

Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons. Podeu copiar-lo, distribuir-lo i transmetre'ls públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada. La llicència completa es pot consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.



# Cultiu Podcast

Memòria

Carlos Bueno Martínez  
Enginyeria Tècnica en Informàtica de Gestió  
Treball Final de Carrera (TFC)

## **Cultiu**

Tècnica de laboratori que té per finalitat obtenir i manejar quantitats elevades de microorganismes per poder-los estudiar o extreure'n substàncies que interessin, com antibiòtics, proteïnes, polisacàrids, etc

## **Podcast**

“A series of audio or video digital-media files which is distributed over the Internet by syndicated download, through Web feeds, to portable media players and personal computers. Though the same content may also be made available by direct download or streaming, a podcast is distinguished from other digital-media formats by its ability to be syndicated, subscribed to, and downloaded automatically when new content is added...”<sup>1</sup>

---

<sup>1</sup> <http://en.wikipedia.org/wiki/Podcasting>

## Taula de continguts

<b>Presentació</b>	<b>5</b>
Memòria	5
Motivació personal	5
Coneixements previs de l'entorn	5
Compromisos	6
Hardware	6
Programari	6
Repositori de codi	7
<b>Objectius</b>	<b>8</b>
Abast de Cultiu Podcast	8
<b>Planificació</b>	<b>9</b>
Fases del projecte	9
Fase 1	9
Fase 2	10
Fase 3	10
Fase 4	11
Fase 5	11
Fase 6	11
Fase 7	12
Gantt del projecte	12
<b>Productes obtinguts</b>	<b>13</b>
Presentació del codi font	13
<b>Anàlisi funcional</b>	<b>15</b>
Flux d'aplicacions seqüencial	15
Flux d'ingesta i captura	17
Arquitectura de CPM	18
Flux de procés d'upload	19

Flux de procés de publicació: L-Commit	20
Flux de procés de publicació: G-Commit	21
Flux d'accés Web	22
Flux d'aplicacions: visió global	23
Flux del sistema escalat	24
<b>Anàlisi orgànica: els mòduls</b>	<b>25</b>
Ingesta i transcodificació, Device Manager ( DM )	25
Diagrama de classes	27
Classe Graph	29
Classe Device	33
Classe PlyAudio	34
Mètodes exportables de la DLL	34
Catalogació i client FTP, Cultiu Podcast Manager ( CPM )	36
Metadata Dinàmica	40
Classe MetadataTag	40
Publicació, Cultiu Podcast File Watch Dog (CPFWD)	44
Classe ObdcDataBase	47
Accés WWW, Cultiu Podcast Center (CPC)	50
Motor de recerca	52
BBDD	57
<b>Conclusions</b>	<b>66</b>
<b>Línies de futur</b>	<b>66</b>
<b>Glossari</b>	<b>67</b>
<b>Bibliografia</b>	<b>69</b>

## Presentació

### ***Memòria***

Aquest document **Memòria** és un recull de dades i informació sobre el Projecte de Final de Carrera (TFC) per a la Universitat Oberta de Catalunya ( UOC ).

Aquest projecte s'emmarca dins l'àrea Ad Hoc de la UOC, que permet presentar qualsevol tipologia de projecte relacionat amb l'Enginyeria Tècnica en Informàtica de Gestió (ITIG) sempre que es consideri interessant i amb un valor afegit d'interès tècnic.

Al final d'aquest document s'haurà deixat clar el treball desenvolupat pel projecte TFC, així com a documentació tècnica per a la seva comprensió i possible ampliació o continuació del mateix.

### ***Motivació personal***

Sóc usuari habitual de podcasts i crec que és un sistema molt útil per accedir a continguts dins la xarxa. Per aquest motiu volia que el TFC cobrés aquesta temàtica.

Per altra banda, donada la meua trajectòria professional, he tingut la sort de treballar en una ampli ventall de llenguatges, sistemes i SGBDs. És per això que volia combinar-ho tot.

El projecte és una barreja de múltiples plataformes i múltiples llenguatges, des de baix nivell com podria ser el tractament de dispositius de captura amb C / C++ i més alt nivell com a C#, Asp-NET.

Respecte a les BBDD he mirat també de fer un ús intensiu amb disseny ER, i tasques de DBA, amb SQL-Server2k8.

### ***Coneixements previs de l'entorn***

Experiència en arquitectura i desenvolupament de software. Professionalment des de fa 8 anys. Gran part d'aquests anys he estat professor en un centre privat, fet que em permet conèixer diverses tecnologies com les que es proposen en el TFC.

He treballat com a Cap de Projectes amb temes de TV / Narrowcasting. Lamentablement no es dóna gaire èmfasi a l'audio i crec que aquesta és una bona oportunitat per a tractar-lo.

Actualment estic com a Consultor a Avanade desenvolupant tasques amb perfil tècnic amb SQL-Server, DTS, C# i Asp.Net.

## **Compromisos**

S'han adquirit, durant el desenvolupament del projecte, els compromisos següents:

Si per causes relacionades amb la salut es produeixen endarreriments en la realització del projecte com a conseqüència d'un rendiment inferior al normal, es dedicaran més hores al TFC un cop superada la malaltia i, si és necessari, es modificarà el pla de treball per contemplar aquestes alteracions i reestructurar el temps de dedicació a cada fase del projecte.

En cas de produir-se algun esdeveniment que faci variar la disponibilitat horària prevista, es farà una reestructuració per tal de recuperar el temps perdut.

Durant el desenvolupament del projecte no han aparegut distorsions greus que hagin influït sobre la pla de treball presentat inicialment. En tot cas s'ha dedicat més temps del previst per un tema de satisfacció personal.

## **Hardware**

En aquest projecte s'utilitzarà l'ordinador personal: Equip de portàtil Dell XPS: Dual Core 2.0GHz, 4 Gb de memòria RAM amb SO Windows Vista.

En cas d'algun tipus de problema, es farà servir el portàtil personal de la feina: Equip de portàtil Dell Latitude 6300: Dual Core a 2.1GHz, 4 Gb de memòria RAM amb SO Windows Vista.

## **Programari**

És necessari un Sistema Operatiu Windows per al desenvolupament del programari i una targeta de hardware que permeti l'entrada d'Audio. L'entrada de microfon és suficient, tot i que la qualitat no és la millor (el ordinadors en tenen habitualment).

També és important disposar d'un dispositiu i d'un cablejat adient per alimentar l'entrada d'audio de l'ordinador que realitza la captura. En el meu cas, he fet servir el meu iPod i he comprat el cable per connectar-los.

Com a eines de desenvolupament faré servir les detallades a continuació, totes elles gratuïtes, o disponibles en versió estudiant.

[Visual Studio Express](#)

[SQL Server 2k8](#)

[ISS](#)

[Filtro/s de Encoding Vorbis](#)

[Subversion](#)

### ***Repositori de codi***

És fa servir com a repositori de codi l'eina Subversion ([SVN](#)), que ens permet un control total de versions en el desenvolupament. Aquest repositori està instal·lat a un disc dur extern i es fan còpies de seguretat cada cap de setmana en un Pen-Drive exclusiu per tal efecte.



## Objectius

### ***Abast de Cultiu Podcast***

L'objectiu del TFC és crear una 'suite' que resolgui tota la línia de producció d'un podcast. És a dir: **captura** d'un senyal d'audio en directe, **transcodificació**, **classificació**, **emmagatzematge** i, per acabar, **difusió** per Internet.

El projecte pren el nom de Cultiu Podcast (CP), i és tota una 'suite' d'eines que tenen com a objectius principals:

- Desenvolupar un producte de qualitat i de dificultat màxima, preparat per a la seva implantació en producció.
- Deslocalitzar geogràficament totes les parts del software desenvolupat. La plataforma CP permet tenir servidors distribuïts regionalment per oferir un millor rendiment. L'upload a les dades és deslocalitzat, per [FTP](#), i l'accés a les dades també ho és per Internet amb WWW.
- Dissenyar un conjunt d'aplicacions escalables i modulars. Oferir una adaptació a la quantitat d'usuaris i a les seves necessitats, tant a nivell de servidors com de software.
- Integrar cadascuna de les seves parts amb productes de tercers fabricants de software.
- Presentar documentació tècnica completa per al posterior seguiment o desenvolupament.

## **Planificació**

### **Fases del projecte**

Donat l'abast del projecte, l'organització és clau per al desenvolupament del mateix. El projecte s'ha dividit en fases lògiques de mòduls.

Fase 1: Anàlisi de requeriments i disseny de l'aplicació

Fase 2: Diagrames E/R i muntatge de BBDD i servidor FTP

Fase 3: Mòdul DM ( captura i transcodificació )

Fase 4: Mòdul CPM ( gestió de captura i transcodificació, classificació )

Fase 5: Mòdul CPFWD ( emmagatzematge )

Fase 6: Mòdul CPC ( difusió )

Fase 7: Documentació i memòria del projecte

### **Fase 1**

Es procedeix a la generació del pla de treball i, paral·lelament, a una anàlisi de requeriments i dissenys de les aplicacions de CP.

En aquesta primera fase és molt importat establir tota l'arquitectura de comunicació i flux de dades que hi ha entre les diverses aplicacions.

Aquest mateix procés, s'itera a cadascun dels diferents mòduls, fent una anàlisi més acurada que detalla amb precisió tot el procés.

D'aquesta forma es desenvolupa una anàlisi funcional i es genera la documentació amb l'arquitectura que s'implementarà. Es destaquen les aventatges i els inconvenients.

En aquest punt també es decidixen els llenguatges pertinents així com les plataformes de desenvolupament.

Es preveuen re-avaluacions al final de cada Fase, com si es tractés d'un procés en espiral.

Dates previstes: del 4 al 21 de setembre.

## **Fase 2**

Es procedeix a la definició d'estructures de dades que emmagatzemen la informació persistent de l'aplicació. Això inclou creació de diagrames E/R i diccionari de dades.

Es generen scripts per a la creació de BBDD, taules, primary keys, foreign keys, constraints, etc.

S'itera la Fase 1 molt ràpidament, revisant tot el sistema en conjunt, evaluant noves idees/problemes amb l'anàlisi de la fase 2, modificant el que es considera necessari per a crear un sistema robust.

Dates previstes: del 22 al 28 de setembre

## **Fase 3**

Es procedeix amb la part més difícil i més tècnica de tot el projecte: la creació del DM per a la captura de dispositius. Les seves fases de desenvolupament són:

- Anàlisi de requisits
- Disseny de requisits
- Disseny del Programa
- Codificació
- Proves
- Integració amb .NET
- Documentació de l'API

En aquest punt es retorna a la Fase 1, per a re-avauar.

Dates previstes: del 29 de setembre al 27 d'octubre

## **Fase 4**

Es procedeix amb la part vistosa de tot el projecte: la creació del CPM. El seu cicle en espiral permet la seva redefinició i constant evolució. Tot i això, es marca una data límit per aconseguir els elements funcionals mínims. El seu procés és:

- Determinació d'objectius
- Anàlisi de riscos
- Planificació
- Codificació i test
- Retorn al primer punt

Es retorna a la Fase 1 per a re-avalar.

Dates previstes: del 28 d'octubre al 25 de novembre

## **Fase 5**

Es desenvolupa un projecte bastant simple però amb complexitat en el seu testeig, donades totes les possibles casuístiques de l'aplicació.

S'escull un cicle tradicional en cascada, idèntic al de la Fase 3.

Retorn a la Fase 1, per a re-avalar.

Dates previstes: del 26 de novembre al 6 de desembre.

## **Fase 6**

Desenvolupament de la part Web de l'aplicació, amb un procés idèntic al de la fase 4. Aquesta fase té com a peculiaritat l'accés a BBDD.

Aquesta fase es pot ampliar, tal i com es pot veure al 'gant', en funció del temps disponible.

Dates previstes: del 7 de desembre al 22 de desembre

## Fase 7

En aquesta fase es procedeix a la documentació de tota la plataforma. En cas d'anar molt bé de temps es pot iterar la fase 5 o 6.

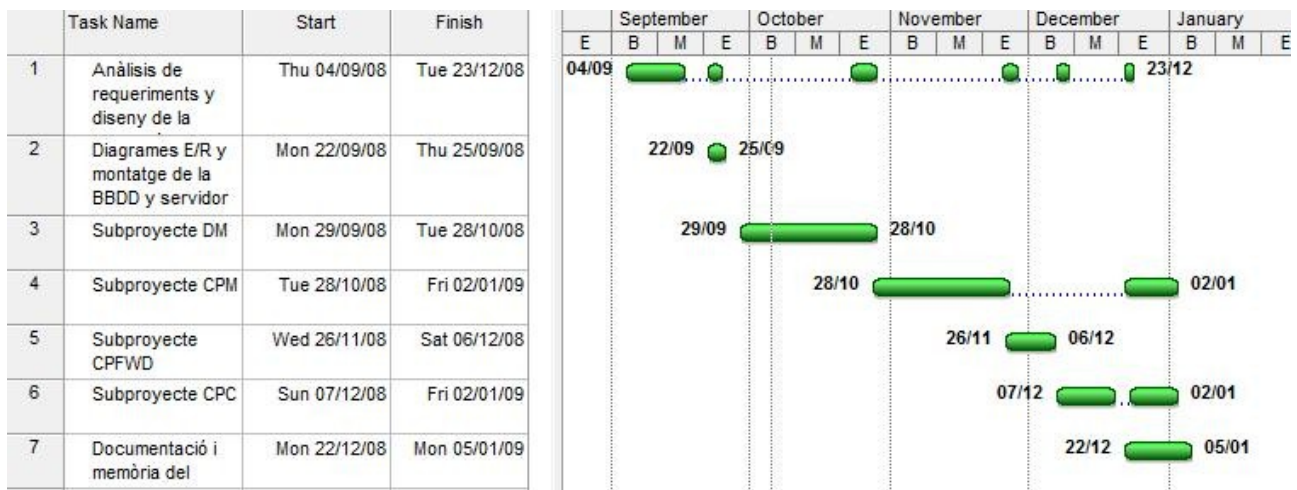
Dates previstes: del 23 de desembre al 6 de gener.

## Gantt de projecte

El gantt del projecte ofereix una visió global i gràfica de les dates definides, així com de l'interactuació de tasques.

Cal observar les contínues re-avaluacions de la fase 1 i la possibilitat d'iterar els mòduls CPM i CPC de cicle de vida en espiral.

Aquesta iteració i re-avaluació fa que al gantt de projecte apareguin tasques solapades. La interacció entre els diferents mòduls és constant.



## Productes obtinguts

El desenvolupament d'aquest treball de final de carrera ha generat els següents productes:

- Memòria on es descriu el projecte, les fases del seu desenvolupament i el resultat del mateix.
- Presentació en PowerPoint on es destaquen els aspectes principals del projecte.
- Codi font de les aplicacions:
  - Device Manager (DM)
  - Cultiu Podcast Manager (CPM)
  - Cultiu Podcast File Watch Dog (CPFWD)
  - Cultiu Podcast Center (CPC)
- Diagrames de base de dades ER.
- Script de creació de la base de dades pel projecte.

### ***Presentació del codi font***

L'entrega del codi font es realitza ordenat en carpetes representatives dels diversos mòduls. Alfabèticament:

- CPC (Cultiu Podcast Center)  
Projecte ASP.NET en C#
- CPFWD (Cultiu Podcast File Watch Dog)  
Projecte Winforms en C#
- Cultiu Podcast Manager (CPM)  
Projecte Winforms en C#
- Device Manager (DM)
  - Projecte Windows en C/C++
  - DM Demo (aplicació de test per DM)  
Projecte Winforms en C#

- Doc
  - CarlosBuenoMEMÒRIA.pdf
  - Presentació\_TFC.ppt
- MSSQL 2k8
  - Scripts CPC\_BBDD.sql
  - Scripts CPC.sql
- Resources
  - cp.ico
  - CultiuPodcast.gif
  - Logo.jpg



## **Anàlisi funcional**

En l'anàlisi funcional trobem Diagrames de Flux de Dades ([DFD](#)) fins a un nivell de profunditat 1 per mostrar el funcionament i processament de dades del sistema Cultiu Podcast.

Es comença des del Flux d'aplicacions i després es detalla cada element individualment, sense entrar en aspectes tècnics d'implementació.

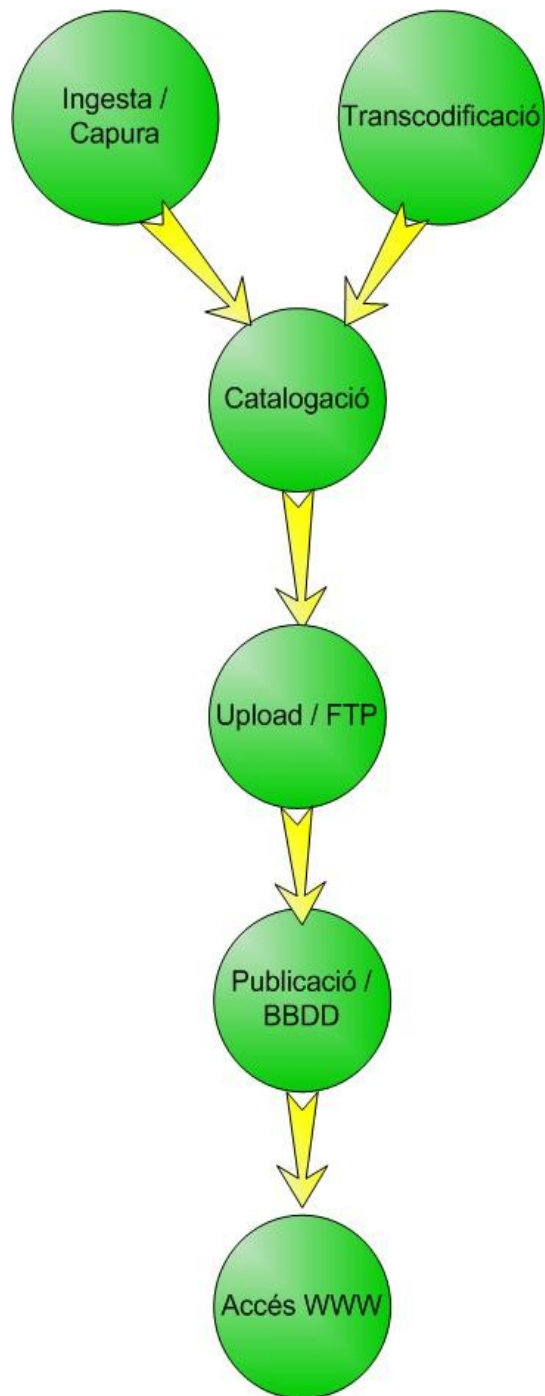
Aquesta part de la documentació ha de proporcionar una visió global de tot el procés per a, posteriorment, poder comprendre l'anàlisi orgànica que tracta aspectes més tècnics.

### ***Flux d'aplicacions seqüencial***

Aquest és el diagrama més genèric, vist des del punt més alt on es pot veure tot el conjunt d'elements que formen el sistema: Ingesta i/o Captura, Catalogació, Emmagatzematge i Publicació/Difusió.

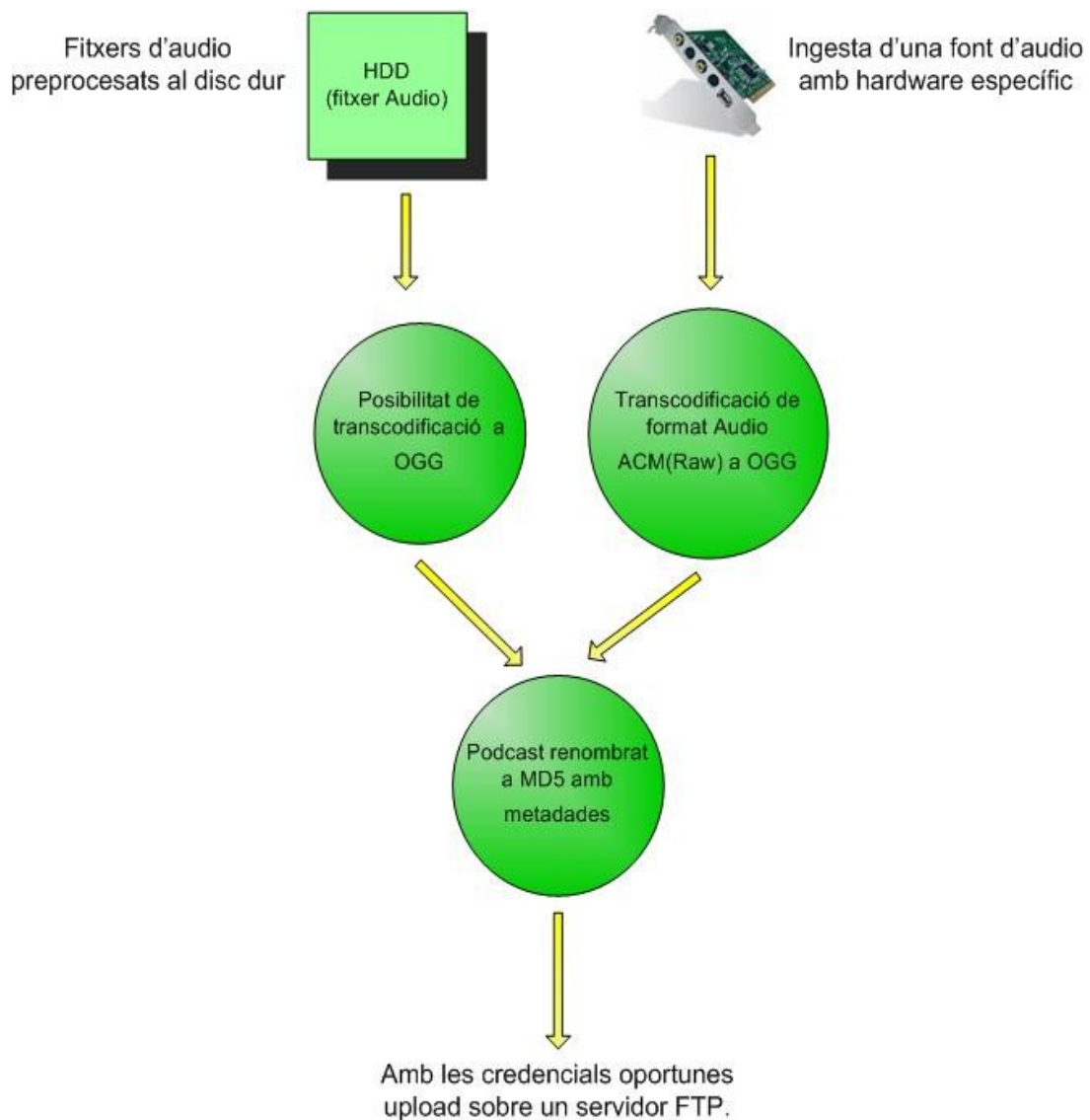
El flux de treball queda molt clar, establint totes les etapes que ha de recórrer un Podcast.





## Flux d'ingesta / Captura

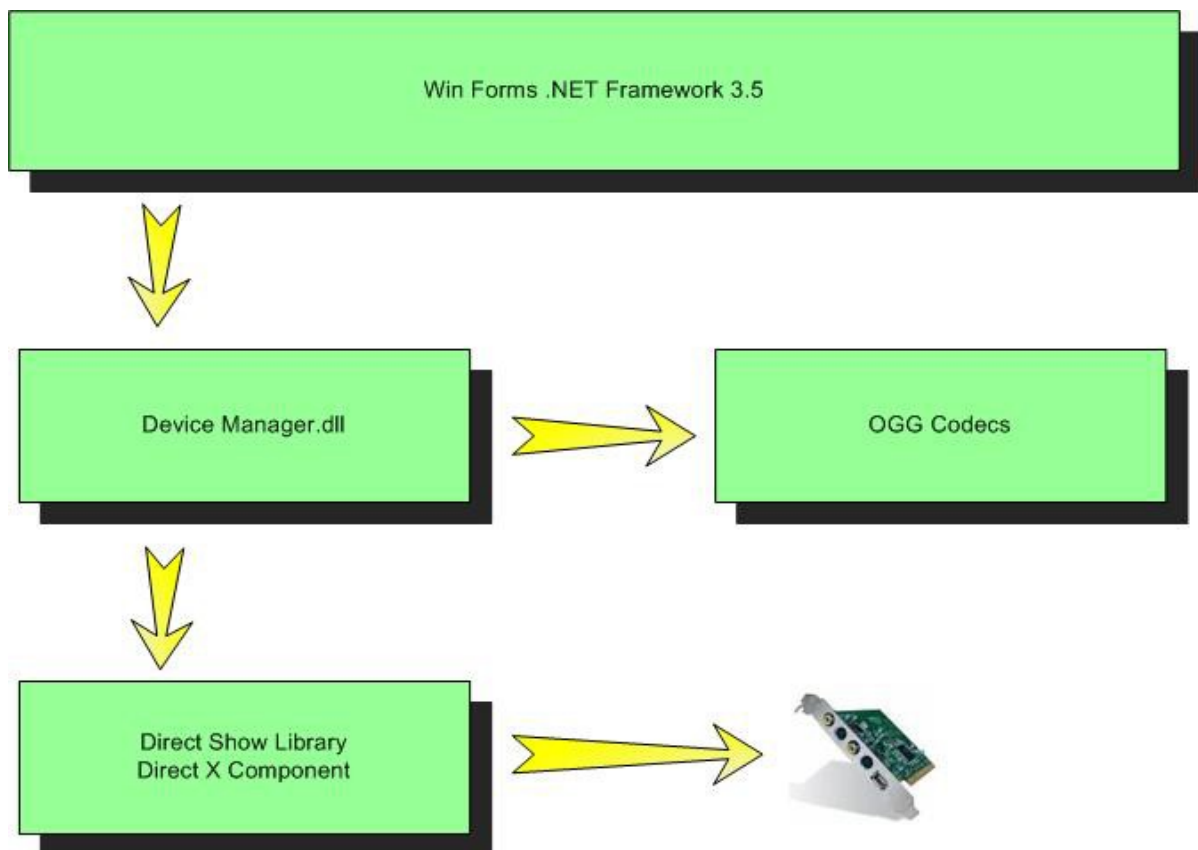
Nivell 1 del procés **Ingesta / Captura** i **Transcodificació** definit al fux d'aplicacions. El sistema ha de garantir que no existeixen col·lisions amb els noms de fitxers Podcast. Per aquest motiu els Podcast són renombrats amb el seu MD5. El nom i característiques originals queden catalogades com a metadata.



## Arquitectura de CPM

El mòdul de CPM es basa amb el Framework 3.5 que accedeix a DM, que li ofereix tota l'arquitectura de DirectShow i, per extensió, l'accés als dispositius de hardware d'audio.

Des de DM s'accedeix també als codecs OGG per a fer transcodificacions.



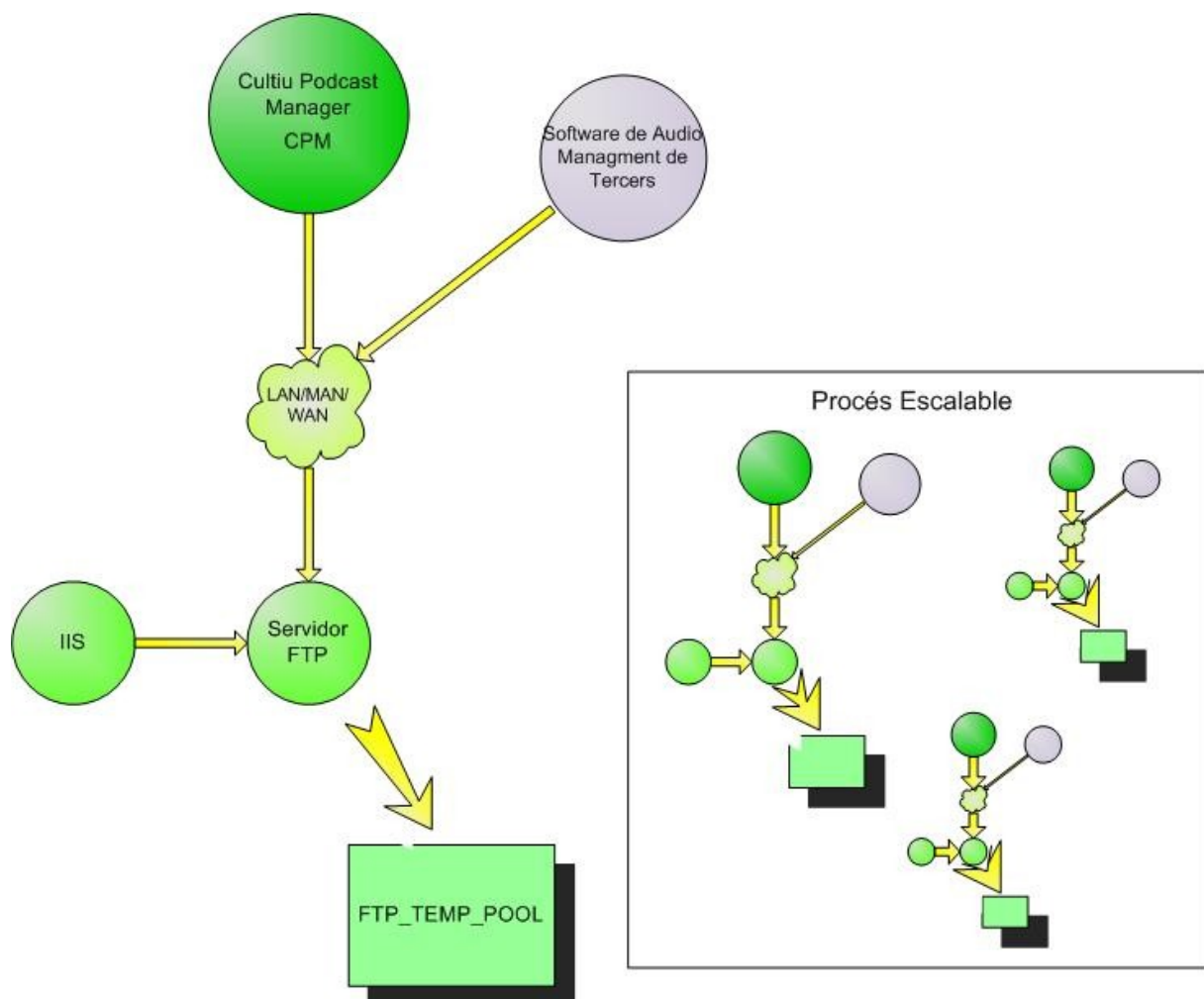
Des del mòdul CPM es procedeix a la seva **catalogació** amb un sistema de Metadata Dinàmic, i amb tecnologia multithread es procedeix a l'**enviament** al servidor FTP.

## Flux de procés d'upload

Amb un servidor IIS ( o similar ) es munta un servidor FTP encarregat de rebre els Podcast per la xarxa. El podcast és enviat per CPM.

Aquests nous Podcast són rebuts a un directori temporal (FTP\_TEMP\_POOL) abans de la seva inclusió al sistema CP.

El procés és altament escalable i modular. Com a funcionalitat extra, permet que un software de tercers, amb les pautes correctes (representat al diagrama amb color lila) pugui també afegir nous podcast al sistema.

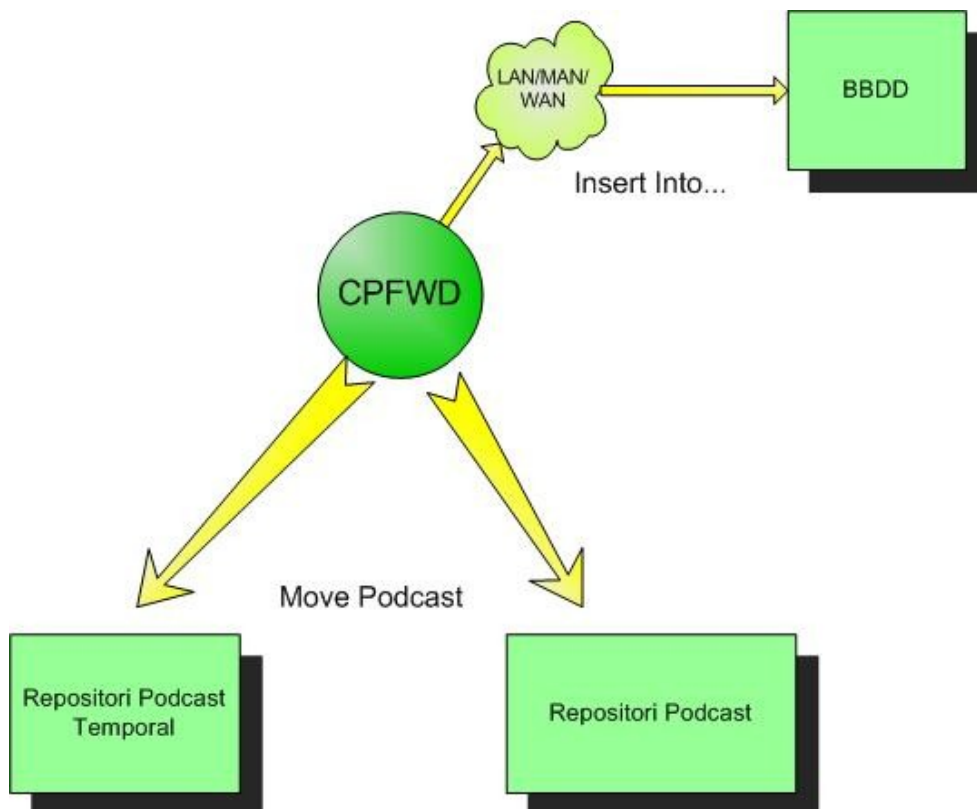


### **Flux de procés de publicació: L-Commit**

Al servidor destinat a rebre els Podcast per FTP, el mòdul CPFWD està monitoritzant l'arribada de nous arxius amb la seva corresponent metadata.

CPFWD és el responsable de, transaccionalment, donar d'alta el nou podcast arribat al servidor de BBDD al qual està configurat, així com de moure el Podcast a la seva ubicació final.

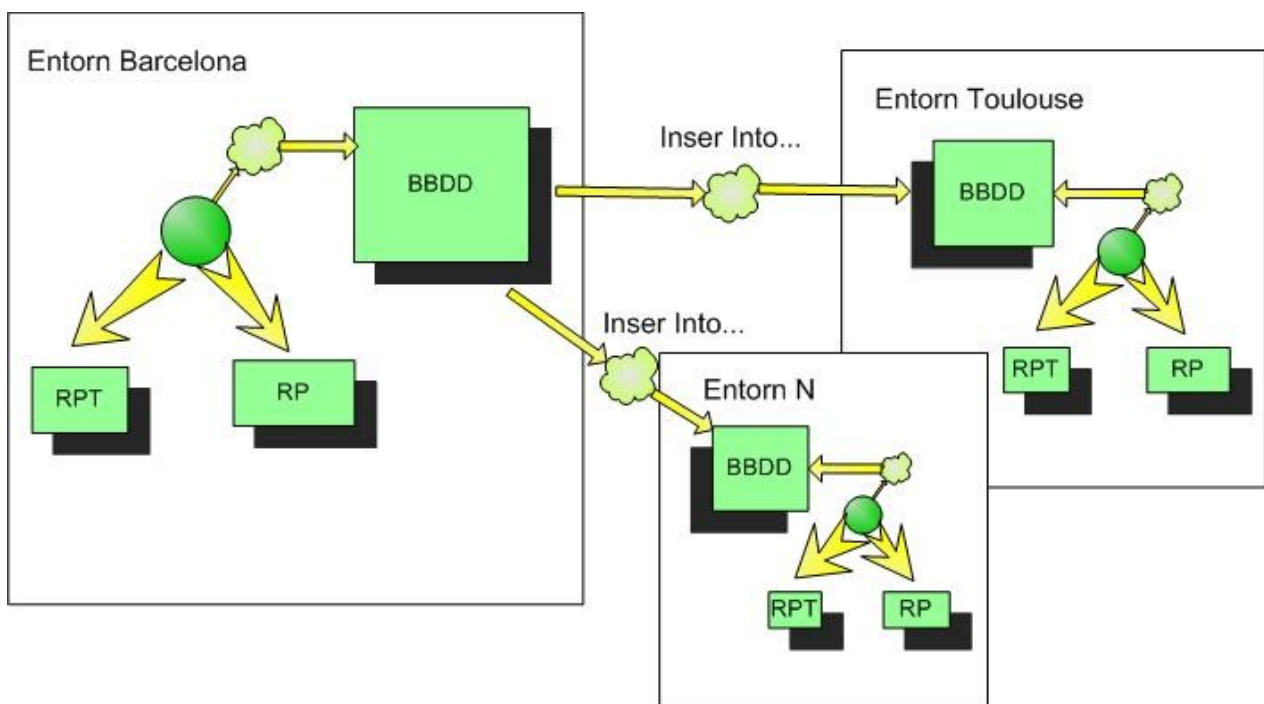
L'ubicació final està ordenada per directoris amb nomenclatura hexadecimal. El primer caràcter del Podcast, prèviament codificat amb un hash MD5, indica on ha de ser guardat.



## Flux de procés de publicació: G-Commit

Es produeix un procés de Commit Global (G-Commit) a la resta de sistemes idèntics instal·lats en altres localitzacions i que vulguin compartir els Podcast.

Aquest procés es realitza amb Jobs, DTS (anomenats IS a partir de la versió 2k5) i rèpliques de SQL Server. El procés de G-Commit es pot dur a terme en un interval de temps customitzat i és de càrrega molt lleugera, ja que és un procés diferencial.



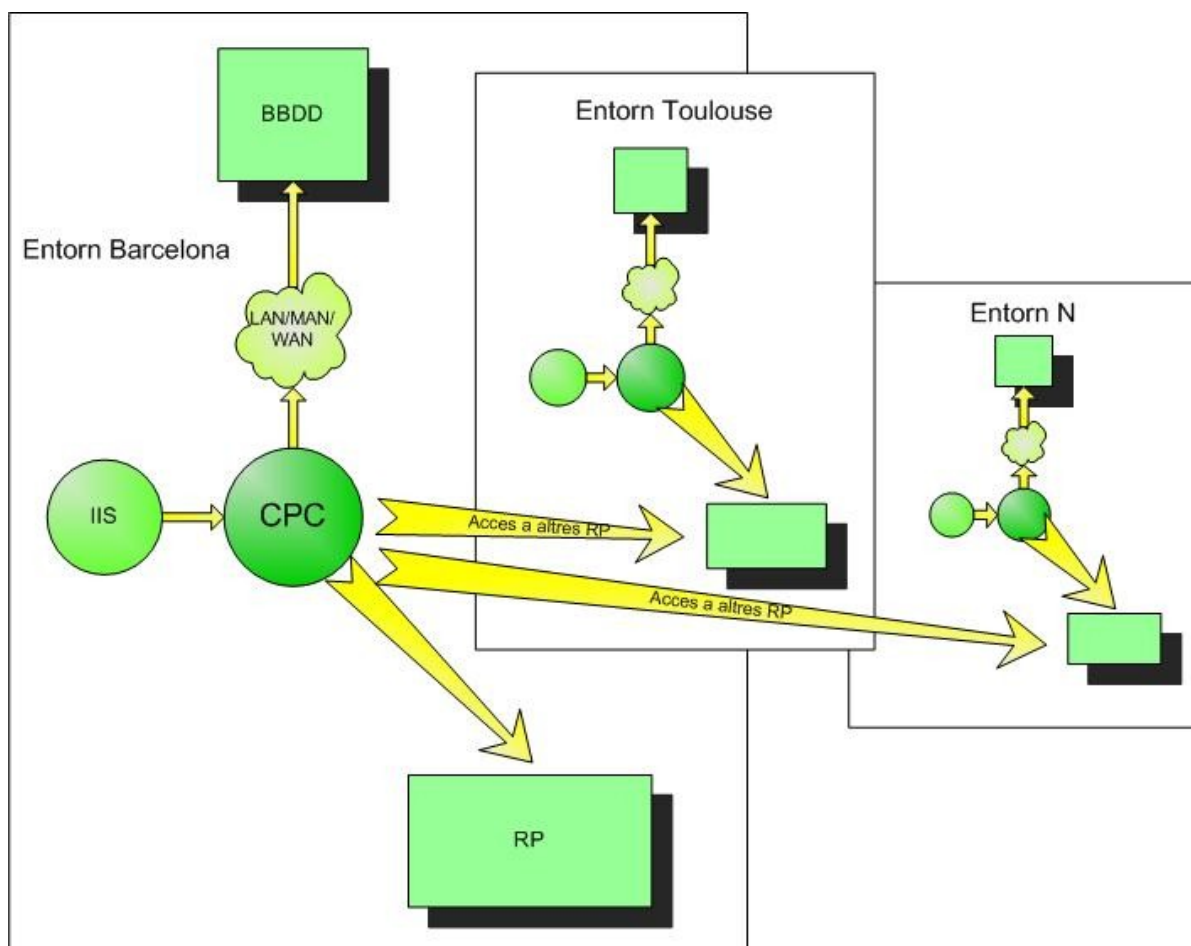
## Flux d'accés Web

L'accés als Podcast està pensat en una arquitectura Web de tres capes:

- Nivell de base de dades, amb un SQL Server 2k8.
- Lògica de negoci amb un IIS amb CPC com a model desenvolupat.
- Lògica de presentació, qualsevol navegador Web.

Aquestes característiques proporcionen una alta escalabilitat i modularitat; a més, permeten deslocalitzar geogràficament la instal·lació de les aplicacions i l'accés de l'usuari.

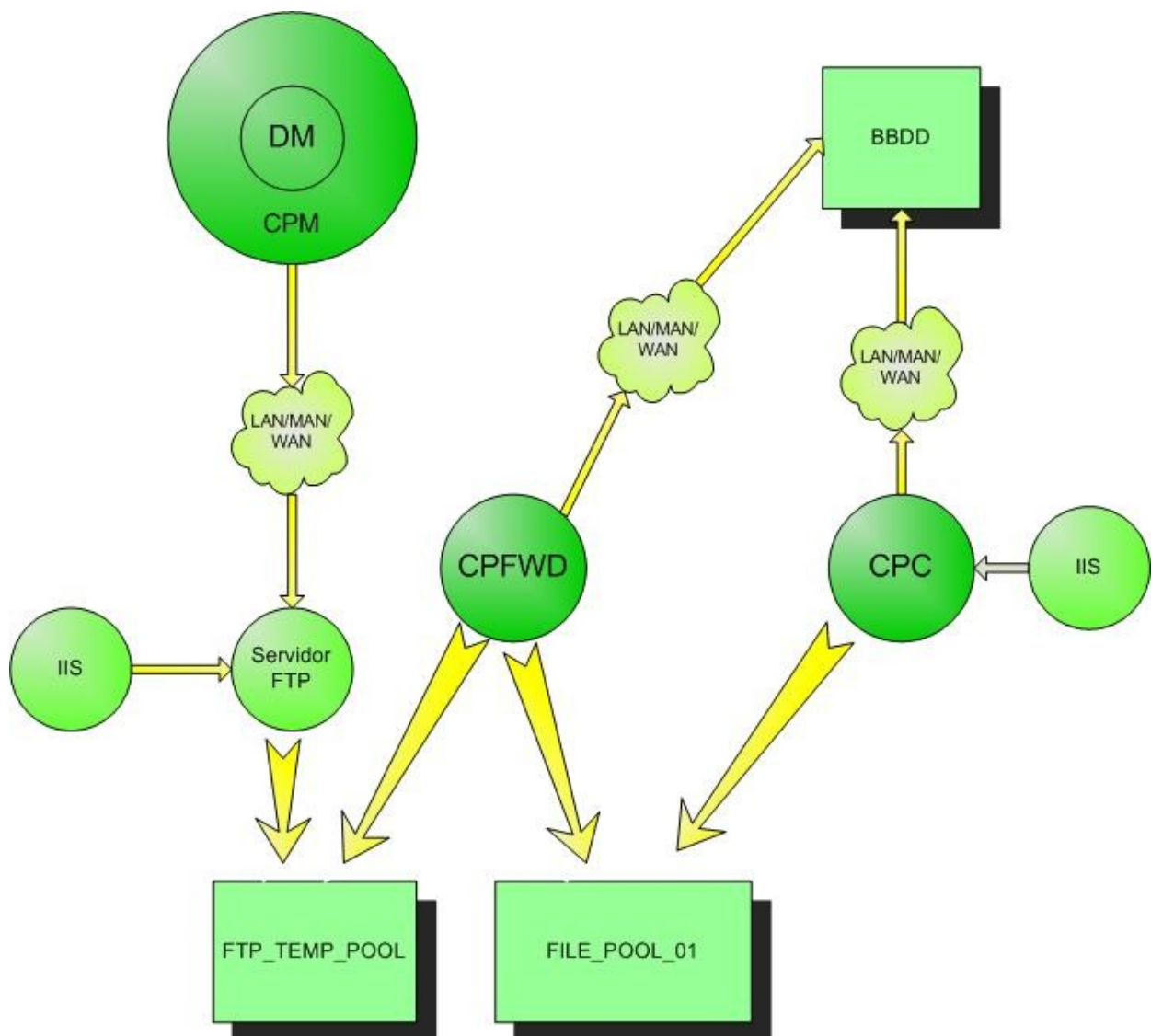
A continuació es veu un diagrama per mostrar com CPC accedeix al seu repositori (RP) o a d'altres, després d'haver accedit a la BBDD on està configurat.



## Flux d'aplicacions: visió global

Es presenta el flux d'aplicacions i processos amb una visió global i completa. És essencial la seva comprensió per tal d'entendre l'abast del projecte; és la seva plasmació en un únic diagrama.

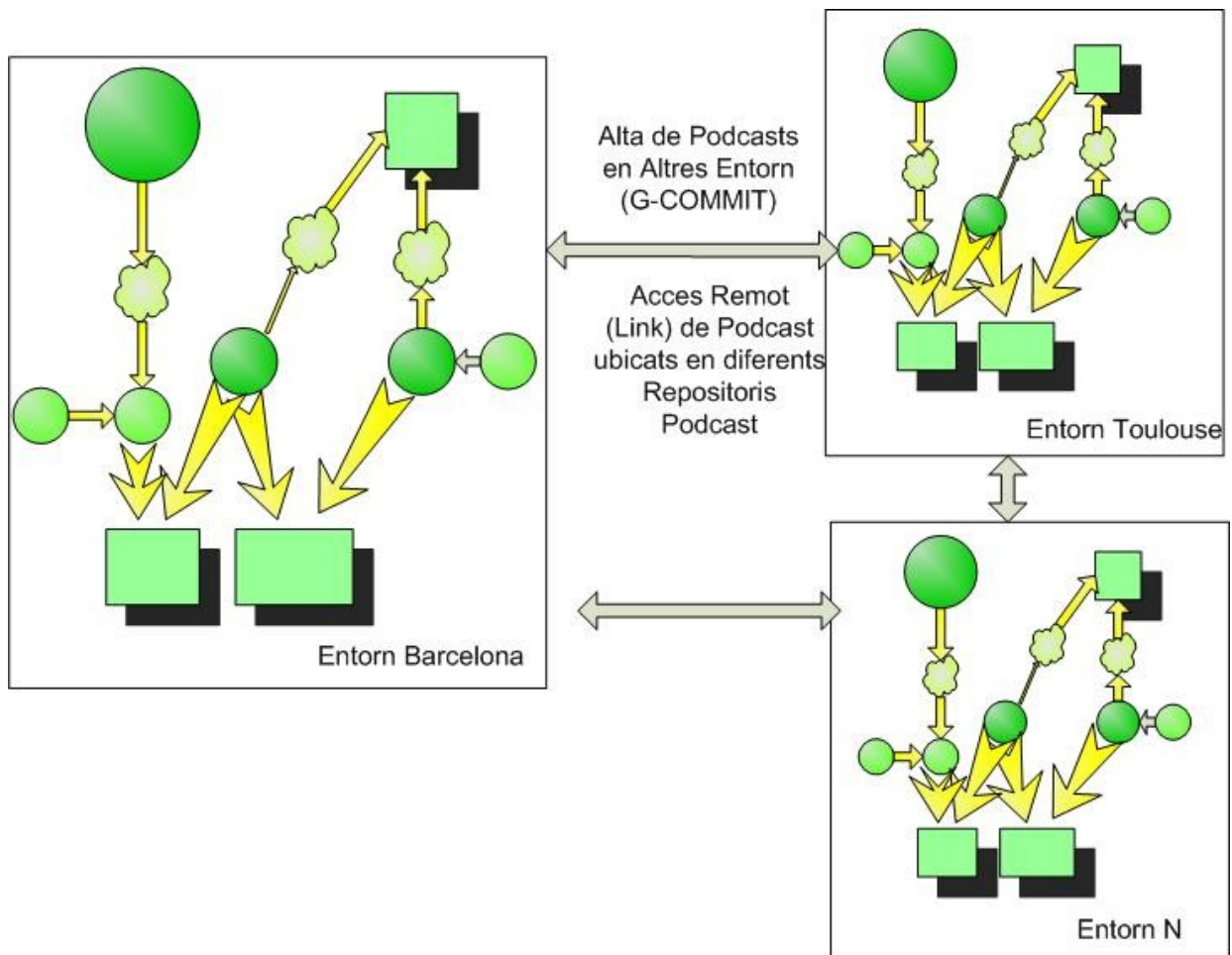
Cal fer notar que FILE\_TEMP\_POOL és l'espai temporal per rebre els Podcast i FILE\_POOL\_01 és la seva ubicació definitiva.





### ***Flux del sistema escalat***

Per últim, una representació del sistema del diagrama anterior escalat en N.



## Anàlisi orgànica: els mòduls

### *Ingesta i transcodificació, Device Manager ( DM )*

**Ingesta / Captura** d'un senyal d'audio amb un dispositiu de hardware com pot ser una targeta de so integrada a la placa base, fent servir Direct Show. L'aplicació suporta qualsevol hardware de captura d'audio.

**Transcodificació** de formats: transcodificació d'un senyal 'x' a un format comprimit com per exemple Ogg, Wma, Mp3, fent servir Direct Show. Inicialment es proveeix de transcodificacions en Ogg de [Vorbis](#).

Una transcodificació és un canvi de codificació o format d'un senyal. En aquest projecte el senyal a transcodificar és d'audio. Per exemple; si es té un dispositiu de hardware per capturar audio en format [PCM](#) (Pulse code modulation), la transformació del format PCM a un format comprimit [Ogg](#) , és una transcodificació.

Possiblement és la part més complexa tècnicament, i per la qual fa falta un nivell tècnic més alt per dur a terme el projecte i per aquest motiu és la part que requereix més temps de dedicació.

Aquesta part es desenvolupa en una [DLL](#) (Dynamic Link Library) amb un llenguatge a baix nivell com és C/C++ amb la API de [Win32](#).

La tecnologia [DirectShow](#) ( inclosa a [DirectX](#) ) és la API que s'utilitza per al control i programació del hardware de captura.

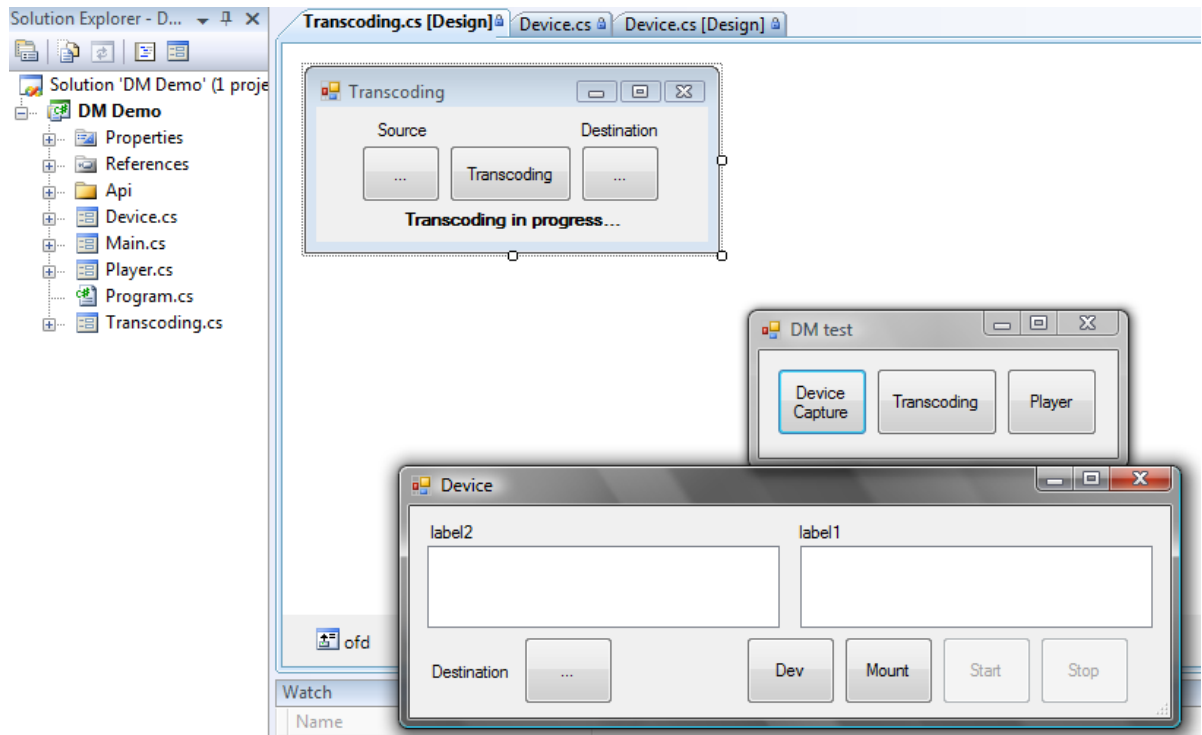
L'entorn de desenvolupament és Visual Studio Express C++.

La metodologia de desenvolupament és en cascada, donat el caràcter rígid de les seves funcionalitats i que es clau per a l'èxit del projecte. Es disposen de tots el requeriments de sortida i es té un coneixement profund de les eines que es fan servir.

Al final de projecte es proveeix d'una API específica per a la DLL, que poden integrar tercers fabricants de software.

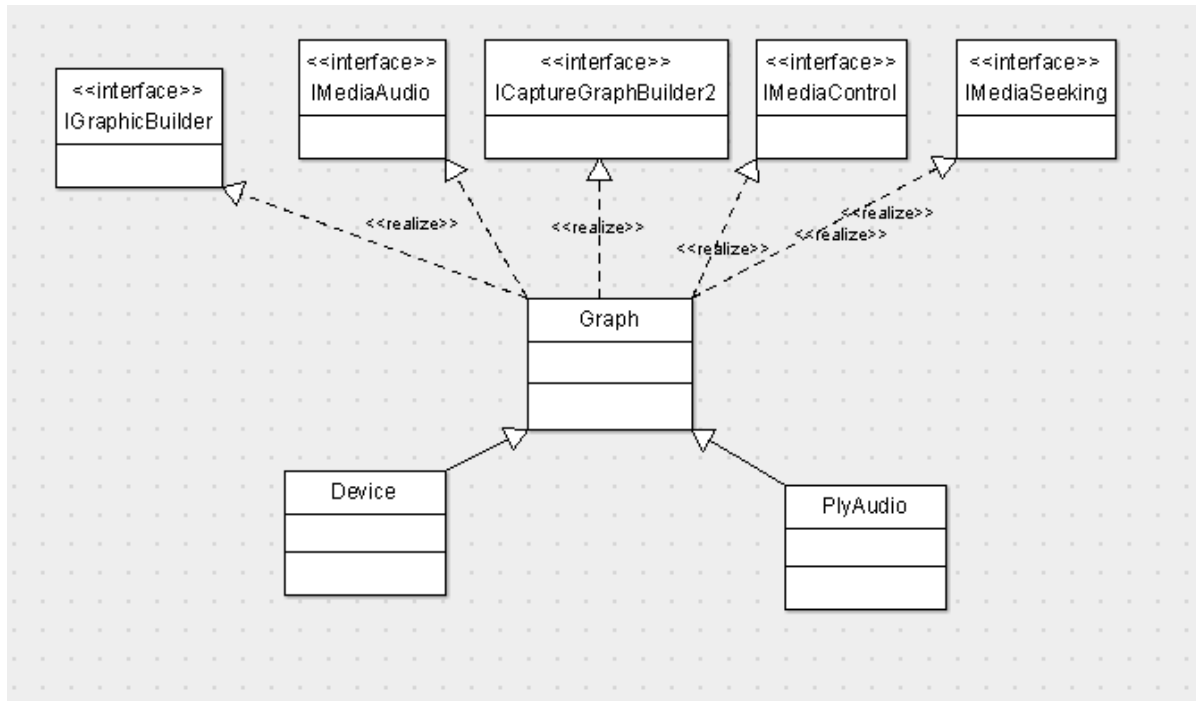
El codecs de transcodificació que s'han implementat són [Ogg](#) de [Vorbis](#).

Es dissenya una aplicació de test **DM Demo** en C# per poder fer el test de l'API en l'etapa prèvia del desenvolupament.



## Diagrama de classes

Es defineix el següent diagrama de classes per DM.



La classe Graph és l'encarregada d'assumir tota la part de gestió d'un graph de DirectShow, per tant, és la que implementa més mètodes i interfaces. Conté el gruix de la lògica de gestió del hardware.

Directament d'ella deriva la classe Device que, sobre els graphs de DirectShow, tracta el control de dispositius de hardware i captura.

També deriva directament PlyAudio que aporta uns mètodes especialitzats en la reproducció d'audio.

### Constants genèriques i macros

Constants de DM:

```

//generic codes
#define DM_OK 0
#define DM_NOT_FOUND 1
#define DM_PLAY 2
#define DM_STOP 3
#define DM_PAUSE 4
#define DM_MOUNT 5
#define DM_UMOUNT 6
  
```

```

//warning codes
#define DM_NOT_ENOUGH_STRING_SIZE 100
#define DM_NOT_INTERFACE_ICONTROL 101
#define DM_ALREADY_PLAY 102
#define DM_ALREADY_STOP 103
#define DM_ALREADY_PAUSE 104
#define DM_NOT_INTERFACE_IBASICAUDIO 105
#define DM_NOT_INTERFACE_ISEEK 106
#define DM_INVALID_AUDIOVALUE 107
#define DM_NOT_DURATION_AVAIABLE 107
#define DM_NOT_INTERFACE_IGRAPH 108
#define DM_ALREADY_MOUNTED 109

//error codes
#define DM_ERROR_COINITIALIZEEX 1000
#define DM_ERROR_COCREATEINSTANCE 1001
#define DM_ERROR_COCREATECLASSENUMERATOR 1002
#define DM_ERROR_BINDINDTOOBJECT 1003
#define DM_ERROR_NULLPOINTER 1004
#define DM_ERROR_SETFILTERGRAPH 1005
#define DM_ERROR_QUERYICONTROL 1006
#define DM_ERROR_QUERYISEEK 1007
#define DM_ERROR_QUERYIBASICADIO 1008
#define DM_ERROR_RUNGRAPH 1009
#define DM_ERROR_STOPGRAPH 1010
#define DM_ERROR_PAUSEGRAPH 1011
#define DM_ERROR_PUTVOLUME 1012
#define DM_ERROR_GETVOLUME 1013
#define DM_ERROR_SETPOSITION 1014
#define DM_ERROR_GETPOSITION 1015
#define DM_ERROR_GETDURATION 1016
#define DM_ERROR_NOT_FIND_PIN 1017
#define DM_ERROR_NOT_CONNECT_PIN 1018
#define DM_ERROR_ISTORAGE_STREAM 1019
#define DM_ERROR_STGCREATE_DOCFILE 1020
#define DM_ERROR_QUERYIPERSISTSTREAM 1021
#define DM_ERROR_PERSIST_SAVE 1022
#define DM_ERROR_STORAGE_COMMINT 1023
#define DM_ERROR_ADD_FILTER 1024
#define DM_ERROR_CHOOSE_ENCODER 1025
#define DM_ERROR_FILTER_ENUMPINS 1026
#define DM_ERROR_QUERY_AUDIOINPUTMIXER 1027
#define DM_ERROR_ENABLE_AUDIOINPUTMIXER 1028
#define DM_ERROR_RENDER_PIN 1029

//Safe Release
#define SAFE_RELEASE(p) {if(p){(p)->Release();(p)=NULL;}}

//Device Constants
#define DM_MAX_ENUM_DEVICES 32
#define DM_MAX_DEVICE_NAME 128
#define DM_MAX_ENUM_PINS 32
#define DM_MAX_PIN_NAME 128
#define DM_DEVICE_OPEN_TAG "#" //define DM_DEVICE_OPEN_TAG "<Devices>"
#define DM_DEVICE_CLOSE_TAG "#" //define DM_DEVICE_CLOSE_TAG "</Devices>"

```

```

#define DM_DEVNAME_OPEN_TAG "$" //define DM_DEVNAME_OPEN_TAG "<DevName>"
#define DM_DEVNAME_CLOSE_TAG "$" //define DM_DEVNAME_CLOSE_TAG "</DevName>"
#define DM_PIN_OPEN_TAG "%" //define DM_PIN_OPEN_TAG "<Pins>"
#define DM_PIN_CLOSE_TAG "%" //define DM_PIN_CLOSE_TAG "</Pins>"
#define DM_PINNAME_OPEN_TAG "&" //define DM_PINNAME_OPEN_TAG "<PinName>"
#define DM_PINNAME_CLOSE_TAG "&" //define DM_PINNAME_CLOSE_TAG "</PinName>"
#define DM_PINCLASS_OPEN_TAG "=" //define DM_PINCLASS_OPEN_TAG "<PinClass>"
#define DM_PINCLASS_CLOSE_TAG "=" //define DM_PINCLASS_CLOSE_TAG "</PinClass>"

//File Async
#define DM_FILE_ASYNC "File Source (Async.)"

//Device Name
#define DM_SOURCE_DEVICE "Source Device"

//Device Name
#define DM_SMART_TEE "Smart Tee"

//Encoder Constants
#define DM_MAX_ENUM_ENCODERS 32
#define DM_MAX_ENCODER_NAME 128
#define DM_ENCODER_OPEN_TAG "?"
#define DM_ENCODER_CLOSE_TAG "?"

//Encoders Allowed
#define DM_VORBIS_OGG_ENCODER "Vorbis Encode Filter"
#define DM_VORBIS_OGG_WRITER "Ogg Mux Filter"

```

## Classe Graph

Es desglossen els seus mètodes públics ordenats per funcionalitat :

<pre>int PlayGraph (double dTcIn=-1.0);</pre>	<p>Posa en 'play' un Graph construït prèviament des del segon especificat al paràmetre dTcIn (segons.milisegons)</p> <p>Per defecte té un valor de -1.0, que indica que ha de començar des del principi.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<pre>int StopGraph (void);</pre>	<p>Posa en estat 'stop' un Graph construït.</p>

	<p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int PauseGraph (void);</code>	<p>Posa en estat 'pause' un Graph construït.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int GetStatus(void);</code>	<p>Retorna l'estatus d'un Graph. Pot ser una de les següents constants:</p> <pre>//generic codes #define DM_OK 0 #define DM_NOT_FOUND 1 #define DM_PLAY 2 #define DM_STOP 3 #define DM_PAUSE 4 #define DM_MOUNT 5 #define DM_UMOUNT 6</pre> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>void SetSaveGraph (bool bEnable);</code>	<p>Habilita\desabilita la generació d'una gràfica visual (*.grf) del graph construït. Paràmetre booleà bEnable.</p> <p>És un element molt important per a depurar l'aplicació en el seu desenvolupament.</p> <p>L'arxiu final es pot visionar amb l'eina <a href="#">Graphedit</a></p>
<code>BOOL GetSaveGraph (void);</code>	<p>Retorna si està habilitada\desabilitada la gravació del graph en un fitxer.</p>
<code>int SetVolume (int value);</code>	<p>Estableix el volum de la reproducció del graph. El paràmetre value ha de tenir valors entre 0 i 100.</p> <p>L'augment de volum és logarítmic, no geomètric.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de</p>

	constants de DM.
<code>int GetVolume (int *value);</code>	<p>Obté el volum actual. Valor entre 0 i 100.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int SetPosition (double position);</code>	<p>Es posiciona en un punt concret del graph de reproducció.</p> <p>El paràmetre position s'expressa amb segons.milisecons</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int GetPosition (double *position);</code>	<p>Per referència en el paràmetre position s'obté el punt de reproducció actual, en format segons.milisecons.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int GetRawPosition (LONGLONG *position);</code>	<p>Per referència en el paràmetre position s'obté el punt de reproducció actual, amb precisió de milisecons * 10.000</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int GetDuration (double *duration);</code>	<p>Per referència en el paràmetre duration s'obtenen els segons.milisecons del media carregat al graph.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int GetRawDuration (LONGLONG *lIDuration);</code>	<p>Per referència en el paràmetre duration s'obtenen els milisecons * 10.000 del media carregat al graph.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per</p>



	consultar els possibles errors veure la llista de constants de DM.
<code>int AddFilter (IBaseFilter *pFilter, char *FilterName);</code>	<p>Afegeix el Filtre pFilter amb el nom FilterName al graph actual.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int Render (IBaseFilter *pFilter);</code>	<p>Renderitza des del filtre pFilter el graph actual.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int Render (IPin *pPin);</code>	<p>Renderitza des del pPin el graph actual.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int Render (char *cFile);</code>	<p>Renderitza des de l'arxiu cFile sobre el graph actual.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>int SaveGraphFile(char *cPath);</code>	<p>Grava en cPath el graph actual per després ser visionat amb Graphedit.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<code>void Unload (void);</code>	Descarrega els filtres del graph actual.

**Classe Device**

Es desglossen els seus mètodes públics ordenats per funcionalitat:

<pre>int Mount (char *cDevFile, char *cEnco, char *cStorage, bool bDev);</pre>	<p>Monta un graph des del device o arxiu origen de transcoding cDevFile, amb el codec cEnco amb destí cStorage.</p> <p>Si el valor bDev és true indica que es tracta d'una captura. Amb valor false es tracta d'una transcodificació.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<pre>int Umount (void);</pre>	<p>Desmonta un Graph muntat prèviament.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<pre>int BlockWhileTranscoding(void);</pre>	<p>Durant un procés de transcodificació, si es crida aquest mètode, la crida queda bloquejada fins que el procés acaba.</p> <p>Aquest procés no bloqueja altres transcodificacions, donat que s'ha implementat la classe amb tecnologia multithread.</p> <p>Retorna True en cas d'èxit; en cas contrari False.</p>
<pre>static int EnumFilters (char *cEnumDev, int iSizeStream);</pre>	<p>Mètode estàtic que per referència cEnumDev retorna una llista de dispositius de hardware de captura d'audio.</p> <p>iSizeStream té el tamany en bytes de cEnumDev.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>

<pre>static int EnumPins (char *sFilterName, char *cEnumDev, int iSizeStream);</pre>	<p>Mètode estàtic que per referència sFilterName retorna una llista dels pins disponibles pel hardware cEnumDev.</p> <p>iSizeStream té el tamany en bytes de sFilterName.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<pre>static int GetFilter (char *sFilterName, REFCLSID clsidFilterClass, IBaseFilter **ppFilter);</pre>	<p>Obté un filtre per referència ppFilter de la classe especificada a clsidFilterClass amb el nom sFilterName.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
<pre>static int GetPin (IBaseFilter *pDevice, char *cSeekPinName, IPin **pPin);</pre>	<p>Obté el pin per referència pPin del device especificat a pDevice, buscant pel text cSeekPinName.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>

### **Classe PlyAudio**

Es desglossen els seus mètodes públics:

<pre>int LoadFile (char *cFile);</pre>	<p>Carrega un arxiu especificat a cFile al graph i el preparara per ser reproduït.</p> <p>Retorna DM_OK en cas d'èxit de la crida. Per consultar els possibles errors veure la llista de constants de DM.</p>
--	---

### **Mètodes exportables de la DLL**

<pre>//Enum Func extern "C" _declspec(dllexport) int EnumDevDMI (char *cEnumDev, int iSize); extern "C" _declspec(dllexport) int EnumEncodersDMI (char *cEnumEncoders, int iSize);</pre>
--

```
//Extern Func Recorder
extern "C" _declspec(dllexport) int CreateDMI (char *cDevFile, char *cEnco, char *cDst, bool bDev);
extern "C" _declspec(dllexport) int DeleteDMI (int hDMI);

extern "C" _declspec(dllexport) BOOL RecDMI (int hDMI);
extern "C" _declspec(dllexport) BOOL StopDMI (int hDMI);
extern "C" _declspec(dllexport) BOOL PauseDMI (int hDMI);
extern "C" _declspec(dllexport) int StatusDMI (int hDMI);
extern "C" _declspec(dllexport) int StatusTrascodingDMI (int hDMI);
extern "C" _declspec(dllexport) int BlockWhileTrascoding (int hDMI);

//Extern Func Player
extern "C" _declspec(dllexport) int CreateDMP (void);
extern "C" _declspec(dllexport) int DeleteDMP (int hDMP);

extern "C" _declspec(dllexport) int LoadFileDMP (int hDMP, char *cFile);
extern "C" _declspec(dllexport) BOOL PlayDMP (int hDMP);
extern "C" _declspec(dllexport) BOOL StopDMP (int hDMP);
extern "C" _declspec(dllexport) BOOL PauseDMP (int hDMP);
extern "C" _declspec(dllexport) int StatusDMP (int hDMP);
```

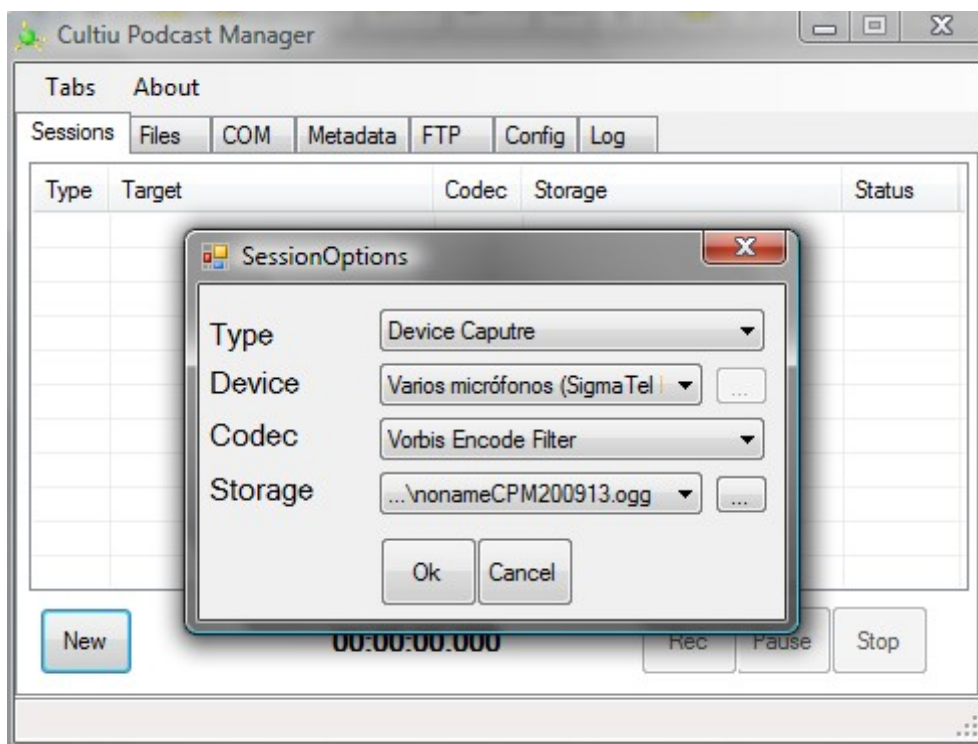
## Catalogació i client FTP, Cultiu Podcast Manager ( CPM )

Cultiu Podcast Manager és una aplicació desenvolupada amb .NET amb una GUI d'usuari que permet realitzar accions de captura/ingesta i upload amb metadades als servidors FTP que són els receptors del [Podcast](#) d'audio.

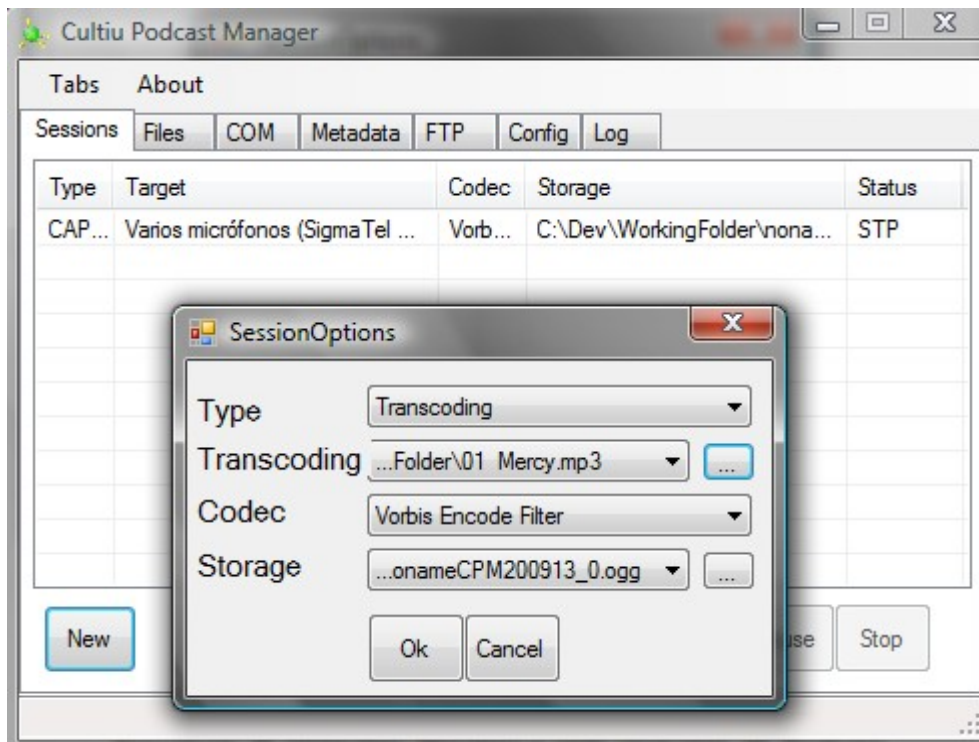
Com a marc de desenvolupament es fa servir VS Express C#. Té com Framework base l'actual (3.5), incorporant elements nous com [LINQ](#). És una aplicació WinForms.

Elements bàsics de CPM:

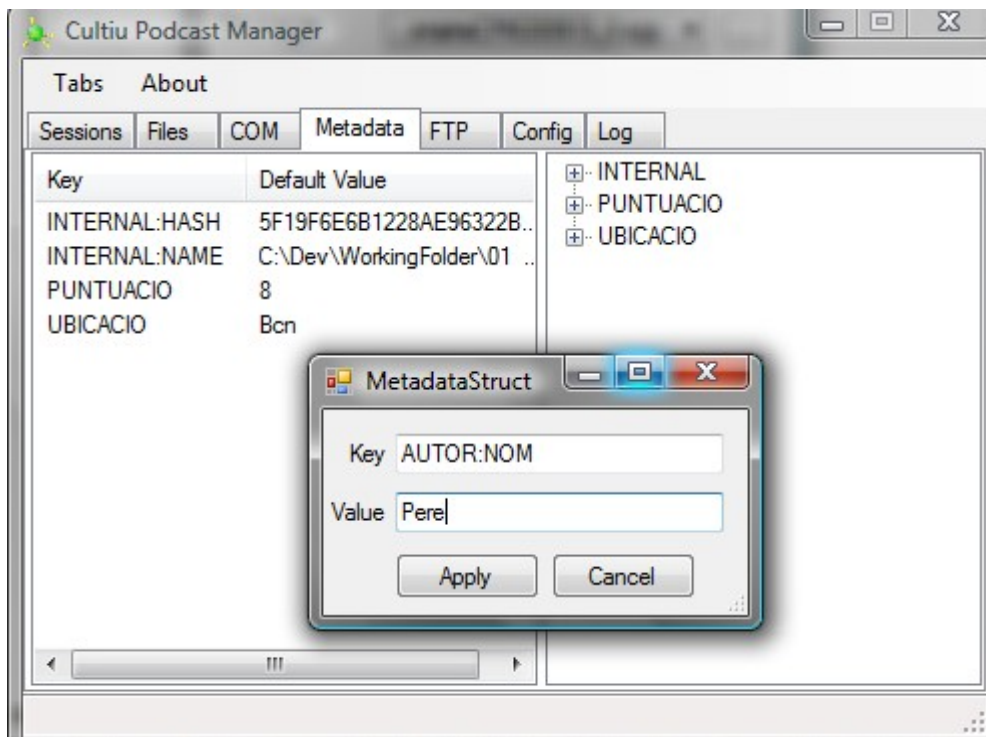
- Possibilitat de Caputa/Ingesta múltiple en paral·lel amb diversos dispositius de captura.



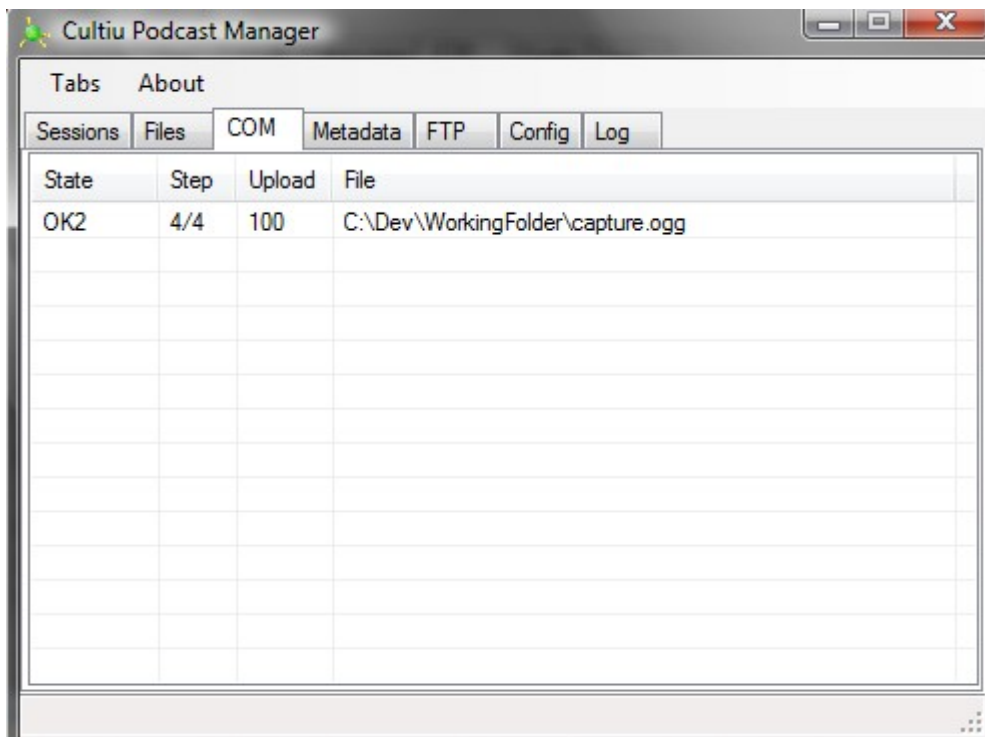
- Transcodificacions d'arxius existents.



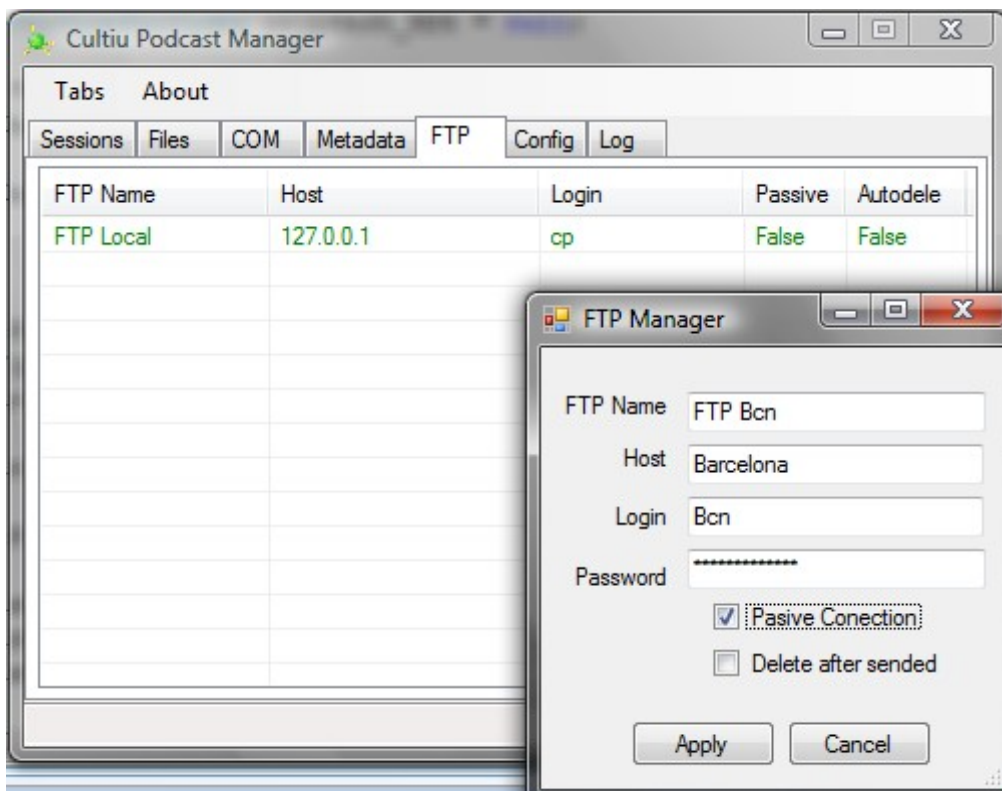
- Associació de Metadata Dinàmica amb jerarquia als Podcast



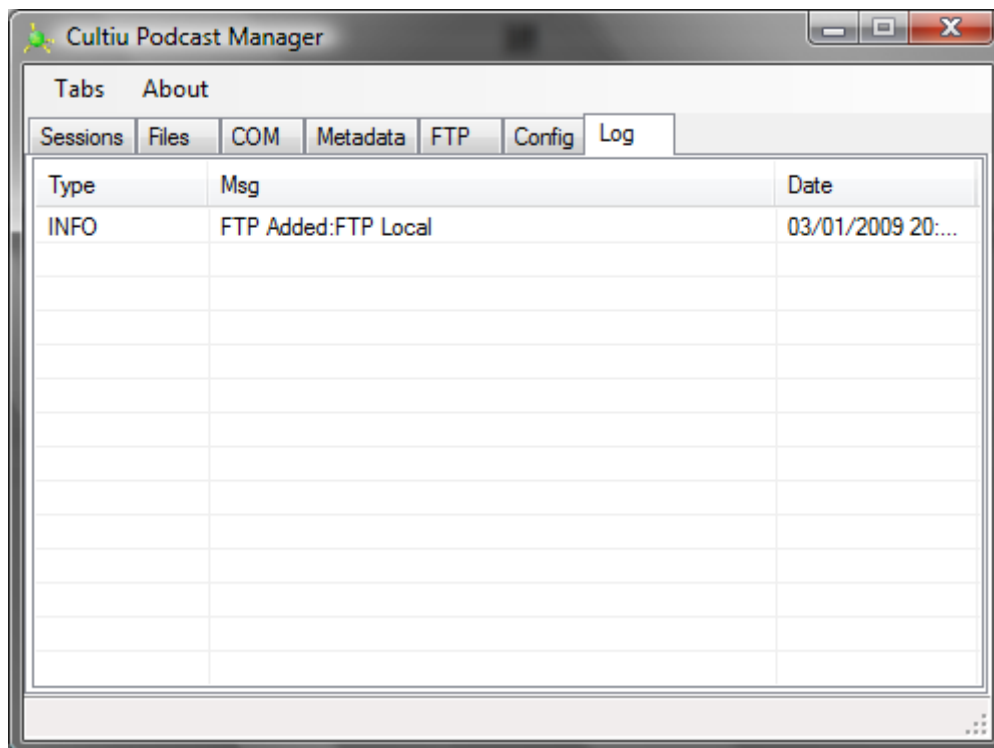
- Control d'estat de les comunicacions amb el servidor destí FTP.



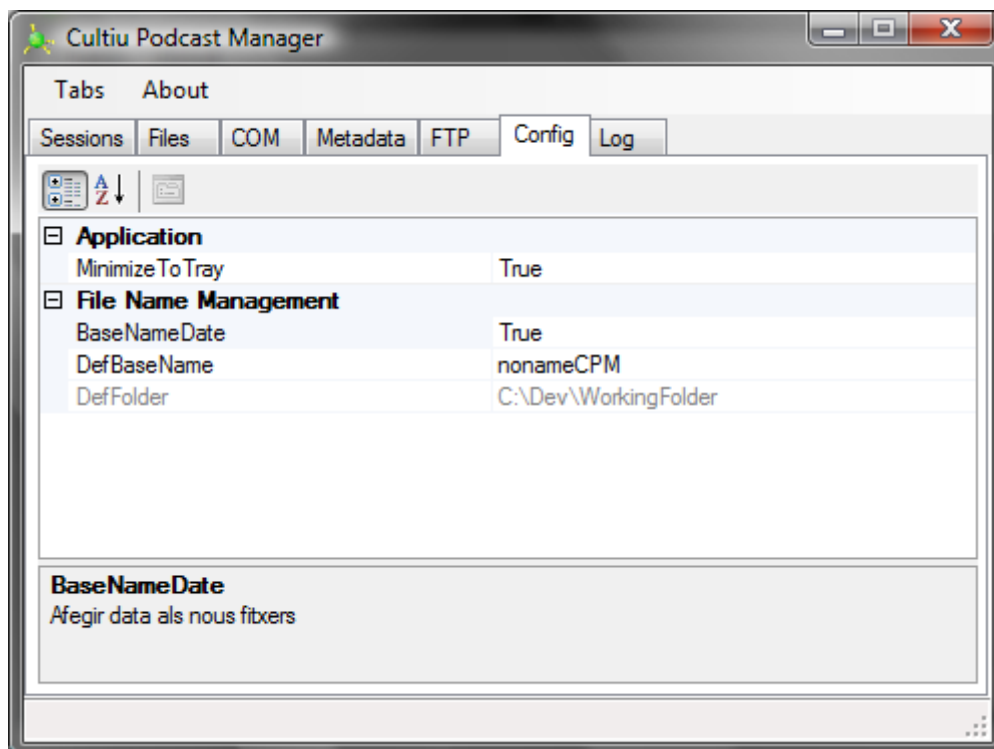
- Configuració de clients FTP.



- Gestor de Logs.



- Configuració de l'aplicació.





La metodologia que emprada per al desenvolupament és en espiral. Això permet tenir un desenvolupament flexible i constant, re-avaluant la situació després d'assolir els objectius.

Amb aquesta metodologia es redueixen riscos en el projecte, donat que sempre es té un 'cicle espiral' acabat, i dóna flexibilitat per introduir constantment detalls nous.

## **Metadata Dinàmica**

La Metadata Dinàmica és un sistema que permet:

- Afegir de 0 a N metadates d'informació a un Podcast.
- Afegir estructures de metadates diferents a arxius en un mateix sistema.
- Crear jerarquia en la creació de metadata.

La metadata té com a element únic i indivisible el Key, que és una cadena de text alfanumèric irrepètible dins d'una mateixa jerarquia.

El Key es guarda sempre en majúscules, fet pel qual es diu que el sistema és 'Case Insensitive'.

Un Key pot tenir un valor associat, però no és imprescindible. Aquest valor s'anomena Value.

Un Tag és un Key i un Value separats per l'operador '='. Exemple: NOM=ANNA.

Per crear jerarquia es fa servir l'operador '!'.

Els Keys INTERNAL:HASH i INTERNAL:NAME estan reservats i no es poden fer servir amb propòsit general.

## **Classe MetadataTag**

Es desglossen els seus mètodes públics ordenats per funcionalitat:

<pre>public MetadataTagErr AddTag(string sTag, string sValue)</pre>	<p>Afegeix un nou Tag en memòria: els paràmetres sTag i sValue.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
---	---

<code>public MetadataTagErr ReplaceTag(string sOldKey, string sNewKey, String sNewValue)</code>	<p>Reemplaça el tag amb Key igual a sOldKey per sNewKey i valor sNewValue.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public MetadataTagErr DelTag(string sKey)</code>	<p>Esborra de memòria el Key del paràmetre sKey.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public bool ChangeValue(string sKey, string sValue)</code>	<p>Canvia el valor del Key sKey pel posat al paràmetre sValue.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public string GetTag(int index)</code>	<p>Obté el Tag de la posició del paràmetre index.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public string GetKey(int index)</code>	<p>Obté el Key de la posició del paràmetre index.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public string GetValue(int index)</code>	<p>Obté el Value de la posició del paràmetre index.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public string GetValueByKey(string sKey)</code>	<p>Obté el Value de Key associat.</p> <p>Retorna un valor de l'enumeració MetadataTagErr.</p>
<code>public string GetTagByKey(string sKey)</code>	<p>Obté el Tag de Key associat.</p> <p>Retorna un valor de l'enumeració</p>

	MetadataTagErr.
<code>public int GetCount()</code>	Retorna el número de Tags en memòria.
<code>public void SetProtected(String sProtected)</code>	Mètode que protegeix de la modificació del Key especificat a sProtected.
<code>public String GetProtected()</code>	Rertorna el Key protegit per no ser modificat.
<code>public SortedList GetStruct()</code>	Retorna una estructura SortedList d'elements ordenats dels Tags que conté en memòria.
<code>public void CopyElements(MetadataTag mt)</code>	Crea una còpia de sí mateix en el paràmetre mt.
<code>public MetadataTagErr Save(string sDest)</code>	Fa persistenst les dades de memòria en un arxiu definit a sDest.  Rertorna el Key protegit per no ser modificat.
<code>public MetadataTagErr Open(string sScr)</code>	Llegeix un arxiu sScr per a carregar les dades guardades prèviament amb el mètode Save.  Rertorna el Key protegit per no ser modificat.

## Enumeracions:

```

NO_ERROR = 0
NOT_ALLOWED_CHAR = 1
NOT_ALLOWED_EMPTY = 2
NOT_ALREADY_EXISTS = 3
SAVE_ERROR = 4
OPEN_ERROR = 5
NOT_ALLOWED_TAG_PROTECTED = 6
KEY_NOT_FOUND=7

```

La classe MetadataTag es fa servir també en el mòdul CPFWD. Es fa un ús intensiu de la filosofia de [POO](#) per reutilització de codi i optimització dels temps de desenvolupament.

També succeix el mateix amb la classe ObdcDataBase que gestiona l'accés a la BBDD entre CPFWD i CPC.

FileHash és un altre cas entre CPM i CPC.

## Publicació, Cultiu Podcast File Watch Dog (CPFWD)

CPFWD és una eina que està monitoritzant el directori del FTP on es dipositen els Podcast enviats prèviament pel CPM.

Un cop detecta un nou Podcast, és l'encarregada de:

- Inserir transaccionalment el nou Podcast a la BBDD.
- Afegir noves Metadates Dinmiques del Podcast a la BBDD.

SQLQuery1.sql - BU1\...\Bu (54))\*

```

select * from dbo.CPC_Podcast;
select * from dbo.CPC_PodcastTag;
select * from dbo.CPC_Tag;

```

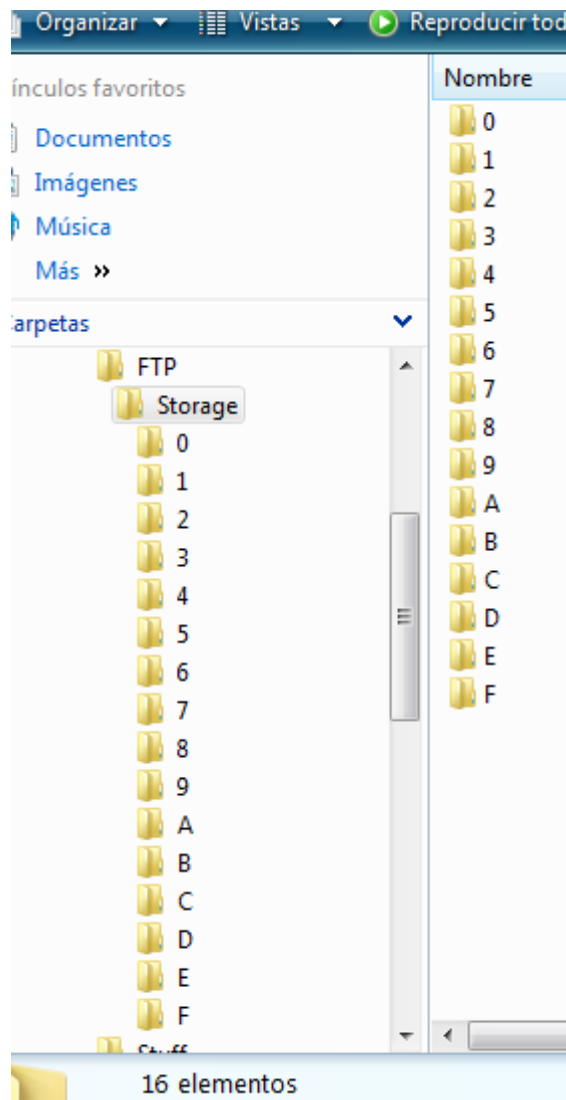
Results Messages

idPodcast	Hash	Path	Name	Extension
1	B66094C5B307D0A3D458DB15690E438C	C:\Dev\FTP\Storage\B\B66094C5B307D0A3D458DB15690...	capture	ogg
2	D41D8CD98F00B204E9800998ECF8427E	C:\Dev\FTP\Storage\D\D41D8CD98F00B204E9800998ECF...	mercy	ogg

idPodcast	idTag	Value
1	2	25
2	1	Ricard
3	1	5
4	1	6
5	1	7
6	1	8
7	2	20
8	2	3

idTag	idParent Tag	Tag
1	NULL	AUTOR
2	1	EDAD
3	1	NOM
4	NULL	INTE...
5	4	HASH
6	4	NAME
7	NULL	PUTU...
8	NULL	UBIC...

- Moure el Podcast a l' 'Storage' definit, a la carpeta correcta.

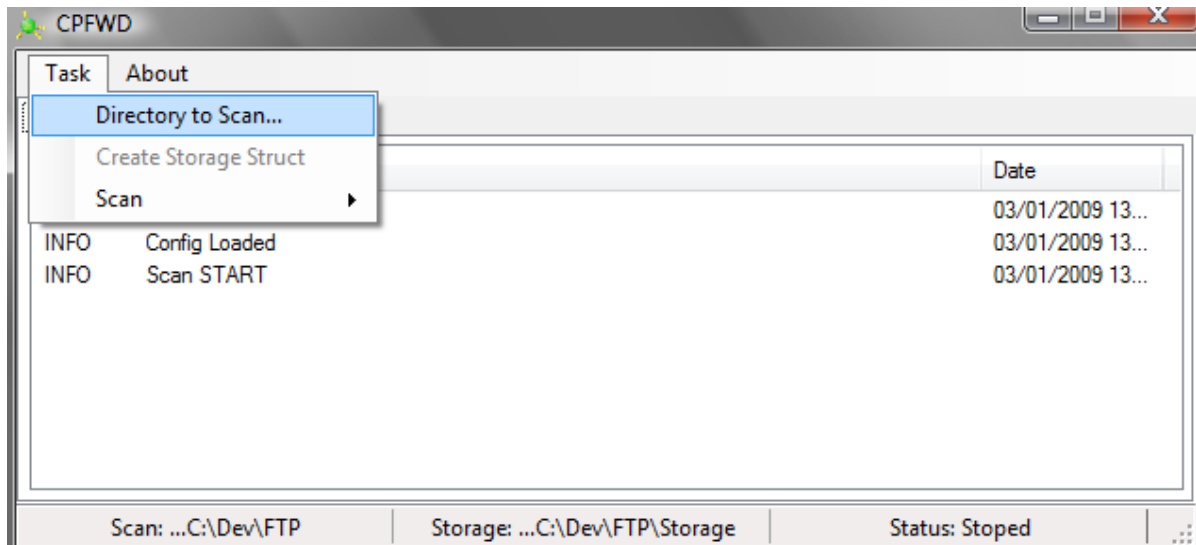


La base és un SQL-Server 2k8. Es fa servir SQL'92, la qual cosa permet poder integrar fàcilment qualsevol servidor de base de dades relacional.

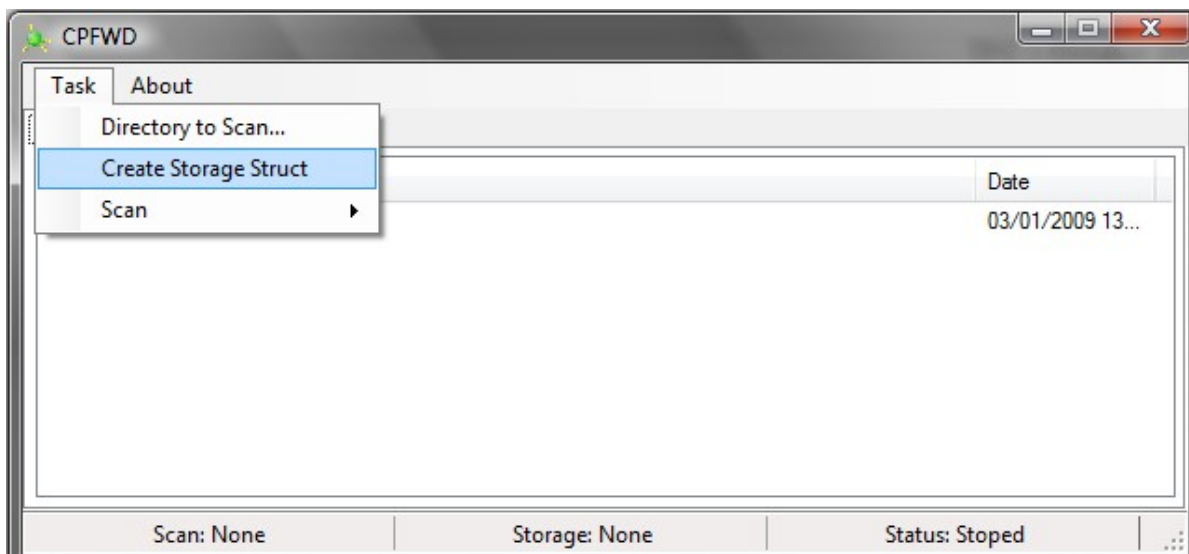
CPFWD també té com a finalitat assegurar la integritat referencial de la BBDD i els arxius al repositori de Podcast. Per tant, realitza tasques de purgació de fitxers erronis o inexistents.

Els elements que s'han de configurar i gestionar en CPFWD per part d'un administrador són:

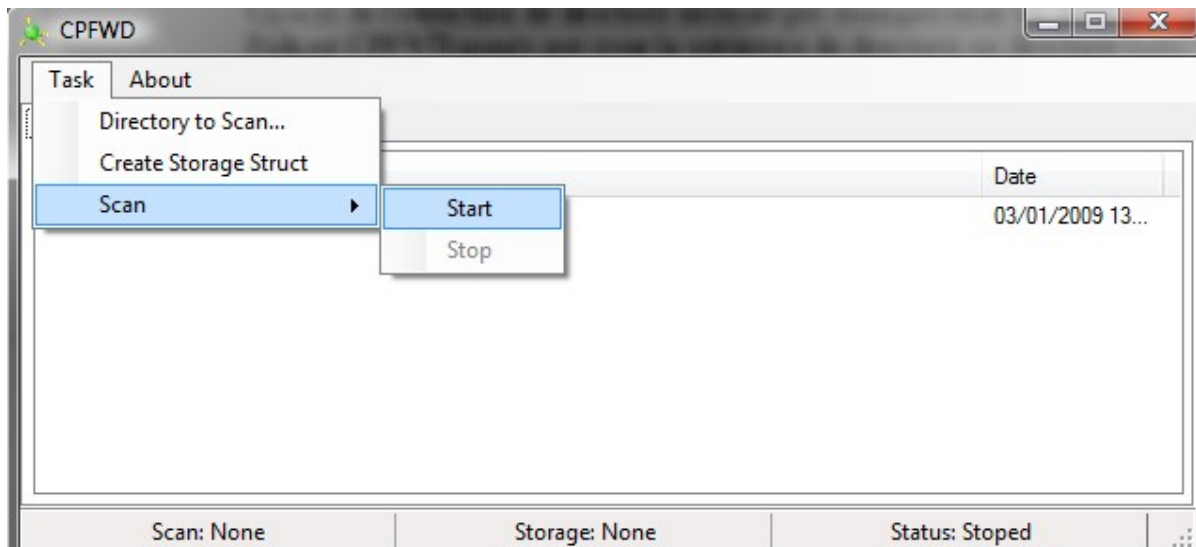
- Configuració d'un directori per rebre els Podcast del CPM.



- Creació de l'estructura de directoris necessària per emmagatzemar els arxius. Podcast CPFWD només pot crear l'estructura de directoris en directoris buits.



- Activar/desactivar la monitorització.



Després de la configuració inicial no és necessària una gestió diària de l'eina, si bé s'ha de fer una monitorització del sistema operatiu per controlar que sempre està funcionant, doncs és vital per a donar d'alta nous arxius.

Com a metodologia de desenvolupament es fa servir el cicle de vida en cascada. Es tracta d'una aplicació que no ha de disposar d'evolucions i d'un caràcter molt rígid per la part funcional de tot el projecte.

El seu desenvolupament es fa amb .NET framework 3.5 i C#.

## **Classe *ObdcDataBase***

La classe *ObdcDataBase* presenta un conjunt de mètodes que ofereix connectivitat amb una BBDD. Basat amb el namespace `System.Data.Odbc`, té tota la lògica de negoci d'accés a un SGBD relacional.

La gestió de possibles errors vinculats a la connexió, errors de sintaxi i integritat referencial de consultes SQL, és tractada amb la propagació d'excepcions.

Exemple del mètode `OpenConnection`:

```
try
{
    m_con = new OdbcConnection(m_sConString);
    m_con.Open();
}
```



```

    }
    catch (OdbcException e)
    {
        throw new Exception("BBDD Error: " + e.Message);
    }

```

Aquesta classe és utilitzada per CPFWD i CPC.

A continuació es descriuen els seus mètodes públics ordenats per funcionalitat:

<code>public void SetTimeout(int Timeout)</code>	Determina el temps d'espera per establir una connexió amb el servidor. Timeout és el paràmetre en segons.
<code>public int GetTimeout()</code>	Retorna el temps d'espera per establir una connexió.
<code>public void SetConnectionString(string ConnectionString)</code>	Estableix la cadena de connexió contra el servidor de base de dades.
<code>public string GetConnectionString()</code>	Retorna la cadena de connexió actual.
<code>public void SetConnectionStringMYSQL(string Ip, int Port, string DatabaseName, string Login, string Pass)</code>	Crea una cadena de connexió específica per un servidor MySQL. Els paràmetres són IP del servidor, Port del servidor, DatabaseName de la base de dades a connectar-se, Login i Pass credencials d'accés.
<code>public void SetConnectionStringSQL2k8(string Ip, string DatabaseName, string Login, string Pass)</code>	Crea una cadena de connexió específica per un servidor SQL Server 2k8. Els paràmetres són IP del servidor, Port del servidor, DatabaseName de la base de dades a connectarse, Login i Pass credencials d'accés.
<code>public void OpenConnection()</code>	Estableix connexió al servidor de BBDD basat en la cadena de connexió.
<code>public void CloseConnection()</code>	Tanca una connexió prèviament establerta amb OpenConnection.

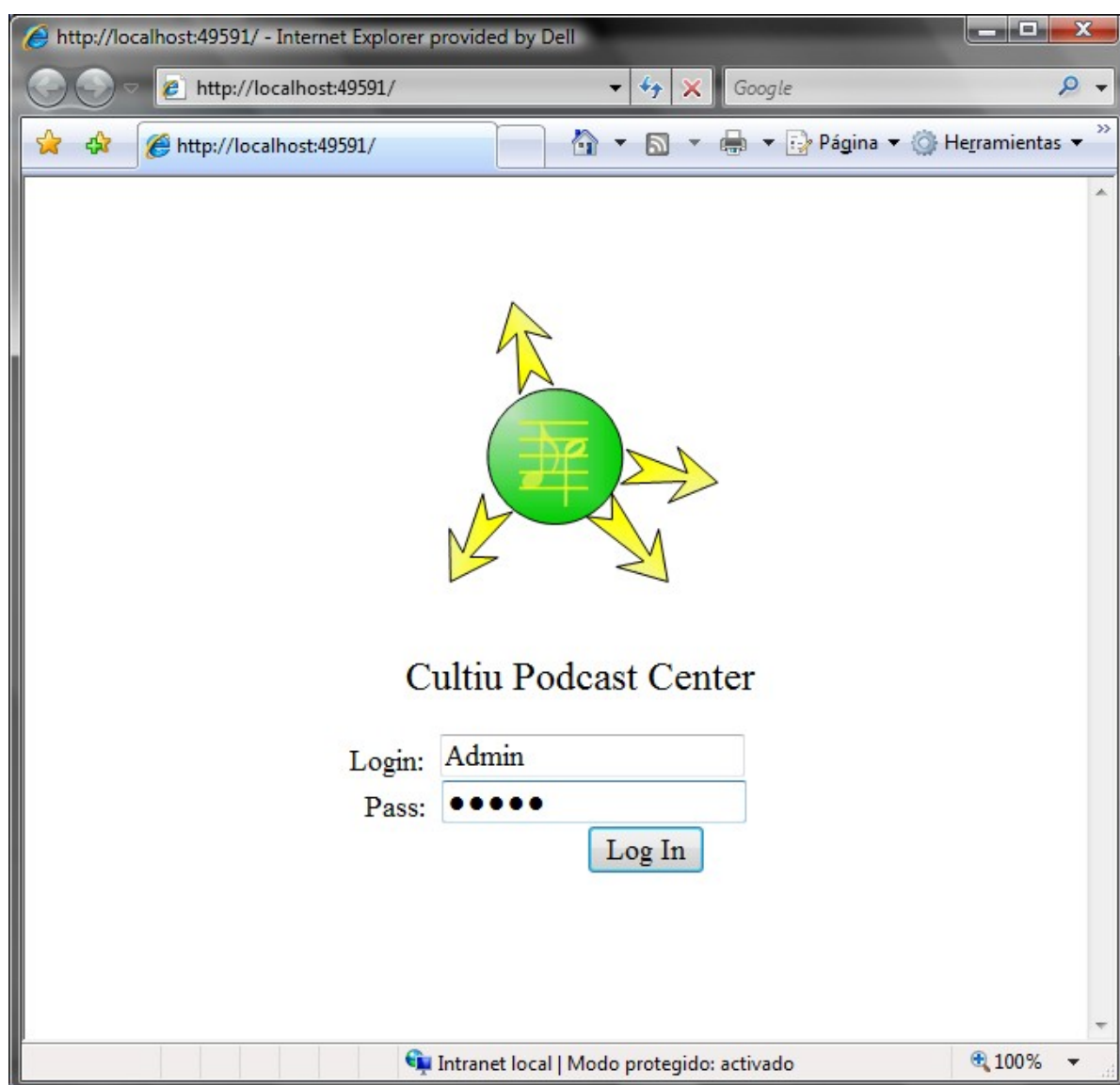
<code>public void TestConnection()</code>	Valida la connectivitat amb el servidor de BBDD configurat.
<code>public void ExecuteScalar(string Query, out int Scalar)</code>	Executa una consulta SQL definida al paràmetre Query. Per referència obté el resultat escalar de la consulta.
<code>public void ExecuteNonQuery(string Query, out int AffectedRows)</code>	Executa una consulta SQL definida al paràmetre Query. Per referència obté el resultat de les files afectades per la consulta.
<code>public void GetNexIDFromTable(string Table, string Field, out int NextID)</code>	Des d'una taula i el camps del paràmetres Table i Field obté el següent valor numèric.

## Accés WWW, Cultiu Podcast Center (CPC)

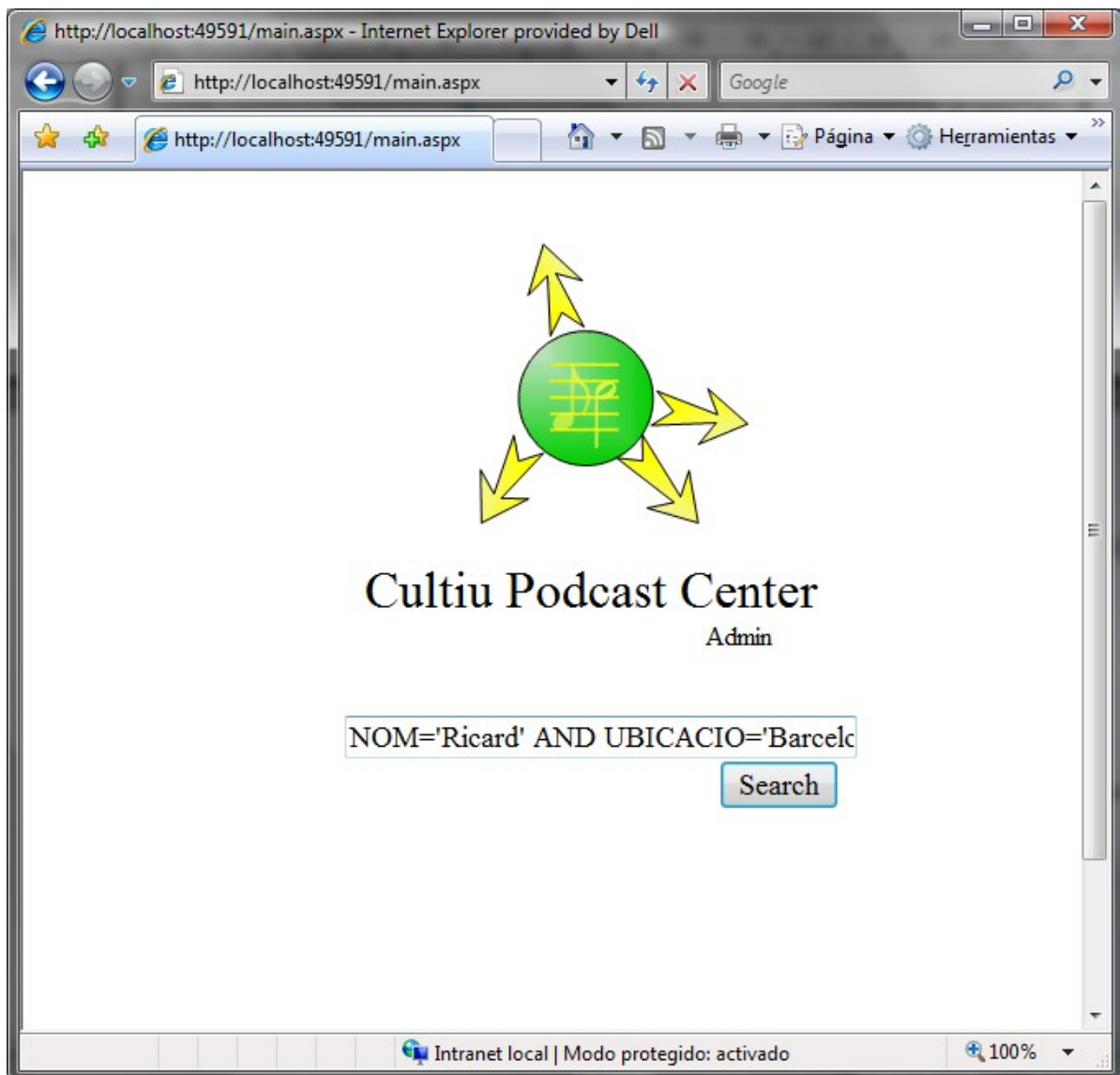
L'accés a tot el contingut de repositori de software es fa amb CPC, que és una aplicació ASP.NET, Framework 3.5 que no pretén ser gaire ambiciosa a nivell de disseny però sí en funcionalitat interna, amb un motor de recerca potent.

Mostra el contingut de la base de dades i pot descarregar un Podcast. Requereix la implantació d'un servidor IIS que sostingui l'aplicació.

Com a pantalla inicial es té una validació de credencials per accedir al CPC. El login i password predefïnits són admin/admin.



Després de la validació es pot fer servir el motor implementat per fer recerques.



L'objectiu de CPC és demostrar que el sistema CP es pot integrar en qualsevol web per fer difusió dels Podcast. En qualsevol cas no és pertenció d'aquest TFC desenvolupar una eina semblant a un [SharePoint](#).

Amb el seu motor de recerca es demostra que en la web es pot fer qualsevol tipus de filtratge a partir de la seva metadata, amb una simple consulta que és transparent per a l'usuari.

Per exemple, un usuari que es connecta a la web i selecciona llengua catalana pot rebre per defecte el contingut de Podcast en català emmagatzemat en el transcurs de l'última setmana.

Per al seu desenvolupament s'ha emprat el desenvolupament en espiral, per les mateixes raons que en el cas de CPM: flexibilitat, possibilitats de millora i evolució constant.

## ***Motor de recerca***

CPC té un motor de recerca que permet, amb una sintaxi còmoda i fàcil, fer recerques de Podcast en el sistema. A continuació es descriu el seu funcionament.

Les recerques es poden realitzar sobre l'existència d'un Key i/o un Tag. De caràcter general els Keys es guarden en format 'Upper Sense' ( Majúscules ) fet pel qual la recerca és sempre 'Case Insesitive'.

### **Recerques sobre Key:**

Es poden fer recerques des de 1 fin a N Keys.

Per defecte es fa ús intensiu del comodí en la cerca sobre la BBDD.

Per a realitzar recerques de múltiples Keys es fan servir les clàusules AND/OR, amb una lògica separació d'espais.

Per defecte la separació de dos Keys per espais implica la utilització de la clàusula OR.

Exemples:



El motor retorna els Podcast que tenen un Key de metadata que conté els caràcters NOM, com són 'nom', 'nomenclatura', 'pronom'. ( equival a \*Nom\* )



Obté els Podcast que tenen un Key que segueix la diposició de \*NOM\* o bé \*CUITAT\*.

Amb aquesta consulta s'obtenen els mateixos resultats que a l'anterior.



S'obtenen els Podcast que tenen el Key que segueix la disposició de \*NOM\* i també el de \*CIUTAT\*.



Amb aquesta recerca s'obtenen els Podcast que tenen el Key \*CIUTAT\* i després tenen el Key \*POBLACIO\* o bé \*NOM\*.

### Recerques sobre Tag:

Es poden fer recerques des de 1 fins a N Tags.

**No** es fa ús del comodí en la cerca sobre la BBDD.

El caràcter que s'utilitza per a separar el Key i el Valor en un Tag és l'igual '='.

Per a realitzar recerques de múltiples Tags es fan servir les clàusules AND/OR, amb una lògica separació d'espais.

Per defecte la separació de dos Tags per espais implica la utilització de la clàusula OR.

Exemples:

El motor retorna els Podcast que tenen un Tag de metadata que conté els caràcters NOM amb el valor de ROBERT assignat ( equival a NOM=ROBERT )

S'obtenen els Podcast que tenen un Tag que segueix la disposició de NOM=ROBERT o bé UBICACIO=BCN.

Amb aquesta consulta s'obtenen els mateixos resultats que a l'anterior.

S'obtenen els Podcast que tenen un Tag que segueix la disposició de NOM=ROBERT i també UBICACIO=BCN.

PCT=20 PCT=80 AND NOM=ANNA

Search

Amb aquesta recerca s'obtenen els Podcast que tenen el Tag PCT=20 o bé PCT=80 i després tenen el Tag NOM=ANNA.

### Recerca combinada:

El motor de recerca permet fer recerques combinades amb tots dos estils tot complint les normes prèvies.

Exemple:

UBICACIO AND NOM=ANNA

Search

Aquest criteri de recerca retorna els Podcast que tenen el Key \*UBICACIO\* i a més tenen el Tag NOM=ANNA.

NOM=PERE NOM=PAU AND SALARI

Search

S'obtenen els Podcast que tenen Tag NOM=PERE o bé NOM=PAU i també tenen un Key \*SALARI\*

### Observacions

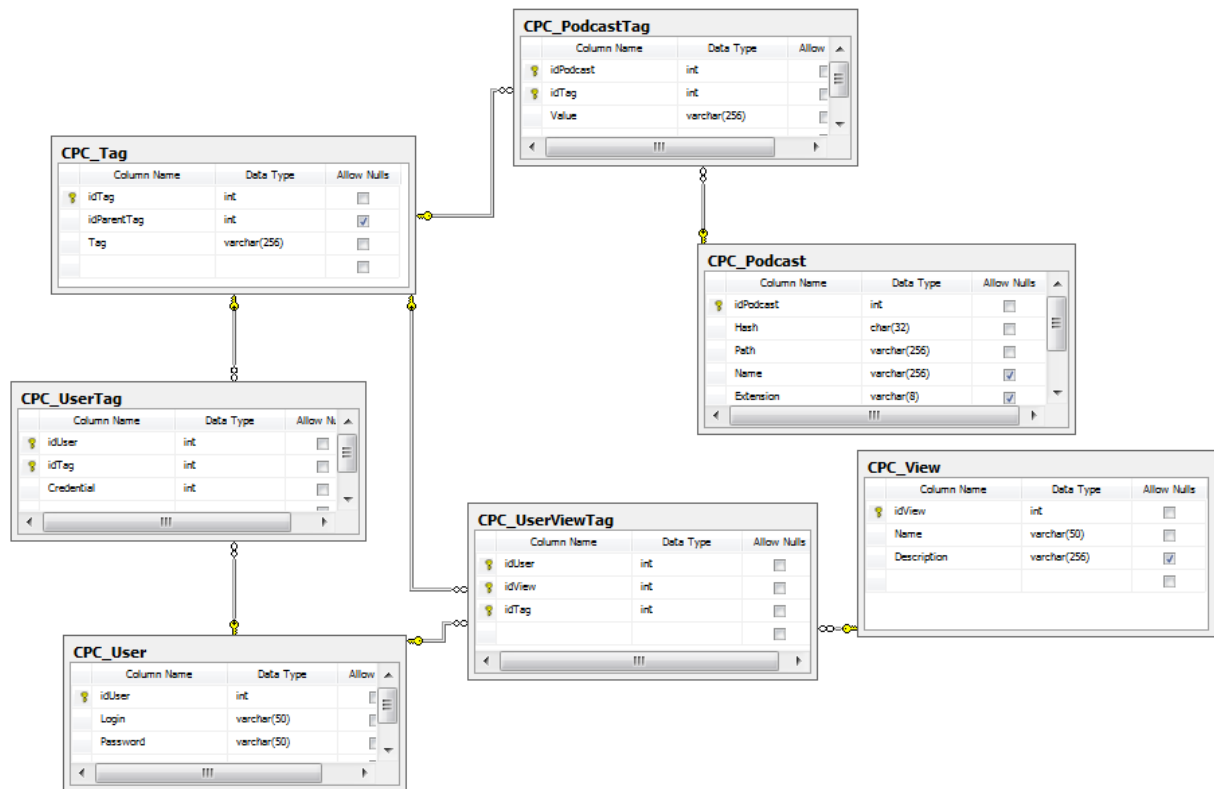
El motor de recerca està preparat per corregir alguns errors en la introducció dels criteris de recerca, com pot ser la inserció de múltiples espais per separar elements de recerca.



Aquesta sintaxi de recerca la faria servir qualsevol programador d'una web que mostrés els Podcast. Això li permetria una fàcil integració al seu sistema. No és pretensió d'aquest projecte crear un entorn web final.

**BBDD**

## Diagrama de BBDD



## Script de BBDD per SQL Server 2k8

```

/* Creació de la BBDD */
USE [master]
GO

CREATE DATABASE [CPC] ON PRIMARY
( NAME = N'Cpc_data',
FILENAME = N'D:\Archivos de programa\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\Cpc_data.mdf',
SIZE = 307200KB , MAXSIZE = 1048576KB , FILEGROWTH = 50%)
LOG ON
( NAME = N'CPC_log',
FILENAME = N'D:\Archivos de programa\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\CPC_log.ldf',
SIZE = 30720KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

```

```

COLLATE Modern_Spanish_CI_AI
GO
EXEC dbo.sp_dbcmptlevel @dbname='CPC', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [CPC].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [CPC] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [CPC] SET ANSI_NULLS OFF
GO
ALTER DATABASE [CPC] SET ANSI_PADDING OFF
GO
ALTER DATABASE [CPC] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [CPC] SET ARITHABORT OFF
GO
ALTER DATABASE [CPC] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [CPC] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [CPC] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [CPC] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [CPC] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [CPC] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [CPC] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [CPC] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [CPC] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [CPC] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [CPC] SET ENABLE_BROKER
GO
ALTER DATABASE [CPC] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [CPC] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [CPC] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [CPC] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [CPC] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [CPC] SET READ_WRITE
GO
ALTER DATABASE [CPC] SET RECOVERY SIMPLE
GO
ALTER DATABASE [CPC] SET MULTI_USER
GO

```

```
ALTER DATABASE [CPC] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [CPC] SET DB_CHAINING OFF
```

Script de taules amb constraints/foreign keys per SQL Server 2k8

```

/***** [dbo].[CPC_Podcast] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CPC_Podcast]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_Podcast]
GO

USE [CPC]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[CPC_Podcast](
    [idPodcast] [int] NOT NULL,
    [Hash] [char](32) NOT NULL,
    [Path] [varchar](256) NOT NULL,
    [Name] [varchar](256) NULL,
    [Extension] [varchar](8) NULL,
    CONSTRAINT [PK_CPC_Podcast] PRIMARY KEY CLUSTERED
(
    [idPodcast] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

/***** Table [CPC_Tag] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CPC_Tag]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_Tag]
GO

```

```

USE [CPC]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[CPC_Tag](
    [idTag] [int] NOT NULL,
    [idParentTag] [int] NULL,
    [Tag] [varchar](256) NOT NULL,
    CONSTRAINT [PK_CPC_Tag] PRIMARY KEY CLUSTERED
(
    [idTag] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

/***** Table [CPC_PodcastTag] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].
[FK_CPC_PodcastTag_CPC_Podcast]') AND parent_object_id = OBJECT_ID(N'[dbo].
[CPC_PodcastTag]'))
ALTER TABLE [dbo].[CPC_PodcastTag] DROP CONSTRAINT
[FK_CPC_PodcastTag_CPC_Podcast]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].
[FK_CPC_PodcastTag_CPC_Tag]') AND parent_object_id = OBJECT_ID(N'[dbo].
[CPC_PodcastTag]'))
ALTER TABLE [dbo].[CPC_PodcastTag] DROP CONSTRAINT
[FK_CPC_PodcastTag_CPC_Tag]
GO

USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].
[CPC_PodcastTag]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_PodcastTag]
GO

USE [CPC]
GO

```

```

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[CPC_PodcastTag](
    [idPodcast] [int] NOT NULL,
    [idTag] [int] NOT NULL,
    [Value] [varchar](256) NOT NULL,
    CONSTRAINT [PK_CPC_PodcastTag] PRIMARY KEY CLUSTERED
(
    [idPodcast] ASC,
    [idTag] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[CPC_PodcastTag] WITH CHECK ADD CONSTRAINT
[FK_CPC_PodcastTag_CPC_Podcast] FOREIGN KEY([idPodcast])
REFERENCES [dbo].[CPC_Podcast] ([idPodcast])
GO

ALTER TABLE [dbo].[CPC_PodcastTag] CHECK CONSTRAINT
[FK_CPC_PodcastTag_CPC_Podcast]
GO

ALTER TABLE [dbo].[CPC_PodcastTag] WITH CHECK ADD CONSTRAINT
[FK_CPC_PodcastTag_CPC_Tag] FOREIGN KEY([idTag])
REFERENCES [dbo].[CPC_Tag] ([idTag])
GO

ALTER TABLE [dbo].[CPC_PodcastTag] CHECK CONSTRAINT
[FK_CPC_PodcastTag_CPC_Tag]
GO

/***** Table [CPC_User] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].
[CPC_User]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_User]
GO

USE [CPC]
GO

```

```

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[CPC_User](
    [idUser] [int] NOT NULL,
    [Login] [varchar](50) NOT NULL,
    [Password] [varchar](50) NOT NULL,
    CONSTRAINT [PK_CPC_User] PRIMARY KEY CLUSTERED
(
    [idUser] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

/***** Table [dbo].[CPC_View] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].
[CPC_View]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_View]
GO

USE [CPC]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[CPC_View](
    [idView] [int] NOT NULL,
    [Name] [varchar](50) NOT NULL,
    [Description] [varchar](256) NULL,
    CONSTRAINT [PK_CPC_View] PRIMARY KEY CLUSTERED
(
    [idView] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

/***** Table [CPC_UserTag] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_CPC_UserTag_CPC_Tag]') AND parent_object_id = OBJECT_ID(N'[dbo].[CPC_UserTag]'))
ALTER TABLE [dbo].[CPC_UserTag] DROP CONSTRAINT [FK_CPC_UserTag_CPC_Tag]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_CPC_UserTag_CPC_User]') AND parent_object_id = OBJECT_ID(N'[dbo].[CPC_UserTag]'))
ALTER TABLE [dbo].[CPC_UserTag] DROP CONSTRAINT [FK_CPC_UserTag_CPC_User]
GO

USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CPC_UserTag]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_UserTag]
GO

USE [CPC]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[CPC_UserTag](
    [idUser] [int] NOT NULL,
    [idTag] [int] NOT NULL,
    [Credential] [int] NOT NULL,
    CONSTRAINT [PK_CPC_UserTag] PRIMARY KEY CLUSTERED
(
    [idUser] ASC,
    [idTag] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[CPC_UserTag] WITH CHECK ADD CONSTRAINT [FK_CPC_UserTag_CPC_Tag] FOREIGN KEY([idUser])
REFERENCES [dbo].[CPC_Tag] ([idTag])

```



```

GO

ALTER TABLE [dbo].[CPC_UserTag] CHECK CONSTRAINT [FK_CPC_UserTag_CPC_Tag]
GO

ALTER TABLE [dbo].[CPC_UserTag] WITH CHECK ADD CONSTRAINT
[FK_CPC_UserTag_CPC_User] FOREIGN KEY([idUser])
REFERENCES [dbo].[CPC_User] ([idUser])
GO

ALTER TABLE [dbo].[CPC_UserTag] CHECK CONSTRAINT [FK_CPC_UserTag_CPC_User]
GO

/***** Table [CPC_UserViewTag] *****/
USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_CPC_UserViewTag_CPC_Tag]') AND parent_object_id = OBJECT_ID(N'[dbo].[CPC_UserViewTag]'))
ALTER TABLE [dbo].[CPC_UserViewTag] DROP CONSTRAINT
[FK_CPC_UserViewTag_CPC_Tag]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_CPC_UserViewTag_CPC_User]') AND parent_object_id = OBJECT_ID(N'[dbo].[CPC_UserViewTag]'))
ALTER TABLE [dbo].[CPC_UserViewTag] DROP CONSTRAINT
[FK_CPC_UserViewTag_CPC_User]
GO

IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_CPC_UserViewTag_CPC_View]') AND parent_object_id = OBJECT_ID(N'[dbo].[CPC_UserViewTag]'))
ALTER TABLE [dbo].[CPC_UserViewTag] DROP CONSTRAINT
[FK_CPC_UserViewTag_CPC_View]
GO

USE [CPC]
GO

IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CPC_UserViewTag]') AND type in (N'U'))
DROP TABLE [dbo].[CPC_UserViewTag]
GO

USE [CPC]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[CPC_UserViewTag](

```

```

        [idUser] [int] NOT NULL,
        [idView] [int] NOT NULL,
        [idTag] [int] NOT NULL,
    CONSTRAINT [PK_CPC_UserViewTag] PRIMARY KEY CLUSTERED
    (
        [idUser] ASC,
        [idView] ASC,
        [idTag] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

ALTER TABLE [dbo].[CPC_UserViewTag] WITH CHECK ADD CONSTRAINT
[FK_CPC_UserViewTag_CPC_Tag] FOREIGN KEY([idUser])
REFERENCES [dbo].[CPC_Tag] ([idTag])
GO

ALTER TABLE [dbo].[CPC_UserViewTag] CHECK CONSTRAINT
[FK_CPC_UserViewTag_CPC_Tag]
GO

ALTER TABLE [dbo].[CPC_UserViewTag] WITH CHECK ADD CONSTRAINT
[FK_CPC_UserViewTag_CPC_User] FOREIGN KEY([idUser])
REFERENCES [dbo].[CPC_User] ([idUser])
GO

ALTER TABLE [dbo].[CPC_UserViewTag] CHECK CONSTRAINT
[FK_CPC_UserViewTag_CPC_User]
GO

ALTER TABLE [dbo].[CPC_UserViewTag] WITH CHECK ADD CONSTRAINT
[FK_CPC_UserViewTag_CPC_View] FOREIGN KEY([idView])
REFERENCES [dbo].[CPC_View] ([idView])
GO

ALTER TABLE [dbo].[CPC_UserViewTag] CHECK CONSTRAINT
[FK_CPC_UserViewTag_CPC_View]
GO

```

## Conclusions

Considero que s'ha afrontat un treball molt ambiciós i s'han assolit els objectius inicialment establerts, si bé es podria dir que les Interfaces d'usuari de les aplicacions són millorables, en el sentit de fer-les més intuïtives per a l'usuari.

Aquest treball m'ha servit per aprofundir en l'estudi de les tecnologies de DirectShow, C#.NET i, en especial, de Framework 3.5 de Microsoft. Alhora m'ha permès ampliar els meus coneixements sobre bases de dades i particularment del sistema SQL Server 2k8.

Després de la realització d'aquest treball considero que l'anàlisi i el disseny de l'aplicació constitueixen els fonaments de la mateixa, i com a tals han de ser sòlids i ben formats, el que suposa una elevada inversió de temps i esforç sense els quals l'aplicació no seria viable.

També cal destacar que la meua experiència prèvia ha estat molt important per a resoldre les dificultats tècniques que han sorgit per assolir el TFC.

Finalment, m'agradaria afegir que en tot aquest procés he trobat a faltar un aspecte que valoro molt del meu dia a dia professional: el treball en equip.

## Línies de futur

Aquest projecte podria servir per ser evolucionat posteriorment en una plataforma més completa, on es poguessin tractar les bases de dades, els repositoris i altres elements pensats en clau de N. Per exemple, tenir N servidors de BBDD (Cloud Networking)

Aquesta evolució crec que es podria fer amb estudiants de la UOC amb un coordinador de tasques.

Finalment m'agradaria poder penjar aquest projecte a Internet amb una llicència GPL.

## Glossari

**Administrador:** persona encarregada del manteniment de l'aplicació.

**API:** Interfície de Programació d'Aplicacions (Application Programming Interface, API), és un conjunt de declaracions que defineix el contracte d'un component informàtic amb qui farà ús dels seus serveis. [Viquipèdia]

**Autenticació:** Procés de seguretat pel qual l'usuari ha d'introduir el seu codi d'usuari i la seva contrasenya i, un cop validades pel sistema, permet l'accés a l'aplicació.

**Codec:** Abreviatura de Compressor-Descompressor. Descriu una especificació desenvolupada en software, hardware o una combinació d'ambdues, capaç de transformar un arxiu a un fluxe de dades (stream) o un senyal. Els codecs poden codificar el fluxe o el senyal (sovint per a la transmissió, l'emmagatzematge o el xifrat) i recuperar-lo o desxifrar-lo de la mateixa manera per a la reproducció o la manipulació en un format més apropiat per a aquestes operacions.

**Codi d'usuari (Login):** Nom únic pel qual es coneix un usuari i que el permet identificar-se al sistema.

**Contrasenya:** Mot associat al codi d'usuari del qual només en són coneixedors l'usuari i el sistema i que permet identificar inequívocament un usuari.

**Cultiu:** Veure pàgina número 2.

**C#:** Llenguatge de programació orientat a objectes desenvolupat i estandaritzat per Microsoft com a part de la seva plataforma .NET, que després va ser aprovada com un estàndard per la ECMA i ISO.

**Framework:** Infraestructura de programari que, en la programació orientada a objectes, facilita la concepció de les aplicacions mitjançant la utilització de biblioteques de classes o generadors de programes. [TERMCAT]

**FTP:** (File Transfer Protocol) és un protocol de xarxa per a la transferència d'arxius entre sistemes connectats a una xarxa TCP, basat en l'arquitectura client-servidor

**HTML:** Llenguatge estàndard que s'utilitza per a etiquetar documents en format d'hipertext, per tal d'indicar a un navegador de quina manera ha de visualitzar un document a la pantalla d'un ordinador. [TERMCAT]

**IDE:** Integrated Development Enviroment. És un programa que es compon d'un conjunt d'eines destinades a la programació.

**Intranet:** Xarxa d'ordinadors d'àrea local privada que proporcionen eines d'Internet per tal de compartir diferents tipus de recursos.

**Podcast:** Veure pàgina número 2.

**SVN:** Subversion és un software de sistema de control de versions dissenyat específicament per a substituir al popular CVS.

**Projecte:** En el context d'aquest TFC, projecte de desenvolupament de software.

**Script:** Programa escrit per a un llenguatge interpretat.

**SGBD:** Sistema Gestor de Bases de Dades. Programari que s'encarrega de gestionar i manipular bases de dades (grans conjunts de dades estructurades) i executar sobre elles les operacions sol·licitades per múltiples clients.

**SQL:** Llenguatge de consulta, d'actualització i de gestió de bases de dades relacionals reconegut com a estàndard per l'Organització Internacional per a la Normalització (ISO).

**Microsoft SQL Server:** Sistema de gestió de bases de dades relacionales (SGBD) basat en el llenguatge Transact-SQL, i específicament en Sybase IQ

**Mòdul:** Part de l'aplicació amb una certa autonomia i característiques diferencials respecte de la resta de l'aplicació, especialment des del punt de vista de l'usuari.

**XML:** eXtensible Markup Language. És un llenguatge d'etiquetes que permet representar i intercanviar informació entre ordinadors o programes.

## Bibliografia

DirectX, DirectShow

<http://msdn.microsoft.com/en-us/directx/default.aspx>

FileZilla

<http://filezilla-project.org/>

Filtres d'encoding:

[http://www.vorbis.com/setup\\_windows/](http://www.vorbis.com/setup_windows/)

<http://www.illiminable.com/ogg/>

Genèrics

<http://ca.wikipedia.org/wiki/Portada>

[http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page)

<http://www.termcat.cat/>

IIS

<http://www.iis.net/downloads/default.aspx?tabid=3>

MSDN

<http://msdn.microsoft.com/es-es/default.aspx>

SQL Server Express

<http://www.microsoft.com/express/>

Subversion SVN

<http://subversion.tigris.org/>

VS Express

<http://www.microsoft.com/express/>

VirtualBox

<http://www.virtualbox.org/>