

UNIVERSITAT POLITÈCNICA DE CATALUNYA

FACULTAT D'INFORMÀTICA DE BARCELONA

GRAU EN ENGINYERIA INFORMÀTICA

Plataforma de desplegament de containers per perfils d'usuari

Autor:
Sergi Soriano Bial

Director:
Alex Pajuelo
Subdirector:
Javier Verdú



14 de juny de 2017

Resum

En aquesta memòria es presenta el desenvolupament del projecte Plataforma de desplegament de containers per perfils d'usuari. Aquest treball s'engloba en una col·laboració amb un projecte de doctorat de la UPC. Es tracta d'una plataforma per poder gestionar containers segons perfils d'usuari d'una organització i la descàrrega d'aquests. El resultat és una plataforma web per la realització de la gestió dels containers i perfils d'usuaris i un executable pel sistema operatiu Linux establert als ordinadors dels usuaris finals per poder descarregar els containers.

Resumen

En esta memoria se presenta el desarrollo del proyecto Plataforma de despliegue de containers según perfiles de usuario. Este trabajo se engloba en una colaboración con un proyecto de doctorado de la UPC. Se trata de una plataforma para poder gestionar containers según perfiles de usuario de una organización y la descarga de estos. El resultado es una plataforma web para la realización de la gestión de containers y perfiles de usuarios, y un ejecutable para el sistema operativo Linux establecido en los ordenadores de los usuarios finales para poder descargar los containers.

Abstract

This document presents the development of the project titled: "Platform for the deployment of containers based on user profiles". This work is the result of a collaboration with a PhD Project at the UPC. The project consists in the development of a platform to manage and download containers based on user profiles. The result is a two-side development: the server side, a web platform to manage the user profiles and containers. The other side is a client software that identifies the user and downloads the appropriate containers.

Dades del projecte

Títol del projecte: Plataforma de desplegament de containers per perfils d'usuari

Nom de l'estudiant: Sergi Soriano Bial

Director: Alex Pajuelo

Codirector: Xavier Verdú

Departament director: Arquitectura de computadors

Data de defensa: 26 de Juny del 2017

Titulació: Grau en Enginyeria informàtica

Especialitat: Tecnologies de la Informació

Centre d'estudis: Facultat d'Informàtica de Barcelona

Departament director: Universitat Politècnica de Catalunya

Membres del tribunal

President: Yolanda Becerra Fontal

Vocal 1: Manel Guerrero Zapata

Vocal 2: Mónica Sanchez Soler

Vocal suplent: Jorge García Vidal

Índex

Índex de figures	6
Índex de taules	7
1 Introducció i contextualització	8
1.1 Context	8
1.2 Actors implicats	9
1.3 Estat de l'art	9
1.4 Formulació del problema	9
1.5 Objectius i requisits	10
1.5.1 Objectiu Principal	10
1.5.2 Objectius secundaris	10
1.5.3 Requisits	11
1.6 Abast	11
1.7 Possibles obstacles i pla de contingència	11
2 Metodologia i rigor	12
2.1 Metodologia en el desenvolupament	12
2.2 Metodologia d'evaluació	12
2.3 Tecnologies usades en el projecte	12
2.3.1 Desenvolupament	13
2.3.2 Control de codi	13
2.3.3 Eines de seguiment	13
2.3.4 Eines de validació	13

3	Planificació inicial	14
3.1	Durada del projecte	14
3.2	Descripció de les tasques	14
3.2.1	Web	16
3.2.2	SO	16
3.3	Valoració d'alternatives i pla d'acció	16
3.4	Diagrama de Gantt	17
4	Gestió econòmica	18
4.1	Aclaracions	18
4.2	Identificació i estimació dels costos	18
4.2.1	Costos de recursos humans	18
4.2.2	Costos de hardware	19
4.2.3	Costos de software	19
4.2.4	Altres costos	19
4.2.5	Imprevistos	19
4.2.6	Contingència	20
4.2.7	Control de gestió	20
4.2.8	Pressupost final	20
5	Anàlisi de la sostenibilitat	21
5.1	Sostenibilitat econòmica	21
5.2	Sostenibilitat social	21
5.3	Sostenibilitat ambiental	22
6	Coneixements previs	23
6.1	Control de codi	23
6.2	Desenvolupament	23
6.3	Gestió i documentació	24
7	Especificació	25
7.1	Descripció del sistema	25

7.2	Actors	26
7.3	Casos d'ús	27
8	Implementació	29
8.1	Servidor	29
8.1.1	Preparació de l'entorn	29
8.1.2	Funcionament	33
8.2	Client	39
9	Canvis respecte la primera planificació	40
10	Test	43
11	Conclusions	44
11.1	Treball futur	44
11.2	Conclusió personal	44
	Bibliografia	46

Índex de figures

3.1	Diagrama de Gantt	17
7.1	Estructura del sistema	25
8.1	Exemple flux d'execució	30
8.2	Pantalla de login	31
8.3	Diagrama de classes	33
8.4	Panell de control	34
8.5	Gestió de containers	34
8.6	Formulari per afegir i editar un container	35
8.7	Gestió de perfils	36
8.8	Formulari per afegit i editar un perfil	36
8.9	Gestió dels usuaris	37
8.10	Gestió dels usuaris	37
8.11	Respostes JSON de la funció userAccess.	38
8.12	Respostes JSON de la funció userAccess.	38
9.1	Diagrama de Gantt definitiu.	42

Índex de taules

3.1	Lliurements GEP	15
4.1	Cost dels recursos humans	19
4.2	Cost del hardware	19
4.3	Pressupost total	20
5.1	Pressupost total	22
9.1	Cost del Software	40
9.2	Pressupost total	41

Capítol 1

Introducció i contextualització

1.1 Context

El següent projecte es realitza com a Treball final de grau en Enginyeria Informàtica en l'especialitat de Tecnologies de la informació. El projecte sorgeix arran d'una col·laboració amb un treball de doctorat. Aquest treball de doctorat, en poques paraules, està enfocat en tenir un únic contenidor per tal de fer servir qualsevol programa sense haver-lo d'instal·lar a cap sistema operatiu. La idea és semblant als dockers, tenir un contenidor i a més a més ens servirà per poder executar el programa a Windows i Linux. La diferència està que en lloc d'haver de tenir instal·lat Dockers a l'ordinador on es vulguin fer servir el container, en el cas del projecte de doctorat no caldrà tenir cap programa instal·lat per executar el fitxer. Un cop tenim aquesta petita introducció a continuació veurem de què tracta el meu treball de final de grau.

Un cop tenim aquests containers, que es poden fer servir als sistemes operatius esmentats, ens cal un lloc d'on poder descarregar-los. Així doncs neix la idea d'aquest projecte, crear una plataforma de desplegament de containers, però a més a més, pensant en què aquests containers estaran associats a uns perfils d'usuaris en concret. És a dir, com sabem, tota organització té diferents perfils d'usuaris que podran o necessitaran fer servir uns programes o altres, per tant és una bona opció que els containers estiguin associats a perfils d'usuari. D'aquesta forma assegurem una personalització adequada a cada usuari.

Com veurem al llarg del document, tenim dos actors principals implicats que faran ús de la plataforma. Per una banda tenim els administradors de la plataforma. Aquests s'encarreguen de gestionar tot el que té a veure amb la part administrativa dels containers i perfils d'usuaris. És a dir, seria l'actor que interactuaria amb l'aplicació web, a l'apartat de la implementació la veurem amb més detall. D'altra banda tenim els usuaris de l'organització. Aquests són els clients del projecte, els beneficiaris de la plataforma. Aquests són els qui a través d'un programa, creat també en aquest projecte, i que estarà als ordinadors de l'organització podran descarregar-se els containers, és a dir, seran els encarregats d'entrar al sistema de l'organització amb usuari i contrasenya i un cop validats podran veure el llistat de containers disponibles i podran seleccionar els que desitgin utilitzar a través de l'executable i descarregar-los. És important remarcar que en cap moment els programes s'estan instal·lant a l'ordinador. Com bé he comentat al principi, són executables que contenen tot el programa i el sistema de fitxers que necessiti fer servir per a executar-se sense tenir dependència del sistema operatiu.

1.2 Actors implicats

A part dels actors que faran servir la plataforma, tenim uns altres actors, els encarregats de dur a terme tot el projecte.

- Desenvolupador de la plataforma. Té l'objectiu d'aconseguir un bon resultat i una plataforma robusta i usable complint tots els requisits.
- Directors del projecte. El professor Alex Pajuelo ha ajudat al desenvolupador i té l'interès que la plataforma funcioni correctament i sigui funcional.
- Professorat de GEP. El professor Xavier Llinàs ha ajudat a la realització de la documentació del Treball de final de Grau.

1.3 Estat de l'art

Abans de procedir a veure com s'ha organitzat el projecte i que hi podrem trobar, cal mirar si existeix quelcom semblant actualment al mercat. Cal veure si és factible crear-ho de nou o bé si es pot aprofitar quelcom. Primer de tot hem de tenir en compte que tot el que volem fer, ha de ser Open Source, per tant ens delimita una mica més la cerca.

Fent una cerca ràpida trobem tecnologies que s'aproximen a una petita part del projecte. Per exemple, hi ha una opció de fer servir FTP (File transport protocol) [1]. És a dir, un protocol per tal d'enviar fitxers entre ordinadors amb qualsevol sistema operatiu. El problema de fer servir el protocol és que ha d'haver-hi alguna cosa que automatitzi el procés de descàrrega i saber on anar a buscar els fitxers, etc. És a dir, podria mirar de fer-ho servir a l'hora de descarregar els fitxers entre l'executable web i la plataforma.

Fent una cerca més extensa no he trobat cap possible similitud que englobi tot el sistema. Al ser una necessitat concreta, sí que poden existir alguns productes existents, però no per cobrir al 100% les necessitats i per tal de tenir un millor manteniment i poder personalitzar amb més facilitat, val la pena invertir temps a crear el sistema de 0.

És cert que hi ha molta informació sobre les tecnologies que faré servir per a desenvolupar la plataforma, fet que ajuda bastant en el procés de desenvolupament. Més endavant veurem de quines tecnologies parlo.

1.4 Formulació del problema

El repte sorgeix en haver de poder distribuir els containers dins de l'organització. Tenir una plataforma que sigui capaç de servir els containers als usuaris que ho necessitin. D'aquesta manera neix el meu projecte, desenvolupar un servidor de gestió de descàrrega de containers depenent dels perfils d'usuari. A més a més, s'ha implementat tota la gestió que té a veure amb els usuaris i permisos de cada un d'aquests per accedir als containers. A continuació veurem amb més de detall el projecte.

La idea és que els ordinadors d'una organització tinguin els discs durs amb els mínims programes instal·lats, evitant així el típic problema de programes desactualitzats o programes instal·lats que l'usuari mai acabarà fent servir. Amb aquesta plataforma oferim que l'usuari es pugui descarregar els containers que necessiti en cada moment. Actualment, els ordinadors d'una companyia

tenen instal·lats els programes a mà segons les necessitats de cada un dels usuaris que facin servir l'ordinador, molts cops s'acaben instal·lant programes que l'usuari no necessita.

Per tal de poder accedir a la plataforma de gestió dels containers, aquest projecte també inclou executables per a Linux per tal de poder-ho fer. Aquest executable permetrà veure el llistat de containers disponibles per l'usuari registrat i poder marcar els containers que es vulgui descarregar. A més a més, la plataforma necessita un manteniment (gestió de containers, perfils, etc.), per fer-ho tenim una aplicació web que és gestionada per uns administradors. Aquests poden gestionar els containers (pujar-ne de nous, pujar noves versions de containers existents al sistema, esborrar els que calguin i a més a més assignar containers a perfils). Addicionalment, també poden gestionar els perfils (crear-ne de nous, assignar usuaris als perfils, esborrar usuaris dels perfils, esborrar perfils, etc.). L'ús d'aquest sistema ens permet tenir un major control de les versions dels programes que es fan servir. A més a més, la plataforma és l'encarregada de servir el llistat de containers a l'usuari i servir els containers que els usuaris demanin. Per tant, també tindrem una disminució de feina al departament de CAU (centre d'atenció a l'usuari), ja que no caldrà que vagin ordinador per ordinador instal·lant programes.

Per tal que el projecte no comporti uns costos molt elevats en llicències, he fet servir el llenguatge PHP (*Hypertext Preprocessor*) per desenvolupar la plataforma web i la part del servidor encarregada de gestionar les descàrregues dels containers, Apache per tal de poder-ho executar i una base de dades MySQL. A més a més, els executables dels sistemes operatius estan escrits en C++.

1.5 Objectius i requisits

Un cop ja hem vist com és el projecte, cal saber quins objectius i requisits es demanaven per tal que el projecte tingués un bon resultat i complís tots els requeriments.

1.5.1 Objectiu Principal

L'objectiu principal d'aquest projecte és crear un sistema de distribució de containers segons perfils d'usuari. És a dir, s'ha de poder donar d'alta containers i associar-los a perfils d'usuaris també donats d'alta. A més a més, el sistema ha de ser capaç de descarregar els containers d'un usuari associats a un o més perfils d'usuari.

1.5.2 Objectius secundaris

A més a més, del principal objectiu, també hi ha d'altres. Aquests són:

- **Donar d'alta containers.** L'administrador ha de poder donar d'alta un container a la plataforma i a més a més ha de poder gestionar-lo, és a dir, modificar les dades d'aquest, donar-lo de baixa, poder assignar a un perfil d'usuari
- **Gestionar perfils d'usuari.** L'administrador ha de poder crear un perfil d'usuari, així com donar-lo de baixa, modificar les dades, assignar containers i assignar usuaris.
- **Automatitzar la descàrrega dels containers.** Cal un agent que pugui descarregar els containers segons perfils d'usuari a l'ordinador de l'usuari.

1.5.3 Requisits

Els requisits que es demanen per aquest projecte són clars, ha de ser una plataforma robusta, fiable i eficient.

1.6 Abast

Pel que fa a l'abast del projecte, per tal de poder crear correctament el sistema de gestió i l'executable de Linux s'ha hagut de seguir diferents passos.

El primer pas ha estat veure les necessitats que tenen les organitzacions i usuaris d'aquestes per tal de fer una aplicació d'àmbit general, útil per les organitzacions que vulguin implementar la plataforma. Un cop tenim els requeriments, s'havia de veure com implementar-los. A més a més a part d'això era necessari posar a punt l'entorn de treball (servidor Apache, base de dades, sistema de fitxers, etc.).

Finalment, un cop he tingut l'entorn de treball creat i amb els requeriments ben clars, només quedava desenvolupar la plataforma. Al següent capítol veurem la metodologia que he fet servir per a dur a terme a temps la creació de la plataforma. L'objectiu final era tenir l'aplicació web funcionant, amb un sistema de gestió de perfils i containers correcte que permetin la descàrrega dels containers demanats i tenir l'executable per Linux per tal que els usuaris puguin seleccionar i descarregar els containers desitjats.

1.7 Possibles obstacles i pla de contingència

En aquest apartat es descriuen els obstacles que s'han hagut de superar en el desenvolupament d'aquest projecte, i com ha ajudat el pla de contingència.

- Un dels possibles obstacles era crear els executables per Windows i per Linux. Com veurem al capítol "Canvis respecte a la primera planificació", per temps no s'ha pogut crear l'executable per Windows però sí per Linux. El meu coneixement en aquest terreny no era gaire extens i era possible que el desenvolupament d'aquests executables hem portés més temps de l'esperat. Cal dir que al fet que no calia que fossin amb interfície gràfica i amb l'ajut del meu director ha estat més senzill el seu desenvolupament.
- Adaptació a MVC (Model-Vista-Controlador) i orientació a objectes en aplicacions web. Tinc alguna experiència en fer pàgines web i alguna aplicació però sense fer servir aquests frameworks. M'ha portat una mica més de temps la corba d'aprenentatge respecte algú que pugui tenir més experiència, però no ha suposat cap problema en el desenvolupament del projecte, ja que s'havia tingut en compte a l'hora de fer la definició.

Capítol 2

Metodologia i rigor

Per tal de dur a terme tot un projecte cal ser ordenat i meticulós amb el procediment. Calia estudiar bé els casos d'ús i desenvolupar-los de forma correcta. A més a més, s'han realitzat reunions amb el professor Alex Pajuelo per tal d'anar veient l'evolució i progrés de la plataforma. Aquestes reunions han servit per ensenyar l'evolució del projecte així com per resoldre dubtes que he tingut i que em pogués donar suggerències per a on seguir o com fer-ho en certs punts del projecte. A més a més, han servit per anar perfilant, és a dir, s'han afegit o modificat funcionalitats per adaptar més la plataforma. A continuació veurem quina metodologia i tecnologies he fet servir al llarg del desenvolupament del projecte.

2.1 Metodologia en el desenvolupament

Com a metodologia d'aquest projecte s'ha triat fer servir una metodologia agile com és Scrum. Hi hagut varis sprints d'uns 15 dies, definint bé les tasques a realitzar a cada un dels sprints. La idea era poder anar mostrant la plataforma amb casos funcionals i anar arreglant els petits bugs o coses a millorar per les següents iteracions.

2.2 Metodologia d'evaluació

Com que he triat Scrum, em serveix també per poder anar rectificant la plataforma a mesura que s'anava evolucionant el projecte. Per altra banda gràcies a l'assignatura de GEP, s'ha anat entregant documentació, rebent suggerències del professor Xavier Llinàs.

2.3 Tecnologies usades en el projecte

En aquest projecte podrem trobar diferents tecnologies usades. A continuació veurem cadascuna d'elles.

2.3.1 Desenvolupament

La tecnologia emprada pel funcionament de la plataforma ha estat una combinació d'Apache [2] i MySQL [3]. Apache és un servidor HTTP (*Hypertext Transfer Protocol*) que ens servirà tenir l'aplicació web i la plataforma encarregada de servir els containers. A més a més MySQL que és un sistema de gestió de bases de dades relacionals i open source, m'ha servit per guardar les dades de la plataforma. Com ja hem vist anteriorment s'ha fet servir PHP pel desenvolupament de la pàgina web i la plataforma. A més a més, també he fet servir el llenguatge C++ per fer l'executable que faran servir els usuaris per descarregar-se els containers, a l'apartat d'implementació veurem com funciona.

Pel desenvolupament necessito XAMPP, que és una eina que ens permet crear un servidor Apache, té una base de dades MySQL i ens permet poder treballar amb PHP. Aquesta eina ens permet amb una sola instal·lació tenir tot l'entorn llest per començar a fer servir.

El fet d'haver triat aquestes tecnologies per desenvolupar el projecte és perquè són Open Source, a més a més tinc experiència amb gairebé totes les tecnologies i hi ha una gran facilitat a poder trobar suport tant a la xarxa o com per part del professorat perquè són tecnologies molt madures, és a dir, fa molts anys es fan servir.

2.3.2 Control de codi

Per tal de tenir un control de versions de codi m'he decantat a fer servir Git [4]. Per tal de poder tenir el codi compartit entre els dos equips de desenvolupament he decidit fer servir BitBucket, que és una pàgina web per emmagatzemar el codi així com totes les versions. A més a més, per tenir un control més clar i fàcil de fer servir, he decidit fer servir Git Kraken, un GUI per facilitar la feina de branques de Git.

2.3.3 Eines de seguiment

Per tal de poder tenir un seguiment i anar veient com evolucionen les tasques he utilitzat Trello [5], una eina d'organització de projectes. És una pàgina web on tu pots definir, per cada projecte, un conjunt de columnes que en el meu cas han estat els estats possibles de les tasques, per exemple: Per definir, per fer, en procés, per testejar, testejant i acabades. A més a més, es poden crear tantes tasques com tingui el teu projecte i les podràs anar movent per les diferents columnes segons l'estat en què estigui la tasca. Així doncs aquesta eina ens permet controlar l'estat de les tasques, ja sigui les pendents, les que estan en curs, les que ja s'han fet, les que estan per provar, etc. Permet tenir una visió global de l'estat del projecte.

2.3.4 Eines de validació

Com hem vist, per tal d'anar veient si s'estava duent a terme correctament el projecte, s'han anat fent reunions amb el director del projecte per tal de rebre feedback sobre l'evolució del projecte i poder fer els canvis que calguessin fer.

Capítol 3

Planificació inicial

En aquest apartat veurem les tasques que s'han de realitzar al llarg del projecte. Les veurem de forma una mica desglossada podent veure així la càrrega de treball que tenen, veient finalment com s'organitzaran en el diagrama de Gantt.

3.1 Durada del projecte

El projecte té una durada de cinc mesos, el mateix que dura el quadrimestre actual. Durant tres d'aquests cinc mesos s'han pensat per desenvolupar el projecte. En aquest període de temps es desenvoluparà tota la plataforma web i els executables. En els dos mesos restants són per les tasques del curs de GEP i tota la documentació necessària entre d'altres.

3.2 Descripció de les tasques

Per tal de tenir una previsió de temps i tasques que s'han de realitzar, m'he creat un diagrama de Gantt per tal de tenir una visió global del projecte, pel que fa a termes de planificació i tasques a realitzar. Incloent totes les tasques relacionades amb GEP i el desenvolupament del projecte (desenvolupament, documentació, recerca, etc.). A continuació veurem amb més detall quines tasques s'han cregut necessàries per a realitzar a cada bloc i quant temps ocupen.

Primerament veiem l'apartat de GEP, que ha consistit en la realització d'uns lliuraments setmanals de documentació relacionada amb el projecte, ajudant a encaminar la documentació final del projecte i ajudant a avançar-la. A més a més, ajuda a planificar de forma correcta el projecte i haver de pensar i organitzar les tasques a realitzar i la forma de dur-les a terme.

Tanmateix, hi hagut lliuraments que han ajudat més a avançar documentació necessària i pensar en quins recursos, materialment parlant, es necessitaven per tal de poder dur a terme el projecte de forma satisfactòria.

El conjunt de lliuraments han comportat una càrrega de 74 hores, aquestes es dividien de la següent manera:

Lliurement	Data entrega	Hores
Definició de l'abast	28/02/2017	25 h
Planificació temporal	06/03/2017	8,25 h
Gestió Econòmica i sostenibilitat	13/03/2017	9,25 h
Primera presentació oral	20/03/2017	6,25 h
Definició de l'abast	27/03/2017	18,25 h
Presentació final i documentació final	27/03/2017	7 h

Taula 3.1: Lliurements GEP

Per altra banda tenim tot el contingut relacionat amb el projecte i el seu desenvolupament. Cal comentar que a cada tasca de desenvolupament s'ha pensat a tenir diverses parts. Aquestes són: anàlisi, desenvolupament, test i documentació de la tasca desenvolupada. A més a més, un cop realitzades les tasques, s'ha reservat una jornada de test de tota la plataforma per veure que tot funcionava de forma correcta i acabar la documentació. A continuació veurem amb detall com invertir les hores en les parts de cada tasca.

- **Anàlisi** - Aquesta part consisteix a fer una anàlisi de les dependències que té la tasca, tenir clars els casos d'ús que havia de tenir i finalment fer un esbós de com ha de ser l'arquitectura o la interfície (si aplica).
- **Desenvolupament** - Aquesta part és on es dedica el temps a desenvolupar tots els casos d'ús pensats en la fase anterior i fer un mínim test per tal de comprovar que tot funciona.
- **Test** - Aquesta fase està dedicada a realitzar un test exhaustiu de la part desenvolupada per tal de trobar tots els bugs i millores a realitzar. Totes les incidències que es vagin trobant, si són errors senzills i ràpids d'arreglar es fa al moment i si no es poden fer al moment, s'apunten i es realitzen en un altre sprint.
- **Documentació** - Un cop realitzat el test i aplicat les millores que pertoquessin s'ha escrit una documentació de la tasca realitzada.

Cal dir que la gran major part del temps està pensat a dedicar-lo a la part de desenvolupament i test però també cal dir que dependent de la tasca a realitzar, podria ser que no es compleixi, ja que hi poden haver tasques a les quals s'hi dediqui més estona a analitzar el cas d'ús que a desenvolupar i testejar.

A part de les tasques de desenvolupament també he cregut necessari afegir tasques a realitzar en el projecte com poden ser hores dedicades a reunions i a la preparació de l'entorn.

Per una banda, com hem vist anteriorment, s'han plantejat unes hores per fer reunions amb el director del projecte per tal d'anar veient l'evolució del projecte i poder anar solucionant dubtes i canviar la funcionalitat per millorar el projecte. A més a més de les reunions també es té en compte les hores fent servir el correu electronic, ja que alguns cops és més ràpid solucionar els dubtes i rebre feedback d'aquesta manera. En total estimo unes 10 hores.

Per l'altra banda, he estimat unes 9 hores a la preparació de l'entorn. Cal dir que la preparació és per l'entorn en l'equip de sobretaula i al portàtil. He contemplat tenir diversos entorns de desenvolupament.

Un cop hem vist les dues tasques que no pertanyen al desenvolupament en si del projecte, procedim a veure les tasques on he centrat més hores. Tal com hem vist al principi, el projecte consisteix a crear una plataforma en un servidor i un executable per a Linux. Per tal de simplificar la planificació he dividit en dos grans grups tot el gruix de desenvolupament. Aquests dos grups els són la part web i la part de SO (sistemes operatius). Per tal de realitzar la part del sistema

operatiu és necessari que la base de la plataforma estigui desenvolupada igual que l'API. Tenint aquest pre-requisit he decidit desenvolupar tota la part web primer, amb l'API inclosa, ja que l'agent la necessita per recuperar la informació de la plataforma, i després la part del sistema operatiu per així poder comprovar que el funcionament de l'agent és correcte.

3.2.1 Web

Aquesta part engloba la gestió de containers, gestió de perfils, gestió d'usuaris, assignació de perfils, etc. (tot ho relacionat amb la web que he fet per tal de gestionar la plataforma) i a més a més també he inclòs en aquesta part, tota la gestió de descàrrega de containers, accessos al sistema de fitxers, etc. Donat que tenia una mica d'experiència en fer aplicacions web, he pensat que aniré més de ràpid desenvolupant aquesta part que no la part del SO. He calculat estar en la part web unes 180 hores per fer tot ho esmentat anteriorment.

3.2.2 SO

Finalment queda per explicar la part de l'agent del sistema operatiu. Aquesta part és on dedicaré més hores en realitzar-la. Considero que no tinc unes nocions molt elevades en creació d'executables de sistemes operatius i he cregut necessari dedicar gran part del meu temps per tal de realitzar aquesta part. Gràcies al meu director de projecte, Alex Pajuelo, ha estat més senzill i més amè de realitzar aquesta part, ja que s'ha ofert ajudar en els dubtes que he anat tenint. Dedicaré unes 270 hores a realitzar aquesta part del projecte.

3.3 Valoració d'alternatives i pla d'acció

Al treballar amb una metodologia Agile, si en un sprint no he pogut acabar les tasques, s'han posat aquestes a fer al següent sprint, fent que aquest darrer s'hagi de treballar més intensament per tal d'acabar totes les tasques. D'altra banda, si en un sprint acabava abans del previst amb les tasques, començava a treballar les del següent sprint. A més a més, en l'estimació d'hores s'ha tingut en compte un marge d'hores, una tercera part del temps estimat, per errors, fent que hi tingui un coixí de temps a cada fase. Per tant, el càlcul d'hores s'ha fet una mica a l'alça per tal tenir una mica de marge per imprevistos.

3.4 Diagrama de Gantt

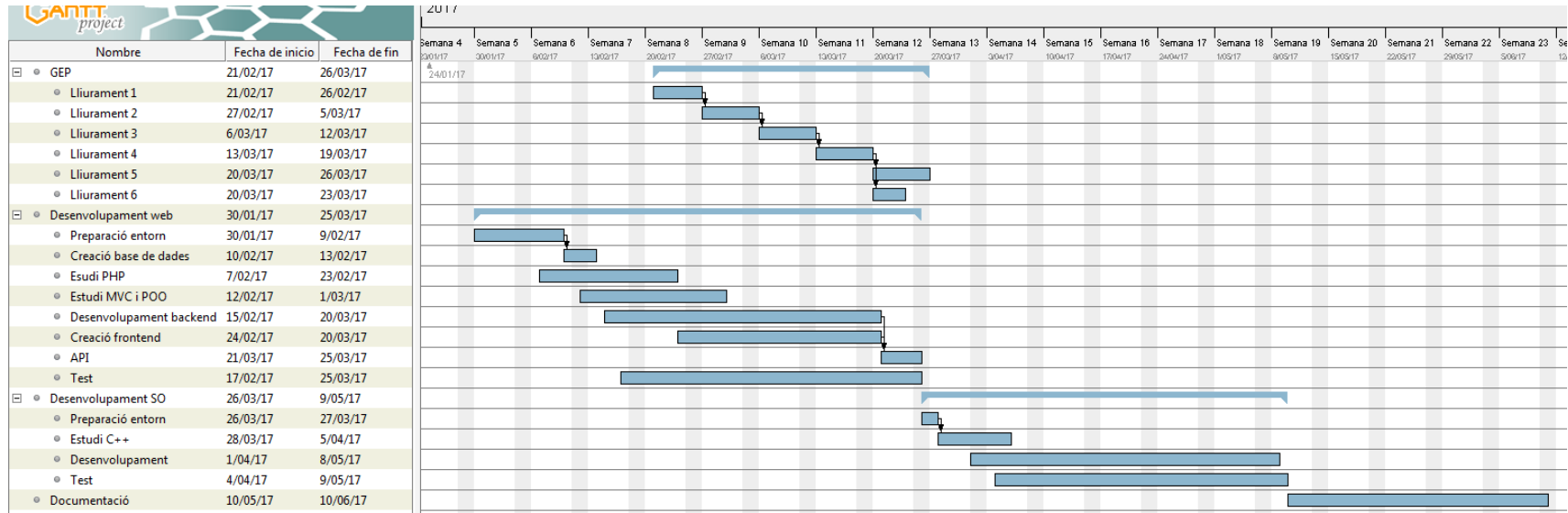


Figura 3.1: Diagrama de Gantt

Capítol 4

Gestió econòmica

Per tal de saber si el projecte és viable o no cal fer un estudi econòmic i d'impacte ambiental. Així doncs començarem fent un càlcul dels costos que tindrem en la realització del projecte.

4.1 Aclaracions

Abans de continuar cal tenir en compte certs detalls. Primerament cal tenir en compte que els costos d'activitat relacionats amb la producció no es tindran en compte donat que el projecte no produeix res, sinó que és un servei software.

4.2 Identificació i estimació dels costos

Per tal de poder fer una estimació correcta dels costos que tindrem cal identificar-los. Els costos seran humans, hardware, software i d'altres. A més a més, en l'estimació de costos tindrem en compte el pla de contingència i possibles costos provocats per imprevistos.

4.2.1 Costos de recursos humans

El projecte es dut a terme per una única persona així que serà fàcil de calcular el cost final, així doncs, totes les tasques les durà a terme aquesta persona (desenvolupament, test, documentació, etc.). Donat que aquesta persona encara és estudiant, aplicarem el mateix preu que la Universitat Politècnica de Catalunya obliga a pagar un mínim a les empreses als estudiants que estiguin treballant en pràctiques, 8 €/hora.

Les hores de treball per aquesta persona s'han aconseguit sumant les hores planificades en el diagrama de Gantt. El càlcul es realitza d'aquesta forma donat que l'única persona encarregada de realitzar el projecte realitzarà totes les tasques (tasques de desenvolupament, reunions i documentació incloses).

Rol	Sou	Tasques	Hores	Total
Desenvolupador	8 €/hora	GEP	74 h	592 €
		Preparació entorn	9 h	72 €
		Reunions	10 h	80 €
		Desenvolupament web	180 h	1.440 €
		Desenvolupament executable	270 h	2.160 €
Total			543 h	4.344 €

Taula 4.1: Cost dels recursos humans

4.2.2 Costos de hardware

Pel que fa als costos associats al hardware, en aquest projecte només es fa servir un equip de sobretaula i un portàtil MacBook. El projecte s'allotjarà a l'ordinador de sobretaula així que pel desenvolupament no cal tenir cap servidor. Però el projecte acabarà funcionant sobre un servidor. Tenint en compte que les companyies tenen servidors, es podria fer ús d'aquests per poder instal·lar la plataforma. He tingut en compte que es faria servir un 15% d'aquest servidor tenint en compte que el servidor té altres usos.

L'amortització està calculada en base la vida útil dels equips i el percentatge d'ús d'aquests en aquest projecte. Aquests costos s'identifiquen com a costos indirectes.

Equip	Vida útil	Preu	Amortització	Cost estimat
Ordinador de sobretaula	5 anys	600 €	70 %	84 €
Portàtil Macbook	4 anys	1.300 €	30 %	97,5 €
Servidor	4 anys	1.000 €	15% %	37,5 €
Total		2.900 €		219 €

Taula 4.2: Cost del hardware

4.2.3 Costos de software

Donat que l'equip de sobretaula fa ús del sistema operatiu Linux, i el portàtil fa ús del Mac OSX, no haurem de pagar cap llicència per mantenir sistemes operatius. A més a més, com hem vist anteriorment, tots els programes fets servir pel desenvolupament i creació de documentació són gratuïts. Per tant al tenir un cost 0 no tindrem en compte cap mena d'amortització donat que no és rellevant. Aquests costos s'identifiquen com a costos indirectes.

4.2.4 Altres costos

A part dels costos de software i hardware hem de tenir en compte altre costos com pot ser les impressions del projecte. Tenint en compte que hi ha tres membres del tribunal i el cost per pàgina és de 0,05 €i una extensió aproximada del projecte de 100 pàgines i contant l'enquadernació de les còpies, el cost aproximat serà d'uns 20 €.

4.2.5 Imprevistos

Com hem vist anteriorment, a les tasques de desenvolupament he deixat un marge de temps per imprevistos, fent que el càlcul d'hores invertides per a imprevistos ja s'hagi tingut en compte en el recompte d'hores de les tasques.

4.2.6 Contingència

He decidit dedicar una part del pressupost a un pla de contingència per si sorgís algun entrebanc. Aquest fet implicarà que el pressupost augmenti. He calculat que destinaré un 10 % del cost total del projecte.

4.2.7 Control de gestió

La part de recursos humans és la que podria variar al llarg de tot el projecte tot i tenir en compte el temps destinat a imprevistos. Tot i així, caldrà fer un control constant per evitar un increment considerable en el pressupost. Aquest control consistirà a anar comparant la data de finalització prevista de les tasques amb la data de finalització real.

A més a més, es mirarà el cost calculat inicialment segons les hores planificades amb el cost de les hores que s'ha treballat realment. Les fases que més poden variar el pressupost inicial amb les seves possibles desviacions són les de desenvolupament. Per realitzar aquest control, es disposa d'una taula Excel en la qual cada dia s'apuntaran les hores dedicades.

4.2.8 Pressupost final

Concepte	Cost
Recursos humans	4.344 €
Recursos hardware	219 €
Recursos software	0 €
Recursos contingència	456,3 €
Altres	20 €
Total	5.039,3 €

Taula 4.3: Pressupost total

Capítol 5

Anàlisi de la sostenibilitat

A continuació veurem detalladament l'impacte del projecte en les diferents modalitats.

5.1 Sostenibilitat econòmica

Per saber si un projecte és viable econòmicament parlant, és necessari fer una avaluació de costos, vist anteriorment.

Tenint la avaluació feta veiem que més d'un 85% del pressupost final són els recursos humans emprats per realitzar el projecte. La resta són despeses indirectes inevitables. A més a més veiem que el projecte és prou viable donat el seu baix cost de desenvolupament.

A més a més, aquest projecte és competitiu donat que la no implementació d'aquest implica un cost molt elevat a la companyia. És a dir, hem de tenir en compte que una companyia més o menys gran pot tenir uns 300 empleats i es tarda entre dues i tres hores en instal·lar tots els programes necessaris per ordinador. Donat aquests valors i que una persona del CAU (centre d'atenció a l'usuari) de la companyia pot cobrar uns 12€/hora, tenim que a la companyia, sense contar el preu de l'electricitat, li costa uns 7.200€. Com podem veure és un cost molt elevat, tenint en compte que amb el nostre sistema no caldria fer cap instal·lació.

Cal comentar que el temps dedicat a cada tasca és proporcional a la seva importància i a més a més tenint en compte les mancances formatives. A més a més, aquest projecte forma part d'una col·laboració amb un projecte de doctorat.

5.2 Sostenibilitat social

Actualment la política del país està estable pel que fa a l'estat del govern. Hi ha disputes entre governs (central i català) però podem dir que està estable la situació. Aquest projecte està en el món de la informàtica més en concret en l'àmbit de la gestió de software i gràcies a aquest projecte millorarà i ajudarà al treball de doctorat que es realitza a la Universitat politècnica de Catalunya (UPC). Cal dir que aquest projecte no millora la situació política i social del país directament, però si millorarà l'usuari que hagi de fer servir la plataforma. A més a més, cap col·lectiu es veurà afectat.

5.3 Sostenibilitat ambiental

Al llarg del projecte es necessitaren en tot moment els dos equips (el portàtil i l'ordinador de sobretaula), fent un ús compaginat entre ells. Un cop la plataforma estigui més sòlida es posarà al servidor per fer petites proves.

Els equips fets servir durant el procés de desenvolupament genera aproximadament 1 kg de CO₂. Com podem veure és gairebé menyspreable el consum que té, tot i així cal tenir-ho en compte. Durant la seva posada en marxa i vida útil del projecte el consum dependrà de la quantitat d'usuaris de la companyia i quina quantitat de containers es descarreguin i amb quina freqüència. Amb tota la casuística és complicat fer una estimació però tenint en compte que el servidor s'aprofitarà per altres usos, podríem dir que depèn del ús que se'n fagi del projecte també serà negligible.

A més a més, el projecte ajudaria a reduir el consum i impacte ambiental donat que com hem vist anteriorment estalviaríem moltes hores de tenir els ordinadors encesos per posar-los a punt.

A part per tal de dur a terme el projecte i la posada en marxa d'aquest, s'han gastat recursos materials que han consumit recursos en la seva construcció i funcionaran amb energia elèctrica. A més a més, aquests recursos es reaprofitaran fins a la fi de la seva vida útil. Tant el portàtil com l'equip de sobretaula utilitzats per realitzar el projecte es seguiran aprofitant per altres usos. I el servidor on vagi el sistema un cop estigui acabat també s'aprofitarà per altres usos de la companyia.

Tenint en compte que tot el material utilitzat majoritàriament és informàtic, el procés de desmantellament anirà a carreg dels centres de reciclatge homologats i certificats. Al ser un producte software no ha calgut tenir en compte el procés de producció per tal de poder-ho reciclar més fàcilment.

Finalment cal comentar que al ser una aplicació de codi obert es podrà reaprofitar parcialment o totalment per a futures modificacions.

	PPP	Vida útil	Riscs
Medi ambient	8	7	-5
Econòmia	6	12	-5
Social	8	10	0
Rang sostenibilitat	22	29	-10
Total		41	

Taula 5.1: Pressupost total

Capítol 6

Coneixements previs

En aquest capítol explicaré quines tecnologies i eines he fet servir per desenvolupar aquest projecte.

6.1 Control de codi

En aquest projecte he fet servir la tecnologia Git per tal de tenir un control de versions. Git ens serveix per a tenir un historial de tots els canvis fets en el codi i poder tirar enrere si fos necessari. A més a més, ens permet crear branques i poder treballar una mateixa implementació de formes diferents sense xafar-se entre elles.

Per altra banda tenim a Bitbucket. És un servei d'allotjament basat en web pels projectes que utilitzen un sistema de control de versions com és Git. M'ha servit bastant donat que al treballar amb dos equips diferents és difícil mantenir el codi per igual però gràcies a tenir el codi a Bitbucket m'ha ajudat a poder treballar tranquil·lament amb els dos equips sense haver de pensar on havia fet les últimes modificacions.

A més a més per simplificar-me la feina he treballat amb GitKraken. És un client de Git per Windows, Mac i Linux. A l'hora de gestionar tot Git local amb el Git de Bitbucket m'ha ajudat bastant donat que és molt visual i fàcil de fer servir. A més a més al començar a desenvolupar el projecte era la segona vegada que feia servir un control de versions i no tenia molta experiència en aquest camp i em van recomanar GitKraken per facilitar-me la feina.

6.2 Desenvolupament

Per fer tota la part del codi de la plataforma he fet servir diverses eines, un dels motius és perquè treballava amb dos equips amb sistemes operatius diferents. Per una banda a l'equip de sobretaula he fet servir Netbeans per desenvolupar la part de la plataforma web. Ja hi havia treballat amb aquest IDE així que ja el tenia més per la mà. A més a més per tal de realitzar les proves he treballat al principi amb XAMPP. És un programa que ens permet tenir Apache i MySQL en un sistema operatiu Windows. Per poder administrar la base de dades faré servir HeidiSQL. És un programa que ajuda a crear les taules necessaries, poder comprobar i canviar dades si calgués, etc. A més a més, quan ja vaig tenir una versió prou estable de la plataforma web vaig instal·lar VirtualBox, una aplicació gratuïta per poder crear màquines virtuals, que m'ha permès poder provar la plataforma web com si fos una aplicació distribuïda. A la màquina virtual, més endavant ho veurem, vaig posar un sistema operatiu Linux i Apache.

Per una altra banda també treballava amb l'equip portàtil. Aquest al portar un sistema operatiu Mac OS vaig decidir treballar amb Sublime Text. Aquest programa és un editor de text que m'ha permès treballar tant en la plataforma web com amb l'agent, ja que permet treballar amb diferents llenguatges de programació. A més a més, per fer les proves també em vaig instal·lar el VirtualBox a l'equip portàtil. També he fet servir el terminal i el compilador G++ per compilar l'executable, l'agent.

Pel que fa als llenguatges que he utilitzat han estat varis, ja que he hagut de desenvolupar per diferents plataformes. Per una banda tenim el PHP com hem vist. És un llenguatge de codi obert que s'utilitza, entre d'altres, per crear aplicacions web i ens permet incrustar codi HTML. Més endavant ho veurem amb més detall però farem servir el PHP per mostrar per pantalla la informació que tenim guardada a la base de dades i a més a més tot el gruix de la gestió de la plataforma. Per fer les consultes a la base de dades he fet servir MySQL. Per fer les consultes a la base de dades he fet servir PDO que és una extensió que ens proporciona una capa d'abstracció d'accés a les dades. En el cas que en algun moment s'hagues de canviar la tecnologia de la base de dades a una altra, aquesta capa d'abstracció amb un simple canvi no caldrà canviar molt codi. A més a més, juntament amb el PHP he fet servir HTML, CSS i jQuery (un framework de JavaScript) per fer la part visual de la web. El CSS ens servirà per als estils de la pàgina i jQuery per fer la web dinàmica. El que hem vist fins ara m'ha servit per fer la part web del projecte. Per una altra banda com hem vist abans farem servir C++ per fer la part de l'agent. C++ és un llenguatge que pot treballar amb programació orientada a objectes. He triat aquest programa perquè ja que tenia experiència amb ell i no m'ha costat recordar a programar amb ell (ja que feia molt temps que no el feia servir).

6.3 Gestió i documentació

Finalment veurem les eines que he fet servir per a organitzar el projecte i realitzar la documentació. Per una banda, per organitzar el projecte he fet servir diverses eines. Trello és una eina que ens servirà per tenir un control dels casos d'ús que cal fer i tenir un desglossament de cada tasca a realitzar. Juntament amb un full de càlcul ens ha servit per tenir un control de les hores invertides a cadascuna de les tasques i veure si anàvem bé durant el sprint. Per una altra banda he fet servir l'e-mail per posar-me en contacte amb el director del projecte per concretar reunions per veure l'evolució del projecte i resoldre dubtes.

Per últim, per fer tota la documentació pertinent del projecte, he decidit fer servir $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ i per no haver de dependre de cap ordinador en especial he decidit aprofitar la plataforma Sharelatex que és una plataforma per tal de crear tota la documentació en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ i accessible des de qualsevol ordinador, ja que és online. Ja havia fet documentació en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ així que no m'ha estat una gran dificultat realitzar-la.

Capítol 7

Especificació

7.1 Descripció del sistema

A la imatge 7.1 es pot veure com és l'estructura en què es basa el nostre sistema, on tenim un servidor web, una base de dades, un repositori i un agent. Anirem un per un veient quines característiques tenen i quins serveix ofereix cada un d'ells.

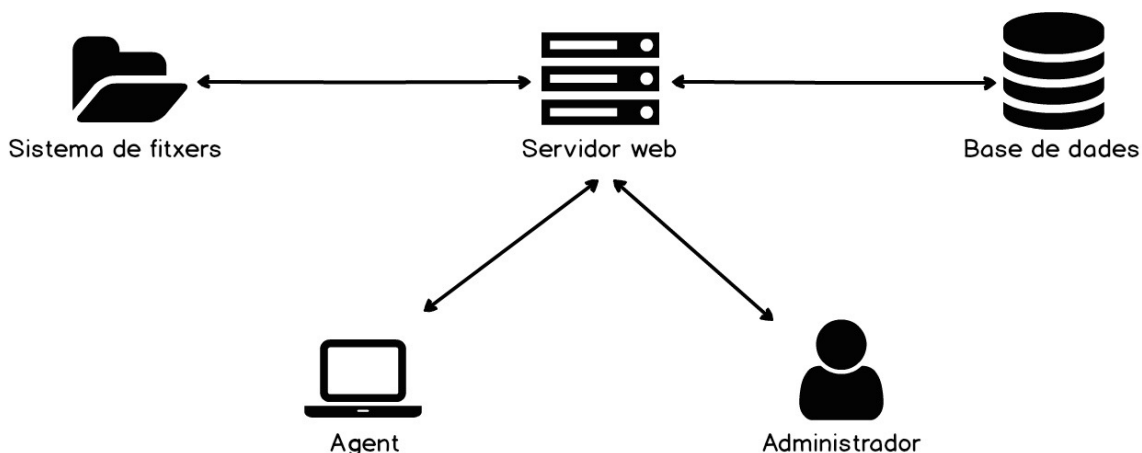


Figura 7.1: Estructura del sistema

Per una banda tenim el servidor web on estarà la plataforma de gestió de containers. Aquesta plataforma s'ha desenvolupat en PHP la part de servidor i HTML, CSS i jQuery per la part visual de la plataforma. Per la part visual he aprofitat un framework anomenat Bootstrap que ha fet més senzill el desenvolupament.

Per poder fer servir la plataforma, al fer servir PHP, necessitarem un servidor HTTP capaç d'executar aquest llenguatge. En aquest cas m'he decantat per Apache perquè és un dels més populars, usats i amb un ampli suport a les comunitats d'Internet. A més a més, com hem vist al llarg del document, el que busquem és minimitzar costos i Apache és gratuït i de codi obert, pel que es posiciona com un gran candidat.

Un cop vist com ha de ser el servidor web passem a veure com ha de ser el lloc on es guarden la informació de la plataforma. Aquesta és la informació sobre els containers (nom, versió, ruta on

tenim guardat el container i el seu MD5), dels perfils (nom del perfil, descripció i els containers assignats al perfil) i dels usuaris (nom, nom d'usuari i els perfils assignats). La solució era muntar un base de dades on poder guardar-ho tot. Vam estar donant-li voltes si calia que aquesta fos una noSql però finalment juntament amb el director del projecte vam decidir fer una base de dades relacional MySQL. A més a més, un MySQL és gratuït i ens serveix per guardar tota la informació així que m'acabo decantant per aquesta opció. Més endavant veurem com està estructurada.

Ja hem vist els sistemes que contindran la plataforma i on es guardarà la informació, ara necessitem veure on es guarden els containers.

Per tal de guardar els containers he decidit guardar-los directament en un directori d'un sistema de fitxers el qual estigui habilitat per poder-ho fer quan l'administrador creï un container a la plataforma i introdueixi un fitxer container. La plataforma haurà de tenir permisos de lectura i escriptura en aquest directori per tal guardar i descarregar els containers quan sigui necessari. Més endavant veurem en detall com es realitzen aquestes dues accions.

Arribats a aquest punt ja hem vist com han de ser els sistemes que conformen tota la part del que és la plataforma de gestió de containers en si. Si no existeixen aquests tres elements, no es podrà fer servir la plataforma creada. Ara bé, necessitarem accedir-hi d'alguna manera per poder-la aprofitar i fer-la servir. Aquí és intervé el últim element de l'arquitectura del nostre sistema, l'agent.

Aquest agent és l'executable que hem anat veiem al llarg del document. És un programa que estarà a l'ordinador de l'usuari final. La idea és que en executar l'agent es puguin arribar a descarregar containers que l'usuari tingui assignats. Per tal que es pugui fer servir en diferents sistemes operatius s'ha desenvolupat en C++. Aquest agent a través d'una API creada a la plataforma web, obtindrà la informació relativa a l'usuari de la plataforma. Més endavant ho veurem amb més profunditat.

7.2 Actors

Un cop ja tenim clara com és l'estructura del sistema cal veure qui farà servir la plataforma i com. Tindrem dos actors principals, l'administrador de la plataforma i l'usuari final. A continuació veurem quin rol desenvolupen a la plataforma cadascun d'ells.

- **Administrador.** Aquest actor és l'encarregat de mantenir la plataforma, és a dir, és la persona que s'encarrega de crear i administrar els perfils d'usuaris, crear, administrar i pujar al sistema els containers disponibles per a l'organització, i crear i mantenir els usuaris que faran ús de la plataforma. Aquest actor està pensat que sigui una persona del departament del centre d'atenció a l'usuari (CAU) del departament de IT. L'administrador ha de ser algú del departament de IT perquè són qui coneixen els programes que fan servir els usuaris i les necessitats d'aquests. A més a més, seran els encarregats de mantenir-los actualitzats.
- **Usuari final.** Finalment veiem el segon i últim actor del nostre sistema. Aquest cas és el qui anomenem usuari final. Aquest és qui acabarà fent servir la plataforma, és a dir, és l'usuari de l'organització que tindrà l'executable al seu sistema operatiu i es triarà quins containers vol descarregar-se al seu ordinador. El perfil d'aquest actor pot ser qualsevol, ja que involucra a cada un dels membres de l'organització.

7.3 Casos d'ús

A continuació veurem els casos d'ús de tot el sistema. Els veurem de forma resumida, ja que més endavant veurem la funcionalitat de cadascun a mesura que vagi explicant com he implementat la plataforma. A més a més comentar que tots els casos d'ús de l'actor administrador són a la plataforma web d'altra banda els casos d'ús de l'usuari final tenen a veure amb l'agent.

- **Cas d'ús:** Login
Actor: Administrador
Descripció: L'administrador vol accedir a la plataforma.
- **Cas d'ús:** Veure tots els containers i la seva informació
Actor: Administrador
Descripció: L'administrador vol veure tots els containers donats d'alta a la plataforma així com la informació bàsica d'aquests.
- **Cas d'ús:** Donar d'alta un container
Actor: Administrador
Descripció: L'administrador vol donar d'alta un nou container a la plataforma indicant la versió, nom i el fitxer container.
- **Cas d'ús:** Editar un container
Actor: Administrador
Descripció: L'administrador ha de poder editar un container que ja existeix a la plataforma.
- **Cas d'ús:** Eliminar un container
Actor: Administrador
Descripció: L'administrador ha de poder eliminar un container existent i tota la informació relacionada amb ell.
- **Cas d'ús:** Veure tots els perfils i la seva informació
Actor: Administrador
Descripció: L'administrador vol veure tots els perfils donats d'alta a la plataforma així com la informació bàsica d'aquests.
- **Cas d'ús:** Donar d'alta un perfil
Actor: Administrador
Descripció: L'administrador vol donar d'alta un nou perfil a la plataforma per relacionar-lo amb els usuaris.
- **Cas d'ús:** Editar un perfil
Actor: Administrador
Descripció: L'administrador ha de poder editar un perfil que ja existeix a la plataforma.
- **Cas d'ús:** Eliminar un container
Actor: Administrador
Descripció: L'administrador ha de poder eliminar un perfil existent i tota la informació relacionada amb ell.
- **Cas d'ús:** Veure tots els usuaris i la seva informació
Actor: Administrador
Descripció: L'administrador vol veure tots els usuaris donats d'alta a la plataforma així com la informació bàsica d'aquests.
- **Cas d'ús:** Donar d'alta un usuari
Actor: Administrador
Descripció: L'administrador vol donar d'alta un nou usuari a la plataforma per relacionar-lo amb perfils d'usuari.

- **Cas d'ús:** Editar un usuari
Actor: Administrador
Descripció: L'administrador ha de poder editar un usuari que ja existeix a la plataforma.
- **Cas d'ús:** Eliminar un usuari
Actor: Administrador
Descripció: L'administrador ha de poder eliminar un usuari existent i tota la informació relacionada amb ell.
- **Cas d'ús:** Relacionar un container amb n perfils diferents
Actor: Administrador
Descripció: L'administrador ha de poder relacionar un container amb perfils existents a la plataforma.
- **Cas d'ús:** Relacionar un usuari amb n perfils diferents
Actor: Administrador
Descripció: L'administrador ha de poder relacionar un usuari amb perfils existents a la plataforma.
- **Cas d'ús:** Accedir a la plataforma
Actor: Usuari final
Descripció: L'agent que executa l'usuari final ha d'accedir a la plataforma a través de l'API.
- **Cas d'ús:** Llistar els containers d'un usuari
Actor: Usuari final
Descripció: A través de l'agent, s'ha de llistar els containers actius que l'usuari té assignats.
- **Cas d'ús:** Descarregar container
Actor: Usuari final
Descripció: A través de l'agent, s'ha de poder descarregar els containers seleccionats.

Capítol 8

Implementació

A continuació veurem la implementació de tots els requisits i casos d'ús en els sistemes corresponents. Començarem veient la part de la plataforma web on està gairebé tot el gruix de feina realitzada.

8.1 Servidor

8.1.1 Preparació de l'entorn

Com hem vist a l'apartat anterior el servidor web haurà de tenir un Apache per tal de poder executar el codi PHP. S'ha desenvolupat en PHP, ja que és un llenguatge que ens serveix per treballar a la banda de servidor i ja tenia experiència treballant-hi amb ell. Cal dir que s'ha programat sense cap mena de framework perquè no tenia cap experiència amb frameworks de PHP i les corbes d'aprenentatge acostumen a ser molt elevades fent un aprenentatge autònom i pel temps que es tenia vaig decidir fer el desenvolupament en PHP pur sense cap framework. No ha sigut cap impediment a l'hora d'avançar amb la creació de la plataforma, així que el resultat ha estat positiu. A més a més, he estructurat el codi per treballar amb Model-Vista-Controlador (MVC) i en programació orientada a objecte (POO). La decisió d'organitzar-ho així ha estat perquè quan s'hagin de modificar funcions, crear noves funcionalitats o quelcom per l'estil, sigui molt més senzill aplicar aquests canvis.

Un exemple de com l'he implementat és la imatge 8.1. Per una banda hi ha un únic fitxer que gestionarà les peticions que es fan, indicant-li un controlador i una acció i aquest executarà l'acció del controlador pertinent. En aquest exemple seria l'acció "edit" del controlador "profileController". El controlador serà l'encarregat de fer les peticions que necessiti als objectes i un cop tingui la informació a través de la funció view passarà aquesta informació a la capa de presentació. En aquest cas cada pàgina que he creat té el seu controlador de tal forma que aquest tindrà totes les accions que es necessiten a la pantalla. A més a més, tots els controladors hereten d'una classe bàsica anomenada BaseController. Aquest objecte té dos mètodes, el view i el redirect. El mètode view ens servirà per enviar informació del controlador a la capa de presentació i el mètode redirect ens servirà per mostrar de forma única la direcció a la capa de presentació per invocar funcions dels controladors.

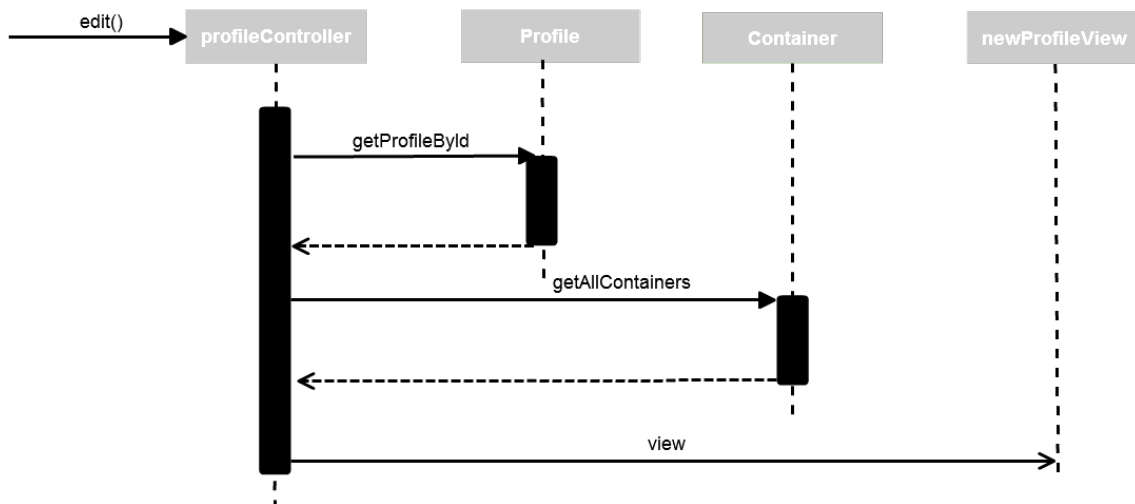


Figura 8.1: Exemple flux d'execució

A part de PHP, com hem vist anteriorment, també he fet servir HTML, CSS i JavaScript per fer la part visual de la plataforma. Al ser una pàgina web on l'administrador haurà de donar d'alta i gestionar tots els containers, usuaris i perfils, havia de ser una mica amigable i fàcil de fer servir. Per aquest motiu em vaig decantar per treballar amb Bootstrap [6]. Bootstrap és un framework que ens ajuda a crear una estructura i una imatge base bastant bona sense haver de dissenyar la pàgina web de zero. A més a més Bootstrap treballa amb jQuery, un framework de Javascript que és molt senzill de fer servir i ajuda molt a desenvolupar funcionalitats. Finalment comentar que aquests dos frameworks ajuden molt a tenir compatibilitat amb la gran majoria de navegadors, és a dir, tota la gestió de compatibilitats de funcions Javascript o tags HTML no l'hem de fer nosaltres.

Per realitzar les proves, he montat el servidor web en una màquina virtual amb Linux, a la que he hagut d'instal·lar Apache i els components necessaris de PHP. Un cop tenim el servidor web instal·lat ja podem col·locar el codi a /var/www/html. Un cop tenim el sistema amb Apache, PHP i el codi instal·lat podem comprovar si funciona correctament, des de la mateixa màquina on tinguem la plataforma, provant a l'explorador web introduint l'URL localhost/plataforma-web. És una prova ràpida per veure si s'hi ha instal·lat tot de forma correcta. Si tot ha anat bé veurem la pàgina inicial per entrar a la plataforma web com a la imatge 8.2.

The image shows a login form with a light gray background. At the top, there is a header box with the text "Please Sign In". Below this, there are two input fields: the first is labeled "Username" and the second is labeled "Password". At the bottom of the form is a green button with the text "Login" in white.

Figura 8.2: Pantalla de login

El següent pas és instal·lar la base de dades. Per fer-ho crearem una nova màquina virtual amb Linux i instal·larem el MySQL i el phpMyAdmin. El phpMyAdmin és una eina que ens permetrà administrar la base de dades MySQL a través d'una pàgina web. Un cop tenim el MySQL instal·lat cal crear les taules i els registres de cada una. A continuació veurem quines taules tenim i quins registres:

- **Administrator.** Aquesta taula ens servirà per guardar els usuaris administradors que tindrà la plataforma web.

- Id. És un valor integer i és la clau primària de la taula.
- Username. És un valor varchar amb una mida de 50 caràcters. Ens servirà per guardar el nom d'usuari del administrador.
- Name. És un valor varchar amb una mida de 50 caràcters. Ens servirà per guardar el nom de l'administrador.
- Password. És un camp varchar amb una mida de 256 caràcters. Ens servirà per guardar la contrasenya de l'administrador que estarà guardada amb un hash de 256 bits. He deixat més espai del necessari per si en un futur s'hagués d'implementar un altre hash que ocupi més.

- **Container.** Aquesta taula ens servirà per guardar la informació dels containers que es donaran d'alta a la plataforma.

- Id. És un valor integer i és la clau primària de la taula.
- Name. És un valor varchar amb una mida de 50 caràcters. Ens servirà per guardar el nom del container.

- Path. És un valor varchar amb una mida de 250 caràcters. Ens servirà per guardar la ruta d'on tenim guardat el fitxer container en el sistema de fitxers.
- Version. És un camp varchar amb una mida de 50 caràcters. Ens servirà per guardar la versió del container.
- MD5. És un camp varchar amb una mida de 250 caràcters. Ens servirà per guardar el MD5 del fitxer container.
- WhenDischarged. És un camp datetime. Ens servirà per indicar quan s'ha eliminat el container de la plataforma.

- **Profile.** Aquesta taula ens servirà per guardar la informació dels perfils que es donaran d'alta a la plataforma.

- Id. És un valor integer i és la clau primària de la taula.
- Name. És un valor varchar amb una mida de 50 caràcters. Ens servirà per guardar el nom del perfil.
- Description. És un valor varchar amb una mida de 250 caràcters. Ens servirà per a una petita descripció del perfil.
- WhenDischarged. És un camp datetime. Ens servirà per indicar quan s'ha eliminat el perfil de la plataforma.

- **User.** Aquesta taula ens servirà per guardar tots els usuaris que tindrem en el sistema.

- Id. És un valor integer i és la clau primària de la taula.
- Name. És un valor varchar amb una mida de 50 caràcters. Ens servirà per guardar el nom de l'usuari.
- Username. És un valor varchar amb una mida de 50 caràcters. Ens servirà per guardar el nom d'usuari de l'usuari.
- WhenDischarged. És un camp datetime. Ens servirà per indicar quan s'ha eliminat el perfil de la plataforma.

- **ContainerPerfil.** Aquesta taula ens servirà per guardar la relació de quins containers té assignats cada perfil.

- Id. És un valor integer i és la clau primària de la taula.
- ContainerId. És un valor integer i és una clau forana a l'id de la taula Container.
- ProfileId. És un valor integer i és una clau forana a l'id de la taula Profile.
- WhenDischarged. És un camp datetime. Ens servirà per indicar quan s'ha eliminat la relació entre el container i el perfil.

- **PerfilUsuari.** Aquesta taula ens servirà per guardar la relació de quins perfils té assignats cada usuari.

- Id. És un valor integer i és la clau primària de la taula.

- ProfileId. És un valor integer i és una clau forana a l'id de la taula Profile.
- UserId. És un valor integer i és una clau forana a l'id de la taula User.
- WhenDischarged. És un camp datetime. Ens servirà per indicar quan s'ha eliminat la relació entre el perfil i l'usuari.

Com podem veure en gairebé tots els objectes principals tenen un registre anomenat "When-Discharged". Aquest camp el farem servir per fer un esborrat lògic en el sistema. La decisió de fer-ho així és perquè si ha estat un error, es pugui tornar en un estat anterior de forma més senzilla que no recuperant un backup. A més a més ens pot servir en un futur per poder fer estadístiques i veure informació sobre la plataforma. Per evitar problemes he decidit fer un borrat no lògic amb els administradors.

Un cop ja tenim la base de dades creada amb la seva estructura pertinent només ens queda saber la ruta del directori que farem servir per guardar tot el repositori de containers. Per simplificar he decidit fer servir la mateixa màquina virtual on tenim el servidor web per fer-lo servir per guardar els containers.

Finalment cal configurar les connexions a la base de dades i al repositori a la plataforma. A la carpeta config tindrem dos fitxers, database.php i config.php. En el primer indicarem els paràmetres per tal de realitzar les connexions (nom d'usuari de la base de dades, nom de la base de dades, IP, i contrasenya), i en el segon a la variable PATH_CONTAINERS indicarem la ruta del directori on guardarem els containers.

L'estructura emprada a la plataforma és la que podem veure a la imatge 8.3. Podem veure com està estructurada la part del Model i hi veiem mètodes i propietats de cada un dels objectes. Cal comentar que tots els objectes tenen els setters i getters corresponents per cada una de les seves propietats.

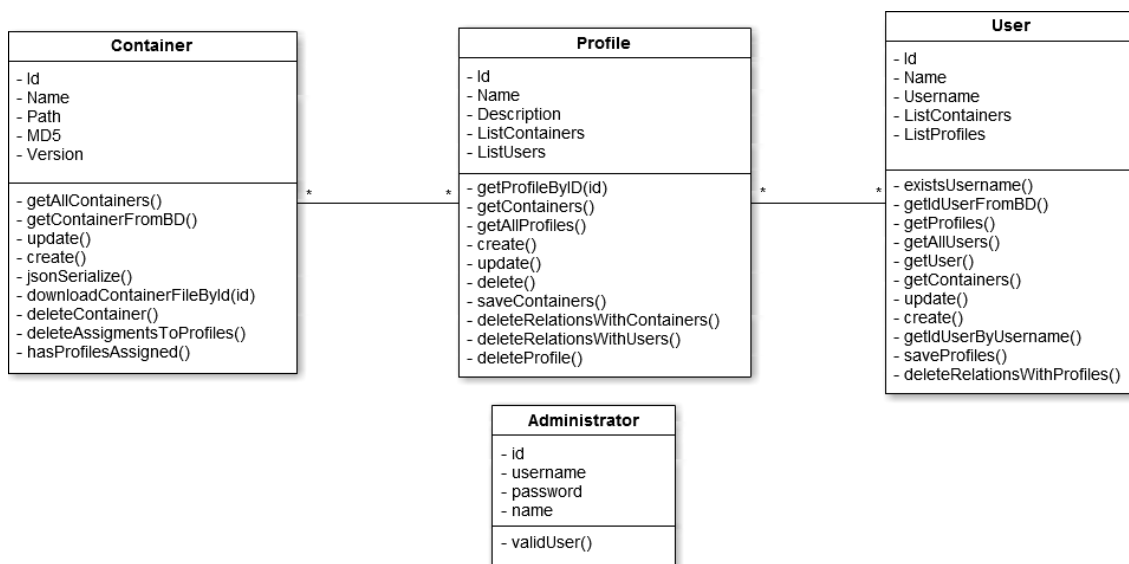


Figura 8.3: Diagrama de classes

8.1.2 Funcionament

Un cop vist el diagrama de classes, ja podem veure el funcionament del sistema i a mesura que vagi explicant la plataforma anirem veient l'ús de les classes i els seus mètodes.

Per poder accedir a la plataforma com a administrador caldrà tenir un donat d'alta a la base de dades, només cal fer una sentència INSERT a la taula Administrator i tenint en compte que la contrasenya se li ha d'aplicar un hash de 256 bits. Un cop tenim el administrador donat d'alta ja podem introduir l'usuari i contrasenya a la pàgina de login. Quan l'administrador introdueixi els paràmetres l'objecte Administrator serà l'encarregat de comprovar que els valors són correctes. En cas de no ser-ho s'avisarà a l'usuari, en cas contrari, el sistema redirigirà a l'usuari a la pàgina inicial de la plataforma, al panell de control. Per tal de tenir un control per saber si tenim un usuari registrat, ens guardarem l'id del administrador en una variable de sessió.

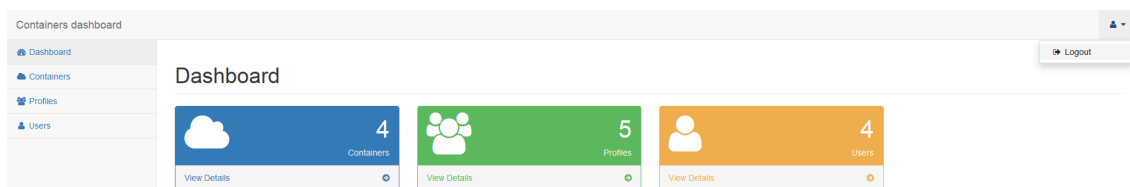


Figura 8.4: Panell de control

Com podem veure a la imatge 8.4 podem diferenciar 3 parts. Per una banda tenim el menú lateral que ens servirà per moure'ns a través de la plataforma, per una altra banda tenim la barra superior on tindrem l'opció de tancar sessió a la plataforma i finalment la part del cos de la web on h tindrem el cos de cadascuna de les pàgines de la plataforma web. Comentar que tant la barra lateral com la barra superior es mantindrà per cada una de les pàgines que té la plataforma.

La idea del panell de control és que sigui un resum de com està la plataforma actualment. En aquest cas podem veure el nombre de containers, perfils i usuaris donats d'alta al sistema. A més a més, la plantilla que he fet servir per construir la part visual ofereix moltes possibilitats per un futur com per exemple, poder fer gràfiques a partir de dades obtingudes a la plataforma.

Containers management

[Add new](#)

List containers

	Name	Version	Path	MD5
	TestContainer	1.0.1	test.txt	d41d8cd98f00b204e9800998ecf8427e
	container2	2.4.2	containerTest.txt	a3ed2248ded884d22a88d0494cfc40ed

Figura 8.5: Gestió de containers

A continuació passem a veure la funcionalitat que ofereix la plataforma de forma senzilla. Per començar tenim l'opció Containers del menú. A l'entrar a aquesta pàgina (imatge 8.5) podem veure un llistat dels containers que ja estan al sistema així com la versió, ruta d'on podem trobar el fitxer container, el nom del container i el MD5. A més a més, per cada un dels containers donats d'alta podem editar tota la seva informació fins i tot a canviar el fitxer container i també els podem eliminar.

A la mateixa pantalla podem veure que tenim l'opció d'afegir un nou container al sistema. El formulari serà el mateix per l'opció d'afegir un nou container a poder editar-ne un. La diferència

que hi haurà serà que quan editem un container, el controlador d'aquesta pàgina anirà a buscar la informació del container seleccionat i la introduirà als camps corresponents. Quan l'administrador pica el botó guardar, el sistema guardarà la informació que li ve dels camps del formulari a l'objecte Container i aquest guardarà la informació a la taula de base de dades Container. Abans de guardar el container a la base de dades, l'objecte guardarà primer el fitxer container a la ruta del sistema de fitxers especificada a la configuració i a continuació guardarà el container a la base de dades. La funció que farem servir per guardar el fitxer és la "move_uploaded_file" nativa de PHP. D'aquesta forma tindrem el MD5 i la ruta d'on tenim guardat el fitxer. La diferència que trobem entre guardar un nou container i guardar un container existent és que quan es pica l'opció guardar, a més a més de la informació dels camps del formulari, s'està enviant el id intern del container i així es podrà identificar quan estem creant un container nou o no.

Edit container

Name container

Version

File

container2.txt

Figura 8.6: Formulari per afegir i editar un container

Com podem veure a la imatge 8.6, veiem que també hi ha l'opció d'eliminar el container. Només apareixerà aquesta opció quan estiguem editant un container, ja que sinó no existirà el container a la base de dades. Per tant si volem eliminar el container, implica que s'esborrarà de forma lògica el container en el sistema i les relacions que tenia amb els perfils. Com hem vist anteriorment fem un esborrat lògic per dos motius, un per poder restaurar en cas d'error de l'administrador o bé per poder fer, en aquest cas per exemple, un històric de quins containers han estat assignats a quin perfil. Quan l'administrador elimina el container, es crida al mètode delete del controlador i s'envia l'id del container que es vol eliminar. Es posarà la data del dia en què es borra al camp whenDischarged de la taula Container de la base de dades del container pertinent i també a la taula ContainerPerfil a tots els registres on estigui l'id del container que s'elimina.

Continuem amb la resta d'opcions de la plataforma, seguim amb la pantalla de gestió dels perfils. Com podem veure a la imatge 8.7 veiem de nou un llistat però en aquest cas sobre tots

els perfils donats d'alta al sistema amb el nom i la descripció de cadascun d'ells. A més a més, tornem a veure les opcions d'editar i eliminar cada un dels perfils donats d'alta a la plataforma. De la mateixa forma que també tenim un botó per crear un nou perfil a la plataforma.

Profiles management

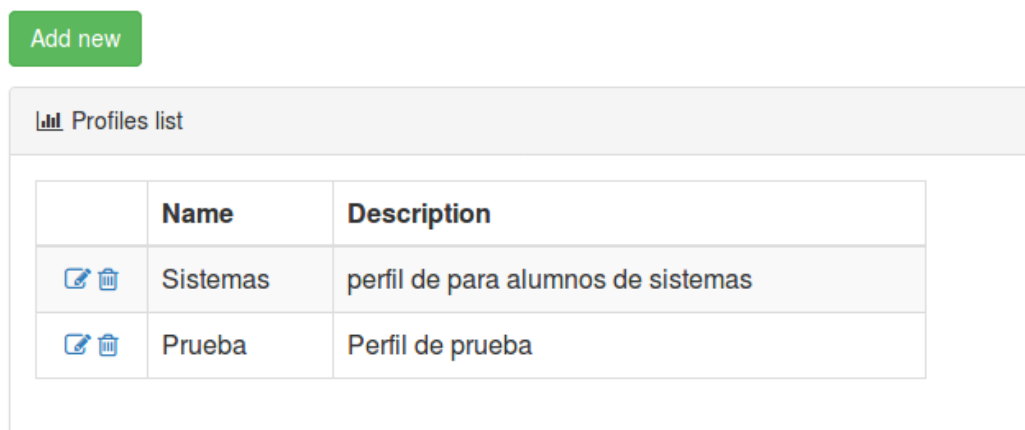


Figura 8.7: Gestió de perfils

De nou com veiem a la imatge 8.8, el formulari per crear un perfil o per editar-lo serà el mateix, en aquest cas el controlador del perfil anirà a buscar a la base de dades la informació del perfil si s'està editant o no, i el sistema posarà la informació als camps corresponents si s'estigués editant. A més a més veiem que al costat dels camps per escriure el nom i la descripció hi ha un llistat de containers amb un checkbox al costat. Aquest llistat ens permetrà assignar, de forma molt senzilla, els containers que vulguem al perfil que estem creant/editant. Automàticament, els usuaris que tinguin assignat el perfil tindran disponible els containers assignats en aquesta llista.

Edit Profile

Name

Description

TestContainer (v.1.0.1)

container2 (v.2.4.2)

Figura 8.8: Formulari per afegit i editar un perfil

Quan esborrem un perfil s'eliminarà les relacions que tinguin amb els containers i amb els usuaris. Així doncs tots els usuaris que tinguessin aquest perfil deixaran de poder fer servir els containers que el perfil tenia associat.

Per últim toca veure la pantalla dels usuaris. La primera pantalla que carrega el controlador a l'accedir a través del menú és el llistat de tots els usuaris donats d'alta al sistema tal com veiem a la imatge 8.9. Seguint el model que hem vist fins ara, a la pantalla de gestió dels usuaris veurem el llistat dels usuaris amb les opcions d'eliminar o editar. En aquest cas l'única informació que es mostrarà en aquest llistat serà el nom de l'usuari i el username. A més a més, tal com hem vist en els casos anteriors, tenim l'opció de crear un nou usuari.

Users management

[Add new](#)

 Users list

	Name	Username
 	sergi Soriano	pim
 	paco	pacomo

Figura 8.9: Gestió dels usuaris

De nou, el formulari per crear i editar un usuari és el mateix en els dos casos, com sempre, si estem editant el controlador, carregarà la informació de l'usuari corresponent. A més a més com podem veure a la imatge 8.10, a la banda esquerra tenim el formulari per afegir l'usuari amb els camps nom i nom d'usuari i a la banda dreta tenim un llistat. Aquest llistat mostra tots els perfils creats a la plataforma i que no estan esborrats. Per tant, marcant-los amb el checkbox podrem assignar perfils a l'usuari de forma senzilla. Automàticament l'usuari podrà accedir als containers que tingui associat el perfil. Si es borra un usuari s'esborrarà les relacions que tenia amb el perfil i a més a més s'inhabilita l'accés a la plataforma.

Edit user

Name

Username

Sistemas
 Prueba

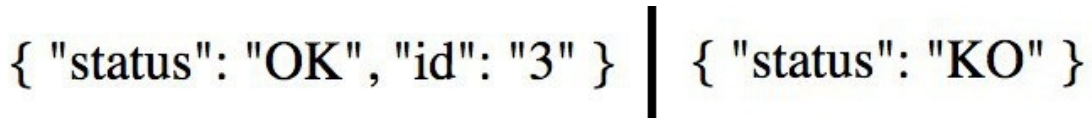
[Delete](#) [Save](#)

Figura 8.10: Gestió dels usuaris

En aquest cas he creat les opcions d'editar la informació bàsica de l'usuari així com la creació de nous, però la idea és que els usuaris no els hàgem de crear un per un al sistema sinó que l'organització on s'implementi la plataforma et proporcioni els usuaris que tenen al seu sistema. L'única opció disponible que hi hauria seria la de poder assignar perfils als usuaris.

A més a més de la part gràfica que hem vist fins ara hi ha un segon mètode per poder accedir a la plataforma. Com que l'agent necessitarà accedir a les dades de la plataforma, he creat una API que permetrà a l'agent de l'usuari final a fer certes operacions. Per poder-hi accedir s'haurà d'entrar via: `nomdelservidor/plataforma-web/api/funció/paràmetre`. Exactament he implementat les següents funcions:

- `userAcces`. Aquesta funció permet a l'agent saber si l'usuari que està executant el programa és un usuari vàlid per la plataforma. A aquesta funció s'envia un paràmetre que serà el username de l'usuari, registrat a l'ordinador on s'està executant l'agent, aplicant-li una funció de hash de 256 bits. La API retornarà una resposta en format JSON indicant si l'usuari és vàlid o no. A continuació veiem un parell de captures on veiem l'estructura de la resposta JSON. A l'esquerra veiem la resposta que torna al registrar un usuari vàlid i a la dreta la resposta d'un no vàlid.



```
{ "status": "OK", "id": "3" } | { "status": "KO" }
```

Figura 8.11: Respostes JSON de la funció `userAccess`.

- `getContainersList`. Aquesta funció ens serveix per poder saber tots els containers que l'usuari pot fer ús. Si l'usuari té dos o més perfils diferents que tinguin assignat un mateix container, la resposta només el retornarà un cop. Com a paràmetres rep el mateix que el cas anterior. Com en el cas anterior, la resposta de la funció serà en format JSON. A continuació podem veure l'exemple d'una resposta.

```
[ { "id": "5", "name": "TestContainer", "version": "1.0.1" }, { "id": "6", "name": "container2", "version": "2.4.2" } ]
```

Figura 8.12: Respostes JSON de la funció `userAccess`.

- `getContainer`. Finalment tenim la funció que ens permetrà descarregar els containers. En aquest cas els paràmetres que rep la funció és el mateix dels casos anteriors i a més a més l'id del container que es vol descarregar. Si es vol descarregar més d'un, s'haurà de fer un bucle a l'agent.

Per tal que es pogués accedir a l'API a través de l'URL fent servir una forma més amigable ha calgut modificar el fitxer `.htaccess`. Aquest fitxer és un fitxer de configuració per cada directori d'on tenim la plataforma. Entre altres opcions que es poden activar en aquest fitxer, l'he utilitzat per fer servir el mod `rewrite` d'Apache. Aquest mòdul ens permet modificar crear direccions URL alternatives. D'aquesta manera l'usuari no haurà de saber quina és la nomenclatura del nom del paràmetre. Internament en el `.htaccess` vincularem la funció a un paràmetre intern de PHP i igual amb els paràmetres que es reben de cada funció.

8.2 Client

Un cop ja hem vist com funciona i com he implementat la plataforma només ens queda veure com funciona l'agent.

Com hem vist anteriorment l'agent és un programa escrit en C++ per tal de descarregar containers de la plataforma. Com ja he comentat, la idea és que l'agent estigui a cada ordinador de l'organització els quals s'hagin de fer servir i així l'agent podrà descarregar els containers que tingui assignat l'usuari que estigui identificat en l'ordinador. Aquest agent serà un programa senzill que s'executi des de la terminal sense cap interfície gràfica. He decidit crear un programa així de senzill, ja que ho ideal seria executar l'agent a l'arrencar el sistema operatiu i així quan l'usuari tingui l'escriptori llest ja tingui els programes instal·lats.

Abans de continuar m'agradaria comentar que he fet servir les llibreries Curl i JSON for Modern C++ per desenvolupar el programa. La primera llibreria ens serveix per fer peticions HTTP, en aquest cas m'ha servit per fer les peticions a l'API de la plataforma i rebre les respostes corresponents. Aquesta llibreria es pot fer servir activant un flag quan compilem el programa. La segona llibreria ens serveix per poder convertir la una cadena de paraules a un objecte, és a dir, poder fer un parser a la resposta de l'API i poder recórrer de forma correcta el JSON que es rep. Aquesta llibreria l'he hagut de descarregar des de la pàgina oficial de JSON. Més endavant veurem com i quan s'utilitzen.

Igual que he fet a la part web del projecte, he desenvolupat l'agent amb programació orientada a objectes. Així doncs tindrem el fitxer principal "agent.cpp" el qual s'encarregarà de fer les crides als mètodes de l'objecte User. L'objecte User tindrà els següents mètodes públics:

- `getUserRegistered`. Aquesta funció retorna el nom d'usuari que està registrat en l'ordinador. Retorna un string.
- `hasAccess`. Aquest mètode serà l'encarregat de fer la petició a l'API per saber si l'usuari que està registrat a l'ordinador té accés a la plataforma. Retorna un boolean.
- `getContainerList`. Aquest mètode serà l'encarregat de fer la petició a l'API per tal de saber el llistat de containers disponibles per l'usuari. Retorna la resposta obtinguda.
- `getFile`. Aquest mètode necessita dos paràmetres, l'id del container i el nom d'aquest. L'id el necessitem per fer la petició a l'API i rebre el fitxer container corresponent. El nom el necessitem per guardar el container al disc de l'ordinador.

Un cop hem vist els mètodes disponibles només queda veure el procediment que es fa. Primer es comprovarà que l'usuari tingui accés a la plataforma. De no ser així el sistema li mostrarà un missatge d'error indicant-li que es posi en contacte amb el personal de TI de la seva organització. Si té accés, el programa farà la petició a l'API per saber els containers que té associats l'usuari. Quan es rep la resposta el programa convertirà el string que es rep a un objecte JSON. Un cop tenim el JSON és més fàcil recorre'l per tal de mostrar per pantalla els containers. A més a més, quan es mostren els containers per pantalla, davant del nom hi ha un número. Aquest número li servirà a l'usuari per seleccionar el container que vol descarregar-se. Un cop l'usuari selecciona el container el programa farà una petició a l'API i enviarà l'id del container que es vol descarregar.

Capítol 9

Canvis respecte la primera planificació

A l'inici d'aquest projecte es va dur a terme una planificació que podem observar a la imatge 3.1. Les tasques de GEP s'han realitzat segons la previsió feta però hi hagut canvis a la resta de les tasques. El primer canvi ha estat amb la tasca de desenvolupament web i el desenvolupament de l'agent. S'ha ampliat el temps a realitzar la tasca del desenvolupament web perquè s'ha decidit fer una interfície gràfica més fàcil d'entendre i usable. A més a més, el temps a desenvolupar l'API ha sigut més del que es tenia previst, ja que no havia tingut mai experiència en fer-ne una. Vaig tenir problemes el fitxer .htaccess i amb la reescriptura de l'URL per tal que fos amigable. Amb aquests canvis el desenvolupament de la web s'ha allargat fins a la setmana 17.

També ha variat el temps de desenvolupament de l'executable. Com que el temps emprat per fer la primera part va ser més gran del plantejat es va haver de variar l'aplicat a l'executable. Tot i així, amb el director del projecte vam comentar que si s'havia de reduir algun punt podia ser l'executable. És a dir, en lloc de fer que fos amb interfície gràfica, fer que l'executable vagi per línia de comandes i a més a més que funcionés per Linux. Gràcies al fet que havíem acordat això va ser senzill adaptar els temps. Per tant, la fase de desenvolupament de l'agent es va dur a terme des de la setmana 17 fins a finals de la setmana 20. Podem veure el gantt final a l'imatge 9.1

Aquest canvi de planificació no ha fet variar el cost final del projecte perquè com veiem, només s'ha variat la durada d'algunes tasques però no la durada total del projecte. A més a més com que no he diferenciat cap rol dins del projecte, no hi hagut cap variació de cost humà. Però tot i així hi hagut un canvi en el software utilitzat. Per problemes externs al projecte l'equip de sobretaula ha estat funcionant amb un Windows Professional 7. Aquest canvi implica una variació en el pressupost perquè s'ha de tenir en compte la llicència del Windows. Cal dir que es podria fer servir una llicència gratuïta que ofereix Windows als estudiants de la Universitat politècnica de Catalunya, però per tenir un pressupost més acurat farem com si no existís aquesta possibilitat.

Producte	Preu	Unitats	Cost
Windows Professional 7	140 €	1	140 €
Total			140 €

Taula 9.1: Cost del Software

Aquest canvi provoca un canvi al pressupost final deixant-lo en 5.179,3 €.

Concepte	Cost
Recursos humans	4.344 €
Recursos hardware	219 €
Recursos software	140 €
Recursos contingència	456,3 €
Altres	20 €
Total	5.179,3 €

Taula 9.2: Pressupost total

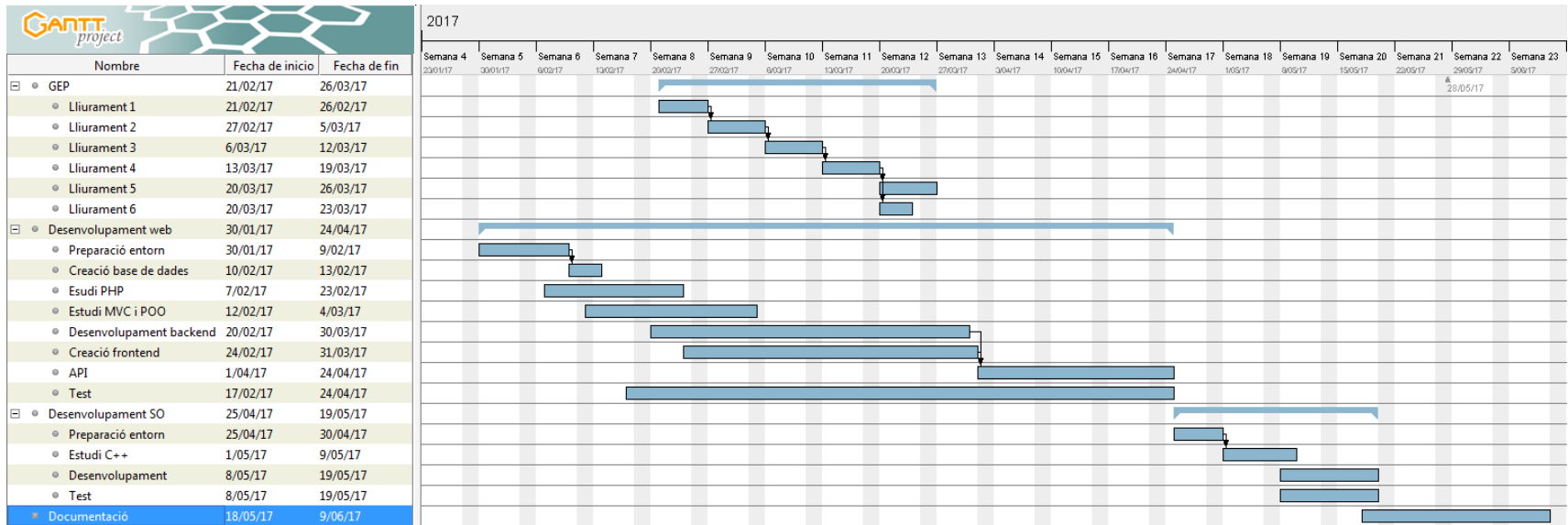


Figura 9.1: Diagrama de Gantt definitiu.

Capítol 10

Test

Un cop tenim el sistema muntat cal fer un test per tal de comprovar que tot funciona de la forma esperada. S'han realitzat dos tipus de tests. El primer ha estat un test de codi, tant a la plataforma com a l'agent.

Primer de tot s'han fet test que els paràmetres i valors que s'obtenen són vàlids i correctes. Aquest test a la plataforma s'ha realitzat debuguejan amb Netbeans, ja que podem posar punts de parada d'execució i anar veient com evolucionen els valors i veient el fil d'execució. A més a més, s'han fet comprovacions en les insercions i actualitzacions a la base de dades evitant duplicitats o insercions o actualitzacions errònies gràcies a HeidiSQL.

En el cas de l'agent, al ser més feixuc la depuració del programa he fet servir un altre mètode. Ha consistit a posar-me logs per pantalla del que anava passant en tot moment per seguir el fil d'execució i veure que els valors els tractava de forma correcta. Per exemple si l'agent agafava de forma correcta el nom d'usuari del sistema o si s'estava connectant bé a l'API de la plataforma, etc. A més a més per tal de veure si s'estava descarregant el container seleccionat, s'han creat diversos containers a la plataforma web i s'ha comprovat que els containers descarregats eren els mateixos dels quals s'havia fet la petició de descàrrega.

El segon tipus de test realitzat ha estat amb usuaris, els quals tenen nocions d'informàtica bàsiques, que fessin servir la plataforma web. El que he volgut aconseguir amb això era veure la reacció dels usuaris en fer servir el sistema, és a dir, si era fàcil de fer servir, si els hi era fàcil l'organització de la web, si no creava confusió, etc. Després dels resultats obtinguts es van aplicar algunes millores per tal que fos més atractiu visualment, ja que com s'havia fet el primer test ja no hi havia errors a l'hora d'editar o crear o veure un container, perfil o usuari.

Capítol 11

Conclusions

L'objectiu inicial d'aquest projecte era tenir una plataforma on poder donar d'alta containers de tal forma que es poguessin assignar a perfils d'usuari i poder-se descarregar a un ordinador a través d'un programa.

Gràcies a tota la informació proporcionada podem dir que s'ha complert correctament amb els requisits. S'ha creat una plataforma web per gestionar tot el tema de crear containers, perfils, assignar containers als perfils, assignar perfils a usuaris, fent que a més a més les pantalles per fer-ho siguin entenedores i clares proporcionant així una bona experiència d'usuari. Podem assegurar també que els fitxers que es puguen a la plataforma es poden descarregar als ordinadors mitjançant una API i un programa que fa servir aquesta API.

Algunes de les dificultats que m'he trobat a l'hora de fer front al desenvolupament del projecte han estat sobretot per què feia molt temps que no programava amb els llenguatges PHP i C++ i a més a més en programació orientada a objectes perquè la meua experiència era molt poca. Afegint que amb el PHP he separat les tres capes de model, vista i controlador, que tampoc hi havia treballat mai amb PHP i algunes coses canvien respecte a altres llenguatges com és VB.Net. Tot i així gràcies als consells del director que m'ha anat guiant quan tenia alguna dificultat o no tenia clar com atacar un tema, m'ha sortit prou bé.

11.1 Treball futur

Es podria dir que el projecte que he desenvolupat és una base perquè s'hi puguin afegir canvis i millores, algunes ja les hem vist al llarg del document. Com per exemple, algunes de les característiques noves que es pot afegir, és tenir un control de quantes vegades es descarreguen els containers per així tenir estadístiques de quin dels containers es fa servir més i a partir d'aquí aplicar millores de rendiment o oferir altres containers o quelcom per l'estil. A més a més, ens pot servir per veure a quines hores o moments del dia hi ha més activitat i si escau posar-hi més recursos en moments concrets del dia per exemple. Com podem veure es poden fer modificacions per tal de tenir una plataforma més completa i competitiva per tal d'oferir un millor servei.

11.2 Conclusió personal

No podria acabar aquest projecte sense fer una valoració personal. Al principi no tenia clar fins on podria arribar. Tenia experiència amb creació de pàgines web però feia molts anys que no tocava

PHP i no tenia clar si recordaria i seria capaç d'abstreure'm del llenguatge amb el qual treballa diàriament. Després de tenir alguns dubtes bastant importants de com enfocar el MVC en PHP i després d'investigar ja ho tenia, ara només quedava fer la resta. Cal remarcar de nou que el suport, consells i recomanacions de part del director del projecte han ajudat molt a no quedar-me encallat en algun punt. L'altre punt on no sabia com aniria era la creació de l'agent, feia molt que no programava en C++ i menys amb orientació orientada a objectes, però de nou, algun consell i tot va anar fluid.

A part dels temors que podria tenir al principi també cal dir que tenia ganes, ja que sabia que obtindria coneixements nous i formes de treballar diferents, més actuals. Treballar amb repositoris, amb diferents llenguatges o la separació de MVC o POO (programació orientada a objectes) són experiències que n'havia tingut però no en molta profunditat. I al final acabes tenint una agilitat que al principi no es tenia. A més a més, són coses que al món laboral es fan servir actualment i trobo que és molt necessari aquest coneixement. A més a més, com ja havia vist al llarg de la meua breu experiència laboral és molt important poder depurar correctament el teu codi i un cop més al llarg del desenvolupament del projecte ho he vist clar. Finalment i per acabar, dir que estic content amb el resultat final del projecte.

Bibliografia

- [1] J. Postel. Rfc 959 file transfer protocol (ftp). [Online]. Disponible: <https://tools.ietf.org/html/rfc959>
- [2] Apache software foundation. [Online]. Disponible: <https://www.apache.org/>
- [3] Mysql. [Online]. Disponible: <https://www.mysql.com/>
- [4] Xampp. [Online]. Disponible: <https://git-scm.com/>
- [5] Trello. [Online]. Disponible: <https://trello.com/>
- [6] Bootstrap. [Online]. Disponible: <http://getbootstrap.com/>