Joel; Estravis Nieto, Sara, dir. Multi-Language Integrated Web-Based IDE for Programming Contests. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/280738>

under the terms of the  license

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

Paz Cepero, Joel; Estravis Nieto, Sara, dir. Multi-Language Integrated Web-Based IDE for Programming Contests. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/280738>

under the terms of the  license

Multi-Language Integrated Web-Based IDE for Programming Contests

Joel Paz Cepero

Resum— Aquest projecte té l'objectiu d'implementar un IDE offline que permeti als participants de concursos de programació basats en el programari Open Source de DOMjudge, com HPCodeWars, programar en diferents llenguatges de programació en la mateixa aplicació web (principalment Python, C++ i Java), millorant així l'experiència d'usuari i l'eficiència del concurs. La implementació de l'IDE es realitzarà a través d'una investigació exhaustiva per a identificar les possibilitats del sistema ja creat, i es centrarà a proporcionar una interfície d'usuari clara i accessible, amb un temps de resposta ràpid. A més, es buscarà automatitzar les correccions amb els jutges automàtics del concurs, i s'investigarà la possibilitat d'agregar compiladors per a millorar el rendiment de l'IDE. El projecte es durà a terme utilitzant metodologies Agile basades en sprints, amb objectius i reunions de seguiment setmanals, i es realitzaran proves per a determinar el compliment dels requisits i rendiment. A més, s'establirà un sistema de control de versions per fer una bona gestió dels canvis.

Paraules clau— Aplicació web, C++, compilador, concurs de programació, DOMjudge, HPCodeWars, IDE, integrat, multi-llenguatge, offline, programació, Python, Symfony.

Abstract— This project has the objective to implement an offline IDE that allows participants in programming contests based on the Open Source DOMjudge software, such as HPCodeWars, to program in different programming languages within the same web application (mainly Python, C++, and Java), thus improving the user experience and the efficiency of the contest. The implementation of the IDE will be carried out through exhaustive research to identify the possibilities of the already created system, and will focus on providing a clear and accessible user interface, with a fast response time. Additionally, automated corrections will be sought using automatic judges for the contest, and the possibility of adding compilers to improve the performance of the IDE will be investigated. The project will be carried out using Agile methodologies based on sprints, with weekly objectives and follow-up meetings, and tests will be conducted to determine compliance with requirements and performance. Moreover, a version control system will be established for effective change management.

Index Terms— C++, Compiler, programming contest, DOMjudge, HPCodeWars, IDE, integrated, multi-language, offline, programming, Python, Symfony, web application.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

CodeWars Barcelona[1] és un concurs de programació organitzat per voluntaris d'HP Inc.[2], en el qual participen estudiants d'entre tercer d'ESO fins a segon de batxillerat. L'objectiu d'aquest concurs és augmentar l'interès en ciències de la computació per als joves, identificar i impulsar el talent entre els estudiants locals, i fomentar l'ús de l'anglès. L'equip guanyador del concurs tindrà un viatge a Houston, on es celebrarà un concurs similar, organitzat Hewlett-Packard Enterprise[3]. El funcionament del concurs es compon d'uns trenta problemes de programació de diferents tipus, des de manipulació d'strings, fins a optimització de camins amb heurístiques. Cada equip pot programar en el llenguatge de programació que vulgui, dintre dels acceptats pel concurs. Els més utilitzats són Python y C++.

Els participants han de portar la seva màquina, per tal de no haver de familiaritzar-se amb un IDE diferent del que estan acostumats, o amb un sistema operatiu diferent.

L'aplicació del concurs està desenvolupada a partir del projecte Open Source DOMjudge[4], que és un sistema automàtic per realitzar concursos de programació. Aquest *software* implementa una aplicació web amb estructura model-vista-controlador, fent ús del *framework* Symfony.

1.1 Motivació

Cada cop més estan apareixent noves tecnologies, i sobretot a la part del desenvolupament de *software*. Algunes d'aquestes tecnologies és la creixent Intel·ligència artificial, que pot suposar un problema a l'hora de mesurar el nivell de programació i resolució de problemes que tenen els participants del concurs. És per aquest motiu que la competició es realitza de manera *offline*, sense cap classe de connexió a Internet. D'altra banda, molts dispositius treballen íntegrament amb el núvol, de manera que en un entorn sense connexió a Internet no poden funcionar correctament.

-
- E-mail de contacte: joel7.99.21@gmail.com
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Sara Estravis Nieto (Ciències de la Computació i Intel·ligència Artificial)
 - Curs 2022/23

Per tant, implementar un IDE sense connexió a Internet durant l'esdeveniment serà molt útil, ja que permetrà que s'eliminin els problemes mencionats, fent que el concurs es realitzi de manera més justa i fluida.

Aquest projecte, però, no només serà útil per a CodeWars, sinó que podria ser una eina la qual podria aprofitar el sistema DOMjudge. Al ser un programari Open Source, podríem contribuir amb aquest projecte per formar part dels *contributors*.

Dissenyar i crear un IDE fàcil d'utilitzar i amb una interfície amigable pot motivar als estudiants a aprendre programació i explorar noves idees. A més, si finalment tots els estudiants hauran d'utilitzar aquest IDE, pot facilitar que els voluntaris ajudin a qualsevol dubte relacionat amb aquest, ja que estaran familiaritzats amb un únic entorn, que funcionarà de la mateixa manera per a tothom, i no un programari diferent per a cada equip, que potser ningú l'ha utilitzat mai, i pot endarrerir el transcurs del concurs.

2 OBJECTIUS

L'objectiu principal del projecte és desenvolupar aquest IDE multi-llenguatge *offline* i integrar-lo a l'aplicació web existent, creant així una nova versió de l'aplicació amb la funcionalitat implementada.

Alguns dels requisits que haurà de complir seran els següents:

- Haurà de ser intuïtiu, amb aspecte amigable i accessible.
- Haurà de tenir un temps de resposta ràpid.
- Permetrà visualitzar el contingut dels tests d'un problema en concret (*inputs* i *expected outputs*).
- Permetrà escollir el llenguatge que s'utilitzarà.
- Permetrà fer *submissions* d'un problema en concret.
- Distingirà entre els diferents rols d'usuari del concurs.
- No permetrà executar codi maliciós.

Es faran proves exhaustives per determinar el compliment dels requisits i el rendiment. Es faran tests funcionals durant tot el desenvolupament. D'altra banda, es faran sessions amb usuaris per fer proves d'usabilitat i avaluar si el programa compleix amb els requisits de disseny, usabilitat i accessibilitat.

A més, es proporcionarà una documentació completa del projecte, de manera que inclourà un manual d'usuari per facilitar la utilització i el manteniment de l'IDE.

S'haurà de fer una investigació per tal de poder integrar l'IDE al programari existent, i d'igual manera es farà un estudi per poder fer servir els compiladors que utilitzen els jutges automàtics.

S'assegurarà la compatibilitat de l'IDE amb els diferents sistemes operatius i navegadors dels participants per permetre fer-hi un ús universal.

Els llenguatges principals seran Python i C++, però s'estudiarà la possibilitat d'ampliar els llenguatges compatibles, com a mínim a Java i C.

3 PLANIFICACIÓ

S'han establert una sèrie de fases que cal seguir de manera sistemàtica per tal de dur a terme el projecte de manera adequada i ordenada.

Podem observar a l'apèndix el diagrama de Gantt que desglossa temporalment totes les tasques a realitzar durant el projecte.

- Anàlisi de requisits: S'ha realitzat una recopilació exhaustiva de tots els requisits necessaris perquè l'IDE funcioni correctament. S'han de tenir en compte les necessitats específiques dels participants en els concursos els quals l'IDE està pensat. A l'apèndix podem observar aquesta llista de requisits elicitats i classificats segons la seva prioritat en quatre nivells: Baixa, Mitja, Alta, Crítica.
- Anàlisi i investigació del sistema actual: Es realitzarà una investigació per determinar les capacitats i limitacions del sistema ja creat, en aquest cas pel programari DOMjudge. Aquesta investigació servirà per determinar el desenvolupament del *software* del projecte.
- Disseny i desenvolupament back-end: Es realitzarà el disseny i el desenvolupament dels components del back-end de l'IDE. S'ha de tenir en compte els requisits analitzats per tal de complir amb els objectius del sistema.
- Disseny i desenvolupament front-end: Es realitzarà el disseny i el desenvolupament dels components del front-end de l'IDE, amb una interfície clara i accessible, seguint amb els requisits analitzats.
- Proves del sistema exhaustives i anàlisi del disseny: S'han de realitzar proves exhaustives del sistema (tant de la part front-end com de la part back-end) per garantir que els components funcionin correctament i de manera integrada. Es realitzaran user testings per determinar la usabilitat i l'eficàcia de l'aplicació.
- Documentació i manual d'usuari: Es farà un document on s'explicaran de manera detallada i precisa els diferents components del sistema. A més, es farà un manual d'usuari enfocat als voluntaris i participants dels concursos.
- Tancament del projecte: Es farà una defensa del projecte davant d'un comitè avaluador, i es farà entrega de la memòria final del projecte.

4 METODOLOGIA

La metodologia de gestió i desenvolupament del projecte serà Scrum[5]. Scrum és una metodologia Agile de gestió de projectes que s'utilitza per desenvolupar i encarregar productes complexos de manera iterativa i incremental. Divideix el projecte en iteracions curtes (sprints). En cada sprint es treballa amb un conjunt de backlog items seleccionats.

Al començament del desenvolupament del projecte es definirà el Product Backlog, que és una llista prioritzada de totes les funcionalitats o tasques que s'han de realitzar per completar el projecte. S'ordenaran aquestes tasques per prioritat temporal. En aquest projecte es treballarà en sprints setmanals, i es farà una reunió de revisió de l'sprint, on s'avaluaran les tasques dutes a terme, i es faran els ajusts al Product Backlog necessaris. A més, en aquesta reunió es planificarà el següent sprint. A la planificació de l'sprint s'escolliran les tasques a fer del Product Backlog. Es farà també una reunió diària de seguiment. Durant la reunió s'analitzaran els avenços realitzats i s'identificaran els obstacles.

5 EINES I TECNOLOGIES

Per al complet desenvolupament del projecte, s'han seleccionat i utilitzat les següents eines i tecnologies.

5.1 Github

Github[6] és una plataforma de control de versions basada en Git. En aquest projecte, Github serà utilitzat per gestionar les versions del codi, treballant sobre la branca actualitzada de DOMjudge 8.2.0, amb les personalitzacions d'HP. Permetrà el seguiment de canvis en el codi assegurant que es manté una història completa del desenvolupament del projecte. A més, Github ofereix funcionalitats de gestió d'incidències.

5.2 Docker

Docker[7] és un programari de contenidors que permet empaquetar una aplicació i les seves dependències en un contenidor lleuger i portable. En aquest projecte, es farà ús de Docker per crear i executar la imatge del sistema localment. Això permetrà treballar en un entorn consistent i aïllat, garantint que l'aplicació funcioni de manera idèntica en diferents màquines i entorns.

5.3 Visual Studio Code

Visual Studio Code[8] és un editor de codi font lleuger i potent desenvolupat per Microsoft. Aquesta eina serà utilitzada per desenvolupar el projecte, ja que ofereix funcionalitats avançades com la depuració de codi, control de versions integrat i una àmplia gamma d'extensions que faciliten la programació en diferents llenguatges.

5.4 Trello

Trello[9] és una eina de gestió de projectes basada en taulers que facilita la planificació i el seguiment dels sprints, organitzant les tasques en llistes i targetes. Aquesta eina proporciona una visió clara del progrés del projec-

te i les tasques pendents. En aquest projecte, ens serà útil per gestionar els sprints setmanals.

5.5 Marvel App

Marvel App[10] és una plataforma en línia de disseny i prototipatge que permet als dissenyadors, desenvolupadors i equips de producte crear i col·laborar en la creació de prototips d'aplicacions i llocs web interactius. En aquest projecte, s'utilitzarà per dissenyar i crear prototips interactius de la solució de programari que es desenvoluparà.

6 ESTAT DE L'ART

En aquest apartat es realitzarà una revisió exhaustiva i crítica de les últimes tendències i desenvolupaments relacionats amb l'objectiu del projecte.

6.1 IDEs existents

A continuació es presenten alguns exemples d'IDEs web existents, analitzant les seves característiques, avantatges i desavantatges en comparació amb el projecte que es vol desenvolupar.

6.1.1 Replit

Replit[11] és un IDE web en línia que permet programar en diversos llenguatges de programació, com Python, Java i C++. Aquest IDE és molt intuïtiu i fàcil d'utilitzar, amb un entorn de desenvolupament lleuger i ràpid. No obstant això, com que és un IDE en línia, requereix connexió a Internet per funcionar, a diferència del projecte que es desenvoluparà en aquest treball, que serà *offline*.

6.1.2 CodeSandbox

CodeSandbox[12] és un IDE web en línia que permet crear aplicacions web i projectes en diversos llenguatges de programació, com React, Vue, Angular, entre d'altres. Ofereix una gran varietat de plantilles per iniciar el desenvolupament ràpidament, però també requereix connexió a Internet, i està més enfocat al desenvolupament d'aplicacions web que al concurs de programació.

6.1.3 Jupyter Notebook

Jupyter Notebook[13] és una aplicació web de codi obert que permet crear i compartir documents amb codi en viu, equacions, visualitzacions i text narratiu. Tot i que és una eina potent per a la programació en Python, no ofereix suport per a altres llenguatges de programació com C++ i Java, i està més orientada a la ciència de dades que als concursos de programació.

6.1.4 Theia

Theia[14] és un IDE basat en la web de codi obert i extensible que permet desenvolupar aplicacions en diversos llenguatges de programació. Pot ser desplegat tant en línia com a local, proporcionant així una solució *offline*. Tot i que Theia ofereix una gran flexibilitat i extensibilitat, pot ser més complex de configurar i mantenir en comparació amb un IDE específicament dissenyat per als concursos de programació DOMjudge, ja que busquem una integració amb els tests dels problemes.

Tots aquests IDEs web tenen les seves avantatges i desavantatges. No obstant això, cap d'ells està específicament dissenyat per a l'ús en un concurs de programació *offline* com el que es busca desenvolupar en aquest treball. El projecte a desenvolupar haurà d'adaptar-se a les necessitats específiques dels participants en els concursos, oferint suport per a múltiples llenguatges de programació, una interfície intuïtiva i ràpida, i la possibilitat de treballar *offline* i fer proves específiques per als problemes.

6.2 Eines disponibles a DOMjudge

A continuació, es realitzarà un anàlisi de les eines que proporciona DOMjudge, amb la finalitat d'identificar quines d'aquestes poden ser útils en la implementació de l'IDE que es planteja en aquest projecte. Es farà especial èmfasi en les eines que faciliten la gestió de codi, la compilació i l'execució de tests.

6.2.1 Editor de codi

DOMjudge proporciona un editor de codi per als membres del jurat, que poden modificar el codi d'una *submission*. Aquesta funcionalitat és interessant per al projecte, ja que pot ser adaptada perquè els concursants puguin editar/crear el seu codi directament a l'aplicació web. No obstant això, l'editor de codi té algunes limitacions, com per exemple el seu disseny, el qual no és gens intuïtiu si no estàs familiaritzat amb l'entorn del concurs, i només permet editar un codi enviat amb anterioritat. Per tant, s'ha decidit desestimar utilitzar aquest mòdul i dissenyar un específic per a l'IDE. D'altra banda, aquest editor fa ús de la llibreria Ace, la qual s'utilitzarà per tal de seguir amb la estructura de l'aplicació.

6.2.2 Judgehosts

Aquests són els components encarregats de compilar, executar i comprovar les *submissions* dels concursants. Els judgehosts són crucials per a la implementació de l'IDE, ja que permeten la interacció amb els compiladors i la realització d'execucions. Cal investigar com integrar aquests components amb l'IDE de manera eficient i sense sobrecarregar-los, de manera que no afecti al rendiment del concurs.

Els judgehosts funcionen de la següent manera: quan una *submission* és enviada, el sistema l'envia als judgehosts per a ser processada. Cada judgehost té la capacitat d'interactuar amb diversos compiladors i interpretats de llenguatges de programació, permetent la execució i avaluació de les solucions.

Per integrar aquests components amb l'IDE, s'haurà de dissenyar una comunicació adequada entre l'entorn d'edició i els judgehosts.

Després de l'anàlisi de les eines disponibles a DOMjudge, es pot concloure que aquestes poden ser útils per a la implementació de l'IDE. L'editor de codi i els judgehosts són dos components clau que es poden adaptar i integrar per proporcionar una experiència d'usuari millorada als concursants.

7 DESENVOLUPAMENT BACK-END

En aquest apartat, es presentarà i es discutirà el desenvolupament del back-end per al projecte. Al llarg d'aquest apartat, es descriuran i justificaran els avanços realitzats en el desenvolupament del back-end, incloent la creació i la configuració de serveis, la gestió de bases de dades, la implementació de lògica d'aplicació i la comunicació amb el front-end.

7.1 Creació d'un tab IDE

A continuació es detallarà el procés de creació del nou tab "IDE" en el menú de navegació de la plataforma web, així com les raons per les quals s'ha decidit diferenciar el contingut del tab entre els membres del jurat i els equips participants.

El tab "IDE" es va crear per facilitar l'accés a l'IDE directament des de la plataforma web i des de qualsevol apartat de l'aplicació. Tot i que el tab "IDE" està present en el menú de navegació tant per als membres del jurat com per als equips, el contingut que cada grup pot visualitzar i utilitzar és diferent. Aquesta diferència es deu a les necessitats i funcions específiques de cada rol en el context del concurs:

- Jurat: Els membres del jurat tindran accés permanent a aquest apartat, a més de disposar de la disponibilitat de totes les funcionalitats.
- Equips: La disponibilitat del tab "IDE" per als equips pot ser opcional i dependrà de les opcions de configuració del concurs. En alguns casos, els organitzadors del concurs poden decidir deshabilitar l'IDE integrat per als equips, per exemple, per fomentar l'ús d'altres eines o plataformes específiques.

Per implementar aquesta diferenciació, es va desenvolupar una lògica de control d'accés basada en els rols dels usuaris. Això permet mostrar el contingut adequat a cada tipus d'usuari i assegurar que les funcions i les dades estiguin correctament protegides i gestionades.

7.2 Editor de codi

En aquest apartat es descriurà l'editor de codi afegit a la plataforma web, que ha estat implementat amb Ace[15].

L'editor de codi està dissenyat perquè els administradors del concurs puguin seleccionar un llenguatge per defecte per a l'IDE, per exemple, Python 3, i així crear un IDE Python.

S'ha afegit la capacitat de modificar el llenguatge de l'IDE, de manera que es pot escollir entre tots els llenguatges disponibles en aquest moment en el concurs. A més, els administradors poden deshabilitar aquesta opció per als equips, de manera que només tinguin permès un llenguatge. També s'ha afegit l'opció de canviar el disseny de l'IDE entre un editor fosc o clar, permetent a cada programador escollir la seva preferència.

L'editor de codi està dissenyat per oferir una experiència de programació eficient i intuïtiva per als usuaris. Aquesta eficiència es pot aconseguir gràcies a les característiques integrades d'Ace, com la destacació de sintaxi.

7.3 Visualitzador de tests

En aquesta secció es presentarà el visualitzador de tests que incorporará l'aplicació, el qual es tracta d'un selector de problema dinàmic, on els usuaris poden indicar per a quin problema volen visualitzar els tests, i un selector de tests.

L'objectiu d'aquesta funcionalitat és permetre als usuaris del concurs accedir de manera còmoda i ràpida als diferents tests d'un problema específic, mostrant per a cada test l'input corresponent i la sortida esperada pel corrector, amb la finalitat d'oferir una visió millorada durant la programació sobre el format esperat.

Aquesta implementació té l'objectiu d'agilitzar la comprensió dels tests i la programació de la solució. Els tests públics, però, són els mateixos que s'indiquen a l'enunciat de cada problema.

Finalment, aquest selector de problema s'utilitzarà per indicar al sistema quin és el problema pel qual s'està fent la *submission*.

Es farà una diferenciació d'usuaris, ja que els membres del jurat tenen la opció de poder visualitzar tots els tests, tant públics com privats, i els membres dels equips participants només poden veure els tests públics.

7.4 Submission

A continuació es detallarà la implementació de l'acció de realitzar una *submission* en el sistema.

Per dur a terme l'enviament d'una resposta per a un problema específic, s'ha utilitzat tecnologies AJAX que permeten fer peticions asíncrones al sistema. D'aquesta manera, es crea un fitxer temporal que conté el codi escrit per l'usuari a la secció de Codi Font de l'IDE, el llenguatge seleccionat al selector de llenguatges, i el problema concret triat al selector de problemes.

Aquest fitxer, juntament amb la informació del llenguatge i el problema, s'incorporen a una variable anomenada "submission", que posteriorment s'envia a la cua de treball dels judgehosts. Els judgehosts prenen els diferents elements de treball en funció de la seva prioritat i retornen el resultat de l'enviament a la llista de *submissions*.

En cas que el sistema accepti l'enviament, surt un missatge per pantalla indicant que l'enviament ha estat exitós. D'altra banda, si l'usuari no ha realitzat un enviament correcte, el sistema mostra un missatge d'error i indica les accions que l'usuari hauria de prendre per corregir-lo.

Si el resultat és correcte, es sumen els punts del problema a la puntuació total de l'equip. Si és incorrecte, s'indica que la solució donada no ha sigut bona.

7.5 Editar enviament anterior

En aquest apartat es tractarà la funcionalitat d'editar un codi font enviat anteriorment.

S'ha afegit un selector d'enviaments anteriors, el qual una vegada has fet un enviament d'un problema en específic, pots seleccionar un dels enviaments d'aquest problema que hagi fet anteriorment.

Una vegada selecciones l'enviament, apareix en l'editor de codi el codi font d'aquest, el qual es pot editar directament, fent que sigui molt més fàcil fer canvis, sense haver de fer un codi des de zero cada vegada que fas un enviament incorrecte.

7.6 Classes desenvolupades

A continuació es farà la explicació de les diferents classes utilitzades per al desenvolupament del projecte.

A l'apartat 15.4 de l'apèndix podem observar un diagrama de classes que representa les relacions entre les diferents classes, i es presenten les classes creades.

A partir d'aquesta classe, s'han fet servir d'altres ja desenvolupades a l'aplicació DOMjudge, les quals són representades com a entitats, ja que fan referència a dades o funcionalitats específiques dins del sistema, como ara la classe *Submission*, *Contest*, *Problem*, etc.

8 DESENVOLUPAMENT FRONT-END

En aquesta secció, s'exposarà i s'analitzarà el procés de desenvolupament del Front-end per al projecte. Durant aquesta secció, es detallaran i es donaran raons sobre les decisions de disseny, a més d'analitzar les estadístiques obtingudes a la prova de prototipatge amb diferents usuaris.

8.1 Layout

A continuació es presentarà l'anàlisi de disseny de la distribució de l'aplicació de l'IDE.

Com podem observar a l'apèndix 15.3, la distribució de l'IDE consta de l'editor de text per al *Source Code*, que serà on l'usuari escriurà el seu codi font de resposta, l'editor de text per a l'*Input*, que servirà per mostrar les entrades al programa per a un dels tests seleccionat, i l'editor de text per a l'*Expected Output*, que serà on es mostrarà el contingut de les sortides del programa esperades per a un determinat test seleccionat.

La decisió d'aquesta distribució es basa en poder tenir una disposició que faciliti als participants dur a terme les tasques principals del concurs. En aquest cas, s'ha decidit donar protagonisme a l'editor de text de Codi Font. S'han realitzat proves per determinar la mida òptima de l'editor de text per aprofitar al màxim l'espai entre els diferents editors. D'altra banda, els editors de text d'Entrades i Sortides Esperades tenen una mida més reduïda que l'edi-

tor principal, ja que la gran majoria dels tests no requereixen un espai més gran.

Finalment, s'han distribuït els diferents editors de manera compacta, però sense sobrecarregar la visió, la qual cosa agilitza les accions de l'usuari de forma significativa.

8.2 Colors

En aquesta secció, es justificarà la selecció dels colors utilitzats a l'aplicació.

En primer lloc, el color principal dels editors de text s'ha basat en estudis que indiquen que el color predominant a l'hora de seleccionar un tema pels editors de codi és el fosc[16], que supera el clar. És per això que, per defecte, els editors es mostren amb un estil de disseny fosc. No obstant això, s'ha implementat un selector d'estils pels editors, on els usuaris poden seleccionar si prefereixen un estil fosc o clar.

En segon lloc, s'ha implementat l'adaptació de l'estil de la font per ajustar-se al llenguatge de programació seleccionat. Això permet destacar la sintaxi específica de cada llenguatge, com ara la distinció dels tipus de variables, els noms de funcions o variables, els comentaris, etc.

A la Figura 1 es pot observar com es destaquen els diferents aspectes del codi escrit en Python, i aquest mateix codi escrit en C++ a la Figura 2, permetent ressaltar-les diferències de comportament segons el llenguatge seleccionat. Aquesta funcionalitat millora la llegibilitat i facilita la comprensió del codi pels desenvolupadors.

Source code

```
1 # Funció que imprimeix "Hello, World!"
2 def hello():
3     print("Hola, món!")
4
5 # Crida a la funció hello
6 hello()
7
```

Figura 1: Hello world escrit en Python

Source code

```
1 #include <iostream>
2
3 // Funció que imprimeix "Hello, World!"
4 void hello() {
5     std::cout << "Hola, món!" << std::endl;
6 }
7
8 // Funció principal
9 int main() {
10     // Crida a la funció hello
11     hello();
12
13     return 0;
14 }
15
```

Figura 2: Hello world escrit en C++

Finalment, s'ha decidit que el botó de fer *Submission* tingui el mateix color que s'utilitza en el logotip d'HP, ja que aquest color destaca i crida l'atenció. Aquesta elecció es basa en la idea de captar l'interès dels usuaris i guiar-los

cap a l'acció desitjada.

El color utilitzat en el logotip d'HP és un color vibrant i distintiu, que sobresurt entre altres colors i captiva l'atenció de l'espectador. A més, aquest color contribueix a la coherència visual i a l'establiment d'una identitat de marca consistent.

8.3 Prototip de Disseny

En aquest apartat es descriurà el prototip de disseny inicial que es va desenvolupar amb l'objectiu d'analitzar l'experiència dels usuaris i utilitzar aquest *feedback* per determinar el disseny definitiu de l'aplicació.

El prototip de disseny va ser creat com una representació visual i interactiva de l'aplicació, que permetia als usuaris experimentar amb les funcionalitats i la interfície de l'aplicació de manera simulada. Aquest prototip va incloure elements com botons, menús, icones i altres elements de disseny que s'esperaven trobar en la versió final de l'aplicació.

Durant aquesta fase del desenvolupament, es va sol·licitar als usuaris que realitzessin una simulació d'enviament de codi per a un problema seleccionat, tenint en compte el context de ser un usuari que havia llegit el problema i sabia què escriuria a la solució.

Els resultats obtinguts van ser molt positius, ja que el 100% dels enquestats van afirmar que la interfície de l'IDE era molt fàcil i intuïtiva, i que no van tenir problemes d'utilització. Van destacar la claredat dels botons i menús, la organització de les funcionalitats i la facilitat per accedir a les diferents seccions de l'aplicació.

A més, els usuaris van expressar la seva satisfacció pel fet que la interfície de l'IDE no els sobre-carregava amb informació innecessària més enllà de les instruccions de context proporcionades pel problema seleccionat. Això els va permetre centrar-se en la seva tasca de desenvolupament sense distraccions ni confusions.

Basant-se en aquesta retroalimentació positiva, el prototip de disseny es va considerar exitós i es van utilitzar aquests resultats per refinar i definir el disseny definitiu de l'aplicació. S'han tingut en compte els suggeriments dels usuaris per millorar encara més la usabilitat i l'experiència de l'usuari en l'aplicació final.

9 PROVES DEL SISTEMA

En aquest apartat, s'abordaran les diferents proves que s'han realitzat per assegurar el compliment de tots els requisits del sistema. A continuació es detallen les proves realitzades en les següents àrees:

9.1 Proves d'usabilitat

S'han realitzat proves exhaustives per avaluar la usabilitat de l'aplicació des del punt de vista dels usuaris. Aquestes proves han implicat la participació activa d'usuaris reals, que han interaccionat amb l'aplicació i han proporcionat retroalimentació sobre la seva experiència d'ús.

S'han tingut en compte factors com la facilitat d'aprenentatge, la intuïtivitat de la interfície, la facilitat de navegació i la claredat de les funcionalitats. Aquestes proves s'han dut a terme amb el prototip de disseny interactiu abans mencionat.

9.2 Proves de rendiment

Per tal de garantir un rendiment òptim de l'aplicació, s'han realitzat proves de rendiment exhaustives que han analitzat el comportament del sistema en situacions de càrrega i amb un volum significatiu de dades. Tot i que en edicions anteriors de la competició d'HPCodeWars hi ha participat més de 100 equips, les proves de rendiment no s'han centrat en tenir un gran nombre d'equips connectats simultàniament, sinó en simular el comportament del concurs mitjançant enviaments de solucions de forma continuada.

El sistema està dissenyat amb una mesura de seguretat per garantir un equilibri en la cua de treball. Es prioritzen els enviaments en funció de paràmetres específics, com ara l'enviament més recent d'un dels equips participants. Això assegura que no és possible abusar de l'aplicació per retardar la correcció dels altres equips. Durant les proves, s'ha comprovat que el sistema manté un bon rendiment, fins i tot en situacions de molts enviaments.

S'ha arribat a la conclusió que el sistema compleix amb els requisits establerts en termes de rendiment. Això significa que l'aplicació pot gestionar amb eficiència un nombre considerable d'enviaments i proporcionar una resposta ràpida i fiable als equips participants.

9.3 Proves de seguretat

La seguretat és una prioritat en el desenvolupament del sistema. Per tant, s'han realitzat proves per identificar i abordar possibles vulnerabilitats de seguretat. S'han dut a terme simulacions d'atac amb diferents *scripts*. Aquestes proves han inclòs intents d'infiltració i intents de robatori d'informació. S'ha treballat per assegurar que les dades dels usuaris estiguin protegides i que l'aplicació estigui immune a possibles amenaces i vulnerabilitats.

Per defecte, el sistema de seguretat dels judgehosts és molt robust, i compta amb diversos mecanismes que fan que sigui molt difícil fer un atac a través d'un enviament de codi.

El sistema de DOMjudge diferencia entre els rols d'usuari i jurats, de manera que els administradors configuren el tipus d'enviament que poden fer els participants, augmentant les mesures de seguretat.

9.4 Proves de compatibilitat

S'han dut a terme proves exhaustives per garantir que l'aplicació sigui plenament compatible amb diferents sistemes operatius i navegadors. L'objectiu d'aquestes proves de compatibilitat és assegurar-se que els usuaris puguin accedir i utilitzar l'aplicació sense problemes en qualsevol entorn.

Les proves de compatibilitat s'han realitzat utilitzant una àmplia varietat de configuracions de navegadors, com ara Google Chrome, Mozilla Firefox, Safari, Opera i Microsoft Edge.

En conclusió, les proves exhaustives de compatibilitat han confirmat que l'aplicació és totalment funcional i compatible amb una àmplia gamma de sistemes operatius i navegadors. Aquesta compatibilitat garantida és essencial per assegurar una experiència d'usuari òptima i garantir l'èxit de la competició en qualsevol entorn informàtic.

9.5 Conclusions

Les proves realitzades en les diferents àrees del sistema han proporcionat resultats satisfactoris i han contribuït a assegurar el compliment dels requisits establerts. Les proves d'usabilitat han demostrat que l'aplicació és fàcil d'utilitzar i que proporciona una experiència positiva als usuaris. Les proves de rendiment han confirmat que el sistema pot gestionar amb eficiència un volum significatiu d'enviaments i mantenir un bon rendiment en tot moment. Les proves de seguretat han posat de manifest la robustesa del sistema i la seva capacitat per resistir intents d'infiltració i robatori d'informació.

10 RESULTATS

En aquest apartat es farà un anàlisi del projecte dut a terme, tenint en compte el desenvolupament i la viabilitat del projecte.

10.1 Objectius assolits

El desenvolupament del projecte ha estat un èxit en termes de complir amb els requisits proposats en la fase de planificació. S'han assolit amb èxit els següents objectius clau:

- **Compliment dels requisits funcionals:** Totes les funcionalitats i característiques requerides per l'aplicació s'han desenvolupat i implementat amb èxit. El sistema permet als participants enviar i avaluar solucions de codi, gestionar els problemes de la competició i interactuar amb la interfície d'usuari de manera eficient i intuïtiva.
- **Rendiment òptim:** S'han realitzat proves de rendiment exhaustives per assegurar un funcionament eficient i ràpid de l'aplicació, fins i tot en situacions de càrrega i volum significatiu de dades.
- **Seguretat robusta:** S'han realitzat proves de seguretat exhaustives per identificar i abordar possibles vulnerabilitats.
- **Usabilitat i compatibilitat:** S'han dut a terme proves d'usabilitat per avaluar l'experiència d'ús dels participants. L'aplicació ha rebut una retroalimentació positiva en termes de facilitat d'aprenentatge, intuïtivitat de la interfície, facilitat de navegació i claredat de les funcionalitats.

S'ha desenvolupat una aplicació que permet realitzar les següents accions:

- Desenvolupar el codi font de l'enviament.
- Seleccionar un problema concret de la llista del concurs.
- Canviar l'estil gràfic de l'editor de codi font.
- Seleccionar un llenguatge de programació concret.
- Editar un codi font enviat anteriorment.
- Visualitzar els tests d'un problema en concret.
- Fer l'enviament de la resposta a través de l'IDE.

Podem observar al diagrama de seqüència de l'apèndix 15.5, les accions que pot realitzar un usuari que utilitza l'aplicació en el concurs.

10.2 Viabilitat del projecte

Amb el desenvolupament del sistema, s'ha pogut concloure que el projecte és viable per a un enginyer informàtic, tenint en compte el temps empleat i els recursos utilitzats. El projecte ha estat desenvolupat per una única persona, qui ha dedicat un temps de 4 hores diàries durant 5 dies a la setmana dedicades al desenvolupament de *software*.

Tots els objectius han estat assolits de manera satisfactòria, demostrant la viabilitat del projecte des d'un punt de vista temporal i de recursos.

11 FUTURES MILLORES

En aquesta secció s'abordarà les possibles futures millores per al sistema desenvolupat. El projecte ha estat desenvolupat amb una arquitectura escalable i modular, que permet la implementació de futures millores sense haver de modificar l'estructura ja existent. Això proporciona una base sòlida per a una aplicació dinàmica i adaptable a diferents situacions, amb la capacitat d'escalar i respondre a les necessitats futures.

11.1 Execució de codi de prova

Durant el desenvolupament de l'IDE per fer enviaments i visualitzar els tests dels problemes ha sorgit la idea d'afegir una funcionalitat extra en implementacions futures, com és la de realitzar execucions de prova mitjançant l'IDE.

Aquesta funcionalitat pot ser molt útil en augmentar l'eficiència dels concursants, permetent-los provar el seu codi ràpidament abans de realitzar enviaments. Per aconseguir això, caldrà investigar les opcions per optimitzar l'ús dels recursos dels judgehosts i considerar la implementació de judgehosts dedicats a aquesta tasca.

11.1.1 Proves amb inputs personalitzats

Una de les implementacions previstes per millorar l'experiència de l'usuari dels participants i augmentar la seva eficiència i comprensió en el desenvolupament del seu codi és habilitar l'execució de tests amb casos personalitzats. Això permet als participants tenir en compte possi-

bles situacions límit (*corner-cases*) que podrien formar part dels tests privats, comprovar que el seu codi funcioni amb diferents inputs i no només amb els tests públics.

L'objectiu d'aquesta funcionalitat és no només permetre als participants desenvolupar el seu codi basant-se en les proves realitzades, sinó també intentar comprendre per què el seu enviament no ha funcionat en cas que els tests públics hagin estat superats.

Mitjançant l'ús d'aquests tests personalitzats, els participants podran explorar diferents situacions d'entrada i avaluar l'eficàcia del seu codi en escenaris més variats. Això els permetrà millorar la robustesa i l'adaptabilitat del seu codi, tenint en compte una àmplia gamma de casos possibles.

12 CONCLUSIONS

A continuació es tractaran les conclusions obtingudes en la realització del projecte, així com els problemes trobats i les seves solucions durant el desenvolupament.

12.1 Problemes i solucions

La planificació del projecte en quant al temps de realització de cada tasca ha sigut bastant optimista, ja que no es planificava que hi hagués cap problema en el desenvolupament que pugues endarrerir el treball, i finalment va aparèixer un problema amb una implementació.

La primera idea que es va desenvolupar dintre del termini de temps va ser la de executar els codis enviats pels concursants a través del servidor de DOMjudge, on arriben totes les peticions dels usuaris, però aquesta opció es va desestimar per temes de seguretat i funcionalitat, ja que en fer proves de seguretat es van trobar algunes vulnerabilitats, com extreure informació del servidor utilitzant comandes al codi font d'enviament. Pel que fa a la funcionalitat, es va considerar que la millor opció era que tots els enviaments fossin executats pel mateix sistema, que era el dels judgehosts.

Finalment, es va arribar a la conclusió que era millor fer ús dels judgehosts, ja que aquests compten amb un sistema de seguretat molt robust, i tots els enviaments utilitzarien els mateixos compiladors i sistema de comparació de codi.

Això va endarrerir el treball i es va tornar a planificar de manera que va suposar una major càrrega de treball per a futures jornades, però finalment es va poder acabar segons el previst.

12.2 Conclusions finals

La realització d'aquest treball ha estat extremadament satisfactòria, ja que s'ha aconseguit desenvolupar una eina que serà d'una gran utilitat per a la realització del concurs d'HPCodeWars. Aquesta competició és un acte que em genera una gran admiració i entusiasme, i estic molt satisfet amb el resultat final del projecte.

Durant el procés de desenvolupament, s'han superat amb èxit els reptes i desafiaments, i s'ha treballat de manera eficient per assolir els objectius establerts. L'aplicació final resulta completa, amb una interfície intuïtiva i un rendiment òptim. Això ha estat possible gràcies a l'esforç personal i els consells de l'equip de desenvolupament d'HP, que han estat de molta ajuda.

El desenvolupament d'aquesta eina ha estat una gran oportunitat per aprendre i aplicar coneixements en el camp de l'enginyeria de programari, ja que ha suposat un repte comprendre el funcionament complet d'una aplicació de gran envergadura i desenvolupar una nova funcionalitat seguint el model de programació establert.

A més, aquesta aplicació ha estat desenvolupada de manera que permeti afegir noves funcionalitats en el futur, i potser s'integrarà en el *software* oficial de l'aplicació de DOMjudge.

Aquest treball m'ha proporcionat un gran coneixement nou i m'ha permès aprofundir en algunes de les assignatures del meu grau.

En definitiva, estic molt satisfet amb els resultats obtinguts i confio que aquesta eina sigui ben rebuda pels participants i contribueixi a enriquir l'experiència dels concursants en l'HPCodeWars.

13 AGRAÏMENTS

Vull expressar el meu agraïment a totes les persones que han contribuït a l'èxit d'aquest Treball de Fi de Grau. El seu suport i orientació han estat fonamentals per al desenvolupament d'aquest projecte.

En primer lloc, vull agrair a HP Inc. per brindar-me l'oportunitat de realitzar les meves pràctiques professionals a la seva empresa. A més, m'han donat el suport per basar el meu treball en un dels projectes que em van oferir, del qual estic molt satisfet d'haver realitzat.

A més, vull expressar la meua gratitud als meus companys de HP Inc. que em van ajudar i col·laborar a solucionar els problemes tècnics que van afectar el flux de desenvolupament, a més de donar el seu feedback. La seva disposició per desbloquejar situacions i proporcionar solucions ràpides va ser fonamental per superar els obstacles i avançar en el projecte.

També vull agrair especialment a la meua tutora, qui em va guiar i assessorar al llarg de tot el procés d'elaboració del Treball de Fi de Grau. Agraeixo la seva dedicació i disponibilitat per respondre les meves preguntes i orientar-me en cada etapa del projecte. A més, agraeixo que hagi accedit a tutoritzar el meu treball amb tant poc de temps per tenir el context del treball.

14 BIBLIOGRAFIA

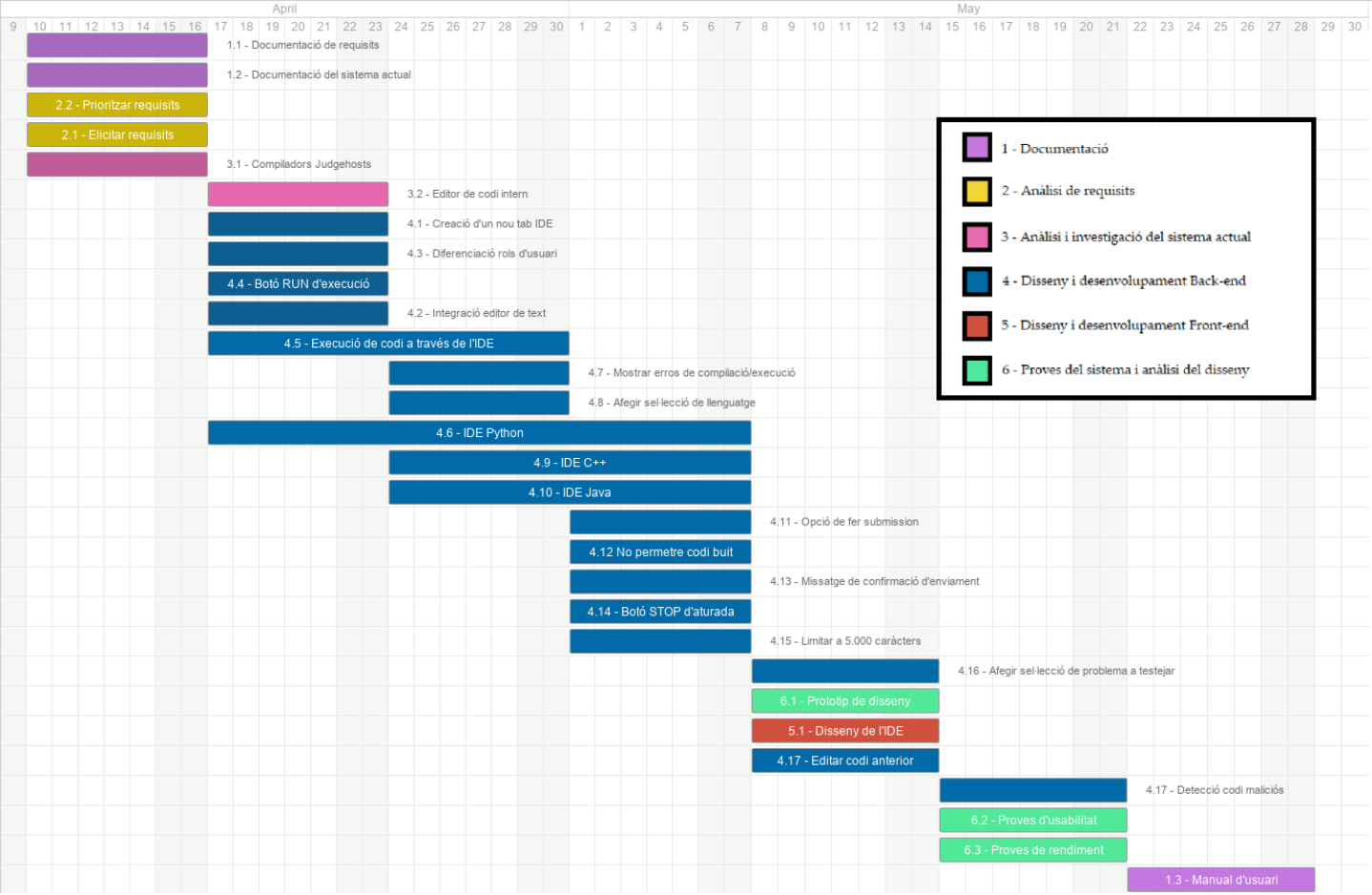
- [1] HP CodeWars Barcelona. "Who are we and what is the HP CodeWars about?", [En línia], Disponible a: <https://www.hpcodewarsbcn.com/about-us/>
- [2] HP Inc. "HP Inc.", [En línia], disponible a: <https://www8.hp.com/us/en/home.html>
- [3] Hewlett-Packard Enterprise. "Hewlett-Packard Enterprise", [En línia], disponible a: <https://www.hpe.com/us/en/home.html>
- [4] DOMjudge. "About the project", [En línia], Disponible a: <https://www.domjudge.org/about>
- [5] Atlassian. "Guía de la metodología scrum: qué es, cómo funciona y cómo empezar", [En línia], Disponible a: <https://www.atlassian.com/es/agile/scrum>
- [6] Github. "The complete developer platform to build, scale, and deliver secure software", [En línia], Disponible a: <https://github.com/about>
- [7] Docker. "Developers bring their ideas to life with Docker", [En línia], Disponible a: <https://www.docker.com/why-docker/>
- [8] Visual Studio Code. "Code editing. Redefined.", [En línia], Disponible a: <https://code.visualstudio.com/#meet-intellisense>
- [9] Trello. "Trello facilita a los equipos la gestión de proyectos y tareas.", [En línia], Disponible a: <https://trello.com/tour>
- [10] Marvel App. "We're on a mission to help the world bring ideas to life", [En línia], Disponible a: <https://marvelapp.com/about-us/>
- [11] Replit. "Make something great.", [En línia], Disponible a: <https://replit.com/>
- [12] CodeSandbox. "Code. Review. Deploy.", [En línia], Disponible a: <https://codesandbox.io/>
- [13] Jupyter Notebook. "Free software, open standards, and web services for interactive computing across all programming languages", [En línia], Disponible a: <https://jupyter.org/>
- [14] Theia. "An Open, Flexible and Extensible Cloud & Desktop IDE Platform.", [En línia], Disponible a: <https://theia-ide.org/>
- [15] Ace. "The high performancecode editor for the web.", [En línia], Disponible a: <https://ace.c9.io/>
- [16] Quora. "¿Por qué los desarrolladores prefieren el tema oscuro en los editores de código?", [En línia], Disponible a: <https://es.quora.com/Por-qué-los-desarrolladores-prefieren-el-tema-oscuro-en-los-editores-de-código>

15 APÈNDIX

15.1 Requisites

ID	TIPUS	REQUISIT	PRIORITAT
RF1	Funcional	El sistema ha de acceptar codi escrit en UTF-8.	Alta
RF2	Funcional	El sistema ha de mostrar un missatge d'error d'execució/compilació en cas de haver-hi.	Mitja
RF3	Funcional	El sistema no permetrà enviar un codi buit.	Alta
RF4	Funcional	El sistema ha de permetre visualitzar els tests d'un problema en concret.	Mitja
RF5	Funcional	El sistema ha de mostrar un missatge de confirmació de l'enviament a un dels problemes.	Baixa
RF6	Funcional	El sistema ha de ser capaç de fer un enviament d'una solució a un problema.	Crítica
RF7	Funcional	El sistema ha de proveir a l'usuari la capacitat d'escriure codi en Python.	Crítica
RF8	Funcional	El sistema ha de proveir a l'usuari la capacitat d'escriure codi en C++.	Crítica
RF9	Funcional	El sistema ha de proveir a l'usuari la capacitat d'escriure codi en Java.	Crítica
RF10	Funcional	El sistema ha de suportar codis de fins a 5000 caràcters.	Mitja
RF11	Funcional	El sistema ha de tenir un sistema d'autorització d'usuaris i permisos que restringeixi l'accés a certes funcionalitats del IDE en funció del rol de cada usuari.	Alta
RF12	Funcional	El sistema no deixarà executar codi potencialment maliciós.	Crítica
RF13	Funcional	El sistema ha de permetre la selecció dels diferents llenguatges de programació.	Crítica
RF14	Funcional	Si l'usuari pressiona el botó RUN, el sistema ha de començar l'execució de l'aplicació.	Alta
RF15	Funcional	El sistema ha de permetre la selecció dels diferents estils d'interfície de programació.	Baixa
RD1	Disseny	El sistema ha de tenir una interfície amigable.	Alta
RD2	Disseny	El sistema ha de ser intuïtiu i fàcil d'utilitzar.	Alta
RD3	Disseny	El sistema ha de seguir l'estil utilitzat en l'aplicació principal (colors, fonts, etc.).	Mitja
RD4	Disseny	El sistema ha de proporcionar colors que facilitin la comprensió del codi.	Alta
RR1	Rendiment	El sistema ha de tenir un temps d'espera màxim (timeout) de 10 segons.	Crítica
RR2	Rendiment	El sistema ha de tenir un temps d'execució mitjà menor a 5 segons.	Alta
RR3	Rendiment	Si hi ha més de 100 usuaris connectats simultàniament, el sistema ha de mantenir una velocitat de resposta de menys de 1 segon per sol·licitud.	Alta
RR4	Rendiment	Tan aviat com l'usuari obri l'aplicació, el sistema ha de carregar la pantalla principal en menys de 2 segons.	Mitja
RR5	Rendiment	Tan aviat com l'usuari selecciona una opció del menú, el sistema ha de carregar la següent pantalla en menys de 1 segon.	Alta

15.2 Diagrama de Gantt



15.3 UI

Select theme:

Dark

Select language:

Python 3

Select problem:

3DBoxDrawing

Select testcase example:

-- select a testcase --

Select previous submission:

-- no submissions yet --

Source code

1

Submit

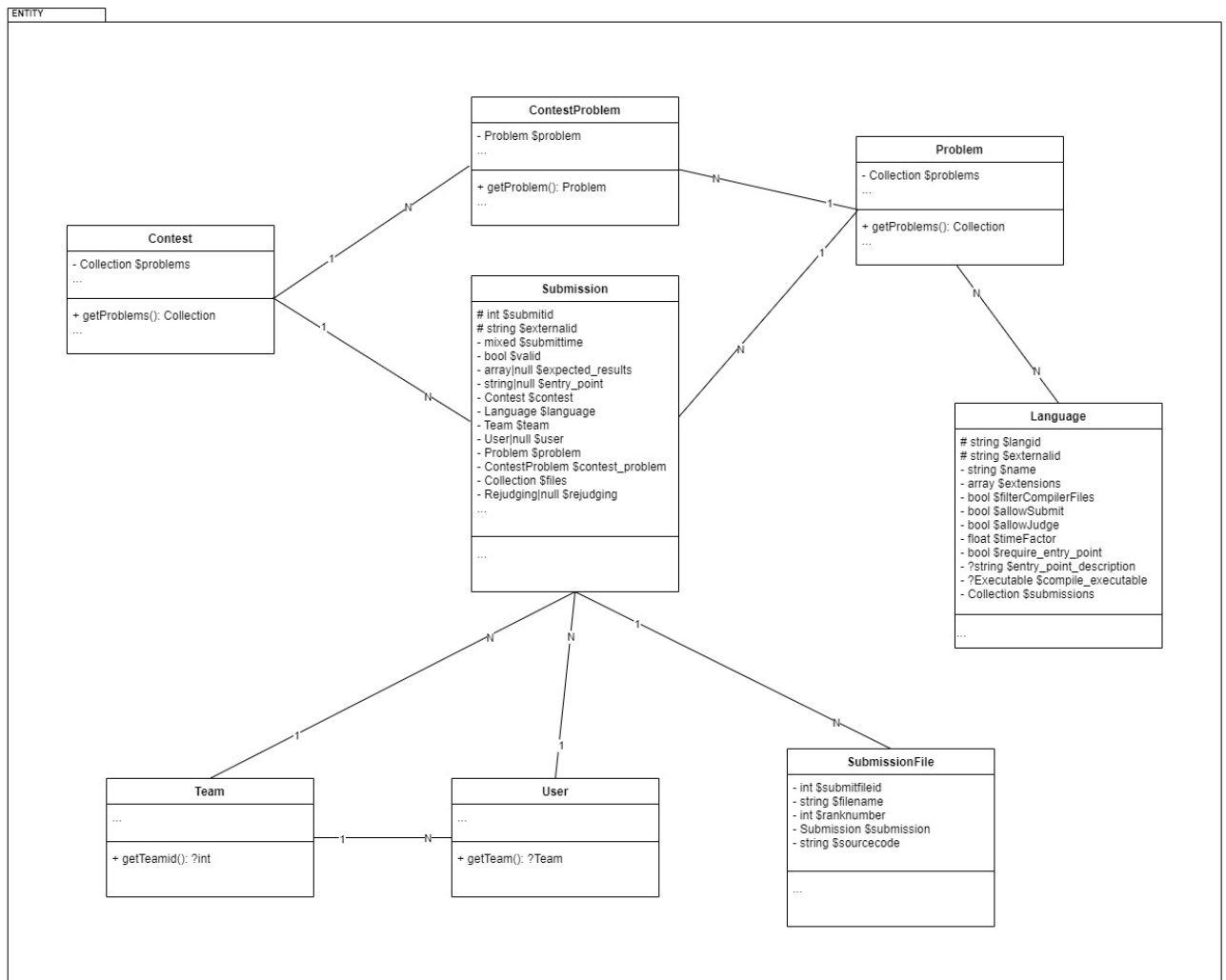
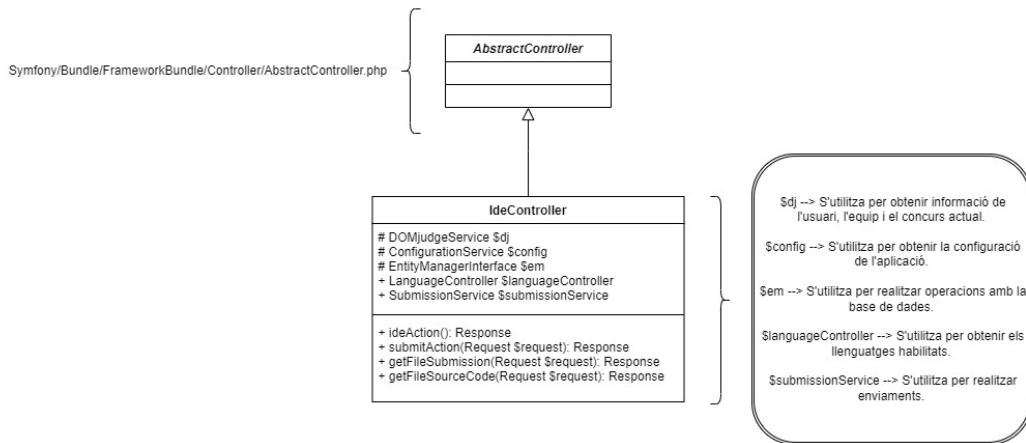
Input

1

Expected Output

1

15.4 Diagrama de classes



15.5 Diagrama de flux

