



This is the **published version** of the bachelor thesis:

Ramon Escandell, Joan; Guichon Aguilar, Rodolfo Alberto, tut. Aplicación IPTV para Windows con soporte para M3U y Xstream Codes. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/317548>

under the terms of the  license

Aplicació IPTV per a Windows amb suport M3U i Xtream Codes

Joan Ramon Escandell

30 de juny de 2025

Resum— Aquest projecte presenta el desenvolupament d'una aplicació IPTV per a Windows capaç de reproduir continguts audiovisuals tant públics (l·listes M3U) com privats (a través d'APIs compatibles amb Xtream Codes). Implementada amb C# i WPF, inclou suport integrat per a fluxos HLS i MPEG-TS gràcies a LibVLCSharp. L'aplicació proporciona gestió multiusuari amb emmagatzematge segur de credencials i ofereix una interfície moderna i intuïtiva. Es tracta d'una eina totalment funcional, preparada per al seu ús real però oberta a futures millores en disseny i experiència d'usuari.

Paraules clau— IPTV, Xtream Codes, M3U, aplicació Windows, reproductor, streaming.

1 INTRODUCCIÓ

En l'actualitat el consum audiovisual a través d'Internet s'ha convertit en una pràctica quotidiana gràcies a la flexibilitat que ofereixen les plataformes digitals. En aquest context, la Televisió per Protocol d'Internet (IPTV) s'ha consolidat com una alternativa atractiva als models tradicionals de televisió per cable o satèl·lit, especialment per la possibilitat d'accedir a contingut diversificat i adaptable a múltiples tipus de dispositius i usuaris.

Aquest projecte sorgeix de la motivació personal de l'autor per explorar en profunditat el funcionament intern dels sistemes IPTV, així com per desenvolupar una aplicació pròpia capaç d'accedir de manera integrada tant a continguts accessibles públicament com a continguts privats que requereixen autenticació mitjançant APIs compatibles amb la plataforma Xtream Codes.

L'objectiu general d'aquest treball és desenvolupar una aplicació IPTV funcional per a Windows, programada en C#, amb una interfície moderna basada en WPF. Aquesta aplicació permetrà carregar i reproduir directament l·listes IPTV en format M3U/M3U8 i, alhora, gestionar connexions dinàmiques a servidors IPTV que utilitzin l'API d'Xtream Codes per accedir a contingut privat. A més, l'aplicació inclourà capacitats avançades, com ara la reproducció integrada de continguts en streaming gràcies a la biblioteca LibVLCSharp, i una gestió multiusuari eficient amb emmagatzematge segur de credencials.

Aquest document s'organitza en diverses seccions: la Secció 2 descriu l'estat de l'art actual en tecnologies IPTV, protocols rellevants i eines emprades en aquest projecte; la Secció 3 detalla la metodologia utilitzada, basada en l'enfocament àgil Kanban; la Secció 4 explica el procés concret

de desenvolupament per fases; la Secció 5 presenta l'arquitectura modular escollida; finalment, les Seccions 6 i 7 exposen els resultats obtinguts, així com les conclusions i possibles línies de continuació del projecte.

2 ESTAT DE L'ART

La IPTV (*Internet Protocol Television*) és una tecnologia de distribució de continguts audiovisuals que utilitza xarxes IP per transmetre canals de televisió en directe i contingut sota demanda. A diferència de la televisió tradicional, la IPTV permet una major flexibilitat en l'accés al contingut i una millor integració amb serveis interactius. Habitualment, el contingut IPTV es transmet mitjançant fluxos encapsulats en protocols específics com MPEG Transport Stream (MPEG-TS) o HTTP Live Streaming (HLS), que ofereixen una bona compatibilitat amb diversos dispositius i capacitat d'adaptació dinàmica segons les condicions de xarxa.

Una de les formes més habituals d'accedir a contingut IPTV és mitjançant l·listes de reproducció M3U o M3U8. Aquestes l·listes contenen enllaços directes als fluxos de vídeo en formats compatibles com HLS (.m3u8), MPEG-TS (.ts), o fins i tot formats d'arxiu estàndard com MP4 per a continguts sota demanda. En particular, el format M3U8 és molt popular en l'streaming en directe per la seva estructura segmentada, que permet ajustar-se dinàmicament a les fluctuacions de la xarxa. Pel que fa al contingut sota demanda, com sèries o pel·lícules, és habitual trobar formats com MKV (Matroska), MP4 o AVI, gràcies a la seva capacitat per emmagatzemar múltiples pistes d'àudio, subtítols i metadades.

En l'àmbit dels serveis IPTV privats, és molt comú l'ús de plataformes que gestionen les subscripcions i generen l·listats personalitzats automàticament mitjançant APIs específiques, com és el cas de Xtream Codes. Aquesta API REST permet autenticar usuaris i entregar-los l·listes M3U específiques segons el seu perfil de subscripció. Actualment, moltes aplicacions IPTV per a ordinadors i dispositi-

- E-mail de contacte: joanramonescandell@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Rodolfo Alberto Guichon Aguilar
- Curs 2024/25

tius mòbils ja ofereixen compatibilitat amb Xstream Codes, però poques combinen de manera senzilla l'accés simultani a llistes públiques (sense necessitat d'autenticació) i a contingut privat gestionat per Xstream Codes.

Des del punt de vista dels reproductors multimèdia, VLC és reconegut per la seva versatilitat i extens suport a gairebé tots els formats existents, així com per la seva capacitat per reproduir fluxos en directe i sota demanda sense complicacions. LibVLCSharp, una biblioteca per a l'entorn .NET basada en LibVLC, ofereix aquesta flexibilitat i robustesa a aplicacions desenvolupades amb tecnologies com C# i WPF. Malgrat que existeixen aplicacions com Kodi o SimpleTV, que suporten en certa mesura aquests usos, la majoria d'elles no ofereixen una interfície moderna amb una bona experiència d'usuari, ni integració fàcil amb l'API Xstream Codes al costat de continguts públics.

En aquest context, el present projecte té com a objectiu cobrir aquest buit existent: oferir una aplicació IPTV integrada, capaç de gestionar continguts tant públics (a través de llistes M3U lliures) com privats (mitjançant autenticació amb Xstream Codes), amb capacitat per reproduir en directe mitjançant protocols HLS, MPEG-TS i sota demanda en formats com MKV o MP4, tot amb la potència i flexibilitat proporcionada per LibVLCSharp i una interfície d'usuari moderna desenvolupada en WPF.

3 METODOLOGIA

Aquest projecte s'ha desenvolupat seguint una metodologia àgil, concretament utilitzant el mètode **Kanban**, caracteritzat per ser visual, adaptable i orientat a la gestió contínua del flux de treball. Kanban ha permès mantenir un control clar i directe sobre l'estat de les tasques, facilitant una identificació ràpida de possibles problemes o desviacions respecte als objectius inicials.

El procés es va gestionar mitjançant un **tauler Kanban digital creat amb Notion**, estructurat en columnes com ara: «Per fer», «En procés», «En revisió» i «Completades». A cada cicle o iteració (també coneguda com a "sprint"), es definien clarament els requisits funcionals a implementar, com per exemple la integració inicial amb Xstream Codes, el desenvolupament del reproductor multimèdia amb LibVLCSharp o la implementació del sistema multiusuari.

A mesura que avançava el desenvolupament, també es registraven en aquest mateix tauler les incidències, bugs o problemes tècnics que sorgien durant les proves i validacions internes. Aquestes tasques es prioritzaven i es resolien dins dels mateixos sprints, assegurant així una evolució contínua i estable del projecte.

Per garantir un correcte control de versions del codi font i facilitar la gestió dels canvis efectuats, s'ha fet ús d'un repositori al servei web **GitHub**. Aquest repositori permetia mantenir organitzades totes les modificacions i evolucions del codi, facilitant alhora la recuperació d'estats previs del projecte en cas necessari. A més, GitHub ha proporcionat un entorn ideal per documentar tècnicament el codi i registrar decisions rellevants sobre arquitectura o implementació.

Altres eines rellevants utilitzades en el desenvolupament del projecte inclouen:

- **Visual Studio**, com a entorn principal per al desenvolupament en C# i WPF.

lupament en C# i WPF.

- **Postman**, per validar ràpidament la connexió amb l'API d'Xstream Codes durant les primeres etapes del projecte.
- **Figma**, emprada en les fases inicials per crear i validar prototips de la interfície d'usuari abans d'implementar-la a WPF.

Aquest enfocament metodològic ha proporcionat flexibilitat suficient per adaptar el projecte a canvis imprevistos, així com per ajustar constantment els objectius en funció dels resultats obtinguts després de cada iteració

4 DESENVOLUPAMENT DEL PROJECTE

El desenvolupament del projecte s'ha estructurat en diverses fases iteratives, seguint un enfocament incremental que ha permès validar progressivament les funcionalitats implementades. Cada fase ha tingut uns objectius concrets i ha estat centrada en un conjunt delimitat de tasques tècniques, tal com es va planificar inicialment en el tauler Kanban utilitzat.

4.0.1 Fase 1: Anàlisi i disseny inicial

Aquesta fase inicial es va dedicar a establir els fonaments tècnics i funcionals del projecte. Es van definir els requisits principals del sistema, tant pel que fa a funcionalitats (càrrega de llistes M3U, connexió a Xstream Codes, reproducció) com a requeriments no funcionals (interfície intuïtiva, modularitat del codi, mantenibilitat).

En paral·lel, es va realitzar el disseny preliminar de l'arquitectura del sistema, basada en tres components principals: **Client**, **Backend** i **Core**. També es van crear els primers prototips de la interfície d'usuari amb Figma, i es va configurar el repositori GitHub per gestionar el codi font des del primer moment.

4.0.2 Fase 2: Implementació del nucli del sistema

Durant aquesta fase es van implementar les funcionalitats bàsiques que permeten carregar contingut IPTV. En concret, es van desenvolupar:

- La pantalla d'inici amb selector de font de contingut (M3U o Xstream Codes).
- El mòdul de càrrega de llistes M3U públiques i el seu processament per extreure canals.
- El sistema de connexió a servidors Xstream Codes amb autenticació via API REST.
- La visualització bàsica de categories i canals obtinguts des del servidor.

També es van realitzar les primeres proves de comunicació amb l'API d'Xstream Codes i es va validar que les dades (canals, sèries, pel·lícules) s'extreguessin correctament.

4.0.3 Fase 3: Reproducció i navegació

En aquesta etapa es va dotar l'aplicació de capacitat real de reproducció de continguts, tant en directe com sota demanda. Les principals tasques implementades van ser:

- La creació del reproductor integrat amb **LibVLCSharp**, amb suport per a fluxos `.m3u8`, `.ts`, `.mp4`, `.mkv`, entre d'altres.
- El mode de visualització a pantalla completa i controls de reproducció bàsics.

4.0.4 Fase 4: Multiusuari i optimitzacions

En aquesta fase es van implementar funcionalitats addicionals centrades en l'experiència d'usuari i en la gestió de múltiples comptes. El sistema va ser ampliat per permetre l'accés fàcil a diferents perfils d'usuari sense necessitat de reintroduir les credencials cada vegada.

Les principals tasques d'aquesta etapa van ser:

- Implementació de la gestió de comptes desats localment.
- Funcionalitat per afegir, seleccionar i eliminar comptes.
- Persistència segura de credencials a nivell local.
- Millores d'interfície per simplificar la navegació entre seccions (canals, pel·lícules, sèries).
- Optimització general del rendiment i correcció d'errors detectats en les fases prèvies. Aquesta fase va servir també per refinar la usabilitat de l'aplicació, millorant el temps de resposta i assegurant una navegació fluida en llistes amb gran volum de contingut.

5 ARQUITECTURA DEL SISTEMA

L'arquitectura del sistema s'ha dissenyat seguint un enfocament modular, amb una separació clara de responsabilitats entre els diferents components. Aquesta estratègia facilita el manteniment, l'escalabilitat i la realització de proves independents per a cada mòdul.

El sistema està dividit en tres capes principals: **JR IPTV.Client**, **JR IPTV.Backend** i **JR IPTV.Core**. El disseny modular es pot observar amb detall a l'apèndix (vegeu la Fig. 15).

5.1 JR IPTV.Client

Aquest component correspon a l'aplicació d'escriptori desenvolupada amb **WPF** (Windows Presentation Foundation) i **C#**. La seva estructura segueix el patró **MVVM** (Model-View-ViewModel), que separa la interfície d'usuari (*View*), la lògica de presentació (*ViewModel*) i les dades (*Model*), permetent un codi més net, testeigable i desacoblant.

El client inclou els següents elements:

- **Vistes (XAML)**: Interfícies gràfiques encarregades de la navegació i la interacció amb l'usuari.

- **ViewModels**: Gestionen la lògica de negoci associada a cada vista i actuen com a intermediaris amb les dades i serveis.
- **Serveis**: Realitzen la comunicació amb el Backend, analitzen llistes M3U i gestionen localment usuaris i sessions.
- **Reproductor de vídeo**: Basat en **LibVLCSharp**, reproduceix contingut IPTV amb protocols com HLS o MPEG-TS directament des dels servidors externs.
- **Emmagatzematge segur de credencials**: Les credencials dels usuaris es xifren localment amb **DPAPI** per assegurar-ne la confidencialitat.

Aquest mòdul representa el punt d'interacció principal amb l'usuari, gestionant tant la reproducció de continguts com la navegació per categories, cerques i seleccions de canals, sèries o pel·lícules. L'aplicació client es distribueix mitjançant instal·ladors estàndards o tecnologies com *ClickOnce*, facilitant-ne la instal·lació i actualització.

5.2 JR IPTV.Backend

Aquest component proporciona una capa d'abstracció entre el client i les fonts de dades remotes (servidors Xstream Codes). Està implementat amb **ASP.NET Core** i desplegat al servei al núvol **Render.com**. Actua com a intermediari per:

- Gestionar de forma segura les peticions d'autenticació amb l'API Xstream Codes mitjançant peticions **HTTPS**.
- Obtenir, estructurar i validar les dades (canals, categories, informació d'episodis) rebudes de l'API Xstream.
- Centralitzar la lògica associada a la connexió amb serveis externs per mantenir el client lleuger i protegir la integritat del sistema.

El Backend proporciona respostes coherents i codis d'estat HTTP adequats en cas d'errors, facilitant una gestió clara d'aquests per part del client.

5.3 JR IPTV.Core

Aquest component funciona com una biblioteca comuna compartida entre Client i Backend, i conté:

- Les entitats del domini (*usuari*, *canal*, *pel·lícula*, *categoria*, etc.).
- Classes DTO (Data Transfer Objects) per a l'intercanvi clar i consistent de dades entre capes.
- Classes de suport i validació comunes a tot el projecte.

El mòdul *Core* assegura la consistència del model de dades a tot el sistema i facilita que futures expansions (per exemple, una aplicació mòbil amb *Flutter*) reutilitzin directament aquestes estructures sense necessitat de modificar-ne el Backend o la lògica interna.

5.4 Flux de dades del sistema

Per il·lustrar millor com flueixen les dades dins del sistema, s'ha dissenyat un *Diagrama de Flux de Dades (DFD)*, que mostra les interaccions principals entre l'usuari, l'aplicació client, el backend i les fonts de dades externes (vegeu la Fig. 16). Aquest diagrama identifica clarament les següents interaccions:

- L'usuari pot introduir directament credencials per accedir al contingut privat via Xstream Codes o proporcionar un enllaç públic de tipus .m3u.
- En cas d'accedir amb credencials, el client autentica l'usuari mitjançant el backend, que al seu torn valida les dades contra Xstream Codes.
- Les dades rebudes (canals, categories, etc.) es retornen al client per a la seva visualització.
- La reproducció de vídeo en directe s'inicia des del client cap al servidor extern (Xstream) directament, sense passar pel backend, utilitzant els protocols HLS o MPEG-TS.
- En cas d'utilitzar un enllaç públic .m3u, el client obté les dades directament del servidor públic, parseja la llista i inicia la reproducció sense intervenció del backend.

5.5 Beneficis de l'arquitectura

Aquest enfocament modular aporta múltiples avantatges per desenvolupar i mantenir l'aplicació IPTV:

- **Escalabilitat:** És senzill afegir noves funcionalitats, integrar altres protocols IPTV o crear nous clients com aplicacions mòbils sense grans canvis estructurals.
- **Mantenibilitat:** Cada mòdul pot evolucionar de manera independent, facilitant el desenvolupament i les proves.
- **Seguretat:** La separació clara de capes i l'ús de mecanismes com HTTPS i DPAPI redueixen significativament els riscos de seguretat i garanteixen una gestió segura de credencials i dades sensibles.
- **Separació de responsabilitats:** L'ús del patró MVVM i la divisió modular redueixen la complexitat del codi i fan més clar l'abast de cada component.
- **Reutilització:** El mòdul *Core* pot ser reutilitzat directament en projectes futurs relacionats amb IPTV, reduint així l'esforç de desenvolupament.

En resum, l'arquitectura proposada garanteix una base sòlida, moderna i extensible per a una aplicació IPTV, assegurant alhora mantenibilitat, seguretat i facilitat d'expansió futura. La representació gràfica dels diferents aspectes de l'arquitectura i el flux de dades es pot consultar a les Fig. 15 i 16, disponibles a l'apèndix.

6 RESULTATS

En aquesta secció es mostren els resultats obtinguts després del desenvolupament de l'aplicació JR IPTV. A través de captures de pantalla es presenten les funcionalitats implementades per facilitar la comprensió dels resultats.

6.1 Pantalla d'autenticació i accés

La figura 1 mostra la pantalla inicial, on l'usuari pot iniciar sessió amb credencials Xstream Codes o enllaços públics M3U.

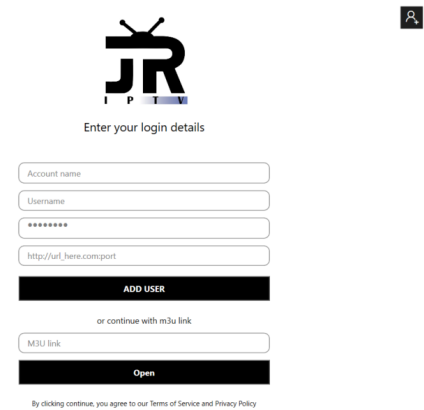


Fig. 1: Pantalla d'autenticació amb suport Xstream Codes i M3U públics.

6.2 Visualització de categories M3U

La figura 2 mostra les categories generades a partir d'un enllaç públic M3U.

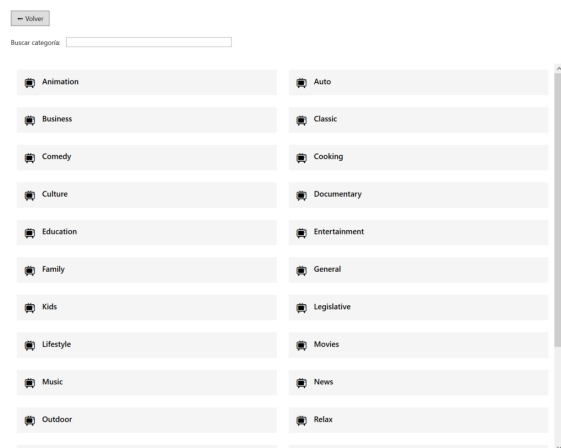


Fig. 2: Categories de canals IPTV generades des d'un fitxer M3U públic. L'enllaç utilitzat en aquest exemple és: <https://iptv-org.github.io/iptv/index.category.m3u>.

6.3 Reproducció de canals (M3U)

La figura 3 il·lustra la reproducció d'un canal seleccionat de tipus M3U, mitjançant el reproductor integrat LibVLCSharp.

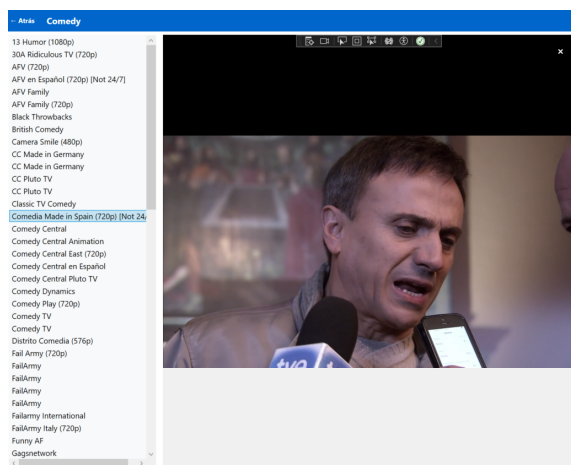


Fig. 3: Reproducció d'un canal IPTV des del fitxer M3U.

6.4 Gestió de comptes d'usuari

La figura 4 permet gestionar comptes IPTV guardats.

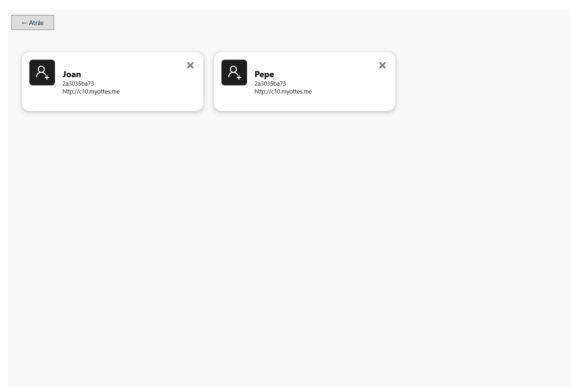


Fig. 4: Pantalla de gestió de comptes d'usuari.

6.5 Pàgina principal de continguts

La figura 5 ofereix accés directe als diferents tipus de contingut IPTV.

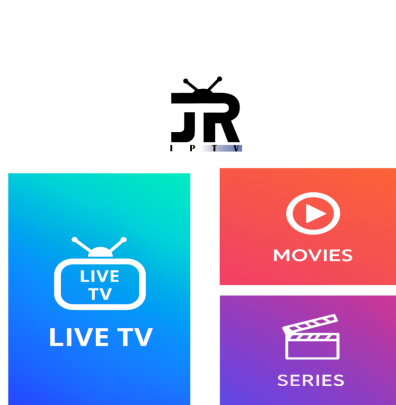


Fig. 5: Accés ràpid als diferents tipus de contingut IPTV.

6.6 Categories de TV en directe

La figura 6 mostra les categories de TV en directe disponibles.

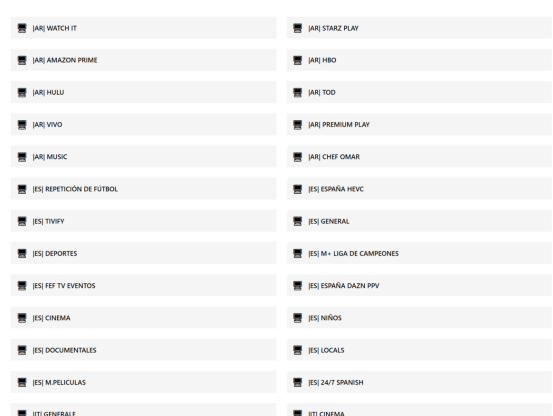


Fig. 6: Categories de TV en directe des de Xstream Codes.

6.7 Reproducció canals TV en directe

La figura 7 il·lustra la reproducció d'un canal de TV en directe.

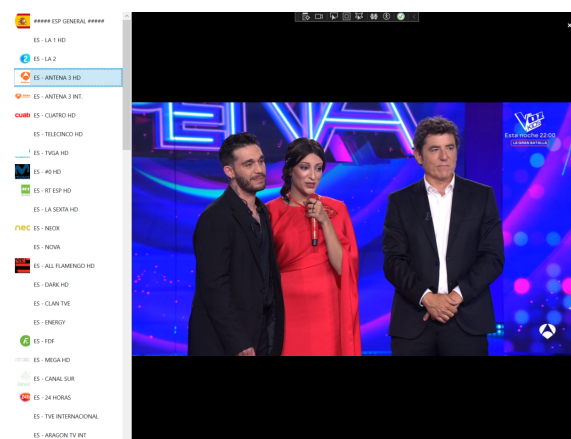


Fig. 7: Reproducció d'un canal de TV en directe.

6.8 Categories de contingut VOD

La figura 8 mostra les categories de contingut sota demanda disponibles.

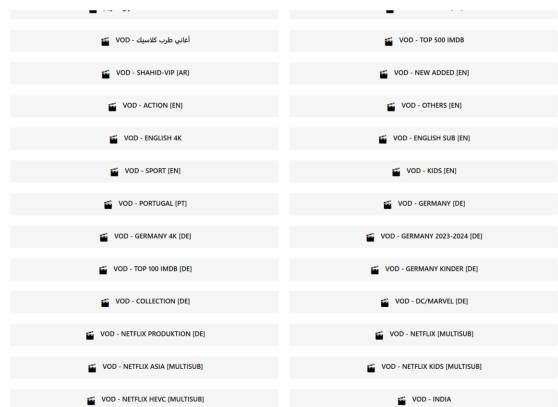


Fig. 8: Categories VOD obtingudes des de Xstream Codes.

6.9 Catàleg de pel·lícules VOD

La figura 9 presenta una vista en graella amb el contingut VOD disponible.

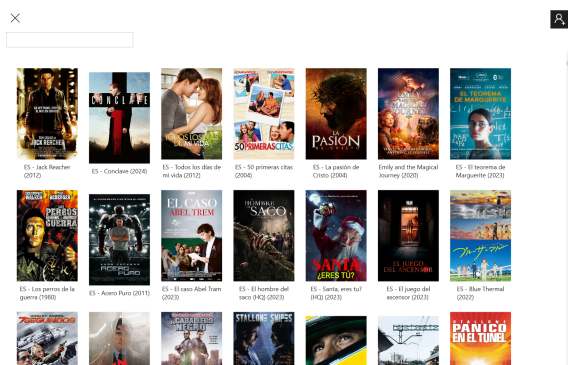


Fig. 9: Graella de pel·lícules VOD.

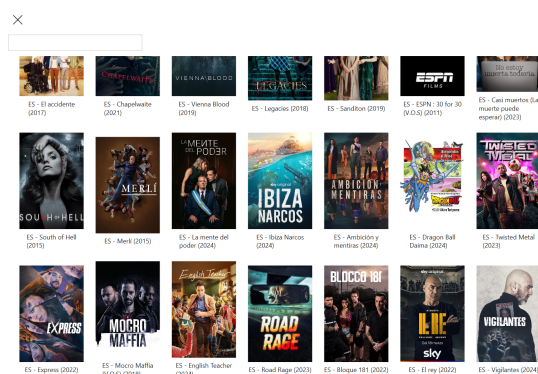


Fig. 12: Graella de sèries IPTV disponibles.

6.10 Detalls d'una pel·lícula

La figura 10 mostra detalls d'una pel·lícula, incloent la seva sinopsi i accés a la reproducció.

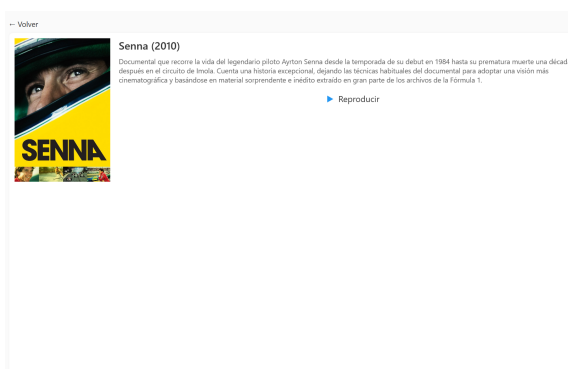


Fig. 10: Detalls d'una pel·lícula.

6.11 Categories de sèries IPTV

La figura 11 mostra categories disponibles per a contingut de sèries.

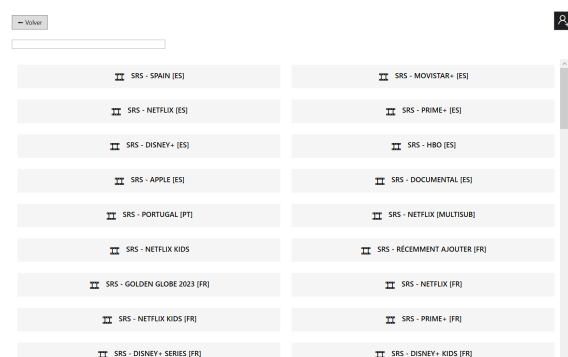


Fig. 11: Categories per a sèries IPTV.

6.12 Catàleg de sèries

La figura 12 presenta la vista en graella del contingut de sèries disponibles.

6.13 Detalls d'una sèrie

La figura 13 mostra informació detallada i els episodis d'una sèrie específica.

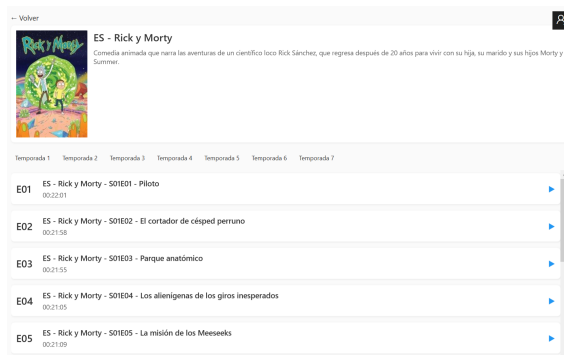


Fig. 13: Detalls i episodis d'una sèrie.

6.14 Reproducció de contingut IPTV

La figura 14 mostra el reproductor integrat utilitzat per a la visualització tant de pel·lícules com d'episodis de sèries. Aquest reproductor es basa en LibVLCSharp i permet una experiència fluida i compatible amb múltiples formats de transmissió.



Fig. 14: Reproductor integrat per a la visualització de contingut IPTV.

7 CONCLUSIONS

En general, estic molt satisfet amb el resultat d'aquest projecte, que ha estat tant un repte tècnic com personal. S'ha aconseguit desenvolupar una aplicació IPTV completament funcional, capaç de gestionar continguts públics i privats, amb suport per a la reproducció de canals en directe i contingut sota demanda.

El sistema implementa una arquitectura sòlida i escalable, i integra tecnologies modernes que garanteixen un bon rendiment i una gestió segura de les dades. Tanmateix, tot i que l'aplicació és plenament operativa, encara hi ha certs detalls de disseny i aspectes relacionats amb l'experiència d'usuari que m'agradaria millorar en el futur, especialment pel que fa a la coherència visual, la disposició de certs elements i la fluïdesa de la navegació.

Aquest projecte m'ha permès consolidar coneixements, aprendre eines noves i enfrontar-me a situacions reals de desenvolupament, i em deixa molt bones sensacions de cara a futurs reptes professionals.

AGRAÏMENTS

Vull expressar el meu agraïment més sincer al meu tutor, Rodolfo Alberto Guichon, per la seva guia, suport i disponibilitat al llarg de tot el desenvolupament del projecte. La seva orientació ha estat fonamental per poder dur a terme aquesta feina amb solidesa i confiança.

També voldria agrair el suport constant de la meua família i amics, que tot i la distància han estat presents en tot moment amb ànims i paraules d'encoratjament. En especial, un agraïment molt especial als meus companys de pis, que han tingut una paciència infinita escoltant-me parlar del projecte encara que moltes vegades no entenguessin ben bé de què tractava. Gràcies per ser-hi sempre.

REFERÈNCIES

- [1] S. Zeadally, H. Moustafa, and F. Siddiqui, *Internet Protocol Television (IPTV): Architecture, Trends, and Challenges*, IEEE Systems Journal, vol. 5, no. 4, pp. 518–525, Mar. 2025. Available: <https://doi.org/10.1109/JSYST.2011.2165601>
- [2] J. Maisonneuve *et al.*, *An Overview of IPTV Standards Development*, IEEE Trans. Broadcasting, vol. 55, no. 2, pp. 315–328, Apr. 2025. Available: <https://doi.org/10.1109/TBC.2009.2020451>
- [3] Wikipedia contributors, *M3U*, Wikipedia, The Free Encyclopedia, Feb. 2025. Available: <https://es.wikipedia.org/wiki/M3U>
- [4] VideoLAN, *LibVLC Documentation*, May 2025. Available: <https://videolan.videolan.me/vlc/libvlc.html>
- [5] Microsoft, *.NET Documentation – .NET Platform*, Apr. 2025. Available: <https://learn.microsoft.com/dotnet>
- [6] Microsoft, *Windows Presentation Foundation (WPF) for .NET – Documentation*, Mar. 2025. Available: <https://learn.microsoft.com/dotnet/desktop/wpf>
- [7] Microsoft, *Visual Studio IDE Documentation*, May 2025. Available: <https://learn.microsoft.com/visualstudio/>
- [8] Postman, *Postman Documentation Overview*, Feb. 2025. Available: <https://learning.postman.com/docs/getting-started/introduction/>
- [9] Figma, *Figma Design Platform – Official Website*, Mar. 2025. Available: <https://www.figma.com>
- [10] Atlassian, *What is Kanban? – Agile Project Management*, Feb. 2025. Available: <https://www.atlassian.com/agile/kanban>
- [11] R. Pantos and W. May, *HTTP Live Streaming*, IETF RFC 8216, Mar. 2025. Available: <https://tools.ietf.org/html/rfc8216>
- [12] Matroska Project, *Matroska Multimedia Container (MKV) – Homepage and Specifications*, May 2025. Available: <https://www.matroska.org>
- [13] Xstream Codes Forum (BlackDream), *[How-To] Player API v2 – Xstream Codes Official API Documentation*, Apr. 2025. Available: <https://github.com/engenex/xstream-codes-api-v2>
- [14] Microsoft, *Data Protection API (DPAPI) - Windows Security*, Mar. 2025. Available: <https://learn.microsoft.com/en-us/dotnet/standard/security/how-to-use-data-protection>
- [15] S. Zeadally, H. Moustafa, and F. Siddiqui, *Internet Protocol Television (IPTV): Architecture, trends, and challenges*, IEEE Systems Journal, Dec. 2011. Available: <https://doi.org/10.1109/JSYST.2011.2165601>
- [16] S. S. Saini and L. S. Sharma, *Investigation of the HTTP Live Streaming (HLS) media protocol's adaptability and performance*, Journal of The Institution of Engineers (India): Series B, Aug. 2024. Available: <https://doi.org/10.1007/s40031-024-01132-w>

A APÈNDIX

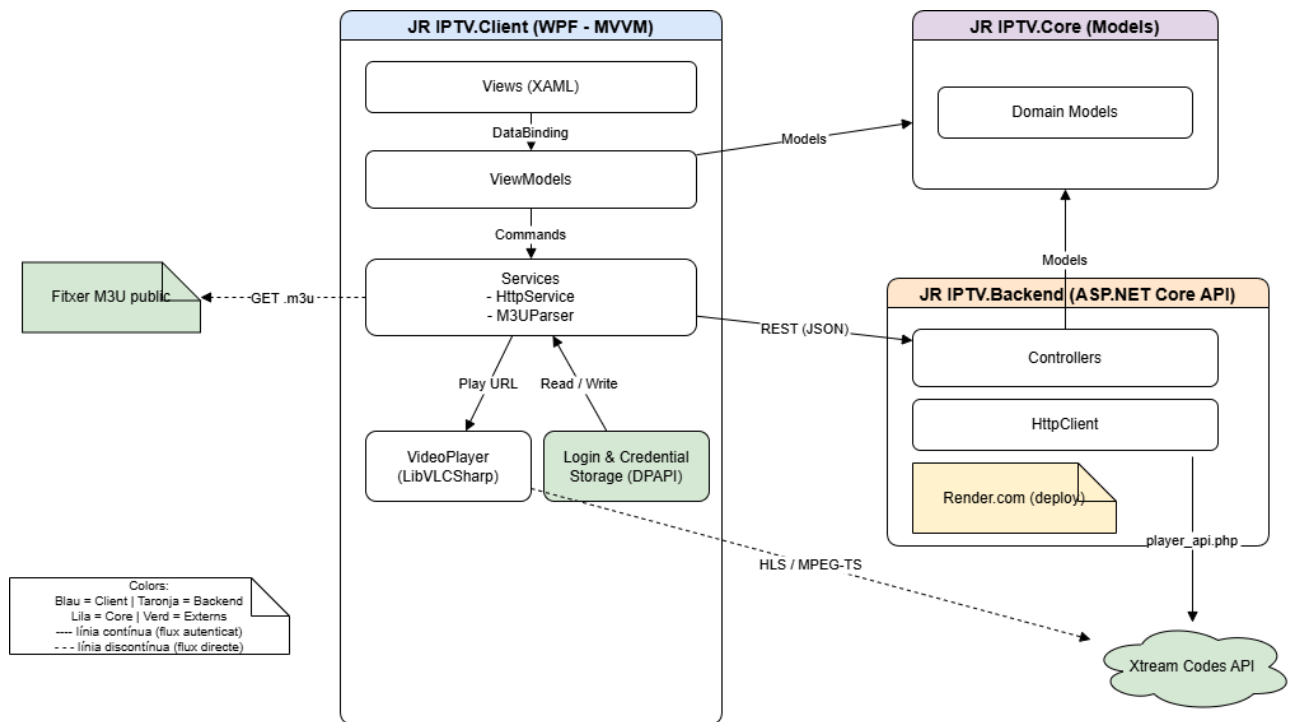


Fig. 15: Diagrama d'arquitectura modular de l'aplicació

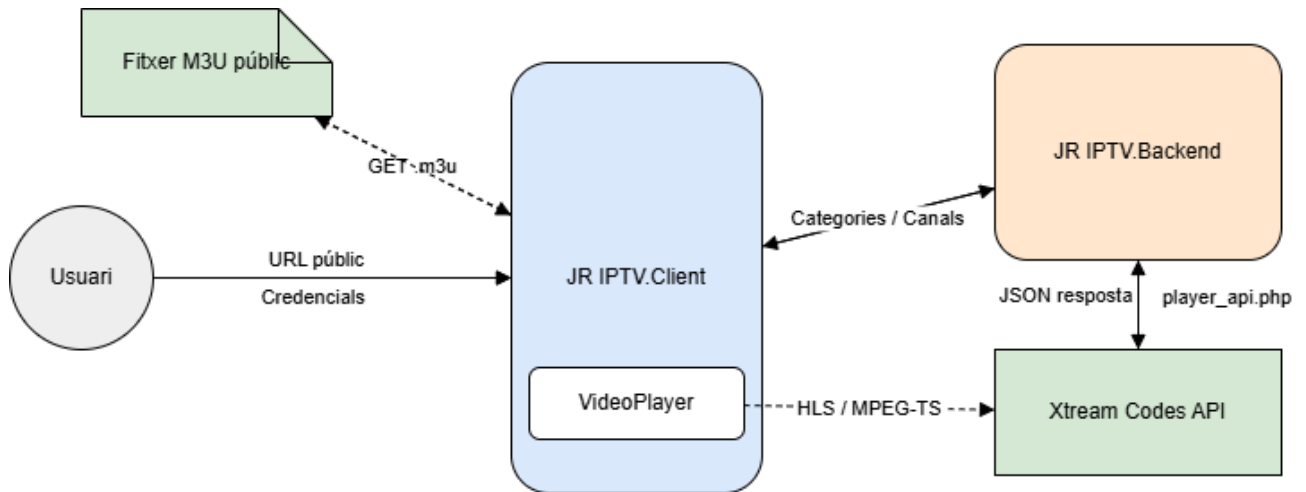


Fig. 16: Diagrama de flux de dades (DFD) del sistema JR IPTV