



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica de Telecomunicació: Especialitat Telemàtica

**RECURSOS PER EDICIÓ D'APLICATIUS BASATS EN
ENCERCLAMENT**

**AUTOR: SARA CORTÉS VIÑALS
PROFESSOR: PERE BARBERAN AGUT**

TARDOR 2008

A la meva família

RESUM

El projecte *Recursos per edició d'aplicatius basats en encerclament* es tracta de la creació d'una aplicació anomenada "A Mida" que permet crear presentacions enfocades a nens amb discapacitat.

Aquesta aplicació consta de sis quadrats que són com botons on ha cada quadrat permet afegir un arxiu ja sigui una imatge, un vídeo, un so o una web, i una imatge a mode de símbol associatiu o icona. A l'hora de reproduir aquesta presentació es pot fer de dues maneres, utilitzant el ratolí de manera convencional, o utilitzant l'encerclament amb el botó dret del ratolí per fer la cerca, i el botó esquerre per fer l'acció. Aquestes presentacions es poden guardar a l'ordinador i reproduir-les o modificar-les posteriorment.

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 *Recursos per edició d'aplicatius basats en encerclament* se trata de la creación de una aplicación llamada "A Mida" que permite crear presentaciones enfocadas a niños con discapacidad.

Esta aplicación consta de seis cuadrados que son como botones donde en cada cuadrado se puede añadir un archivo ya sea una imagen, un video, un sonido o una web, i una imagen a modo de símbolo asociativo o icono. En el momento de reproducir esta presentación se puede hacer de dos formas, utilizando el ratón de forma convencional, o utilizando el barrido con el botón derecho del ratón para hacer la búsqueda, i el botón izquierdo para hacer la acción. Estas presentaciones se pueden guardar en el ordenador i reproducirlas o modificarlas posteriormente.

Para la realización de esta aplicación se ha utilizado el lenguaje Visual Basic .NET i el Visual Studio 2008 Express Edition.

ABSTRACT

Recursos per edició d'aplicatius basats en encerclament essay is based on the creation of an application called "A Mida" which allows to create presentations for disabled children.

This application has six squared buttons. Each button allows to add a file, which can be an image, video, sound or a web, and another image which is an associative symbol or icon. This presentation can be watched in two different ways: using the mouse or the scanning method, where the right button of the mouse is for searching and the left button is to do the action. This presentation has the options of being saved, reproduced or modified.

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 Maresme.....	1
1.2 Tecnologia i discapacitat.....	2
1.2.1 Commutadors.....	2
1.2.2 Encerclament.....	3
2. Objectius.....	5
2.1 Objectius del projecte.....	5
2.2 Objectius personals.....	6
3. Antecedents.....	7
3.1 Programa Toca la Pantalla.....	7
3.2 Programa Descobrir.....	9
3.3 Programa Mirar i tocar.....	10
4. Planificació.....	13
5. Entorn de desenvolupament.....	15
5.1 Java.....	15
5.2 Adobe Director.....	15
5.3 Flash.....	16
5.4 Macromedia Authorware.....	16
5.5 Visual Basic 2008 Express Edition.....	16
5.6 NSIS.....	17
6. Disseny.....	19
6.1 Mode Edició.....	19
6.1.1 Casos d'ús.....	20
6.2 Mode Presentació.....	22
6.2.1 Casos d'ús.....	23
7. Desenvolupament.....	27
7.1 Introducció a VB .NET i Framework .NET.....	27
7.2 Programa A Mida.....	28
7.2.1 Pantalla d'edició.....	28
7.2.2 Pantalla de presentació.....	44
7.3 Instal·lador.....	53

II INDEX

8. Pressupost.....	61
9. Possibles millores.....	63
10. Conclusions.....	65
ANNEX I: Manual d'usuari.....	67
ANNEX II: Contingut del CD.....	71
Bibliografia.....	73

INDEX DE FIGURES

Figura 1: Ratolí adaptat.....	3
Figura 2: Encerclament.....	4
Figura 3: Toca la pantalla 1.....	8
Figura 4: Toca la pantalla 2.....	8
Figura 5: Descobrir 1.....	9
Figura 6: Descobrir 2.....	10
Figura 7: Mirar i tocar.....	11
Figura 8: Pantalla d'edició.....	19
Figura 9: Diagrama Casos d'ús Edició.....	20
Figura 10: Pantalla presentació.....	23
Figura 11: Diagrama Casos d'ús Presentació.....	23
Figura 12: Framework .NET.....	28
Taula 1: Variables principals.....	29
Taula 2: Altres variables.....	30
Figura 13: Afegir arxiu.....	30
Figura 14: Codi Afegir arxiu.....	30
Figura 16: Triar Tipus.....	31
Figura 17: Codi mostrar nom arxiu.....	31
Figura 18: Codi triar tipus (imatge).....	32
Figura 19: Codi triar tipus (so, imatge, vídeo, web).....	33
Figura 20: Carregar arxius.....	33
Figura 21: Codi carregar icona.....	34
Figura 22: Codi carregar arxiu (So, Imatge amb So, Vídeo).....	35
Figura 23: Codi carregar arxiu (Web).....	35
Figura 24: Navegador.....	36
Figura 25: Vista edició amb arxiu introduït.....	36
Figura 26: Encerclament desactivat.....	37
Figura 27: Encerclament activat.....	37
Figura 28: Codi activar/desactivar l'encerclament.....	37
Figura 29: Codi provar presentació.....	38

Figura 30: Opcions.....	38
Figura 31: Exemple arxiu .mida.....	39
Figura 32: Codi guardar presentació.....	40
Figura 33: Mètode AddText.....	41
Figura 34: Codi carregar presentació.....	41
Figura 35: Codi al obrir des de fitxer.....	42
Figura 36: Codi per llegir de l'arxiu .mida.....	43
Figura 37: Codi nova presentació.....	44
Figura 38: Botó amb imatge.....	45
Figura 39: Codi reproduir (Imatge).....	46
Figura 40: Codi reproduir (So i Vídeo).....	46
Figura 41: Reproduir So.....	46
Figura 42: Codi reproduir (Imatge amb so).....	47
Figura 43: Reproduir imatge amb so.....	47
Figura 44: Codi reproduir (Web).....	48
Figura 45: Reproduir Web.....	48
Figura 46: Codi MouseUp quadre 1.....	49
Figura 47: Codi MouseUp quadre 2.....	50
Figura 48: Codi MouseUp quadre 3.....	50
Figura 49: Codi MouseUp quadre 4.....	51
Figura 50: Codi MouseUp quadre 5.....	51
Figura 51: Codi MouseUp quadre 6.....	52
Figura 52: Codi sense encerclament quadre 1.....	52
Figura 53: Codi sense encerclament quadre 2.....	52
Figura 54: Codi sense encerclament quadre 3.....	52
Figura 55: Codi sense encerclament quadre 4.....	53
Figura 56: Codi sense encerclament quadre 5.....	53
Figura 57: Codi sense encerclament quadre 6.....	53
Figura 58: Codi instal·lador (inici).....	54
Figura 59: Codi instal·lador (Secció obligatòria).....	55
Figura 60: Codi instal·lador (Part opcional).....	56
Figura 61: Codi desinstal·lador.....	57
Figura 62: Codi registerExtension.....	58

Figura 63: Codi unregisterExtension.....	59
Taula 3: Recursos Hardware.....	61
Taula 4: Recursos Software.....	61
Taula 5: Recursos Humans.....	61
Taula 6: Pressupost.....	61

1. Introducció

La idea del projecte *Recursos per edició d'aplicatius basats en encerclament* 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 nens en edat escolar amb afectació neurològica que interfereix greument en el seu procés de desenvolupament i aprenentatge.

La idea és oferir un programa on aquests nens puguin navegar a dins d'ell ja sigui mitjançant polsadors o amb el ratolí, i reproduir cançons, imatges, vídeos o webs, prèviament introduïdes a l'aplicació.

Aquesta aplicació els permet tant treballar la resposta amb estímuls causa-efecte, com a obtenir la independència de poder triar dins de la presentació *A Mida* el que volen fer o veure.

1.1. La Fundació 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 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 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.

Els problemes d'aquest disseny es fan encara més evidents si parlem de persones amb discapacitat, quan la impossibilitat, encara que sigui mínima de realitzar algun moviment, tenen efectes catastròfics a l'hora d'utilitzar aquestes eines.

A l'usuari, davant no poder millorar el disseny d'aquests perifèrics, només li queda la possibilitat de adaptar-los o utilitzar perifèrics alternatius.

En el cas de persones amb discapacitat física podríem resoldre el problema configurant la velocitat de resposta del teclat i del ratolí al ritme del usuari, i si això no fos suficient utilitzar ratolins alternatius com serien dos polsadors.

És aquí on s'ha adaptat l'aplicació realitzada en aquest projecte que permet treballar tant amb ratolí com amb polsadors adaptats per tant de poder fer l'encerclament, és a dir, el botó esquerre permet fer el clic convencional del ratolí i el botó dret permet passar d'un control a l'altre, i per tant ens desfem de la necessitat de haver de moure el ratolí, acte que resulta complicat per a les persones amb discapacitat.

1.2.1. Commutadors

Un commutador és un dispositiu que actua com a polsador, i permet ser utilitzat per persones amb discapacitats motrius i sensorials. N'hi ha de diferents formes, mides, que poden funcionar per pressió, per moviment, per aire, etc. amb el que poden adaptar-se a moltes situacions.

Tenen l'objectiu de facilitar l'accés a les persones amb discapacitat motriu, ja que són susceptibles de ser accionats sense cap mena de precisió, i a les persones amb discapacitat psíquica perquè permet disposar d'un entorn amb elements simples amb el que interactuar.

Si s'adapta el ratolí amb dues entrades per a poder connectar aquests commutadors, que emularan les accions assignades del ratolí, la persona que utilitza l'ordinador pot tenir al davant només els commutadors i la pantalla, eliminant així altres elements del seu abast com el teclat, el ratolí, cables, etc.



Figura 1: Ratolí Adaptat.

1.2.2. Encerclament

L'encerclament, “Barrido” en castellà i “Scanning” en anglès, com el seu nom indica serveix per marcar totes les possibilitats actives de la pantalla de manera graduada, és a dir, una després de l'altre. És un sistema adequat per usuaris amb discapacitat motriu i/o amb trastorns de relació amb l'entorn.

Existeixen dos tipus d'encerclament, l'automàtic i el dirigit:

- **Encerclament automàtic:** Només s'utilitza el botó esquerra del ratolí per escollir la opció i va passant d'una zona marcada a una altre de manera automàtica amb una temporalització que generalment pot ser programada per l'usuari.

- **Encerclament Dirigít:** El pas d'una opció marcada a una altre es fa de manera manual, utilitza els botons del ratolí de manera que el dret és el que permet moure's per les diferents opcions, fa les funcions de cerca, i l'esquerre permet escollir-les, fa l'acció.

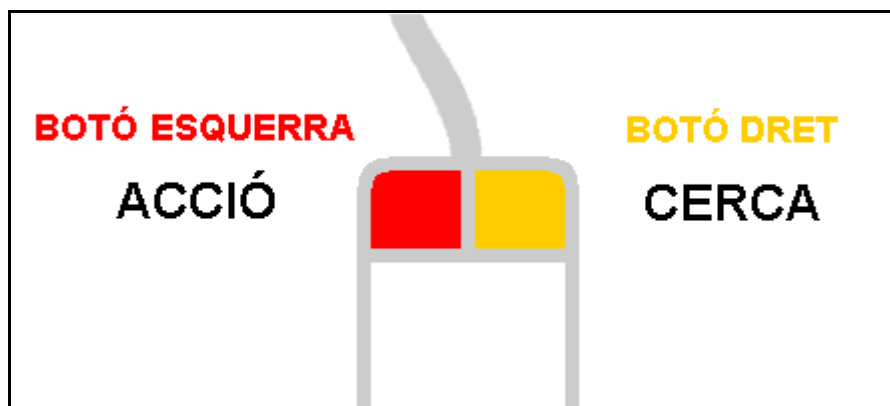


Figura 2: Encerclament.

2. Objectius

2.1. Objectius del projecte

L'objectiu principal d'aquest projecte és oferir un programa obert, on cadascú pugui afegir el contingut (imatge, so, vídeo o web) que desitgi.

Es tracta d'una aplicació enfocada principalment a nens amb discapacitats. Aquesta aplicació ha de constar de dues parts, la part d'edició que serà manipulada per l'usuari a càrrec del nen, i la part de presentació, que serà el resultat final que es veurà.

La part més important d'aquest programa és que sigui una aplicació totalment configurable per l'usuari (pare o educador del nen) i permeti afegir qualsevol arxiu que estigui en el propi ordinador o on-line.

L'aplicació ha de constar de sis quadrats que seran com botons. En el mode d'edició, cada quadrat permetrà afegir una imatge a mode de símbol associatiu, i un arxiu ja sigui la mateixa imatge, un so, un vídeo o una web.

Un cop s'hagi finalitzat l'edició, l'usuari final podrà iniciar la presentació, on podrà navegar dins l'aplicació utilitzant el ratolí, de manera convencional o mitjançant l'encerclament, és a dir, utilitzant el botó dret del ratolí per fer el recorregut d'un botó a l'altre i el botó esquerre per a fer la selecció.

Aquestes presentacions creades amb l'aplicació s'han de poder guardar per a poder recuperar-les posteriorment i utilitzar-les sempre que es vulgui.

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. L'objectiu fonamental per a mi és aconseguir crear una aplicació que compleixi tot el que demanen i sobretot que per aquests nens i persones amb discapacitat els hi sigui útil el que jo he fet.

Un altre objectiu personal d'aquest projecte és aprendre un llenguatge de programació, per a mi nou, com el Visual Basic, que potser si no hagués fet aquest projecte no l'hauria d'haver utilitzat i no n'hagués après.

3. Antecedents

Avui en dia existeixen alguns programes enfocats a persones amb discapacitat, la majoria creats per fundacions o usuaris que cedeixen el codi a qualsevol que el vulgui utilitzar o millorar.

Aquí recollim algun d'aquests programes cercats per la web, que són semblants al realitzat en aquest projecte i han ajudat a decidir les eines que posteriorment seràn utilitzades per a la realització d'aquest.

3.1. Programa Toca la Pantalla

Aquest programa consisteix en treballar la causa-efecte, es tracta de que al fer clic a la pantalla, apareix una imatge acompanyada d'un so, es a dir, ofereix un estímul de resposta a l'actuació de l'usuari.

Existeixen dues formes d'utilitzar-lo:

Toca abans: L'usuari toca la pantalla i apareix una imatge acompanyada d'un só, (opcional) on a posat el cursor i a fet clic.

L'altre forma és toca després: apareix la imatge automàticament i l'usuari a de fer el clic sobre d'ella per a que aparegui una altre imatge, acompanyada de so també.

Permet configuracions, es pot seleccionar les imatges i els sons, el color de fons de la pantalla, etc.

Està desenvolupat amb Visual Basic i el codi font està disponible.

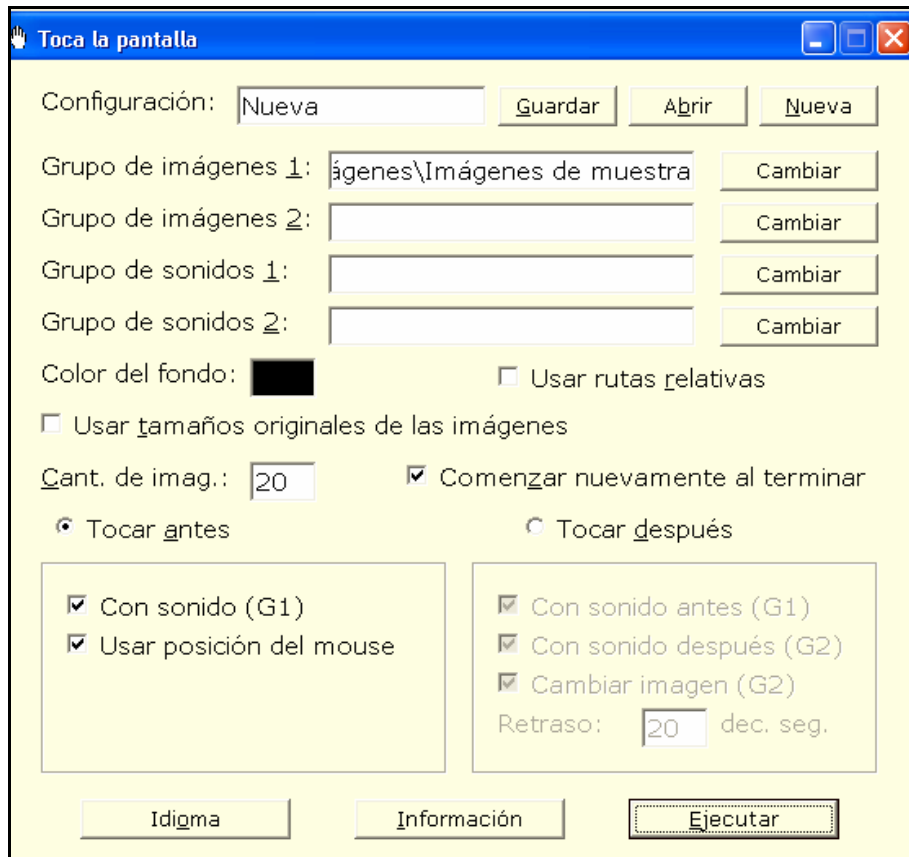


Figura 3: Toca la pantalla 1.

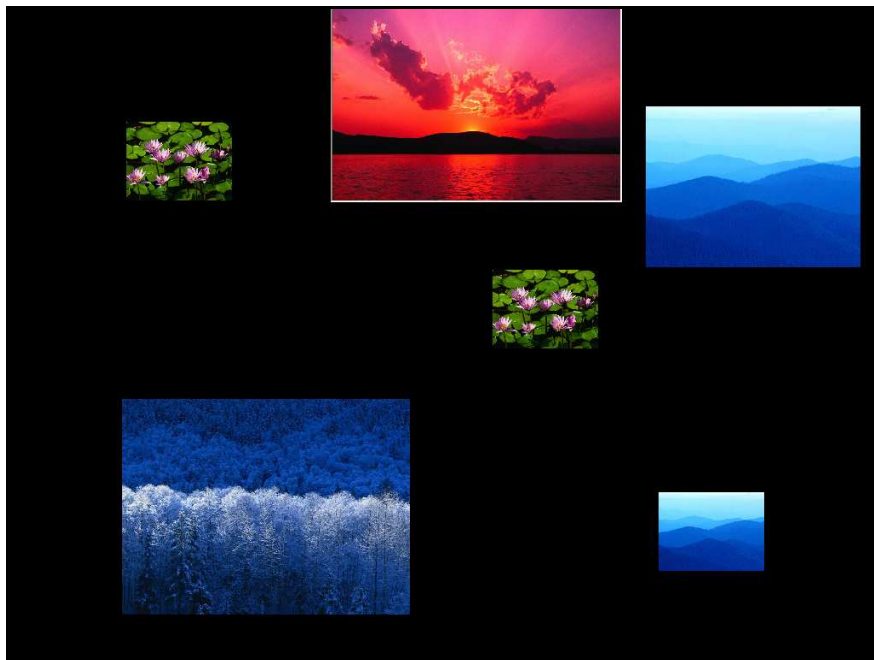


Figura4: Toca la pantalla 2.

3.2. Programa Descobrir

Aquest programa consisteix en afegir una imatge que es divideix en cel·les, tantes com escollim i al fer clic va apareixent la imatge amagada.

Permet escollir la imatge, els sons al fer clic i el so de la imatge final. Es pot guardar per a tornar a carregar-la posteriorment. També es pot escollir el nombre de cel·les en que queda dividida la pantalla.

Desenvolupat amb visual basic i codi font disponible.

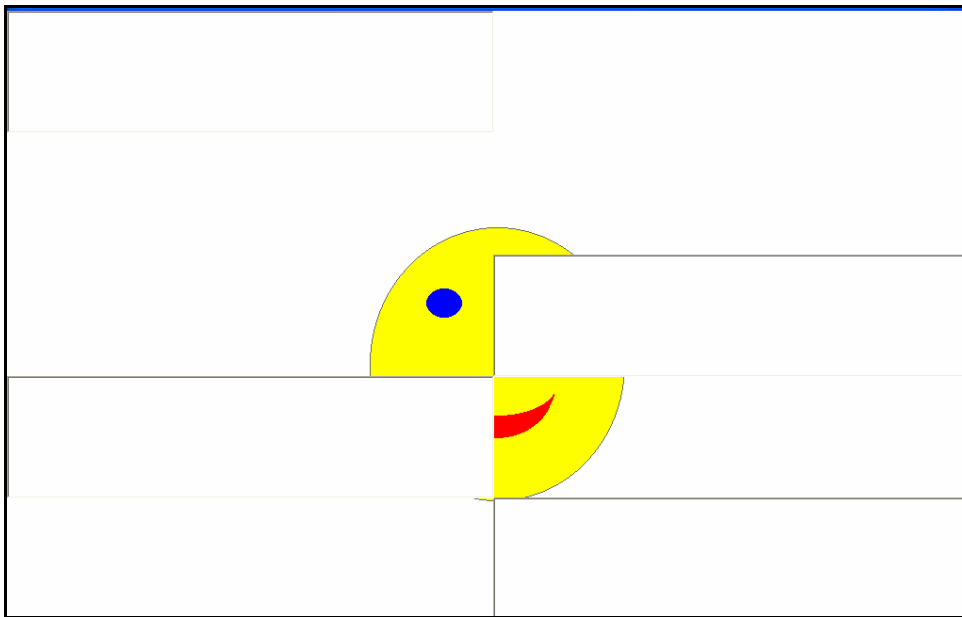


Figura 5: Descobrir 1.

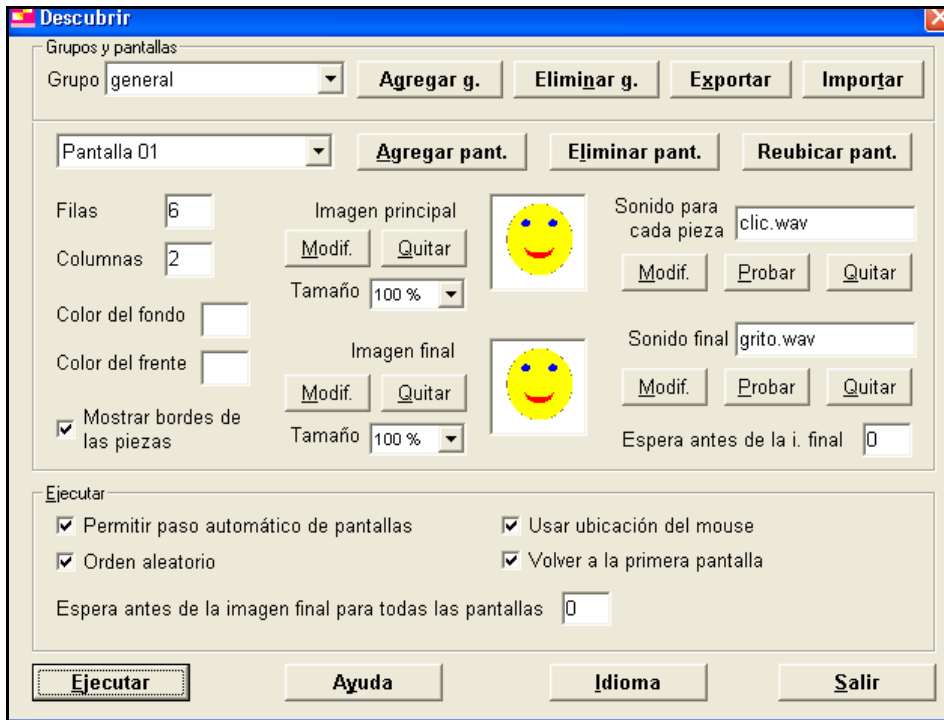


Figura 6: Descubrir 2.

3.3. Programa Mirar i tocar

Apareix a la pantalla un cercle o una figura i fins que no se li faci clic no apareix un altre estímul. Genera informe final amb els temps de resposta, etc.

No té codi font disponible.

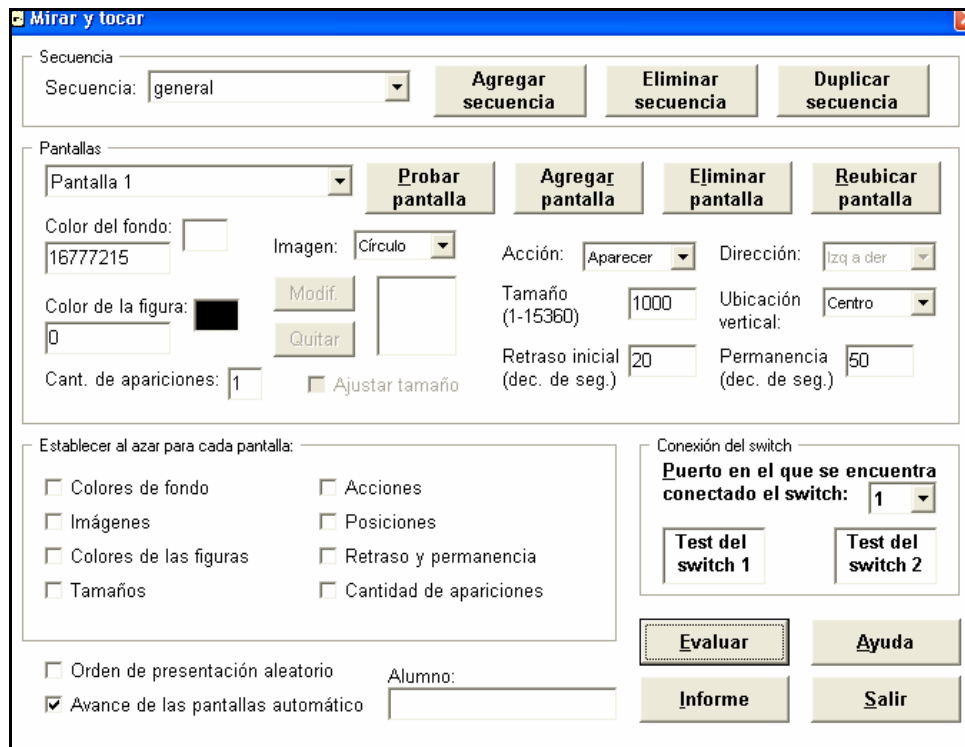


Figura 7: Mirar i tocar.

De la llista de aplicacions consultades, s'han seleccionat aquestes 3 anteriors com ha més importants, ja que eren orientades a nens amb discapacitats i eren totalment configurables, que era un dels principals objectius de la nostra aplicació.

Tot i ser aplicacions totalment configurables cap d'elles permet realitzar l'encerclament, part imprescindible de l'aplicació, aquesta mancança motiva a la realització d'un programa totalment nou el qual permet-hi configuracions i a més a més l'opció d'utilitzar l'encerclament.

4. Planificació

La planificació per a la realització del projecte s'ha realitzat de la següent manera:

- 1r. Reunió amb el tutor del projecte i la responsable de la Fundació Maresme per a fixar els objectius del projecte.
- 2n. Cerca d'antecedents de l'aplicació per a obtenir idees de com realitzar-lo.
- 3r. Cerca de software existent que permeti realitzar l'aplicació. I triar-ne un.
- 4t. Obtenir documentació sobre el software que s'utilitzarà, en aquests cas, obtenir tota la documentació necessària de Visual Basic 2008.
- 5é. Desenvolupament de l'aplicació. Aquesta és la part més important del projecte, que s'ha anat fent paral·lela a la memòria i fent periòdiques visites amb els responsables de la Fundació Maresme, per a poder obtenir la informació necessària i els canvis o millores que s'haguessin d'anar realitzant.
- 6é. Un cop finalitzada gairebé l'aplicació, se li va donar una copia a la responsable de la Fundació Maresme per que la proves, tant ella com els nens i detectés possibles errades o millores, i així poder fer una segona iteració en el desenvolupament de l'aplicació
- 7é. Finalment la creació de l'instal·lador amb la versió final del programa.
- 8é. Redacció de la memòria. Que s'ha anat fent paral·lelament amb el desenvolupament de l'aplicació.
- 9é. Per últim. Un cop finalitzada l'aplicació, la realització d'un manual d'usuari que anirà juntament amb el programa.

5. Entorn de desenvolupament

La primera part de realització del projecte és decidir quin software s'utilitzarà per al desenvolupament de l'aplicació. A continuació es citaran les possibilitats i perquè s'ha triat el que finalment s'ha utilitzat.

5.1. Java

Utilitzar java és una de les possibilitats per a realitzar aquesta aplicació, ja que és el llenguatge més utilitzat en les assignatures de la carrera, i el que més és domina.

El programa a realitzar ha de ser una aplicació multimèdia que ha de tractar amb vídeo, web, sons, etc, per tant s'ha mirat també per ha treballar conjuntament amb el Java un conjunt de llibreries anomenada SDL: Simple DirectMedia Layer, desenvolupades amb C++ que proporcionen funcions bàsiques per a realitzar operacions de dibuix en 2D, gestió de efectes de so i musica, carregar i gestió de imatges i vídeos.

Encara i estar programada amb C++ permet utilitzar altres llenguatges com el java, el visual basic, etc. Proporciona eines per a desenvolupar aplicacions multimedia.

Es tracta d'una llibreria multiplataforma, és a dir, que tant es suportada amb windows com amb linux, macOS, etc.

5.2. Adobe Director

Aquest software permet generar presentacions multimèdia, permet incorporar pel·lícules, múltiples tipus de mitjans con imatges amb format jpeg, bmp, png, gif, psd, tiff, etc, vídeo mov, avi, et, sons wav, aiff, etc o animacions en Flash. Inclou editors bàsics de text, mapa de bits, vectors i sons. També permet programació amb el llenguatge Lingo.

És un software utilitzat per la Fundació Maresme alhora de realitzar les seves aplicacions web amb encerclament, per tant també era una bona opció.

5.3. Flash

Flash és una aplicació en forma d'estudi d'animació que treballa sobre fotogrames destinats a la producció i entrega de contingut interactiu i multimèdia multiplataforma.

Adobe Flash incorpora el llenguatge de programació ActionScript.

Com Director, es un software que utilitza la Fundació Maresme per a realitzar les seves aplicacions per a nens amb discapacitat.

5.4. Macromedia Authorware

S'usa per a crear programes interactius que poden integrar sons, textos, gràfics, animacions simples i pel·lícules. Es poden integrar arxius flash i director, permet programar en JavaScript.

El Macromedia Authorware s'ha triat com una de les millors eines per a realitzar la aplicació però finalment s'ha descartat per que Macromedia ha deixat de desenvolupar-la i la versió última existent és bastant antiga, i tampoc existeix gaire documentació de com utilitzar-lo.

5.5. Visual Basic 2008 Express Edition

Finalment s'ha decidit utilitzar Visual Basic 2008 Express Edition, tot i ser la primera vegada que es programa en Visual Basic, és un codi del qual existeix molta documentació i que permet afegir característiques multimèdia amb facilitat, permeten per exemple afegir el windows media player per a reproduir sons i vídeos, i també permet afegir amb facilitat un navegador web.

Visual Basic és una programació molt visual que permet programar interfícies gràfiques amb facilitat i per les necessitats d'aquest projecte és necessari un entorn de

desenvolupament que permeti programar gràficament 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 programar l'instal·lador, però no es creu que sigui un problema ja que existeixen múltiples alternatives per a crear aquest instal·lador.

5.6 NSIS

Com s'ha mencionat a l'apartat anterior, amb el Visual Basic 2008 Express Edition no es pot programar l'instal·lador, i per tant s'ha optat per utilitzar NSIS (NullSoft Scriptable Install System), un programa OpenSource que mitjançant un script permet crear l'instal·lador del programa amb facilitat.

6. Disseny

L'aplicació consta de dos modes d'utilització, una és la part d'edició, utilitzada per els pares o educadors del nen. I l'altre mode és el de presentació, que seria el resultat final a la edició anterior.

6.1. Mode Edició

Aquest mode és la pantalla que permet afegir els arxius que es desitgin. De la idea sorgida de la Fundació Maresme, aquesta pantalla ha de tenir 6 quadrats i permetre guardar els canvis realitzats en un arxiu .mida.

S'ha realitzat la pantalla de tal manera que permet afegir la icona com a símbol associatiu, i també l'arxiu que posteriorment en el mode de presentació, es reproduirà.

Aquest mode també permet seleccionar si l'arxiu final serà, utilitzant l'encerclament, o de manera convencional utilitzant el ratolí.

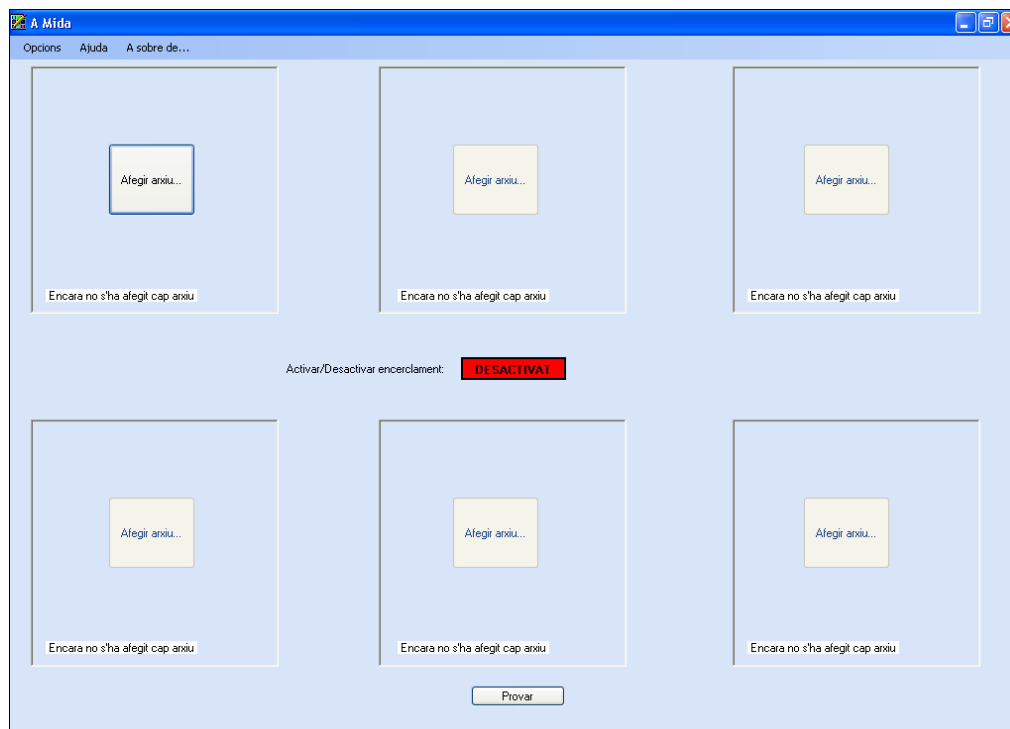


Figura 8: Pantalla d'edició.

6.1.1. Casos d'ús

Breu explicació dels casos d'ús.

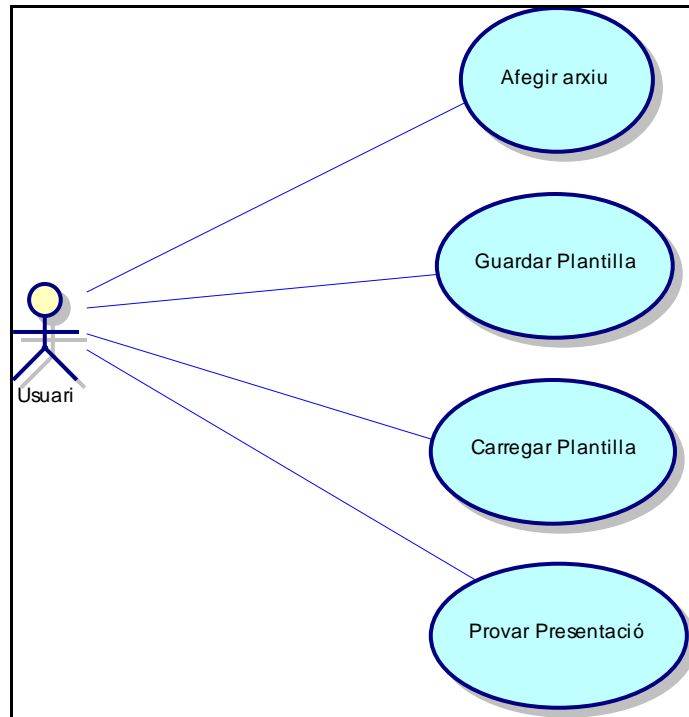


Figura 9: Diagrama Casos d'ús Edició.

- **Cas d'ús Afegir arxiu:**

Actor: L'usuari que manipula el programa en mode edició.

Descripció: L'usuari afegeix els arxius que desitgi en l'aplicació.

Flux Normal:

1. L'usuari prem el botó d'afegir arxiu.
2. Apareix la pantalla de selecció de tipus d'arxiu i tria imatge.
3. L'usuari introdueix la imatge que desitja buscant-la a l'ordinador.

4. L'usuari repeteix els passos 1, 2 i 3 tantes vegades com quadrats desitgi omplir de la presentació.

Flux Alternatiu:

- 2.a. A la pantalla de selecció l'usuari escull so, imatge amb so, o vídeo
- 3.a. L'usuari introdueix la imatge que farà d'icona, i el arxiu a reproduir, ja sigui so o vídeo
- 2.b. A la pantalla de selecció l'usuari escull web.
- 3.b. L'usuari introdueix la imatge que farà d'icona i executa el navegador que permetrà cercar i triar la web.

• **Cas d'ús Guardar plantilla:**

Actor: L'usuari que manipula el programa en mode edició.

Descripció: L'usuari vol guardar la presentació.

Flux Normal:

1. L'usuari prem l'opció guardar plantilla.
2. L'usuari tria el nom i la ruta on es guardarà l'arxiu.
3. L'usuari prem guardar i la plantilla es guarda.

Flux Alternatiu:

- 1.a. L'usuari encara no ha executat cap vegada el cas d'ús afegir arxiu, i per tant apareix un missatge que no deixa guardar la plantilla ja que encara es buida.

• **Cas d'ús Carregar plantilla:**

Actor: L'usuari que manipula el programa en mode edició.

Descripció: L'usuari obre una presentació en mode edició guardada prèviament.

Flux Normal:

1. L'usuari prem l'opció carregar plantilla.
2. Tria el arxiu que desitja obrir buscant-lo al ordinador.
3. L'arxiu s'obre i es visualitza a la pantalla en mode edició.

- **Cas d'ús Provar presentació:**

Actor: L'usuari que manipula el programa en mode edició.

Descripció: L'usuari obté una visió preliminar de la presentació.

Flux Normal:

1. L'usuari prem el botó provar.
2. S'obre la pantalla en mode presentació i l'usuari pot visualitzar-la.

Flux Alternatiu:

2.a. L'usuari encara no ha executat cap vegada el cas d'ús afegir arxiu i per tant no hi ha cap arxiu que reproduir a la presentació, apareix un missatge que ho notifica.

6.2. Mode Presentació

Aquesta part és el resultat final que veurà el nen, en un principi constava sempre de 6 quadrats, per finalment s'ha decidit de permetre que l'usuari en mode edició n'afegeixi els que vulgui, per tant el resultat final pot variar depenent de els quadrats que s'hagin omplert.

Hi ha dues maneres d'accedir a aquesta pantalla. Des de el mode d'edició, utilitzant el botó provar, on podrem veure un vista prèvia del resultat final. O un cop l'arxiu .mida ja s'ha guardat, obrint aquest directament.

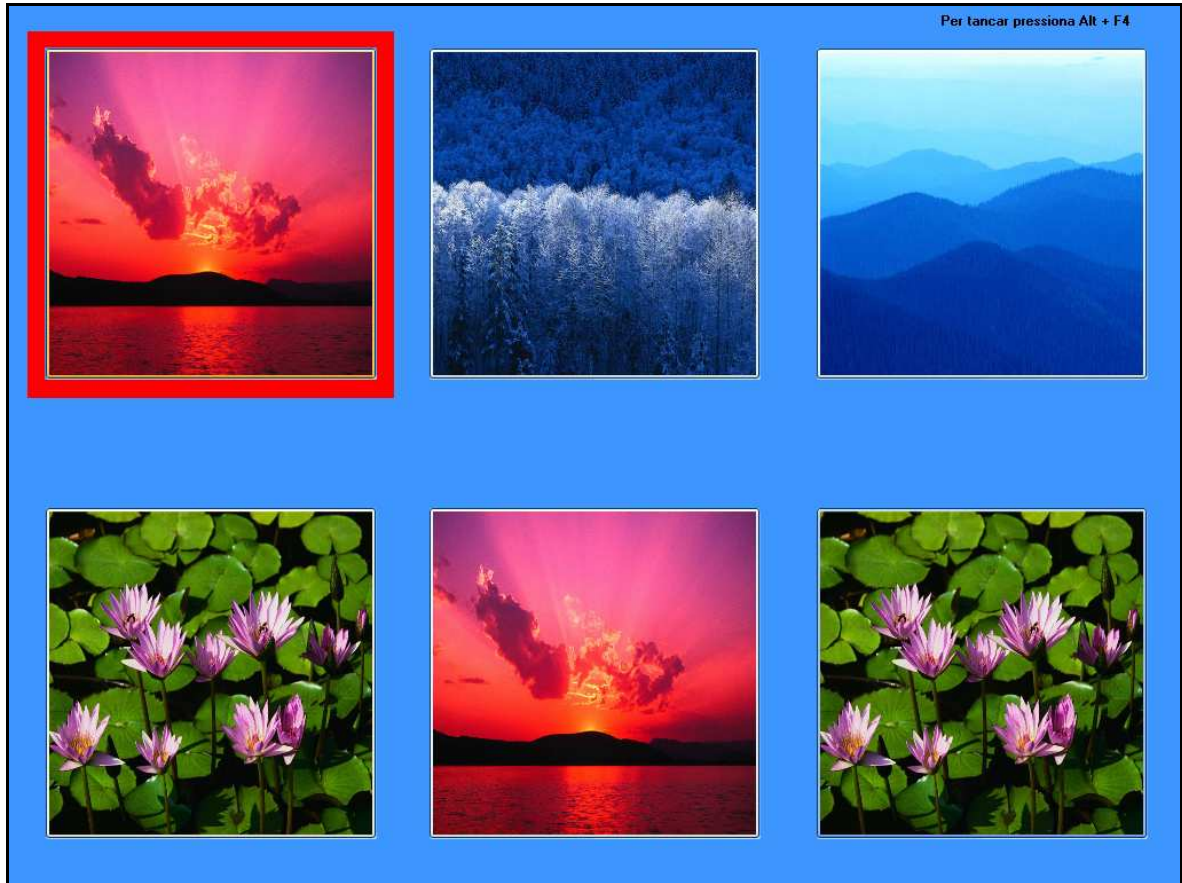


Figura 10: Pantalla presentació.

6.2.1. Casos d'ús:

Breu explicació dels casos d'ús.

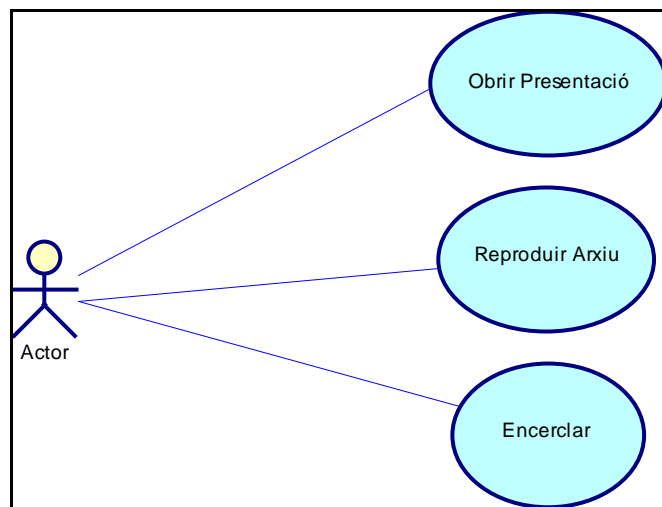


Figura 11: Diagrama Casos d'ús Presentació.

- **Cas d'ús obrir presentació:**

Actor: Nen o responsable

Descripció: S'executa la presentació.

Flux Normal:

1. Sobre l'arxiu .mida que es desitgi.
2. S'inicia la presentació.

- **Cas d'ús Reproduir Arxiu:**

Actor: Nen

Descripció: S'obre i reproduceix l'arxiu que hi ha associat al botó.

Flux Normal:

1. El nen clica sobre el botó que desitgi reproduir.
2. S'obre la imatge a pantalla completa.
3. El nen torna a fer clic amb el botó dret o esquerre i es tanca la pantalla i torna a la de presentació.

Flux Alternatiu:

2.a. Si l'arxiu associat al botó es un so, aquest es reproduceix. Si és un vídeo es reproduceix el vídeo a pantalla completa.

2.b. Si l'arxiu associat es una web, s'obre el navegador i carrega la web.

- **Cas d'ús Encerclar:**

Actor: Nen

Descripció: El nen va seleccionant els botons mitjançant el botó dret i el botó esquerre del ratolí, o amb dos polsadors.

Flux Normal:

1. Es prem el botó esquerre del ratolí i s'inicia el cas d'ús Reproduir arxiu.
2. Es prem el botó dret del ratolí i es passa a seleccionar el següent botó de la pantalla.
3. Es repeteixen els passos 1 i 2 tantes vegades com es desitgi.
4. Es prem Alt + F4, i el programa es tanca i finalitza.

7. Desenvolupament

7.1. Introducció a VB .NET i Framework .NET

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

1. Dissenyar la interfície. El formulari, col·locant en ell els controls i objectes necessaris.
2. Dona 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 com a l'hora 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 altre banda, la biblioteca de classes .NET Framework proporciona una jerarquia de classes orientades a objectes 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.

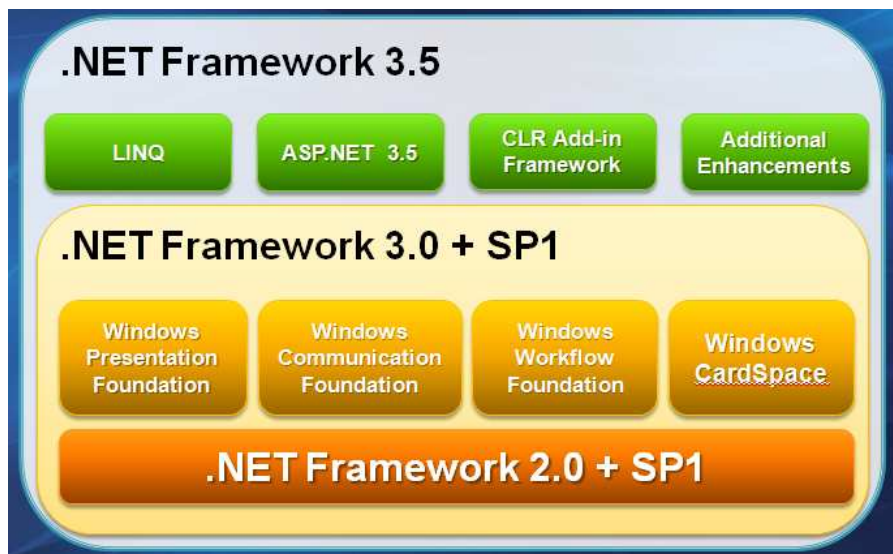


Figura 12: Framework .NET

7.2. Programa A Mida

A continuació s'explicarà com ha estat desenvolupat el programa, les variables utilitzades, i els mètodes més rellevants. No s'afegirà el codi complet de l'aplicació sinó les parts que s'han considerat més importants.

7.2.1. Pantalla d'edició

El programa és basa principalment en tres variables, que són les que guarden totes les dades, que es mostren a la següent taula:

Nom variable	Tipus	Mida	Valors
RutaArxiu	Array de String	6 posicions (0,5)	Guarda la ruta de l'arxiu que l'usuari afegeix.
RutaIcona	Array de String	6 posicions (0,5)	Guarda la ruta de la imatge que l'usuari escull com a icona.
Tipus	Array de String	6 posicions (0,5)	Guarda el tipus d'arxiu que s'ha afegit, pot prendre els valors de "Web", "Imatge", "Vídeo", "So" o "ImatgeSo".

Taula 1: Variables principals.

També hi ha altres variables utilitzades en l'aplicació que són les següents:

Nom variable	Tipus	Mida	Valors
NomArxiu	Array de String	6 posicions (0,5)	Guarda el nom de l'arxiu que l'obté de la variable RutaArxiu.
NomIcona	Array de String	6 posicions(0,5)	Guarda el nom de l'arxiu utilitzat com icona.
Encerclament	Booleà		True o False
ObertAmbArxiu	Booleà		True o False
Mida	Integer		Indica el nombre de quadrats utilitzats, 6 com a màxim

Taula 2: Altres variables.

Dins d'aquesta pantalla d'edició l'usuari clica sobre el primer afegir arxiu, ja que la resta encara no estan activats per a que l'usuari estigui obligat a introduir els arxius en ordre.



Figura 13: Afegir arxiu.

```
'Obre la pantalla per afegir un arxiu al quadrat.
Private Sub Afegir1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Afegir1.Click
    Abrir.Abrir(0)
    Afegir2.Enabled = True
End Sub
```

Figura 14: Codi Afegir arxiu.

Al prémer sobre aquest botó es genera l'event buttonclick que fa que s'obri el següent formulari que permet escollir quin arxiu es desitja afegir, cada botó té el seu propi event que executa el mètode per afegir, i passa com a paràmetre la posició del botó, en aquest cas 0 ja que és el primer botó.

Al cridar el mètode obrir s'executa el següent codi:

```
Public Sub Abrir(ByVal posicio As Integer)
    Me.Posicio = posicio
    Me.Visible = True
End Sub
```

Figura 15: Codi Obrir.

Que fa visible la pantalla per escollir el tipus d'arxiu i indica a la classe quina es la posició des de la qual s'ha cridat al mètode.

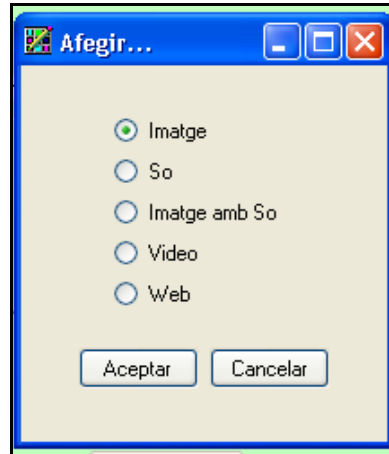


Figura 16. Triar Tipus.

Permet triar entre les diferents opcions: Imatge, So, Imatge amb So, Vídeo i Web.

Depenent la opció que es triï, s'executarà un mètode o un altre.

Si l'opció triada es Imatge, s'obre el quadre de diàleg que permet seleccionar la imatge triada. S'ha posat com a carpeta d'inici c:\.

Al haver triat una imatge, la icona i l'arxiu són les mateixes, per tant es guarda a les variables RutaArxiu i RutaIcona, els mateixos valors (Figura 18). Un cop carregada la imatge es posa al lloc on toqui, depenent de la posició, de fons del botó per a que l'usuari sàpiga quin arxiu ha afegit, i també escriu al quadrat del botó el nom de l'arxiu afegit, aquest codi és comú per a tots els tipus d'arxiu. El següent codi es el que escriu el nom de l'arxiu a la pantalla (Figura 17).

```
Public Sub MostrarNomArxiu(ByVal posicio As Integer)
    Select Case posicio
        Case 0
            Form1.Nom1.Text = Form1.NomArxiu(0)
        Case 1
            Form1.Nom2.Text = Form1.NomArxiu(1)
        Case 2
            Form1.Nom3.Text = Form1.NomArxiu(2)
        Case 3
            Form1.Nom4.Text = Form1.NomArxiu(3)
        Case 4
            Form1.Nom5.Text = Form1.NomArxiu(4)
        Case 5
            Form1.Nom6.Text = Form1.NomArxiu(5)
    End Select
End Sub
```

Figura 17: Codi mostrar nom arxiu.

```

Private Sub OK_Button_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles OK_Button.Click
    If Imatge.Checked Then
        Form1.Tipus(Posicio) = "Imatge"
        AbrirArchivo.InitialDirectory = "c:\"
        AbrirArchivo.Filter =
"JPG(*.jpg)|*.jpg|PNG(*.png)|*.png|GIF(*.gif)|*.gif|Todos(*.Jpg,
*.Png, *.Gif, *.Tiff, *.Jpeg, *.Bmp)|*.Jpg; *.Png; *.Gif; *.Tiff;
*.Jpeg; *.Bmp"
        AbrirArchivo.FilterIndex = 1
        AbrirArchivo.RestoreDirectory = True
        If AbrirArchivo.ShowDialog() =
System.Windows.Forms.DialogResult.OK Then
            Try
                miStream = AbrirArchivo.OpenFile()
                If (miStream IsNot Nothing) Then
                    Form1.RutaIcona(Posicio) =
AbrirArchivo.FileName.ToString()
                    Form1.RutaArxiu(Posicio) =
AbrirArchivo.FileName.ToString()
                    Form1.NomArxiu(Posicio) =
Path.GetFileName(Form1.RutaArxiu(Posicio))
                    Form1.NomIcona(Posicio) =
Path.GetFileName(Form1.RutaArxiu(Posicio))
                    Select Case Posicio
                        Case 0
                            Form1.Panel1.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(0))
                        Case 1
                            Form1.Panel2.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(1))
                        Case 2
                            Form1.Panel3.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(2))
                        Case 3
                            Form1.Panel4.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(3))
                        Case 4
                            Form1.Panel5.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(4))
                        Case 5
                            Form1.Panel6.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(5))
                    End Select
                    Abrir2.MostrarNomArxiu(Posicio)
                End If
            Catch Ex As Exception
                MessageBox.Show("Error: No es pot carregar la
imatge " & Ex.Message)
            End Try
        End If
    End If

```

Figura 18: Codi triar tipus (imatge).

Si l'opció triada es So, Imatge amb So, o Vídeo, (Figura 19) s'obre una altre pantalla en la qual hem de triar la imatge que farà d'icona, i l'arxiu que volem introduir (el So o el Vídeo).

```

ElseIf So.Checked Then
    Form1.Tipus(Posicio) = "So"
    Abrir2.Abrir2(Posicio)

ElseIf ImatgeSo.Checked Then
    Form1.Tipus(Posicio) = "ImatgeSo"
    Abrir2.Abrir2(Posicio)

ElseIf Video.Checked Then
    Form1.Tipus(Posicio) = "Video"
    Abrir2.Abrir2(Posicio)

ElseIf Web.Checked Then
    Form1.Tipus(Posicio) = "Web"
    Abrir2.Abrir2(Posicio)
End If
Me.Close()
End Sub

```

Figura 19: Codi triar tipus (so, imatge, vídeo, web).



Figura 20: Carregar arxius.

Quan premem carregar icona (Figura 21), s'obre el mateix quadre de diàleg per afegir una imatge, i ens permet cercar-la a l'ordinador.

```

Private Sub CargarFondo_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CargarFondo.Click
    OpenFileDialog1.InitialDirectory = "c:\"
    OpenFileDialog1.Filter =
"JPG(*.jpg)|*.jpg|PNG(*.png)|*.png|GIF(*.gif)|*.gif|Todos(*.Jpg,
*.Png, *.Gif, *.Tiff, *.Jpeg, *.Bmp)|*.Jpg; *.Png; *.Gif; *.Tiff;
*.Jpeg; *.Bmp"
    OpenFileDialog1.FilterIndex = 1
    OpenFileDialog1.RestoreDirectory = True
    If OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK Then
        Try
            miStream = OpenFileDialog1.OpenFile()
            If (miStream IsNot Nothing) Then
                Form1.RutaIcona(Posicio) =
OpenFileDialog1.FileName.ToString()
                Form1.NomIcona(Posicio) =
Path.GetFileName(Form1.RutaIcona(Posicio))
                Select Case Posicio
                    Case 0
                        Form1.Panel1.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(0))
                    Case 1
                        Form1.Panel2.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(1))
                    Case 2
                        Form1.Panel3.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(2))
                    Case 3
                        Form1.Panel4.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(3))
                    Case 4
                        Form1.Panel5.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(4))
                    Case 5
                        Form1.Panel6.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(5))
                End Select

            End If
            Catch Ex As Exception
                MessageBox.Show("Error: No es pot carregar la
imatge " & Ex.Message)
            End Try
        End If
    End Sub

```

Figura 21: Codi carregar icona.

Posteriorment premem carregar arxiu (Figura 22) i mostra el quadre de diàleg per buscar un arxiu al ordinador i depenent del tipus que sigui filtra pel tipus d'extensió, és a dir, si és un so mostra els arxius .mp3, si es un vídeo filtra pel format .avi, encara que permet mostrar tots els tipus d'arxius.

```

Private Sub CargarArxiu_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CargarArxiu.Click
    If Form1.Tipus(Posicio) = "So" Or Form1.Tipus(Posicio) =
"Video" Or Form1.Tipus(Posicio) = "ImatgeSo" Then
        OpenFileDialog1.InitialDirectory = "c:\"
        If Form1.Tipus(Posicio) = "So" Or Form1.Tipus(Posicio)
= "ImatgeSo" Then
            OpenFileDialog1.Filter = "mp3 files
(*.mp3)|*.mp3|All files (*.*)|*.*"
        ElseIf Form1.Tipus(Posicio) = "Video" Then
            OpenFileDialog1.Filter = "avi files
(*.avi)|*.avi|All files (*.*)|*.*"
        End If
        OpenFileDialog1.RestoreDirectory = True
        If OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK Then
            Try
                miStream = OpenFileDialog1.OpenFile()
                If (miStream IsNot Nothing) Then
                    Form1.RutaArxiu(Posicio) =
OpenFileDialog1.FileName.ToString()
                    Form1.NomArxiu(Posicio) =
Path.GetFileName(Form1.RutaArxiu(Posicio))
                    MostrarNomArxiu(Posicio)
                End If
            Catch Ex As Exception
                MessageBox.Show("Error: No es pot carregar la
imatge" & Ex.Message)
            End Try
        End If
    End If

```

Figura 22: Codi carregar arxiu (So, Imatge amb So, Vídeo).

Per última opció si el que s'ha triat a sigut web la icona la carrega igual que abans, però al pressionar carregar arxiu (Figura 23) obre un petit navegador desenvolupat (Figura 24) que permet cercar per internet la web i guardar el link d'aquesta pressionant l'opció guardar web.

```

ElseIf Form1.Tipus(Posicio) = "Web" Then
    AbrirWeb.Abrir(Posicio)
End If
Me.Close()
End Sub

```

Figura 23: Codi carregar arxiu (Web).

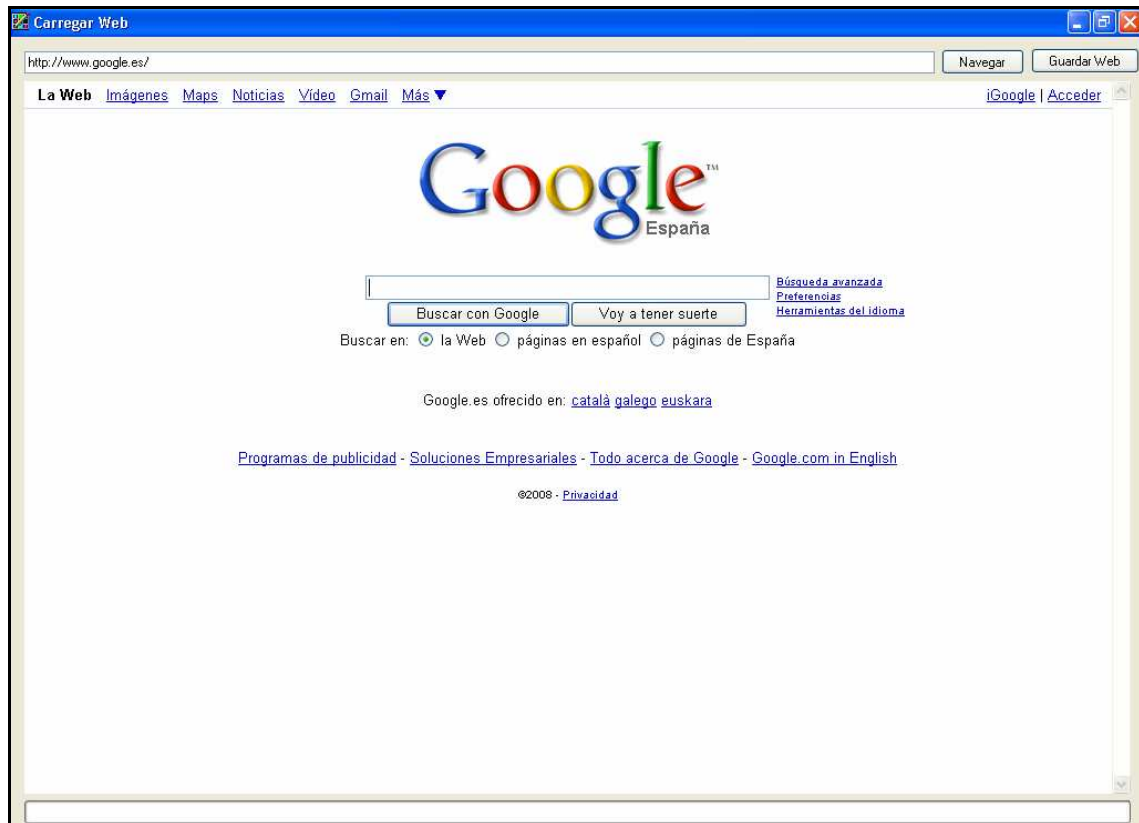


Figura 24: Navegador.

L'aplicació permet afegir fins a 6 botons ja que és el disseny que s'havia decidit, però també permet afegir-ne menys.

Quan carreguem la icona i l'arxiu, la pantalla ens mostra la icona que apareixerà i el nom de l'arxiu introduït:



Figura 25: Vista edició amb arxiu introduït.

Tal com s'ha desenvolupat l'aplicació, el que fa és guardar totes les rutes dels arxius i de les icones en les variables RutaArxiu i RutaIcona abans esmentades, on cada posició fa referència a quin quadrat s'han associat aquells arxius.

L'usuari pot decidir si vol que el programa funcioni utilitzant l'encerclament (Figura 27) o de manera convencional (Figura 26), per fer-ho a de modificar l'opció d'encerclament que es mostra al mig de la pantalla que el que fa és modificar un booleà que també queda emmagatzemat.

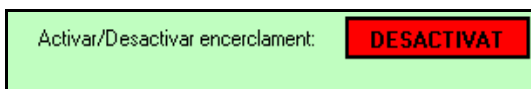


Figura 26: Encerclament desactivat.

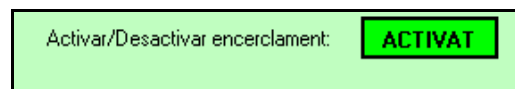


Figura 27: Encerclament activat.

El codi associat a aquests mètodes és el següent:

```
'Modifica l'opció de utilitzar encerclament o no.
Private Sub CheckBoxEncerclament_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBoxEncerclament.CheckedChanged
    If CheckBoxEncerclament.Checked Then
        Encerclament = True
        CheckBoxEncerclament.Text = "ACTIVAT "
    Else
        Encerclament = False
        CheckBoxEncerclament.Text = "DESACTIVAT "
    End If
End Sub
```

Figura 28: Codi activar/desactivar l'encerclament.

Un cop l'usuari creu que ha finalitzat la seva edició pot provar la presentació premen el botó de provar (Figura 29), al fer això s'executa el segon formulari que seria passar a mode presentació on ja podem reproduir els arxius introduïts.

```

'Executa el programa.
Private Sub Provar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Provar.Click
    If RutaArxiu(0) = "" And RutaArxiu(1) = "" And RutaArxiu(2)
= "" And RutaArxiu(3) = "" And RutaArxiu(4) = "" And RutaArxiu(5) =
"" And RutaIcona(0) = "" And RutaIcona(1) = "" And RutaIcona(2) =
"" And RutaIcona(3) = "" And RutaIcona(4) = "" And RutaIcona(5) =
"" Then
        MessageBox.Show("Has d'omplir almenys un quadrat avans
d'iniciar", "Atenció")
    Else
        If RutaArxiu(0) <> "" And RutaIcona(0) <> "" Then
            Form2.Cuadro1.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(0))
            Form2.Cuadro1.Visible = True
        End If
        If RutaArxiu(1) <> "" And RutaIcona(1) <> "" Then
            Form2.Cuadro2.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(1))
            Form2.Cuadro2.Visible = True
        End If
        If RutaArxiu(2) <> "" And RutaIcona(2) <> "" Then
            Form2.Cuadro3.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(2))
            Form2.Cuadro3.Visible = True
        End If
        If RutaArxiu(3) <> "" And RutaIcona(3) <> "" Then
            Form2.Cuadro4.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(3))
            Form2.Cuadro4.Visible = True
        End If
        If RutaArxiu(4) <> "" And RutaIcona(4) <> "" Then
            Form2.Cuadro5.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(4))
            Form2.Cuadro5.Visible = True
        End If
        If RutaArxiu(5) <> "" And RutaIcona(5) <> "" Then
            Form2.Cuadro6.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(5))
            Form2.Cuadro6.Visible = True
        End If
        Form2.Visible = True
    End If
End Sub

```

Figura 29: Codi provar presentació.

També pot guardar la plantilla (Figura 32), crear una plantilla nova (Figura 37), o carregar alguna plantilla ja existent (Figura 34).



Figura 30: Opcions.

A l'hora de guardar les plantilles s'ha decidit fer-ho guardant només les rutes dels arxius i de les icones, ja que s'ha considerat que fer còpies directes dels arxius, podria dur com a conseqüència que els arxius fossin massa pesats, ja que per exemple s'hi podrien afegir 6 pel·lícules, i aquestes 6 pel·lícules estarien duplicades a l'ordinador.

L'únic problema que s'ha trobat de treballar amb les rutes ha estat que s'ha de treballar sempre en local, l'arxiu *A Mida* ha d'haver estat fet al teu ordinador amb els teus arxius, si no, no funcionarà.

Aquestes rutes es guarden amb una extensió pròpia que se l'hi ha anomenat *.mida* (Figura 31), en aquest arxiu també es guarda informació sobre si s'ha d'utilitzar l'encerclament o no. En resum, aquests arxius guarden els arrays comentats anteriorment de les rutes, els icones i els tipus, i a més a més si hi ha o no encerclament.

A continuació podem veure un exemple d'un arxiu *.mida*:

```

5
C:\Projecte\Puesta de sol.jpg
C:\Projecte\03.jpg
C:\Projecte\Colinas azules.jpg
C:\Projecte\los-simpson-aniversario.gif
C:\Projecte\Nenúfares.jpg
C:\Projecte\Invierno.jpg
C:\Projecte\Puesta de sol.jpg
C:\Projecte\Zapatillas\06 - Vuelve.mp3
C:\Projecte\Zapatillas\01 - Canciones.mp3
C:\Projecte\los simpson 16x13 - homer-movil.avi
www.google.es
C:\Projecte\Invierno.jpg
Imatge
So
ImatgeSo
Video
Web
Imatge
False

```

Figura 31: Exemple arxiu .mida

El primer que es guarda són el nombre de quadrats que s'han omplert, a continuació, la seqüència d'icones escollides, posteriorment els arxius afegits, després el tipus d'arxiu i per últim True o False que indica si hi ha encerclament o no.

El que fa el programa es llegir tantes línies com indica el primer nombre (començant per 0) i les afegeix primer al array d'icones, després a l'array d'arxius i després al array de tipus.

```
'Mètode per a guardar la plantilla
Private Sub GuardarToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
GuardarToolStripMenuItem.Click
    If RutaArxiu(0) = "" And RutaArxiu(1) = "" And RutaArxiu(2)
= "" And RutaArxiu(3) = "" And RutaArxiu(4) = "" And RutaArxiu(5) =
"" And RutaIcona(0) = "" And RutaIcona(1) = "" And RutaIcona(2) =
"" And RutaIcona(3) = "" And RutaIcona(4) = "" And RutaIcona(5) =
"" Then
        MessageBox.Show("Has d'omplir almenys un quadrat avans
de guardar", "Atenció")
    Else
        Mida = RutaArxiu.Length - 1
        SaveFileDialog1.Filter = "Mida |*.mida"
        SaveFileDialog1.Title = "Guardar plantilla "
        SaveFileDialog1.DefaultExt = ".mida"
        SaveFileDialog1.ShowDialog()
        If SaveFileDialog1.FileName <> "" Then
            Dim arxiu As System.IO.FileStream = CType _
(SaveFileDialog1.OpenFile(), System.IO.FileStream)
            AddText(arxiu, Mida & Environment.NewLine)
            For i = 0 To Mida
                AddText(arxiu, RutaIcona(i) &
Environment.NewLine)
            Next
            For i = 0 To Mida
                AddText(arxiu, RutaArxiu(i) &
Environment.NewLine)
            Next
            For i = 0 To Mida
                AddText(arxiu, Tipus(i) & Environment.NewLine)
            Next
            If Encerclament Then
                AddText(arxiu, "True" & Environment.NewLine)
            Else
                AddText(arxiu, "False" & Environment.NewLine)
            End If
        End If
    End If
End Sub
```

Figura 32: Codi guardar presentació.

El mètode privat AddText (Figura 33), afegeix a l'arxiu creat el text que passem com a paràmetre.

```
'Mètode que escriu en l'arxiu les rutes de la plantilla
Private Shared Sub AddText(ByVal fs As FileStream, ByVal value As
String)
    Dim info As Byte() = New UTF8Encoding(True).GetBytes(value)
    fs.Write(info, 0, info.Length)
End Sub
```

Figura 33: Codi mètode AddText.

Un cop guardada una plantilla, i ha dues opcions de reproduir-la, carregant-la directament des de l'aplicació amb l'opció carregar plantilla i el programa la recupera en mode edició, o obrint directament el arxiu que hem guardat, que s'obre ja en mode presentació, ja que com veurem més endavant el programa associa la extensió .mida a ell mateix.

Els mètodes utilitzats per a carregar la plantilla són els següents:

```
'Mètode per a carregar un arxiu .mida guardat
Private Sub CarregarToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CarregarToolStripMenuItem.Click
    Dim miStream As IO.FileStream = Nothing
    OpenFileDialog1.Filter = "Mida |*.mida"
    If OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK Then
        Try
            miStream = OpenFileDialog1.OpenFile()
            If (miStream IsNot Nothing) Then
                Dim Ruta = OpenFileDialog1.FileName.ToString()
                Dim Archivo = Path.GetFileName(Ruta)
                'Llegeix del arxiu .mida guardat
                LleguirArxiu(Archivo)
            End If
        Catch Ex As Exception
            MessageBox.Show("Error: No es pot obrir l'arxiu " &
Ex.Message)
        End Try
    End If
End Sub
```

Figura 34: Codi carregar presentació.

El programa filtra els arxius que es puguin escollir utilitzant la extensió .mida, per tant només deixa obrir aquest tipus d'arxius.

I l'altre mètode és el que detecta si hi ha paràmetres d'entrada, és a dir, que si s'ha obert fent clic directament a l'arxiu .mida per tant obre el programa en mode presentació directament.

El que fa és generar l'event clic del botó provar i així executar l'aplicació.

```
'Comprova si té parametres d'entrada (S'ha obert desde un fitxer)
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim Arxiu As String
    'Comprovem que no estigui buit
    If Not Command() = vbNullString Then
        Arxiu = Trim(Command)
        'Comprovem que no tingui ""
        If Instr(1, Arxiu, Chr(34)) <> 0 Then
            Arxiu = Mid(Arxiu, 2, Len(Arxiu) - 2)
        End If
        LleguirArxiu(Arxiu) 'Crida al metode lleguir arxiu
        ObertAmbArxiu = True ' Indica que el programa s'executa
per un arxiu ja existent
        Provar.PerformClick()
        Me.Visible = False
    Else
        Me.Visible = True
    End If
End Sub
```

Figura 35: Codi al obrir des de fitxer.

Ambdós mètodes anteriors criden al mètode privat LleguirArxiu() (Figura 36) que té com a paràmetre l'arxiu .mida que es vol obrir, i el mètode llegeix els valors d'aquest. També carrega les icones i els noms de l'arxiu per a que ho vegi l'usuari.

```

'Llegeix del arxiu .mida guardat
Private Sub LleguirArxiu(ByVal arxiu As String)
    Dim Boolea As Boolean
    Dim ArchivoLeer As StreamReader = New StreamReader(arxiu)
    Mida = ArchivoLeer.ReadLine
    For i = 0 To Mida
        RutaIcona(i) = ArchivoLeer.ReadLine()
    Next
    For i = 0 To Mida
        RutaArxiu(i) = ArchivoLeer.ReadLine()
    Next
    For i = 0 To Mida
        Tipus(i) = ArchivoLeer.ReadLine()
    Next
    Boolea = ArchivoLeer.ReadLine() 'Llegeix l'atribut boolean
de si es amb encerclament o no
    If Boolea = "True" Then
        Encerclament = True
        CheckBoxEncerclament.Checked = True
    ElseIf Boolea = "False" Then
        Encerclament = False
    End If
    ArchivoLeer.Close()
    ArchivoLeer.Dispose()
    If RutaIcona(0) <> "" Then
        Me.Panell1.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(0))
        Nom1.Text = Path.GetFileName(RutaArxiu(0))
    End If
    If RutaIcona(1) <> "" Then
        Me.Panel2.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(1))
        Nom2.Text = Path.GetFileName(RutaArxiu(1))
    End If
    If RutaIcona(2) <> "" Then
        Me.Panel3.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(2))
        Nom3.Text = Path.GetFileName(RutaArxiu(2))
    End If
    If RutaIcona(3) <> "" Then
        Me.Panel4.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(3))
        Nom4.Text = Path.GetFileName(RutaArxiu(3))
    End If
    If RutaIcona(4) <> "" Then
        Me.Panel5.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(4))
        Nom5.Text = Path.GetFileName(RutaArxiu(4))
    End If
    If RutaIcona(5) <> "" Then
        Me.Panel6.BackgroundImage =
System.Drawing.Image.FromFile(RutaIcona(5))
        Nom6.Text = Path.GetFileName(RutaArxiu(5))
    End If
End Sub

```

Figura 36: Codi per llegir de l'arxiu .mida.

Apart de guardar o carregar, també tenim la opció de desfer tot el que hem fet clicant sobre Nova Plantilla, que torna a deixar-ho tot buit, tal i com estava inicialment. Per a començar a treballar de nou.

```
'Torna a posar tot buit per a començar a fer una presentació de
nou.
Private Sub NovaToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NovaToolStripMenuItem1.Click
    Array.Clear(NomArxiu, 0, 5)
    Array.Clear(RutaArxiu, 0, 5)
    Array.Clear(NomIcona, 0, 5)
    Array.Clear(RutaIcona, 0, 5)
    Array.Clear(Tipus, 0, 5)
    Panel1.BackgroundImage = Nothing
    Panel2.BackgroundImage = Nothing
    Panel3.BackgroundImage = Nothing
    Panel4.BackgroundImage = Nothing
    Panel5.BackgroundImage = Nothing
    Panel6.BackgroundImage = Nothing
    Nom1.Text = "Encara no s'ha afegit cap arxiu"
    Nom2.Text = "Encara no s'ha afegit cap arxiu"
    Nom3.Text = "Encara no s'ha afegit cap arxiu"
    Nom4.Text = "Encara no s'ha afegit cap arxiu"
    Nom5.Text = "Encara no s'ha afegit cap arxiu"
    Nom6.Text = "Encara no s'ha afegit cap arxiu"
    CheckBoxEncerclament.CheckState = CheckState.Unchecked
End Sub
```

Figura 37: Codi nova presentació.

7.2.2. Pantalla de presentació

La pantalla de presentació és la que mostrarà els arxius introduïts anteriorment, a pantalla completa. Com ja s'ha comentat, s'accedeix a ella de dues formes, una és utilitzant el botó provar del formulari d'edició, i l'altre obrint directament una plantilla guardada.

Aquesta pantalla es pot utilitzar de dues formes diferents, amb encerclament o sense, depenent de la configuració que hagi triat l'usuari.

Cada botó permet reproduir el contingut afegit prèviament. A continuació es mostra un exemple per a cada tipus d'arxiu.

- **Botó amb imatge:**



Figura 38: Botó amb imatge.

Quan premem el botó que té associat la imatge, s'obre un formulari que el que fa és mostrar la imatge a pantalla completa. Tant si utilitzem l'encerclament com si no, aquesta pantalla es tancarà quan cliquem amb el ratolí el botó dret o l'esquerre.

Distingeix amb quin botó del ratolí s'ha fet el clic, si és amb el dret passa el focus del programa al següent control, i si és amb el botó esquerre crida al mètode reproduir que s'explica a continuació:

```
Private Sub Reproduir(ByVal Posicio As Integer)
    If (Form1.Tipus(Posicio) = "Imatge") Then
        Imagen.PictureBox1.ImageLocation =
Form1.RutaArxiu(Posicio)
        Imagen.Visible = True
    End If
End Sub
```

Figura 39: Codi reproduir (Imatge).

Aquesta és la part del mètode Reproduir() que indica que si el tipus és "Imatge" es mostrarà la imatge a pantalla completa.

- **Botó amb so:**

Per a reproduir sons i vídeos s'ha optat per la opció d'afegir una component del windows media player a l'aplicació, i així permetre reproduir aquests arxius.

Quan pressionem aquest botó s'executa el windows media player amb el so introduït.

```

        ElseIf (Form1.Tipus(Posicio) = "So" Or Form1.Tipus(Posicio)
= "Video") Then
            Video.WindowsMediaPlayer1.URL =
Form1.RutaArxiu(Posicio)
            Cursor.Position = New Point(Cursor.Position.X - 50,
Cursor.Position.Y + 400)
            Video.Visible = True
    
```

Figura 40: Codi reproduir (So i Video).

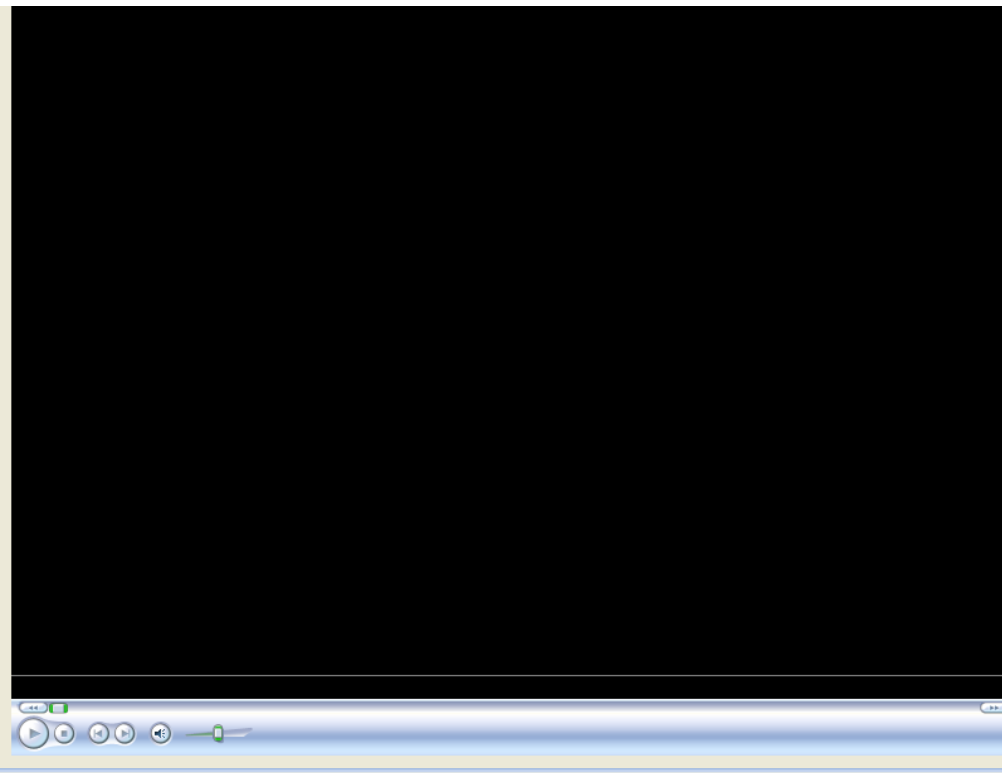


Figura 41: Reproduir So.

- **Botó amb imatge amb so:**

Aquesta opció, es una unió de les dues anteriors, el que fa es mostrar la imatge afegida com a icona a pantalla completa i reproduir el so que s'hagi afegit.

La part del mètode reproduir que afecta aquesta opció és la següent:

```

ElseIf (Form1.Tipus(Posicio) = "ImatgeSo") Then
    ImagenSo.BackgroundImage =
System.Drawing.Image.FromFile(Form1.RutaIcona(Posicio))
    ImagenSo.WindowsMediaPlayer1.URL =
Form1.RutaArxiu(Posicio)
    ImagenSo.Visible = True

```

Figura 42: Codi reproduir (Imatge amb so).

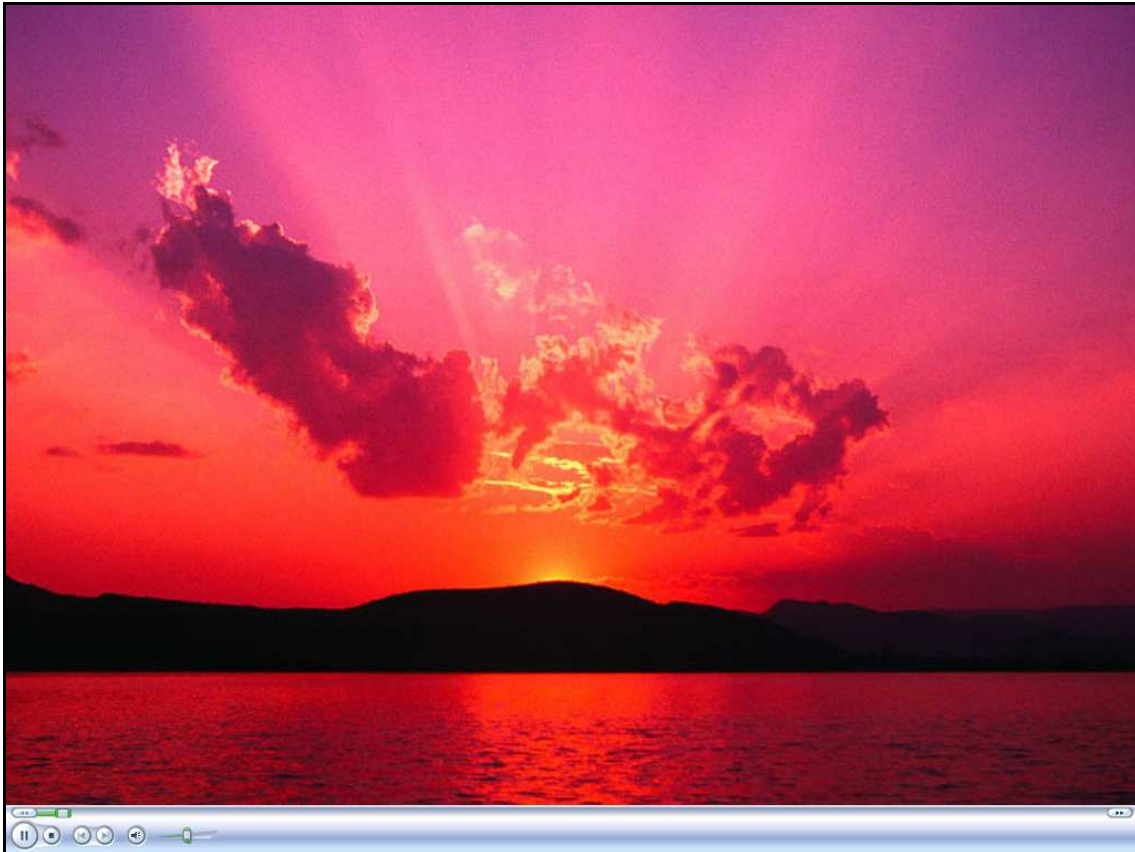


Figura 43: Reproduir imatge amb so.

- **Botó amb vídeo:**

Aquesta opció, com la del só, també utilitza la component del windows media player, i al clicar sobre el botó reproduceix el vídeo a pantalla completa que s'hagi introduït, per a tancar també cal clicar amb el botó dret o esquerra del ratolí.

S'utilitza el mateix fragment de codi que a l'hora de reproduir un so, ja que s'obre de la mateixa manera tant sigui un so, com un vídeo.

- **Botó amb web:**

L'últim tipus d'arxiu que permet afegir són les webs. Al clicar sobre el botó que té associada una web, s'obre un navegador a pantalla completa que et deixa utilitzar el ratolí, per si el que s'ha afegit es algun joc on-line per exemple. Inicialment si la opció triada era encerclament, el ratolí quedava amagat i si feies clic la web es tancava, aquesta opció es va considerar que no era apropiada, ja que per a moltes webs era necessari poder utilitzar el ratolí. Finalment s'ha decidit de fer-lo visible.

El codi del mètode reproduir per aquest tipus d'arxiu és:

```

ElseIf (Form1.Tipus(Posicio) = "Web") Then
    Web.WebBrowser1.Navigate(Form1.RutaArxiu(Posicio))
    Web.Visible = True
    Cursor.Show()
End If
End Sub

```

Figura 44. Codi reproduir (Web).

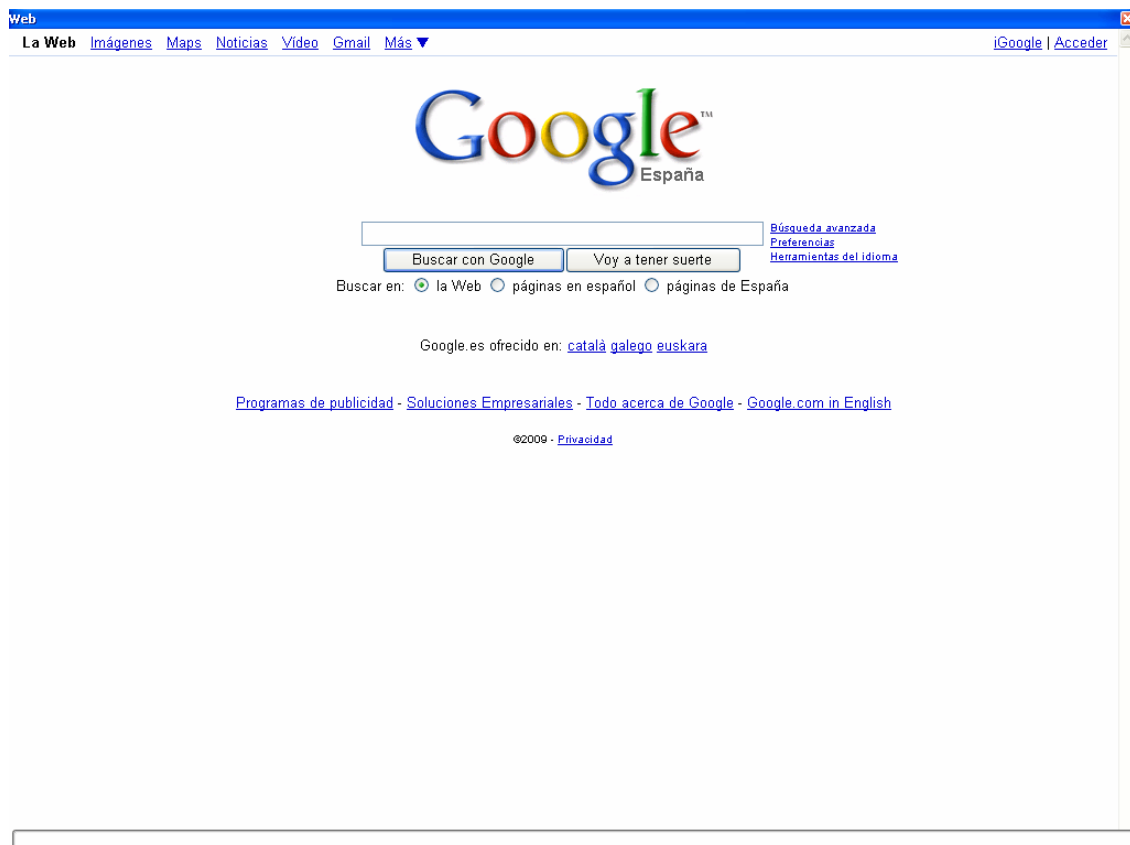


Figura 45: Reproduir Web.

- **Encerclament:**

Per a que el programa permeti realitzar l'encerclament s'ha realitzat de la següent manera, s'han afegit al formulari de presentació 6 imatges vermelles, associades a cada quadrat.

Cada quadrat té el seu event MouseUp, que detecta quan s'ha deixat de prémer un botó del ratolí, i permet distingir si ha sigut el botó dret o el botó esquerre, partint d'aquesta condició el programa executa una part del codi o una altre, executant per tant la part de reproduir si s'ha premut el botó esquerre, o realitzant el salt al següent quadrat, fent desaparèixer la imatge vermella del quadrat actual i posant a true la propietat visible del següent. Com es mostra en els següents events de cada quadrat en concret.

```

Private Sub Cuadro1_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Cuadro1.MouseUp
    If Form1.Encerclament Then
        If e.Button = MouseButton.Left Then
            Me.Cuadro1.Focus()
            Me.Reproducir(0)
        ElseIf e.Button = MouseButton.Right Then
            If Form1.RutaArxiu(1) = "" Then
                Me.Seleccio1.Visible = True
                Cursor.Clip = New Rectangle(Cuadro1.Location,
Cuadro1.Size)
            Else
                Me.Seleccio1.Visible = False
                Me.Seleccio2.Visible = True
                Cursor.Clip = New Rectangle(Cuadro2.Location,
Cuadro2.Size)
            End If
        End If
    End If
End Sub

```

Figura 46: Codi MouseUp quadre 1.

```

Private Sub Cuadro2_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Cuadro2.MouseUp
    If Form1.Encerclament Then
        If e.Button = MouseButton.Left Then
            Me.Cuadro2.Focus()
            Me.Reproduir(1)

            ElseIf e.Button = MouseButton.Right Then
                Me.Seleccio2.Visible = False
                If Form1.RutaArxiu(2) = "" Then
                    Me.Seleccio1.Visible = True
                    Cursor.Clip = New Rectangle(Cuadro1.Location,
Cuadro1.Size)
                Else
                    Me.Seleccio3.Visible = True
                    Cursor.Clip = New Rectangle(Cuadro3.Location,
Cuadro3.Size)
                End If
            End If
        End If
    End Sub

```

Figura 47: Codi MouseUp quadre 2

```

Private Sub Cuadro3_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Cuadro3.MouseUp
    If Form1.Encerclament Then
        If e.Button = MouseButton.Left Then
            Me.Cuadro3.Focus()
            Me.Reproduir(2)

            ElseIf e.Button = MouseButton.Right Then
                Me.Seleccio3.Visible = False
                If Form1.RutaArxiu(3) = "" Then
                    Me.Seleccio1.Visible = True
                    Cursor.Clip = New Rectangle(Cuadro1.Location,
Cuadro1.Size)
                Else
                    Me.Seleccio4.Visible = True
                    Cursor.Clip = New Rectangle(Cuadro4.Location,
Cuadro4.Size)
                End If
            End If
        End If
    End Sub

```

Figura 48: Codi MouseUp quadre 3.

```

Private Sub Cuadro4_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Cuadro4.MouseUp
    If Form1.Encerclament Then
        If e.Button = MouseButton.Left Then
            Me.Cuadro4.Focus()
            Me.Reproduir(3)

        ElseIf e.Button = MouseButton.Right Then
            Me.Seleccio4.Visible = False
            If Form1.RutaArxiu(4) = "" Then
                Me.Seleccio1.Visible = True
                Cursor.Clip = New Rectangle(Cuadro1.Location,
Cuadro1.Size)
            Else
                Me.Seleccio5.Visible = True
                Cursor.Clip = New Rectangle(Cuadro5.Location,
Cuadro5.Size)
            End If
        End If
    End If
End Sub

```

Figura 49: Codi MouseUp quadre 4.

```

Private Sub Cuadro5_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Cuadro5.MouseUp
    If Form1.Encerclament Then
        If e.Button = MouseButton.Left Then
            Me.Cuadro5.Focus()
            Me.Reproduir(4)

        ElseIf e.Button = MouseButton.Right Then
            Me.Seleccio5.Visible = False
            If Form1.RutaArxiu(5) = "" Then
                Me.Seleccio1.Visible = True
                Cursor.Clip = New Rectangle(Cuadro1.Location,
Cuadro1.Size)
            Else
                Me.Seleccio6.Visible = True
                Cursor.Clip = New Rectangle(Cuadro6.Location,
Cuadro6.Size)
            End If
        End If
    End If
End Sub

```

Figura 50: Codi MouseUp quadre 5.

```

Private Sub Cuadro6_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Cuadro6.MouseUp
    If Form1.Encerclament Then
        If e.Button = MouseButton.Left Then
            Me.Cuadro6.Focus()
            Me.Reproduir(5)

        ElseIf e.Button = MouseButton.Right Then
            Me.Seleccio6.Visible = False
            Me.Seleccio1.Visible = True
            Cursor.Clip = New Rectangle(Cuadro1.Location,
Cuadro1.Size)
        End If
    End If
End Sub

```

Figura 51: Codi MouseUp quadre 6.

Si l'opció triada no ha sigut l'encerclament, els events que s'utilitzen, són els següents, ja que tenen en compte el clic directe del ratolí sobre el quadrat escollit.

```

Private Sub Cuadro1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cuadro1.Click
    If Not Form1.Encerclament Then
        Reproduir(0)
    End If
End Sub

```

Figura 52: Codi sense encerclament quadre 1.

```

Private Sub Cuadro2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cuadro2.Click
    If Not Form1.Encerclament Then
        Reproduir(1)
    End If
End Sub

```

Figura 53: Codi sense encerclament quadre 2.

```

Private Sub Cuadro3_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cuadro3.Click
    If Not Form1.Encerclament Then
        Reproduir(2)
    End If
End Sub

```

Figura 54: Codi sense encerclament quadre 3.

```

Private Sub Cuadro4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cuadro4.Click
    If Not Form1.Encerclament Then
        Reproduir(3)
    End If
End Sub

```

Figura 55: Codi sense encerclament quadre 4.

```

Private Sub Cuadro5_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cuadro5.Click
    If Not Form1.Encerclament Then
        Reproduir(4)
    End If
End Sub

```

Figura 56: Codi sense encerclament quadre 5.

```

Private Sub Cuadro6_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cuadro6.Click
    If Not Form1.Encerclament Then
        Reproduir(5)
    End If
End Sub

```

Figura 57: Codi sense encerclament quadre 6.

7.3. Instal·lador

Com ja s'ha comentat anteriorment, la versió Express del Visual Studio 2008 no permet realitzar l'instal·lador, per tant s'ha optat per utilitzar un programa lliure que permet mitjançant un script crear l'instal·lador de l'aplicació.

El programa utilitzat a sigut el NSIS (Nullsoft Scriptable Install System) 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 la extensió .mida creada per al nostre programa a aquest mateix, per a que el sistema la reconegui i l'obri directament amb el programa *A Mida*.

```

; -----
;
; Instal·lador del projecte
;
; Autora: Sara Cortes Viñals
; -----

!include "registerExtension.nsh"

; Nom del instal·lador
Name "Instal·lacio AMida"

; Nom del exe resultant
OutFile "AMidaInstall.exe"

; Directori per defecte on es farà la instal·lació
InstallDir $PROGRAMFILES\AMida

RequestExecutionLevel admin

; -----
; Pages

Page components
Page directory
Page instfiles

UninstPage uninstConfirm
UninstPage instfiles

```

Figura 58: Codi instal·lador (inici).

En aquesta part de 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 programa AMida (obligatori)"

SectionIn RO

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

; Fitxers que es descomprimiran amb l'instal·lació
File /r *.*

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

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

; Registrem la extensió
${registerExtension} "$INSTDIR\amida.exe" ".mida" "Plantilla AMida"

SectionEnd

```

Figura 59 : 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 el arxiu amida.exe.

També crida a la funció `registerExtension` que és la que s'encarrega de indicar-li a l'ordinador que els arxius amb extensió `.mida` s'han d'obrir amb el programa `amida.exe`.

```

; -----
; Aquesta secció es per fer accessos directes. Opcional per l'usuari
; -----

Section "Instal·lació accés directe"

  CreateDirectory "$SMPROGRAMS\AMida"
  CreateShortCut "$SMPROGRAMS\AMida\Uninstall.lnk"
"$INSTDIR\uninstall.exe" "" "$INSTDIR\uninstall.exe" 0
  CreateShortCut "$SMPROGRAMS\AMida\Amida.lnk" "$INSTDIR\amida.exe" ""
"$INSTDIR\amida.exe" 0
  CreateShortCut "$DESKTOP\Amida.lnk" "$INSTDIR\amida.exe" ""
"$INSTDIR\amida.exe" 0

SectionEnd

```

Figura 60: 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. Es una part opcional, on l'usuari 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\AMida"
DeleteRegKey HKLM SOFTWARE\AMida

; Remove files and uninstaller
Delete $INSTDIR\install.nsi
Delete $INSTDIR\uninstall.exe

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

; Remove directories used
RMDir "$SMPROGRAMS\AMida"
RMDir "$INSTDIR\AMida"

${unregisterExtension} ".mida" "Plantilla AMida"

SectionEnd

```

Figura 61: 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 al mètode unregisterExtension per a desenregistrar l'extensió .mida de l'aplicació *A Mida*.

En un altre script anomenat registerExtension.nsh (Figura 62 i 63) és on hi han les dues funcions per registrar la extensió, cridades des de l'script principal, aquestes funcions són les següents:

```

!macro registerExtension executable extension description
    Push "${executable}" ; "full path to my.exe"
    Push "${extension}" ; ".mkv"
    Push "${description}" ; "MKV File"
    Call registerExtension
!macroend

; back up old value of .opt
Function registerExtension
!define Index "Line${__LINE__}"
    pop $R0 ; ext name
    pop $R1
    pop $R2
    push $1
    push $0
    ReadRegStr $1 HKCR $R1 ""
    StrCmp $1 "" "${Index}-NoBackup"
    StrCmp $1 "OptionsFile" "${Index}-NoBackup"
    WriteRegStr HKCR $R1 "backup_val" $1
"${Index}-NoBackup:"
    WriteRegStr HKCR $R1 "" $R0
    ReadRegStr $0 HKCR $R0 ""
    StrCmp $0 "" 0 "${Index}-Skip"
        WriteRegStr HKCR $R0 "" $R0
        WriteRegStr HKCR "$R0\shell" "" "open"
        WriteRegStr HKCR "$R0\DefaultIcon" "" "$R2,0"
"${Index}-Skip:"
    WriteRegStr HKCR "$R0\shell\open\command" "" ""$R2" "%1"
    WriteRegStr HKCR "$R0\shell\edit" "" "Edit $R0"
    WriteRegStr HKCR "$R0\shell\edit\command" "" ""$R2" "%1"
    pop $0
    pop $1
!undef Index
FunctionEnd

```

Figura 62: Codi registerExtension.

On es passen com a paràmetre el executable, la extensió i la descripció de l'arxiu, aquest codi a estat trobat a la web de NSIS.

I el següent fragment de script és el de la funció unregisterExtension, que desfà els passos fets a la funció anterior:

```

!macro unregisterExtension extension description
    Push "${extension}" ;
    Push "${description}" ;
    Call un.unregisterExtension
!macroend

Function un.unregisterExtension
    pop $R1 ; description
    pop $R0 ; extension
!define Index "Line${__LINE__}"
    ReadRegStr $1 HKCR $R0 ""
    StrCmp $1 $R1 0 "${Index}-NoOwn" ; only do this if we own it
    ReadRegStr $1 HKCR $R0 "backup_val"
    StrCmp $1 "" 0 "${Index}-Restore" ; if backup="" then delete the whole key
    DeleteRegKey HKCR $R0
    Goto "${Index}-NoOwn"
"${Index}-Restore:"
    WriteRegStr HKCR $R0 "" $1
    DeleteRegValue HKCR $R0 "backup_val"
    DeleteRegKey HKCR $R1 ;Delete key with association name settings
"${Index}-NoOwn:"
!undef Index
FunctionEnd

```

Figura 63: Codi unregisterExtension.

8. Pressupost

El costos a l'hora de realitzar l'aplicació són els que es mostren en les següents taules:

Recursos Hardware	Euros
Equip informàtic	1.200 €
Total	1.200 €

Taula 3: Recursos Hardware.

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

Taula 4: Recursos Software.

Recursos Humans	Euros
Anàlisis d'aplicacions existents: 30 h x 15 €	450 €
Programació: 150 h x 15 €	2.250 €
Proves i correccions: 60 h x 15 €	900 €
Documentació: 50 h x 15 €	750 €
Redacció Memòria: 90 h x 15 €	1.350 €
Total	5.700 €

Taula 5: Recursos Humans.

Activitat	Euros
Recursos Hardware	1.200 €
Recursos Software	0€
Recursos Humans	5.700 €
Total	6.900 €

Taula 6: Pressupost.

9. Possibles millores

Una de les possibles millores per a aquesta aplicació, és que permetés afegir més tipus d'arxius, per exemple una idea enfocada per a nens, ja no amb dificultats, si no de poca edat, seria que permetés afegir programes, com per exemple jocs, per tant el nen obrint la plantilla *A Mida* podria triar a quin joc vol jugar i no caldria que el busques a l'ordinador, cosa que probablement no sàpiga fer.

Un altra idea de possible millora, seria que es poguessin també afegir arxius .mida dins del mateix programa, per tant ja no es limitarien només a 6 opcions per a introduir, si no que aquestes es multiplicarien ja que podria haver-hi a dins de cada quadrat una altre plantilla *A Mida*, podent fer així per exemple un menú, és a dir, el primer quadrat serien webs, i al obrir-lo sortiria una altre presentació on els 6 quadrats contindrien 6 webs, el següent quadrat serien pel·lícules, i al prémer s'obriria una altre presentació que contindria 6 pel·lícules, i així successivament.

També per a millorar el programa, seria interessant que tot és pogués afegir on-line, tant els arxius com les icones, per tant ja no hi hauria el problema de haver de treballar sempre en local, ja que la ruta de l'arxiu seria una direcció d'internet i per tant s'hi podria accedir des de qualsevol ordinador que tingués el software *A Mida* instal·lat.

Això seria interessant, per exemple, per si els responsables de la Fundació Maresme, volguessin penjar a la seva web, des d'on es descarregarà el software, plantilles ja fetes per ells i que els usuaris les poguessin també descarregar i reproduir sense problemes.

10. Conclusions

El projecte té com objectius crear un programa que permetés crear presentacions que l'usuari pogués configurar i afegir els arxius que es desitgi.

Els objectius que tenia aquest projecte han estat assolits, l'aplicació permet crear presentacions personalitzables amb 6 botons on a cada botó pots afegir l'arxiu que vulguis cercant-lo a l'ordinador.

L'altre objectiu fonamental és l'ús de l'encerclament, objectiu que també ha estat assolit, ja que l'usuari pot fer funcionar l'aplicació sense la necessitat de moure el ratolí. S'ha afegit funcionalitat a l'aplicació permetent triar si es vol utilitzar encerclament o no, permetent així també utilitzar el ratolí de manera convencional.

L'aplicació permet guardar les plantilles creades i recuperar-les posteriorment.

Les opcions principals d'arxius que havia de tenir el programa eren vídeo, so, imatge i web, opcions que permet reproduir l'aplicació, i a més s'ha afegit la opció de imatge amb so.

En conclusió es creu que s'han assolit tots els objectius que demanava l'aplicació amb èxit.

ANNEX I

1. Manual d'usuari

1.1. Instal·lació A Mida

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

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

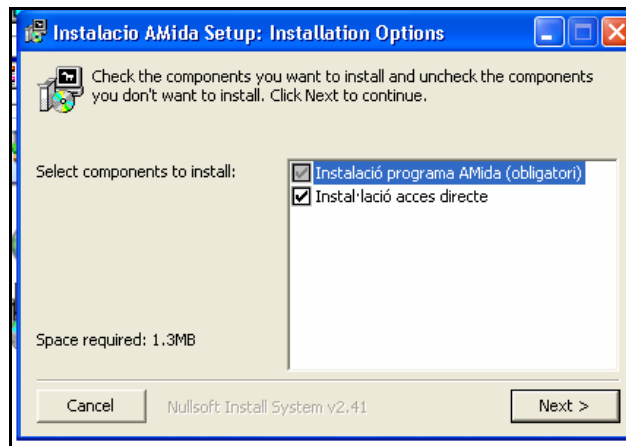


Figura 1: Pantalla d'instal·lació.

1.2. Crear un presentació A Mida

Obrir el programa A Mida i es mostrarà la pantalla d'edició. Per a afegir arxius clicar sobre el botó afegir arxiu i s'obrirà el menú que ens permet triar quin tipus d'arxiu volem afegir.



Figura 2: Boto per afegir



Figura 3: Menú de tipus d'arxiu

Un cop triat el tipus d'arxiu s'obrirà la pantalla per afegir la imatge si hem triat imatge, o la pantalla per afegir la icona i l'arxiu si hem triat la resta d'opcions.



Figura 4: Carregar.

Prémer el botó per carregar la icona, on ens deixarà escollir una imatge, i posteriorment prémer el botó de carregar l'arxiu, on triarem l'arxiu que es vulgui afegir (vídeo, so o web). Si hem escollit una web, caldrà prémer el botó guardar web per a que quedi guardada.



Figura 5: Carregar web.

Comprovar que s'han afegit correctament els arxius mirant la pantalla i veient que ha cada quadrat es mostra la imatge escollida com a icona, i el nom del arxiu afegit, si no és així, tornar a afegir.



Figura 6. Icona i arxiu afegit.

Un cop s'hagin afegit tots els arxius que es vulguin, a la pantalla apareix l'opció per a utilitzar l'encerclament o no. L'encerclament és el que permet desplaçar el cursor sense haver de moure el ratolí, és a dir, amb el botó dret del ratolí passem al següent botó, i amb el botó esquerre fem clic sobre el control.

Si l'opció està en vermell vol dir que el encerclament està desactivat, i si està verda és que si que està activat.

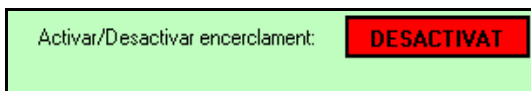


Figura 7: Encerclament desactivat

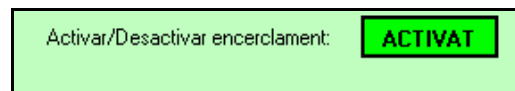


Figura 8: Encerclament activat

Fent clic al botó provar podem fer una vista prèvia de la presentació. I si es desitja, guardar la presentació des de opcions, guardar plantilla. També es pot triar fer una plantilla nova, o carregar alguna ja existent.

1.3. Presentacions A Mida

Per obrir una presentació *A Mida* ja creada, cal obrir directament l'arxiu, o carregar-lo des del programa. Al obrir l'arxiu s'obre la presentació, que si està en mode d'encerclament podem veure utilitzant el botó dret i l'esquerre del ratolí, i si no amb el ratolí de manera convencional.

Per obrir els arxius cal fer clic sobre la icona que es desitgi, i per tancar la finestra fer clic amb el boto dret o el boto esquerre.

Per finalitzar el programa s'ha de prémer Alt + F4.

1.4. Desinstal·lació A Mida

Per desinstal·lar el programa executar l'arxiu uninstall.exe que s'ha creat a la carpeta on s'ha instal·lat l'aplicació.

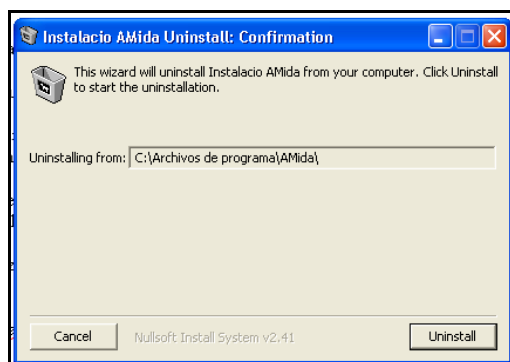


Figura 9: Pantalla de desinstal·lació.

Prémer sobre uninstall i l'aplicació es desinstal·larà.

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 a instal·lar l'aplicació.

Bibliografia

Referències bibliogràfiques:

[1] Francisco Charre Ojeda, *Programación con Visual Basic.NET*, Anaya Multimedia

[2] Brian Siler & Jeff Spotts, *Visual Basic 6 Edición Especial*, Prentice Hall

Referències web:

[3] <http://softwarelibre.uca.es/wikijuegos/Portada>

[4] http://es.wikipedia.org/wiki/Simple_DirectMedia_Layer

[5] <http://www.programatium.com/director.htm>

[6] <http://software.computadora-discapacidad.org/>

[7] http://www.antoniosacco.com.ar/docu/pautas_desarrollo_soft.pdf

[8] <http://authorware-iufront.blogspot.com/2008/07/authorware.html>

[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.lamosqueta.cat>

[15] <http://www.tecnologiaydiscapacidad.es>

