

THE MERCHANT

TFC APLICACIONS MÒBILS

MEMÒRIA

JOAN MANUEL GUTIÉRREZ VÁZQUEZ

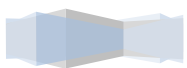
COORD. JOAN VICENT ORENGA



A l'Anna per ser-hi, per creure-hi, per apretar i per tots
els sopars que li dec

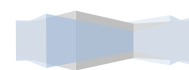
Als meus tres bitxos, pel temps perdut, que no tornarà,
però que es compensarà

Als amics, per preocupar-se, i a en Dani, per les seves
idees

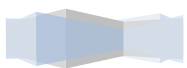


1 Taula de contingut

2	Introducció als jocs on-line per smartphone	5
2.1	Clash of clans	5
2.2	Dragons age legends	5
2.3	Juego de mafiosos	6
3	Objectius	7
4	Funcionalitats principals.....	7
4.1	Pantalla d’inici – Registre d’usuari	7
4.2	Gestió principal de funcions.....	7
4.3	Àrea social	7
4.4	Gestió de tripulació	7
4.5	Gestió de la nau.....	7
4.6	Navegació.....	8
4.7	Comerç	8
5	Planificació del projecte	8
6	Anàlisi funcional	9
6.1	Actors	9
6.2	Model de casos d’ús.....	9
6.3	Fitxes de casos d’ús	10
7	Anàlisi funcional	10
7.1	Experiència del jugador	10
7.2	Evolució procedural de l’entorn.....	10
7.3	Prototipus d’alta fidelitat	10
7.3.1	Inici de l’aplicació, registre o inici de sessió	11
7.3.2	Pantalla principal (Home).....	12
7.3.3	Àrea Social	13
7.3.4	Àrea de comerç (trade)	14
7.3.5	Informació jugador	15
7.3.6	Informació de navegació	17
8	Disseny tècnic.....	19
8.1	Diagrama de components de l’arquitectura	19
8.1.1	Servidor	19



8.1.2	Dispositiu mòbil.....	19
8.1.3	Google Cloud Messaging	19
8.2	Interfície amb el servidor	20
8.3	Disseny relacional de la base de dades	21
8.3.1	Diagrama E/R.....	21
8.3.2	Taules	22
8.4	Diagrama de classes	26
8.4.1	Descripció de classes	27
8.5	Diagrames de seqüència	29
8.5.1	Registre de l'usuari.....	29
8.5.2	Enviament de mail.....	30
8.5.3	Enviament de chat.....	30
8.5.4	Navegació	31
9	Implementació	31
9.1	Google Cloud Messaging	31
9.1.1	Registre al Google Cloud Messaging	31
9.1.2	GMC al servidor	32
9.1.3	GMC al dispositiu.....	33
9.2	Localització	33
9.3	Llistes d'objectes. ListViews i Adapters.....	33
9.4	InputDialog modals	36
9.5	Missatges Toast.....	37
9.6	RESTful asíncron	37
10	Manual d'usuari	40
10.1	Pantalla d'inici	40
10.1.1	Creació de nou usuari.....	41
10.1.2	Inici de sessió amb usuari existent.....	42
10.1.3	Menú principal	42
10.1.4	Àrea social	43
10.1.5	Navegació	44
10.1.6	Àrea tripulació.....	45
10.1.7	Àrea Flota	46
10.1.8	Àrea comerç	47
11	Conclusions	49



12	Futures millores fora de l'àmbit del projecte.....	49
13	Fons d'informació	50
13.1	Internet.....	50
13.2	Bibliografia	50



2 Introducció als jocs on-line per smartphone

Els telèfons mòbils han gaudit de funcionalitats d'oci des de els principis de la popularització del mòbil pel públic en general. Començant per l'Snake de Nokia i seguint amb els principis de l'smartphone compatible amb Java, multitud de jocs han estat a disposició dels usuaris.

Amb la irrupció al mercat dels smartphones amb connexió a Internet un nou tipus de joc ha estat possible, aquell que uneix el caràcter "casual" dels primers jocs amb un component online i una complexitat afegida que pugui recompensar els jugadors en funció de la seva dedicació. Es a dir, jocs en que no es guanya o es perd en qüestió de minuts, sinó que poden tenir una evolució llarga però discontinua.

Es així com s'introdueixen els jocs denominats "persistents" que inclouen una part d'interacció de l'usuari i una part de gestió online centralitzada. Aquests poden, a mes, incloure un aspecte social afegit que permet als usuaris comparar els seus progressos amb es d'altres usuaris, i competir de manera directa (PvP) com indirecta (PvE).

Veiem a continuació alguns exemples de jocs per a smartphone:

2.1 Clash of clans

Es tracta d'un joc multi jugador, molt similar a "Age of empires", amb un component PvP molt destacat que inclou aspectes de city-building i un molt destacat aspecte de monetització de l'experiència, així com alguns elements de col·laboració entre jugadors.

Les activitats es realitzen mitjançant cicles d'acció i on l'usuari ha d'esperar per poder realitzar una nova jugada



2.2 Dragons age legends

Aquest es un joc d'estratègia per torns on el jugador controla un grup d'herois lluitant contra missions predefinides (PvE) o contra altres jugadors per pujar en el ranking, Es pot destacar



aquest joc per l'ús que fa de la monetització. Els jugadors no compren directament els herois que volen, sinó que compren possibilitats aleatòries d'aconseguir herois, així doncs la sort s'introdueix ja en la mecànica inicial amb un alt component adictiu.

2.3 Juego de mafiosos

Joc d'estratègia ambientat als anys 30, i amb similituds amb "El Padrino". Amb molt poca component gràfica, on es tracta de complir missions per pujar nivells i poder realitzar accions més complexes.

Tot i que inicialment podria semblar el contrari, és un joc que genera una adicció prou gran i on la component social es va fent més necessària a mesura que s'avança al joc (integració en un grup mafiós, guerres entre clans contraris)



3 Objectius

L'objectiu d'aquest document és descriure el joc **the Merchant**, tant des del punt de vista tècnic com funcional, així com proporcionar el seu manual d'usuari.

Es tracta d'un joc multi jugador, ambientat a l'espai, que té com a objectiu comerciar amb productes entre planetes. Cada planeta actua de manera independent, aplicant unes fluctuacions als preus que permeten guanyar o perdre molt ràpidament, i avançar o perdre posicions al ranking. També inclou una part social (chat i missatgeria interna)

L'elecció d'aquest tipus de desenvolupament m'ha permès utilitzar moltes de les tècniques que he après durant els estudis d'enginyeria tècnica, i també m'ha obert la porta a un molt professional que cada dia guanya més importància a la nostra societat.

Des del punt de vista tècnic, he fet servir diferents tecnologies, com ara Java per al desenvolupament Android, C# amb MVC per al servidor, VB.net com a generador de l'univers programàtic i motor del món virtual, REST (json) com a protocol de comunicació entre entorn, i la utilització del serveis cloud de Google i Amazon.

Ha estat un inici complicat amb el món Android, amb una corba d'aprenentatge llarga i amb moments de frustració, però un cop desbloquejats aquets problemes, puc avaluar l'experiència com profitosa i engrescadora, que segur que aprofitaré al món laboral, i que per suposat, no s'aturarà amb aquest projecte.

4 Funcionalitats principals

4.1 Pantalla d'inici – Registre d'usuari

Quan arrenca l'aplicació, s'ha d'escollir entre registrar-se com a usuari o accedir a l'aplicació amb un usuari existent.

4.2 Gestió principal de funcions

Menú principal de l'aplicació. Un panell de control on es pot veure on ets actualment, cap a on vas i les teves estadístiques principals

4.3 Àrea social

Secció on l'usuari pot contactar amb usuaris connectats, enviar missatges i xats i mirar el ranking general

4.4 Gestió de tripulació

En aquesta pantalla, l'usuari pot veure la tripulació assignada durant la creació de l'univers programàtic, així com les seves estadístiques i foto.

4.5 Gestió de la nau

En aquesta pantalla, l'usuari pot veure els mòduls assignats a la seva nau durant la creació de l'univers programàtic, així com les seves estadístiques i aparença.



4.6 Navegació

Pantalla de visualització dels planetes més mes propers, i de la distància on es troben. L'usuari podrà escollir si viatja a algun d'ells.

4.7 Comerç

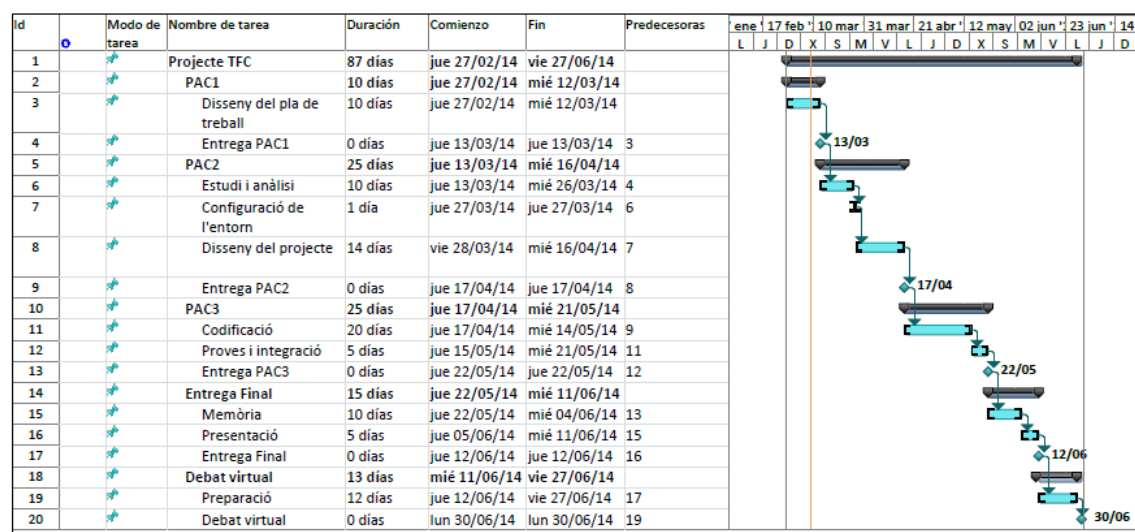
Pantalla de gestió de compra i venda d'objectes, amb certes restriccions preestablertes.

5 Planificació del projecte

S'ha realitzat la planificació tenint en compte els milestones establerts pels consultors de l'assignatura, sent aquetes les dates principals:

Data clau	Document	Descripció
12/03/14	PAC1	Elaboració del pla de treball, detall dels requisits i planificació del projecte.
16/04/14	PAC2	Estudi de l'arquitectura i anàlisi general del projecte. Instal·lació i configuració del entorn de programació.
21/05/14	PAC3	Implementació i codificació del projecte.
11/06/14	PAC4	Data de finalització del projecte amb l'entrega de la memòria, la presentació final i el producte final.
27/06/14	Debat virtual	Defensa del projecte.

Diagrama de Gantt de la planificació temporal de les tasques que componen el projecte:



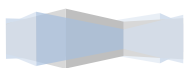
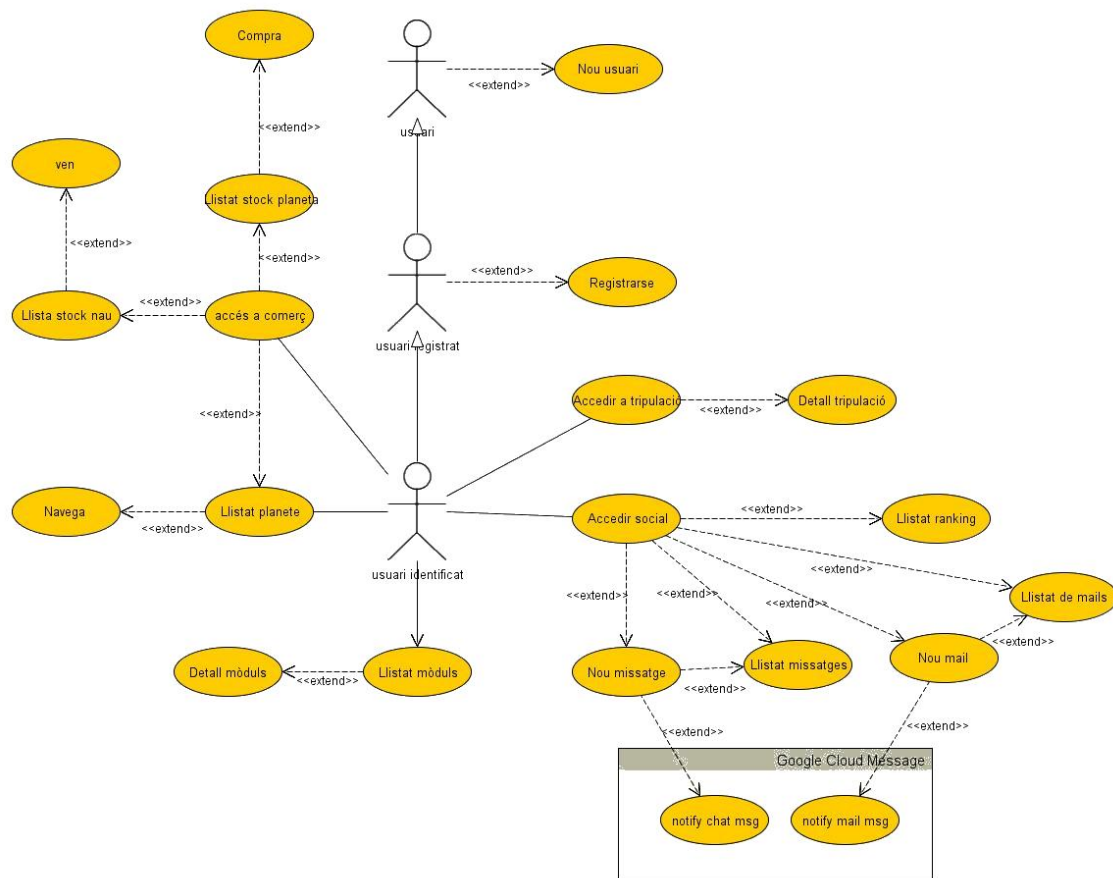
6 Anàlisi funcional

6.1 Actors

Usuari: Representa el dispositiu client de l'aplicació (Android), controlat per l'usuari que el fa servir

Sistema: Part servidora, que gestiona les comunicacions, la persistència de les dades, i la variabilitat de l'univers actual mitjançant uns paràmetres aleatoris que fan que tant l'usuari com els elements virtuals amb els que interactuïn ho facin de forma dinàmica.

6.2 Model de casos d'ús



6.3 Fitxes de casos d'ús

7 Anàlisi funcional

7.1 Experiència del jugador

Un cop iniciat el joc, el jugador pot enregistrar-se al sistema com un nou jugador o fer login amb les seves credencials existents. Des de la pantalla principal el jugador pot accedir directament a totes les pantalles secundàries, que es on es durà a terme tota la gestió del joc. Això inclou la pantalla social, on el jugador es relaciona amb altres jugadors mitjançant un chat i un sistema de missatgeria, la pantalla de navegació, on el jugador pot escollir la seva destinació a cada moment, la pantalla de comerç, on el jugador compra i ven mercaderies i la pantalla de la nau, on el jugador pot veure les capacitats de la seva nau i les millores instal·lades

L'experiència del jugador a "The Merchant" està basada en el tràfic de mercaderies en un univers dinàmic, es a dir, la intervenció dels jugadors té una influència en el joc, però alhora el propi univers té uns mecanismes d'evolució que s'encarreguen de dinamitzar les condicions. Això permet introduir un element d'exploració ja que els diferents planetes van canviant de manera autònoma amb el temps.

7.2 Evolució procedural de l'entorn

Els diversos planetes tenen uns mecanismes d'evolució que permeten que la seva situació vagi evolucionant, incrementant o disminuint els seus stocks de mercaderies i també canviant la demanda i oferta que aquets tenen, aquest aspecte es controla mitjançant un conjunt de fórmules matemàtiques que el servidor de l'aplicació calcula a cada cicle.

7.3 Prototipus d'alta fidelitat

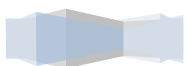
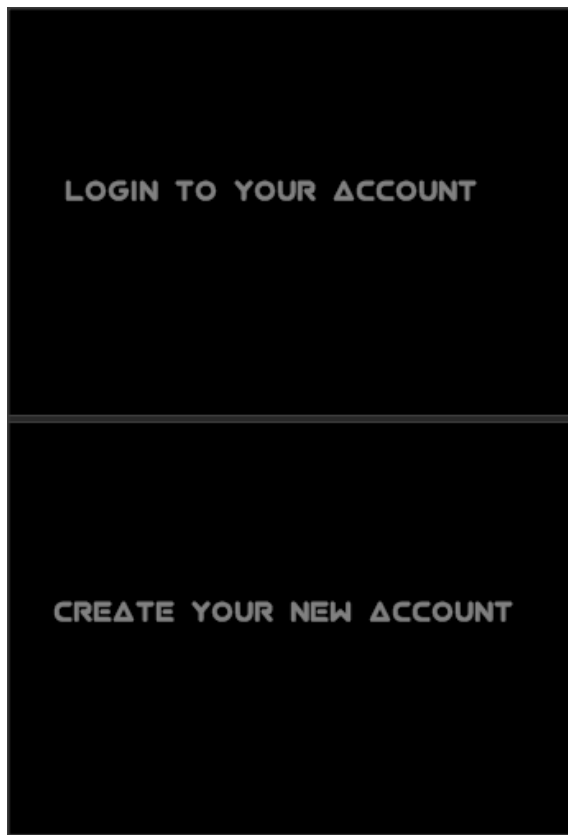
A continuació detallo un prototipus realitzat amb l'eina *Just in Mind Prototyper*, representant el nombre més ampli possible de funcionalitats.

Estèticament, aquesta eina pot generar algunes limitacions (fonts, visualització de graelles) però permet validar el flux d'informació i la usabilitat de la Interface.

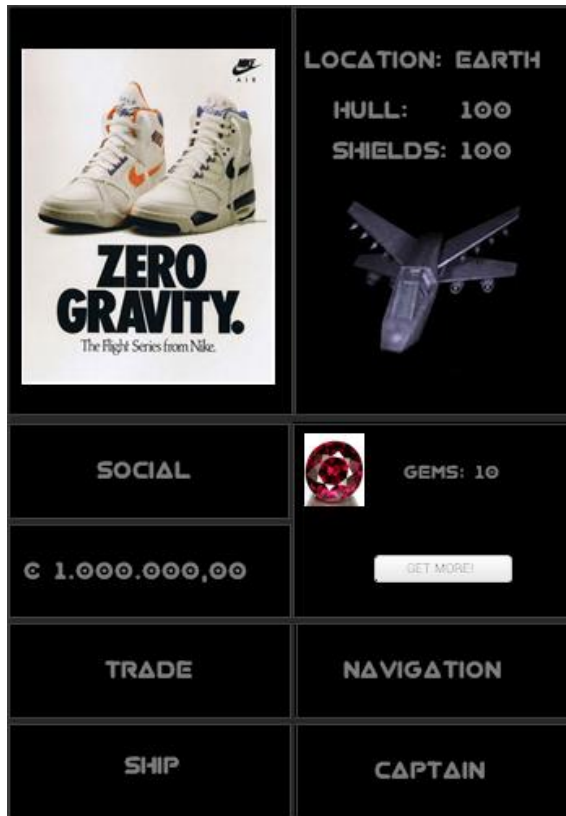
També remarcar que les imatges associades a les naus, tripulació, publicitat, ... són només indicatives d'ubicació, i seran reemplaçades a la versió definitiva.



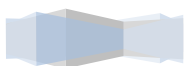
7.3.1 Inici de l'aplicació, registre o inici de sessió



7.3.2 Pantalla principal (Home)



Conté l'accés a cada una de les àrees gestionades al joc. També inclou la part de monetització de l'aplicació (publicitat / missatges generals) i gestió de Gemes (super-moneda)



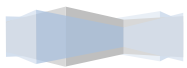
7.3.3 Àrea Social



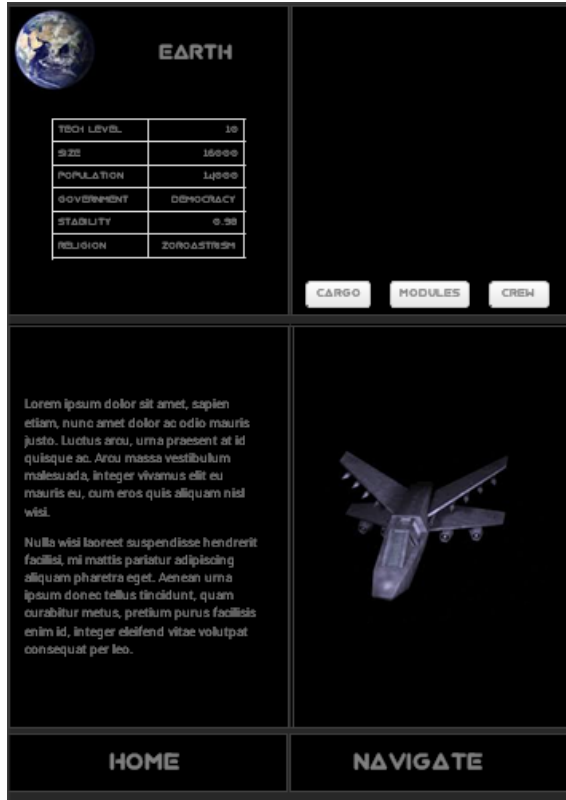
Visualització general de l'àrea social



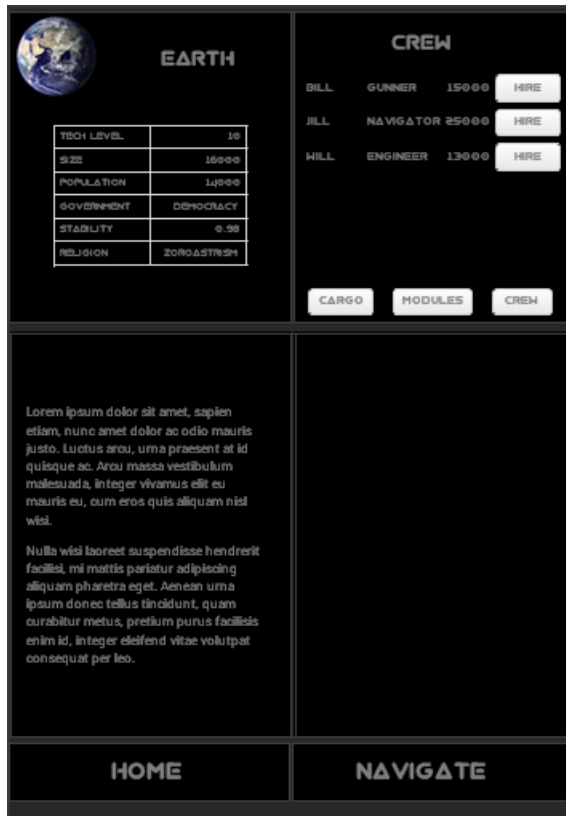
Xat públic (amb tots el usuaris connectats)



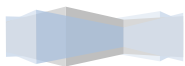
7.3.5 Informació jugador



Informació sobre la tripulació que es pot contractar en un planeta



Mostra la informació del planeta actual, informació bàsica sobre la nau, i accés a la càrrega, mòduls i tripulació actual





BIOGRAPHY

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus aro, urna praesent sit id quisque ac. Arou massa vestibulum molestuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisi wisi.

Nulla wisi laoreet suspendisse hendrerit facilisi, mi mattis pariatur adipiscing aliquam pharetra eget. Aenean urna ipsum donec tellus tincidunt, quam curabitur metus, pretium purus facilisis

CREW INFO

BILL

Granary	+3
Hand to Hand	+2



CREW MEMBERS

Bill
Jim
Walt

HOME

Informació sobre la tripulació (capità i personal contractat)

MODULES

Ammor 1
Glass 1
Shield 1


STAT SUMMARY


Ammor	150
Shield	200
Energy Damage	55
Energy Regeneration	5
Cargo	15000
Shield Regeneration	2
Fuel	3000

MODULE INFO

SHIELD 1

Shield	+3
Shield regeneration	+1

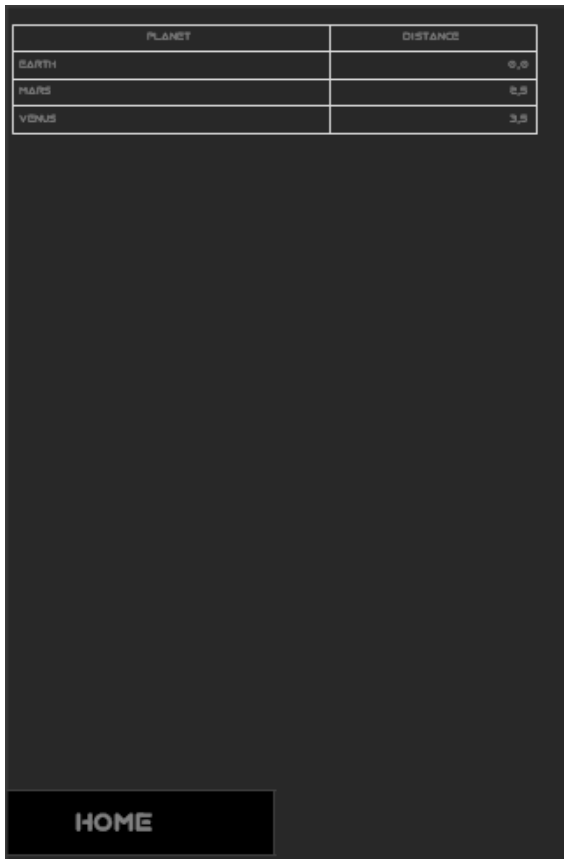




HOME

Informació sobre mòduls actualment disponibles a la nau

7.3.6 Informació de navegació



Llista de planetes a l'abast



Informació del planeta on estàs actualment





EARTH

TECH LEVEL	10
SIZE	10000
POPULATION	10000
GOVERNMENT	DEMOCRACY
STABILITY	0.99
RELIGION	ZOROASTRIAN

MODULES

ARMOR 1	1 1000	<input type="button" value="BUY"/>
SHIELD 1	1 2000	<input type="button" value="BUY"/>
GUNS 1	1 3000	<input type="button" value="BUY"/>

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisi wisi.

Nulla wisi laoreet suspendisse hendrerit facilisi, mi mattis partur adipiscing aliquam pharetra eget. Aenean urna ipsum donec tellus tincidunt, quam curabitur metus, pretium purus facilisis enim id, integer eleifend vitae volutpat consequat per leo.

HOME
NAVIGATE

Informació del planeta on estàs actualment (Mòduls)



VENUS

TECH LEVEL	7
SIZE	3500
POPULATION	0000
GOVERNMENT	ANARCHIST
STABILITY	0.00
RELIGION	NONE

REMOTE TRADE NOT ALLOWED

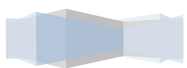


Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu massa vestibulum malesuada, integer vivamus elit eu mauris eu, cum eros quis aliquam nisi wisi.

Nulla wisi laoreet suspendisse hendrerit facilisi, mi mattis partur adipiscing aliquam pharetra eget. Aenean urna ipsum donec tellus tincidunt, quam curabitur metus, pretium purus facilisis enim id, integer eleifend vitae volutpat consequat per leo.

HOME
NAVIGATE

Informació d'un planeta remot (notar el bloqueig d'interacció)



8 Disseny tècnic

8.1 Diagrama de components de l'arquitectura

L'arquitectura del sistema la conformen diverses tecnologies que detallem a continuació:

- Servidor de l'aplicació
- Gestor d'univers virtual
- Dispositiu mòbil android
- Servidor de Google cloud Messaging

8.1.1 Servidor

Es tracta d'un servidor Windows Server 2008 amb IIS 7.5, que emmagatzema les dades (base de dades MSAccess) i les imatges.

Els webservices han estat desenvolupats en C# i es serveixen mitjançant ASP.NET i MVC3.

Adicionalment s'executa un procés que gestiona el cicle del joc, on es realitza l'aplicació de les fórmules per al càlcul de necessitats dels planetes, l'increment de saldo, així com l'evolució de desplaçaments de les naus.

Aquest procés s'ha desenvolupat en VB6 utilitzant la cap de dades DataEnvironment.

8.1.2 Dispositiu mòbil

Es tracta del client del joc de l'aplicació.

Aquest client es comunicará amb el servidorr GCM per identificar-se i per rebre missatges PUSH i amb el servidor que proporciona webservice MVC via HTTP encapsulat amb JSON.

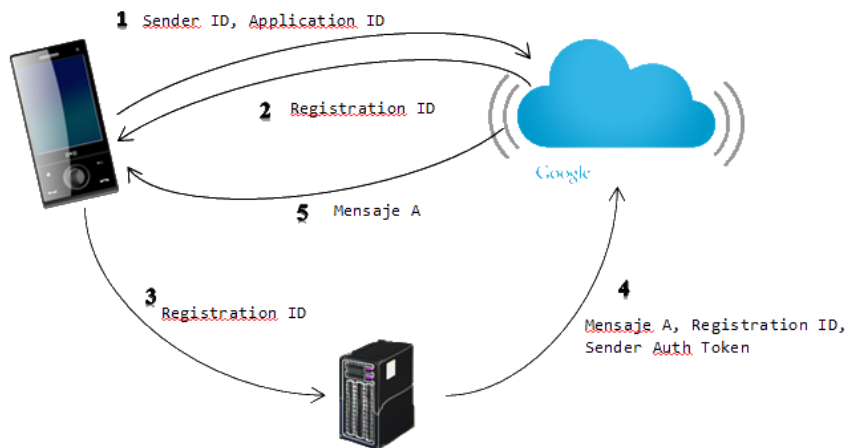
No hi ha cap emmagatzemament local de dades, ja que les dades del servidor canvien constantment.

8.1.3 Google Cloud Messaging

Google Cloud Messaging és un servei que proporciona Google que permet fer notificacions PUSH al clients, evitat que hi hagi una comunicació constant entre el client i el servidor, i així reduir el consum d'ample de banda i bateria.

El procés seguit en la utilització del servei és l'indicat al gràfic inferior:





1. El client indica al servidor GCM quin és el seu identificador de client i amb quina aplicació s'ha de comunicar.
2. El servidor GCM retorna un identificador únic a on el servidor enviarà la comunicació
3. El client Android informa al servidor de l'aplicació quin és el seu identificador
4. El servidor de l'aplicació envia missatges al servidor GCM indicant el destinatari
5. El servidor GCM envia la notificació PUSH al dispositiu Android.
6. A partir d'aquest moment, el dispositiu Android realitza les tasques pertinents (sincronització) amb el servidor de l'aplicació

8.2 Interfície amb el servidor

El servidor enviarà al dispositiu mòbil notificacions PUSH per indicar que ha d'executar una petició HTTP.

Aquestes peticions PUSH s'activen quan hi ha un nou missatge de mail, un nou missatge de xat o si hi ha un canvi en la posició de la nau.

El servidor, mitjançant una estructura MVC 3, s'encarregarà de tractar les peticions del client i retornarà informació en format JSON, o determinats codis de escapament.

Les accions controlades pel servei MVC són les següents:

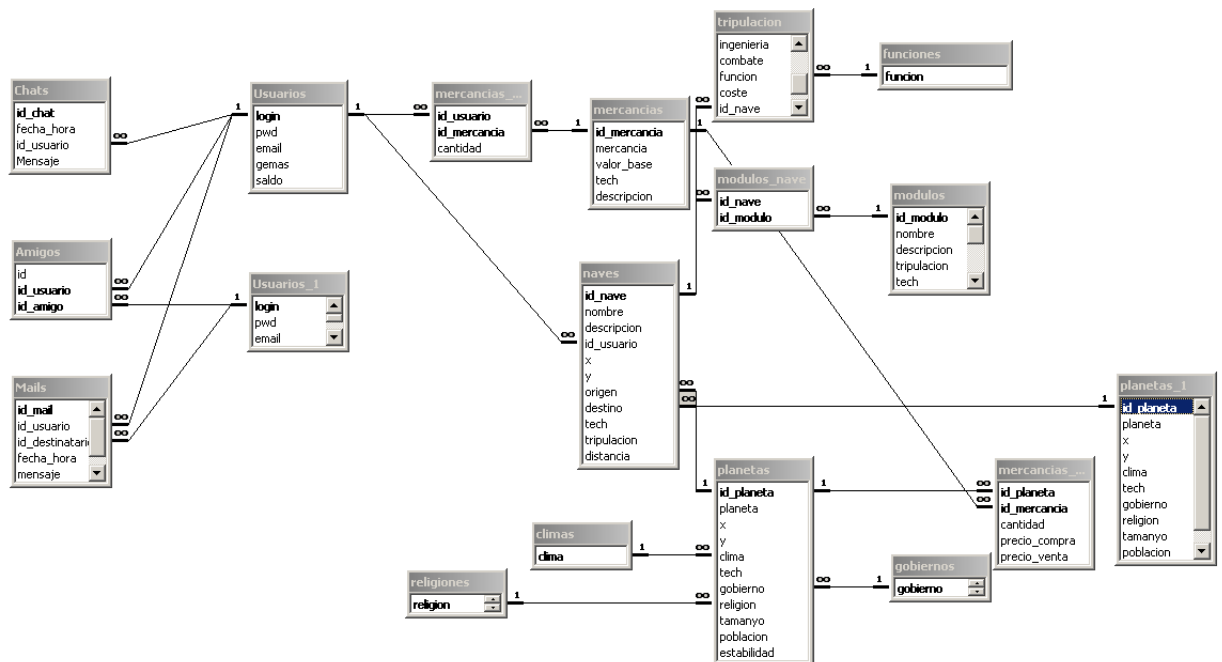
Routing MVC	Parametre	Acció	Retorna
http://Serverip/ServiceWebREST/capitan/capitan	id d'usuari	Retorna la informació del capità assignat a la nau del jugador	JSON
http://Serverip/ServiceWebREST/cargoplaneta/cargo	JSON	Actualitza el registre de mercaderies d'usuari i de planeta (compra/venda)	codi d'escapament
http://Serverip/ServiceWebREST/cargoplaneta/cargos/	id d'usuari	Retorna mercaderies de planeta	JSON
http://Serverip/ServiceWebREST/cargousuario/cargo	JSON	Actualitza el registre de mercaderies d'usuari i de planeta (compra/venda)	codi d'escapament
http://Serverip/ServiceWebREST/cargousuario/cargos	id d'usuari	Retorna mercaderies d'usuari	JSON
http://Serverip/ServiceWebREST/chat/chat	JSON	inserta un missatge de chat a la cua. Envia missatge PUSH	codi d'escapament
http://Serverip/ServiceWebREST/chat/chats	-	Retorna llista de missatges de la cua de chat (els 10 primers)	JSON

http://Serverip/ServiceWebREST/friends/friend	JSON	Afegeix un amic a la llista d'amics	JSON
http://Serverip/ServiceWebREST/friends/friends	id d'usuari	Retorna la llista d'amics d'un jugador	JSON
http://Serverip/ServiceWebREST/mails/mail	JSON	inserta un missate de de mail.Envia missatge PUSH	codi d'escapament
http://Serverip/ServiceWebREST/mails/mails	id d'usuari	Retorna llista de missatges de mail	JSON
http://Serverip/ServiceWebREST/modules/module	JSON	Retorna les dades detallades d'un mòdul	JSON
http://Serverip/ServiceWebREST/modules/modules	id d'usuari	Retorna llista de mòduls d'un jugador	JSON
http://Serverip/ServiceWebREST/planetas/planeta	JSON	Retorna les dades detallades d'un planeta	JSON
http://Serverip/ServiceWebREST/planetas/planetas	id d'usuari	Retorna llista de planetes ordenats per proximitat al jugador	JSON
http://Serverip/ServiceWebREST/ranking/ranking	JSON	Retorna les dades detallades de ranking d'un usuari	JSON
http://Serverip/ServiceWebREST/ranking/rankings	-	Retorna el ranking de jugadors ordenat de més gran a més petit	JSON
http://Serverip/ServiceWebREST/tripulacion/tripulacion	JSON	Retorna les dades detallades d'un tripulant	JSON
http://Serverip/ServiceWebREST/tripulacion/tripulaciones	id d'usuari	Retorna llista de tripulants assignats a un usuari	JSON
http://Serverip/ServiceWebREST/Usuarios/Usuario	JSON	Retorna les dades detallades d'un usuari/ Inserta un usuari	JSON / codi d'escapament

8.3 Disseny relacional de la base de dades

8.3.1 Diagrama E/R

A continuació mostrem el diagrama d'entitat / relació del disseny de BBDD del servidor.



8.3.2 Taules

8.3.2.1 Taula climas

Taula auxiliar que conté els diferents climes que pot tenir un planeta

Climas				
Atribut	Descripció	Tipus	Longitud	Clau
Clima	Identificador del clima d'un planeta	Texte	50	s

8.3.2.2 Taula chats

Taula que emmagatzema cadascun dels missatges de chat creats per un usuari

Chats				
Atribut	Descripció	Tipus	Longitud	Clau
id_chat	identificador únic d'un missatge	autonumeric		s
fecha_hora	data i hora d'enviament	fecha/hora		
id_usuario	usuari emissor	texto	50	
mensaje	contingut del missatge	texto	200	

8.3.2.3 Taula funciones

Taula auxiliar que conté les diferents funcions d'un tripulant

funciones				
Atribut	Descripció	Tipus	Longitud	Clau
funciones	identificador de la funció d'un tripulant	texto	50	

8.3.2.4 Taula gobiernos

Taula auxiliar que conté els diferents tipus de govern per a un planeta, així com un paràmetre de variabilitat que influirà en el càlcul de preus i futures necessitats

gobiernos				
Atribut	Descripció	Tipus	Longitud	Clau
gobiernos	identificador d'un tipus de govern d'un planeta	Texto	50	s
variabilidad	variabilitat d'aquest planeta (pot intervenir en els calculs de necessitat)	Double		



8.3.2.5 Taula mails

Taula que conté cadascun dels mails enviats a un destinatari per par d'un jugador.

mails				
Atribut	Descripció	Tipus	Longitud	Clau
id_mail	identificador únic d'un missatge	autonumeric		s
id_usuario	usuari emissor	Texto	50	
id_destinatario	usuari destinatari	Texto	50	
fecha_hora	data i hora d'enviament	fecha/hora		
mensaje	contingut del missatge	Texto	200	

8.3.2.6 Taula mercancias

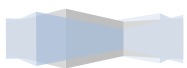
Taula de mercaderies, on trobem informació bàsica per a la compra (tipus de tecnologia necessària per accedir-hi, descripció i preu inicial, utilitzat durant la generació programàtica de l'univers virtual.

Mercancías				
Atribut	Descripció	Tipus	Longitud	Clau
id_mercancia	identificador únic de la mercaderia	autonumeric		s
Mercancías	nom de la mercaderia	texto	50	
valor_base	cost inicial	double		
Tech	nivell tecnològic necessari per ser comprat / venut	double		
Descripción	Descripció de la mercaderia	texto	50	

8.3.2.7 Taula modulos

Conté la informació relativa a cada un dels mòduls disponibles per ser vinculat a una nau. Aquest tipus de mòdul pot estàr vinculat a més d'una nau.

modulos				
Atribut	Descripció	Tipus	Longitud	Clau
id_modulo	identificador únic del mòdul	autonumeric		s
nombre	nom del mòdul	texto	50	
descripcion	descripció del mòdul	texto	255	
tripulacion	capacitat de tripulació	double		
tech	tecnologia que aporta	double		



carga	càrrega que permet	double		
motor	capacitat de motor	double		
armadura	protecció del fuselatge	double		
escudos	escut del mòdul	double		
recarga	temps necessari per recuperar-se en un atac	double		
ataque	capacitat d'atac	double		
coste	cost del mòdul	double		

8.3.2.8 Taula naues

Taula que emmagatzema les naus. Cada una d'aquestes naus està assignada només a un usuari

La distància anirà variant en cada cicle del joc sempre i quant el destí estigui assignat.

naues				
Atribut	Descripció	Tipus	Longitud	Clau
id_nave	identificador únic de la nau	autonumeric		s
nombre	nom de la nau	texto	50	
descripcion	descripció de la nau	texto	255	
id_usuario	usuari assignat	texto	50	
x	posició x actual	double		
y	posició y actual	double		
origen	planeta d'origen	double		
destino	planeta de destí	double		
tech	tecnologia que aporta la nau	double		
tripulacion	capacitat de tripulació	double		
distancia	distancia actual fins a l'objectiu	double		

8.3.2.9 Taula planetes

Taula de planetes, on emmagatzemem els paràmetres utilitzats pel càlcul així com la seva posició

planetes				
Atribut	Descripció	Tipus	Longitud	Clau
id_planeta	identificació única de planeta	autonumeric		s
planeta	nom del planeta	texto	50	
x	posició x del planeta	double		



y	posició y del planeta	double		
clima	clima del planeta	texto	50	
tech	tecnologia mínima per interactuar amb ell	double		
gobierno	tipus de govern	texto	50	
religion	tipus de religió	texto	50	
tamanyo	tamany màxim de població	double		
poblacion	població actual	double		
estabilidad	estabilitat del planeta	double		

8.3.2.10 Taula religions

Diferents tipus de religions

religiones				
Atribut	Descripció	Tipus	Longitud	Clau
religion	tipus de religió	texto	50	s

8.3.2.11 Taula tripulacion

Taula que emmagatzema a cada un dels tripulants, contenint la informació relativa a les seves capacitats. Cada tripulant pot estar assignat a una única nau.

tripulacion				
Atribut	Descripció	Tipus	Longitud	Clau
id_tripulacion	identificació única de la tripulació	autonumeric		s
nombre	nom del tripulant	texto	50	
navegacion	capacitat de navegació	double		
pilotaje	capacitat de pilotatge	double		
artilleria	capacitat d'atac en nau	double		
ingenieria	capacitat de reparació de la nau	double		
combate	capacitat de combat al planeta	double		
funcion	funció del tripulant	texto	50	
coste	cost de contractació	double		
id_nave	nau assignada	double		
biografia	dbiografia del tripulant	texto	255	



8.3.2.12 Taula usuarios



8.4.1 Descripció de classes

A continuació es descriuen breument totes les classes de l'aplicació:

Classe	Funció
CaptainActivity	Gestiona l'activitat de la secció Captaion. S'encarrega de mostrar la informació i comunica amb la classe assincrona LoadTroop per obtenir les dades
CargoPlanetListAdapter	Enllaça el layout del item de la llista de mercaderies de planetes amb l'array de dades
CargoUserListAdapter	Enllaça el layout del item de la llista de mercaderies dels usuaris amb l'array de dades
CrewListAdapter	Enllaça el layout del item de la llista de tripulants amb l'array de dades
CustomButton	Control d'usuari personalitzat per mostrar la font del joc i els colors basat en el Button standard

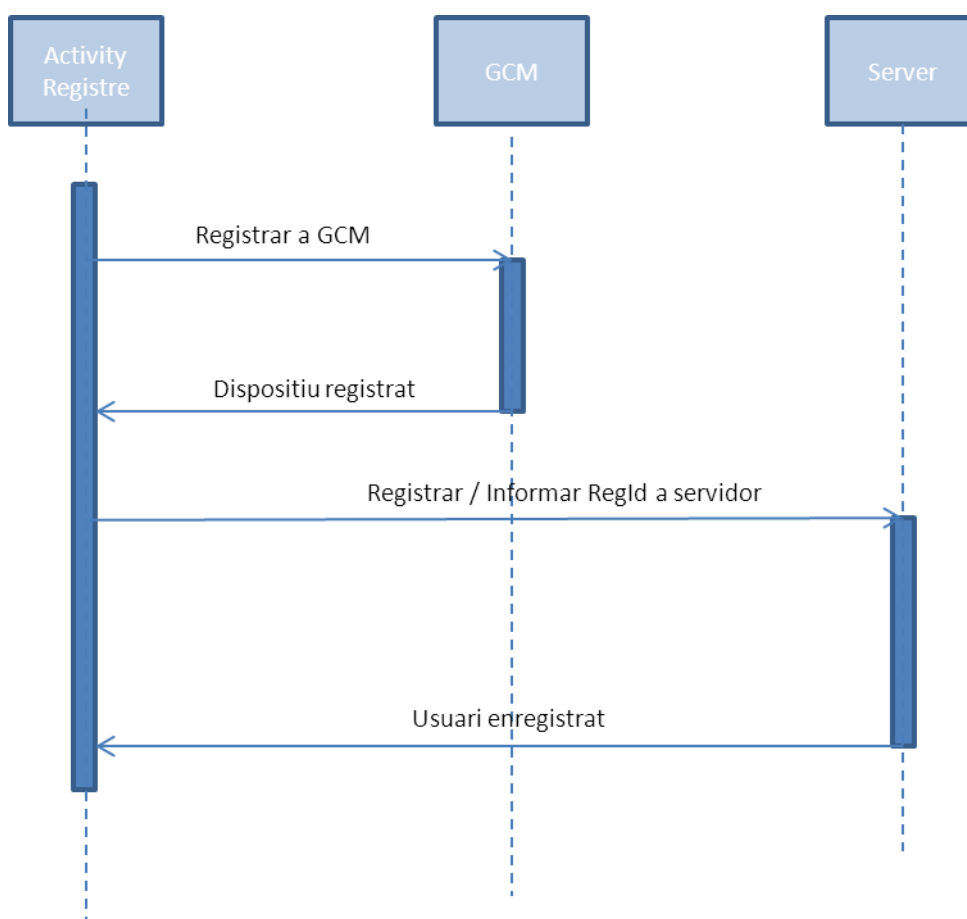
CustomTextView	Control d'usuari personalitzat per mostrar la font del joc i els colors basat en el TextView standard
ChatListAdapter	Enllaça el layout del item de la llista de missatges de chat amb l'array de dades
FriendListAdapter	Enllaça el layout del item de la llista d'amics amb l'array de dades
GCMIntentService	Classe que gestiona els events de GCM
GsonDateAdapter	Classe que gestiona els objectes JSON
JSONParser	Classe que gestiona els objectes JSON
LoadCargos	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre les mercaderies dels planetes i dels jugadors. També gestiona la petició de compra i venda de productes
LoadChats	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre els chats actius. També sol·licita la inserció en la bbdd de nous chats
LoadFriends	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre els amics del jugador. També sol·licita la inserció en la bbdd de nous amics associats
LoadMails	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre els mails del jugador. També sol·licita la inserció en la bbdd de nous mails
LoadModules	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre els mòduls de les naus
LoadPlanets	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre els planetes.
LoadRankings	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació sobre el ranking dels usuaris
LoadTroop	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació de la tripulació
LoginActivity	Activity d'inici de sessió
MailListAdapter	Enllaça el layout del item de la llista de mails amb l'array de dades
MainActivity	Activitat de la pantalla de selecció de registre / login
MenuActivity	Activitat de la pantalla principal de l'aplicació
ModuleListAdapter	Enllaça el layout del item de la llista de moduls de les naus amb l'array de dades
NavigateActivity	Activitat de la pantalla de navegació de planetes
PlanetListAdapter	Enllaça el layout del item de la llista de planetes amb l'array de dades
RankingListAdapter	Enllaça el layout del item de la llista de ranking amb l'array de dades
RegisterActivity	Activity de registre de nou usuari
Session	Classe que enmagatzema la informació de l'usuari (login, ID de GCM) així com determinades constants (server)
ShipActivity	Activity de la secció de naus
SocialActivity	Activity de la secció social

SplashActivity	Activity de la pantalla inicial (splash)
TareaInfo	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació de l'usuari referent als saldos i ubicacions
TareaLogin	Classe asíncrona que comunica amb el servidor MVC per obtenir la informació de l'usuari i registrar l'ID de GMC
TareaRegistrar	Classe asíncrona que comunica amb el servidor MVC per a registrar un nou usuari i registrar l'ID de GMC
TradeActivity	Activity de la secció de comerç
Utils	Utilitats varies (conversió de dates en JSON...)

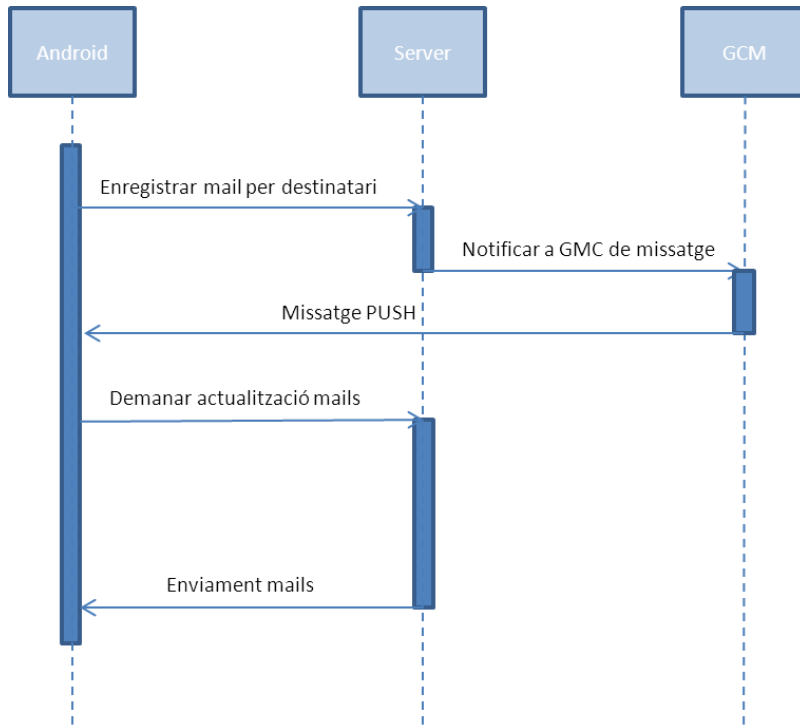
8.5 Diagrames de seqüència

A continuació detallo els diagrames de seqüència més complexes. Els altres diagrames són només de petició o d'actualització de dades.

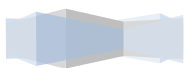
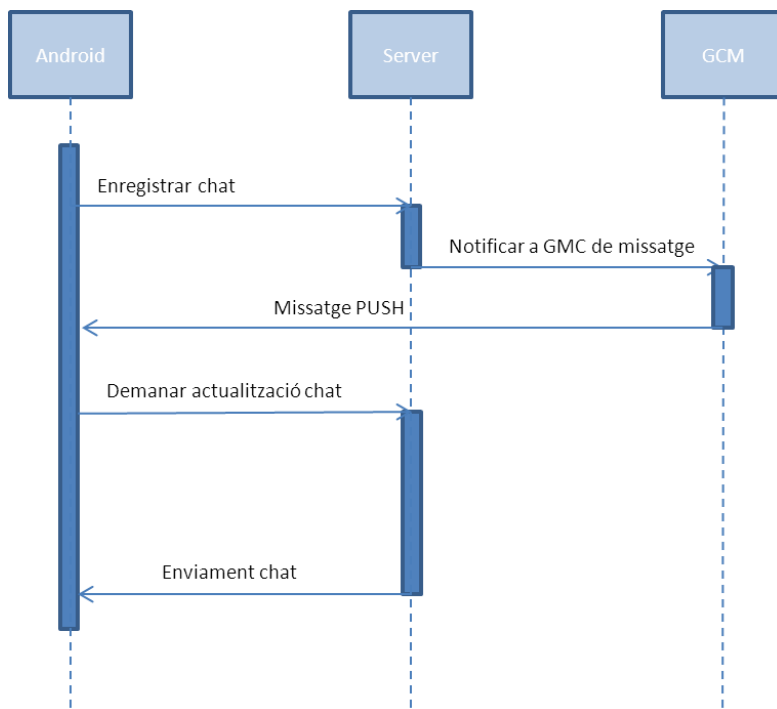
8.5.1 Registre de l'usuari



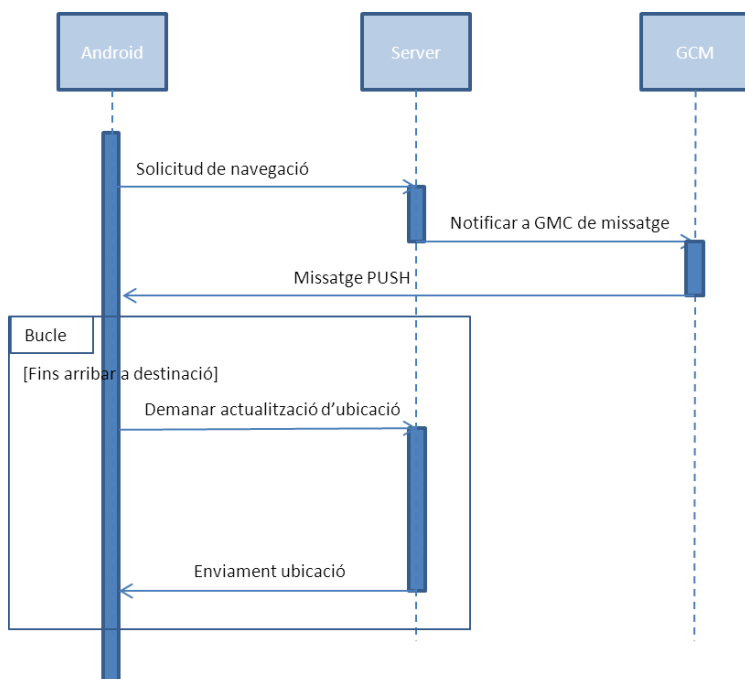
8.5.2 Enviament de mail



8.5.3 Enviament de chat



8.5.4 Navegació



9 Implementació

9.1 Google Cloud Messaging

En aquest aplicació, la sincronització de les dades amb el servidor és important, ja que hi ha un moviment de planetes, i una fluctuació de les dades que fa necessari tenir la informació detallada freqüentment. També disposa d'una àrea social que necessita refresc de dades en temps real.

En una primera versió, es va desenvolupar una versió amb una actualització basada en intervals de temps, però, després de fer les proves amb els testejadors, vàrem descobrir que consumia molta bateria i molt ample de banda. Per aquest motiu es va decidir canviar vers un model basat en missatges PUSH

9.1.1 Registre al Google Cloud Messaging

Hem registrat l'aplicació a la consola Google APIS, per obtenir els codis necessaris per realitzar la comunicació:



< TheMerchant

- Overview
- APIs & auth
 - APIs
 - Credentials**
 - Consent screen
 - Push
- Permissions
- Settings
- Support
- App Engine
- Compute Engine
- Cloud Storage
- Cloud Datastore
- Cloud SQL

OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

Create new Client ID

Compute Engine and App Engine [Learn more](#)

Client ID	889238[redacted]s.googleusercontent.com
Email address	889238[redacted]eveloper.gserviceaccount.com

[Download JSON](#)

Public API access

Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

[Learn more](#)

Create new Key

Key for browser applications

API key	AIzaSyBFLD[redacted]p-wxykXKS4
Referers	Any referer allowed
Activation date	Jun 18, 2014 12:05 PM
Activated by	totama@gmail.com (you)

[Edit allowed referers](#)
[Regenerate key](#)
[Delete](#)

9.1.2 GMC al servidor

Al servidor hem utilitzat ASP.Net amb C# . Per realitzar les comunicacions, hem utilitzat una classe existent a “the code project”

```

using System.Collections.Specialized;

public class AndroidGCMPushNotification
{
    public AndroidGCMPushNotification()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public string SendNotification(string deviceId, string message)
    {
        string GoogleAppID = "AIzaSyBFLDg[redacted]";
        var SENDER_ID = "88[redacted]";
        var value = message;
        WebRequest tRequest;
        tRequest = WebRequest.Create("https://android.googleapis.com/gcm/send");
        tRequest.Method = "post";
        tRequest.ContentType = " application/x-www-form-urlencoded;charset=UTF-8";
        tRequest.Headers.Add(string.Format("Authorization: key={0}", GoogleAppID));

        tRequest.Headers.Add(string.Format("Sender: id={0}", SENDER_ID));

        string postData = "collapse_key=score_update&time_to_live=108&delay_while_idle=1&data.message=" + value + "&data.time_to_live=" + System.DateTime.Now.ToString() + "&registration_id=" + deviceId + "";
        Console.WriteLine(postData);
        Byte[] byteArray = Encoding.UTF8.GetBytes(postData);
        tRequest.ContentLength = byteArray.Length;

        Stream dataStream = tRequest.GetRequestStream();
        dataStream.Write(byteArray, 0, byteArray.Length);
        dataStream.Close();

        WebResponse tResponse = tRequest.GetResponse();

        dataStream = tResponse.GetResponseStream();

        StreamReader tReader = new StreamReader(dataStream);

        String sResponseFromServer = tReader.ReadToEnd();

        tReader.Close();
        dataStream.Close();
        tResponse.Close();
        return sResponseFromServer;
    }
}

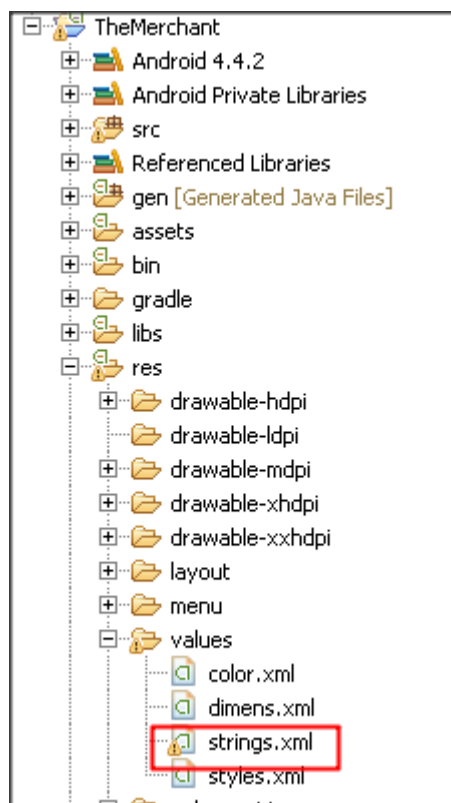
```

9.1.3 GMC al dispositiu

Seguien les instruccions de developers.android.com, utilitzem la classe `GCMIntentService` per gestionar el tractament de les notificacions

9.2 Localització

Tot i que el software està actualment només en anglès, s’han seguit les recomanacions de disseny per a la localització del soft, tenint totes les cadenes de text emmagatzemades al fitxer “strings.xml”.



Per recuperar el text i assignar-ho a un missatge en temps d’execució hem utilitzat la següent comanda:

```
ctvResultado.setText (getResources () .getString (R.string.error_creating_user) ) ;
```

Per fer-ho en temps de disseny, hem seguit l’standard del dissenyador:

```
android:text="@string/action_register"
```

9.3 Llistes d’objectes. ListViews i Adapters

Les dades de l’aplicació es mostren principalment en Listviews.

Per utilitzar correctament aquest component, hem de treballar amb tres tipus d’objectes:



- Un layout gràfic de cada element. En alguns casos on ens interessava amagar determinada informació, hem utilitzat controls ocults, amb la propietat “visibility:gone” per evitar consumir espai
- Una classe que representi un element
- Un adaptador que extengui BaseAdapter i que s’encarrega de assignar el contingut lògic a la capa de visualització
- Una classe que s’encarrega d’accedir al servei REST de manera asíncrona i generar un adaptador que nodreixi el component listview.

A continuació, trobem un exemple dels tres fitxers utilitzats a una llista molt senzilla

Adaptador que enllaça el contingut de l’array amb la part visual: FriendListAdapter.java

```
package com.jmgv.themerchant;

import java.util.ArrayList;
import java.util.HashMap;
import com.example.themerchant.R;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

public class FriendListAdapter extends BaseAdapter {
    private Context mContext;
    LayoutInflater inflater;
    private final ArrayList<HashMap<String, String>> urls;
    HashMap<String, String> resultp = new HashMap<String, String>();

    public FriendListAdapter(Context context,
        ArrayList<HashMap<String, String>> items) {
        mContext = context;
        urls = items;
    }

    public int getCount() {
        return urls.size();
    }

    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return getItem(position).hashCode();
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        TextView friend_friend;

        inflater = (LayoutInflater) mContext
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View view = inflater.inflate(R.layout.custom_list_friend, parent,
            false);

        resultp = urls.get(position);

        friend_friend = (TextView) view.findViewById(R.id.txtFriend);

        friend_friend.setText(resultp.get(SocialActivity.TAG_FRIEND_FRIEND));

        return view;
    }
}
```



```
}
```

Layout de visualització d'un item individual: Custom_list_friend.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="0sp" >

    <TextView
        android:id="@+id/txtFriend"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="5sp"
        android:layout_marginRight="3sp"
        android:layout_marginTop="3sp"
        android:text="TextView" />

</LinearLayout>
```

Clase d'accés al servei RESTFUL asíncrona, que s'encarrega d'omplir l'array i d'assignar-lo al listview: LoadRankings.java

```
class LoadRankings extends AsyncTask<String, String, String> {
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... args) {
        // Building Parameters
        List<NameValuePair> params = new ArrayList<NameValuePair>();

        // getting JSON string from URL
        String json = jsonParser.makeHttpRequest(Session.URL + "ranking/rankings"
, "GET",params);

        // Check your log cat for JSON reponse
        Log.d("Mails JSON: ", "> " + json);

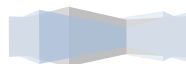
        return json;
    }

    @Override
    protected void onPostExecute(String json) {
        try {
            ranking = new JSONArray(json);

            if (ranking != null) {
                // looping through All albums
                for (int i = 0; i < ranking.length(); i++) {
                    JSONObject c = ranking.getJSONObject(i);
                    // Storing each json item values in variable
                    String friend= c.getString(TAG_RANKING_LOGIN);
                    String tripulacion=
                    String.valueOf((int)c.getDouble(TAG_RANKING_TRIPULACION));

                    String armadura=
                    String.valueOf((int)c.getDouble(TAG_RANKING_ARMADURA));

                    String escudos=
                    String.valueOf((int)c.getDouble(TAG_RANKING_ESCUDO));
```



```
String saldo=
String.valueOf((int)c.getDouble(TAG_RANKING_SALDO));

String valor=
String.valueOf((int)c.getDouble(TAG_RANKING_VALOR));

// creating new HashMap
HashMap<String, String> map = new HashMap<String,
String>();

// adding each child node to HashMap key => value
map.put(TAG_RANKING_LOGIN, friend);
map.put(TAG_RANKING_TRIPULACION, tripulacion);
map.put(TAG_RANKING_ARMADURA, armadura);
map.put(TAG_RANKING_ESCUDO, escudos);
map.put(TAG_RANKING_SALDO, saldo);
map.put(TAG_RANKING_VALOR, valor);

// adding HashList to ArrayList
rankingList.add(map);
}

mAdapter = new RankingListAdapter(SocialActivity.this,
rankingList);
listRanking.setAdapter(mAdapter);

// dismiss the dialog after getting all albums
pDialog.dismiss();
} else {
Log.d("Rankings: ", "null");
}

} catch (JSONException e) {
e.printStackTrace();
} catch (JsonParseException e) {
e.printStackTrace();
} catch (Exception e) {
e.printStackTrace();
}
}
}
}
```

També es important controlar els events que succeeixen a cada item de la llista (normalment el click).

Aquí podem veure el codi font que gestiona aquest event:

```
CustomButton btnAddFriend = (CustomButton) findViewById(R.id.cbtFriend );
btnAddFriend.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
EditText txtFriend =(EditText) findViewById(R.id.edtFriend ); ;
LoadFriends tarea = new LoadFriends();
tarea.execute(Session.login, txtFriend.getText().toString());
txtFriend.setText("");
}
});
```

9.4 InputBox modals

Per agilitzar les tasques a l'usuari, hem evitat el màxim possible la utilització de finestres addicionals. Ara bé, en el cas de la compra i de la venda de mercaderies, utilitzem els diàlegs modals per a la introducció de dades

Aquí podem veure l'exemple del botó de compra. Cal remarcar que en aquest codi es força que l'object textEdit tingui una entrada únicament numèrica, per evitar errors innecessaris.



```

btnBuy = (CustomButton) findViewById(R.id.cbtBuy);
btnBuy.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        final EditText input = new EditText(mContext);
        input.setInputType(InputType.TYPE_CLASS_NUMBER);
        new AlertDialog.Builder(mContext)
            .setTitle("Buying quantity")
            .setMessage("Enter quantity")
            .setView(input)
            .setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                    Editable value = input.getText();
                    new
                    LoadCargos().execute("BUY",materialIdBuy,value.toString(),p
                    lanetTargetId);
                }).setNegativeButton("Cancel", new
                DialogInterface.OnClickListener() {
            });
    });
}

```

9.5 Missatges Toast

Per mostrar informació a l'usuari de manera temporal, he optat per utilitzar els missatges TOAST, que no són invasius a la pantalla i proporcionen informació de manera senzilla a l'usuari sobre les accions realitzades:

```

Toast.makeText(getApplicationContext(),
    (String)getResources().getString(R.string.planet_error) ,
    Toast.LENGTH_LONG).show();

```

9.6 RESTful asíncron

Aquesta ha sigut una de les parts més complexes de gestionar, ja que, al tractar els diferents comportaments dintre d'una mateixa classe podria retornar diferent tipus d'informació: un únic objecte JSON, un array d'objectes JSON, un codi de sortida exitosa o diversos codis de sortida no exitosos, que s'haurien de tractar de manera diferent.

Ha sigut important comprendre les accions que es podien dur a terme a cada un dels events, principalment `doInBackground()` i `onPostExecute()`.

A continuació mostro el codi d'una de la classe gestora de crides asíncrones

```

class LoadModules extends AsyncTask<String, String, String> {

    String login;
    String id_nave ;
    String id_modulo;
    String nombre;
    String descripcion;
    String tripulacion ;
    String tech ;
    String carga ;
    String motor ;
    String armadura;
    String escudos ;
    String recarga ;
    String ataque ;
    String coste ;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected String doInBackground(String... args) {
        HttpClient httpClient = new DefaultHttpClient();
        String respStr="";
    }
}

```



```

String json="";

try {
    if (args.length>0) {
        try
        {
            HttpGet del = new HttpGet(Session.URL + "modules/module/" + args[0]);

            del.setHeader("content-type", "application/json");

            HttpResponse resp = httpClient.execute(del);
            respStr = EntityUtils.toString(resp.getEntity());

            JSONObject c = new JSONObject(respStr);

            //nombre= respJSON.getString(TAG_NOMBRE);

            URL url= new URL(Session.URL+ "img/modules/" + args[0] + ".jpg");
            bmpModule = BitmapFactory.decodeStream(url.openConnection().getInputStream());
            json="OK";

            login= c.getString(TAG_MODULE_LOGIN);
            id_nave = String.valueOf((int)c.getDouble(TAG_MODULE_ID_NAVE));
            id_modulo = String.valueOf((int)c.getDouble(TAG_MODULE_ID_MODULO));
            nombre= c.getString(TAG_MODULE_NOMBRE);
            descripcion= c.getString(TAG_MODULE_DESCRIPCION);
            tripulacion = String.valueOf((int)c.getDouble(TAG_MODULE_TRIPULACION));
            tech =String.valueOf((int)c.getDouble(TAG_MODULE_TECH));
            carga = String.valueOf((int)c.getDouble(TAG_MODULE_CARGA));
            motor = String.valueOf((int)c.getDouble(TAG_MODULE_MOTOR));
            armadura= String.valueOf((int)c.getDouble(TAG_MODULE_ARMADURA));
            escudos = String.valueOf((int)c.getDouble(TAG_MODULE_ESCUDOS));
            recarga =String.valueOf((int)c.getDouble(TAG_MODULE_RECARGA));
            ataque =String.valueOf((int)c.getDouble(TAG_MODULE_ATAQUE));
            coste =String.valueOf((int)c.getDouble(TAG_MODULE_COSTE));

        }
        catch(Exception ex)
        {
            Log.e("ServicioRest","Error!", ex);

            json = "KO";
        }
    }
    else {
        List<NameValuePair> params = new ArrayList<NameValuePair>();

        // getting JSON string from URL
        json = jsonParser.makeHttpRequest(Session.URL + "modules/modules/" + Session.login,
                                        "GET",
                                        params);

        // Check your log cat for JSON response
        Log.d("Modules JSON: ", "> " + json);
    }
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

return json;
}

@Override
protected void onPostExecute(String json) {
    try {
        if (json!="OK" && json!="KO")
        {
            modules = new JSONArray(json);
            modulesList.clear();
            if (modules != null) {
                // looping through All albums
                for (int i = 0; i < modules.length(); i++) {
                    JSONObject c = modules.getJSONObject(i);

                    // Storing each json item values in variable
                    login= c.getString(TAG_MODULE_LOGIN);
                    id_nave =
                    id_modulo =
                    nombre= c.getString(TAG_MODULE_NOMBRE);
                    descripcion= c.getString(TAG_MODULE_DESCRIPCION);
                    tripulacion =
                    tech
                    carga =
                    motor =
                    armadura=
                    escudos =
                    String.valueOf((int) c.getDouble(TAG_MODULE_ID_NAVE));
                    String.valueOf((int) c.getDouble(TAG_MODULE_ID_MODULO));

                    String.valueOf((int) c.getDouble(TAG_MODULE_TRIPULACION));
                    =String.valueOf((int) c.getDouble(TAG_MODULE_TECH));
                    String.valueOf((int) c.getDouble(TAG_MODULE_CARGA));
                    String.valueOf((int) c.getDouble(TAG_MODULE_MOTOR));
                    String.valueOf((int) c.getDouble(TAG_MODULE_ARMADURA));
                    String.valueOf((int) c.getDouble(TAG_MODULE_ESCUDOS));
                }
            }
        }
    }
}

```



```

=String.valueOf((int)c.getDouble(TAG_MODULE_RECARGA));
=String.valueOf((int)c.getDouble(TAG_MODULE_ATAQUE));
=String.valueOf((int)c.getDouble(TAG_MODULE_COSTE));

recarga
ataque
coste

// creating new HashMap
HashMap<String, String> map = new HashMap<String, String>();
// adding each child node to HashMap key => value
map.put(TAG_MODULE_LOGIN, login);
map.put(TAG_MODULE_ID_NAVE, id_nave);
map.put(TAG_MODULE_ID_MODULO, id_modulo);
map.put(TAG_MODULE_NOMBRE, nombre);
map.put(TAG_MODULE_DESCRIPCION, descripcion);
map.put(TAG_MODULE_TRIPULACION, tripulacion);
map.put(TAG_MODULE_TECH, tech);
map.put(TAG_MODULE_CARGA, carga);
map.put(TAG_MODULE_MOTOR, motor);
map.put(TAG_MODULE_ARMADURA, armadura);
map.put(TAG_MODULE_ESCUDOS, escudos);
map.put(TAG_MODULE_RECARGA, recarga);
map.put(TAG_MODULE_ATAQUE, ataque);
map.put(TAG_MODULE_COSTE, coste);

// adding HashList to ArrayList
modulesList.add(map);
}

mAdapter = new ModuleListAdapter(ShipActivity.this,
modulesList);
listModules.setAdapter(mAdapter);
mAdapter.notifyDataSetChanged();

// dismiss the dialog after getting all albums
//pDialog.dismiss();
} else {
Log.d("Modules: ", "null");
}
} else
{
if (json.equals("OK")) {
imgModule.setImageBitmap bmpModule);

txtTripulacion.setText(tripulacion);
txtDescripcion.setText(descripcion);
txtTech.setText(tech);
txtCarga.setText(carga);
txtMotor.setText(motor);
txtArmadura.setText(armadura);
txtEscudo.setText(escudos);
txtRecarga.setText(recarga);
txtCoste.setText(coste);

}
else
{
Toast.makeText(getApplicationContext(),
(String)getResources().getString(R.string.social_not_existing_friend) ,
Toast.LENGTH_LONG).show();
}
}
} catch (JSONException e) {
e.printStackTrace();
} catch (JsonParseException e) {
e.printStackTrace();
}
}

}

class LoadRankings extends AsyncTask<String, String, String> {
@Override
protected void onPreExecute() {
super.onPreExecute();
}

@Override
protected String doInBackground(String... args) {
List<NameValuePair> params = new ArrayList<NameValuePair>();

// getting JSON string from URL
String json = jsonParser.makeHttpRequest(Session.URL + "ranking/ranking/" +
Session.login, "GET",
params);

// Check your log cat for JSON response
Log.d("Mails JSON: ", "> " + json);

return json;
}

@Override
protected void onPostExecute(String json) {
try {
JSONObject c = new JSONObject(json);;

```




```
tripulacion.setText(  
String.valueOf((int)c.getDouble(SocialActivity.TAG_RANKING_TRIPULACION));  
  
armadura.setText(String.valueOf((int)c.getDouble(SocialActivity.TAG_RANKING_ARMADURA));  
escudos.setText(String.valueOf((int)c.getDouble(SocialActivity.TAG_RANKING_ESCUDO));  
saldo.setText(String.valueOf((int)c.getDouble(SocialActivity.TAG_RANKING_SALDO));  
valor.setText(String.valueOf((int)c.getDouble(SocialActivity.TAG_RANKING_VALOR));  
  
    } catch (JSONException e) {  
        e.printStackTrace();  
    } catch (JSONException e) {  
        e.printStackTrace();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    }  
}
```

10 Manual d'usuari

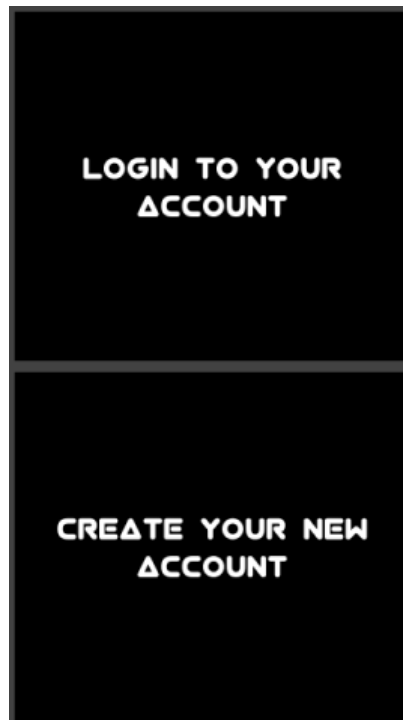
10.1 Pantalla d'inici

Quan l'aplicació s'executa, apareixerà una finestra durant 3 segons amb el nom de l'aplicació:



Un cop desaparegui aquesta imatge, podrem seleccionar si ens voler registrar o si pel contrari, ja disposem d'un usuari



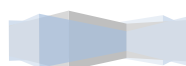


10.1.1 Creació de nou usuari

Per crear un nou usuari, introduïrem l'e-mail, el nom d'usuari i la contrasenya i la verificació.

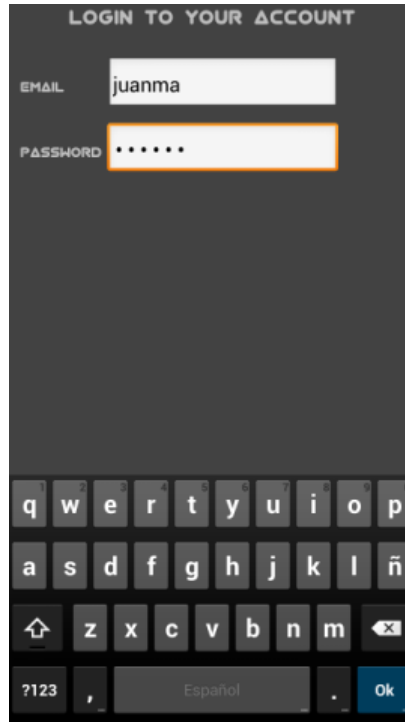


Si hem comés un error en la verificació de contrasenya o l'usuari ja existeix, l'aplicació ens donarà un missatge com a la imatge de la dreta



10.1.2 Inici de sessió amb usuari existent

Si ja disposem d'un usuari, escollirem l'opció corresponent i introduïrem el nom d'usuari i el seu password. Posteriorment pitjarem el botó *Play*



10.1.3 Menú principal

Un cop validats a l'aplicació, ens situarem al menú principal de l'aplicació



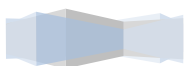


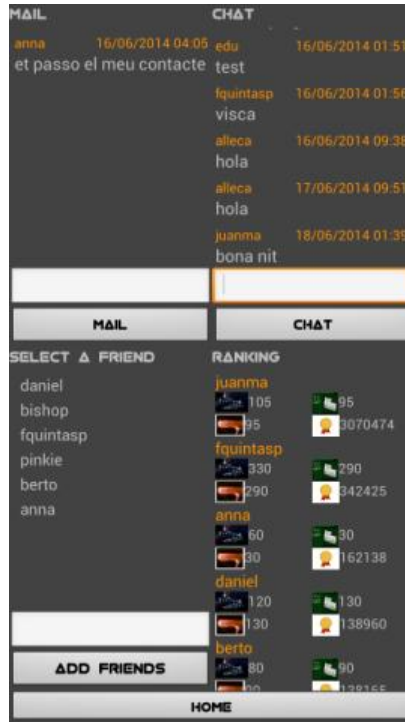
En aquesta pantalla, podem veure les estadístiques bàsiques (escut, armadura i planeta actual), així com el saldo en monedes i en gemes (actualment deshabilitat)

10.1.4 Àrea social

Quan accedim a l'àrea social, trobarem una pantalla dividida en quatre:

- A la banda inferior dreta, trobem el ranking d'usuaris (podem desplaçar-nos per la llista tot fent un scroll arrossegant)
- A la banda inferior esquerra, podem trobar els nostres contactes. Per afegir un contacte, escriurem el nom que apareix al ranking i pitjarem sobre "add friend"
- A la banda superior esquerra, podem trobar el mail. Per enviar un mail, escriurem el missatge al textbox situat sobre el botó mail, seleccionarem un contacte i el botó de mail canviarà cap a "mail to xxxx"
- A la banda superior dreta, podem escriure missatges instantanis que seran vistos pels altres jugadors connectats

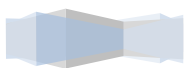




10.1.5 Navegació

Per poder canviar de planeta, accedirem a la pantalla Navegació. Aquí podrem veure una llista de tots els planetes ordenats del més proper al més llunyà.

Per viatjar a un d'aquests planetes, el seleccionarem i el botó inferior canviarà a "navigate to xxxx".



En pitjar el botó, sortirem de la pantalla, i tornarem a la pantalla principal. Allà trobarem que tenim com a destí el nou planeta i a quina distància en anys llum (LY) ens trobem. Aquesta distància s'anirà reduint fins que arribem a la destinació escollida.



10.1.6 Àrea tripulació

Dintre del àrea de tripulació, trobarem la biografia del capità assignat, així com una foto d'aquest / aquesta.

A la banda inferior dreta, veiem a la informació bàsica de la resta de la tripulació. Selecciónant a un dels membres, podrem veure la seva foto.



CAPTAIN **BIOGRAPHY**



El Capitán Haddock es un personaje de historietas de Las aventuras de Tintin. Como mejor amigo de Tintin, Archibaldo Haddock se dio a conocer en El cangrejo de las pinzas de oro y poco a poco fue ganando protagonismo en las aventuras del

CREW PHOTO **CREW MEMBERS**



capita haddock

nav.	clima	rel.	eng.	Func.	cost
5	1	3	3	captain	500

piloto 2

nav.	clima	rel.	eng.	Func.	cost
2	15	0	6	pilot	400

marine 2

nav.	clima	rel.	eng.	Func.	cost
1	3	3	8	marine	300

marine

nav.	clima	rel.	eng.	Func.	cost
1	3	5	8	marine	300

HOME

10.1.7 Àrea Flota


En aquesta àrea, trobarem la informació sobre els mòduls de que disposa la nostra nau. Ens informa de les estadístiques general a la banda dreta superior, i, seleccionant una entrada del llistat superior esquerre, podrem veure una foto i les estadístiques d'aquets mòdul.

MODULE LIST **STATS SUMMARY**

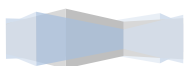
Energy 1	Crew	105
Energy 3	Armor	95
Load 8	Shield	95
	Balance	3050254
	Value	3071254

ENERGY 3 MODULE INFO **ENERGY 3 PICTURE**

Desc.	Energy 3
Crew	2
Tech.	70
Load	450
Motor	20
Armor	20
Shield	10
Reload	10
Cost	350



HOME



10.1.8 Àrea comerç

Aquesta es la secció on podrem comerciar amb altres planetes.

A la banda esquerra superior trobem els materials disponibles per a la venda. Els que estan marcats en vermell, no es poden vendre, ja que són materials que no estan assignats per defecte al planeta al que ens trobem.

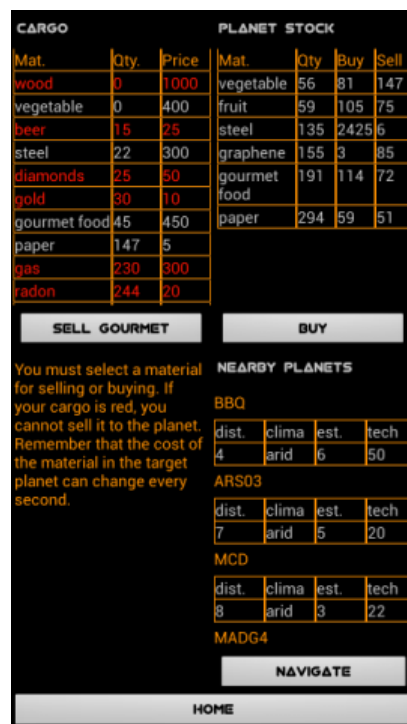
A la banda dreta superior, trobem els bens que tenen els planetes, amb un preu de venda i un preu de compra, que anirà fluctuant.

Tant pel procés de compra com pel procés de venda, seguirem les mateixes passes:

Seleccionarem el material a comprar o vendre i el botó canviarà de nom. Un cop pitjat el botó de compra / venda, ens demanarà quantes unitats volem vendre / o comprar.

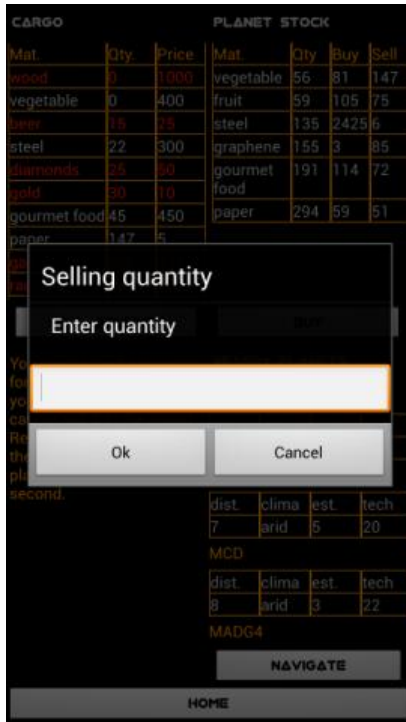
Si estem en disposició d'efectuar la transacció (tenim saldo suficient, stock suficient, la quantitat és positiva...), ens apareixerà un missatge informant-nos de la transacció.

Si no ha sigut possible, ens indicarà el motiu.

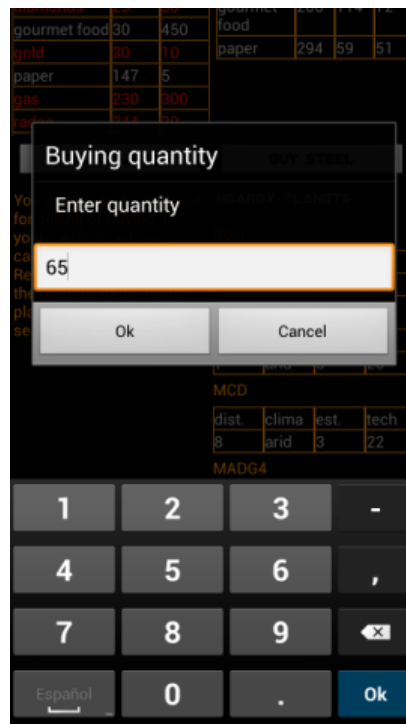


Aquí podem apreciar la selecció del producte a vendre





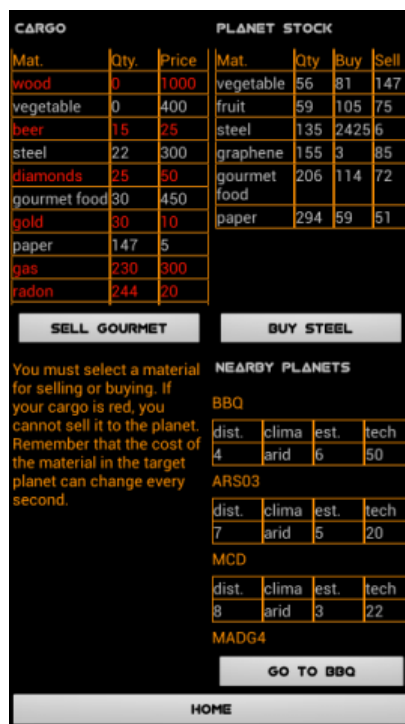
Aquí ens demana quantes unitats volem vendre i tenim el missatge de confirmació



Funciona de la mateixa manera per la venda. Notar que el teclat està limitat a dígitos per evitar errors innecessaris.



També trobem una breu explicació de qué hem de fer en aquesta secció i un accés més reduït a la funcionalitat de navegació, per, un cop efectuades les gestions comercials, poder navegar a un nou destí.



11 Conclusions

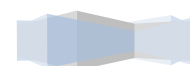
La realització d'aquest projecte ha estat molt dura i amb moments de frustració. Però un cop assolit el resultat, m'ha generat una gran satisfacció i una sensació que per aquests tipus de coses val la pena dedicar-se al món de la informàtica.

A mesura que els problemes amb la plataforma Android es solucionaven he anat passant d'una certa aversió per les eines que utilitzava (Java, Eclipse,...) a una raonable apreciació de les mateixes (val a dir que el meu background és, sobretot, en eines Microsoft...)

Vull millorar les mancances que té aquest producte (que són moltes) i utilitzar els coneixements obtinguts per poder crear un producte amb una qualitat acceptable per ser utilitzat per més gent.

12 Futures millores fora de l'àmbit del projecte

La pressió del projecte, la inexperiència en aquest entorn i la combinació del desenvolupament amb les obligacions personals i laborals, han fet que el producte fos molt més incomplet del que era a la meua imaginació quan vaig començar-lo.



Les properes funcionalitat que s'afegiran al projecte, fora de l'àmbit lectiu seran:

- Àrea social
 - Identificar usuaris connectats o desconnectats
 - Búsqueda d'amics
 - Millorar aspecte gràfic del rànking
- Àrea Comerç
 - Indicar quin és el balanç econòmic en cada moment
- Tripulació i Nau
 - Permetre comerciar amb els mòduls i tripulació
- Execució de l'univers
 - Afegir més intel·ligència als càlculs aplicant regles més complexes (basades en la religió, govern i clima del planeta, així com a fets aleatoris)
- Imatge
 - Reemplaçar les imatges per unes dissenyades específicament per a l'aplicació, amb un aire més minimalista
- general
 - Donar informació a l'inici sobre els esdeveniments que han ocorregut mentre l'usuari no estava connectat.
 - Afegir l'apartat *gems* per monetitzar l'aplicació
 - Migrar BBDD a mySql

13 Fonts d'informació

13.1 Internet

Per al desenvolupament del projecte s'han consultat les següents pàgines

- Android Developer (<http://developer.android.com/index.html>)
- Sgoliver.net (<http://www.sgoliver.net/>)
- Java Code Geeks (<http://www.javacodegeeks.com>)
- The code project (<http://www.codeproject.com/>)
- Android Example (<http://androidexample.com/>)
- UPV Android: Introduccion a la programación (<http://cursointroduccionandroid.upvx.es/ficha>)

13.2 Bibliografia

- El gran libro de programación avanzada con Android – José Enrique Amaro Soriano
- Desarrollo de videojuegos I-II-III-IV (UCLM) – David Vallejo Fernandez / Cleto Martín Angelina
- Artificial intelligence for games – Ian Millington



- Beginning Android Games – Mario Zechner / Robert Green
- Game Programming game AI by exemple – Mat Buckland
- Game engine architecture – Jason Gregory

