

# MINIDLNA\_TWEAKER

---

## APLICACIÓ PER A CONFIGURAR I PREVISUALITZAR EL SERVIDOR MiniDLNA PER A LINUX

Treball Fi de Carrera

Jordi Martínez Lloveras

Enginyeria Tècnica en Telecomunicació, especialitat Telemàtica

Universitat Oberta de Catalunya (UOC)

Tutor acadèmic: Manel Llopart Vidal

Juny 2013

## **MiniDLNA\_tweaker, personalitza el teu servidor MiniDLNA**

El poder compartir una mateixa biblioteca multimèdia disponible per tota la llar i no haver d'estar movent la informació amb dispositius extraïbles o de gravació és un problema que està sent enfocat des de diferents àmbits de la programació i de l'electrònica. Un d'ells és la possibilitat d'instal·lar un servidor al ordinador d'escriptori de la llar per a poder compartir la informació que hi és a dins o que es pot anar descarregant d'Internet.

L'aplicació MiniDLNA per a GNU/Linux és un servidor lleuger que compleix els estàndards DLNA/UPnP configurable a través d'un simple arxiu de text, això la fa ideal per al propòsit esmentat de la implantació d'un servidor que ofereixi els continguts a tots els dispositius que compleixin els estàndards esmentats.

Aquest treball ve a emplenar el buit existent que té aquest programari en quant a una eina gràfica que serveixi per a poder controlar d'una manera senzilla i intuïtiva aquest servidor, que al ser part d'un programari més gran destinat a altres tasques, no disposa d'aquesta eina esmentada.

L'aplicació MiniDLNA\_tweaker desenvolupada en aquest treball emplena aquest buit tot permetent tres aspectes bàsics relacionat amb el servidor: Configurar els seus paràmetres, controlar el seu funcionament i poder escollir la informació multimèdia que s'ofereix d'una manera molt més intuïtiva que l'existent.

## **MiniDLNA\_tweaker, personaliza tu servidor MiniDLNA**

El poder compartir una misma biblioteca multimedia disponible para todo el hogar y no tener que mover la información en dispositivos extraíbles o de grabación es un problema que está siendo enfocado desde diferentes ámbitos de la programación y de la electrónica. Uno de ellos es la posibilidad de instalar un servidor en el ordenador de sobremesa del hogar para poder compartir la información que está dentro del o se descarga de internet.

La aplicación MiniDLNA para GNU/Linux es un servidor ligero que cumple con los estándares DLNA/UPnP configurable a través de un simple archivo de texto, esto la hace ideal para el propósito dicho de la implantación de un servidor que ofrezca contenidos multimedia a los dispositivos del hogar que cumplan estos estándares.

La aplicación MiniDLNA\_tweaker desarrollada en este trabajo llena este vacío permitiendo los tres aspectos básicos de control de este servidor: Configurar sus parámetros, controlar su funcionamiento y poder escoger la información multimedia ofrecida de una manera mucho más intuitiva que la existente.

## **MiniDLNA\_tweaker, a MiniDLNA server customizer**

The possibility to share the same multimedia library in every home device, and so to avoid the necessity of moving the information in removable or recorded devices is a problem that nowadays is being treated from different points of view of the programming and electronic areas. One of them is the possibility to install a server in the desktop computer in order to share the multimedia information inside or downloadable from Internet.

MiniDLNA application for GNU/Linux is a lightweight server that fits the DLNA/UPnP standards which configuration remains in just one text file. This makes the application perfect for the purpose to have a multimedia server that offers the information to every home device that applies these protocols.

The application MiniDLNA\_tweaker developed in this work fits this gap making possible to control the three main items of this server: To set up its parameters, control of the running up and down, and the possibility of choosing this information to share in a very much familiar way than the actual possibilities.

## Introducció

Els servidors DLNA són una de les eines ideals per a configurar una xarxa multimèdia a la llar. Una vegada configurat el servidor, permet a tots els dispositius que tenen aquesta implementació accedir a la base de dades d'arxius multimèdia i reproduir-los sense haver de fer cap configuració extra en el dispositiu de reproducció, només fa falta que el servidor i el dispositiu final de reproducció estiguin connectats a la mateixa xarxa local.

Per als sistemes operatius més populars, MS Windows, Mac OS i Linux, existeixen potents eines que posen a disposició de l'usuari la possibilitat de configurar un servidor multimèdia d'aquestes característiques. En el cas més conegut de MS Windows, des del reproductor de mitjans subministrat pel mateix sistema operatiu de forma gratuïta es pot configurar aquest mateix tipus de servidor.

Per a les distribucions GNU/Linux més contemporànies, existeix un servidor de baix consum i pes de memòria anomenat MiniDLNA, que té l'ànim de ser totalment compatible amb els dispositius DLNA / UPnP. Les seves més valuoses característiques per aquest treball és que és un servidor lleuger, de codi obert (*open source*) i totalment configurable només amb la modificació d'uns paràmetres en el seu fitxer de configuració d'arrencada. Està desenvolupat per un empleat de l'empresa NETGEAR per al seu software ReadyNAS, una eina per a configurar emmagatzemaments tipus NAS.

La mancança que ve a intentar solucionar l'aplicació en la que es basa aquest treball és la falta d'una interfície gràfica senzilla per a poder controlar aquest servidor. Una eina que s'obri, modifiqui i es tanqui sense residir permanentment en la memòria i que sigui independent del programari del servidor. L'aplicació està pensada i desenvolupada per a poder controlar el servidor i les seves tasques més bàsiques, per a poder implementar un nou fitxer de configuració o per a poder modificar un d'existent, per a poder visualitzar la llista d'arxius multimèdia de la base de dades que crea el servidor i poder treure de la mateixa els arxius que no interessin que estiguin a la disposició del dispositiu de reproducció i per últim, poder fer una petita visualització o reproducció, depenent del contingut, dels fitxers multimèdia.

Després d'explorar les diverses possibilitat per tal de crear una aplicació en entorn Linux que fos capaç de fer totes aquestes tasques el més compacta i integra possible, l'entorn de desenvolupament escollit és l'entorn del *Qt Project*, un entorn de

llibries desenvolupades en el llenguatge C++, que es capaç de gestionar Interfícies gràfiques d'usuari, connectar i gestionar bases de dades SQL, gestionar fitxers i carpetes sense sortir del propi entorn, executar comandes de Linux com si fossin processos que s'obren i es tanquen i implementar un petit reproductor de mitjans totalment personalitzable gràcies a l'entorn multimèdia del Qt *Phonon*. També ofereix in Entorn de desenvolupament integrat, el *Qt Creator*, que fa que les tasques de desenvolupament d'aplicacions gràfiques sigui una tasca menys complicada que si s'hagués de baixar més nivells en la compilació o la programació d'aplicacions.

El desenvolupament i la implementació ha estat marcat per l'estudi de tot el que és l'entorn de llibries Qt, havent tingut l'avantatge de poder trobar en un mateix lloc tot el material d'estudi necessari per a la implementació de l'aplicació, no així de tots els processos anteriors, paral·lels i posteriors del desenvolupament d'una aplicació en entorn gràfic, que si be alguns dels treballs necessaris ja eren coneguts, altres has suposat un aprenentatge més profund en altres matèries relacionades.

## Agraïments

A la meva companya la Sandra, per haver-me aguantat tots aquests anys i per recolzar-me quan vaig canviar d'orientació d'estudis

## Índex de continguts

MiniDLNA_tweaker, personalitza el teu servidor MiniDLNA .....	1
MiniDLNA_tweaker, personaliza tu servidor MiniDLNA.....	1
MiniDLNA_tweaker, a MiniDLNA server customizer .....	2
Introducció .....	3
Agraïments .....	5
Índex de taules .....	9
Glossari .....	10
Definició del projecte.....	11
Títol.....	11
Resum.....	11
Paraules clau .....	12
Objectius principals .....	12
Beneficis .....	13
Entregables.....	13
Motivació.....	13
Estat de l'art.....	15
Divisió de tasques per a investigar .....	15
Disciplines d'estudi.....	18
L'entorn d'interfície d'usuari i els diferents toolkits emprats i IDEs .....	18
Interfícies simples de diàlegs i interacció .....	18
Llibreries addicionals a llenguatges estàndard (GUI Toolkits):.....	19
Integració amb comandes Linux, ordres SQL i multimèdia .....	29
Integració de les bases de dades Sqlite.....	29
Integració de la informació multimèdia.....	30
Disseny.....	32
Introducció .....	32
Aparença.....	33
Objectiu 1: Tractament del fitxer de configuració: .....	33
Objectiu 2: Administració del servidor .....	35
Objectius 3 i 4: Base de dades i reproductor:.....	35
Desenvolupament en pseudo-codi i diagrames.....	38
Implementació .....	46
Desenvolupament dels objectius .....	53
Tractament del fitxer de configuració .....	53

Administració del servidor:.....	58
Base de dades.....	61
Objectiu addicional 4 – Visualitzar i reproduir els arxius de la biblioteca .....	62
Traducció.....	65
Documentació.....	69
Compilació i execució .....	69
Execució: .....	69
Compilació .....	70
Demostració .....	71
Conclusions i línies de futur .....	73
Fitxers entregats: .....	75
Bibliografia – Recursos emprats .....	76



Il·lustració 1 - Tasques a investigar .....	17
Il·lustració 2 - Exemple utilització del Zenity .....	18
Il·lustració 3 - Exemple utilització del Kdialog (entrada) .....	19
Il·lustració 4 - Exemple utilització del Kdialog (sortida).....	19
Il·lustració 5 - Captura de pantalla de la eina Fluid.....	20
Il·lustració 6 - Exemple d'aplicació FLTK (paquet fltk1.3-games a Linux Wheezy) .....	21
Il·lustració 7 - Captura d'aplicació feta amb FOX .....	22
Il·lustració 8 - Aplicació en wxwidgets, FileZilla (funcionant a MS Windows 7).....	23
Il·lustració 9 - Captura de pantalla de l'eina de disseny Glade (GTK).....	24
Il·lustració 10 - Captura de l'eina de desenvolupament Anjuta (GTK).....	25
Il·lustració 11 - Clementine, un reproductor d'àudio fet amb Qt.....	26
Il·lustració 12 - Qt Creator, un IDE complet per a crear aplicacions en QT.....	27
Il·lustració 13 - Captura de la taula SQL del MiniDLNA "details" .....	31
Il·lustració 14 - Esbós de la pestanya de configuració.....	34
Il·lustració 15 - Esbós de la pestanya de configuració amb botons i menú.....	35
Il·lustració 16 - Esbós de la pestanya de la base de dades i el reproductor .....	36
Il·lustració 17 - Esbós del menú .....	36
Il·lustració 18 - Flux d'interacció de l'aplicació.....	37
Il·lustració 19 - Diagrama inici aplicació .....	38
Il·lustració 20 - Diagrama opció aturar servidor .....	39
Il·lustració 21 - Diagrama opció arrencar servidor .....	40
Il·lustració 22 Diagrama opció aplicar canvis .....	41
Il·lustració 23 - Diagrama Funció per a comprovar paràmetres .....	43
Il·lustració 24 - Diagrama Obrir arxiu.....	44
Il·lustració 25 - Visualització Base de Dades.....	45
Il·lustració 26 - Entorn de desenvolupament Qt Creator en el mode gràfic.....	47
Il·lustració 27 - Entorn de desenvolupament Qt Creator en mode programació.....	48
Il·lustració 28 - Codi XML generat pel disseny gràfic del Qt Creator.....	49
Il·lustració 29 - Plasmació de l'esbós gràfic al Qt Creator.....	50
Il·lustració 30 - Menú contextual ->Go to slot... .....	51
Il·lustració 31 - Senyals definibles .....	51
Il·lustració 32 - Definició del codi de la senyal.....	52
Il·lustració 33 - Mostra cartell error permisos .....	55
Il·lustració 34 - Captura pantalla aplicació en mode reproductor .....	64
Il·lustració 35 - Captura pantalla Qt Linguist, eina de traducció del Qt .....	66
Il·lustració 36 - Captura pantalla aplicació idioma català .....	67

Il·lustració 37 - Captura pantalla aplicació idioma castellà .....	68
Il·lustració 38 - Captura pantalla aplicació idioma anglès .....	68

## Índex de taules

Taula 1 - Descripció Ubuntu app developer website .....	10
Taula 2 – Descripció web documentació MiniDLNA.....	11
Taula 3 - Descripció web GTK+ 3 Reference manual .....	12
Taula 4 – Descripció Llibre MySQL:Building user interfaces .....	12
Taula 5 – Descripció The MPEG standards .....	13
Taula 6 – Comparativa Lliberies par a construir GUI .....	26

## Glossari

**DLNA** – Sigles de *Digital Living Network Alliance*, organització de fabricants per a estandarditzar els protocols de xarxa entre dispositius. Certifica dispositius amb un estàndard amb el mateix nom.

**IDE** – Sigles de *Integrated Development Environment* (Entorn de desenvolupament integrat), eina informàtica per al desenvolupament de programari comoda, ràpida i integrada.

**Home Network o Xarxa Local o Xarxa de la Llar:** Xarxa que integren tots els aparells que estan a una llar sota connexió normalment d'un mateix router

# Definició del projecte

## Títol

MiniDLNA\_tweaker - Aplicació per a configurar i previsualitzar el servidor MiniDLNA per a Linux.

## Resum

- El servidor MiniDlna és un servidor que pretén complir els estàndards DLNA (Digital Living Network Alliance, en català *Aliança per l'estil de vida digital en xarxa*) i UPnP-AV (Universal Plug n' Play, en català *Connectar i reproduir universal*). Interacciona amb els clients que certifiquen el compliment d'aquests dos serveis per a oferir una connexió a una base de dades de documents multimèdia (àudio, vídeo i imatges), i a poder des de el client accedir a aquests continguts per a ser reproduïts. L'avantatge d'aquest sistema es que des de el sistema client (certificat amb els estàndards anomenats abans), sense configuració, només amb la connexió d'aquest equip a la mateixa xarxa que el servidor, ja es te accés a tota aquesta biblioteca de continguts.
- En l'entorn Windows, des de les últimes versions, es pot configurar des de els mitjans que ofereix el mateix sistema operatiu (Reproductor de mitjans, tauler de control i explorador d'arxius) la biblioteca de mitjans audiovisuals a la qual els clients tenen accés. A Linux existeix el programari Minidlna, un servidor que s'instal·la com a servei del sistema operatiu. Aquest servidor es configura a través d'un fitxer de text (minidlna.conf) al qual es poden definir la sèrie de paràmetres que configuren el servei, tant de configuració de xarxa com dels directoris que formaran part de la biblioteca de mitjans audiovisuals.
- Queda clar que per molt que a l'ajuda i al mateix fitxer de configuració les diferents opcions de configuració queden explicades, no es la millor opció de definir la configuració d'un sistema que pel seu objectiu, que es disposar d'accés "familiar" d'uns continguts emmagatzemats al servidor, normalment l'ordinador familiar, hauria de ser senzill i adaptable a un públic no expert en les vicissituds de configurar un servidor mitjançant un fitxer de text.

- La idea d'aquest projecte o treball és al seu acabament de disposar d'una eina gràfica de configuració d'aquest servidor que permeti d'una manera senzilla i objectiva definir els paràmetres de configuració del fitxer de text i poder veure d'una manera clara els components de la biblioteca audiovisual, oferint alguna eina de visualització dels mateixos continguts al servidor.
- Els coneixements necessaris per a fer el projecte seran la programació orientada a objectes, la programació de base de dades, la programació i la gestió de serveis en Linux, el coneixement de gestió de projectes, la interacció persona-ordinador, tots aquests són adquirits de la carrera.
- Els coneixements necessaris addicionals per a desenvolupar el projecte seran: la programació d'interfícies d'usuari, aspectes concrets de l'administració de sistemes Linux, i com ajuntar tot lo après a la carrera en quant a programació en una aplicació que ho faci servir tot.

## Paraules clau

- Linux, servidor, multimèdia, media-center, DLNA, gui, database

## Objectius principals

- La consecució d'una interfície d'usuari per a configurar els paràmetres del servidor DLNA de Linux (minidlna), mitjançant un formulari clar i intuïtiu
- Poder inicialitzar, parar o restablir el servei
- L'extracció de la informació de la base de dades del servidor i la conversió a un format que permeti extreure informació dels arxius multimèdia
- Objectius addicionals
- Tractament de la informació de la base de dades: Ordenar, canviar de nom...
- Reproducció dels arxius de la biblioteca
- Estètica de l'aplicació millorada: icones...

## Beneficis

- Els beneficis d'aquest treball fi de carrera personalment seran l'ampliació de conceptes relacionats amb la programació d'interfícies d'usuari, el tractament dels fluxes de la xarxa, la configuració de xarxes i el tractament de les bases de dades. Això ho podré aplicar professionalment ja que en el present i futur, els continguts multimèdia que es poden visualitzar ja sigui al televisor o ja sigui en qualsevol altra visualitzador, ja no venen necessàriament d'una senyal aèria fixa, sinó que el tractament d'aquestes senyals, l'emmagatzemament de continguts, la visualització dels mateixos en qualsevol dels punts connectats a la xarxa local, serà com evolucionarà l'oci multimèdia a casa.
- Els beneficis intencionats per a la comunitat serà de disposar d'una eina de configuració senzilla actualment inexistent per al servidor miniDLNA.

## Entregables

- Els entregables del treball Fi de Carrera proposat seran:
- La Memòria del treball
- Un programa, en binari i codi font, que faci el proposat en aquest treball
- Un possible vídeo (real o screencast) de com funciona l'aplicació

## Motivació

- La motivació d'aquest treball és múltiple. Podria començar per les motivacions personals, sempre he estat admirador des de fa molt de temps de les distribucions Linux i com es pot fer un sistema operatiu complet basat en un sistema cooperatiu en el qual cadascú aporta en mida de les seves possibilitats. Crec que a més a més, la seva configuració de seguretat i la seva política d'usuari fa que sigui una opció idònia per a l'entorn familiar, allunyant-lo del malware i dels virus que inunden l'entorn Windows.
- Donada la política d'actualització de distribucions de Linux, m'he vist forçat a instal·lar el servidor en varies ocasions, i al no estar fixa en una distribució, m'he afrontat amb diversos problemes: Configuració de xarxa, de permisos d'usuari en els directoris de la biblioteca, d'establiment del servei, etc. Ha donat

la casualitat que al anar avançant en la carrera, he pogut comprendre i solucionar millor aquests problemes que m'anaven sortint.

- Sempre he trobat interessant la programació d'interfícies d'usuari i m'agradaria aprofundir més en aquesta matèria ja que ala carrera no s'entra en aquest aspecte, sinó en aspectes més interns del sistema operatiu, i trobo que es buit que vull emplenar.
- Actualment treballa en el disseny i desenvolupament de televisors i productes associats a l'àrea mecànica, i crec que és un treball fi de carrera que em pot anar molt be per a la projecció professional, doncs em prenc la part de la programació d'interfícies d'usuari com a un oportunitat en poder en un futur treballar en l'automatització de processos administratius, doncs encara s'emplena bastant quantitat de paper que no seria necessari repetir cada vegada. I en un futur m'agradaria canviar d'àmbit professional, i pel que he vist que estan fent els meus companys de software, la reproducció de continguts en xarxa, la interconnexió d'aparells domèstics, i el tenir dintre del televisor un petit sistema operatiu el qual et permeti tota aquesta interrelació, connexió i accés a contingut per a reproduir o l'accés a xarxes 2.0, tot això d'una manera més fluida i natural que l'actual i a l'abast de totes les cases, no només les d'alt poder adquisitiu, serà la pròxima revolució que veurem a dintre de les nostres cases en un futur proper.

# Estat de l'art

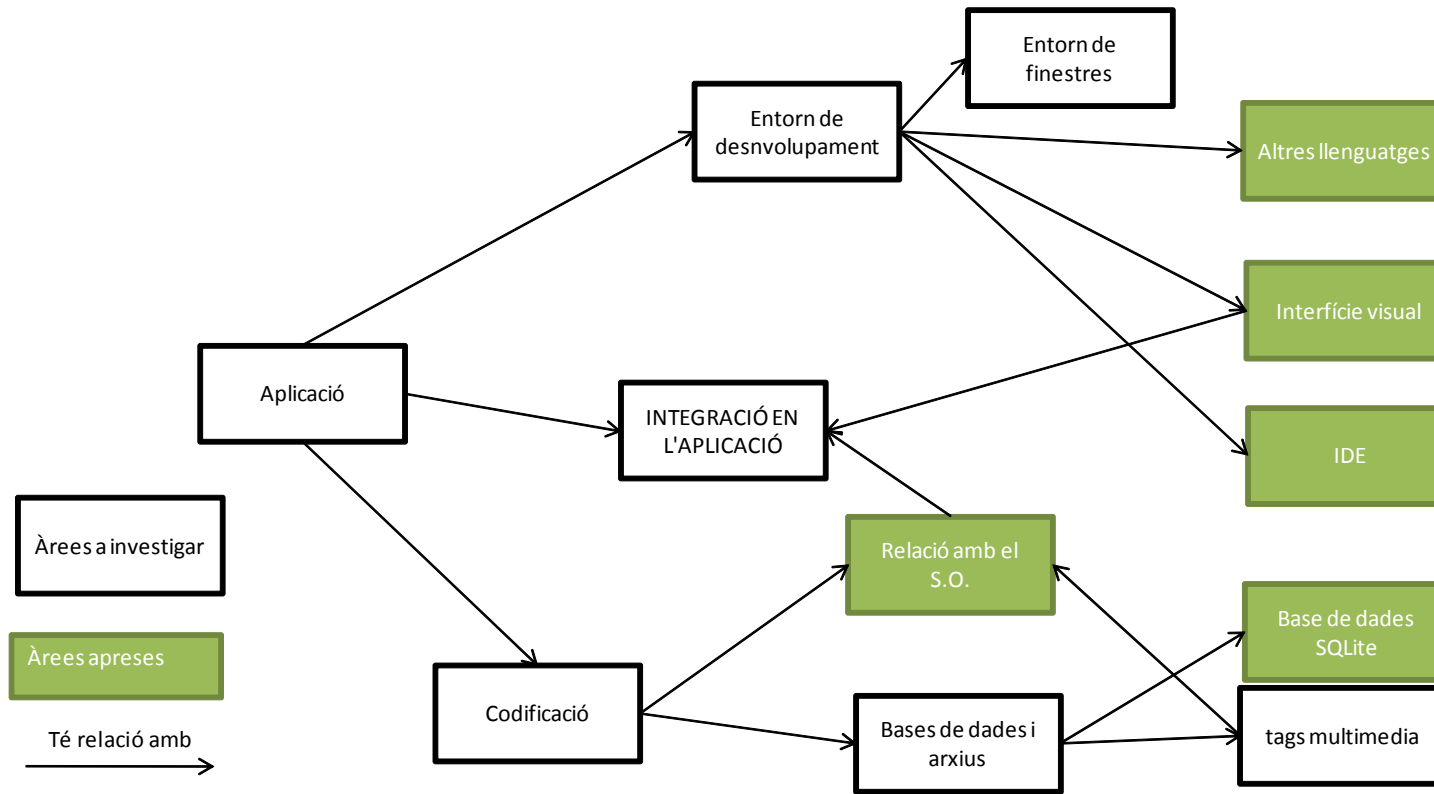
## Divisió de tasques per a investigar

- L'objectiu del treball, fer una interfície gràfica per a configurar el servidor MiniDLNA (<http://sourceforge.net/projects/minidlna/>) a Linux, fa que el treball d'investigació de l'estat de l'art es pugui dividir en diferents tasques al no marcar una disciplina general, sinó que serà un treball compartimentat en el que les diferents etapes d'implementació dependran de diferents decisions que s'hagin de prendre en acabar l'estat de l'art i veure la forma més factible i més interessant, que pot ser no sigui la més senzilla de totes per a les possibilitats d'aprenentatge, per exemple, que es poden derivar d'una altra.
- Les diferents divisions que es poden marcar per a la investigació de l'estat de l'art i de la implementació del projecte o aplicació són:
  - Entorn de desenvolupament de l'aplicació
    - Dintre d'aquesta primera decisió dependran altres, com la interfície de desenvolupament més adequada, si s'han d'aprendre algun llenguatge auxiliar per a desenvolupar l'aplicació, la familiarització amb el conjunt de llibreries de desenvolupament de la interfície, la possibilitat d'inserir diversos llenguatges dintre de l'aplicació.
    - Interfícies de desenvolupament (IDE)
  - Codificació
    - Aquest apartat, pel que hem pogut anar investigant en el projecte tindrà tres àrees ben diferenciades:



- Programació de la interfície en si (visual, finestra, entrada de text, botons, etc, que estarà completament relacionada amb la part abans descrita i la seva dificultat estarà lligada a l'entorn escollit.
  
- Programació de les funcions que ha de fer la interfície:
  - Llegir fitxer de text
  
  - Llegir entrades de text i comprovar-les
  
  - Substitució d'opcions de configuració per les entrades
  
  - Activació, parada i restabliment d'un servei
  
- Relació amb la base de dades i arxius multimèdia:
  - Llegir base de dades, modificar, ordenar, modificar, visualitzar sortides i formatar-les.
  
  - Llegir informació d'arxius multimèdia i poder-la fer servir per als propòsits de la informació a la base de dades.

En forma gràfica



II-lustració 1 - Tasques a investigar

## Disciplines d'estudi

Les interfícies gràfiques d'usuari (GUI) i els diferents llenguatges emprats i IDEs (*Integrated development environment, Entorn de desenvolupament integrat*)

La integració de diferents llenguatges en una aplicació: Linux script, el tractament de les bases de dades SQLite i les etiquetes (*tags*) multimèdia

## L'entorn d'interfície d'usuari i els diferents toolkits emprats i IDEs

En entorn Linux, per la seva naturalesa de codi obert i lliure, existeixen moltes possibilitats per a desenvolupar aplicacions amb entorn gràfic.

De fet, en realitat, l'entorn no és l'essencial per a desenvolupar aplicacions gràfiques, el que és essencial és que totes les llibreries estiguin disponibles per a l'entorn de sistema operatiu i que existeixi un compilador per aquest sistema.

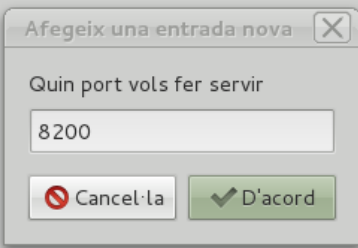
Es pot anar augmentant de complexitat poc a poc i així es pot trobar:

## Interfícies simples de diàlegs i interacció

- Zenity (<https://help.gnome.org/users/zenity/stable/index.html.es>) :

*Zenity, Zen dialogs*, és una reescriptura de *gdialogs*, inclòs a l'entorn d'escriptori, que serveix per donar "vida" als scripts *shell* que es desenvolupin en Linux. Disposa d'una sèrie d'entrades i sortides estàndard que es poden fer servir per interactuar amb el que executa el programa que poden servir, per exemple, per a escollir una opció entre moltes en un formulari, per a escollir una data de calendari, etc. És basa en les llibreries GTK i GLADE (explicitats més endavant), i és molt senzill d'utilitzar, un exemple per aconseguir una resposta per part de l'usuari, veiem que la resposta ve per la sortida estàndard.

```
jordi@kasa:~$ szAnswer=$(zenity --entry --text "Quin port vols fer servir" --entry-text "8200"); echo $szAnswer
8200
jordi@kasa:~$
```



### Il·lustració 2 - Exemple utilització del Zenity

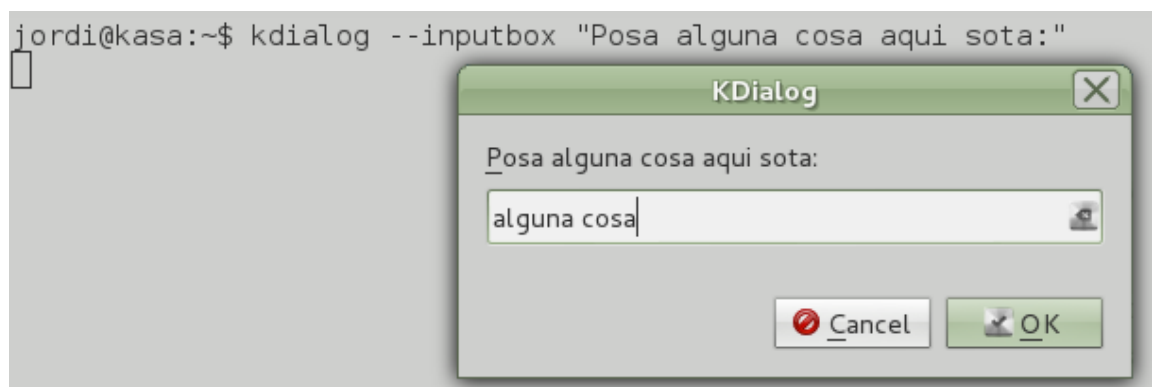
Evidentment, no es tracta d'un sistema complet d'interfície gràfica d'usuari, però per a

simples scripts o per a combinar-lo amb aplicacions GTK, és un sistema senzill i ràpid d'interactuar. Els seus requeriments per a poder programar són molt baixos i senzills, ja que només s'ha d'adaptar als requeriments del programador, si és factible, ja que les opcions són limitades.

- Kdialog

[http://techbase.kde.org/index.php?title=Development/Tutorials/Shell Scripting with KDE Dialogs#Introduction and Scope](http://techbase.kde.org/index.php?title=Development/Tutorials/Shell_Scripting_with_KDE_Dialogs#Introduction_and_Scope)

El mateix que *Zenity* és per a l'entorn *GTK/GLADE/Gnome*, *kdialog* ho és per a l'entorn *qt/KDE* (explicitat més endavant). L'ús és molt semblant:



II-Il·lustració 3 - Exemple utilització del Kdialog (entrada)

I la resposta per la sortida estàndard:

```
jordi@kasa:~$ kdialog --inputbox "Posa alguna cosa aqui sota:"
alguna cosa
jordi@kasa:~$
```

II-Il·lustració 4 - Exemple utilització del Kdialog (sortida)

Al principi de la programació de l'aplicació, en veure els requeriments, es pot decidir si es volen usar tots els *toolkits* (conjunt d'eines o conjunt de *widgets*) dels que disposen GTK i QT o ja amb aquesta forma senzilla es poden solucionar els requeriments de l'aplicació.

### Librerries addicionals a llenguatges estàndard (GUI Toolkits):

Aquest apartat està basat principalment en els *toolkits* més populars basats en C++ per la formació rebuda a la carrera fortament basada en C / C++, deixarem per tant les opcions basades en Java o altres llenguatges.

FLTK - Fast Light Toolkit (<http://www.fltk.org/index.php>)

Característiques:

- Basat en C++.
- Possibilitat d'inserir el codi com a subclasse C++.
- Molt lleuger
- Integració amb altres llenguatges de programació (*Python, Ruby*)
- Modular
- Llicència LGPL
- Multiplataforma (*Unix/Linux, Windows i MAC OSX*)
- Les eines d'interacció amb usuari: botons, formularis són molt completes
- Ofereix l'eina fluid per a dissenyar les interfícies d'usuari. (captura funcionant a MacOS)
- Contres:
  - Fluid és una eina molt limitada
  - Poc suport per a impressió, xarxa.
  - Poc estès.



II-Il·lustració 5 - Captura de pantalla de la eina Fluid

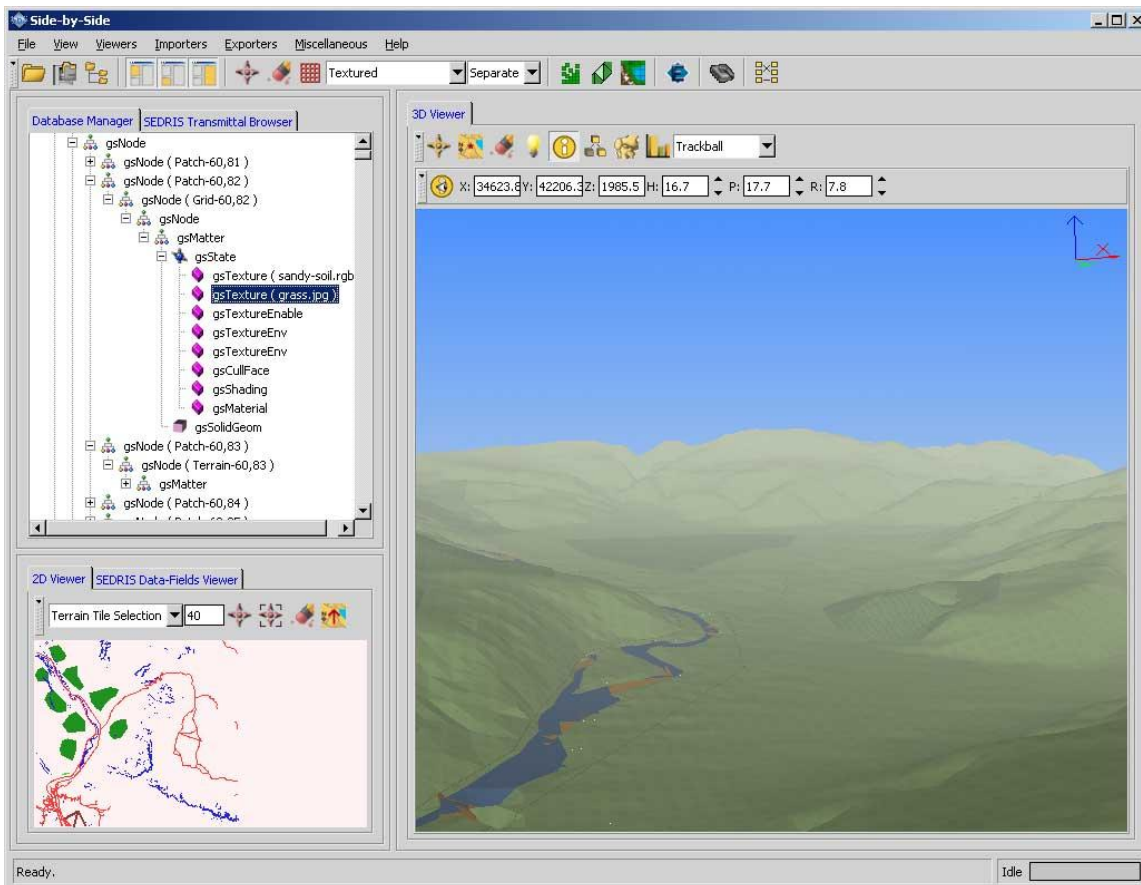


II-lustració 6 - Exemple d'aplicació FLTK (paquet fltk1.3-games a Linux Wheezy)

FOX toolkit (<http://www.fox-toolkit.org/>)

Característiques:

- Basat en C++.
- Molt lleuger
- Facilitat de programació
- Modular
- Llicència *GNU Lesser General Public License* (amb una excepció per a permetre el enllaç estàtic)
- Integració amb altres llenguatges (*Python, Ruby*)
- Independent de plataforma, dibuixa el seu propi GUI, per això pot estar poc integrat a l'entorn d'usuari. No suporta MAC OS.
- *Widgets* molt avançats
- Contres:
  - Molt basat en l'aspecte GUI, poc suport per a altres tasques
  - No te cap software per a dissenyar Interfícies d'usuari, tots els intents son discontinus.
  - Relativament poc estès.

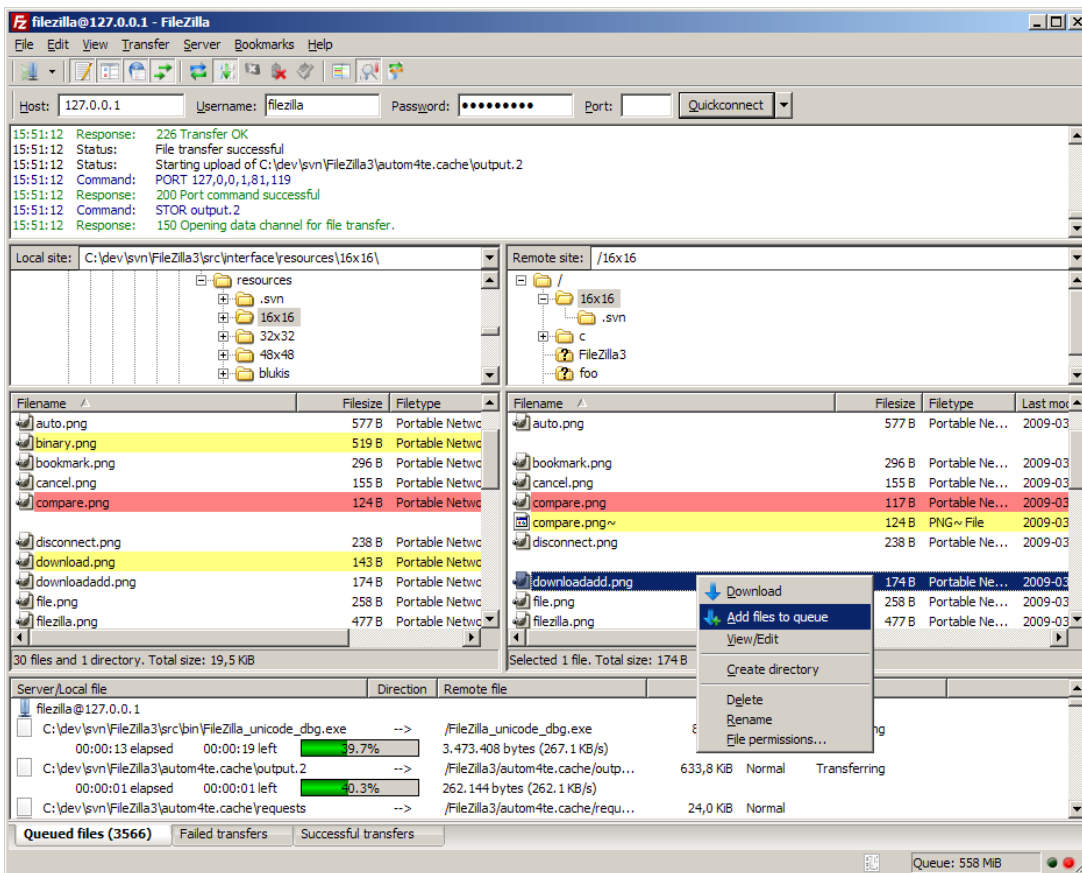


II-Il·lustració 7 - Captura d'aplicació feta amb FOX

WxWidgets – Cross Platform Gui Library (<http://www.wxwidgets.org/>)

- Natiu de C++, actualment disponible en tots els llenguatges més populars.
- Un dels més complets *toolkits*, possiblement el més complet fora dels que tenen un complet sistema d'escriptori darrere.
- Modular
- Llicència *GNU Lesser General Public License*, però més oberta, és lliure per a ús públic i comercial
- Allà on pot, usa el sistema de la plataforma on està, per això s'integra més en la plataforma però les seves aplicacions presenten un aspecte menys homogeni entre elles depenent de la plataforma.
- *Widgets* molt avançats, suporta, impressió, xarxa.
- Disposa d'eines avançades per a la creació de *GUIs*, algunes comercials.
- Bastant estès i amb aplicacions populars (ex. *Filezilla* (<http://filezilla-project.org/>))
- Contres:
  - Manca d'estabilitat comparat amb els *toolkits* més estesos

- Al fer ser molts *widgets* propis, i ser una plataforma molt estesa, pot tenir molts *bugs* (errades de codi sense solucionar) que depenen a quina plataforma esta funcionant.
- No suporta temes (nativament)



## II-Il·lustració 8 - Aplicació en wxwidgets, FileZilla (funcionant a MS Windows 7)

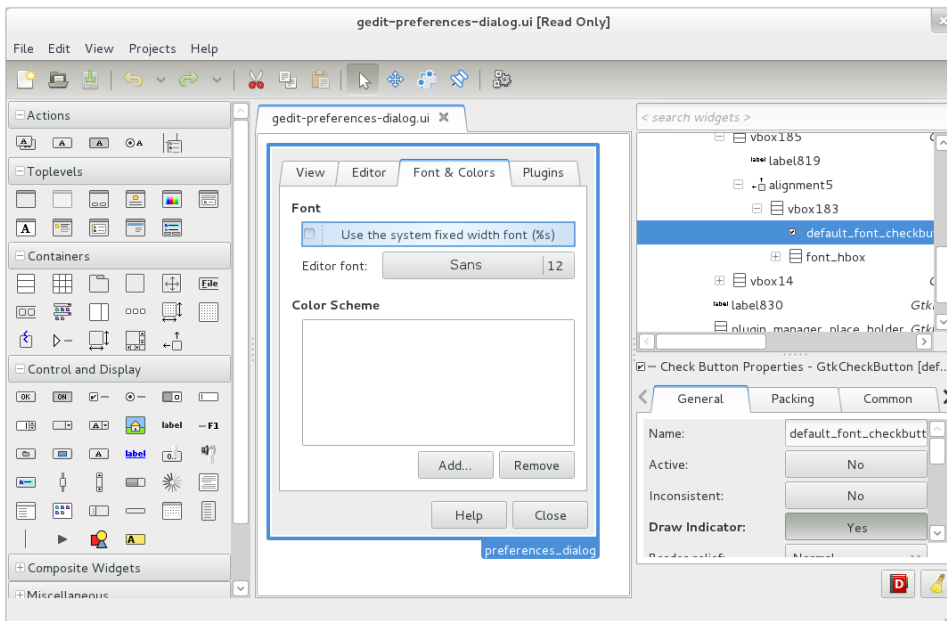
Biblioteques completes per a dissenyar Interfícies d'usuari, moltes vegades donant o modificant funcions del mateix llenguatge, amb un entorn d'escriptori complet al darrere:

GTK+ - The GTK+ Project (<http://www.gtk.org/>)

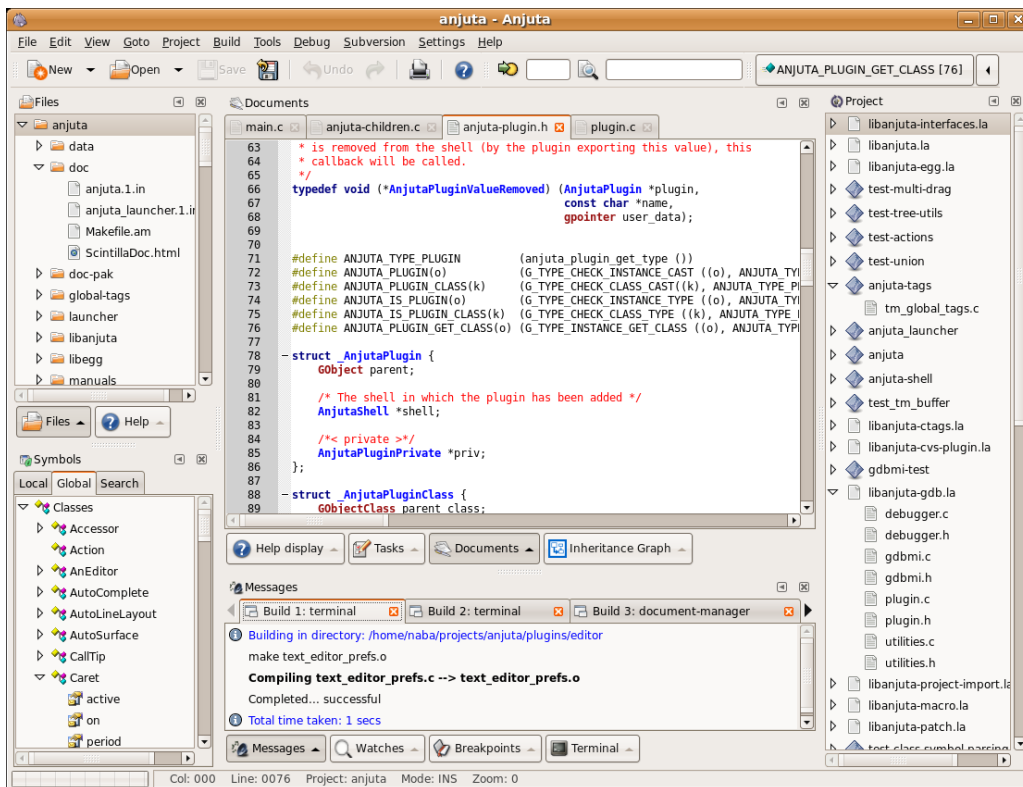
- Nadiu de C, disposa d'una extensió en C++, *gtkmm*, molt popular.
- Un dels més complets *toolkits*
- És la llibreria primària per a desenvolupar *Gnome*, *Xfce* i altres entorns d'escriptori de Linux
- Llicència GNU LGPL versió 2.1
- Fa servir els seus propis *widgets*, vol dir que no 's'integra a altres plataformes com Windows (nativament)
- *Widgets* molt avançats, suporta, impressió, xarxa, i temes
- Disposa d'eines avançades per a la creació de *GUIs*, algunes comercials.
- Molt estès amb aplicacions de tot tipus



- Fa servir C, amb la seva simplicitat
- Documentació estesa i abundant
- Al ser a sota d'un entorn com *Ubuntu Linux*, té el suport comercial de Canonical Ltd.. Això ha fet que la seva integració amb el llenguatge *Python* sigui perfecte.
- Disposa d'*Anjuta* i *Glade*, un entorn de programació i disseny natiu per a GTK+ molt potents
- Contres:
  - Més lent i menys lleuger que altres entorns menys potents
  - Fortament lligat a *Unix/Linux*
  - Fa servir C, amb la seva simplicitat



II-Il·lustració 9 - Captura de pantalla de l'eina de disseny Glade (GTK)

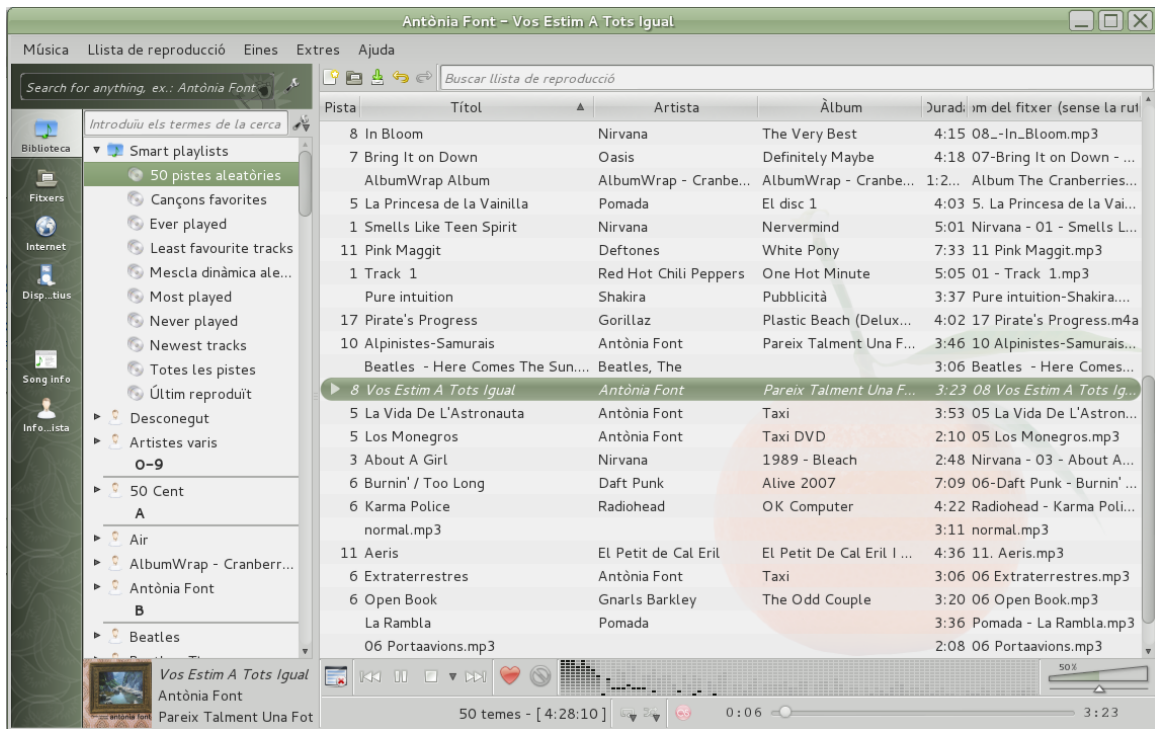


II-Il·lustració 10 - Captura de l'eina de desenvolupament Anjuta (GTK)

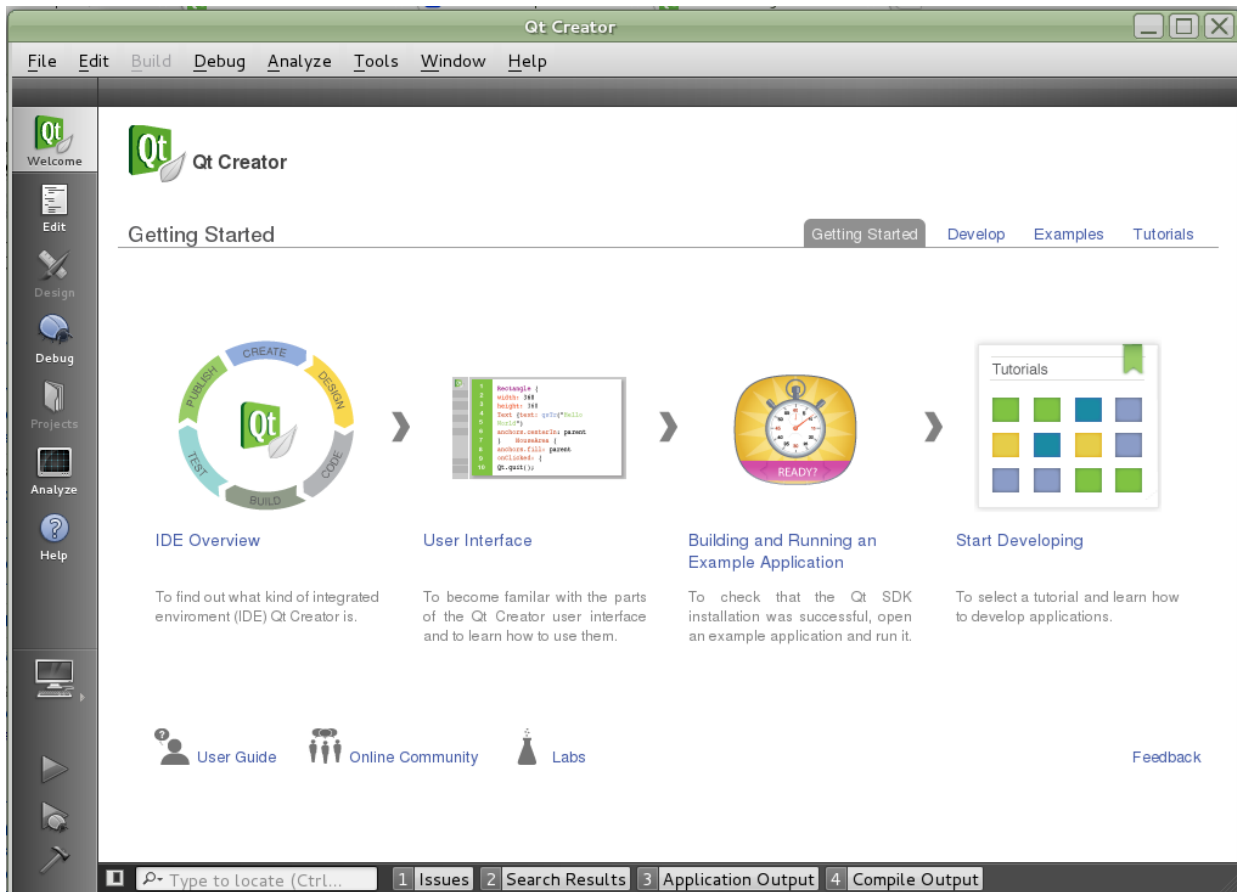
### Qt – The Qt project

- Nadiu de C++
- Possiblement el *toolkits* lliure més complet.
- És la llibreria primària per a desenvolupar KDE, un dels dos entorns d'escriptori de Linux més estesos
- A partir de la versió 4.5, lliure també per a fer aplicacions comercials
- Fa servir els seus propis *widgets*, vol dir que no s'integra a altres plataformes com Windows (nativament)
- *Widgets* molt avançats, suporta, impressió, xarxa, i temes. Amplia funcionalitats del C++.
- Disposa de moltes eines avançades per a la creació de *GUIs*, algunes comercials.
- Molt estès amb aplicacions de tot tipus
- Documentació estesa, abundant i per a tots els nivells.
- *Nokia* és propietària de *Qt* a través de *Digia*. Continua sent de codi obert i no només controlat per *Digia*, sinó que es governa també a través d'una comunitat.
- Compatible amb quasi tots els entorns actuals, inclosos els dispositius mòbils.
- Contres:
  - Amplia funcionalitats del C++. amb el que suposa a nivell de compilador i hardware
  - Més lent i menys lleuger que altres entorns menys potents

- Situació de *Nokia* no gaire clara dins el mercat de telefonia mòbil, cosa que pot fer
- Molt poc lleuger, ha de tenir les llibreries instal·lades al sistema per a poder-se utilitzar.



II-lustració 11 - Clementine, un reproductor d'àudio fet amb Qt



II-lustració 12 - Qt Creator, un IDE complet per a crear aplicacions en QT

Petit diagrama de la situació actual per al disseny d'aplicacions a Linux

Llibreries gràfiques	GTK+ (2.20)	GTK+ (3.8)	GTK+ (3.8)	QT (5)	FOX (1.6.47)	wxWidgets (2.9.4)	FLTK (1.3)
Llenguatge de programació	Llenguatge C	Llenguatge C	Llenguatge C	Llenguatge C++	Llenguatge C++	Llenguatge C++	Llenguatge C++
Escriptori	Xfce	Gnome	Ubuntu (unity)	KDE	Independent	Independent	Independent
Llenguatge recomanat de programació d'aplicacions	C++ a través de XFCE Foundation Classes - XML	C / C++ - XML	Python - XML	C++ - QML (Dispositius mòbils)	C++	C++	C++
Suport altres llenguatges (Python, Ruby, SQL...) i llibreries completes d'escriptori	Si	Si	Si	Si	No	Si	No complet
Entorn de disseny d'interfícies d'usuari	Glade XFCE (XML)	Glade (XML)	Glade (XML)	QT creator (XML) QT Quick (QML)	No	WxBUILDER / wxFormBuilder	Fluid
Entorn de programació d'aplicacions	No nadiu	Anjuta	Gedit - quickly	QT creator	No	Si	No
Patrocinat comercialment	No	No	Si	Si	No	No	No
Documentació / Facilitat aprenentatge	Molt Be	Molt Be	Be (Python necessari)	Molt Be	Regular	Be	Regular

## Integració amb comandes Linux, ordres SQL i multimèdia

### Integració de les bases de dades SQLite

*SQLite* és una llibreria del llenguatge C de domini públic que implementa una base de dades relacional. És una novetat per que ha trencat amb el paradigma client-servidor de les bases de dades SQL, tenint en un sol fitxer \*.db les definicions de la base de dades i les dades mateixes. Suposen un avanç en que els programes poden tenir accés a una base de dades sense implementar i connectar amb un servidor, només fent una crida o implementant-la com si fos una llibreria (include).

La part dolenta és que te un mecanisme bastant dur de bloqueig en mode escriptura que fa que només un usuari pugui accedir en aquest mode.

És molt més ràpida que els altres sistemes SQL més populars com el *MySQL* o el *PostgreSQL*, i unit al tema de la manca de necessitat de servidor, fa una opció molt atractiva per a programadors.

La instal·lació és molt senzilla, totes les distribucions de Linux més populars tenen compilats paquets en els seus repositoris i si no a la web de *SQLite* tenen tant binaris precompilats per a la majoria d'entorns tant mòbils com d'escriptori i el codi font per a poder compilar-lo manualment. Al ser una base de dades SQL, les instruccions per a poder manipular-la són molt semblants a qualsevol altra base de dades SQL amb alguna modificació al no intervenir servidor ni consola.

El seu futur sembla bastant consolidat al ser els seus patrocinadors maques tan reconegudes com *Bloomberg*, *Mozilla*, *Nokia*, *Oracle*, *Adobe* o *Bentley*.

NO hi ha problema per al control d'aquestes bases de dades per part dels llenguatge més moderns, al existir llibreries que ho implementen, els llenguatges basats en C / C++ no ho necessiten ja que *SQLite* és una llibreria de C.

Python	Python-sqlite (SQLite 2) – Pysqlite2 (SQLite 3)
Qt	libqt4-sql-sqlite
Gambas	gambas3-gb-db-sqlite3
WxWidgets	wxSQLite3
Ruby	ruby-sqlite3

GTK+, FLTK i FOX no tenen un embolcall especial per a SQLite, encara que hi han

projectes a la xarxa en GTK i FLTK especialment que fan que la seva combinació no tingui cap problema.

## Integració de la informació multimèdia

Els *tags* multimèdia són la informació de l'autor, àlbum, durada, etc.. que està inserida a cada un dels fitxers estàndards multimèdia més populars. El problema que es veu a priori es que cada format te la seva pròpia manera d'inserir aquesta informació al fitxer. Estudiant el codi font del servidor *miniDlna*, trobem la funció que fa aquesta tasca i insereix aquesta informació a la base de dades *SQLite* (*metadata.c* i *metadata.h*). En el seu encapçalament fa referència a tot un conjunt de llibreries del sistema:

*libexif/exif-loader.h* (manipulació de la metadata (EXIF) dels arxius *dimatge*), llibreria amb llicència GNU feta per [www.exif.org](http://www.exif.org)

*jpeglib.h* (Llibreria del grup independent *JPEG*)

*avutil.h* – Llibreria del projecte *FFmpeg*

*avcodec.h* - Llibreria del projecte *FFmpeg*

*avformat.h* Llibreria del projecte *FFmpeg*

*tagutils/tagutils.h* – Llibreria treta del projecte *ReadyNas*

que son les que tenen definides les estructures de cada tipus d'arxiu i les característiques de cada arxiu que porta inserides. Així defineix també una sèrie de funcions pròpies per aprofitar aquesta informació per al *miniDlna* que identifiquen quin tipus de fitxer és per la seva extensió, i depèn del tipus ja extrauen certa informació o una altra. Quan la te extreta la guarda a la base de dades a la taula *DETAILS*.

Després d'investigar una mica podem descobrir que existeixen diversos visors de bases de dades *SQLite*, és a dir, que ens deixen obrir la base de dades com si fos un arxiu i podem observar les seves taules i com estan definides. Això es de gran ajuda pel propòsit del projecte, ja que simplificarà molt la tasca de la extracció de les metadades o *tags* dels arxius multimèdia. Obrim una taula creada amb un servidor *miniDlna* amb uns pocs arxius de cada tipus (àudio, vídeo i fotografia) i veiem la seva estructura. En concret ens interessa la taula *DETAILS*:

SQLite Database Browser - /home/jordi/.minidlna/cache/files.db

File Edit View Help

Database Structure Browse Data Execute SQL

Table: DETAILS

ID	PATH	SIZE	TITLE	DURATION	BITRATE	SAMPLERATE	ARTIST	ALBUM	GENRE
133	133		- All Artists -						Alternative Metal
134	134	/home/jordi/Música/01- Prison Song.mp3	Prison Song	0:03:21.563	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
135	135		Toxicity				System Of A Do		Nu Metal / Hardcore
136	136		System Of A Do						Nu Metal / Hardcore
137	137		- All Albums -				System Of A Do		Nu Metal / Hardcore
138	138		Nu Metal / Hard						
139	139		- All Artists -						Nu Metal / Hardcore
140	140	/home/jordi/Música/02 Digital Bath.mp3	Digital Bath	0:04:15.328	320000	48000	Deftones	White Pony	Alternative Metal
141	141	/home/jordi/Música/02 Estoy Por Que He Ver	Estoy Por Que He Ver	0:04:08.711	320000	44100	Lírico	Un Antes Y Un C	
142	142	/home/jordi/Música/02- Needles.mp3	Needles	0:03:13.831	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
143	143	/home/jordi/Música/03- Deer Dance.mp3	Deer Dance	0:02:55.128	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
144	144	/home/jordi/Música/03 Elite.mp3	Elite	0:04:01.863	320000	48000	Deftones	White Pony	Alternative Metal
145	145	/home/jordi/Música/03 Juntos En Esto (Con K	Juntos En Esto (Con K	0:03:52.672	320000	44100	Lírico	Un Antes Y Un C	
146	146	/home/jordi/Música/04 De Vuelta Al Pasado.r	De Vuelta Al Pasado	0:04:06.804	320000	44100	Lírico	Un Antes Y Un C	
147	147	/home/jordi/Música/04- Jet Pilot.mp3	Jet Pilot	0:02:06.540	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
148	148	/home/jordi/Música/04 RX Queen.mp3	RX Queen	0:04:27.736	320000	48000	Deftones	White Pony	Alternative Metal
149	149	/home/jordi/Música/05 Street Carp.mp3	Street Carp	0:02:41.535	320000	48000	Deftones	White Pony	Alternative Metal
150	150	/home/jordi/Música/05 Summer Love.mp3	Summer Love	0:03:05.260	320000	44100	Lírico	Un Antes Y Un C	
151	151	/home/jordi/Música/05- X.mp3	X	0:01:58.625	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
152	152	/home/jordi/Música/06- Chop Suey!.mp3	Chop Suey!	0:03:30.315	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
153	153	/home/jordi/Música/06 Frio Invierno.mp3	Frio Invierno	0:02:27.226	320000	44100	Lírico	Un Antes Y Un C	
154	154	/home/jordi/Música/06 Teenager.mp3	Teenager	0:03:20.655	320000	48000	Deftones	White Pony	Alternative Metal
155	155	/home/jordi/Música/07- Bounce.mp3	Bounce	0:01:54.288	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
156	156	/home/jordi/Música/07 Hay Una Fiesta (Con	Hay Una Fiesta (Con	0:04:24.907	320000	44100	Lírico	Un Antes Y Un C	
157	157	/home/jordi/Música/07 Knife Prty.mp3	Knife Prty	0:04:49.215	320000	48000	Deftones	White Pony	Alternative Metal
158	158	/home/jordi/Música/08- Forest.mp3	Forest	0:04:00.172	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
159	159	/home/jordi/Música/08 Korea.mp3	Korea	0:03:23.607	320000	48000	Deftones	White Pony	Alternative Metal
160	160	/home/jordi/Música/08 Suave (Smooth Oper	Suave (Smooth Oper	0:04:08.450	320000	44100	Lírico	Un Antes Y Un C	
161	161	/home/jordi/Música/09- ATWA.mp3	ATWA	0:02:56.277	320000	44100	System Of A Do	Toxicity	Nu Metal / Hardcore
162	162	/home/jordi/Música/09 Con Calma.mp3	Con Calma	0:03:47.604	320000	44100	Lírico	Un Antes Y Un C	
163	163	/home/jordi/Música/09 Passenger.mp3	Passenger	0:06:08.559	320000	48000	Deftones	White Pony	Alternative Metal
164	164	/home/jordi/Música/10 Change (In The Hous	Change (In The Hous	0:05:01.239	320000	48000	Deftones	White Pony	Alternative Metal

1 - 177 of 177

Go to: 0

II-lustració 13 - Captura de la taula SQL del MiniDLNA "details"

Veiem que amb la taula DETAILS podrem aconseguir tots els detalls que ens interessin per a l'aplicació i que podrem intentar aconseguir una bona visualització. Igualment, la estructura de la funció *metadata*, fa que també es pugui aprofitar part del codi per als propòsits particulars.



# Disseny

## Introducció

L'aplicació pensada és una aplicació gràfica per a configurar el servidor miniDLNA en entorn Linux. La MiniDLNA és una aplicació de codi obert que implementa un servidor que compleix els estàndards DLNA en un entorn Linux. Està compilada i es pot trobar en els repositoris de quasi totes les distribucions Linux, segur en les més populars, i a la web del projecte es pot descarregar tant els fitxers font de l'aplicació per tal de compilar-la com una versió ja compilada (estàtica), per a descomprimir-la i executar-la directament en qualsevol entorn Linux actual.

L'aplicació MiniDLNA es fonamenta en dues estructures fonamentals: Un fitxer de configuració en mode text, que conté els paràmetres fonamentals del servidor, i una base de dades SQLite, a on el servidor emmagatzema els detalls de tots els arxius multimèdia que hi ha als directoris especificats en el fitxer de configuració.

L'aplicació a dissenyar té tres objectius clars diferenciats i imprescindibles que tenen a veure amb les diferents funcionalitats del servidor:

Tractar el fitxer de configuració, això vol dir:

Crear el fitxer si no existeix

Importar els paràmetres d'un fitxer existent

Modificar els paràmetres, implementar-los tot fent una còpia de seguretat de l'arxiu de configuració existent.

Administració del servidor:

Arrencar el servidor

Aturar el servidor

Resetejar el servidor amb els paràmetres nous

Base de dades

Visualitzar la base de dades en un format intel·ligible (camí al fitxer, artista, títol...) que permeti veure els arxius que hi ha al servidor.

Aplicar els canvis als directoris del servidor

Eliminar fitxers de la base de dades

Objectiu addicional marcat:

Poder visualitzar o reproduir els fitxers multimèdia en algun tipus de reproductor encastrat a l'aplicació.

## Aparença

### Objectiu 1: Tractament del fitxer de configuració:

Anàlisi dels paràmetres del fitxer de configuració:

Nom	Descripció	Necessari (N) o per omission (D)	Caixetí necessari
port (8200)	Port pel tràfic http	N	Text ("8200" defecte)
network_interface	Interfície de xarxa	N	Text ("eth0" defecte)
media_dir	Directoris multimèdia	N Acumulable	1. Llista desplegable (A,V,P, ) 2. Escollir directori (estàndar Qt?)
friendly_name	Nom que apareix als clients	N	Text ("Servidor Dlna" defecte)
db_dir	Directoris per base de dades i artwork	N	Text / Estàndar Qt ("home/\$USER/.minidlna/cache" defecte)
log_dir	Directoris per arxius de registre	N	Text / Estàndar Qt ("home/\$USER/.minidlna/log" defecte)
album_art_names	Tipus arxius portada	D	
inotify	Auto descobrir arxius nous	D(si)/N - Opcional	Thick ✓
enable_tivo		D (No)	
strict_dlna		D (No)	
presentation_url		D (Comentat)	
notify_interval	Interval de notificació	D (900)	
serial		D(12345678)	
model_number		D(1)	
root_container	Directoris d'inici	N	Llista desplegable (Default (.),B,M,V,P, )

Per tant, aquest és l'esbós de l'aplicació en la vista de definició dels paràmetres del servidor:

The image shows a configuration window with two tabs: 'Parametres' and 'Base de Dades'. The 'Base de Dades' tab is active. The configuration options are as follows:

- Port: Text box
- Network int: Text box
- Friendly name: Text box
- root\_container: Dropdown menu with 'Audio|Video|Imatges' selected and a 'Selecciona' button.
- inotify: Radio button labeled 'Activat'.
- Directoris: Dropdown menu with 'Audio|Video|Imatges' selected and a 'Selecciona' button.

Below the Directoris dropdown is a list box containing the following items:

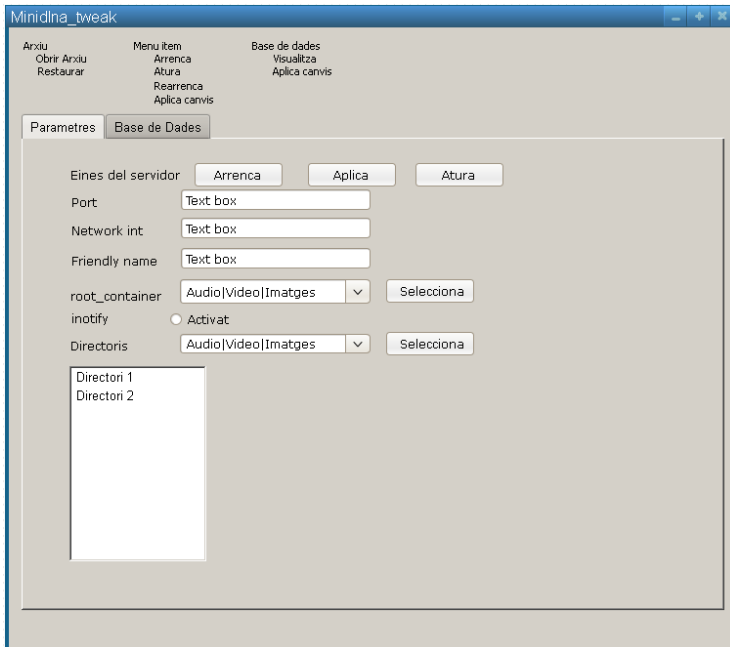
- Directori 1
- Directori 2

#### II·l·lustració 14 - Esbós de la pestanya de configuració

L'opció escollida per tal de separar els aspectes de configuració de servidor a modificar en el fitxer de text i l'altre aspecte enfocat a la visualització de la llista de fitxers multimèdia i la seva visualització és la de separar-los en pestanyes dintre de la mateixa aplicació.

## Objectiu 2: Administració del servidor

El mètode escollit per tal de controlar el servidor i poder engegar-lo, aturar-lo o aplicar els canvis fets en els paràmetres és la instal·lació de tres botons a sobre de la definició dels paràmetres:

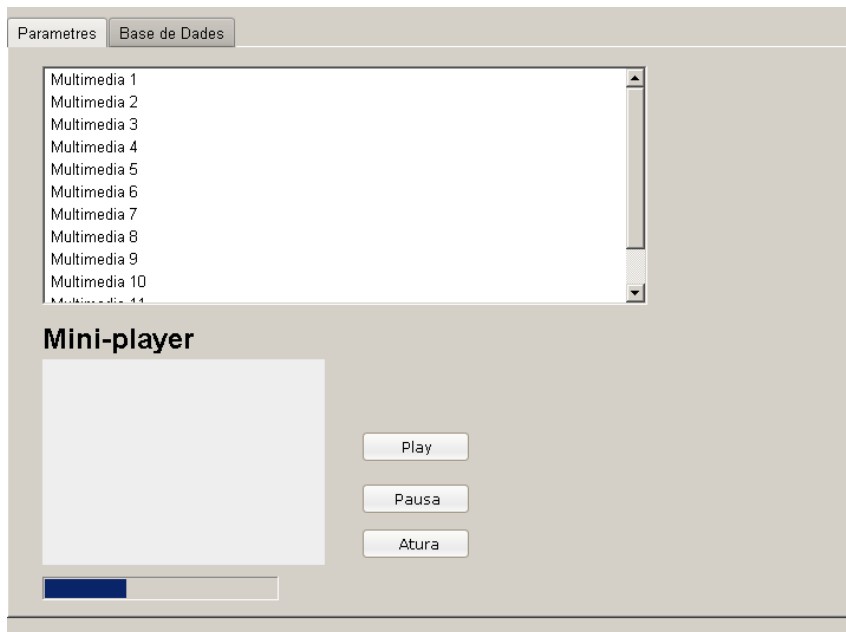


II-lustració 15 - Esbós de la pestanya de configuració amb botons i menú

## Objectius 3 i 4: Base de dades i reproductor:

A la pestanya de la base de dades apareixerà una llista navegable dels arxius multimèdia i una finestra per a un reproductor amb uns controls mínims

L'aparença per a la pestanya de la base de dades seria:



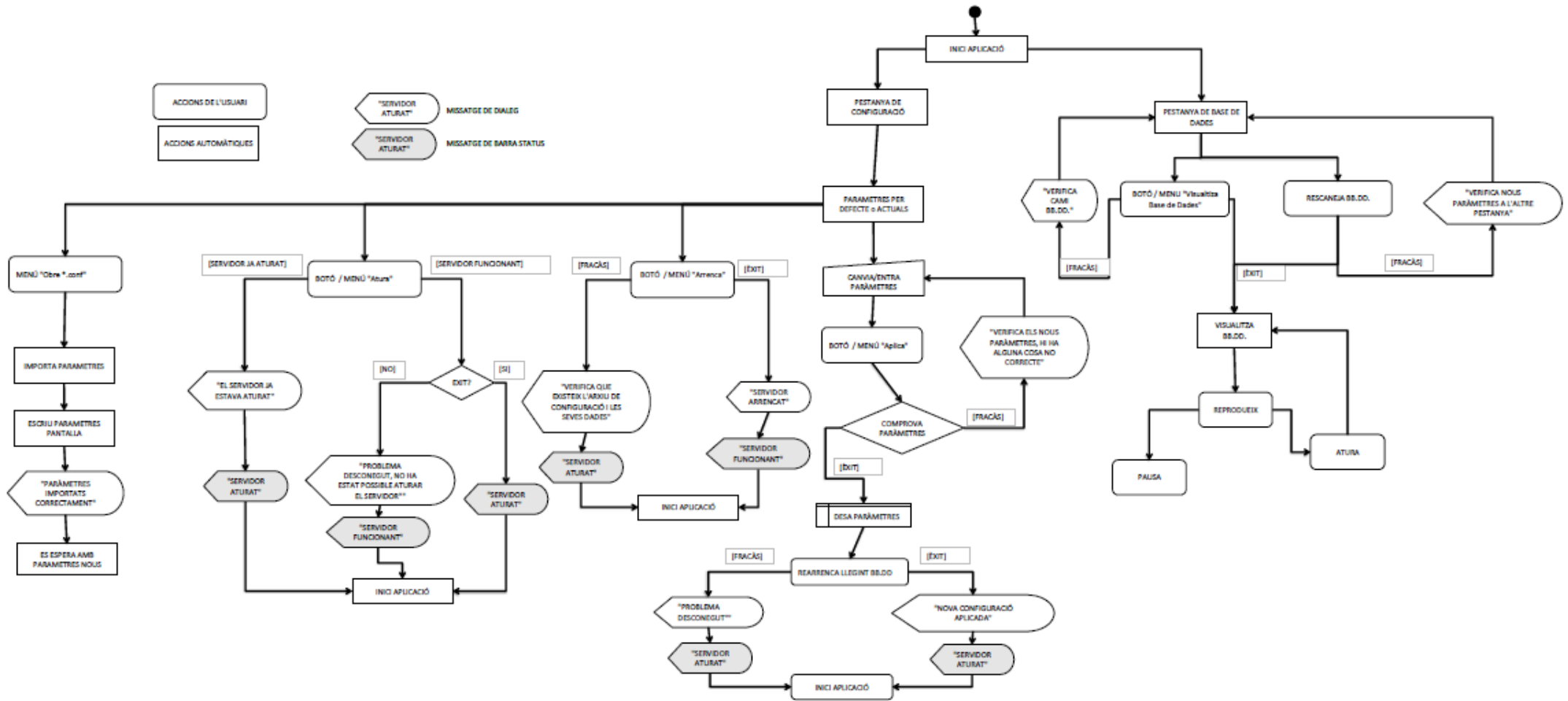
#### II-lustració 16 - Esbós de la pestanya de la base de dades i el reproductor

Al menú de l'aplicació estan reflectides totes les funcions que l'aplicació podrà fer dividides en els diferents aspectes:

Arxiu	Menu item	Base de dades
Obrir Arxiu	Arrenca	Visualitza
Restaurar	Atura	Aplica canvis
	Rearrenca	
	Aplica canvis	

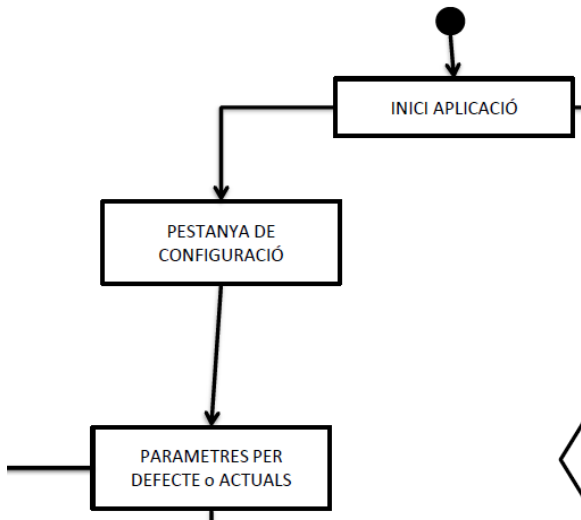
#### II-lustració 17 - Esbós del menú

# Flux d'interacció de l'aplicació



Il·lustració 18 - Flux d'interacció de l'aplicació

## Desenvolupament en pseudo-codi i diagrames



II-Iustració 19 - Diagrama inici aplicació

L'aplicació comença amb la pestanya de configuració per defecte i els quadres de text amb les entrades de configuració emplenades amb les opcions de configuració més comunes allà on l'opció sigui clara.

Els paràmetres per defecte han de ser visualitzables per l'usuari i han de ser els més comuns, fent que l'usuari, si ho desitja, només hagi d'escollir els directoris multimèdia per a compartir.

Els paràmetres per defecte comuns per al servidor i que estaran introduïts són:

Paràmetre	Valor
Port	8200
Network interface	Buit. El servidor les autodescobreix
Server name	"Servidor Multimedia"
Root_container	"Navegar"
db_dir	"home/NOM_USUARI/.minidlna/cache"
log_dir	"home/NOM_USUARI/.minidlna/"

Per tant, la llista de funcions/accions serà:

Mostra l'aplicació

Pestanya per defecte-> Pestanya de configuració de servidor.

Aconseguir nom Usuari

Formatar paràmetres amb el nom d'usuari

Emplenar paràmetres per defecte

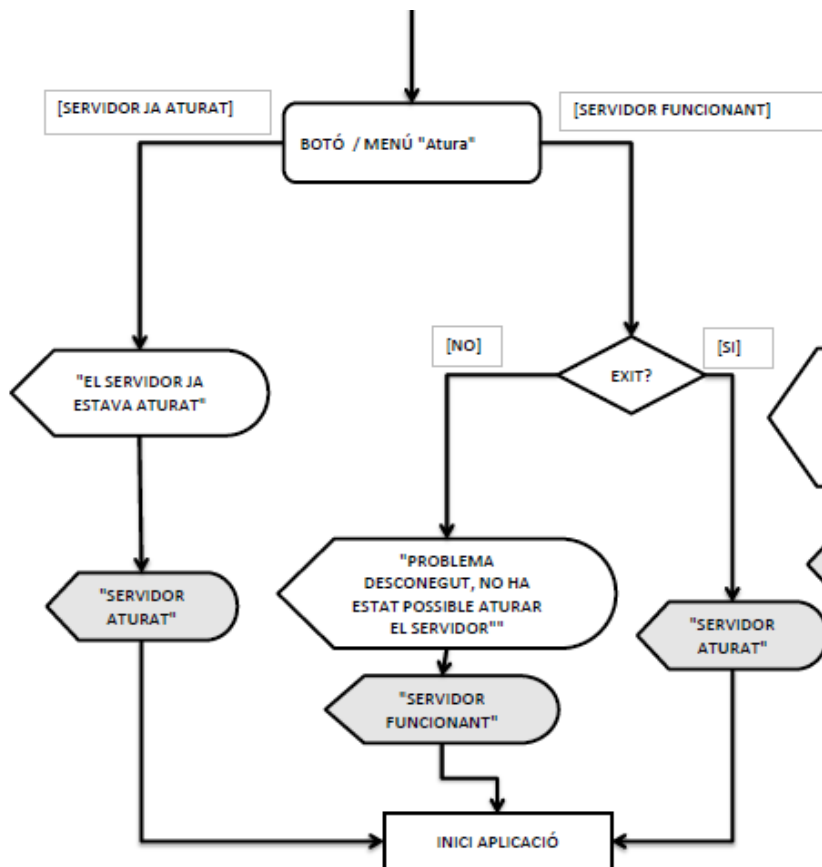
Mostrar status servidor barra status (FUNCIÓ COMÚ A MOLTS PROCESSOS) Definició:

SI (servidor funciona) LLAVORS

"Servidor funcionant"

SINÓ "servidor aturat"

Aplicació en espera d'acció d'usuari



II-lustració 20 - Diagrama opció aturar servidor

### BOTO ATURA – MENU ATURA

Ordre per aturar el servidor

Comprovar funcionament del servidor

SI (servidor no funciona) LLAVORS

Mostra **diàleg** “El servidor ja estava aturat”

Mostrar status servidor barra status

SINÓ

Ordre aturar procés servidor

SI (servidor funciona) LLAVORS

Mostra **diàleg** “Problema desconegut, no ha estat possible aturar el servidor”

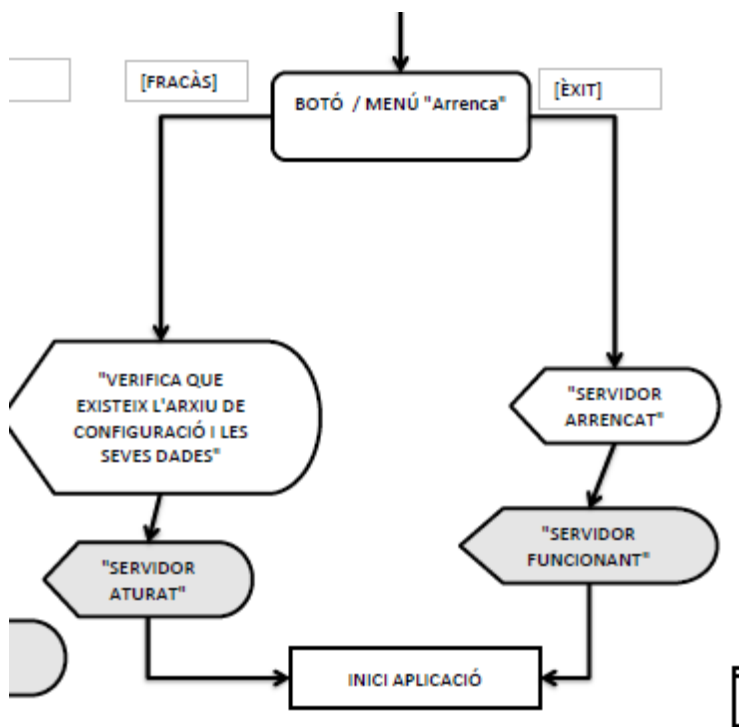
Mostrar status servidor barra status

SINO

Mostra **diàleg** “Servidor Aturat”

Aplicació en espera d'acció d'usuari





Il·lustració 21 - Diagrama opció arrencar servidor

**BOTO ARRENCA – MENU ARRENCA (Arrenca el servidor MiniDLNA amb la configuració que hi ha al directori per defecte)**

Ordre per arrencar el servidor

SI (servidor funciona) LLAVORS

Mostra **diàleg** “Servidor arrencat”

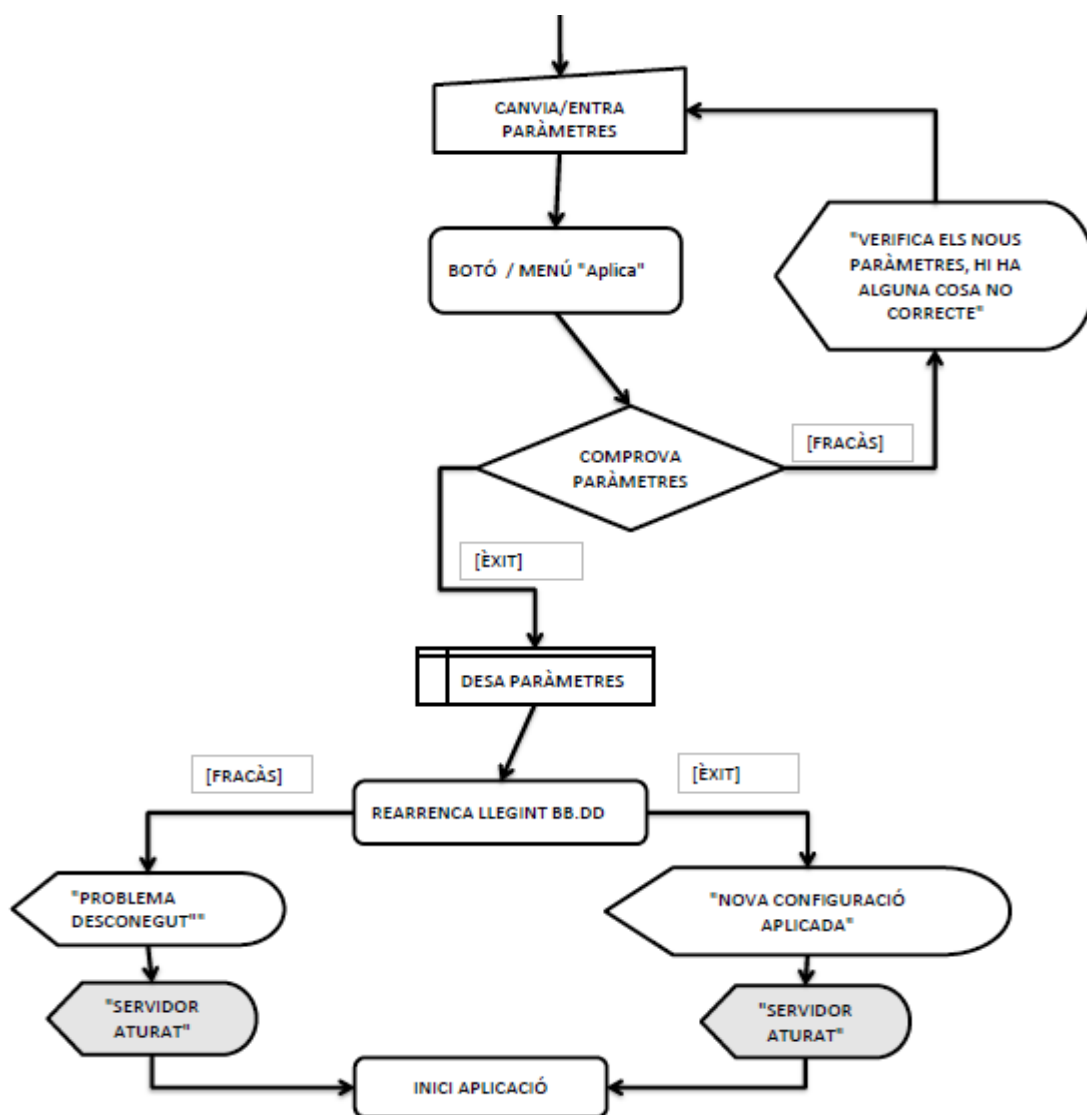
Mostrar status servidor barra status

SINÓ

Mostra **diàleg** “Verifica que existeix la configuració i les seves dades”

Mostrar status servidor barra status

Aplicació en espera d’acció d’usuari



Il·lustració 22 Diagrama opció aplicar canvis

**BOTO APLICA – MENU APLICA (Aplica la configuració entrada al servidor MiniDLNA i l’arrenca o el rearrenca)**

Aplicació en espera d’acció d’usuari

SI (Comprova paràmetres) – *Explicació funció al final LLAVORS*

Desa paràmetres

Arrenca o rearrenca el servidor (**mateixa comanda**)

SI (servidor funciona) LLAVORS

Mostra **diàleg** “Nova configuració aplicada”

Mostrar status servidor barra status

SINÓ

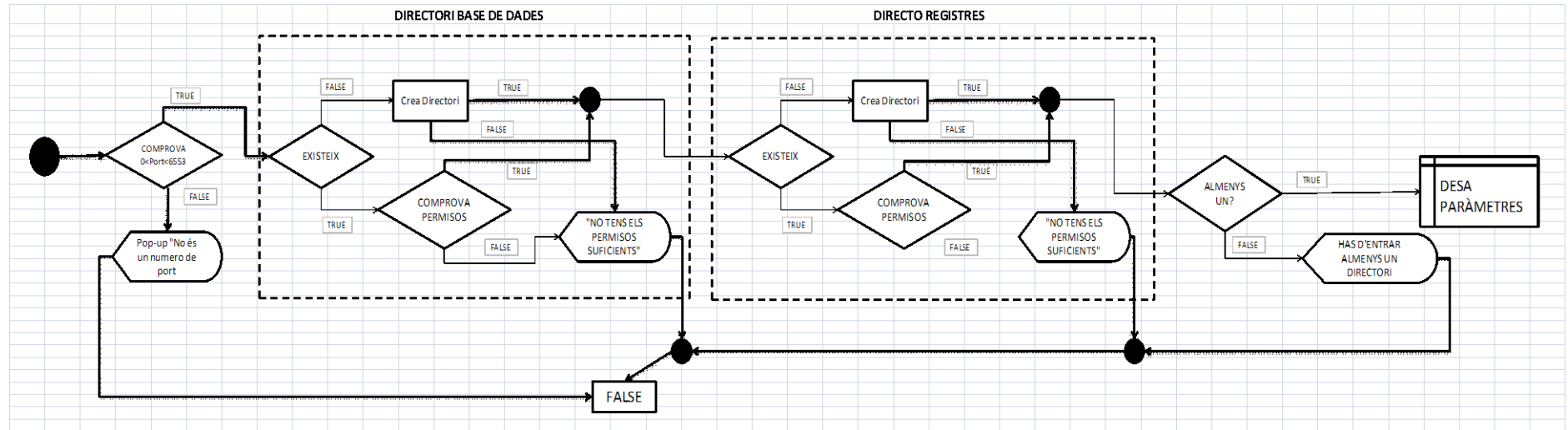
Mostra **diàleg** “Problema desconegut, servidor aturat”

Mostrar status servidor barra status

SINÓ Mostra **diàleg** “Verifica els nous paràmetres, hi ha alguna entrada no correcte”

Aplicació en espera d'acció d'usuari

## FUNCIÓ COMPROVAR PARÀMETRES



II-lustració 23 - Diagrama Funció per a comprovar paràmetres

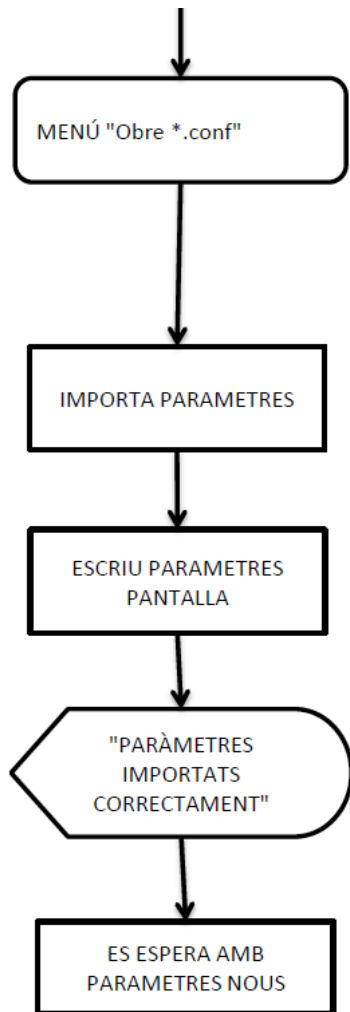
Aquest funció comprova si:

El numero de port està entre 0 i 65535

Els directoris existeixen, i si existeixen si l'usuari té permisos

Si no existeixen, s'intenten crear, si no es pot, s'assumeix que no es tenien permisos

Si hi ha almenys un directori multimèdia entrat



#### Il·lustració 24 - Diagrama Obrir arxiu

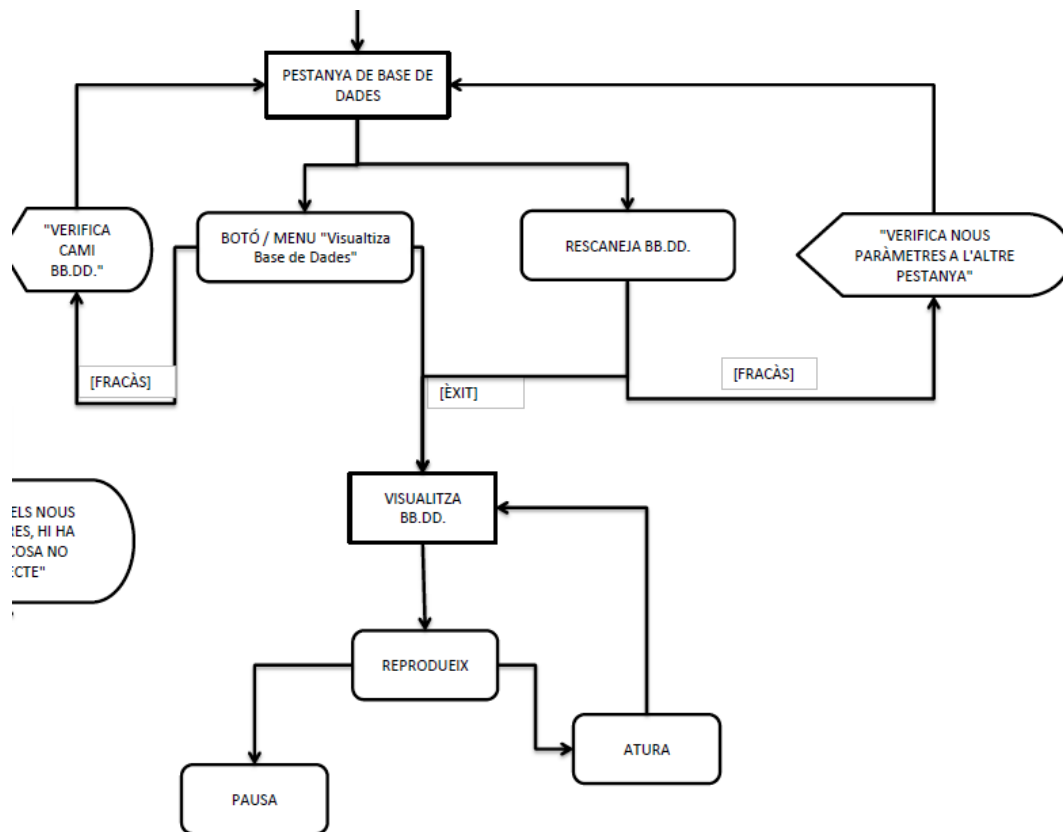
**MENÚ "OBRE \*.CONF"** – Obrir un arxiu de configuració existent, importa els paràmetres, els comprova i els escriu per pantalla en les caselles corresponents

Obre l'arxiu

Importa els paràmetres

Els escriu per pantalla

Mostra un diàleg amb la confirmació de la importació



## II-lustració 25 - Visualització Base de Dades

BOTO / MENÚ "Visualitza base de dades" i Reescaneja Base de dades, aquesta última funció està pensada per si s'introdueixen nous directoris multimèdia, aplica els nous paràmetres fent servir la mateixa funció que a la primera pestanya i visualitza la base de dades de la mateixa manera que amb l'altre botó.

Per a visualitzar la base de dades:

Connectar amb la base de dades (ordre SQL)

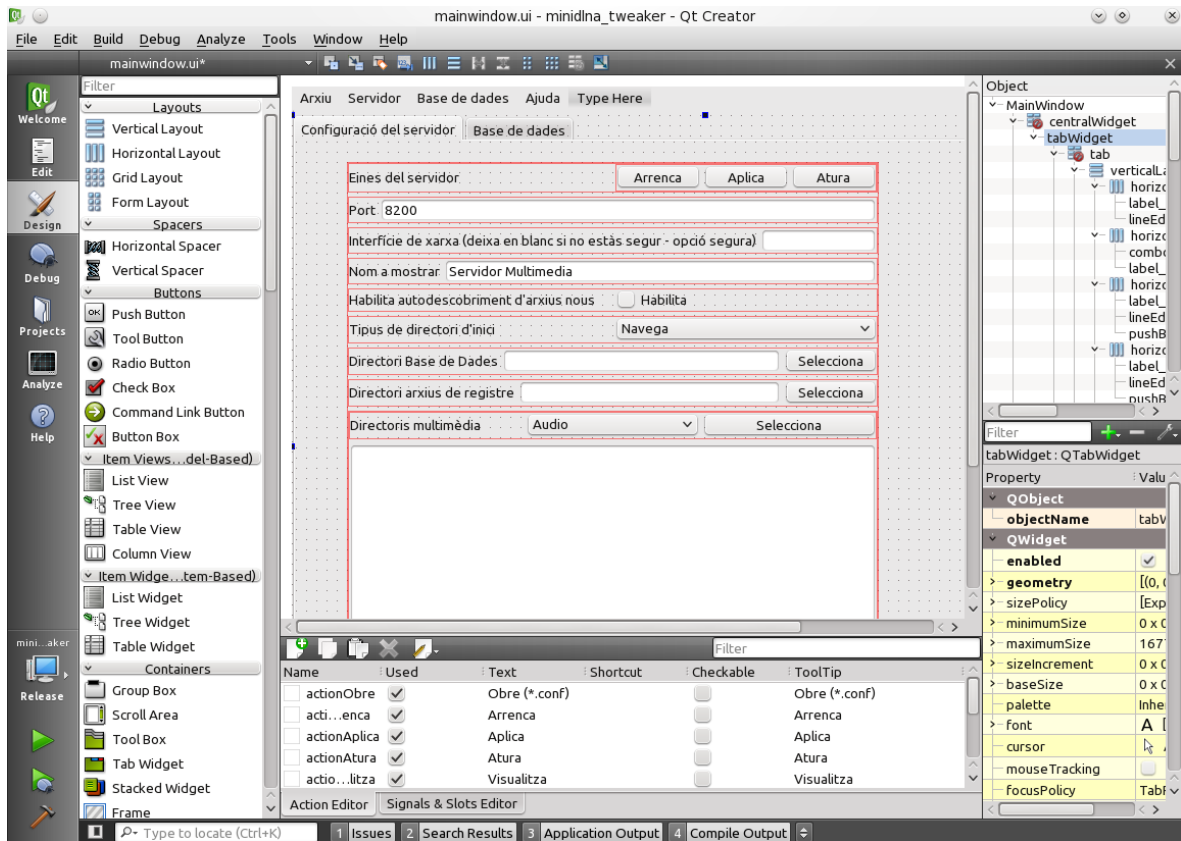
Visualitzar la base de dades (Visualitzar resultat ordre SQL en una finestra)

Per a reproduir els fitxers es podrà fer doble click al fitxer a la llista de la base de dades visualitzada o mitjançant botons de reproductor, hi ha l'esquema clàssic d'un reproductor.

## Implementació

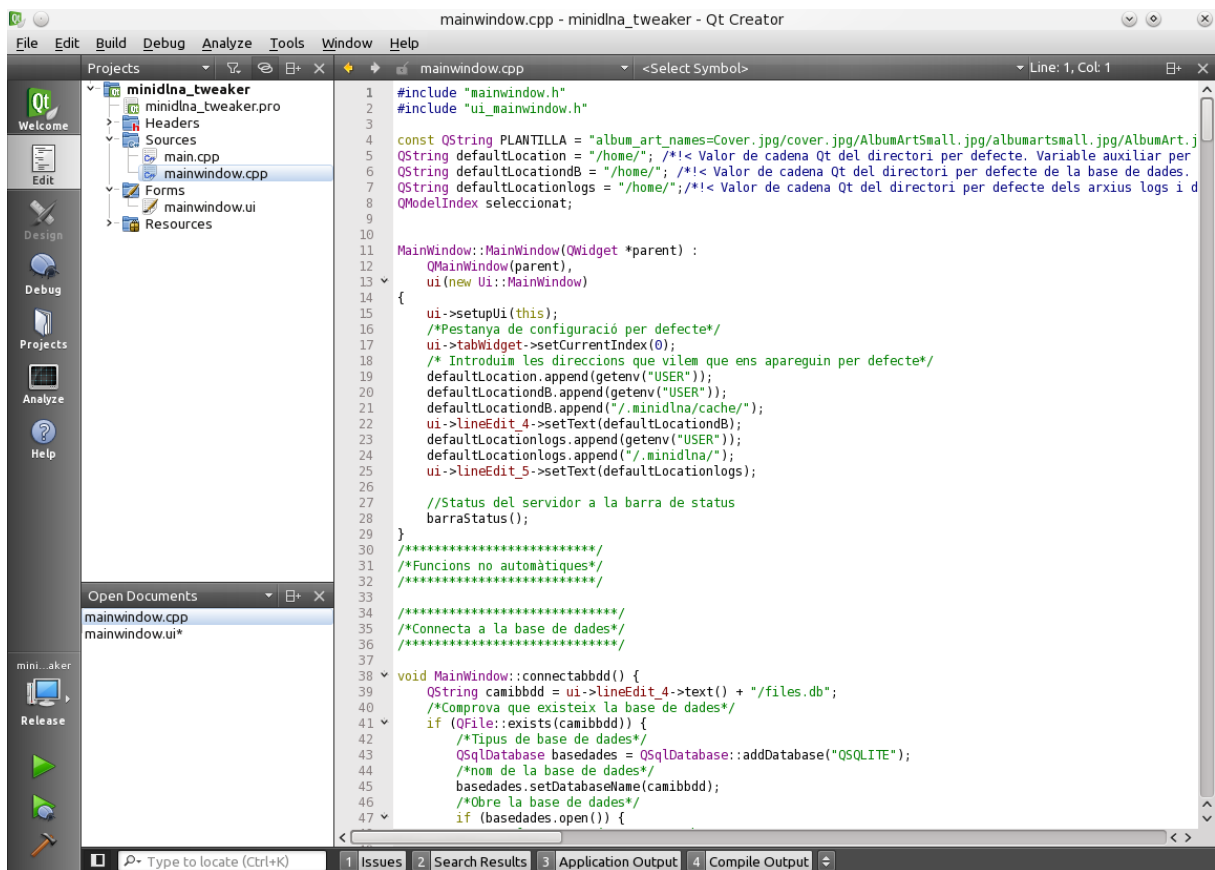
Després d'investigar les possibles opcions per al desenvolupament de l'aplicació a l'estat de l'art, l'entorn de desenvolupament escollit per a desenvolupar l'aplicació és el de les llibreries Qt. Te les característiques ideals per aquest projecte, les seves principals llibreries estan "de sèrie" en les principals distribucions Linux, ofereix un entorn de programació molt senzill, intuïtiu i amb una gran implementació de l'ajuda per les seves llibreries en aquest entorn. L'entorn Qt no és només que una sèrie de llibreries desenvolupades en el llenguatge C++, que implementen un entorn gràfic a part de millorar i donar suport a altres tecnologies que son típicament usades en la programació d'aplicacions d'escriptori com son les bases de dades o el suport multimèdia. La possibilitat de desenvolupar l'aplicació dintre d'un mateix entorn ha estat un factor decisoris per a escollir aquest entorn.

L'entorn de creació és el *Qt Creator* , un entorn que permet definir l'aplicació gràficament en un entorn WYSIWYG (El que veus és el que obtindràs), i al mateix temps definir les funcions que interaccionaran amb aquell entorn gràfic, les classes addicionals si es necessiten, i també és un entorn per a poder fer la depuració i la compilació molt senzill.



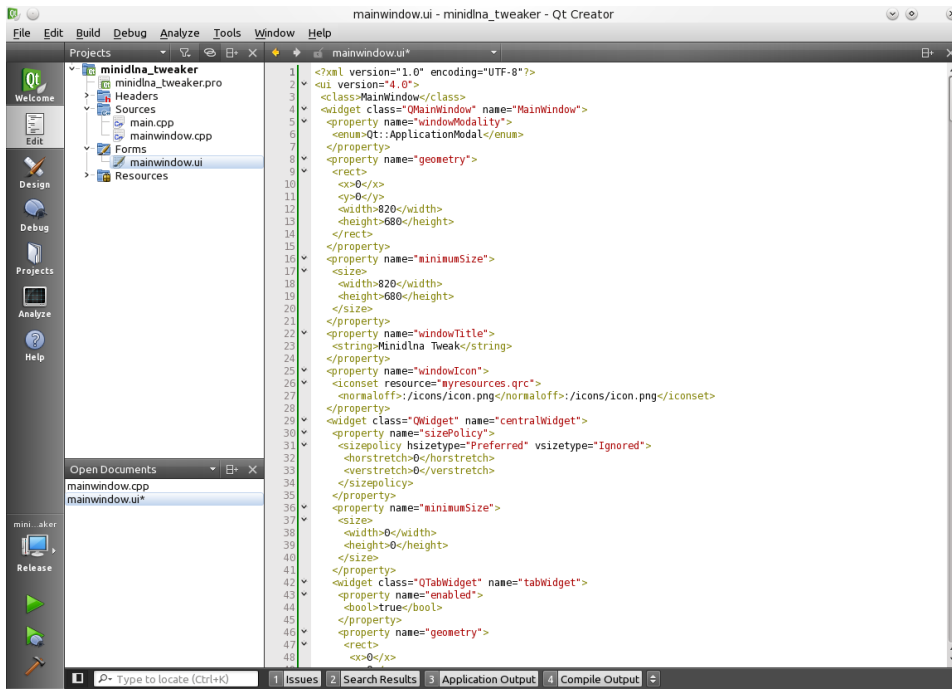
II-Il·lustració 26 - Entorn de desenvolupament Qt Creator en el mode gràfic





## II-lustració 27 - Entorn de desenvolupament Qt Creator en mode programació.

L'avantatge d'utilitzar un IDE amb un entorn gràfic integrat es que el software genera el codi necessari per a generar l'entorn gràfic d'usuari, la distribució gràfica es guarda en un arxiu \*.ui en format xml:

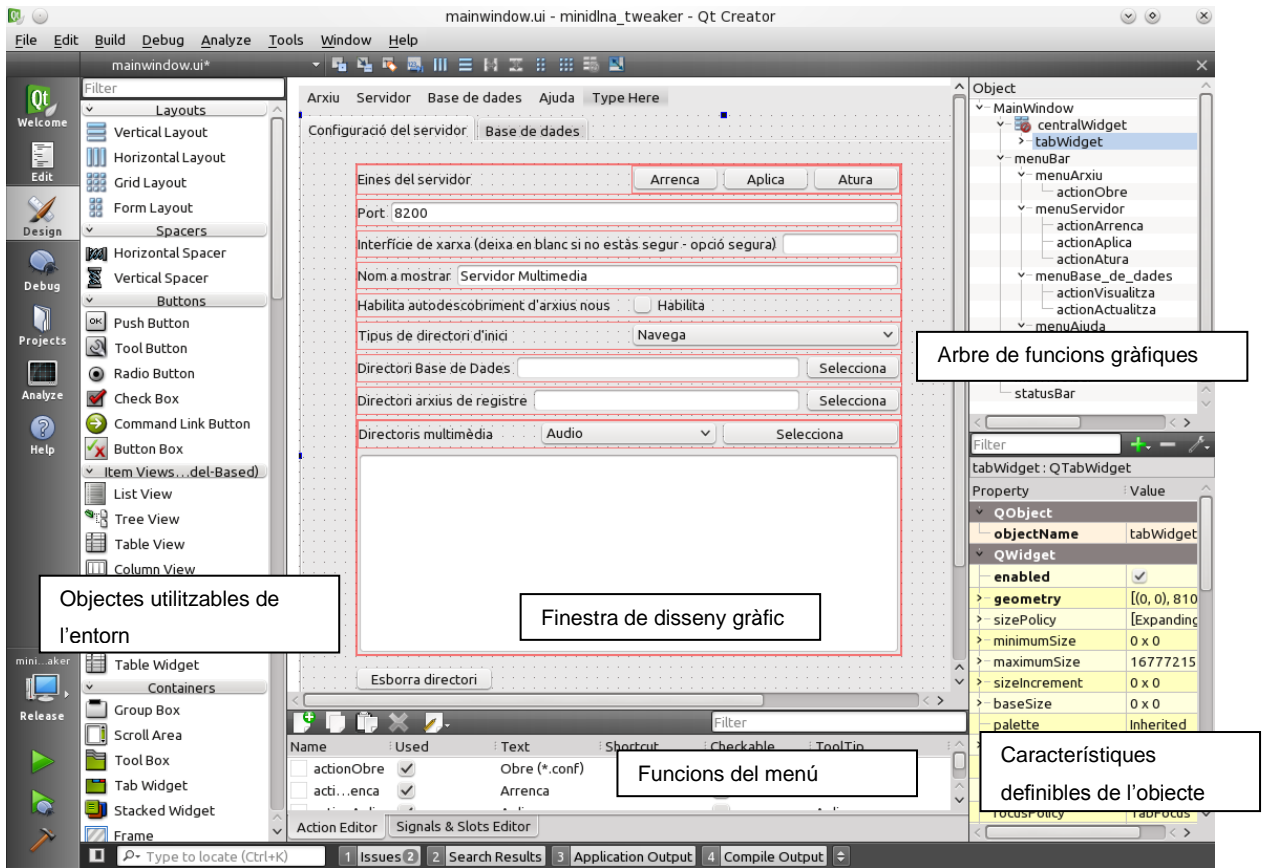


## II-lustració 28 - Codi XML generat pel disseny gràfic del Qt Creator

Aquest codi XML genera en temps de compilació els fitxers C++ necessaris per tal de compilar el codi sencer de l'aplicació amb un compilador C++, en aquest cas GCC-C++.

Amb aquestes característiques, una vegada conegut el funcionament de les llibreries Qt, i dominat l'entorn, es pot començar a programar d'una manera molt senzilla.

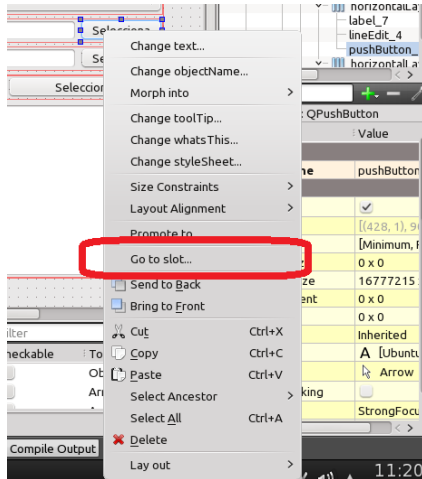
El primer pas ha estat plasmar els esbossos de l'aplicació en el Qt Creator en la funció del disseny gràfic de l'aplicació.



II-Il·lustració 29 - Plasmació de l'esbós gràfic al Qt Creator

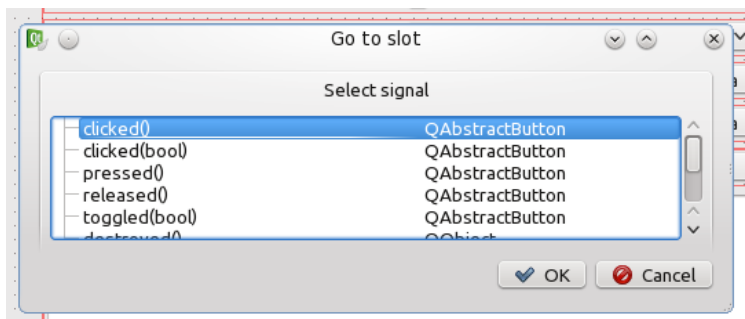
Per tant, tot l'entorn gràfic i les seves característiques es poden gestionar gràficament des d'aquest entorn.

Les interaccions amb l'usuari es defineixen molt intuïtivament, per exemple el botó de selecció d'un directori multimèdia es defineix de la següent manera:



II-Il·lustració 30 - Menú contextual ->Go to slot...

Automàticament s'obre una finestra amb les opcions de la interacció que es volen definir amb el botó.



II-Il·lustració 31 - Senyals definibles

Automàticament al pitjar el botó OK, la finestra per a definir el codi queda oberta per tal que el programador pugui definir la funció que s'ha d'executar al pitjar el botó, en aquest cas la selecció d'un directori:

```
340 p->start("/bin/bash", QStringList() << "-c" << QString("kill pid "+ nrpId));
341 p->waitForFinished();
342 barraStatus();
343 }
344 else {
345     QMessageBox::information(this, "Proces no aturat", "No hi havia no no s'ha trobat proces actual");
346     barraStatus();
347 }
348 }
349
350 /*****
351  *Directori base de dades*/
352 *****/
353
354 void MainWindow::on_pushButton_4_clicked()
355 {
356     //Selecciona el directori
357     QString dir;
358     dir = QFileDialog::getExistingDirectory(this, tr("Selecciona directori"), defaultLocationdB, QFileDialog:
359     //Si no es cancela, entra el directori, si no no fa res
360     if (dir != "") {
361         ui->lineEdit_4->setText(dir);
362     }
363 }
364
365 /*****
366  *Directori de logs*/
367 *****/
368
369 void MainWindow::on_pushButton_5_clicked()
370 {
371     //Selecciona el directori
372     QString dir;
```

II-lustració 32 - Definició del codi de la senyal

## Desenvolupament dels objectius

### Tractament del fitxer de configuració

Ítems a desenvolupar:

Crear el fitxer si no existeix

Importar els paràmetres d'un fitxer existent

Modificar els paràmetres d'un arxiu existent fent una còpia de seguretat de l'arxiu de configuració existent.

a. Crear el fitxer si no existeix:

El fitxer de configuració és crea amb els paràmetres que defineix l'usuari més els paràmetres decidits que es fixin per defecte. Per tant primer s'ha creat una constant de text amb els paràmetres per defecte:

Codi:

```
const          QString          PLANTILLA          =
"album_art_names=Cover.jpg/cover.jpg/AlbumArtSmall.jpg/albumartsmall.j
pg/AlbumArt.jpg/albumart.jpg/Album.jpg/album.jpg/Folder.jpg/folder.jpg
/Thumb.jpg/thumb.jpg\nenable_tivo=no\nstrict_dlna=no\nnotify_interval=
900\nserial=12345678\nmodel_number=1";
```

Aquí ja es pot veure la primera llibreria usada de l'entorn Qt, la *QString*, una llibreria semblant al *String* de C++ però amb una sintaxis i unes operacions implementades molt més complertes.

Per tal d'escriure tots els paràmetres a un nou arxiu de configuració, s'ha fet servir un arxiu intermig on s'escriuen els paràmetres, així, d'aquesta manera, com que els paràmetres es comproven abans d'escriure, si hi ha algun paràmetre incorrecte, només es mostrarà el missatge d'error i s'esborrarà aquest arxiu intermig sense modificar el arxiu de configuració actual si existia.

```
/*Escriure parametres a l'arxiu temporal*/
```

```
bool MainWindow::escriureParametres(QString descripcio, QString
parametre)
{
    //Directori de treball
    QDir::setCurrent(defaultLocationlogs);
    //Obre l'arxiu
    QFile nom("nou.txt");
    if (!nom.open(QFile::Append | QFile::Text)) {
```

```

        QMessageBox::information(this, "Error", "Hi han problemes
amb els permisos d'usuari");
        return false;
    }
    else {
        //Escriu a l'arxiu
        QTextStream out(&nom);
        out << descripcio << parametre << "\n";
        //tanca l'arxiu
        nom.flush();
        nom.close();
        return true;
    }
}

```

Aquí s'utilitzen quatre classes Qt :

QDir: Classe per a manipular directoris. S'ha fet servir per definir el directori per defecte

QFile: Classe per a manipular fitxers (la manipulació és més senzilla que amb C++ o amb ordres Linux de manipulació)

Es defineix el nom del fitxer, s'obre per a afegir informació en format text, si hi ha algun problema amb l'obertura, sortirà un missatge indicant l'error.

Al final de la funció, tanquem el fitxer amb *flush* i *close*.

QTextStream: Classe per a l'entrada i sortida de text.

Es defineix la direcció de memòria de l'arxiu com a la sortida de text i hi enviem els paràmetres i la descripció entrats amb la funció.

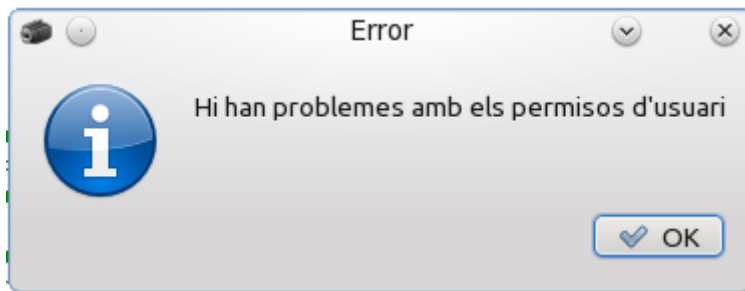
QMessageBox és la classe per a mostrar diàlegs amb botó d'acceptació, per exemple si s'executa el codi anterior:

```

QMessageBox::information(this, "Error", "Hi han problemes amb els
permisos d'usuari");

```

El resultat és:



II-lustració 33 - Mostra cartell error permisos

La funció es defineix com a booleana per si es volen manipular els errors a la funció pare que la crida.

#### b. Importar els paràmetres d'un fitxer existent

Aquesta opció només està disponible des del menú horitzontal superior degut a que és ben visible ja que te una opció per si mateix:

La implementació de la funció està dividida en dues opcions, una que obre el fitxer i llegeix línia per línia i per cada una invoca la funció de processament. Els paràmetres detectats són escrits a la casella corresponent.

<pre>bool MainWindow::llegeixArxiuConf(QString arxiuConf)</pre>	<pre>void MainWindow::processaLineaConf(QString line)</pre>
<pre>QFile file(arxiuConf); if(!file.open(QIODevice::ReadOnly)) {     QMessageBox::information(0, "error", file.errorString());     return false; } QTextStream in(&amp;file); while(!in.atEnd()) {     QString line = in.readLine();     processaLineaConf(line); } return true;</pre> <p>-----FI DE CODI-----</p>	<pre>if (line.startsWith("#")) {return;}  if (line.startsWith("port=")) {     line.replace("port=", "");     ui-&gt;lineEdit-&gt;setText(line);     return; } if (line.startsWith("network_interface=")) { line.replace("network_interface=", "");     ui-&gt;lineEdit_2-&gt;setText(line);     return; } if (line.startsWith("friendly_name=")) { line.replace("friendly_name=", "");     ui-&gt;lineEdit_3-&gt;setText(line);     return; } if (line.startsWith("db_dir=")) {     line.replace("db_dir=", "");     ui-&gt;lineEdit_4-&gt;setText(line);     return; } if (line.startsWith("log_dir=")) {</pre>
<p>L'entorn Qt gràcies a la llibreria QFile té una forma senzilla de processar els fitxers amb aquestes característiques. Éls passos son: És defineix l'arxiu S'obre Es defineix l'arxiu com a entrada de la</p>	



classe QTextStream per a poder processar les línies de text de l'arxiu

Mentrestant no s'arribi al final de l'arxiu, es llegeix cada línia i es processa.

```
line.replace("log_dir=", "");
ui->lineEdit_5->setText(line);
return;
}
if (line.startsWith("inotify=yes")) {
    ui->checkBox->setChecked(true);
    return;
}
if (line.startsWith("inotify=no")) {
    ui->checkBox->setChecked(false);
    return;
}
if (line.startsWith("media_dir=")) {
    line.replace("media_dir=", "");
    ui->listWidget->addItem(line);
    return;
}
if
(line.startsWith("root_container=")) {
line.replace("root_container=", "");

    if (line=="B") {
        ui->comboBox-
>setCurrentIndex(0);
        return;
    }
    if (line=="P") {
        ui->comboBox-
>setCurrentIndex(1);
        return;
    }
    if (line=="M") {
        ui->comboBox-
>setCurrentIndex(2);
        return;
    }
    if (line=="V") {
        ui->comboBox-
>setCurrentIndex(3);
        return;
    }
    if (line==".") {
        ui->comboBox-
>setCurrentIndex(4);
        return;
    }
    return;
}
}
---- FI DEL CODI ----
```

Degut al format que tenen els fitxers de configuració del miniDLNA, processem les línies d'aquesta manera:

Si la línia comença pel títol d'un paràmetre més el símbol igual, esborra aquest text detectat i així a la línia només queda el valor del paràmetre que s'envia a la casella corresponent.

c. Modificar els paràmetres d'un arxiu existent fent una còpia de seguretat de l'arxiu de configuració existent.

La funció més llarga del programa és la d'aplicar els nous paràmetres al servidor multimèdia, ja que ha de fer les següents tasques:

Comprovar els paràmetres i passar-los a l'arxiu intermig. La manipulació de directoris i de fitxers és el més complicat, la comprovació de paràmetres com el port o saber el paràmetre que s'ha d'escriure en funció del tipus de característica escollida és un problema d'algorísmia senzill.

Esborra la còpia de seguretat antiga si existeix.

Crea la còpia de seguretat si existeix l'arxiu de configuració actual

Renombrar l'arxiu intermig a arxiu de configuració

Aturar el servidor si està en funcionament

Engegar el servidor aplicant els paràmetres nous

Per a les funcions comuns s'ha utilitzat una funció per a tots els processos o la funció general de l'aplicació. S'han definit funcions individuals per a:

Passar els paràmetres a l'arxiu intermig. (escriureParametres), ja que s'ha fet d'aquesta manera utilitzant un arxiu intermig, podem fer servir la mateixa funció que per a crear un nou arxiu.

Per a crear arxius, esborrar i renombrar s'ha fet servir la llibreria QFile de manipulació de fitxers.

<pre>/*Esborra la copia de seguretat antiga*/ /*Esborra la copia de seguretat antiga*/  /*Si existeix esborra - Si no esborra correctament, llença un missatge*/ if (QFile::exists("minidlna_old.conf")) {     if (!QFile::remove("minidlna_old.conf")) {  QMessageBox::information(this, "Informe error", "No esborrat minidlna_old.conf correctament");     } }  /*Creació de la còpia de seguretat de l'arxiu de configuració actual*/  /*Passar la ubicació dels logs i la configuració al directori especificat*/ defaultLocationlogs = ui-&gt;lineEdit_5- &gt;text();</pre>	<p>Les funcions de la classe QFile remove i rename ens serveixen per a fer aquestes gestions després de la comprovació de la existència dels fitxers.</p> <p>Si alguna cosa va malament en la creació del nou fitxer de configuració, l'arxiu intermedi s'esborrarà, per al apretar el botó aplica, el procés és crear el fitxer i afegir text, si no l'esborressim, acumularia paràmetres repetits.</p>
---	--

```
    /*Si existeix còpia (renombra) al
    directori nou - Si no còpia correctament,
    llença un missatge*/
    if (QFile::exists("minidlna.conf")) {
        if (!QFile::rename("minidlna.conf",
defaultLocationlogs + "/minidlna_old.conf")) {

QMessageBox::information(this, "Error backup",
"Backup no fet correctament, assegura't que
els noms dels directoris son correctes i
existeixen i tens permis per escriure-hi");
            return;
        }
    }
}
/*****/

/*Renobrem l'arxiu nou*/
/*****/

if (!QFile::rename("nou.txt",
defaultLocationlogs + "/minidlna.conf"))
{

QMessageBox::information(this, "Error",
"Ops, assegura't que els noms dels
directoris son correctes i existeixen i
tens permis per escriure-hi!");
    QFile::remove("nou.txt");
}
}
```

Els objectius de l'aplicació en quant al control del servidor MiniDLNA eren

**Administració del servidor:**

- Arrencar el servidor
- Aturar el servidor
- Resetejar el servidor amb els paràmetres nous

Les tres opcions s'han resolt d'una manera similar. La classe Qt QProcess és feta servir per a executar programes externs a l'aplicació i la comunicació amb ells. De la pàgina d'ajuda del MiniDLNA de la distribució Ubuntu extrèiem les comandes necessàries per a:

Arrencar el servidor:

```
minidlna -f /home/$USER/.minidlna/minidlna.conf -P  
/home/$USER/.minidlna/minidlna.pid
```

Aturar el servidor:

```
xargs kill </home/$USER/.minidlna/minidlna.pid
```

Aplicar els canvis a la base de dades:

```
minidlna -f /home/$USER/.minidlna/minidlna.conf -R
```

L'arxiu minidlna.pid és un arxiu opcional que crea el servidor MiniDLNA quan arrenca

S'ha preferit enlloc de executar la comanda per a aturar el servidor, aconseguir el numero de procés del fitxer minidlna.pid i executar la comanda per a enviar le senyal kill al procés.

La funció d'arrencar el servidor s'ha implementat de dues maneres segons un parametre booleà entrat. Segons si és true o false, el servidor arrenca rellegint o no la base de dades.

### Arrencar Servidor

```
/*  
/*****  
*/  
/*Arrenca el servidor  
*  
*True - Arrenca  
*False - Arrenca amb escaneig de la base de dades  
**/  
  
void MainWindow::arrencaServidor(bool opcio) {  
    if (QFile::exists(defaultLocationlogs + "/minidlna.conf")) {  
        QProcess *p = new QProcess();  
        if (opcio) {  
            p->start("/bin/bash", QStringList() << "-c" << QString("minidlna -f "  
/home/$USER/.minidlna/minidlna.conf -P /home/$USER/.minidlna/minidlna.pid"));  
            p->waitForFinished();  
        }  
        else {  
            p->start("/bin/bash", QStringList() << "-c" << QString("minidlna -f "  
+ defaultLocationlogs + "minidlna.conf -P " + defaultLocationlogs +  
"/minidlna.pid -R"));  
            p->waitForFinished();  
        }  
        sleep(1);  
        if (!QFile::exists(defaultLocationlogs+"minidlna.pid")) {  
            QMessageBox::information(this, QString::fromUtf8("Error  
inicialització del servidor"), QString::fromUtf8("Assegurat que el directori de  
treball (logs) sigui el correcte i que existeixi l'arxiu de configuració amb els
```

```

paràmetres necessaris"));
    }
}
else {
    QMessageBox::information(this,QString::fromUtf8("Error inicialització del
servidor"), QString::fromUtf8("Assegurat que el directori de treball (logs) sigui
el correcte i que existeixi l'arxiu de configuració amb els paràmetres
necessaris"));
    }
    barraStatus();
}

```

El procés és similar, és crea una instància procés, es formata en una cadena QString i s'executa. Com es pot veure hi ha una funció (waitforfinished()) que fa que no es continuï l'execució fins que el procés no s'acabi.

Es veu avall les funcions necessàries per tal de saber el pid emmagatzemat a l'arxiu minidlna.pid.

Definir la ruta al fitxer, obrir el fitxer en mode lectura, llegir la línia i a parir d'aquí, formatar l'ordre.

#### Atura el servidor

```

void MainWindow::on_pushButton_3_clicked() {
    /*Definim el camí cap a l'arxiu que conté el PID del miniDLNA*/
    QString camiaPID = ui->lineEdit_5->text() + "minidlna.pid";
    /*Si l'arxiu existeix, aturem el procés*/
    if (QFile::exists(camiaPID)) {
        /*Definim l'arxiu sobre el qual operar*/
        QFile arxiuPID(camiaPID);
        /*Obrim l'arxiu a prova d'errors*/
        if (!arxiuPID.open(QIODevice::ReadOnly | QIODevice::Text)) {
            QMessageBox::information(this,"Error", "Arxiu PID no obert");}
        /*Llegim la primera i unica linea que conte el PID*/
        QTextStream in(&arxiuPID);
        QString nrpid = in.readLine();
        /*Definim un process intern per a executar una ordre*/
        QProcess *p = new QProcess();
        /*Executem l'ordre de matar el procés del servidor: kill pid "n° de
pid"*/
        p->start("/bin/bash",QStringList() << "-c" << QString("kill pid "+
nrpid));
        p->waitForFinished();
        /*Mostrem a la barra de status que hem aturat*/
        barraStatus();
    }
}

```

```

    }
    else {
        QMessageBox::information(this,QString::fromUtf8("Procés no aturat"),
        QString::fromUtf8("No hi havia no no s'ha trobat proces actual"));
        barraStatus();
    }
}

```

La funció d'aplicar els canvis ja s'ha explicat en els objectius del tractament del fitxer de configuració.

## Base de dades

Visualitzar la base de dades en un format intel·ligible (camí al fitxer, artista, títol...) que permeti veure els arxius que hi ha al servidor.

Aplicar els canvis als directoris del servidor

Eliminar fitxers de la base de dades

El Qt provés d'una classe anomenada QTableView que serveix per a visualitzar tables extretes d'un model definit. En els cas d'una taula SQLite, els passos a seguir per a poder visualitzar una comanda SQL a una instància QTableview.

Connectar la base de dades:

Definir el camí a la base de dades

Definir el tipus de base de dades

Obrir la base de dades

Definir el model i visualitzar-lo

Definir el model com un model d'ordre SQL

Definir l'ordre SQL

Connectar el model amb la vista de taula.

Les ordre posteriors de formatació de la taula són estètiques.

Per tal de fer aquests processos, s'han de definir dues instàncies de classes addicionals del Qt:

QSqlDatabase, per a definir i tractar la base de dades.

QSqlQueryModel, per a definir l'ordre SQL que volem com a model.

Codi:

```

void MainWindow::connectabdd() {
    QString camibdd = ui->lineEdit_4->text() + "/files.db";
    /*Comprova que existeix la base de dades*/
    if (QFile::exists(camibdd)) {

```

```

        /*Tipus de base de dades*/
        QSqlDatabase basedades = QSqlDatabase::addDatabase("QSQLITE");
        /*nom de la base de dades*/
        basedades.setDatabaseName(camibbdd);
        /*Obre la base de dades*/
        if (basedades.open()) {
            /*Informe progrés, es pot esborrar*/
            QMessageBox::information(this,"Informe progrés", "BBDD oberta");
        }
        else {
            QMessageBox::information(this,"EEEEII", "BBDD no oberta, si ");
        }
        /*Apliquem una ordre SQL i la introduim a la vista de tabla*/
        QSqlQueryModel *model = new QSqlQueryModel(this);
        model->setQuery("SELECT [TITLE],[ALBUM],[MIME],[PATH] FROM [DETAILS] WHERE MIME
<>'");
        ui->tableView->setModel(model);
        ui->tableView->resizeColumnsToContents();
        ui->tableView->setAlternatingRowColors(true);
    }
    /*Si no existeix al camí per defecte, demana la introducció del camí correcte*/
    else {
        QMessageBox::information(this,"Informe error", "La base de dades no està al
directori habitual, entra el camí correcte a la pestanya de configuració");
        return;
    }
}
}
}

```

Aplicar els canvis als directoris del servidor

La funció del botó d'actualitzar la base de dades el que fa és un *reset* de la vista de la taula, crida a la funció d'aplicar els canvis als paràmetres i a la funció anterior de connectar la base de dades amb la vista de taules.

CODI:

```

/*Actualitza base de dades*/
void MainWindow::on_pushButton_8_clicked()
{
    /*Esborra la taula*/
    ui->tableView->reset();
    /*Aplica canvis*/
    on_pushButton_2_clicked();
    /*Mostra de nou la biblioteca*/
    on_pushButton_7_clicked();
}

```

Eliminar fitxers de la base de dades

Simplement treu el fitxer de la taula amb una funció que dona la classe QSqlViewTable.

#### Objectiu addicional 4 – Visualitzar i reproduir els arxius de la biblioteca

L'entorn Qt, disposa de les classes que implementen les llibreries Phonon per a permetre una visualització dels arxius multimèdia.

En l'aspecte gràfic, a l'aplicació s'ha implementat un objecte de la classe *videoPlayer* i un de la classe *seekSlider* per a poder visualitzar i controlar la reproducció.

S'han implementat les funcions més bàsiques d'un reproductor (Play, pause, stop), a traves de botons que criden a les funcions implementades en la classe *VideoPlayer()*.

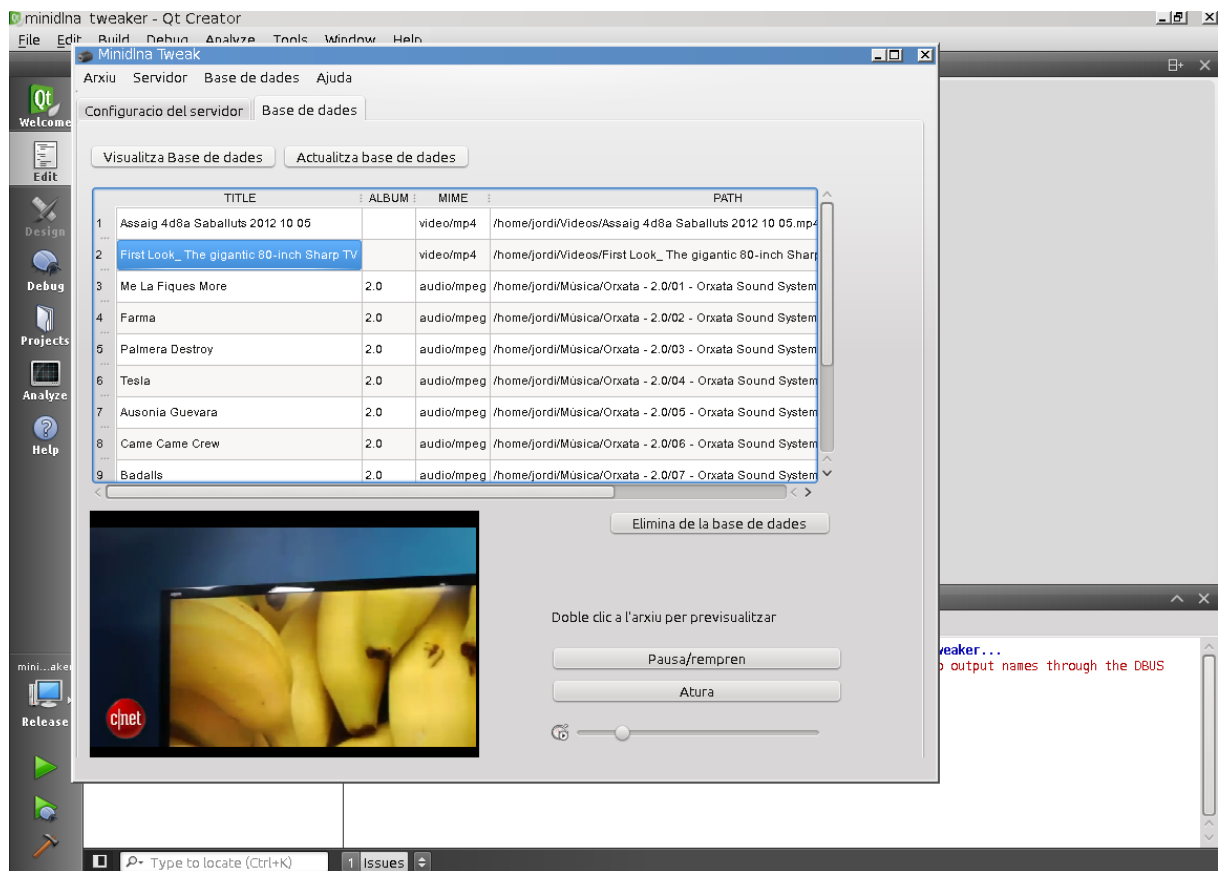
Aquí la dificultat rau en controlar l'objecte seleccionat de la taula i fer les conversions necessàries entre classes per a poder enviar al reproductor el camí a l'arxiu que es vol reproduir:

Codi de la implementació de la funció de reproducció quan es fa doble click a un objecte de la taula:

```
void MainWindow::on_tableView_doubleClicked(const QModelIndex &index)
{
    /*Si està reproduint primer para la reproducció actual*/
    if (ui->videoPlayer->isPlaying()) {
        ui->videoPlayer->stop();
    }
    /*Guardem el nuero de fila*/
    int fila = index.row();
    /*Mitjançant la classe QVariant, que es com un container de tipus, guardem el camí*/
    QVariant camiaMVariant = ui->tableView->model()->data(ui->tableView->model()-
>index(fila,3));
    /*Converti el camí a cadena*/
    QString camiaM = camiaMVariant.toString();
    /*Carreguem, reproduim i enllacem el control de progrès*/
    ui->videoPlayer->load(camiaM);
    ui->videoPlayer->play();
    ui->seekSlider->setMediaObject(ui->videoPlayer->mediaObject());
}
```



## Captura de reproducció:



II-lustració 34 - Captura pantalla aplicació en mode reproductor

## Traducció

L'Entorn Qt ofereix la possibilitat d'integrar la traducció de les aplicacions a les mateixes aplicacions.

Les modificacions que s'han de fer al codi font són senzilles. A les cadenes traduïbles se les ha d'embolcallar en un parèntesi `tr()`, per exemple, a la funció d'obertura de la base de dades:

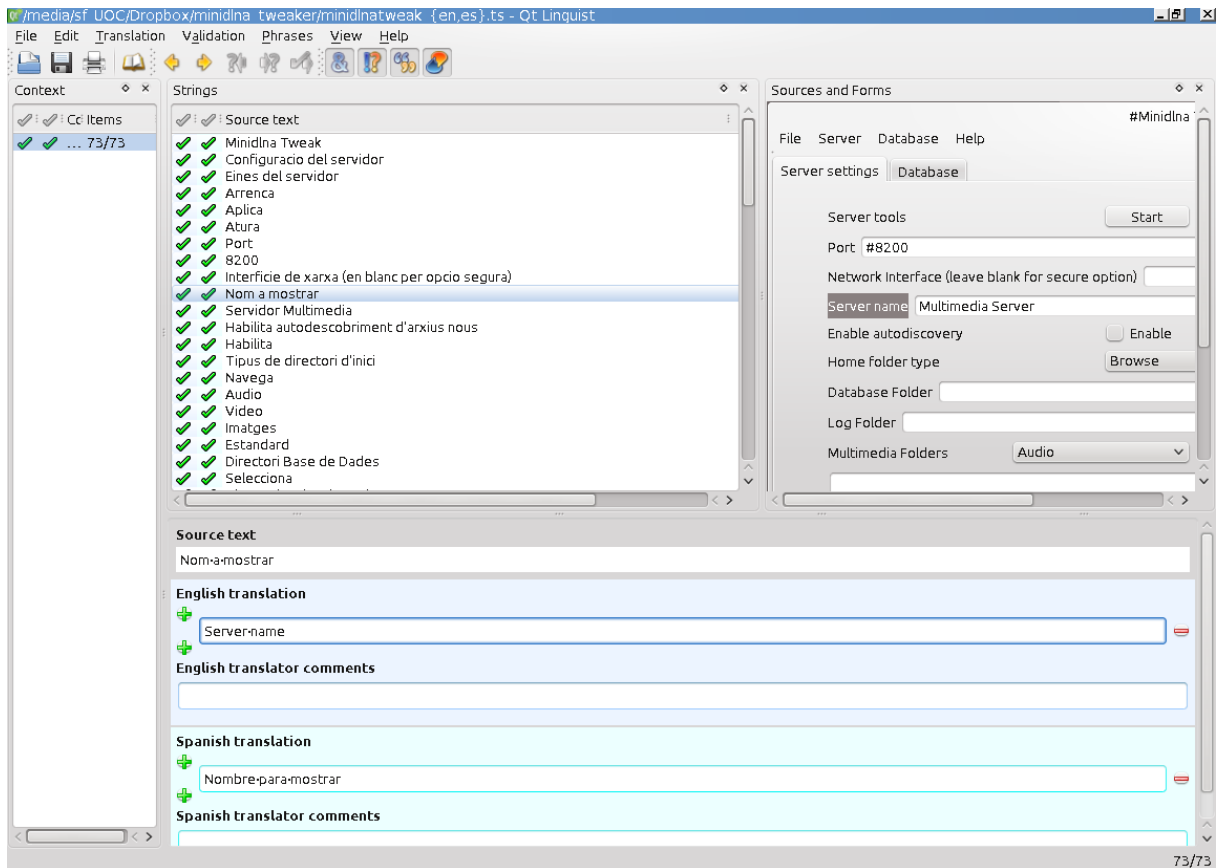
```
if (!basedades.open()) {
    QMessageBox::information(this, tr("Error"),    tr("Error a
l'obertura de la base de dades"));
}
```

Veiem que les cadenes “Error” i “Error a l'obertura de la base de dades”, les hem posat en un parèntesi `tr` que indica al programa que és una cadena traduïble.

Qt ofereix tres eines, Qt Linguist, `lupdate` i `lrelease` que ajuden a fer la traducció. Una vegada preparat el codi font i preparat també l'arxiu del projecte amb les traduccions que es volen fer:

```
TRANSLATIONS = minidlnatweak_es.ts \
..... minidlnatweak_en.ts
```

l'eina `lupdate` crea uns arxius XML de traducció que són llegibles per l'eina Qt Linguist, que mitjançant una eina gràfica, ajuda a fer la feina de traducció:



II-Il·lustració 35 - Captura pantalla Qt Linguist, eina de traducció del Qt

Quan ja estan fetes les traduccions, el QT Linguist et crea un arxiu \*.qm per a cada traducció que és el que l'arxiu busca per a cada llenguatge.

Per tal de saber el llenguatge correcte, es fa que el programa cerqui la localització de l'usuari i carregui el llenguatge correcte:

CODI:

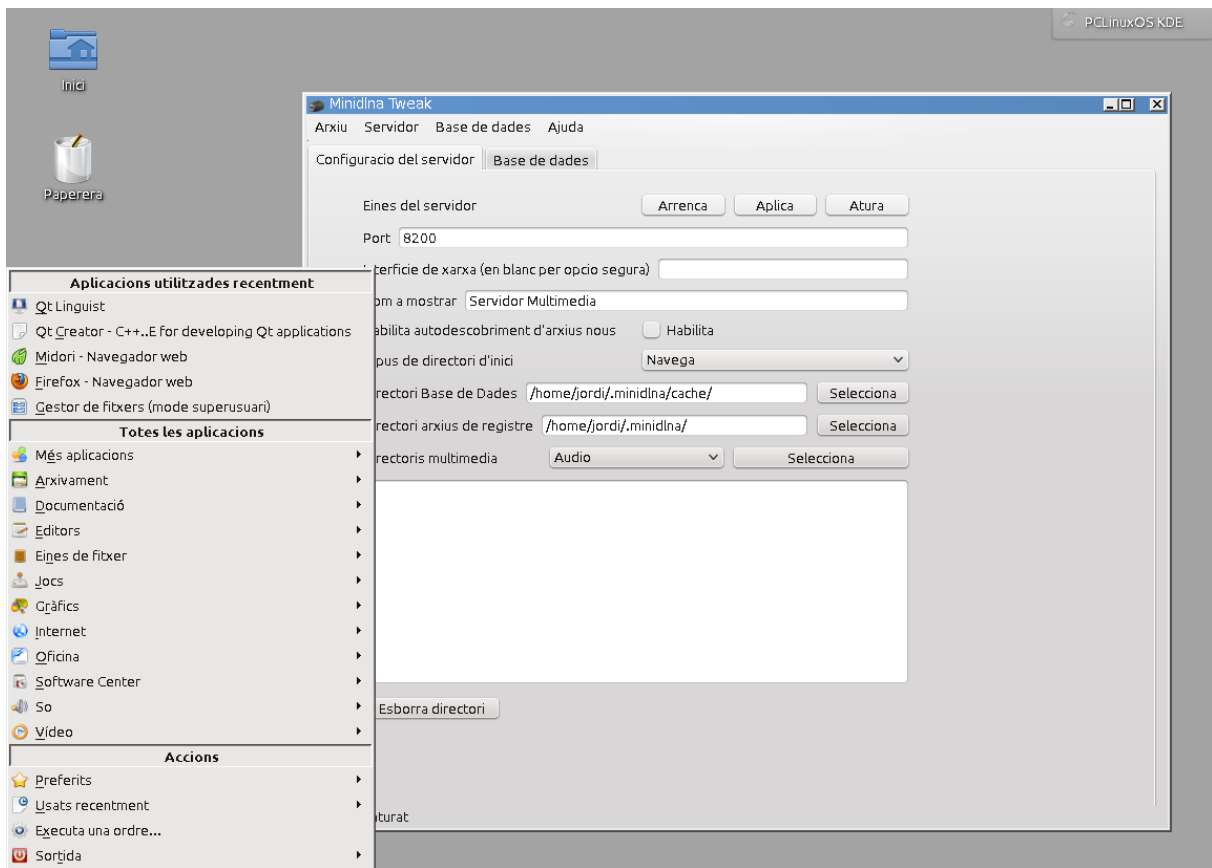
```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv); /*!<Crea l'aplicació "a". */

    QString locale = QLocale::system().name(); /*Cerca el locale de l'usuari*/
    QTranslator translator; /*Crea una instància translator*/
    translator.load(QString("minidlnatweak_")+locale); /*Carrega l'arxiu corresponent a
la localització*/
    a.installTranslator(&translator); /*instala l'arxiu carregat*/

    MainWindow w; /*!<Crea la finestra "w". */
    w.show(); /*!<Mostra la finestra "w". */

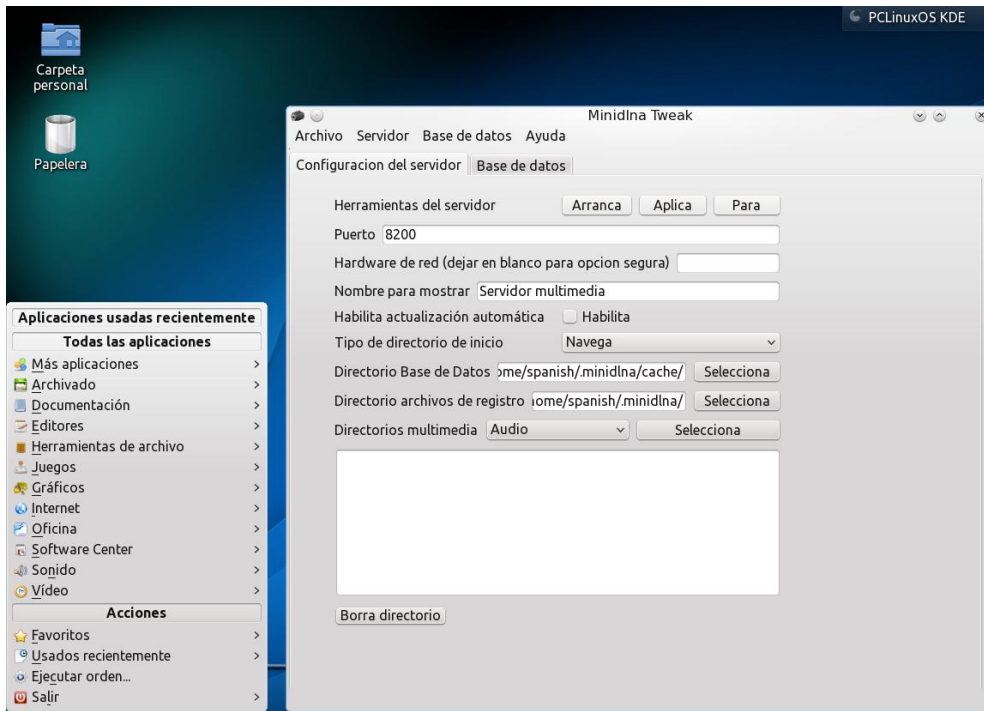
    return a.exec(); /*!<Executa l'aplicació "a". */
}
```

Captura de pantalla amb un usuari idioma català



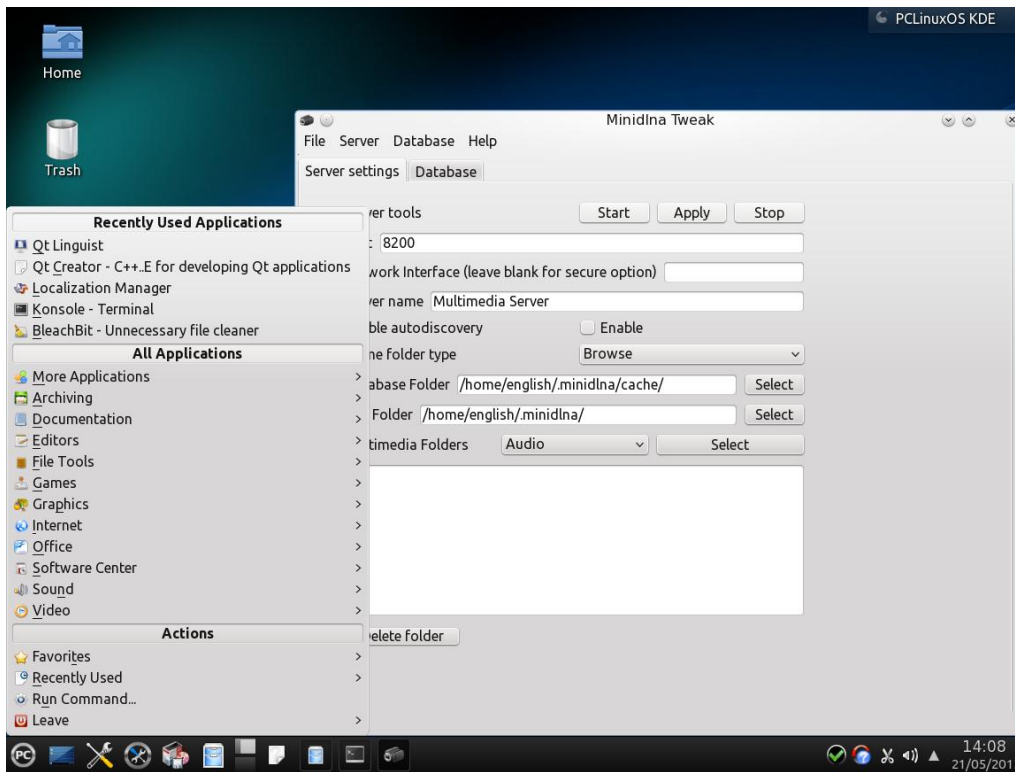
Il·lustració 36 - Captura pantalla aplicació idioma català

Captura de pantalla amb un usuari idioma castellà:



II-Ilustració 37 - Captura pantalla aplicació idioma castellà

Captura de pantalla amb un usuari idioma anglès:



II-Ilustració 38 - Captura pantalla aplicació idioma anglès

## Documentació

Per la documentació de l'aplicació s'ha fet servir l'eina Doxygen, que consisteix a comentar el codi font d'una manera específica per a que el programa Doxygen interpreti aquests comentaris per a crear unes pàgines de documentació HTML. També és necessari editar un document de configuració per que les opcions per defecte estan pensades per a un tipus d'aplicació que no es un treball d'universitat (ex. No crea documentació de funcions privades)

## Compilació i execució

### Execució:

Per a la correcta execució de l'aplicació, per dependències s'ha d'instal·lar el següent paquet (Ubuntu 12.04). Si es volen les instruccions per alguna altra distribució, contactar amb mi.

Per a una correcta execució com a usuari no com a servei global:

```
sudo apt-get install minidlna
sudo service minidlna stop
sudo update-rc.d minidlna disable
```

Per dependències del programa:

```
sudo apt-get install phonon libphonon4 libegl1-mesa-lts-quantal
```

Fer com a executable el fitxer:

```
chmod a+x minidlna_tweaker
```

I executar-lo:

```
./minidlna_tweaker
```

Si es te el sistema operatiu en castellà o anglès, si es tenen els arxius \*.qm a la mateixa carpeta, automàticament l'aplicació s'ha de visualitzar en aquests idiomes

## Compilació

La instal·lació del QT Creator a la maquina ja suposa tenir tota una sèrie d'eines i de llibreries per a poder compilar el programa. A part de les llibreries de desenvolupament de l'entorn Qt, s'han d'instal·lar per a poder visualitzar al Qt Creator i per a poder compilar les llibreries de desenvolupament del Phonon, l'entorn (framework) multimèdia del Qt, el paquet corresponent a la distribució de paquet phonon, la llibreria libphonon4 i les llibreries de desenvolupament phonon-devel.

Val a dir que per evitar problemes de dependències, s'ha provat de compilar l'aplicació de manera estàtica:

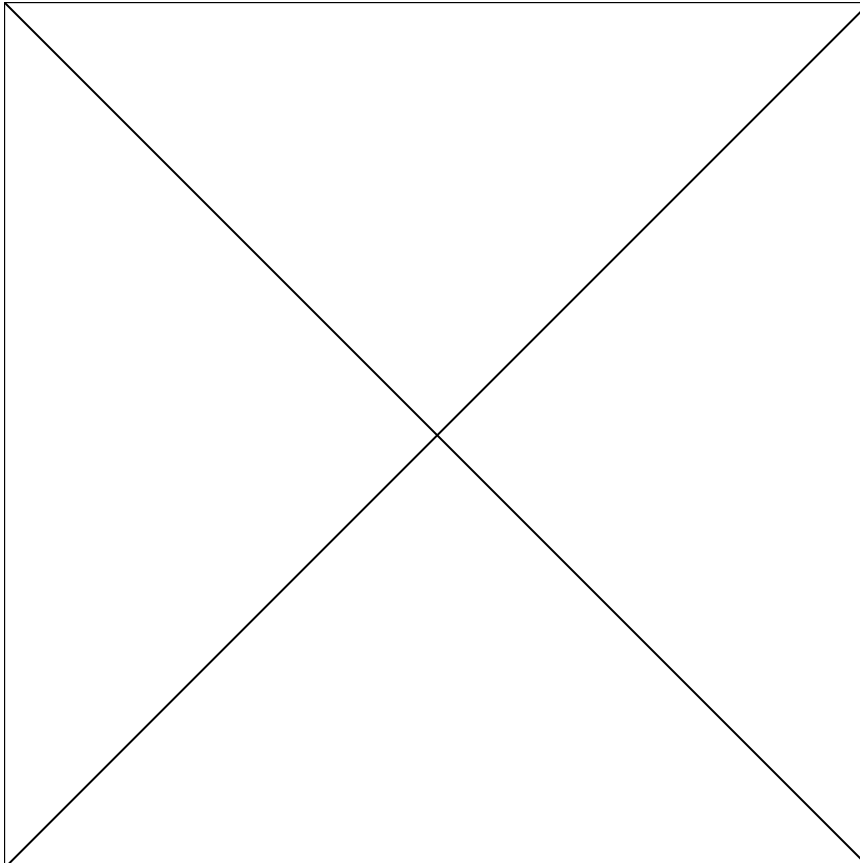
Després de compilar l'entorn Qt de manera estàtica i fer els canvis adequats al codi font, la compilació i l'execució no son optimes degut a que les classes i llibreries relacionades amb el control de les bases de dades SQL i l'entorn multimèdia Phonon no poden ser compilades estàticament a dintre de l'aplicació, sinó que en quant al SQL, s'han d'instal·lar com a plugin, feina aconseguida i de la qual es disposa de la implementació del codi font, però a hores d'ara encara no s'ha aconseguit com fer que l'aplicació disposi del reproductor de vídeo en compilació estàtica.

## Demostració

A continuació hi han dos vídeos per a demostrar el funcionament del programa, un per a comprovar que el programa funciona i un altre per demostrar que el servidor està funcionant.

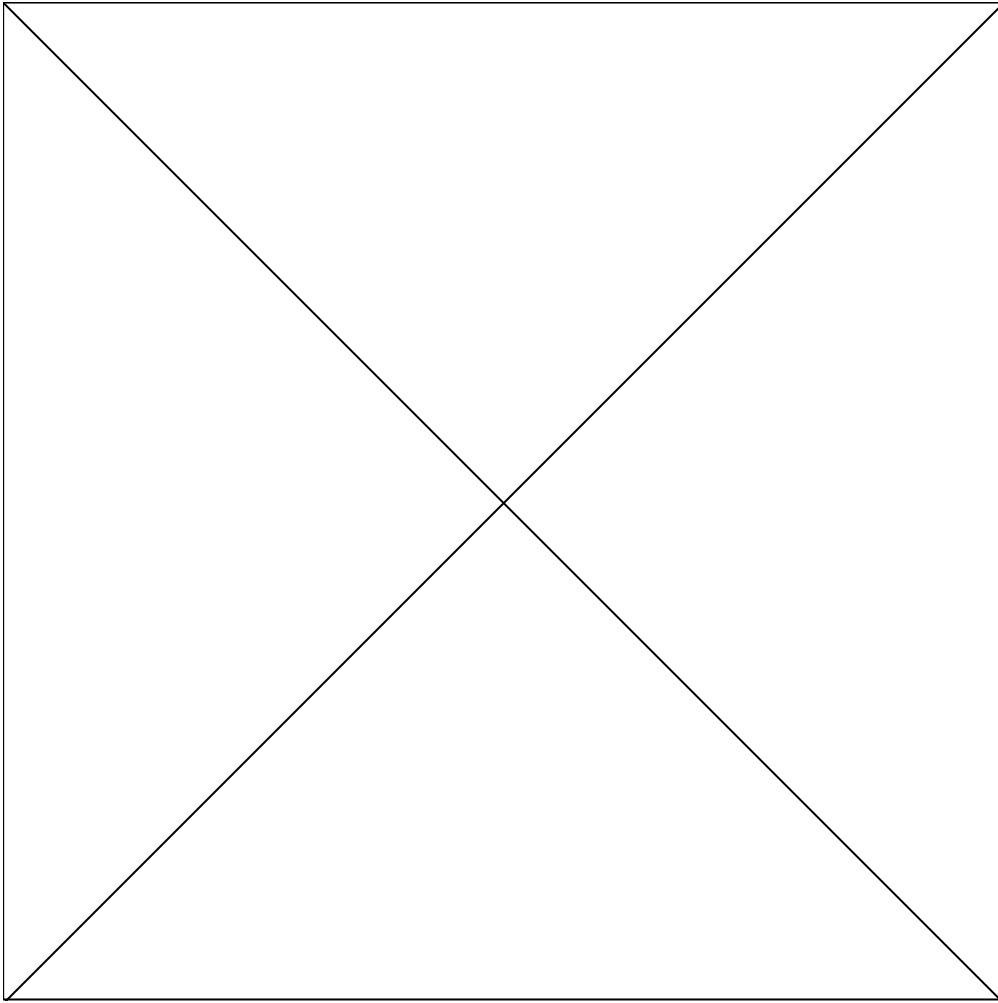
Les direccions són (si el vídeo encastat no funciona):

<https://www.youtube.com/watch?v=w0XyxbwyY54>





<https://www.youtube.com/watch?v=4WVhcW2DTBc>



## Conclusions i línies de futur

Les actuals tecnologies de les que disposa un desenvolupador d'aplicacions són molt complertes, essent molt senzill la programació d'una aplicació senzilla i no tenint-se que preocupar dels aspectes més bàsics de la programació d'aplicacions gràfiques, l'aparició d'eines popularment conegudes com WYSIWYG (*What you see is what you get* – El que veus és el que tindràs) en el camp de les aplicacions per a escriptori facilita enormement el treball de disseny de les aplicacions i la connexió entre l'aplicació gràfica i els seus components (botons, menús, llistes) i les funcions cridades per ells, havent-se de preocupar el desenvolupador dels aspectes més particulars de l'aplicació, l'enllaç entre els diferents components de l'aplicació, etc.

En particular, l'ús de les llibreries Qt ha permès no sortir d'aquest entorn en tota la programació de l'aplicació, només per a engegar els processos per a controlar el servidor s'han hagut de fer servir eines independents a l'entorn Qt, s'han tingut que obrir processos independents que cridaven a comandes Linux. Tant com en l'aspecte de la programació de codi font, on les eines són molt completes, com en els aspectes menys tècnics de l'aplicació com pot ser per exemple les diferents traduccions possibles de l'aplicació, l'entorn Qt s'ha mostrat molt complet. Queda per a un futur l'estudi de l'empaquetat de l'aplicació per a diverses distribucions i/o arquitectures de processador, no iniciat no per la complexitat, sinó per la quantitat de temps disponible.

La característica del llenguatge d'estar implementat en C++, fa que sigui fàcilment ampliable en les funcions i classes necessàries per al desenvolupament del projecte. L'entorn Qt ara es troba en un estat de transició, com gairebé totes les eines per a crear aplicacions, doncs les fronteres entre els dispositius d'escriptori i els dispositius mòbils són cada vegada més confuses. És per això que ara tot l'entorn Qt està més centralitzat en desenvolupar el llenguatge QML i les eines associades al seu desenvolupament i que sigui igual de senzill el desenvolupar aplicacions amb les llibreries Qt4 com amb el llenguatge QML. No és necessari explicar en aquest capítol en les causes, només dir que és un llenguatge vàlid per crear aplicacions per a escriptori com per a crear aplicacions per a dispositius mòbils. La possibilitat de crear aplicacions en que la compatibilitat entre sistemes operatius estigui garantida és un punt molt important en un futur, guanyant importància llenguatges com el QML o l'HTML5.

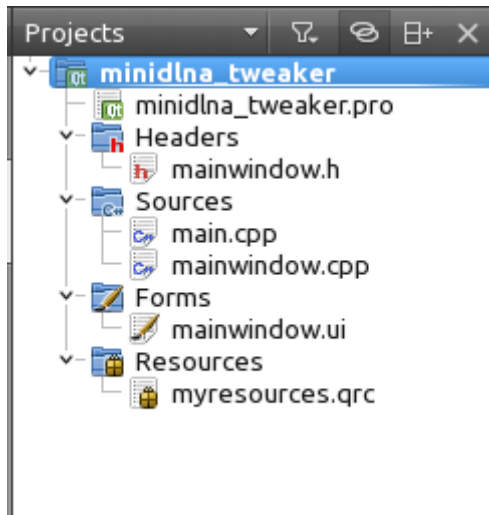
En aquest camp de la configuració de servidors per a la xarxa local, val a dir que és un camp en actual expansió, la possibilitat de interconnectar tots els dispositius locals de la llar a la mateixa biblioteca multimèdia és un concepte clarament a desenvolupar en els pròxims anys tant pels fabricants d'aplicacions per a configurar aquests servidors, com per als fabricants de dispositius finals. L'estàndard DLNA facilita enormement la tasca de la connexió una vegada els dispositius es troben en la mateixa xarxa local, i sembla que la xarxa wireless 804.11 (popularment wi-fi) en les seves variants també s'està convertint en estàndard per a la capes més baixes, encara que en aquest camp si que estan sortint protocols diferents (NFC, diferents implementacions del bluetooth...) que pot ser que facin variar aquests estàndards, ja que les implementacions per a crear les xarxes locals seran diferents.

Encara no està clar quin serà el tipus de dispositiu que en un futur implementi aquesta funció de servidor multimèdia, si serà més semblant a un ordinador d'escriptori, si serà els dispositius de reproducció ampliat amb més capacitat o si aquests serveis en un futur estaran directament allotjats en el "núvol" d'Internet, passant a ser tots els dispositius eines d'accés a aquest núvol.

## Fitxers entregats:

Document jordi122\_PAC4.pdf, aquest document.

El codi font i tots els arxius del projecte són al directori “src”, obrint al QT Creator l’arxiu QT es tindrà aquest arbre d’arxius:



El codi compilat és al directori “compilat”, hi és l’executable “minidlna\_tweaker”, els arxius “\*.qm” de traducció, i els arxius que genera el QT Creator a partir del disseny de la GUI

## Bibliografia – Recursos emprats

Documentació GTK+

Xfce Foundation Classes - <http://xfc.xfce.org/>

Xfce Desktop Environment - <http://xfce.org/>

Ubuntu app developer website - <http://developer.ubuntu.com/>

The Python GTK+ 3 Tutorial

Release 3.4 30/03/2013

GNU Free Documentation License

<http://python-gtk-3-tutorial.readthedocs.org/en/latest/>

Glade - A User Interface Designer - <http://glade.gnome.org/>

Gnome Library - <https://developer.gnome.org/>

Documentació Minidlna i codi Font:

Comunitat Ubuntu - <https://help.ubuntu.com/community/MiniDLNA>

Comunitat arch - <https://wiki.archlinux.org/index.php/MiniDLNA>

Projecte MiniDLNA - <http://sourceforge.net/projects/minidlna/>

Documentació QT

Qt Project Qt Creator - <http://qt-project.org/doc/qtcreator-2.7/>

Qt4 tutorial - <http://www.zetcode.com/gui/qt4/>

- C++ Qt tutorials a [www.voidrealms.com](http://www.voidrealms.com)  
(<http://www.youtube.com/playlist?list=PL2D1942A4688E9D63>)
- Documentació oficial Qt: <http://qt-project.org/doc/qt-4.8/>

Documentació SQLite

SQLite Home Page - <http://www.hwaci.com/sw/sqlite/index.html>

- The w3 schools - <http://www.w3schools.com/>

Documentació Zenity / Kdialogs

A complete zenity dialog examples - <http://linux.byexamples.com/archives/259/a-complete-zenity-dialog-examples-1/>

Zenity and KDialog - <http://www.linux-magazine.com/Issues/2009/99/Zenity-and-KDialog>

Manual del Zenity - <https://help.gnome.org/users/zenity/stable/>

Gui toolkit comparison: dialogs(zenity, kdialog) vs gtkdialog vs toolkits(qt, gtk) - <http://thepathprogramming.blogspot.com.es/2010/11/gui-toolkit-comparison-dialogszenity.html>

- **Linux**

- *Advanced Linux Programming*
  - *by Mark Mitchell, Jeffrey Oldham, and Alex Samuel, of CodeSourcery LLC*
  - *published by New Riders Publishing*
  - *ISBN 0-7357-1043-0*
  - *First Edition, June 2001*

Toolkits / Altres llenguatges

Fox Toolkit - <http://www.fox-toolkit.org/>

Fast Light Toolkit - <http://www.fltk.org/>

Erco's FLTK Cheat Page - <http://seriss.com/people/erco/fltk/>

WxWidgets - <http://www.wxwidgets.org/>

Altres

18 Herramientas para programar en GNU/Linux - <http://usemoslinux.blogspot.com/2012/09/18-herramientas-para-programar-en.html>

Difference between Qt and GTK Part One - <http://techfreaks4u.com/blog/posts/difference-between-qt-and-gtk/>

Cuestión de API's: GTK+ vs Qt vs WxWidget... - <http://softwarelibre.uca.es/node/822>

Decide que librería gráfica usar: wxWidgets vs GTK+ vs QT - <http://usemossoftwarelibre.wordpress.com/2011/04/28/decide-que-libreria-grafica-usar-wxwidgets-vs-gtk-vs-qt/>