



Universitat  
Oberta  
de Catalunya

## *Aplicació MultipliCat*

**Francesc Roig i Feliu**

Grau d'Enginyeria Informàtica

Desenvolupament aplicacions dispositius mòbils (iOS)

**Jordi Ceballos Villach**  
**Carles Garrigues Olivella**

*14 de juny de 2017*



Aquesta obra està subjecta a una llicència de:

[Reconeixement-NoComercial-CompartirIgual](https://creativecommons.org/licenses/by-nc-sa/3.0/)

[3.0 Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Aplicació MultipliCat</i>
<b>Nom de l'autor:</b>	<i>Francesc Roig i Feliu</i>
<b>Nom del consultor/a:</b>	<i>Jordi Ceballos Villach</i>
<b>Nom del PRA:</b>	<i>Carles Garrigues Olivella</i>
<b>Data de lliurament:</b>	<i>14/06/2017</i>
<b>Titulació o programa:</b>	<i>Grau en Enginyeria Informàtica</i>
<b>Àrea del Treball Final:</b>	<i>Desenvolupament aplicacions dispositius mòbils (iOS)</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>taules, escola, primària</i>

### Resum del Treball:

En aquesta memòria es descriu la implementació d'una aplicació per a la plataforma iOS (iPhone, iPad) que té com objectiu facilitar l'aprenentatge de les taules de multiplicar per a nens de primària.

L'aplicació utilitza un enfoc totalment nou pel que fa a la forma de motivar a l'alumne a estudiar les taules. A diferència d'altres aplicacions que podem trobar a l'*App Store* on l'única motivació pel nen a seguir «jugant» amb l'aplicació és l'atractiu visual que aquesta pugui tenir, a *MultipliCat* la motivació és un premi pactat amb el pares o educadors un cop aconseguit un determinat nombre de punts. L'aplicació en aquest cas simplement actua com a eina d'aprenentatge formulant les preguntes i comptabilitzant els punts, però deixa la motivació de l'alumne a un objectiu extern, el premi pactat.

S'ha de dir que a diferència de la resta d'aplicacions del mercat on l'alumne perd ràpidament l'interès pel «joc», i consegüentment la motivació per aprendre les taules, amb *MultipliCat* la motivació es manté molt més, accelerant el procés d'aprenentatge de les taules de forma significativa.

Com a conclusió i en base a les proves fetes, puc afirmar que amb aquest mètode s'obtenen uns resultats molt satisfactoris on la resta de sistemes més tradicionals esdevenen poc eficaços per a determinats nens, sobretot els més moguts i amb poca paciència.

### Abstract:

This report describes the implementation of an application for iOS platform (iPhone, iPad) which aims to facilitate the learning of times tables for primary school children.

The application uses an entirely new approach regarding how to motivate students to study these tables. Unlike other applications that can be found in the App Store where the only motivation for the child to continue "playing" with the application is the visual appeal that may have, in *MultipliCat* the motivation is the achievement of a prize previously agreed with parents or educators (the prize is achieved at an agreed number of points). The application in this case simply acts as a learning tool by asking questions and counting points, but leaves the motivation of students to an external Award previously agreed.

It must be said that unlike other applications of the market where the student quickly loses interest in the "game", and therefore the motivation to learn tables, with *MultipliCat* motivation is accelerated.

In conclusion and based on some tests done, I can say that this method allows to obtain very satisfactory results where other more traditional systems become ineffective for some children, especially those most restless and with little patience.



## Índex de continguts

1 Pla de treball.....	9
1.1 Context i Justificació del treball.....	9
1.2 Objectius del treball.....	10
1.3 Enfocament del mètode seguit.....	10
1.4 Planificació del treball.....	11
1.4.1 Recursos necessaris.....	11
1.4.2 Tasques a realitzar.....	11
1.5 Productes obtinguts.....	13
1.6 Breu descripció dels altres capítols de la memòria.....	13
2 Eines de treball.....	13
2.1 Software de desenvolupament <i>Xcode</i> .....	13
2.2 Software de suport.....	14
2.3 Components gràfics, classes i controladors utilitzats.....	14
3 Usuari i context d'ús.....	15
3.1 Mètode d'indagació.....	15
3.1.1 L'alumne (el nen); procés d'indagació.....	15
3.1.2 Els pares i educadors; procés.....	16
3.2 Perfils d'usuari identificats.....	16
3.2.1 Característiques del perfil 'alumne' (el nen).....	16
3.2.2 Característiques del perfil 'pares i educadors'.....	16
3.2.3 Context d'ús.....	16
3.2.4 Anàlisi de tasques.....	17
3.2.5 Característiques de l'aplicació.....	17
4 Disseny conceptual.....	18
4.1 Escenaris d'ús.....	18
4.1.1 Multiplicació directa.....	18
4.1.2 Multiplicació inversa.....	18
4.2 Fluxos d'interacció.....	18
5 Sketches inicials i prototipatge horitzontal.....	20
5.1 Sketches inicials.....	20
5.2 Prototipus horitzontal.....	22
6 Implementació de l'aplicació.....	26
6.1 Arquitectura de l'aplicació.....	26
6.1.1 ViewController.....	26
6.1.2 ViewControllerTaula.....	27
6.1.3 ViewControllerGrafic.....	28
6.1.4 ViewControllerPass.....	29
6.1.5 ViewControllerAjustos.....	30
6.1.6 UINavigationController.....	31
6.1.7 AppDelegate.....	32
6.2 Canvis respecte el prototipus de partida.....	33
6.3 Altres aspectes relatius a la implementació.....	34
7 Avaluació del projecte.....	35
7.1 Obtenció d'informació sobre els usuaris (pares i educadors).....	35
7.2 Avaluació del producte (l'aplicació).....	35
7.3 Preguntes referent a les tasques.....	36
7.3.1 Preguntes als pares i educadors (usuaris tipus 1).....	36
7.3.2 Preguntes als nens (usuaris tipus 2).....	36
8 Conjunt de proves.....	37
8.1 Metodologia de les proves.....	37
8.2 Proves de funcionals de «joc».....	37
8.2.1 «Joc» directe.....	41
8.2.2 «Joc» invers.....	41

8.3 Proves de configuració.....	42
8.3.1 Reset d'estadístiques.....	42
8.3.2 Selecció de taules de «joc».....	45
8.3.3 Ajust del temps màxim de resposta.....	45
8.3.4 Inicialització del comptador de punts.....	45
8.3.5 Inicialització del comptador a punts objectiu.....	45
8.3.6 Canvi de contrasenya.....	46
8.3.7 Avisos acústics.....	46
8.3.8 Proves amb usuari real.....	47
8.3.9 Proves addicionals multi dispositiu.....	47
8.4 Provar l'aplicació en Xcode.....	48
9 Valoració econòmica.....	49
10 Conclusions.....	50
11 Referències bibliogràfiques.....	51
12 Annex I. Llistats de les classes.....	52
12.1 ViewController.....	52
12.2 ViewControllerTaula.....	54
12.3 ViewControllerPass.....	60
12.4 ViewControllerAjustos.....	62
12.5 ViewControllerGrafic.....	69
12.6 AppDelegate.....	71

# Agraïments

*Vull agrair a la meva família el suport que m'ha donat tot  
el temps que ha durat aquest projecte i,  
molt especialment, al meu fill que ha estat un dels principals  
col·laboradors amb el munt de proves que li he fet fer.*





# 1 Pla de treball

## 1.1 Context i Justificació del treball

L'aprenentatge de les *taules de multiplicar* és una cosa per la que tots hi hem passat en un moment o altre de la nostra vida. Per tal de facilitar aquest aprenentatge i en funció del moment, s'han utilitzat uns mètodes o uns altres. En el moment actual, com no podia ser cap altra manera, les apps o aplicacions s'han convertit en un dels mètodes més escollits tant per a pares com per a educadors com a facilitador d'aquest aprenentatge.

Aquest treball de final de grau aborda el desenvolupament d'una aplicació per iOS que facilita l'aprenentatge de les taules de multiplicar per a nens de primària.

La idea d'abordar aquest tema en el treball de final de grau, neix a partir de l'experiència viscuda amb el meu fill, quan va haver d'aprendre les taules de multiplicar. Com a persona vinculada al món de les tecnologies de la informació, tant pel moment tecnològic que viu la nostra societat com també pel fet que el tema m'apassiona, vaig començar a buscar solucions d'aquest tipus que poguessin ajudar al meu fill en aquest procés. Com era d'esperar, en vaig trobar un munt. He de dir, però, que com que a casa únicament tenim dispositius Apple, la cerca d'aplicacions va ser únicament en aquesta plataforma.

Totes les que vaig poder avaluar basen l'èxit de l'aprenentatge en motivar la pràctica de les taules a base de captar l'atenció del nen a amb colors llampancs i gràfics vistosos. El problema amb el que em vaig trobar amb el meu fill, és que per molts colors llampancs i gràfics vistosos que tinguessin les aplicacions, no aconseguien retenir ni l'atenció ni la motivació del nen per seguir jugant. Al cap de 5 o 10 minuts de jugar-hi, el nen ja n'estava cansat. Per tant vaig intuir que, o bé aquestes aplicacions no funcionaven per a tots els nens, o bé alguna cosa estava fallant en el seu plantejament.

Després de provar diferents maneres de plantejar l'aprenentatge de les taules com un joc, vaig arribar a la conclusió que una cosa que per a ell no era percebut com un «joc» (o com a mínim un «joc» atractiu per a ell) mai aconseguiria captar l'atenció del nen.

En el cas del meu fill, i d'algun altre nen de la seva classe, el que ens va funcionar millor als pares va ser incentivar l'aprenentatge amb petits premis. Premis que el nen aconseguia si, per exemple, era capaç de respondre bé 10 multiplicacions seguides que se li plantegessin.

Amb aquesta estratègia vàrem poder mantenir la motivació del nen al llarg de tot el procés d'aprenentatge de les taules.

A partir d'aquesta experiència, he pensat que estaria bé crear una aplicació de taules de multiplicar que, amb aquesta estratègia d'aprenentatge, faciliti plantejar i controlar les fites que el nen ha d'assolir (les taules que ha de memoritzar) per aconseguir el premi pactat.

Actualment, de les aproximadament 120 aplicacions disponibles en el App store d'Apple, no n'hi ha cap que abordi l'aprenentatge de les taules de multiplicar de la forma que s'aborda en aquest projecte. És per això que el desenvolupament d'aquesta aplicació està plenament justificat pel fet que dona resposta a unes necessitats concretes que la resta d'aplicacions disponible no cobreixen.

Per altra banda, la major part de les aplicacions d'aquest tipus disponibles en el App store són en anglès. També se'n pot trobar algunes en espanyol, però cap en català. És per això que un segon objectiu d'aquest projecte és contribuir a expandir l'ús del català en aquest àmbit, el de les aplicacions, que encara és força pobre.

## 1.2 Objectius del treball

Els objectius del TFG són:

1. Proporcionar una solució tecnològica en forma d'aplicació per a iOS que doni suport a l'alumne en el procés d'aprenentatge de les taules de multiplicar utilitzant un enfoc diferent de la resta d'aplicacions iOS del mercat.
2. Aportar una nova eina al món de l'educació primària gratuïta, en forma de programari lliure, i amb unes funcionalitats diferents de les que es poden trobar actualment en el *App store*.
3. Expandir l'ús del català en el món de les aplicacions.

L'aplicació estarà disponible per la plataforma iOS, concretament per les famílies de dispositius iPhone (iPhone 5, ES, 6, 6 Plus, 7 i 7Plus) i iPad (iPad mini 2, mini 4, Air 2, Pro).

No es preveu per aquest projecte que l'aplicació estigui disponible per altres plataformes. En qualsevol cas, com que es tracta d'un projecte sota la llicència *open source*, el projecte pot ser continuat i desenvolupat per altres plataformes.

L'aplicació es desenvoluparà en català per complir amb un dels seus objectius ja esmentats.

## 1.3 Enfocament del mètode seguit

Es tracta d'una aplicació que en el seu desenvolupament no depèn de cap tecnologia especial, per tant, es desenvolupa a partir de les llibreries estàndards que proporciona Xcode.

Tampoc es preveu cap tipus de connectivitat amb l'exterior. L'aplicació treballa de forma exclusivament local. Al tractar-se d'una aplicació d'ús infantil, el fet que no s'envii ni es rebi res de l'exterior garanteix un nivell de confidencialitat més alt, cosa que la pot fer més atractiva en aquest sentit tant pels pares com pels educadors. Per altra banda, això permet que l'aplicació pugui ser utilitzada off-line sense problemes.

A diferència d'altres aplicacions d'aquest tipus, on la part gràfica esdevé clau a l'hora motivar al nen, en aquest cas no cal. És per això que es prescindeix d'una interface gràfica excessivament decorada a favor de més claredat i simplicitat que en faciliti l'ús tant pels nens com pels pares o educadors. Tot i això, s'ha buscat un aspecte gràfic amb una paleta de colors agradable que faciliti la concentració dels nens.

Pel que fa a la forma d'abordar el desenvolupament del projecte, es fa a partir d'una metodologia àgil que permet una gestió més dinàmica del projecte.

## 1.4 Planificació del treball

### 1.4.1 Recursos necessaris

El desenvolupament de l'aplicació no requereix de recursos especials. Amb un ordinador MAC i el software de desenvolupament que Apple proporciona, i que és gratuït, ja n'hi ha prou per abordar aquest projecte. Els requisits són:

- un ordinador MAC amb sistema operatiu OS X 10.11.5 o posterior,
- i el software de desenvolupament *Xcode* versió 8.2 o posterior (recomanat).

En el cas que l'aplicació es vulgui distribuir en el App store d'Apple, caldrà adquirir una llicència de desenvolupador que costa 99 € anuals.

### 1.4.2 Tasques a realitzar

Les tasques a realitzar s'emmarquen en cinc grans blocs o tasques principal. Els tres primers coincideixen amb les tasques previstes per a complir els objectius previstos en cadascuna de les tres PACs, mentre que les dues últimes ja s'emmarquen sobretot en la preparació i lliurament dels lliurables del projecte (memòria i presentació).

Un cop presentada i aprovada la proposta del projecte, les tasques principals a realitzar són les següents:

- **PAC 1.** Aquest primer bloc consisteix en definir alguns aspectes del projecte com són el context i justificació del treball, objectius del treball, i l'enfoc o la planificació. En aquest bloc també es preveu començar la redacció de la memòria, lliurant com a PAC 1 el primer capítol on es tracten els aspectes abans esmentats.

També cal crear i lliurar una aplicació de prova tipus «Hola món» com a primera presa de contacte amb la plataforma de desenvolupament Xcode.

Id	Tasca	Durada	Inici
1	<b>PAC 1</b>		22/02/17
2	Preparació document memòria	1	
3	Elaborar PAC 1	5	
4	Preparar «Hola món»	1	
5	Lliurar PAC 1	1	

- **PAC 2.** Aquest segon bloc aborda sobretot els aspectes que tenen a veure amb el disseny de les funcionalitats, el disseny general de les pantalles de l'aplicació, i d'altres aspectes no vinculats directament en la codificació de les funcionalitats.

Id	Tasca	Durada	Inici	
6	<b>PAC 2</b>		09/03/17	
7	Establir les funcionalitats bàsiques	4		
8	Disseny de la dinàmica de «joc»	3		
9	Definir les possibles configuracions	4		
10	Disseny iconogràfic general	4		
11	Selecció de la paleta de colors	2		
12	Disseny conceptual de les pantalles de «joc»	2		
13	Disseny conceptual de les pantalles de configuració	2		
14	Disseny de la navegació entre pantalles	1		
15	Establir objectius de «joc» predefinitos	2		
16	Disseny de la pantalla de benvinguda	3		
17	Disseny de les pantalles de joc	3		
18	Disseny de les pantalles de configuració	3		
19	Redacció de la PAC 2	4		
20	Lliurar PAC 2	1		

- PAC 3.** En aquest bloc ja s'entra de ple en el desenvolupament del producte en sí, és a dir, l'aplicació. Aquí es realitzen tasques com el disseny de l'aplicació amb components gràfics real, el disseny i implementació de la lògica de control, així com tot el conjunt de proves i refinaments necessaris que calgui per a obtenir al producte definitiu segons les especificacions requerides. Al final d'aquesta bloc ja tindrem la versió definitiva de l'aplicació.

Id	Tasca	Durada	Inici	
21	<b>PAC 3</b>		06/04/17	
22	Programació de les pantalles de «joc»	8		
23	Proves de les pantalles de «joc» amb usuaris test	2		
24	Anàlisi de les proves i refinament del disseny	1		
25	Programació de les pantalles de configuració	8		
26	Proves de les pantalles de configuració amb usuaris test	2		
27	Anàlisi de les proves i primer refinament del disseny	1		
28	Disseny de la icona de la aplicació	2		
29	Creació de la primera beta l'aplicació	1		
30	Proves de la primera beta amb usuaris test	4		
31	Anàlisi de les proves i segon refinament del disseny (si cal)	4		
32	Integració de canvis i creació de la segona beta l'aplicació	4		
33	Proves de la segona beta amb usuaris test	2		
34	Anàlisi de les proves i segon refinament del disseny (si cal)	2		
35	Integració de canvis i creació de versió definitiva	2		
36	Creació de la versió definitiva (v 1.0)	1		
37	Redacció de la PAC 3	4		
38	Lliurar PAC 3	1		

- Redacció de la memòria definitiva.** Aquest bloc està centrat exclusivament en la memòria del projecte. Aquí s'aborda la integració dels diferents apartats que conformen el document final, així com les revisions que calgui per obtenir el document lliurable amb tota la documentació.

Id	Tasca	Durada	Inici	
39	<b>REDACCIÓ DE LA MEMÒRIA DEFINITIVA</b>		18/05/17	
40	Integració dels diferents apartats de la memòria	8		
41	Redacció de l'abstract en català i anglès	4		
42	Primera revisió de la memòria	2		
43	Segona revisió de la memòria	2		

- Elaboració de la presentació i lliurament.** Aquí es realitzen les tasques vinculades amb la presentació virtual del TFG. Des de la preparació dels continguts fins a la gravació. En acabar es lliurarà el projecte amb tot el que es demana.

Id	Tasca	Durada	Inici	
44	<b>ELABORACIÓ DE LA PRESENTACIÓ i LLIURAMENT</b>		01/06/17	
45	Disseny de la presentació	3		
46	Preparació i proves de la presentació	4		
47	Gravació del vídeo	4		
48	Muntatge del vídeo	2		
49	Revisió de la presentació	1		
50	Lliurar projecte i presentació	1		

El total d'hores estimat per aquest projecte és de 126h. La dedicació setmanal serà d'aproximadament **2 hores diàries, incloent el cap de setmana**. En funció de la setmana, la distribució d'aquest temps pot variar lleugerament.

## 1.5 Productes obtinguts

Com a resultat de la realització del projecte s'obtidrà:

- una aplicació per a la plataforma iOS (es lliurarà la carpeta del projecte),
- una memòria del projecte en forma de document escrit
- i una presentació del projecte en format vídeo.

## 1.6 Breu descripció dels altres capítols de la memòria

En la resta dels capítols de la memòria es tracten tots els aspectes relacionats amb el desenvolupament del programari. S'aborden els aspectes de disseny de l'aplicació a nivell gràfic, d'usabilitat, del control per part dels pares o educadors, i de com fer seguiment del progrés del nen en l'aprenentatge de les taules. També s'aborda l'ús de l'aplicació, tant en l'entorn de la llar com en els centres educatius.

En els següents capítols s'aborden en detall els aspectes de dissenys, des de la interface gràfica fins als aspectes més programàtics.

Es tracta en detall cadascuna de les tasques portades a terme en la planificació del projecte (apartat 1.4.2), justificant les decisions tècniques o de disseny que s'hagin pres al llarg del desenvolupament.

També s'incorpora una guia d'utilització de l'aplicació que permeti treure'n el màxim profit a l'usuari.

Finalment, es detallen les conclusions a les que s'arriba un cop finalitzat el projecte.

# 2 Eines de treball

## 2.1 Software de desenvolupament Xcode

Es tracta del software de desenvolupament estàndard que proporciona Apple per a desenvolupar aplicacions destinades, sobretot, als seus productes. Xcode és un entorn de desenvolupament que, si bé es pot utilitzar com editor per a múltiples llenguatges de programació, està especialment dissenyat per a programar en *Objective C* i *Swift*. En el cas de desenvolupar d'aplicacions per a dispositius mòbils basats en iOS (iPhones, iPad, iPod, etcètera), únicament es poden utilitzar aquests dos llenguatges. En aquest sentit, val la pena

remarcant que, si bé *Objective C* ha estat durant molts anys l'únic llenguatge disponible per aquest tipus de desenvolupaments, el 2014 es va presentar el nou llenguatge de programació *Swift* (desenvolupat per Apple) amb la intenció de facilitar la programació i evitar la barrera que suposa pels nous programadors un llenguatge tan complicat com l'*Objective C*.

Aquest projecte s'ha desenvolupat amb Xcode 8.3 i *Swift 3*.

## 2.2 Software de suport

Adicionalment s'ha utilitzat el software *GIMP v2.8* i *Sketch v43.2* pel disseny gràfic de l'aplicació.

## 2.3 Components gràfics, classes i controladors utilitzats

Els components gràfics utilitzats són els estàndards que proporciona Xcode pel desenvolupament d'aplicacions per a iOS. Concretament s'han utilitzat les següents classes i controladors:

**UIButton.** Es tracta del component gràfic estàndard tipus botó i s'ha utilitzat en diverses pantalles de l'aplicació.

**UILabel.** En aquest cas també es tracta d'un altre component gràfic estàndard àmpliament utilitzat en el desenvolupament d'aplicacions. És un component passiu que la seva funció principal és mostrar textos informatius dins les pantalles de les aplicacions.

**UISegmentedControl.** Es tracta d'un component que simula un conjunt de botons agrupats, el quals actuen a mode de commutador.

**UISwitch.** Es tracta d'un component que simula un interruptor on/off. En l'aplicació s'utilitza per a activar o desactivar funcionalitats en la configuració d'aquesta.

**UISlider.** És un component que permet seleccionar un valor numèric a partir del desplaçament d'un element gràfic per la pantalla. En aquest cas s'ha utilitzat per definir els *time-outs* de resposta de l'aplicació.

**UITextField.** Es tracta d'un component similar a una etiqueta (UILabel) però, a diferència d'aquest, el component permet al usuari editar-ne el contingut. En l'aplicació s'utilitza, per exemple, per a canviar la contrasenya d'accés al menú de configuració.

**Stack View.** Si bé no es tracta d'un element com els anteriors que tenen una clara representació gràfica dins l'aplicació, es un element que la seva funció és la d'organitzar la representació d'altres components gràfics. Aquest component permet crear agrupacions verticals i horitzontals de components cosa que facilita la representació d'aquests dins l'aplicació. També permet agrupar altres Stack View creant estructures més complexes.

A nivell de controladors, s'ha usat el **UIViewController** i el **UINavigationController**. El primer és el que s'utilitza com a suport bàsic pels diferents components que formen les pantalles de l'aplicació, mentre que el segon és el responsable de la navegació entre pantalles.

## 3 Usuari i context d'ús

### 3.1 Mètode d'indagació

Abans de justificar el mètode d'indagació, vaig a posar una mica en context l'«usuari» del l'aplicació. L'usuari en aquest cas serà, per un costat l'alumne (el nen) com principal objectiu de l'aplicació, i per l'altre els pares o professors com a parametritzadors d'aquesta.

Tot i que podem identificar dos usuaris distints, no els podem tractar com a perfils diferents d'usuaris que utilitzen l'aplicació de la mateixa manera. Per exemple, en el cas d'un joc, tot i podent haver-hi perfils d'usuari diferent, tots comparteixen un mateix objectiu amb l'aplicació, jugar. En aquest cas però, l'objectiu és ben diferent. Els pares o educadors interactuaran amb l'aplicació per a configurar els exercicis, mentre que els nens simplement «jugaran»<sup>1</sup>.

A partir d'aquesta premissa, s'han triat mètodes d'indagació diferents per a cadascun d'ells. El mètode escollit pels nens és el de *Shadowing*, mentre que pels pares o educadors s'ha escollit el de les *entrevistes*.

#### 3.1.1 L'alumne (el nen); procés d'indagació

Tal com ja s'ha apuntat anteriorment, el mètode escollit per obtenir informació de l'usuari «nen» ha estat *Shadowing*.

L'objectiu ha és trobar la forma més eficient, des del punt de vista d'aprenentatge, de participació de l'alumne en el «joc». En aquest sentit, s'ha buscat: (1) la forma més senzilla i intuïtiva pel nen d'interactuar amb l'aplicació, (2) un disseny gràfic agradable i que faciliti la interacció i la concentració del nen en l'aplicació, i (3) una dinàmica de joc àgil que no cansi a l'usuari (en aquest cas el nen).

Sobre la forma d'interactuar amb l'aplicació (1), es van provar dos mètodes en forma d'exercicis escrits. El primer consistia en què el nen havia de contestar una sèrie de preguntes del tipus  $6 \times 4 = ?$  on ell escrivia el que considerava que era la resposta. I el segon consistia en què el nen simplement escollia la resposta correcta d'entre 10 possibles respostes. De les dues opcions, la que va funcionar millor va ser la segona, ja que un mateix període de temps, el nen podia respondre més preguntes.

Descartada la primera opció, es va fer una tercera prova. Consistia en utilitzar com possibles respostes el conjunt de valors possibles de la taula preguntada. Per exemple, si la pregunta és  $4 \times 5 = ?$ , les possibles respostes són: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40. Amb aquesta tècnica, el nen anava més guiat i li era més senzill intuir la resposta correcta. En aquest cas l'aprenentatge va ser més ràpid, i per tant, més eficient.

Paral·lelament, d'acord amb algunes converses mantingudes amb professors d'aquesta edat, el mètode de triar la resposta en lloc d'escriure el resultat correcte, resulta més eficient degut a què en aquesta edat es treballa molt la memòria visual.

---

1 Es fa servir el terme «joc» per indicar l'ús de l'aplicació en mode aprenentatge, és a dir, quan l'utilitza el nen.

### 3.1.2 Els pares i educadors; procés

En aquest cas, el mètode escollit és el de cercar informació a través d'entrevistes i preguntes tant a pares (aquí també m'hi incloc) com a educadors. Tots coincidim que l'aplicació ha de ser una eina senzilla d'utilitzar, no només pels nens, sinó també per a nosaltres.

L'aplicació ha de permetre programar els exercicis que ha de fer el nen d'una forma ràpida i senzilla. En aquest sentit, tenim clar que l'eina d'aprenentatge (l'aplicació) té un objectiu primordial, facilitar la pràctica de les taules de multiplicar, per tant, el que esperem com a pares o educadors és que l'eina sigui prou flexible per adaptar-se a les necessitats de cada alumne. Per exemple, pot haver-hi nens que siguin capaços de practicar diverses taules alhora, mentre que d'altres els va millor d'una en una; l'aplicació ha de ser prou flexible per adaptar-se a cada cas.

Pel que fa als aspectes gràfics i de dissenys, tots coincidim que el més important és la simplicitat i, sobretot, que sigui intuïtiva d'utilitzar.

## 3.2 Perfils d'usuari identificats

### 3.2.1 Característiques del perfil 'alumne' (el nen)

Es tracta sobretot de nens de 7 a 9 anys, però tret de la franja d'edat, no hi ha cap tret especial important en aquest perfil. No obstant, aquesta aplicació pot esdevenir especialment interessant en casos on les tècniques habituals de motivació a l'estudi de les taules no donin bons resultats.

També pot esdevenir una eina interessant en els casos que el nen necessiti una eina pro-activa que l'ajudi a l'estudi de les taules de multiplicar. A diferència del que seria un mètode «tradicional», la utilització d'aquesta aplicació permet marcar un ritme en el procés d'aprenentatge que, d'altre manera, dependria només de propi alumne.

### 3.2.2 Característiques del perfil 'pares i educadors'

El fet que, com a mínim inicialment, s'hagi previst desenvolupar l'aplicació únicament per a la plataforma iOS, orienta l'aplicació cap un mercat d'usuaris d'un perfil econòmic sobretot mitjà i alt. Aquest fet pot semblar discriminatori respecte la plataforma Android, amb molts més usuaris, però això no treu que la distribució d'usuaris sigui igual a tots els centres. Si bé en una gran majoria de centres la plataforma Android serà la més estesa, també hi ha centres on passa tot el contrari (com és el cas de l'escola del meu fill). És per això que un desenvolupament per aquesta plataforma queda totalment justificat.

Més enllà d'aquestes consideracions prèvies, el perfil dels pares o educadors entraria en el que és ja habitual en la nostra societat, persones acostumades a l'ús de les tecnologies de la informació que veuen com la utilització d'aquesta tecnologia en el procés educatiu dels seus fills pot ser altament beneficiós.

Pel que fa als mestres o educadors, podríem establir un perfil similar.

### 3.2.3 Context d'ús

Es tracta d'una aplicació pensada sobretot per a ser utilitzada tant en un iPhone com en un iPad. Amb això no es descarta que en un futur no es pugui portar a altres dispositius iOS, però, a hores d'ara, aquesta possibilitat



no es té en compte. El fet de poder ser instal·lada en telèfons intel·ligents, permet que pugui ser utilitzada en una gran varietat de llocs; per exemple, quan els pares acompanyen el nen cap a l'escola en cotxe, aquest la pot utilitzar durant el trajecte en el iPhone del seus pares.

Si ens traslладem en l'àmbit de l'escola, l'aplicació podria ser utilitzada durant la classe mitjançant iPads del centre, com una eina per a fer exercicis dins l'aula.

Des del punt de vista d'entorn tecnològic, al tractar-se d'una aplicació que no necessita connectivitat a Internet per a ser utilitzada, l'entorn d'utilització queda completament obert.

Si ens fixem en l'entorn físic, l'aplicació no requereix de cap condició especial (llum, soroll, etcètera), en tot cas, haurà de ser un entorn mínimament adequat perquè el nen es pugui concentrar en respondre les preguntes. En aquest sentit, s'ha de dir que com que les preguntes que fa l'aplicació no estan immerses en cap context previ, són preguntes ràpides ( $7 \times 8 = ?$ ), i el fet que hi pugui haver petites distraccions no distorsiona el procés d'aprenentatge; recordem que els nen es distreuen fàcilment.

### 3.2.4 Anàlisi de tasques

Des del punt de vista de l'alumne, l'única tasca que haurà de realitzar és simplement anar contestant les preguntes en forma de multiplicacions simples del tipus  $7 \times 8 = ?$  o  $7 \times ? = 56$  que l'aplicació li planteja. Amb les respostes anirà acumulant punts que li permetran assolir l'objectiu de punts pactat amb el pares o educadors. Si encerta la resposta sumarà 1 punt, i si la falla en restarà 8. Amb aquesta estratègia anirà practicant i memoritzant les taules sense adonar-se'n, amb la motivació d'assolir l'objectiu de punts que li permetrà, per exemple, aconseguir algun tipus de premi pactat.

Des de l'altra basant, la dels pares o educadors, la tasca que hauran de dur a terme serà la configuració de l'exercici. Caldrà definir alguns paràmetres com quines taules es practicaràn o el mode de «joc» (multiplicació directa o indirecta).

### 3.2.5 Característiques de l'aplicació

L'aplicació caldrà que disposi de dues àrees diferents, una per a la sessions de joc del nen, i una altre de configuració. Tots sabem que els nens se les empesquen totes per aconseguir els seus objectius, i en aquest sentit, si deixéssim l'accés a l'àrea de configuració obert, no dubteu que el primer que farien seria modificar la configuració del joc per a facilitar aconseguir els punts pactats. És per això que cal protegir l'accés a l'àrea privada amb un sistema de contrasenya.

També és interessant que hi hagi algun sistema gràfic que permeti fer un seguiment de l'aprenentatge de les diferents taules. En aquest sentit, es disposarà d'una gràfic que mostrarà a quines taules els nens tenen més problemes, cosa que permetrà als pares o educadors planificar millor les sessions de «joc» i fer el procés d'aprenentatge més eficient.

## 4 Disseny conceptual

### 4.1 Escenaris d'ús

L'aplicació no contempla que l'escenari d'ús variï en funció del context on s'estigui utilitzant. Independentment del context on es trobi l'usuari (en aquest cas l'alumne), l'escenari en concret no variarà.

Es contemplen dos escenaris d'ús diferents en funció de l'objectiu concret que es vulgui aconseguir amb l'ús de l'aplicació. Per un costat (1) tenim la pràctica de multiplicacions directes, que bàsicament respon a l'aprenentatge directe de les taules ( $7 \times 8 = ?$ ), i per l'altre (2) tenim la pràctica inversa ( $7 \times ? = 56$ ). L'objectiu de la pràctica inversa és consolidar l'aprenentatge de les taules i prepara a l'alumne per a les divisions.

S'ha de dir que ambdós escenaris d'ús l'objectiu pel nen és el mateix, aconseguir els punts pactats.

#### 4.1.1 Multiplicació directa

En aquest escenari l'alumne (el nen) aprèn les taules de multiplicar de forma directa ( $7 \times 8 = ?$ ). Simplement ha de triar quina és la resposta correcta de les opcions que se li plantegen.

L'objectiu pels pares o educadors és, òbviament, que el nen arribi a memoritzar les taules de multiplicar. Mentre que pel nen l'objectiu és arribar als punts que li permetin aconseguir el premi pactat. Quan encerta una resposta, l'aplicació li marcarà l'encert i li suma un punt, mentre que si falla la resposta, l'aplicació li marca quina és la correcta (amb la finalitat que el nen l'apregui per quan li torni a sortir) i el penalitza traient-li 8 punts. La puntuació mai pot ser inferior a 0.

Per a començar a aprendre una taula no li cal cap tipus d'informació addicional. És a partir de les equivocacions que el nen aprèn les respostes.

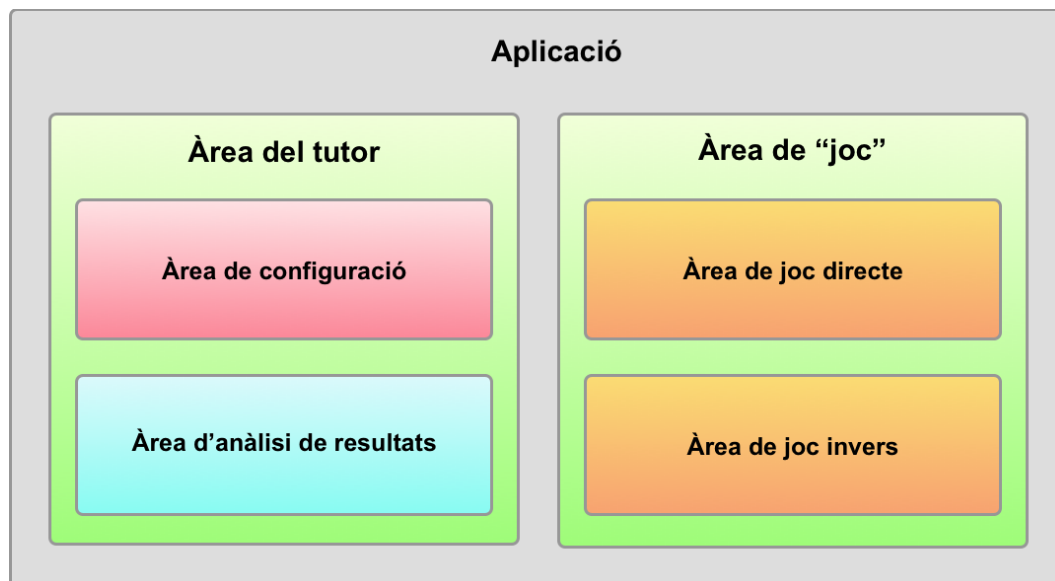
#### 4.1.2 Multiplicació inversa

Aquest és el segon escenari en el que ens podem trobar. En aquest cas, el nen ja ha pogut memoritzar les taules però li cal aprofundir una mica més en el concepte de cara a preparar les divisions.

En aquest cas tot funciona pràcticament igual, tret que ja no es pregunta, per exemple, *quan és 5 per 4*, sinó que se li pregunta *quin número multiplicat per 5 dóna 20*.

## 4.2 Fluxos d'interacció

En el gràfics següent es mostra l'estructura de l'aplicació:



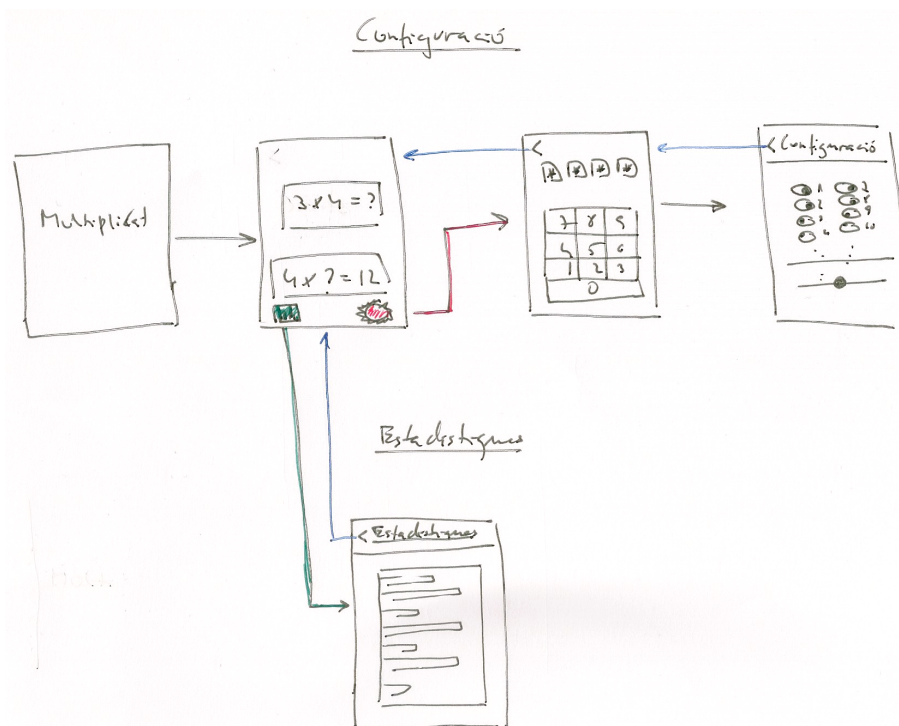
Podem observar les dues àrees principal, (1) l'àrea del tutor (pares o educadors) i (2) l'àrea de joc. Aquestes àrees limiten la interacció de l'usuari amb l'aplicació. Mentre que l'àrea del tutor està reservada als pares o educadors. L'àrea on interactuaran els nens serà únicament l'àrea de «joc».

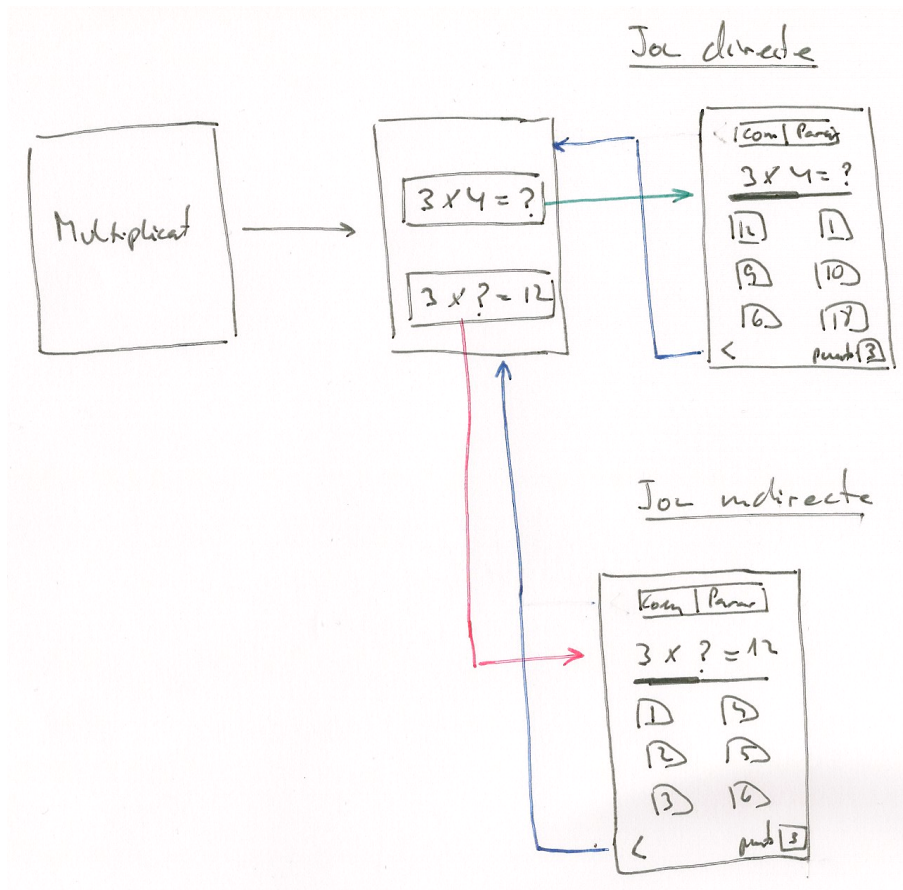
Dins l'àrea o zona del tutor hi trobem la part de configuració, on els pares o educadors poden personalitzar la sessió de joc. També hi ha una àrea de resultats on es poden veure les estadístiques d'ús.

En la zona de «joc» hi podem trobar dues subàrees que mostren les dues funcionalitats de «joc» que té l'aplicació, la directe i la inversa.

## 5 Sketches inicials i prototipatge horitzontal

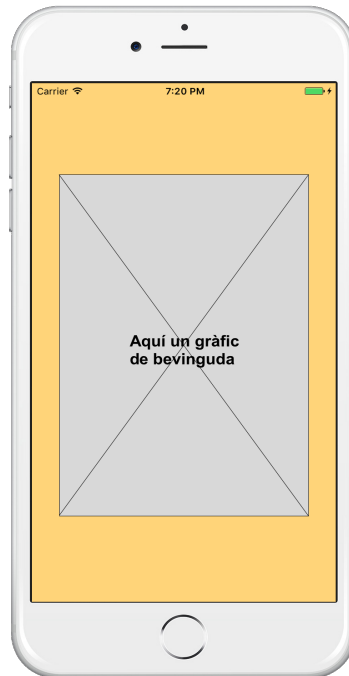
### 5.1 Sketches inicials



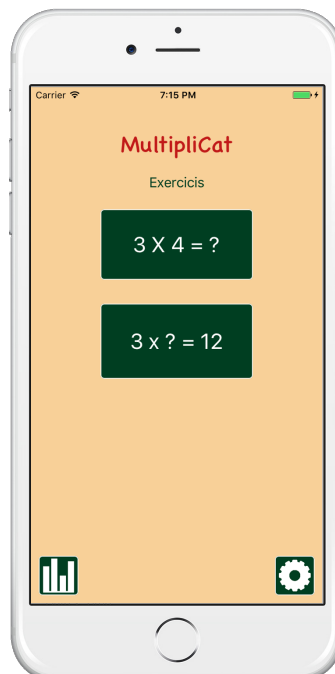


## 5.2 Prototipus horitzontal

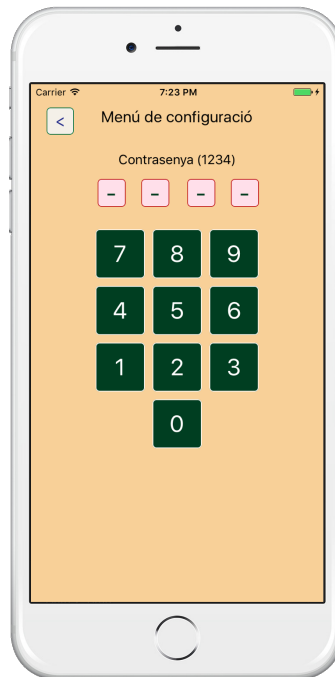
- Pantalla d'inici d'aplicació:



- Pantalla de selecció de l'exercici:



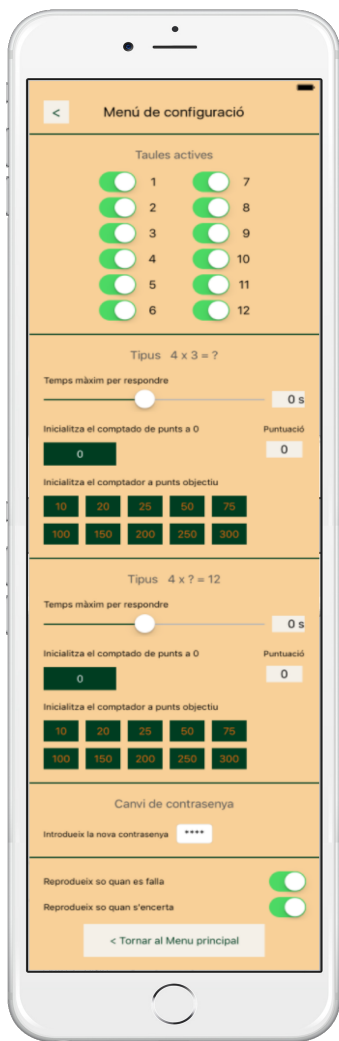
- Pantalla d'accés al menú de configuració:



- Pantalla del gràfic seguiment de l'aprenentatge:

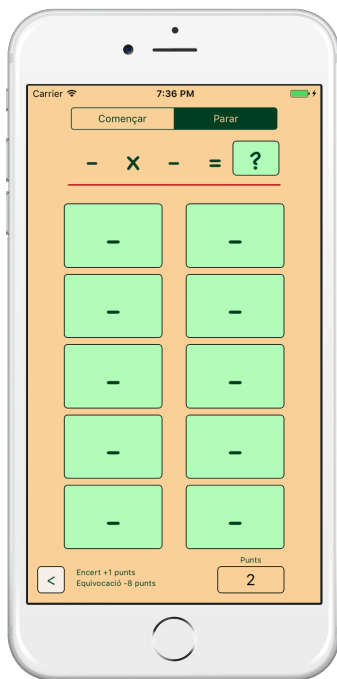


- Pantalla de configuració (desplaçament vertical):

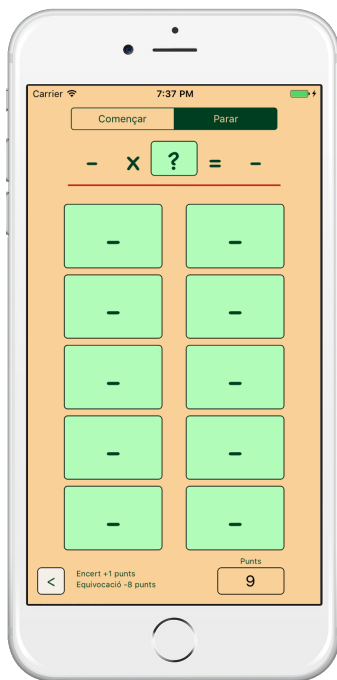




- Pantalla de «joc» mode directe:



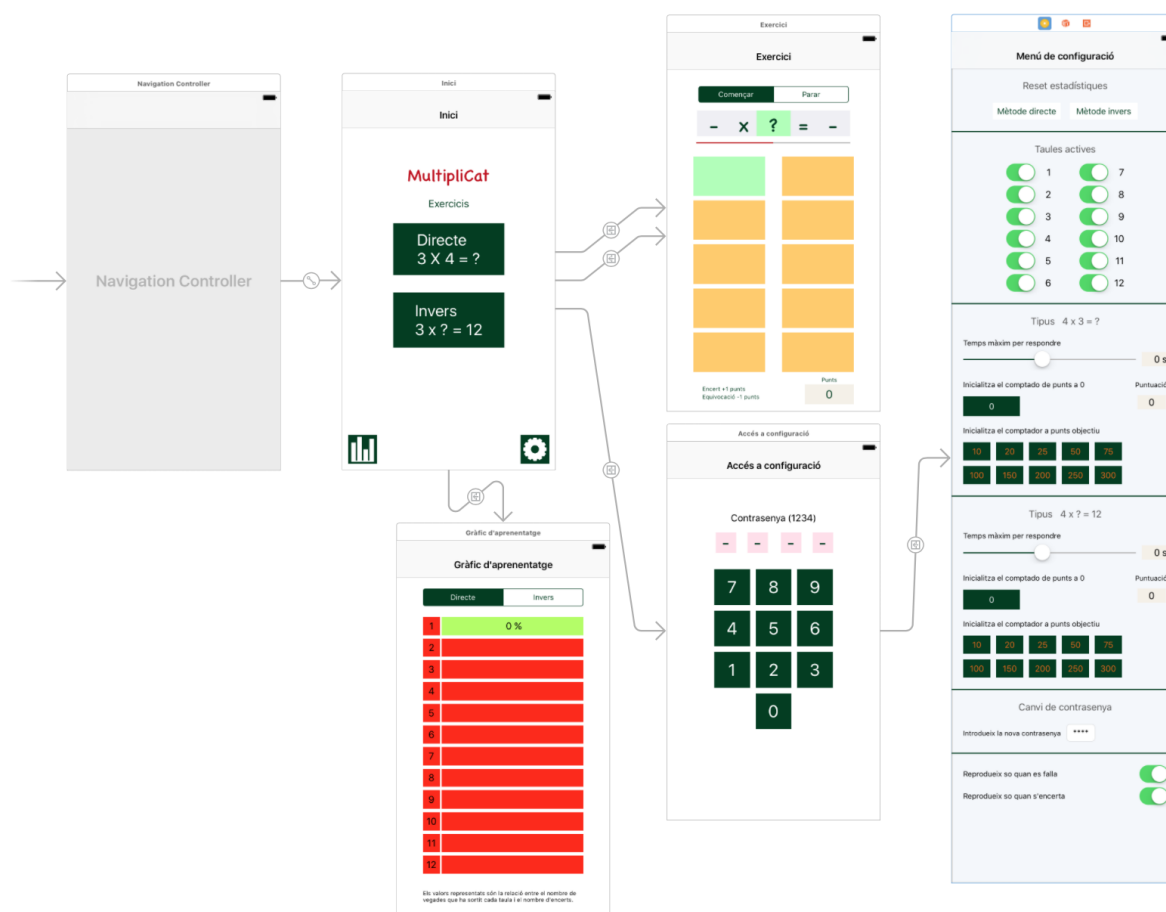
- Pantalla de «joc» mode indirecte o invers:



## 6 Implementació de l'aplicació

### 6.1 Arquitectura de l'aplicació

L'arquitectura de l'aplicació és bàsicament plana i segueix exactament l'estructura de pantalles. A continuació podem veure l'estructura de l'aplicació a partir del *Storyboard* de la pròpia aplicació:



L'aplicació està formada principalment per 4 classes, que es corresponen amb els controladors de les vistes de les pantalles. A continuació s'analitzen les seves funcionalitats.

**Nota.** A l'*Annex I* d'aquesta memòria es poden trobar els llistats corresponents a les classes del projecte. El codi està auto comentat per tal de facilitar la comprensió.

#### 6.1.1 ViewController

Aquesta és la classe per defecte que crea Xcode quan seleccionem el patró d'aplicació *Single View Application*. Es tracta d'un controlador de vistes del tipus *UIViewController* associat a la pantalla d'aplicació inicial, la qual també es crea per defecte en iniciar un nou projecte amb aquest patró.

Aquesta classe està associada a la pantalla «Inici» que apareix quan s'inicia l'aplicació (just després de la pantalla de llançament). És la pantalla central des d'on s'inicia tota la navegació a la resta de pantalles.

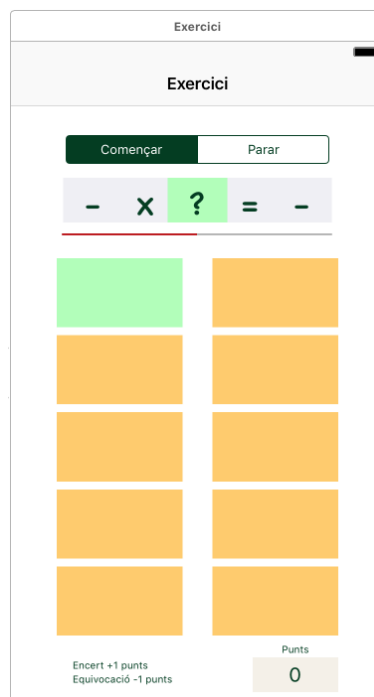


Aquesta classe, a banda dels components gràfics que hi podem trobar, s'utilitza també per a definir algunes constants globals a nivell de tota l'aplicació com, per exemple, els colors d'alguns components gràfics o la seva tipografia de lletra. També s'utilitza per a inicialitzar les principals variables de funcionament de l'aplicació a partir de la informació desada en el propi dispositiu (recordem que tota la informació de l'aplicació es desa en local). La càrrega d'aquesta informació es produeix en el mètode *viewDidLoad*, mètode que és invocat després que el controlador de la vista es carrega a memòria.

El llistat d'aquesta classe es troba en l'Annex I, apartat 12.1 d'aquesta memòria.

### 6.1.2 ViewControllerTaula

Es tracta també d'una classe del tipus *UIViewController* com el cas anterior, que en aquest cas està associada a la pantalla «Exercici». Vindria a ser la classe principal de l'aplicació i és des d'on es controla tot el procés de «joc» d'aquesta. Ambdós tipus d'exercicis (Directe i Invers) es realitzen des d'aquesta pantalla. També és la pantalla des d'on podem anar a les pantalles de configuració i d'estadístiques.

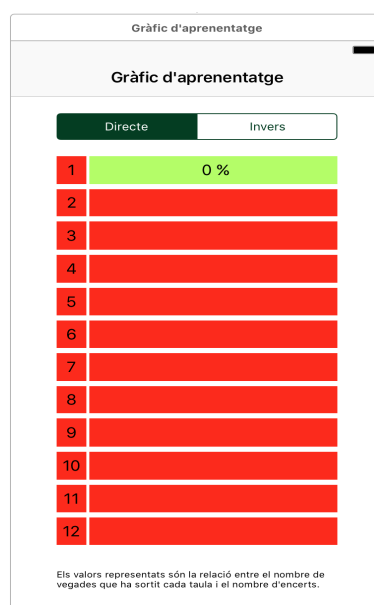


A banda de gestionar els components gràfics, el controlador implementa les classes que gestionen l'execució de l'exercici, és a dir, tota la lògica del «joc».

El llistat d'aquesta classe es troba en l'Annex I, apartat 12.2 d'aquesta memòria.

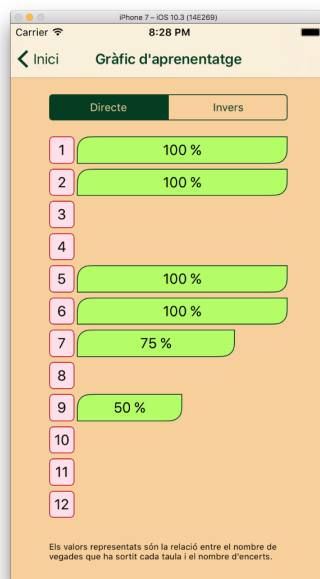
### 6.1.3 ViewControllerGrafic

Al igual que en els casos anteriors, aquí també es tracta d'una classe tipus *UIViewController*. La classe està associada a la pantalla que mostra les estadístiques de «joc».



La responsabilitat d'aquesta classe és gestionar la representació gràfica dels components gràfics utilitzats per a representar l'estadística, en aquest cas, components del tipus *UILabel*.

Un aspecte interessant d'aquesta classe és que s'ha modificat el component *UILabel* (que s'utilitza per a generar les barres de l'estadística) per tal de donar-li un aspecte més atractiu tal com es pot apreciar a la figura següent:

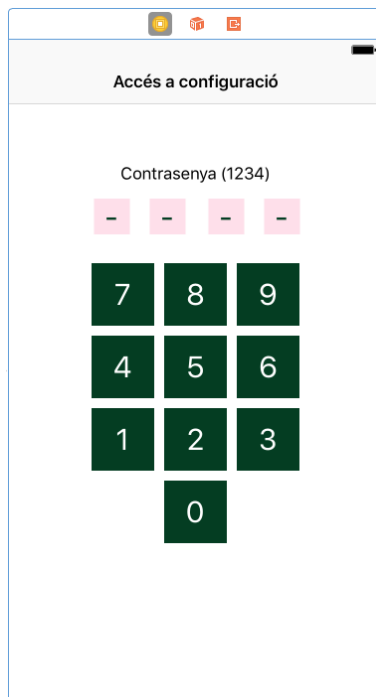


Aquesta classe també s'encarrega de gestionar l'actualització i el control de l'estadística.

El llistat d'aquesta classe es troba en l'Annex I, apartat 12.5 d'aquesta memòria.

#### 6.1.4 ViewControllerPass

Aquí també es tractaria d'una classe tipus *UIViewController*, en aquest cas associada a la pantalla encarregada de sol·licitar la contrasenya al usuari, que li permeti accedir a la pantalla de configuració.

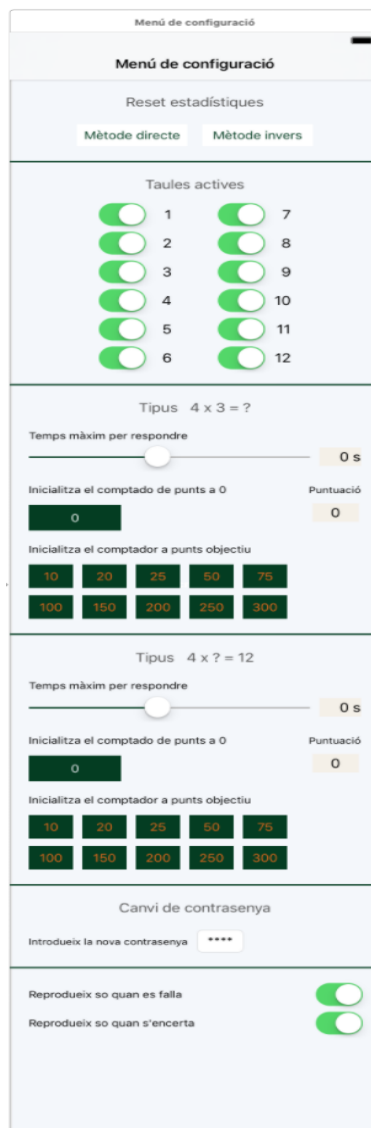


Des d'aquesta classe s'accediria a la pantalla de configuració un cop validada la contrasenya.

El llistat d'aquesta classe es troba en l'Annex I, apartat 12.3 d'aquesta memòria.

### 6.1.5 ViewControllerAjustos

Aquesta classe, també del tipus *UIViewController*, estaria associada a la pantalla de configuració de l'aplicació. S'accediria a aquesta pantalla des de la pantalla «Accés a configuració» i, com es pot apreciar, seria la pantalla amb més components.

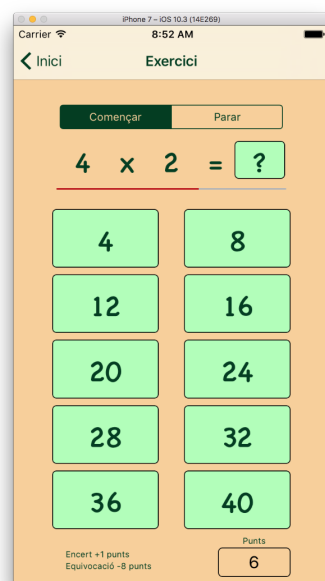


En aquesta pantalla hi podem trobar el següents tipus de components: *UILabel* (etiqueta), *UIButton* (botó), *UISwitch* (interruptor), *UISlider* (selector per desplaçament) i *UITextField* (camp de text).

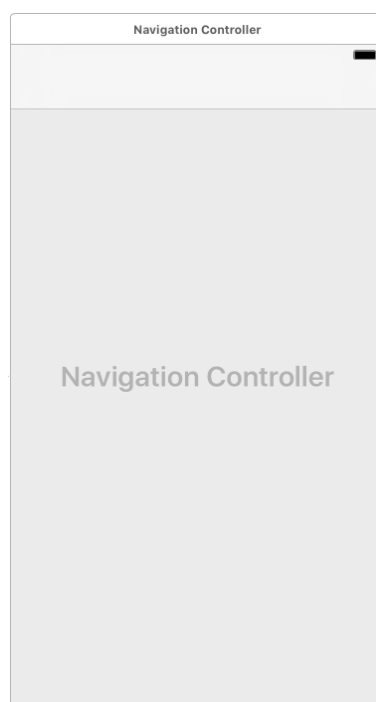
El llistat d'aquesta classe es troba en l'Annex I, apartat 12.4 d'aquesta memòria.

### 6.1.6 UINavigationController

Pel que fa a la navegació de l'aplicació entre les diferents pantalles, d'això s'encarrega la classe *UINavigationController*. Aquesta classe és la responsable de fer aparèixer una barra de navegació a la part superior de totes les pantalles i de gestionar la navegació.



A l'estructura de l'aplicació el controlador de navegació es pot veure representat d'aquesta forma:



### 6.1.7 AppDelegate

A banda de les classes anteriors, la classe que també s'ha modificat respecte a l'original, que Xcode crea per defecte, és la classe *AppDelegate*. Es tracta d'una classe que actua transversalment a tota l'aplicació i és la que té la responsabilitat principal d'interaccionar amb el sistema. Aquesta classe implementa una sèrie de mètodes delegats que poden ser utilitzats per l'aplicació i que permeten, per exemple, detectar quan l'aplicació ha estat llançada o quan torna a estar activa després d'haver passat a la inactivitat. En aquest projecte s'han utilitzat aquests dos mètodes per a definir el context de color general de la barra de navegació i



per a retardar l'ocultació de la pantalla de benvinguda de l'aplicació. A continuació podem veure la implementació d'aquests mètodes:

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
    [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.
    // Modifica els color per defecte del component UINavigationController per a tota l'app.

    UINavigationController.appearance().tintColor = TINT_COLOR
    UINavigationController.appearance().titleTextAttributes = [NSForegroundColorAttributeName:
                                                                TINT_COLOR]

    return true
}

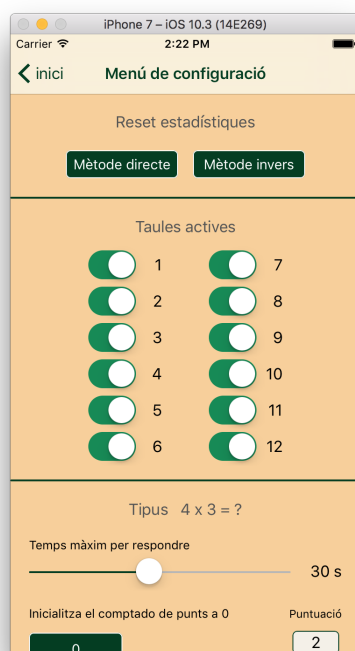
func applicationDidBecomeActive(_ application: UIApplication) {
    // Restart any tasks that were paused (or not yet started) while the application
    // was inactive. If the application was previously in the background, optionally
    // refresh the user interface.

    // Aguanta la pantalla LaunchScreen uns segons més.
    Thread.sleep(forTimeInterval: 4)
}
```

El llistat sencer d'aquesta classe es troba en l'Annex I, apartat 12.6 d'aquesta memòria.

## 6.2 Canvis respecte el prototipus de partida

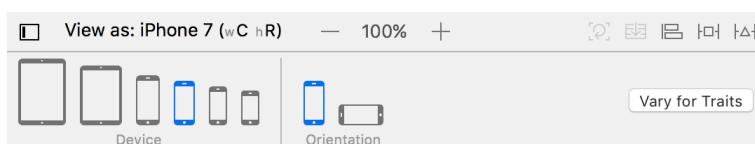
Si bé les funcionalitats principals no han canviat respecte al prototipus original (apartat 5.2), s'han introduït dos petits canvis a nivell funcional que milloren la usabilitat de l'aplicació i la fan més intuïtiva. En primer lloc tenim la introducció d'una barra superior de navegació, la qual permet a l'usuari moure's per les diferents pantalles de l'aplicació d'una manera més senzilla i intuïtiva (anteriorment s'havia previst fer-ho mitjançant botons). I en segon lloc tenim el mecanisme de *reset* de les estadístiques, que s'ha introduït a la pantalla de configuració, i que permet fer un seguiment més precís de l'evolució de l'alumne. A la figura següent podem veure aquests dos elements:



### 6.3 Altres aspectes relatius a la implementació

Una implementació d'aquest tipus, on hi ha més d'un dispositiu objectiu amb diferents configuracions de pantalla, comporta certes dificultats d'implementació, sobretot pel que fa a la *interface gràfica*. La dificultat rau en com representar adequadament els components gràfics de l'aplicació quan les dimensions de la pantalla varien en funció del dispositiu i de la seva orientació.

En aquest cas s'ha previst que l'aplicació pugui ser executada en qualsevol dispositiu de les famílies iPad i iPhone, cosa que tot i que només s'ha previst permetre una orientació vertical dels dispositius, encara implica una varietat important de dimensions de pantalla. La forma d'abordar el problema ha estat utilitzant els mecanismes que Xcode ens proporciona per a fer front a aquesta problemàtica, les *classes* de pantalla i les restriccions. Aquests mecanismes permeten que la distribució dels components en la pantalla es faci de forma dinàmica en temps d'execució, adaptant la representació dels components gràfics al tamany i orientació del dispositiu en cada moment.



Pel que fa a la part programàtica de l'aplicació no hi ha cap aspecte important a ressaltar. La implementació s'ha fet amb la versió 3 del llenguatge *Swift*, però degut a que no incorpora cap classe especial que difereixi d'altres versions anteriors, és molt probable que no hi hagi incompatibilitats amb versions anteriors del llenguatge en el cas que fos necessari utilitzar-ne alguna.

## 7 Avaluació del projecte

### 7.1 Obtenció d'informació sobre els usuaris (pares i educadors)

Per tal d'avaluar la idoneïtat del nou mètode d'aprenentatge de les taules, es proposen una sèrie de preguntes orientades als pares i educadors que permetin obtenir informació relacionada amb l'ús dels dispositius així com la visió dels pares i educadors envers del mètode. L'anàlisi de les respostes pot permetre avaluar millor la disposició a adoptar aquestes noves tecnologies d'aprenentatge.

- Quins dispositius *Apple* tens a casa teva o a classe?
- Deixes que el nen (o nens) els utilitzin?
- Creus que la utilització de dispositius com smartphones o tauletes poden ser útils per l'educació?
- Creus que els mètodes clàssics d'aprenentatge de les taules de multiplicar funcionen amb els nens d'avui dia?
- Creus que la utilització de dispositius com smartphones o tauletes poden facilitar aquest aprenentatge?

### 7.2 Avaluació del producte (l'aplicació)

Per tal d'avaluar el resultat el producte obtingut, en aquest cas l'aplicació, caldrà establir un conjunt de proves que permetin verificar les funcionalitats de l'aplicació d'acord a amb les especificacions requerides.

Un cop implementada l'aplicació és important avaluar-la, no només a nivell funcional com es faria des d'una basant de funcionament, sinó tenint en compte els aspectes d'interacció amb l'usuari. Per aquesta avaluació s'han previst una sèrie de tasques que l'usuari hauria de realitzar i que permetrien avaluar aquests aspectes. Les tasques que els usuaris haurien de realitzar amb la finalitat d'avaluar l'aplicació serien configurar una sèrie d'exercicis per tal d'avaluar la facilitat d'ús d'aquesta i les seves possibilitats. La realització d'aquestes tasques permetrà també afinar respostes a les preguntes de l'apartat anterior així com detectar problemes d'usabilitat que pugui haver.

**Tasca 1.** Configurar una sessió de treball en mode *directe* per a practicar les taules del 2 al 4, permetent un temps màxim per a respondre de 20s.

**Tasca 2.** Inicialitzar el comptador de punts del mode *directe* a 0.

**Tasca 3.** Inicialitzar el comptador de punts a 25 amb l'objectiu de 0 (descomptant).

**Tasca 4.** Configurar una sessió de treball en mode *Invers* per a practicar les taules del 2 al 12.

**Tasca 5.** Canviar la contrasenya d'accés.

Adicionalment, també caldrà fer algunes proves amb nens a partir d'alguns exercicis que els pares o educadors configuren per tal d'avaluar la resposta d'aquests amb l'aplicació.

## 7.3 Preguntes referent a les tasques

Un cop realitzades les tasques proposades a l'apartat anterior, procedirem a obtenir el feedback corresponent a partir de les respostes d'una sèrie de preguntes que farem als pares i educadors, i també als nens com a «usuaris finals».

### 7.3.1 Preguntes als pares i educadors (usuaris tipus 1)

- S'entén bé el funcionament de l'aplicació com a mètode d'aprenentatge de les taules de multiplicar?
- T'ha costat entendre l'operativitat de l'aplicació?
- Creus que el nen té dificultats per a utilitzar-la?
- El menú de configuració és prou intuïtiu?
- Creus que faltaria parametritzar alguna cosa més?
- T'agrada la paleta de color seleccionada?
- Trobes a faltar alguna cosa en l'aplicació?
- Et sembla correcta la informació presentada en el gràfic?
- Creus que l'ús d'aquesta aplicació agilitza el procés d'aprenentatge de les taules de multiplicar?

### 7.3.2 Preguntes als nens (usuaris tipus 2)

- Trobes que aquesta app és senzilla d'utilitzar?
- T'agrada aprendre les taules de multiplicar d'aquesta manera?
- T'agraden els colors que surten?

## 8 Conjunt de proves

En aquest apartat es descriuen les proves que s'han fet per a validar el funcionament de l'aplicació. L'objectiu d'aquestes és comprovar que l'aplicació es comporta tal com s'ha previst i detectar possibles *bugs* abans d'alliberar la versió final.

### 8.1 Metodologia de les proves

Degut a la diversitat de dispositius i versions on l'aplicació es pot instal·lar, les proves s'han fet, sobretot, en el simulador del Xcode. L'execució de l'aplicació en un equip simulat dóna suficients garanties de funcionament com per a validar l'aplicació per a un dispositiu real.

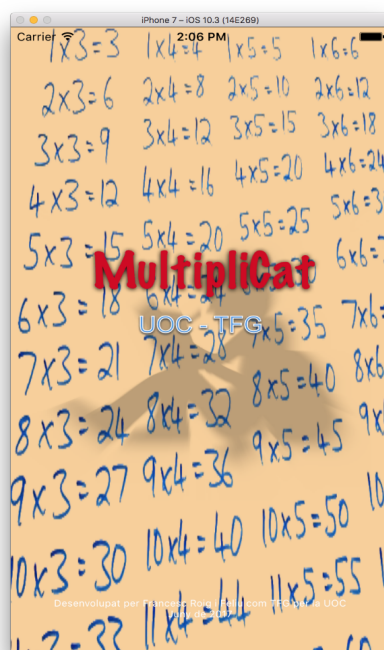
S'estableixen dues categories de proves. La primera correspon a proves de tipus funcional, les que tenen a veure amb la sessió de «joc» de l'alumne. I la segona les de configuració. En les de configuració s'han provat diferents configuracions per tal de verificar que l'aplicació és comporta segons el previst. Addicionalment també s'ha establert una prova de navegació per les diferents pantalles de l'aplicació.

Apart de les proves funcionals, també s'han fet proves amb un nen (un usuari real) per a comprovar si l'aplicació proporciona els resultats previstos.

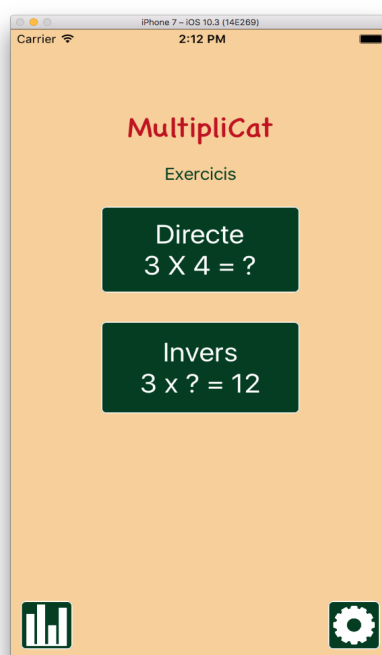
### 8.2 Proves de funcionals de «joc»

Abans d'abordar les dues categories de proves, s'ha fet una prova de navegació per l'aplicació amb la finalitat de verificar que es pot accedir a qualsevol pantalla d'acord amb el que està previst. Aquesta prova és simple, només cal moure's per les pantalles de l'aplicació. El resultat d'aquesta prova ha estat el següent.

**Pantalla inicial de benvinguda.** S'arriba a aquesta pantalla en el moment d'iniciar l'aplicació.



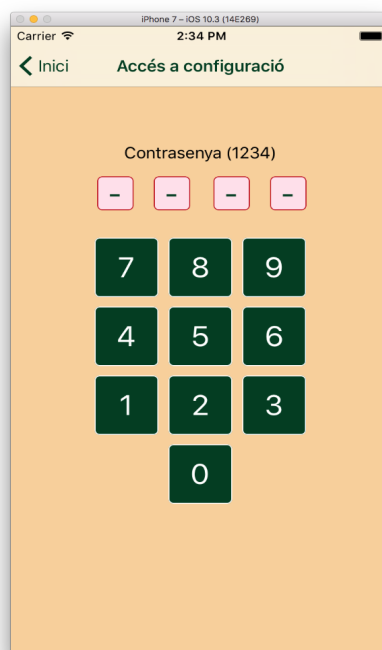
**Pantalla principal.** En aquesta pantalla hi trobem els botons que permeten al nen triar quin tipus d'exercici vol (o ha) de practicar a més dels botons d'accés als menús de configuració i d'estadístiques.



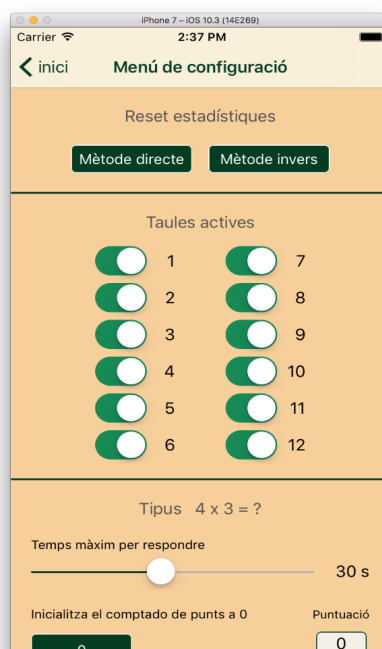
**Pantalla d'estadístiques.** Des de la pantalla principal podem navegar a la pantalla d'estadístiques on hi l'estadística dels exercicis per taules. El valor numèric representa el percentatge d'encert respecte al nombre total de vegades que surt cadascuna de les taules.



**Pantalla d'accés a la pantalla de configuració.** És la pantalla que dóna accés a la pantalla de configuració a través d'una contrasenya.

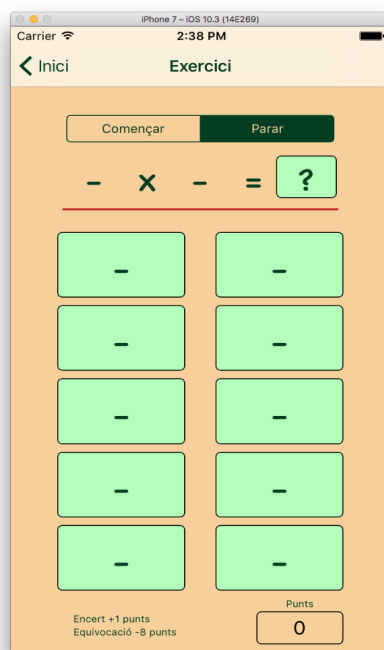


**Pantalla de configuració.** És des d'on es poden configurar tots els paràmetres de l'aplicació. Es tracta d'una pantalla amb scroll.

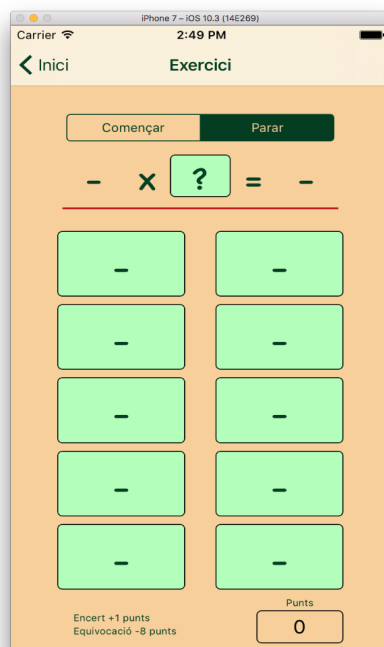


**Pantalla de «joc» directe.** Es tracta de la pantalla de treball on el nen anirà respondent les preguntes que l'aplicació li faci. En aquesta pantalla podem trobar: un botó d'inici i parada (pausa) de la sessió de «joc», la

zona on es fa la pregunta, una barra que indica el temps de resposta que queda en cada moment ( timeout), els botons amb les possibles respostes i el nombre de punts aconseguits.



**Pantalla de «joc» invers.** Es tracta d'una pantalla molt similar a l'anterior però, en aquest cas, la pregunta varia d'acord amb aquesta opció de joc. La resta és igual.

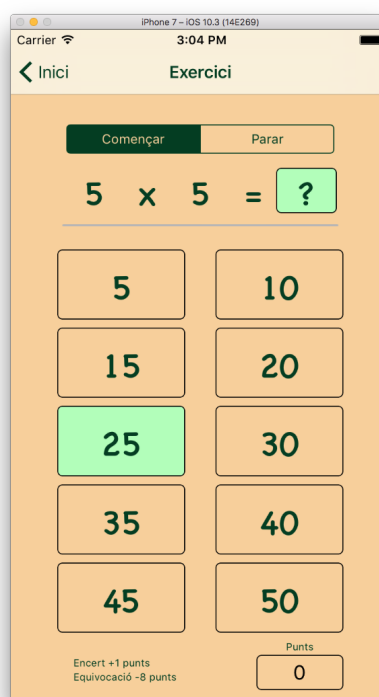
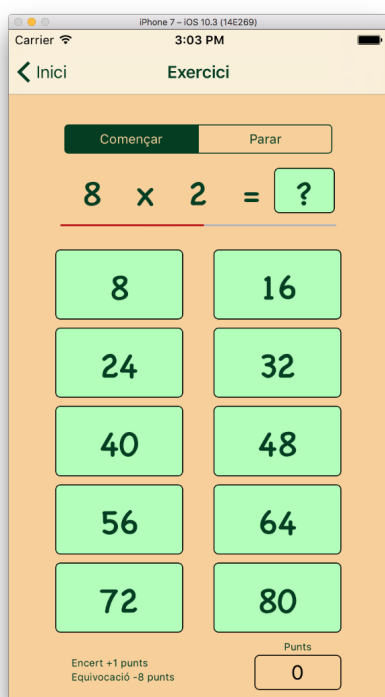




### 8.2.1 «Joc» directe

Aquesta prova consisteix en fer una sessió de «joc» directe i anar comprovant que els encerts i els error es comptabilitzen correctament. Es comprova també que, en el cas que la resposta sigui incorrecta, es marca la resposta errònia i es mostra la correcta.

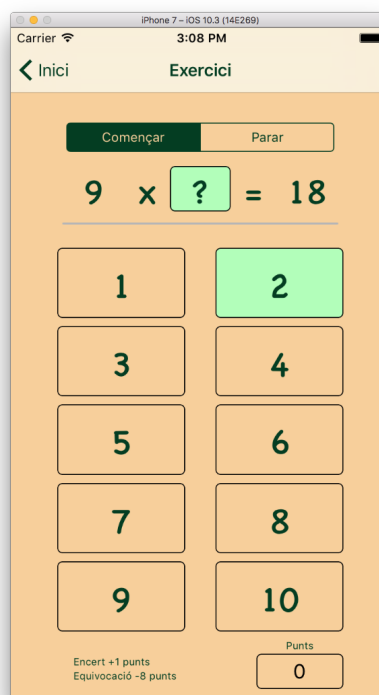
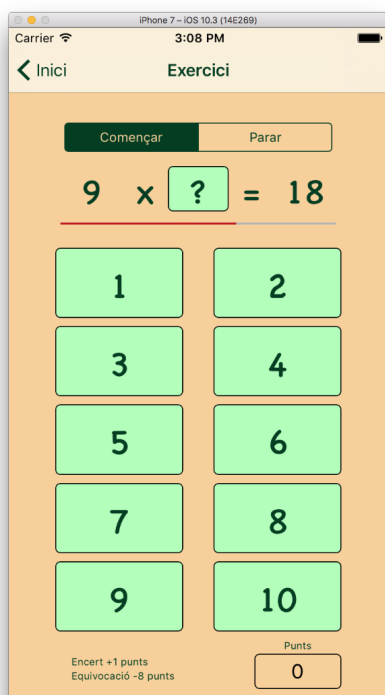
També es comprova el temps (timeout) de resposta i com, després d'haver transcorregut aquest temps, es passa a l'estat de resposta incorrecta.



### 8.2.2 «Joc» invers

Aquesta prova consisteix en fer una sessió de «joc» invers i anar comprovant que els encerts i els error es comptabilitzen correctament. Es comprova també que, en el cas que la resposta sigui incorrecta, es marca la resposta errònia i es mostra la correcta.

Al igual que en l'apartat anterior, aquí també es comprova el temps (timeout) de resposta i com, després d'haver transcorregut aquest temps, es passa a l'estat de resposta incorrecta.



## 8.3 Proves de configuració

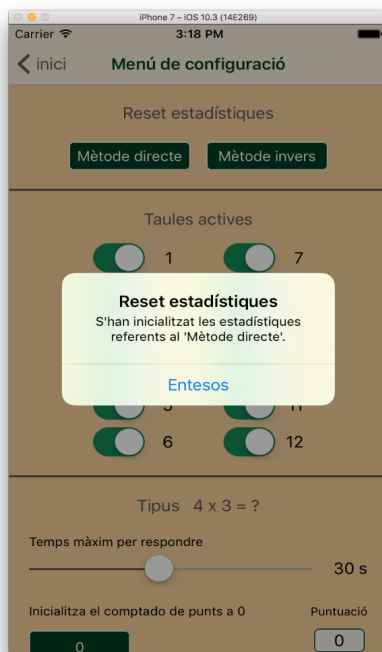
### 8.3.1 Reset d'estadístiques

Aquesta prova consisteix en verificar que després d'haver acumulat uns resultats de joc, prement els botons de reset les estadístiques es posen a 0.

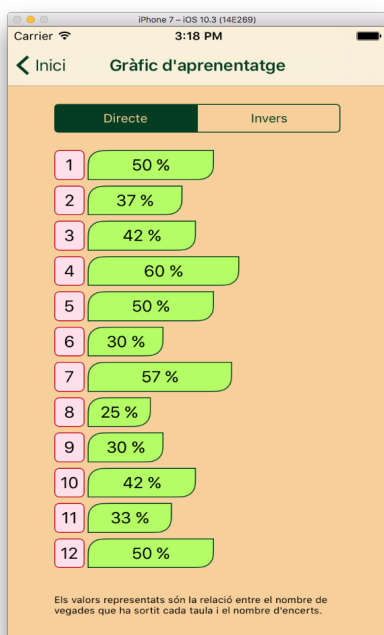
**Botons de reset:**



Quan es prem qualsevol d'aquests botons apareix l'avís informatiu de l'acció realitzada:



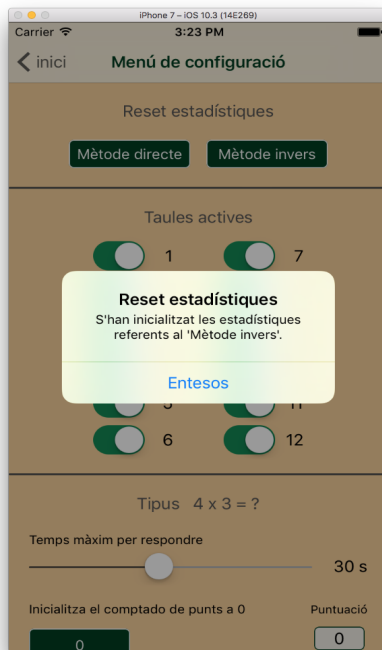
**Resultat (abans/després):**



En el cas de prémer el botó de «Mètode invers», passaria el mateix.

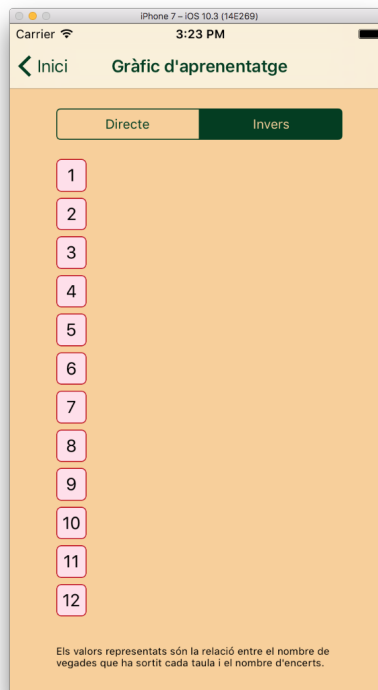
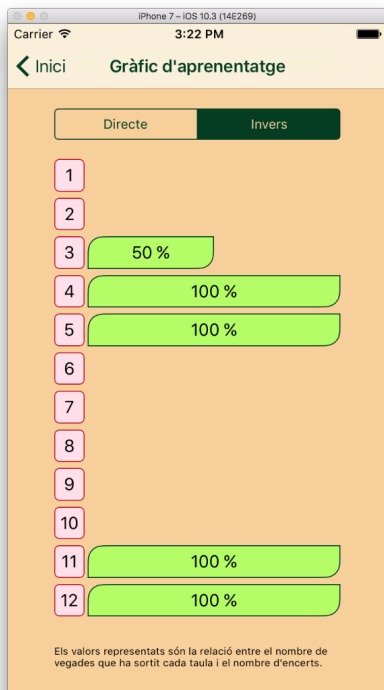
**Botons de reset:**





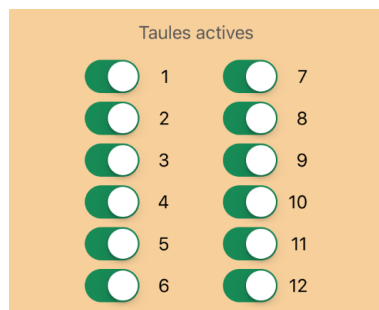
Quan es prem qualsevol d'aquests botons apareix l'avís informatiu d'acció realitzada:

**Resultat (abans/després):**



### 8.3.2 Selecció de taules de «joc»

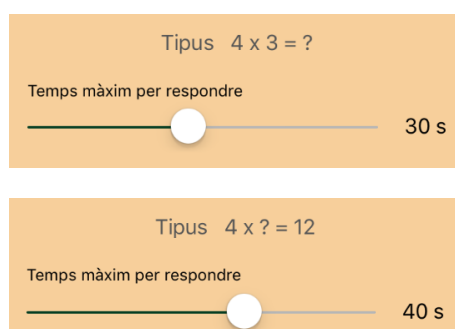
La prova consisteix en comprovar que després de deshabilitar algunes taules, aquestes no apareixen en les preguntes que es fan. Pot passar, això sí, que si per exemple es deshabilita la taula del 6 però la del 3 està habilitada, aparegui *la pregunta*  $3 \times 6 = ?$ ; la que no apareixeria seria la de  $6 \times 3 = ?$ .



Aquesta prova s'ha fet tant pel mètode directe com pel invers.

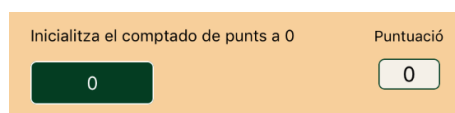
### 8.3.3 Ajust del temps màxim de resposta

La prova consisteix en comprovar que si es modifica el temps màxim de resposta, tant pel mètode directe com pel invers, aquest canvi s'aplica en la sessió de «joc»



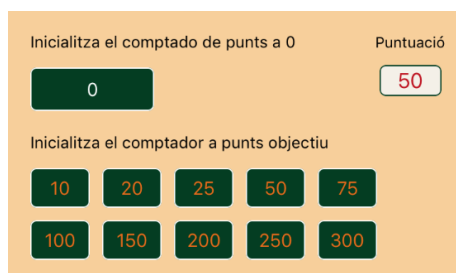
### 8.3.4 Inicialització del comptador de punts

La prova consisteix en comprovar que si es prem el botó de reset dels punts, el comptador es posa a 0. El reset de punts es pot fer de forma independent pel mètode directe i invers.



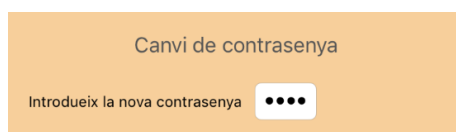
### 8.3.5 Inicialització del comptador a punts objectiu

La prova consisteix en comprovar que si es prem un botó amb un determinat nombre punts, aquesta puntuació objectiu passa al comptador de punts. En aquest cas, cada resposta correcta anirà restant 1 punt fins arribar a 0.

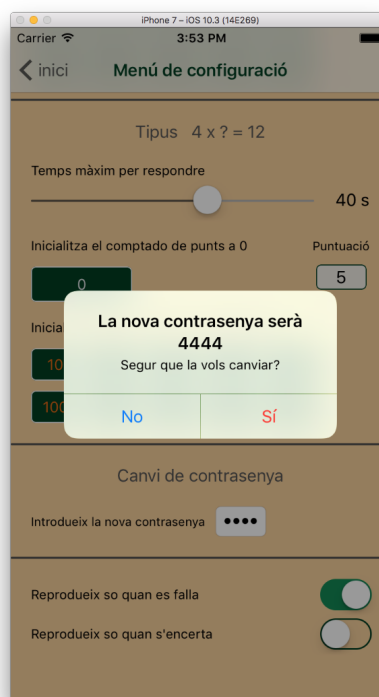
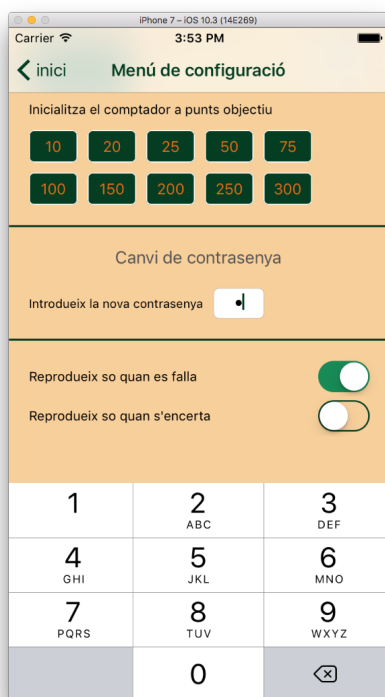


### 8.3.6 Canvi de contrasenya

Aquesta prova consisteix en canviar la contrasenya d'accés al *Menú de configuració*, per defecte 1234.

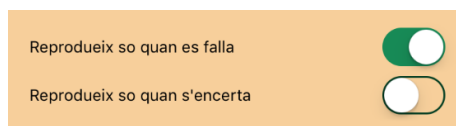


Comprovem com apareix el teclat numèric per a poder introduir el nou codi i com després es sol·licita la confirmació del canvi:



### 8.3.7 Avisos acústics

Aquesta prova consisteix en habilitar i deshabilitar els avisos acústics que pot fer l'aplicació quan l'usuari (el nen) respon a una pregunta. Per defecte només es troba habilitat l'avís de resposta incorrecta.



### 8.3.8 Proves amb usuari real

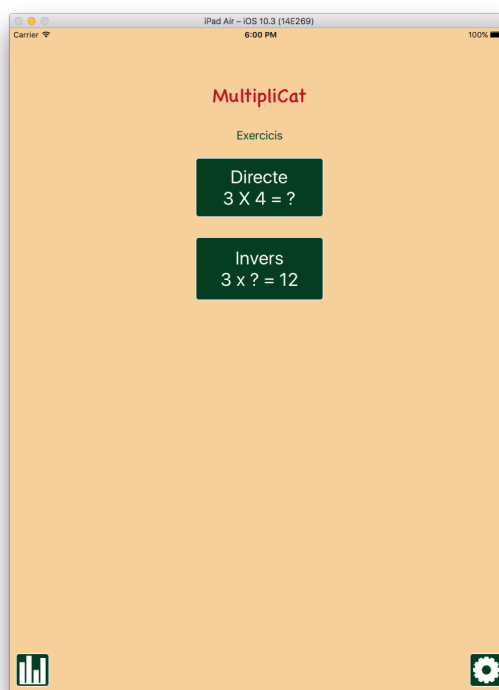
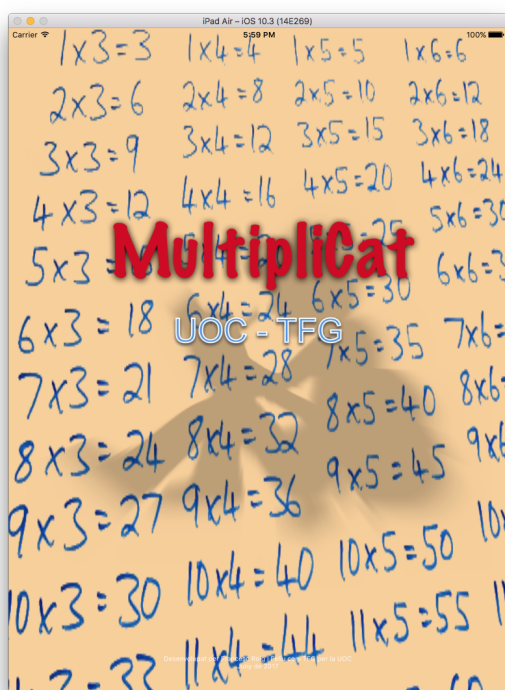
Aquestes proves les he anat fent durant tota l'etapa de desenvolupament de l'aplicació amb el meu fill.

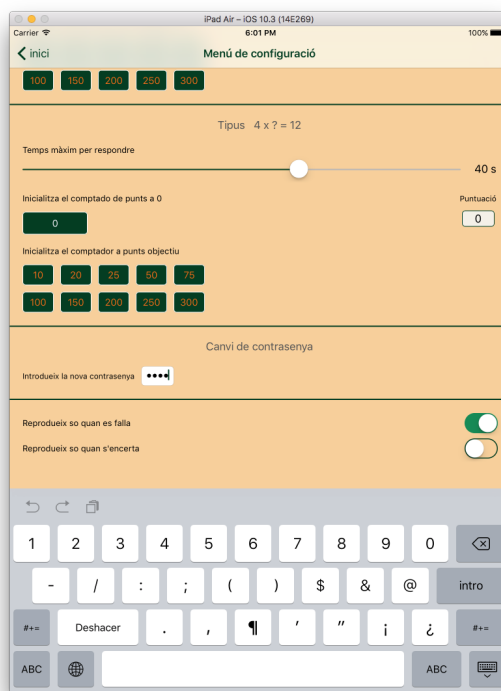
A més d'anar comprovant el funcionament de l'aplicació en un entorn real (en aquest cas en un iPhone 6 i amb un nen de 8 anys), Aquestes proves m'han permès anar ajustant els paràmetres per defecte a partir de les diferents sessions de joc que li havia preparat. També m'ha permès avaluar el grau d'efectivitat del mètode, que he de dir que ha resultat molt satisfactori.

### 8.3.9 Proves addicionals multi dispositiu

S'han fet proves de funcionament de l'aplicació en dispositius de la família iPad. En aquest cas, l'única cosa que canvia és la mida dels components.

Aquí podem veure algunes mostres:





## 8.4 Provar l'aplicació en Xcode

Per a executar el projecte cal tenir Xcode actualitzat a l'última versió (8.3.2). Potser en versions anterior funciona, però no ho he pogut comprovar. Simplement cal obrir el fitxer *MultipliCat.xcodeproj* en Xcode i des d'aquest ja es pot executar la simulació.



## 9 Valoració econòmica

Al tractar-se d'un projecte *open source* no contemplo un cost d'hores de desenvolupament (representa que, al estar fora de l'horari laboral, el cost d'oportunitat d'aquests temps és 0), però sí que la publicació de l'aplicació en *App Store* pot implicar unes despeses fixes anuals.

Adquirir una llicència de desenvolupador que permeti publicar i mantenir aplicacions en el *App Store* d'Apple, comporta unes despeses anuals de **99 €**. En el cas de no disposar d'una llicència d'aquest tipus caldrà tenir en compte aquest cost.

## 10 Conclusions

Després de finalitzar aquest projecte, puc dir que valoro molt positivament l'experiència que m'ha aportat abordar el desenvolupament d'una aplicació com aquesta. M'ha motivat especialment el fet que, a banda de l'objectiu de fer un TFG, hagi estat el meu fill un dels beneficiats d'aquest projecte.

Per altra banda, també ha estat especialment interessant el poder abordar el projecte no només des de la basant purament tecnològica, sinó també des d'una basant metodològica. Penso que aquest projecte obre les portes a una forma alternativa d'abordar les aplicacions d'aquest tipus, amb uns plantejaments de funcionament diferents als que s'apliquen actualment.

Un altre dels objectius que es perseguia, i que crec que també s'aconsegueix, és el d'augmentar la presència del català en les aplicacions. Ja sigui per motius econòmics o estratègics, l'ús del català en les aplicacions encara és molt baix, i crec que amb iniciatives com aquesta on es prioritza l'ús d'aquesta llengua, es contribueix a fomentar-la.

Des del punt de vista tecnològic ha estat un projecte on la part gràfica té un pes especial, com no podia ser de cap altra manera en una aplicació d'aquest tipus. En general s'han utilitzat els components gràfics estàndards, però també podem trobar una modificació interessant a nivell programàtic d'un d'ells per a fer-lo més atractiu visualment parlant (pantalla d'estadístiques).

Per acabar, puc dir que els projectes que he fet al llarg de la meua vida professional sempre m'han permès conèixer coses noves, aprofundir en les que ja sabia i, sobretot, agafar experiència per abordar nous projectes i reptes tecnològics. Aquest projecte ha permès acostar-me al món de l'educació infantil i entendre millor com la tecnologia pot ajudar en el procés d'aprenentatge dels nostres fills, cosa que ha estat especialment gratificant pel fet d'haver-ho pogut aplicar també amb el meu.

## 11 Referències bibliogràfiques

- |                                 |  |
|---------------------------------|--|
| Apple (iBook)                   | <i>The Swift Programming Language. Swift 3.1</i> , Apple (2017)  |
| Paul Hudson (web)               | <i>Hacking with Swift</i> (2016), <a href="http://www.hackingwithswift.com">www.hackingwithswift.com</a> |
| Vandad Nahavandipoor (O'Reilly) | <i>iOS 8 Swift Programming Cookbook</i> , O'Reilly (2014)  |
| Steven Lipton (iBook)           | <i>Swift Swift View Controllers</i> , iBooks Store (2016)  |

## 12 Annex I. Llistats de les classes

### 12.1 ViewController

```
//
// ViewController.swift
// MultipliCat
//
// Created by Francesc Roig i Feliu.
// Copyright © 2017 Francesc Roig i Feliu.
//

import UIKit

let COLOR_FONS_PANTALLA = UIColor.init(red: 248/255, green: 208/255, blue: 152/255, alpha: 1)
let COLOR_FONS_BOTONS = UIColor.init(red: 9/255, green: 139/255, blue: 84/255, alpha: 1)
let TINT_COLOR = UIColor.init(red: 0, green: 62/255, blue: 33/255, alpha: 1)
let COLOR_VERD_2 = UIColor.init(red: 6/255, green: 114/255, blue: 66/255, alpha: 1)
let COLOR_VERD_3 = UIColor.init(red: 9/255, green: 139/255, blue: 84/255, alpha: 1)
let COLOR_VERMELL = UIColor.init(red: 193/255, green: 14/255, blue: 25/255, alpha: 1)
let COLOR_ROSA = UIColor.init(red: 255/255, green: 223/255, blue: 234/255, alpha: 1)
let COLOR_BLAUCEL = UIColor.init(red: 204/255, green: 255/255, blue: 255/255, alpha: 1)

let COLOR_FONS_NORMAL = UIColor(red: 176/255, green: 255/255, blue: 184/255, alpha: 1)
let COLOR_FONS_ENCERT = UIColor(red: 150/255, green: 1.0, blue: 150/255, alpha: 1)
let COLOR_FONS_ERROR = UIColor(red: 1.0, green: 100/255, blue: 100/255, alpha: 1)
let COLOR_TRANSPARENT = UIColor.white.withAlphaComponent(0.0)

let COLOR_FONS_GRIS = UIColor(red: 0.85, green: 0.85, blue: 0.85, alpha: 1)
let COLOR_TEXT_BOT_SOL = UIColor(red: 0.0, green: 0.0, blue: 0.5, alpha: 1)

let COLOR_FONS_BARRA_GRAFIC = UIColor(red: 0.7, green: 1, blue: 0.36, alpha: 1)

//let ALPHA_BOTONS = CGFloat(0.5)
let ALPHA_BOTONS = CGFloat(1)
let ALPHA_BOTONS_RESP = CGFloat(0)

//{ Perpet comprovar si s'ha executat algun cop aquesta app.
var appInicialitzada = false

var met_directe = true

// Desa els punts minims a partir dels quals ja no se'n restaran més.
var punts_minims_dir = 0
var punts_minims_ind = 0

// Taules actives.
var taules = [true, true, false, false, false, false, false, false, false, false, false]

var password:String = "1234"

var soOnError = true
var soOnEncert = true

// Estadístiques.
var encerts1:[Int] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
var errors1:[Int] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
var encerts2:[Int] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
var errors2:[Int] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

/*
 *
 *
 */
class ViewController: UIViewController {

    @IBOutlet weak var botoMultDir: UIButton!
    @IBOutlet weak var botoMultiInd: UIButton!
    @IBOutlet weak var botoConfig: UIButton!
    @IBOutlet weak var botoEstad: UIButton!
```

```

/*
 * Mètode viewDidLoad
 */
override func viewDidLoad() {
    super.viewDidLoad()

    botoMultDir.titleLabel?.textAlignment = .center
    botoMultDir.titleLabel?.numberOfLines = 0
    botoMultDir.layer.borderWidth = 1
    botoMultDir.layer.cornerRadius = 5
    botoMultDir.layer.borderColor = UIColor.white.cgColor

    botoMultiInd.titleLabel?.textAlignment = .center
    botoMultiInd.titleLabel?.numberOfLines = 0
    botoMultiInd.layer.borderWidth = 1
    botoMultiInd.layer.cornerRadius = 5
    botoMultiInd.layer.borderColor = UIColor.white.cgColor

    botoConfig.layer.borderWidth = 1
    botoConfig.layer.cornerRadius = 5
    botoConfig.layer.borderColor = UIColor.white.cgColor

    botoEstad.layer.borderWidth = 1
    botoEstad.layer.cornerRadius = 5
    botoEstad.layer.borderColor = UIColor.white.cgColor

    self.view.backgroundColor = COLOR_FONS_PANTALLA

    let defaults = UserDefaults.standard

    // Comprova si és el primer cop que s'executa la app en el dispositiu
    if defaults.bool(forKey: "AppInicialitzada") {
        // Carrega les taules actives.
        taules = defaults.object(forKey: "TaulesActives") as? [Bool] ?? [Bool]()

        password = (defaults.object(forKey: "Pass") as? String)!

        soOnError = defaults.bool(forKey: "SoOnError")
        soOnEncert = defaults.bool(forKey: "SoOnEncert")

        encerts1 = defaults.array(forKey: "Encerts1") as? [Int] ?? [Int]()
        errors1 = defaults.array(forKey: "Errors1") as? [Int] ?? [Int]()
        encerts2 = defaults.array(forKey: "Encerts2") as? [Int] ?? [Int]()
        errors2 = defaults.array(forKey: "Errors2") as? [Int] ?? [Int]()
    } else {
        // Inicialitzem les variables.
        // Inicialitza la configuració de les taules (totes actives)
        taules = [true, true, true, true, true, true, true, true, true, true, true, true]
        defaults.set(taules, forKey: "TaulesActives")

        // Inicialitza avisos
        defaults.set(true, forKey:"SoOnError")
        defaults.set(false, forKey:"SoOnEncert")

        // Inicialitza timeouts preguntes.
        defaults.set(30, forKey: "TimeOutDir")
        defaults.set(40, forKey:"TimeOutInd")

        // Inicialitza password.
        password = "1234"
        defaults.set(password, forKey: "Pass")

        // Inicialitza estadístiques.
        defaults.set(encerts1, forKey: "Encerts1")
        defaults.set(errors1, forKey: "Errors1")
        defaults.set(encerts2, forKey: "Encerts2")
        defaults.set(errors2, forKey: "Errors2")

        defaults.set(true, forKey:"AppInicialitzada")
        print("Valors inicials restaurats ...")
    }
}

/*
 *
 */

```

```
override func viewWillAppear(_ animated: Bool) {
    self.navigationController?.navigationBar.isHidden = true
}

/*
 *
 */
@IBAction func botoMultDirecte(_ sender: AnyObject) {
    met_directe = true
}

/*
 *
 */
@IBAction func botoMultInvers(_ sender: AnyObject) {
    met_directe = false
}
}
```

## 12.2 ViewControllerTaula

```
//
// ViewControllerTaula.swift
// MultipliCat
//
// Created by Francesc Roig i Feliu.
// Copyright © 2017 Francesc Roig i Feliu. All rights reserved.
//

import UIKit
import AVFoundation

class ViewControllerTaula: UIViewController {

    let inc_puntuacio = 1

    // Fixa el decrement de punts.
    let dec_puntuacio = 8

    // Punts de la sessió en curs.
    var punts_sessio = 0

    var mult_1 = 0
    var mult_2 = 0
    var resultat = 0

    @IBOutlet weak var text_view_punts: UILabel!

    @IBOutlet weak var eti_num_1: UILabel!
    @IBOutlet weak var eti_num_2: UILabel!
    @IBOutlet weak var eti_per: UILabel!
    @IBOutlet weak var eti_igual: UILabel!
    @IBOutlet weak var eti_res: UILabel!

    @IBOutlet weak var eti_encert: UILabel!
    @IBOutlet weak var eti_error: UILabel!

    @IBOutlet weak var barra_temps: UIProgressView!

    @IBOutlet weak var segmenIniciParada: UISegmentedControl!

    @IBOutlet var bot_sol: [UIButton]!

    // Component reproductor
    var audioPlayer: AVAudioPlayer!

    // Timeout de la pregunta
    var timeout_pregunta = Timer()
```

```

//
var TIME_OUT:Float = 0 // x10 segons

var comptado_timeout:Float = 0.0
var increment_comptado_timeout:Float = 0.01

/*
 * Mètode viewDidLoad
 */
override func viewDidLoad() {
    super.viewDidLoad()

    self.view.backgroundColor = COLOR_FONS_PANTALLA

    text_view_punts.backgroundColor = UIColor.white.withAlphaComponent(0)
    text_view_punts.layer.borderWidth = 1
    text_view_punts.layer.cornerRadius = 5
    text_view_punts.layer.masksToBounds = true

    eti_num_1.backgroundColor = COLOR_TRANSPARENT
    eti_per.backgroundColor = COLOR_TRANSPARENT
    eti_igual.backgroundColor = COLOR_TRANSPARENT

    if met_directe {
        eti_num_2.text = "-"
        eti_res.text = "?"
        eti_num_2.backgroundColor = COLOR_TRANSPARENT
        eti_res.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(ALPHA_BOTONS)
        eti_res.layer.borderWidth = 1.0
        eti_res.layer.cornerRadius = 5
        eti_res.layer.masksToBounds = true
    } else {
        eti_num_2.text = "?"
        eti_res.text = "-"
        eti_num_2.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(ALPHA_BOTONS)
        eti_res.backgroundColor = COLOR_TRANSPARENT
        eti_num_2.layer.borderWidth = 1.0
        eti_num_2.layer.cornerRadius = 5
        eti_num_2.layer.masksToBounds = true
    }

    for b in bot_sol {
        b.setTitleColor(TINT_COLOR, for: .disabled)
        b.setTitleColor(TINT_COLOR, for: UIControlState())
        b.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(ALPHA_BOTONS)
        // b.backgroundColor = COLOR_ROSA
        b.isEnabled = false
        b.layer.borderWidth = 1.0
        b.layer.cornerRadius = 5
        b.layer.masksToBounds = true
    }
}

/*
 *
 */
override func viewWillAppear(_ animated: Bool) {
    self.navigationController?.navigationBar.isHidden = false
}

/*
 *
 */
override func viewWillDisappear(_ animated: Bool) {
    timeout_pregunta.invalidate()

    for b in bot_sol {
        b.isEnabled = false
        b.setTitle("-", for: UIControlState())
        b.layer.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(0).CGColor
    }
    if met_directe {
        eti_num_1.text = "-"
        eti_num_2.text = "-"
    }
}

```

```

        eti_res.text = "?"
    } else {
        eti_num_1.text = "-"
        eti_num_2.text = "?"
        eti_res.text = "-"
    }
}

/*
 *
 */
override func viewDidAppear(_ animated: Bool) {

    for b in bot_sol {
        let h = CGFloat(b.frame.height / 2.0)
        b.titleLabel!.font = UIFont(name: "Chalkboard SE", size: h)
        b.setTitle("-", for: UIControlState())
    }

    let defaults = UserDefaults.standard

    if met_directe {
        punts_sessio = defaults.integer(forKey: "Punts1")
        let t = defaults.integer(forKey: "TimeOutDir")
        encerts1 = defaults.object(forKey: "Encerts1") as? [Int] ?? [Int]()
        errors1 = defaults.object(forKey: "Errors1") as? [Int] ?? [Int]()
        if t > 0 {
            TIME_OUT = (Float)(t/10)
        } else {
            TIME_OUT = 3.0
        }
    } else {
        punts_sessio = defaults.integer(forKey: "Punts2")
        let t = defaults.integer(forKey: "TimeOutInd")
        encerts2 = defaults.object(forKey: "Encerts2") as? [Int] ?? [Int]()
        errors2 = defaults.object(forKey: "Errors2") as? [Int] ?? [Int]()
        if t > 0 {
            TIME_OUT = (Float)(t/10)
        } else {
            TIME_OUT = 4.0
        }
    }

    punts_minims_dir = defaults.integer(forKey: "PuntsMinimDir")
    punts_minims_ind = defaults.integer(forKey: "PuntsMinimInd")

    if punts_sessio < 0 {
        text_view_punts.backgroundColor = COLOR_ROSA
        text_view_punts.text = "\\((-1*punts_sessio))"
    } else {
        text_view_punts.textColor = UIColor.black
        text_view_punts.text = "\\(punts_sessio)"
    }

    eti_encert.text = "Encert +\\(inc_puntuacio) punts"
    eti_error.text = "Equivocació -\\(dec_puntuacio) punts"

    barra_temps.setProgress(1.0, animated: false)

    segmenIniciParada.selectedSegmentIndex = 1
}

/*
 *
 */
@IBAction func controlIniciParada(_ sender: UISegmentedControl) {

    if sender.selectedSegmentIndex == 0 {
        novaMultiplicacio()
    } else {
        timeout_pregunta.invalidate()
        for b in bot_sol {
            b.isEnabled = false
            b.setTitle("-", for: UIControlState())
            b.layer.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(0).CGColor
        }
        if met_directe {

```



```

        eti_num_1.text = "-"
        eti_num_2.text = "-"
        eti_res.text = "?"
    } else {
        eti_num_1.text = "-"
        eti_num_2.text = "?"
        eti_res.text = "-"
    }
}

/*
 *
 */
@IBAction func accioBotoSol(_ sender: UIButton) {

    if met_directe {
        if resultat == Int(sender.currentTitle!)! {
            marcaEncert()
        } else {
            marcaError()
            sender.backgroundColor = COLOR_FONS_ERROR
        }
    } else {
        if mult_2 == Int(sender.currentTitle!)! {
            marcaEncert()
        } else {
            marcaError()
            sender.backgroundColor = COLOR_FONS_ERROR
        }
    }
}

/*
 * Gennera una nova multiplicació
 */
func novaMultiplicacio() {

    // Desa dades.
    let defaults = UserDefaults.standard
    if met_directe {
        defaults.set(punts_sessio, forKey: "Punts1")
        defaults.set(encerts1, forKey: "Encerts1")
        defaults.set(errors1, forKey: "Errors1")
    } else {
        defaults.set(punts_sessio, forKey: "Punts2")
        defaults.set(encerts2, forKey: "Encerts2")
        defaults.set(errors2, forKey: "Errors2")
    }

    if punts_sessio<0 {
        text_view_punts.backgroundColor = COLOR_ROSA
        text_view_punts.text = "\\((-1 * punts_sessio))"
    } else {
        if punts_minims_dir<0 && met_directe {
            text_view_punts.backgroundColor = COLOR_BLAUCEL //{ Indica punt positiu de premi en el
mode directe.
        } else {
            text_view_punts.textColor = UIColor.black
        }
        if punts_minims_ind<0 && !met_directe {
            text_view_punts.backgroundColor = COLOR_BLAUCEL //{ Indica punt positiu de premi en el
mode indirecte.
        } else {
            text_view_punts.textColor = UIColor.black
        }
        text_view_punts.text = "\\(punts_sessio)"
    }

    // Genera un número que estigui dins del rang de taules actives
    var a = true
    while(a) {
        mult_1 = Int(arc4random_uniform(12) + 1)
        if taules[mult_1-1] {
            a = false
        }
    }
}

```

```

    mult_2 = Int(arc4random_uniform(10) + 1)

    resultat = mult_1 * mult_2

    if met_directe {
        eti_num_1.text = "\(mult_1)"
        eti_num_2.text = "\(mult_2)"
    } else {
        eti_num_1.text = "\(mult_1)"
        eti_res.text = "\(resultat)"
    }

    var i = 1
    for b in bot_sol {
        if met_directe {
            b.setTitle("\(mult_1*i)", for: UIControlState())
        } else {
            b.setTitle("\(i)", for: UIControlState())
        }
        i += 1
        b.layer.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(ALPHA_BOTONS).CGColor
    }

    barra_temps.setProgress(1.0, animated: false)
    comptado_timeout = TIME_OUT
    timeout_pregunta = Timer.scheduledTimer(timeInterval: 0.1, target: self, selector:
#selector(decrementaTimeout), userInfo: nil, repeats: true)

    for b in bot_sol {
        b.isEnabled = true
    }
}

/*
 *
 */
func marcaError() {
    timeout_pregunta.invalidate()

    // Actualitza estadística.
    if met_directe {
        errors1[mult_1] += 1
    } else {
        errors2[mult_1] += 1
    }

    punts_sessio -= dec_puntuacio
    if punts_sessio < punts_minims_dir && met_directe {
        punts_sessio = punts_minims_dir
    }

    if punts_sessio < punts_minims_ind && !met_directe {
        punts_sessio = punts_minims_ind
    }

    if soOnError {
        reproduceixAudio(nom: "uhoh", player: &audioPlayer)
        audioPlayer.play()
    }

    for b in bot_sol {
        if met_directe {
            if resultat == (b.currentTitle! as NSString).integerValue {
                b.backgroundColor = COLOR_FONS_NORMAL
            } else {
                b.backgroundColor = COLOR_FONS_GRIS.withAlphaComponent(0)
            }
        } else {
            if mult_2 == (b.currentTitle! as NSString).integerValue {
                b.backgroundColor = COLOR_FONS_NORMAL
            } else {
                b.backgroundColor = COLOR_FONS_GRIS.withAlphaComponent(0)
            }
        }
        b.isEnabled = false
    }
    Timer.scheduledTimer(timeInterval: 4, target: self, selector: #selector(tornaEstatNormal),
userInfo: nil, repeats: false)
}

```

```

}

/*
 *
 */
func marcaEncert() {
    // Actualitza estadística.
    if met_directe {
        encerts1[mult_1] += 1
    } else {
        encerts2[mult_1] += 1
    }

    if soOnEncert {
        reproduceixAudio(nom: "beep", player: &audioPlayer)
        audioPlayer.play()
    }

    timeout_pregunta.invalidate()

    punts_sessio += inc_puntuacio

    for b in bot_sol {
        if met_directe {
            if resultat == (b.currentTitle! as NSString).integerValue {
                b.backgroundColor = COLOR_FONS_NORMAL
            } else {
                b.backgroundColor = COLOR_FONS_GRIS.withAlphaComponent(ALPHA_BOTONS_RESP)
            }
        } else {
            if mult_2 == (b.currentTitle! as NSString).integerValue {
                b.backgroundColor = COLOR_FONS_NORMAL
            } else {
                b.backgroundColor = COLOR_FONS_GRIS.withAlphaComponent(ALPHA_BOTONS_RESP)
            }
        }
        b.isEnabled = false
    }
    Timer.scheduledTimer(timeInterval: 1, target: self, selector: #selector(tornaEstatNormal),
        userInfo: nil, repeats: false)
}

/*
 *
 */
func tornaEstatNormal() {
    for b in bot_sol {
        b.backgroundColor = COLOR_FONS_NORMAL.withAlphaComponent(ALPHA_BOTONS)
        b.isEnabled = true
    }

    novaMultiplicacio()
}

/*
 *
 */
func decrementaTimeout() {
    if comptado_timeout > increment_comptado_timeout {
        comptado_timeout -= increment_comptado_timeout
        barra_temps.setProgress(comptado_timeout/TIME_OUT, animated: true)
    } else {
        marcaError()
        timeout_pregunta.invalidate()
        barra_temps.setProgress(0, animated: true)
    }
}
}

```

```

/*
 * Reprodueix audio.
 */
func reprodueixAudio(nom:String, player: inout AVAudioPlayer!) {
    print("Audio: \(nom)")

    guard let sound = NSDataAsset(name: nom) else {
        print("Recurs d'audio no trobat!!")
        return
    }
    do {
        try AVAudioSession.sharedInstance().setCategory(AVAudioSessionCategoryPlayback)
        try AVAudioSession.sharedInstance().setActive(true)
        player = try AVAudioPlayer(data: sound.data, fileTypeHint: AVFileTypeMPEGLayer3)
    } catch let error as NSError {
        print("Error de reproducció: \(error.localizedDescription)")
    }
}

/*
 *
 */
@IBAction func botoTornarMenu(_ sender: AnyObject) {
    timeout_pregunta.invalidate()
}
} // Fi classe

```

## 12.3 ViewControllerPass

```

//
// ViewControllerPass.swift
// MultipliCat
//
// Created by Francesc Roig i Feliu.
// Copyright © 2017 Francesc Roig i Feliu. All rights reserved.
//

import UIKit

class ViewControllerPass: UIViewController {

    @IBOutlet weak var etiPass1: UILabel!
    @IBOutlet weak var etiPass2: UILabel!
    @IBOutlet weak var etiPass3: UILabel!
    @IBOutlet weak var etiPass4: UILabel!

    @IBOutlet weak var boto0: UIButton!
    @IBOutlet weak var boto1: UIButton!
    @IBOutlet weak var boto2: UIButton!
    @IBOutlet weak var boto3: UIButton!
    @IBOutlet weak var boto4: UIButton!
    @IBOutlet weak var boto5: UIButton!
    @IBOutlet weak var boto6: UIButton!
    @IBOutlet weak var boto7: UIButton!
    @IBOutlet weak var boto8: UIButton!
    @IBOutlet weak var boto9: UIButton!
    @IBOutlet weak var botoC: UIButton!

    var boto = [UIButton]()
    var etiPass = [UILabel]()

    var posDigit = 0

    /*
     *
     */
    override func viewDidLoad() {
        super.viewDidLoad()
    }
}

```

```

self.view.backgroundColor = COLOR_FONS_PANTALLA

boto = [boto0, boto1, boto2, boto3, boto4, boto5, boto6, boto7, boto8, boto9]
etiPass = [etiPass1, etiPass2, etiPass3, etiPass4]

for b in boto {
    b.layer.borderColor = UIColor.white.cgColor
    b.layer.cornerRadius = 5
    b.layer.borderWidth = 1
}

for e in etiPass {
    e.layer.borderColor = COLOR_VERMELL.cgColor
    e.layer.cornerRadius = 5
    e.layer.borderWidth = 1
    e.layer.masksToBounds = true
}
}

/*
 *
 */
override func viewWillAppear(_ animated: Bool) {
    self.navigationController?.navigationBar.isHidden = false
}

/*
 *
 */
@IBAction func accioBotoTeclat(_ sender: UIButton) {

    let str_num = (sender.titleLabel!.text! as NSString)

    var pass_correcte = true

    let pass_c1 = password[password.characters.index(password.startIndex, offsetBy:
0)...password.characters.index(password.startIndex, offsetBy: 0)]
    let pass_c2 = password[password.characters.index(password.startIndex, offsetBy:
1)...password.characters.index(password.startIndex, offsetBy: 1)]
    let pass_c3 = password[password.characters.index(password.startIndex, offsetBy:
2)...password.characters.index(password.startIndex, offsetBy: 2)]
    let pass_c4 = password[password.characters.index(password.startIndex, offsetBy:
3)...password.characters.index(password.startIndex, offsetBy: 3)]

    switch posDigit {
    case 0:
        etiPass1.text = "*"
        if pass_c1 != str_num as String {
            pass_correcte = false
        }
    case 1:
        etiPass2.text = "*"
        if pass_c2 != str_num as String {
            pass_correcte = false
        }
    case 2:
        etiPass3.text = "*"
        if pass_c3 != str_num as String {
            pass_correcte = false
        }
    default:
        etiPass4.text = "*"
        if pass_c4 != str_num as String {
            pass_correcte = false
        }
        if pass_correcte {
            // Contrasenya correcte
            posDigit = -1
            performSegue(withIdentifier: "segue_config", sender: nil)
        } else {
            // Mostra el missatge d'error
            let alerta = UIAlertController(title: "Contrasenya incorrecte !!",
                message: "Si no recordes la contrasenya, l'única forma de
restituir" +
                " l'original (1234) és reinstal·lant l'app. Per a fer-
ho," +

```

```

        " cal esborris l'app del dispositiu i la tornis a" +
        " instal·lar des de l'App Store.",
        preferredStyle: UIAlertControllerStyle.alert)
    let accioDacord = UIAlertAction(title: "D'acord",
        style: UIAlertActionStyle.default)
    { (action) in
        self.posDigit = -1
        self.etiPass1.text = "-"
        self.etiPass2.text = "-"
        self.etiPass3.text = "-"
        self.etiPass4.text = "-"
        self.navigationController?.popToRootViewController(animated: true)
    }
    alerta.addAction(accioDacord)
    present(alerta, animated: true, completion: nil)
}
}
posDigit += 1
} // Fi "@IBAction func accioBotoTeclat()"
} // Fi classe

```

## 12.4 ViewControllerAjustos

```

//
// ViewControllerAjustos.swift
// MultipliCat
//
// Created by Francesc Roig i Felu.
// Copyright © 2017 Francesc Roig i Felu. All rights reserved.
//

import UIKit

class ViewControllerAjustos: UIViewController {

    @IBOutlet weak var botoResetDirecte: UIButton!
    @IBOutlet weak var botoResetInvers: UIButton!

    @IBOutlet weak var sliderTimeDir: UISlider!
    @IBOutlet weak var sliderTimeInd: UISlider!
    @IBOutlet weak var etiTimeOutDir: UILabel!
    @IBOutlet weak var etiTimeOutInd: UILabel!

    @IBOutlet weak var etiPuntsDir: UILabel!
    @IBOutlet weak var etiPuntsInd: UILabel!
    @IBOutlet weak var botMenu: UIButton!

    @IBOutlet weak var switchTaula1: UISwitch!
    @IBOutlet weak var switchTaula2: UISwitch!
    @IBOutlet weak var switchTaula3: UISwitch!
    @IBOutlet weak var switchTaula4: UISwitch!
    @IBOutlet weak var switchTaula5: UISwitch!
    @IBOutlet weak var switchTaula6: UISwitch!
    @IBOutlet weak var switchTaula7: UISwitch!
    @IBOutlet weak var switchTaula8: UISwitch!
    @IBOutlet weak var switchTaula9: UISwitch!
    @IBOutlet weak var switchTaula10: UISwitch!
    @IBOutlet weak var switchTaula11: UISwitch!
    @IBOutlet weak var switchTaula12: UISwitch!

    @IBOutlet weak var switchSoError: UISwitch!
    @IBOutlet weak var switchSoEncert: UISwitch!

    var switchTaula = [UISwitch]()

    @IBOutlet weak var textFieldPass: UITextField!

    @IBOutlet weak var botPuntsObj_D10: UIButton!
    @IBOutlet weak var botPuntsObj_D20: UIButton!
    @IBOutlet weak var botPuntsObj_D25: UIButton!
    @IBOutlet weak var botPuntsObj_D50: UIButton!
}

```

```

@IBOutlet weak var botPuntsObj_D75: UIButton!
@IBOutlet weak var botPuntsObj_D100: UIButton!
@IBOutlet weak var botPuntsObj_D150: UIButton!
@IBOutlet weak var botPuntsObj_D200: UIButton!
@IBOutlet weak var botPuntsObj_D250: UIButton!
@IBOutlet weak var botPuntsObj_D300: UIButton!
@IBOutlet weak var botPunts_D0: UIButton!

@IBOutlet weak var botPuntsObj_I10: UIButton!
@IBOutlet weak var botPuntsObj_I20: UIButton!
@IBOutlet weak var botPuntsObj_I25: UIButton!
@IBOutlet weak var botPuntsObj_I50: UIButton!
@IBOutlet weak var botPuntsObj_I75: UIButton!
@IBOutlet weak var botPuntsObj_I100: UIButton!
@IBOutlet weak var botPuntsObj_I150: UIButton!
@IBOutlet weak var botPuntsObj_I200: UIButton!
@IBOutlet weak var botPuntsObj_I250: UIButton!
@IBOutlet weak var botPuntsObj_I300: UIButton!
@IBOutlet weak var botPunts_I0: UIButton!

var botPuntsObj = [UIButton]()

/*
 *
 */
override func viewDidLoad() {
    super.viewDidLoad()

    self.view.backgroundColor = COLOR_FONS_PANTALLA

    // Desplaça la vista quan surt el teclat.
    NotificationCenter.default.addObserver(self, selector:
#selector(ViewControllerAjustos.keyboardWillShow(_:)), name: NSNotification.Name.UIKeyboardWillShow,
object: nil)
    NotificationCenter.default.addObserver(self, selector:
#selector(ViewControllerAjustos.keyboardWillHide(_:)), name: NSNotification.Name.UIKeyboardWillHide,
object: nil)

    switchTaula = [switchTaula1, switchTaula2, switchTaula3, switchTaula4, switchTaula5,
switchTaula6, switchTaula7, switchTaula8, switchTaula9, switchTaula10, switchTaula11, switchTaula12]

    for i in 0...11 {
        switchTaula[i].tintColor = TINT_COLOR
        switchTaula[i].onTintColor = COLOR_VERD_3
        if taules[i] {
            switchTaula[i].isOn = true
        } else {
            switchTaula[i].isOn = false
        }
    }

    switchSoError.tintColor = TINT_COLOR
    switchSoError.onTintColor = COLOR_VERD_3
    switchSoEncert.tintColor = TINT_COLOR
    switchSoEncert.onTintColor = COLOR_VERD_3

    let defaults = UserDefaults.standard

    soOnError = defaults.bool(forKey: "SoOnError")
    soOnEncert = defaults.bool(forKey: "SoOnEncert")

    if soOnError {
        switchSoError.isOn = true
    } else {
        switchSoError.isOn = false
    }
    if soOnEncert {
        switchSoEncert.isOn = true
    } else {
        switchSoEncert.isOn = false
    }

    etiTimeoutDir.backgroundColor = UIColor.white.withAlphaComponent(0)
    etiTimeoutInd.backgroundColor = UIColor.white.withAlphaComponent(0)

    var time_out_dir = defaults.integer(forKey: "TimeOutDir")
    if time_out_dir == 0 {
        time_out_dir = 30
    }
}

```

```

var time_out_ind = defaults.integer(forKey: "TimeOutInd")
if time_out_ind == 0 {
    time_out_ind = 30
}

let punts_dir = defaults.integer(forKey: "Punts1")
if punts_dir < 0 {
    etiPuntsDir.textColor = COLOR_VERMELL
    etiPuntsDir.text = "\\((-1*punts_dir))"
} else {
    etiPuntsDir.textColor = UIColor.black
    etiPuntsDir.text = "\\(punts_dir)"
}

let punts_ind = defaults.integer(forKey: "Punts2")
if punts_ind < 0 {
    etiPuntsInd.textColor = COLOR_VERMELL
    etiPuntsInd.text = "\\((-1*punts_ind))"
} else {
    etiPuntsInd.textColor = UIColor.black
    etiPuntsInd.text = "\\(punts_ind)"
}

sliderTimeDir.setValue(Float(time_out_dir/10)*2, animated: true)
etiTimeOutDir.text = "\\(time_out_dir) s"

sliderTimeInd.setValue(Float(time_out_ind/10)*2, animated: true)
etiTimeOutInd.text = "\\(time_out_ind) s"

botPuntsObj = [botPuntsObj_D10, botPuntsObj_D20, botPuntsObj_D25, botPuntsObj_D50,
botPuntsObj_D75, botPuntsObj_D100, botPuntsObj_D150, botPuntsObj_D200, botPuntsObj_D250,
botPuntsObj_D300, botPunts_D0, botPuntsObj_I10, botPuntsObj_I20, botPuntsObj_I25, botPuntsObj_I50,
botPuntsObj_I75, botPuntsObj_I100, botPuntsObj_I150, botPuntsObj_I200, botPuntsObj_I250,
botPuntsObj_I300, botPunts_I0]

for b in botPuntsObj {
    b.layer.borderColor = UIColor.white.cgColor
    b.layer.cornerRadius = 5
    b.layer.borderWidth = 1
}

etiPuntsDir.layer.borderColor = TINT_COLOR.cgColor
etiPuntsDir.layer.cornerRadius = 5
etiPuntsDir.layer.borderWidth = 1
etiPuntsDir.layer.masksToBounds = true

etiPuntsInd.layer.borderColor = TINT_COLOR.cgColor
etiPuntsInd.layer.cornerRadius = 5
etiPuntsInd.layer.borderWidth = 1
etiPuntsInd.layer.masksToBounds = true

botoResetDirecte.backgroundColor = TINT_COLOR
botoResetDirecte.setTitleColor(UIColor.white, for: UIControlState.normal)
botoResetDirecte.layer.borderColor = UIColor.white.cgColor
botoResetDirecte.layer.cornerRadius = 5
botoResetDirecte.layer.borderWidth = 1

botoResetInvers.backgroundColor = TINT_COLOR
botoResetInvers.setTitleColor(UIColor.white, for: UIControlState.normal)
botoResetInvers.layer.borderColor = UIColor.white.cgColor
botoResetInvers.layer.cornerRadius = 5
botoResetInvers.layer.borderWidth = 1
}

/*
 *
 */
override func viewWillAppear(_ animated: Bool) {
    self.navigationController?.navigationBar.isHidden = false
}

/*
 *
 */
override func viewWillDisappear(_ animated: Bool) {

```



```

    // Salta a la primera pantalla (es salta la pantalla se password).
    self.navigationController?.popToRootViewController(animated: false)
}

/*
 *
 */
@IBAction func accioBotoResetDirecte(_ sender: Any) {

    for i in 1...12 {
        encerts1[i] = 0
        errors1[i] = 0
    }

    let defaults = UserDefaults.standard
    defaults.set(encerts1, forKey: "Encerts1")
    defaults.set(errors1, forKey: "Errors1")

    //{ Mostra un missatge d'avís.
    let alerta = UIAlertController(title: "Reset estadístiques",
        message: "S'han inicialitzat les estadístiques referents al
'Mètode directe'.",
        preferredStyle: UIAlertControllerStyle.alert)
    let accioDacord = UIAlertAction(title: "Entesos", style: UIAlertActionStyle.default) { (action)
in }
    alerta.addAction(accioDacord)
    present(alerta, animated: true, completion: nil)
}

/*
 *
 */
@IBAction func accioBotoResetInvers(_ sender: Any) {

    for i in 1...12 {
        encerts2[i] = 0
        errors2[i] = 0
    }

    let defaults = UserDefaults.standard
    defaults.set(encerts2, forKey: "Encerts2")
    defaults.set(errors2, forKey: "Errors2")

    //{ Mostra un missatge d'avís.
    let alerta = UIAlertController(title: "Reset estadístiques",
        message: "S'han inicialitzat les estadístiques referents al
'Mètode invers'.",
        preferredStyle: UIAlertControllerStyle.alert)
    let accioDacord = UIAlertAction(title: "Entesos", style: UIAlertActionStyle.default) { (action)
in }
    alerta.addAction(accioDacord)
    present(alerta, animated: true, completion: nil)
}

/*
 *
 */
@IBAction func accioSwitchTaulas(_ sender: AnyObject) {

    //{ Comprova que com a mínim una taula està activa.
    var act = 0
    for i in 0...11 {
        if switchTaula[i].isOn {
            act += 1
        }
    }

    if act==0 {
        //{ Mostra un missatge d'alerta
        let alerta = UIAlertController(title: "Error seleccionant taules !!",
            message: "S'ha de seleccionar una taula com a mínim.",
            preferredStyle: UIAlertControllerStyle.alert)
        let accioDacord = UIAlertAction(title: "Entesos",
            style: UIAlertActionStyle.default) { (action) in }
    }
}

```

```

        alerta.addAction(accioDacord)
        present(alerta, animated: true, completion: nil)

        //{ Restaura posició de l'últim switch
        for i in 0...11 {
            if taules[i] {
                switchTaula[i].isOn = true
            } else {
                switchTaula[i].isOn = false
            }
        }
    } else {
        //{ Actualitza el vector taules amb els nous valors.
        for i in 0...11 {
            if switchTaula[i].isOn {
                taules[i] = true
            } else {
                taules[i] = false
            }
        }
        //{ Desa la configuració.
        let defaults = UserDefaults.standard
        defaults.set(taules, forKey: "TaulsActives")
    }
}

//***** Tipus 4 x 3 = ? *****/

/*
 *
 */
@IBAction func accioSliderTimeOutDir(_ sender: UISlider) {

    let seg = Int(10 * round(sender.value*10)/20)
    etiTimeOutDir.text = "\(seg) s"
}

/*
 *
 */
@IBAction func accioSliderFiDragDir(_ sender: UISlider) {

    let seg = Int(10 * round(sender.value*10)/20) //{ "0 perquè l'slider va de 1 a 12 (5 a 60s)

    let defaults = UserDefaults.standard
    defaults.set(seg, forKey: "TimeOutDir")
    print(seg)
}

/*
 *
 */
@IBAction func accioBotoResetDir(_ sender: AnyObject) {

    let defaults = UserDefaults.standard
    defaults.set(0, forKey: "Punts1")

    punts_minims_dir = 0
    defaults.set(punts_minims_dir, forKey:"PuntsMinimDir")

    etiPuntsDir.text = "0"
    etiPuntsDir.textColor = UIColor.black
}

/*
 * Acció dels botons del punts objectiu
 */
@IBAction func accioBotoPuntsObj_D(_ sender: UIButton) {

```

```

    let punts = (sender.currentTitle! as NSString).integerValue

    let defaults = UserDefaults.standard
    defaults.set(-1 * punts, forKey: "Punts1")

    punts_minims_dir = -1 * punts
    defaults.set(punts_minims_dir, forKey:"PuntsMinimDir")

    etiPuntsDir.text = "\(punts)"
    etiPuntsDir.textColor = COLOR_VERMELL
}

//***** Tipus 4 x ? = 12 *****/

/*
 *
 */
@IBAction func accioSliderTimeOutInd(_ sender: UISlider) {

    let seg = Int(10 * round(sender.value*10)/20)
    etiTimeOutInd.text = "\(seg) s"
}

/*
 *
 */
@IBAction func accioSliderFiDragInd(_ sender: UISlider) {

    let seg = Int(10 * round(sender.value*10)/20)
    let defaults = UserDefaults.standard
    defaults.set(seg, forKey: "TimeOutInd")
}

/*
 *
 */
@IBAction func accioBotoResetInd(_ sender: AnyObject) {

    let defaults = UserDefaults.standard
    defaults.set(0, forKey: "Punts2")

    punts_minims_dir = 0
    defaults.set(punts_minims_dir, forKey:"PuntsMinimInd")

    etiPuntsInd.text = "0"
    etiPuntsInd.textColor = UIColor.black
}

/*
 *
 */
@IBAction func accioBotoPuntsObj_I(_ sender: UIButton) {

    let punts = (sender.currentTitle! as NSString).integerValue

    let defaults = UserDefaults.standard
    defaults.set(-1 * punts, forKey: "Punts2")

    punts_minims_ind = -1 * punts
    defaults.set(punts_minims_ind, forKey:"PuntsMinimInd")

    etiPuntsInd.text = "\(punts)"
    etiPuntsInd.textColor = COLOR_VERMELL
}

// *****/

/**
 *
 */
@IBAction func accioTextFieldPass(_ sender: UITextField) {

```

```

let str_in = sender.text! as NSString

if str_in.length==4 {
    textFieldPass.resignFirstResponder()
    let num_pass = (sender.text! as NSString).integerValue

    let alerta = UIAlertController(title: "La nova contrasenya serà \"\(num_pass)\",
        message: "Segur que la vols canviar?",
        preferredStyle: UIAlertControllerStyle.alert
    )

    let accioNo = UIAlertAction(title: "No",
        style: UIAlertActionStyle.default) { (action) in }

    let accioSi = UIAlertAction(title: "Sí",
        style: UIAlertActionStyle.destructive)
    { (action) in
        let defaults = UserDefaults.standard
        password = str_in as String
        defaults.set(str_in, forKey: "Pass")
        // print("Nou password: \"\(password)")
    }

    alerta.addAction(accioNo)
    alerta.addAction(accioSi)
    present(alerta, animated: true, completion: nil)
}

}

/*
 *
 */
@IBAction func accioSwitchSoError(_ sender: UISwitch) {

    let defaults = UserDefaults.standard

    if sender.isOn {
        soOnError = true
        defaults.set(true, forKey:"SoOnError")
    } else {
        soOnError = false
        defaults.set(false, forKey:"SoOnError")
    }
}

/*
 *
 */
@IBAction func accioSwitchSoEncert(_ sender: UISwitch) {

    let defaults = UserDefaults.standard

    if sender.isOn {
        soOnEncert = true
        defaults.set(true, forKey:"SoOnEncert")
    } else {
        soOnEncert = false
        defaults.set(false, forKey:"SoOnEncert")
    }
}

/*
 *
 */
func keyboardWillShow(_ notification: Notification) {

    if let keyboardSize = (notification.userInfo?[UIKeyboardFrameBeginUserInfoKey] as?
    NSValue)?.cgRectValue {
        self.view.frame.origin.y -= keyboardSize.height
    }
}
}

```

```

/**
 *
 */
func keyboardWillHide(_ notification: Notification) {
    if let keyboardSize = (notification.userInfo?[UIKeyboardFrameBeginUserInfoKey] as?
    NSValue)?.cgRectValue {
        self.view.frame.origin.y += keyboardSize.height
    }
}
} //{ Fi classe

```

## 12.5 ViewControllerGrafic

```

//
// ViewControllerGrafic.swift
// MultipliCat
//
// Created by Francesc Roig i Feliu.
// Copyright © 2017 Francesc Roig i Feliu. All rights reserved.
//

import UIKit

class ViewControllerGrafic: UIViewController {

    @IBOutlet var barraGrafic: [UILabel]!
    @IBOutlet var etiBarraGrafic: [UILabel]!
    @IBOutlet weak var selectorMetode: UISegmentedControl!

    @IBOutlet weak var stackGrafic: UIStackView!

    var maxBarraGrafic:CGFloat!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.

        self.view.backgroundColor = COLOR_FONS_PANTALLA

        // Obtenim l'amplada màxima en pìxels del gràfic.
        maxBarraGrafic = barraGrafic[0].frame.width

        for (i, eti_barra) in etiBarraGrafic.enumerated() {
            if taules[i] {
                eti_barra.backgroundColor = COLOR_ROSA
            } else {
                eti_barra.backgroundColor = UIColor.clear
            }
            eti_barra.layer.borderColor = COLOR_VERMELL.cgColor
            eti_barra.layer.borderWidth = 1
            eti_barra.layer.cornerRadius = 5
            eti_barra.layer.masksToBounds = true
        }

        actualitzaGrafic(selector: 0)
    }

    /**
     *
     */
    override func viewWillAppear(_ animated: Bool) {
        self.navigationController?.navigationBar.isHidden = false
    }

    /**
     *
     */
    override func viewDidAppear(_ animated: Bool) {
        selectorMetode.selectedSegmentIndex = 0
    }
}

```

```

    actualitzaGrafic(selector: 0)
}

/*
 * Acció de selector de l'estadística a mostrar, mètode Directe o Invers.
 */
@IBAction func accioSelectorDirecteInvers(_ sender: UISegmentedControl) {
    actualitzaGrafic(selector: sender.selectedSegmentIndex)
}

/*
 * Actualitza el gràfic (0 -> directe, altrament -> invers)
 */
func actualitzaGrafic(selector: Int) {
    if selector == 0 {
        for num_barra in 1..12 {
            let total = errors1[num_barra] + encerts1[num_barra]
            if total > 0 && taules[num_barra-1] {
                let valor = Float(encerts1[num_barra])/Float(errors1[num_barra] +
encerts1[num_barra])
                setBarra(n_barra: num_barra-1, valor: valor)
            } else {
                setBarra(n_barra: num_barra-1, valor: 0)
            }
        }
    } else {
        for num_barra in 1..12 {
            let total = errors2[num_barra] + encerts2[num_barra]
            if total > 0 && taules[num_barra-1] {
                let valor = Float(encerts2[num_barra])/Float(errors2[num_barra] +
encerts2[num_barra])
                setBarra(n_barra: num_barra-1, valor: valor)
            } else {
                setBarra(n_barra: num_barra-1, valor: 0)
            }
        }
    }
}

/*
 * Modifica l'amplada de barra -> valor [0...1].
 */
func setBarra(n_barra: Int, valor: Float) {
    barraGrafic[n_barra].text = "\\(Int(valor*100)) %"
    barraGrafic[n_barra].backgroundColor = COLOR_FONS_BARRA_GRAFIC
    barraGrafic[n_barra].frame = CGRect(x: barraGrafic[n_barra].frame.origin.x, y:
barraGrafic[n_barra].frame.origin.y, width: CGFloat(valor) * maxBarraGrafic, height:
barraGrafic[n_barra].frame.height)
    // Es defineix la màscara de contorn per barres (UILabel).
    let forma = CAShapeLayer()
    forma.bounds = barraGrafic[n_barra].frame
    forma.position = barraGrafic[n_barra].center
    forma.path = UIBezierPath(roundedRect: barraGrafic[n_barra].bounds, byRoundingCorners:
[.bottomRight, .topLeft], cornerRadii: CGSize(width: 15, height: 15)).cgPath
    // S'aplica la màscara.
    barraGrafic[n_barra].layer.mask = forma
    // Es defineix el contorn
    let contorn = CAShapeLayer()
    contorn.path = forma.path
    contorn.fillColor = UIColor.clear.cgColor
    contorn.strokeColor = TINT_COLOR.cgColor
    contorn.lineWidth = 2
    contorn.frame = barraGrafic[n_barra].bounds
    // Esborra la capa anterior abans d'afegir la nova redimensionada.
    if let sl = barraGrafic[n_barra].layer.sublayers {
        for l in sl {
            l.removeFromSuperlayer()
        }
    }
}

```

```

    }
  }
  // Afegeix la nova capa del contron.
  barraGrafic[n_barra].layer.addSublayer(contorn)
}

} // Fi classe ViewControllerGrafic

```

## 12.6 AppDelegate

```

//
// AppDelegate.swift
// MultipliCat
//
// Created by Francesc Roig i Feliu.
// Copyright © 2017 Francesc Roig i Feliu.
//

import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    /*
     *
     */
    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.

        // Modifica els color per defecte del component UINavigationController per a tota l'app.
        UINavigationController.appearance().tintColor = TINT_COLOR
        UINavigationController.appearance().titleTextAttributes = [NSForegroundColorAttributeName:TINT_COLOR]

        return true
    }

    func applicationWillResignActive(_ application: UIApplication) {
        // Sent when the application is about to move from active to inactive state. This can occur for
        certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the
        user quits the application and it begins the transition to the background state.
        // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering
        callbacks. Games should use this method to pause the game.
    }

    func applicationDidEnterBackground(_ application: UIApplication) {
        // Use this method to release shared resources, save user data, invalidate timers, and store
        enough application state information to restore your application to its current state in case it is
        terminated later.
        // If your application supports background execution, this method is called instead of
        applicationWillTerminate: when the user quits.
    }

    func applicationWillEnterForeground(_ application: UIApplication) {
        // Called as part of the transition from the background to the active state; here you can undo
        many of the changes made on entering the background.
    }

    /*
     *
     */
    func applicationDidBecomeActive(_ application: UIApplication) {
        // Restart any tasks that were paused (or not yet started) while the application was inactive.
        If the application was previously in the background, optionally refresh the user interface.

        // Aguanta la pantalla LaunchScreen uns segons més.
        Thread.sleep(forTimeInterval: 4)
    }
}

```

```
func applicationWillTerminate(_ application: UIApplication) {  
    // Called when the application is about to terminate. Save data if appropriate. See also  
    applicationDidEnterBackground:  
}  
  
}
```