



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA APLICADA

## *Videoclub iPhone iStreamrTV*

PROYECTO FIN DE CARRERA

Javier Ramis Munilla

Director del proyecto: Javier Lluch Crespo  
Tutor de la empresa: Germán Renau

---

11-07-2010



## **AGRADECIMIENTOS**

Este trabajo no habría sido posible sin el apoyo de muchas personas que han contribuido con ideas, conversaciones y consejos. Mi más sincero agradecimiento a todos aquellos que me han ayudado a llevar adelante este proyecto.

Quiero expresar mi aprecio a la empresa Red Azul Multimedia SL por su confianza depositada y por brindarme esta oportunidad, en concreto a Juan Montilla, por escogerme para la realización del proyecto, y a Germán Renau, tutor del proyecto y de donde nació la idea. También mencionar a mis compañeros de la plataforma con los que he tenido que trabajar conjuntamente y de los que he ido aprendiendo día a día.

Por otro lado estoy enormemente agradecido con el profesor Javier Lluch Crespo por su aceptación como director del proyecto, por servirme de guía y por su dedicación y paciencia mostrada durante todo este tiempo.

Finalmente quiero dar las gracias a mi familia, amigos y novia por su apoyo incondicional durante toda mi etapa de estudios, a mis padres por hacer todo lo posible para que pudiera llegar a donde estoy y a mi hermano por iluminarme durante mis estudios y trabajos. Gracias por haber confiado en mis capacidades.

# TABLA DE CONTENIDOS

<b>1. INTRODUCCIÓN</b> .....	<b>9</b>
<b>1.1</b> Introducción.....	9
<b>1.2.</b> Problema y formulación conceptual.....	10
<b>1.3.</b> Objetivo .....	10
<b>2. CONCEPTOS PREVIOS GENERALES</b> .....	<b>12</b>
<b>2.1.</b> Situación de los videoclubs .....	12
<b>2.2.</b> Videoclub online StreamrTV .....	12
<b>2.3.</b> Contenidos multimedia .....	12
<b>2.4.</b> Evolución de los dispositivos móviles, un poco de historia... ..	13
<b>2.5.</b> iPhone y iPod Touch .....	14
<b>3. CONCEPTOS PREVIOS TÉCNICOS</b> .....	<b>17</b>
<b>3.1.</b> Aplicaciones nativas .....	17
<b>3.2.</b> Cocoa y Cocoa Touch.....	19
<b>3.3.</b> MVC.....	22
<b>3.4.</b> Amazon S3 .....	23
<b>4. ISTREAMRTV</b> .....	<b>24</b>
<b>4.1.</b> El proyecto .....	24
<b>4.2.</b> Usabilidad .....	25
<b>5. REQUERIMIENTOS</b> .....	<b>26</b>
<b>5.1.</b> Casos de uso .....	26
5.1.1. Actores .....	26
5.1.2. Acceso .....	27
5.1.2.1. Iniciar sesión .....	28
5.1.2.2. Cerrar sesión.....	29
5.1.3. Listado y gestión online.....	30
5.1.3.1. Listar películas en alquiler .....	31
5.1.3.2. Buscar película.....	32
5.1.3.3. Ver tráiler .....	34
5.1.3.4. Alquilar película.....	35
5.1.3.5. Votar película .....	36
5.1.4. Configuración y gestión offline .....	37
5.1.4.1. Descargar película .....	38
5.1.4.2. Ver película.....	39
5.1.4.3. Cambiar configuración .....	40
5.1.4.4. Enviar email a soporte.....	41
5.1.5. Periódicos.....	42
5.1.5.1. Borrar película .....	43
<b>5.2.</b> Glosario.....	44

<b>6. TECNOLOGÍAS .....</b>	<b>45</b>
<b>6.1. Lenguajes de programación .....</b>	<b>45</b>
<b>6.2. Interfaz de usuario .....</b>	<b>48</b>
6.2.1. NextSTEP Interface Builder .....	48
6.2.2. Interfaz mediante Objective-C .....	49
<b>6.3 Servlets y JSON .....</b>	<b>49</b>
<b>7. DISEÑO Y REALIZACIÓN .....</b>	<b>50</b>
<b>7.1 Arquitectura.....</b>	<b>50</b>
7.1.1 Cliente iStreamrTV .....	51
7.1.2. Conversor JSON .....	52
7.1.3. Servidor iStreamrTV .....	52
<b>7.2. Arquitectura de clases.....</b>	<b>53</b>
7.2.1. Controladores de vistas.....	53
7.2.2. Clases lógicas.....	54
7.2.3. Otras.....	55
<b>7.3. Diagramas de secuencia.....</b>	<b>56</b>
7.3.1. Iniciar sesión.....	56
7.3.2. Listar películas.....	57
7.3.3. Alquilar película.....	58
<b>7.4. Prototipo.....</b>	<b>59</b>
7.4.1. Navegación.....	59
7.4.2. Mi cuenta .....	60
7.4.3. Login .....	61
7.4.4. Destacados, género y búsqueda .....	62
7.4.5. Alquiler películas .....	63
7.4.6. Ver tráilers y votación .....	64
7.4.7. Descarga películas y visionado.....	65
7.4.8. Configuración y gestión de datos.....	68
<b>7.5. Iconos.....</b>	<b>70</b>
<b>8. TESTEO DEL SOFTWARE.....</b>	<b>71</b>
<b>8.1. Base de datos.....</b>	<b>71</b>
<b>8.2. Estructura iStreamrTV .....</b>	<b>72</b>
<b>8.3. Interfaz de usuario .....</b>	<b>72</b>
<b>9. COMPARACIÓN CON APLICACIONES SIMILARES.....</b>	<b>74</b>
<b>9.1. App Store .....</b>	<b>74</b>
<b>9.2. mSpot.....</b>	<b>75</b>
<b>10. POSIBLES MEJORAS .....</b>	<b>76</b>
<b>10.1. Encriptación/Desencriptación de películas .....</b>	<b>76</b>
<b>10.2. Redes sociales.....</b>	<b>76</b>
<b>10.3. iStreamrTV en el iPad.....</b>	<b>77</b>
<b>11. SUMARIO .....</b>	<b>78</b>

<b>12. MANUAL DE USUARIO .....</b>	<b>79</b>
<b>12.1. Mediante la App Store .....</b>	<b>79</b>
<b>12.2. Mediante el simulador .....</b>	<b>79</b>
<b>12.3. Mediante compilación sobre el dispositivo .....</b>	<b>80</b>
<b>13. SOFTWARE UTILIZADO .....</b>	<b>81</b>
<b>13.1. Software para la aplicación .....</b>	<b>81</b>
<b>13.2. Software para los servidores .....</b>	<b>82</b>
<b>14. BIBLIOGRAFÍA .....</b>	<b>83</b>
<b>15. ADJUNTOS.....</b>	<b>85</b>
<b>15.1 Listado de las ilustraciones .....</b>	<b>85</b>

## **Lista de abreviaturas y acrónimos**

### **3G**

Tercera-generación de transmisión de voz y datos a través de telefonía móvil.

### **App (Application)**

Aplicación (programa).

### **DVD (Digital Versatile Disc)**

Disco Versátil Digital, o también Digital Video Disc, es un dispositivo de almacenamiento óptico.

### **GPRS (General Packet Radio Service)**

Servicio general de paquetes vía radio, extensión del Sistema Global para Comunicaciones Móviles para la transmisión de datos no conmutada.

### **HTML (HyperText Markup Language)**

Lenguaje de Marcado de Hipertexto, es el lenguaje de marcado predominante para la elaboración de páginas web

### **HTTP**

Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la World Wide Web.

### **JSON (JavaScript Object Notation)**

Formato ligero para el intercambio de datos.

### **JSP (JavaServer Pages)**

JSP es un lenguaje de programación basado en Java que se ejecuta en el servidor.

### **OS (Operative System)**

Sistema Operativo.

**REDAMM**

Red Azul Multimedia, empresa proponente del proyecto.

**S3 (Amazon Simple Storage Service)**

Solución de almacenamiento infinito en Internet.

**SDK (Software Development Kit)**

Kit de desarrollo de software

**SQL (Structured Query Language)**

Lenguaje de consulta sobre bases de datos.

**UML**

Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software.

**VHS (Video Home System)**

Sistema de grabación y reproducción analógica



## 1. Introducción

### 1.1 Introducción

La evolución de la tecnología en los teléfonos móviles está teniendo mucho auge y desarrollo en estos últimos años desde sus inicios a finales de la década de los 70. Actualmente es más ya que un dispositivo sofisticado cotidiano, es un objeto cultural de pleno derecho. Y no sólo por su gran implantación social, sino porque en un sentido plenamente biográfico e identitario, forma parte del día a día en un entorno laboral o familiar, en un espacio público o íntimo. Paralelamente corre el desarrollo de una plataforma digital que demarca un territorio del mercado y del consumo, con su consiguiente reflejo en imágenes de marca, de productos.

Esta situación lleva a las empresas a invertir grandes cantidades de dinero y esfuerzo en aplicaciones que enriquecen la variedad y las posibilidades dentro del mercado. De esta forma se consigue motivar a los fabricantes de telefonía móvil en seguir desarrollando nuevos aparatos con mayores prestaciones provocando así un bucle en continuo crecimiento. En la última década, y con la implantación y consolidación de Internet gracias al nacimiento del GPRS y wifi, se ha abierto un nuevo abanico de posibilidades que todavía está por explotar.

Actualmente se encuentran infinidad de programas para móviles en la red donde según los estudios, sea en 2010 el año donde lleguen a duplicarse las descargas de aplicaciones y sigan aumentando hasta el 2012.

La motivación de este proyecto coincide con la revolución que está causando la llegada del iPhone y iPod Touch el mercado español, un dispositivo móvil pionero que permite novedosas incorporaciones en lo que a posibilidades en aplicaciones se refiere.

## 1.2. Problema y formulación conceptual

El proyecto fue propuesto por el departamento de tecnología de la empresa “Red Azul Multimedia SL”, REDAMM. Todos los requerimientos y especificaciones que el software ha necesitado han sido proporcionados por un pre-proceso, en orden de encajar y trabajar conjuntamente con la infraestructura ya existente de la plataforma de alquiler de vídeos online, *Streamr.tv*.

La plataforma *Streamr.tv* se trata de una página web dedicada a la distribución y difusión de productos multimedia. Mediante un contrato, alquiler, los usuarios pueden acceder al visionado de películas desde el propio aparato. Resumiendo, *Streamr.tv* se trata de un videoclub online.

El objetivo de esta memoria es la de mostrar el concepto, funcionamiento y arquitectura estructural de *iStreamrTV*, nombre del software creado para los dispositivos iPhone y iPod Touch con el que nos referiremos a él en adelante.

*iStreamrTV* nace de la idea de transferir y portar toda la estructura y funcionalidad de la página web a los dispositivos nombrados. Los contenidos alojados en servidores pueden utilizarse dado que se mantienen los estándares web en este tipo de aplicaciones, por lo que es posible manejar los datos sin problemas. De esta forma se aprovecha el éxito actual de esta tecnología para expandir el mercado de los productos ofrecidos en *Streamr.tv*.

Por tanto la principal tarea consiste en mantener las opciones ofrecidas previamente, así como la usabilidad, características que fácilmente se pierden en las portaciones de aplicaciones cuando se hace entre plataformas tan dispares.

## 1.3. Objetivo

Basándonos en lo comentado en el apartado anterior, *iStreamrTV* tiene distintos objetivos que cumplir que coinciden con las funcionalidades a transferir.

El proyecto realizado requiere de la creación de un interfaz programado para que los dispositivos móviles iPhone e iPod Touch puedan acceder a una serie de contenidos almacenados en servidores.

En primer lugar hay que tener en cuenta que la aplicación y servidores deben ser capaces de poder realizar una comunicación entre ambos mediante llamadas. Esto requiere un desarrollo e implementación de una adaptación sobre la infraestructura de los servicios aportando nuevas funcionalidades que se acoplen a los requisitos.

Mediante el acceso, que requiere de conexión a Internet, el usuario debe poder entrar al sistema donde podrá visualizar los tráileres disponibles gracias a la integración de un reproductor de medios para el dispositivo. Para ello tendrá la opción de poder acceder, listar y navegar por toda la información obtenida. A partir de aquí el usuario tendrá la posibilidad de registro que le permitirá acceder a mayor cantidad de contenidos y que a su vez, tendrá ligada una cuenta con un crédito disponible para la adquisición de los servicios ofrecidos. De esta

forma se podrán visualizar las películas alquiladas de forma íntegra mediante el reproductor integrado de manera sin ningún tipo de conexión a Internet. Finalmente, y dado que se trata de un alquiler, la película se eliminará una vez expire el tiempo contratado.

Con futuro desarrollo es posible incluir nuevas funcionalidades para hacer un software más complejo, con más opciones y más seguro. Se han tenido en cuenta estas ideas durante la programación para facilitar la tarea en el caso de incluirlas. Se explicarán con detalle más adelante en el capítulo "9. Posibles mejoras".

## 2. Conceptos previos generales

### 2.1. Situación de los videoclubs

Los videoclubs, como tienda física, sufren una situación de desaparición que se acentúa a medida que pasa el tiempo. Experimentó su mayor auge en la década de los ochenta con la aparición del VHS. En los noventa, sin embargo, con la aparición de las televisiones privadas, el consumo de vídeos descendió y ya entonces muchas de las tiendas cerraron. Con el cambio de formato de VHS a DVD se produjo un ligero rebrote pero es en el año 2006 cuando se produjo una reducción de un 70% de los alquileres en estos establecimientos, unos números delicados que provocan que las previsiones muestren que acabarán desapareciendo casi en su totalidad en los próximos años. Las causas podrían establecerse en:

- una gran aceptación por parte del público de la inclusión de contenido filmico en internet. En este apartado la piratería juega un papel importante dado que es entonces donde se registra un excesivo crecimiento, siendo los videoclubs los mayores perjudicados.
- la falta de innovación en el mercado. La negativa de los establecimientos hacia nuevas ideas para atraer al cliente ha provocado un estancamiento en el modo de operar.

### 2.2. Videoclub online StreamrTV

StreamrTV es una plataforma web en continuo desarrollo dedicada a la producción, distribución, difusión de productos multimedia y otros contenidos. En funcionamiento desde el año 2009, la página ofrece el alquiler de películas vía streaming convirtiéndose por tanto en un videoclub online.

Para poder contratar los servicios de la empresa es necesario registrarse en el sitio web (con dirección [www.streamr.tv](http://www.streamr.tv)) y acceder con la cuenta. A través de ingresos por tarjeta de crédito o por Paypal, es posible obtener créditos que se canjean por alquileres.

### 2.3. Contenidos multimedia

A día de hoy Internet se encuentra lleno de contenidos multimedia, dado que en resumen lo que los usuarios buscan. Más concretamente y refiriéndonos a la temática que nos abarca, el vídeo es uno de los principales recursos que se demandan. Se puede dividir en dos formas de acceder a ellos:

- el streaming o la distribución de vídeo de forma continua.
- descargas o transferencia completa de datos.

Aunque con distinta función, ambas opciones son empleadas frecuentemente en función de las necesidades y de otros parámetros. Aspectos técnicos, aspectos funcionales, aspectos legales, aspectos económicos... son algunos de los motivos que hacen decantarse entre una u otra.

El streaming o la distribución de vídeo de forma continua permite la reproducción del contenido sin descargas necesarias dado que se utiliza un búfer. Éste almacena el contenido que se mostrará dando la posibilidad de permitir al usuario de un visionado casi instantáneo provocando que, en caso de querer volver a reproducirlo, se tenga que volver a acceder al sitio en cuestión.

La descarga o transferencia completa de datos permite copiar el contenido de una fuente en el dispositivo y visionarlo completamente cuando se desee. Para que esto se produzca, es necesario esperar a que toda la información haya sido transferida.

#### **2.4. Evolución de los dispositivos móviles, un poco de historia...**

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales.

Casi cualquier dispositivo móvil actual permite funciones que van más allá de la comunicación básica, permitiendo el envío de contenido de verdadero valor. Las nuevas redes y la proliferación de servicios para la transmisión de datos lo hacen más accesible incluso.

La evolución de los dispositivos móviles desde su primera utilización en 1973 ha sido espectacular. El primer prototipo desarrollado por Motorola llamado "DynaTac", más comúnmente apodado "El Ladrillo" empezó a comercializarse una década después, El coste del terminal fue de entre 3.000 y 4.000 dólares y pesaba cerca de un kilo.



Figura 1 – Evolución del dispositivo móvil

Hoy en día podemos encontrar terminales de menos de 100 gramos con multitud de funcionalidades y con un coste muy bajo para las compañías. La revolución en este campo no obstante, ha sido la posibilidad de usar estos dispositivos para la transmisión de datos. Ello ha permitido integrar en este dispositivo multitud de aplicaciones que han mejorado notablemente la productividad para sus usuarios. Una de las funcionalidades más utilizadas en estos dispositivos es la recepción y envío de correo electrónico.

Compañías como Nokia Siemens están ya advirtiendo del riesgo de colapso de las redes por el motivo del alto consumo de datos que estos aparatos realizan. Este fabricante señala que el tráfico de datos móviles anual llegará a los 23 Hexabytes en 2015, lo que equivale a cerca de 6.300 millones de personas descargándose un libro digital cada día, cifras que indican claramente hacia donde se mueve la red.

### 2.5. iPhone y iPod Touch

Los dispositivos de Apple han causado una revolución en el mundo de la telefonía en los últimos años. El iPhone es un teléfono inteligente multimedia con conexión a internet a través de WIFI o 3G, pantalla táctil (con tecnología multitáctil) y una interfaz de hardware minimalista. Se puso a la venta en Estados Unidos el 29 de junio de 2007 llegando al resto de países de forma escalonada meses después. Por otro lado, el iPod Touch se muestra como un iPhone sin funcionalidad telefónica, por lo que está más orientado a la reproducción multimedia y que fue distribuido en el mes de septiembre de 2007.

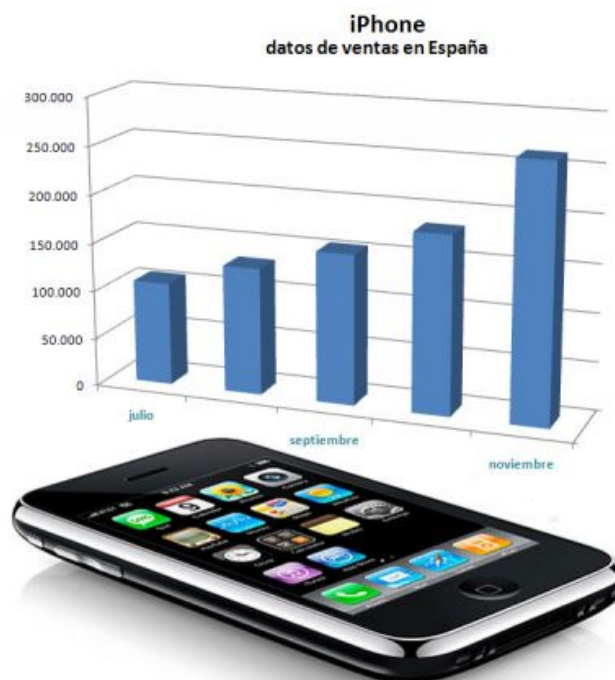


Figura 2 - Gráfico de ventas del iPhone en España en el año 2008

La gráfica de la figura 2 muestra el gran incremento de ventas de iPhone y iPod Touch solamente en España que ha seguido incrementándose, llegando incluso a duplicar las ventas entre meses posteriores.

Con datos internacionales, estas cifras son aún más espectaculares si tenemos en cuenta que se trata de un dispositivo con un sistema operativo recién salido al mercado y que compite con empresas que están en el negocio desde sus inicios.

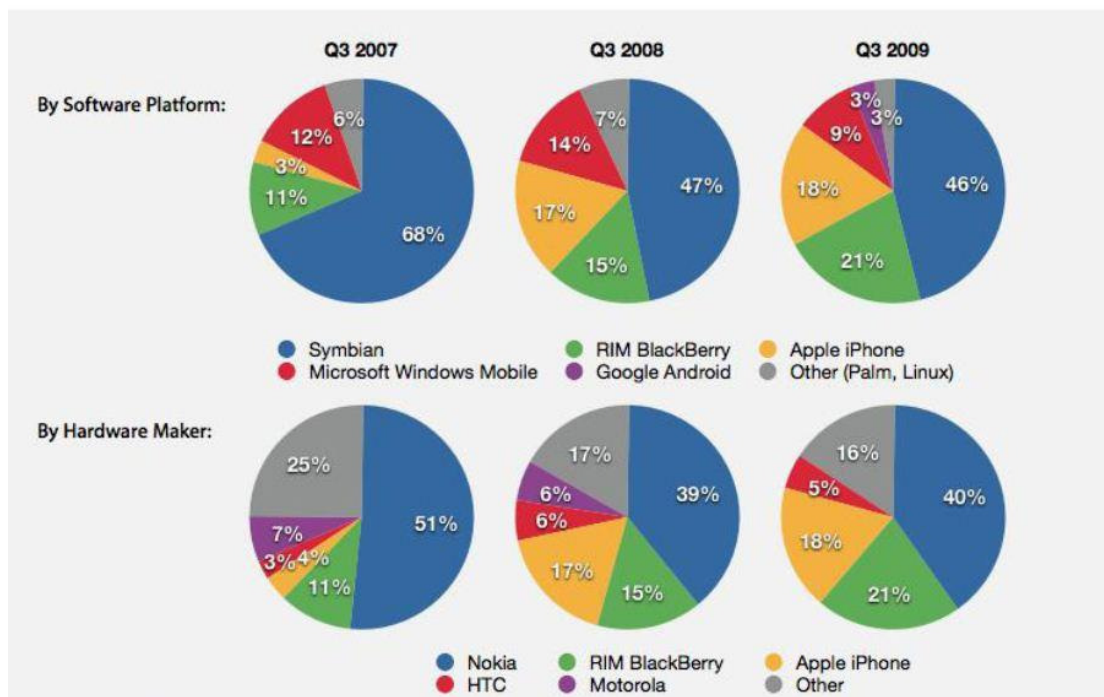


Figura 3 – Cuota del mercado global de dispositivos Smartphone durante los años 2007 y 2009

Con estos datos, junto a cifras récords como la venta de un millón de la primera versión dispositivos en su primera semana o de un millón y medio en la salida de la última versión hasta el momento (iPhone 4) también en la primera semana, hacen que se trate del producto que más ha triunfado en el sector.

Además del gran éxito, estos aparatos comparten unas características comunes que hacen apropiada la inserción de una plataforma para el videoclub online:

- Navegación por Internet mediante Wifi y 3G, necesaria para la navegación de la información localizada en Internet y la descarga de películas.
- Reproducción de vídeos en formato .mp4, donde con una recodificación previa de la videoteca, optimizada para la pantalla de los dispositivos, se hará la reproducción sobre éstos.
- Plataforma preparada para inserción y divulgación de aplicaciones gracias a la App Store, desarrollada por Apple, permite fácilmente la distribución de la aplicación.

## Videoclub iPhone iStreamrTV

- Las aplicaciones se encargan de la gestión de datos y ficheros (en vez del propio usuario), dando la posibilidad de mantener fuera del alcance el material alquilado, borrándolo una vez expira el periodo de alquiler.

	iPhone 8GB	iPod Touch 8GB
GSM/Edge	Yes	No
Wi-Fi	Yes	Yes
Bluetooth	Yes	No
Video output	No	Yes
Camera	Yes	No
Multitouch screen	Yes	Yes
Screen size	3.5 inch	3.5 inch
Screen resolution	480x320	480x320
Weight	135g	120g
Height	11.5 cm	11 cm
Width	6.1 cm	6.18 cm
Thickness	11.6 cm	8 cm
Operating system	OS X	OS X
PDA Apps	Yes	No
Battery - Audio	24 hours	22 hours
Battery - Video	7 hours	5 hours
USB port	Yes	Yes
Earphone connector	Recessed	Standard
Microphone	Yes	No
Price	\$399	\$299

Figura 4 – Características iPhone y iPod Touch



### 3. Conceptos previos técnicos

El desarrollo de software para el iPhone y iPod Touch supone una programación cerrada dada la política de Apple de estandarizarlas. Esto provoca que los programas creados tengan grandes similitudes en cuanto a aspecto y funcionalidad trayendo ventajas e inconvenientes. Como ventajas tenemos software unificado por unas reglas estrictas que permiten a los usuarios identificar fácilmente los elementos y llevar un mejor y mayor control. Por otro lado, como inconveniente tenemos que estas herramientas provocan algunas limitaciones en su funcionalidad que evitan exprimir un software más libre en cuanto a diseño y funcionalidad se refiere. Sin embargo, existen distintas alternativas para la gestión e interacción externa de los datos con los que se trabajan.

En este capítulo presentará las técnicas que han guiado a cómo la información se debe manejar y estructurar para dar solución al problema planteado.

#### 3.1. Aplicaciones nativas

Para el desarrollo de aplicaciones nativas es necesario el uso de Cocoa Touch, set de herramientas para crear programas sobre la plataforma Mac OS X y donde se incluye el SDK para iPhone. Su estructura es la siguiente:

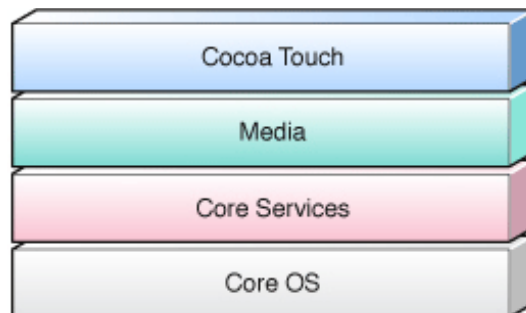


Figura 5 – Estructura de una aplicación nativa en iPhone

Los 4 elementos mostrados son todas las capas con las que trabaja el iPhone o iPod Touch, empezaremos explicando de abajo hacia arriba:

- **Core OS:** es la base o el núcleo del sistema operativo y es responsable, entre otras cosas, de gestionar los archivos, la memoria, seguridad y la comunicación.
- **Core Services:** provee los servicios fundamentales de la aplicación, por ejemplo puedes acceder a la lista de contactos, al SQLite o a las preferencias.
- **Media:** aquí está todo lo referente a Multimedia: vídeo, audio, archivos de imágenes, Core Animation, OpenGL, etc.

- **Cocoa Touch:** no es más que el Framework de desarrollo con el que se ha trabajado durante este proyecto.

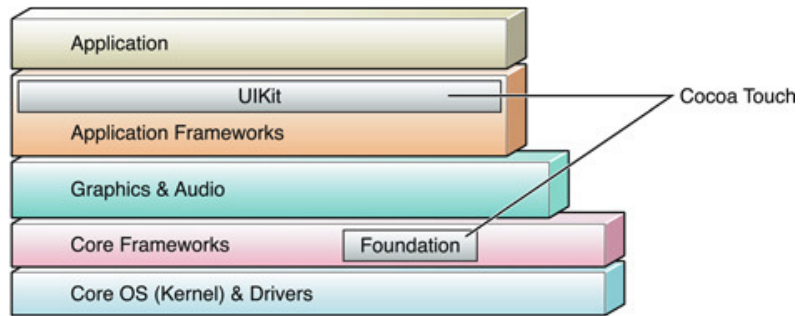


Figura 6 – Cocoa en la arquitectura iPhone OS

### 3.2. Cocoa y Cocoa Touch

Cocoa es un conjunto de tecnologías que provienen del desarrollo del sistema operativo NeXTSTEP. En 1987 la compañía NeXT presentó su sistema operativo basado en un BSD Unix de Berkley, y desde entonces el sistema operativo incluía una serie de frameworks para que los programadores desarrollasen sobre el lenguaje Objective C. Tras el abandono del negocio del hardware por parte de NeXT en 1993, NeXTSTEP se migró a diferentes plataformas (Intel x86, SPARC, Alpha y PA-RISC) y el proyecto pasó a denominarse OpenStep. En 1996, tras los infructuosos intentos por parte de Apple de desarrollar un digno sucesor de su sistema Mac OS 7, se encontró en OpenStep una opción viable. Apple adquirió NeXT y durante los siguientes 5 años se desarrolló el siguiente sistema operativo para Macintosh, denominado inicialmente Rhapsody. Rhapsody pretendía ser una fusión del antiguo Mac OS 7 (y siguientes) con las tecnologías de OpenStep y se convirtió en lo que hoy conocemos como Mac OS X, y por su parte OpenStep se convirtió en Cocoa, y que sirve de base para Cocoa Touch.

La diferencia básica entre Coca y Cocoa Touch es la forma en la que el usuario invoca comandos. En Cocoa se tiene un teclado físico y un ratón o trackpad, en Cocoa Touch se usa un teclado virtual y los dedos directamente sobre la pantalla táctil. Más adelante veremos las diferencias en los frameworks dadas sus diferencias en la interacción.

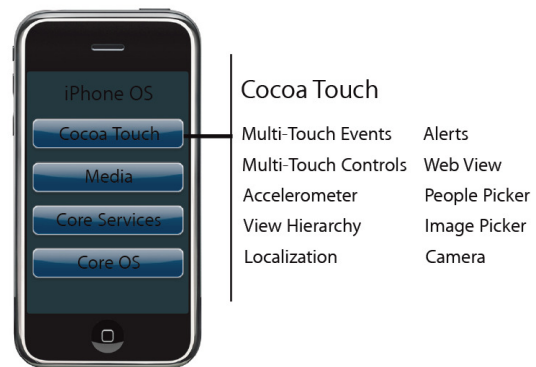


Figura 7 – Cocoa Touch

Cocoa está dividido en 4 niveles de profundidad en función del nivel de sistema al que afecte. Seguidamente describimos brevemente los dos niveles superiores, ya que es donde se encuentran empaquetadas las clases más importantes y, por tanto, sobre los que se han tenido que interactuar.

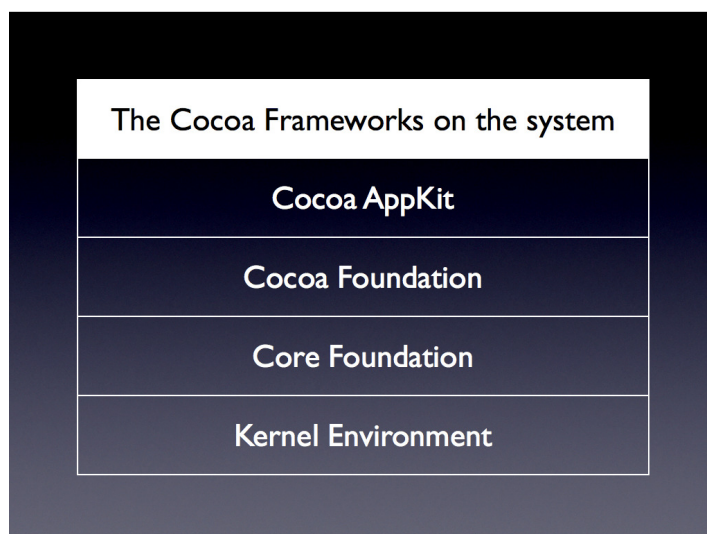


Figura 8 – Estructura de los frameworks de Cocoa

El AppKit Framework contiene más de 120 clases y funciones destinadas a mostrar en pantalla los elementos de la interfaz de usuario, gestionar la comunicación entre ellos y los dispositivos de video o manejar los eventos generados por el usuario.

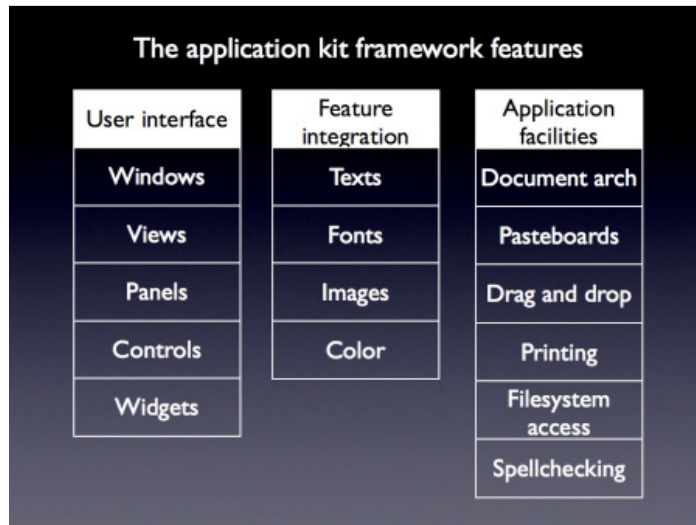


Figura 10 – Cocoa AppKit FrameWork

Sin embargo, las aplicaciones nativas para el iPhone hacen uso de las librerías contenidas en el framework UIKit, que se trataría del equivalente del AppKit (observar figura X) y utiliza el tiempo de ejecución del dispositivo para construir y gestionar la interfaz de usuario. Provee la infraestructura para la implementación de los gráficos, la estructura de la aplicación, el control de los eventos, el manejo de la interfaz, la representación de las vistas y controles así como el soporte de texto y contenido web. Soporta a su vez código para el manejo del acelerómetro, la cámara, la librería de fotos o toda la información que posee el dispositivo.

Aunque el conjunto de clases es mucho más extenso, las más significativas se pueden observar en la siguiente figura:



Figura 11 – Resumen de las clases más importantes contenidas en el framework UIKit

Por otro lado el Foundation Framework proporciona más de 80 clases y funciones que establece una serie de paradigmas a través de los cuales se definen convenciones de uso para ciertos mecanismo y objetos.

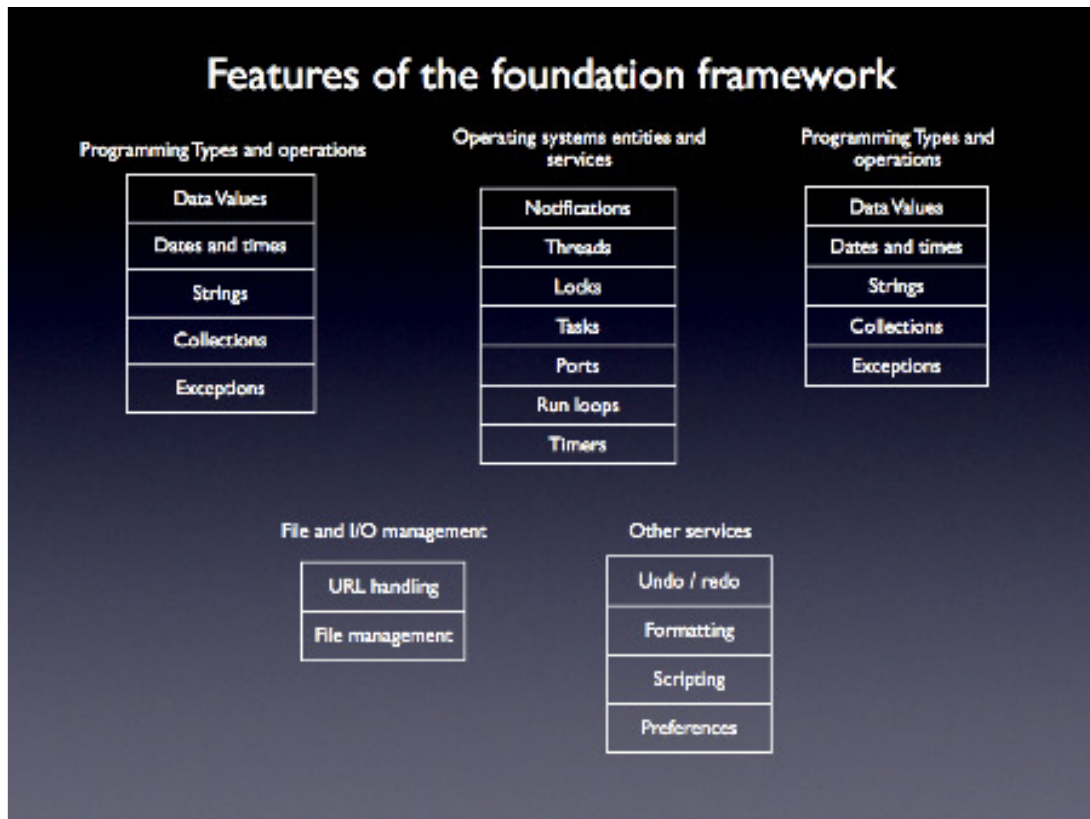


Figura 12 – Cocoa Foundation Framework

Así pues, las diferencias entre Cocoa y Cocoa Touch se podrían resumir en el siguiente esquema:

Cocoa		Cocoa Touch	
AppKit	NSView NSControl	UIKit	UIView UIControl
Foundation	NSString NSArray NSObject	Foundation	NSString NSArray NSObject

Figura 13– Tabla resumen de las diferencias entre Cocoa y Cocoa Touch

### 3.3. MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

El modelo MVC divide todas las funcionalidades en tres categorías distintas:

- **Modelo:** Las clases que contienen datos de la aplicación
- **Vista:** Constituido por las ventanas, controles, y otros elementos que el usuario puede ver e interactuar.
- **Controlador:** Enlaza el modelo y vista juntos y es la lógica de aplicación que decide cómo manejar las entradas del usuario.

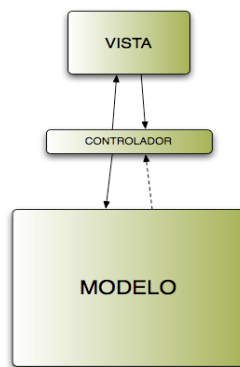


Figura 14 – Esquema MVC

Una aplicación bien diseñada MVC tiene todos sus datos importantes encapsulados en objetos. Cualquier dato que forma parte del estado persistente de la solicitud (ya sea que el estado persistente se almacene en archivos o bases de datos) debe residir en los objetos una vez que el modelo de datos se carga en la aplicación. Debido a que representan los conocimientos y experiencia relacionada con el ámbito del problema específico, que tienden a ser reutilizables.

El objetivo en MVC es hacer que los objetos que implementan estos tres tipos de código, se diferencien el uno del otro como sea posible. Cualquier objeto que se escriba debe ser fácilmente identificable y pertenecer a una de las tres categorías. Un objeto que implementa un botón, por ejemplo, no deben contener el código para procesar los datos cuando se toca el botón, y el código que implementa una cuenta bancaria no debería contener el código para dibujar una tabla para mostrar sus transacciones.

### 3.4. Amazon s3

En este capítulo abordaremos el funcionamiento del hosting contratado por la empresa REDAMM, Amazon, para el almacenamiento de datos. Para la organización de los archivos, Amazon ha creado tres conceptos:

- **Buckets:** son algo parecido a un directorio o carpeta de un sistema operativo, donde se colocan los archivos. Los nombres de los buckets están compartidos entre toda la red de Amazon S3, por lo que si se crea un bucket, nadie más podrá usar ese nombre para un nuevo bucket.
- **Objects:** son las entidades de datos en sí, es decir, los archivos. Un object almacena tanto los datos como los metadatos necesarios para S3, y pueden ocupar entre 1 byte y 5 Gigabytes.
- **Keys:** son una clave única dentro de un bucket que identifica a los objects de cada bucket. Un object se identifica de manera unívoca dentro de todo S3 mediante su bucket+key.

En cuanto a seguridad, Amazon ha implementado un sistema de permisos de acceso a archivos por usuario a los que se le pueden dar capacidad de "Lectura", "Escritura" o "Control Total".

Se dispone de una API para la permitir la comunicación de las aplicaciones con S3, que acepta mediante HTTPRequest peticiones cifradas con Crypt/HMAC. Cada acceso que se haga a esta API deberá estar validado mediante dos claves que se nos proporciona y que, junto a un hash basado en una semilla temporal, información de acceso, y la key de destino, generan una firma que el sistema validará. Estas peticiones HTTP permiten la subida de archivos, modificación de permisos, creación de buckets, eliminación de objects... En definitiva, todas las acciones necesarias para administrar los S3.

## 4. iStreamrTV

iStreamrTV es el nombre que ha recibido la aplicación para dispositivos iPhone y iPod Touch desarrollada para la empresa REDAMM. Su nombre proviene del resultado de la combinación del nombre de la plataforma web (streamrTV) para el alquiler de vídeos adaptado al popular prefijo que Apple proporciona a sus productos "i" (el cual su significado es el de "Internet"). El logo utilizado es el que representa a la página y donde la S de streamerTV sobre el uso de una textura metálica que representa avance tecnológico.

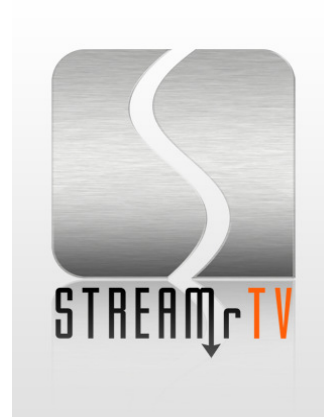


Figura 15 – Logo streamrTV

### 4.1. El proyecto

Partiendo de una arquitectura y unas bases predefinidas, y poseyendo un modelo referenciado (UML), es posible crear un programa capaz de cumplir los objetivos propuestos. La falta de software en la comercialización de películas mediante el uso de dispositivos portables es todavía bastante alta. Este vacío pretende ser llenado poco a poco con iStreamrTV, una potente y simple herramienta que da una nueva alternativa cuidando la usabilidad. Dado que su uso requiere el conocimiento mínimo necesario para trabajar con las aplicaciones típicas encontradas en la App Store (gracias a la política de estandarizar los objetos), cualquier usuario que sepa manejar el iPhone o iPod Touch tendrá facilidad para manejar iStreamrTV.

La siguiente figura resume esquemáticamente las dos funcionalidades básicas de la aplicación:

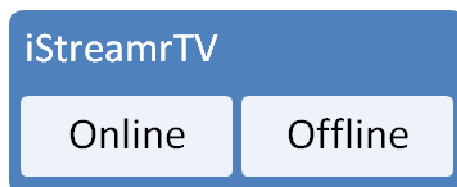


Figura 16 – Esquema funcionalidad iStreamrTV

La parte online servirá para navegar por los contenidos ofrecidos (tráilers y películas) para una posterior descarga de éstos. Por lo tanto toda la información mostrada y por la que se navega está en continua actualización. Por otro lado, la configuración básica y el visionado de las películas corresponden a la parte offline utilizando el propio reproductor incorporado sobre los archivos ya localizados en el aparato.



## 4.2. Usabilidad

Dado que se trata de software para comercialización de productos, iStreamrTV está pensado para abarcar la mayor cantidad de usuarios posibles. Por lo tanto se encontraría en el campo del ocio y abarcando a todos los usuarios sin importar la edad o el sexo. Como requisito previo sería necesaria la adquisición de uno de los productos sobre los que trabaja la aplicación ya citados anteriormente donde, conociendo el funcionamiento de estos, no requeriría ningún tipo de aprendizaje especial dada la homogeneización en el uso.

Un ejemplo de usabilidad y homogeneización se verá más adelante, en la comparación de aplicaciones similares en el capítulo 9.



Figura 17 – iPhone

## 5. Requerimientos

Basado en los objetivos de iStreamrTV (1.3 "Objetivos"), el siguiente capítulo describe los requerimientos desde el punto de vista del usuario final.

### 5.1. Casos de uso

Los casos de uso servirán para conocer posibles las acciones que pueden efectuar los actores y capturar los requisitos del sistema.

#### 5.1.1. Actores

Antes de empezar veremos una visión global de la estructura de los actores con el sistema.

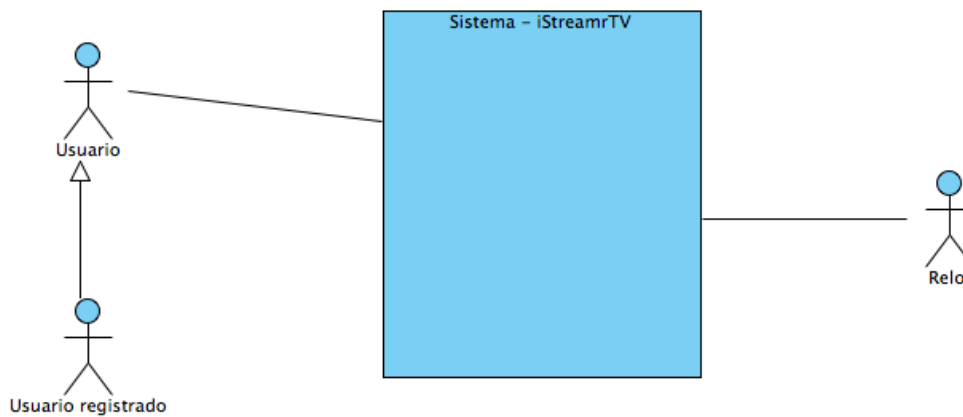


Figura 18 – Diagrama casos de uso. Actores

Podemos observar que se distingue entre un usuario (anónimo) siendo una generalización de un usuario registrado. También existe un reloj, un actor ficticio que dispara un caso de uso de forma periódica y automática. A continuación se dará una visión detallada de los casos de uso, con un análisis del flujo de eventos y una enumeración de las extensiones síncronas asociadas.

### 5.1.2. Acceso

El apartado llamado "acceso" contiene distintos métodos para mantener y gestionar las acciones referentes a la autenticación y autorización de la sesión. Existen tres casos de usos distintos que puede acceder cualquier usuario dadas unas pre-condiciones.

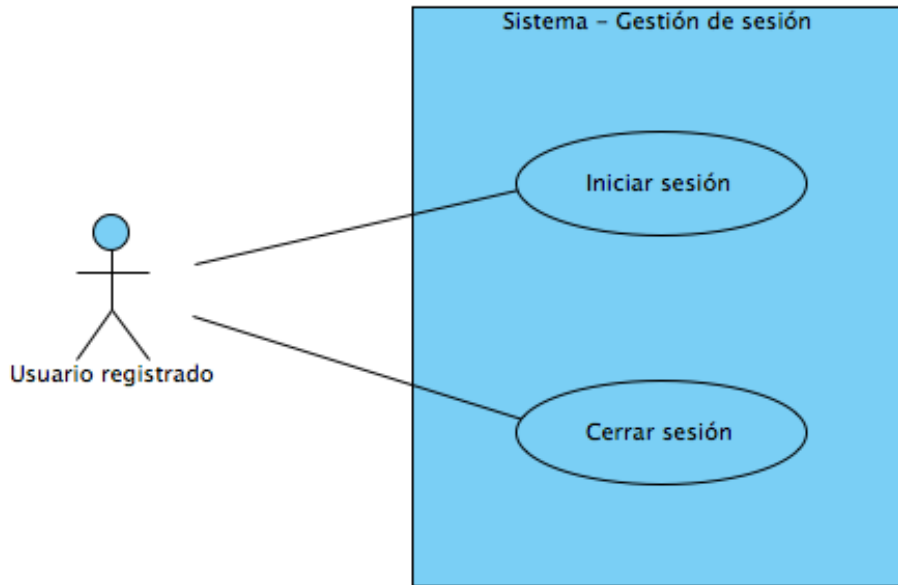


Figura 19 – Caso de uso. Acceso (gestión de sesión)

Como podemos observar, el inicio y cierre de sesión quedan restringidos en su uso solo para usuario registrados.

### 5.1.2.1. Iniciar sesión

Un usuario puede autenticarse con el sistema e iniciar una sesión al comienzo de la aplicación introduciendo correctamente el nombre de usuario y su contraseña.

#### Detalles del caso de uso

**Nombre:** Iniciar sesión.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario no puede tener ninguna sesión activa.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario introduce el usuario y la contraseña.	
2.	El usuario pulsa el botón de inicio de sesión.	
3.		El sistema valida los datos recibidos.
4.		El sistema hace una petición al servidor para conocer si existe alguna combinación con el usuario y la contraseña recibida.
5.		El sistema indica que se ha iniciado la sesión redireccionando al menú principal.

#### Extensiones síncronas

#1 En 4, si no se encuentra el usuario y la contraseña. -> Error, se muestra mensaje informando de ello.

### 5.1.2.2. Cerrar sesión

Un usuario siempre tendrá la opción de cerrar su sesión con el sistema tan pronto como haya iniciado una sesión.

#### Detalles del caso de uso

**Nombre:** Cerrar sesión.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario debe haber iniciado una sesión anteriormente.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario pulsa el boto de cierre de sesión.	
2.		El sistema cierra la sesión del usuario.
3.		Se actualiza el menú principal de pantalla para permitir nuevos inicios de sesión.

#### Extensiones síncronas

No existen.

### 5.1.3. Listado y gestión online

Aquí abarcaremos las posibles interacciones que tiene un usuario registrado respecto a la visualización y gestión de los contenidos online ofrecidos. Esta parte del sistema contiene un peso importante dentro de iStreamrTV. Se pueden distinguir los siguientes casos de uso:

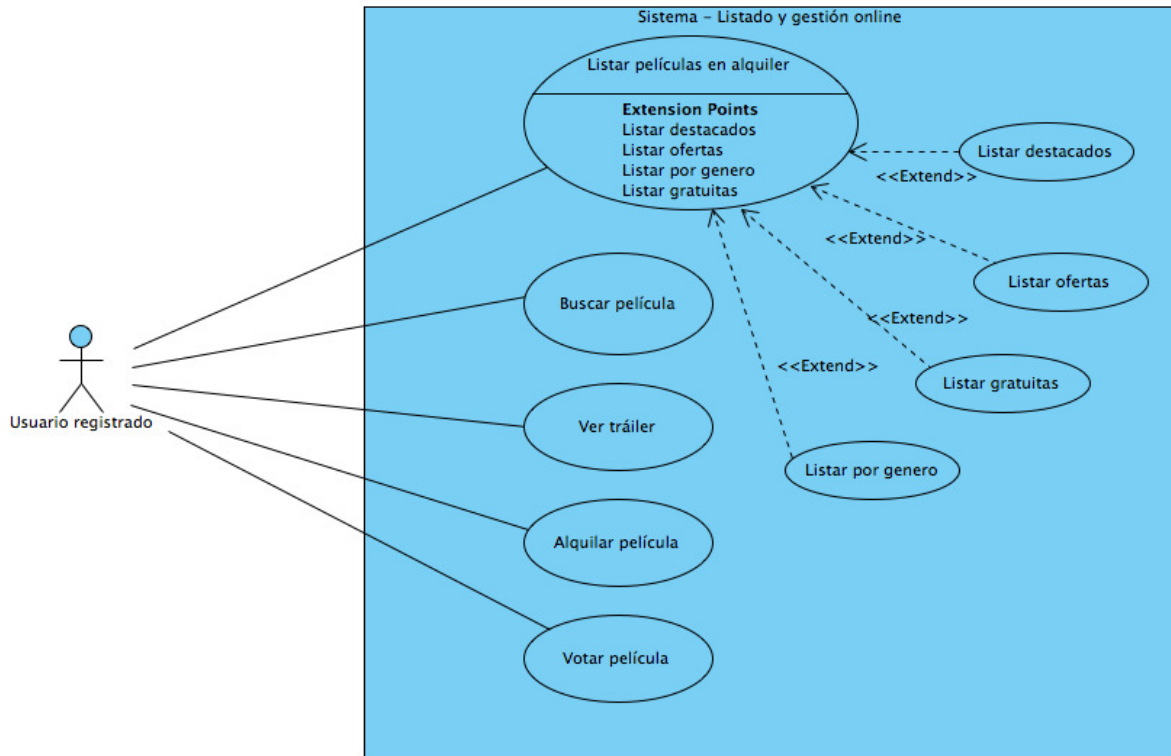


Figura 20 – Casos de uso. Listado y gestión online

Al igual que en la gestión de la sesión, los casos de uso mostrados solo pueden ser accionados por un usuario registrado que haya iniciado la sesión, que quedará reflejado en las condiciones de los detalles en los próximos casos de uso.

### 5.1.3.1. Listar películas en alquiler

El listado de películas permite dar acceso a todo el contenido de películas en el que se incluyen numerosas extensiones dada la cantidad de filtros establecidos. Por lo tanto su funcionamiento es compartido y que detallaremos seguidamente.

#### Detalles del caso de uso

**Nombre:** Listar películas en alquiler.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario debe haber iniciado una sesión anteriormente.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario selecciona pulsa en uno de los distintos botones de listados (filtro).	
2.		El sistema comprueba el filtro seleccionado.
3.		El sistema hace una petición al servidor para obtener el listado de películas
4.		El servidor devuelve el listado y se muestran en pantalla.
5.	El usuario extiende el listado	
6.		El sistema repite el proceso de listado para las siguientes películas.

#### Extensiones síncronas

**#1** En 3, si se cumple el tiempo de espera (timeout). -> Error, el tiempo de llamada ha expirado y se muestra la tabla vacía advirtiendo al usuario.

**#2** En 6, al igual que en 3, si se cumple el tiempo de espera (timeout). -> Error, el tiempo de llamada ha expirado y se muestra la tabla vacía advirtiendo al usuario.

### 5.1.3.2. Buscar película

La búsqueda de películas guarda bastantes similitudes con el listado, pero dadas sus notables diferencias hacen que se muestre como un caso de uso independiente evitando por tanto que se trate de una extensión de éste.

#### Detalles del caso de uso

**Nombre:** Buscar película.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario debe haber iniciado una sesión anteriormente.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario selecciona el menú de listado de películas mediante búsqueda.	
2.	El usuario introduce el texto y pulsa el botón de buscar.	
3.		El sistema comprueba los datos introducidos.
4.		El sistema hace una petición al servidor para obtener el listado de películas según el texto introducido.
5.		El servidor devuelve el listado y se muestran en pantalla.
6.	El usuario extiende el listado	
7.		El sistema repite el proceso de listado para las siguientes películas.



## Extensiones síncronas

#1 En 4, si se cumple el tiempo de espera (timeout). -> Error, el tiempo de llamada ha expirado y se muestra la tabla vacía advirtiendo al usuario.

#2 En 7, al igual que en 4, si se cumple el tiempo de espera (timeout). -> Error, el tiempo de llamada ha expirado y se muestra la tabla vacía advirtiendo al usuario.

### 5.1.3.3. Ver tráiler

Un usuario puede ver un tráiler si la película lo contiene en cualquier momento.

#### Detalles del caso de uso

**Nombre:** Ver tráiler.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario debe haber iniciado una sesión anteriormente. La película debe poseer tráiler, de lo contrario el botón estaría desactivado.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario, desde el detalle de película, pulsa el botón de ver tráiler.	
2.		El sistema lanza un mensaje de confirmación.
3.	El usuario confirma su elección.	
4.		El sistema hace una petición al servidor para recibir mediante streaming el contenido.
5.		El tráiler se visualiza mediante el reproductor de vídeo.

#### Extensiones síncronas

- #1 En 3, el usuario cancela su petición. -> Se muestra la vista detalle de la película.
- #2 En 4, la película está corrupta. -> Error, se muestra un mensaje de advertencia.
- #3 En 5, el usuario detiene el visionado del tráiler. -> Se cierra del reproductor de vídeo y se muestra la vista detalle de la película.

#### 5.1.3.4. Alquilar película

Del mismo modo que la visualización de tráilers, los usuarios pueden alquilar películas siempre y cuando tengan créditos suficientes disponibles.

##### Detalles del caso de uso

**Nombre:** Alquilar película.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario debe haber iniciado una sesión anteriormente.

**Postcondiciones:** Los créditos del usuario deben haberse reducido correctamente una vez se ha alquilado la película.

##### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario, desde el detalle de película, pulsa el botón de alquilar película.	
2.		El sistema lanza un mensaje de confirmación.
3.	El usuario confirma su elección.	
4.		El sistema obtiene los créditos del usuario y comprueba que es mayor al precio.
5.		Se notifica al servidor el movimiento para que quede registrado.
6.		Se descuentan los créditos del usuario.
7.		Se añade un registro en la aplicación para incluir la película en el menú "mis películas".

##### Extensiones síncronas

#1 En 3, el usuario cancela su petición. -> Se muestra la vista detalle de la película.

#2 En 4, el usuario no posee suficientes créditos. -> Error, se muestra un mensaje de advertencia y se muestra la vista detalle de la película.

### 5.1.3.5. Votar película

En este último caso de uso del listado y gestión online, los usuarios pueden valorar las películas libremente del 1 al 10. Estos datos quedarán registrados en la base de datos evitando que pueda votar otra vez la misma película.

#### Detalles del caso de uso

**Nombre:** Votar película.

**Actores:** Usuario registrado.

**Precondiciones:** El usuario debe haber iniciado una sesión anteriormente.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario, desde el detalle de película, pulsa el botón de votar película.	
2.		El sistema muestra una ventana para introducir la votación.
3.	El usuario introduce un valor y pulsa en el botón de votar.	
4.		El sistema recibe la valoración y se muestra un mensaje confirmándolo.
5.		Se notifica al servidor de la votación y se añade un registro.

#### Extensiones síncronas

#1 En 1, el usuario ya había votado con anterioridad la película. -> Se muestra un mensaje de advertencia.

#2 En 3, el usuario cancela la votación. -> Se muestra la vista detalle de la película.

#### 5.1.4. Configuración y gestión offline

En este subcapítulo detallaremos los casos de usos cuyo actor será un usuario no identificado o no registrado. Aquí encontramos distintas acciones de configuración, así como de descarga y visionado de la película. Aunque evidentemente la descarga requiere de conexión a Internet para funcionar, no requiere haber iniciado una sesión, es por ello por lo que se agrupa en este conjunto.

La configuración y gestión offline se trata del otro peso importante en el sistema, tal y como se ha visto en el esquema de la figura X.

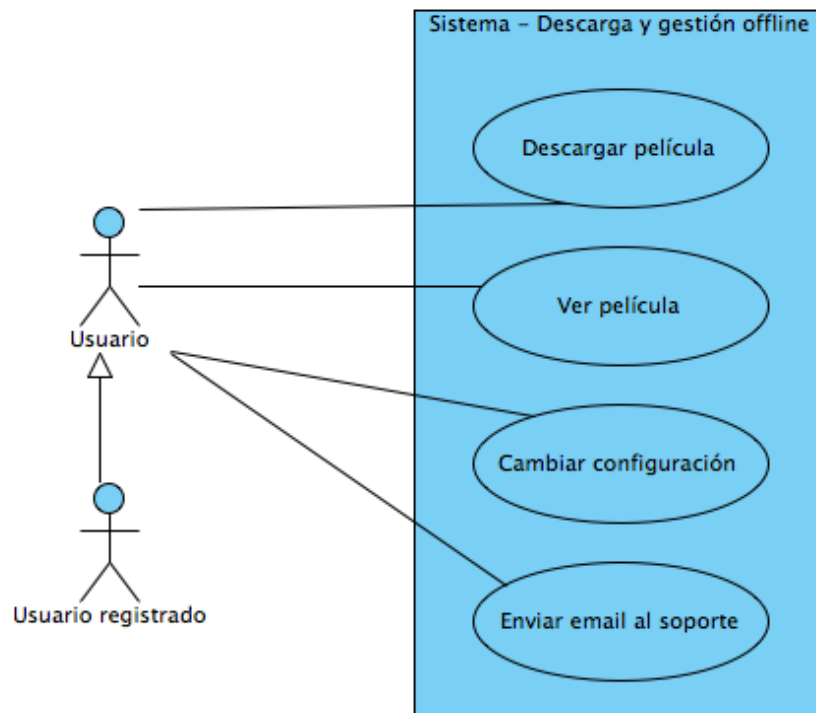


Figura 21 – Casos de uso. Configuración y gestión offline

#### 5.1.4.1. Descargar película

Los usuarios finales podrán descargar las películas que hayan alquilado desde el propio aparato, por ello no requiere una identificación.

#### Detalles del caso de uso

**Nombre:** Descargar película.

**Actores:** Usuario, usuario registrado.

**Precondiciones:** El usuario debe haber alquilado una película previamente.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario pulsa el botón de descargar película.	
2.		El sistema obtiene el estado de la descarga.
3.		El sistema envía al servidor la orden de transferencia de datos de la película seleccionada a partir del trozo correspondiente.
4.		El sistema recibe los trozos y los va juntando en un único archivo.
5.		Se cambia el botón de descarga por el de activar la película.

#### Extensiones síncronas

**#1** En 3 y 4, la aplicación se cierra o pierde la conexión a Internet. -> Se muestra un mensaje de advertencia y se modifica el estado de la descarga a parcial.

**#2** En 4, si el trozo descargado no es el último. -> Se modifica el estado de descarga y se vuelve al punto 3.

#### 5.1.4.2. Ver película

Una vez descargada, se podrán visualizar tantas veces como se desee (dentro del periodo de alquiler) los vídeos.

#### Detalles del caso de uso

**Nombre:** Descargar película.

**Actores:** Usuario, usuario registrado.

**Precondiciones:** El usuario debe haber alquilado una película previamente y haberla activado.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario pulsa el botón de ver película.	
2.		El sistema busca y selecciona el fichero correspondiente.
3.		Se reproduce la película mediante el reproductor de vídeo.

#### Extensiones síncronas

#1 En 2, si el fichero está corrupto. -> Se muestra un mensaje de advertencia y se borran los datos asociados para una nueva descarga.

#3 En 3, el usuario detiene el visionado de la película. -> Se cierra del reproductor de vídeo y se muestra la vista detalle de la película alquilada.

### 5.1.4.3. Cambiar configuración

Se pueden establecer y modificar ciertos parámetros que cambiarán la funcionalidad de algunas opciones de la aplicación.

#### Detalles del caso de uso

**Nombre:** Cambiar configuración.

**Actores:** Usuario, usuario registrado.

**Precondiciones:** Ninguna.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario pulsa el botón de configuración.	
2.		El sistema muestra en pantalla las distintas opciones modificables
3.	El usuario modifica la configuración y guarda el estado.	
4.		El sistema aplica los cambios asignados.

#### Extensiones síncronas

No existen.



#### 5.1.4.4. Enviar email a soporte

También se tiene la posibilidad de enviar un email al soporte técnico de streamTV en caso de consulta o problema.

#### Detalles del caso de uso

**Nombre:** Enviar email a soporte.

**Actores:** Usuario, usuario registrado.

**Precondiciones:** Tener configurada una cuenta de correo por defecto en el dispositivo.

**Postcondiciones:** Ninguna.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario pulsa el botón de envío de email al soporte.	
2.		El sistema muestra en pantalla el formulario para el envío.
3.	El usuario rellena la información en los campos.	
4.	Se pulsa el botón de enviar.	
5.		El sistema envía un email al destinatario solicitado con la información adjuntada.

#### Extensiones síncronas

#1 En cualquier momento el usuario puede cancelar el envío y guardar el mensaje para un futuro envío. -> Se cierra el formulario y se redirecciona al menú principal.

### 5.1.5. Periódicos

Por último se recoge el caso de uso accionado por el reloj, actor que ejecuta de forma periódica el único caso de uso de este subcapítulo, el borrado de las películas marcando la finalización del alquiler.

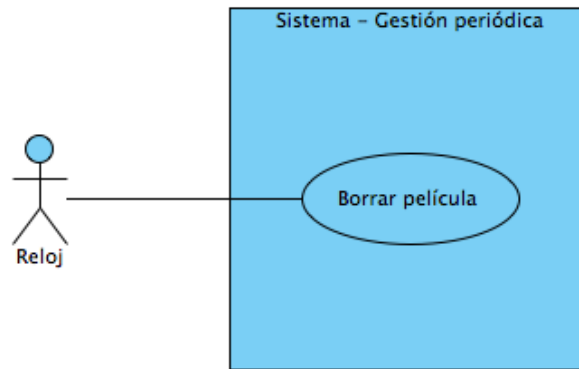


Figura 22 – Casos de uso. Gestión periódica

### 5.1.5.1. Borrar película

En el momento de iniciación de la aplicación, se ejecutará el caso de uso que borrará todas las películas cuyo tiempo de alquiler haya expirado.

#### Detalles del caso de uso

**Nombre:** Borrar película.

**Actores:** Reloj.

**Precondiciones:** Ninguna.

**Postcondiciones:** Se ha borrado toda la información relativa a la película en el dispositivo.

#### Flujo de eventos

	Intención de Usuario	Obligaciones del Sistema
1.	El usuario inicia la sesión.	
2.		El sistema comprueba y actualiza los estados de las películas descargadas y alquiladas.
3.		Se borran todos los archivos y referencias a las películas activadas cuya fecha de alquiler haya expirado.

#### Extensiones síncronas

#1 Se trata de un caso de uso periódico lanzado por el reloj. En este caso se inicia cada vez que se ejecuta el programa.

#2 En 3, Si no existe la película, se borran solamente las referencias a ella.

## 5.2. Glosario

Toda la terminología utilizada en el capítulo de "Casos de Uso" puede ser encontrada aquí.

- **Usuario final:** se trata de un usuario normal antes de identificarse en el sistema.
- **Usuario o usuario registrado:** un usuario es el actor principal en iStreamrTV. Para ello ha sido necesario una identificación donde se le asocian unos datos relacionados a su cuenta.
- **Usuario no registrado:** usuario que utiliza la aplicación sin identificarse en el sistema.

## 6. Tecnologías

Para elegir las herramientas necesarias para la implementación del proyecto, es necesario orientar los requisitos de un usuario según los objetivos formulados. Por otro lado, una funcionalidad eficiente, así como una interfaz sencilla y usable, son una importante base para el desarrollo de un programa para un uso masivo de usuarios.

### 6.1. Lenguajes de programación

Con la incorporación del iPhone en el mercado, Apple ha decidido blindar el desarrollo de aplicaciones para éste haciéndolo exclusivo únicamente para sus herramientas. Con esto se prohíbe el uso de intérprete intermedios y compiladores cruzados, es decir, se excluyen el uso de las API privadas siendo útiles las propias de la compañía. Estas aplicaciones deben ser escritas en Objective-C, C, C++ o JavaScript pero esta última no puede hacer uso de las API documentadas, haciendo por tanto útiles las tres primeras opciones. De esta manera se consigue evitar aplicaciones que no son directamente controladas por el propio iPhone.

Entre las tres opciones, Objective-C es la opción obligada (siendo las otras dos opcionales y complementarias), ya que el framework utilizado (Cocoa Touch framework) hace uso de este lenguaje, por lo que para aprovechar todas las funcionalidades, partiendo desde las más básicas, hacen obligada su elección. La principal característica de Objective-C es su similitud con C++ (extiende el lenguaje estándar ANSI C) aunque la sintaxis básica, así como el diseño, están basados en SmallTalk.

Los diferentes tipos de fichero existentes se pueden resumir en la siguiente tabla:

Extensión	Tipo
.h	de cabecera. Contiene las declaraciones de clases, el tipo, funciones y constantes.
.m	Fichero de recursos. Contiene el código en Objective-C.
.mm	Fichero de recursos. Contiene el código en C++. Sólo se utiliza para referenciar a código anteriormente creado y que se quiere enlazar en la aplicación.

Figura 23 – Tabla tipos de fichero y sus extensiones

Debido a que Objective-C es una extensión de C compatible hacia atrás, muchas de las características de la sintaxis de C han sido heredadas, entre ellas:

- Sentencias de control de flujo.
- Los tipos de datos fundamentales, estructuras y punteros.
- Conversiones implícitas y explícitas entre tipos.
- Los ámbitos de las variables: Globales, estáticas y locales.
- Las funciones y su sintaxis.
- Las directivas del preprocesador (a las que se añaden nueva directivas así como la inclusión de directivas del compilador).

La gran diferencia que distingue este lenguaje de otros es que se trata de un lenguaje muy dinámico ya que muchas decisiones de muchos lenguajes se toman en tiempo de compilación mientras que Objective-C las deja para el tiempo de ejecución. Así mismo, toda la memoria utilizada es dinámica evitando problemas como desbordamientos de pila. Todas estas características hacen este lenguaje idóneo para la programación en nuestro caso.

```

1 #import "arbreDigital.h"
2 #import "tabString.h"
3
4 @interface Lzw:Object
5
6
7
8
9
10
11
12
13 @interface Compresseur:Lzw
14 {
15     NoeudDigital* strTab;
16     int tailleFichierInput;
17     int tailleFichierOutput;
18     unsigned int seuilBits;
19 }
20 - initWithInput:(char*)input Output:(char*)output;
21 - free;
22 - compresser;
23 - (void)ecrire:(unsigned int)code estDernier:(BOOL)estDernier;
24 @end
25
26 @interface Decompresseur:Lzw
27
28
29
30
31
32
33
34
35
36
37 //gcc -o m arbreDigital.o main.o lzw.o inOut.o myString.o -lobjc
38

```

```

1 #import "lzw.h"
2 #import "constant.h"
3
4 static unsigned long outputBuf = 0;
5 static int outputBitCompteur = 0;
6
7 static unsigned long inputBuf = 0;
8 static int inputBitCompteur = 0;
9
10 @implementation Lzw
11
12
13
14
15
16
17
18
19
20
21 @implementation Compresseur
22 - initWithInput:(char*)input Output:(char*)output
23 {
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 24 – Ejemplo de código en Objective-C

Las clases en Objective-C constan de una interfaz y una implementación. La interfaz indica la estructura del objeto y la implementación, como su nombre indica, contiene la implementación de sus métodos.

Normalmente una clase siempre deriva de otra clase a las que hacen referencia ya que de lo contrario no puede utilizar las ventajas del runtime de Objective-C. Aunque es posible que una clase no derive de ninguna clase, esto sólo se hace cuando la clase se va a utilizar como almacén de datos (estilo de estructuras de C), pero no se van a ejecutar métodos sobre ella.

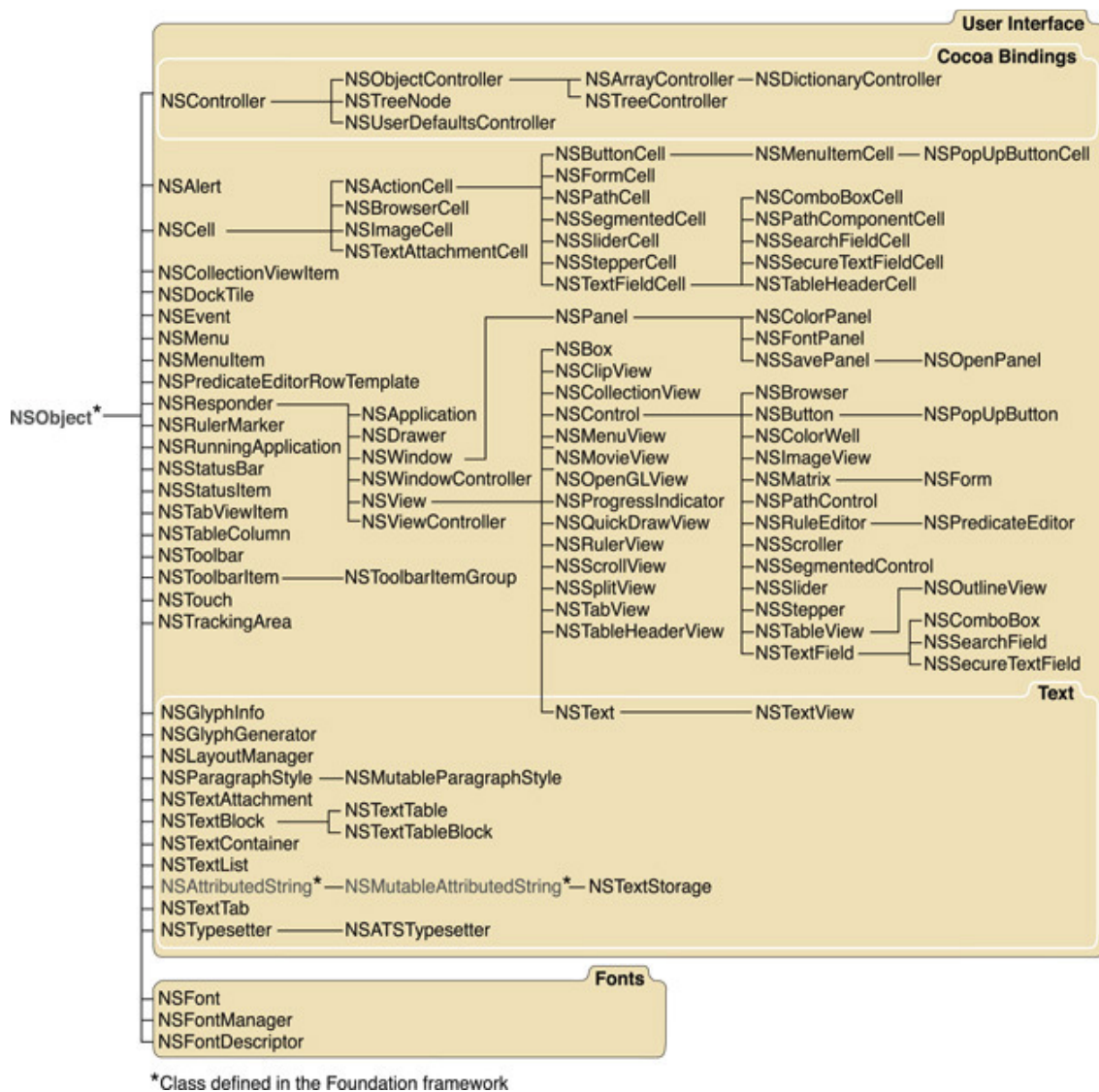


Figura 25 – Estructura de clases de la interfaz

## 6.2. Interfaz de usuario

La interfaz de usuario es el único medio por el que el usuario interactúa con el programa. Cualquier error o inconsistencia puede provocar un rechazo inmediato aunque se haya trabajado una estructura lógica sólida.

### 6.2.1. NextSTEP Interface Builder

Un archivo "NextSTEP Interface Builder" (nib) es una interfaz de usuario generada mediante el uso de "drog and drop" (arrastrar y soltar) donde se establecen los elementos, controladores y conexiones presentes en la aplicación. El código generado en estos ficheros de descripción de la componente gráfica se cargan al arrancar la aplicación y usa Cocoa Touch para pintar la ventana con sus componentes, vistas y controles en la pantalla. Este tipo de técnicas ahorran mucho tiempo y tienen un resultado óptimo.

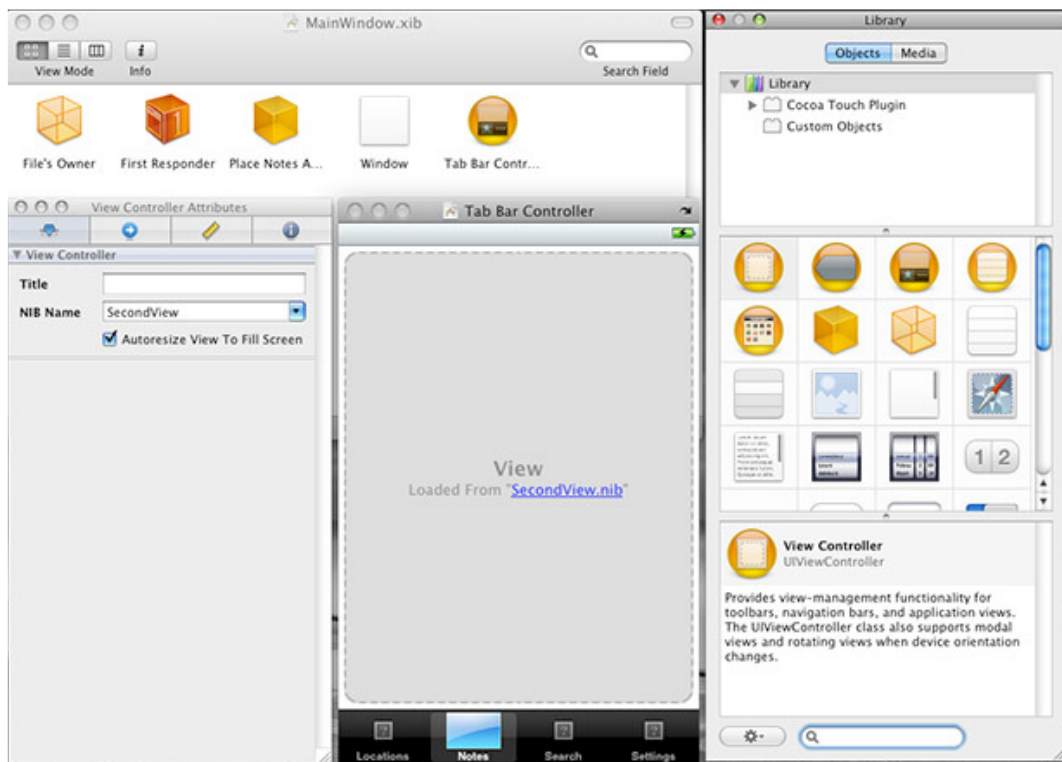


Figura 26 – Interface Builder



### 6.2.2. Interfaz mediante Objective-C

También es posible la creación de interfaces mediante el propio código de programación. Para ellos es necesario Hacer uso de las distintas clases UI (User Interface) teniendo las ventajas e inconvenientes que esto presenta. Esta opción presenta una riqueza mayor aunque un trabajo más costoso, aunque en diseños muy sencillos ahorra más tiempo.

Es posible combinar ambas opciones para aprovechar las ventajas de cada una, aunque esto puede provocar una desorganización en el diseño del software.

### 6.3 Servlets y JSON

Una combinación entre servlets y llamadas JavaScript Object Notation (JSON) es un potente método para realizar llamadas a un servidor con unos parámetros definidos. El cliente envía una petición con unos parámetros llamando a un servlet en concreto. Esta petición es recogida por el servidor que analiza la información y ejecuta funciones, mediante una programación en Java, que devuelve una respuesta con datos estructurados y en un formato establecido.

Aunque inicialmente este sencillo proceso es utilizado generalmente para la creación de páginas web (generalmente en archivos JSP), se ha aprovechado estas funcionalidades para crear un sistema de comunicación eficaz.

## 7. Diseño y realización

Toda la implementación y el proceso de diseño de iStreamrTV están documentados en el capítulo 7. Se describen los módulos del proyecto, la estructura de la base de datos, como las clases trabajan conjuntamente y la interfaz creada para el prototipo.

### 7.1 Arquitectura

En la arquitectura del proyecto podemos distinguir distintas capas o conjuntos. En concreto encontramos tres distintos módulos que están conectados entre ellos: Cliente iStreamrTV, Conversor JSON y Servidor iStreamrTV.

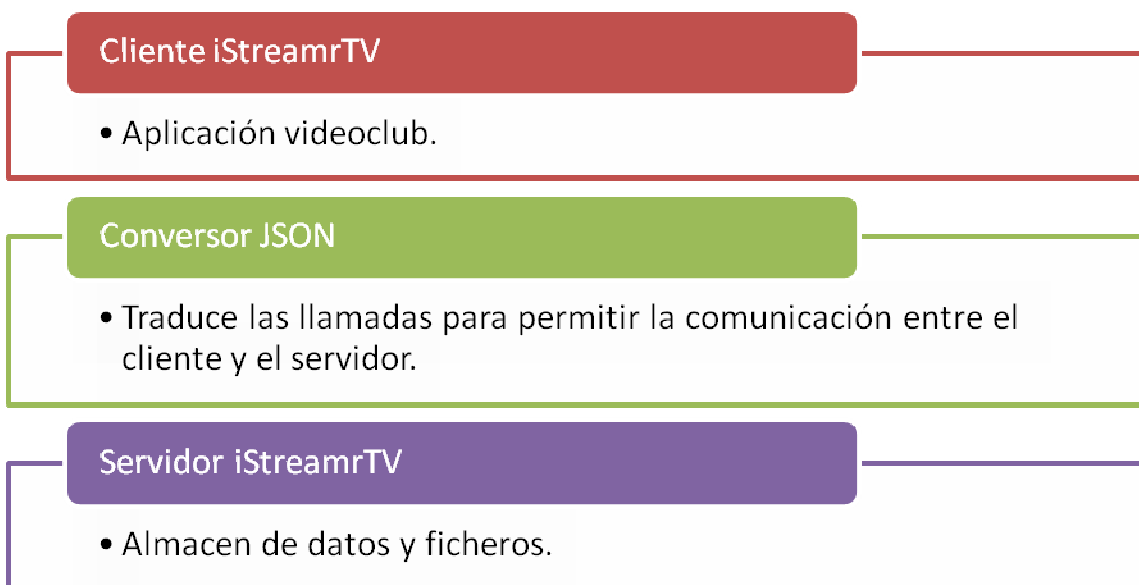


Figura 27 – Esquema de la arquitectura del proyecto

Cuya conexión quedaría establecida de la siguiente forma:

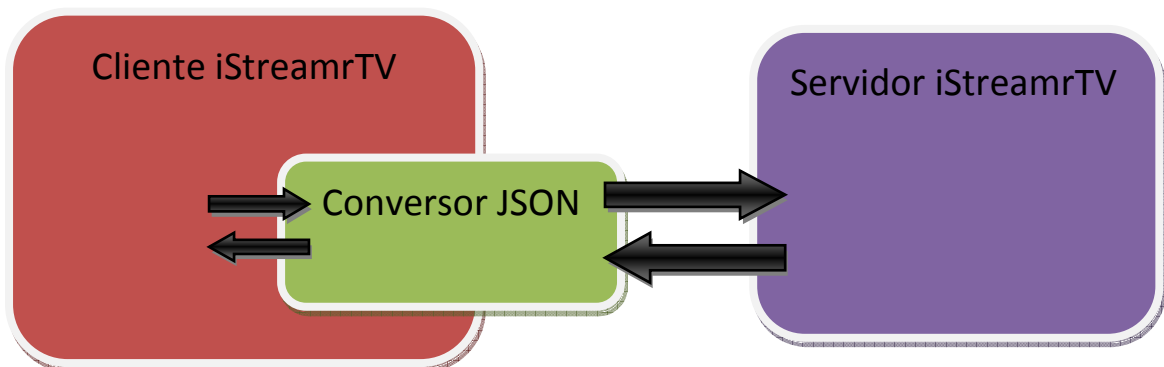


Figura 28 – Conexión de los módulos

### 7.1.1 Cliente iStreamrTV

Basado en el modelo MVC, distinguimos internamente tres capas que trabajan conjuntamente. El funcionamiento de este módulo se podría explicar en forma de ejemplo. El usuario interactúa con la interfaz de usuario pulsando sobre la pantalla táctil. El controlador, que posee y gestiona los eventos gracias a una extensa librería, recibe (por parte de los objetos de la interfaz- vista) la notificación de la acción solicitada por el usuario. Así pues el controlador accede al modelo, actualizándolo y modificándolo de forma adecuada a la acción solicitada por el usuario. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario y la vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. Finalmente la vista espera nuevas interacciones del usuario volviendo a empezar el ciclo.

Este proceso es programable mediante las clases estándar creadas y que se ponen a disposición del usuario. Las capas son:

- **Vista:** a través del template "vista" se han creado archivos nib (con extensión .xib) que contienen las clases necesarias para gestionar los objetos de la interfaz y las interacciones sobre ellos.
- **Controlador:** en el proyecto encontramos distintos templates para gestionar esta capa. Las clases con terminación "\*Controller.h" y "\*Controller.m" poseen gran cantidad de métodos que se encargan de dar funcionamiento a la recepción de los delegados que han sido enlazados previamente con los objetos de la interfaz.
- **Modelo:** las clases que contienen una lista de tareas, el estado de cada una de ellas y que permite que los datos sean persistentes. Corresponderían con el resto de clases, un ejemplo sería la clase "DownloadManager" entre otras.

El cliente iStreamrTV es básicamente la aplicación para el iPhone/iPod Touch que tiene la arquitectura que se acaba de explicar y que enlaza con los otros módulos para un funcionamiento total.

### 7.1.2. Conversor JSON

Alojado internamente en el programa iStreamrTV, este módulo se trata de un conjunto de clases importadas que contienen distintos métodos para tratar las llamadas JSON.

De esta forma las clases traducen las llamadas de forma íntegra tanto para las salientes como para las entrantes. Así pues, a partir de los métodos es posible comunicarse con el servidor a través de una conexión que permite la transferencia de datos y ficheros entre el dispositivo y el servidor donde se aloja la información. Su integración en el programa iStreamrTV permite la llamada entre clases para conseguir un sencillo acceso de funcionalidades desde el Cliente iStreamrTV.

### 7.1.3. Servidor iStreamrTV

El módulo, integrado en servidores de REDAMM, recoge todas las peticiones enviadas por el usuario y las transforma en peticiones para el servidor. Una vez analizada la petición, se prepara una respuesta (o una excepción) que puede ser controlada por la aplicación contactando con la base de datos como proveedor de información. El intercambio de datos, trabajando en conjunto con el conversor JSON y sus métodos, se puede hacer de forma síncrona y asíncrona, según las necesidades de cada momento.

Aquí también dos capas diferenciadas, business y servlet:

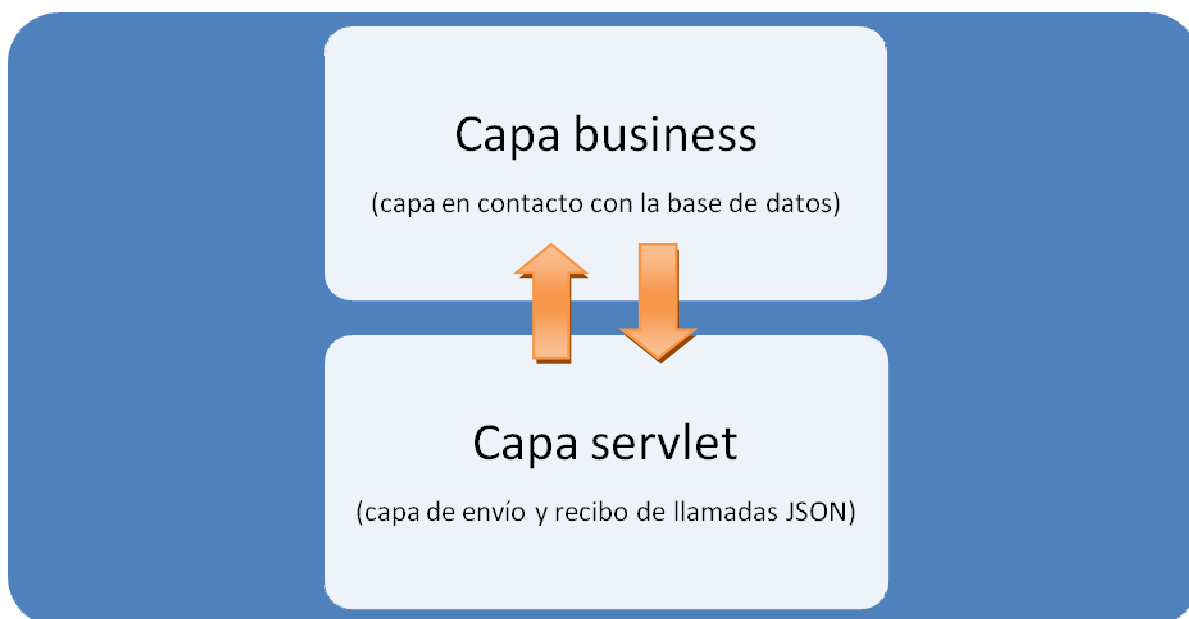


Figura 29 – Capas del servidor iStreamrTV

El esquema de trabajo es similar al de MVC, a diferencia que la capa vista pierde toda la importancia quedándose en solo dos módulos. La capa business se encarga de la gestión y ejecución lógica de la aplicación ya que contiene las clases "manager" y contactando con la base de datos a través de llamadas SQL. Por otro lado, la capa servlet recibe las llamadas (request) y trabaja con ellas según la acción solicitada y política establecida.

## 7.2. Arquitectura de clases

Para utilizar los datos e información en el sistema es necesaria una estructuración de las clases. A continuación mostraremos los modelos asociados a los distintos grupos de clase, que se caracterizan por la herencia para crear subclases adaptadas a los requerimientos.

Dada la gran cantidad de atributos y métodos para las distintas clases, se ha optado por ocultar estos para evitar extensos y confusos diagramas, mostrando solamente éstos en los diagramas más simples.

### 7.2.1. Controladores de vistas

Dada la gran cantidad de vistas presentadas en cualquier aplicación iPhone y iPod Touch, provoca que el modelo quede más compleja que el resto de bloques.

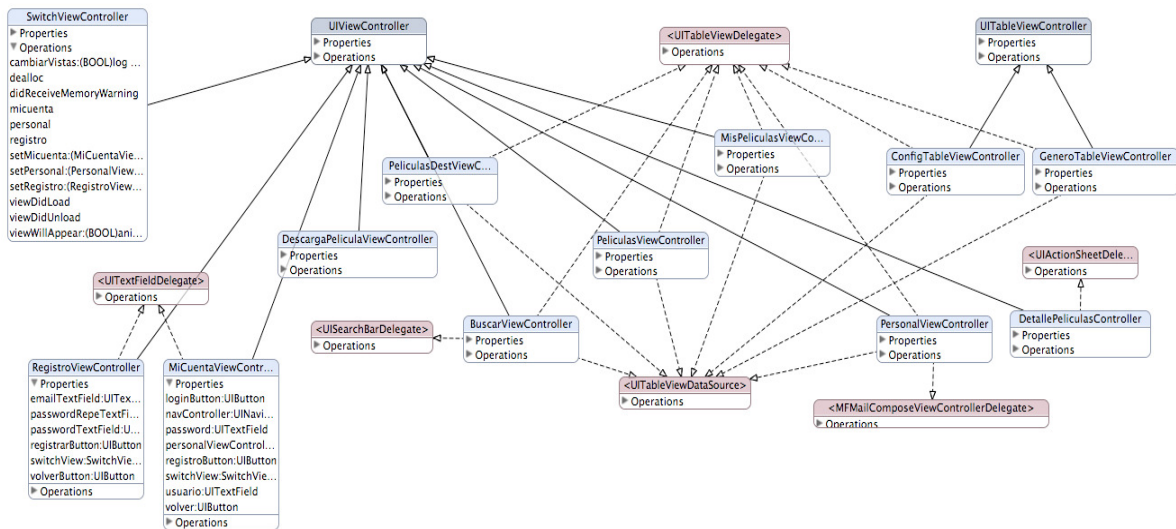


Figura 30 – Modelo de clases controladoras de vistas

Podemos observar, a través de la figura 30, que todos los controladores heredan de dos clases globales llamadas UIViewController y UITableViewController según si lo que contienen es una tabla en su interior o no. En esta arquitectura queda representada la herencia comentada y el funcionamiento de la lógica las vistas, configuradas independientemente para establecer su finalidad.

### 7.2.2. Clases lógicas

A continuación se muestra la estructura de las clases encargadas de llevar la lógica para la gestión y ejecución de la aplicación.

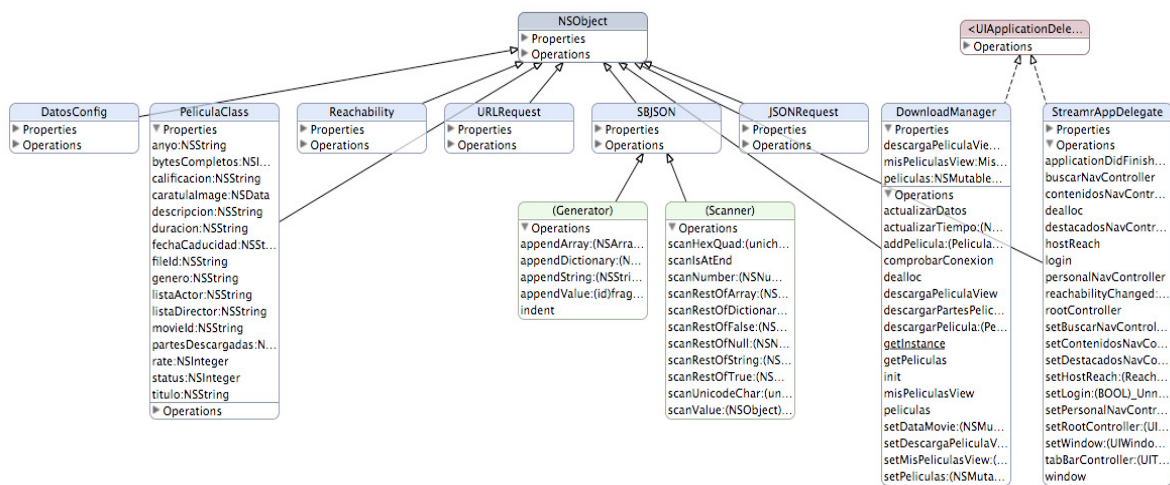


Figura 31 – Modelo de clases lógicas

NSObject es la clase raíz en la jerarquía de la cual el resto de clases han heredado de esta super-clase sus métodos sobrecargándolos.

También encontramos el uso del singleton en el proyecto, esto es instancias únicas de la clase para evitar creaciones de nuevas instancias (regulando el constructor de forma privada) y servir como punto de control y acceso global a ella. Este es el caso, por ejemplo, de la clase "DownloadManager".

### 7.2.3. Otras

Tenemos también modelos de clase para tratar la navegación a través de un controlador y las celdas de las tablas sobre la que se han diseñado distintas subclases para los distintos tipos de celdas existentes en iStreamrTV.

En el caso de la navegación queda la siguiente estructura:

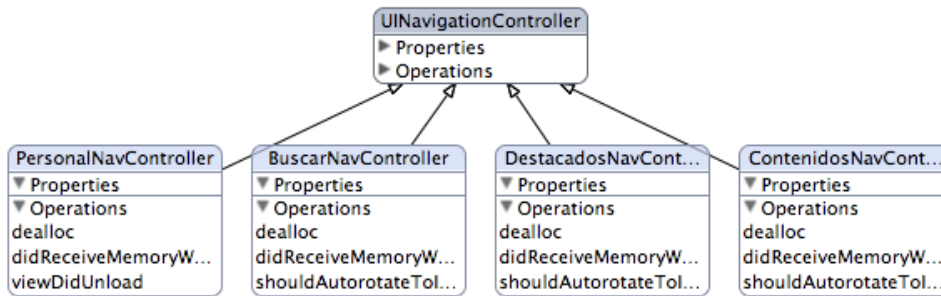


Figura 32 – Modelo de clases para la navegación

Y para las clases referentes a al diseño y funcionamiento de las celdas:

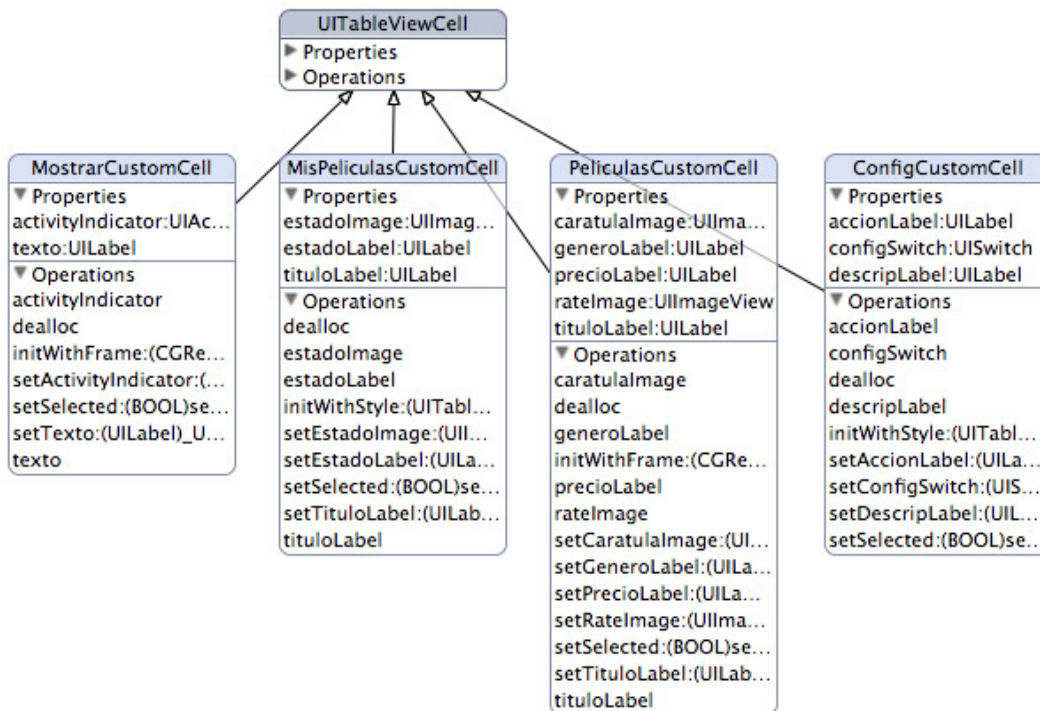


Figura 33 – Modelo de clases para las celdas

Con un esquema más sencillo, el funcionamiento y disposición es similar al de los controladores de vistas.

### 7.3. Diagramas de secuencia

Los diagramas de secuencia nos permiten ver la interacción de un conjunto de objetos en una aplicación a través del tiempo de un método de una clase. Seguidamente se explicarán los métodos más importantes utilizados.

NOTA: se ha utilizado la nomenclatura utilizada en el lenguaje estandarizado UML:

- IU: Interface User
- OC: Object Control

#### 7.3.1. Iniciar sesión

El inicio de sesión identifica al usuario en el sistema. Para autorizar un usuario, el nombre de usuario y la contraseña se envían mediante una llamada de tipo "POST" al servidor. El servidor busca un usuario que coincide con los datos suministrados y después redirecciona al usuario al menú principal haciendo desaparecer la vista de acceso.

Una vez iniciada la sesión se muestran los datos de la cuenta y desaparece la información relacionada con el inicio de sesión.

Si la base de datos no encuentra un usuario con la combinación de nombre y contraseña dada, se devuelve un error que se muestra al usuario mediante una ventana de información.

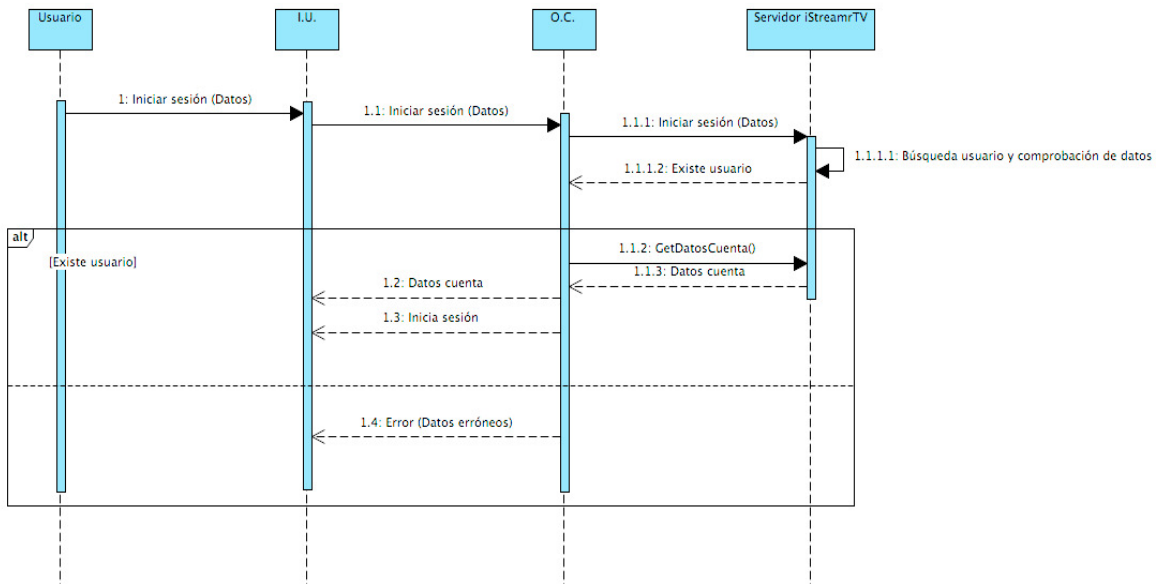


Figura 34 – Diagrama de secuencia. Iniciar sesión



### 7.3.2 Listar películas

Es el único medio por el que un usuario puede navegar por los contenidos ofrecidos. Mediante el listado de películas, el usuario establece un filtro por defecto y realiza una petición al servidor que le devuelve el listado de películas solicitado.

A través de un time-out se controla el tiempo de respuesta del servidor para evitar un bloqueo si éste no responde. En ese caso se muestra la lista vacía.

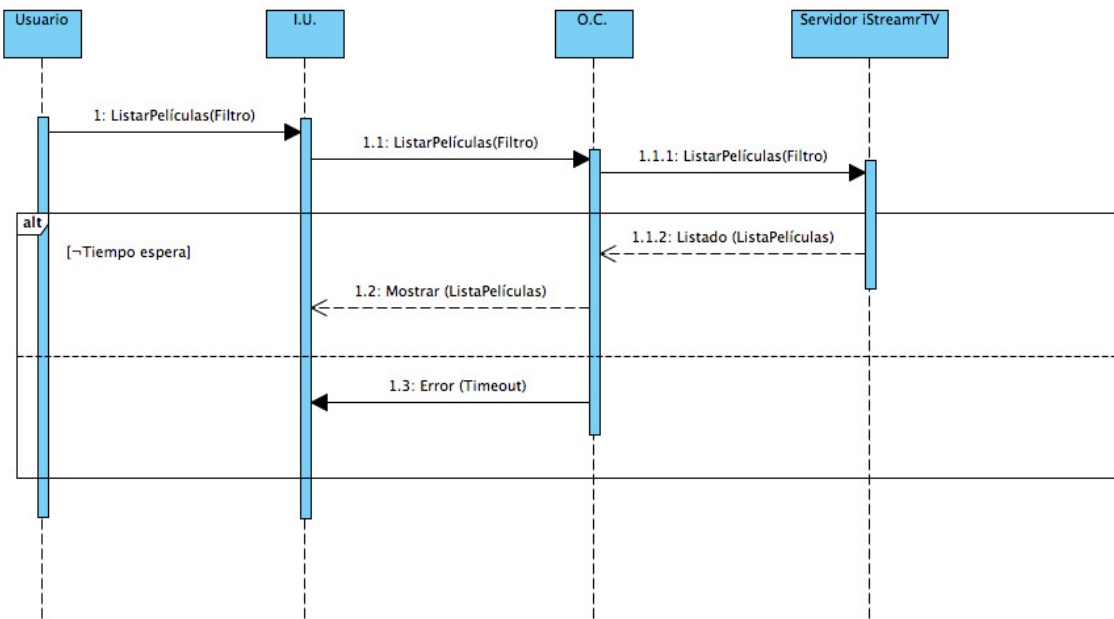


Figura 35 – Diagrama de secuencia. Listar películas

### 7.3.3. Alquilar película

Solamente los usuarios registrados que han iniciado una sesión pueden alquilar películas. Solamente es necesario el envío del número de identificación de la película que, una vez es identificada, se comprueba si el usuario posee suficientes créditos para realizar el alquiler. Si lo tiene, se registra el movimiento y la película se añade a la lista de "mis películas" para una posterior descarga; en caso contrario se alertará al usuario mediante una ventana de que no dispone de créditos suficientes.

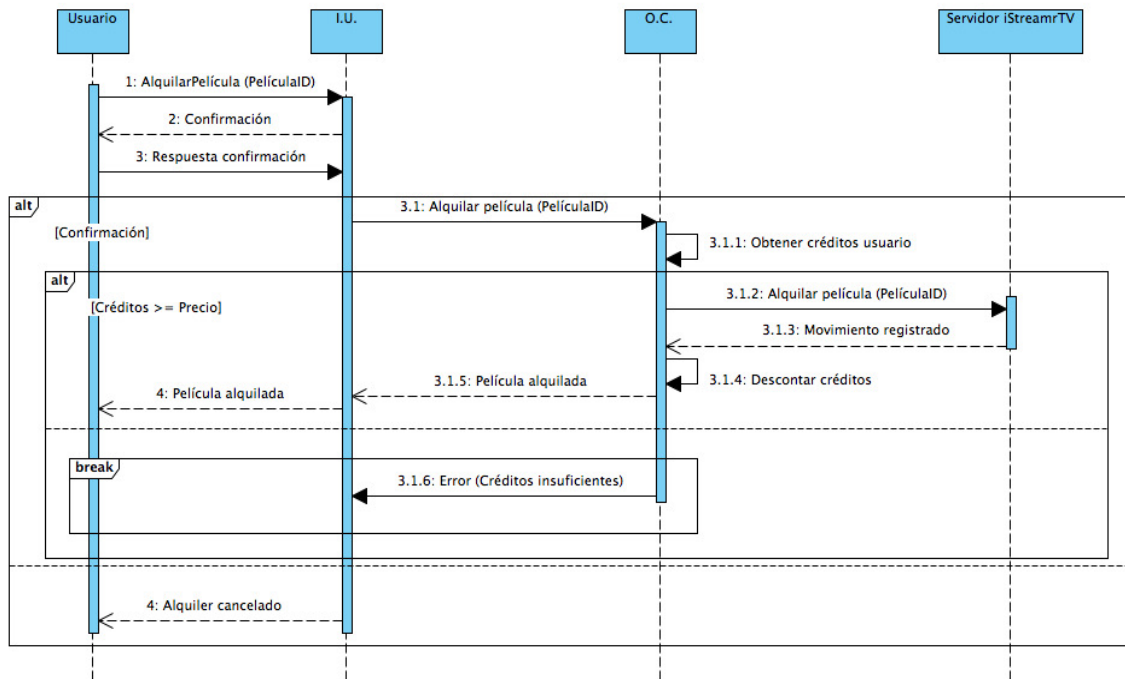


Figura 36 – Diagrama de secuencia. Alquilar película

#### 7.4. Prototipo

A continuación se mostrará la versión final del prototipo explicando la funcionalidad y enlazándola con los casos de uso ya explicados. Las imágenes utilizadas se han generado mediante el uso de Adobe Photoshop para reforzar la interfaz de usuario. Por otro lado, se han empleado los iconos de navegación estándar del propio programa de creación de interfaces.



Figura 37 – Aplicación iStreamrTV en iPhone

##### 7.4.1. Navegación

iStreamrTV se divide en cuatro menús diferenciados, los tres primeros sirven para el listado, filtrado y búsqueda de películas mientras que el cuarto se trata del menú personal del usuario:

- **Destacados:** se recogen las últimas novedades y las películas destacadas del momento.
- **Favoritos:** permite el filtrado de las películas por su género.
- **Buscar:** dada una cadena de entrada, se realiza una búsqueda de las películas que la contengan.
- **Mi cuenta:** menú principal que sirve de configuración y gestión de la cuenta personal. Permite además el acceso a las películas alquiladas.



Figura 38 – Menú de navegación

La navegación se realiza de forma táctil, por lo que todas las acciones y toda la navegación deberán realizarse pulsando con el dedo sobre los elementos de la pantalla.

#### 7.4.2. Mi cuenta

Se trata del menú inicial de la aplicación aunque posee distinta vista dependiendo de si el usuario ha iniciado o no. Desde aquí se inicia y se cierra la sesión, se detalla la información del usuario, da acceso a la gestión, y configuración de iStreamrTV, así como también al listado de películas alquiladas.



Figura 39 – Menú Mi cuenta (principal)



Figura 40 – Menú Mi cuenta (principal) con sesión iniciada

### 7.4.3. Login

Tal y como el caso de uso de inicio de sesión describe, el usuario final necesita rellenar los campos de nombre de usuario y contraseña para identificarse en el sistema. Si se rellena erróneamente alguno de los campos, el sistema advertirá de que se ha introducido alguno de los datos de forma equivocada. Una vez identificado el usuario podrá acceder a los menús de búsqueda y listado de películas, así como a sus datos personales redireccionándolo al menú principal.



Figura 41 – Vista inicio de sesión



Figura 42 – Error en el inicio de sesión

Se ha añadido un control de los datos introducidos avisando de la obligación de rellenar ambos campos. Además, el nombre de los usuarios registrados siempre coincide con el nombre de una cuenta de correo, si se ha introducido mal, también se mostrará un mensaje advirtiendo del error. Esto ayuda a evitar hacer envíos de datos innecesarios que provoquen en el servidor una búsqueda que ya de antemano se sabe que no encontrará la combinación de usuario y contraseña.

#### 7.4.4. Destacados, género y búsqueda

Como ya hemos dicho anteriormente, esta es la forma por la que el usuario navega por los contenidos ofrecidos. Según el listado (filtrado) se mostrarán unas películas u otras. En todas las tablas, la cantidad de películas por defecto es de 20 películas a mostrar, aunque se pueden ir incrementando si el usuario hace uso de la última celda hasta que finalmente quedan todas las películas mostradas (siempre mediante el filtro asociado). Se requiere del inicio de sesión para el acceso, si no se ha producido una identificación, el programa mostrará una alerta recordándonoslo.



Figura 43 – Vista carga de listado



Figura 44 – Intento de listado sin inicio de sesión

En el caso de destacadas, encontramos un submenú de navegación superior que permiten unos filtrados que comúnmente son utilizados por los usuarios. Se trata de películas destacadas por la empresa (como últimas novedades), ofertas y películas gratuitas.

Para los géneros se muestra una tabla previa que permite la selección del género donde seguidamente se mostrarán las películas que lo posean en su información.

Por último el usuario también puede hacer uso de un motor de búsqueda que encontrará aquellas películas que contengan tanto en el título, como en cualquier otro campo de la información, los caracteres introducidos. Mientras se realiza la introducción de datos, se deshabilita la tabla dando un tintado grisáceo y se vuelve a activar si se pulsa en ella.

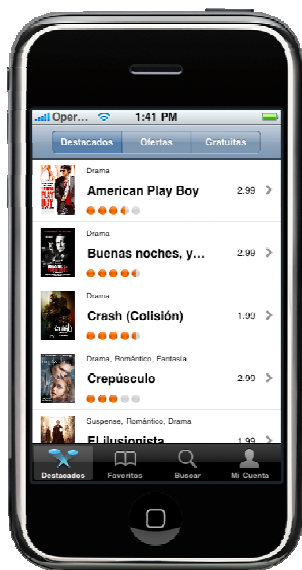


Figura 45 – Listado destacadas



Figura 46 – Listado favoritos (por género)

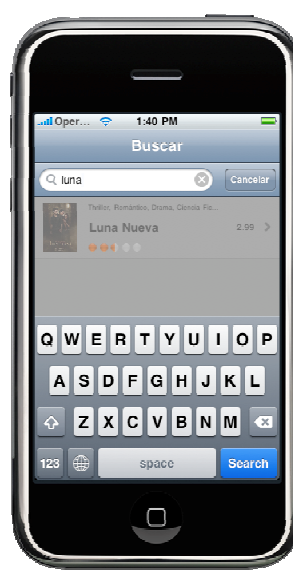


Figura 47 – Listado búsqueda

Para todos los casos, el diagrama de secuencia para el caso de uso "listar películas" descrito con anterioridad recoge y modela los procesos mostrados.

#### 7.4.5. Alquiler películas

Una vez el usuario selecciona una película, la vista cambia al detalle de la película donde se muestran los datos de ésta y la opción de alquilarla. Al pulsar sobre el botón, se mostrará un mensaje de confirmación para evitar el alquiler por error. Si se confirma el alquiler, se realizará una petición y al usuario se le descontarán los créditos de su cuenta y se añadirá la película al menú de "mis películas".



Figura 48 – Vista confirmación de alquiler



Figura 49 – Mensaje película alquilada

En caso de no tener suficientes créditos, la película no se alquilará y se muestra un mensaje de error informando de la situación.



Figura 50 – Mensaje no se dispone de créditos suficientes

#### 7.4.6. Ver tráilers y votación

Desde la vista de detalle de película, el usuario también puede ver el tráiler (si se dispone de él) y dar una valoración numérica del 0 al 10 para votarla. Por otro lado, la votación se restringe a un único voto por usuario y película, por lo que si un usuario lo intenta dos veces, se le informará de que no puede realizar el segundo voto.



Figura 51 – Votación de una película



Figura 52 – Votación efectuada



En el caso de que el usuario seleccione la opción de ver el tráiler, el dispositivo conectará con el servidor iStreamrTV para realizar un visionado mediante streaming a través del reproductor integrado. Si se produce algún error en la conexión, se alerta al usuario y la vista vuelve al detalle de la película.

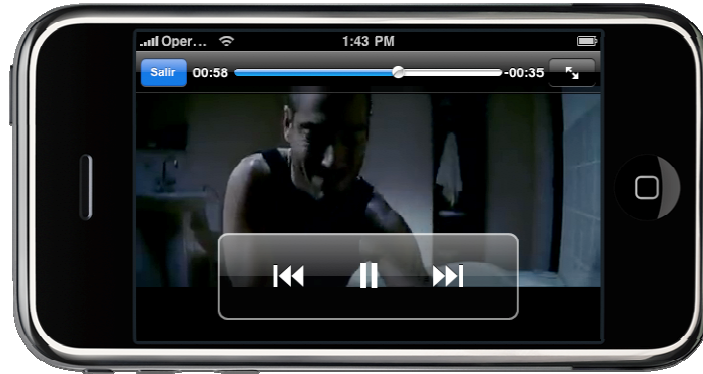


Figura 53 – Tráiler (streaming)

#### 7.4.7. Descarga películas y visionado

Para acceder a la descarga de películas es necesario hacerlo a través del menú principal y entrar en la vista de "mis películas". Ésta ofrece una vista previa de las películas alquiladas y de su estado (no descargada, descargada, activada) Una vez seleccionada la película, la descarga se efectúa de forma manual donde se inicia un hilo secundario para evitar el bloqueo del programa y permitir múltiples descargas al mismo tiempo. Estos hilos se encargan de gestionar todo el proceso descargando la película por trozos y permitir, de esta forma, poder pausarlas y reanudarlas sin pérdidas de datos.

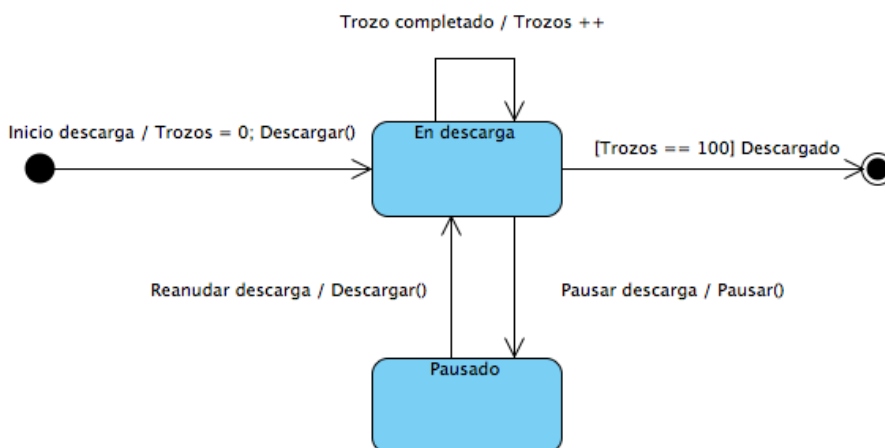


Figura 54 – Diagrama de estados de descarga película

## Videoclub iPhone iStreamrTV



Figura 55 – Vista detalle de descarga



Figura 56 – Vista detalle película en descarga



Figura 57 – Vista detalle película descargada

Existen eventos programados que comprueban la conexión de internet o incluidos en las librerías la batería del aparato para poder salvar la información y evitar, también de la misma forma, las pérdidas de los datos ya descargados e información del estado de la descarga.



Figura 58 – Pérdida de conexión a Internet

Una vez descargada la película, será necesario activarla para que el usuario pueda hacer uso de ella. En el momento que se activa, comienza el periodo de alquiler y se guarda la fecha para que la película sea borrada automáticamente del dispositivo una vez expira el tiempo. En caso de que la película estuviese corrupta y fuera imposible la reproducción, el programa detectará el error y borrará automáticamente la película para poder descargarla de nuevo.

El contenido de la película (el detalle) se almacena de manera offline para que el usuario pueda tener acceso tanto a los datos de la película como a la reproducción de esta sin tener que establecer una conexión a Internet. Toda esta información desaparecerá cuando la película sea borrada del iPhone o iPod Touch.



Figura 59 – Película activada

Las descargas no requieren de identificación del usuario, por lo que se puede establecer que las películas quedan ligadas al dispositivo sobre el cual se ha efectuado la descarga y no al usuario que ha adquirido el servicio.

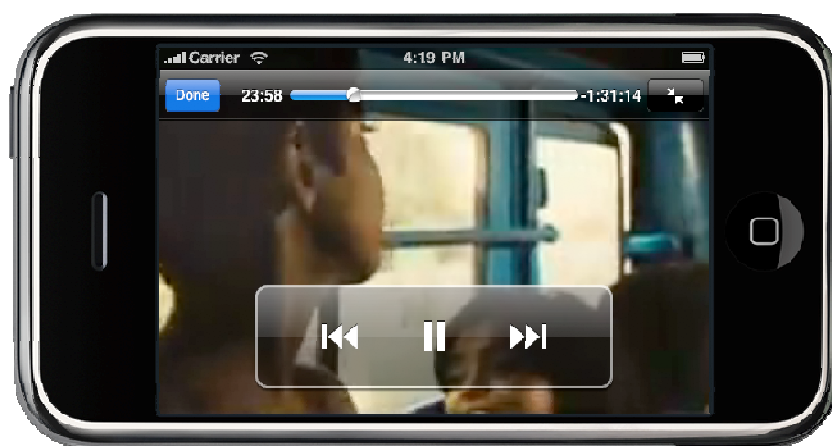


Figura 60 – Reproducción de la película alquilada

#### 7.4.8. Configuración y gestión de datos

iStreamrTV ofrece distintas funcionalidades para la gestión y administración de la aplicación. Desde el menú principal encontramos la configuración y el contacto por email con el soporte.

En el primer caso, desde la configuración, encontramos dos opciones distintas:

- Autologin automático, en el momento que arranca la aplicación, inicia la última sesión iniciada por un usuario.
- Descargas automáticas, cuando una descarga se detiene por la pérdida de conexión, al activar esta opción se reanudan automáticamente las descargas que estaban en curso.

La configuración establecida permanecerá guardada en la memoria del programa para mantenerla cada vez que se inicie iStreamrTV.

En el segundo caso encontramos una opción para ponerse en contacto con el soporte técnico de StreamrTV en el caso de que surja algún problema. El usuario tiene la opción de enviar un correo utilizando la configuración y la cuenta de correo establecida en el gestor de correo integrado en los dispositivos iPhone y iPod Touch.



Figura 61 – Vista configuración



Figura 62 – Descargas automáticas tras conexión reestablecida

## Videoclub iPhone iStreamrTV

En el segundo caso tenemos el envío de emails desde dentro de la aplicación. El usuario puede contactar con el servicio técnico de streamrTV a través de este servicio que por defecto ya lleva la dirección de correo escrita.

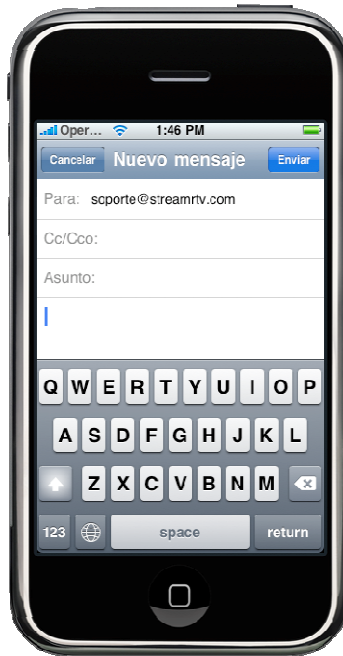


Figura 63 – Envío de email

Como ya se ha mencionado, es necesario tener una cuenta de correo configurada con la aplicación de gestión de correo que ofrece Apple en los dispositivos. De esta forma, el envío de cualquier correo se realizara a través de la cuenta configurada.

Este sistema permite también salvar el texto escrito para poder cargarlos en un futuro y ser enviados cuando se requiera.

### 7.5. Iconos

Existen multitud de empresas que ofrecen venta de iconos y diseño de logos por sitios web aunque también se ofrecen de forma gratuita en otras páginas de la red. Sin embargo, en iStreamrTV se ha optado por generar estas imágenes a medida que se requerían. Para ello ha sido necesaria la generación de distintos logos e iconos.

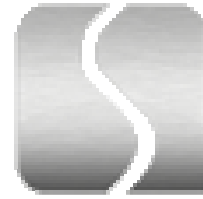


Figura 64 – Logo streamrTV

En el caso de la valoración (rates.png y ratesx3.png), se ha optado por utilizar un mapeado de la imagen dentro del propio programa para obtener el sub-contenido requerido en cada ocasión.

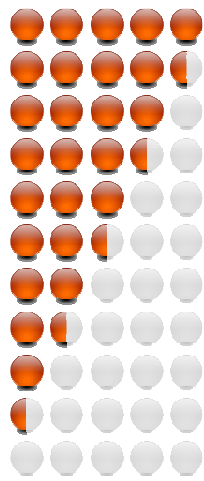


Figura 65 – Imagen selección de voto

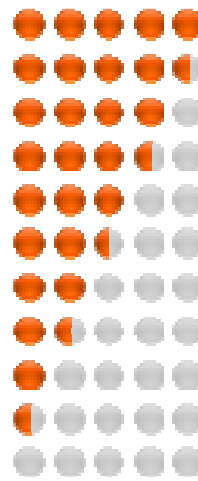


Figura 66 – Imagen valoración película

## 8. Testeo del software

Esta es una de las etapas más importantes dentro del desarrollo de una aplicación y que muestran parte de la calidad de éste. El testeo ha sido realizado de forma sistemática diferenciando las distintas estructuras, de esta forma se garantiza y menor esfuerzo y una mejor efectividad del trabajo y permitiendo un menor coste de mantenimiento en el futuro. Dado que las estructuras están separadas y conectadas entre sí, a través de distintas herramientas, se han testeado por partes.

### 8.1. Base de datos

Para comprobar que la base de datos es sólida y funciona de forma correcta, se ha forzado ésta utilizando masivamente los métodos implementados de forma manual a través de Sequel Pro y órdenes forzadas que actualizan, añaden y borran los datos directamente de la base de datos.

Para comprobar el correcto funcionamiento, sobre dos copias de la base de datos, las mismas órdenes enviadas por la aplicación se han reproducido también de forma manual comprobando después que ambas coincidían en resultados y sin problemas.

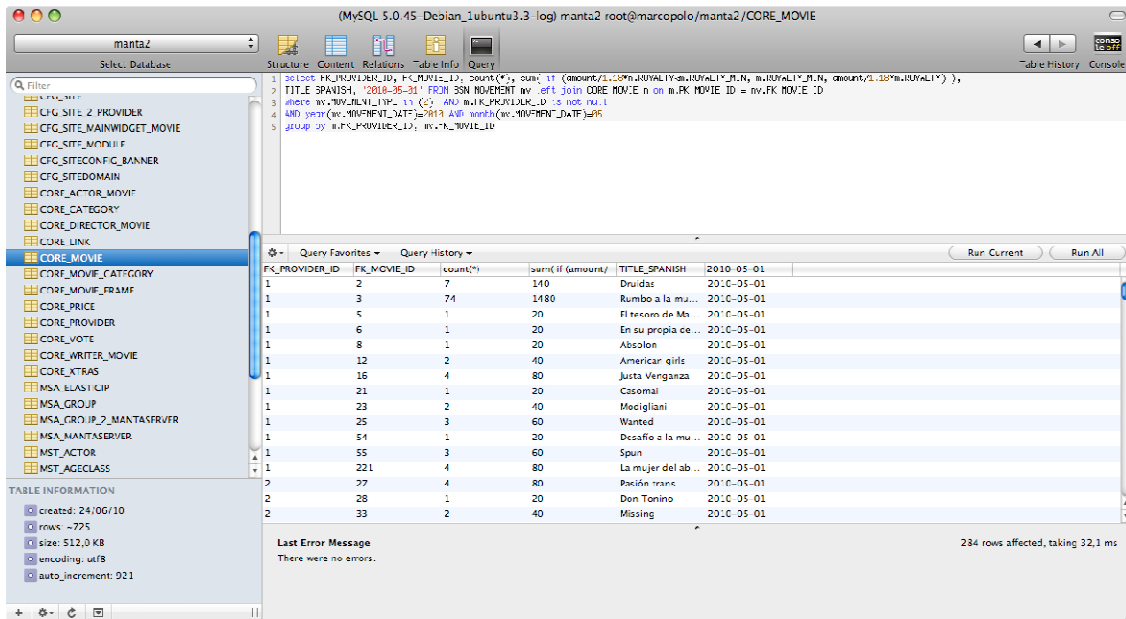


Figura 67 – Pruebas en la base de datos

## 8.2. Estructura iStreamrTV

La gestión de los recursos en un dispositivo móvil es un tema crítico que requiere de la optimización del código dada las carencias (de procesamiento y duración de batería) presentadas por estos aparatos. Es importante realizar un código eficiente y de bajo consumo para conseguir una aplicación estable y fiable en este tipo de plataformas.

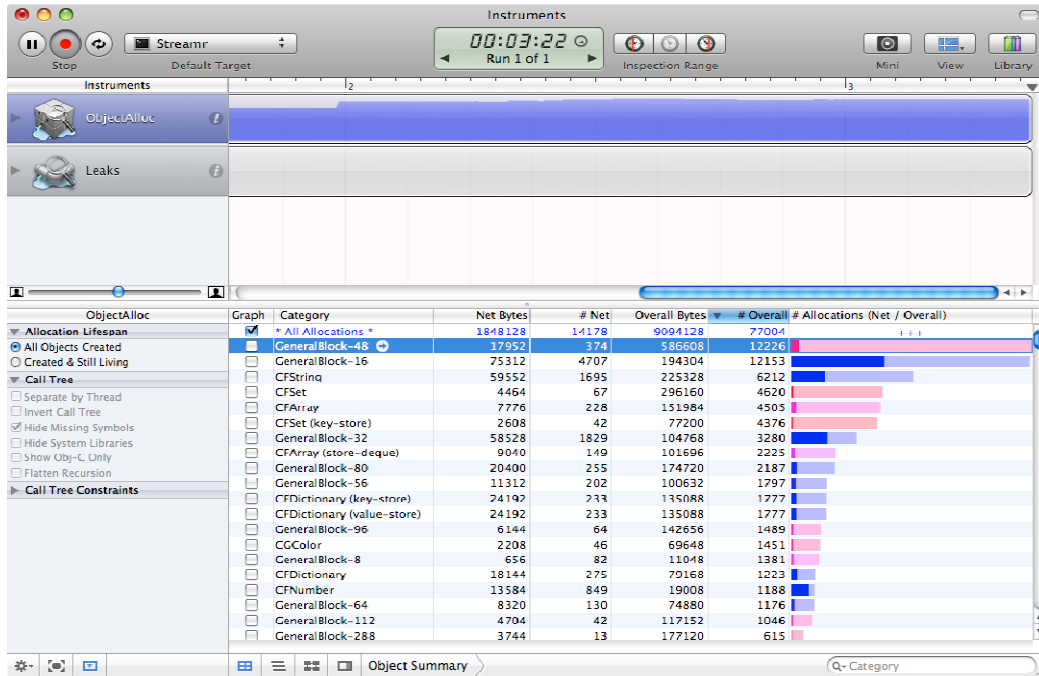


Figura 68 – Instruments sobre iStreamrTV

memoria. En la parte superior encontramos en una barra de tiempo, la cantidad de memoria reservada, siendo la barra inferior los leaks o pérdidas de memoria debido a errores en la programación. Esto se produce cuando hay bloques de memoria no liberados y se han perdido todas las referencias a él. Es importante mantener un programa libre de estas fugas, sobretodo en plataformas móviles donde los recursos son escasos.

La información inferior muestra estadísticas de la memoria reservada y liberada durante la ejecución.

## 8.3. Interfaz de usuario

El testeo de la interfaz de usuario se puede hacer de forma automática o de forma manual. En primer lugar cabe destacar la importancia de realizar todas las pruebas automáticas posibles para después realizar las pruebas manuales necesarias.

Se requiere de un script o código generado que siga unos pasos determinados. En primer lugar será necesario interrogar a la aplicación sobre que se está mostrando en pantalla para capturar los distintos objetos. En segundo lugar es necesaria la selección de estos estados para



posteriormente realizar modificaciones sobre la interfaz. Finalmente, y para tener un script más maduro, es necesario que éste pueda interpretar los resultados obtenidos.

Para la versión 4 del sistema operativo de Apple, se incluirá una clase nueva que permite, a través de sus métodos y captura de eventos, realizar estas pruebas automáticas con mayor fluidez y facilidad (UIAutomation Instrument).

Por otro lado, las pruebas manuales sirven sobre todo para mejorar la apariencia visual de la aplicación, así como su usabilidad. Los usuarios finales y testers son los que deben hacer estas pruebas cuyo fin es mejorar el aspecto y no el buen funcionamiento (finalidad del testeo automático y de las pruebas sobre la estructura explicada en el capítulo 8.2).

## 9. Comparación con aplicaciones similares

El siguiente capítulo abarca aplicaciones que han sido implementadas o diseñadas de una manera similar a iStreamrTV.

### 9.1. App Store

App Store es un servicio creado por la compañía Apple que permite a los usuarios navegar y descargar aplicaciones para sus dispositivos iPhone o iPod Touch.



Figura 69 – App Store

En rasgos generales, iStreamrTV comparte muchas características con App Store, así como la estructura y la disposición de los elementos. A partir de tablas y listados, se muestran los contenidos a los que fácilmente se acceden a los detalles y descripciones, teniendo una navegación bastante similar. Sin embargo, por objetivos, hay distintas funcionalidades muy diferentes ya que iStreamrTV debe trabajar con contenidos y configuraciones offline y con diversas opciones que difieren de la plataforma de Apple, como es la gestión y reproducción de vídeos desde el propio aparato.

## 9.2. mSpot

La siguiente aplicación es también un videoclub para móviles con Android en su sistema operativo (y en desarrollo para los dispositivos de Apple) comercializado en Estados Unidos y Canadá. Aunque inicialmente su funcionamiento era mediante móviles con un navegador que soportara HTML, posteriormente nació la aplicación de funcionalidad pareja.



Figura 70 – Logo mSpot

Además de triviales aspectos en el diseño, mSpot posee una gran diferencia respecto a iStreamrTV, los vídeos se visualizan por streaming únicamente (con las ventajas e inconvenientes que esto supone) mientras que iStreamrTV requiere de una descarga previa completa permitiendo después un visionado offline.

Por otro lado, el pago se realiza mediante una transferencia bancaria sin obligación de registro y a través de la propia aplicación, se efectúa el pago. En el caso de iStreamrTV, se posee una cantidad de créditos asociados a la cuenta registrada y que, realizando una llamada al servidor, se reducen los créditos como ya se ha explicado en capítulos anteriores.

## 10. Posibles mejoras

Según las necesidades del mercado, iStreamrTV se ha trabajado sobre una base estructurada en vista de incluir posibles futuros añadidos. En este capítulo explicaremos algunas de las ampliaciones posibles para la aplicación, así de cómo podrían ser implementadas.

### 10.1. Encriptación/Desencriptación de películas

Las películas alojadas en el servidor se encuentran troceadas para su descarga. El reproductor integrado en el iPhone (y en general en la mayoría de reproductores) se permite la visualización del fichero hasta una posición dañada o corrupta. En ese caso, de ahí en adelante se hace imposible su visualización. Tomando esto como base, la idea sería la encriptación del primer trozo de cada película que a posteriori se podría desencriptar en el propio teléfono a modo de seguridad. Dado que iPhone tiene soporte para la encriptación RSA (Rivest, Shamir y Adleman) se realizaría mediante el uso de la clave pública del cliente la encriptación y la clave privada para su desencriptación posteriormente.

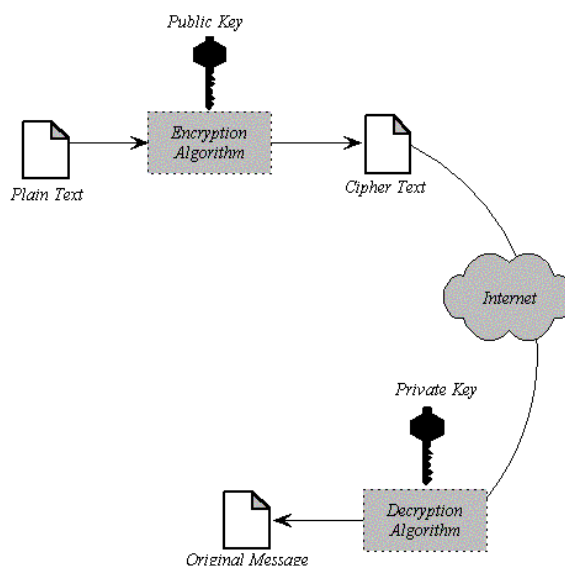


Figura 71 – Esquema básico de la encriptación

### 10.2. Redes sociales

En el último año, el incremento del uso de redes sociales entre la gente ha hecho que las empresas vean éstos como un buen sitio para promocionar sus productos. Actualmente la plataforma web StreamrTV está desarrollando un módulo para que los usuarios puedan insertar y compartir comentarios en los tableros personales de facebook y en la propia página. iStreamrTV podría emularlo mediante una ampliación haciendo uso de "Facebook Connect", una SDK que permite la gestión de datos almacenados en los servidores de facebook, así como el uso de herramientas y funciones para el desarrollo en aplicaciones nativas.

### 10.3. iStreamrTV en el iPad

Aunque en principio pueda parecer que el iPhone y el iPad solo tengan semejanzas en el diseño físico, ciertamente ambos dispositivos comparten un sistema operativo muy similar, iPhone OS4 y iPad OS4, aunque este último incluye ciertas mejoras añadidas para exprimir así las superiores prestaciones que tiene frente al iPhone y iPod Touch.



Figura 72 – iPad

Es por ello que, gracias a la reutilización de código presentada por la arquitectura MVC, construyendo una nueva interfaz adaptada al iPad, sería posible incluir toda la funcionalidad que presenta iStreamrTV. Importando el código lógico, se podrían también incluir nuevas funcionalidades dadas las ventajas que ofrece el nuevo producto de Apple.

## 11. Sumario

El objetivo de esta documentación era mostrar y explicar el diseño y descripción de la aplicación de alquiler de vídeos mediante el iPhone y iPod Touch basada en la plataforma web StreamrTV a la que hemos llamado iStreamrTV.

Para conseguirlo ha sido necesario el seguimiento de un proceso jerárquico, empezando por el estudio de la idea, de las técnicas de desarrollo posibles y de una información base sobre estos procesos. Seguidamente se ha realizado un estudio de los requisitos desde el punto de vista de los usuarios finales para generar una especificación para el software. Todas estas especificaciones han servido para la elección y comprensión de las distintas técnicas y métodos empleados que se han utilizado para la creación del prototipo y primera versión de iStreamrTV.

Durante la investigación de la documentación para el estudio del framework y de la arquitectura estándar del software para este tipo de aplicaciones, no encontré la misma información que se puede encontrar ahora, que es mucho menor que la que se encontrará en un futuro cercano. Y es que durante el tiempo de desarrollo, la todavía explotación de las posibilidades de las herramientas y la documentación asociada a ella no era tan extensa como es ahora, siendo ésta escasa en castellano hasta los actuales días que se empiezan a encontrar mayor número de contenidos.

Dado que el desarrollo de todo el conjunto referente a iStreamrTV constaba de dos partes bien diferenciadas: la aplicación y la programación sobre los servidores, opté por centrar los esfuerzos en crear una interfaz básica funcional partiendo de los modelos referenciales. Una vez que cada parte de la interfaz se implementaba, se pasaba a realizar las llamadas correspondientes al servidor mediante las llamadas JSON. Este hecho provocó que fuera alternando el trabajo entre la aplicación y la preparación de servidores según los casos de uso estudiados. De esta forma, cuando un bloque era funcional, se complementaba con el resto del proyecto haciendo una rutina de programación escalonada y por pasos. La inclusión de clases importadas y desarrolladas para la lectura del formato de los datos y su comprensión por parte del dispositivo fue esencial para avanzar a estas alturas del proyecto.

Completamente implementado, este hecho hará que iStreamrTV sea una aplicación única en el actual mercado español. Basado en la plataforma web de iStreamrTV, esta portación para iPhone y iPod Touch permite tener acceso a un videoclub desde el propio dispositivo móvil, una propuesta todavía no explotada. Por otro lado destacar que los datos se recogen siempre desde internet, lo que hace que la aplicación tenga la información en constante actualización con su "hermana mayor".

## 12. Manual de usuario

Este capítulo tratará de explicar cómo instalar el proyecto iStreamrTV en uno de los dispositivos compatibles. Si bien es cierto que existirán dos formas de hacerlo, mediante la descarga a través de la App Store o a partir de una compilación del proyecto sobre el móvil, solamente la segunda opción es, en estos momentos, la que funciona.

### 12.1. Mediante la App Store

Como hemos comentado con anterioridad, la App Store es un soporte mediante la cual se pueden descargar aplicaciones directamente para el iPhone o iPod Touch. Estas aplicaciones están disponibles para ser compradas y libres de costo, dependiendo de cada una.

Para que la aplicación se encuentre en la lista es necesario poseer una licencia de desarrollo y enviar el proyecto previo relleno de un formulario o contrato. Después de esperar a una comprobación de contenidos y aceptación del software, se recibe un código de activación que será necesario para poner finalmente el programa en la lista y a disposición del usuario.

Actualmente iStreamrTV no se encuentra todavía en la App Store aunque se espera que en un breve periodo de tiempo esté disponible a petición de la empresa REDAMM.

### 12.2. Mediante el simulador

A través del programa "Simulador iPhone" es posible ejecutar iStreamrTV desde el propio Mac. Para ello es necesario abrir el proyecto con el XCode a través del fichero "Streamr.xcodeproj". Una vez abierto se debe tener instalado el "Simulador iPhone", que viene integrado por defecto en el entorno de desarrollo de XCode. Con una configuración de compilación sobre el simulador con la versión 3.1.2, solo es necesario compilar y ejecutar el proyecto. Si el ordenador posee conexión a Internet, es posible realizar todas las opciones online de iStreamrTV sin ningún otro tipo de configuración previa.

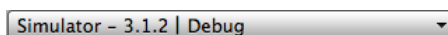


Figura 73 – Compilación sobre el simulador

### 12.3. Mediante compilación sobre el dispositivo

Para la compilación e instalación de iStreamrTV sobre un dispositivo iPhone o iPod Touch es necesario seguir los mismos pasos planteados que en el simulador. Sin embargo, antes de esto es necesario que la ID del dispositivo donde se quiere realizar la instalación debe estar registrada en la cuenta de desarrollador para Apple. Además, el propio ordenador debe tener instalados los certificados (que proveen unas llaves de identificación) donde el desarrollador aprueba que esa máquina es válida para poder transferir el programa.

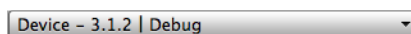


Figura 74 – Compilación sobre el dispositivo

Finalmente, una vez configurado, habría que marcar la opción de compilación "iPhone Device" sobre la versión 3.1.2 y ejecutarlo. Por supuesto, el dispositivo previamente ha de ser enchufado mediante un cable USB al ordenador. Una vez se transfiere la aplicación, ya no es necesario volver a realizar la instalación ya que el propio aparato ya contendrá iStreamrTV.

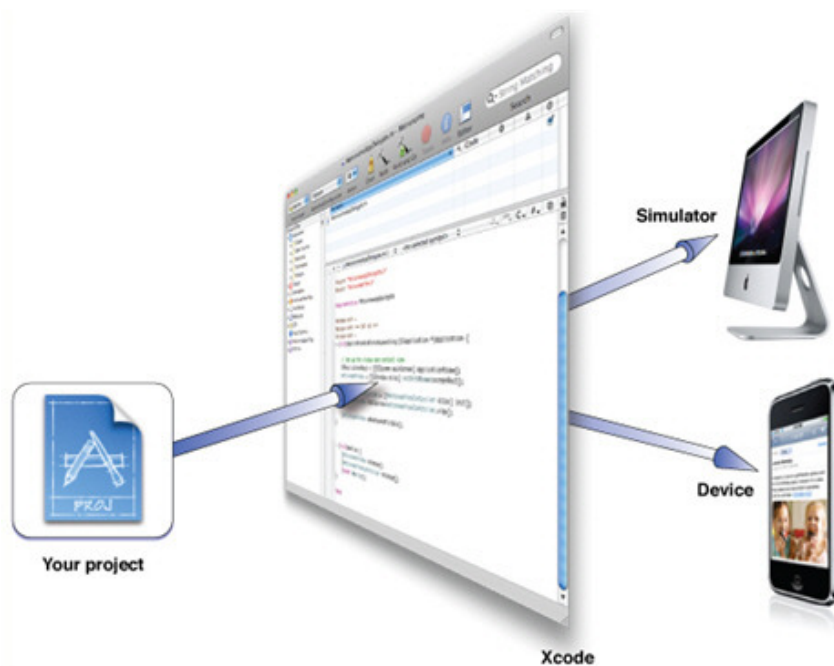


Figura 75 – Diagrama de instalación del proyecto



### 13. Software utilizado

Para la realización del proyecto y de todas las tareas planteadas se han utilizado los siguientes programas, cada uno con su propio fin.

#### 13.1. Software para la aplicación

- **XCode**, entorno de desarrollo integrado (IDE, en sus siglas en inglés) de Apple. Utilizado para la programación en Objective-C del proyecto iStreamrTV.
- **Interface Builder**, constructor de interfaces que trabaja conjuntamente con XCode para permitir la conexión de los objetos de la vista con el modelo lógico.
- **Simulador iPhone**, simulador sobre el que se han hecho todas las pruebas antes de testear sobre dispositivos reales.
- **Adobe PhotoShop CS4**, herramienta de dibujo donde se han generado las imágenes necesarias para completar el interfaz.
- **Instruments**, programa con distintas funciones que examinan y analizan el rendimiento de las aplicaciones creadas.

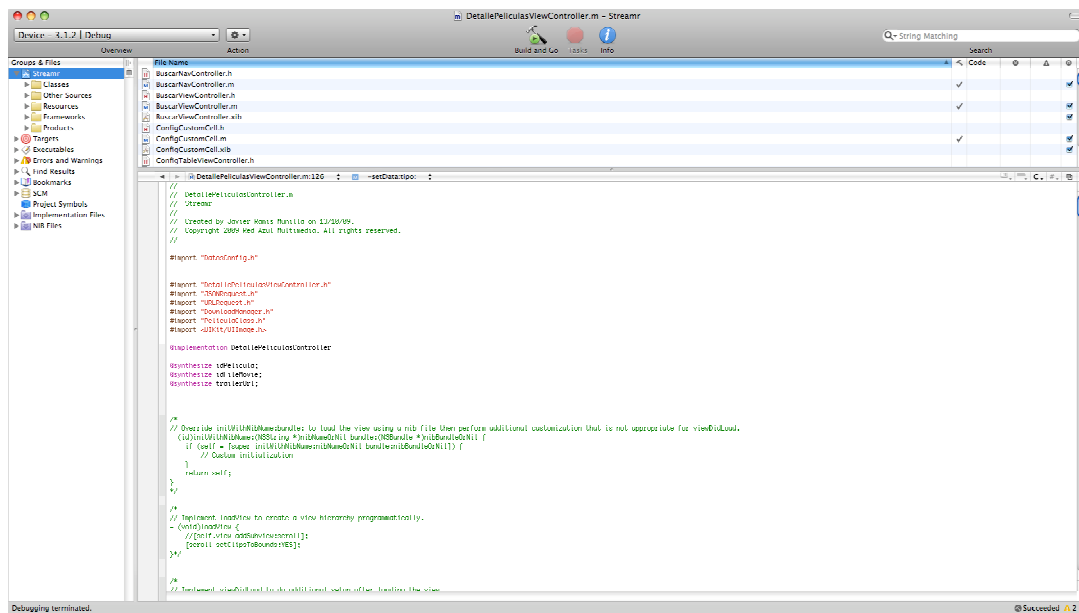


Figura 76 – Entorno de trabajo Xcode

### 13.2. Software para los servidores

- **Eclipse**, entorno de desarrollo integrado de código abierto multiplataforma sobre el que se ha preparado, mediante programación en Java, a los servidores para establecer llamadas JSON entre estos y los dispositivos de Apple.
- **Sequel Pro**, gestor de base de datos en MySQL gratuito para Mac, sobre el que se ha preparado la base de datos según los requerimientos necesitados por este proyecto.

## 14. Bibliografía

- DALRYMLPLE, Mark y KMASTER, Scott. *Learn Objective-C on the Mac*. Nueva York, Apress. 2009.
- MARK, Dave y LAMARCHE, Jeff. *Beginning iPhone Development: Exploring the iPhone SDK*. Nueva York, Apress 2009.
- FROUFE QUINTAS, Agustín. *Java 2 (4ª edición) Actualizada a la versión J2SE 5*. Madrid, Alfa Omega. 2000.
- Apple Dev [Página web]  
[Visitada: 01/2010 - 02/2010]  
<http://developer.apple.com/>
- iPhone Dev [Página web]  
[Visitada: 12/2009 - 02/2010]  
<http://www.iphonedevsdk.com/>
- MacProgramadores [Página Web]  
[Visitada 12/2009]  
<http://macprogramadores.org/>
- MySQL Dev [Página Web]  
[Visitada: 02/2010]  
<http://dev.mysql.com/>
- Servlets [Página Web]  
[Visitada 02/2010]  
[http://java.sun.com/j2ee/tutorial/1\\_3-fcs/doc/Servlets.html](http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html)

- Servlets 2 [Página Web]  
[Visitada 02/2010]  
*[http://www.programacionfacil.com/java\\_servlets/start](http://www.programacionfacil.com/java_servlets/start)*
  
- StackOverFlow [Página Web]  
[Visitada: 12/2009 - 02/2010]  
*<http://stackoverflow.com/>*
  
- Stanford iPhone Dev [Página Web]  
[Visitada: 12/2009 - 01/2010]  
*<http://www.stanford.edu/class/cs193p/cgi-bin/drupal/downloads-2010-winter>*
  
- Wikipedia [Página Web]  
[Visitada: 02/2010]  
*<http://es.wikipedia.org/>*
  
- Youtube [Videotutoriales]  
[Visitada: 01/2010]  
*<http://www.youtube.com/>*

## 15. Adjuntos

### 15.1 Listado de las ilustraciones

Figura 1: Evolución del dispositivo móvil .....	13
Figura 2: Gráfico de ventas del iPhone en España en el año 2008 .....	14
Figura 3: Mercado global de dispositivos Smartphone durante los años 2007 y 2009 .....	15
Figura 4: Características iPhone y iPod Touch .....	16
Figura 5: Estructura de una aplicación nativa en iPhone .....	17
Figura 6: Cocoa en la arquitectura iPhone OS.....	18
Figura 7: Cocoa Touch.....	19
Figura 8: Estructura de los frameworks de Cocoa .....	19
Figura 10: Cocoa AppKit FrameWork .....	20
Figura 11: Resumen de las clases más importantes contenidas en el framework UIKit.....	20
Figura 12: Cocoa Foundation FrameWork .....	21
Figura 13: Tabla resumen de las diferencias entre Cocoa y Cocoa Touch .....	21
Figura 14: Esquema MVC .....	22
Figura 15: Logo streamrTV .....	24
Figura 16: Esquema funcionalidad iStreamrTV .....	24
Figura 17: iPhone.....	25
Figura 18: Diagrama casos de uso. Actores.....	26
Figura 19: Caso de uso. Acceso (gestión de sesión).....	27
Figura 20: Casos de uso. Listado y gestión online .....	30
Figura 21: Casos de uso. Configuración y gestión offline.....	37
Figura 22: Casos de uso. Gestión periódica .....	42
Figura 23: Tabla tipos de fichero y sus extensiones.....	45
Figura 24: Ejemplo de código en Objective-C .....	46
Figura 25: Estructura de clases de la interfaz.....	47
Figura 26: Interface Builder.....	48
Figura 27: Esquema de la arquitectura del proyecto.....	50
Figura 28: Conexión de los módulos .....	50
Figura 29: Capas del servidor iStreamrTV .....	52
Figura 30: Modelo de clases controladoras de vistas .....	53
Figura 31: Modelo de clases lógicas.....	54
Figura 32: Modelo de clases para la navegación.....	55
Figura 33: Modelo de clases para las celdas .....	55
Figura 34: Diagrama de secuencia. Iniciar sesión.....	56
Figura 35: Diagrama de secuencia. Listar películas.....	57
Figura 36: Diagrama de secuencia. Alquilar película.....	58
Figura 37: Aplicación iStreamrTV en iPhone .....	59
Figura 38: Menú de navegación.....	59
Figura 39: Menú Mi cuenta (principal).....	60
Figura 40: Menú Mi cuenta (principal) con sesión iniciada .....	60
Figura 41: Vista inicio de sesión .....	61
Figura 42: Error en el inicio de sesión .....	61
Figura 43: Vista carga de listado .....	62
Figura 44: Intento de listado sin inicio de sesión .....	62
Figura 45: Listado destacadas .....	63
Figura 46: Listado favoritos (por género).....	63

Figura 47: Listado búsqueda .....	63
Figura 48: Vista confirmación de alquiler .....	63
Figura 49: Mensaje película alquilada .....	63
Figura 50: Mensaje no se dispone de créditos suficientes .....	64
Figura 51: Votación de una película .....	64
Figura 52: Votación efectuada .....	64
Figura 53: Tráiler (streaming).....	65
Figura 54: Diagrama de estados de descarga película .....	65
Figura 55: Vista detalle de descarga .....	66
Figura 56: Vista detalle película en descarga .....	66
Figura 57: Vista detalle película descargada .....	66
Figura 58: Pérdida de conexión a Internet.....	66
Figura 59: Película activada.....	67
Figura 60: Reproducción de la película alquilada.....	67
Figura 61: Vista configuración.....	68
Figura 62: Descargas automáticas tras conexión reestablecida .....	68
Figura 63: Envío de email .....	69
Figura 64: Logo streamrTV .....	70
Figura 65: Imagen selección de voto.....	70
Figura 66: Imagen valoración película.....	70
Figura 67: Pruebas en la base de datos.....	71
Figura 68: Instruments sobre iStreamrTV .....	72
Figura 69: App Store.....	74
Figura 70: Logo mSpot.....	75
Figura 71: Esquema básico de la encriptación .....	76
Figura 72: iPad.....	77
Figura 73 – Compilación sobre el simulador .....	79
Figura 74 – Compilación sobre el dispositivo.....	80
Figura 75: Diagrama de instalación del proyecto.....	80
Figura 76 – Entorno de trabajo Xcode .....	81

