Módulo de streaming de Picta para el televisor inteligente cubano con sistema operativo GNU/Linux Nova

Facultad 1

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Esteban Manuel Landrian Amaro

Tutores: M. Sc. Juan Manuel Fuentes Rodríguez

Ing. Aldy León García

Ing. Javier Junquera Falcón

La Habana, mayo 2022

Año 65 de la Revolución

Cuando alguien ama lo que hace se nota, cuando no amas lo que haces se nota aún más.

Steve Jobs

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis que tiene por título: Módulo de streaming de Picta para el televisor inteligente y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Esteban Manuel Landrian Amaro

_____

Firma del Autor

Ing. Aldy León García

_____

Firma del Tutor

Ing. Javier Junquera Falcón

_____

Firma del Tutor

M. Sc. Juan Manuel Fuentes Rodríguez

_____

Firma del Tutor

DATOS DE CONTACTO

Autores:

Esteban Manuel Landrian Amaro

Universidad de las Ciencias Informáticas (UCI)

e-mail: emlandrian@estudiantes.uci.cu

Tutor:

Ing**.** Aldy León García

Universidad de las Ciencias Informáticas (UCI)

e-mail: algarcia@uci.cu

Tutor:

Ing**.** Javier Junquera Falcón

Universidad de las Ciencias Informáticas (UCI)

e-mail: jejunquera@uci.cu

Tutor:

M. Sc. Juan Manuel Fuentes Rodríguez

Universidad de las Ciencias Informáticas (UCI)

e-mail: jfuentesr@uci.cu

AGRADECIMIENTOS

A mi madre, a mi hermana,a mi sobrino y a papá que a pesar de no siempre estar presente se siente muy orgulloso de que su hijo termine los estudios y tenga un título como profesional.

A mi tía Roxanita por ser mi referente de vida y el verdadero ejemplo de superación.

A mi pareja Maylin por apoyarme en todo y ser capaz de sacarme una sonrisa en mis peores momentos. Por no soltarme la mano cuando más lo he necesitado.

A mi suegra, casi una madre para mí, Lazara Linares, gracias por haberme aportado tanto en tan poco tiempo que llevamos conociéndonos, y gracias a toda esa familia que se han portado tan bien conmigo.

A mi familia por compartir mis logros, en especial a mis primos Raidel y Yamilka

A mi profesor de la vida Mauricio el tinieblo, gracias por su apoyo, regaños y constate ánimos.

A mi familia de la UCI, Emilio, Jasan, Karlucha, Roxy, claudia mi venenito, Pochy, Leo el colombófilo, Leandro, Suset, Osvaldo, Miguel Luis sharkboy Adam Medina, Johan Bravo, Armin Vázquez, Yuniel, El Friky,El kaneka, Daniel la salsa, Enrico mi pelú favorito, Roger shelby, Letuce, Dutel,El emba ,Hairo,El Loba.

A mi grupo Nefilim, que rompimos siempre, el mejor grupo de baile de la Uci.

A mi Tim que los llamo los machis: Leonel Eduardo, Kilmer, Enier la eminencia, Ruru ,Gallardo, Ferni, Jasiel por darme todos esos buenos momentos desde un café hasta un full party.

A mis cuñis Yesenia, Alianna, Rusmery.

A los amigos de siempre, a los amigos UCI, a los amigos de la vida, que siguieron página a página esta investigación desde un café, un cigarro, un dominó, una llamada.

¡A todos Gracias!

DEDICATORIA

A todas aquellas personas que nunca dejaron de creer en mí en especial a todos que cuando tuvieron la oportunidad de ayudarme lo hicieron, en especial a mi familia, a mi compañera de vida y amigos.

RESUMEN

Los avances de las Tecnologías de la Información y la Comunicación, han alcanzado todos los sectores de la vida, principalmente en los dispositivos que usan internet; aportando a las personas numerosos beneficios, ya sea por trabajo, entretenimiento o comunicación. Entre todos estos dispositivos que forman parte del internet de las cosas, se tienen los televisores inteligentes o Smart tv como también se les conoce. También existen plataformas que facilitan el acceso a contenido audiovisual como lo es Picta. Actualmente no se puede acceder a los contenidos de Picta desde los televisores inteligentes cubanos haciendo uso de la internet. La presente investigación se propone, desarrollar un módulo de conexión que sea capaz de consumir servicios de internet de Picta y poder visualizarlos en los televisores inteligentes fabricados en nuestro país. La propuesta está concebida como una opción complementaria a este proceso, con apoyo en los fundamentos de módulos de conexión y los beneficios que estos proponen. Para su desarrollo se concibe la unión de las interfaces de Kodi y el uso de la API de Picta, siguiendo la metodología AUP-UCI, Visual Paradigm para UML v15.1 como herramienta de modelado, como lenguaje de programación Python 3.5, Visual Studio Code como IDE de desarrollo. Se obtuvo como resultado un módulo de conexión que consume los servicios de *streaming* en los televisores inteligentes cubanos con sistema operativo NOVA diseñado sobre la base de los patrones GRASP. Además, se le aplicaron pruebas a nivel de unidad, funcionalidad e integración, para las cuales se obtuvo resultados satisfactorios**.**

Palabras clave: módulo, televisor inteligente, *streaming*

ABSTRACT

Advances in Information and Communication Technologies have reached all sectors of life, mainly in devices that use the Internet; providing people with numerous benefits, whether for work, entertainment or communication. Among all these devices that are part of the internet of things, there are smart televisions or Smart TV as they are also known. There are also platforms that facilitate access to audiovisual content, such as Picta. Picta content cannot currently be accessed from Cuban smart televisions using the internet. The present investigation proposes to develop a connection module that is capable of consuming Picta internet services, which is currently the quintessential audiovisual platform in Cuba, and to be able to view them on smart televisions manufactured in our country. The proposal is conceived as a complementary option to this process, based on the fundamentals of connection modules and the benefits that they propose. For its development, the union of the Kodi interfaces and the use of the Picta API is conceived, following the AUP-UCI methodology, Visual Paradigm for UML v15.1 as a modeling tool, as a programming language Python 3.5, Visual Studio Code as development IDE. As a result, a connection module was obtained that consumes streaming services in Cuban smart televisions with the NOVA operating system designed on the basis of the GRASP and GOF patterns. In addition, tests were applied at the unit, functionality and integration level, for which satisfactory results were obtained.

Keywords: *module, smart TV, streaming*

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) son parte indispensable de la sociedad, son tecnologías que utilizan la informática, la microelectrónica y las telecomunicaciones para crear nuevas formas de comunicación a través de herramientas de carácter tecnológico y comunicacional, esto con el fin de facilitar la emisión, acceso y tratamiento de la información (1). De manera cotidiana y en aspectos tan importantes como el entorno familiar, escolar y social. Su uso extensivo ha revolucionado las formas tradicionales de acceder a la información, de comunicarse y de relacionarse, facilitando mucho las tareas del día a día.

La evolución tecnológica siempre ha significado un avance en los procesos de comunicación humana, la tecnología y la comunicación han soportado uno a uno los escalones de la evolución natural de nuestra especie. La búsqueda del ser humano para mejorar su forma de vida ha favorecido la constante innovación (2).

El televisor se ha convertido en un aparato electrodoméstico habitual, cotidiano y normal con amplia presencia en los hogares de todo el mundo. La masificación de este aparato se considera el inicio de la era de las telecomunicaciones en la humanidad y entre sus funciones tiene a la televisión como medio de comunicación(3).

La televisión es un sistema de emisión y recepción a distancia de sonido e imágenes a través de ondas hercianas**,** juega un rol preponderante entre los medios de comunicación, siendo el medio de comunicación masiva por excelencia (4). El televisor es el aparato idóneo para recibir y visualizar dichas ondas. Estos dispositivos se han adaptado a los nuevos tiempos y a la revolución digital de manera extraordinaria, han cumplido con las expectativas y han satisfecho nuestras necesidades.

Un televisor inteligente es capaz de conectarse a Internet y utilizar la conexión doméstica para el consumo de servicios. El sistema operativo de estos da la posibilidad de actualizar e instalar aplicaciones. Es posible usar redes sociales, servicios online, gestionar una biblioteca multimedia y mantenerse informado.

La Universidad de las Ciencias Informáticas posee una alianza con la Empresa Industrial para la Informática, las Comunicaciones y la Electrónica (GEDEME). La soberanía tecnológica en Cuba está estrechamente ligada al *software* y al *hardwar*e libre y los concibe como recursos, herramientas y medios para construir y ejercer dicha soberanía, que a pesar de las restricciones impuestas por el bloqueo de Estados Unidos a la Revolución cubana, el proceso de globalización no ha dejado de impulsar las nuevas tecnologías para Cuba. Las computadoras, laptops, tabletas, televisores inteligentes y celulares por la alianza estratégica, serán sumistradas con la distribución cubana de GNU/Linux Nova. El televisor inteligente cubano como proyecto está definido como un cliente ligero con la capacidad de ejecutar el sistema operativo GNU/Linux Nova. La proyección de este sería a través de un dispositivo de salida multimedia, siendo este un monitor, televisor o proyector que cuente con entrada de video HDMI o VGA.

El televisor inteligente cubano utiliza como gestor multimedia y de servicios de internet el *software* de código libre Kodi siendo este multiplataforma. Es ejecutable en Android, Microsoft Windows, Mac OS y GNU/Linux. Es un completo gestor multimedia que permite reproducir múltiples formatos de video y audio. Entre sus virtudes permite jugar juegos, consumir servicios de *streaming*, leer noticias de los órganos de prensa internacionales y consultar sus redes sociales preferidas. Este conjunto de funcionalidades se brindan a través módulos desarrollados por la propia comunidad de código libre de Kodi ofreciendo posibilidades infinitas de agregar funcionalidades a Kodi.(5)

Actualmente las posibilidades que ofrece este centro multimedia crecen a medida que más desarrolladores centran su atención en extender sus funcionalidades. Una de las novedades que destaca es poder añadir módulos que consuman servicios de Internet, sean estos Netflix, YouTube o cualquier servicio de transmisión de video conocido. La plataforma de *streaming* Picta es un producto cubano de excelencia en el campo audiovisual. Picta como producto nacional tiene como visión llegar a todos los ámbitos sociales, escuelas, hogares, centros de trabajo y de ocio.

La plataforma de audiovisuales Picta cuenta con dos variantes de acceso actualmente: a través de su plataforma web y mediante su aplicación móvil haciendo uso de una API. Con el surgimiento de los televisores inteligentes cubanos y el marcado crecimiento de los usuarios

de Picta, surge la variante de conexión entre los televisores y la plataforma para consumir servicios de *streaming*, funcionalidad no implementada hasta el momento de la realización de la presente investigación.

Teniendo en cuenta lo antes planteado se identifica el siguiente **problema de la investigación**: ¿Cómo consumir los servicios de *streaming* de la plataforma Picta desde los televisores inteligentes cubanos con sistema operativo Nova?

Se define como **objeto de estudio**: Los métodos para consumir servicios de *streaming* de las plataformas de audiovisuales, enmarcado en el **campo de acción:** Los métodos para consumir servicios de *streaming* en Picta. El presente trabajo tiene como **objetivo general:** Desarrollar un módulo para consumir los servicios de *streaming* de la plataforma Picta para los televisores inteligentes con sistema operativo Nova.

Los **objetivos específicos** son**:**

1. Elaborar el marco teórico de la investigación sobre el proceso de consumir los servicios de *streaming* de la plataforma Picta en los televisores inteligentes.
2. Diseñar un módulo para consumir los servicios de *streaming* de Picta para televisores inteligentes.
3. Implementar un módulo para consumir los servicios de *streaming* de Picta para televisores inteligentes.
4. Evaluar el módulo de consumo de servicios de *streaming* de Picta para televisores inteligentes mediante la aplicación de técnicas de pruebas.

Para dar cumplimiento al objetivo planteado se definen las siguientes **tareas de investigación:**

- Realización del marco teórico de la investigación a partir de los elementos que intervienen en el desarrollo de los métodos para consumir servicios de *streaming* en Picta.
- Caracterización de los módulos para consumir servicios de *streaming* de las plataformas audiovisuales.

- Identificación los requisitos funcionales y no funcionales a tener en cuenta en la propuesta de solución.
- Implementación del módulo para consumir los servicios de *streaming* de la plataforma Picta.
- Validación del módulo para consumir los servicios de *streaming* de la plataforma Picta.

Para dar cumplimiento a las tareas de la investigación se utilizaron los siguientes **métodos científicos:**

Métodos teóricos:

- ✦ Analítico –Sintético: mediante este método se llevó a cabo la construcción del marco teórico de esta investigación, partiendo del análisis de los referentes teóricos que están relacionados con el consumo de servicios de *streaming* de Picta para televisores inteligentes, se hizo una síntesis de los elementos más relevantes.
- ✦ Modelación: se utiliza para el proceso de diseño, la creación de modelos, propuestas y alternativas mediante la abstracción de sus elementos fundamentales, se representan las características, relaciones y componentes de la solución. Se evidencia en la modelación de los artefactos ingenieriles generados a partir de la metodología definida.
- ✦ Histórico -Lógico: después de un estudio de la evolución de las tecnologías de la información y las comunicaciones y la adaptabilidad de los televisores a estas tecnologías, se hizo una síntesis de los elementos más relevantes e históricos que aportan al desarrollo de dicha investigación, referente a el consumo de los servicios de *streaming* de diferentes aplicaciones en televisores inteligentes.

Método empírico:

- ✦ Entrevista: Se aplicó a los trabajadores del proyecto Picta para conocer la necesidad del desarrollo de una propuesta que dé solución a la problemática, además para definir sus funcionalidades, identificar las particularidades y las restricciones que se imponen (Ver Anexo 1).
- ✦ Observación: este método se utilizó en el análisis del proceso de consumo de servicios de *streaming* de las diferentes plataformas de audiovisuales que existen en la

actualidad. Para su aplicación se utilizó la guía de observación para el proceso de creación de un módulo que haga uso de servicios de *streaming*.

Este informe está conformado por introducción, tres capítulos, conclusiones, referencias bibliográficas y anexos. Los capítulos se estructuran como se muestra a continuación:

Capítulo 1: Fundamentos de la investigación, metodología y tecnologías empleadas para la realización del módulo de consumo de servicios de streaming de Picta.

En este capítulo se realiza un estudio de los fundamentos teóricos del ámbito televisivo, se definen conceptos asociados al mismo y se analizan soluciones homólogas para que en conjunto con las herramientas y la metodología escogida, proponer una posible solución al problema de investigación.

Capítulo 2: Análisis y diseño de la propuesta de solución

En este capítulo se realiza la exploración e inicialización de la solución propuesta de acuerdo a la metodología de desarrollo de software. Se describe el modelo de dominio, que permite una mejor comprensión de la solución. Se enumeran y detallan los requisitos funcionales y no funcionales, agrupados en historias de usuario, generando una visión para la creación del módulo a desarrollar. Se escoge la arquitectura que se debe seguir para el desarrollo de la misma, la cual brindará una organización básica para la implementación. Además, se definen los patrones de diseño a utilizar.

Capítulo 3: Desarrollo y validación de la solución propuesta

En este capítulo se realiza el diagrama de componente aportando una mejor compresión para el desarrollo del módulo. Se definen los estándares de codificación usando las reglas de PEP 8. Además, se realizan pruebas a niveles de unidad, de sistema y de integración, finalizando en un análisis de los resultados de las mismas, que permiten la validación de la solución propuesta.

CAPÍTULO I: Fundamentos de la investigación, metodología y tecnologías empleadas para la realización del módulo de streaming de Picta.

En este capítulo se abordan los conceptos asociados a esta investigación, y se realiza un análisis de sistemas homólogos para fijar ciertos criterios de comparación, se definirán las tecnologías empleadas y se identificará la metodología más acorde para la solución de la problemática. Para lograr una mejor explicación del contexto del problema, se definen conceptos que son fundamentales a tener presente para el desarrollo del módulo de consumo de servicios de *streaming* en los televisores inteligentes cubanos con el sistema operativo Nova.

1.1 Conceptos fundamentales

El **televisor** es un aparato electrónico, permite la recepción y reproducción de señales de televisión (6).

Un **televisor inteligente** cuenta con un sistema operativo, puede conectarse a internet y descargar multitud de aplicaciones. Esto permite realizar funciones no disponibles en un TV normal según define (7). Se caracteriza por estar preparado para ofrecer diversos servicios digitales. De este modo, en un televisor inteligente, una persona puede buscar contenidos en la *Web* y visualizarlos en la pantalla. También cuenta con la posibilidad de grabar en vivo programas en un disco duro y de interactuar, de diversos modos, con los canales de televisión a través de distintas aplicaciones (8). Los televidentes pueden ver contenido multimedia desde internet sin necesidad de ver transmisiones estándar lineales. Por supuesto, la navegación *web* no es muy cómoda a través de un control remoto, por lo tanto los televisores inteligentes ofrecen una funcionalidad para conectar varios dispositivos inalámbricos (9).

Dentro de las características de los televisores inteligentes está el uso de los **sistemas operativos**.(11) define un sistema operativo como el software que coordina y dirige todos los servicios y aplicaciones que utiliza el usuario, por eso es el más importante y fundamental. Se trata de programas que permiten y regulan los aspectos más básicos del sistema. Los sistemas operativos, también llamados núcleos o *kernels*, suelen ejecutarse de manera privilegiada respecto al resto del *software*, sin permitir que un programa cualquiera realice cambios de

importancia sobre él que puedan comprometer su funcionamiento. Los sistemas operativos más utilizados son Windows, Linux, OS/2 y DOS.

Una **plataforma virtual** o **plataforma *web*** es un sistema que permite la ejecución de diversas aplicaciones bajo un mismo entorno, dando a los usuarios la posibilidad de acceder a ellas a través de Internet. Esto quiere decir que, al utilizar una plataforma virtual, el usuario no debe estar en un espacio físico determinado, sino que sólo necesita contar con una conexión a la *Web* que le permita ingresar a la plataforma en cuestión y hacer uso de sus servicios(10).

El **servicio *streaming*** se refiere a cualquier contenido de medios, ya sea en vivo o grabado, que se puede disfrutar en computadoras, televisores inteligentes y aparatos móviles a través de Internet y en tiempo real. Las películas, los programas de TV y los videos musicales son tipos comunes de contenido de *streaming*. Los archivos de música, video y otros tipos de archivos de medios se preorganizan y transmiten en paquetes secuenciales de datos, a fin de que se pueda hacer *streaming* de ellos de forma simultánea. Y, a diferencia de las descargas tradicionales que se guardan en el dispositivo que se esté utilizando, los archivos de medios se eliminan automáticamente luego de reproducirlos. Todo lo que se necesita para hacer *streaming* es una conexión a Internet de alta velocidad rápida y confiable, acceso o suscripción a un servicio o aplicación para hacer *streaming* y un aparato compatible (12).

Nova es una distribución GNU/Linux creada por la Universidad de las Ciencias Informáticas (UCI) en Cuba. Esta forma de distribución se establece bajo la plataforma de *software* libre y código abierto, que trata de independizar al país cubano de la adquisición de *software* con licencias pagas. Siendo más específicos, la idea principal fue desarrollar una distribución compatible con el *software* manejado en la universidad (UCI) (13). Un sistema operativo está conformado básicamente por cuatro módulos:

- Núcleo o *Kernel*.
- Administrador de memoria.
- Sistema de entrada/salida.
- Administrador de archivos.

De acuerdo a (14) un **módulo** es un componente de *software* o parte de un programa que contiene una o más rutinas. Uno o más módulos desarrollados independientemente forman un programa. Una aplicación de *software* de nivel empresarial puede contener varios módulos diferentes, y cada módulo sirve operaciones comerciales únicas y separadas. Los módulos facilitan el trabajo de un programador al permitir que el programador se concentre en un solo área de la funcionalidad de la aplicación de *software*. Los módulos se incorporan típicamente en el programa (*software*) a través de interfaces.

De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas. Particularmente, en el caso de la programación, los módulos suelen estar (no necesariamente) organizados jerárquicamente en niveles, de forma que hay un módulo principal que realiza las llamadas oportunas a los módulos de nivel inferior.

Cuando un módulo es convocado, recibe como entrada los datos proporcionados por otro del mismo o superior nivel, el que ha hecho la llamada; luego realiza su tarea. A su vez este módulo convocado puede llamar a otro u otros módulos de nivel inferior si fuera necesario; cuando ellos finalizan sus tareas, devuelven la salida pertinente al módulo inmediato llamador, en secuencia reversa, finalmente se continúa con la ejecución del módulo principal (15).

1.2 Análisis de streaming de Picta

La plataforma de contenidos audiovisuales Picta desarrollada por la Universidad de Ciencias Informáticas (UCI), es uno de los repositorios y servicios de *streaming* más importante del país. El sitio ofrece dos servicios principales: televisión en vivo a través de *streaming* de video y la posibilidad de visualizar y descargar audiovisuales. Por la facilidad del dominio (.cu), la mayor recomendación es visionar en vivo o desde el propio espacio. Pero si se desea disponer una y otra vez del mismo material, lo aconsejable es descargar para lo cual tienen varias opciones de alta, media o baja resolución.

Actualmente la plataforma no cuenta con un módulo capaz de consumir servicios de *streaming* desde un televisor inteligente.

1.2.1 Análisis de módulos de streaming

Los archivos de música, video y otros tipos de archivos de medios se organizan y transmiten en paquetes secuenciales de datos, a fin de que se pueda hacer *streaming* de ellos de forma simultánea. Y, a diferencia de las descargas tradicionales que se guardan en el dispositivo que se esté utilizando, los archivos de medios se eliminan automáticamente luego de reproducirlos. Todo lo que se necesita para hacer *streaming* es una conexión a Internet de alta velocidad rápida y confiable, acceso o suscripción a un servicio o aplicación para hacer *streaming* y un dispositivo compatible(16).

1.2.2 Servicios de Streaming en plataformas de audiovisuales.

YouTube es un portal del Internet que permite a sus usuarios subir y visualizar videos. Fue creado en febrero de 2005 por Chad Hurley, Steve Chen y Jawed Karim. Esta plataforma cuenta con un reproductor online basado en *Flash*, el formato desarrollado por **Adobe Systems**. Una de sus principales innovaciones fue la facilidad para visualizar videos en *streaming*, es decir, sin necesidad de descargar el archivo a la computadora. Los usuarios, por lo tanto, pueden seleccionar qué video quieren ver y reproducirlo al instante, esto es posible a través de la API que usa que le permite hacer *streaming* en cualquier dispositivo que quiera consumir el contenido de esta plataforma (17).

Según (18) **Star Plus** es el hogar de series de televisión y películas de los estudios de contenido de The Walt Disney Company, incluidos Disney Television Studios, FX, 20th Century Studios, entre otros, así como el servicio de *streaming* de deportes en vivo de ESPN, que sus primeras transmisiones deportivas en vivo estarían disponibles en un servicio de *streaming* a través de la API que usa Strar Plus.

Netflix Es un servicio de transmisión o *streaming* que permite ver una gran variedad de series, documentales y películas en cualquier dispositivo con acceso a internet como celulares, computadoras, tabletas o 'Smart tv' mediante el pago de una tarifa fija mensual. Para empezar a usarlo, se debe abrir una cuenta en Netflix. Ofrece la posibilidad de hacer *streaming* a través de su API la cual da su servicios solamente si se está vinculado a una cuenta existente, en caso de no tenerla debe crearse una.(19).

Para que las plataformas de audiovisuales puedan ofrecer servicios de *streaming* deben hacer uso de una **API**, que es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el *software* de las aplicaciones, permitiendo la comunicación entre dos aplicaciones a través de un conjunto de reglas. Se establece cómo un módulo de un *software* que se comunica o interactúa con otro para cumplir una o muchas funciones(26).

A continuación (ver Tabla 1), se hace un resumen de las principales plataformas de audiovisuales existentes que consumen servicios de *streaming* utilizadas en la actualidad, para establecer las diferencias y similitudes con la solución propuesta en cuanto a:

- Consumo de servicios sin necesidad de una suscripción.
- Permiso de mostrar contenido sin autenticación.
- Acceso a su API.
- Complejidad de la arquitectura.

Tabla 1. plataformas de audiovisuales existentes que consumen servicios de streaming*.*
Fuente elaboración propia.

Plataformas de servicios en streaming.	**Consumo de servicios sin necesidad de una suscripción**.	**Permiso de mostrar contenido sin autenticación**	**Acceso a su API**	**Complejidad de la arquitectura**
YouTube	si	no	si	media
Netflix	no	no	si	Alta
Twitch	si	no	si	Alta

El análisis anterior sobre las plataformas de audiovisuales que consumen servicios de *streaming*, demuestra la necesidad de desarrollar una nueva solución, ya que estas no resuelven las carencias que presenta la creación de un módulo de *streaming* de Picta para el televisor inteligente cubano con sistema operativo GNU/Linux Nova. Se identificaron como

limitaciones:

- YouTube no muestra su contenido de *streaming* sin autenticarse.
- Twitch cuenta con una complejidad en su arquitectura alta, lo cual dificulta entender su funcionamiento en su totalidad, se debe tener en cuenta que dicha plataforma está orientada a utilizarse principalmente para transmisiones que son en directo.
- Netflix no permite consumir servicios de su plataforma sin una suscripción previa, no muestra su contenido de *streaming* sin autenticarse y presenta una complejidad en su arquitectura alta.

En la búsqueda realizada no se encontraron aplicaciones que den solución al problema de la investigación por lo que es necesario comenzar a desarrollar el módulo de *streaming* para Picta que forma parte de la solución propuesta.

Teniendo en cuenta la experiencia obtenida con el estudio anterior la propuesta de solución debe combinar los aspectos de consumir servicios de *streaming* sin necesidad de una suscripción, permitir mostrar los contenidos sin autenticación, tener acceso a su API y que la complejidad de la arquitectura sea media o preferiblemente baja. El módulo a desarrollar está dirigido a la plataforma de audiovisuales Picta, ya que a partir de esta se podrán consumir servicios de *streaming* en televisores inteligentes cubanos.

1.2.3 Kodi como centro multimedia para que los televisores inteligentes puedan hacer uso de plataformas de streaming.

Una de las mejores ideas para que el televisor sea un Smart TV es a través de dispositivos externos que se conectan vía HDMI tales como el **Chromecast** de Google, a través de un dispositivo que se denomina *dongle*, que es muy parecido a un pincho USB, pero con conexión HDMI y un pequeño ordenador dentro para hacer que sus funciones permitan consumir servicios de *streaming* de aplicaciones como YouTube. El *dongle* **webOS** es el sistema operativo más reciente de las Smart TV de LG que transforma el uso del televisor para hacerlo más intuitivo, rápido y divertido (22). Según(23) El sistema está basado en Linux y hace uso de inteligencia artificial, entre sus funcionalidades encontramos los servicios de *streaming*

desde un Smart TV, que tiene como ventaja poder disfrutar de contenidos audiovisuales con total comodidad en cuestión de segundos. Según (24) los **Apple TV** son los *dongle* que usa Apple permite conectar el televisor inteligente a contenidos de *streaming,* pero debe cumplir con una conexión a internet de al menos 5Mbps de velocidad.

A pesar de que los *dongles* les permiten a los televisores inteligentes consumir servicios de *streaming* de las plataformas de audiovisuales, tienen como desventaja que son específicos para cada televisor en dependencia de su fabricante. Por lo tanto, no se puede usar cualquier *dongle* en un Smart TV al azar. Como solución se puede crear un centro multimedia interconectado donde se hace uso de servidores locales en nuestro ordenador o televisor por medio de aplicaciones tales como Plex, Kodi, entre otras.

Los centros multimedia se utilizan para mostrar cualquier objeto o sistema que utiliza múltiples medios de expresión para presentar o comunicar información. Los medios pueden ser de texto, imágenes, animación, sonido, vídeo, entre otros. Las aplicaciones que se utilizan como Plex que se puede usar de forma gratuita, aunque su código no es libre*,* Stremio que también es gratis, pero no es de código libre y para servicios de *streaming* depende de un *dongle* y Kodi que si es de código libre y para ofrecer contenidos de *streaming* solo depende de la manipulación de su código.

Kodi es una aplicación con la que puedes convertir tu ordenador en un centro multimedia en el que se puede ver con una interfaz limpia y clara todo tipo de contenidos. Una ventaja de Kodi es que su código es libre, por lo que todos los usuarios de Internet pueden acceder a su código fuente y manipularlo para adaptarlo a cualquier sistema operativo o dispositivo en el que lo necesiten. Gracias a eso Kodi tiene versiones tanto para Windows como para GNU/Linux, macOS, iOS, Android, Raspberry Pi. Otra de sus principales características es que es una aplicación totalmente modular que puedes adaptar a cualquier entorno. No sólo porque tiene temas para cambiar su interfaz, sino porque también tiene un sistema de add-ons con los que añadirle diferentes tipos de funcionalidades según cuales sean tus características. Estos add-ons te permiten por ejemplo poder ver en Kodi diferentes canales de televisión a través de Internet, mostrar predicciones meteorológicas, o conectarte a servicios como Plex, SoundCloud o YouTube. También se puede acceder a algunos de los servicios de

almacenamiento en la nube (25).Otra ventaja es que se puede consumir servicios de *streaming* de las plataformas de audiovisuales sin necesidad de utilizar un *dongle* en el Smart TV.

1.3 Metodología de desarrollo de software

Según (27) la metodología de desarrollo de *software* es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de *software* informático. Es importante señalar que existen varias, de manera que es una decisión de cada equipo.

Para el desarrollo de la solución se utilizará la metodología de desarrollo de software AUP-UCI.

Metodología de desarrollo de software AUP-UCI

La metodología que será utilizara para el desarrollo de la propuesta de solución es Variación de AUP para la UCI, la cual fue elaborada en la Universidad de las Ciencias Informáticas. Esta metodología logró estandarizar el proceso de desarrollo de *software* en la universidad, dando cumplimiento a las buenas prácticas y logrando hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Variación de AUP para la UCI es la metodología de desarrollo de *software* usada en CESOL para el cual se desarrolla este trabajo de diploma. La misma cuenta con tres fases inicio, ejecución y cierre. Además define cuatro escenarios posibles, en los que la propuesta de solución puede enmarcarse (28) .

Inicio: durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

El desarrollo de la solución que se propone, se enmarca en el escenario número 4 de la metodología antes definida. Debido a que el proyecto no es extenso, se muestra un alto vínculo entre el cliente y el desarrollador; y el negocio a informatizar está bien definido.

1.4 Lenguajes y herramientas para el modelado de la solución

En el modelado de la propuesta de solución se utilizaron los siguientes lenguajes y herramientas.

1.4.1 Lenguaje de modelado

El lenguaje de modelado es cualquier lenguaje informático gráfico o textual que provee el diseño y construcción de estructuras y modelos siguiendo un conjunto sistemático de reglas y marcos. Como lenguaje de modelado se utiliza el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) en su versión 2.1, que consiste en un lenguaje diseñado para visualizar, especificar, construir y documentar *software* orientados a objetos (29).

1.4.2 Herramientas para el modelado

Las herramientas de desarrollo de *software* son diversos productos informáticos que dan soporte a una tarea concreta, facilitan y aseguran entregar un sistema con calidad. Se realizó un estudio de las herramientas CASE más utilizadas en el ciclo de vida de desarrollo del software, donde se destacan: Oracle Designer, Visual Paradigm y Power Designer. La herramienta escogida para el modelado de la aplicación es Visual Paradigm. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos.

Visual Paradigm v15.1 para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para ingenieros de software, analistas de sistemas y arquitectos de sistemas que están interesados en la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (30).

Visual Paradigm también ofrece (30):

- Navegación intuitiva entre la escritura del código y su visualización.
- Potente generador de informes en formato PDF/HTML.
- Documentación automática.

- Ambiente visualmente superior de modelado.

1.5 Herramientas y tecnologías de la implementación

En el desarrollo de la propuesta de solución se utilizaron las siguientes herramientas y tecnologías.

1.5.1 Lenguaje de programación

Un lenguaje de programación es una interfaz de usuario con la cual los seres humanos pueden elaborar procesos que serán ejecutados por una máquina de cómputo. Los mismos tienen reglas y parámetros establecidos para poder realizar diferentes acciones, existen diferentes lenguajes y dependen del tipo de programación (31). Para el desarrollo del módulo de conexiones inalámbricas para televisores inteligentes con sistema operativo Nova se seleccionó Python como lenguaje de programación.

Python 3.5 es un lenguaje potente, flexible y con una sintaxis clara y concisa que no requiere dedicar tiempo a su compilación debido a que es interpretado. Es de código abierto y como tal cualquiera puede contribuir a su desarrollo y divulgación. Además, no es necesario pagar ninguna licencia para distribuir el *software* desarrollado en este lenguaje. Hasta su intérprete se distribuye de forma gratuita. Puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar los más populares, Windows y Linux; pero además puede funcionar en teléfonos inteligentes (32).

Las principales características que tiene este lenguaje son (33) :

- Lenguaje de programación multiparadigma (programación orientada a objetos, programación estructurada y programación funcional).
- Utiliza el tipo de dato dinámico y *reference counting* (conteo de referencia) para el manejo de memoria.
- Permite la resolución dinámica de nombres, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado ligadura dinámica de métodos).

- Puede utilizarse como un lenguaje de extensión para módulos y aplicaciones que necesitan de una interfaz programable.

1.5.2 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE) es un sistema de *software* para el diseño de aplicaciones que combina herramientas comunes para desarrolladores en una sola interfaz de usuario gráfica (GUI). Los IDE permiten que los desarrolladores comiencen a programar aplicaciones nuevas con rapidez, ya que no necesitan establecer ni integrar manualmente varias herramientas como parte del proceso de configuración. Otras funciones comunes del IDE se encargan de ayudar a los desarrolladores a organizar su flujo de trabajo y solucionar problemas. Los IDE analizan el código mientras se escribe, así que las fallas causadas por errores humanos se identifican en tiempo real. Gracias a que hay una sola GUI que representa todas las herramientas, los desarrolladores pueden ejecutar tareas sin tener que pasar de una aplicación a otra (34).

Visual estudio code es un editor de código fuente desarrollado por Microsoft. Es *software* libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. Tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación (35).

Conclusiones del capítulo

El avance tecnológico de los medios de comunicación ha propiciado que la televisión como medio de difusión adopte nuevas formas para transmitir su información. La definición de los principales conceptos asociados a este medio y su entorno permite una mayor comprensión para el desarrollo del módulo de *streaming* de Picta para los televisores inteligentes cubanos con sistema operativo GNU/Linux Nova.

Los módulos de *streaming* que utilizan las plataformas de audiovisuales existentes para la difusión de información aportan ideas a la solución del problema en cuestión, y sirven de base para desarrollar una idea general de cómo se debe representar la solución final. El marco de trabajo para el desarrollo del módulo está definido por el uso de herramientas y tecnologías reconocidas y con la metodología AUP-UCI.

CAPÍTULO 2: Análisis y diseño de la propuesta de solución

En el presente capítulo, se desarrolla el análisis y diseño de la propuesta de solución tomando como punto de partida las características los elementos planteados por la metodología Variación de AUP-UCI en el escenario 4. Se definen los requisitos del sistema y la arquitectura del mismo

2.1 Propuesta de Solución

Con el fin de concretar la propuesta de solución y darle respuesta al objetivo de la presente investigación se define como línea principal de esta investigación: el desarrollo un módulo para televisores inteligentes con sistema operativo GNU/Linux Nova, que permita consumir servicios de *streaming* a través de la API de Picta. La solución debe mostrarles a los usuarios cuáles son los contenidos que puede disfrutar de Picta, permitiéndole hacerlo en *streaming* desde centro multimedia Kodi, siendo este multiplataforma y al ser de código libre, es posible adaptarlo al resultado que se desea obtener para dar cumplimiento al objetivo definido en la investigación.

2.2 Modelo de dominio

El modelado de dominio es uno de los modelos más utilizados en la ingeniería de software. Un esfuerzo de modelado de dominio relativamente pequeño es una gran herramienta para controlar la complejidad del sistema en desarrollo. Puede ayudar a resolver innumerables ambigüedades tanto en los requisitos como en la intención del diseño. El modelado de dominio refleja nuestra comprensión de las entidades del mundo real y sus relaciones y responsabilidades que cubren el dominio del problema (36). En la figura 1 se muestra el modelo conceptual de la propuesta de solución.

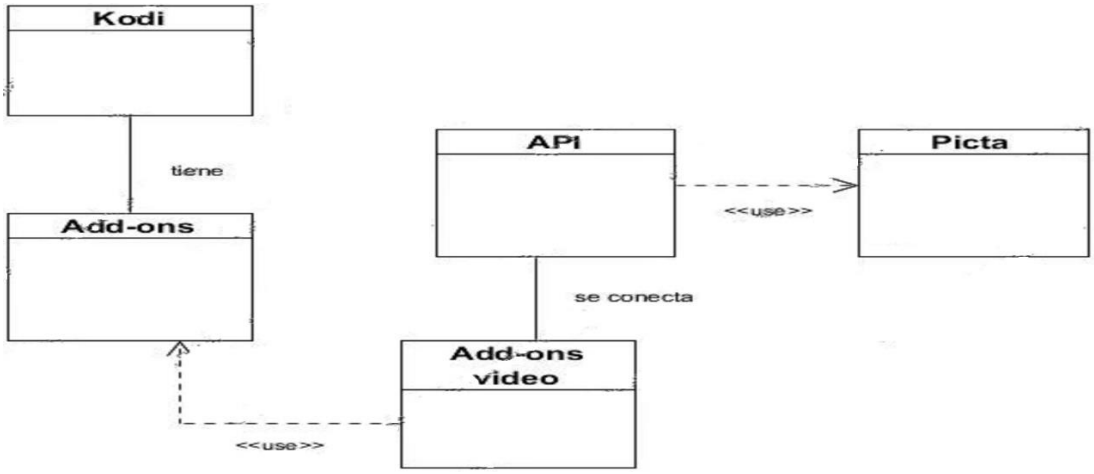


figura 1:Modelo de conceptual. Fuente: Elaboración propia.

Este modelo evidencia, como Kodi tiene add-ons(módulos) que cada uno es específicamente para una funcionalidad. En este caso el módulo a utilizar es add-ons video que permite la funcionalidad de poder reproducir videos en Kodi. Add-ons video se comunica con a la API que utiliza Picta ya que no es posible conectar Kodi y Picta directamente, siendo la segunda quien tiene el contenido que se desea obtener y mostrar.

2.3 Requisitos

La Ingeniería de Requisitos es una de las partes cruciales en el éxito de todo proyecto *software*. La aparición de errores o carencias durante la recogida de requisitos implica un descenso en la productividad del proceso de desarrollo y, por lo tanto, un incremento del coste del mismo. Incluir una adecuada ingeniería de requisitos en el ciclo de vida del *software* minimizará la posibilidad de que esto ocurra. La Ingeniería de Requisitos se convierte en pieza clave para poder medir la calidad de un sistema informático al poder iniciar la definición de la batería de pruebas que el sistema debe pasar, garantizando que éstas satisfacen los requisitos establecidos y por lo tanto el sistema es válido y funcionalmente es correcto(37).

2.3.1 Técnicas de identificación de requisitos

Las técnicas agrupadas como generales son las que permiten investigar aspectos generales para posteriormente ser especificados con un mayor detalle con el apoyo de técnicas más específicas. Estas técnicas son más abiertas y requieren ser adecuadamente orientadas para cubrir la información que se requiere capturar, es importante que para sacar el mayor provecho

de estas técnicas se debe tener un dialogo lo más abierto posible entre las organizaciones de desarrollo de software y las empresas cliente.(38)

Entrevista

Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Debe quedar claro que no basta con hacer preguntas para obtener toda la información necesaria. Es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista.(39). La entrevista fue realizada al personal que desarrollo Picta (Ver Anexo 1).

Observación

Por medio de esta técnica el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método permite observar la forma en que se llevan a cabo los procesos y, por otro, verificar que realmente se sigan todos los pasos especificados.(39)

2.3.2 Validación de los requisitos.

Desarrollo de prototipos

Un modelo prototipo o modelo de desarrollo evolutivo es utilizado principalmente en el desarrollo de *software* para ofrecer al usuario una visión previa de cómo será el programa o sistema. Se le dice de desarrollo evolutivo al modelo de prototipo porque evoluciona hasta convertirse en el producto final.(40)

Con los requisitos obtenidos mediante la aplicación de la Entrevista y la Observación se realizaron los prototipos de las interfaces lo que dio la posibilidad de verificar la consistencia y completitud de los mismos. La aplicación de estas técnicas propició la identificación de los requisitos que se describen a continuación.

2.3.3 Especificación de requisitos de software.

El objetivo principal de la Especificación de Requisitos del Sistema (ERS) es servir como medio de comunicación entre clientes, usuarios, ingenieros de requisitos y desarrolladores. En la ERS deben recogerse tanto las necesidades de clientes y usuarios (necesidades del negocio, también conocidas como requisitos de usuario, requisitos de cliente, necesidades de usuario, etc.) como los requisitos que debe cumplir el sistema *software* a desarrollar para satisfacer dichas necesidades (requisitos del producto, también conocidos como requisitos de sistema o requisitos *software*).(41) A continuación se relacionan y describen los requisitos funcionales y no funcionales definidos para la implementación de la propuesta de solución.

Requisitos funcionales

Un requerimiento funcional es una descripción del servicio que debe ofrecer el *software*. Describe un sistema de *software* o su componente. Solo se implementan entradas en el sistema de *software*, su transporte y salidas. Puede ser un cálculo, manipulación de datos, proceso comercial, interacción del usuario o cualquier otra funcionalidad específica que defina qué función probablemente realizará su sistema.(42) En la tabla 2 se listan los requisitos funcionales de la propuesta de solución, donde su prioridad fue definida por el usuario.

Tabla 2:Descripcion de requisitos funcionales Fuente: Elaboración propia

Numero	**Requisito**	**Descripción**	**Prioridad**
RF1	Conectar a los contenidos de *streaming* de la API de Picta.	Conectar desde Kodi con la plataforma de Picta mediante internet, utilizando la API de la misma.	Alta
RF2	Mostrar categorías.	Mostrar los resultados de la petición realizada a las categorías de la plataforma de Picta.	Alta

RF3	Obtener categorías.	Obtener los contenidos de la petición realizada a la plataforma de Picta.	Alta
RF4	Mostrar videos de los canales.	Mostrar los videos de los canales según su tipo.	Alta
RF5	Obtener videos de los canales	Obtener los videos de los canales según su tipo.	Baja
RF6	Mostrar canales	Mostar los canales	Media
RF7	Obtener canales	Obtener los canales	Baja
RF8	Obtener metadatos	Obtener metadatos de videos, series, canales	Media
RF9	Mostrar metadatos	Mostrar metadatos de videos, series, canales.	Media
RF10	Reproducir videos	Reproducir el video que se solicita	Baja

Requisitos no funcionales

Se refieren a las cualidades, restricciones y características del *software*. A diferencia de los funcionales, no determinan una funcionalidad del sistema a desarrollar.(43)

En la tabla 3 se muestran los requisitos no funcionales con sus clasificaciones, basándose en las normas de ISO-25010.

Adecuación funcional.

RNF1 -El módulo no funcionara correctamente si el usuario no cuenta con ciertas características de *hardware*. La interfaz se mostrará en un dispositivo con entrada VGA, HDMI o con acceso a internet.

RNF2 -El sistema funciona correctamente y los componentes se encuentran acorde a las pautas de diseño definidas.

Portabilidad.

RNF3 -El servidor estará corriendo sobre el sistema operativo GNU/Linux-NOVA.

RNF4 -Adapta sus componentes a la resolución de pantalla seleccionada por el usuario.

Usabilidad

RNF5- El idioma de todas las interfaces de la aplicación será español.

Eficiencia de desempeño

RNF6- El sistema realizará las operaciones indicadas en cada momento.

2.4Historia de Usuario

Las historias de usuario describen las características y necesidades de un *software* desde la perspectiva de un usuario, ayudando a alinear expectativas y evitar errores críticos en el futuro. Una historia de usuario puede considerarse como una preparación para establecer los requisitos del *software*. La descripción de historias de usuario permite planificar, priorizar y estructurar el trabajo, mediante un enfoque visual donde podrás ver, priorizar y evaluar los requisitos(44).

Para la descripción de los requisitos funcionales de la propuesta de solución se define una historia de usuario para cada uno, para un total de 10 historias de usuarios. A continuación, se muestran las HU “Conectar a los contenidos de *streaming* de la API de Picta.” “Mostrar categorías” “Obtener categorías” y “Mostrar videos de los canales” por ser las que describen las funcionalidades que tienen mayor prioridad para el cliente. El resto de la historias de usuarios (Ver anexo 2).

A continuación, se definen los elementos que conforman las HU:

- **Número:** Representa el identificador de la historia de usuario, con el fin de emplearlo como referencia de esta en la fase de validación.
- **Nombre del requisito:** Define el nombre del requisito que describe.

- **Programador:** Especifica el nombre y apellidos del programador que se encargará de desarrollar el requisito descrito.
- **Iteración asignada:** Enumera según la metodología la iteración en la que se encuentra la implementación del requisito.
- **Riesgo en desarrollo:** Define los riesgos que pudieran imposibilitar la implementación del requisito.
- **Tiempo estimado:** Establece el tiempo en días que se estimó para el desarrollo del requisito.
- **Tiempo real:** Especifica el tiempo en horas que tomó al programador, la implementación del requisito.
- **Descripción:** Realiza una descripción detallada de las condiciones que debe cumplir el requisito según exige el cliente en diferentes escenarios posibles.
- **Observaciones:** Se especifican los elementos bajo los que el requisito funcional debe ejecutarse.

Tabla 3.Historia de usuario: Conectar a los contenidos de streaming de la API de Picta. Fuente: Elaboración propia.

Historia de Usuario	
Número: 1	**Nombre del Requisito:** Conectar a los contenidos de streaming de la API de Picta
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:** 72 horas
Riesgo en Desarrollo: Alta	**Tiempo Real:** 96 horas

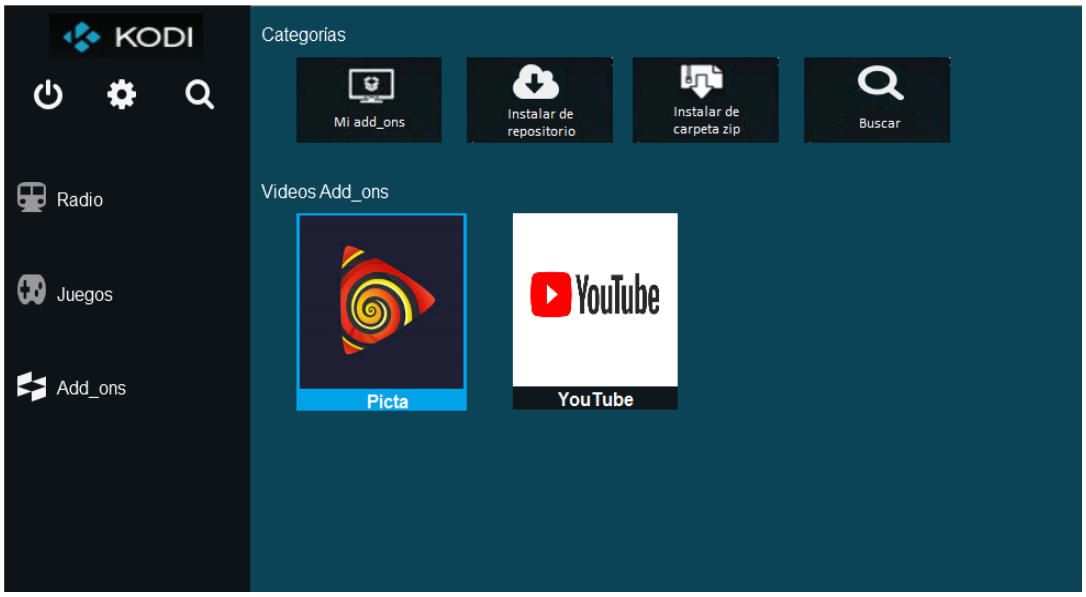
Descripción: El módulo debe permitirle a el usuario conectarse desde Kodi con la plataforma de Picta mediante internet, utilizando la API de la misma.
Observaciones: El usuario debe verificar que tiene conexión a internet
Prototipo de interfaces: *figura 2.Prototipo de add-ons de video de Picta en Kodi* *Fuente: Elaboración propia* 

Tabla 4.Historia de Usuario Mostrar categorías Fuente: Elaboración propia

Historia de Usuario	
Número: 2	**Nombre del Requisito:** Mostrar categorías
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 1
Prioridad: baja	**Tiempo Estimado:** 50 horas

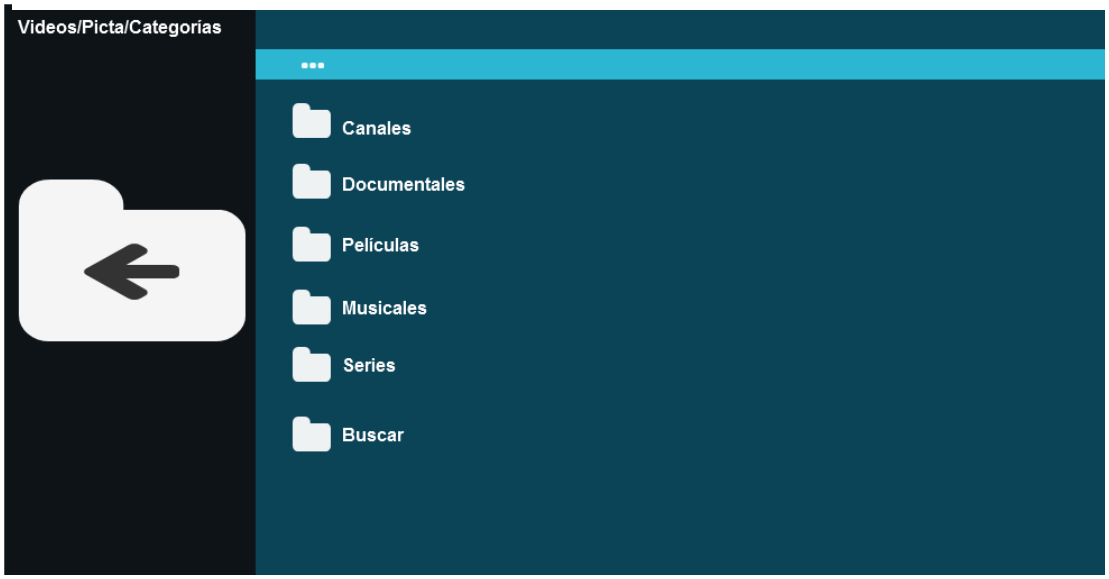
Riesgo en Desarrollo: baja	**Tiempo Real:** 70 horas
Descripción: El módulo debe mostrarle al usuario los resultados de la petición realizada a las categorías de la plataforma de Picta.	
Observaciones: NA	
Prototipo de interfaces: *figura 3 Prototipo de mostrar categorías Fuente: Elaboración propia.* 	

Tabla 5.Historia de usuario Obtener categorías Fuente:Elaboración propia.

Historia de Usuario	
Número: 3	**Nombre del Requisito:** Obtener categorías

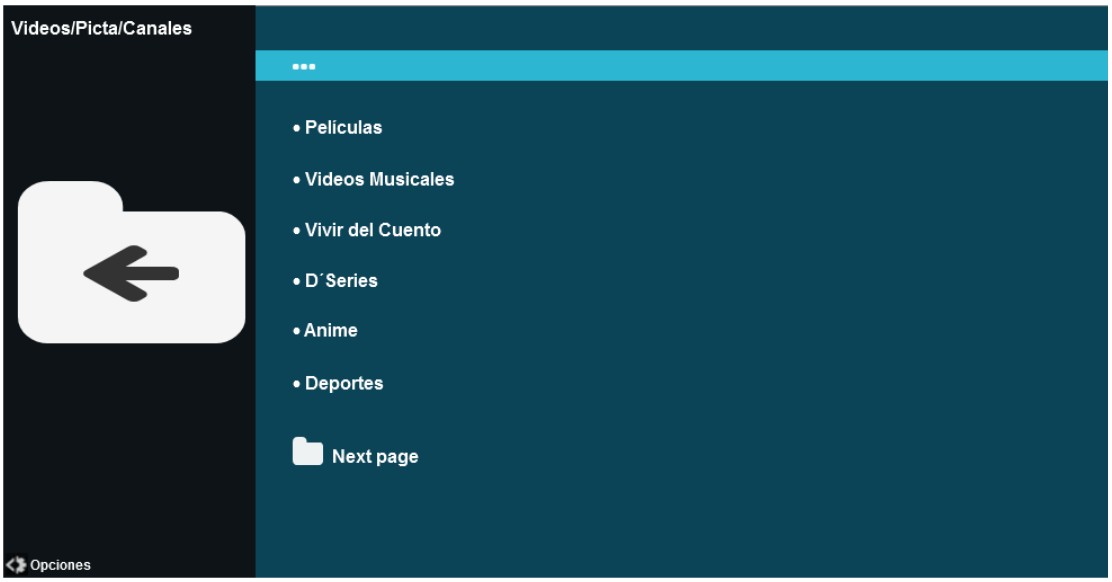
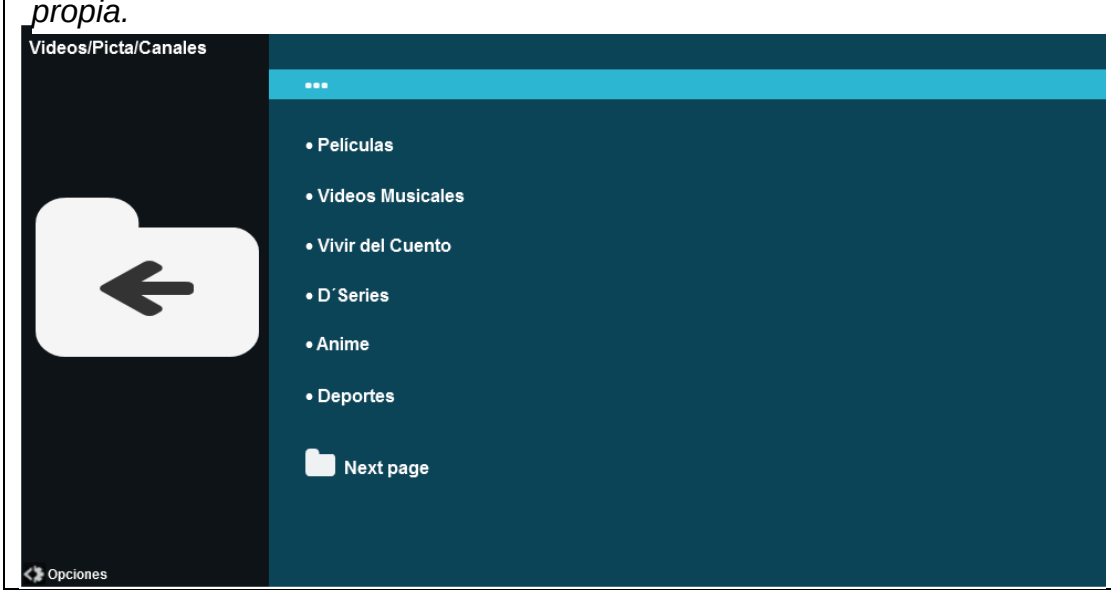
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:** 45 horas
Riesgo en Desarrollo: baja	**Tiempo Real:** 60 horas
Descripción: El usuario debe tener respuesta de la solicitud realizada a la Api de Picta y poder visualizarlos en Kodi	
Observaciones: NA	
Prototipo de interfaces: *figura 4: Prototipo de obtener categorías. Fuente: Elaboración propia.* 	

Tabla 6.Historia de usuario Mostrar videos de los canales Fuente: Elaboración propia

Historia de Usuario	
Número: 4	**Nombre del Requisito:** Mostrar videos de los canales
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:** 70 horas
Riesgo en Desarrollo: Alta	**Tiempo Real:** 79 horas
Descripción: El módulo debe permitirle al usuario visualizar desde Kodi con la plataforma de Picta utilizando la API de la misma los videos de los canales solicitados.	
Observaciones:	

Prototipo de interfaces:

figura 5 Prototipo de mostrar los videos de canales Fuente: Elaboración propia.



2.5 Modelo de Diseño

El **diseño de *software*** es el proceso mediante el cual un agente crea una especificación de un artefacto de *software* destinado a lograr objetivos , utilizando un conjunto de componentes primitivos y sujeto a restricciones .El diseño de *software* puede referirse a toda la actividad involucrada en la conceptualización, elaboración, implementación, puesta en marcha y, en última instancia, modificación de sistemas complejos o la actividad que sigue a la especificación de requisitos y antes de la programación. El diseño de *software* generalmente implica la resolución de problemas y la planificación de una solución de *software*. Esto incluye un diseño de algoritmos y componentes de bajo nivel y un diseño de arquitectura de alto nivel (45).

2.5.1 Descripción del estilo arquitectónico Modelo Vista Controlador (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. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo(46).

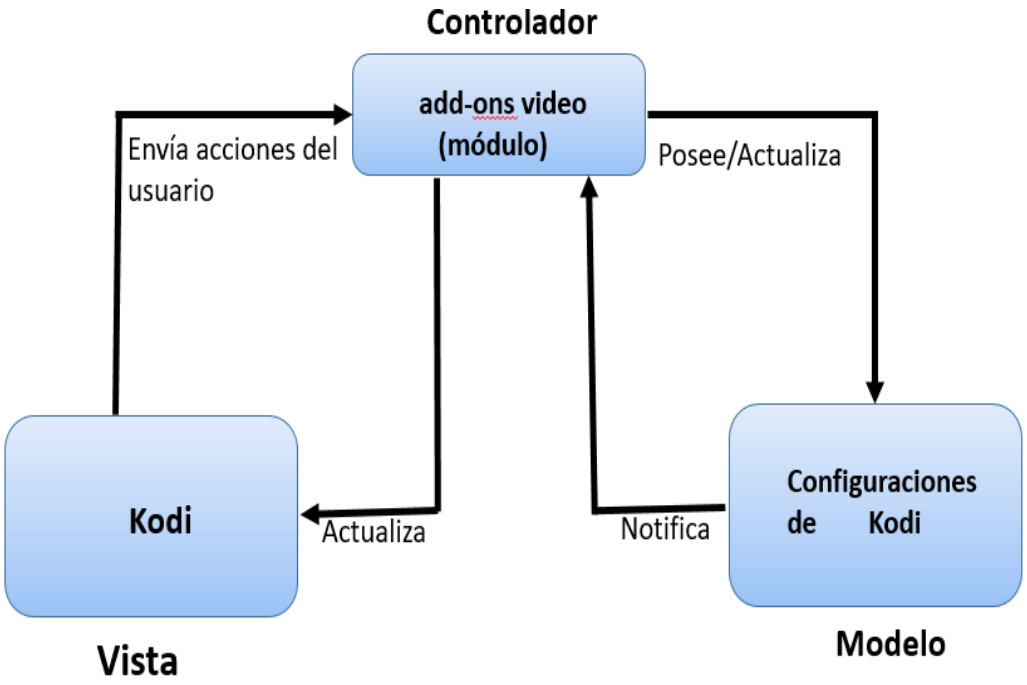


figura 6: Modelo Vista controlador. Fuente: Elaboración propia.

- El Modelo es quien contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, es la que compone la información que se envía al cliente.
- El Controlador actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

2.5.2 Patrones de Diseño

Los patrones de diseño, son una solución general, reutilizable y aplicable a diferentes problemas de diseño de *software*. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error(47).

GRASP (Patrones de *software* de asignación de responsabilidad general) es un patrón de diseño en el desarrollo de *software* orientado a objetos utilizado para asignar responsabilidades para diferentes módulos de código. Como herramienta para desarrolladores de software, GRASP proporciona un medio para resolver problemas organizacionales y ofrece una forma común de hablar sobre conceptos abstractos. El patrón de diseño establece responsabilidades para objetos y clases en orientado a objetos. GRASP con programación orientada a objetos clasifica los problemas y sus soluciones juntos como patrones. Con estos problemas y soluciones bien definidos, se pueden aplicar en otros casos similares(48).

Además, asigna tipos de roles a clases y objetos para hacer una clara delimitación de responsabilidades. Estos roles son:

Experto en información

Este patrón implica que cada objeto es responsable de almacenar su propia información, no la de los demás conceptos(49). Este patrón se evidencia en el caso que desee obtener información de una serie, mediante la función **def get_series** se obtiene toda la información que se desea referente a series, como también ocurre con **def get_episodios** entre otros. A continuación, se muestra un fragmento de código donde se evidencia como una función tiene la responsabilidad de almacenar toda su información.

```
def get_series(next_page: int = COLLECTION["next_href"]) -> List["Serie"]:
    Get list of Series
      :return: the list of Series
    :rtype: list
    COLLECTION[SERIES] = []
    url_series=f"{API_BASE_URL}serie/?page={next_page}&page_size=10&ordering=-id"
    r = requests.get(url_series)
    result = r.json()
  for v in result["results"]:
```

```
        generos = ", ".join(g["nombre"] for g in v["genero"])
        COLLECTION[SERIES].append(
            {
                "name": v["nombre"],
                "id": v["pelser_id"],
                "thumb": v["imagen_secundaria"] + "_380x250",
                "genre": generos,
                "cant_temp": v["cantidad_temporadas"],
            }
        )
    COLLECTION["next_href"] = int(result.get("next") or 0)
    return COLLECTION[SERIES]
```

Creador

Este patrón implica que un objeto debe responsabilizarse de crear otros:

- Si contiene o agrega varios objetos del tipo de los creados.
- Si se encarga de registrar objetos del tipo de los creados.
- Si utiliza mucho los objetos creados.
- O si contiene los datos para crear los del tipo creados(49).

Este patrón se evidencia en la clase **requests** es la única que conoce donde están los *endpoints* de cada tipo de objeto, ya sea película, series, o cualquier otro.

```
import requests
result = {}
    COLLECTION[category] = []
    if category == DOCUMENTALES:
```

```
        url_docum =
f"{API_BASE_URL}publicacion/?page={next_page}&page_size=10&tipologia_nombre_raw=Documental&ordering=-fecha_creacion"
        r = requests.get(url_docum)
        result = r.json()
    elif category == PELICULAS:
        url_pelic =
f"{API_BASE_URL}publicacion/?page={next_page}&page_size=10&tipologia_nombre_raw=Pel%C3%ADcula&ordering=-fecha_creacion"
        r = requests.get(url_pelic)
        result = r.json()
    elif category == MUSICALES:
        url_musicales =
f"{API_BASE_URL}publicacion/?page={next_page}&page_size=10&tipologia_nombre_raw=Video%20Musical&ordering=-fecha_creacion"
        r = requests.get(url_musicales)
        result = r.json()
```

Controlador

El controlador se define como el primer objeto más allá de la capa de la interfaz de usuario que recibe y coordina ("controla") una operación del sistema. El controlador debe delegar el trabajo que debe realizarse a otros objetos; coordina o controla la actividad. No debería hacer mucho trabajo por sí mismo. El controlador GRASP se puede considerar como parte de la capa de aplicación en un sistema orientado a objetos con capas comunes. en una arquitectura lógica de sistema de información(50).Como se evidencia en el fichero **pluging.py**.

Patrones GoF se tratan de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a través de prueba y error. Existen 23 patrones de diseño comunes.

Cada uno de ellos define la solución para resolver un determinado problema, facilitando además la reutilización del código fuente. Los tipos de patrones de diseño son 3:

- Patrones creacionales
- Patrones estructurales
- Patrones de comportamiento.

Los patrones GoF a utilizar en la propuesta de solución son:

- **Decorator** que permite añadir funcionalidades a objetos colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades.

  @hostname=api.picta.cu

  @port=443

- **Facade** que proporciona una interfaz simplificada a una biblioteca, un framework o cualquier otro grupo complejo de clases.

  r=requests.get(url_temp)

  result=r.json()

- **Singleton** que permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.

  EPISODIOS: list [¨video¨]=[ ]

  url_temp= f¨{API_Base_Url}

  r=requests.get(url_temp)

  result=r.json()

Conclusiones del capítulo

En el análisis y diseño de la solución propuesta quedaron definidos los requisitos de *software*, patrones y arquitectura a utilizar. El módulo para la reproducción de contenidos de *streaming* de la plataforma Picta para el centro multimedia Kodi, utilizará como lenguaje de programación Python para proporcionar una estructura más solida y permitir el empleo de buenas prácticas de programación orientada a objetos.

CAPÍTULO 3: DESARROLLO Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

La concepción de la propuesta del sistema presentada en el capítulo anterior, ayuda al programador a entender mejor las funcionalidades que busca el cliente y, por tanto, ser capaz de llevarlas a código entendible por la computadora, o lo que es lo mismo implementar el sistema. Durante toda la fase de implementación y después de esta se desarrollan un conjunto de pruebas con el objetivo de asegurar la calidad de la aplicación y que la misma cumpla con todas las peticiones del cliente. Como principales elementos en este capítulo se definen el diagrama de componentes y el diagrama de despliegue. Además, de especificar los casos de pruebas aplicados a la solución desarrollada para validar su correcto funcionamiento.

3.1 Despliegue de la solución.

1. Para desplegar la solución propuesta se necesita una computadora de propósito general con:
    - 4gb de RAM.
    - 10gb de disco duro.
    - Sistema operativo Nova GNU/Linux en su versión 8.0 conectada a un dispositivo de visualización, monitor, o televisor conectado por un puerto HDMI o VGA.
2. Para la instalación de Kodi:
    - acceder al sito https://kodi.tv/download/ descargar para Sistema operativo Linux, la versión Kodi v19.4 (Matrix).
    - utilizar los siguientes comandos y ejecutar en ese mismo orden que se muestran:
        - sudo apt install software -properties -common
        - sudo apt -apt -repository -y ppa:team -xbmc/ppa
        - sudo apt install kodi
        - sudo apt install kodi -inputstream -adaptive kodi -inputstream -ffmpegdirect

3. Para la instalación del módulo desarrollado se deben realizar dos acciones:
    - seleccionar en Kodi la opción instalar desde carpeta zip
    - seleccionar la carpeta plugin.video.picta-kodi_19.zip
4. Seleccionar el contenido que se desea ver de Picta en Kodi.
5. Reproducción del contenido seleccionado en forma de *streaming* sin necesidad de descargarlo.

3.2 Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos(51).

3.2.1 Diagrama de componente

El diagrama de componentes ilustra la relación que existe entre componentes de *software*, así como la ubicación de cada uno de ellos dentro del módulo, como se implementan las clases en término de componentes. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularizarían el entorno de implementación y en el lenguaje de programación utilizado. Además, muestra las dependencias entre componentes (52). A continuación, se muestra el diagrama de componentes del módulo para consumir servicios de Picta.

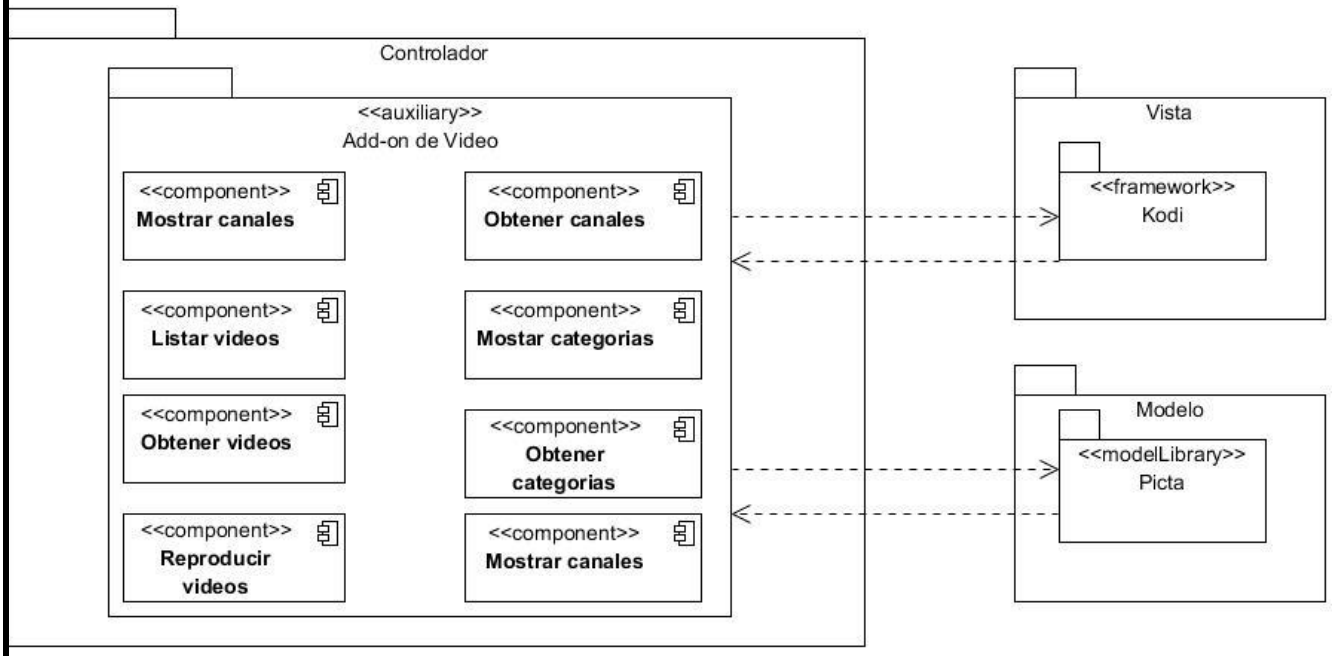


figura 7. Diagrama de componentes Fuente: Elaboración propia

Donde, el paquete Vista contiene la interfaz de usuario la que interactúa con el mismo, el paquete Controlador contiene las funciones principales que gestionan la lógica y los datos de los procesos, el paquete Modelo contiene el componente el cual se considera una base de datos volátil no persistente en ficheros JSON.

3.3 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del *software* y para obtener un buen rendimiento(53).

Reglas de indentación en PEP 8

Python define la lógica de su código utilizando bloques indentados, por lo tanto, se recomienda indentar usando 4 espacios, dado que los espacios ocupan el mismo tamaño en casi todos los tipos de fuente y 4 es un número aceptable para la separación visual de los bloques. Como se observa en la figura 4.

```
    if category == DOCUMENTALES:
        url_docum = f"{API_BASE_URL}publicacion/?
        r = requests.get(url_docum)
        result = r.json()
```

figura 8. Estándares de Python Fuente: Visual Studio Code

Importaciones de código

Para las importaciones de paquetes existen reglas.

- El uso de import debe de ser en líneas separadas.
- El orden de importación es: librería estándar, librerías de terceros, librerías propias o locales.
- Las importaciones absolutas son los recomendados.
- Las importaciones relativas solo pueden ser explícitas y preferiblemente simples y cortas.
- Importaciones usando * (wildcard) deben de ser evitadas.

```
 8
 9  import sys
10  from typing import TYPE_CHECKING, Any, Dict, List, Union
11  from urllib.parse import parse_qsl, unquote_plus, urlencode
12
13  import requests
14  import xbmc
15  import xbm  Loading...
16  import xbmcgui
17  import xbmcplugin
18  import xbmcvfs
```

figura 9.Importaciones en Python Fuente: Visual Studio Code

La PEP 8 contempla muchas reglas sobre cómo tratar los comentarios que se pueden ver a continuación:

- Los comentarios pueden ser contradictorios, y requieren que se actualicen con el código, por tanto, si se pueden evitar haciendo código más claro, mejor.
- Deben de componerse de frases completas.
- Deben de comenzar en mayúscula a no ser que comiencen con el nombre de un identificador, el cual se respetará siempre su tamaño.
- Los bloques de comentarios se componen de varios párrafos terminados en puntos, y terminar en dos espacios en blancos salvo la última frase.
- Los comentarios deben de ser en Ingles a no ser «que se esté 120%» seguro de que se leerán siempre en otro idioma y serán claros.
- Los bloques se aplican al código justo después de donde se encuentran y con la misma indentación.
- Los comentarios de línea comienzan con al menos 2 espacios y un # y un espacio después.

```
xbmcplugin.setPluginCategory(_HANDLE, name)
xbmcplugin.setContent(_HANDLE, "season")
# Note:
# La cantidad_temporadas para una serie no está actualizada mediante la API
# Ejemplo:
#   Serie = {"pelser_id": 107, "nombre": "Rick and Morty", "cantidad_temporadas": 4}
# Si consultas {{temporadaBySerieEndpoint}} "count": 5,
cant_temp = int(temporada) + 1
```

figura 10. Como comentar en Python Fuente: Visual Studio Code

Convenciones de nombres

Las siguientes reglas aplican para los nombres de cada tipo de objeto.

- Caracteres a evitar: o, O, l y L (letra O y letra L) dado que se pueden confundir con un 0 o un 1 dependiendo del tipo de fuente usada.
- Caracteres ASCII: se utilizan en la librería estándar siempre.
- Nombre de módulos y paquetes: se escriben en minúsculas y se desaconseja el uso de barras bajas, manteniendo los nombres lo más cortos posibles.
- Nombrado de clases: se usan el Formato Capital, donde la primera letra de cada palabra es mayúscula.
- Nombres de tipos de variables: en Formato Capital.
- Excepciones: igual que el de clases, pero con el sufijo «Error».
- Nombres de variables y funciones: se usan palabras en minúsculas separadas por barras bajas.
- Métodos en clases: el primer parámetro para métodos de instancia es self y para métodos de clase cls.
- Constantes: en mayúsculas.

figura 11. Reglas de Python Fuente: Visual Studio Code

3.5 Validación de la propuesta de solución

Tabla 7: Estrategia de pruebas. Fuente: Elaboración propia.

Tipo de prueba	Técnica de prueba	Validación
Unitaria	Caso de prueba (Caja blanca)	Valida los métodos y funciones individuales de las clases, componentes o módulos que usa el *software*.
Funcional	Casos de prueba (Caja Negra)	Valida las funcionalidades diseñadas para el sistema.
Integración	Prueba de integración de sistema	Valida la relación entre interfaces de sistemas externos.

3.6 Pruebas unitarias

Las pruebas unitarias de *software*, conocidas también como *unit testing* o test unitarios, pueden definirse como un mecanismo de comprobación del funcionamiento de las unidades de menor tamaño de un programa o aplicación en específico. Estas pruebas forman parte de la estrategia de metodología ágil del trabajo del desarrollo, donde se busca ofrecer piezas pequeñas de *software* en funcionamiento en un corto periodo de tiempo, con el objetivo de aumentar la satisfacción del cliente(55).

A continuación, se muestran una de las pruebas unitarias realizadas, las demás se encuentran en Anexos 3, donde para su realización se utilizó como herramienta la librería **pytest** de Python y comprobar su correcto funcionamiento.

```python
import requests
API_BASE_URL = "https://api.picta.cu/v2/"
def test_get_canales():
    url_canales = f"{API_BASE_URL}canal/?page={next_page}&page_size=10&ordering=-cantidad_suscripciones"
    r = requests.get(url_canales)

    assert r.status_code == 200

```

figura 12: Prueba unitaria al método get_canales

```
C:\Users\Hp _PC\Desktop\plugin.video.picta-kodi_19\tests>pytest
================================== test session starts ==================================
platform win32 -- Python 3.7.7rc1, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\Hp _PC\Desktop\plugin.video.picta-kodi_19\tests
collected 1 item

test_plugin.py .                                                                  [100%]

=================================== 1 passed in 0.59s ===================================
```

figura 13:Resultado de la prueba unitaria al método get_canales

Luego de realizadas las pruebas unitarias se obtuvo como resultado que el flujo de trabajo de las funcionalidades del módulo para consumir servicios de streaming de la plataforma Picta es correcto; pues se comprobó que cada sentencia del código fuente se ejecuta al menos una vez.

3.7 Pruebas Funcionales

Las pruebas funcionales se basan en la comprobación de los resultados de la ejecución de las funcionalidades que se han diseñado previamente. Para llevarlas a cabo, hay que diseñar un plan de pruebas que se pueda ejecutar y revisar a lo largo de las validaciones que se realicen en la funcionalidad entregada. Se divide en una serie de etapas nos aportarán la correcta realización de las mismas(54).

- ✓ Analizar requisitos y documentación.

En esta etapa, se revisa toda la documentación de las funcionalidades a probar. Una vez que se ha revisado todo, se realiza un esqueleto de plan de pruebas y cuando los requisitos están finalizados, se completa.

- ✓ Diseñó del plan de prueba.

Cuando concluye la revisión de la documentación, se diseña y especifica las pruebas que garantizarán las funcionalidades. Las pruebas tienen que ser claras y concisas y sobre todo determinar una cobertura total.

- ✓ Ejecución del plan de pruebas.

Se ejecuta el plan de pruebas que se ha diseñado anteriormente. Para ello, hay que seguir cada paso de los casos de prueba y reportar el resultado del mismo.

- ✓ Gestión de defectos

Una vez que se ejecuta el plan de pruebas, aparecerán unos defectos que serán solucionados por las personas que han realizado la funcionalidad entregada. A continuación, se presentan los casos de prueba para las Historias de Usuarios Mostar categorías y Mostrar videos de los canales utilizando el método de caja negra bajo la técnica de partición de equivalencia.

Tabla 8: Caso de Prueba Mostrar categorías Fuente: Elaboración propia

Escenario	Descripción	Respuesta del sistema	Flujo central
EC1 Mostrar categorías con conexión a internet.	Mostrar los resultados de la petición realizada a las categorías de la plataforma de Picta.	Se mostrarán las categorías existentes en Picta.	Seleccionar la carpeta de nombre categoría.

EC1.2 Mostrar categorías sin conexión a internet.	Mostrar los resultados de la petición realizada a las categorías de la plataforma de Picta, con un mensaje de error en la petición al registro de la API	No se mostrarán las categorías existentes en Picta.	No se podrá seleccionar la carpeta de nombre categoría.

Tabla 9: Caso de Prueba Mostrar videos de los canales Fuente: Elaboración propia

Escenario	Descripción	Respuesta del sistema	Flujo central
EC1 Mostrar videos de los canales usando un add-ons de video	Mostrar los videos de los canales según su tipo.	Se mostrarán los videos de los canales disponibles en Picta.	Seleccionar un canal disponible.
EC1.2 Mostrar videos de los canales sin usar un add-ons de video.	Mostrar los videos de los canales según su tipo, el cual se muestra según el add-ons que se esté usando en ese momento.	No se mostrarán los videos de los canales disponibles en Picta debido a que el add-ons que se usó no es de video.	No se encuentran disponibles los canales.

3.7.1Resultados de las pruebas funcionales realizadas

Los problemas detectados en el período de pruebas de validación se clasificaron en: No conformidades significativas **(NCS)** y en No conformidades no significativas **(NCNS)**. A continuación, se describen los aspectos que se tuvieron en cuenta en cada clasificación.

NCS: son las no conformidades referentes a las funcionalidades de la aplicación: validaciones incorrectas o respuestas de la aplicación diferentes a lo descritas previamente.

NCNS: son las no conformidades en cuanto al diseño de la propuesta de solución y errores ortográficos.

Fueron realizadas 3 iteraciones de pruebas, en la primera iteración se detectaron 3 NCS y 2 NCNS. Fueron resueltas satisfactoriamente 2 NCS en la misma iteración, quedando pendientes 2 NCNS y 1 NCS.

No conformidades significativas

- El módulo mostró la información de una categoría distinta a la seleccionada por el usuario.
- El módulo no mostró ningún resultado al ejecutar una búsqueda en el listado de contenidos almacenados en la API de Picta, conociéndose que existe este contenido y que coincide con el criterio de búsqueda introducido**.**
- El módulo se detuvo al intentar ver un contenido sin una adecuada conexión.

No conformidades significativas No Significativas

- El módulo tiene faltas de ortografía en el texto mostrado en la vista “Add-ons video” en la interfaz principal de Kodi.
- El módulo tiene faltas de ortografía en los textos de descripción de la “serie“ que se quiere visualizar en pantalla.

En la segunda iteración no se detectaron nuevas no conformidades y de las 3 pendientes, se solucionan 1 NCS y 1 NCNS, siendo solucionadas

No Conformidades Significativas:

- El módulo no mostró ningún resultado al ejecutar una búsqueda en el listado de contenidos almacenados en la API de Picta, conociéndose que existe este contenido y que coincide con el criterio de búsqueda introducido**.**

No conformidades significativas No Significativas:

- El módulo tiene faltas de ortografía en el texto mostrado en la vista “Add-ons video” en la interfaz principal de Kodi.
- El módulo tiene faltas de ortografía en los textos de descripción de la “serie“ que se quiere visualizar en pantalla.

En la tercera iteración no se detectaron nuevas no conformidades y la quedaba pendiente,1 NCNS fue solucionada.

No conformidades significativas No Significativas:

- El módulo tiene faltas de ortografía en los textos de descripción de la “serie“ que se quiere visualizar en pantalla.

A continuación, se muestra un gráfico con un resumen de los resultados obtenidos tras la realización de las pruebas:

figura 14. Resultado de las pruebas funcionales. Fuente: Elaboración propia

3.8 Pruebas de integración

Las pruebas de integración se definen como un mecanismo de testeo de *software*, donde se realiza un análisis de los procesos relacionados con el ensamblaje o unión de los componentes, sus comportamientos con múltiples partes del sistema ya sea de archivos operativos o de *hardware*. De modo que las pruebas de integración están a cargo del examen de las interfaces entre los subsistemas o los grupos de componentes del programa o aplicación que se analiza, lo que contribuye a garantizar su correcto funcionamiento(56).A partir del requisito funcional 1: Conectar a los contenidos de streaming de la API de Picta, se deben tener en cuentas los siguientes aspectos:

1. El sistema operativo que se va a usar es GNU/Linux NOVA en su versión 8.0, en caso

   de estar usando otro sistema operativo, se debe instalar el antes mencionado, que se puede encontrar en *ucistore.cu* que está disponible de forma gratuita.

2. Verificar que se tiene conexión a internet.
3. Visitar el sitio Kodi download e Instalar la opción para Linux
4. Para la instalación de Kodi utilizar los siguientes comandos y ejecutar en ese mismo orden que se muestran:

   sudo apt install software -properties -common

   sudo apt -apt -repository -y ppa:team -xbmc/ppa

   sudo apt install kodi

   sudo apt install kodi -inputstream -adaptive kodi -inputstream -ffmpegdirect

5. Instalar extensión (add-ons) de video de Picta en Kodi, que sería el módulo desarrollado que va a permitir consumir servicios de *streaming* de la API de la plataforma Picta.
6. Para la instalación del módulo desarrollado solo se debe realizar dos acciones: seleccionar en Kodi la opción instalar desde carpeta zip y seleccionar la carpeta plugin.video.picta-kodi_19.zip

7. Seleccionar el contenido que se desea ver el cual hace una petición a la API de Picta, esta API se conecta a la plataforma y envía una respuesta a Kodi.
8. Reproducción del contenido seleccionado en forma de *streaming* sin necesidad de descarga.

A partir de lo anterior se demuestra (ver Anexos) que la solución desarrollada por el investigador resuelve el problema planteado.

Al finalizar la realización de las pruebas de integración se concluye que:

• La ejecución de las pruebas de integración permitió verificar el trabajo conjunto de los componentes del sistema en cuestión.

• No se encontró ninguna no conformidad. Se llega a la conclusión que existe una correcta integración entre los componentes internos del sistema.

Conclusiones del capítulo

La descripción del despliegue de la solución permitió la planificación de cada una de las iteraciones diseñadas para el completo desarrollo del módulo. En cada iteración se obtuvo un conjunto de componentes, relacionados en el diagrama de componentes, aportando una mejor compresión de la solución en términos de desarrollo. Además, las estrategias de pruebas realizadas permitieron identificar errores y corregirlos antes de liberar el producto.

CONCLUCIONES

Una vez finalizada la investigación se desarrolló un módulo de *streaming* de Picta para el televisor inteligente cubano con sistema operativo GNU/Linux Nova**,** concluyendo lo siguiente:

El estudio del proceso de como consumir servicios de *streaming*, específicamente en Kodi permitió comprender los principales conceptos para el desarrollo de un módulo en el televisor inteligente cubano.

Los módulos existentes para consumir servicios de *streaming* que se estudiaron, no representan una solución al problema, pero sirvieron de base para elaborar una idea general de cómo se desea representar la solución final.

El modelo conceptual enmarcó el contexto donde se usará el módulo y permitió la captura de requisitos funcionales y no funcionales, que fueron agrupados en historias de usuario generando una visión para la creación de un módulo de *streaming* de Picta para el televisor inteligente cubano con sistema operativo GNU/Linux Nova.

La implementación del módulo desarrollado posibilitó darles respuestas a los requisitos definidos y con ello a las necesidades del cliente.

La ejecución de pruebas al módulo desarrollado, siguió una estrategia de pruebas, que contempla pruebas unitarias, pruebas funcionales y pruebas de integración, arrojando como resultado un total de 5 no conformidades, corregidas en tres iteraciones, lo cual permitió la validación de todos los requisitos identificados y la aceptación por parte del cliente.

El módulo desarrollado tiene un gran aporte social, debido a que cualquier persona puede disfrutar servicios de *streaming* de la plataforma cubana de audiovisuales por excelencia Picta de manera gratuita. Aunque la solución de la presente investigación está orientada a los televisores inteligentes se puede hacer uso de este módulo en cualquier dispositivo, entre ellos, computadoras de escritorios, pc portátiles, entre otros.

RECOMENDACIONES

El objetivo general trazado en el presente trabajo de diploma fue alcanzado; no obstante, se recomienda:

- Desarrollar nuevos módulos para Picta en Kodi ya que este permite varios add-ons y el desarrollado fue orientado solo a consumir servicios de *streaming* de videos.
- Permitir que los usuarios puedan dejar sus comentarios desde Kodi una vez que disfrute el contenido de su preferencia.
- Permitir que los usuarios que tengan una cuenta en Picta puedan acceder a esta desde Kodi.

REFERENCIAS BIBLIOGRÁFICAS

1. CLARO. ¿Qué son las TIC? Y ¿Por qué son tan importantes? Online. 2019. [Accessed 9 June 2022]. Available from: https://www.claro.com.co/institucional/que-son-las-tic/

2. JIMÉNEZ, Network Contact: November 19, 2003 Categories: Latin America Western Europe Pensamiento EstratÃ©gico Academic Institutions Digital Television by Specific Theme Site Codes Latin America Give it 1/5 Give it 2/5 Give it 3/5 Give it 4/5 Give it 5/5 Your rating: None (13 votes) Los Medios de Comunicación frente a la Revolución de la Información Nhuna Daiana. Los Medios de Comunicación frente a la Revolución de la Información. *The Communication Initiative Network*. Online. 2021. [Accessed 9 June 2022]. Available from: https://www.comminit.com/content/los-medios-de-comunicaci%C3%B3n-frente-la-revoluci%C3%B3n-de-la-informaci%C3%B3nLos Medios de Comunicación frente a la Revolución de la Información Nhuna Daiana Jiménez J.Periodista.Durante cinco años ha sido Coordinadora de Información para la Región Central del "Circuit

3. SCHOOL, Euroinnova Business. Funciones de la televisión como medio de comunicación. *Euroinnova Business School*. Online. [Accessed 17 November 2022]. Available from: https://www.euroinnova.ec/blog/funciones-de-la-television-como-medio-de-comunicacionTodos conocemos el papel que tiene la televisión en la sociedad. Pero, ¿Conoces las funciones de la televisión como medio de comunicación? ¡Te lo contamos!

4. SCHOOL, Euroinnova Business. Funciones de la televisión como medio de comunicación. *Euroinnova Business School*. Online. 2022. [Accessed 9 June 2022]. Available from: https://www.euroinnova.ec/blog/funciones-de-la-television-como-medio-de-comunicacionTodos conocemos el papel que tiene la televisión en la sociedad. Pero, ¿Conoces las funciones de la televisión como medio de comunicación? ¡Te lo contamos!

5. LINARES, Iván. Cómo usar Kodi en Android para ver los canales de la TDT. *Xataka Android*. Online. 18 May 2021. [Accessed 14 June 2022]. Available from: https://www.xatakandroid.com/tutoriales/como-usar-kodi-android-para-ver-canales-tdt-1Kodi es un completo gestor de contenido multimedia que permite reproducir todo lo que tengas almacenado, también en Android. Y dispone de complementos para...

6. UCHA, Florencia. Televisor. *Definición ABC*. Online. 2011. [Accessed 9 June 2022]. Available from: https://www.definicionabc.com/tecnologia/televisor.phpEl televisor es un aparato electrónico que permite la recepción y reproducción de señales de televisión. Generalmente, consta de una pantalla y mandos o controles; fue creado el...

7. ANGULO, Mario. ¿Qué es Smart TV y para qué sirve? (Lista de todas funciones). *Qué TV Comprar*. Online. 14 August 2021. [Accessed 9 June 2022]. Available from: https://quetvcomprar.com/que-es-smart-tv-para-que-sirve/¿Qué es y para qué sirve una Smart TV? Aquí te cuento todas las funciones y cómo elegir tu Smart TV ideal, según las características varias.

8. JULIÁN PORTO and MERINO, María. Definición de smart TV — Definicion.de. *Definición.de*. Online. 2016. [Accessed 9 June 2022]. Available from: https://definicion.de/smart-tv/Smart TV es el concepto inglés que se utiliza para nombrar a los denominados televisores inteligentes. Se trata de un tipo de dispositivo con características…

9. VARGIC, Radoslav, TRÚCHL, Peter and PODHRADSK, Pavol. *Tecnologías inteligentes*. . 2019.

10. Definición de plataforma virtual - Definicion.de. *Definición.de*. Online. [Accessed 15 November 2022]. Available from: https://definicion.de/plataforma-virtual/Plataforma es un concepto con varios usos. Por lo general se trata de una base que se halla a una cierta altura o de aquello...

11. Sistema Operativo - Concepto, usos, tipos, funciones y ejemplos. *Concepto*. Online. [Accessed 9 June 2022]. Available from: https://concepto.de/sistema-operativo/Sistema operativo ✓ Te explicamos qué es un sistema operativo, cuáles son sus tipos, usos y componentes. Además, sus funciones y ejemplos.

12. ¿Qué es streaming? - Definición, significado y explicación | Verizon Fios. Online. [Accessed 15 November 2022]. Available from: https://espanol.verizon.com/info/definitions/streaming/

13. LINUXERO. Nova, la distribución GNU/Linux hecha en Cuba. *Desde Linux*. Online. 17 November 2015. [Accessed 9 June 2022]. Available from: https://blog.desdelinux.net/nova-la-distribucion-gnulinux-hecha-en-cuba/Nova es una distribución GNU/Linux creada por la Universidad de las Ciencias Informáticas (UCI) en Cuba. Esta forma de distribución se establece bajo la

14. TECHOPEDIA. ¿Qué es un módulo? - definición de techopedia - Desarrollo - 2022. Online. 2022. [Accessed 9 June 2022]. Available from: https://es.theastrologypage.com/module

15. ACADEMIC. Módulo (informática). *Los diccionarios y las enciclopedias sobre el Académico*. Online. 2022. [Accessed 10 June 2022]. Available from: https://es-academic.com/dic.nsf/eswiki/817057Para otros usos de este término, véase Módulo. En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente,…

16. Qué es el streaming - Definición, significado y explicación. *Verizon Fios*. Online. [Accessed 25 October 2022]. Available from: https://espanol.verizon.com/info/definitions/streaming/Definición y explicación de streaming. Obtén más información acerca del significado de streaming con el diccionario de términos técnicos de Verizon.

17. Definición de YouTube — Definicion.de. *Definición.de*. Online. [Accessed 10 June 2022]. Available from: https://definicion.de/youtube/YouTube es un portal del Internet que permite a sus usuarios subir y visualizar videos. Fue creado en febrero de 2005 por Chad Hurley, Steve…

18. Star Plus: qué es, cuál es su contenido y cuánto cuesta. *Digital Trends Español*. Online. 7 June 2022. [Accessed 10 June 2022]. Available from: https://es.digitaltrends.com/entretenimiento/star-plus-que-es-contenido-precio/Se ha convertido rápidamente en una de las plataformas de streaming más exitosas del mercado. Te contamos qué es Star Plus, qué ofrece y cuánto cuesta.

19. BBVA. Netflix ¿Qué es y cómo funciona? Algunas recomendaciones. *BBVA NOTICIAS*. Online. 18 May 2018. [Accessed 10 June 2022]. Available from: https://www.bbva.com/es/netflix-que-es-y-como-funciona/Todo lo que necesita conocer sobre Netflix, la plataforma de transmisión de películas y series que conquistó a los paraguayos.

20. Llega a Picta la televisión cubana en streaming (+ Apk). *Cubadebate*. Online. 13 March 2020. [Accessed 10 June 2022]. Available from: http://www.cubadebate.cu/noticias/2020/03/13/llega-a-picta-la-television-cubana-en-streaming-apk/La plataforma cubana de contenidos audiovisuales Picta, desarrollada por la Universidad de las Ciencias Informáticas, sumó cuatro canales de la televisión cubana a las diversas propuestas que ya tiene la aplicación.

21. FERNÁNDEZ, Yúbal. Chromecast: qué es, cómo funciona y qué se puede hacer con él. *Xataka*. Online. 15 May 2022. [Accessed 14 June 2022]. Available from: https://www.xataka.com/basics/chromecast-que-como-funciona-que-se-puede-hacerVamos a explicarte todo lo que necesitas saber sobre el Chromecast de Google, explicándote qué es este dispositivo, cómo funciona y qué puedes llegar a hacer...

22. Queremos ayudarte | LG España. *LG ES*. Online. [Accessed 14 June 2022]. Available from: https://www.lg.com/es/posventa/guias-y-soluciones/television/como-saber-que-version-de-webos-tengoNODATA

23. webOS de LG: cómo actualizar, versiones y más del sistema de Smart TV. *El Output*. Online. [Accessed 14 June 2022]. Available from: https://eloutput.com/productos/smart-tv/lg-webos-smart-tv/Descubre todo lo que necesitas saber sobre webOS (incluida la versión 6.0), el sistema operativo de los televisores inteligentes de LG.

24. Qué es Apple TV y cómo instalarlo. *PcComponentes*. Online. [Accessed 14 June 2022]. Available from: https://www.pccomponentes.com/que-es-apple-tv-como-instalarPara convertir tu televisor en un smart TV con Apple TV. Un gran número de aplicaciones y acceso a las principales plataformas de vídeo. Ahora también con Apple TV de Cuarta generación.

25. FERNÁNDEZ, Yúbal. Kodi: qué es, cómo funciona y todo lo que puedes hacer con él. *Xataka*. Online. 29 April 2022. [Accessed 8 September 2022]. Available from:

https://www.xataka.com/basics/kodi-que-es-y-como-funcionaTe explicamos qué es Kodi y cómo puedes realizar la configuración inicial para empezar a utilizarla

26. FERNÁNDEZ, Yúbal. API: qué es y para qué sirve. *Xataka*. Online. 23 August 2019. [Accessed 6 October 2022]. Available from: https://www.xataka.com/basics/api-que-sirveVamos a explicarte qué son las API y para qué sirven estos protocolos que son una parte fundamental en el funcionamiento de las aplicaciones y webs actuales....

27. Metodologías de desarrollo de software | Universitat Carlemany. Online. 2020. [Accessed 10 June 2022]. Available from: https://www.universitatcarlemany.com/actualidad/metodologias-de-desarrollo-de-software

28. PIÑEIRO CÁRDENAS, Javier. *Herramienta web para la creación de personalizaciones de Nova Servidores*. . Universidad de ciencias informaticas, 2017.

29. GONZÁLEZ, Carlos Yordan, ARAGÓN, Yaniel Lázaro, HERNÁNDEZ, Eylin and BARCENA, Yarileidis. Módulo para el diseño de modelos entidad relación en la plataforma RDB-learnin. . 2017. Vol. 10, p. 11–25.

30. Visual Paradigm para UML. *Software.com.ar*. Online. [Accessed 10 June 2022]. Available from: https://software.com.arBusca por p visual-paradigm-para-uml.

31. CISNEROS, ALVARO HUMBERTO. Conceptos básicos y generales de la programación. *Educación Virtual Cisneros*. Online. [Accessed 10 June 2022]. Available from: https://www.evirtualcisneros.xyz/articulos/8tecnologia/25-conceptos-basicos-ygenerales-de-laprogramacion.EVIRTUALCISNEROS, Plataforma de educación virtual, tecnología, artículos, asesorias, plataforma colegios, evirtual,

32. PÉREZ, Andres. Primeros pasos en Python. Online. 2017. [Accessed 10 June 2022]. Available from: https://pybaq.co/blog/primeros-pasos-en-python/

33. MEDUSA: una nueva herramienta para el desarrollo de sistemas Brain-Computer Interface basada en Python. Online. [Accessed 10 June 2022]. Available from: https://uvadoc.uva.es/handle/10324/31372

34. El concepto de IDE. Online. [Accessed 14 June 2022]. Available from: https://www.redhat.com/es/topics/middleware/what-is-ideUn entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas comunes para desarrolladores en una sola interfaz de usuario gráfica (GUI).

35. Qué es Visual Studio Code y qué ventajas ofrece. *OpenWebinars.net*. Online. 22 July 2022. [Accessed 14 June 2022]. Available from: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/Si quieres saber más sobre el editor de código más utilizado en la actualidad, en este artículo vamos a profundizar en las virtudes de Visual Studio Code.

36. MENDOZA, José Eder Guzmán. S2C3-Modelo_Dominio. *HCI Collab*. Online. [Accessed 2 November 2022]. Available from: https://hci-collab.com/libro/modelo_dominio/Diseño del Modelo del Dominio Josefina Guerrero GarcíaFacultad de Ciencias de la ComputaciónBenemérita Universidad Autónoma de Pueblajosefina.guerrero@correo.buap.mx Resumen: En este capítulo se revisa el modelo de dominio, el cual es una representación visual de clases conceptuales u objetos de situaciones reales en un dominio. Los modelos de dominio también se han

37. JUNTA DE ANDALUCÍA. Ingeniería de requisitos | Marco de Desarrollo de la Junta de Andalucía. Online. 2022. [Accessed 19 September 2022]. Available from: https://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/ingenieria/ingenieria-requisitos

38. Requerimientos de Software - ProQuest. Online. [Accessed 19 September 2022]. Available from: https://www.proquest.com/openview/5dd30d795aa48c298bf4352792a93059/1?pq-origsite=gscholar&cbl=2027443Explore millions of resources from scholarly journals, books, newspapers, videos and more, on the ProQuest Platform.

39. Obtención de Requerimientos. Técnicas y Estrategia. *SG Buzz*. Online. [Accessed 19 September 2022]. Available from: https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategiaComo sabemos, un área de conocimiento de gran importancia en el desarrollo de software, es la ingeniería de requerimientos. Esta comprende las actividades de obtención (captura, descubrimiento y adquisición), análisis (y negociación), especificación, y validación de requisitos. Además, establece una actividad de gestión de requerimientos para manejar los cambios, mantenimiento y rastreabilidad de los requerimientos.

40. Modelo de prototipos: ¿qué es y cuáles son sus etapas? | Blog | Hosting Plus Perú. *Hosting Plus*. Online. 6 July 2021. [Accessed 19 September 2022]. Available from: https://www.hostingplus.pe/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/Cuando se desarrolla un programa o aplicación es habitual recurrir a un modelo de prototipos para poder presentar una versión previa y funcional que sirva como presentación o muestra del proyecto. La elaboración de prototipos es muy interesante para conseguir feedback en cuanto a requisitos, funcionalidad y operatividad, de forma que se pueda afrontar el desarrollo final del software de forma … Continued

41. Especificación de Requisitos del Sistema | Marco de Desarrollo de la Junta de Andalucía. Online. [Accessed 19 September 2022]. Available from: http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407

42. ADMIN. ¿Qué son los requisitos funcionales? Especificación, tipos, EJEMPLOS. *Ebooks Online*. Online. 1 January 2020. [Accessed 19 September 2022]. Available from: https://ebooksonline.es/que-es-un-requisito-funcional-especificacion-tipos-ejemplos/¿Quieres

averiguar mas sobre los requisitos funcionales en 2022?. Con solo un clic AQUÍencontrarás los mejores tipos y ejemplos

43. Requerimientos NO funcionales | OVA2_IngenieriaDeSoftware. Online. [Accessed 19 September 2022]. Available from: https://repositorio.konradlorenz.edu.co/micrositios/001-1527/requerimientos_no_funcionales.html

44. ¿Qué son las historias de usuario? *Blog ITDO - Agencia de desarrollo Web, APPs y Marketing en Barcelona*. Online. 10 August 2021. [Accessed 6 October 2022]. Available from: https://www.itdo.com/blog/que-son-las-historias-de-usuario/Las historias de usuario o User Story, forman parte del enfoque Agile y describen las características y necesidades de un software desde la perspectiva de un usuario, ayudando a alinear expectativas y evitar errores críticos en el futuro. Una historia de usuario puede considerarse como una preparación para establecer los

45. HMONG.WIKI. Diseño de software Descripción generalyConceptos de diseño. Online. [Accessed 6 October 2022]. Available from: https://hmong.es/wiki/Software_designEl diseño de software es el proceso mediante el cual un agente crea una especificación de un artefacto de software destinado a lograr objetivos , utilizando un conjunto de componentes primitivos y sujeto a restricciones . [1] El diseño de software puede referirse a “toda la actividad involucrada en la conceptualización, elaboración, implementación, puesta en marcha y, en última instancia, modificación de sistemas complejos” o "la actividad que sigue a la especificación de requisitos y antes de la programación , como ... [en] un software estilizado proceso de ingeniería ". [2]

46. ALICANTE, Servicio de Informática Universidad de. Modelo vista controlador (MVC). Online. [Accessed 6 October 2022]. Available from: https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.htmlServicio de Informática ASP.NET MVC 3 Framework. Introducción al modelo vista controlador (MVC)

47. MIRIAM. Qué son los Patrones de Diseño de software / Design Patterns. *Profile Software Services*. Online. 24 June 2020. [Accessed 7 October 2022]. Available from: https://profile.es/blog/patrones-de-diseno-de-software/Los Patrones de Diseño (Design Patterns) son una solución general, reutilizable y aplicable a diferentes problemas de diseño de software.

48. GRASP (Patrones de software de asignación de responsabilidad general) – Tech-dir. Online. [Accessed 7 October 2022]. Available from: https://tech-dir.net/grasp-patrones-de-software-de-asignacion-de-responsabilidad-general/

49. Los patrones del diseño software GRASP • JnjSite.com. *JnjSite.com*. Online. 3 February 2022. [Accessed 7 October 2022]. Available from: https://jnjsite.com/los-patrones-del-diseno-software-grasp/Continuando con la serie de posts sobre los principios y patrones de diseño del software llegamos a los básicos patrones GRASP. Estos patrones GRASP significan según sus siglas en inglés General…

50. HMONG.WIKI. GRASP (diseño orientado a objetos) PatronesyVer también. Online. [Accessed 4 November 2022]. Available from: https://hmong.es/wiki/GRASP_(object-oriented_design)Patrones (o principios) de software de asignación de responsabilidad general , abreviado GRASP , es un conjunto de “nueve principios fundamentales en el diseño de objetos y asignación de responsabilidad” [1] : 6 publicado por primera vez por Craig Larman en su libro de 1997 [cita requerida] Aplicación de UML y patrones .

51. ¿Qué es un modelo de implementación? Online. [Accessed 3 November 2022]. Available from: https://www.centrobanamex.com.mx/que-es-un-modelo-de-implementacion

52. GONZÁLEZ RODRÍGUEZ, Arlenis. *Subsistema de gestión de perfil de usuario para el buscador Orión*. Online. bachelorThesis. Universidad de las Ciencias Informáticas. Facultad 1, 2017. [Accessed 3 November 2022]. Available from: https://repositorio.uci.cu/jspui/handle/123456789/7963El presente trabajo propone un subsistema de gestión de perfil de usuario para el buscador Orión. Se analizaron diferentes fuentes bibliográficas relacionadas con las características, funcionalidades y utilización de perfiles de usuario en Sistemas de Recuperación de Información a nivel nacional e internacional. La solución consiste en un subsistema capaz de gestionar perfiles de usuario en el buscador Orión definiéndose un compendio de información personal, preferencias y comportamiento de navegación en la web. El proceso de desarrollo estuvo guiado por la metodología de software AUP en su versión UCI, seleccionándose como principales tecnologías: el marco de trabajo Symfony 2.7.9, el lenguaje de programación PHP 5.6, el sistema de gestión de base de datos PostgreSQL 9.5 y Visual Paradigm 8.0como herramienta para el modelado. Las pruebas de software aplicadas al subsistema de gestión de perfil de usuario, demostraron que es una solución funcional, segura, con un rendimiento adecuado y que se integra sin dificultad al buscador Orión. Los resultados de la investigación evidenciaron que el sistema desarrollado posee un alto valor para el buscador Orión al facilitar la identificación de las preferencias del usuario basado en su navegación.Accepted: 2019-09-12T15:33:51Z

53. GUILARTE DOMÍNGUEZ, Claudia Rafaela. *Sistema para la gestión de procesos en la Dirección de Extensión Universitaria.* Online. bachelorThesis. Universidad de las Ciencias Informáticas. Facultad 4, 2019. [Accessed 3 November 2022]. Available from: https://repositorio.uci.cu/jspui/handle/123456789/10152La Universidad de la Ciencias Informáticas es uno de los centros pertenecientes al Ministerio de Educación Superior (MES). Para su funcionamiento se distribuye en diferentes áreas entre las que se encuentra la Dirección de Extensión Universitaria, la cual realiza diferentes procesos que permiten mejorar el funcionamiento de la misma. Entre los cuales se encuentra la divulgación de actividades, gestión de concursos, reservación de los locales de la Universidad, así como la reservación de actividades externas. A pesar de la gestión que se les realiza a los procesos todavía estos tienen limitantes. Los sistemas relacionados con la Extensión Universitaria, no satisfacen los problemas que existen en la universidad. La presente investigación presenta un sistema para la gestión de procesos de la Dirección de Extensión Universitaria, el cual permite la gestión de los diferentes procesos que se realizan en la Dirección de Extensión Universitaria a partir de varias interfaces que facilitan la automatización de la gestión de dichos procesos. Se presenta

una estrategia de pruebas realizada al sistema, con el objetivo de entregar al cliente una solución confiable y libre, que pueda ser empleada como apoyo para un mejor funcionamiento de la Dirección de Extensión Universitaria.Accepted: 2022-05-05T12:32:00Z

54. Introducción al Testing: Pruebas funcionales. Online. [Accessed 3 November 2022]. Available from: https://www.qalovers.com/2021/04/pruebas-funcionales.htmlQALovers está especializado en el estudio, definición e implantación de servicios de pruebas de software.

55. KEEPCODING, Redacción. ¿Qué son las pruebas unitarias de software? Online. 1 August 2022. [Accessed 3 November 2022]. Available from: https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/Conoce aquí qué son las pruebas unitarias de software, sus características y utilidades relevantes para el desarrollo de programas o aplicaciones.

56. KEEPCODING, Redacción. ¿Qué son las pruebas de integración? Online. 5 August 2022. [Accessed 3 November 2022]. Available from: https://keepcoding.io/blog/que-son-las-pruebas-de-integracion/En este artículo aprenderás qué son las pruebas de integración, así como sus características, funciones y propiedades de mayor relevancia.

•

ANEXOS

Anexo 1: Entrevista realizada al personal que desarrollo Picta identificar las deficiencias

I. Entrevista realizada al personal que desarrollo Picta. Usted ha sido seleccionado para esta entrevista por su amplia experiencia en el proceso de desarrollo de la plataforma. Es necesario que responda a las siguientes preguntas basándose en su experiencia práctica y teórica como desarrollador.
Pregunta 1: ¿Usted utiliza Picta? ¿Cómo?
Pregunta 2: ¿Qué tipo de contenido prefieren consultar los usuarios? ¿Por qué?
Pregunta 3: ¿Cómo usted utiliza la tecnología en su día a día?
Pregunta 4: ¿Cuánto tiempo se necesita para poder consumir los servicios de Picta una vez que se realiza una petición?
II. Entrevista realizada a los usuarios para identificar las deficiencias en el proceso de consumir los servicios que ofrece la plataforma de audiovisuales Picta: Usted ha sido seleccionado para esta entrevista por ser un usuario que utiliza dicha aplicación. Es necesario que responda a las siguientes preguntas basándose en su experiencia como usuario.
Pregunta 1: ¿usted desde cuando usa Picta? ¿Cómo?
Pregunta 2: ¿Qué tipo de contenido usted prefiere consultar? ¿Por qué?
Pregunta 3: ¿Cómo utiliza la tecnología en su día a día?

Pregunta 4: ¿Tiene usted televisor inteligente en casa? ¿cómo consume servicios a internet con él?

Anexo 2: Historias de Usuario de los requisitos funcionales

Tabla 10.Historia de Usuario: Obtener videos de los canales. Fuente: Elaboración propia.

Historia de Usuario	
Número: 5	**Nombre del Requisito:** Obtener videos de los canales
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 2
Prioridad: Alta	**Tiempo Estimado:** 72 horas
Riesgo en Desarrollo: Alta	**Tiempo Real:** 96 horas
Descripción Obtener los videos de los canales según su tipo.	
Observaciones: El usuario debe verificar que tiene conexión a internet	
Prototipo de interfaces:	

Tabla 11.Historia de usuario: Mostrar canales Fuente: Elaboración propia.

Historia de Usuario	
Número: 6	**Nombre del Requisito:** Mostrar canales

Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 2
Prioridad: Alta	**Tiempo Estimado:** 54 horas
Riesgo en Desarrollo: Alto	**Tiempo Real:** 60 horas
Descripción: el usuario debe visualizar los canales a los que tiene acceso	
Observaciones: NA	
Prototipo de interfaces:	

Tabla 12.Historia de usuario: Obtener canales Fuente: Elaboración propia.

Historia de Usuario	
Número: 7	**Nombre del Requisito:** Obtener canales
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 2
Prioridad: baja	**Tiempo Estimado:** 14 horas
Riesgo en Desarrollo: bajo	**Tiempo Real:** 20 horas

Descripción: el usuario debe visualizar una respuesta de los canales que obtuvo después de la petición realizada.
Observaciones: NA
Prototipo de interfaces:

Tabla 13.Historia de usuario: Mostrar metadatos Fuente: Elaboración propia.

Historia de Usuario	
Número:8	**Nombre del Requisito:** mostrar metadatos
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 2
Prioridad: media	**Tiempo Estimado:** 20 horas
Riesgo en Desarrollo: medio	**Tiempo Real:** 22 horas
Descripción: Mostrar metadatos de videos, series, canales	
Observaciones: NA	
Prototipo de interfaces:	

Tabla 14.Historia de usuario: Reproducir videos Fuente: Elaboración propia.

Historia de Usuario	
Número:10	**Nombre del Requisito:** reproducir videos
Programador: Esteban Manuel Landrian Amaro	**Iteración Asignada:** 3
Prioridad: alta	**Tiempo Estimado:** 86 horas
Riesgo en Desarrollo: alto	**Tiempo Real:** 92 horas
Descripción: Reproducir el video que es solicitado por el usuario.	
Observaciones: NA	
Prototipo de interfaces:	

Anexo 3 Pruebas unitarias

```python
import requests
API_BASE_URL = "https://api.picta.cu/v2/"


def test_get_temporada():
    id=232
    url_temp = f"{API_BASE_URL}temporada/?serie_pelser_id={id}&ordering=nombre"
    r = requests.get(url_temp)

    assert r.status_code == 200
```

figura 15 Prueba unitaria al método get_temporada

```
C:\Users\Hp _PC\Desktop\plugin.video.picta-kodi_19\tests>pytest
============================================================ test session starts ============================================================
platform win32 -- Python 3.7.7rc1, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\Hp _PC\Desktop\plugin.video.picta-kodi_19\tests
collected 1 item

test_plugin.py .                                                                                                                    [100%]

============================================================ 1 passed in 0.84s =============================================================
```

figura 16: Resultado de la prueba unitaria al método get_temporada

Anexo 4 Incorporación del módulo de Picta a Kodi.

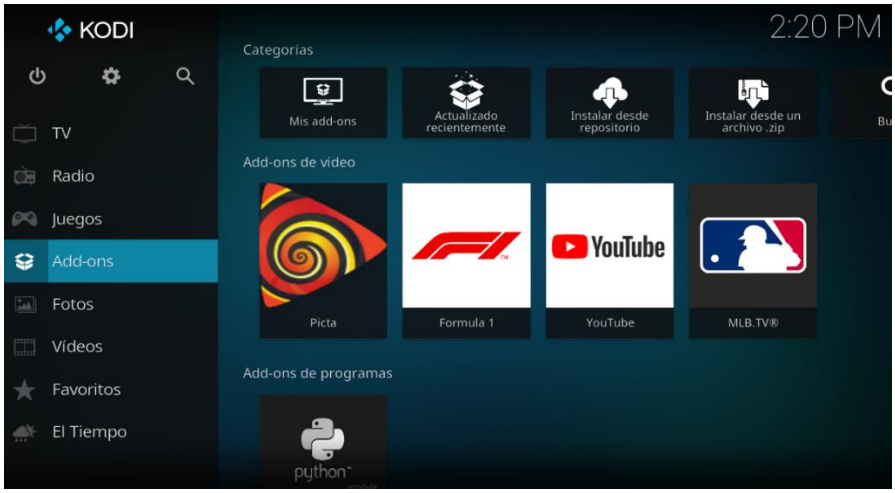


figura 17: Add-ons de video de Picta en Kodi.

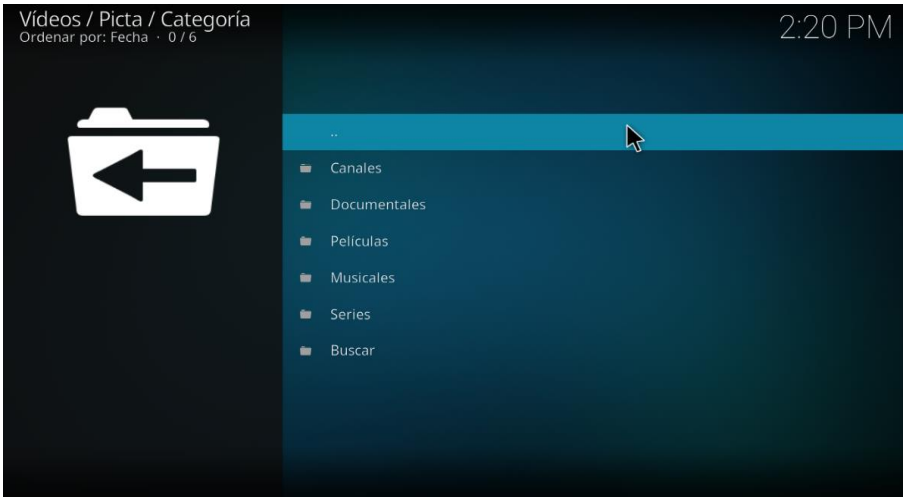


figura 18: Lista de videos por categorías.

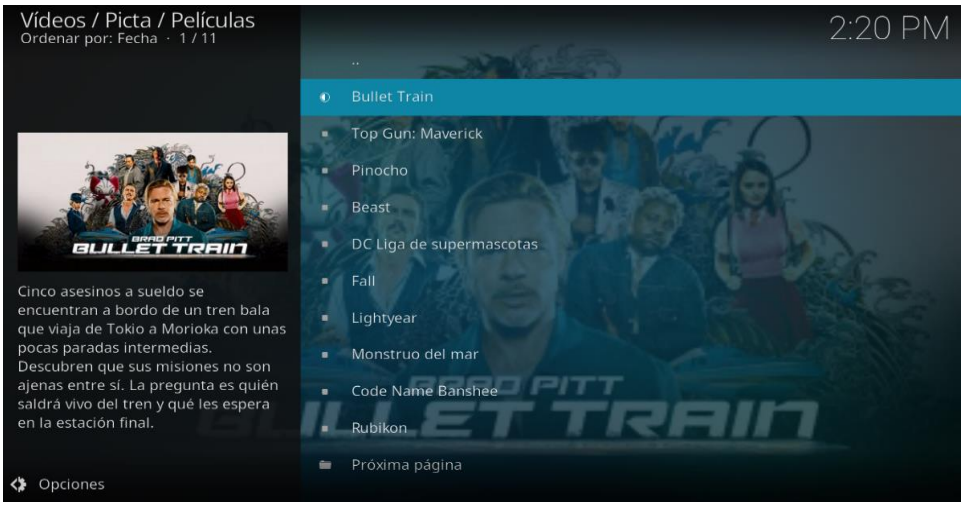


figura 19: Lista de películas con todos sus metadatos.

figura 20: Reproducción en streaming de una película.