



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica en Informàtica de Gestió

**WARCOMRADES
VOLUM I**

**DANIEL RUIZ PONCE
ENRIC SESA NOGUERAS**

TARDOR 2009

Resum

Warcomrades és un projecte de desenvolupament d'un joc de rol web. Actualment hi ha molts productes en el sector, amb diverses temàtiques, i depenent del producte el nombre de jugadors pot arribar a ser de centenars de milers.

El que pretén aquest projecte és el desenvolupament d'una maqueta funcional com a primera iteració, on el requisit primordial sigui l'escalabilitat, la facilitat de manteniment i que el producte sigui modular.

El producte permetrà a l'usuari configurar-lo al seu gust, per tal de millorar la interacció persona - ordinador, fer-la més fluida. Aquest es considera un punt rellevant, ja que es pensa que pot ser un clar avantatge competitiu.

Tal com es detalla en aquest document, la elecció de les tecnologies per l'elaboració s'ha fet orientat a poder complir els requisits anteriors.

En definitiva es vol una maqueta que permeti publicar una versió alfa tancada en la següent iteració centrant-se més en aspectes visuals, de continguts i sobretot en la jugabilitat. Que permeti la modificació total o parcial, de continguts, normes, temàtiques i tecnologies.

Summary

WarComrades is a development project of a web-browser role-playing game.

Currently, there are many products on the sector, with different topics and, depending on the product; the number of players may arrive to be hundreds of thousands.

This project expects a functional model development as first iteration, being the most important requirements: the scalability, easy maintenance, and modularity.

The product will allow the users to configure it to their liking, with the aim of improving human-computer interaction, making it more fluid. This is considered an important point as it is a clear competitive advantage.

As this document details, the development technologies choice has been oriented to satisfy the mentioned requirements.

Definitely, the objective is to make a model that permit to publish a closed alpha version at the next development iteration. It will focus more in visual aspects, especially in contents and playability. This model should allow total or partial modification of content, standards, and technology topics.

Resumen

Warcomrades es un proyecto de desarrollo de un juego de rol web.

Actualmente, hay muchos productos en el sector de diferentes temáticas y, dependiendo del producto, el número de jugadores puede llegar a ser de centenares de miles.

Lo que pretende este proyecto es obtener una maqueta funcional en la primera iteración, donde el requisito principal sea la escalabilidad, la facilidad de mantenimiento y que el producto sea modular.

El producto permitirá al usuario configurarlo a su gusto, con el objetivo de mejorar la interacción persona-ordenador, haciéndola más fluida. Este se considera un punto importante ya que se cree que es una clara ventaja competitiva.

Tal como se detalla en el documento, la elección de las tecnologías para la elaboración se ha orientado a poder satisfacer los requisitos mencionados.

En resumen, se quiere obtener una maqueta que permita publicar una versión alfa cerrada en la siguiente iteración de desarrollo. Se centrará más en aspectos visuales, contenidos y sobretodo en la jugabilidad. Esta maqueta debe permitir la modificación total o parcial de contenidos, normas, temáticas y tecnologías.

Index

	Pàg.
1. Introducció.....	1
1.1 Propòsit del projecte.....	1
1.2 Descripció del projecte.....	1
1.3 Requeriments.....	3
1.4 Situació actual del projecte.....	5
2. Tecnologies.....	9
2.1 El Framework Symfony.....	9
2.2 PHP POO.....	20
2.3 SGBD.....	20
2.4 AJAX.....	22
3. Desenvolupament.....	23
3.1 Model Relacional.....	23
3.1.1 Submodel de Localització.....	23
3.1.2 Submodel d'Habilitats.....	24
3.1.3 Submodel de possessions materials.....	25
3.2 Casos d'ús.....	26
3.2.1 Gestió de Jugador.....	26
3.2.2 Gestió de Viles.....	32
3.2.3 Gestió atacs/moviments mapa.....	37
3.2.4 La Millora.....	40
3.2.5 Consultes.....	43
3.3 Interfície gràfica.....	48
3.4 L'arquitectura.....	51
4. Actualitat del projecte.....	57
5. Conclusió.....	61
Annex A.....	63
Annex B.....	75
Annex C.....	79
Annex D.....	101
Annex E.....	103
Bibliografia.....	109

1- Introducció.

1.1- Propòsit del projecte.

El projecte té com a objectiu el desenvolupament d'un joc en línia que pugui resultar atractiu per els jugadors. S'ha d'elaborar primant l'estructura i compatibilitats de l'aplicatiu.

L'objectiu és obtenir un producte que sigui un joc d'una temàtica, i amb unes normes i continguts, per fer la prova al mercat. Per poder fer un anàlisi posterior, i aplicant el mateix model de procés fer possible refer el joc invertint el mínim de recursos.

En definitiva, es vol obtenir un producte final que permeti analitzar errors comesos, possibles millores, canvis a fer, i que es pugui modificar la gran majoria del producte amb un temps i uns costos reduïts.

1.2- Descripció del projecte.

WarComrades és un joc de rol de navegador web. Un "joc de rol" és la traducció que s'usa en català per definir *role-playing game* (RPG). Un dels aspectes més característics d'aquests jocs és, tal com indica el seu nom, que un o més jugadors interpreten un determinat rol, paper o personalitat. Acostumen a propiciar una trama on el personatge juga un paper important, i segons el joc, disposen de complexos sistemes per millorar, investigar, construir, expandir-se, negociar, guanyar experiència, relacionar-se, etc.

WarComrades representa un món que es divideix en quatre parts (anomenades àrees), en cada àrea hi habita una raça amb característiques que les diferencia de la resta. Així doncs, a l'hora de crear un personatge (anomenat heroi), l'usuari ha de seleccionar una raça dependent dels avantatges que, segons al seu criteri, aquesta li proporcioni. No només això afecta a les característiques de l'heroi, també es pot configurar els seus atributs i habilitats a mesura que vagi guanyant experiència lluitant, millorant l'equip, investigant, aprenent habilitats (màgies i atacs especials, etc.), aprenent un ofici (segons

els oficis, els herois s'especialitzen en uns atributs o altres; tenen diferents bonificacions –anomenades *bonus*– per professió i inclús diferents habilitats).

El jugador comença amb una vila principal (ubicada segons la zona geogràfica del jugador) i el seu primer heroi (en podrà fer varies).

Les opcions que tindrà el joc per el jugador, a grans trets, seran:

- Millorar les característiques del personatge explorant el mapa i lluitant contra enemics i altres jugadors.
- Anar en busca d'una mascota que l'ajudi en combat, encara que també la pot fer servir per comerciar o combinar-ne dues per fer-ne una de millor.
- Fer missions individualment o en grup per tal d'aconseguir objectes o habilitats.
- Millorar les característiques de la seva vila: construint edificis, investigant i creant unitats d'atac i defensa per tal de conquerir viles enemigues que li proporcionin recursos.
- Comerciar amb altres jugadors.
- Crear noves armes i armadures o bé tot tipus de pocions amb elements del joc.
- Encantar armes i armadures amb elements màgics.
- Crear o unir-se a un Clan (associació de diferents jugadors) i conquerir els quatre castells del mapa, els quals ofereixen atributs que afecten a tot el Clan.
- Relacionar-se amb altres jugadors mitjançant la xarxa social que proporcionarà WarComrades.
- Fer combats jugador contra jugador (PVP) i clan contra clan (CVC).
- Configurar la interfície del joc per potenciar la interacció, incloent una versió reduïda per experts (menys recolzament textual, i elements desplaçats per augmentar l'espai de joc).

Algunes d'aquestes característiques es poden dur a terme segons la professió que selecciona el jugador a mesura que el nivell de l'heroi augmenta i gràcies a les diferents habilitats que aquest pot aprendre durant el joc.

Les professions s'escullen en certs nivells i serveixen per a especialitzar l'heroi en diferents camps; com són la lluita cos a cos o a distancia, la màgia o les habilitats de mercader. Cada una d'aquestes professions ofereix un conjunt de camins a seguir que especialitzen encara més a l'heroi i li dona unes certes característiques i habilitats.

Les habilitats són diferents segons la professió de l'heroi, i es poden aprendre a mesura que l'heroi guanya experiència lluitant. L'àmplia varietat d'habilitats que pot triar un heroi fa que s'hagi d'especialitzar segons el tipus d'ús que se li vulgui donar. Tot això fa que augmenti la diversitat en els personatges del joc.

A WarComrades els equips de lluita (heroi i mascota) varien molt entre jugadors, aconseguint una varietat molt àmplia de configuracions. (Tot això es gràcies al gran nombre de factors que afecten als atributs i habilitats d'un heroi o mascota, com la raça, la professió, els diferents equips, i altres aspectes relacionats).

El joc tindrà un sistema de pagament que, tot i no oferir avantatges en la jugabilitat, ofereix, a grans trets, serveis als usuaris com per exemple: canviar la imatge de l'heroi, de la mascota, reiniciar l'arbre d'habilitats, reiniciar els atributs, treure la publicitat, etc.

1.3- Requeriments

WarComrades ha de tenir unes característiques mínimes necessàries sense les quals l'èxit del producte es veuria compromès. Algunes d'aquestes són implícites en les necessitats que hi ha en qualsevol altre tipus de web, i algunes són pròpies del tipus de producte.

Robustesa: L'aplicatiu ha de ser robust, no s'ha de permetre que els usuaris que acabin en el domini del joc tinguin la sensació que l'aplicació no té un bon funcionament, no

s'ha de permetre que passin coses com *rollbacks* de base de dades, domini caigut, pèrdua de dades, etc.

Usabilitat: L'aplicatiu s'ha de diferenciar de la resta de productes del sector aportant un alt grau d'usabilitat. Es vol que quan l'usuari entri tingui la sensació de que pot configurar-ho tot i adaptar-ho al seu gust. Se li han de facilitar bones eines de xarxa social sempre amb bona accessibilitat.

Mitjançant AJAX s'ha d'aconseguir crear una impressió el més interactiva possible de la interfície.

Rapidesa: L'aplicació ha de tenir una càrrega mitja-ràpida (2,35 segons aprox.). Amb aquest nivell de càrrega estaria just a la mitja de Internet.

Visites: El bon posicionament web i la publicitat en altres llocs són essencials per l'èxit de WarComrades.

Consistència: És imprescindible que l'usuari tingui la sensació d'estar usant un producte homogeni final i que li doni la sensació d'estar interactuant amb un joc.

Mantenible: El joc ha de facilitar el seu manteniment, la seva ampliació i la seva modificació.

Multicultural: Warcomrades ha de suportar un nombre el més elevat possible d'idiomes per atraure quants més usuaris millor.

Ingressos: WarComrades és un projecte amb ànim de lucre i per això s'ha d'oferir un espai per a la publicitat de tercers. A més, també s'han d'oferir serveis de pagament que encara falta concretar.

1.4- Situació actual del projecte.

1.4.1- Anàlisi DAFO de WarComrades.

1.4.1.1- Debilitats.

- **Falten recursos.** Des d'un inici era clar que **no es disposa de recursos** per desenvolupar una interfície gràfica de nivell professional. Per el moment, la interfície gràfica es una maqueta prèvia, que es considera temporal, vàlida per la versió alfa tancada, si es creu oportú es pot publicar la beta oberta també amb aquesta interfície ja que és de la propietat de WarComrades, però s'ha de mirar de millorar molt.
- **Nul·la publicitat.** Degut a la falta de recursos el projecte no disposarà a priori de publicitat en altres llocs web; que és necessària ja que hi ha una relació directa entre les pàgines que vinculen la web i les visites que rep.
- **Pocs recursos humans.** Degut a la falta de recursos econòmics només dues persones treballen com a desenvolupadors del joc. Això fa que aquest avanci molt lentament respecte els competidors.
- **Aprenentatge de noves tecnologies.** WarComrades està desenvolupat mitjançant noves tecnologies del sector. El disseny i el desenvolupament es veuen afectats per la corba d'aprenentatge.
- **Poca experiència amb projectes d'aquestes dimensions.**

1.4.1.2- Amenaces.

- **Força competència.** En els últims anys la competència ha crescut desmesuradament i les empreses que actualment tenen un impacte molt gran en el sector acostumen a tenir al darrera un ampli equip, també són empreses consolidades. Això fa que fer-hi la competència sigui quelcom complex.
- **Copyright.** Degut al Copyright hi ha la obligació de crear o comprar totes les imatges.

1.4.1.3- Fortaleses.

- **Capacitat Innovadora.** WarComrades pretén innovar en molts aspectes de cara al seu atractiu visual i funcional per a l'usuari, la qual cosa no està gaire imposada al sector.
- **Enginyeria.** S'ha observat que la majoria dels jocs de navegador han estat desenvolupats per dissenyadors web. Aquests són jocs perfectament funcionals però creiem que no n'hi ha prou. WarComrades ha estat desenvolupat per futurs enginyers en informàtica, la qual cosa garanteix un avantatge des del punt de vista del manteniment i la manipulació respecte als competidors. Per exemple, si es veïés que el joc no és agradable en algun aspecte, es modifica fàcilment i es millora. És més, tal i com s'ha pensat i desenvolupat, serà possible canviar, gairebé totalment, el temari i la funcionalitat del joc en cas de que no funcionés o se'n volgués crear un altre.

- **Noves tecnologies:**
 - **Funcionalitat.** El joc és desenvolupat amb tecnologia innovadora, aplicant patrons com: rapidesa en les peticions, fortalesa davant atacs, disminució de la càrrega d'informació al servidor i al client mitjançant la memòria cache, etc.
 - **Programari lliure.** Davant la idea de no re inventar la roda cada vegada, *scripts* i *frameworks* lliures que fan més ràpida la programació de l'aplicació i possibiliten un funcionament més adequat i un manteniment modular.
 - **Potència.** S'ha estudiat amb deteniment cada pas a l'hora de triar la tecnologia per a crear WarComrades. Així s'ha assegurat que, per exemple, la base de dades que s'usa (PostgreSQL) sigui més potent, ràpida i segura, que la que solen usar la gran majoria de jocs de navegador (MySQL) o que el *framework* utilitzat (Symfony basat en PHP5) sigui més segur, més estructurat i més ràpid que la tecnologia que usen la gran majoria de competidors (PHP o ASP).

- **Mantenir el joc en línia no és car.** Davant el gran creixement i la demanda que té Internet, hi ha molta competència a l'hora d'oferir un bon servei i barat en servidors de *hosting*.

1.4.1.4- Oportunitats.

- **Forta demanda.** El mercat creix i el nombre d'usuaris interessats en el sector també.

- **Innovació.** Al mercat actual no s'ha trobat cap aplicació que ofereixi alguns dels aspectes que es té pensat per a WarComrades, així que s'intentarà atraure usuaris per aquest front.

- **La publicitat.** El creixement dels llocs i negocis en línia ha fet que una web amb molt tràfic d'usuaris (com la que pretén ser WarComrades) tingui uns beneficis extres si incorpora algun apartat amb publicitat.
- **Reutilització de codi.** En el cas de que el joc funcionés seria possible, invertint moltes menys hores de les invertides inicialment, fer-ne un altre de similars característiques. O en cas de fracàs, fer un estudi i modificar-lo identificant els errors que hagin sorgit i relançar l'aplicació.
- **Bon posicionament web.** Actualment, sembla que l'èxit dels productes del sector no és degut al seu bon posicionament web, perquè s'ha realitzat proves amb paraules claus relacionades, i no tenen un bon posicionament. Es pretén posicionar el millor possible el joc, per tal de captar el màxim d'usuaris possibles que cerquin aspectes relacionats amb la temàtica.

1.4.2- Anàlisi de la competència.

S'ha realitzat un estudi sobre la situació de la competència del sector. Està detallat a l'Annex A.

2- Tecnologies.

2.1- El *framework* Symfony.

2.1.1- Introducció.

D'es d'un inici, es tenia clar que WarComrades seria dissenyat i implementat mitjançant un *framework* ja que aquests permeten:

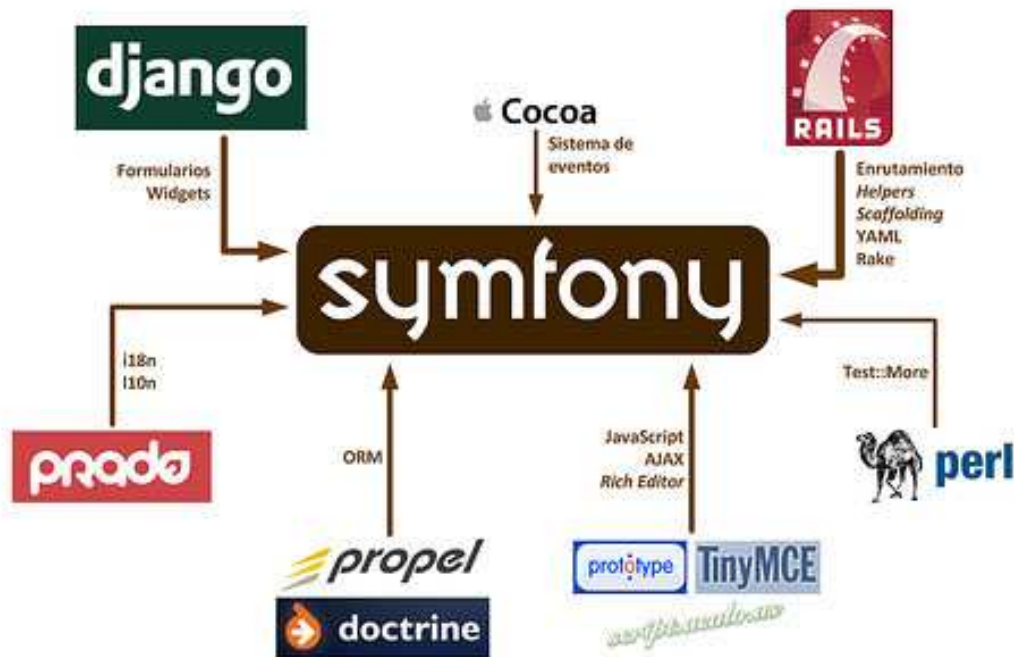
- Simplificar el desenvolupament d'una aplicació mitjançant l'**automatització d'alguns dels patrons** utilitzats per a resoldre les tasques comunes.
- Proporcionar **estructura al codi font**, forçant al desenvolupador a crear codi més llegible i més fàcil de mantenir.
- Facilitar la programació d'aplicacions, ja que encapsulen operacions complexes en instruccions senzilles.

Després d'estudiar i provar diferents *frameworks* com van ser *Cake* o *Zend*, entre d'altres, va ser conclouent que el que més s'adequava a les necessitats de WarComrades era el *framework* anomenat Symfony, que és un complet *framework* dissenyat per a optimitzar, gràcies a les seves característiques, el desenvolupament de les aplicacions web.

El *framework* Symfony és un conjunt de *frameworks* independents que s'han adaptat per treballar conjuntament per el benefici del desenvolupament d'aplicacions. Algunes de les característiques que criden més l'atenció són les següents:

- Separa la lògica de negoci, la lògica de servidor i la presentació de l'aplicació web.
- Proporciona **diverses eines i classes** encaminades a reduir el temps de desenvolupament d'una aplicació web complexa.

- **Automatitza les tasques més comunes**, permetent al desenvolupador dedicar-se per complet als aspectes específics de cada aplicació.
- Està desenvolupat completament amb **PHP 5**.
- Ha estat **provat** en nombrosos projectes reals i s'utilitza en llocs web de comerç electrònic de primer nivell.
- És **compatible amb la majoria de gestors de bases de dades**, com MySQL, PostgreSQL, Oracle i SQL Server de Microsoft.
- Es pot executar tant en plataformes ***nix** (Unix, Linux, etc.) com en plataformes **Windows**.
- Té al darrera una ampla comunitat que l'avalua. Està prevista una versió 2 amb un augment del rendiment aproximat de fins al 70% sobre la versió 1.2 amb la que treballem.



Esquema de les tecnologies externes que incorpora Symfony.

2.1.2- Característiques de Symfony.

Symfony automatitza la majoria d'elements comuns dels projectes web, com per exemple:

- La capa d'**internacionalització** que inclou Symfony permet la traducció de les dades i de la interfície, així com l'adaptació local dels continguts.
- La capa de presentació utilitza **plantilles** (*layouts*) que poden ser creats per dissenyadors HTML sense cap tipus de coneixement del *framework*. Els ajudants (*helpers*) inclosos permeten minimitzar el codi utilitzat en la presentació, ja que encapsulen grans blocs de codi que es poden executar amb simples funcions.
- Els formularis inclouen **validació automatitzada i farciment automàtic de dades** (*repopulation*), el que assegura l'obtenció de dades correctes i millora l'experiència d'usuari.
- Les dades inclouen **mecanismes d'escapament** que permeten una millor protecció contra els atacs produïts per dades corruptes.
- La **gestió de la *caché*** redueix l'ample de banda utilitzat i la càrrega del servidor.
- L'**autenticació i la gestió de credencials** simplifiquen la creació de seccions restringides i la gestió de la seguretat d'usuari.
- El sistema d'**enrutament** i les *URL* netes permeten considerar a les adreces de les pàgines com part de la interfície, a més d'estar optimitzades per als cercadors.
- El suport de correu electrònic inclòs i la gestió de *APIs* permeten a les aplicacions web interactuar més enllà dels navegadors.
- Els llistats són més fàcils d'utilitzar a causa de la paginació automatitzada, el filtrat i l'ordenació de dades.
- Els ***plugins***, les **factories** (patró de disseny Factoria) i els ***mixins*** permeten realitzar extensions a mida per a Symfony.
- Les interaccions amb *Ajax* són molt fàcils d'implementar mitjançant els *helpers* que permeten encapsular els efectes *Javascript* compatibles amb tots els navegadors en una única línia de codi.

2.1.3- Altres característiques rellevants de Symfony.

Entre altres motius, es va triar Symfony perquè es va dissenyar perquè s'ajustés als següents requisits:

- **Fàcil d'instal·lar i configurar** en la majoria de plataformes (i amb la garantia que funciona correctament en els sistemes Windows i *nix estàndards).
- Independent del sistema gestor de bases de dades.
- Prou **flexible** com per a adaptar-se als casos més complexos.
- Basat en la premissa de "convenir en comptes de configurar", en la qual el desenvolupador només ha de configurar allò que no és convencional.
- Segueix la majoria de les millors **pràctiques i patrons de disseny** per a la web.
- Preparat per a aplicacions empresarials i **adaptable** a les polítiques i arquitectures pròpies de cada empresa, a més de ser prou **estable** com per a desenvolupar aplicacions a llarg termini.
- Codi fàcil de llegir que inclou comentaris de "*phpDocumentor*" i que permet un **manteniment** molt senzill.
- Fàcil d'estendre, el que permet la seva integració amb llibreries desenvolupades per tercers.

2.1.4- Entorn de desenvolupament i eines proporcionats per Symfony.

Symfony pot ser completament personalitzat per a complir amb els requisits de les empreses que disposen de les seves pròpies polítiques i regles per a la gestió de projectes i la programació d'aplicacions.

Per defecte incorpora diversos entorns de desenvolupament diferents i inclou diverses eines que permeten automatitzar les tasques més comunes de l'enginyeria del programari:

- Les **eines** que generen automàticament codi han estat dissenyades per a fer prototipatge d'aplicacions i per a crear fàcilment la part de gestió de les aplicacions.
- El *framework* de desenvolupament de **proves** unitàries i funcionals proporciona les eines ideals per al desenvolupament basat en proves (*test-driven development*).
- La **barra de depuració** web simplifica la depuració de les aplicacions, ja que mostra tota la informació que els programadors necessiten sobre la pàgina en la qual estan treballant.
- La interfície de **línia de comandes** automatitza la instal·lació de les aplicacions entre servidors.
- És possible realitzar canvis "en calent" de la configuració (sense necessitat de reiniciar el servidor).

El complet sistema de **log** permet als administradors accedir fins a l'últim detall de les activitats que realitza l'aplicació.

2.1.5- L'ORM.

La informació de la base de dades relacional ha de ser mapejada a un model d'objectes. Això es pot fer amb una eina *ORM*, o Mapejador, i afortunadament, Symfony suporta dos d'ells: Propel i Doctrine. En el cas del projecte s'usarà **Propel**.

El *ORM* necessita una descripció de les taules i les seves relacions per a crear les classes relacionades. Hi ha dues maneres de crear aquest esquema de descripció: **enginyeria inversa** d'una base de dades existent o **creant-lo a mà**. En el nostre cas usarem la primera opció.

PDO és la capa d'abstracció de base de dades que ve amb PHP. El *framework* Symfony suporta totes les Base de dades suportades per *PDO* (MySQL, PostgreSQL, SQLite, Oracle, MSSQL, ...).

Propel treballa amb **quatre classes** per taula. Per exemple, per a la taula Player és necessari:

- **Player:** Un objecte d'aquesta classe representa un únic registre de la taula Player. La classe està buida per defecte.
- **BasePlayer:** La classe pare de Player.
- **PlayerPeer:** La classe defineix els mètodes estàtics que majoritàriament retorna col·leccions d'objectes Player.
- **BasePlayerPeer:** La classe pare de PlayerPeer.

Les taules s'han creat en la base de dades, però no hi ha dades en elles. Per a qualsevol aplicació web, hi ha tres tipus de dades:

- **Dades Inicials:** Les dades inicials són necessaris perquè l'aplicació funcioni (sinó es donen errors de claus foranies). També necessitem un administrador per a cada taula per a poder accedir al *backend*.
- **Dades de Prova:** Les dades de prova són necessàries perquè l'aplicació sigui provada. Cada vegada que s'executen les proves, pot ser necessària una base de dades neta amb algunes dades noves de prova en ella.
- **Les Dades de l'Usuari:** Les dades de l'usuari són creades pels usuaris durant la vida normal de l'aplicació.

Cada vegada que Symfony crea les taules en la base de dades, les dades es perden. Per tant a emplenar la base de dades amb alguns les dades inicials, hem creat una sèrie d'*scripts* amb sentències SQL.

2.1.6- Backend.

El primer pas és crear l'aplicació *backend*. Fins i tot si el *backend* només serà utilitzat pels administradors, és convenient habilitar totes les funcions de seguretat ja incorporades en Symfony.

Ja que només pot tenir un *index.php* per directori, Symfony crea un *index.php* per al primer controlador frontal de producció i nomena als altres amb el nom de l'aplicació.

Els mòduls d'administració tenen moltes més funcions que els simples mòduls. Sense escriure una sola línia de PHP, cada mòdul proporciona aquestes característiques:

- La llista d'objectes paginada.
- La llista és ordenable.
- La llista pot ser filtrada.
- Els Objectes poden ser creats, editats, i eliminats.
- Els objectes seleccionats poden ser eliminats en batch.
- La validació habilitada.
- Els Missatges Flash donen informació immediata a l'usuari.

El Generador d'Administrador proporciona totes les funcions necessàries per a crear una interfície *backend* en un paquet fàcil de configurar.

2.1.7- L'enrutament.

En un context web, una *URL* és l'identificador únic d'un recurs web. Quan s'accedeix a una *URL*, s'està demanant al navegador obtenir un recurs identificat per aquest. Per tant, ja que l'adreça *URL* és la interfície entre la pàgina web i l'usuari, ha de transmetre informació significativa sobre algun recurs al que fa referència.

Però les "tradicionals" *URLs* realment no descriuen al recurs, sinó que exposen l'estructura interna de l'aplicació. A l'usuari no li importa que el lloc web sigui desenvolupat amb el llenguatge PHP o que una certa classe tingui un cert identificador en la base de dades. Exposar el funcionament intern de la aplicació és perjudicial pel que fa a la seguretat.

Les *URLs* són tan importants que existeix tot un *framework* dedicat a la seva gestió: el ***framework d'enrutament***. L'enrutament gestiona el *URI* (*Uniform Resource Identifier*) intern i la *URL* externa. Quan una petició arriba, l'enrutament analitza la *URL* i la converteix en un *URI* intern.

L'*URI* intern està fet de diverses parts: el mòdul, l'acció i la cadena de consulta, que inclou els paràmetres a passar a l'acció.

Les *URLs* amigables són importants perquè transmeten informació a l'usuari. És també útil per a optimitzar el lloc web per als motors de recerca.

2.1.8- Els formularis.

Qualsevol lloc web té formularis; des del simple formulari de contacte fins als complexos amb desenes de camps.

Crear formularis és també una de les més complexes i tedioses tasques d'un desenvolupador web: és necessari crear l'*HTML* del formulari, implementar les regles de validació per a cada camp, processar els valors per a després guardar-los en la base de dades, mostrar missatges d'error, emplenar els camps en cas d'errors, etc.

Per resoldre el problema Symfony proporciona un *framework* per a facilitar l'administració d'un formulari. El ***framework de formularis*** està compost de tres parts:

- **Validació:** El sub-*framework* “*validation*” ofereix classes per a validar les entrades (enter, cadenes, adreça de correu electrònic, ...)
- **Widgets:** El sub-*framework* de “*widgets*” ofereix classes per a la sortida dels camps *HTML* (*input*, *textarea*, *select*, ...)
- **Formularis:** La classes *form* representen formularis fets de *widgets* i validadors i donen mètodes per a ajudar a gestionar el formulari. Cada camp del formulari té el seu propi validador i el seu *widget*.

2.1.9- Sessions.

Symfony gestiona persistentment dades entre les peticions *HTTP*. El desenvolupador no ha de manipular directament les sessions, sinó que usa l'objecte *sfUser*, que representa a l'usuari final de l'aplicació.

Emmagatzemar els objectes directament en la sessió està totalment desaconsellat. El motiu és que l'objecte de la sessió es serialitza entre una petició i una altra i es desa en un arxiu. Quan la sessió és deserialitza, la classe de l'objecte desat s'ha d'haver carregat prèviament i aquest no sempre és el cas. A més de que poden haver objectes de tipus *stalled* si es desen objectes de Propel.

Donat un identificador, el mètode `sfUser::getAttribute()` obté els valors de la sessió d'usuari. Per contra, el mètode `setAttribute()` emmagatzema qualsevol variable PHP en la sessió, per a un determinat identificador. Aquest mètode també té un valor predeterminat opcional a retornar si l'identificador no està encara definit.

Flashes: Un *flash* és un efímer missatge emmagatzemat en la sessió d'usuari que s'eliminarà automàticament després de la pròxima petició. És molt útil quan es necessita mostrar un missatge a l'usuari després d'un redireccionament. El generador d'administració utilitza *flashes* per a mostrar informació a l'usuari quan es guarda un lloc de treball, s'esborra o estén la seva validesa.

Un flash s'estableix utilitzant el mètode `setFlash()` de `sfUser`. El primer argument és l'identificador del *flash* i el segon és el missatge a mostrar. Es pot definir qualsevol flash que es vulgui, però *notice* i *error* són dos dels més comuns (que són utilitzats intensivament pel generador d'administració).

2.1.10- Seguretat.

2.1.10.1- Autenticació.

Quan un usuari no autenticat intenta accedir a una acció segura, es remet la petició a l'acció de *login* tal com s'ha configurat a l'arxiu YAML (routing.yml).

S'ha configurat que tot l'aplicatiu requereixi autenticació exceptuant el casos d'ús de: *login* i alta de jugador.

2.1.10.2- Autorització.

Quan un usuari està autenticat, l'accés a algunes accions pot ser encara més restringit per la definició de credencials. Un usuari ha de tenir les credencials per a accedir a la pàgina que sol·licita.

El sistema de credencials de Symfony és bastant simple i poderós. Permet la definició de diferents rols d'usuari, i permet limitar l'accés a les diferents pàgines a segons quins usuaris.

2.1.11 Els Entorns.

Hi ha dos arxius (`index.php` y `frontend_dev.php`) que són anomenats controladors frontals: totes les peticions a l'aplicació es fan a través d'ells. Però, per què hi ha dos controladors frontals si hem definit només una aplicació? Ambdós arxius apunten a la mateixa aplicació però per a diferents entorns.

Quan es desenvolupa una aplicació, excepte si es desenvolupa directament en el servidor de producció, són necessaris diversos entorns:

- **L'entorn de desenvolupament:** Aquest és l'ambient utilitzat per desenvolupadors web per a afegir noves funcions, corregir els errors, ...
En l'entorn de desenvolupament, l'aplicació necessita registrar tots els detalls d'una petició per a facilitar la depuració, ha de mostrar l'excepció en el navegador, però la *cache* ha de ser deshabilitada perquè tots els canvis realitzats al codi es tinguin en compte immediatament.
- **L'entorn de prova:** Aquest entorn s'utilitza per a provar automàticament l'aplicació.
- **L'entorn “staging”:** Aquest entorn és utilitzat pel client per a posar a prova l'aplicació i informar errors o característiques que falten.

- **L'entorn de producció:** Aquest és l'entorn on un usuari final interactua. En l'entorn de la producció, l'aplicació haurà de mostrar missatges d'error personalitzats en lloc d'excepcions, i per descomptat, la capa del *cache* ha d'estar activada. Aquest ha de ser optimitzat per al rendiment i l'experiència de l'usuari final.

Resumint: un “entorn Symfony” és un conjunt únic d'ajustaments de configuració. El *framework* Symfony inclou tres d'ells: *dev*, *test*, i *prod*.

2.1.12 - Internacionalització i Localització.

2.1.12.1- La cultura de l'usuari.

Les característiques i18n i l10n de Symfony es basen en la cultura de l'usuari. La cultura és la combinació del llenguatge i el país de l'usuari. Per exemple, la cultura per a un usuari que parla francès és fr i la cultura per a un usuari de França és fr_FR.

El llenguatge està codificat en dos minúscules, d'acord amb la *ISO 639-1 standard*, i el país està codificat amb dos caràcters en majúscula, d'acord amb la *ISO 3166-1 standard*.

2.1.12.2- Internacionalització.

Diferents idiomes tenen diferents conjunts de caràcters. L'Anglès és l'idioma més simple ja que només usa els caràcters ASCII, el català o el francès són un xic més complexes, amb caràcters accentuats com "à", i les llengües com el rus, xinès o àrab són molt més complexes que tots els seus caràcters ja que estan fora del rang ASCII. Aquests idiomes es defineixen amb diferents conjunts de caràcters.

Quan es tracta de dades internacionalitzades, és millor utilitzar la norma *Unicode*. La idea darrera de *Unicode* és establir un conjunt universal de caràcters que conté tots els caràcters de tots els idiomes. El problema que té és que un sol caràcter es pot representar amb una quantitat de 21 bits. Per tant, per a la web, vam usar UTF-8, que mapeja el codi *Unicode* apuntant-lo a una seqüència de longitud variable de octets.

2.1.12.3- Localització.

Suportar diferents cultures també significa suportar les diferents maneres de formatar dates i nombres. Per això existeixen plantilles on diversos helpers estan a disposició per ajudar a tenir en compte molts dels aspectes que es basen en l'actual cultura de l'usuari.

2.2- PHP POO.

El llenguatge de programació que inicialment es volia fer servir era PHP per la seva rapidesa i senzillesa, però quina versió? S'ha optat per la versió 5.3 perquè soluciona carències que tenen les anteriors versions en *POO*, i perquè és la compatible amb la última versió estable de Symfony.

PHP és un llenguatge molt utilitzat en el desenvolupament d'aplicacions web, és un llenguatge que porta actiu molts anys, i per els quals hi ha diferents servidors lliures multiplataforma i hi ha diferents *frameworks* que faciliten treballar-hi.

2.3- Sistema Gestor de Base de dades PostgreSQL.

PostgreSQL es un Sistema Gestor de Base de Dades molt potent. Té més de 15 anys de desenvolupament amb molta activitat, i una arquitectura més que provada que té molt bona reputació sobre fiabilitat, integritat de les dades i exactitud. Funciona en la majoria de sistemes operatius, incloent Linux, Unix (La majoria de versions) i Windows.

Es considerat el millor SGBD lliure del mercat. És considerat anàlog a Oracle, i encara ho hem de provar però ens hauria de permetre una carrega similar a Oracle, o sinó millor.

Té característiques molt sofisticades com per exemple el: *Multiversion Concurrency Control* (MVCC), *point in time recovery*, *tablespaces*, replicació asíncrona, *nested transactions* (punt de guardat o *savepoints*), backups en línia o un sofisticat planificador/optimitzador de consultes. Suporta sets de caràcters internacionals, codificació de caràcters *multibyte*, *Unicode*.

Té patents sobre concurrència de sistemes que no proporciona Oracle, (com un sistema per la concurrència sense elements serialitzables).

És escalable tant en quantitat de dades com en la quantitat de usuaris que poden treballar-hi. Hi ha sistemes muntats amb PostgreSQL en l'actualitat que gestionen més de 4TB de dades. Les limitacions més generals de PostgreSQL són les següents:

Límit	Valor
Mida màxima de la BBDD	Il·limitada
Mida màxima de Taula	32TB
Mida màxima de una fila	1.6TB
Mida màxima d'un camp	1GB
Files màximes per taula	Il·limitades
Camps màxims d'una taula	250-1600 (depenent del tipus de camps)
Nombre màxim de índexs per taula	Il·limitat

2.4- AJAX.

Ajax és un acrònim de *Asynchronous JavaScript and XML*, és una tècnica de desenvolupament web per crear aplicacions interactives o *RIA* (*Rich Internet Applications*). Aquestes aplicacions s'executen en el client, és a dir, en el navegador dels usuaris mentre es manté en comunicació asíncrona amb el servidor en segon pla.

D'aquesta manera és possible realitzar canvis sobre les pàgines sense necessitat de recarregar-les, el que significa augmentar la interactivitat, velocitat i usabilitat en les aplicacions.

Les aplicacions de la denominada *Web 2.0* inclouen nombroses interaccions en el costat del client, efectes visuals complexes i comunicacions asíncrones amb els servidors. Tot el que s'ha comentat es realitza amb *JavaScript*, però programar-lo manualment és tediós i requereix de molt de temps per corregir els possibles errors.

Symfony inclou una sèrie de *helpers* que automatitzen molts dels usos comuns de *JavaScript* en les plantilles. La majoria de comportaments en el costat del client es poden programar sense necessitat d'escriure gaires línies de codi *JavaScript*. Els programadors s'han de preocupar de l'efecte que volen incloure i *Symfony* s'encarrega de fer-ho amb la sintaxi necessària tenint en compte les possibles incompatibilitats entre navegadors.

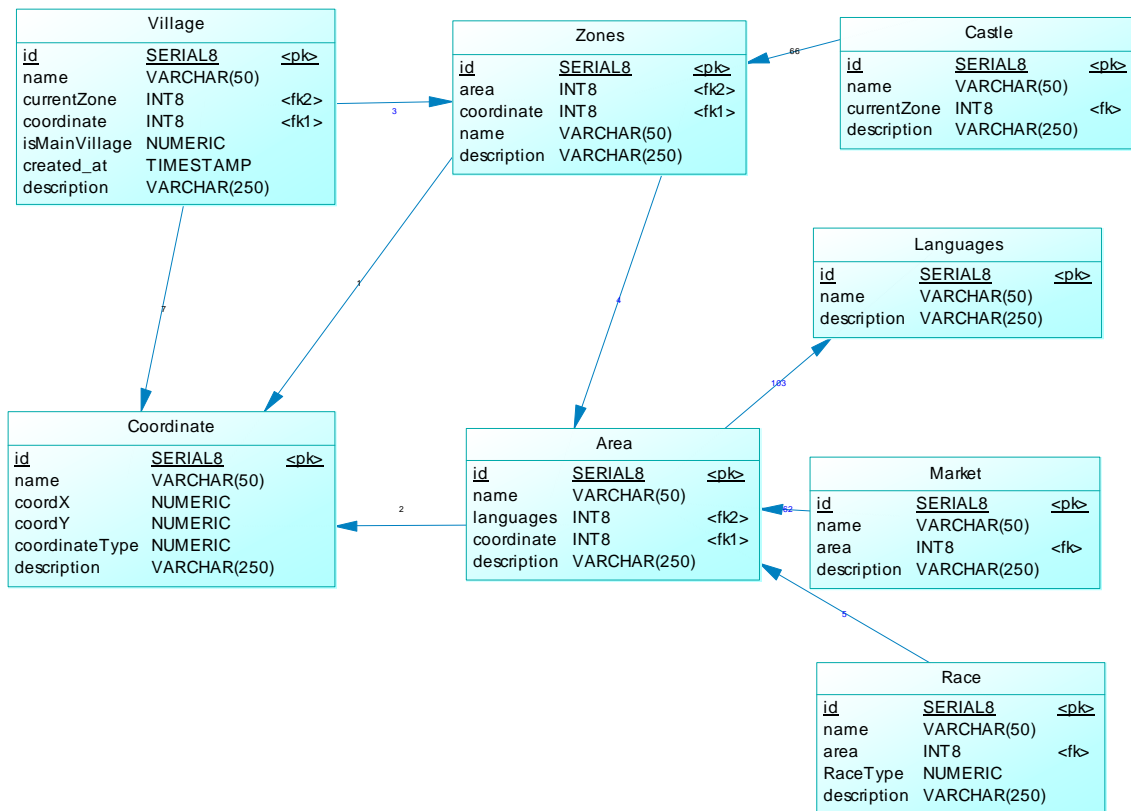
3- Desenvolupament.

El model de domini està detallat a l'Annex B.

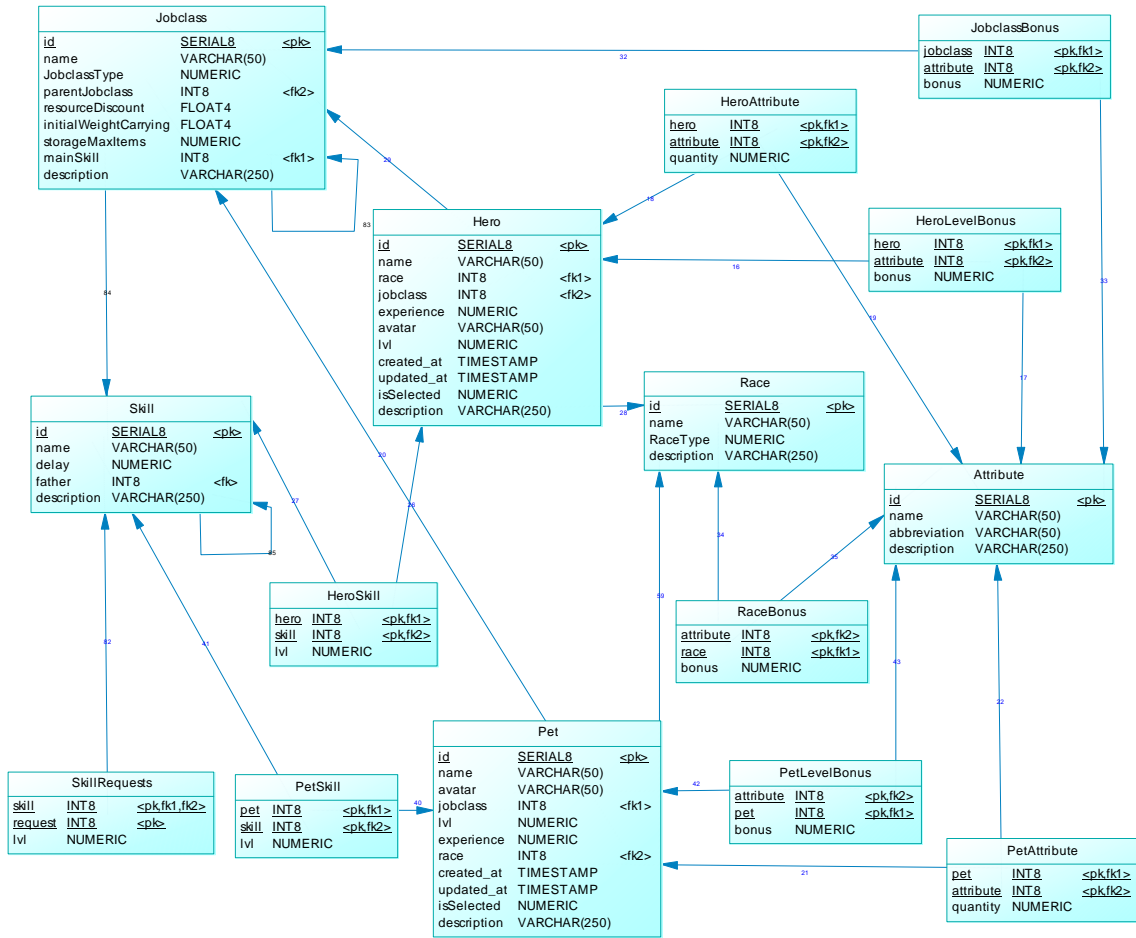
3.1- Model Relacional.

El diagrama d'entitat relació s'ha dividit en diversos submodels degut a que resulta complex de visualitzar-ho complet. Els submodels es detallen a continuació.

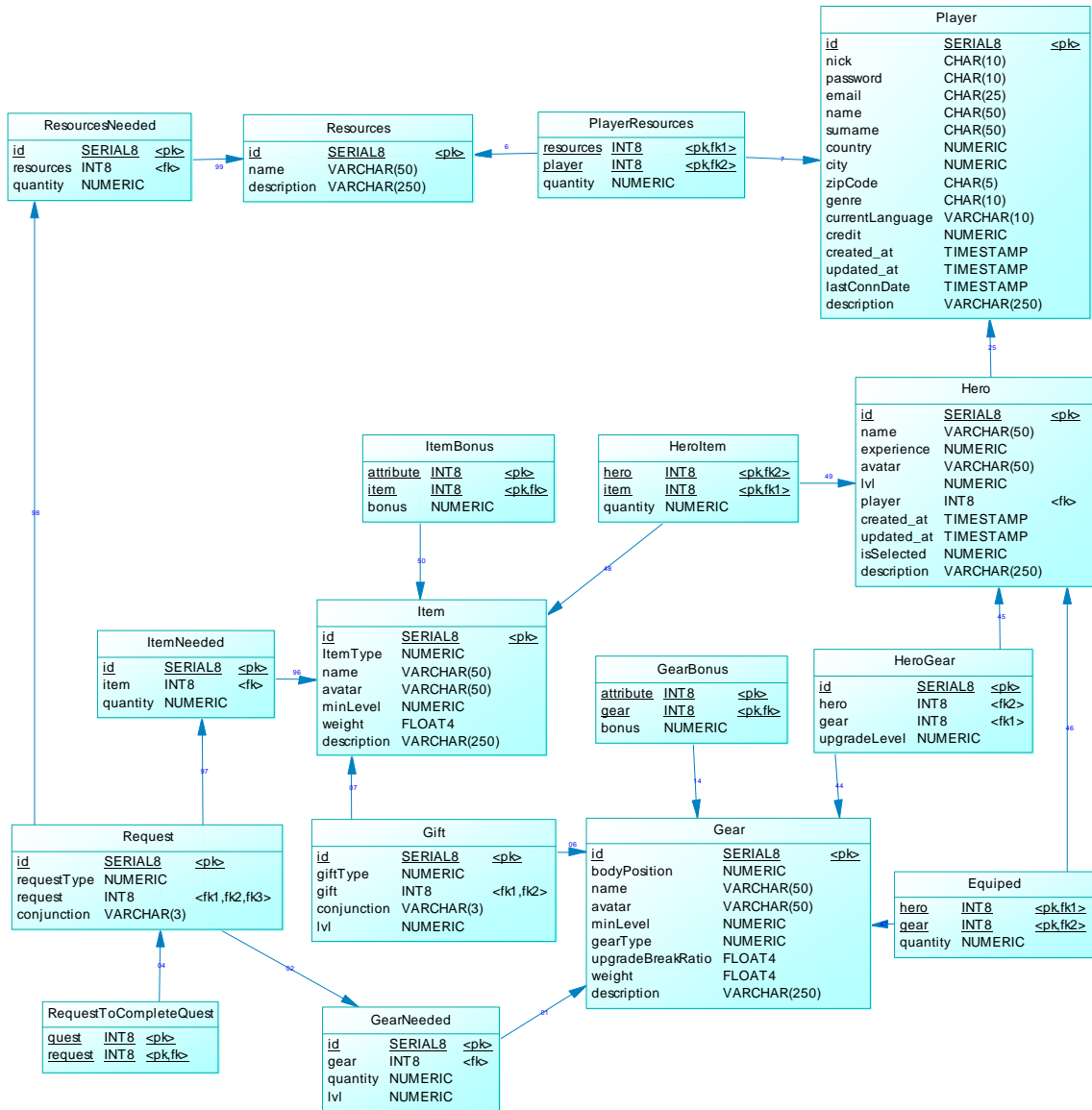
3.1.1- Submodel de Localització (location).



3.1.2- Submodel d'habilitats (skills).



3.1.3- Submodel de possessions materials.



3.2- Casos d'ús.**3.2.1- Gestió Jugador.****3.2.1.1- Alta Jugador.**

Nom	Alta Jugador.
Descripció	El Jugador es registra al Sistema.
Actors	Jugador.
Requisits	El Jugador ha de proporcionar les dades necessàries per realitzar el registre.
Restriccions	El Jugador no ha d'existir al sistema (no es pot repetir l'alias de Jugador ni el correu electrònic).
Resultat esperat	El Jugador és enregirat al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix al registre. 2. El Sistema demana les dades al Jugador. 3. El Jugador introdueix les dades i confirma l'alta. 4. El Sistema valida les dades. 5. S'executa el cas d'ús "Crear Heroi". 6. S'executa el cas d'ús "Crear Vila". 7. El Jugador és donat d'alta al Sistema. 8. El Sistema assigna els recursos al nou Jugador. 9. El Sistema confirma la correctesa de l'alta al Jugador.
Flux alternatiu	<ol style="list-style-type: none"> 4.1 Les dades introduïdes per el Jugador són incorrectes i el Sistema retorna error i torna al punt 2. 7.1. El cas d'ús "Crear Heroi" retorna error i torna al punt 5. 7.2. El cas d'ús "Crear Vila" retorna error i torna al punt 6. 9.1. El Sistema no pot donar d'alta al Jugador per x motiu. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.1.2- Modificar Jugador.

Nom	Modificar Jugador.
Descripció	El Jugador modifica les seves dades.
Actors	Jugador.
Requisits	El Jugador ha d'existir al Sistema.
Restriccions	Al modificar l'alias o el correu electrònic ha de seguir-se complint que no es poden repetir.
Resultat esperat	La informació del Jugador és modificada al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la modificació de les dades. 2. El Sistema mostra les dades actuals del Jugador de manera editable. 3. El Jugador modifica les dades i confirma la modificació. 4. El Sistema valida les dades. 5. El Sistema enregistra les dades. 6. El Sistema confirma la correctesa de la modificació.
Flux alternatiu	<ol style="list-style-type: none"> 4.1. Les dades són incorrectes i el Sistema retorna error. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.1.3- Baixa de Jugador.

Nom	Baixa de Jugador.
Descripció	El Jugador es dona de baixa al Sistema.
Actors	Jugador.
Requisits	El Jugador ha d'existir al Sistema.
Restriccions	El Jugador no es pot donar de baixa si està realitzant alguna interacció amb algun altre Jugador (combats, quests, etc.).
Resultat esperat	El Jugador es marca com a donat de baixa al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a donar-se de baixa. 2. El Sistema demana confirmació de la baixa sol·licitant contrasenya. 3. El Jugador introdueix la contrasenya i accepta. 4. El Sistema valida la contrasenya. 5. El Sistema marca com a donat de baix al Jugador i les seves viles, mascotes, herois, etc. 6. El Sistema confirma la correctesa de la baixa.
Flux alternatiu	<ol style="list-style-type: none"> 4.1. La contrasenya és incorrecta i el Sistema retorna error. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.1.4- Log In de Jugador.

Nom	Log In de Jugador.
Descripció	Quan un Jugador vol entrar al sistema, se li demana <i>login</i> : mail i contrasenya.
Actors	Jugador.
Requisits	El Jugador ha d'estar creat, i ha de coincidir el mail i la contrasenya.
Restriccions	Si el Jugador no recorda el mail o la contrasenya, pot demanar un recordatori, que se li enviarà al mail que te assignat a la seva conta de Jugador.
Resultat esperat	El sistema començarà una sessió per al jugador i cridarà el cas d'ús de: Tria de personatges, per que faci la tria.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador va a l'apartat de Log In de jugadors 2. El Jugador introdueix el mail i la contrasenya i clica à entrar. 3. El sistema crida el cas d'ús: <i>login</i>. 4. El sistema mostra al Jugador que està <i>loguejat</i> i quins són els personatges seleccionats.
Flux alternatiu	<p>2.1 El mail o la contrasenya són incorrectes, el sistema li indica al Jugador l'error, i li demana que ho introdueixi de nou. Si es la tercera vegada que s'equivoca, es posarà la conta en estat de <i>baneig</i> temporal de 1 hora. El sistema li enviarà un correu electrònic al Jugador indicant que hi ha hagut 3 intents erronis d'accés a la seva conta, amb un enllaç per tornar a activar la conta sense esperar 1 hora, per si ha sigut ell que ha fallat molt.</p> <p>3.1 El Jugador no té més de un heroi i una mascota, llavors no es crida el cas d'ús Tria de personatges, directament entra al joc amb els que té, posant-los com a equip de batalla seleccionat.</p> <p>3.2 El Jugador no té Heroi (pot haver-los esborrat tots...), llavors es cridarà el cas d'ús Crear Heroi.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.1.5- Log Out de Jugador.

Nom	<i>Logout</i> de Jugador.
Descripció	Un Jugador que està <i>loguejat</i> vol sortir de WarComrades
Actors	Jugador.
Requisits	El Jugador ha d'estar <i>loguejat</i> .
Restriccions	No s'observen restriccions.
Resultat esperat	El sistema <i>deslogueja</i> al Jugador correctament.
Flux normal	1. El Jugador clica sobre l'enllaç de log out 2. El sistema es fa log out del Jugador.
Flux alternatiu	No s'observen, l'únic requisit es estar <i>loguejat</i> , i si no es compleix el Jugador no podrà accedir a aquest cas d'ús. (*). El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.1.6- Assignar coordenades.

Nom	Assignar coordenades.
Descripció	Quan el Jugador es dona d'alta, crea la seva vila principal, aquesta vila ha de ser assignada per proximitat geogràfica amb els altres jugadors.
Actors	Sistema.
Requisits	El Jugador ha de proporcionar les dades necessàries per realitzar l'assignació de coordenades.
Restriccions	Aquest cas d'ús només es cridarà en el moment en que un Jugador s'està donant d'alta i se li vol ubicar en el mapa la seva vila principal.
Resultat esperat	Les coordenades de la nova vila són enregistrades al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El sistema està en procés de creació de Jugador-vila-heroi, amb les dades de La localització física real del Jugador, s'ha de "calcular" on s'ha d'ubicar la seva Vila principal en el mapa. 2. El Sistema calcula la proximitat de Jugadors amb localització similar en el mapa. 3. El Sistema assigna les coordenades concretes a la nova vila del nou Jugador.
Flux alternatiu	<ol style="list-style-type: none"> 2.1. no existeixen Jugadors amb localització similar. 2.2. per el motiu que sigui el sistema detecta varies opcions on ubicar <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.2- Gestió Viles.**3.2.2.1- Crear Vila.**

Nom	Crear Vila.
Descripció	El Jugador crea una nova vila.
Actors	Jugador.
Requisits	El Jugador ha de proporcionar les dades necessàries per realitzar la creació de la vila. El Jugador ha de disposar dels recursos necessaris per crear una vila.
Restriccions	El Jugador no pot sobrepassar el nombre límit de viles per Jugador amb la nova creació.
Resultat esperat	La nova vila es enregistrada al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la creació d'una nova vila. 2. El Sistema demana les dades necessàries al Jugador. 3. El Jugador introdueix les dades i confirma la creació. 4. El Sistema valida les dades. 5. El Sistema modifica les dades dels recursos del Jugador. 6. La nova vila és creada al Sistema. 7. El sistema crida el cas d'ús: assignar coordenades. 8. El sistema crida el cas d'ús: crear construccions.
Flux alternatiu	<p>4.1 Les dades introduïdes per el Jugador són incorrectes i el Sistema retorna error i torna al punt 2.</p> <p>4.2 El Jugador no té els recursos necessaris per crear la vila, el Sistema retorna l'error.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.2.2- Modificar Vila.

Nom	Modificar Vila.
Descripció	El Jugador vol modificar dades d'una vila que posseeix.
Actors	Jugador.
Requisits	El Jugador ha de proporcionar les dades necessàries per realitzar la modificació.
Restriccions	El Jugador no pot modificar dades restringides (coordinades i altres), només es podran modificar dades tipus: nom de la vila, descripció, etc.
Resultat esperat	Els canvis en la vila són enregistrats per el sistema i el fet es notificat al jugador.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la modificació de les dades d'una de les seves viles. 2. El Sistema demana les dades necessàries al Jugador i mostra un botó de acceptar. 3. El Jugador introdueix les dades i accepta la modificació. 4. El Sistema valida les dades. 6. Les noves dades són enregistrades al sistema.
Flux alternatiu	<ol style="list-style-type: none"> 4.1 Les dades introduïdes per el Jugador són incorrectes i el Sistema retorna error i torna al punt 2. 4.2 El Jugador no té els recursos necessaris per crear la vila, el Sistema retorna l'error. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.2.3- Eliminar Vila.

Nom	Eliminar Vila.
Descripció	El Jugador vol eliminar una vila que posseeix.
Actors	Jugador.
Requisits	El Jugador ha de proporcionar confirmar amb dades personals la eliminació.
Restriccions	El Jugador no pot eliminar la seva vila principal. El jugador no pot eliminar la seva vila si està rebent un atac, o si amb una vila està enviant un atac, o interaccionant d'alguna altre manera amb la vila cap a un altre jugador.
Resultat esperat	La eliminació de la vila es enregistrada i el fet es confirmat al jugador
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la eliminació d'una de les seves viles. 2. El Sistema mostra les dades de la vila al Jugador i mostra un botó de Eliminar. 3. El Jugador clica Eliminar. 4. El Sistema demana confirmació introduint el correu electrònic i la contrasenya. 5. El Jugador introdueix les dades i confirma la eliminació 6. El sistema elimina la vila i ho confirma al Jugador.
Flux alternatiu	<ol style="list-style-type: none"> 1.1 El jugador no clica Eliminar. 4.1 El sistema torna error perquè la vila es la vila principal del jugador. 5.1 Les dades introduïdes per el jugador són incorrectes, a la tercera vegada que es compleixi aquest punt, la conta passarà a tenir un ban de 1 hora i s'enviarà un correu electrònic a la conta del jugador informant del intent de eliminar una vila. Amb un enllaç per poder activar la conta immediatament, per si es ell realment però ha introduït malament les dades. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.2.4- Fer Investigació.

Nom	Fer Investigació.
Descripció	Un jugador vol fer una investigació fent servir la/les construcció/ns que ho permeten de alguna de les seves viles.
Actors	Jugador.
Requisits	El jugador ha de disposar de la/les construcció/ns necessària/es per investigar.
Restriccions	Només es podrà fer una investigació per vila alhora.
Resultat esperat	El Jugador realitza la investigació i obté els seus. Si el quest requeria objectes o elements, li seran descomptats de la seva "motxilla".
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica en el edifici de investigacions. 2. El sistema li ensenya les investigacions que pot començar. 3. El Jugador li indica quina investigació vol. 4. El sistema li descompta els elements o recursos necessaris per iniciar la investigació, i li indica al jugador que ha començat a investigar, indicant-li el temps de demora. 5. Quan el sistema detecta que el temps de realitzar la investigació ha acabat, li indica al jugador que ha finalitzat amb un missatge a la bústia.
Flux alternatiu	<ol style="list-style-type: none"> 2.1 El jugador no compleix els requisits de cap investigació. 5.1 El Jugador cancel•la la investigació. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.2.5- Millora de Construccions.

Nom	Millora de Construccions.
Descripció	Un jugador vol fer una construcció, ja sigui fer el primer nivell d'una construcció de la qual no te edificació, o vol fer un nivell més a una edificació que ja te construïda.
Actors	Jugador.
Requisits	El jugador ha de complir els requisits que demani la construcció en qüestió que vulgui realitzar, ja sigui de nivell d'alguna construcció, de nivell d'heroi.. o mascota..., raça, professió, possessió d'elements o equipacions, habilitats (<i>skills</i>), etc.
Restriccions	Només es podrà fer una construcció alhora per vila.
Resultat esperat	El Jugador acaba la construcció i obté els beneficis de la mateixa.
Flux normal	<ol style="list-style-type: none"> 1. Jugador clica en el apartat de construccions. 2. El sistema li mostra les construccions de la vila. 3. El Jugador li indica que vol realitzar la construcció d'un edifici concret. 4. El sistema redueix els recursos o elements necessaris per fer la construcció, i li indica amb un temporitzador el temps que falta per la finalització. 5. El sistema detecta que la construcció ha finalitzat i li envia un missatge a la bústia al jugador indicant-li.
Flux alternatiu	<p>5.1 El jugador cancel·la la construcció abans de la finalització, el sistema la cancel·la.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.3- Gestió atacs/moviments mapa.

3.2.3.1- Especificar ordre d'habilitats i elements.

Nom	Especificar ordre d'habilitats i elements.
Descripció	El Jugador configura les habilitats a usar en combat.
Actors	Jugador.
Requisits	Les habilitats/elements usables a configurar han de pertànyer a l'heroi. Hi ha d'haver espais d'habilitats per poder posar-n'hi una.
Restriccions	Un heroi no pot usar habilitats/elements usables que no ha après/obtingut. No es pot equipar una habilitat/element usable si la taula d'habilitats configurades ja és plena.
Resultat esperat	El Jugador té configurat l'ordre d'atac.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a configurar les habilitats o elements usables per el combat. 2. El Sistema mostra la taula corresponent amb les habilitats/elements usables possibles a configurar. 3. El Jugador configura una de les habilitats/elements usables en l'ordre que els vol usar. 4. El Sistema afegeix l'habilitat/element usable seleccionat a la taula de configuració. 5. El Jugador accepta la configuració. 6. El Sistema guarda la configuració i informa al Jugador.
Flux alternatiu	<ol style="list-style-type: none"> 1.1 La taula de configuració és plena i el Sistema retorna error. 5.1 El Jugador vol afegir mes elements/habilitats, torna al punt 3. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.3.2- Enviar atac a zona.

Nom	Enviar atac a zona.
Descripció	El Jugador envia un atac a una zona.
Actors	Jugador.
Requisits	El Jugador ha d'estar a la zona o bé que aquesta sigui accessible per a l'heroi.
Restriccions	Un heroi no pot atacar una zona on no hi pot accedir.
Resultat esperat	El Jugador ataca un zona.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a atacar una zona. 2. El Sistema fa els càlculs corresponents i modifica les dades(càlculs del combat). 3. El Sistema executa el cas d'ús "Repartir <i>gifts</i>" i ho informa al Jugador .
Flux alternatiu	<ol style="list-style-type: none"> 2.1 El Jugador no es troba a la zona a atacar i el Sistema executa el cas d'ús "Viatjar a zona" abans de procedir el cas d'ús actual. 2.2 El Jugador perd la batalla, el Sistema el retorna al mercat visitat més pròxim i n'informa al Jugador. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.3.3- Viatjar a zona.

Nom	Viatjar a zona.
Descripció	El Jugador selecciona viatjar a una Zona.
Actors	Jugador.
Requisits	La zona ha de ser accessible per el Jugador.
Restriccions	Un heroi no pot viatjar a una zona on no ha atacat prèviament.
Resultat esperat	El Jugador viatja a una zona.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a atacar una zona. 2. El Sistema fa els càlculs corresponents, modifica les dades i informa al Jugador el resultat de la batalla. 3. El Jugador confirma el resultat de la batalla.
Flux alternatiu	<p>2.1 El Jugador perd la batalla, el Sistema el retorna al mercat visitat més pròxim i n'informa al Jugador.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.3.4- Veure mapa.

Nom	Veure mapa.
Descripció	El Jugador selecciona que vol veure el mapa.
Actors	Jugador.
Requisits	El Jugador ha de tenir seleccionat almenys l'heroi de l'equip de batalla.
Restriccions	És necessari tenir instal·lat flash en el navegador del client.
Resultat esperat	El Jugador visualitza el mapa del mon de WarComrades.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica en el mapa per visualitzar-lo 2. El Sistema carrega el mapa de WarComrades a alt nivell (àrees i zones)
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.4- La millora.**3.2.4.1- Millorar equipament(Upgrade).**

Nom	Millorar equipament (<i>Upgrade</i>).
Descripció	Un jugador vol millorar un equipament que posseeix mitjançant un WCC.
Actors	Jugador, WCC de millores.
Requisits	El jugador ha de disposar de les equipacions que vol provar de millorar, i els elements necessaris per realitzar el intent de millora.
Restriccions	Només es podrà millorar una equipació alhora, només ho podrà fer un WCC de millores.
Resultat esperat	El sistema fa la millora o la fallida i li deixa o destrueix la equipació respectivament, sempre li consumeix els elements necessaris per la prova de millora.
Flux normal	<ol style="list-style-type: none"> 1. Jugador clica en el WCC de millores. 2. El WCC li dona la benvinguda, i li indica que li pot fer millores, li demana quin objecte vol millorar. 3. El Jugador clica la equipació que vol millorar, i li indica de fer la millora. 4. El WCC li demana confirmació per fer el intent de millora de la equipació seleccionada. 5. El Jugador confirma l'intent. 6. El WCC fa l'intent de millora. 7. El Sistema li augmenta el nivell de millora a la equipació , o li fa desaparèixer, segons si ha obtingut <i>Success</i>, o <i>Failed</i>.. 8. El WCC li demana si vol seguir fent proves de millores o si vol sortir. 9. El Jugador li indica que vol seguir fent millores, i es torna al punt 3.
Flux alternatiu	<p>3.1 El Jugador no te els elements necessaris per fer la millora, el WCC li indica i va al punt 8</p> <p>5.1 El Jugador cancel•la l'intent.</p> <p>9.1 El Jugador indica que vol sortir de les millores.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.4.2- Encantar equipament.

Nom	Encantar equipament.
Descripció	Un jugador vol encantar un equipament que posseeix mitjançant un WCC.
Actors	Jugador, WCC de encantaments.
Requisits	El jugador ha de disposar de les equipacions que vol encantar, i els elements necessaris per realitzar l'encantament.
Restriccions	Només es podrà encantar una equipació alhora, només ho podrà fer un WCC de encantaments.
Resultat esperat	El sistema fa el encantament o el failed i li deixa encantat o no, sempre li consumeix els elements necessaris per la prova d'encantament.
Flux normal	<ol style="list-style-type: none"> 1. Jugador clica en el WCC de encantaments 2. El WCC li dona la benvinguda, i li indica que li pot fer encantaments, li demana quin objecte vol encantar. 3. El Jugador clica l'equipació que vol encantar, i li indica quin encantament vol. 4. El WCC li demana confirmació per fer l'intent d'encantament de l'equipació seleccionada 5. El Jugador confirma l'intent. 6. El WCC fa l'intent d'encantament. 7. El Sistema li encanta l'equipació, o no, segons si ha obtingut Success, o Failed.. i sempre li desconta el/s element/s necessaris per fer l'intent. 8. El WCC li demana si vol seguir fent proves d'encantament o si vol sortir. 9. El Jugador li indica seguir fent encantaments, i tornem al punt 3.
Flux alternatiu	<ol style="list-style-type: none"> 4.1 El Jugador no te els elements necessaris per fer l'encantament, el WCC li indica i va al punt 8 5.1 El Jugador cancel·la el intent. 9.1 El Jugador indica que vol sortir dels encantaments. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.4.3- Realitzar Missió (quest).

Nom	Realitzar Missió (quest).
Descripció	Un jugador vol realitzar una Missió.
Actors	Jugador.
Requisits	El jugador ha de disposar dels requisits que demani la missió en qüestió que vulgui realitzar, ja siguin de: nivell necessari, raça, professió, possessió de elements o equipacions, habilitats d' <i>skill</i> , etc.
Restriccions	Només es podrà fer una Missió alhora.
Resultat esperat	El Jugador realitza la Missió i obté els beneficis de la mateixa. Si es requerien objectes o elements, li seran descomptats de la seva "motxilla".
Flux normal	<ol style="list-style-type: none"> 1. Jugador clica en el WCC de missions 2. El WCC li dona la benvinguda, i li indica que pot començar missions amb ell, li demana quina vol realitzar. 3. El Jugador li indica quina missió vol. 4. El WCC li demana confirmació per començar la missió. 5. El jugador confirma que vol començar. 6. El WCC redueix els recursos o elements necessaris per la missió del jugador, i li demana al jugador fer les proves per la missió (proves, preguntes, etc..) 7. El jugador realitza les proves o preguntes de la missió, i li indica al WCC. 8. El WCC li atorga al jugador el premi del quest realitzat, i finalitza la missió per el jugador. 9. El WCC li demana si vol seguir fent missions. 10. El Jugador li indica seguir fent missions, i tornem al punt 3.

Flux alternatiu	<p>1.1 el WCC no és de missions.</p> <p>3.1 El jugador indica que vol sortir sense fer cap missió dels que té.</p> <p>5.1 El Jugador cancel·la el començament de la missió.</p> <p>6.1 El WCC indica al jugador que no compleix els requisits per començar la missió que ha seleccionat.</p> <p>7.1 El jugador no pot realitzar la missió i li indica al WCC que vol cancel·lar-la, el missió finalitza sense cap benefici per el jugador.</p> <p>10.1 El jugador li indica que vol deixar de fer missions i vol sortir, el WCC es despedeix.</p>
------------------------	---

3.2.5- Consultes.

3.2.5.1- Cerca de Jugador.

Nom	Cerca de Jugador.
Descripció	Un jugador vol consultar la informació disponible d'un altre jugador.
Actors	Jugador1, Jugador2.
Requisits	El jugador ha de conèixer el l'alias del jugador que vol cercar.
Restriccions	Només es podrà cercar a un jugador alhora.
Resultat esperat	El Jugador troba la informació del jugador que cercava.
Flux normal	<ol style="list-style-type: none"> 1. Jugador1 clica en el apartat de jugadors? 2. El sistema li demana l'alias a cercar 3. El Jugador1 li indica l'alias del jugador2 i clica cercar. 4. El sistema li mostra la informació de que disposa al jugador2.
Flux alternatiu	4.1 El jugador2 no existeix. (*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.2- Consultar Heroi.

Nom	Consultar Heroi.
Descripció	Un jugador vol consultar la informació de l'heroi que té seleccionat.
Actors	Jugador.
Requisits	El jugador ha de tenir un heroi seleccionat.
Restriccions	Aquest cas d'ús està restringit per els herois del propi jugador.
Resultat esperat	El Jugador troba la informació del heroi que ha indicat.
Flux normal	1. El Jugador clica en el botó de informació del seu heroi. 2. El sistema li mostra la informació del heroi que ha indicat.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.3- Consultar Mascota.

Nom	Consultar Mascota.
Descripció	Un jugador vol consultar la informació del mascota que te seleccionat.
Actors	Jugador.
Requisits	El jugador ha de tenir un mascota seleccionat.
Restriccions	Aquest cas d'ús està restringit per les mascotes del propi jugador.
Resultat esperat	El Jugador troba la informació del mascota que ha indicat.
Flux normal	1. El Jugador clica en el botó de informació del seu mascota. 2. El sistema li mostra la informació del mascota que ha indicat.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.4- Consultar Viles.

Nom	Consultar Viles.
Descripció	Un jugador vol consultar la informació de les viles que té.
Actors	Jugador.
Requisits	El jugador ha de tenir viles.
Restriccions	Aquest cas d'ús està restringit per les viles del propi jugador.
Resultat esperat	El Jugador troba la informació de totes les seves viles.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica en el botó de informació de les viles. 2. El sistema li mostra la informació de les viles (un llistat amb les viles que posseeix, amb coordenades, nom, situació, producció i algun vincle per fer servir d'accés ràpid)
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.5- Consultar Construccions Vila.

Nom	Consultar Construccions Vila.
Descripció	Un jugador vol consultar les construccions d'una de les seves viles.
Actors	Jugador.
Requisits	El jugador ha de tenir viles.
Restriccions	Aquest cas d'ús està restringit per les viles del propi jugador.
Resultat esperat	El Sistema li mostra les construccions de la vila seleccionada al jugador.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica sobre d'una vila. 2. El sistema li mostra que ha entrat a la vila i li mostra les construccions.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.6- Consultar Investigacions Vila.

Nom	Consultar Investigacions Vila.
Descripció	Un jugador vol consultar les investigacions d'una de les seves viles.
Actors	Jugador.
Requisits	El jugador ha de tenir viles, ha de tenir la/les construcció/ns que li permet/en investigar.
Restriccions	Aquest cas d'ús està restringit per les viles del propi jugador.
Resultat esperat	El Sistema li mostra les investigacions de la vila seleccionada al jugador.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador estant dintre d'una vila, clica en l'edifici/s de les investigacions en el botó de: veure investigacions 2. El sistema li mostra les investigacions de la vila.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.7- Consultar Ranking.

Nom	Consultar Ranking.
Descripció	Un jugador vol consultar el ranking, per nivells de heroi, mascotes o nombre de viles.
Actors	Jugador.
Requisits	El jugador ha de seleccionar un dels rankings per visualitzar.
Restriccions	No s'observen.
Resultat esperat	El Sistema li mostra el ranking.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica <i>rankings</i>, i per defecte li surt el de nivell 2. El Jugador selecciona el <i>ranking</i> que vol veure i clica rank 3. El sistema mostra el resultat de la consulta en l'apartat de <i>rankings</i>.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

3.2.5.8- Consultar combats realitzats.

Nom	Consultar combats realitzats.
Descripció	Un jugador vol consultar els combats que ha realitzat el/s seu/s heroi/s.
Actors	Jugador .
Requisits	El jugador ha de tenir combats realitzats.
Restriccions	El nombre de combats que emmagatzema el sistema és limitat.
Resultat esperat	El Sistema li mostra el llistat de combats que ha fet.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica veure combats. 2. El sistema fa la consulta, i li mostra en pantalla els últims combats realitzats ordenats cronològicament de forma creixent.
Flux alternatiu	<p>2.1 El jugador no te combats realitzats, o ha esborrat els informes, i el sistema li indica que no hi ha informes de combats.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.2.5.9- Consultar el Mercat.

Nom	Consultar el Mercat.
Descripció	Un jugador vol consultar un mercat del joc.
Actors	Jugador .
Requisits	El Heroi del jugador ha de estar en un mercat.
Restriccions	El Heroi seleccionat ha de tenir après la llengua del mercat.
Resultat esperat	El Sistema li mostra el mercat.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica sobre el mercat. 2. El sistema li mostra el mercat.
Flux alternatiu	<p>2.1. El Heroi del jugador no coneix el llenguatge del mercat, i només pot veure els WCC.</p> <p>2.2. El Heroi del joc no està a la mateixa zona que el mercat.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

3.3- Interfície gràfica.

3.3.1- Accessibilitat.

La web és compatible i s'ha testejat amb: Mozilla Firefox, Internet Explorer, Safari, i Opera. El joc passa el tests de la W3C referents a l'*HTML* i al *CSS*, garantint la màxima accessibilitat en la majoria de navegadors en les seves diverses versions.

La interfície del joc és multi idioma, com a llengua per defecte té l'anglès, però si detecta algun altre dels contemplats mostra la web en la llengua detectada, encara que per el moment només s'ha afegit la traducció a l'espanyol.

3.3.2- Consistència.

Sempre que es va a fer alguna addició, modificació o eliminació es torna a comprovar que encara sigui possible, que segueixi respectant les regles, ja que el moment de càrrega de la pantalla del cas d'ús no serà segurament el mateix moment que quan s'executin modificacions, és a dir, per exemple:

L'usuari carrega mercat, per fer la carrega s'han calculat entre d'altres coses, els recursos del jugador. Si es disposa a comprar, quan es vagi a fer l'últim pas de la compra, es torna a comprovar que encara disposi dels diners i que el mercat encara tingui els recursos suficients per fer la venda, i que el preu no hagi variat, sinó es el cas, se li informa amb un missatge.

3.3.3- Interactivitat.

Mitjançant text de recolzament se li dona informació al jugador fent aparèixer la informació de l'element sobre el que tingui el punter del ratolí. En els formularis se li mostra al jugador dinàmicament la informació de les dades que està manipulant.

El jugador pot configurar la ubicació i importància de les diferents parts de la web i es desa de manera persistent.

La interfície es pot veure en mode reduït, per a jugadors experts, per potenciar la dinàmica entre el jugador i l'aplicació.



Captura de pantalla de WarComrades en la que es mostra un canvi en la configuració de la visualització de la pantalla (construccions).

3.3.4- Tecnologia.

La interfície gràfica està feta amb *HTML*, *CSS* i *JavaScript*, generat amb *php5*, s'ha fet servir el *framework* *Prototype* per la implementació de la majoria de la interactivitat gràfica en *JavaScript*.

Els elements del lloc web: menú, utilitats, accessos, seccions, contingut, etc. es generen dinàmicament de la persistència, és a dir, per exemple: si s'ha de mostrar viles, es mostraran tantes com tingui el jugador.

Cada secció de la web i els apartats que conté estan preparats per poder ser actualitzats dinàmicament amb *AJAX* independentment de la resta de la web.



Captura de pantalla de WarComrades (unitats de les viles).



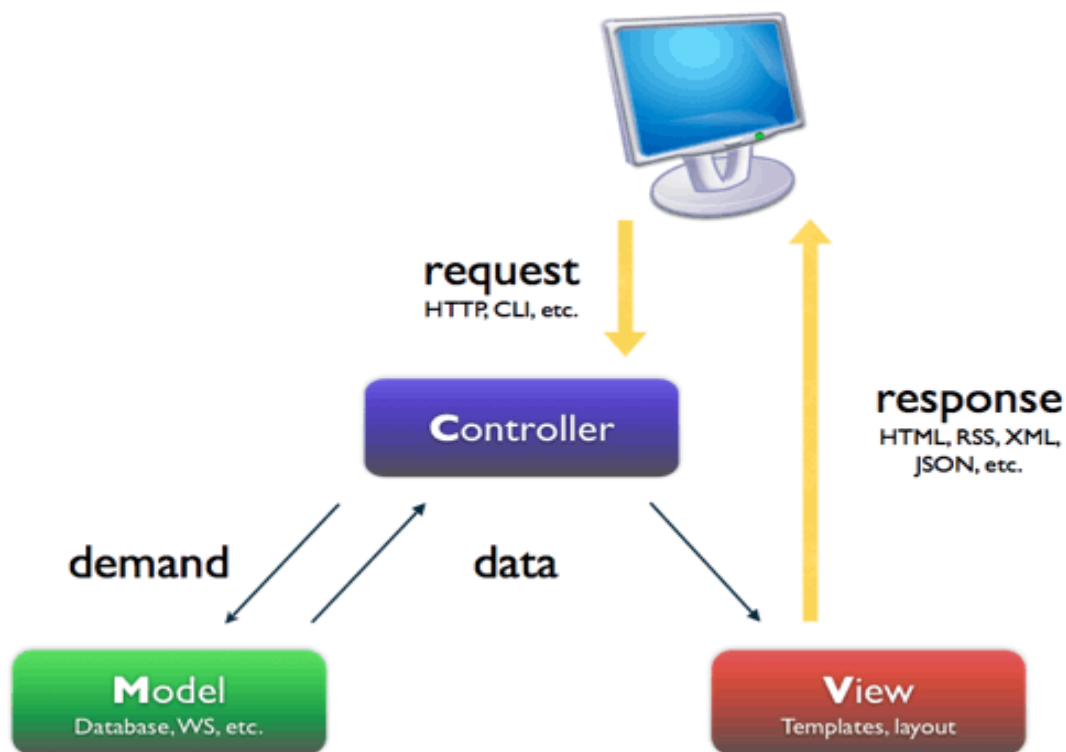
Captura de pantalla de WarComrades (mercat).

3.4- L'arquitectura.

3.4.1- Patró: MVC.

Per a desenvolupament web, la solució més comuna per a l'organització del seu codi d'avui dia és el patró de disseny Model Vista Controlador (MVC). En resum, el patró de disseny MVC defineix una manera d'organitzar el codi d'acord a la seva naturalesa. Aquest patró separa el codi en **tres capes**:

- **La capa Model** defineix la lògica de negoci (la base de dades pertany a aquesta capa).
- **La Vista** és amb el que l'usuari interactua (un motor de plantilles és part d'aquesta capa). En Symfony, la vista és principalment la capa de plantilles PHP.
- **El Controlador** és la peça de codi que crida al Model per a obtenir algunes dades que li passa a la Vista per a la presentació al client. Les sol·licituds són gestionades per un controlador frontal que delega la veritable labor a les accions. Aquestes accions són, lògicament, agrupades en mòduls.



3.4.2- Patró creador.

A la classes d'accions (*Actions*) es tenen les accions definides per cada mòdul. En les accions es creen els objectes de classe que representen les entitats.

La classe *Actions* és la “creadora” d'aquestes entitats.

3.4.3- Patró expert.

Aquest és molt utilitzat en el projecte, ja que Propel és la llibreria externa que s'utilitza per realitzar la capa d'abstracció de dades en el model, encapsula tota la lògica de dades i aquestes són generades les classes amb totes les funcionalitats comuns de les entitats.

3.4.4- Patró Controlador Frontal.

Totes les peticions web són gestionades per un sol controlador frontal (*sfActions*), que es l'únic punt d'entrada de tota la aplicació per un determinat entorn.

Quan el controlador frontal rep una petició, utilitza el sistema d'enrutament per associar el nom d'una acció i el nom d'un mòdul amb la *URL* introduïda per l'usuari.

3.4.5- Singleton.

Assegura l'existència d'una única instància per una classe i la creació d'un mecanisme d'accés global a aquesta instància.

En el controlador frontal hi ha una crida a *sfContext::getInstance()*. En una acció, el mètode *getContext()*, un objecte molt útil que emmagatzema una referència a tots els objectes del nucli del *framework*.

3.4.6- Patró Factoria.

Permet treballar amb objectes de diferents famílies de manera que aquestes no es barregen entre si, fent transparent el tipus de família concreta que s'estigui fent servir.

Quan es necessita per exemple crear un nou objecte per una petició, es cerca en la definició de la factoria el nom de la classe que s'ha de fer servir per la tasca. Un exemple en el projecte és *sfUser*.

3.4.7- Patró Decorador.

La gran part de cadascuna de les pàgines d'una web té el mateix aspecte. Ja que la duplicació de codi no és aconsellable en absolut, ja sigui codi *HTML* o *PHP*, necessitem trobar una manera de prevenir aquests elements de vista comuna resultants de la duplicació de codi.

Per solucionar el problema hem utilitzat un altre patró de disseny: el **patró de disseny decorador**, que resol el problema al revés:

El contingut de la plantilla s'integra en el *layout*, o mirat des d'un altre punt de vista, el *layout* decora la plantilla.

Aquest comportament es una implementació del patró de disseny anomenat: Composite.

3.4.8- Patró Composite.

Permet tractar objectes compostos com si es tractés d'un de simple. Serveix per construir objectes complexes a partir d'altres simples i similars entre ells, gràcies a la composició recursiva i a una estructura en forma d'arbre. Això simplifica el tractament dels objectes creats, ja que al disposar tots ells d'una interfície comuna, es tracten tots ells de la mateixa forma. Un exemple és la manera en que *Symfony* recupera els objectes en el model des de la persistència.

3.4.9- Patró Command.

Patró de disseny que es usat per representar i encapsular tota la informació requerida per fer la crida a un mètode. Aquesta informació inclou el nom del mètode i l'objecte al que pertany i valors dels seus paràmetres.

Tres termes relacionats amb aquest patró acostumen a ser: client, invocador (*invoker*) i receptor (*receiver*).

El client instància l'objecte *command* i facilita la informació requerida per cridar el mètode a posteriori. El que invoca decideix quan el mètode ha de ser cridat. El receptor és una instància de la classe que conté el codi del mètode.

Usant aquest patró es facilita la construcció de components generals que necessiten delegar, seqüències o executar crides a mètodes sense necessitat de saber el propietari o els paràmetres necessaris.

3.4.10- Patró Filtre Interceptor (*Intercepting Filter Pattern*).

Es disposa de filtres connectables per processar serveis comuns de forma estàndard sense requerir canvis de nucli del codi del processament de la petició. Els filtres interceptors agafen l'entrada de les peticions (*requests*) i donen com a sortida les respostes (*responses*) permetent pre-processament i post-processament. S'afegiran i es treuran sense necessitar canvis en el codi existent.

El processament principal és completat amb una varietat de serveis comuns com: seguretat, *login*, depuració, etc.

Aquests filtres són components independents de codi de l'aplicació i s'afegeixen o treuen de forma declarativa.

3.4.11- Patró Vista en dos passos (*Two-Step View Pattern*).

Transforma les dades del model a *HTML* en dos passos, el primer és crear la presentació lògica de les dades del model sense especificar formats, el segon és convertir la presentació lògica amb el format que es requereixi.

Això permet, entre altres coses, fer canvis globals a un entorn sense alterar els altres.

3.4.12- Patró ajudant de la vista. (*View Helper Pattern*)

Els canvis de la capa de presentació són continus i complexes de desenvolupar i mantenir, degut a la combinació de la lògica del negoci per l'accés a les dades i la lògica del formateig de la presentació, això fa que el sistema sigui menys flexible, menys reutilitzable, i menys adaptable als canvis.

Aquí és on prenen un paper important els *helpers*. La vista delega la gestió del contingut als seus ajudants (*helpers*), que serveixen com a model de dades i adaptadors per les dades. La lògica de la presentació és encapsulada en l'ajudant i es situa entre la vista i la capa de negoci.

3.4.13- Patró de porta d'enllaç d'una taula de base de dades. (*Table Data Gateway Pattern*)

Mesclar SQL en la lògica de l'aplicació pot causar molts problemes. Molts desenvolupadors no es troben còmodes desenvolupant amb SQL, o simplement no l'escriuen bé. L'administrador de *BBDD* necessita que l'SQL es pugui trobar de forma senzilla per poder fer *tunning* de la *BBDD* per fer-la evolucionar.

Es crea un objecte que actua de *gateway* a una taula de la base de dades. Es requereix una única instància per gestionar totes les files de la taula.

Amb aquesta instància es pot accedir a una taula simple o vista i fer: *selects*, *inserts*, *updates* o *deletes*. Que seran cridats des de mètodes d'altres classes.

3.4.14- Patró de Registre Actiu. (*Active Record Pattern*).

Es té un objecte que fa referència a una registre de una taula o de una vista de la base de dades, encapsula l'accés a la base de dades, i afegeix lògica del domini a les dades.

L'objecte carrega les dades i el comportament. Gran part de les dades és persistent i necessita estar desada en una base de dades. Aquest patró fa servir l'aproximació més òbvia, posant lògica de l'accés de dades en el objecte del domini. D'aquesta manera tot desenvolupador pot saber com llegir i escriure les dades en la base de dades.

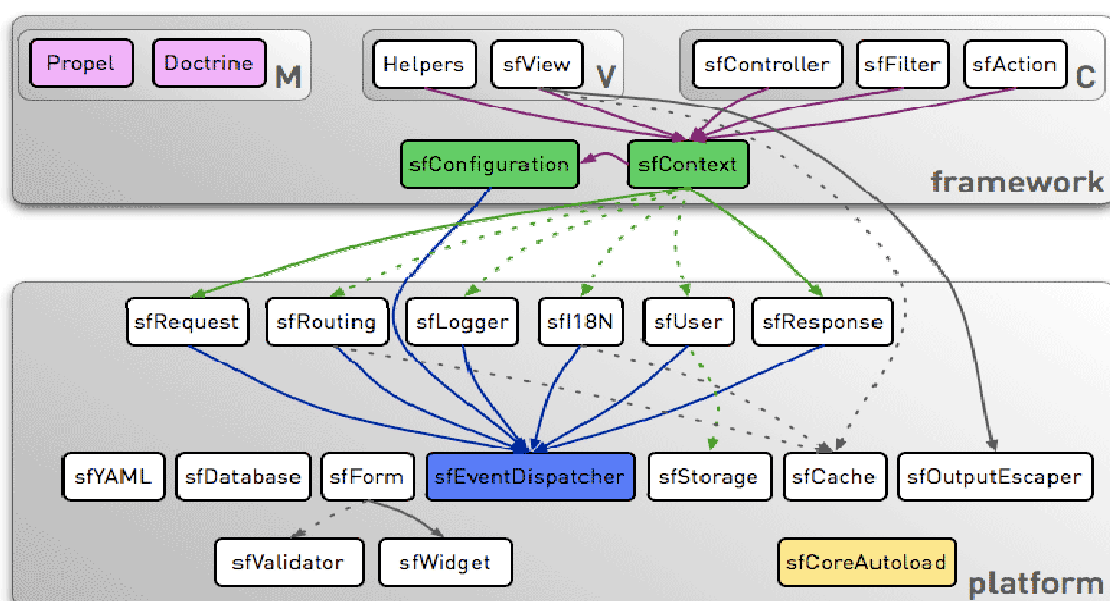
3.4.15- Patró herència simple d'una taula. (*Single Table Inheritance Pattern*).

Les bases de dades relacionals no suporten la herència, quan es mapejen els objectes d'una base de dades s'ha de considerar com representar les complexes estructures d'herències en taules relacionals.

Quan es mapeja una base de dades relacional, s'ha de minimitzar les relacions (*joins*) que fàcilment van augmentant a mesura que es processa una estructura d'herència amb múltiples taules.

Aquest patró mapeja tot els atributs de totes les classes mantenint una estructura jeràrquica en una sola classe. Representa la jerarquia d'herència de classes com una única classe que té tot els atributs de les taules que involucra.

3.4.16- Diagrama de l'arquitectura.



4- Actualitat del projecte.

4.2- Implementacions pendents.

Aquest apartat està detallat a l'Annex D.

4.2.1- Previsió a curt termini.

Una vegada presentat el projecte, es vol fer una posta en comú de totes els aspectes rellevants que hem trobat desenvolupant o aprenent durant el projecte. I s'ha de parlar sobre la següent iteració, i estudiar el temps necessari, i la viabilitat.

S'ha determinat que la propera iteració serà la versió alfa tancada ja penjada.

Comprendrà entre altres coses:

- Tornar a llegir tota la documentació del *framework* i veure exemples. Revisat tot el que s'ha desenvolupat per veure si es pot fer un millor ús d'alguna part, o si hi ha eines disponibles que no fem servir, etc.
- Actualització de la base de dades, s'ha de preparar canvis que hem vist que són necessaris per penjar, encara que sigui amb pocs testers coneguts i en versió tancada. S'han d'analitzar indexacions, i altres aspectes.
- Actualització de la seguretat, s'ha de implementar millor la seguretat per cada jugador concret, i fer proves.
- Revisió de la temàtica i jugabilitat, ens haurem de centrar amb analitzar les parts que s'havien pensat i afegir-hi el modificat durant el desenvolupament. Tenim acordada una sèrie de reunions posteriors a la presentació, amb un *game tester* i traductor, amb el que revisarem tota la versió anglesa del joc, i ens oferirà ajuda de dissenyador, per revisar la jugabilitat, ratis, usabilitats, i altres idees que pugui aportar. Això no s'ha fet encara, perquè de cara a la versió: Maqueta per la presentació, no era prioritària la temàtica.

- Revisió de la interfície. S'ha de revisar la part gràfica, i si és possible millorar la interfície de la web, que sembli més professional. Es farà una reunió amb una dissenyadora per assessorar-nos sobre els possibles canvis a fer. (ja se'n va fer una, però no es va poder concretar gaire perquè la maqueta estava a mig realitzar, i no es va poder fer una idea dels continguts i abast que tindríem, ara sí serà possible).
- Revisar codi. Abans de seguir avançant s'ha de fer un *benchmark* del codi possible per les implementacions que més es fan, i prendre sempre al codificar la versió que en les proves hagi resultat més ràpida i/o amb menys càrrega de memòria.
- S'ha de fer un anàlisi final per veure la modularitat de l'aplicatiu, si es pot estructurar millor les parts ja fetes, s'hauran d'aplicar els canvis pertinents.
- S'han de definir i implementar els casos d'ús que es decideixi que ha d'haver funcionals per la versió alfa tancada.
- S'ha de configurar els servidor *online*, i fer testeig previ a la versió amb l'aplicatiu funcionant en aquest sistema.
- S'ha de tornar a testejar la compatibilitat entre navegadors de tot l'aplicatiu.
- S'ha d'implementar les eines de relacions personals, i facilitar als *testers* la comunicació amb nosaltres. Segurament s'implementi un entorn especialitzat per ells. (es a dir alguns senzills casos d'ús i un entorn amb altre format, orientat al *feedback* dels errors, o aspectes a comentar).
- S'hauria de revisar i estar al dia de les versions existents i les que s'han fet servir, per mirar de fer servir la versió primant primer estabilitat, i segon velocitat.

S'ha d'analitzar encara les funcionalitats necessàries, i tot el temps que requerirà. Però estimem que la següent iteració es marcarà un temps màxim de 3 mesos aproximadament per finalitzar, això no es vol allargar gaire més perquè es prefereix testejar en tancat però *online* menys casos d'ús però fer-ho abans ja en el seu sistema final.

4.2.2- Previsió a llarg termini.

Una vegada penjada la versió alfa tancada s'haurà d'analitzar que ha d'incloure la següent iteració, però a priori ja podríem assegurar que al menys comprendrà:

- Testeig de tot l'aplicatiu *online*, de seguretat, de funcionament, de consistència, de velocitat, de càrrega de memòria de l'usuari i la del servidor.
- S'haurà d'anar corregint tots els errors que es vagin reportant, o que detectem.
- S'haurà de fer les modificacions necessàries en l'apartat gràfic i funcional.
- Si encara es requereix s'haurà de millorar la part gràfica de la web.
- Si es requereix s'haurà de millorar el sistema que s'hagi fet per el feedback amb els testers.

L'objectiu que es té a llarg termini, al menys inicialment, és desenvolupar tot el possible de l'aplicatiu sense fer *outsourcing*, però amb el temps, si el projecte té èxit, i s'aconsegueix volum d'usuaris, s'hauria de mirar que totes les parts siguin el més professional possible. És a dir, si el disseny encara no s'ha modificat gaire, s'ha de fer que tingui un nivell professional, que la gestió de la base de dades estigui afinada i sigui eficient, s'ha de testejar molts aspectes *online*.

És un condició *sine qua non* que la jugabilitat s'hagi perfeccionat per la versió oberta.

5- Conclusió.

La competitivitat del sector és molt alta, hi ha empreses consolidades i molt professionals al darrera. Però de la mateixa forma sempre que s'ha provat un dels jocs es troben molts aspectes millorables.

S'ha trobat que són pocs els jocs que aporten innovacions en la usabilitat, i aquestes acostumen a ser merament bona fluïdesa d'interacció propiciada per l'AJAX.

El projecte ha pretès fer la maqueta d'un joc que sigui suficientment addictiu i aprofitar possibles avantatges competitius. S'ha volgut prendre avantatge amb la competència propiciant que l'usuari pugui estructurar d'interfície com li sigui més fàcil interactuar-hi. Les tecnologies que s'han utilitzat han fet possible assolir aquest punt.

El desenvolupament ha pretès fer el producte adaptable a diferents llengües per tal d'augmentar el nombre d'usuaris potencials. El sistema implementat permet incorporar llenguatges de manera molt senzilla.

El projecte ha volgut tenir com a prioritat la escalabilitat i facilitat de manteniment de l'aplicació. El *framework* utilitzat i altres tecnologies han facilitat que això s'aconsegueixi.

Annex.

Annex A - Anàlisi de la competència.

A.1- Objectius.

Els objectius de l'anàlisi són avaluar què estan fent els productes del sector. Una vegada realitzat l'anàlisi es vol poder fer una aproximació del potencial del producte a un any vista.

A.2- La mostra.

Les mostres que s'han triat han estat un conjunt de jocs ja funcionals que estan en línia.

Ens hem centrat en cinc productes “de referència” en el sector: www.ogame.es , www.travian.es , www.hatrick.org , www.darkonline.net, www.seafight.es

Cadascun dels cinc objectes d'estudi té el seu funcionament:

- **Ogame:** és un joc de naus espacials amb aproximadament 8 anys al mercat, on el jugador colonitza planetes, pot crear flotes de naus amb les que pot atacar a altres jugadors, recollir residus, pot fer llunes i fer-hi construccions especials, té muntat un sistema de clans per atacar i colonitzar zones estratègicament. L'univers el tenen dividit en sistemes solars (centenars) i a cada sistema hi ha 9 posicions disponibles per colonitzar un planeta, en els planetes es pot construir i investigar.

Té una part de pagament amb la que es treu la publicitat i facilita algunes eines, com, per exemple, posar en cua construccions perquè no requereixi la intervenció del jugador, entre la finalització d'una construcció i la següent. També afegeix un resum de l'estat de tots els planetes, que resulta bastant útil donada la complexitat del joc.



Captura de pantalla del joc Ogame.

- Travian:** Travian és un videojoc d'estratègia en línia multijugador massiu (MMORTS), l'objectiu del qual és colonitzar i conquerir enfront els altres jugadors. En aquest joc gratuït poden participar milers de jugadors simultàniament en un món virtual històric ambientat en la caiguda de l'Imperi Romà. El joc està programat en PHP. Actualment ha estat traduït a més de 40 idiomes i compta amb més de 2,5 milions d'usuaris registrats en prop de 250 servidors. La seva interfície gràfica és molt pobre i poc amigable, encara que ofereix una bona jugabilitat.

Latencia 10 ms
Hora del servidor: 17:25:44

Plus

65/800 65/800 65/800 81/800 2/12

Aldea de behere

Producción:

Madera:	8 por hora
Barro:	8 por hora
Hierro:	8 por hora
Cereales:	10 por hora

Tropas:
ninguno



Captura de pantalla del joc Travian.

- **Hattrick:** És un joc creat a Suècia l'any 1997. Es tracta d'un joc en el qual el jugador administra el seu propi club de futbol. Actualment compost per 124 lligues nacionals, traduït a 48 idiomes i amb més d'un milió d'entrenadors i equips de tot el món, combina la gestió econòmica, l'estratègia esportiva i la diversió que proporciona la comunitat mundial de *mánagers*, amb les seves conferències i federacions.

Pel que fa a la interfície, és de les més simples entre els jocs de navegador i és molt directa a tots els vincles de la web. També inclou un flash per a fer alineacions i veure els comentaris dels partits a temps real.

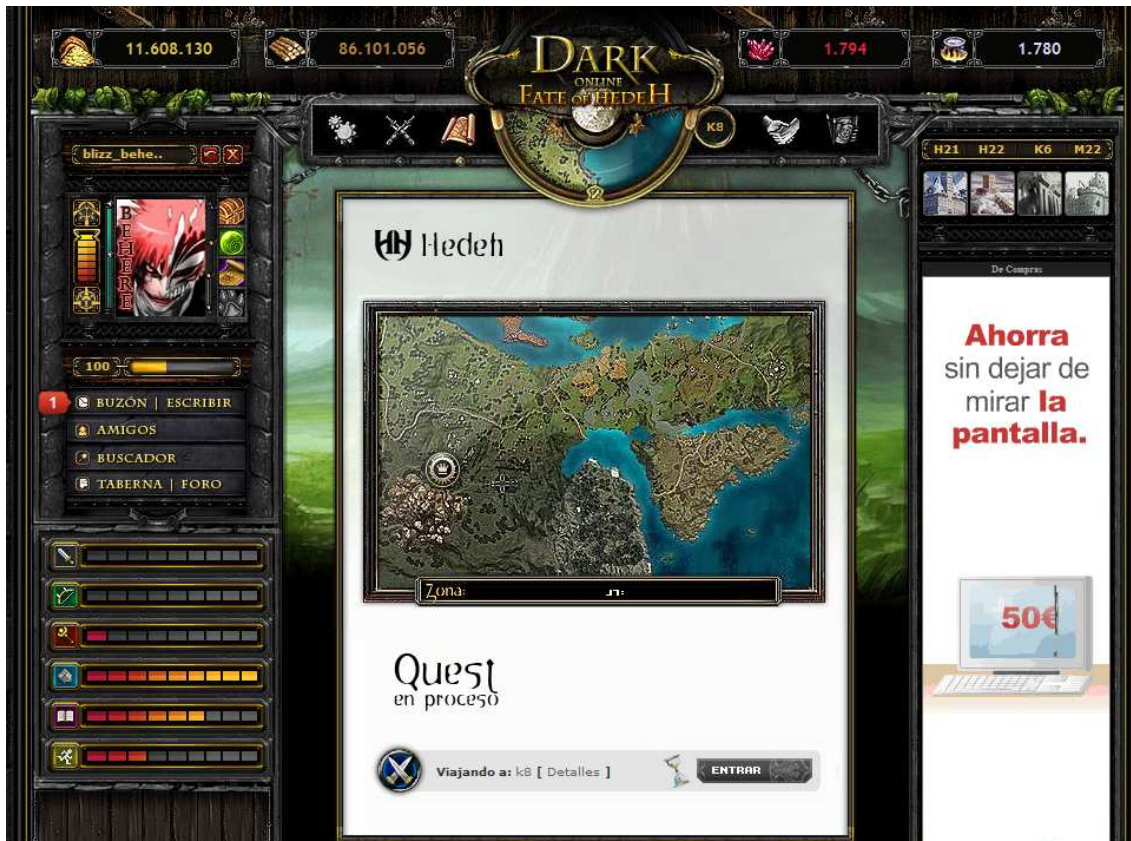
A més, abonant una quota temporal, l'usuari pot convertir-se en *supporter*, que encara que no suposa cap avantatge per al joc respecte a la resta dels jugadors, amplia les característiques de Hattrick, accedint a tot un seguit de funcionalitats accessòries com per exemple escut d'equip, notes de premsa, estadístiques, favorits, federacions i veure la cara dels jugadors, entre d'altres.



Captura de pantalla del joc Hatrck.

- **Darkonline:** és un joc que porta 3 anys en línia, on el jugador conquereix castells en diferents zones del món (n'hi ha desenes), té diferents equipaments, habilitats i elements (*items*). Té un sistema de clans que els permet liderar una zona i això dona beneficis de recursos al clan. Té molta diversitat d'enemics *NPC* (personatges del sistema) repartits per totes les zones del món, cadascun amb diferents ratis per, al vèncer, obtenir certs elements i recursos.

El jugador té un heroi i unes tropes amb les que ataca. Els atacs estan limitats amb temps, i la mobilitat per el mapa també sent proporcional a la distància. El mercat permet vendre objectes d'equipament a altres jugadors, i el comerç de recursos. Es pot fer PVP(combats jugador contra jugador). Té parts de pagament com: reinici d'atributs d'heroi, reinici d'habilitats d'heroi, canvi de facció, etc.



Captura de pantalla del joc Darkonline

- **Seafight:** és un joc de pirates amb 5 anys al mercat. On el jugador controla un vaixell en temps real en un mapa separat per quadrants. El jugador pot recórrer els mars amb els vaixells que pugui anar comprant, quant més recursos tingui, millors vaixells i armes tindrà. Els vaixells tenen diferents característiques, que els fan, uns ser més ràpids per viatjar i cercar, però dolents en combat, i altres més grans i lents, però que resulten molt poderosos en combat.

Cada vaixell pot equipar canons amb els que disparar l'enemic, hi ha de diferents potencies de danys i diferents velocitats de càrrega, i bales de canó també de diferents tipus amb les mateixes variants en velocitat i potència. Per tot això en el mapa sovint el jugador té la sensació de trobar vaixells molt diversos. El sistema situa vaixells *NPC* (personatge del sistema) per el mar, alguns agressius, i alguns que només ataquen si són provocats.

Té una part de pagament que permet comprar diners del joc directament.



Captura de pantalla del joc Seafight (mode menú).



Captura de pantalla del joc Seafight (mode jugable).

A.3- Criteris a avaluar.

Idiomes: Nombre de idiomes que suporta l'aplicació.

Llocs que el vinculen: nombre d'anuncis que tenen en altres pàgines publicitant-los.

Ingressos: Benefici directe dels *adsense* de google (diaris)

Número d'usuaris: nombre d'usuaris (actius).

Valoració preu web: Valoració aproximada en dòlars proporcionada www.cubestat.com.

Avantatges / Desavantatges: que siguin bastant significatius.

Visites diàries: mitja del nombre de visites que rep el joc al dia.

Temps de càrrega: Temps mig de càrrega del joc, calculat per l'índex Alexa.

web	Criteris				
	Idiomes	Llocs que el vinculen	Ingressos per adsense diaris	Número usuaris	Valoració preu web
www.ogame.com.es	32	616	\$579,00	30.000.000	\$423.189
www.travian.net	30	385	\$972	2.500.000	\$709.568
www.hattrick.org	48	2.956	\$4.285	899.302	\$3.128.572
www.darkonline.net	1	23	\$21	15.000	\$15.665
www.seafight.com	28	177	\$125	25.500.924	\$91.793

web	Criteris		
	Avantatges / desavantatges competitiu	Visites diàries	Temps de càrrega
www.ogame.com.es	La jugabilitat estratègica està molt ben elaborada.	193.200	Mig (2.266 Segons), 54% dels llocs són més lents.
www.travian.net	És de les webs més ràpides de carga del món.	324.045	Molt ràpid (0.45 Segons), 96% dels llocs són més lents.
www.hattrick.org	El joc requereix massa dedicació en comparació amb la competència.	1.428.572	Mig (2.507 Segons), 52% dels llocs són més ràpids.
www.darkonline.net	Només té un idioma. La càrrega és molt lenta.	7.153	Molt lent (7.286 Segons), 91% dels llocs són més ràpids.
www.seafight.com	Primer dels patrocinats de Google.	41.915	Mig (2.031 Segons), 58% dels llocs són més lents.

Index Info For: ogame.com.es

Alexa Rank: 5,175

Quantcast Rank: 695,261

Competee Rank: 1,738,153

PageRank: 4

Backlinks: 0

Google Indexed Pages: 17,600

Google Indexed Images: 5,980

Yahoo Indexed Pages: 0

Live Indexed Pages: 4,230

Dmoz Listing: No

Host IP: 212.227.100.164

Site Hosted in: Germany

Domain Suffix: Spain

Archive: [ogame.com.es in the past](#)

Compete Rank

Index Info For: travian.net

Alexa Rank: 3,086

Quantcast Rank: 260,421

Competee Rank: 651,053

PageRank: 5

Backlinks: 0

Google Indexed Pages: 0

Google Indexed Images: 464

Yahoo Indexed Pages: 0

Live Indexed Pages: 21,700

Dmoz Listing: No

Host IP: 92.51.158.111

Site Hosted in: Germany

Owner: Travian Games GmbH

ICANN Registrar: PSI-USA, INC.

Site Age: 7 years

Created: 2003-05-28

Expires: 2010-05-28

Updated: 2009-05-29

Domain Suffix: Network

Archive: [travian.net in the past](#)

Compete Rank

Index Info For: hattrick.org

Alexa Rank: 700

Quantcast Rank: 132,154

Competee Rank: 116,812

PageRank: 6

Backlinks: 0

Google Indexed Pages: 0

Google Indexed Images: 796

Yahoo Indexed Pages: 0

Live Indexed Pages: 142,000

Dmoz Listing: Yes

Dmoz Title: Hattrick

Dmoz Description: - Træne dit hold og handle spillere i samtidig konkurrence med hundredetusinder af spillere fra hele verden.

Dmoz Category: World: Dansk:

Spil: Internet: Hattrick

Host IP: 193.34.189.76

Site Hosted in: Switzerland

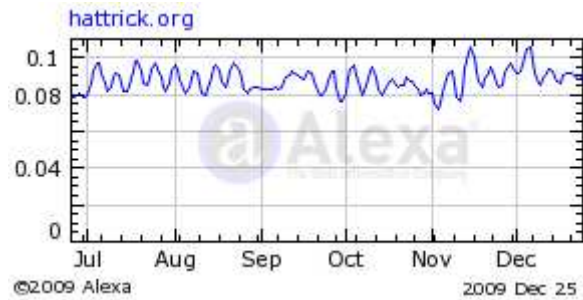
Owner: Hattrick Ltd

ICANN Registrar: Public Interest Registry

Site Age: 10 years

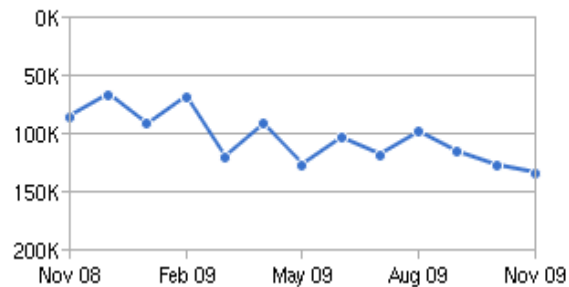
Created: 2000-04-13

Domain Suffix: Organization



Compete Rank

hattrick.org



Index Info For: darkonline.net

Alexa Rank: 139,821

Quantcast Rank: 0

Competee Rank: 0

PageRank: 0

Backlinks: 0

Google Indexed Pages: 0

Google Indexed Images: 193

Yahoo Indexed Pages: 0

Live Indexed Pages: 435

Dmoz Listing: No

Host IP: 87.98.227.38

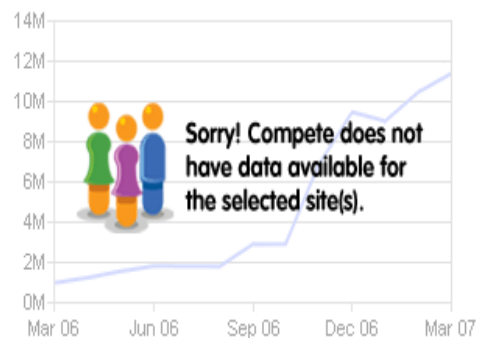
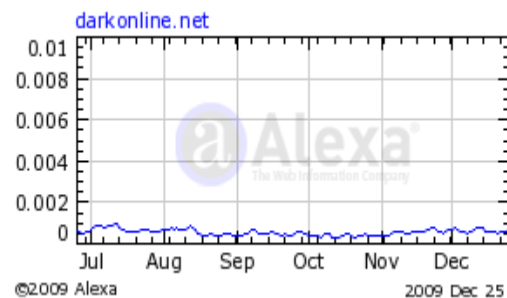
Site Hosted in: Spain

ICANN Registrar: ONLINENIC, INC.

Site Age: 3 years

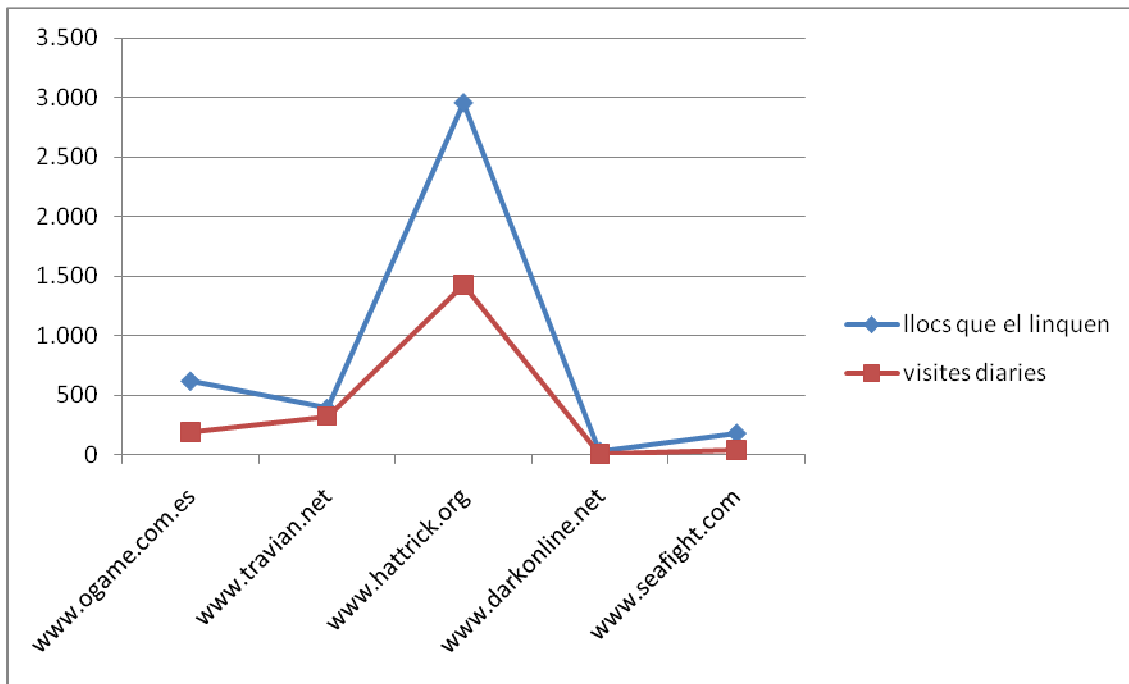
Created: 2007-04-12

Domain Suffix: Network



A.4- Interpretació de les dades.

Observem una relació directament proporcional de llocs que vinculen al lloc web i el nombre de visites diàries que rep. Podem extrapolar el mateix resultat per iteracions



posteriors del projecte.

Relació entre llocs que enllacen al joc i nombre de visites diàries que obté.

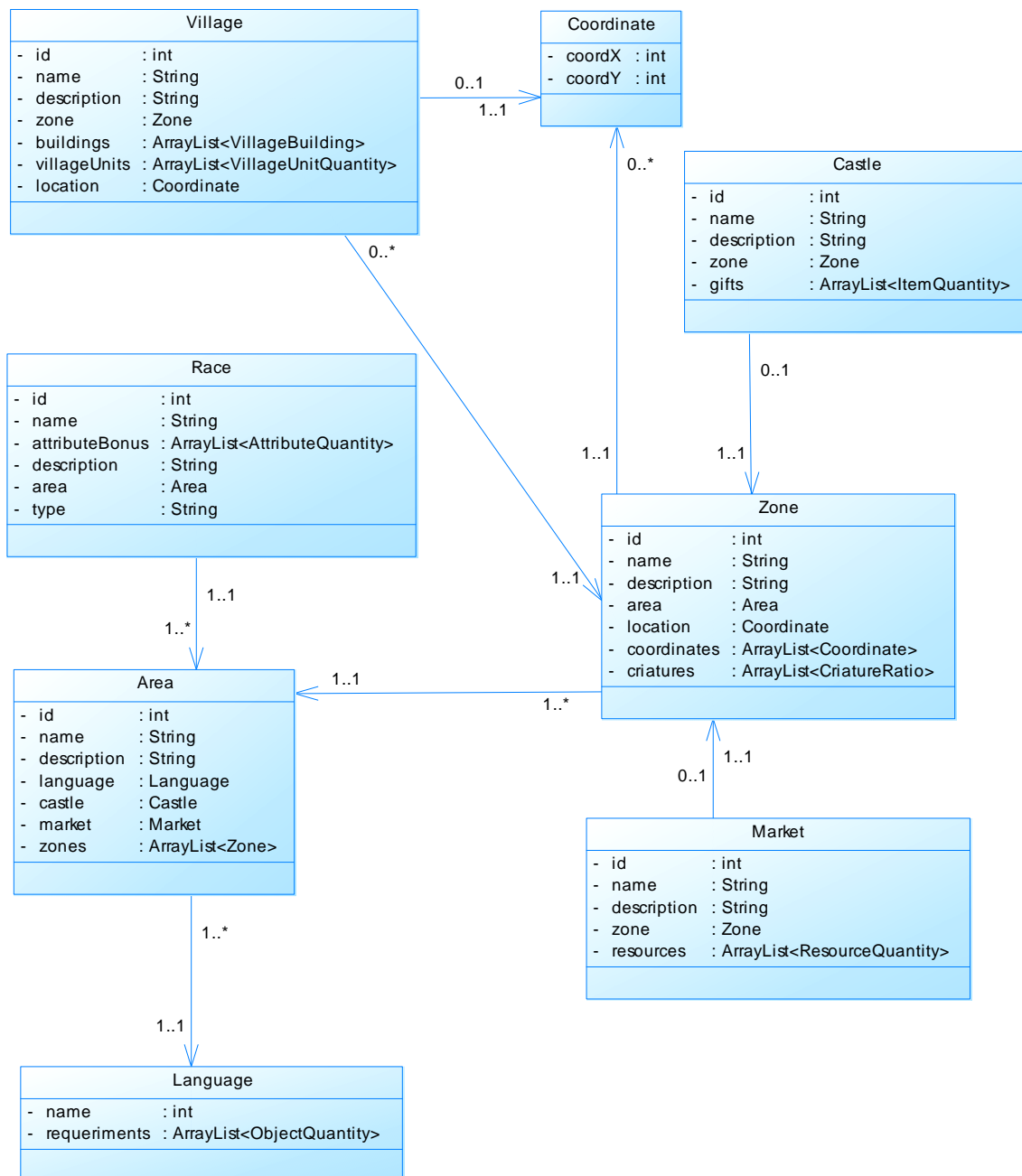
Tots els llocs web estudiats ofereixen un lloc bastant usable, alguns tenen parts implementades amb AJAX per millorar la interactivitat, tots requereixen una mica d'hores de joc per la fluïdesa, encara que ja és el normal en qualsevol *RPG* de navegador.

Quasi la totalitat dels productes disposen de més d'una vintena d'idiomes, això els proporciona molts més usuaris potencials. Tots tenen publicitat i sistemes de pagament per algunes parts. S'ha comprovat que els més vistosos gràficament parlant, també són els més lents de carregar. No s'ha observat cap relació directa entre l'aspecte visual del lloc web i el nombre d'usuaris que té.

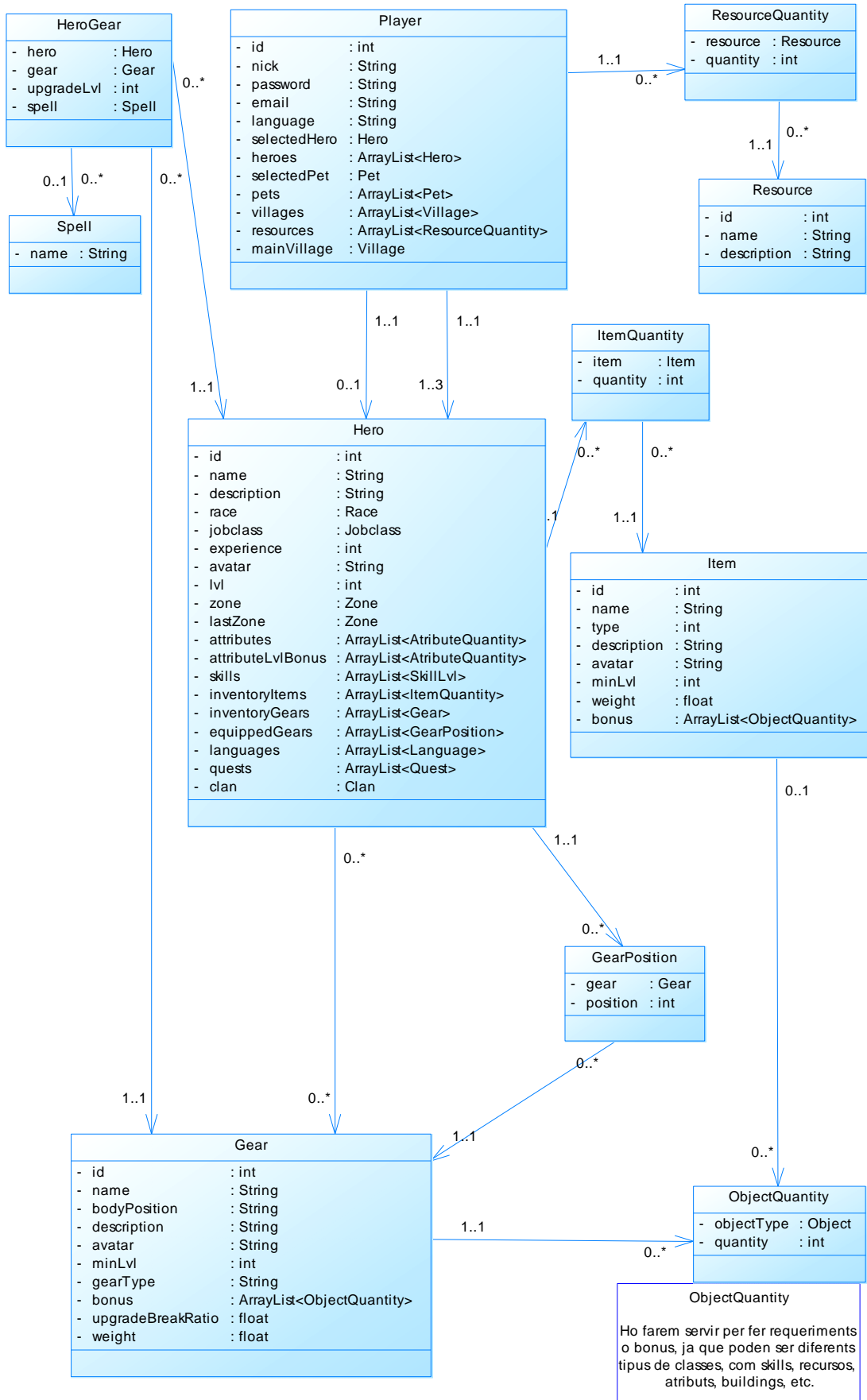
Annex B - Model de Domini.

El diagrama de classes del model de domini s'ha dividit en diversos submodels degut a que resulta complex de visualitzar-ho complert. Els submodels es detallen a continuació.

B.1- Submodel de Localització (*Location*).



B.3- Submodel de Possessions materials.



Annex C - Casos d'ús.

C.1- Gestió Heroi/Mascota.

C.1.1- Crear Heroi.

Nom	Crear Heroi.
Descripció	El Jugador crea un nou heroi.
Actors	Jugador.
Requisits	El Jugador ha de seleccionar entre les opcions que li permet l'aplicació tal com vulgui configurar el seu heroi.
Restriccions	El Jugador no pot sobrepassar el límit de número de herois per Jugador, fent la creació. El Jugador ha d'introduir les dades que siguin requerides per fer la creació.
Resultat esperat	El nou heroi és enregistrat al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la creació d'un nou Heroi. 2. El Sistema demana les dades al Jugador. 3. El Jugador introdueix les dades configurant l'Heroi i confirma la creació. 4. El Sistema valida les dades. 5. El Sistema confirma la creació al Jugador.
Flux alternatiu	<ol style="list-style-type: none"> 1.1 El Jugador ja té el numero límit de herois, i el sistema no li permet crear de nous. 4.1 Les dades introduïdes per a l'Heroi són incorrectes i el Sistema retorna error i torna al punt 2. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.1.2- Modificar Heroi.

Nom	Modificar Heroi.
Descripció	El Jugador vol modificar un heroi ja creat (nom, avatar, etc.).
Actors	Jugador.
Requisits	El Jugador ha de seleccionar entre les opcions que li permet modificar l'aplicatiu tal com vulgui configurar el seu heroi.
Restriccions	Només es pot modificar un heroi alhora, no es pot canviar la raça ni els estats.
Resultat esperat	Els canvis del heroi són enregistrats al Sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la modificació d'un Heroi ja creat per ell. 2. El Sistema li mostra les dades actuals i li permet canviar les que es poden. 3. El Jugador modifica les dades configurant l'Heroi i clica finalitzar la modificació. 4. El Sistema valida les dades. 5. El Sistema confirma la modificació al Jugador.
Flux alternatiu	<ol style="list-style-type: none"> 3.1 El Jugador no confirma la modificació. 4.1 Les dades introduïdes per al Heroi són incorrectes i el Sistema retorna error i torna al punt 2. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.1.3- Esborrar Heroi:

Nom	Esborrar Heroi.
Descripció	El Jugador vol esborrar un heroi ja creat.
Actors	Jugador.
Requisits	El Jugador ha de tenir com a mínim un heroi.
Restriccions	Només es pot eliminar un heroi alhora. No pot estar realitzant/rebent cap atac ni interacció amb altres jugadors.
Resultat esperat	El heroi del jugador es eliminat i passa a tenir un <i>slot</i> lliure.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a la eliminació d'un Heroi ja creat per ell. 2. El Sistema li mostra les dades actuals del heroi i un botó per eliminar-lo. 3. El Jugador clica que vol eliminar-lo. 4. El Sistema demana confirmació introduint el correu electrònic i la contrasenya. 5. El Jugador introdueix el correu electrònic i la contrasenya i confirma la eliminació. 6. El sistema elimina el heroi del jugador i li indica la correctesa de l'eliminació.
Flux alternatiu	<ol style="list-style-type: none"> 3.1 El Jugador no confirma l'eliminació del heroi. 5.1 Les dades introduïdes per el jugador són incorrectes, a la tercera vegada que es compleixi aquest punt, la conta passarà a tenir un ban de 1hora i s'enviarà un correu electrònic a la conta del jugador informant de l'intent d'eliminar un heroi.

C.1.4- Tria de personatges.

Nom	Tria de personatges.
Descripció	Quan un Jugador es <i>logueja</i> en l'aplicatiu o selecciona canviar personatges s'executa aquest cas d'ús, podrà seleccionar quin heroi i quina mascota vol fer servir.
Actors	Jugador.
Requisits	El jugador ha de tenir com a mínim dos herois o un heroi i dos mascotes.
Restriccions	El Jugador no tindrà cap restricció alhora de anar canviant entre els seus personatges (herois i mascotes), els podrà combinar com vulgui i tantes vegades com vulgui.
Resultat esperat	El sistema seleccionarà l'equip (heroi/mascota) que ha triat el Jugador com el seu equip de batalla.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador indica que vol seleccionar personatges. 2. El Sistema li mostra el conjunt de herois i mascotes dels que disposa, marcant-li els que té seleccionats actualment, i permetent-li canviar entre d'altres que tingui el Jugador. 3. El Jugador fa la selecció corresponent i indica al sistema que ho confirma. 4. El sistema assigna l'equip de batalla que ha triat el Jugador com el seu actual.
Flux alternatiu	<ol style="list-style-type: none"> 2. El Jugador no disposa de herois i/o mascotes, el sistema li indicarà al Jugador que ha de crear com a mínim un heroi. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.1.5- Assignació d'estats.

Nom	Assignació d'estats.
Descripció	Quan un heroi o mascota és creat, o puja de nivell, se li concedeixen uns punts a repartir entre els possibles atributs que el configuren per la batalla i altres aspectes. El Jugador pot decidir què pujar o deixar-ho per un altre moment, ja que els punts només se li reduiran al repartir-los, no s'esborren o es perden al sortir ni de cap altre manera que no sigui utilitzar-los per millorar els atributs del heroi o mascota en qüestió.
Actors	Jugador.
Requisits	El Jugador ha de tenir punts disponibles a repartir als atributs.
Restriccions	Aquest cas d'ús només serà cridat si el Jugador té suficients punts a repartir segons l'habilitat que vulgui pujar.
Resultat esperat	Els nous valors dels atributs del heroi o mascota, són actualitzats al sistema.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador indica que vol editar els atributs del seu heroi o mascota. 2. El Sistema li mostra en detall els atributs que formen el seu personatge i li permet utilitzar els punts aconseguits per repartir-los en la millora. 3. El Jugador incrementa els atributs que vol i clica confirmar l'acció. 4. El sistema li confirma la modificació al Jugador.
Flux alternatiu	<p>2.1. El Sistema li mostra en detall els atributs que formen el seu personatge, però no li permet millorar cap atribut perquè no té punts suficients a repartir.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no ho emmagatzema.</p>

C.1.6- Tria de professió (jobclass).

Nom	Tria de professió
Descripció	El Jugador tria una professió per el seu heroi.
Actors	Jugador.
Requisits	L'heroi ha de complir els requisits necessaris per aconseguir ser d'una professió (quests realitzades, nivell de l'heroi, raça de l'heroi, una certa professió actual, etc.).
Restriccions	Un heroi no pot ser d'una professió si no compleix els requisits necessaris d'aquesta. Un heroi no pot ser de dues professions alhora.
Resultat esperat	L'heroi es fa d'una certa professió.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a aprendre una professió. 2. El Sistema mostra les professions disponibles per a l'heroi. 3. L'heroi selecciona aprendre una de les professions disponibles. 4. El Sistema actualitza les dades de l'aplicació i confirma la correctesa al Jugador.
Flux alternatiu	<p>2.1. L'heroi no té professions disponibles i el Sistema ho reporta al Jugador.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.1.7- Aprendre habilitat (skill).

Nom	Aprendre habilitat
Descripció	El Jugador tria aprendre una habilitat.
Actors	Jugador.
Requisits	L'heroi ha de complir els requisits necessaris per aconseguir una habilitat (quests realitzades, nivell de l'heroi, raça de l'heroi, una certa professió actual, habilitats ja aconseguides, etc.).
Restriccions	Un heroi no pot aconseguir una habilitat si no compleix els requisits necessaris d'aquesta.
Resultat esperat	L'heroi aprèn una habilitat nova.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a aprendre una habilitat. 2. El Sistema mostra les habilitats disponibles per a l'heroi. 3. L'heroi selecciona aprendre una de les habilitats disponibles. 4. El Sistema actualitza les dades de l'aplicació corresponents i confirma la correctesa al Jugador.
Flux alternatiu	<p>2.1. L'heroi no té habilitats disponibles i el Sistema ho reporta al Jugador.</p> <p>1.1. L'heroi no compleix els requisits per a aprendre la habilitat i el Sistema retorna error. (això només en el cas de que la habilitat es pugui seleccionar tot i no complir els requisits, que en tot cas es veurà desenfocada).</p> <p>(* El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.1.8- Visualitzar l'arbre d'habilitats.

Nom	Visualitzar l'arbre d'habilitats.
Descripció	El Jugador visualitza l'arbre d'habilitats i camins disponibles (apreses i per aprendre) de la professió del seu heroi.
Actors	Jugador.
Requisits	L'heroi ha de pertànyer a la professió de l'arbre d'habilitats.
Restriccions	Un jugador no pot accedir a un arbre d'habilitats que no pertany a la professió del seu heroi.
Resultat esperat	El Jugador visualitza l'arbre d'habilitats.
Flux normal	1. El Jugador accedeix a visualitzar l'arbre d'habilitats. 2. El Sistema mostra l'arbre d'habilitats.
Flux alternatiu	2.1. El jugador no té permisos per veure l'arbre d'habilitats i el Sistema li reporta l'error. (*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

C.2- Gestió Inventari/Equipament.

C.2.1- Afegir element a l'inventari.

Nom	Afegir element a l'inventari.
Descripció	El Jugador vol afegir un element a l'inventari del heroi que te equipat.
Actors	Jugador.
Requisits	L'objecte ha de cabre a l'inventari, l'Heroi ha de poder suportar el pes de l'inventari.
Restriccions	Aquest cas d'ús esta restringit a elements que rebí l'heroi mitjançant un <i>dropeig</i> d'algun monstre, d'algun comerç, de quest, o de agafar-lo del inventari.
Resultat esperat	1. El Sistema afegeix un objecte al inventari de l'heroi.
Flux normal	1. L'objecte es suma al inventari de l'heroi.
Flux alternatiu	1.1 El inventari de l'heroi és ple i el Sistema retorna error. 1.2 El pes de l'objecte sumat al pes de la resta del inventari és superior el pes que l'heroi pot suportar i el Sistema dona error.

C.2.2- Eliminar element de l'inventari.

Nom	Eliminar element de l inventari.
Descripció	El Jugador vol eliminar 1 element del inventari de l'heroi actual.
Actors	Jugador.
Requisits	Es requereix que el jugador confirmi la eliminació.
Restriccions	No es pot eliminar un objecte del inventari de l'heroi si aquest està sent utilitzat.
Resultat esperat	L'objecte es resta del inventari de l'heroi.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica eliminar un objecte del inventari del seu heroi. 2. El sistema demana confirmació per eliminar l'objecte. 3. El jugador confirma l'eliminació de l'objecte. 4. El Sistema esborra l'objecte de l'heroi. 5. El Sistema executa el cas d'ús "Consultar Inventari".
Flux alternatiu	<p>1.1 L'objecte seleccionat a esborrar està sent utilitzat per l'heroi i el Sistema retorna error.</p> <p>3.1. El jugador cancel·la l'eliminació de l'objecte, el sistema va al pas 5.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.2.3- Equipar/Usar element de l'inventari.

Nom	Equipar/Usar element del inventari.
Descripció	El Jugador equipa/usa l'element del inventari.
Actors	Jugador.
Requisits	No es pot equipar/usar un objecte del inventari de l'heroi si aquest ja està sent utilitzat.
Restriccions	L'objecte ha de poder ser usat per l'heroi (restriccions de classe, raça, nivell, etc.).
Resultat esperat	L'heroi usa/s'equipa l'objecte.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a usar/equipar un objecte del inventari del seu heroi. 2. El Sistema actualitza les dades corresponents i el inventari de l'heroi. 3. El Sistema executa el cas d'ús "Consultar Inventari".
Flux alternatiu	<ol style="list-style-type: none"> 2.1. L'objecte seleccionat a usar/equipar està sent utilitzat per l'heroi i el Sistema retorna error. 2.2. L'objecte no pot ser usat per l'heroi i el Sistema retorna error. <p>(* El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.2.4- Desequipar element.

Nom	Desequipar element.
Descripció	El Jugador vol desequipar un objecte del heroi que té en ús.
Actors	Jugador.
Requisits	L'objecte ha d'estar equipat. L'objecte ha de cabre al inventari.
Restriccions	No es pot desequipar un objecte del inventari de l'heroi si aquest no està sent utilitzat.
Resultat esperat	L'heroi es desequipa l'objecte.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador accedeix a desequipar-se un objecte del seu heroi. 2. El Sistema actualitza les dades corresponents i el inventari de l'heroi. 3. El Sistema executa el cas d'ús "Consultar Inventari".
Flux alternatiu	<ol style="list-style-type: none"> 1.1 El inventari de l'heroi és ple i el Sistema retorna error. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.3- Comerç.

C.3.1- Comprar element a comerciant (wcc).

Nom	Comprar element a comerciant (WCC)
Descripció	El mon de WarComrades disposarà de WCC, WarComradeCharacters. Seran personatges amb els que els jugadors podran interactuar. En aquest cas d'ús concret, per fer la compra d'elements del joc.
Actors	Jugador.
Requisits	El Jugador ha de disposar dels diners suficients per fer la compra que realitzi.
Restriccions	La compra no podrà en cap cas fer sobrepassar la capacitat de carrega del equip de combat (motxilla) de combat que tingui seleccionat el jugador.
Resultat esperat	El sistema fa la transacció, descomptant l'import al jugador, i posant-li els elements en el inventari
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica en el WCC amb el que vol fer comerç 2. El WCC dona la benvinguda al jugador i li ofereix els tipus de productes dels que disposa, li dona la opció de vendre-li, i també la opció de fer intercanvi. 3. El jugador selecciona un element per comprar i selecciona la quantitat d'ells que vol comprar, el sistema li indica el preu que li costa alhora, etc. i el Jugador clica comprar. 4. El sistema li descompta l'import de la compra al jugador, i crida el cas d'ús "Afegir element al inventari" amb l'objecte comprat.
Flux alternatiu	<ol style="list-style-type: none"> 2.1 El WCC seleccionat no es de comerç. 2.2 El Jugador clica vendre-li elements al WCC, llavors es crida el Cas d'ús: Vendre element a comerciant. 3.1 L'element que vol comprar el jugador no permet fer compra indicant quantitat. <p>Equipacions o altres elements que es considerin "rars" si el Jugador vol comprar varis haurà de anar fent el pas 3... tantes vegades com unitats vulgui comprar.</p> <ol style="list-style-type: none"> 3.2 El cas d'ús "Afegir element al inventari" retorna error per sobrepassar el pes del inventari. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el</p>

	<p>Sistema no emmagatzema res.</p>
--	------------------------------------

C.3.2- Vendre element a comerciant (WCC).

Nom	Vendre element a comerciant (WCC).
Descripció	Un jugador vol comerciar amb un WCC per vendre elements que té.
Actors	Jugador, WCC.
Requisits	El Jugador ha de disposar d'elements, el WCC ha de permetre vendre-li (ser comerciant).
Restriccions	La venda d'elements sempre s'ha de confirmar.
Resultat esperat	El sistema fa la transacció, augmenta l'import de la venda als diners del jugador, i eliminant els elements del inventari.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica en el WCC amb el que vol fer comerç 2. El WCC dona la benvinguda al jugador i li ofereix els tipus de productes dels que disposa, també li dona la opció de vendre-li, i també la opció de fer intercanvi. 3. El jugador selecciona un element per vendre i selecciona la quantitat d'ells que vol vendre, el sistema li indica el preu que li paga alhora, i el Jugador clica vendre. 4. El sistema li demana confirmació indicant-li el preu. 5. El Jugador accepta la venda per l'import indicat. 6. El sistema li augmenta els diners de la venda al jugador, i li treu els elements de la seva "motxilla".
Flux alternatiu	<ol style="list-style-type: none"> 2.1 El WCC seleccionat no es de comerç. 2.2 El WCC no permet vendre-li elements. 3.1 L'element que vol comprar el jugador no permet fer venda indicant quantitat. Equipacions o altres elements que es considerin "rars" si el Jugador vol vendre varis Haurà de anar fent el pas 3 tantes vegades com unitats vulgui vendre. 3.2 L'element que seleccioni el Jugador no es pot vendre a WCC <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.3.3- Intercanvi de recursos a comerciant (WCC).

Nom	Intercanvi de recursos a comerciant (WCC).
Descripció	Un jugador vol comerciar amb un WCC per fer intercanvi de recursos. El WCC indicarà tots els recursos amb els que comercia. I a cada recurs hi posarà el preu base amb el que el ven, i amb el que compra també. Quanta mes quantitat es vulgui comprar, mes barat serà obtenir el recurs, i quant menys li quedi al WCC mes car ho vendrà després, etc.
Actors	Jugador.
Requisits	El Jugador ha de disposar de recursos, el WCC ha de permetre intercanvi (ser comerciant, i disposar de recursos).
Restriccions	El intercanvi d'elements sempre s'ha de confirmar.
Resultat esperat	El sistema fa la transacció, disminueix el recurs ofert per el jugador al jugador, i li augmenta el que ha demanat. En la quantitat especificada en ambdós casos.
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica en el WCC amb el que vol fer comerç 2. El WCC dona la benvinguda al jugador i li ofereix els tipus de productes dels que disposa, li dona la opció de vendre-li, i també la opció de fer intercanvi. 3. El jugador selecciona fer intercanvi. 4. El WCC li mostra per cada recurs, la quantitat de que disposa, i el preu al que el compra i el preu al que el ven. 5. El jugador selecciona un recurs per obtenir, indicant la quantitat, selecciona l'element amb el que el vol intercanviar. 6. El Jugador accepta la transacció per la quantitat que li demana el WCC. 7. El sistema fa la reducció i augment de recursos en el jugador i el WCC respectivament ,i actualitza el preu al WCC, si li hem reduït l'estoc el comerciant pujarà el preu, serà més car. En canvi si li venem no alterarem el preu del producte que ell ven.
Flux alternatiu	<ol style="list-style-type: none"> 2.1 El WCC seleccionat no es de comerç. 2.2 El WCC no permet intercanvi d'elements. 3.1 El Jugador selecciona compra o venda. S'executarà el cas d'ús respectiu en cada cas. 6.1 El Jugador no accepta la transacció. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.3.4- Comerciar entre Jugadors.

Nom	Comerciar entre jugadors
Descripció	Un jugador vol comerciar amb un altre jugador per fer intercanvi de recursos o elements.
Actors	Jugador.
Requisits	Els jugadors han de tenir elements o recursos per comerciar.
Restriccions	El intercanvi d'elements/recursos sempre s'ha de confirmar.
Resultat esperat	El sistema fa la transacció, disminueix els recursos/elements que ha ofert cada jugador i li augmenta a l'altre. En la quantitat especificada en ambdós casos.
Flux normal	<ol style="list-style-type: none"> 1. Jugador1 clica en comerciar amb un altre jugador. 2. Jugador1 indica el <i>nick</i> del jugador2 amb el que vol fer comerç i clica començar el comerç 3. El Jugador1 selecciona el que vol intercanviar, ja siguin recursos,elements,etc.. 4. El Jugador1 clica enviar transacció. 5. El Jugador2 rep la transacció a falta de que introdueixi el que vol comerciar 6. El Jugador2 introdueix els elements que vol comerciar, i clica enviar transacció 7. El Jugador1 rep una situació de la transacció final, i el sistema li demana confirmació. 8. El jugador2 rep una situació de la transacció final, i el sistema li demana confirmació. 9. El Jugador1 i el Jugador2 accepten la transacció, i el sistema reparteix els elements tan com ells s'han volgut intercanviar
Flux alternatiu	<ol style="list-style-type: none"> 2.1 El jugador2 (<i>nick</i> que ha introduït el jugador1) te deshabilitat el comerç entre jugadors. 2.2 El jugador2 no existeix. 4.1 El jugador1 cancel·la la transacció. 6.1 El jugador2 cancel·la la transacció. 7.1 El jugador1 no confirma la transacció. 7.2 El jugador2 no confirma la transacció. 9.1 El Jugador1 o/i Jugador2 no accepta la transacció <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.4- Component Social.**C.4.1- Enviar Missatge.**

Nom	Enviar Missatge.
Descripció	Un jugador vol enviar un missatge a la bústia d'un altre jugador.
Actors	Jugador1, Jugador2.
Requisits	El jugador ha de conèixer l'alias del jugador2.
Restriccions	El missatge haurà de contenir només text. Es limitarà l'ús de paraules malsonants (tacos) censurant la majoria amb * El missatge haurà d'anar destinat a un únic destinatari. Només es podrà enviar un missatge cada 30seg.
Resultat esperat	El Jugador2 rep el missatge del jugador1
Flux normal	<ol style="list-style-type: none"> 1. El Jugador1 està a la bústia i clica el botó d'enviar missatge. 2. El sistema li carrega la pantalla de enviament de missatge. 3. El Jugador1 indica l'alias del jugador2 al que vol adreçar-se, escriu el missatge i clica enviar. 4. El sistema li indica que el missatge ha estat enviat.
Flux alternatiu	<ol style="list-style-type: none"> 3.1 El jugador2 no existeix. El sistema li indica sense esborrar el missatge. 4.1 El format del missatge no es correcte. El sistema li indica sense esborrar el missatge. <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

C.4.2- Llegir Missatge.

Nom	Un jugador vol llegir un missatge de la seva bústia.
Descripció	Jugador.
Actors	El jugador ha d'estar a la bústia per poder executar aquest cas d'ús.
Requisits	No s'observen restriccions.
Restriccions	El Jugador llegeix el missatge.
Resultat esperat	El Jugador llegeix el missatge.
Flux normal	1. El Jugador1 està a la bústia d'entrada i clica sobre el missatge que vol llegir. 2. El sistema li carrega el missatge, i l'enregistra com a llegit.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

C.4.3- Eliminar missatge.

Nom	Eliminar missatge.
Descripció	Un jugador vol eliminar un missatge de la bústia.
Actors	Jugador.
Requisits	El jugador ha de tenir missatges.
Restriccions	El jugador només podrà executar aquest cas d'ús des de la bústia d'entrada.
Resultat esperat	El missatge es eliminat de la bústia del jugador.
Flux normal	1. El Jugador està a la bústia d'entrada i clica sobre eliminar un o varis missatge/s. 2. El Sistema elimina el/s missatges i li confirma 3. El sistema executa el cas d'ús: consultar bústia d'entrada.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

C.4.4- Consultar bústia d'entrada.

Nom	Consultar bústia d'entrada.
Descripció	Un jugador vol consultar la bústia seva bústia d'entrada de missatges.
Actors	Jugador.
Requisits	No se'n s'observen.
Restriccions	El jugador només tindrà una bústia d'entrada, la bústia tindrà una limitació del nombre de missatges que pot tenir emmagatzemats.
Resultat esperat	El sistema carrega la bústia de missatges del jugador.
Flux normal	1. El Jugador clica sobre la bústia de missatges. 2. El Sistema carrega la bústia de missatges.
Flux alternatiu	(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.

C.5- Interfície i Organització.

C.5.1- Càlcul de Recursos.

Nom	Càlcul de Recursos.
Descripció	El sistema amb AJAX va actualitzant automàticament el nombre de recursos de que disposa el jugador, ja que són sempre visibles des de qualsevol part de l'aplicatiu. Encara s'ha de precisar el interval de actualització.
Actors	Sistema, Jugador.
Requisits	El jugador ha d'estar <i>loguejat</i> .
Restriccions	No s'observen.
Resultat esperat	Els recursos són actualitzats per el sistema.
Flux normal	<ol style="list-style-type: none"> 1. El sistema calcula per el interval de refresc que te configurat quin increment de recursos que li ha d'aplicar al jugador. 2. El sistema li actualitza en pantalla els recursos al jugador 3. Mentre el jugador no executi el cas d'ús: <i>Logout</i> el sistema va repetint el pas1 i 2.
Flux alternatiu	1.1 El jugador executa el cas d'ús Log Out i surt de l'aplicatiu.

C.5.2- Modificar la visualització de les utilitats i de les dreceres.

Nom	Modificar la visualització de les utilitats i de les dreceres.
Descripció	Un jugador vol modificar la visualització de les utilitats, se li permet canviar les utilitats d'ubicació de la web de manera interactiva arrossegant, i minimitzar-les o maximitzar-les. També se li permet ordenar la barra de dreceres.
Actors	Jugador.
Requisits/Restriccions	Els jugadors <i>premium</i> tindran la configuració que triïn amb persistència, sinó es <i>premium</i> només es desarà fins que faci <i>logout</i> .
Resultat esperat	El Jugador configura la interfície gràfica de l'aplicació gestionant la disponibilitat i posició de les utilitats, i ordenant al seu gust la barra de " <i>shortkeys</i> ".
Flux normal	<ol style="list-style-type: none"> 1. El Jugador clica per moure o maximitzar o minimitzar alguna utilitat o "<i>shortkey</i>". 2. El Sistema mou l'element fins que el jugador el deixa en una zona vàlida. 3. El Sistema emmagatzema les dades.
Flux alternatiu	<p>2.1 El Jugador deixa l'element en una zona no vàlida, l'element torna a la seva posició d'origen.</p> <p>(*) El Jugador pot sortir del joc en qualsevol moment i el Sistema no emmagatzema res.</p>

Annex D - Implementació realitzada.

En aquests moments WarComrades compta amb:

- La implementació de la base de dades amb PostgreSQL.
- Implementació del model de domini, juntament amb l'estructura que defineix el *framework* Symfony i els patrons mencionats anteriorment.
- La relació entre el model de domini i la persistència mitjançant l'*ORM* Propel (estructures definides amb *YAML*).
- Poblament inicial de la base de dades i joc de dades de proves.
- *Backend* de l'aplicació juntament amb les funcions d'agregar, modificar i editar d'administració de les dades del joc.
- Els formularis d'entrada de dades.
- La gestió de les sessions i els seus atributs.
- La seguretat de l'aplicació; concretament l'autenticació de l'usuari i l'autorització que rep en cada apartat de l'aplicació.
- S'ha definit la cultura de l'usuari; la internacionalització i la localització. De moment hi ha definits dos idiomes; castellà i anglès.
- Casos d'ús implementats:
 - Alta, Modificació, Baixa i Consulta de jugador.
 - *Login* i *Logout* de jugador.
 - Crear, Modificar, Esborrar i Consultar Heroi.
 - Crear, Modificar, Eliminar i Consultar Vila.
 - Consultar Mascota.
 - Creació, Millora i Consulta de Construccions.
 - Consultar Investigacions.

- Visualitzar Mapa.
 - Intercanvi de Recursos al Comerciant (*WCC*).
 - Consultar bústia d'entrada.
 - Enviar i Llegir Missatges.
 - Consultar *Ranking*.
 - Consultar el Mercat.
 - Modificar la visualització de les utilitats i les dreces.
 - Tria de professió.
 - Mode Expert (visualització reduïda de la interfície).
- Primera versió de la interfície gràfica.

Annex E- Glossari.

E.1- Glossari de termes tècnics.

ADSENSE: Sistema de publicitat ideat per Google en el que els desenvolupadors web poden unir-se a aquest sistema per activar textos i imatges publicitàries en les seves pàgines web.

AJAX: *Asynchronous JavaScript And XML*. És un conjunt interrelacionat de tècniques de desenvolupament web que s'executen a l'ordinador client per a crear aplicacions web interactives.

Apache: Servidor web que permet penjar una aplicació web a Internet.

API: *Application Programming Interface*. Interfície implementada per un programa software que habilita la interacció amb els demés programes. Són implementades per aplicacions, llibreries i sistemes operatius.

ASCII: *American Standard Code for Information Interchange*. Esquema de codificació de caràcters originalment basat en l'alfabet anglès. Representa el text en els ordinadors, equips de comunicació i altres mecanismes que usen text.

Backup: Procés de creació d'una còpia de seguretat de la base de dades.

Banner: Finestra inclosa en la interfície del joc que s'usa per a publicitat o per a ressaltar alguna cosa.

Batch: Quan es parla d'eliminar quelcom en *bach* significa fer-ho de forma automatitzada.

Benchmarking: Procés de comparar els processos de negoci incloent costos, temps, productivitat o qualitat de quelcom enfront la seva competència.

Browser: Navegador.

Caché: Col·lecció de dades que es dupliquen i s'emmagatzemen en algun lloc de l'ordinador. Es pot interpretar com un magatzem de dades temporal que serveix perquè l'ordinador pugui accedir a aquestes dades ràpidament i sense haver-les de tornar a demanar.

Cake: *framework* de desenvolupament amb PHP que proveeix una extensa arquitectura per desenvolupar, mantenir, i compilar aplicacions.

CSS: *Cascading Style Sheets*. Llenguatge de fulles d'estil usat per a descriure presentacions gràfiques mitjançant un document de marques.

DAFO: en anglès SWOT (*Strengths, Weaknesses, Opportunities, Threats*) és una metodologia per estudiar la situació competitiva d'una empresa en el seu mercat (situació externa) i de les característiques internes (situació interna) d'aquesta, a efectes de determinar les seves Debilitats, Amenaces, Fortaleses i Oportunitats.

Framework: Arquitectura de programari que modela les relacions generals de les entitats del domini. Proveeix una estructura i una metodologia de treball la qual estén o utilitza les aplicacions del domini.

Frontend / Backend: Terme general que es refereix al inici i al final d'un procés. El *frontend* o *front-end* és el responsable de acollir la entrada de dades de l'usuari mitjançant varis formularis i processant-los perquè el *backend* o *back-end* els pugui usar. El *frontend* és un tipus d'interfície entre l'usuari i el *backend*.

Game Tester: Persona que testeja el joc.

Gateway: Porta entre dos programes o sistemes. Actua com un portal que permet compartir dades mitjançant un certs protocols.

GNU: acrònim recursiu de "*GNU is Not Unix*". Sistema operatiu compost enterament de software gratuït. Com el seu nom indica, no té res a veure amb Unix.

GUI: *Graphical User Interface*. Tipus de interfície d'usuari que permet als usuaris interactuar amb programes de maneres diferents.

Helper: Funcions d'objectes integrades en Symfony.

Hosting: *Web Hosting Service*. Tipus de servei d'Internet que permet als usuaris accedir al lloc web per mitjà de Internet.

HTML: *Hyper Text Markup Language*. És el llenguatge predominant de les pàgines web.

HTTP: *Hypertext Transfer Protocol*. Protocol a nivell d'aplicació per distribuir sistemes d'informació d'hipermèdia.

Javascript: Llenguatge de programació orientat a objectes i està basat en el llenguatge *script*. S'usa per habilitar accés i modificacions als objectes en l'ordinador del client.

Join: En l'àmbit de les bases de dades és una connexió que relaciona dues taules.

Layout: Plantilla que es fa servir amb el patró decorador per decorar un contingut.

Login: Identificar-se i accedir al joc.

Logout: Sortir del joc i perdre la sessió.

Link: Vincle que direcciona a algun lloc.

Mixin: En el llenguatge orientat a objectes, *Mixin* és una classe que proveeix certes funcionalitats per ser afegides a les subclasses i d'aquesta manera no haver-la d'instanciar.

MVC: Patró Model Vista Controlador.

MVCC: *Multiversion Concurrency Control*. Aplicació que habilita mètodes de control concurrent que s'usen per a donar concurrència a bases de dades, arxius, etc.

Nested transactions: És una transacció que és iniciada per una instrucció que es troba dins d'una altra transacció encara existent.

ORM: *Object-Relational Mapping*. Tècnica de programació que converteix les dades entre sistemes incompatibles de bases de dades relacionals i llenguatges de programació orientats a objectes.

PHP: *HyperText Processor*. És usat generalment per a desenvolupament web per a produir pàgines web dinàmiques. El codi PHP és traduït per el servidor a text HTML i *JavaScript*, que se li envia al client com a text pla.

Plugin: També anomenat *plug-in*, *addin*, *add-in*, *addon*, *add-on*, *snap-in* o *snapin* consisteix en un programa d'ordinador que interactua amb una aplicació host (com per exemple un navegador o un client de correu electrònic) per proveir funcions específiques sota demanda.

Point-in-time recovery: Sistema en el que un conjunt de dades o bé unes configuracions determinades poden ser restaurades des d'un punt de recuperació preestablert.

POO: *Originalment Object Oriented Programming* (OOP). Programació Orientada a Objectes.

Prototype: *Framework JavaScript* que incorpora utilitats i funcions lliures per fer més ric i fàcil el desenvolupament web.

RIA: *Rich Internet Application*. Aplicació web que té la majoria de les característiques de les aplicacions d'escriptori.

Rollback: En les tecnologies de les bases de dades, és una operació que retorna la base de dades a algun estat previ determinat.

Rollover: Efecte de *JavaScript* que s'activa al passar el ratolí per sobre.

RPG: *Role-Playing Game* (joc de rol).

Savepoint: Punt de guardat.

Script: Llenguatge de programació que permet control a una o més aplicacions software.

SGBD: Sistema Gestor de Base de Dades.

sfUser: Classe característica de Symfony que és emmagatzemada en sessió per tal de gestionar-les (substitut que ofereix el *framework* a treballar a la sessió).

Stalled: Referent als objectes que estan inacabats o sense processar.

Tablespace: Zona d'emmagatzematge en la que les dades actuals d'una base de dades estan contingudes.

Tunning: Modificació.

Unicode: Estàndard de la indústria de la computació que permet als ordinadors interpretar i manipulat text en la majoria dels sistemes d'escriptura.

URI: *Uniform Resource Identifier*. Composició de caràcters que s'usa per identificar un nom o un recurs a Internet.

URL: *Uniform Resource Locator*. Subconjunt del *URI* que especifica en quin lloc un recurs està disponible i la manera d'arribar-hi.

versió alfa (alpha): Primera versió tancada d'una aplicació en la que es permet accedir a uns certs usuaris per tal de testejar-la.

Web: *World Wide Web* (WWW). Sistema interrelacionat de documents d'hipertext continguts a Internet.

Website: Lloc web. Conjunt de pàgines web, imatges, vídeos, etc. Correlacionades que representen adreces d'un mateix nom de domini.

Widget: També anomenat control, és un element de la GUI que disposa d'informació que pot ser canviada en qualsevol moment per l'usuari, com una finestra o una caixa de text.

YAML: acrònim recursiu de *YAML Ain't Markup Language*. Format de dades serialitzable comprensible per a les persones que pren conceptes de la programació de C, Perl, Python i XML.

Zend: *Framework* basat en la simplicitat i orientat a objectes. Es basa en oferir a les aplicacions seguretat i serveis ja funcionals.

E.2- Glossari de termes relatius als jocs de rol.

Ban: Estat de penalització a un jugador.

CVC: *Clan Versus Clan*. Lluita entre clans.

Drop: Objecte llençat per personatges del joc al morir.

Gear: Objecte del joc que s'usa com a equipament per als personatges.

Gift: Objecte que es dona com a recompensa.

Item: Objecte.

Jobclass: Professi3 de personatges del jugador.

Nick: Identificador de l'usuari en el joc enfront els demés usuaris.

NPC: *Non-Playing Character*. Personatge del joc que és controlat per el Sistema i amb el qual els usuaris poden interactuar d'una manera determinada.

PVP: *Player Versus Player*. Lluita entre jugadors.

Ranking: Llista ordenada per nivell.

Reset: Retornar quelcom a l'estat inicial.

Shortcutkey: Tecles o accessos ràpid de la interfície del joc.

Skill: Habilitat dels personatges del jugador.

Upgrade: Millora.

WCC: *WarComrades Character*. NPC de WarComrades.

Bibliografía

PHP:

- [1] www.jourmoly.com.ar/introduccion-a-mvc-con-php-primera-parte/, JOURMOLY, Introducción a MVC con PHP, primera parte. *September 19, 2007.*
- [2] www.desarrolloweb.com/manuales/58/, desarrolloweb.com, *Manual de PHP5.*
- [3] www.phpbench.com/, phpbenchmark, *the php code benchmark.*

JavaScript:

- [4] www.webreference.com/programming/javascript/mk/column2/, WebReference, *How to Drag and Drop in JavaScript.*
- [5] www.desarrolloweb.com/manuales/xajax.html, desarrolloweb.com, *Trabajo con AJAX en PHP.*
- [6] jqueryfordesigners.com/jquery-tabs/, jquery, *jquery for designers.*

Symfony:

- [7] www.symfony-project.org/, *Symfony, Open-Source PHP Web Framework.*
- [8] Fabien Potencier, *Practical Symfony 1.2 for Propel.* SESSIOLABS Books.
- [9] Fabien Potencier, François Zaninotto, *Symfony 1.2, la guía definitiva.* SESSIOLABS Books.

Altres:

- [10] www.en.wikipedia.org/, Wikipedia, *The Free Encyclopedia*.
- [11] www.noupe.com/css/47-excellent-ajax-css-forms.html, noupe, 47+ *Excellent Ajax CSS Forms*.
- [12] blog.objetivonegocio.com/2009/07/estudio-mercado/, Objetivo Negocio, *Como realizar un estudio de mercado*.
- [13] usability.gov/guidelines/index.html, usability.gov, *guidelines*.