



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica en Informàtica de Gestió

**APLICACIÓ MULTIMÈDIA PER A PERSONES AMB
DISCAPACITAT VISUAL**

**Sara Cortés Viñals
Pere Barberan Agut**

TARDOR 2009

A la meva família

RESUM

El projecte *Aplicació multimèdia per a persones amb discapacitat visual* es tracta de crear una aplicació que permet a persones amb discapacitat visual reduïda o nul·la poder jugar a diferents activitats utilitzant una tauleta digital.

Aquestes activitats es caracteritzen per ser totalment configurables per l'usuari, ja que aquest pot afegir les imatges i els sons que posteriorment aniran a l'activitat. Cada activitat creada amb l'aplicació pot ser guardada i utilitzar-se tantes vegades com es desitgi. En concret, aquestes activitats són: Parelles, Busca Imatge, Resposta Correcte, Memori i Laberint.

Per tal de que l'usuari pugui interactuar amb els jocs utilitzant la tauleta digital, l'aplicació té l'opció de poder imprimir la pantalla per a poder adaptar-la en relleu i col·locar-la a sobre de la tauleta. D'aquesta manera l'usuari utilitzant el rastreig tàctil i l'oïda pot realitzar les diferents activitats.

Per a la realització d'aquesta aplicació s'ha utilitzat el llenguatge Visual Basic .Net i el Visual Studio 2008 Express Edition.

RESUMEN

El proyecto *Aplicació multimèdia per a persones amb discapacitat visual* se trata de crear una aplicación que permita a personas con discapacidad visual reducida o nula poder jugar a diferentes actividades usando una tableta digital.

Estas actividades se caracterizan por ser totalmente configurables por el usuario, ya que este puede añadir las imágenes y los sonidos que posteriormente irán en la actividad. Cada actividad creada con la aplicación puede ser guardada y utilizarse tantas veces como se desee. En concreto las actividades son: Parejas, Busca Imagen, Respuesta Correcta, Memori y Laberinto.

Para que el usuario pueda interactuar con los juegos utilizando la tableta digital, la aplicación tiene la opción de imprimir la pantalla para poder adaptarla en relieve y colocarla sobre la tableta. De esta manera el usuario utiliza el rastreo táctil y el oído para realizar las diferentes actividades.

Para la realización de esta aplicación se ha utilizado el lenguaje de programación Visual Basic .Net y el Visual Studio 2008 Express Edition

ABSTRACT

The project *Aplicació multimèdia per a persones amb discapacitat visual* consist on creating an application which allows blind and visually impaired people to play with different activities using a digital tablet.

The particularity of these activities is that they can be completely configured by the user, who can add images and sound. Each activity created can be saved and used the times they want. These activities are: Pairs, Find the Image, The Correct Answer, Memory and Laberinth.

The user can interact with the games using the digital tablet because the application has the possibility to print the screen. You can adapt the copy and have it in relief and then put it on the digital tablet. The user uses the tactile scanning and the hearing to do the different activities.

Visual Basic .Net language and Visual Studio 2008 Express Edition have been used to develop this application.

INDEX

1. Introducció.....	1
1.1 La Fundació el Maresme.....	2
1.2 Tecnologia i discapacitat.....	2
1.2.1 Tauleta digital.....	3
2. Objectius.....	5
2.1 Objectiu del projecte.....	5
2.2 Objectius personals.....	6
3. Antecedents.....	7
3.1 Jocs utilitzats per la Fundació.....	7
3.2 Programa <i>A Mida</i>	8
4. Entorn de desenvolupament.....	11
4.1 Visual Studio 2008 Express Edition.....	11
4.2 NSIS.....	13
5. Metodologia.....	15
6. Planificació.....	17
7. Anàlisi.....	19
7.1 Anàlisi de requisits.....	19
7.2 Especificació casos d'ús.....	23
7.2.1 Diagrama de casos d'ús.....	23
7.2.2 Casos d'ús detallats.....	24
8. Disseny.....	35
8.1 Disseny de la capa Presentació.....	36
8.2 Disseny de la capa de Domini.....	42
8.3 Disseny de la capa de Persistència.....	43
9. Implementació.....	45
9.1 Implementació de la capa de Domini.....	45
9.1.1 Classe Joc.....	45
9.1.2 Classe Fitxa.....	48
9.1.3 Classe Arxiu.....	49
9.1.4 Classe Parelles.....	50

9.1.5 Classe BuscaImatge.....	51
9.1.6 Classe RespostaCorrecte.....	51
9.1.7 Classe Pregunta.....	54
9.1.8 Classe Memori.....	55
9.1.9 Classe Laberint.....	58
9.2 Implementació de la capa de Persistència.....	59
9.3 Instal·lador.....	64
10. Pressupost.....	69
11. Possibles millores.....	71
12. Conclusions.....	73
ANNEX I: Manual d'usuari.....	75
ANNEX II: Contingut del CD.....	83
Bibliografia.....	85

INDEX DE FIGURES

Figura 1: Tauleta Digital.....	3
Figura 2: Plantilles adaptades.....	8
Figura 3: A Mida Configuració.....	9
Figura 4: A Mida Presentació.....	9
Figura 5: Framework .NET.....	12
Figura 6: Procés software.....	15
Figura 7: Taula planificació.....	18
Figura 8: Diagrama de casos d'ús.....	23
Figura 9: Model en capes.....	35
Figura 10: Pantalla Configuració Parelles.....	36
Figura 11: Pantalla Joc Parelles.....	37
Figura 12: Pantalla Configuració BuscaImatge.....	37
Figura 13: Pantalla Joc BuscaImatge.....	38
Figura 14: Pantalla Configuració RespostaCorrecte.....	38
Figura 15: Pantalla Afegir Pregunta.....	39
Figura 16: Pantalla Joc Resposta Correcte.....	39
Figura 17: Pantalla Configuració Memori.....	40
Figura 18: Pantalla Joc Memori.....	40
Figura 19: Pantalla Configuració Laberint.....	41
Figura 20: Pantalla Joc Laberint.....	41
Figura 21: Disseny del Domini.....	42
Figura 22: Disseny de Persistència.....	43
Figura 23: Classe Joc.....	45
Figura 24: Mètode afegirFitxa.....	46
Figura 25: Mètode clicarFitxa.....	47
Figura 26: Mètode comprovarParella.....	48
Figura 27: Classe Fitxa.....	48
Figura 28: Classe Arxiu.....	49
Figura 29: Classe Parelles.....	50
Figura 30: Mètode clicarFitxa Parelles.....	50

Figura 31: Classe BuscaImatge.....	51
Figura 32: Classe RespostaCorrecte.....	51
Figura 33: Mètode afegirPregunta.....	52
Figura 34: Mètode eliminarPregunta.....	52
Figura 35: Mètode clicarFitxa RespostaCorrecte.....	53
Figura 36: Classe Pregunta.....	54
Figura 37: Classe Memori.....	55
Figura 38: Mètode desordenarFitxes (Part 1).....	56
Figura 39: Mètode desordenarFitxes (Part 2).....	57
Figura 40: Mètode clicarFitxa Memori.....	58
Figura 41: Classe Laberint.....	58
Figura 42: Taula d'extensions.....	60
Figura 43: Classe GestorDades.....	60
Figura 44: Mètode guardarParelles.....	62
Figura 45: Mètode carregarParelles.....	63
Figura 46: Exemple de joc guardat.....	64
Figura 47: Codi instal·lador (inici).....	65
Figura 48: Codi instal·lador (secció obligatòria).....	66
Figura 49: Codi instal·lador (part opcional).....	67
Figura 50: Codi desinstal·lador.....	68
Figura 51: Taula Recursos Hardware.....	69
Figura 52: Taula Recursos Software.....	69
Figura 53: Taula Recursos Humans.....	69
Figura 54: Pressupost.....	69

1. Introducció

La idea del projecte *Aplicació multimèdia per a persones amb discapacitat visual* sorgeix de la Fundació el Maresme. Aquesta fundació és una entitat que treballa amb persones amb discapacitat de totes les edats. Aquest projecte va enfocat principalment a persones amb discapacitat intel·lectual i visió reduïda o nul·la, encara que pot ser utilitzat per altres col·lectius.

Es tracta de poder oferir diferents activitats per tal de que aquestes persones, mitjançant una tauleta digital, puguin interactuar amb elles. Aquestes activitats estan reforçades amb estímuls sonors i amb una plantilla impresa que anirà a sobre la tauleta per tal de que utilitzant el tacte i l'oïda puguin jugar.

Actualment la Fundació el Maresme està utilitzant activitats que es caracteritzen per:

- Tenir so, de manera que quan hi ha un encert s'informa auditivament, i paral·lelament hi ha estímuls sonors indicadors de la interacció.
- Són jocs estàtics, és a dir, els elements no es mouen, això permet transformar allò que es veu a la pantalla en una plantilla en relleu.
- Conservar al màxim la configuració. S'utilitzen activitats que permeten ser descarregades, per tal de conservar les diferents configuracions.

Els objectius d'aquestes activitats són per a treballar la memòria, el rastreig tàctil i l'aprenentatge de nous conceptes.

Dins d'aquestes activitats es treballen:

- El manteniment de les competències bàsiques.
- Centres d'interès i temàtiques diverses.
- Ampliació de vocabulari.
- L'autonomia personal i social.

Les eines per a treballar aquestes i altres àrees són diverses, però cada vegada s'utilitza més l'ús de les noves tecnologies.

A part de l'ordinador, predomina l'ús de material sobre paper (fitxes), fet que limita l'accés als usuaris que tenen dèficits visuals. És per això que la Fundació aposta per l'ús de la tauleta digital. Actualment, el temps que han de dedicar a la cerca i adaptació de les activitats on-line per tal que siguin d'utilitat per a ells és molt elevat.

És per aquest motiu que sorgeix la idea de la creació d'aquest projecte.

1.1. La Fundació el Maresme

La Fundació Privada el Maresme Pro Persones amb disminució psíquica és una entitat d'iniciativa social sense ànim de lucre nascuda l'any 1966 amb l'objectiu de promoure i impulsar la integració social de les persones amb discapacitat psíquica. El seu àmbit d'actuació és la comarca del Maresme.

Actualment la Fundació el Maresme atén, en els diferents centres i serveis que componen la xarxa assistencial que van des de la primera infància fins a l'edat adulta, un col·lectiu de més de 1.500 persones.

1.2. Tecnologia i discapacitat

L'ús de l'ordinador depèn molt dels perifèrics d'entrada. Els més habituals són el teclat i el ratolí als quals s'han adaptat la major part dels programes.

En el desenvolupament d'aquests perifèrics no sempre s'han aplicat els criteris del disseny per a tots, i en conseqüència, hi ha grups d'usuaris que tenen dificultats per a poder utilitzar-los. En aquest cas, les persones amb discapacitat visual necessiten adaptacions per tal de poder utilitzar els ordinadors. Algunes d'aquestes adaptacions existents actualment són:

- Lectors de pantalla:

Els lectors de pantalla són aplicacions per identificar i interpretar allò que es mostra a la pantalla. Aquesta interpretació es representa a l'usuari mitjançant sintetitzadors de text a veu, o amb una sortida braille.

- Magnificadors de pantalla

Els magnificadors de pantalla són programes per a la accessibilitat que permeten ampliar els caràcters i configurar els colors depenent de la necessitat de l'usuari.

1.2.1. Tauleta Digital

Les tauletes digitals o tauletes gràfiques són perifèrics que permeten a l'usuari introduir gràfics o dibuixos a mà, tal com ho faria amb un llapis i un paper. També permet apuntar i senyalar els objectes que es troben a la pantalla.

Consisteix en una superfície plana sobre la qual l'usuari pot dibuixar una imatge utilitzant el llapis que ve junt amb la tauleta.



Figura 1: Tauleta Digital.

Tot i no ser eines dissenyades per a l'ús específic de persones amb discapacitat visual, poden resultar molt útils, ja que permeten afegir una plantilla en relleu a sobre de la tauleta per tal de que l'usuari pugui treballar amb el tacte i el rastreig.

Treballant amb la tauleta, les persones a qui va dirigit aquest projecte poden gaudir de les mateixes activitats que la resta del grup. Però, a més a més, els companys poden gaudir d'aquestes activitats, sense tenir problemes de visió, ja que la tauleta i el material que es dissenya per utilitzar-la és un recurs pedagògic ric en estímuls tàctils que faciliten la interacció amb l'ordinador.

2. Objectius

2.1. Objectius del projecte

L'objectiu principal d'aquest projecte és oferir una aplicació que permeti als educadors inserir una sèrie d'imatges i sons dins d'una plantilla estàndard. Aquesta plantilla s'ha de poder imprimir per tal d'adaptar-la físicament amb relleu i posteriorment col·locar-la a sobre de la tauleta.

D'aquesta manera es permet crear plantilles tàctils de manera pràctica, sempre amb la mateixa estructura i a la vegada familiars per a la persona que ha d'interactuar amb elles.

L'aplicació ha de crear diferents activitats que siguin totalment configurables per l'usuari. Concretament, els entorns multimèdia que ha de tenir l'aplicació són els següents:

- Activitat 1: Dos espais ben definits amb quatre graelles cadascun en les que es pugui inserir imatges i sons. L'activitat a realitzar amb aquest entorn és la d'associar una graella d'un espai amb la de l'altre.
- Activitat 2: Dos espais ben definits, un amb única graella i l'altre amb quatre graelles. L'activitat seria relacionar la graella d'un espai amb la de l'altre.
- Activitat 3: Tres graelles en les que es puguin inserir imatges, sons. Es sentirà una pregunta i s'haurà de triar la resposta entre les que es plantegen a les tres graelles.
- Memori: Poder inserir imatges i sons, per parelles, relacionades amb qualsevol tema. Ha d'haver-hi la possibilitat de triar el número de graelles per jugar.
- Laberints: Crear un entorn multimèdia per practicar el rastreig amb les mans. Però ha de permetre la configuració d'un fil conductor lligat a una història, per fer-lo el més estimulants possible.

Totes les creacions fetes amb l'aplicació han de tenir l'opció de poder ser guardades, per poder utilitzar-les o modificar-les posteriorment.

2.2. Objectius personals

Quan se'm va oferir la possibilitat de realitzar aquest projecte em va semblar una molt bona idea, ja que no només es tractava de realitzar un treball fi de carrera, sinó que es tractava de desenvolupar una eina que tenia una finalitat i que més endavant molta gent li podria treure profit.

Aquest ha sigut un dels principals motius a l'hora de triar aquest projecte. El meu objectiu principal era crear una aplicació que compleixi tots els requisits demanat i sobretot que sigui útil per aquestes persones amb discapacitat.

En quan al tema tecnològic, Un altre objectiu personal d'aquest projecte era aprendre un llenguatge de programació, com el Visual Basic .Net, que potser si no hagués fet aquest projecte no l'hauria d'haver utilitzat i no n'hagués après.

3. Antecedents

Actualment la Fundació el Maresme treballa amb diferents activitats cercades on-line i les adapta per a poder treballar amb elles. A continuació es mostraran algunes d'aquestes activitats, i també el programa *A Mida*.

3.1. Jocs utilitzats per la Fundació

Els educadors de la Fundació utilitzen diferents jocs que no han estat dissenyats expressament per a persones amb deficiència visual, i s'adapten imprimint manualment la pantalla i adaptant aquesta amb relleu. Alguns dels jocs que utilitzen són els següents:

- Conte SPC Sant Jordi. Activitats: Joc de laberints on el ratolí és una rosa que has de guiar-la dins del laberint fins arribar a la princesa.
Web: http://www.edu365.cat/primaria/contes/contes_spc/jordi/index.htm
- Childtopia. Comprensió oral: Histories narrades amb preguntes i tres respostes a triar. Web:
<http://childtopia.com/index.php?module=home&func=coce&myidioma=spa&idphpx=juegos-educativos>
- Memori: Joc del memori on a l'adaptació que han realitzat s'ha de donar relleu als quadrats, i a més enumerar cada requadre per a que l'usuari tingui més pistes.
<http://educacion.once.es/>
- Piano i números: Piano en el qual s'imprimeix la imatge i amb la tauleta es pot tocar. Web: <http://www.poissonrouge.com>



Figura 2: Plantilles adaptades.

3.2. Programa A Mida

L'aplicació *A Mida* és l'aplicació que vaig realitzar com a projecte final de carrera de Eng. Tècnica de Telecomunicacions especialitzada en Telemàtica. Aquest també va ser un projecte desenvolupat a petició de la Fundació el Maresme, i estava enfocat principalment a nens amb discapacitat física i psíquica.

El programa *A Mida* consisteix en una aplicació multimèdia que consta de sis caselles on s'hi pot afegir imatges, sons, vídeos i webs. Utilitzant polsadors i la tècnica de l'encerclament els nens que utilitzen aquesta aplicació podran accedir als continguts que ells desitgin afegits anteriorment, acte que, utilitzant l'ordinador de manera convencional els hi és impossible.

El llenguatge utilitzat per a desenvolupar l'aplicació va ser el Visual Basic .Net. El principal motiu per a triar aquest llenguatge va ser que era necessària la creació d'un navegador web (per tal de buscar continguts webs que es volguessin afegir) i en l'estudi que es va fer de possibles llenguatges a utilitzar, Visual Basic .Net ho permetia fer d'una manera senzilla. A més, tota la programació de la part visual es pot realitzar de manera força intuïtiva, cosa que fa que no hagi de dedicar-hi més temps del necessari a aquesta part del projecte.

Introdueixo aquest projecte com precedent ja que he decidit seguir utilitzant el mateix entorn de treball que en el cas anterior, per tal de poder reutilitzar codi del projecte i aprofitar-ne l'aprenentatge obtingut.

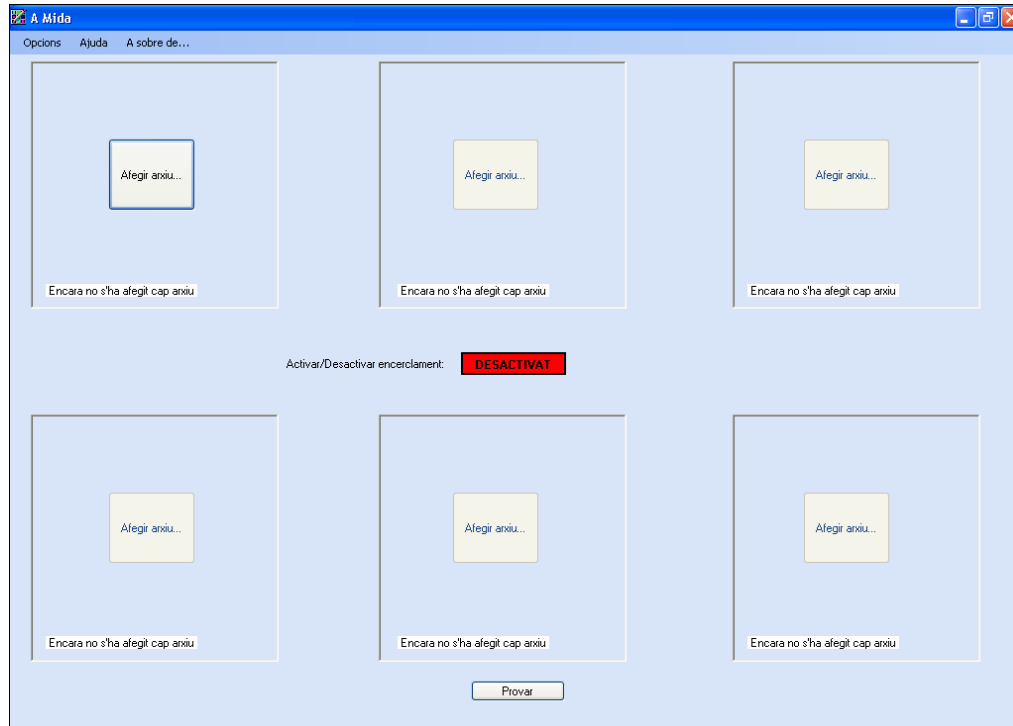


Figura 3: A Mida Configuració.

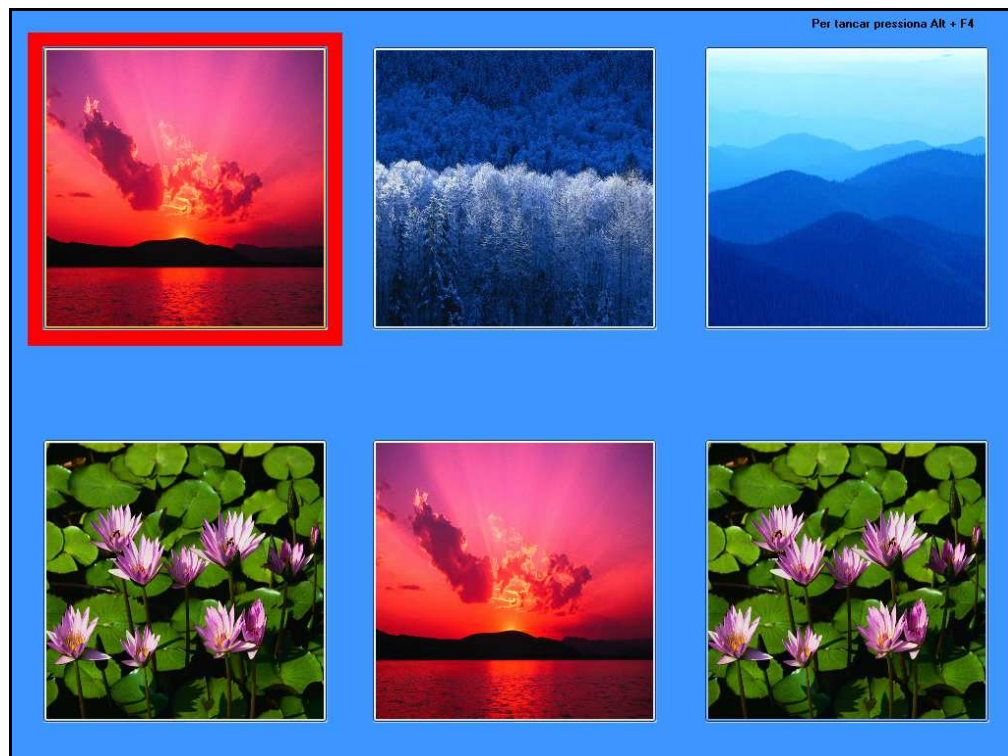


Figura 4: A Mida Presentació.

4. Entorn de desenvolupament

Com s'ha comentat anteriorment s'ha decidit seguir amb l'entorn de desenvolupament del projecte anterior, que ha estat el Visual Basic .net. i el Visual Studio 2008 Express Edition.

4.1 Visual Studio 2008 Express Edition

Visual Basic és una llenguatge molt visual que permet programar interfícies gràfiques amb facilitat. Per les necessitats d'aquest projecte és necessari un entorn de desenvolupament que permeti programar gràficament de manera àgil ja que no es tracta d'una aplicació que hagi de realitzar molts processos complexos.

Un altre motiu per el qual s'ha decidit treballar amb Visual Basic 2008 Express Edition, és que aquesta edició, enfocada a estudiants, és totalment gratuïta, per tant no cal adquirir llicència.

La única limitació que té al ser una edició Express és que no permet realitzar l'instal·lador. Aquesta limitació no és problemàtica ja que existeixen en el mercat diverses eines que permeten fer-ho.

Per a crear una aplicació en Visual Basic, s'han de realitzar tres passes principals.

1. Dissenyar la interfície. S'ha de dissenyar l'estructura dels formularis, col·locant en ell els controls i objectes necessaris.
2. Donar valors a les propietats dels objectes col·locats al formulari.
3. I per últim escriure el codi als objectes per a indicar les accions i events que es realitzaran quan l'usuari interaccioni amb aquests objectes. Aquest pas és conegut com el de donar funcionalitat a l'aplicació.

Per a realitzar aquesta aplicació s'ha utilitzat com a eina de desenvolupament el Visual Studio 2008 i .NET Framework 3.5. Visual Studio 2008 és un IDE que permet realitzar aplicacions basades en Windows.

.NET Framework és un entorn per a construir, instal·lar i executar serveis web i altres aplicacions. Es compon de tres parts:

- El Common Language Runtime (CLR)
- Les classes Framework
- Motor de generació de formularis, per a crear les nostres aplicacions.

Qualsevol executable, depèn d'una forma o altre d'una sèrie de llibreries, ja siguin en temps d'execució o en temps de compilació. Doncs el CLR es això, una sèrie de llibreries utilitzades en temps d'execució per a que els nostres executables o qualsevol basats en .NET puguin funcionar.

Per una altra banda, la biblioteca de classes .NET Framework proporciona una jerarquia de classes orientades a objectes que estan disponibles per a qualsevol dels llenguatges basats en .NET com per exemple el Visual Basic. Ofereix una visió orientada a objectes del sistema operatiu Windows i inclou centenars de classes que encapsulen els objectes més importants del nucli de Windows. D'aquesta manera podem accedir a una sèrie de funcionalitats del sistema operatiu.

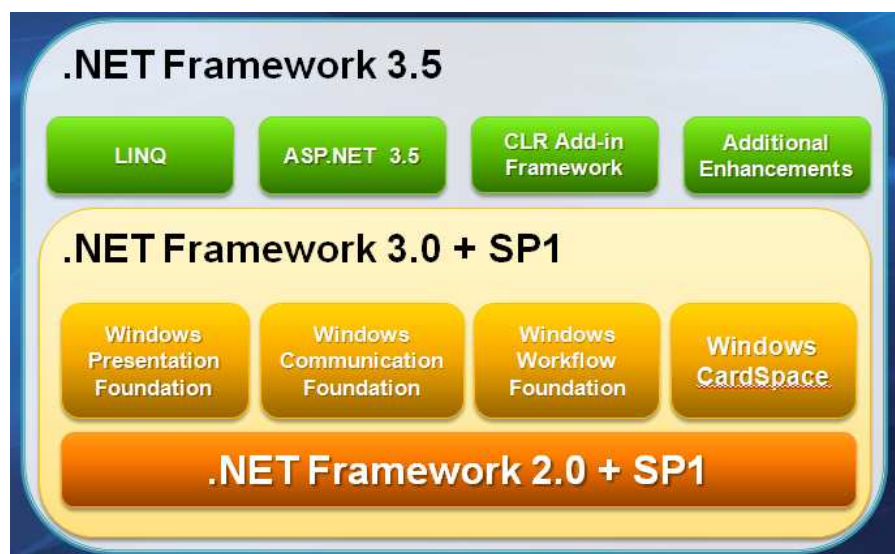


Figura 5: Framework .NET

4.2 NSIS

Com s'ha mencionat a l'apartat anterior, amb el Visual Basic 2008 Express Edition no es pot programar l'instal·lador. L'eina triada per realitzar-lo ha sigut NSIS (NullSoft Scriptable Install System), un programa OpenSource que mitjançant un script permet crear l'instal·lador del programa amb facilitat. NSIS és un software usat àmpliament cosa que facilita la cerca d'exemple i tutorials a la web.

5. Metodologia

La metodologia que s'utilitzarà per al desenvolupament del projecte està basada en la programació seqüencial.

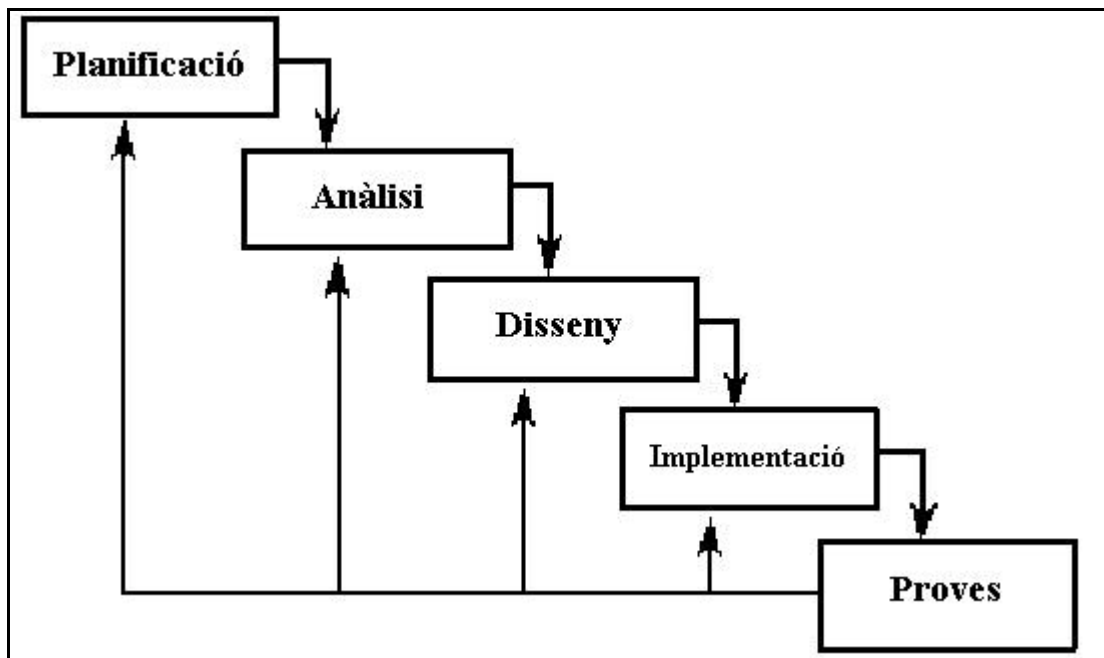


Figura 6: Procés software.

- **Fase de planificació:**

En aquesta fase es farà una planificació temporal del projecte, indicant en quin període es farà cada activitat, i quantes hores es preveuen per a la seva realització.

- **Fase d'anàlisi**

És aquí on es farà l'anàlisi del projecte per tal de tenir ben definides totes les funcionalitats que haurà de tenir l'aplicació. En aquesta fase s'obtidran els requisits ha desenvolupar posteriorment.

- **Fase de disseny**

En aquesta fase es convertirà l'anàlisi de requisits obtingut anteriorment en un diagrama de classes del qual partir a l'hora de començar la implementació. També en aquesta etapa es farà el disseny de la part de persistència, i el disseny de les interfícies gràfiques d'usuari.

- **Fase d'implementació**

Es portarà a terme la programació del projecte tenint en compte tot l'esmentat anteriorment.

Tot i tractar-se d'una metodologia seqüencial, es pot dir que també s'ha aplicat la metodologia iterativa. De les reunions que es tenien periòdicament amb la Fundació el Maresme sorgien noves funcionalitats i canvis en els requeriments inicials, que s'han anat desenvolupant al llarg del projecte.

6. Planificació

Una part molt important de qualsevol projecte és la seva planificació, per tal de deixar clar abans d'iniciar el desenvolupament, els passos que s'hauran de seguir per poder complir els objectius esmentats.

S'ha previst un total de 320 hores per a la realització del projecte. Aquestes hores s'assignaran a cada etapa del projecte. Les etapes en les que s'ha dividit són les següents:

Estudi previ:

- Reunió amb el tutor del projecte i la Fundació el Maresme per a fixar els objectius del projecte.
- Estudi d'antecedents i familiarització amb els recursos que actualment utilitzen. Adquisició de les eines necessàries per a la realització del projecte, en aquest cas, la tauleta digital.
- Decidir la metodologia que s'utilitzarà i l'entorn de desenvolupament més adequat, i instal·lació del software necessari. (En aquest cas, no ha sigut necessari ja que s'ha decidit seguir utilitzant el mateix entorn de treball que en el projecte anterior).
- Si fos necessari, fase d'aprenentatge de noves tecnologies.

Anàlisi:

- Anàlisi dels requisits del projecte.

Disseny:

- Disseny del domini, persistència i interfície d'usuari.

Implementació:

- Implementació de l'aplicació, durant aquesta etapa es realitzaran varies reunions amb la Fundació el Maresme, per tal de que puguin veure l'evolució

de l'aplicació, i així poder tenir la informació necessària per si sorgeix algun canvi en els requisits o alguna millora.

- Finalment la creació de l'instal·lador amb la versió final del programa.

Proves:

- Fase de proves.

Documentació:

- Redacció de la memòria, que es farà paral·lelament amb el desenvolupament de l'aplicació.
- Redacció d'un manual d'usuari que anirà junt amb l'aplicació.

Per tant, el temps en hores que és dedicarà a cada etapa del projecte és el que es mostra a la següent taula:

ETAPA	HORES
Estudi previ	32
Anàlisi	44
Disseny	64
Implementació	88
Proves	20
Documentació	72

Figura 7: Taula planificació.

7. Anàlisi

7.1 Anàlisi de requisits

Per tal de poder realitzar el projecte primer s'ha de fer un estudi dels requeriments que són necessaris per cobrir tota la funcionalitat demanada. Aquests requisits han estat obtinguts a partir de les reunions que s'han fet amb els responsables del projecte a la Fundació el Maresme.

L'aplicació constarà de dos perfils d'usuaris diferents. Per una part l'usuari encarregat d'afegir tots els continguts als jocs, i per l'altre, l'usuari final que serà qui jugui a aquests jocs. Per tant, tots els jocs tindran dues parts: la part de configuració, i la part de joc.

L'usuari encarregat de la part de configuració serà una persona sense problemes de visió, per tant no caldrà cap tipus d'adaptació i podrà utilitzar l'ordinador de manera convencional.

L'usuari final, haurà de poder jugar amb els jocs creats mitjançant la plantilla impresa i la tauleta digital. Aquesta plantilla impresa s'haurà d'adaptar a la mida de la tauleta digital proporcionada per la Fundació el Maresme.

Les característiques concretes que haurà de tenir cada joc o activitat són les següents:

- **Activitat 1:**

Aquesta activitat constarà de dos espais ben definits amb quatre graelles cadascun. L'usuari en mode configuració ha de poder afegir per a cada graella una imatge, i un so, tenint també l'opció de que alguna o totes les graelles només tinguin imatges. L'usuari haurà d'informar de quina és la relació entre graelles. Un cop creada l'activitat es tindrà la opció de guardar-la, i també la de carregar-ne alguna d'existent. També ha de tenir l'opció d'imprimir la plantilla.

En el mode de joc, l'usuari final utilitzant la plantilla impresa prèviament i la tauleta digital podrà seleccionar cada botó del joc i intentar relacionar-lo amb el de l'altre espai. Segons com hagi estat la configuració del joc, cada vegada que premi un botó haurà de sonar el so associat a aquella graella, i un cop l'usuari hagi seleccionat dues graelles, l'aplicació haurà d'informar de forma sonora si ha encertat l'associació o si ha estat incorrecte.

Quan l'usuari hagi completat totes les associacions l'aplicació haurà d'informar de forma sonora que ha acabat el joc.

A aquesta activitat l'he anomenat *Parelles*.

- **Activitat 2:**

El funcionament d'aquest joc és molt semblant a l'activitat anterior, el canvi és que en aquest cas en un dels espais només tens una graella i en l'altre en tens quatre. El funcionament en mode de configuració és exactament igual que l'activitat *Parelles* únicament que aquest cas només constarà d'una associació.

En el mode de joc, haurà de constar d'un botó per a poder triar entre jugar o només escoltar els sons, ja que al tractar-se d'una activitat amb només una associació podria ser que per atzar triessin a la primera tirada la resposta correcte. Per tant l'usuari podrà escollir entre seleccionar una graella o només escoltar el so que aquest té associat.

Aquesta activitat la he anomenat *Busca Imatge*.

- **Activitat 3:**

En aquest cas es tractarà de fer com un test, on l'usuari en mode configuració podrà afegir una història en un so, i a partir d'aquí afegir preguntes relacionades amb aquesta història.

Aquestes preguntes constaran d'un text amb un so associat, que serà la pregunta, i tres possibles respostes, que seran tres graelles on s'hi podrà inserir imatges i sons. També haurà d'indicar quina graella és la resposta correcte. L'usuari podrà afegir tantes preguntes com desitgi.

En aquest cas es pot imprimir una plantilla i utilitzar-la per a totes les preguntes, o imprimir cada pregunta com a plantilla única, això ho pot escollir l'usuari.

En el mode joc, l'usuari primer escoltarà la història i posteriorment sentirà la primera pregunta i tindrà la opció (com en el cas anterior) de només escoltar les possibles respostes, o de seleccionar-ne una. L'usuari podrà passar d'una pregunta a la següent sempre que vulgui, i tornar a enrere per respondre alguna que no hagués contestat.

Aquesta activitat l'he anomenat *Resposta Correcte*.

- **Memori:**

En el mode configuració s'haurà de poder escollir el nivell de dificultat a jugar, amb un màxim de 6 parelles. L'usuari ha d'afegir una imatge i un so per a cada parella (només una vegada per parella). En aquest cas no caldrà definir associacions, són automàtiques, cada casella està associada a ella mateixa.

En el mode de joc, l'usuari haurà d'escollir dues caselles, que al ser seleccionades es faran visibles i s'escoltarà el so que s'hagi afegit. Si les dues parelles són correctes quedaran descobertes i s'haurà de sentir un so que indiqui que ha estat correcte. Si el usuari torna a prémer una casella que ja ha estat descoberta s'haurà d'indicar mitjançant un so.

Aquest joc s'anomena *Memori*.

- **Laberint:**

L'activitat constà de 5 nivells de dificultat que van augmentant conforme es finalitza el nivell anterior.

En aquest tipus d'activitat l'usuari que la configuri haurà de afegir una història per tal de servir com a fil conductor del joc, també haurà d'afegir la imatge del personatge que vol que recorri el laberint. Es des d'aquí on es podrà seleccionar el laberint que es vol imprimir.

En el mode de joc, l'usuari mitjançant la tauleta haurà de seguir el recorregut del laberint, un cop arribi al final d'aquest, l'aplicació haurà de reproduir un avís sonor, i passar al següent nivell de dificultat.

Aquest joc s'anomena *Laberint*.

Totes les activitats hauran de tenir l'opció d'afegir un títol al joc, amb text, amb imatge i amb so. També s'han de poder guardar i carregar totes les activitats, ja sigui des de la pantalla de configuració com obrint l'arxiu que s'ha guardat directament.

7.2. Especificació de casos d'ús.

7.2.1. Diagrama de casos d'ús.

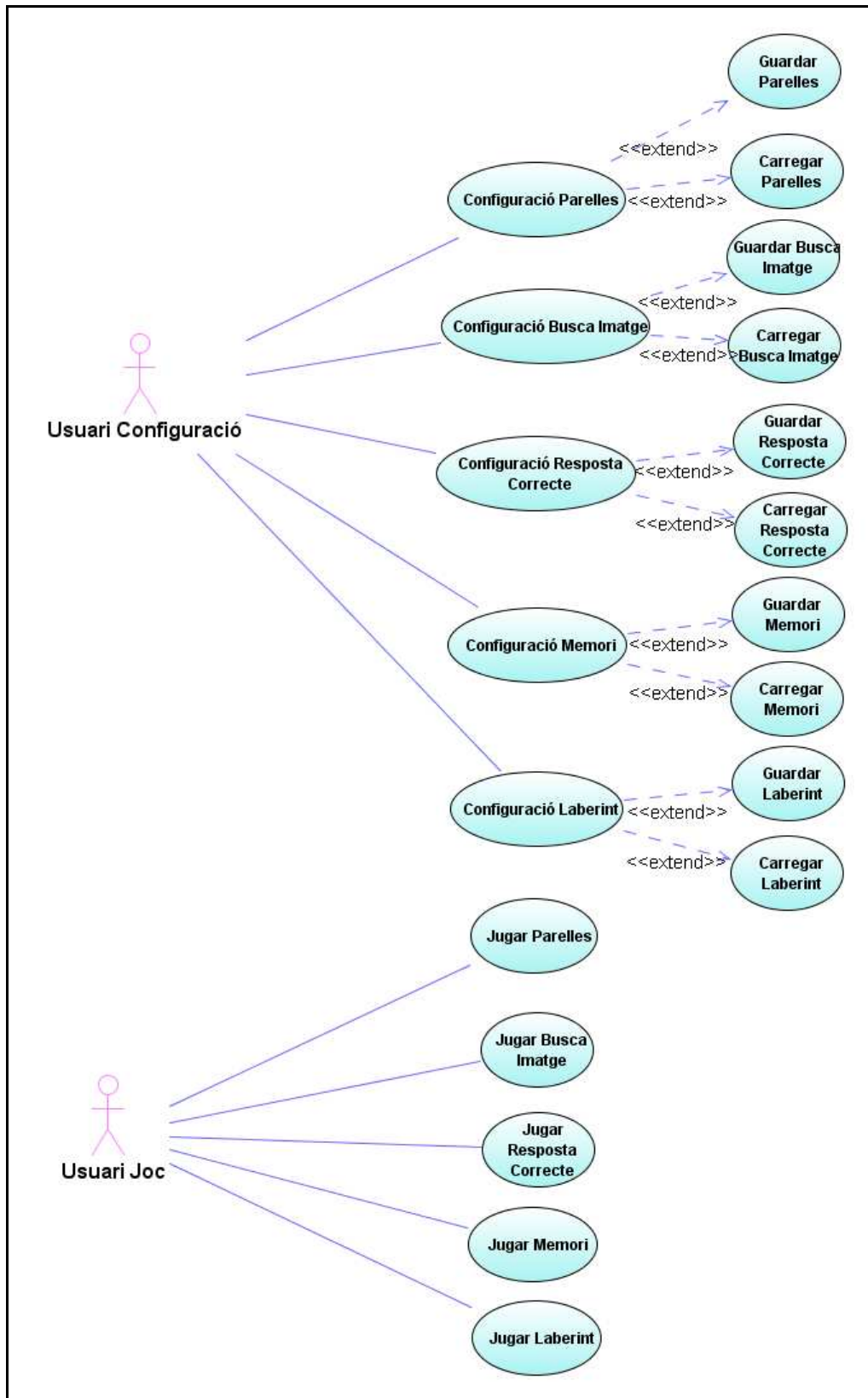


Figura 8: Diagrama de casos d'ús.

7.2.2. Casos d'ús detallats.

- **Cas d'ús Configuració Parelles:**

Actor: Usuari Configuració

Descripció: L'usuari afegeix el contingut que desitgi per crear un joc del tipus Parelles.

Flux normal:

1. L'usuari afegeix un títol al joc.
2. L'usuari afegeix una imatge per a cada casella.
3. El sistema mostra la imatge i el nom a la casella.
4. L'usuari afegeix un so per a cada casella. (opcional)
5. El sistema mostra el nom del so a la casella.
6. L'usuari associa dues caselles.
7. L'usuari prem el botó per iniciar el joc.
8. El sistema comprova que el joc es correcte.
9. S'inicia el cas d'ús *Jugar Parelles*.

Flux alternatiu:

- 1a. L'usuari carrega un joc existent.
 - 1.1a. El sistema carrega el joc en la pantalla de configuració.
 - 1.2a. L'usuari pot modificar el joc (pas 2) o seguir al pas 7 o el 7b.
- 7b. L'usuari tria guardar el joc.
 - 7.1.b. El sistema guarda el joc.
 - 7.2.b. Segueix al pas 7 o l'usuari tanca l'aplicació i finalitza el cas d'ús.
- 8c. Alguna dada introduïda és errònia
 - 8.1c. El sistema avisa de l'error.

Postcondicions: Si el joc s'ha guardat, el sistema ha de crear una carpeta amb totes les dades i l'arxiu del joc.

• **Cas d'ús Configuració Busca Imatge:**

Actor: Usuari Configuració

Descripció: L'usuari afegeix el contingut que desitgi per crear un joc del tipus Busca Imatge.

Flux normal:

1. L'usuari afegeix un títol al joc.
2. L'usuari afegeix una imatge per a cada casella.
3. El sistema mostra la imatge i el nom a la casella.
4. L'usuari afegeix un so per a cada casella. (opcional)
5. El sistema mostra el nom del so a la casella.
6. L'usuari associa la casella de l'esquerra amb una de les de la dreta.
7. L'usuari prem el botó per iniciar el joc.
8. El sistema comprova que el joc es correcte.
9. S'inicia el cas d'ús *Jugar Busca Imatge*.

Flux alternatiu:

- 1a. L'usuari carrega un joc existent.
 - 1.1a. El sistema carrega el joc en la pantalla de configuració.
 - 1.2a. L'usuari pot modificar el joc (pas 2) o seguir al pas 7 o el 7b.
- 7b. L'usuari tria guardar el joc.
 - 7.1.b. El sistema guarda el joc.
 - 7.2.b. Segueix al pas 7 o l'usuari tanca l'aplicació i finalitza el cas d'ús.
- 8c. Alguna dada introduïda és errònia
 - 8.1c. El sistema avisa de l'error.

Postcondicions: Si el joc s'ha guardat, el sistema ha de crear una carpeta amb totes les dades i l'arxiu del joc.

• **Cas d'ús Configuració Resposta Correcte:**

Actor: Usuari Configuració

Descripció: L'usuari afegeix el contingut que desitgi per crear un joc del tipus Resposta Correcte.

Flux normal:

1. L'usuari afegeix un títol al joc.
2. L'usuari afegeix un so com a història.
3. El sistema mostra el nom del so afegit com a història.
4. L'usuari prem el botó per afegir nova pregunta.
5. El sistema mostra la pantalla per afegir preguntes.
6. L'usuari afegeix un text per a la pregunta i un so.
7. L'usuari afegeix tres possibles respostes amb imatge i so.
8. L'usuari indica al sistema quina és la resposta correcte.
9. L'usuari prem el botó guardar pregunta i el sistema la guarda.
10. El sistema mostra a la pantalla de configuració un llistat de les preguntes afegides.
11. L'usuari prem el botó per iniciar el joc.
12. El sistema comprova que el joc es correcte.
13. S'inicia el cas d'ús *Jugar Resposta Correcte*.

Flux alternatiu:

* Es repetiran els passos del 4 al 9 tantes vegades com l'usuari desitgi.

- 1a. L'usuari carrega un joc existent.
 - 1.1a. El sistema carrega el joc en la pantalla de configuració.
 - 1.2a. L'usuari pot modificar el joc (pas 2) o seguir al pas 7 o el 7b.
- 10b. L'usuari prem el botó per modificar una pregunta existent
 - 10.1b. El sistema torna al pas 5.
- 10c. L'usuari prem el botó per esborrar una pregunta existent
 - 10.1c. El sistema borra la pregunta seleccionada.
- 11d. L'usuari tria guardar el joc.
 - 11.1d. El sistema guarda el joc.

11.2d. Segueix al pas 11 o l'usuari tanca l'aplicació i finalitza el cas d'ús.

12e. Alguna dada introduïda és errònia

12.1e. El sistema avisa de l'error.

Postcondicions: Si el joc s'ha guardat, el sistema ha de crear una carpeta amb totes les dades i l'arxiu del joc.

• **Cas d'ús Configuració Memori:**

Actor: Usuari Configuració

Descripció: L'usuari afegeix el contingut que desitgi per crear un joc del tipus Memori.

Flux normal:

1. L'usuari afegeix un títol al joc.
2. L'usuari escull el nivell de dificultat.
3. El sistema mostra les caselles a omplir corresponents al nivell.
4. L'usuari afegeix una imatge per a cada casella.
5. El sistema mostra la imatge i el nom a la casella.
6. L'usuari afegeix un so per a cada casella.
7. El sistema mostra el nom del so a la casella.
8. L'usuari prem el boto per iniciar el joc.
9. El sistema comprova que el joc és correcte.
10. S'inicia el cas d'ús *Jugar Memori*.

Flux alternatiu:

- 1a. L'usuari carrega un joc existent.
 - 1.1a. El sistema carrega el joc en la pantalla de configuració.
 - 1.2a. L'usuari pot modificar el joc (pas 2) o seguir al pas 8 o el 8b.
- 8b. L'usuari tria guardar el joc.
 - 8.1.b. El sistema guarda el joc.
 - 8.2.b. Segueix al pas 8 o l'usuari tanca l'aplicació i finalitza el cas d'ús.
- 9c. Alguna dada introduïda és errònia

9.1c. El sistema avisa de l'error.

Postcondicions: Si el joc s'ha guardat, el sistema ha de crear una carpeta amb totes les dades i l'arxiu del joc.

- **Cas d'ús Configuració Laberint:**

Actor: Usuari Configuració

Descripció: L'usuari afegeix el contingut que desitgi per crear un joc del tipus Laberint.

Flux normal:

1. L'usuari afegeix un títol al joc.
2. L'usuari afegeix un so com a historia.
3. El sistema mostra el nom de l'arxiu de so de la historia.
4. L'usuari afegeix una imatge com a personatge.
5. El sistema mostra la imatge i el nom de la imatge.
6. L'usuari escull un laberint i el sistema l'imprimeix.
7. L'usuari prem el boto per iniciar el joc.
8. El sistema comprova que el joc és correcte
9. S'inicia el cas d'ús *Jugar Laberint*.

Flux alternatiu:

- 1a. L'usuari carrega un joc existent.
 - 1.1a. El sistema carrega el joc en la pantalla de configuració.
 - 1.2a. L'usuari pot modificar el joc (pas 2) o seguir al pas 7 o el 7b.
- 7b. L'usuari tria guardar el joc.
 - 7.1.b. El sistema guarda el joc.
 - 7.2.b. Segueix al pas 7 o l'usuari tanca l'aplicació i finalitza el cas d'ús.
- 8c. Alguna dada introduïda es errònia
 - 8.1c. El sistema avisa de l'error.

Postcondicions: Si el joc s'ha guardat, el sistema ha de crear una carpeta amb totes les dades i l'arxiu del joc.

• **Cas d'ús Joc Parelles:**

Actor: Usuari Joc

Descripció: L'usuari juga al joc Parelles utilitzant la tauleta digital i la plantilla impresa.

Precondicions: S'ha d'haver iniciat anteriorment el cas d'ús *Configuració Parelles* i haver imprès la plantilla.

Flux normal:

1. L'usuari selecciona una casella de l'esquerra.
2. El sistema reproduïx el so associat a la casella.
3. L'usuari selecciona una altra casella de la dreta.
4. El sistema reproduïx el so associat a la casella.
5. El sistema comprova si les dues caselles estan associades.
6. El sistema reproduïx el so de correcte.
7. El sistema canvia la imatge de les caselles encertades.

* Es repeteixen les passes 1 a la 5 fins que s'han trobat totes les parelles i el joc a finalitzat.

Flux alternatiu:

* Si es prem alt + F4 en qualsevol moment, el joc es tanca i finalitza el cas d'ús.

- 6a. L'usuari no ha encertat l'associació.
 - 6.1a. El sistema reproduïx el so d'incorrecte.
 - 6.2a. Tornem al pas 1.

• **Cas d'ús Joc Busca Imatge:**

Actor: Usuari Joc

Descripció: L'usuari juga al joc Busca Imatge utilitzant la tauleta digital i la plantilla impresa.

Precondicions: S'ha d'haver iniciat anteriorment el cas d'ús *Configuració Busca Imatge* i haver imprès la plantilla.

Flux normal:

1. L'usuari selecciona la casella de la dreta.
2. El sistema reproduueix el so associat a la casella.
3. L'usuari selecciona una de les caselles de l'esquerra.
4. El sistema reproduueix el so associat a la casella.
5. L'usuari prem el botó per a triar resposta.
6. L'usuari selecciona la casella de la dreta.
7. El sistema reproduueix el so associat a la casella.
8. L'usuari selecciona una de les caselles de l'esquerra.
9. El sistema reproduueix el so associat a la casella.
10. El sistema comprova si l'associació es correcte.
11. El sistema reproduueix el so de correcte.

* Es repeteixen les passes 1 a la 4 tants cops com l'usuari vulgui.

Flux alternatiu:

* Si es prem alt + F4 en qualsevol moment, el joc es tanca i finalitza el cas d'ús.

- 10a. L'usuari no ha encertat l'associació.
 - 10.1a. El sistema reproduueix el so d'incorrecte.
 - 10.2a. Tornem al pas 6.
 - 10.2b. L'usuari prem el botó per a no triar resposta
 - 10.3b. Tornem al pas 1.

• **Cas d'ús Joc Resposta Correcte:**

Actor: Usuari Joc

Descripció: L'usuari juga al joc Resposta Correcte utilitzant la tauleta digital i la plantilla impresa.

Precondicions: S'ha d'haver iniciat anteriorment el cas d'ús *Configuració Resposta Correcte* i haver imprès la plantilla.

Flux normal:

1. El sistema reproduueix el so de la historia.
2. El sistema reproduueix el so de la pregunta.
3. L'usuari selecciona una de les caselles de resposta.
4. El sistema reproduueix el so associat a la casella.
5. L'usuari prem el botó per a triar resposta.
6. L'usuari selecciona una casella de resposta.
7. El sistema reproduueix el so associat a la casella.
8. El sistema comprova si la resposta és correcte.
9. El sistema reproduueix el so de correcte.
10. L'usuari prem el botó per a passar a la següent pregunta.
11. Tornem al pas 2.

* Es repeteixen les passes 3 i 4 tants cops com l'usuari vulgui.

* Es repeteixen totes les passes (1 a la 11) tantes vegades com preguntes s'hagin afegit.

Flux alternatiu:

* Si es prem alt + F4 en qualsevol moment, el joc es tanca i finalitza el cas d'ús.

- 10a. L'usuari no ha encertat l'associació.
 - 10.1a. El sistema reproduueix el so d'incorrecte.
 - 10.2a. Tornem al pas 6.
 - 10.2b. L'usuari prem el botó per a no triar resposta.
 - 10.3b. Tornem al pas 2.

• **Cas d'ús Joc Memori:**

Actor: Usuari Joc

Descripció: L'usuari juga al joc Memori utilitzant la tauleta digital i la plantilla impresa.

Precondicions: S'ha d'haver iniciat anteriorment el cas d'ús *Configuració Memori* i haver imprès la plantilla.

Flux normal:

1. El sistema desordena les caselles.
2. L'usuari selecciona una casella.
3. El sistema fa visible la casella.
4. El sistema reproduueix el so associat a la casella.
5. L'usuari selecciona una altre casella.
6. El sistema fa visible la casella.
7. El sistema reproduueix el so associat a la casella.
8. El sistema comprova que les caselles són iguals.
9. El sistema reproduueix el so de correcte.
10. El sistema deixa descobertes les caselles encertades.

* Es repeteixen les passes 2 i 10 fins a trobar totes les parelles.

Flux alternatiu:

* Si es prem alt + F4 en qualsevol moment, el joc es tanca i finalitza el cas d'ús.

- 9a. El sistema comprova que les parelles no són correctes.
 - 9.1a. El sistema reproduueix el so d'incorrecte.
 - 9.2a. El sistema torna a amagar les imatges de les caselles seleccionades.
 - 9.3a. Tornem al pas 2.

• **Cas d'ús Joc Laberint:**

Actor: Usuari Joc

Descripció: L'usuari juga al joc Laberint utilitzant la tauleta digital i la plantilla impresa.

Precondicions: S'ha d'haver iniciat anteriorment el cas d'ús *Configuració Laberint* i haver imprès la plantilla.

Flux normal:

1. El sistema reproduceix el so de la historia.
2. El sistema col·loca el personatge a la sortida del laberint
3. L'usuari mou el llapis per la tauleta.
4. El sistema mou el personatge dins del laberint seguint el moviment de l'usuari.
5. El personatge arriba al final del laberint
6. El sistema reproduceix el so de correcte.
7. El sistema passa al següent nivell del laberint.
8. Tornem al pas 2.

* Es repeteixen les passes de la 2 a la 8 fins que finalitzen els cinc nivells del laberint.

Flux alternatiu:

* Si es prem alt + F4 en qualsevol moment, el joc es tanca i finalitza el cas d'ús.

- 4a.El personatge es surt fora del laberint.
 - 4.1a. El sistema reproduceix el so d'incorrecte.
 - 4.2a. Tornem al pas 2.

8. Disseny

En aquesta etapa es farà el disseny de l'aplicació, utilitzant els diagrames de classes per a complir amb les funcionalitats anteriorment esmentades.

Per tal de que l'aplicació sigui el més mantenible i ampliable possible s'ha decidit implementar-la dividint l'aplicació en tres capes. Per tant l'aplicació constarà de la capa de Presentació, la capa de Domini, i la capa de Persistència.

La capa de presentació és on estaran tots els formularis de l'aplicació així com la lògica associada als events que és rebien, el domini on residirà tota la lògica de l'aplicació i la de persistència la que gestionarà les dades i crearà i llegirà els fitxers on es guardaran les activitats.

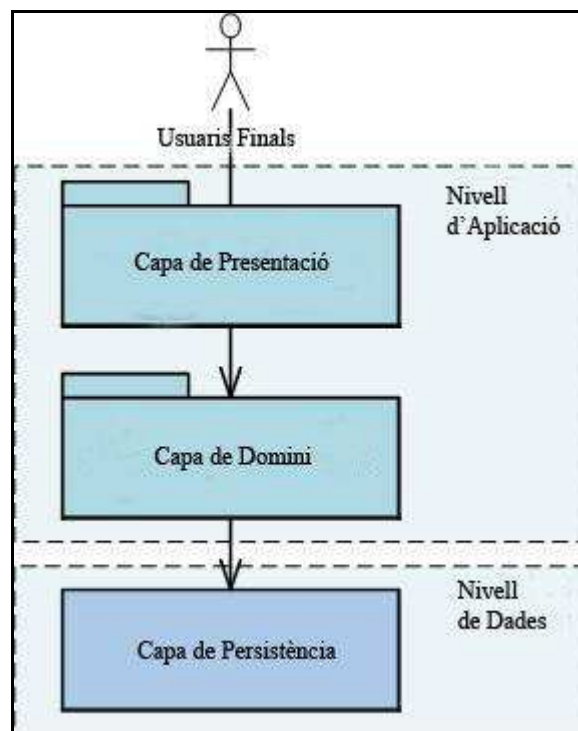


Figura 9: Model en capes.

8.1. Disseny de la Capa de Presentació.

El disseny de la interfície gràfica d'usuari ha estat decidit amb la supervisió dels responsables de la Fundació el Maresme i afegint les modificacions que s'han cregut oportunes.

- **Disseny interfície Parelles:**

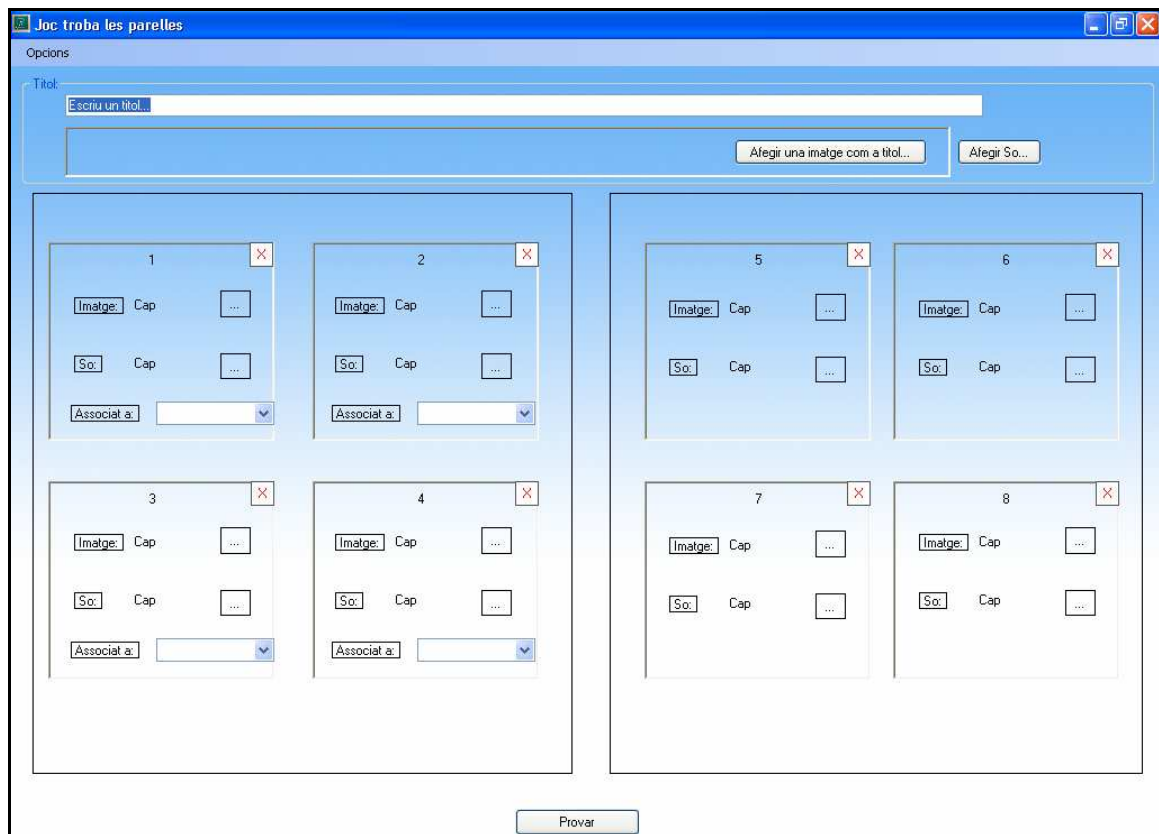


Figura 10: Configuració Parelles

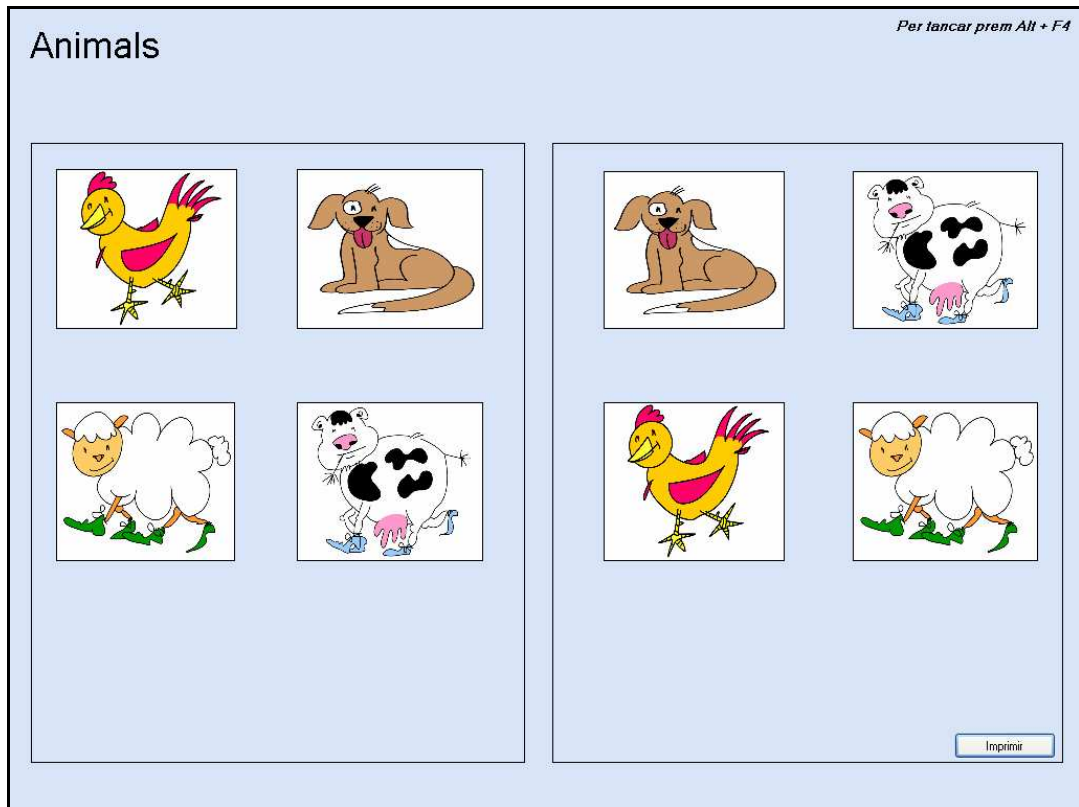


Figura 11: Pantalla Joc Parelles

- Disseny interfície Busca Imatge:

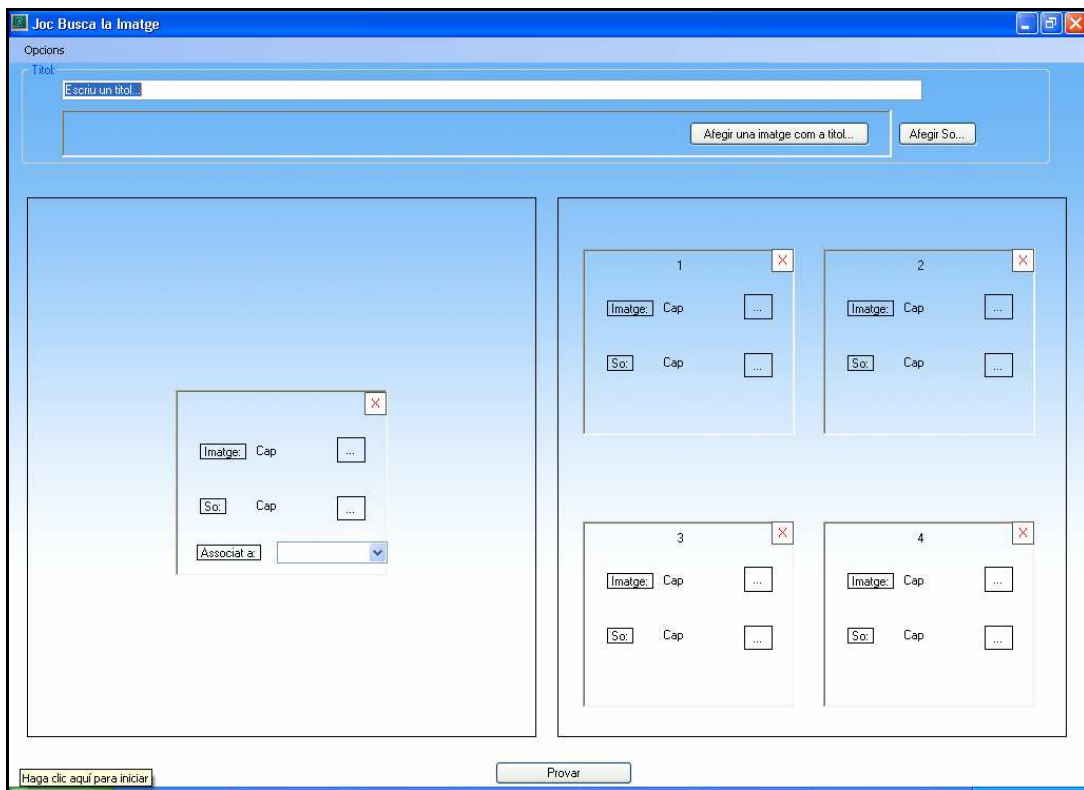


Figura 12: Pantalla Configuració Busca Imatge

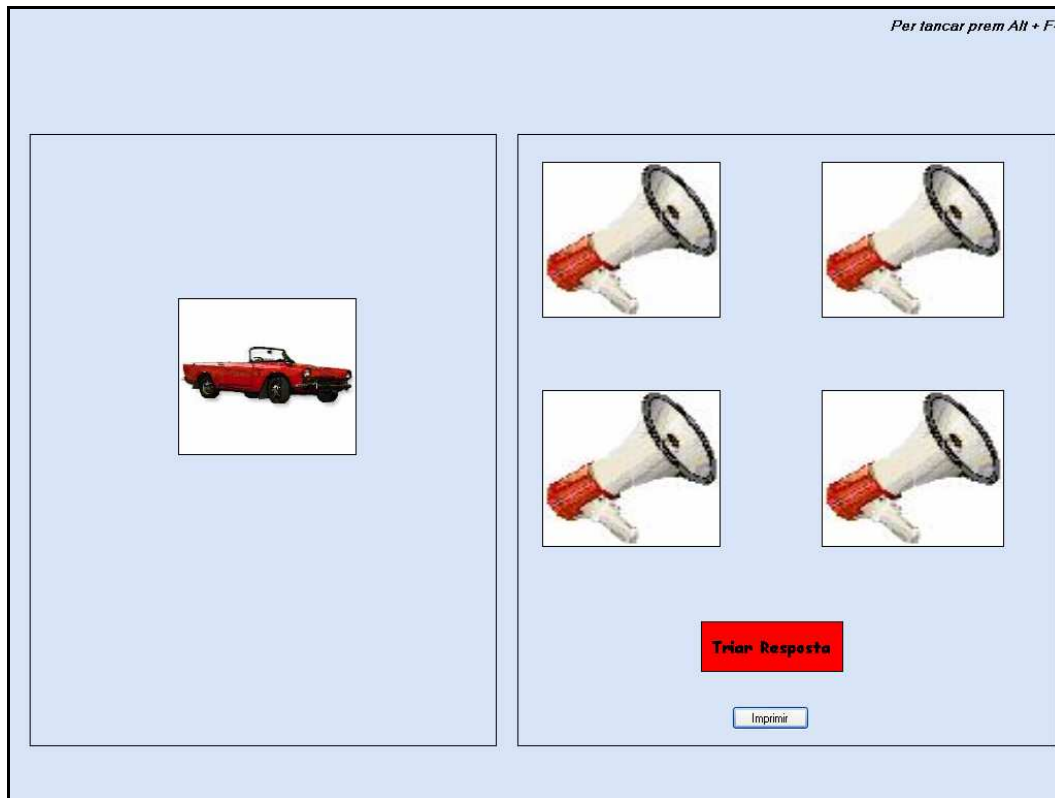


Figura 13: Pantalla Joc Busca Imatge

- Disseny interfície Resposta Correcte:

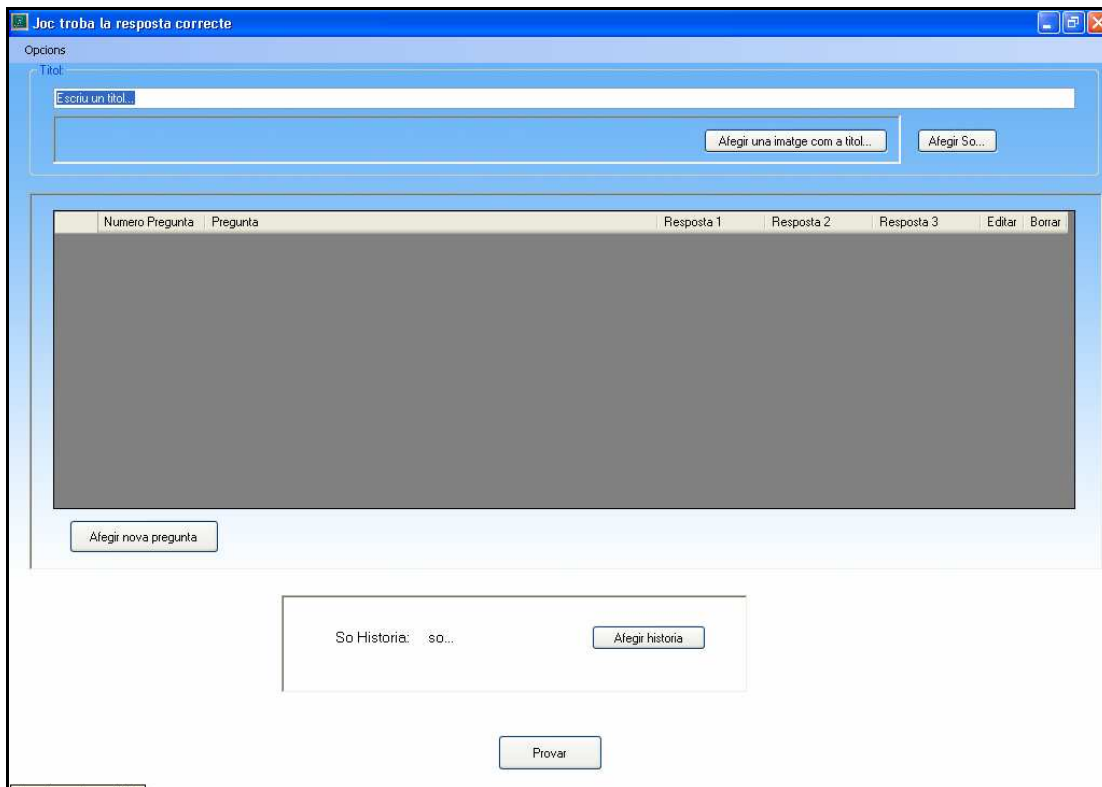


Figura 14: Pantalla Configuració Resposta Correcte

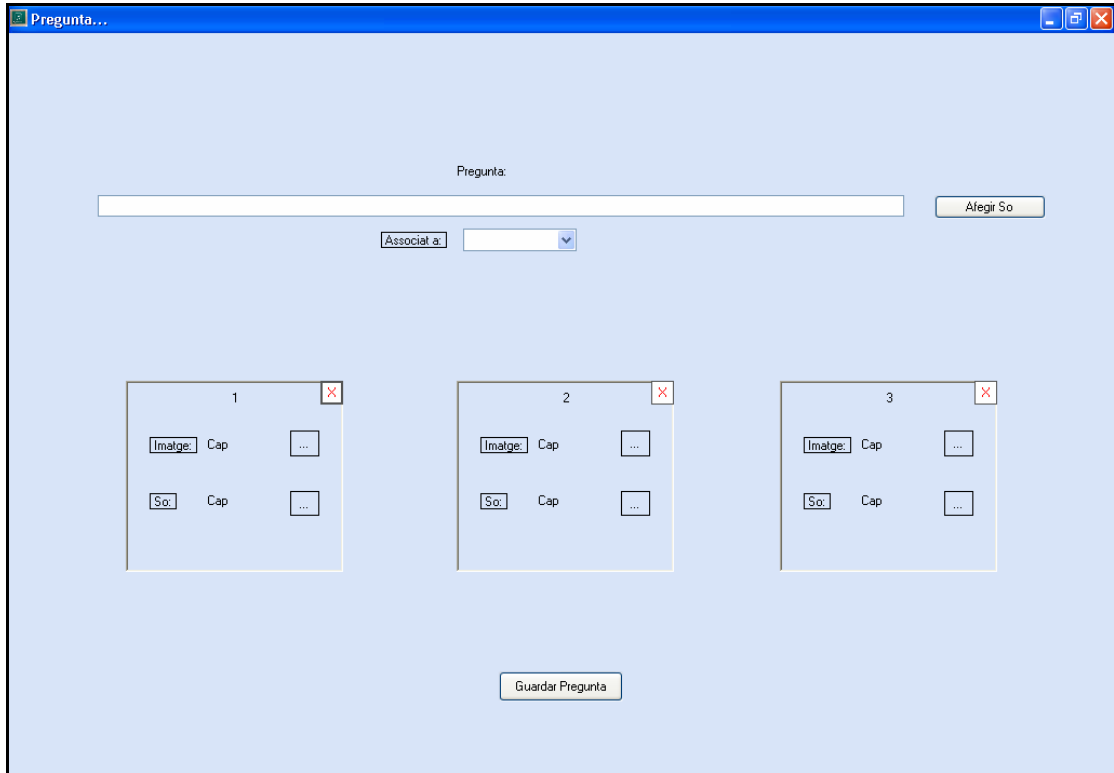


Figura 15: Pantalla Afegir Pregunta



Figura 16: Pantalla Joc Resposta Correcte

- Disseny interfície Memori:

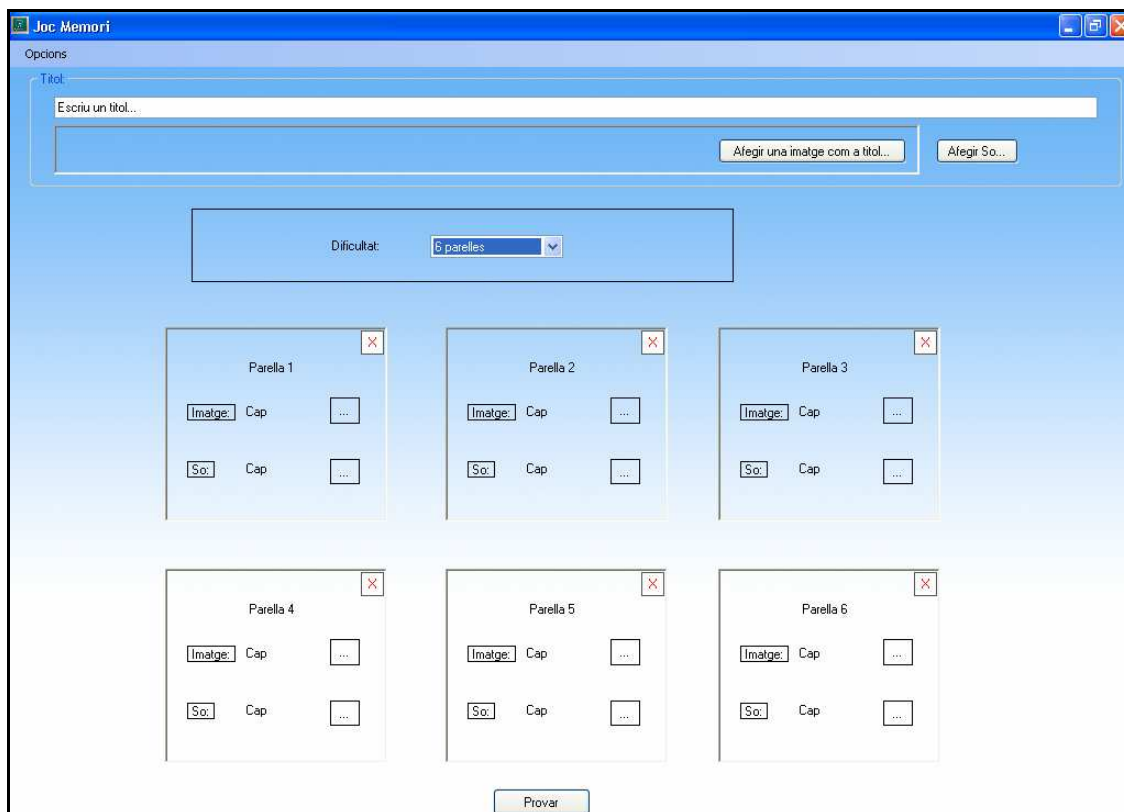


Figura 17: Pantalla Configuració Memori



Figura 18: Pantalla Joc Memori

- Disseny interfície Laberint:

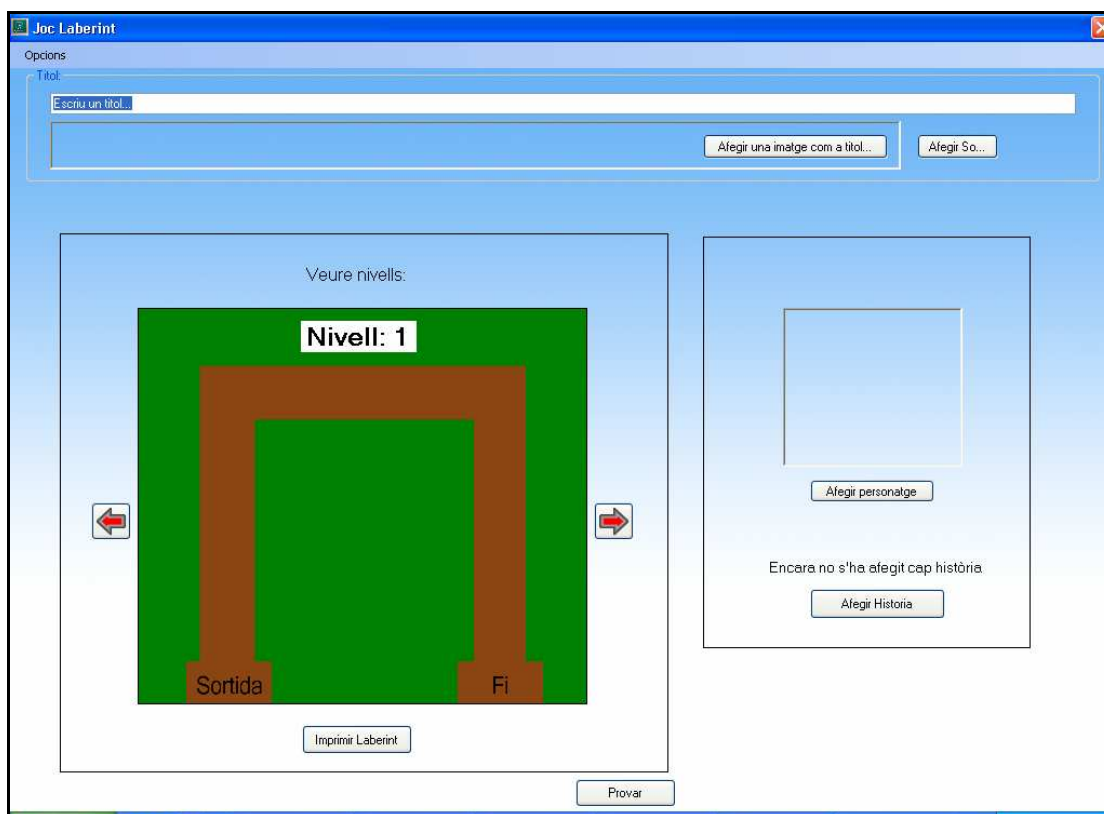


Figura 19: Pantalla Configuració Laberint.

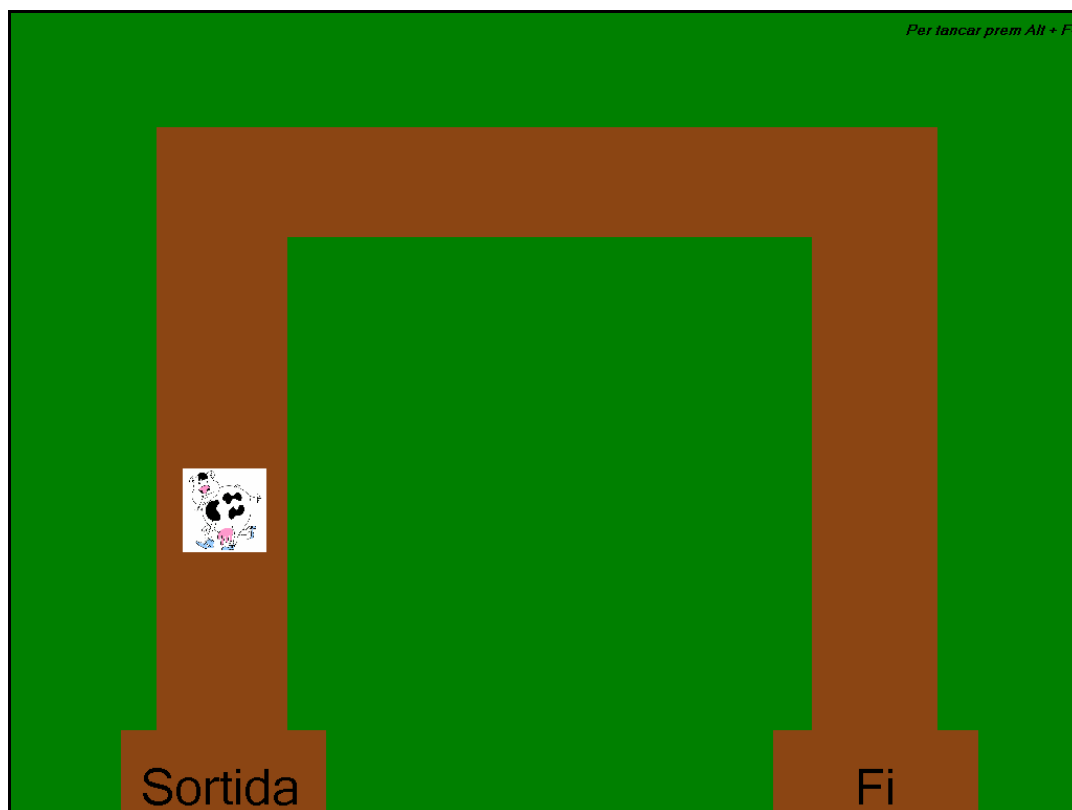


Figura 20: Pantalla Joc Laberint

8.2. Disseny de la Capa de Domini.

El diagrama de classes dissenyat per a l'aplicació ha estat el següent:

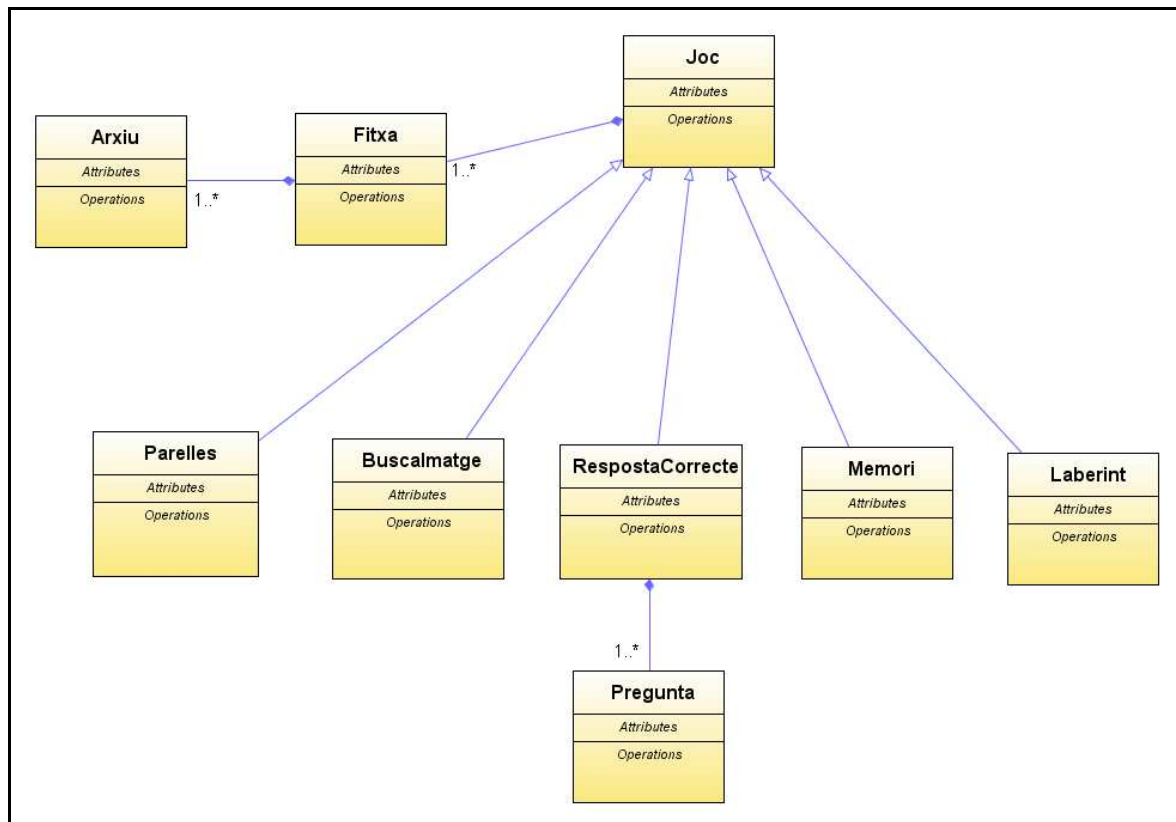


Figura 21: Disseny del Domini.

On la classe principal de l'aplicació és la classe *Joc*, de la qual hereten les classes *Parelles*, *BuscaImatge*, *RespostaCorrecte*, *Memori* i *Laberint*.

De la classe *RespostaCorrecte* hi ha una relació de contingència amb la classe *Pregunta* que és la classe que defineix els objectes de les preguntes que es crearan en aquest tipus de joc.

També tenim la classe *Fitxa* amb una relació de contingència amb la classe *Joc*. La classe *Fitxa* és la que contindrà les dades de cada casella de cada joc, és a dir les imatges i els sons que aquestes tinguin. La classe *Arxiu* és la que guardarà aquestes imatges i aquests sons.

8.3. Disseny de la Capa de Persistència.

L'aplicació havia de comptar amb la opció de poder guardar totes les activitats que es creessin i poder-les carregar sempre que es volgués. Per tal de poder complir amb aquesta funcionalitat s'ha decidit guardar les dades necessàries en arxius de text.

Per a la capa persistència s'ha creat la classe GestorDades que és la que llegirà i escriurà als arxius de text que guardaran els jocs. Aquest arxius contindran per a cada joc, el títol i les rutes de tots els arxius tant imatges com sons que contingui l'activitat.

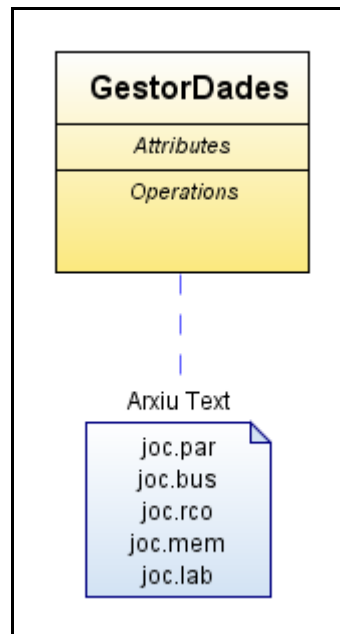


Figura 22: Disseny persistència.

9. Implementació

En aquesta etapa es realitzarà la implementació de l'aplicació, tenint en compte el que s'ha comentat en els apartats anteriors tant en l'etapa d'Anàlisi com en l'etapa de Disseny.

Es realitzarà la programació de les diferents classes que componen les capes de presentació, domini i persistència.

Per poder explicar com ha estat la implementació de l'aplicació es començarà per mostrar totes les classes del domini detallades i per últim quina serà la implementació de la capa de persistència.

9.1. Implementació de la Capa de Domini

A continuació es passarà a explicar com s'ha realitzat la implementació de cada classe del domini, tant els seus atributs com els mètodes més rellevants. No s'ha considerat necessària l'enumeració i explicació de tots els mètodes creats a l'aplicació, per tant només s'esmenten els més importants.

9.1.1 Classe Joc

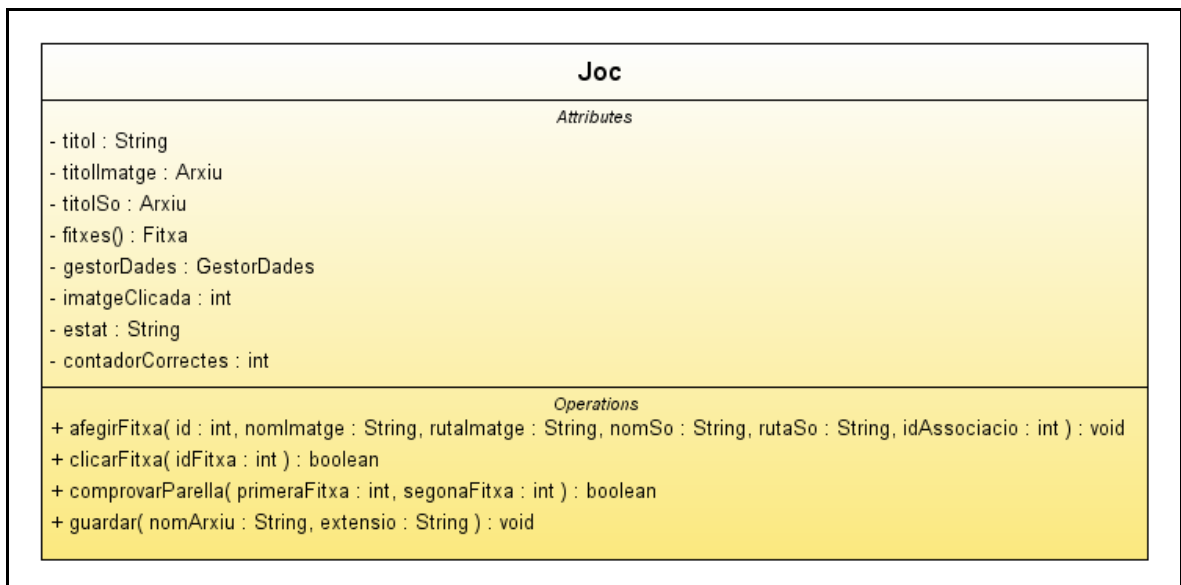


Figura 23: Classe Joc

Aquesta és la classe principal de l'aplicació, d'ella hereten la resta de tipus de jocs. Els atributs principals d'aquesta classe són:

- **títol:** Atribut del tipus String que conté el text del títol del joc.
- **títolImatge:** Atribut del tipus Arxiu (classe que comentarem més endavant) que conte el títol en forma de imatge del joc, s'hi és que s'ha afegit algun.
- **títolSo:** Atribut del tipus Arxiu que conté el so del títol del joc si l'usuari n'ha afegit algun.
- **Fitxes():** Atribut del tipus contenidor on cada posició conté un objecte del tipus Fitxa (classe que explicarem més endavant). La dimensió d'aquest contenidor anirà en funció del tipus de joc, ja que cada joc conté un nombre de fitxes diferent.
- **gestorDades:** Aquest atribut és el que crea una instància de la classe de persistència GestorDades, per tal de poder guardar i carregar les activitats.
- **imatgeClicada:** Atribut del tipus Integer que indica quina és la fitxa que l'usuari ha clicat en el moment de jugar.
- **contadorCorrectes:** Atribut del tipus Integer que indica el nombre de encerts que porta l'usuari per tal de saber quan ha finalitzat el joc.

A més dels mètodes gets i sets per a tots els atributs i el constructor de la classe, els mètodes més importants de la classe són els següents:

- **afegirFitxa:** Aquest mètode és el que s'encarrega de afegir una nova fitxa al joc que està creant l'usuari en aquell moment. El codi Visual Basic .Net associat a aquest mètode es el següent:

```
Public Sub afegirFitxa(ByVal id As Integer, ByVal nomI As String,
ByVal rutaI As String, ByVal nomS As String, ByVal rutaS As String,
ByVal idAss As Integer)
    Dim f = New Fitxa(id, nomI, rutaI, nomS, rutaS, idAss)
    fitxes(id - 1) = f
End Sub
```

Figura 24: Mètode afegirFitxa.

El mètode afegeix la nova fitxa a la posició indicada segons el seu identificador.

- **clicarFitxa:** Aquest és el mètode que s'executa quan l'usuari prem una casella del joc. El mètode comprova en quin estat es troba el joc. Si està en estat d'inici només reproduïx el so que aquesta casella té associada i guarda l'identificador de la casella que s'ha clicat. Per una altra banda, si l'estat del joc és en espera, el joc crida al mètode privat `comprovarParella`, reproduïx el so de correcte o incorrecte segons el cas, i torna a posar el joc en estat inici.

```
Public Overridable Function clicarFitxa(ByVal fitxa1 As Integer)
    Dim correcte = False
    If Not (Me.getSoFitxa(fitxa1).getRuta() = "") Then
        My.Computer.Audio.Play(Me.getSoFitxa(fitxa1).getRuta(),
AudioPlayMode.WaitToComplete)
    End If
    If estat = "inici" Then
        imatgeClicada = fitxa1
        estat = "espera"
    ElseIf estat = "espera" Then
        If (Me.comprovarParella(imatgeClicada, fitxes(fitxa1 -
1).getId())) Then
            correcte = True
            My.Computer.Audio.Play(My.Resources.CORRECTE,
AudioPlayMode.WaitToComplete)
            fitxes(imatgeClicada - 1).setEncertada(True)
            fitxes(fitxa1 - 1).setEncertada(True)
            'Reproduir So de correcte
        Else
            correcte = False
            My.Computer.Audio.Play(My.Resources.INCORRECTE,
AudioPlayMode.WaitToComplete)
            'Reproduir so de incorrecte
        End If
        estat = "inici"
    End If
    Return correcte
End Function
```

Figura 25: Mètode `clicarFitxa`.

- **comprovarParella:** Aquest és el mètode privat que comprova si la primera casella que s'ha clicat i la segona estaven associades. El tipus de retorn es booleà. El codi d'aquest mètode és el següent:

```

Public Function comprovarParella(ByVal primeraImatge As Integer, ByVal
segonaImatge As Integer)
    Dim correcte As Boolean
    correcte = (fitxes(primerImatge -
1).getIdAssociacio).Equals(segonaImatge)
    Return correcte
End Function

```

Figura 26: Mètode comprovarParella.

- **Guardar:** Aquest mètode crida al mètode per a guardar el joc de la classe GestorDades passant-li els paràmetres necessaris.
- **Carregar:** Aquest mètode crida al mètode carregar de la classe GestorDades amb el nom del arxiu que es vol carregar.

9.1.2 Classe Fitxa

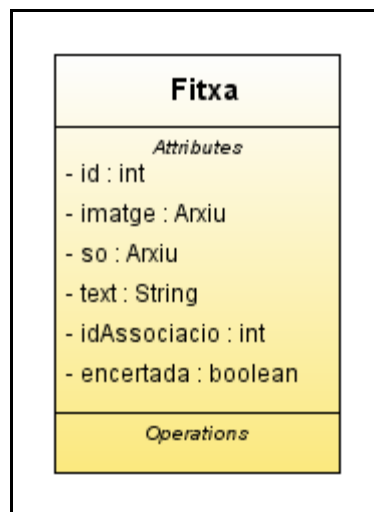


Figura 27: Classe Fitxa

La classe Fitxa és la que conté totes les dades necessàries per a cada casella. Com s'ha comentat anteriorment a cada casella s'hi podrà afegir imatge i so, i a més haurà d'estar associada a una altra seleccionada per l'usuari.

Per tal de que aquesta classe fos útil per a tots els tipus d'activitats s'ha programat de manera que té com atributs qualsevol variable que pugui tenir una fitxa, com per

exemple el text, ja que la pregunta dels jocs del tipus Resposta Correcte també la considerarem com una fitxa. Les històries que s'han de afegir als jocs Resposta Correcte i Laberint, també les considerarem com una fitxa.

La descripció dels atributs és la següent:

- **Id:** Atribut que conté l'identificador de la fitxa.
- **Imatge:** Atribut del tipus Arxiu que conté la imatge que correspon a la fitxa.
- **So:** Atribut del tipus Arxiu que conté el so que correspon a la fitxa.
- **Text:** Atribut del tipus String que només s'utilitzarà si la fitxa a crear és una pregunta, ja que afegirà el text de la pregunta que es vol fer.
- **idAssociacio:** Atribut del tipus Integer que conté l'identificador de la fitxa que està associada a l'actual. Aquest atribut només s'utilitzarà quan es tracti de fitxes que tinguin associació.

Els mètodes implementats en aquesta classe han estat els sets i els gets per a cada atribut, i els constructors, un per a cada tipus de fitxa que es volgués crear.

9.1.3 Classe Arxiu

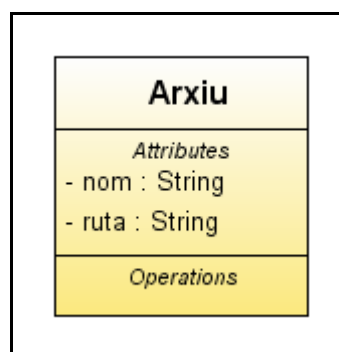


Figura 28: Classe Arxiu.

Aquesta classe és la que conté les dades dels arxius que s'han afegit, ja sigui una imatge o un so.

- **Ruta:** Conté la ruta completa de l'arxiu que s'ha afegit.
- **Nom:** Conté el nom de l'arxiu que s'ha afegit, amb extensió inclosa

9.1.4 Classe Parelles

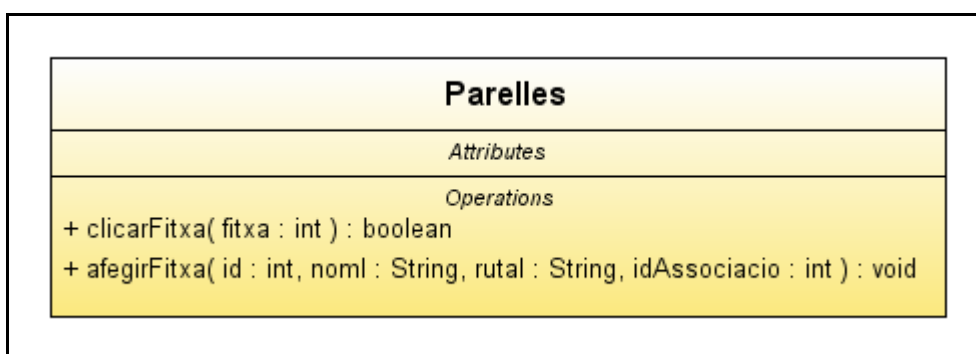


Figura 29: Classe Parelles.

Aquesta classe hereta tots els atributs de la classe principal Joc, i sobreesciu els mètodes clicarFitxa i afegirFitxa.

- **clicarFitxa:** Sobreesciu el mètode per tal de poder portar el control de quantes caselles s'han resolt i poder indicar que el joc ha finalitzat. El mètode crida al mètode clicarFitxa de la classe Joc. El codi associat a aquest mètode és el següent:

```

Public Overloads Function clicarFitxa(ByVal fitxa1 As Integer)
    Dim correcte = MyBase.clicarFitxa(fitxa1)
    If (correcte) Then
        contadorCorrectes = contadorCorrectes + 1
    End If
    If contadorCorrectes = 4 Then
        My.Computer.Audio.Play(My.Resources.FINAL_2,
AudioPlayMode.WaitToComplete)
        MyBase.possarContadorAzero()
        'fi del joc, poder tornar a jugar o sortir.
    End If
    Return correcte
End Function
  
```

Figura 30: Mètode clicarFitxa Parelles.

9.1.5 Classe BuscaImatge

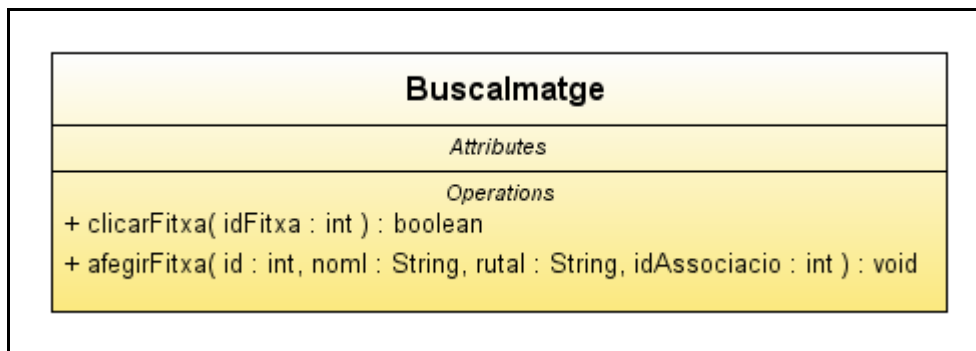


Figura 31: Classe BuscaImatge

Aquesta classe, com l'anterior, hereta tots els atributs de la classe Joc i sobrescriu els mateixos mètodes per tal d'obtenir la mateixa funcionalitat que en l'apartat anterior, però en aquest cas per a les activitats Busca Imatge.

9.1.6 Classe RespostaCorrecte

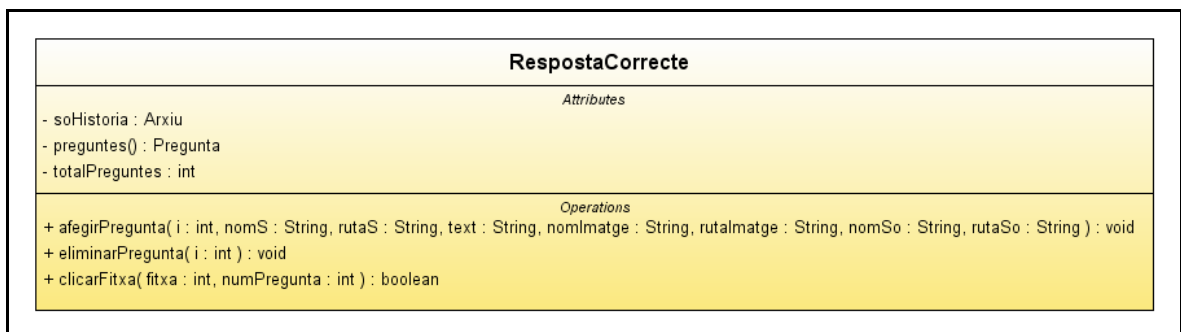


Figura 32: Classe RespostaCorrecte.

La classe Resposta Correcte a més d'heretar tots els atributs de la classe Joc afegeix atributs propis. Aquest són:

- **soHistoria:** Atribut del tipus Arxiu que conté les dades del so que s'ha afegit per a ser utilitzar com a fil conductor de la història.
- **Preguntes:** Atribut del tipus Array de Preguntes, que conté totes les preguntes que l'usuari hagi afegit a l'activitat.

- **totalPreguntes:** Atribut del tipus Integer que conté la mida de l'atribut anterior, és a dir, el nombre total de preguntes que conté el joc.

A més del mètodes heretats de la classe Joc, aquesta classe implementa els mètodes següents:

- **afegirPregunta:** Aquest mètode afegeix una nova Pregunta al joc del tipus Resposta Correcte. Aquesta pregunta constarà d'una pregunta i tres respostes (Classe Pregunta, que explicarem més endavant), i afegeix aquesta nova pregunta a la llista de preguntes de la classe. El codi associat a aquest mètode és el següent:

```
Public Sub afegirPregunta(ByVal i As Integer, ByVal nomS As String,
ByVal rutaS As String, ByVal idAss As Integer, ByVal text As String,
ByVal nomImatge() As String, ByVal rutaImatge() As String, ByVal
nomSo() As String, ByVal rutaSo() As String)
    Dim p = New Pregunta()
    p.afegirFitxaPregunta(i, nomS, rutaS, idAss, text)
    For x = 0 To 2
        p.afegirFitxa(x + 1, nomImatge(x), rutaImatge(x),
nomSo(x), rutaSo(x))
    Next
    preguntes(i - 1) = p
End Sub
```

Figura 33: Mètode afegirPregunta.

- **eliminarPregunta:** Aquest mètode s'executarà quan l'usuari en el mode de configuració decideixi eliminar una pregunta que hagués afegit prèviament. El codi es el següent:

```
Public Sub eliminarPregunta(ByVal i As Integer)
    i = i - 1
    totalPreguntes = totalPreguntes - 1
    For x = i To totalPreguntes
        If x = totalPreguntes Then
            preguntes(x) = Nothing
        Else
            preguntes(x) = preguntes(x + 1)
        End If
    Next
End Sub
```

Figura 34: Mètode eliminarPregunta.

- **clicarFitxa:** Aquest mètode sobreesciu el mètode de la classe Joc amb el mateix nom. El mètode comprova si per la pregunta que en aquell moment s'està responent, l'usuari ha triat la resposta correcta.

```

Public Overloads Function clicarFitxa(ByVal fitxa As Integer, ByVal
numPregunta As Integer)
    Dim correcte = False
    If Not (preguntes(numPregunta -
1).getFitxaResposta(fitxa).getSo().getRuta() = "") Then
        My.Computer.Audio.Play(preguntes(numPregunta -
1).getFitxaResposta(fitxa).getSo().getRuta(),
AudioPlayMode.WaitToComplete)
    End If
    If (preguntes(numPregunta -
1).getFitxaPregunta().getIdAssociacio() = preguntes(numPregunta -
1).getFitxaResposta(fitxa).getId()) Then
        correcte = True
        My.Computer.Audio.Play(My.Resources.CORRECTE,
AudioPlayMode.WaitToComplete)
        preguntes(numPregunta -
1).getFitxaPregunta().setEncertada(True)
        preguntes(numPregunta -
1).getFitxaResposta(fitxa).setEncertada(True)
        'Reproduir So de correcte
        contadorCorrectes = contadorCorrectes + 1
        If contadorCorrectes = totalPreguntes Then
            My.Computer.Audio.Play(My.Resources.FINAL_2,
AudioPlayMode.WaitToComplete)
            MyBase.possarContadorAzero()
        End If
    Else
        correcte = False
        My.Computer.Audio.Play(My.Resources.INCORRECTE,
AudioPlayMode.WaitToComplete)
        'reproduir so de incorrecte
    End If
    Return correcte
End Function

```

Figura 35: Mètode clicarFitxa RespostaCorrecte.

9.1.7 Classe Pregunta

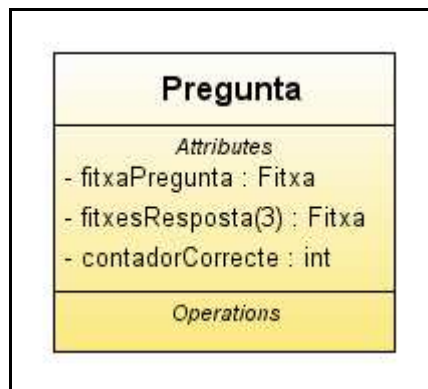


Figura 36: Classe Pregunta.

Aquesta classe conté tota la informació de cadascuna de les preguntes del joc RespostaCorrecte. Els atributs d'aquesta classe són els següents:

- **fitxaPregunta:** Atribut del tipus Fitxa. Conté la pregunta del test, aquesta pregunta és un objecte del tipus Fitxa amb text i so.
- **fitxaResposta:** Atribut del tipus Array de Fitxa. Aquest atribut conté les tres possibles respostes que tindrà la pregunta en concret.
- **contadorCorrecte:** Atribut del tipus Integer que conté el nombre de preguntes encertades que porta l'usuari en moment d'execució.

Els mètodes d'aquesta classe són els constructors i tots els gets i els sets corresponents a cada atribut.

9.1.8 Classe Memori

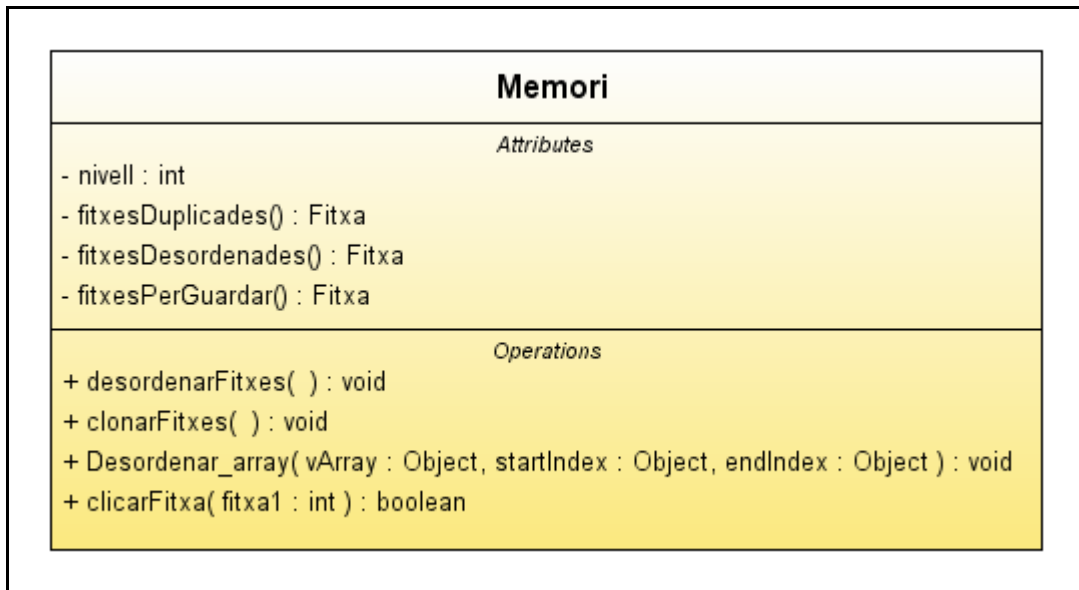


Figura 37: Classe Memori.

Aquesta classe implementa l'activitat Memori, els atributs que afegeix a més dels que ha heretat són:

- **Nivell:** Atribut del tipus Integer que conté el nivell de dificultat del joc.
- **fitxesDuplicades:** Atribut del tipus Array de Fitxes que conté les fitxes que ha afegit l'usuari duplicades per tal de tenir una parella de cada fitxa.
- **fitxesDesordenades:** Atribut del tipus Array de Fitxes que conté els mateixos valors que l'atribut fitxesDuplicades però en posicions aleatòries del vector, per tal de que a cada partida les fitxes estiguin en caselles diferents. Aquest atribut pren valor al executar-se el mètode desordenarFitxes().
- **fitxesPerGuardar:** Atribut del tipus Array de Fitxes que conté les fitxes que s'han de guardar si l'usuari vol guardar el joc, és a dir, les fitxes sense duplicar. S'ha cregut necessària la implementació d'aquest atribut ja que l'atribut fitxes de la classe principal, que és el que inicialment conté aquests valors, serà utilitzat per a

guardar, en temps d'execució i quan l'usuari estigui jugant, els valors de les fitxes desordenat per tal de poder utilitzar la funció clicarFitxa de la classe principal.

Els mètodes més rellevants implementats en aquesta classe són els següents:

- **desordenarFitxes:** Com s'ha comentat anteriorment aquest mètode té la funció de desordenar de manera aleatòria les fitxes afegides per l'usuari. Per la seva implementació, s'utilitza el mètode privat Desordenar_array. Les passes que realitza són: Primer redimensiona els contenidors de fitxes segons el nivell de dificultat de l'activitat, i duplica les fitxes. Després crea un vector amb els valors duplicats per cridar al procediment de desordenar_array, indicant-li quin serà el seu valor màxim i mínim. Un cop obtenim aquests nombres aleatoris els utilitzem per tal de, mitjançant una iteració, anar col·locant les fitxes al vector de fitxesDesordenades.

```
Public Sub desordenarFitxes()
    ReDim fitxesDuplicades(0 To (nivell * 2 + 1))
    ReDim fitxesDesordenades(0 To (nivell * 2 + 1))
    Dim xNivell As Integer
    Select Case nivell
        Case 1
            fitxesDuplicades(0) = fitxes(0)
            fitxesDuplicades(1) = fitxes(1)
            fitxesDuplicades(2) = fitxes(0)
            fitxesDuplicades(3) = fitxes(1)
            xNivell = 3
        Case 2
            fitxesDuplicades(0) = fitxes(0)
            fitxesDuplicades(1) = fitxes(1)
            fitxesDuplicades(2) = fitxes(2)
            fitxesDuplicades(3) = fitxes(0)
            fitxesDuplicades(4) = fitxes(1)
            fitxesDuplicades(5) = fitxes(2)
            xNivell = 5
        Case 3
            fitxesDuplicades(0) = fitxes(0)
            fitxesDuplicades(1) = fitxes(1)
            fitxesDuplicades(2) = fitxes(2)
            fitxesDuplicades(3) = fitxes(3)
            fitxesDuplicades(4) = fitxes(0)
            fitxesDuplicades(5) = fitxes(1)
            fitxesDuplicades(6) = fitxes(2)
            fitxesDuplicades(7) = fitxes(3)
            xNivell = 7
    End Select
End Sub
```

Figura 38: Mètode desordenarFitxes() (Part 1)

```

Case 5
    fitxesDuplicades(0) = fitxes(0)
    fitxesDuplicades(1) = fitxes(1)
    fitxesDuplicades(2) = fitxes(2)
    fitxesDuplicades(3) = fitxes(3)
    fitxesDuplicades(4) = fitxes(4)
    fitxesDuplicades(5) = fitxes(5)
    fitxesDuplicades(6) = fitxes(0)
    fitxesDuplicades(7) = fitxes(1)
    fitxesDuplicades(8) = fitxes(2)
    fitxesDuplicades(9) = fitxes(3)
    fitxesDuplicades(10) = fitxes(4)
    fitxesDuplicades(11) = fitxes(5)
    xNivell = 11
End Select
ReDim UnArray(xNivell)
' Agrega els valors al vector
For i = 0 To xNivell
    UnArray(i) = i
Next
' Desordena el elements
Call Desordenar_array(UnArray, LBound(UnArray),
UBound(UnArray))
'Passar les fitxes desordenades al nou vector
For i = 0 To nivell * 2 + 1
    fitxesDesordenades(i) = fitxesDuplicades(UnArray(i))
Next
clonarFitxes()
ReDim fitxes(fitxesDesordenades.Length - 1)
For i = 0 To fitxesDesordenades.Length - 1
    Me.fitxes(i) = fitxesDesordenades(i)
Next
End Sub

```

Figura 39: Mètode desordenarFitxes() (Part 2).

- **clonarFitxes:** Com s'ha comentat aquest mètode copia els objectes que conté l'atribut fitxes de la classe principal a l'atribut fitxesPerGuardar de la classe Memori. D'aquesta manera obtenim les fitxes que interessa guardar sense valors duplicats.
- **desordenarArray:** Aquest mètode és el que realment genera nombres aleatoris sense repetició. Per fer-ho s'ha utilitzat la sentència Randomize(). Cal indicar el nombre màxim i el nombre mínim d'aquests nombres al cridar a la funció.
- **clicarFitxa:** Aquest mètode, que és comú a tots els jocs exceptuant els de tipus Laberint, redefineix el mètode de la classe principal del mateix nom per tal de

poder tenir un control de si la partida ha acabat o no. La implementació és la següent:

```

Public Overloads Function clicarFitxa(ByVal fitxa1 As Integer)
    Dim correcte = MyBase.clicarFitxa(fitxa1)
    If (correcte) Then
        contadorCorrectes = contadorCorrectes + 1
    End If
    If contadorCorrectes = nivell + 1 Then
        My.Computer.Audio.Play(My.Resources.FINAL_2,
        AudioPlayMode.WaitToComplete)
        contadorCorrectes = 0
        'fi del joc, poder tornar a jugar o sortir.
    End If
    Return correcte
End Function

```

Figura 40: Mètode clicarFitxa Memori.

9.1.9 Classe Laberint

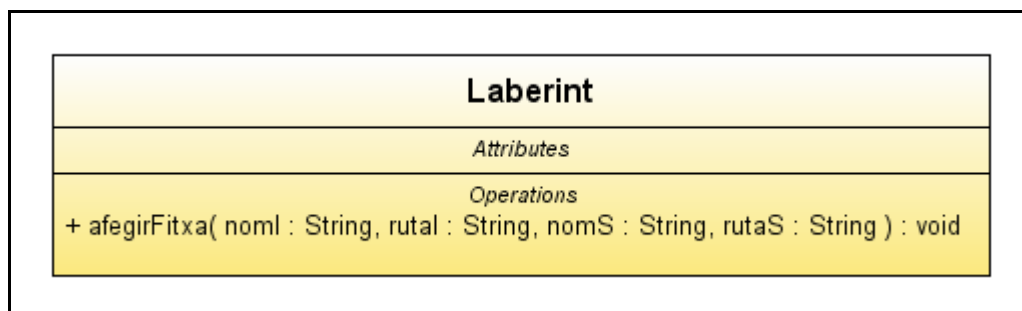


Figura 41: Classe Laberint.

Aquesta classe no implementa cap atribut propi, ja que només són necessaris els que hereta de la classe joc. Redefineix el mètode afegirFitxa, ja que en aquest cas, no es tractarà de una fitxa convencional com en els altres casos, però s'ha decidit seguir utilitzant la classe Fitxa perquè igualment ens era útil. En aquest cas els valors de l'objecte fitxa seran la imatge del personatge i el so de la historia, no té cap tipus d'associació.

Els mètodes per comprovar si l'usuari s'ha sortit de la pantalla o ha arribat al final estan implementats a la capa de presentació ja que responen a events que genera la pantalla.

Per una altra banda, per a tots els tipus de jocs, era necessari poder imprimir la pantalla a mode de plantilla per col·locar-la a sobre de la tauleta, aquests mètodes d'imprimir estan implementats a la capa de presentació de cada formulari en concret.

Aquestes plantilles tenien que encaixar amb el model de tauleta *wide screen design tablet TB – 7300* de la marca Trust. Així doncs s'ha implementat de manera que imprimeixi amb les mides d'aquesta tauleta.

9.2. Implementació de la Capa de Persistència

En aquest apartat s'explicarà com s'ha implementat la part de persistència de l'aplicació.

Un dels objectius fonamentals del projecte era que les activitats creades utilitzant l'aplicació es poguessin guardar i carregar. S'ha decidit complir aquesta funcionalitat de la mateixa manera que es va fer en l'aplicació *A Mida*.

Quan un usuari vol guardar un joc, el programa crea una carpeta amb el nom que hagi afegit l'usuari i a dins d'aquesta copia tots els arxius necessaris per a fer-la funcionar (imatges, sons, etc.). A dins d'aquesta carpeta es crea també un arxiu de text (.txt) amb les indicacions de quin contingut haurà d'anar a cada casella. Inicialment es guardava directament la ruta dels arxius en aquest .txt sense crear la carpeta, fet que, si es treballava en local no donava problemes (ja que els arxius estaven en els llocs indicats), però si es decidia agafar l'arxiu i intentar executar-lo en un ordinador diferent amb el software instal·lat donava errors, ja que en aquell ordinador no es trobaven els arxius que el programa estava buscant. Per solucionar aquest problema es va decidir treballar amb carpetes, i per tant per a poder utilitzar un joc ja creat en un altre terminal, es necessitarà la carpeta sencera del joc.

Un altre punt important era que el joc s'executés directament al obrir l'arxiu, per aconseguir aquest objectiu s'ha decidit que l'aplicació canviï la extensió d'aquest .txt per una pròpia, seguint els criteris de la taula següent:

Tipus de Joc	Extensió
Parelles	.par
Busca Imatge	.bus
Resposta Correcte	.rco
Memori	.mem
Laberint	.lab

Figura 42: Taula d'extensions.

D'aquesta manera, si associem la extensió pròpia al nostre aplicatiu, quan l'usuari obri l'arxiu s'obrirà l'activitat directament.

S'ha decidit posar una extensió diferent per a cada tipus de joc per dos motius. El primer és que l'usuari pugui veure de quina activitat es tracta a simple vista, i l'altre per que l'aplicatiu pugui distingir quin tipus de joc s'ha obert.

La classe encarregada d'aquestes funcions és la classe GestorDades que es mostra a continuació:

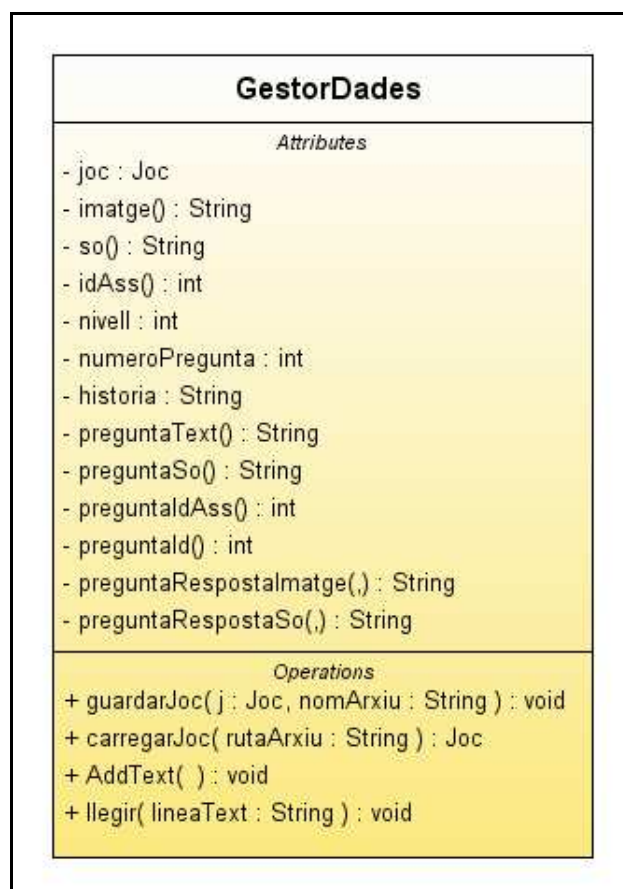


Figura 43: Classe GestorDades.

Els atributs de la classe són els següents:

- **Joc:** Atribut del tipus `Joc` que contindrà el joc que s'ha carregat i serà el retorn del mètode `carregarJoc`.
- **Imatge:** Array de Strings que contindrà els noms de les imatges llegides de l'arxiu que s'estigui carregant.
- **So:** Atribut del tipus Array de Strings que contindrà els nom de l'arxiu de so llegit del fitxer que s'estigui carregant.
- **idAss:** Aquest atribut que és del tipus Array de Integer conté el id de la fitxa a la que està associada, també serà llegit del fitxer.
- **Nivell:** Contindrà el nivell del joc (no serà necessari per a tots els jocs, només per els de tipus memòria).
- **numeroPregunta:** Quan és llegit del fitxer del joc i aquest sigui del tipus `RespostaCorrecte` aquesta variable contindrà el nombre de preguntes que té l'activitat.
- **Historia:** Nom de l'arxiu de so que es farà servir com història de l'activitat.
- **preguntaText:** Atribut del tipus Array de String que contindrà els textos de les preguntes de l'activitat que s'està carregant, tindrà tantes posicions com preguntes existeixin.
- **preguntaSo:** Igual que el cas anterior però amb el nom de l'arxiu de so de la pregunta.
- **preguntaIdAss:** Array de String que conté les associacions de les fitxes.

- **preguntaId:** Array que conté a cada posició l'identificador de la fitxa.
- **PreguntaRespostaImatge:** Matriu que contindrà les tres possibles imatges de la resposta d'una pregunta en concret.
- **PreguntaRespostaSo:** Igual que el cas anterior però amb els noms dels sons.

El mètodes més importants d'aquesta classe són els guardarJoc i carregarJoc, s'intentarà explicar sense entrar en detall el funcionament d'aquests mètodes.

- **guardarJoc:** Aquest mètode guarda el joc que l'usuari li hagi indicat, aquesta funcionalitat s'ha implementat de manera que es guarda la part comú a tots els jocs al codi d'aquest mètode, i depenent de quin tipus de joc es desitja guardar es crida als mètodes privats:
 - GuardarParelles: El mètode escriu en un arxiu de text els valors del joc, utilitzant etiquetes per tal de que la localització de les dades sigui més senzilla alhora de carregar. També és el mètode encarregat de fer còpies dels arxius necessaris i col·locar-los a la nova carpeta.

```

Private Sub guardarParelles()
    For i = 1 To 8
        AddText(arxiu, "Fitxa Imatge " & i & ": " &
joc.getImatgeFitxa(i).getNom() & Environment.NewLine)
        AddText(arxiu, "Fitxa So " & i & ": " &
joc.getSoFitxa(i).getNom() & Environment.NewLine)
        AddText(arxiu, "Fitxa idAssociacio " & i & ": " &
joc.getFitxa(i).getIdAssociacio() & Environment.NewLine)
        If Not My.Computer.FileSystem.FileExists(carpetamod + "\"
+ joc.getImatgeFitxa(i).getNom()) Then

My.Computer.FileSystem.CopyFile(joc.getImatgeFitxa(i).getRuta(),
carpetamod + "\" + joc.getImatgeFitxa(i).getNom())
        End If
        If Not My.Computer.FileSystem.FileExists(carpetamod + "\"
+ joc.getSoFitxa(i).getNom()) And Not joc.getSoFitxa(i).getNom() = ""
Then

My.Computer.FileSystem.CopyFile(joc.getSoFitxa(i).getRuta(),
carpetamod + "\" + joc.getSoFitxa(i).getNom())
        End If
    Next
End Sub

```

Figura 44: Mètode guardarParelles()

- GuardarBuscaImatge: Funciona igual que l'anterior però guardant les dades del joc d'aquest tipus.
 - GuardarRespostaCorrecte.
 - GuardarMemori.
 - GuardarLaberint.
- **addText:** Mètode d'escriptura al fitxer de text.
 - **carregarJoc:** Aquest és el mètode que carrega el joc que l'usuari vol obrir. Rep per paràmetre el nom de l'arxiu que s'ha d'obrir, i llegint línia per línia, Es recupera la informació del fitxer llegint seqüencialment línia a línia. Igual que en el cas anterior existeixen els mètodes privats de carregar per a cada joc en concret, que es criden segons el cas. Un exemple seria el codi de carregarJocParelles:

```

Private Sub carregarParelles(ByVal arxiu As StreamReader)
    Dim lineaText As String
    joc = New Parelles()
    While Not arxiu.EndOfStream
        lineaText = arxiu.ReadLine()
        llegir(lineaText)
    End While
    For i = 0 To 7
        If so(i) = "" Then
            joc.afegirFitxa(i + 1, imatge(i),
                Path.GetFullPath(imatge(i)), so(i), so(i), idAss(i))
        Else
            joc.afegirFitxa(i + 1, imatge(i),
                Path.GetFullPath(imatge(i)), so(i), Path.GetFullPath(so(i)),
                idAss(i))
        End If
    Next
End Sub

```

Figura 45: Mètode carregarParelles.

- **Llegir:** Mètode que rep com a paràmetre cada línia del arxiu de text que s'està carregant, i depenent quina etiqueta tingui, l'afegeix a l'atribut que li correspon.

A continuació es mostrà un exemple del fitxer que guarda un joc del tipus Parelles:

```
Titol: prova
Fitxa Imatge 1: gall.gif
Fitxa So 1: gallCurt.wav
Fitxa idAssociacio 1: 7
Fitxa Imatge 2: gos dret.GIF
Fitxa So 2: gosCurt.wav
Fitxa idAssociacio 2: 5
Fitxa Imatge 3: ovella.gif
Fitxa So 3: ovellaCurt.wav
Fitxa idAssociacio 3: 8
Fitxa Imatge 4: vaca.gif
Fitxa So 4: vacaCurt.wav
Fitxa idAssociacio 4: 6
Fitxa Imatge 5: gos dret.GIF
Fitxa So 5:
Fitxa idAssociacio 5: 2
Fitxa Imatge 6: vaca.gif
Fitxa So 6:
Fitxa idAssociacio 6: 4
Fitxa Imatge 7: gall.gif
Fitxa So 7:
Fitxa idAssociacio 7: 1
Fitxa Imatge 8: ovella.gif
Fitxa So 8:
Fitxa idAssociacio 8: 3
```

Figura 46: Exemple de joc guardat.

9.3. Instal·lador

Per a crear l'instal·lador de l'aplicació s'ha utilitzat NSIS en la seva versió 2.41.

A continuació es mostra el codi utilitzat per a realitzar l'instal·lador i el codi necessari per a associar les nostres extensions pròpies al nostre programa, per a que el sistema les reconegui i les obri amb aquest.

```
;-----  
;  
; Autora: Sara Cortes Viñals  
;-----  
  
!include "registerExtension.nsh"  
  
; Nom del instal·lador  
Name "Instalacio Jocs per tauleta"  
  
; Nom del exe resultant  
OutFile "JocsTauletaInstall.exe"  
  
; Directori per defecte on es farà la instal·lació  
InstallDir $PROGRAMFILES\JocsTauleta  
  
; Request application privileges for Windows Vista  
RequestExecutionLevel admin  
  
;-----  
; Pages  
  
Page components  
Page directory  
Page instfiles  
  
UninstPage uninstConfirm  
UninstPage instfiles
```

Figura 47: Codi instal·lador (inici).

En aquesta part del codi s'especifica el nom de l'instal·lador, el nom que tindrà l'arxiu .exe final i el directori on es farà la instal·lació.


```

;-----
; Instal·lació bàsica
;-----
Section "Instal·lació Jocs Tauleta (obligatori)"

SectionIn RO

; Set output path to the installation directory.
SetOutPath $INSTDIR

; Fitxers que es descomprimiran amb l'instal·lació, tots en aquest cas
File /r *.*

; Ens guardem al registre el directori de la instal·lació
WriteRegStr HKLM SOFTWARE\JocsTauleta "Install_Dir" "$INSTDIR"

; Write the uninstall keys for Windows
WriteRegStr HKLM
"Software\Microsoft\Windows\CurrentVersion\Uninstall\JocsTauleta"
"DisplayName" "Uninstall JocsTauleta"
WriteRegStr HKLM
"Software\Microsoft\Windows\CurrentVersion\Uninstall\JocsTauleta"
"UninstallString" "$INSTDIR\uninstall.exe"
WriteRegDWORD HKLM
"Software\Microsoft\Windows\CurrentVersion\Uninstall\JocsTauleta" "NoModify" 1
WriteRegDWORD HKLM
"Software\Microsoft\Windows\CurrentVersion\Uninstall\JocsTauleta" "NoRepair" 1
WriteUninstaller "uninstall.exe"

; Registrem la extensió
${registerExtension} "$INSTDIR\JocsTauleta.exe" ".par" "Joc Parelles"
${registerExtension} "$INSTDIR\JocsTauleta.exe" ".bus" "Joc Busca Imatge"
${registerExtension} "$INSTDIR\JocsTauleta.exe" ".rco" "Joc Resposta Correcte"
${registerExtension} "$INSTDIR\JocsTauleta.exe" ".mem" "Joc Memori"
${registerExtension} "$INSTDIR\JocsTauleta.exe" ".lab" "Joc Laberint"

SectionEnd

```

Figura 48: Codi instal·lador (secció obligatòria).

En aquesta part comprimeix a l'instal·lador tots els arxius del programa, els arxius que estan a la carpeta on guardem l'script a l'hora de compilar-lo, en aquest cas és la carpeta del projecte en Visual Basic que conté tots els arxius necessaris per a que l'aplicació funcioni i l'arxiu JocsTauleta.exe.

També crida a la funció `registerExtension` que és la que s'encarrega de indicar-li a l'ordinador que els arxius amb la extensió indicada s'han d'obrir amb l'aplicació. Aquesta funció juntament amb la de `unregisterExtension` es troben en un altre fitxer anomenat `registerExtension.nsh`.

```
;-----  
; Aquesta secció es per fer accessos directes. Opcional per l'usuari  
;-----  
  
Section "Instal·lació accés directe"  
  
    CreateDirectory "$SMPROGRAMS\JocsTauleta"  
    CreateShortCut "$SMPROGRAMS\JocsTauleta\Uninstall.lnk"  
"$INSTDIR\uninstall.exe" "" "$INSTDIR\uninstall.exe" 0  
    CreateShortCut "$SMPROGRAMS\JocsTauleta\JocsTauleta.lnk"  
"$INSTDIR\JocsTauleta.exe" "" "$INSTDIR\JocsTauleta.exe" 0  
    CreateShortCut "$DESKTOP\JocsTauleta.lnk" "$INSTDIR\JocsTauleta.exe" ""  
"$INSTDIR\JocsTauleta.exe" 0  
  
SectionEnd
```

Figura 49: Codi instal·lador (Part opcional).

En aquesta secció és on es creen els accessos directes de l'aplicació a l'escriptori i al menú inici. És una part opcional per l'usuari, ja que pot decidir si s'han de crear els accessos directes o no.

```

; -----
; Desinstal·lador
; -----

Section "Uninstall"

; Remove registry keys
DeleteRegKey HKLM
"Software\Microsoft\Windows\CurrentVersion\Uninstall\JocsTauleta"
DeleteRegKey HKLM SOFTWARE\JocsTauleta

; Remove files and uninstaller
Delete $INSTDIR\install.nsi
Delete $INSTDIR\uninstall.exe
Delete $INSTDIR\JocsTauleta.application
Delete $INSTDIR\JocsTauleta.exe
Delete $INSTDIR\JocsTauleta.exe.manifest
Delete $INSTDIR\JocsTauleta.pdb
Delete $INSTDIR\JocsTauleta.xml
Delete $INSTDIR\AxInterop.WMPLib.dll
Delete $INSTDIR\Interop.WMPLib.dll
Delete "$INSTDIR\Manual d'usuari.pdf"
Delete $INSTDIR\registerExtension.nsh

; Remove shortcuts, if any
Delete "$SMPROGRAMS\JocsTauleta\*.*)"
Delete "$INSTDIR\JocsTauleta\*.*)"
Delete "$DESKTOP\JocsTauleta.lnk"

; Remove directories used
RMDir $SMPROGRAMS\JocsTauleta
RMDir $INSTDIR\JocsTauleta

${unregisterExtension} ".par" "Joc Parelles"
${unregisterExtension} ".bus" "Joc Busca Imatge"
${unregisterExtension} ".rco" "Joc Resposta Correcte"
${unregisterExtension} ".mem" "Joc Memori"
${unregisterExtension} ".lab" "Joc Laberint"

SectionEnd

```

Figura 50: Codi desinstal·lador.

La part del desinstal·lador és la que borra tots els arxius creats quan s'instal·la l'aplicació, elimina els accessos directes de l'escriptori i del menú inici (si existeixen) i crida a la funció `unregisterExtension` per a desenregistrar les extensions.

10. Pressupost.

Els costos associats a l'aplicació són els que es mostren en les següents taules:

Recursos Hardware	Euros
Equip informàtic	1.200 €
Tauleta Digital TB-7300 Trust	164,99 €
Total	1.364,99 €

Figura 51: Taula Recursos Hardware.

Recursos Software	Euros
Visual Studio 2008 Express Edition	0 €
Microsoft .NET Framework 3.5	0 €
NSIS	0 €
Total	0 €

Figura 52: Taula Recursos Software.

Recursos Humans	Euros
Estudi previ: 20 h x 15 €	300 €
Anàlisi: 40 h x 15 €	600 €
Disseny: 80 x 15 €	1.200 €
Implementació: 120 x 15 €	1.800 €
Proves: 60 h x 15 €	900 €
Documentació: 90 h x 15 €	1.350 €
Total	6.150 €

Figura 53: Taula Recursos Humans.

Activitat	Euros
Recursos Hardware	1.364,99 €
Recursos Software	0€
Recursos Humans	6.150 €
Total	7.514,99 €

Figura 54: Taula Pressupost.

Si comparem la taula de recursos humans amb l'estimació de temps que és va fer en l'apartat de planificació podem comprovar que algunes etapes s'han allargat més del temps que havíem previst inicialment.

Això es deu, principalment a la manca d'experiència alhora de fer planificacions de projectes. Per una altra banda, el desconeixement del llenguatge Visual Basic .Net a fet que el temps dedicat a la implementació hagi sigut més elevat de l'esperat.

Un altre motiu per a l'increment de temps ha estat que durant el projecte s'han concretat o afegit nous requisits que en el moment de fer la estimació no havien estat previstos.

11. Possibles millores.

Algunes de les possibles millores per aquesta aplicació podrien ser el permetre més tipus de arxius de so, ja que actualment només permet el format .wav ja que era l'únic que no requeria de cap reproductor per a reproduir-se.

Un altre millora seria que l'usuari pogués escollir la mida de la plantilla que s'imprimirà, ja que actualment i a petició de la Fundació el Maresme, la plantilla sempre s'imprimeix en la mateixa mida, que és la de la tauleta que ells utilitzen, i per tant si algun usuari volgués utilitzar l'aplicació amb una tauleta d'unes altres mides la plantilla no encaixaria.

Per una altre banda també es podria millorar, encara que no es una característica indispensable, la interfície gràfica de l'aplicació. Tot i així pot resultar una millora interessant ja que d'aquesta manera es podria fer més atractiva l'aplicació per a altres tipus de col·lectius.

12. Conclusions.

El projecte té com a objectius la creació d'un programa que permeti crear cinc tipus d'activitats diferents totalment configurables.

Aquest objectiu principal ha estat assolit. L'aplicació creada permet crear tant activitats del tipus parelles, com busca imatge, resposta correcte, memòria i laberint. Totes elles són configurables per l'usuari, ja que permet afegir les imatges i sons escollides. D'aquesta manera es permet per exemple crear diverses activitats amb temàtiques diferents.

Un altre objectiu assolit a estat la possibilitat de interactuar amb aquestes activitats mitjançant la tauleta digital, acció que es pot realitzar perfectament amb l'aplicació ja que aquesta inclou la opció de imprimir la plantilla directament des del programa amb les mides necessàries.

Per als cinc tipus d'activitats diferents es té la opció de poder guardar i carregar, i a més a més utilitzar-la en qualsevol màquina que tingui el software instal·lat.

Per concloure comentar que també s'han assolit nous coneixements sobre el llenguatge utilitzat, en aquest cas el Visual Basic .Net, i s'ha adquirit experiència alhora de treballar en projectes que no són només acadèmics si no que posteriorment tindran un ús en la societat.

Un altre punt on també s'ha adquirit experiència ha sigut en el tracte amb l'usuari, ja que he tingut que assistir periòdicament a reunions amb l'usuari que definia els requeriments del projecte. Aquest tipus d'experiència difícilment s'adquireix durant la vida acadèmica.

ANNEX I

Manual d'usuari:

Instal·lació:

Per instal·lar l'aplicació, clicar sobre l'arxiu JocsTauletaInstall.exe, i s'executarà l'instal·lador. Si es desitja que es creï un accés directe a l'escriptori marcar la opció d'instal·lació d'accés directe.

Prémer el botó next i el programa s'instal·larà, un cop la instal·lació ha finalitzat prémer el botó close.

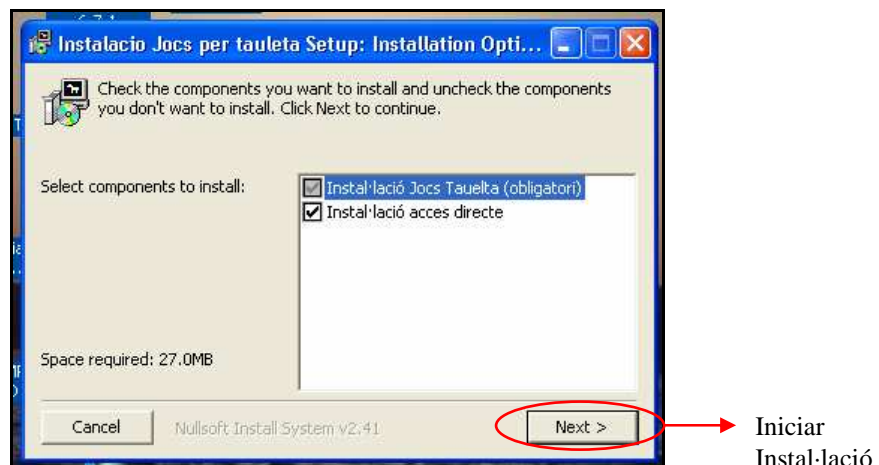


Figura 1: Pantalla d'instal·lació.

Crear un joc:

Obrir el programa JocsTauleta i es mostrarà la pantalla de selecció de joc:

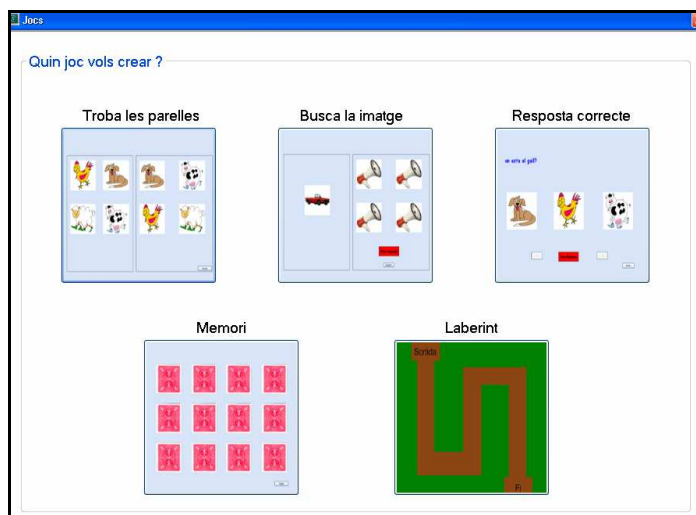


Figura 2: Selecció de Joc.

Prémer el botó del joc que es vulgui crear entre els cinc possibles.

Parelles:

Si l'activitat seleccionada és la de parelles s'obrirà la pantalla de configuració d'aquest joc. Omple totes les caselles amb imatges i sons, o només imatges i indica a quina casella esta associada cadascuna.

Per a afegir imatges o sons, clicar els botons marcats amb [...]. Per a seleccionar l'associació selecciona un numero de la llista.

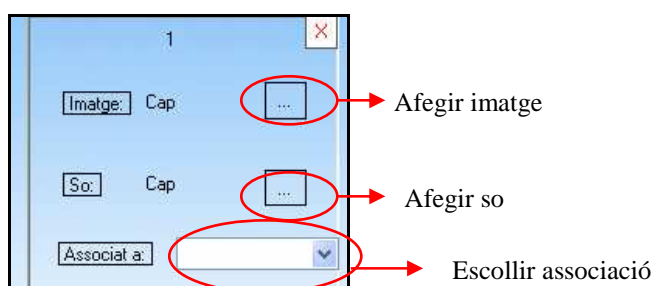


Figura 3: Afegir dades.

Un cop afegides les vuit caselles, introdueix un títol al joc.

Finalitzada la configuració de l'activitat es pot iniciar directament l'activitat clicant el botó Provar.

BuscaImatge:

Si l'activitat seleccionada és la de busca imatge s'obrirà la pantalla de configuració d'aquesta, introdueix els valors a les 5 caselles com en el cas anterior, i afegint l'associació de la única casella de l'esquerra. Un cop finalitzada la edició es pot provar l'activitat prement el botó Provar.

RespostaCorrecte:

Si l'activitat seleccionada és la de resposta correcte, s'obrirà la pantalla de configuració d'aquesta, primer introdueix un so per a la historia clicant el botó afegir historia.

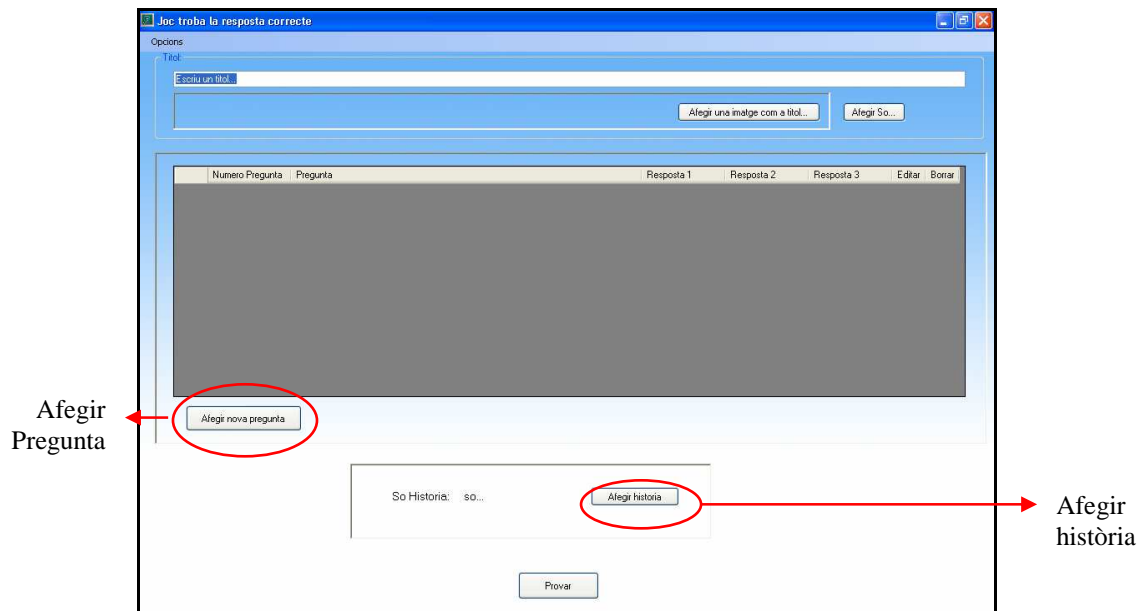


Figura 4: Configuració Resposta Correcte.

Posteriorment, afegeix tantes preguntes com desitgis clicant el botó afegir nova pregunta:

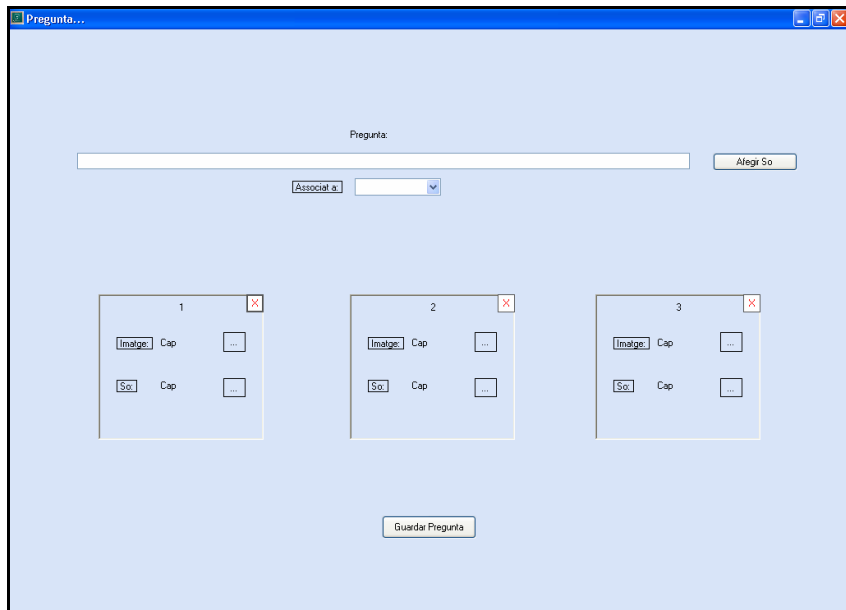


Figura 4: Afegir pregunta.

En la secció de pregunta afegeix un text per a la pregunta i un arxiu per al so, indica a quina resposta està associada seleccionant un nombre de la llista.

Per afegir les tres possibles respostes, afegeix els arxius com en el cas de joc parelles.

Un cop afegida la pregunta prem el botó guardar pregunta, l'aplicació tornarà a la pantalla de configuració on es mostrarà una llista de totes les preguntes que s'han afegit, des d'aquesta llista es poden esborrar les preguntes o modificar-les:

	Numero Pregunta	Pregunta	Resposta 1	Resposta 2	Resposta 3	Editar	Borrar
	1	on esta el gall?	gos dret.GIF	gall.gif	vaca.gif		
	2	quin poble es malaro?	PortMataro.jpg	arenys.jpg	canel.jpg		
	3	on esta la vaca?	gos dret.GIF	ovella.gif	vaca.gif		

Figura 5: Llistat preguntes.

Es poden afegir tantes preguntes com es desitgin. Un cop finalitzada l'edició prémer el botó provar per iniciar el joc.

Memori:

Si s'ha seleccionat aquesta activitat, s'obrirà la pantalla de configuració del memori. Primer selecciona el nivell de dificultat de la llista:

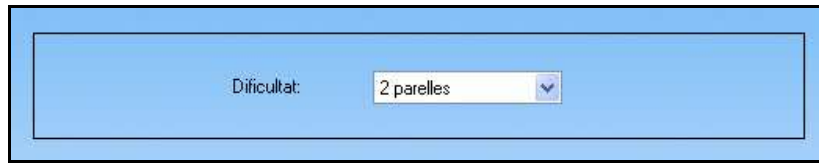


Figura 6: Dificultat.

Posteriorment es faran visibles les caselles necessàries segons el nivell de dificultat escollit, omplir aquestes caselles igual que en el cas de l'activitat parelles.

Un cop completada la edició prémer el botó provar per iniciar el joc.

Laberint:

Si l'activitat seleccionada ha estat el laberint s'obrirà la pantalla de configuració d'aquest. Afegeix el arxiu que farà de personatge, i el so que farà de història segons es mostra en la següent figura:

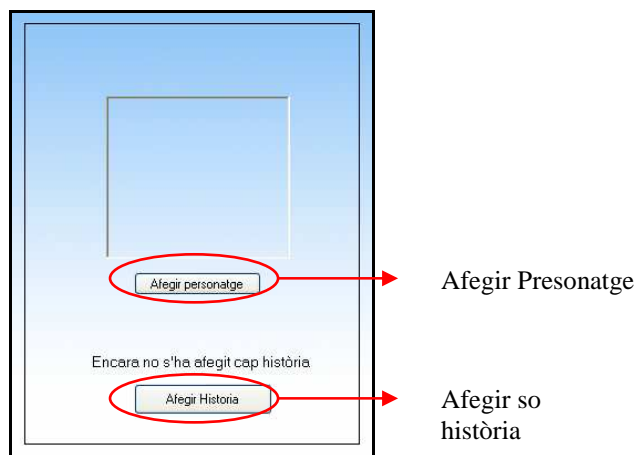


Figura 7: Afegir personatge.

Clicant el botó imprimir laberint, es pot imprimir el laberint seleccionat per tal de utilitzar-lo posteriorment com a plantilla.

Guardar i carregar activitat:

Per a poder guardar o carregar una activitat existent, des del menú opcions de qualsevol joc, seleccionar guardar o carregar segons sigui el cas.

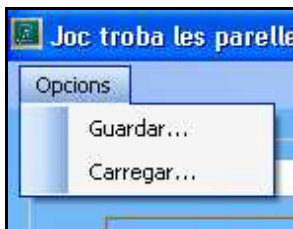


Figura 8: Guardar i Carregar.

Si es vol guardar, s'haurà d'introduir el nom amb el qual es vol que es guardi l'arxiu. Si es vol carregar un joc existent, s'haurà d'indicar quin.

Jugar:

Si es desitja jugar a qualsevol activitat només cal carregar o obrir directament l'arxiu de l'activitat que es vol utilitzar.

Si es vol jugar fent ús de la tauleta digital, cal imprimir les plantilles per posar a sobre d'aquesta, totes les plantilles es poden imprimir directament des del mateix joc. Només cal prémer el botó imprimir de l'activitat i s'imprimirà amb les mides ajustades a la tauleta. En el cas d'activitats del tipus laberint, s'haurà d'imprimir prèviament el laberint desitjat des de la pantalla de configuració.

Per tancar les activitats s'ha de prémer Alt+F4

Desinstal·lació:

Per a desinstal·lar l'aplicació cal executar l'arxiu uninstall.exe que s'ha creat a la carpeta on s'ha instal·lat l'aplicació. Prémer el botó Uninstall i l'aplicació es desinstal·larà.

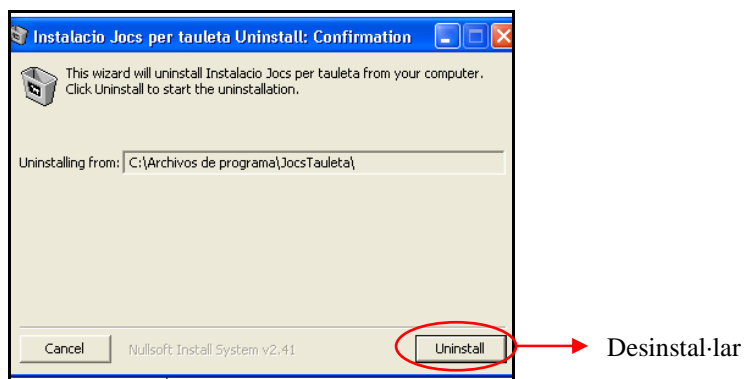


Figura 9: Pantalla de desinstal·lació.

ANNEX II

Contingut del CD

El CD que s'adjunta amb la memòria inclou:

- **Resum:** Un resum del projecte en català, castellà i anglès.
- **Article:** Article-resum de quatre planes del projecte.
- **Memòria:** La memòria en format .pdf
- **Aplicació:** Conté tota la carpeta del projecte de l'aplicació, el codi font, i l'arxiu per instal·lar l'aplicació.
- **Exemples:** Carpetes que contenen jocs ja creats com a exemples.
- **Software necessari:** Visual Studio 2008 Express Edition, Microsoft Framework .NET 3.5

Bibliografia:

Referències bibliogràfiques:

- [1] Francisco Charte Ojeda, *Programación con Visual Basic .NET*, Anaya Multimedia
- [2] Brian Siler & Jeff Spotts, *Visual Basic 6 Edición Especial*, Prentice Hall
- [3] Dossier assignatura Enginyeria del Software 2.
- [4] Xavier Cruz Pla, *Messenger Visual*, Projectes EUPMT Telemàtica, Primavera 2009.
- [5] Xavier Cruz Pla, *Assistent Personal per a persones amb necessitats especials*, Projectes EUPMT Informàtica, Tardor 2008.
- [6] Sara Cortés Viñals, *Recursos per edició d'aplicatius basats en encerclament*, Projectes EUPMT Telemàtica, Tardor 2008.

Referències web:

- [7] <http://software.computadora-discapacidad.org/>
- [8] http://www.antoniosacco.com.ar/docu/pautas_desarrollo_soft.pdf
- [9] <http://forums.microsoft.com/>
- [10] <http://www.recursosvisualbasic.com>
- [11] <http://vbasic.astalaweb.com/>
- [12] <http://www.foro.vb-mundo.com>
- [13] http://nsis.sourceforge.net/File_Association
- [14] <http://www.tecnologiaydiscapacidad.es>
- [15] http://dotnetjunkies.com/WebLog/vbpuntonet/articles/poo_vbnet_3.aspx
- [16] http://www.elguille.info/colabora/NET2005/Percynet_PrintDocument_PrintDialog.htm
- [17] <http://msdn.microsoft.com/>
- [18] <http://bancoimagenes.isftic.mepsyd.es/>

