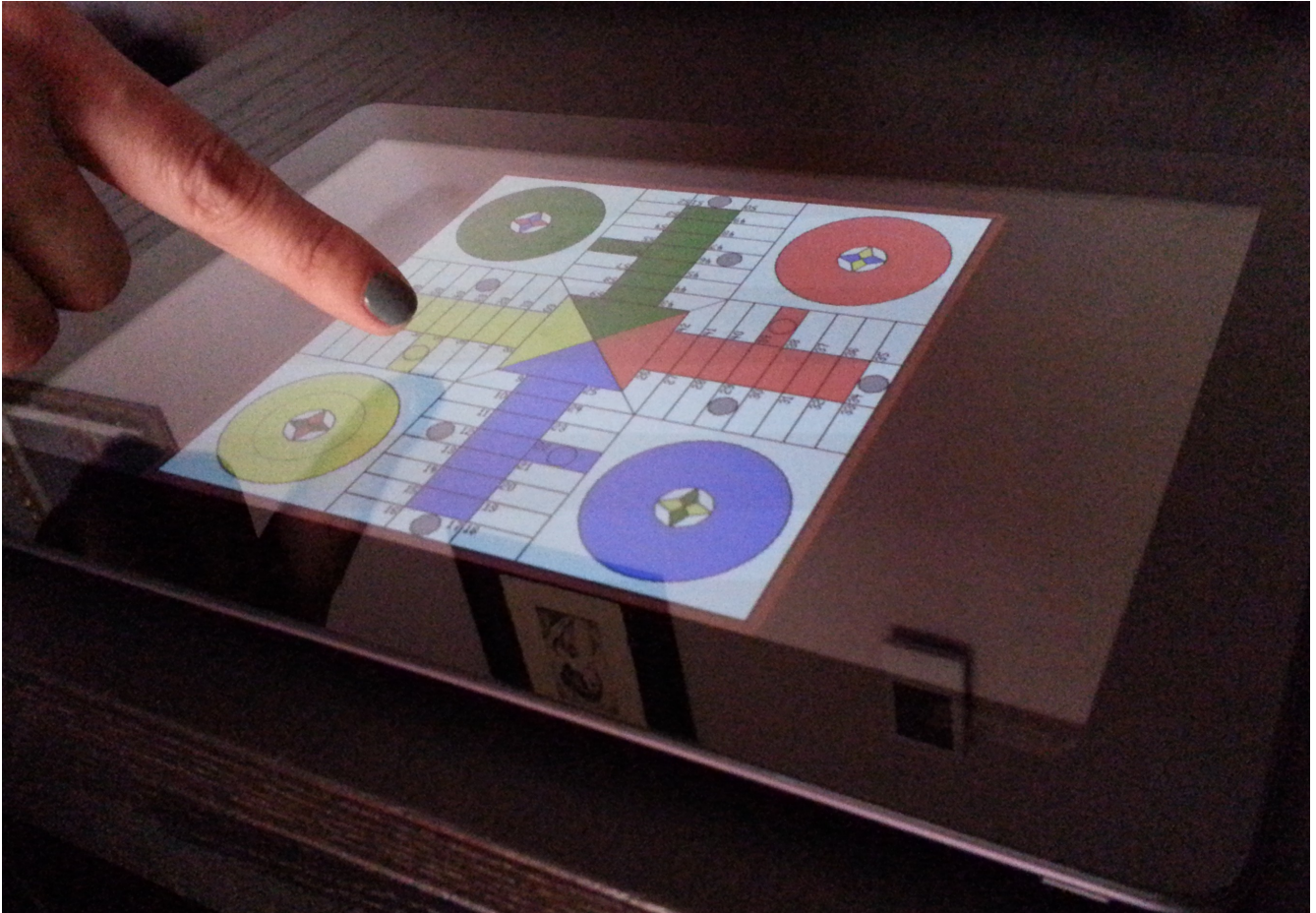


# DJT - Digitalitzador de jocs de taula



**Alumne: Jordi Montoya Ortega**

**Tutor: Carlos Sanchez Rosa**

**Àrea: Programació d'aplicacions per a mòbils usant HTML5**

# Índex

Descripció.....	4
Objectius.....	4
Objectius del projecte.....	4
Línies obertes del projecte.....	5
Diari de desenvolupament (Introducció).....	5
Que necessitem?.....	6
Requeriments hardware.....	6
Requeriments software.....	6
Eines per al desenvolupament.....	7
Diari de desenvolupament (Eines de treball).....	7
Planificació.....	8
Gantt.....	9
Tasques.....	10
Recursos.....	10
Fites del projecte.....	11
Fita 1.....	11
Fita 2.....	11
Fita 3.....	12
Fita extraordinària.....	12
Fita final, Entrega del projecte.....	12
Cicle de vida del projecte.....	12
Diari de desenvolupament (Planificació).....	13
Anàlisi funcional.....	14
Casos d'ús.....	14
Descripció dels casos d'ús.....	15
Detall del Context 1.....	18
Detall del Context 2.....	22
Detall del Context 3.....	24
Disseny de pantalles.....	27
Selecció de un joc.....	27
Jugar a un joc.....	28
Manteniment d'un joc.....	30
Diari de desenvolupament (Anàlisi funcional).....	33
Anàlisi tècnic.....	34
Arquitectura.....	34
Base de dades.....	34
Diagrama entitat-relació.....	35
Disseny de la aplicació.....	36
Diagrama de classes - Entitats.....	37
Diagrama de classes - Model.....	37
Diari de desenvolupament (Anàlisi tècnic).....	39
Implementació.....	39
Base de dades.....	39
Model relacional.....	39
Model de objectes.....	40
Mètodes d'accés a dades.....	41
Selecció de joc.....	43

Manteniment de joc.....	44
Càmera/Galeria.....	46
Taula de fitxes i taula de daus.....	47
Carregar.....	49
Guardar.....	50
Jugar a un joc.....	50
Carregar joc.....	51
Carregar joc. El taulell.....	52
Carregar joc. Les fitxes.....	52
Carregar joc. Els daus.....	54
Principals problemes durant la implementació.....	56
Transaccions asíncrones.....	56
Jquerymobile, recarregues de pantalla.....	57
Jquerymobile, visualització de inputs en formularis.....	57
Rendiment en la pantalla de selecció de joc.....	58
Rendiment en la pantalla de jugar amb un joc.....	58
Depuració.....	58
Problemes coneguts.....	58
Diari de desenvolupament (Implementació).....	59
Conclusions.....	60
Glossari de termes.....	61
Bibliografia.....	62

## Descripció

Avui dia existeixen milers de jocs de taula formats per tres parts: un taulell, unes fitxes i uns daus. L'us d'aquests jocs de taula no és tan freqüent com fa uns anys degut a la gran importància d'altres mètodes d'entreteniment com per exemple els videojocs.

El que es vol fer en aquest projecte és acostar els clàssics jocs de taula als temps més moderns que corren avui dia, utilitzant la tecnologia més actual però mantenint la manera de jugar de tota la vida.

Existeixen moltes versions de jocs de taula en format digital. El problema d'aquests, és que son massa automatitzats, és a dir, un tira un dau digital i la resta es fa automàticament. Això fa que es perdi la “màgia” dels jocs de taula, i és possible que sigui la causa que aquests no siguin molt populars.

El digitalitzador de jocs de taula, es tracta d'una eina que ofereix la possibilitat de passar un gran nombre de jocs de taula a format digital, per el posterior ús d'aquests tal com s'ha fet tota la vida.

El que es pretén amb això és mantenir l'essència dels jocs de taula, on cadascú que juga es pot inventar les seves pròpies normes, o pot intentar fer trampes, o en definitiva utilitzar-l'ho com ho ha fet sempre. A part, si es te a casa un joc al que s'ha jugat sempre, es podrà tenir la mateixa versió d'aquest en format digital i no un de semblant, cosa que pot donar un sentiment especial al utilitzar-l'ho.

Així doncs el projecte tractarà en desenvolupar una aplicació per dispositius mòbils, preferentment de pantalla gran, on poder mantenir jocs fets per l'usuari.

## Objectius

### ***Objectius del projecte***

Els objectius del projecte son el desenvolupament de tots els elements necessaris per que l'aplicació sigui funcional.

Aquests objectius seran:

- Visualització de jocs: S'han de poder visualitzar tots els jocs que hi ha en l'aplicació, i poder interactuar amb aquests.
- Manteniment de jocs: L'aplicació ha de poder crear, modificar i eliminar un joc. Aquest manteniment ha de ser persistent, és a dir, si sortim de l'aplicació cal que es quedin guardats tots els jocs creats.
  - Captura de taulell: S'ha de poder realitzar la captura d'un taulell tant amb la càmera del dispositiu com des d'un fitxer de imatge que tinguem.
  - Selecció de elements de joc: S'han de poder seleccionar els elements necessaris del joc.

Aquests elements seran fitxes i daus. Es podran escollir tantes fitxes com es vulgui i de variats colors i tants daus com es vulgui.

- Utilització de jocs: L'aplicació ha de ser capaç d'obrir un joc guardat, carregar totes les fitxes i daus.
- Interacció amb el dau: Els daus han de permetre ser “llançats” generant un valor aleatori.
- Interacció amb les fitxes: Les fitxes s'han de poder moure per damunt del taulell arrossegant-les amb el dit.
- El taulell ha de aparèixer el màxim gran possible mantenint l'aspecte.

### ***Línies obertes del projecte***

Es tracta d'objectius extraordinaris queden fora del desenvolupament del projecte. Aquest contingut ha estat pensat com a possible millora al funcionament bàsic de l'aplicació.

Aquests objectius son:

- Zoom en la pantalla de jugar.
- Guardar un posicionament inicial sobre el taulell de les fitxes, de manera que en obrir el joc, apareguin ja en el lloc que vulguem.
- Guardar partida: Guardar l'estat d'una partida per poder-la reprendre en qualsevol moment.
- Compartició de jocs: Permetre l'enviament dels jocs entre dispositius per tal que es puguin utilitzar per més gent sense que s'hagin de crear varies vegades.
- Taulell amb gravetat: Simular el comportament d'un taulell real fent que si el movem les fitxes es caiguin o es moguin tal com farien en el món real.
- Editor d'imatges: Al capturar un nou taulell, poder editar l'imatge retallant-la o escalant-la per tal que sigui una imatge amb més qualitat.
- Fitxes i daus en 3D per una experiència més real.

### **Diari de desenvolupament (Introducció)**

La idea d'aquest apartat i els següents amb el mateix nom, és tenir en narrativa un resum del que ha anat succeint durant el desenvolupament del projecte.

La elecció del projecte no va ser fàcil. Primer de tot es va haver d'escollir l'àrea del projecte. El que es volia fer era una aplicació en Android degut a que és el sistema operatiu mòbil més accessible per a tothom degut a que és gratuït i es pot desenvolupar des de qualsevol plataforma. Com que no existia cap àrea a escollir per desenvolupar aplicacions natives per SO mòbils, la que més va interessar era la de programació d'aplicacions mòbils usant HTML5. Això va suposar més dificultat en l'aprenentatge degut a que tots els llenguatges que s'haurien d'utilitzar eren del tot, o gairebé del tot desconeguts. En canvi, el desenvolupament natiu en Android, al ser Java, i degut a que ja s'havia “tocat” una mica anteriorment, hauria estat molt més senzill.

Estàvem davant d'una nova decisió. Ara ja es tenia l'àrea de treball, però encara s'havia de decidir quin projecte fer. Qualsevol cosa que se'ns acudia ja estava inventada, era massa senzilla o massa complexa per ser un projecte final de carrera tenint en compte les hores que es disposaven.

De les propostes que hi havia per escollir no hi havia cap que fes el pes, de manera que es va decidir que se'n faria una pròpia.

La idea de l'aplicació no era nova. Ja s'havia pensat feia temps en veure que molts videojocs que simulen jocs de taula, es converteixen en jocs absurds on només cal tirar el dau i mirar com va responant el joc davant del resultat del dau.

Es va proposar al tutor, i li va semblar que era una bona idea, així que de seguida ens vam posar a definir quins serien els objectius a complir la tecnologia a utilitzar i la planificació a seguir en el pla de treball.

## **Que necessitem?**

Primer de tot s'ha fet un estudi de les eines que existeixen per tal de detectar quines són les que cobriran les necessitats que tenim en el desenvolupament de l'aplicació.

Ens disposem a fer una aplicació en HTML5. La principal avantatge que això té és que l'aplicació serà multi-plataforma. Avui dia hi ha diversos sistemes operatius per a dispositius mòbils. El que es pretén és que la nostra aplicació sigui apta per tots ells o si més no, sigui el més fàcil de portar possible.

Per tal d'aconseguir això existeixen diferents frameworks. Un d'ells és el Phonegap. Phonegap és un framework que ens permet desenvolupar aplicacions en HTML5, CSS3 i javascript fent que aquestes s'executin en dispositius mòbils. És totalment compatible amb els SO mòbils més usats, que són iOS i Android i Windows Phone. Pot accedir a tots els elements hardware com ara acceleròmetre, gps, càmera, notifikacions...

Aquest framework està agafant molta popularitat ja que la tecnologia que es fa servir per desenvolupar és molt coneguda i estesa i les possibilitats que ens dona a la hora de portar-la entre dispositius són molt elevades.

## **Requeriments hardware**

Per que la nostra aplicació pugui funcionar, és necessari un telèfon mòbil o tableta que disposi de càmera, pantalla tàtil i memòria per guardar els jocs.

## **Requeriments software**

Phonegap: S'han utilitzat les llibreries de Phonegap les quals s'encarreguen de comunicar el codi que s'ha desenvolupat amb el dispositiu on es fa la instal·lació de l'aplicació.

Sqlite: Es requereix que el dispositiu disposi d'un sistema de base de dades Sqlite. Phonegap es comunica amb aquest i fa d'intermediari del codi de l'aplicació amb la base de dades.

Kineticjs: Es tracta d'una llibreria gràfica 2D per l'element canvas d'HTML5. S'ha utilitzat per fer el

desenvolupament de la pantalla de jugar amb un joc. En aquesta es creen objectes, es permet arrossegar-los, es poden crear daus i llançar-los etc.

Jquerymobile: Son unes llibreries JavaScript que donen molta potencia a la hora de desenvolupar webs per mòbils. S'han utilitzat aquestes llibreries per fer les vistes de les pantalles de l'aplicació.

## ***Eines per al desenvolupament***

Per al desenvolupament de l'aplicació s'utilitzarà:

- Ubuntu 12.10 (<http://www.ubuntu.com/>): El sistema operatiu que s'ha utilitzat per fer el desenvolupament de tot el projecte.
- Eclipse Indigo EE (<http://www.eclipse.org/>): IDE de desenvolupament de aplicacions.
- Android SDK (<http://developer.android.com/sdk/index.html>): Kit de desenvolupament de aplicacions Android.
- **Nexus 10**: Tauleta Android que s'ha utilitzat per desplegar i testejar l'aplicació.
- LibreOffice 3.5.4.2 (<http://es.libreoffice.org/>): Suite ofimàtica que s'ha utilitzat per fer la documentació.
- Planner (<https://live.gnome.org/Planner>): Eina per realitzar planificacions de les tasques.
- InkScape (<http://inkscape.org/>): Eina que s'ha utilitzat per dissenyar les pantalles.
- DIA (<http://projects.gnome.org/dia/>): Eina d'edició de diagrames.
- Phonegap (<http://phonegap.com/>): Framework de desenvolupament d'aplicacions per mòbils.
- Jquerymobile (<http://jquerymobile.com/>): Framework de desenvolupament d'aplicacions per mòbils.
- Kineticjs (<http://kineticjs.com/>): Llibreria gràfica.

## **Diari de desenvolupament (Eines de treball)**

Les eines que es volien fer servir, estaven clares des d'un principi. Durant tota la carrera a la UOC s'ha fet servir sempre que s'ha pogut eines opensource. Moltes d'elles no estan al nivell d'eines comercials, però la gran majoria de vegades, per no dir sempre, han permès fer les tasques per les que es necessitaven.

Per al projecte final de carrera, no seria diferent. Totes les eines que s'han utilitzat son opensource, gratuïtes i suficientment potents com per dur a terme el projecte.

De les eines que escollides no havíem utilitzat anteriorment les següents: Planner, Phonegap, Jquerymobile i Kineticjs.

L' Ubuntu s'havia utilitzat des de fa ja uns quants anys durant la major part de la carrera i en l'àmbit personal. L'eclipse fa anys que l'utilitzo en la meua vida professional, de manera que es coneix molt i es te molt per ma. L'Android SDK es coneix relativament poc. S'ha fet una aplicació senzilla nativa en Android anteriorment. De tota manera no ha fet falta conèixer pràcticament res d'aquest per fer el

projecte. El Libreoffice és el processador de textos que sempre s'ha utilitzat en la UOC i en la resta d'àmbits excepte en casos on s'hagi imposat la variant de Microsoft. El Planner, ha estat una eina nova que s'ha utilitzat per fer planificacions. És senzilla però ha servit per fer el que necessitàvem. L'Inkscape ja l'havia utilitzat en la assignatura d'Interfícies multimèdia i ha servit per dissenyar les pantalles. El DIA, s'ha utilitzat en moltes assignatures durant la carrera, per exemple les assignatures de base de dades on es necessitava fer models relacionals etc.

Jquerymobile ha estat el que mes mal de caps ha portat. Ha costat molt fer-se amb ell. En molts moments s'ha estat a punt de treure a aquest del projecte i utilitzar directament Javascript i estils CSS3. No s'ha fet ja que se sap que és una eina que ha de aportar molt en quant a desenvolupament. La sensació que s'ha tingut és que abans de treballar amb Jquerymobile hauria estat molt positiu treballar amb Javascript, tenir coneixements una mica mes grans sobre com programar sense Jquerymobile i després afegir aquesta tecnologia, que segurament aporta només que millores. Per fer això es evident que no es disposava de temps en el projecte així que s'ha hagut de “patir” una mica mes del compte.

Kineticjs, ha permès treballar a un nivell mes alt, definint objectes i sprites. Sense aquesta llibreria hauria estat molt mes complicat el desenvolupament de l'apartat de jugar (canvas).

## Planificació

Per la planificació s'ha contat que el temps laborable en una setmana és de 2 dies. En alguns casos excepcionals ha pogut ser de mes o menys dies degut a vacances o altres situacions que han pogut comportar que es dediqui mes o menys temps. A part s'han contat varis dies festius del calendari com a laborables per al projecte.

Es disposa aproximadament de 225 hores per realitzar totes les funcionalitats que inclou el projecte final de carrera. Ha calgut retallar molta funcionalitat “extra” i extreure-la del projecte degut a que no ha estat possible desenvolupar aquestes parts, que aporten valor afegit a l'aplicació, però no son indispensables per al seu ús.

Segons la planificació el projecte quedarà enllestit el dia 2 de gener, un dimecres. Aquesta data es escollida com a final degut a que al desembre es disposaran de uns quants dies extra on podrem aprofitar per concloure totes les tasques que quedin obertes.

S'han definit 5 tasques:

**Formació:** La tasca de formació son un total de 32 hores on s'estudiaran les tecnologies que utilitzarem en el desenvolupament del projecte. Aquestes tecnologies son: Android, HTML5, canvas, Kineticjs, Jquerymobile, Planner i InkScape.

**Planificació:** A la tasca de planificació se li dediquen 8 hores. És la tasca on es definiran ella mateixa i la resta de tasques, el temps i recursos de cadascuna i l'ordre en que cal executar-les.

**Disseny:** La tasca de disseny serà la part on es definiran les especificacions de l'aplicació. Disposem de 40 hores per realitzar-la. Esta composta per l'Anàlisi funcional, el disseny de les pantalles i l'anàlisi tècnic.

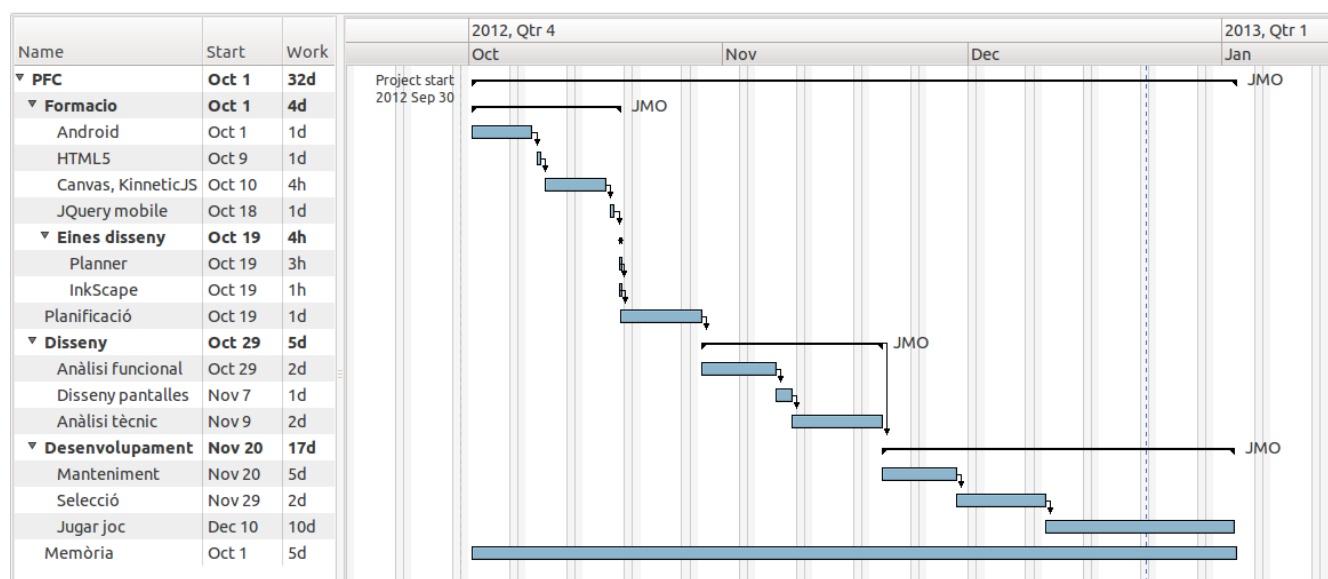
**Desenvolupament:** Aquesta tasca inclou tota la implementació del que hem dissenyat a la tasca



anterior. Disposem de 136 hores de desenvolupament.

**Memòria:** Aquesta tasca s'inicia juntament amb l'inici del projecte i finalitza al entregar aquest. Inclou tota la feina feta per poder redactar el present document. Aproximadament unes 40 hores per realitzar-la.

## Gantt



**Inici del projecte: 1 de Octubre del 2012.**

**Final del projecte: 2 de Gener del 2013.**

## Tasques

Name	Start	Finish	Work	Duration
▼ <b>PFC</b>	<b>Oct 1</b>	<b>Jan 2</b>	<b>32d</b>	<b>68d</b>
▼ <b>Formació</b>	<b>Oct 1</b>	<b>Oct 19</b>	<b>4d</b>	<b>14d 4h</b>
Android	Oct 1	Oct 8	1d	6d
HTML5	Oct 9	Oct 9	1d	1d
Canvas, KineticJS	Oct 10	Oct 17	4h	6d
JQuery mobile	Oct 18	Oct 18	1d	1d
▼ <b>Eines disseny</b>	<b>Oct 19</b>	<b>Oct 19</b>	<b>4h</b>	<b>4h</b>
Planner	Oct 19	Oct 19	3h	3h
InkScape	Oct 19	Oct 19	1h	1h
Planificació	Oct 19	Oct 29	1d	6d
▼ <b>Disseny</b>	<b>Oct 29</b>	<b>Nov 20</b>	<b>5d</b>	<b>16d</b>
Anàlisi funcional	Oct 29	Nov 7	2d	7d
Disseny pantalles	Nov 7	Nov 9	1d	2d
Anàlisi tècnic	Nov 9	Nov 20	2d	7d
▼ <b>Desenvolupament</b>	<b>Nov 20</b>	<b>Jan 2</b>	<b>17d</b>	<b>31d</b>
Manteniment	Nov 20	Nov 29	5d	7d
Selecció	Nov 29	Dec 10	2d	7d
Jugar joc	Dec 10	Jan 2	10d	17d
Memòria	Oct 1	Jan 2	5d	68d

## Recursos

Name	Short nam	Type	Group	Email	Cost
Jordi Montoya Ortega	JMO	Work			0
Carlos Sanchez Rosa	CSR	Work			0

## Fites del projecte

Per tal de fer un seguiment correcte, s'han definit 3 fites cadascuna en una data concreta que cal complir per tal de arribar al final del projecte amb èxit.

Aquestes fites son:

### ***Fita 1***

31 d'Octubre, anàlisi funcional i anàlisi tècnic.

Aquesta fita inclou una primera versió de l'anàlisi funcional, del disseny de les pantalles i de l'anàlisi tècnic. Aquesta versió inicial no és definitiva. Juntament amb el tutor s'iterarà fins que arribem a una feina que es cregui satisfactòria. Aquest motiu fa que es pugui allargar fins al 20 de novembre la tasca de disseny.

Aquesta documentació funcional-tècnica definirà:

- Casos d'us.
- Disseny de totes les pantalles de l'aplicació.
  - Pantalla de selecció de joc.
  - Pantalla de manteniment d'un joc.
  - Pantalla de jugar a un joc.
- Definició de elements de joc possibles.
  - Tipus de fitxes.
  - Tipus de daus.
- Definició de la BBDD.
- Definició de entitats.
- Definició de arquitectura de l'aplicació.

Un cop entregat el document es procedirà a iniciar la segona fita. Si hi ha alguna modificació en la primera, caldrà executar-la paral·lelament a la segona.

### ***Fita 2***

28 de Novembre, primera versió de l'aplicació.

El que es fa en aquesta fita és un “break” per redactar un petit document on s'analitza l'estat de l'aplicació. En aquest document s'hi recolliran tots els elements destacables que hagin ocorregut durant el desenvolupament.

A part del document caldrà entregar el codi font implementat on s'espera, però no te perquè ser exactament, que hi hagi:

- Manteniment total de un joc. (Alta/modificació/eliminació)
- Pantalla de selecció de joc.

### ***Fita 3***

2 de Gener, versió final de l'aplicació.

El dia 2 de gener l'aplicació ha de estar acabada. A part de l'aplicació cal que estigui feta la memòria i la presentació del projecte.

En la tercera fita tindrem en l'aplicació les parts de la fita 2 mes:

- Pantalla de joc.
- Carrega de joc.
- Fitxes funcionals
- Daus funcionals.
- Redacció de la memòria.
- Gravació de la presentació.

### ***Fita extraordinària***

Nomes en cas que la fita 3 finalitzi abans del previst, executarem la fita extraordinària. Aquesta fita inclou la implementació dels objectius extraordinaris que no entren en el projecte. És una fita que pot aportar valor afegit, però també és un risc. No ens podem permetre “trencar” res en el moment de la execució de la fita extraordinària, de manera que en cas de executar-la serà nomes si es te el total convenciment de que no estem desestabilitzant l'aplicació que hem d'entregar.

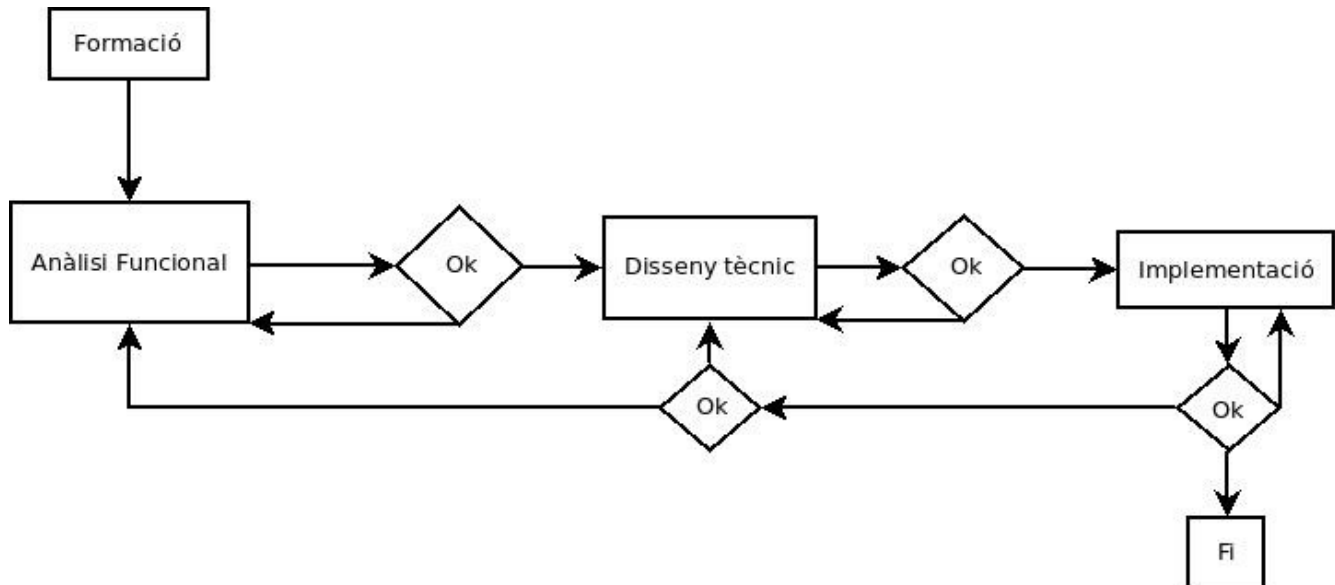
### ***Fita final, Entrega del projecte***

Aproximadament la entrega serà el dia 4 de gener del 2013.

## **Cicle de vida del projecte**

El projecte te un cicle de vida iteratiu en varies fases. No es passarà d'una fase a la següent fins que s'estigui segur que es va sobre el camí correcte. Un cop validat que això és així, s'avançarà. Això no vol dir que no es pugui tornar mai enrere, sempre es podrà tornar si és necessari, però si que es minimitzaran les probabilitats de que haguem de fer-ho.

Les etapes a seguir son les del següent gràfic.



Primer de tot es farà la formació, la qual té uns dies determinats i no es pot allargar. Durant la resta de etapes sempre hi ha formació inclosa, però serà sempre com una part d'aquesta i mai un tot.

Es continuarà amb l'anàlisi funcional. Un cop assegurat i validat amb el tutor, es podrà passar a la següent etapa. En cas que no sigui suficient, es continuarà treballant en ell abans de passar.

Seguidament es passarà al disseny tècnic on novament estarem iterant en ell fins que s'estigui d'acord en que està tot correcte.

Finalment, es passarà a la implementació. En aquest punt es podrà tornar a etapes anteriors en cas que sigui necessari, ja que és probable que es trobi alguna deficiència en el disseny en aquesta etapa. Un cop finalitzada la implementació finalitzarà la part del projecte corresponent aquesta i es procedirà a tancar els temes oberts com la memòria o la presentació.

## Diari de desenvolupament (Planificació)

Abans de començar la assignatura del projecte final de carrera es pensava que per fer aquest es tindria molt més temps. Be, el que es pensava de fet és que el temps que es tindria donaria per molt més, i es podrien realitzar més funcionalitats.

De tot el que es voldria haver posat en el projecte, només hi ha la part que s'ha cregut que era possible arribar a fer amb el temps que es disposava. Per tant s'han fet primer les parts principals abans que les menys importants.

La planificació ha estat modificada a posteriori. Això és degut a que, primerament, abans de poder planificar les tasques, es volia dedicar un temps a realitzar les de formació. D'aquesta manera es podria ajustar més bé els temps de cada tasca i definir aquestes més precisament, degut a que ja es tindria el coneixement necessari per fer estimacions.

La planificació inicial, era molt semblant amb la excepció que era molt més genèrica. No hi havia definides les subtasques. Només es tenien identificades les tres tasques principals, “Formació, Disseny i Desenvolupament”.

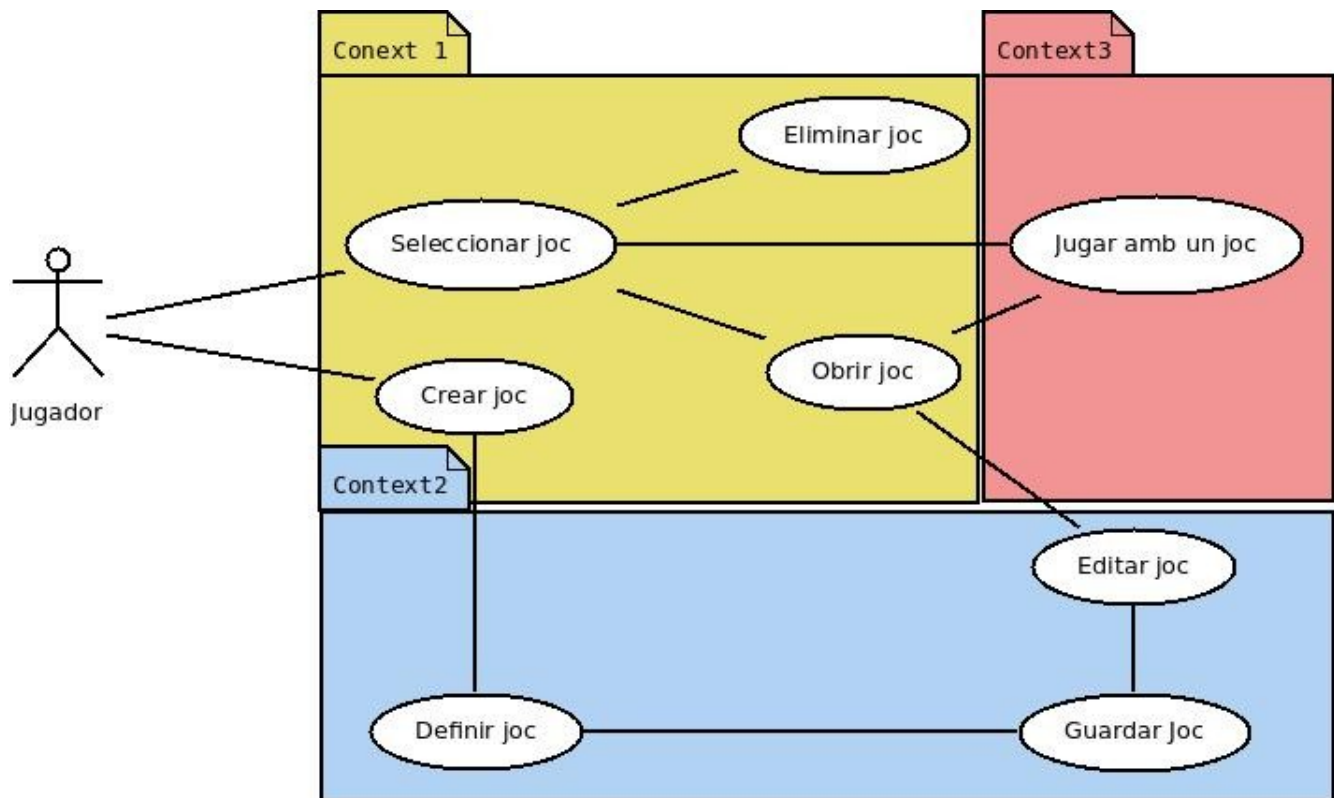
Quan es va modificar la planificació no van tocar les dates, ja que el temps que teníem no donava per fer gaires modificacions. El que es va fer és detallar les tasques que ja teníem amb les subtasques que es dividia cadascuna.

Pel que fa a les fites, les dates es corresponen amb les entregues de les PAC de l'assignatura. En cada fita tenim una part important concreta del projecte. Aquesta divisió ha servit per portar un control del temps, de manera que en tot moment ens ha permès veure si anàvem bé o malament respecte a la planificació.

## Anàlisi funcional

Aquest apartat detalla els requeriments funcionals que s'espera que compleixi l'aplicació DJT.

### Casos d'ús



## **Descripció dels casos d'ús**

Es defineixen 3 contextos diferents on dividir els casos d'ús.

El primer context, Context 1, engloba les accions que es poden realitzar sobre la llista de jocs que existeixen en l'aplicació.

El segon context, Context 2, engloba les accions que es poden realitzar durant la creació o edició d'un joc.

El tercer context, Context 3, engloba les accions que es poden realitzar quan juguem amb a un joc.

<b>Cas d'ús</b>	Seleccionar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Generalització
<b>Propòsit</b>	Escollir un joc per executar una acció sobre aquest.
<b>Resum</b>	El cas d'ús defineix la acció de seleccionar un joc per a realitzar alguna acció sobre aquest. És una generalització dels casos d'ús Eliminar joc, Jugar amb un joc i Obrir un joc.
<b>Precondicions</b>	Cal que existeixi com a mínim un joc per poder executar el cas d'us.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Jugar joc
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Eliminar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Eliminar un joc que existeix en l'aplicació.
<b>Resum</b>	Aquest cas d'ús defineix la acció de eliminar un joc que tenim ja creat en l'aplicació.
<b>Precondicions</b>	Cal que existeixi com a mínim un joc per poder executar el cas d'us.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Eliminar joc
<b>Subflux</b>	No hi ha Subfluxes.

<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.
-------------------	---

<b>Cas d'ús</b>	Jugar amb un joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Jugar amb un joc que tenim creat en l'aplicació.
<b>Resum</b>	Aquest cas d'ús defineix les accions que es poden realitzar quan estem jugant a un joc.
<b>Precondicions</b>	Cal que existeixi com a mínim un joc per poder executar el cas d'us.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Jugar amb un joc
<b>Subflux</b>	S1 --> Seleccionar un joc --> Obrir joc --> Jugar amb un joc
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Obrir joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Generalització
<b>Propòsit</b>	Realitzar accions sobre un joc que tenim en l'aplicació.
<b>Resum</b>	Aquest cas d'ús és una generalització de les accions que es poden fer sobre un joc que comporten un canvi de context.
<b>Precondicions</b>	Cal que existeixi com a mínim un joc per poder executar el cas d'us.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Obrir joc
<b>Subflux</b>	Obrir joc S1 --> Jugar amb un joc S2 --> Editar joc
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Crear joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic



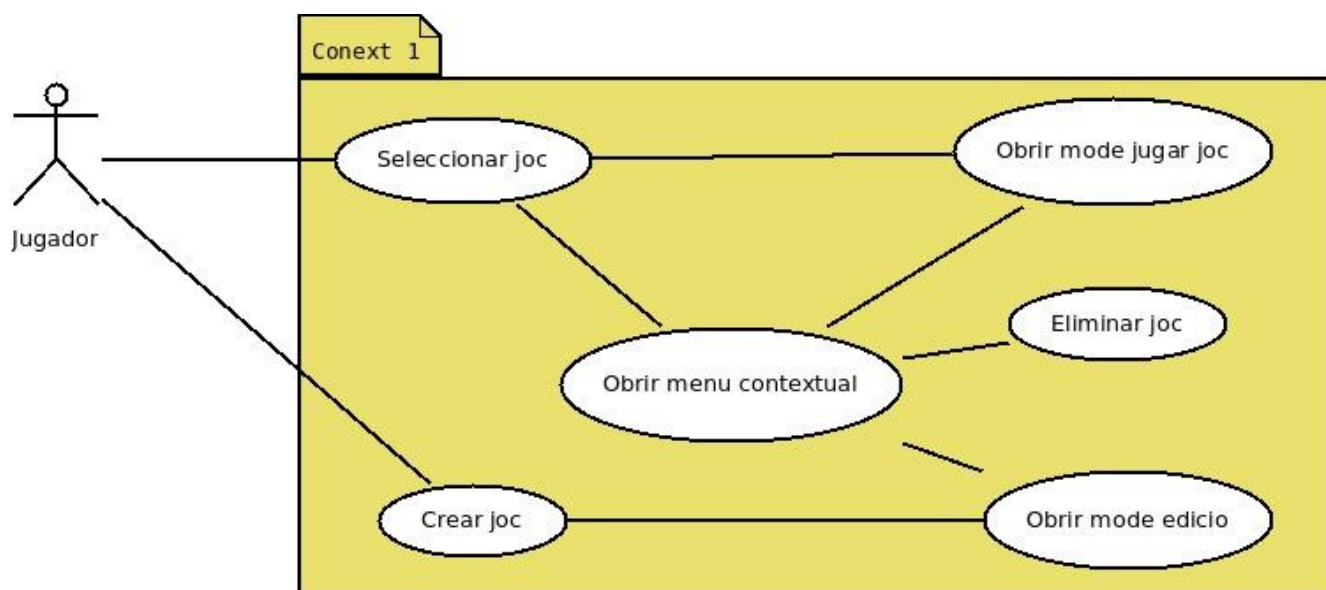
<b>Propòsit</b>	Crear un nou joc en l'aplicació.
<b>Resum</b>	Aquest cas d'ús defineix la acció que cal realitzar per crear un joc nou.
<b>Precondicions</b>	No hi ha cap precondició.
<b>Flux principal</b>	Entrar a l'aplicació --> Crear joc
<b>Subflux</b>	No hi ha Subfluxes
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Editar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Modificar un joc que tinguem en l'aplicació.
<b>Resum</b>	Aquest cas d'ús defineix totes les accions que es poden realitzar sobre un joc que existeix en el context de edició d'aquest.
<b>Precondicions</b>	Cal que existeixi com a mínim un joc per poder executar el cas d'us.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Obrir joc --> Editar joc
<b>Subflux</b>	No hi ha Subfluxes.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Definir joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Inclusió
<b>Propòsit</b>	Definir un nou joc en l'aplicació.
<b>Resum</b>	Aquest cas d'ús inclou el crear joc. Es tracta del cas d'ús que ens permet crear tots els elements del joc, seleccionar el taulell i definir els elements.
<b>Precondicions</b>	No hi ha precondicions.
<b>Flux principal</b>	Entrar a l'aplicació --> Crear joc --> Definir joc
<b>Subflux</b>	No hi ha Subfluxes.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Guardar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Inclusió
<b>Propòsit</b>	Guardar un joc en l'aplicació
<b>Resum</b>	Aquest cas d'ús inclou el definir joc i editar joc. Es tracta del cas d'ús que ens permet un cop hem definit un joc nou, o editat un d'existent guardar els canvis que hem realitzat sobre aquest.
<b>Precondicions</b>	Haver definit un joc.
<b>Flux principal</b>	Entrar a l'aplicació --> Crear joc --> Definir joc --> Guardar joc
<b>Subflux</b>	S1 --> Entrar a l'aplicació --> Seleccionar un joc --> Obrir joc --> Editar joc --> Guardar joc
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

### Detall del Context 1



En aquest context tenim els casos d'us

<b>Cas d'ús</b>	Seleccionar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Generalització
<b>Propòsit</b>	Escollir un joc per fer alguna acció sobre ell.
<b>Resum</b>	Es tracta del cas d'ús que permet seleccionar un joc. Hi ha dos possibles accions a realitzar. Una és obrir directament el joc. Això s'aconsegueix tocant el joc en la llista de jocs. L'altra és obrir un menú contextual per poder seleccionar més accions a realitzar. Les accions són els casos d'ús Obrir mode jugar joc, Eliminar Joc, Obrir mode edició joc.
<b>Precondicions</b>	Ha de existir un joc com a mínim.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc
<b>Subflux</b>	No hi ha Subfluxes.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Obrir mode jugar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Inclusió
<b>Propòsit</b>	Obrir un joc
<b>Resum</b>	Es tracta del cas d'ús que permet passar de la pantalla de selecció d'un joc a la de jugar amb aquest. Inclou la selecció del joc.
<b>Precondicions</b>	Ha de existir un joc com a mínim.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Obrir mode jugar joc
<b>Subflux</b>	S1 Entrar a l'aplicació --> Seleccionar un joc --> Obrir menú contextual --> Obrir mode jugar joc
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Eliminar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Inclusió
<b>Propòsit</b>	Eliminar un joc
<b>Resum</b>	Es tracta del cas d'ús que permet esborrar de l'aplicació un joc existent.
<b>Precondicions</b>	Ha de existir un joc com a mínim.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Obrir menú contextual --> Eliminar joc
<b>Subflux</b>	No hi ha Subfluxes.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

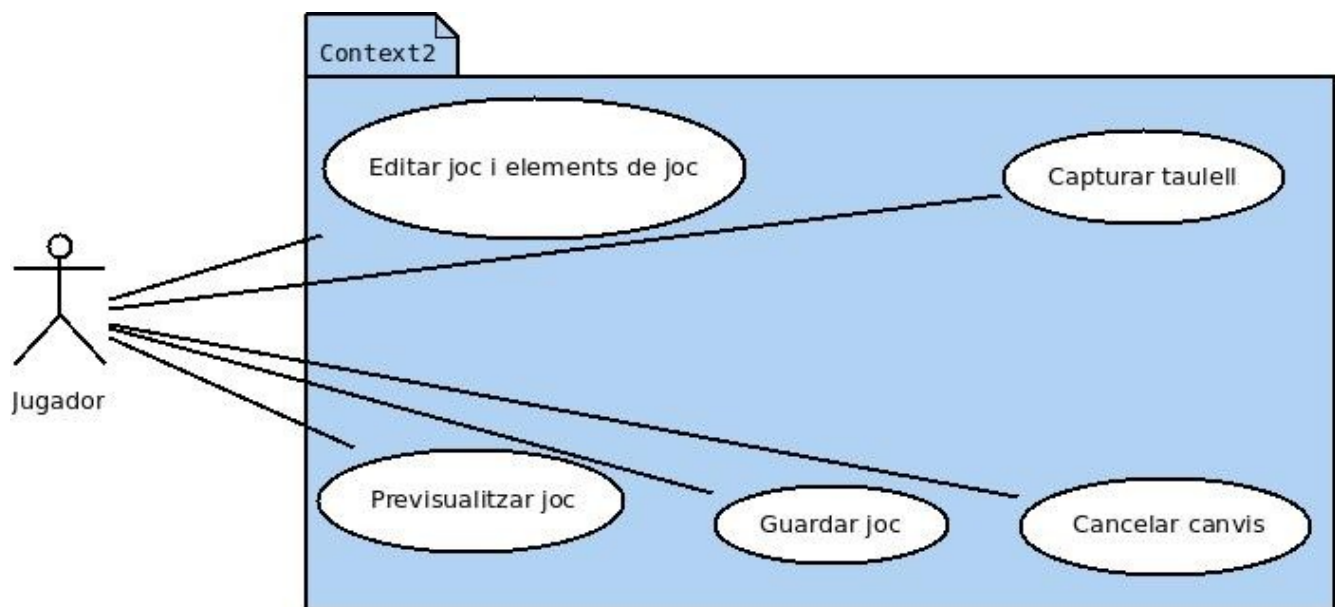
<b>Cas d'ús</b>	Obrir menú contextual
<b>Actors</b>	Jugador
<b>Tipus</b>	Inclusió
<b>Propòsit</b>	Realitzar mes accions sobre un joc
<b>Resum</b>	Mantenint uns segons tocat un joc de la llista de jocs, s'obre el menú contextual on podem accedir a mes casos d'us. Aquest cas d'ús inclou el seleccionar joc.
<b>Precondicions</b>	Ha de existir un joc com a mínim.
<b>Flux principal</b>	Entrar a l'aplicació --> Seleccionar un joc --> Obrir menú contextual
<b>Subflux</b>	No hi ha Subfluxes.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Crear joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Crear un nou joc en l'aplicació.
<b>Resum</b>	Existeix la possibilitat de crear un joc nou, tocant l'últim element de la llista de jocs.
<b>Precondicions</b>	No hi ha.

<b>Flux principal</b>	Entrar a l'aplicació --> Crear joc
<b>Subflux</b>	No hi ha Subfluxes
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Obrir mode edició
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Navegar a la part de crear o modificar un joc.
<b>Resum</b>	Aquest cas d'ús és desencadenat automàticament en executar el cas d'ús Crear joc. També és el que s'executa quan volem editar un joc.
<b>Precondicions</b>	No hi ha.
<b>Flux principal</b>	Entrar a l'aplicació --> Crear joc --> Obrir mode edició
<b>Subflux</b>	Entrar a l'aplicació --> Seleccionar joc --> Obrir menú contextual --> Obrir mode edició
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

## Detall del Context 2



<b>Cas d'ús</b>	Editar joc i elements de joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Definir el joc
<b>Resum</b>	Aquest cas d'ús ens permet definir com volem que sigui el joc. Escollir els elements de joc, i definir el nom i les instruccions.
<b>Precondicions</b>	No hi ha.
<b>Flux principal</b>	Entrar a l'aplicació --> Crear joc --> Obrir mode edició -->Editar joc i elements de joc
<b>Subflux</b>	Entrar a l'aplicació --> Seleccionar joc --> Obrir menú contextual --> Obrir mode edició -->Editar joc i elements de joc
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Capturar taulell
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Definir el taulell del joc.
<b>Resum</b>	Aquest cas d'ús ens permet escollir el taulell del joc. En la pantalla de edició del joc, podem realitzar aquest cas d'ús tocant la part dedicada a mostrar el taulell.
<b>Precondicions</b>	No hi ha.
<b>Flux principal</b>	Obrir mode edició --> Capturar taulell
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

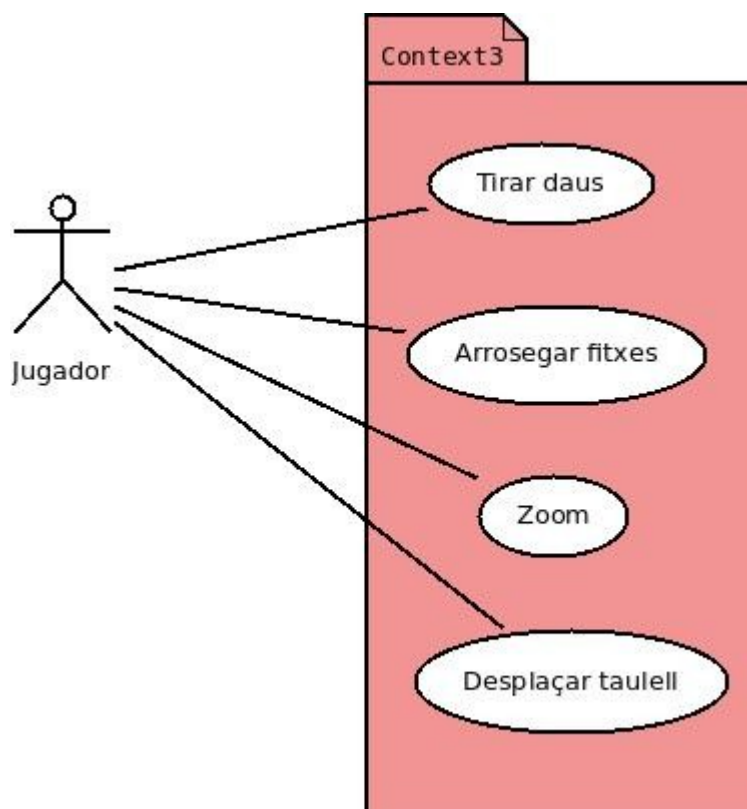
<b>Cas d'ús</b>	Previsualitzar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Permet veure com queda un joc que estem editant, en mode joc.
<b>Resum</b>	Aquest cas d'ús ens permet mentre estem definint un joc veure com queda. El que fa és mostrar en la pantalla de joc com queda el joc abans que el guardem.
<b>Precondicions</b>	No hi ha.
<b>Flux principal</b>	Obrir mode edició --> Previsualitzar joc
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Guardar joc
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Permet guardar un joc.
<b>Resum</b>	Aquest cas d'ús ens permet guardar el joc que estem editant.
<b>Precondicions</b>	No hi ha.

<b>Flux principal</b>	Obrir mode edició --> Guardar joc
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No es pot guardar si falten camps obligatoris, o si no es passen les validacions corresponents al botó guardar.

<b>Cas d'ús</b>	Cancel·lar canvis
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Permet cancel·lar els canvis en un joc.
<b>Resum</b>	Aquest cas d'ús ens permet sortir de la pantalla de edició del joc, sense que els canvis fets hagin afectat.
<b>Precondicions</b>	No hi ha.
<b>Flux principal</b>	Obrir mode edició --> Cancel·lar canvis
<b>Subflux</b>	No hi ha
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

### ***Detall del Context 3***





<b>Cas d'ús</b>	Tirar daus
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Permet fer el llançament de un dau.
<b>Resum</b>	Aquest cas d'ús ens permet fer el llançament d'un dau. Tocant al damunt de un dau s'executa el cas d'ús.
<b>Precondicions</b>	Estar en la pantalla de jugar a un joc. El joc ha de tenir daus creats.
<b>Flux principal</b>	Obrir aplicació --> Seleccionar joc --> Tirar daus
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.








<b>Cas d'ús</b>	Arrossegar fitxes
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Moure les fitxes per la pantalla.
<b>Resum</b>	Aquest cas d'ús ens permet desplaçar fitxes per damunt del taulell.
<b>Precondicions</b>	Estar en la pantalla de jugar a un joc. El joc ha de tenir fitxes.
<b>Flux principal</b>	Obrir aplicació --> Seleccionar joc --> Arrossegar fitxes
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Zoom
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Fer zoom en el joc
<b>Resum</b>	Aquest cas d'ús ens permet fer zoom en el taulell, permetent apropar la imatge en cas de necessitat per una precisió més gran. Es realitza mitjançant el gest de pinçar la pantalla cap endins per fer zoom enrere i cap a fora per fer zoom endavant.
<b>Precondicions</b>	Estar en la pantalla de jugar a un joc.
<b>Flux principal</b>	Obrir aplicació --> Seleccionar joc --> Zoom
<b>Subflux</b>	No hi ha
<b>Excepcions</b>	No hi ha excepcions en aquest cas d'ús.

<b>Cas d'ús</b>	Desplaçar taulell
<b>Actors</b>	Jugador
<b>Tipus</b>	Bàsic
<b>Propòsit</b>	Moure el taulell per centrar el punt que vulguem
<b>Resum</b>	Aquest cas d'ús ens permet desplaçar el taulell de manera que podem centrar el punt que vulguem. Això pot ser molt útil en cas de fer zoom. Per realitzar aquesta acció cal tocar la pantalla amb dos dits i sense deixar de tocar fer el desplaçament cap on vulguem.
<b>Precondicions</b>	Estar en la pantalla de jugar a un joc.
<b>Flux principal</b>	Obrir aplicació --> Seleccionar joc --> Desplaçar taulell
<b>Subflux</b>	No hi ha.
<b>Excepcions</b>	No es pot desplaçar si està tot complert en la pantalla.

## Disseny de pantalles

### Selecció de un joc

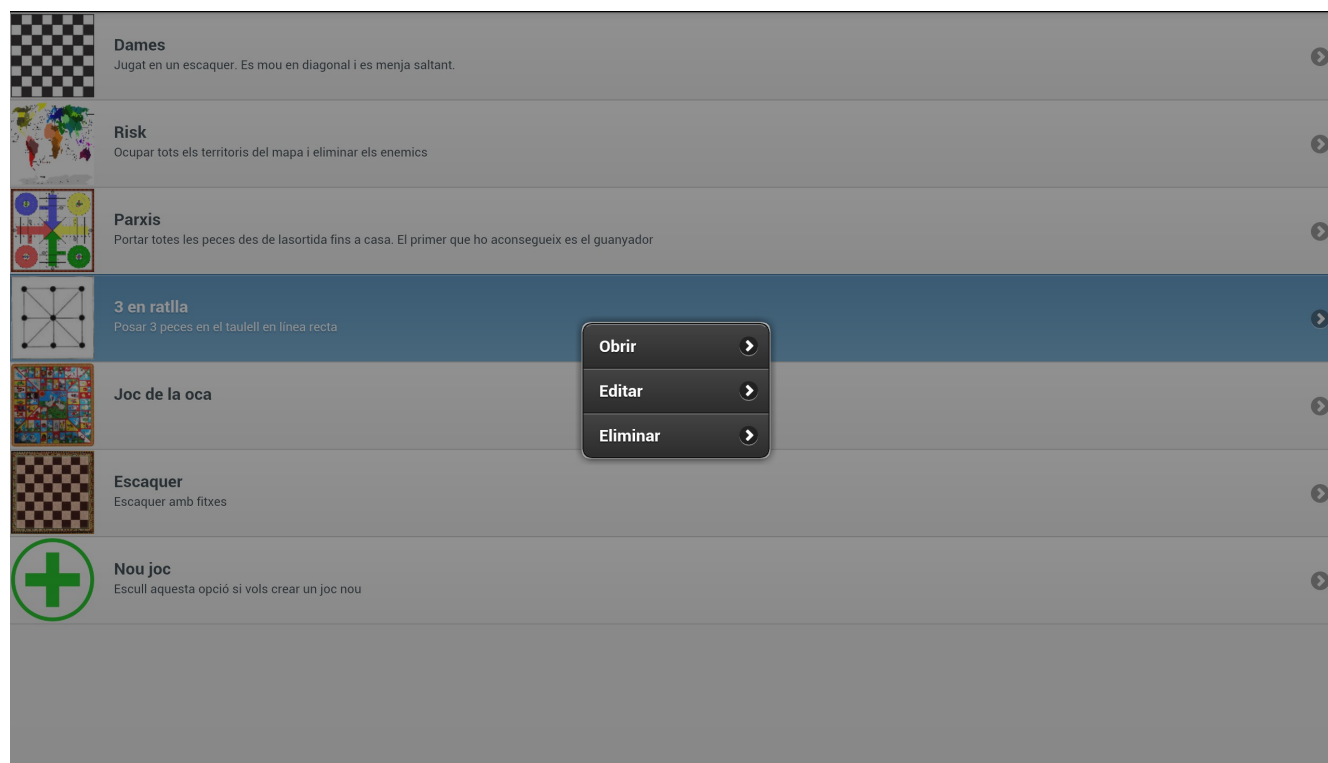
	<b>Dames</b> Jugat en un escaquer. Es mou en diagonal i es menja saltant.	>
	<b>Risk</b> Ocupar tots els territoris del mapa i eliminar els enemics	>
	<b>Parxis</b> Portar totes les peces des de la sortida fins a casa. El primer que ho aconsegueix es el guanyador	>
	<b>3 en ratlla</b> Posar 3 peces en el tauler en línia recta	>
	<b>Joc de la oca</b>	>
	<b>Escaquer</b> Escaquer amb fitxes	>
	<b>Nou joc</b> Escull aquesta opció si vols crear un joc nou	>

Cada pantalla representa un context de casos d'ús. Aquesta pantalla correspon al Context 1. Està formada únicament per una llista on hi haurà tots els jocs de taula que tinguem creats. L'últim element de la llista és un element diferent que tindrà la funció de crear un nou joc. En cas que no hi hagi cap joc, apareixerà aquest únic element.

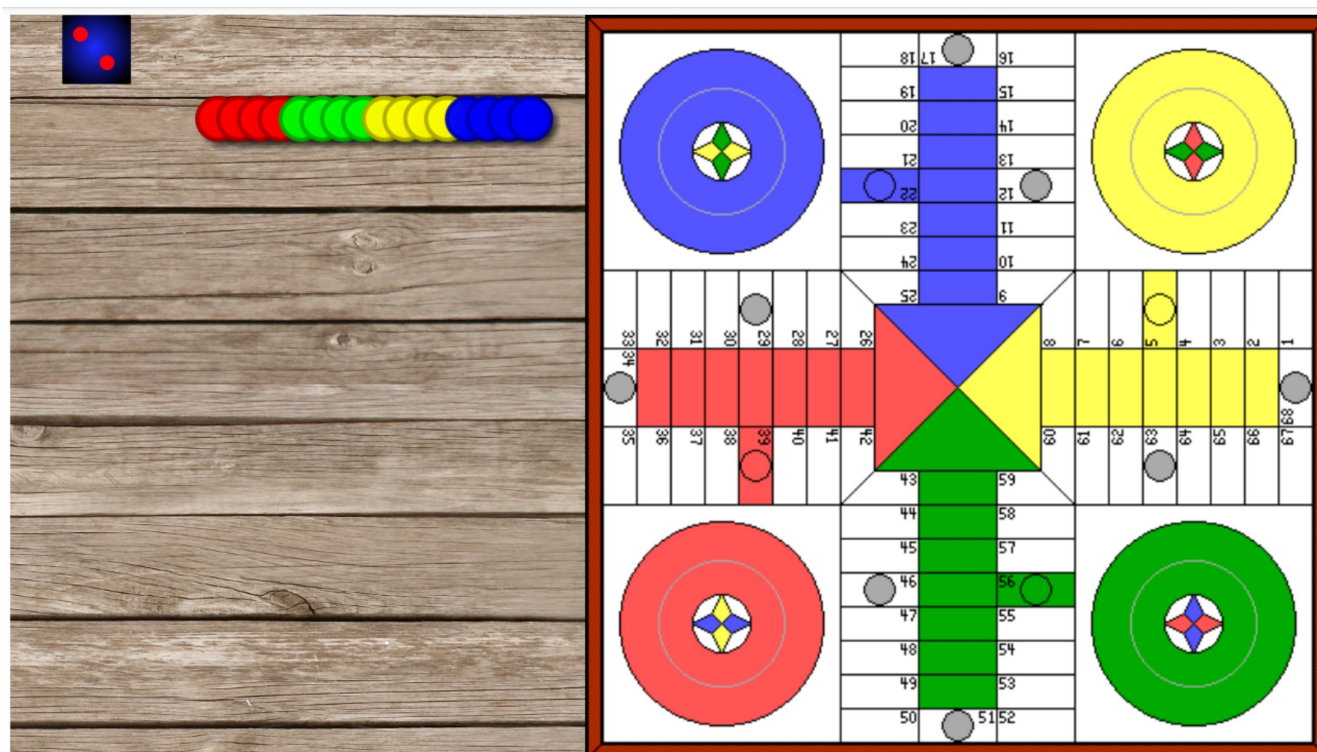
Acció	Cas d'ús	Descripció
Tocar al damunt d'un element de la llista	Seleccionar joc. Obrir mode jugar joc.	Anirem a la pantalla de jugar i s'obrirà amb el joc seleccionat.
Scroll sobre la llista	Seleccionar joc.	Ens podrem desplaçar per la llista de jocs fent scroll amb el dit.
Clic al damunt de l'últim element de la llista	Crear joc Obrir mode edició	Anirem a la pantalla de edició de jocs amb el formulari buit per tal de crear un joc nou.

llista (Afegir joc)		
Mantenir element de la llista polsat	Obrir menú contextual	Es desplega el menú contextual (veure imatge Menú contextual 1).
Editar (Menu1)	Obrir mode edició	Ens movem a la pantalla de edició del joc amb totes les dades del joc carregades per tal de poder editar-les.
Eliminar (Menu1)	Eliminar joc	Eliminarà totalment el joc del dispositiu.
Obrir (Menu1)	Obrir mode jugar joc	Igual que tocar al damunt de la llista, anirem a la pantalla de jugar i s'obrirà amb el joc seleccionat.

## Menú contextual 1



## Jugar a un joc



Aquesta pantalla es correspon al Context 3. Ens permet jugar a un joc. Esta formada per dos parts:

El taulell i els elements de joc.

Es presentarà el fons com si fos una taula, amb una textura de fusta.

El taulell mantindrà l'aspecte original de la imatge.

El taulell apareixerà el màxim a la dreta deixant un petit marge per que sembli que es troba damunt d'una taula.

Els elements de joc apareixeran a la esquerra al damunt de la taula. Aquests elements no estaran ordenats de cap manera concreta però si que apareixeran en una posició còmoda, un al costat de l'altre de esquerra a dreta i saltant de línia quan no hi càpiguen mes.

Acció	Cas d'ús	Descripció
Arrossegar fitxes	Arrossegar fitxes	Al tocar una fitxa la podrem arrossegar per la pantalla per posar-la on toqui.
Tocar daus	Tirar daus	Al tocar un dau aquest farà una petita animació

		canviant la cara mostrada. Sonarà durant la animació el so de un dau al ser tirat. La animació durarà 2 segons.
Fer zoom (no implementat)	Zoom	Utilitzant el gest de pinçar, es podrà fer zoom sobre el taulell per tal de augmentar la precisió en cas de necessitat. El zoom cap endins es realitza pinçant cap a fora i el zoom cap a fora es realitza pinçant cap endins.
Desplaçament (no implementat)	Desplaçar taulell	Al tenir la possibilitat de fer zoom apareix la necessitat de realitzar desplaçaments del taulell per centrar la atenció al punt que ens interressi. Això es realitza tocant la pantalla amb dos dits a la vegada i arrossegant aquests cap on volem realitzar el desplaçament.

## Manteniment d'un joc

Nom:

Instruccions:  
 Portar totes les peces des de la sortida fins a casa. El primer que ho aconsegueix es el guanyador

Camera
 Galeria

Quantitat	Tipus	Color	Tamany		Quantitat	Tipus	Min	Max	
<input type="text" value="4"/>	<input type="button" value="Cercle"/>	<input type="button" value="Vermell"/>	<input type="text" value="20"/>	<input type="button" value="-"/>	<input type="text" value="1"/>	<input type="button" value="Dau estandar"/>	<input type="text" value="1"/>	<input type="text" value="6"/>	<input type="button" value="-"/>
<input type="text" value="4"/>	<input type="button" value="Cercle"/>	<input type="button" value="Verd"/>	<input type="text" value="20"/>	<input type="button" value="-"/>	<input type="button" value="+"/>				
<input type="text" value="4"/>	<input type="button" value="Cercle"/>	<input type="button" value="Groc"/>	<input type="text" value="20"/>	<input type="button" value="-"/>					
<input type="text" value="4"/>	<input type="button" value="Cercle"/>	<input type="button" value="Blau"/>	<input type="text" value="20"/>	<input type="button" value="-"/>					
<input type="button" value="+"/>									

Guardar

Aquesta pantalla es correspon al Context 2. És la pantalla on es realitza el manteniment de un joc en concret. Com a manteniment entenem la alta/modificació de un joc.

Un joc consta de:

Nom: És un camp obligatori. Serà el nom del joc.

Instruccions: Un camp de text opcional on s'explicarà si es vol les instruccions del joc.

Taulell: Apareixerà una icona en cas que no la tinguem definida sinó sortirà la imatge del taulell.

Fitxes: Apartat per definir les fitxes del joc

Daus: Apartat per definir els daus del joc.

Acció	Cas d'ús	Descripció
Prémer botó “Guardar”	Guardar joc	Es valida que el nom no estigui buit i que no existeixi. Es guarda el joc i tornem a la pantalla de selecció del joc. En cas contrari es mostra el missatge “Nom de joc ja existent” o “El nom del joc no pot estar buit”, i ens quedem en la mateixa pantalla.
Prémer botó “Enrere”	Cancel·lar canvis	Es cancel·len els canvis i sortim a la llista de jocs.
Prémer el boto càmera.	Capturar taulell	S'obre l'aplicació de la càmera del dispositiu.
Prémer el botó galeria.	Capturar taulell	S'obre la galeria del dispositiu.
Retornar de captura/selecció	Capturar taulell	Es mostra en el requadre del taulell la imatge seleccionada.

### **Fitxes**

En aquest apartat es defineixen les fitxes del joc.

L'apartat s'organitza en files. En cada fila tenim 4 elements:

- El primer es tracta d'un camp per escollir el numero de fitxes.
- El segon serà una llista desplegable on apareixeran tots els tipus de fitxes que puguem escollir.
- El tercer serà un camp per escollir el color de les fitxes.
- El quart serà un camp per escollir el tamany de les fitxes.

Sempre apareix com a mínim una fila amb un botó per afegir una nova fila.

### **Tipus de fitxes**

Es podran escollir entre els següents tipus de fitxes. Totes elles podran ser de qualsevol color.



### **Daus**

En aquest apartat es defineixen els daus del joc.



L'apartat s'organitza en files. En cada fila hi tenim:

- Un primer camp per seleccionar el numero de daus.
- Un segon camp de tipus llista desplegable per escollir el tipus de dau.
- Només en cas que el tipus sigui numèric, apareixeran dos camps més on es podrà escollir el valor màxim i mínim del dau (no es fa servir actualment).

### **Tipus de daus**

Hi ha 2 tipus de daus diferents.

- Dau amb punts, amb les cares de 1 punt fins a 6 punts.
- Dau de poker.
- Dau numèric, amb valor mínim i màxim a escollir per el jugador.(no implementat)

## **Diari de desenvolupament (Anàlisi funcional)**

Començant per els casos d'us, ha estat molt complicat definir aquests per arribar a tenir el que hi ha finalment.

En una primera versió no s'havien definits contextos, els casos d'ús no eren del tot clars i possiblement massa genèrics.

Després de donar-hi moltes voltes, la definició d'aquests s'ha anat acostant a un model de casos d'us, que ell mateix és auto-explicatiu i ens permet veure exactament que fa l'aplicació.

Els contextos ens permeten veure com està dividida l'aplicació i com es fa per passar d'un context a un altre.

En el primer diagrama s'han intentat definir els casos d'ús globals de l'aplicació, a continuació s'han detallat els casos d'ús de cada context.

Tot i que teníem clar el que es volia que fes l'aplicació, com ja s'ha comentat, fer el diagrama de casos d'ús no ha estat senzill i ha calgut refer-lo varies vegades fins aconseguir el que s'ha realitzat finalment.

Pel que fa al disseny de les pantalles, es pot veure que tots els estils són de Jquerymobile. En un principi es van definir unes pantalles semblants, però que no utilitzaven aquests estils.

Després de veure diferents aplicacions i webs fetes amb Jquerymobile, i en vista de la dificultat de crear uns estils propis totalment a part de Jquerymobile, s'ha decidit que no tenia sentit.

Utilitzar els estils de Jquerymobile ens aportarà moltes més avantatges. Per començar, la gent que està acostumada a fer servir el mòbil per navegar o ja ha fet servir altres aplicacions fetes amb Jquerymobile, les quals són moltes, trobaran la nostra més amigable, com si ja estiguessin acostumats a utilitzar-la degut a que tenen una aparença semblant.

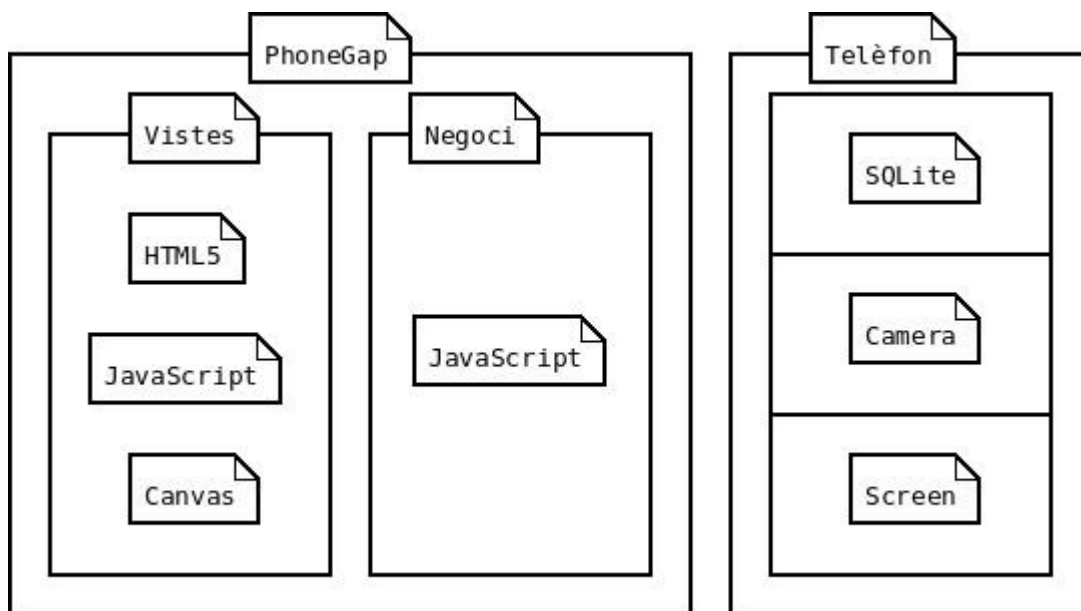
Un altre aspecte és que amb els estils de Jquerymobile ja ens assegurem que els elements de la pantalla encaixin correctament aprofitant tot l'espai d'aquesta i que tinguin una interfície usable amb dispositius

tàctils.

## Anàlisi tècnic

### Arquitectura

Per la realització de l'aplicació s'utilitza Phonegap. L'estructura que té la nostra aplicació és la següent:



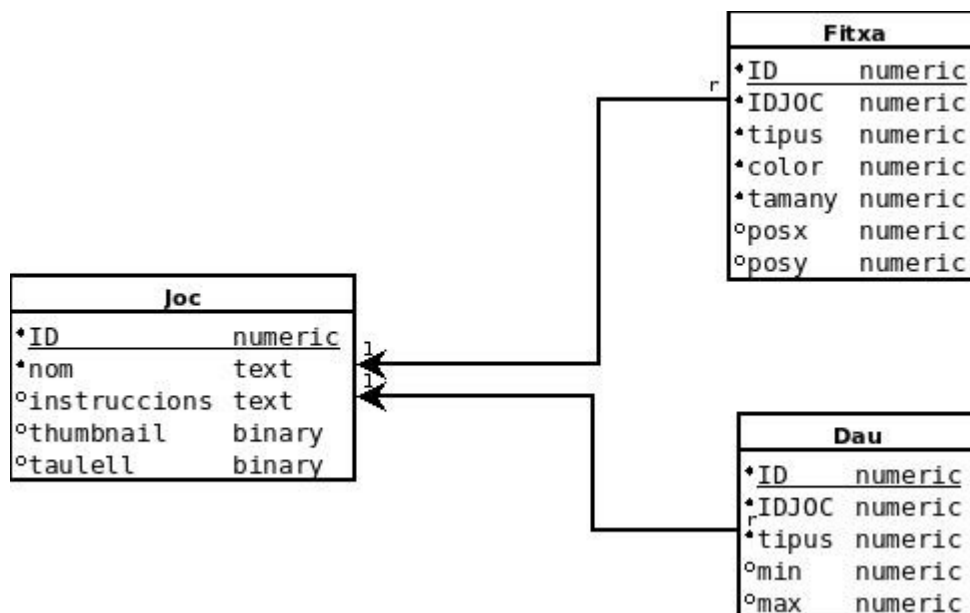
Com es pot veure en la imatge tenim que, les vistes es realitzaran utilitzant d'una banda HTML5 i JavaScript i de l'altra canvas (que no deixa de ser html5 i JavaScript).

Aquestes vistes utilitzaran una capa de negoci, on estarà tota la funcionalitat de l'aplicació implementada amb JavaScript. Aquesta capa de negoci alhora accedirà al hardware del telèfon, concretament a l'emmagatzematge en una base de dades SQLite, a la càmera i a la pantalla.

### Base de dades

Degut a que l'aplicació està desenvolupada per a dispositius mòbils d'última generació, es pot utilitzar per tal de guardar la informació una petita base de dades SQLite la qual es pot crear, mantenir i utilitzar amb Phonegap.

## Diagrama entitat-relació



### Taula Joc

La taula Joc és on es guarden els jocs que tenim en l'aplicació.

El camp **ID**, és l'identificador únic de la fila.

El camp **nom**, és obligatori i és únic en la taula.

El camp **instruccions** serveix per escriure les instruccions del joc.

El camp **thumbnail** és un camp on es guarda la imatge del taulell a tamany reduït per mostra-ho a les llistes.

El camp **taulell** és un camp de tipus binary on es guarda la imatge del taulell.

### **Taula Fitxa**

La taula fitxa és on es guardaran totes les fitxes definides en els jocs i on es relacionaran aquestes amb cada joc.

El camp **ID**, és l'identificador únic de la fila.

El camp **IDJOC**, és una clau forana i relaciona la fitxa amb el joc.

El camp **tipus**, defineix quin tipus de fitxa es.

El camp **color**, defineix el color de la fitxa.

El camp **tamany**, defineix el tamany de la fitxa.

El camp **posx**, defineix en l'eix de les x la posició de la fitxa en el joc.

El camp **posy**, defineix en l'eix de les y la posició de la fitxa en el joc.

### **Taula Dau**

El camp **ID**, és l'identificador únic de la fila.

El camp **IDJOC**, és una clau forana que relaciona el dau amb el joc.

El camp **tipus**, defineix el tipus de dau que es tracta.

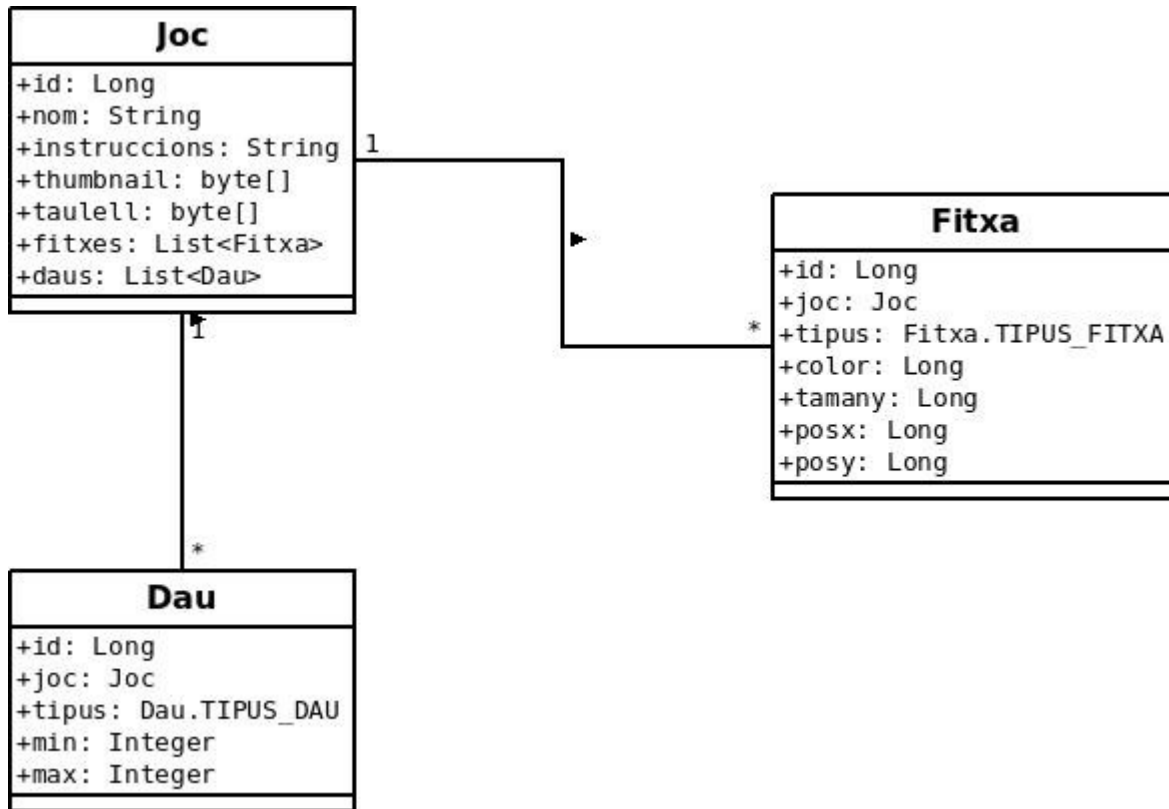
El camp **min**, serveix per als daus numèrics i defineix el valor mínim del dau.

El camp **max**, serveix per als daus numèrics i defineix el valor màxim del dau.

### ***Disseny de l'aplicació***

La lògica de negoci de l'aplicació es realitza en JavaScript. Existeixen tècniques per poder utilitzar classes en JavaScript. Degut a que això ens pot ajudar molt a estructurar correctament el projecte i a reutilitzar molt codi, utilitzarem classes.

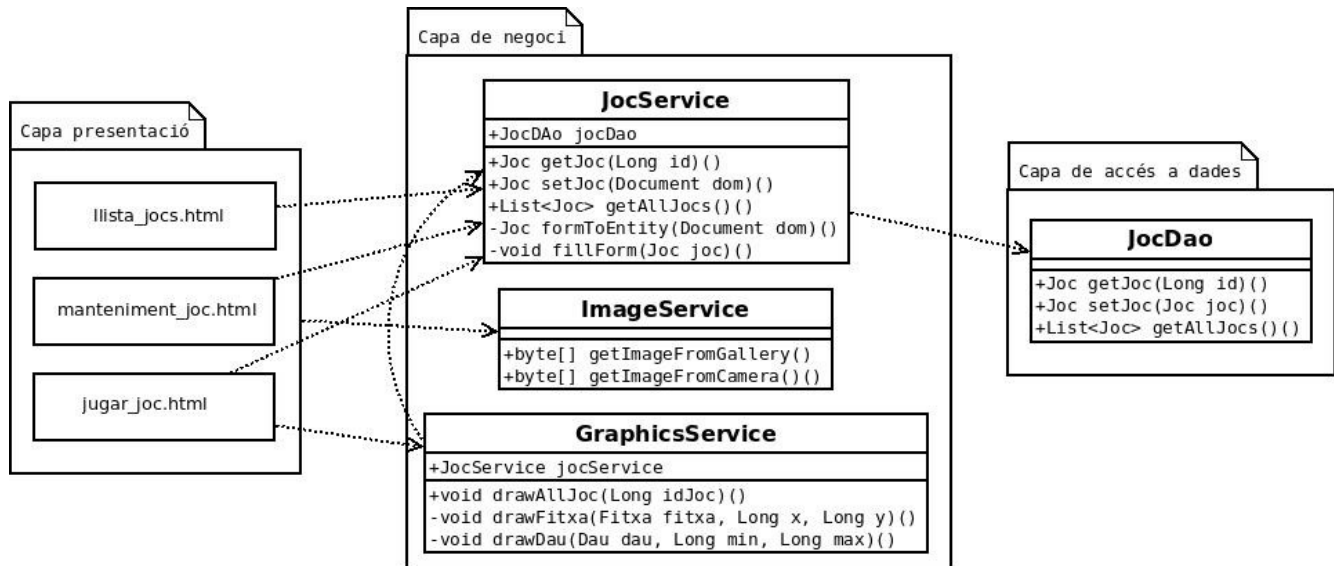
## Diagrama de classes - Entitats



Les classes que existeixen tenen una estructura similar a les taules de la base de dades. El que es pretén amb això és poder treballar amb objectes podent passar aquests de un context a un altre de l'aplicació.

## Diagrama de classes - Model

Sense complir estrictament amb cap model de disseny concret, l'aplicació quedarà distribuïda en tres capes. Una capa de presentació on es posaran totes les vistes realitzades amb HTML. Una capa de negoci que farà les operacions necessàries de l'aplicació i una capa d'accés a dades que com el seu nom indica s'encarregarà de fer les operacions de carregar, guardar, eliminar o qualsevol operació relacionada amb la base de dades.



### Capa d'accés a dades

En la capa d'accés a dades tenim una classe *JocDao* que disposa dels mètodes necessaris per poder guardar o actualitzar un joc a la base de dades, poder recuperar un joc ja guardat o poder recuperar la llista de jocs.

### Capa de negoci

En la capa de negoci tenim tres classes diferents. La primera *JocService* la qual te com a dependència la classe de la capa d'accés a dades *JocDao* degut a que la utilitza en els seus propis mètodes.

Aquesta classe és capaç de recollir la informació que li envia la capa de presentació, per poder modificar-la i preparar-la per poder ser persistida o processada de la manera que pertorqui.

La classe *ImageService* te la capacitat d'obtenir imatges tant de la càmera com de la galeria del mòbil. S'utilitza des de la capa de presentació, en la pantalla de *manteniment\_joc.html* per tal de seleccionar el taulell.

La classe *GraphicsService* te tota la lògica que fa falta en la pantalla de *jugar\_joc.html*. És a dir te la capacitat de dibuixar un joc en la pantalla i crear els elements que pertorquin, així com permetre fer les accions pertinents a arrossegar fitxes i llençar daus. Te com a dependència la classe *JocService*, de manera que ella sola és capaç amb l'identificador del joc (atribut id) de pintar tot un joc sencer.

### Capa de presentació

En la capa de presentació hi tenim les tres pantalles en format HTML. Aquestes pantalles seran el mes semblants possible al disseny definit en aquest mateix document en l'apartat [Disseny de pantalles](#).

## Diari de desenvolupament (Anàlisi tècnic)

Pel que fa al disseny tècnic de l'aplicació, s'ha fet una separació de totes les parts del codi en capes. Així tenim que, tot el codi necessari per accedir a base de dades queda definit en un sol fitxer i des de la resta de l'aplicació treballarem sempre amb entitats.

Això és una gran avantatge, sobretot en casos on sigui possible que aquesta part hagi de canviar. Només s'ha realitzat el desenvolupament per Android, tot i que en principi no hi ha cap part que no hagi de funcionar en la resta de plataformes compatibles amb Phonegap.

Si algun dispositiu on es vol portar l'aplicació no disposa de Sqlite i s'ha de treballar per exemple amb fitxers, no caldria tocar res més que la llibreria que s'ha fet, adaptar-la per treballar amb fitxers en comptes de Sqlite i l'aplicació seguirà funcionant igual. També es podria voler que les dades es guardessin en una base de dades comuna per tothom. Novament caldria canviar la llibreria d'accés a dades, i la resta de l'aplicació quedaria exactament igual.

En el disseny es pot veure també que les vistes HTML queden separades de la lògica de negoci. Per un tema primerament de comoditat i seguidament per que els mètodes van molt lligats al que veiem en una pantalla concreta i només cal utilitzar-los en un sol lloc, els fitxers HTML tenen la lògica de negoci. Això no vol dir que no existeixi aquesta separació, però sí que és més lògica que no pas física. Cada pantalla té un funcionament totalment diferent, de manera que no ha calgut fer una llibreria de negoci que sigui utilitzada per més de una vista.

## Implementació

### *Base de dades*

### Model relacional

Per tal de crear el model físic de la base de dades s'ha fet un script que s'executa el primer cop que obrim l'aplicació. Aquest script crea les taules i les relacions entre elles:

```
CREATE TABLE IF NOT EXISTS JOC (id INTEGER PRIMARY KEY, nom TEXT, instruccions TEXT, thumbnail BLOB, taulell BLOB);

CREATE TABLE IF NOT EXISTS FITXA (id INTEGER PRIMARY KEY, idjoc INTEGER REFERENCES JOC(id), tipus INTEGER, color TEXT, tamany INTEGER, posx INTEGER, posy INTEGER);

CREATE TABLE IF NOT EXISTS DAU (id INTEGER PRIMARY KEY, idjoc INTEGER REFERENCES JOC(id), tipus INTEGER, min INTEGER, max INTEGER);
```

## Model de objectes

Per tal de treballar amb objectes s'han creat les entitats necessàries per vincular les taules de la base de dades.

A continuació es mostra una petita extracció de codi on es pot veure com s'implementen objectes en Javascript. S'han eliminat tots els mètodes de get i set excepte els de fitxes, daus, i id, ja que la resta són exactament iguals per cada atribut.

Com podem veure s'ha creat un mètode per afegir daus i un altre per afegir fitxes a un objecte joc, de manera que des de fora es pugui treballar amb objectes joc alhora de fer crides a mètodes d'accés a dades.

```
function Joc(pid, pnom, pinstruccions, pthumbnail, ptaulell) {  
  
    this.id = pid;  
    this.nom = pnom;  
    this.instruccions = pinstruccions;  
    this.thumbnail = pthumbnail;  
    this.taulell = ptaulell;  
    this.fitxes = [];  
    this.daus = [];  
  
    this.getFitxes = function() {  
        return this.fitxes;  
    };  
  
    this.addFitxa = function(fitxa) {  
        this.fitxes[this.fitxes.length] = fitxa;  
    };  
  
    this.getDaus = function() {  
        return this.daus;  
    };  
  
    this.addDau = function(dau) {  
        this.daus[this.daus.length] = dau;  
    };  
  
    this.getId = function() {  
        return this.id;  
    };  
  
    this.setId = function(idin) {  
        this.id = idin;  
    };  
  
    // La resta de getters i setters per cada atribut  
  
}
```



```
function Fitxa(pid, pidJoc, ptipus, pcolor, ptamany, pposx, pposy) {
    this.id = pid;
    this.idJoc = pidJoc;
    this.tipus = ptipus;
    this.color = pcolor;
    this.tamany = ptamany;
    this.posx = pposx;
    this.posy = pposy;

    //Getters i setters de cada atribut
}

function Dau(pid, pidJoc, ptipus, pmin, pmax) {

    this.id = pid;
    this.idJoc = pidJoc;
    this.tipus = ptipus;
    this.min = pmin;
    this.max = pmax;

    // Getters i setters de cada atribut
}
```

## Mètodes d'accés a dades

Pel que fa a mètodes d'accés a dades, s'han creat 4 funcions que ens permeten realitzar totes les accions que es requereixen.

```
function getJocById(pid, callback){
    //Implementació
}
```

Aquesta funció carrega un joc, concretament el que tenim a la base de dades amb l'id que se li passa en el paràmetre *pid*.

Degut a que en Phonegap les transaccions a la BBDD son asíncrones, ha calgut afegir un paràmetre més, el qual és l'identificador del mètode callback que es cridarà un cop acabat de carregar el joc.

El que fa el mètode es seleccionar primerament un el joc amb l'id indicat de la taula JOC. Un cop seleccionat el joc, amb la fila resultant de la base de dades es crea un nou objecte de la entitat Joc. A continuació es carreguen de la mateixa manera totes les fitxes, les quals creen un nou objecte fitxa per

cada fila retornada i s'afegeix aquest objecte al objecte joc creat. Finalment es fa exactament el mateix amb els daus.

Un cop carregades totes les dades s'executa el mètode que hem informat de callback el qual rebrà com a paràmetre l'objecte Joc (amb totes les fitxes i daus).

```
function setJoc(joc, callback) {  
    //Implementació  
}
```

Aquest mètode rep un objecte de tipus Joc i el persisteix a la base de dades. Un cop persistit llança un mètode callback en el qual informa del id del joc que s'ha inserit.

En Sqlite els id son auto-generats, de manera que fins que no inserim un joc, no tenim l'id d'aquest.

L'id és necessari degut a que si no el tenim, el segon cop que guardem es pensarà que és un objecte Joc nou i crearà una nova fila amb el joc (és a dir, duplicarà el joc).

La implementació del mètode és fixa si existeix ja a la base de dades el joc que estem guardant. Per saber-ho només cal veure si l'id ja té valor. En aquest cas el que fa és actualitzar el joc amb els nous canvis que s'hagin pogut fer. Si en canvi l'id no té valor s'afegirà una fila nova.

Per guardar les files i els daus del joc, el que es fa sempre és eliminar les que ja existien per al joc que guardem i a continuació inserir les que es volen guardar.

```
function getAllJocs(callback){  
    //Implementació  
}
```

Aquest mètode carrega tots els elements de la taula JOC i crea un objecte Joc per cadascun d'ells. No carrega ni fitxes ni daus. S'utilitza per la pantalla de selecció de jocs, per llistar-los tots.

El mètode callback que rep el crida en tenir tots els jocs carregats i passa una llista de jocs com a paràmetre.

```
function deleteJoc(idJoc){  
    //Implementació
```

```
}
```

L'ultima funció, serveix per eliminar un joc. Se li passa l'id del joc que es vol eliminar i dins s'executen les sentències per eliminar tant les fitxes com els daus i el joc.

Tot el codi relacionat amb base de dades es troba concentrat en la llibreria Javascript amb nom **djtdatabase.js**.

S'ha agrupat tot en un fitxer ja que seran mètodes que cal fer servir des de tota l'aplicació. A mes a mes és una part susceptible de ser extreta per ser canviada en cas que es vulgui realitzar una ampliació de l'aplicació.

Arribat a aquest punt de la implementació ja podem treballar a nivell de negoci utilitzant les funcions exposades.

## ***Selecció de joc***

La pantalla de selecció de joc es correspon al fitxer llista\_jocs.html. Aquesta pagina HTML utilitza les llibreries i estils de Jquerymobile. També utilitza la nostra llibreria d'accés a base de dades i per tant la llibreria de Phonegap. És la vista que es carrega al entrar a l'aplicació.

Mètodes:

```
function pintarLlistaJocs(jocs) {  
    //Implementació  
}
```

Aquest mètode rep una llista de objectes Joc i és capaç de crear amb Jquerymobile la llista amb les imatges en miniatura i la informació pertinent.

```
function eliminarJoc(id){  
    //Implementacio  
}
```

Aquest mètode és el que crida al mètode que elimina un joc de la llibreria de base de dades i refresca la llista de jocs.

El funcionament és el següent:

Al carregar la pagina es crida el mètode “**getAllJocs(pintarLlistaJocs);**”. Aquesta és una crida a la funció definida en la llibreria djtdatabase.js i com a paràmetre de callback se li passa la funció definida en el fitxer HTML on ens trobem (llista\_jocs.html).

Això produeix que es carreguin els jocs guardats a la base de dades.

A part dels elements retornats del a base de dades, sempre s'afegeix a la llista l'element de crear un joc nou.

### Funcionalitat de cada element de la llista

Utilitzant Jquerymobile, s'ha afegit funcionalitat a cada element de la llista. S'ha definit el mètode que s'executa en fer taphold (aguantar tocat durant un segons) a la llista. Aquest mètode reconeix l'element on es fa taphold i obre un popup.

Resumint el codi que fa això és el següent:

```
$('#listId').on('taphold', function(e){  
    ...  
    $('#popupMenu').popup('open');  
});
```

El popup s'ha definit amb Jquerymobile. Això s'ha fet creant un div amb l'atribut data-role=popup.

El menú contextual disposa de 3 links, el primer per obrir el joc, el segon per editar el joc i el tercer crida la funció que elimina el joc (definida en aquest mateix fitxer HTML).

### ***Manteniment de joc***

La estructura de la pàgina s'ha creat amb Jquerymobile. S'han defineixen 5 àrees.

- Camps de text.
- Imatge del taulell.
- Taula de fitxes.
- Taula de daus.
- Botonera inferior

S'han definit dos distribucions possibles. Una per a pantalles grans, on la taula de fitxes i la de daus es troben de costat i una altra distribució per a pantalles petites on la taula de fitxes i la de daus es troben

una al damunt de l'altra. Per fer-ho, s'han creat diferents divs niats. Aquests divs tenen un l'atribut class que identifica com cal mostrar el div.

Segons el tamany de la pantalla es mostra una configuració o una altra d'aquests divs.

L'estructura per a pantalles grans s'ha definit així:

```
<div id="pare" data-role="page">
  <div id="pare" data-role="content">
    <div id="gridfila1" class="ui-grid-a">
      <div id="divnom" class="ui-block-a">
        <!-- CAMPS DE TEXTE -->
      </div>
      <div id="divimatge" class="ui-block-b">
        <!-- Imatge del taulell -->
      </div>
    </div>
    <div id="gridfila2" class="ui-grid-a">
      <div id="divfitxes" class="ui-block-a">
        <!-- Taula de fitxes -->
      </div>
      <div id="divdaus" class="ui-block-b">
        <!-- Taula de daus -->
      </div>
    </div>
    <div id="gridbotons" class="ui-grid-solo">
      <div id="divbotons" class="ui-block-a">
        <!-- Botons -->
      </div>
    </div>
  </div>
</div>
```

La estructura per a pantalles petites s'ha definit així:

```
<div id="pare" data-role="page">
  <div id="pare" data-role="content">
    <div id="gridfila1" class="ui-grid-a">
      <div id="divnom" class="ui-block-a">
        <!-- CAMPS DE TEXTE -->
      </div>
      <div id="divimatge" class="ui-block-b">
        <!-- Imatge del taulell -->
      </div>
    </div>
    <div id="gridfila3" class="ui-grid-solo">
      <!-- Taula de fitxes -->
    </div>
    <div id="gridfila4" class="ui-grid-solo">
```

```
        <!-- Taula de daus -->
    </div>

    <div id="gridbotons" class="ui-grid-solo">
        <div id="divbotons" class="ui-block-a">
            <!-- Botons -->
        </div>
    </div>
</div>
```

A continuació es detallen les parts de codi mes rellevant de la lògica d'aquesta part.

## Càmera/Galeria

Phonegap disposa d'accés a la càmera del dispositiu així com d'accés a la Galeria d'aquest per tal d'obtenir imatges o fitxers i utilitzar-los en la nostra aplicació. Els mètodes de Phonegap que hem utilitzat son:

```
//Obtenir imatge des de camera
navigator.camera.getPicture(metodeCallbackCamera,                callbackError,
{ destinationType : destinationType.DATA_URL});

//Obtenir imatge des de fitxer
navigator.camera.getPicture(metodeCallbackGaleria,                callbackError,
{ destinationType : destinationType.FILE_URI});
```

El que s'ha implementat a continuació és el que es vol fer amb les imatges que ens ha retornat.

Els mètodes de callback que se li passa a aquestes funcions reben com a paràmetre la imatge. El de la càmera rep una cadena en base64, en canvi el de la galeria rep una URL amb la ruta del fitxer escollit.

Un cop obtinguda la imatge, s'ha implementat un mètode en Javascript que crea una nova imatge re-escalada a partir de la original.

Per fer això s'ha utilitzat l'element canvas de HTML5. El funcionament és el següent:

1 – Es crea un element canvas del tamany que volem la imatge.

```
var canvas = document.createElement("canvas");
canvas.width = 100;
canvas.height = 100;
```

2 - En el context del canvas es pinta la imatge i se li diu en quin punt comença i en quin acaba

```
canvas.getContext("2d").drawImage(sourceImage, 0, 0, 100, 100);
```

### 3 – S'exporta el canvas a imatge

```
canvas.toDataURL();
```

En aquest punt es té la imatge original i la imatge re-escalada.

## Taula de fitxes i taula de daus

La implementació d'aquestes dos taules és pràcticament idèntica. El que val per una val per l'altre.

Que es puguin afegir i treure files de la taula, no ha estat senzill. Tampoc ho ha estat construir-la a partir d'objectes Fila o Dau ni de recórrer-la per guardar els valors que s'hagin introduït.

A continuació s'explica el funcionament basant-se en la taula de fitxes, però cal dir que per la de daus és exactament igual. No és el mateix codi ja que les taules són diferents, però l'algoritme que es fa servir és idèntic.

Per començar, existeix sempre la taula definida en HTML:

```
<table id="taulaFitxes">
<thead>
<tr><td>Quantitat</td><td>Tipus</td><td>Color</td><td>Tamany</td><td></td></tr>
</thead>
<tbody>
</tbody>
<tfoot>
<tr><td colspan="5"></td>
</tr>
</tfoot>
```

Aquesta taula té la capçalera amb els noms de les columnes. El body el té buit i al peu té una imatge de la icona que permet afegir files i es defineix quin és el mètode que es crida al fer clic en ella.

Existeix la funció:

```
function afegeixFilaFitxa(quantitat, tipus, color, tamany) {
    //Implementació
}
```

Aquesta funció, crea una nova fila en la taula. Per fer això utilitza JQuerymobile. La manera de fer-ho és la següent:

```
$('#taulaFitxes tbody').append(fila);
```

On fila és el codi HTML que defineix una fila, taulaFitxes és l'identificador de la taula i tbody és un element HTML de la taula.

La funció es pot cridar des de dos llocs diferents. El primer d'ells és quan es fa clic al botó afegir fila de la taula. En aquest cas es crea la fila nova amb els valors per defecte.

El segon lloc des d'on es pot cridar la funció és al carregar un joc existent. En aquest cas se li passen com a paràmetres els valors que tenim guardats a la base de dades.

En aquest cas, quan es carrega, no es disposa d'una relació de fitxes amb files de la taula directa. És a dir, en una fila de la taula hi poden haver varies fitxes definides, totes elles iguals. Ha calgut implementar un sistema que identifiqui les fitxes iguals per adjuntar-les totes en una mateixa fila abans de crear-la.

Per tal que es carreguin els estils de Jquerymobile, cal fer la següent crida sempre que modifiquem la taula:

```
$('#taulaFitxes').trigger("create");
```

Si no es fa aquesta crida, les files afegides no tenen cap estil aplicat.

Un altra funció, en aquest cas una de molt senzilla però que és interessant, és la que es fa servir per eliminar una fila.

```
function removeRow(td) {  
    $(td).parent().remove();  
}
```

La ultima columna de cada fila, sempre és la que conté la icona de esborrar fila.

En la cel·la es defineix que al fer clic es cridi la funció removeRow(td) de la següent manera:

```
<td onclick="removeRow(this);">
```

Això passa la cel·la seleccionada al mètode, i dins d'aquest obtenim l'element pare (és a dir, la fila) i l'eliminem.

Amb aquests mètodes es poden afegir i esborrar files i configurar el joc com es vulgui. El que cal fer ara, es convertir les dades de la taula en entitats Fila i Dau i construir l'objecte Joc quan es premi el botó guardar.



Cal doncs recórrer les taules d'alguna manera i anar extraient les dades de dins dels diferents controls que hi ha en cada cel·la d'aquesta.

El codi que es fa servir per recórrer les dades i construir els objectes és el següent:

```
$("#taulaFitxes tbody tr").each(function(index) {  
    var camp0, camp1, camp2, camp3;  
    $(this).children("td").each(function(index2) {  
        switch (index2) {  
            case 0:  
                camp0 = $(this).find('input').val();  
                break;  
            case 1:  
                camp1 = $(this).find('select').val();  
                break;  
            case 2:  
                camp2 = $(this).find('select').val();  
                break;  
            case 3:  
                camp3 = $(this).find('input').val();  
                break;  
        }  
    })  
  
    for ( var i = 0; i < camp0; i++) {  
        joc.addFitxa(new Fitxa(null, null, camp1, camp2, camp3, '0', '0'));  
    }  
});
```

El que està fent el codi es, per cada fila de dins del body de la taula i per cada columna dins d'aquesta fila, obtenir el valor de l'element que hi ha dins (en aquest cas tenim input per els text box i select per els combos). Aquests valors es guarden en les variables camp0, camp1... per cada fila. És llavors, per cada fila, que s'executa el bucle que veiem a baix, on es mira quantes fitxes s'han definit en la fila actual, i es creen tantes noves fitxes amb els valors obtinguts com s'han definit. Llavors continua amb següent fila.

## Carregar

Amb la informació que tenim fins ara d'aquesta pantalla, ja ens podem fer una idea de com es fa per convertir una entitat Joc, amb les seves fitxes i daus al que finalment es veu per pantalla.

El que es fa primer de tot, és una crida al mètode “*getJocById(id, callback)*” de la llibreria de base de dades que s'ha implementat.

Es defineix el mètode de callback en el fitxer on ens trobem. Aquest mètode és el següent:

```
function carregaJoc(joc) {  
    // implementació  
}
```

El paràmetre és l'entitat Joc. Dins d'aquest mètode es van obtenint els diferents elements de la pantalla. El nom del joc, les instruccions, la imatge, i es carreguen en els seus controls.

Després es fa el calcul que s'ha comentat en l'apartat anterior per agrupar les fitxes i els daus iguals en files i es crida, per cada fila el mètode explicat també en l'apartat anterior que és el que afegeix les files a la taula.

## Guardar

Per guardar s'utilitza la funció següent:

```
function guardarJocAmbImatge(){  
    // implementació  
}
```

El que fa és construir un objecte Joc, amb tots els valors que hi ha en els elements de la pantalla. A part dels visibles tenim un valor ocult on es guarda l'id del joc, de manera que sabem quan estem fent l'actualització d'un joc i quan estem fent la inserció de un nou.

S'utilitza després el mètode que recorre les taules i crea els objectes Fitxa i els objectes Dau, i els afegeix al objecte Joc creat.

Finalment es crida la funció que tenim en la llibreria de base de dades “setJoc(joc, callback)” i aquesta guarda el joc i ens retorna l'id inserit o actualitzat.

## Jugar a un joc

Per realitzar el desenvolupament de la pantalla de joc, s'ha utilitzat la llibreria gràfica de desenvolupament en Javascript i que treballa sobre un element canvas de HTML5 comentada anteriorment Kineticjs.

S'utilitzen les llibreries de Phonegap i la djtdatabase.js creada per nosaltres mateixos per encapsular tot el que te a veure amb base de dades.

En aquesta pantalla no s'utilitza Jquerymobile.

La definició de la pàgina en HTML es molt senzilla. Nomes existeix un div que disposa de id.

```
<body>
```

```
<div id="container"></div>
</body>
```

A mes a mes s'ha definit un estil per aquest div.

```
<style type="text/css">
#container {
  overflow: hidden;
  width: 100%;
  height: 100%;
  background-image: url('img/woodtexture.jpg');
  background-repeat: no-repeat;
  background-size: 100%;
}
</style>
```

El que s'ha fet amb aquest estil és posar la imatge de la taula de fusta en el background i s'ha definit que el tamany del div és tota la pantalla.

Tota la resta d'elements que es veuen es construeixen amb Javascript utilitzant la llibreria gràfica de Kineticjs.

Kineticjs ens ha permès treballar en capes. S'ha aprofitat aquesta característica per definir 3 capes. La primera el taulell, la segona les fitxes i la tercera els daus. D'aquesta manera no tenim cap mena d'interferència entre cadascun d'aquests elements.

Una segona característica de Kineticjs que s'ha aprofitat és el treballar amb shapes (formes). Gràcies a aquesta llibreria es poden definir formes i el comportament d'aquestes d'una manera molt neta i amb poca dificultat.

També ens ha proporcionat la possibilitat de treballar amb transicions de tota mena, per exemple de moviment, de transparència o de tamany dels elements. S'ha intentat aprofitar aquestes característiques al màxim per que la sensació dels elements sigui el més real possible.

Finalment una característica que s'ha utilitzat de Kineticjs son els sprites. Totes les llibreries gràfiques disposen d'aquest element. Es tracta de la utilització d'una imatge on hi ha totes les posicions possibles d'un objecte (en el nostre cas, totes les cares d'un dau). El que es fa és carregar un sol cop en memòria la imatge, i mitjançant codi mostrar la zona de la imatge que es correspon a l'estat actual. Al tenir la imatge carregada, el canvi entre les àrees que mostrem és immediat i dóna sensació de moviment.

## Carregar joc

Un cop més, al carregar la pàgina es crida el mètode de la llibreria djtdatabase.js que ens retorna un objecte Joc a partir del seu id.

En la pàgina s'ha definit una funció la qual se li passa de callback a aquest mètode. La funció és la següent:

```
function carregarJocComplet(joc) {  
    //Implementacio  
}
```

Dins d'aquesta funció, es carrega tot el joc de la manera següent:

## Carregar joc. El taulell

La imatge del taulell la es te en l'objecte Joc. Sempre s'ajusta la imatge depenent de la seva alçada, és a dir, s'ajusta la imatge per que l'alçada sigui la mateixa que la de la pantalla. Aquest factor de escalat es doncs l'alçada total que tenim, dividit entre l'alçada de la imatge.

Es multipliquem doncs per aquest valor tant l'amplada (width ) com l'alçada (height) de la imatge, i així s'aconsegueix que mida d'aquesta sigui la que encaixa en la nostra pantalla i que manté l'aspecte original.

Es crea un element específic de Kinetiics per imatges i es fa que es posicioni aquesta ajustada a la dreta. Per fer-ho cal trobar el punt inicial, que és el que es correspon amb la longitud total de la pantalla menys la longitud total de la imatge.

Aquest element és pintat en una capa de Kinetiics i aquesta capa és afegida a la escena.

## Carregar joc. Les fitxes

A continuació es carreguen les fitxes. Per fer això s'ha fet un bucle que recorre totes les fitxes i les va creant. Es disposa de tota la informació guardada en la entitat Fitxa de manera que només cal construir la fitxa segons aquesta informació.

A continuació es mostra com és la definició d'una fitxa rodona:

```
shape = new Kinetic.Circle({  
    x : fitxa.getPosx(),  
    y : fitxa.getPosy(),  
    radius : fitxa.getTamany(),  
    fill : fitxa.getColor(),  
    stroke : getColorStroke(fitxa.getColor()),  
    opacity : 0,  
    strokeWidth : 4,  
    draggable : true,  
    shadow : {  
        color : 'black',  
        blur : 10,
```

```
        offset : [ 5, 5 ],  
        opacity : 0.6  
    },  
    scale : {  
        x : 1,  
        y : 1  
    }  
});
```

Aquesta fitxa rodona ja es podria arrossegar per la pantalla. Això s'ha fet amb l'atribut “**draggable : true**”. El color i el tamany s'agafen de l'objecte fitxa, així com el tipus. Les fitxes tenen un *stroke*, que és el voltant d'aquestes, de un color similar al de la fitxa però amb menys intensitat. A més a més també tenen ombra. Aquests dos elements junts aporten realitat ja que dóna més la sensació que la fitxa està al damunt del taulell.

Amb “*opacity : 0*” s'ha definit que la fitxa és transparent. Totes les fitxes es creen transparents i s'aplica una transició quan ja estan totes carregades per que apareguin sobre el taulell. Aquesta característica, a part de quedar bonica, s'ha fet per tal de solucionar un possible error de la llibreria en dispositius mòbils. El comportament incorrecte de l'error és que allà on es pinta per primer cop un objecte, no es refresca mai la escena, de manera que encara que moguem la fitxa de lloc, sembla com si en quedés una fixa a sota sempre.

En aplicar aquesta transició, això ja no ocorre, o si més no, al ser transparent la fitxa no es veu.

Per tal de millorar la experiència al utilitzar el joc, s'han creat uns efectes sobre les fitxes al arrossegar-les. Aquests efectes son que en tocar-la, aquesta augmenta de tamany i en deixar-la anar es torna a fer petita amb un efecte elàstic. El codi que fa això és el següent:

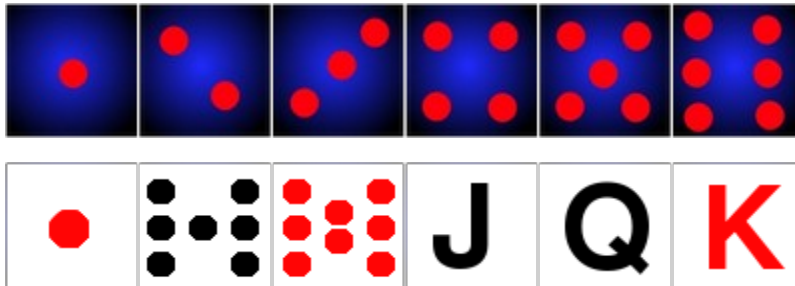
```
shape.on('dragstart', function() {  
    if (trans) {  
        trans.stop();  
    }  
  
    shape.moveToTop();  
    shape.setAttrs({  
        scale : {  
            x : 1.2,  
            y : 1.2  
        }  
    });  
});  
  
shape.on('dragend', function() {  
    trans = shape.transitionTo({  
        duration : 0.5,  
        easing : 'elastic-ease-out',  
        scale : {
```

```
        x : 1,  
        y : 1  
    }  
  })  
});
```

On shape és una fitxa.

## Carregar joc. Els daus

Quan es carreguen els daus es necessiten crear uns objectes sprites per cadascun d'ells. S'han implementat dos possibles daus, les imatges per els sprites dels quals son les següents:



Aquestes dos imatges es carreguen 1 sol cop en memòria en el cas que hi hagi daus. Amb aquesta mateixa imatge podem crear tants sprites com sigui necessari.

Per tal de definir els sprites amb Kineticjs, s'ha de definir un array de animacions. Nomes tenim una animació que és la de tirar el dau. Dins de la animació es defineix un nou array de objectes on cadascun d'ells defineix un estat del sprite, és a dir, una àrea de la imatge. Per tots dos daus tenim la següent animació:

```
var animations = {  
  tirar : [ {  
    x : 0,  
    y : 0,  
    width : 67,  
    height : 67
```

```
    }, {  
      x : 67,  
      y : 0,  
      width : 67,  
      height : 67  
    }, {  
      x : 133,  
      y : 0,  
      width : 67,  
      height : 67  
    }, {  
      x : 200,  
      y : 0,  
      width : 67,  
      height : 67  
    }, {  
      x : 267,  
      y : 0,  
      width : 67,  
      height : 67  
    }, {  
      x : 333,  
      y : 0,  
      width : 67,  
      height : 67  
    } ]  
  } ]  
};
```

El que es defineix en la animació, com es pot veure, és el punt inicial i final de cada cara del dau.

A continuació es mostrarà com crear l'objecte sprite que tindrà la imatge del dau i la animació per tirar el dau que hem definit.

El codi per fer això és el següent:

```
var dau = new Kinetic.Sprite({  
  x : ix,  
  y : iy,  
  image : imatgeDau,  
  animation : 'tirar',  
  animations : animations,  
  frameRate : 5  
});
```

On ix, i iy son la posició on volem posar el dau.

ImatgeDau és l'objecte Image on tenim una de les dos imatges del dau.

El quart element, 'tirar' és l'animació que es vol que es faci de totes les que tenim definides en la array

de animacions, que li passem en el cinquè element.

Finalment el `frameRate` : 5 és la velocitat en que canviem de imatge.

Per poder tirar el dau, finalment ens cal engegar l'animació quan toquem el dau, i fer que aquesta s'aturi en un moment aleatori.

Per fer això el que s'ha utilitzat és el següent codi:

```
dau.on('touchstart', function() {  
    dau.setAnimation('tirar');  
    dau.start();  
  
    dau.afterFrame(Math.floor((Math.random() * 6)), function() {  
        dau.stop();  
    });  
});
```

El que fa aquest codi és definir el comportament d'un dau quan el toquem. Primer de tot s'estableix que l'animació d'executar és la de llençar (no en tenim cap mes) i s'inicia l'animació.

Se li ha definit que s'executi la funció que atura aquesta animació després de un frame que es correspon a un valor aleatori entre 1 i 6.

D'aquesta manera s'ha aconseguit l'efecte de que estem llançant un dau.

## ***Principals problemes durant la implementació***

Durant la implementació de l'aplicació s'han produït diferents factors o dificultats que ens han fet consumir a vegades mes temps del desitjat per realitzar algunes parts del desenvolupament.

Aquests problemes han estat:

### **Transaccions asíncrones**

El primer problema amb que ens hem trobat, ha estat amb els mètodes asíncrons de Phonegap per realitzar determinades operacions. En concret, el que ens ha afectat és el fet que no es poden executar transaccions síncrones a la base de dades.

En un principi no érem coneixedors d'això, de manera que en el disseny inicial s'havia pensat i implementat un sistema que ha hagut de canviar per adaptar-se a aquesta limitació. D'haver sabut que això era així, s'hauria realitzat el disseny acord a aquest fet, però el que ha passat és que ens en hem adonat al moment de fer la implementació.

El principal problema que ha portat això és al moment de guardar un joc. El problema es troba en que cal guardar x fitxes i y daus. Fins que el joc no està guardat no es pot saber quin és l'identificador



d'aquest de manera que no es pot seguir guardant. En aquest cas ha calgut moure dins dels callbacks de les transaccions les següents sentències a executar, enlloc de fer una darrera l'altre.

## Jquerymobile, recarregues de pantalla

El segon problema ha estat a causa de Jquerymobile, o mes aviat a que ha estat molt difícil treballar amb agilitat utilitzant aquesta tecnologia. Un dels problemes ha estat el repintat de les pantalles, que s'executa més vegades que les desitjades. Això comportava que en fer servir el boto de “enrere” que tenen els dispositius Android, es recarregués un altre cop la mateixa pantalla enlloc de anar enrere.

Aquest problema ha estat resolt a tot arreu. Hi ha un lloc però que la manera de solucionar-ho no ha estat la que s'hauria volgut en un principi. Aquest lloc és al escollir la imatge del taulell. Inicialment s'ha fet que en fer clic al damunt de la imatge, s'obris un popup on es pogués escollir si es vol la imatge des de fitxer o des de la càmera. L'acció d'obrir el popup produïa que es recarregués la pagina i no s'ha aconseguit que això no passi. És un punt critic ja que en un procediment normal, es donaria el següent cas:

Clic imatge -> Menú popup -> Opció càmera -> Fer foto i acceptar -> Pantalla manteniment amb el taulell carregat -> Guardar -> Botó enrere (per anar a la llista de jocs) -> **Pantalla del taulell sense imatge ni dades.**

La sensació que es dona és que no s'ha guardat, tot i que en realitat sí que ho ha fet.

El que s'ha fet per solucionar-ho és posar 2 botons que no recarreguen la pagina directament a la pantalla, un per la càmera i l'altre per la galeria.

Pel que s'ha pogut descobrir, hi ha elements que recarreguen la pantalla depenent de on es produeixin. Per exemple, un element “<input type=“button”>” dins d'un formulari, recarrega la pantalla. En canvi un element “<button type=“button”>” no ho fa.

No s'ha pogut trobar el perquè succeeix això en el cas del popup, de manera que s'ha optat per la opció que explicada.

## Jquerymobile, visualització de inputs en formularis

El següent problema, relacionat amb Jquerymobile també, ha estat la creació dels apartats fitxes i daus. Aquests apartats tenen controls numèrics, de colors i de llistes desplegable. A part de la dificultat de afegir els controls dinàmicament, s'ha trobat que en un navegador es veu d'una manera, en un dispositiu concret d'una altra i en un altre dispositiu d'una tercera diferent.

Mentre en un navegador de ordinador, concretament amb el Chrome per Linux es veuen tots els controls d'una manera correcta, en els dispositius mòbils no.

Per començar no surten, en els elements numèrics, els botons per puja i baixar el valor sense fer servir el teclat. Aquest no és un problema, ja que podem introduir el valor amb el teclat.

El que pitjor funciona és l'element de seleccionar colors. De fet aquest no funciona gens. Si ho

visualitzem en el navegador de l'ordinador, el control ens permet seleccionar qualsevol color des de una paleta on apareixen tots, però des del mòbil, només podem escriure el valor hexadecimal del color. S'ha començat a implementar amb canvas un sistema semblant al que surt en els navegadors, però degut a falta de temps s'ha deixat aparcat i s'ha optat per posar un llistat desplegable amb els principals colors per escollir.

## **Rendiment en la pantalla de selecció de joc**

Un altre problema ha estat el rendiment de la llista. Ha passat que en carregar les imatges directament en la llista, en haver-hi 5 o 6 es ralentitzava el dispositiu degut a que consumia massa memòria. Per tal de solvatar això ha calgut modificar l'schema de la base de dades i afegir una columna més amb el thumbnail del taulell. A part ha calgut crear un mètode per escalar la imatge capturada i fer-ne una de petita.

En aquest cas estem davant d'un error de disseny que podríem haver previst.

## **Rendiment en la pantalla de jugar amb un joc**

En la pantalla de jugar, es pot veure que el rendiment en determinats dispositius és pitjor del que es voldria. Fent varies proves, sembla que el rendiment té a veure amb la resolució de la pantalla i molt probablement amb el motor del navegador que s'utilitza.

El que s'ha pogut veure també, és que en el navegador del mòbil, en concret amb el Chrome per Android, els mateixos exemples amb canvas funcionen força millor que amb Phonegap. No s'ha pogut descobrir quina és la causa d'això. La diferència no és exagerada, però sí és perceptible.

No obstant, no podem esperar que el rendiment sigui igual que en una aplicació nativa.

## **Depuració**

Una limitació que ha fet perdre molt temps ha estat el no poder (o almenys no s'ha aconseguit) depurar l'aplicació en el dispositiu. Degut a que es necessiten les llibreries de Phonegap i les taules SQLite, la depuració en el navegador de l'ordinador no ha servit. El que s'ha fet és utilitzar *alerts* en el codi o la consola de logs del dispositiu per veure que és el que es va produint en cada moment. És un sistema molt incommode, però no ha estat possible trobar una alternativa millor.

## **Problemes coneguts**

### **Llançament del dau**

Quan tenim més d'un dau, al llançar -lo, sempre es llança l'últim que s'ha llançat abans. Sembla que es queda una referència creuada amb el dau anterior. S'ha revisat el codi i no s'ha aconseguit solucionar. Possiblement és un error de la llibreria Kineticjs.

### **Posicionament moltes fitxes o molts daus**

Si es posen moltes fitxes o molts daus, es probable que acabin sortint de la pantalla i aquests no es puguin utilitzar.

### **Escalat imatge thumbnail**

Així com en la part de jugar, les imatges surten ben escalades mantenint l'aspecte, en el thumbnail es perd la proporció i es queda la imatge sempre quadrada.

### **Àrees en la pantalla de manteniment de joc**

Al construir-se la pantalla dinàmicament, no queda ben dimensionada i cal fer scroll en aquesta, quan podria evitar-se.

## **Diari de desenvolupament (Implementació)**

En arribar a la implementació començava la part més divertida del projecte. Fins aquest moment tot havia estat llegir documentació fer algun exemple bàsic utilitzant la tecnologia que faríem servir, i pensar i redactar els anàlisis i planificacions.

Es va començar a treballar doncs, primerament amb la llibreria que tracta tot l'accés a dades. Com ja he comentat, es va fer tota la implementació de la llibreria sense saber que els mètodes que estàvem utilitzant de Phonegap eren asíncrons.

Quan es va descobrir que això passava es van adaptar tots els mètodes implementats per treballar amb callbacks.

Hi ha hagut molts moments de poca productivitat, sobretot treballant amb Jquerymobile. La navegació entre pantalles, el posicionament dels elements, les dimensions d'aquests o el comportament que tenen, etc. han estat molt difícils de manejar. Semblava que per més que es definís com es volia que fos cada element, Jquerymobile no funcionés com s'esperava.

La major part de tots aquests moments han estat sobretot fruit de la inexperiència. Mes avançada la implementació, s'han vist coses de Jquerymobile molt pràctiques i que ens han permès fer molt en poques línies de codi.

La part més interessant d'implementar ha estat la pantalla de jugar.

De la primera versió feta, que era molt senzilla, a la última que s'ha aconseguit finalment, els detalls han millorat molt. Primer es va desenvolupar una pantalla de joc molt bàsica, la qual era força senzilla i no donava sensació de joc real. Finalment però el resultat sí que dona sensació de realitat.

## Conclusions

La realització del projecte final de carrera ha estat una experiència molt positiva. S'ha descobert una tecnologia que està pràcticament acabada de néixer.

El fet de que el projecte tracti de una idea pròpia, ha fet que la motivació per fer-ho sigui molt alta. Estar motivat ha ajudat a persistir davant els problemes que s'han tingut durant el desenvolupament de tota l'aplicació. En un principi no s'esperava que fossin tants.

Des de la planificació a la implementació, passant per la definició funcional, tècnica etc. s'ha intentat posar el màxim de cura en tot el que s'ha fet, no deixant mai res sense explicar o a mitges sense motiu.

Una sensació que queda, és el saber com hauria estat el desenvolupament de l'aplicació en Android natiu enlloc de Phonegap. Aquesta curiositat no seria tant gran si el rendiment de l'aplicació fos optim, però ja hem vist que sembla que podria ser millor en alguns aspectes.

S'han vist força aplicacions fetes en Phonegap i Jquerymobile, i totes elles semblen força menys fluides que una aplicació nativa. Sobretot s'ha vist que els events que responen a accions tàctils, no responen de la mateixa manera que en les aplicacions natives.

En un principi es pensava que el temps del que es disposava per fer el projecte donaria per mes. Gràcies al tutor que va posar especial atenció en aquest punt, i en tot moment va recomanar que es definís molt bé el que havia d'entrar i el que no, i que es tingués en compte el temps que es disposa. De no haver estat així es podria haver donat el cas de haver-nos perdut en el desenvolupament degut a que es volien fer mes coses de les que es podien.

També cal remarcar la flexibilitat amb la que s'ha pogut treballar, on s'ha definit entre el tutor i alumne les fites, tant amb dates com amb contingut, de manera conjunta i això ha ajudat molt a portar el projecte al dia i a saber en tot moment quin era l'estat i possibles desviacions.

Pel que fa al dia a dia, degut al punt anterior i al esforç fet, no hi han hagut desviacions en la planificació. Tot ha anat desenvolupant-se sobre les dates previstes. Hi ha potser algun punt del desenvolupament que podria haver quedat mes be, o si mes no hauria estat positiu que quedés mes be, però en termes generals s'han assolit els objectius.

Un altre aspecte a destacar, és el temps que cal dedicar a fer totes les tasques que no son la implementació. Des del primer moment on s'escull el projecte fins el moment en que s'acabi de fer la presentació, hi ha hagut molta feina en planificar, definir, aprendre, etc. que en un principi no s'és conscient que sigui tant rellevant. Tota aquesta feina però, ha estat molt ben feta i que és un dels motius principals d'èxit del projecte.

Finalment i pel que fa al producte final, la sensació és que ha quedat un producte vistós, que te un bon funcionament i que permet jugar a molts jocs diferents amb les eines que proporciona.

## Glossari de termes

Terme	Definició
Android	Sistema operatiu basat en Linux desenvolupat per Google.
Android SDK	Kit de desenvolupament de aplicacions Android.
Canvas	Element HTML5 que permet la generació de gràfics dinàmicament
Cas d'ús	Descripció dels passos a seguir per dur a terme un procés.
CSS3	Llenguatge utilitzat per definir presentacions de documents HTML o XML.
DIA	Aplicació per crear diagrames de tipus.
Eclipse	Aplicació per programar en diversos llenguatges.
Entitat	En el model entitat relació, és la representació d'un objecte del món real que té existència independent.
Framework	Estructura de suport per utilitzar en el desenvolupament d'aplicacions.
HTML5	Hypertext Markup Language: Llenguatge per definir pàgines web.
InkScape	Editor de gràfics.
iOS	Sistema operatiu mòbil de la empresa Apple Inc.
Javascript	Llenguatge de programació interpretat.
Jquerymobile	Framework per desenvolupar aplicacions per mòbils
Kineticjs	Framework Javascript per desenvolupar aplicacions damunt d'un element canvas de HTML5
Libreoffice	Suite ofimàtica lliure i gratuïta
Multi-plataforma	Aplicació que pot funcionar en diferents plataformes, com ara Android, IOS, Windows etc.
Open Source	Terme amb el que es coneix el programari distribuït i desenvolupat lliurement.
Phonegap	Framework per crear aplicacions mòbils usant HTML.
Planner	Eina per planificar, programar i seguir projectes.
SQLite	Sistema de gestió de base de dades relacional.

Sprite	En videojocs son utilitzats per crear animacions a partir de una imatge amb totes les posicions possibles de la animació.
Thumbnail	Versió reduïda de una imatge per previsualitzar.
Ubuntu	Sistema operatiu Linux
Widows phone	Sistema operatiu mòbil desenvolupat per Microsoft.
Zoom	Apropar o allunyar la imatge per enfocar una zona concreta

## Bibliografia

Pilgrim, Mark. HTML5: Up and Running. First Edition. United States of America: O'Really Media inc. August 2010. 208 pag.

Phonegap. Apache Cordova Documentation: Getting Started Guides [en línia]. The Apache Software Foundation, 2012, versió 2.2.0 [Consulta: Octubre-Desembre de 2012] Disponible en web: <[http://docs.phonegap.com/en/2.2.0/guide\\_getting-started\\_index.md.html](http://docs.phonegap.com/en/2.2.0/guide_getting-started_index.md.html)>

jQueryMobile. JQuery mobile demos [en línia]. JQuery Foundation and other contributors, 2012, versió 1.2.0 Final [Consulta: Octubre-Desembre de 2012] Disponible en web: <<http://jquerymobile.com/demos/1.2.0/>>

KineticJS i canvas. Html5CanvasTutorials [en línia]. Eric Rowell 2011 – 2013, versió 4.2.0 [Consulta: Octubre-Desembre de 2012] Disponible en web: <<http://www.html5canvastutorials.com/kineticjs/html5-canvas-events-tutorials-introduction-with-kineticjs/>>

jQueryMobile. Manual de jQueryMobile [en línia]. Desarrolloweb.com, 2012, versió 1.0 [consulta: Octubre-Desembre de 2012] Disponible en web: <<http://www.desarrolloweb.com/manuales/manual-jquery-mobile.html>>