**Universidad Tecnológica de La Habana “José Antonio Echeverría”**

**Facultad de Ingeniería Informática.**

**Filial de Ciencias Técnicas “Diez de Octubre”.**

****

****

**Sistema informático para la implementación del Número Internacional Normalizado para la Música en Cuba.**

**Trabajo de diploma para optar por el título de Ingeniería Informática.**

|  |  |
| --- | --- |
| **Autor:** | Antonio Cruz González  [antoniocruzglez24@gmail.com](mailto:antoniocruzglez24@gmail.com) |
| **Tutor:** | MsC. Juan Luis Rodríguez Dágueles |
|  | [juan.luis@desoft.cu](mailto:juan.luis@desoft.cu) |

**La Habana, Cuba**

**Mayo 2024**

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Grupo de Gestión de la Cámara Cubana del Libro de la Agencia Cubana del ISBN y a la Facultad de Ingeniería Informática para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 24 días del mes de mayo del 2024.





|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Antonio Cruz González |  |

**OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA**

El Trabajo de Diploma, titulado “Sistema informático para la implementación del Número Internacional Normalizado para la Música en Cuba”, fue realizado para el Grupo de Gestión de la Cámara Cubana del Libro (GGCCL). Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

X Totalmente

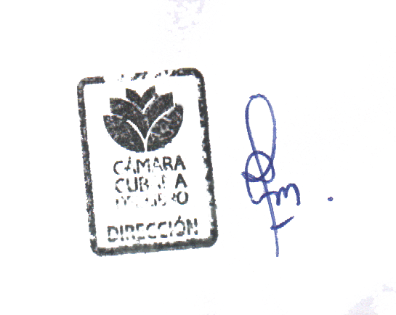
 Parcialmente en un \_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

* Una aplicación informática que permite una gestión automatizada de los datos referentes a las solicitudes y asignaciones de Números Internacionales Normalizados para la Música a piezas musicales de producción nacional.
* Ofrecer un nuevo servicio a sus clientes que les permite poder acceder a toda la colección de producciones musicales almacenadas en la entidad, así como la información, código de barra e ISMN asociado para cada publicación musical independiente, de una manera rápida, sencilla y con una vista novedosa y atractiva.
* Una vía rápida de exportar reportes útiles para el estudio de comportamiento de la aceptación y rechazos de las solicitudes, listado de existencia de publicaciones musicales, editoriales y editores independientes asociados y estadísticas de las solicitudes.
* Una mejor y más rápida gestión de las inscripciones de editoriales a la entidad.
* Un ahorro considerable en concepto de materiales e insumos de oficina

Como resultado de la implantación de este trabajo se reporta un efecto económico total que asciende a 32.000 CUP, 20.000 CUP por concepto de ahorro respecto al coste del software en el mercado internacional y 12.000 CUP por concepto de ahorro en materiales de oficina y salario de trabajadores.

Y para que así conste, se firma la presente a los \_24\_ días del mes de \_mayo\_ del año \_2024\_

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nombre del representante de la entidad Cargo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Resumen**

El presente proyecto está encaminado a desarrollar una herramienta informática para insertar en Cuba todo lo relacionado con la asignación, control y gestión del Número Internacional Normalizado para la Música en la Cámara Cubana del Libro y la Agencia del ISMN en Cuba.

El objetivo principal de esta inserción en el país es la identificación única de las publicaciones de música notada para la producción, publicidad, envíos, estadísticas, control de existencia, búsquedas, catalogación y otros usos en editoriales y agencias, todo de forma digital.

En la actualidad la gestión manual de esta información acarrea diversas dificultades debido a la naturaleza compleja de todos los datos a tener en cuenta para afrontar los distintos escenarios de uso. Mediante el uso de esta aplicación, los especialistas de la Agencia solo se encargarán de revisar las solicitudes y una vez aceptadas el sistema automáticamente generará los dígitos de control, códigos de barra, datos de la publicación, alertas, estadísticas y otras informaciones necesarias para ser enviadas a las editoriales o personas no jurídicas.

Para el análisis y diseño se emplearán prácticas del Proceso Unificado de Rational (RUP) y para la implementación se utilizarán tecnologías de software libre para el desarrollo Web principalmente el marco de trabajo Django basado en el lenguaje de programación Python.

**Palabras clave:** ISMN, ISBN, Numero Internacional Normalizado, código de partitura, Agencia de ISMN en Cuba, codificación de obras musicales.

**Abstract**

This project is aimed at developing a computer tool to digitize everything related to the allocation, control and management of ISMN, International Standard Music Number, in the Cuban ISMN Agency and Cuban Book Chamber.

The main objective of the ISMN insertion in the country is the unique identification of noted music publications for production, advertising, shipping, billing, statistics, stock control, search, cataloging and other uses in publishers and agencies, all in a digital way.

Currently, the manual management of this information entails several difficulties due to the complex nature of all the data to be taken into account to address the different use scenarios. Through the use of this application, the Agency’s specialists will only be in charge of reviewing the applications and once accepted, the system will automatically generate the control digits, barcodes, publication data, alerts, statistics and other information necessary to be sent to publishers or applicants.

For the analysis and design, Rational Unified Process (RUP) practices will be used and for the implementation, free software technologies for Web development will be used, mainly the Django framework based on the Python programming language.

**Keywords:** International Standard Music Number, ISMN, ISBN, score music code, ISMN Cuban Agency, codification of musical works.

Índice

[**Introducción** 1](#_Toc167378318)

[Antecedentes 1](#_Toc167378319)

[Área de Desarrollo 2](#_Toc167378320)

[Situación Problemática 3](#_Toc167378321)

[Problema 3](#_Toc167378322)

[Objeto de estudio 3](#_Toc167378323)

[Campo de acción 3](#_Toc167378324)

[Objetivo general 4](#_Toc167378325)

[Objetivos Específicos 4](#_Toc167378326)

[Hipótesis del trabajo 6](#_Toc167378327)

[Valor práctico 6](#_Toc167378328)

[Estructuración del contenido. 7](#_Toc167378329)

[**Capítulo 1 Fundamentos teóricos.** 9](#_Toc167378330)

[1.1 Introducción 9](#_Toc167378331)

[1.2 Objetivos estratégicos de la organización. 9](#_Toc167378332)

[1.2.1 Flujo actual de los procesos. 9](#_Toc167378333)

[1.2.2 Análisis crítico de la ejecución de los procesos. 10](#_Toc167378334)

[1.3 Procesos objeto de automatización. 11](#_Toc167378335)

[1.4 Sistemas automatizados existentes vinculados al campo de acción. 12](#_Toc167378336)

[1.5 Fundamentos de los objetivos. 13](#_Toc167378337)

[1.6 Tendencias y tecnologías actuales para el desarrollo web. 13](#_Toc167378338)

[1.6.1 Metodologías de desarrollo de software 14](#_Toc167378339)

[1.6.2Herramientas CASE (Computer Aided Software Engineering) 16](#_Toc167378340)

[1.6.3 Tendencias tecnológicas 17](#_Toc167378341)

[1.6.4 Lenguajes de programación. 18](#_Toc167378342)

[1.6.5 Frameworks de desarrollo web más utilizados en la actualidad. 21](#_Toc167378343)

[1.6.6 Entorno de desarrollo integrado (IDE) 26](#_Toc167378344)

[1.6.7 Sistemas Gestores de Bases de Datos (SGBD / DBMS) 27](#_Toc167378345)

[1.6.8 Servidor Web 29](#_Toc167378346)

[1.6.9 Tecnologías seleccionadas 32](#_Toc167378347)

[1.7 Análisis crítico de las fuentes y bibliografías utilizadas (Estado del arte). 33](#_Toc167378348)

[1.8 Conclusiones 33](#_Toc167378349)

[**Capítulo 2. Modelo de negocio.** 34](#_Toc167378350)

[2.1 Introducción 34](#_Toc167378351)

[2.2 Modelo de Negocio Actual 34](#_Toc167378352)

[2.3 Reglas del negocio 34](#_Toc167378353)

[2.4 Actores del negocio 36](#_Toc167378354)

[2.5 Diagrama de casos de uso del negocio 36](#_Toc167378355)

[2.6 Trabajadores del negocio 37](#_Toc167378356)

[2.7 Casos de uso del negocio 38](#_Toc167378357)

[2.7.1 Caso de Uso del Negocio “Solicitar ISMN” 38](#_Toc167378358)

[2.7.2 Caso de Uso del Negocio “Solicitar registro de Inscripción” 39](#_Toc167378359)

[2.7.3 Diagrama de actividades del caso de uso “Solicitar ISMN” 40](#_Toc167378360)

[2.7.4 Diagrama de actividades del caso de uso “Solicitar registro de Inscripción”. 41](#_Toc167378361)

[2.8 Modelo de objetos 42](#_Toc167378362)

[2.8.1 Descripción de las Entidades del Negocio 43](#_Toc167378363)

[2.9 Conclusiones 45](#_Toc167378364)

[**Capítulo 3 Requisitos** 46](#_Toc167378365)

[3.1 Introducción 46](#_Toc167378366)

[3.2 Definición de los requisitos funcionales 46](#_Toc167378367)

[3.3 Actores del sistema a automatizar 49](#_Toc167378368)

[3.4 Jerarquía de actores del sistema 51](#_Toc167378369)

[3.5 Paquetes y sus relaciones 52](#_Toc167378370)

[3.5.1 Diagrama de Paquetes del Sistema 52](#_Toc167378371)

[3.6 Diagrama de casos de uso del sistema a automatizar 53](#_Toc167378372)

[3.6.1 Diagrama de Casos de Uso del Sistema del paquete “Seguridad” 54](#_Toc167378373)

[3.6.2 Diagrama de Casos de Uso del Sistema del paquete “Reportes y Alertas” 55](#_Toc167378374)

[3.6.3 Diagrama de Casos de Uso del Sistema del paquete “Nomencladores” 56](#_Toc167378375)

[3.6.4 Diagrama de Casos de Uso del Sistema del paquete “Gestión” 56](#_Toc167378376)

[3.7 Descripción de los casos de uso 57](#_Toc167378377)

[3.8 Definición de los requisitos no funcionales. 61](#_Toc167378378)

[3.9 Conclusiones 63](#_Toc167378379)

[**Capítulo 4 Descripción de la solución propuesta.** 64](#_Toc167378380)

[4.1 Introducción. 64](#_Toc167378381)

[4.2 Diagrama de clases del diseño. 64](#_Toc167378382)

[4.2.1 Diagrama de diseño. Paquete “Seguridad” 65](#_Toc167378383)

[4.2.2 Diagrama de diseño. Paquete “Gestión” 66](#_Toc167378384)

[4.2.3 Diagrama de diseño. Paquete “Nomencladores” 67](#_Toc167378385)

[4.2.3 Diagrama de diseño. Paquete “Reportes y Alertas” 68](#_Toc167378386)

[4.3 Principios de diseño. 69](#_Toc167378387)

[4.3.1 Interfaz de usuario 70](#_Toc167378388)

[4.3.2 Formato de salida de los reportes. 72](#_Toc167378389)

[4.4 Tratamiento de errores 74](#_Toc167378390)

[4.5 Diseño de la base de datos. 75](#_Toc167378391)

[4.5.1 Modelo lógico de datos. 77](#_Toc167378392)

[4.5.2 Modelo físico de datos. 78](#_Toc167378393)

[4.6 Diagrama de despliegue. 79](#_Toc167378394)

[4.6.1 El flujo de las aplicaciones Django y el Modelo-Vista-Plantilla (MVT). 82](#_Toc167378395)

[4.6.2 Estructuración en Capas. 84](#_Toc167378396)

[4.7 Conclusiones 85](#_Toc167378397)

[**Capítulo 5. Validación y factibilidad de la solución propuesta.** 86](#_Toc167378398)

[5.1 Introducción. 86](#_Toc167378399)

[5.2 Tipos de pruebas ejecutadas. 86](#_Toc167378400)

[5.3 Diseño de los casos de prueba. 89](#_Toc167378401)

[5.3.1 Pruebas de validación 89](#_Toc167378402)

[5.3.2 Pruebas de Funcionalidad 94](#_Toc167378403)

[5.4 Resultados obtenidos en las pruebas. 98](#_Toc167378404)

[5.5 Beneficios tangibles e intangibles. 99](#_Toc167378405)

[5.5.1 Introducción 99](#_Toc167378406)

[5.5.2 Planificación basada en casos de uso 99](#_Toc167378407)

[5.5.1 Beneficios tangibles: 107](#_Toc167378408)

[5.5.2 Beneficios intangibles: 107](#_Toc167378409)

[5.6 Análisis de costos y beneficios 107](#_Toc167378410)

[5.7 Conclusiones 109](#_Toc167378411)

[**Conclusiones** 110](#_Toc167378412)

[**Recomendaciones** 111](#_Toc167378413)

[**Referencias** 112](#_Toc167378414)

Índice de tablas

[Tabla 1. Descripción de los actores del negocio. 35](#_Toc167362371)

[Tabla 2. Descripción de los trabajadores del negocio 37](#_Toc167362372)

[Tabla 3. Caso de Uso del Negocio "Solicitar ISMN" 38](#_Toc167362373)

[Tabla 4. Caso de Uso del Negocio "Solicitar registro de nuevo Editor" 38](#_Toc167362374)

[Tabla 5. Descripción de las Entidades del Negocio 44](#_Toc167362375)

[Tabla 6. Actores del sistema a automatizar 50](#_Toc167362376)

[Tabla 7. Descripción del caso de uso “Gestionar ISMN”. 57](#_Toc167362377)

[Tabla 8. Descripción del caso de uso "Visualizar registro de Editores" 57](#_Toc167362378)

[Tabla 9. Descripción del caso de uso "Gestionar Editores" 58](#_Toc167362379)

[Tabla 10. Descripción del caso de uso "Enviar Solicitud de ISMN" 59](#_Toc167362380)

[Tabla 11. Descripción del caso de uso "Realizar búsqueda general" 60](#_Toc167362381)

[Tabla 12. Descripción caso de uso de prueba <Autenticar, Registrar Usuario> 92](#_Toc167362382)

[Tabla 13. Descripción caso de uso de prueba <Gestionar Solicitud> 96](#_Toc167362383)

[Tabla 14. Factores de Peso de los actores sin ajustar 99](#_Toc167362384)

[Tabla 15. Complejidad de los actores del sistema. 99](#_Toc167362385)

[Tabla 16. Factor de peso de los casos de uso sin ajustar. 100](#_Toc167362386)

[Tabla 17. Pesos de los casos de uso sin ajustar 101](#_Toc167362387)

[Tabla 18. Factores de complejidad técnica. 102](#_Toc167362388)

[Tabla 19. Factores ambientes del sistema 103](#_Toc167362389)

[Tabla 20. Estimación de esfuerzo en horas. 104](#_Toc167362390)

[Tabla 21. Análisis de costo y beneficio. 107](#_Toc167362391)

Índice de figuras

[Figura 1. Fases del ciclo de vida de RUP. 15](#_Toc167189979)

[Figura 2. Lenguajes de programación más populares en el año 2020. 21](#_Toc167189980)

[Figura 3. Frameworks backend más populares en el año 2022. 24](#_Toc167189981)

[Figura 4. Número de empleos disponibles en 2022 para los Frameworks Frontend más utilizados. 25](#_Toc167189982)

[Figura 5. Entorno de desarrollo integrado (IDE) más usados en el año 2019. 27](#_Toc167189983)

[Figura 6. Sistemas Gestores de Bases de Datos más populares en marzo de 2022. 29](#_Toc167189984)

[Figura 7. Comparativa entre los servidores web. 32](#_Toc167189985)

[Figura 8. Diagrama de casos de uso. 37](#_Toc167189986)

[Figura 9. Diagrama de actividades del caso de uso "Solicitar ISMN". 40](#_Toc167189987)

[Figura 10. Diagrama de actividades del caso de uso "Solicitar registro de nuevo Editor. 41](#_Toc167189988)

[Figura 11. Modelo de objetos. 42](#_Toc167189989)

[Figura 12. Jerarquía de actores del sistema. 51](#_Toc167189990)

[Figura 13. Diagrama de Paquetes del Sistema. 52](#_Toc167189991)

[Figura 14. Diagrama de Casos de Uso del Sistema del paquete "Seguridad". 54](#_Toc167189992)

[Figura 15. Diagrama de Casos de Uso del Sistema del paquete "Reportes y Alertas". 54](#_Toc167189993)

[Figura 16. Diagrama de Casos de Uso del Sistema del paquete "Nomencladores". 55](#_Toc167189994)

[Figura 17. Diagrama de Casos de Uso del Sistema del paquete "Gestión" 56](#_Toc167189995)

[Figura 18. Diagrama de diseño. Paquete "Seguridad". 64](#_Toc167189996)

[Figura 19. Diagrama de diseño. Paquete "Gestión. 65](#_Toc167189997)

[Figura 20. Diagrama de diseño. Paquete "Nomencladores". 66](#_Toc167189998)

[Figura 21. Diagrama de diseño. Paquete "Reportes y Alertas". 67](#_Toc167189999)

[Figura 22. Página de inicio de la aplicación. 70](#_Toc167190000)

[Figura 23. Página de Lista de Editores. 71](#_Toc167190001)

[Figura 24. Página de Colecciones Musicales. 71](#_Toc167190002)

[Figura 25. Reporte Estadístico de Solicitudes Página 1. 72](#_Toc167190003)

[Figura 26. Reporte Estadístico de Solicitudes Página 2. 72](#_Toc167190004)

[Figura 27. Error que muestra el sistema al intentar insertar una Publicación Musical mientras aún hay solicitudes ISMN pendientes. 73](#_Toc167190005)

[Figura 28. Error que muestra el sistema al intentar acceder a una URL que no existe en el sistema. 74](#_Toc167190006)

[Figura 29. Transición del modelo lógico de datos al modelo físico. 75](#_Toc167190007)

[Figura 30. Modelo lógico de datos. 76](#_Toc167190008)

[Figura 31. Modelo físico de datos. 77](#_Toc167190009)

[Figura 32. Diagrama de despliegue. 78](#_Toc167190010)

[Figura 33. Colaboración entre los componentes de un MVC. 80](#_Toc167190011)

[Figura 34. Django Patrón MTV. 81](#_Toc167190012)

[Figura 35. Estructuración en capas. 84](#_Toc167190013)

# **Introducción**

## Antecedentes

El International Standard Music Number o Número Internacional Normalizado para la Música (Música escrita o Música impresa), en lo adelante ISMN por sus siglas en inglés, ha sido bastante exitoso durante los más de treinta años de su existencia. Actualmente sesenta y cinco agencias ISMN son miembros activos del sistema como representantes de países y de regiones.[1]

Sería deseable que se desarrollaran más aplicaciones del ISMN en la industria de la música ya que la globalización se refleja hoy en día, a través del crecimiento de los proveedores en Internet, quienes ofrecen gran cantidad de partituras y así reducen la brecha entre los países tradicionales productores de música y los amantes de la música en todo el mundo. El ISMN es el identificador ideal para este material en Internet, en virtud de ser compatible con los sistemas de identificación más especializados y aprobado por la ISO TC 46/SC 9 [2].

La aplicación de este estándar no era objetivo en Cuba, pero durante el curso del año 2021 ante un crecimiento en los registros de la música notada por la Cámara del Libro y llegar a almacenar más de cuatrocientas publicaciones musicales ese año, el país solicita ser miembro de la Agencia Internacional del ISMN y por tanto se ha venido preparando metodológicamente y técnicamente para implementar su uso, esto permitirá tener un registro sistemático de información estadística, año a año, con las piezas musicales inscritas por las empresas y editoriales.

En este nuevo año 2024 están registradas en la Cámara Cubana del Libro noventa empresas de grabaciones y ediciones musicales de las cuales existen setenta publicando anualmente de manera activa, con alrededor de cinco mil editores, compositores y asociados en todo el país y alrededor de diez mil números musicales archivados y un promedio de varios cientos de nuevas piezas por año entre partituras, partituras de estudio y vocales, libretos, publicaciones electrónicas, antologías, microformas, Brailles, entre otras. Estos datos representan inconvenientes para un manejo rápido y efectivo de la información mediante la realización manual de procesos importantes para las editoriales y editores independientes, como pueden ser:

* Identificación única para publicaciones de música impresa.
* Pedidos y distribución de publicaciones.
* Contabilidad y procesamientos ágiles.
* La compilación de directorios (música impresa).
* Creación y asignación de códigos de barras o códigos QR a las publicaciones musicales.
* Facturación de contratos.
* Gestión de derechos de autor.
* Catalogación por copia en las bibliotecas.
* Compilación de estadísticas actualizadas.
* Administración a nivel nacional del derecho de préstamo público.
* Enlaces con estándares relacionados.

### Área de Desarrollo

La Agencia Cubana del ISBN, encargada de la organización de la Ferie del Libro en Cuba, establece la obligación de un uso de estandarización para todas las publicaciones del país, incluida la música notada publicada en Cuba tal como ocurre en diferentes países del mundo. Su asignación solo podrá ser solicitada y asignada a los órganos editores según establece la [Resolución no. 87 del 27 de septiembre de 2007](https://www.claustrofobias.com/resolucion-no-87-2007-reglamento-del-registro-nacional-del-internacional-standard-book-number-isbn), del ministerio de Cultura. A raíz de la reciente incorporación de Cuba a la Agencia Internacional ISMN, las editoriales, artistas y editores independientes tienen la obligatoriedad de inscribirse en el Registro Nacional de ISMN para obtener el ISMN o de lo contrario obtenerlo a través de una empresa ya registrada [3].

A solicitud del Ministerio de Cultura, será la Agencia Cubana de ISBN adscripta a la Cámara Cubana del Libro (CCL) cita en calle 15 No. 605 esq. C, Vedado La Habana, la responsable de atender todo lo relacionado con la asignación, control y gestión de los ISMN.

A mediados del año 2022 un grupo de especialistas de la Agencia Cubana de ISBN comenzó la realización de las primeras asignaciones ISMN para algunas de las piezas archivadas más solicitadas y relevantes, almacenando los datos de las piezas junto al código ISMN, generado manualmente, en tablas de Excel. A diario pueden llegar a recibir más de 40 solicitudes provenientes de diferentes clientes, por lo que su administración, revisión y consecuente asignación de los códigos es más engorrosa y compleja, ha provocado duplicidades, pérdida de tiempo, rectificaciones y otros factores de riesgos humanos, además del notable aumento de documentos o tablas generadas para su gestión. La Agencia Cubana desestimó la asignación manual del ISMN tras estos errores.

## Situación Problemática

Actualmente Cuba no cuenta con una forma rápida y segura para identificar única e inequívocamente la música notada, la inserción y asignación manual de los ISMN, la generación de sus dígitos de verificación y códigos de barra, así como la reasignación de rangos de prefijos y los reportes estadísticos en el elevado número de producciones musicales tiende al incumplimiento de los tiempos establecidos, es propenso a errores, duplicidades e innumerables fallos.

## Problema

¿Cómo insertar el método de codificación ISMN en las publicaciones musicales de Cuba de una manera rápida y segura?

## Objeto de estudio

**Escenario del cliente/usuario**

* Procesos de registros, reglamentos y procedimientos para la identificación única de la música notada en Cuba.

**Escenario Informático**

* Disciplina de ingeniería de *software* y tendencias actuales del desarrollo web para análisis y gestión de la información.

## Campo de acción

**Escenario del cliente/usuario**

* Metodologías, manuales y normas empleadas en el uso de estándar internacional ISMN (Norma en particular utilizada).
* Resoluciones y reglamentos redactados por el MINCULT dirigido a las editoriales de la música y editores independientes.
* Arquitectura de la información almacenada en el centro.

**Escenario Informático**

* Patrones de arquitectura de software.
* *Frameworks* para el desarrollo web.
* Generación de tablas y códigos de barra en la web.
* Sistemas de alerta.
* Implementación de servicios web.

## Objetivo general

Desarrollar un sistema informático para la Agencia Cubana de la Música que permita el registro, gestión y asignación del ISMN en las existentes y nuevas producciones musicales que se vayan generando en Cuba.

## Objetivos Específicos

Para dar cumplimiento al objetivo general, se han trazado los siguientes objetivos específicos:

1. Describir el entorno de trabajo en la Agencia Cubana y sus requerimientos y estudiar normas y regulaciones sobre el correcto uso del ISMN en publicaciones musicales. Demostrar la factibilidad del proyecto
2. Realizar la modelación, análisis y diseño necesarios para la gestión y el procesamiento automatizado de la información.
3. Implementar un sistema que garantice la realización de los procesos relacionados con la aplicación del ISMN a nivel nacional.
4. Realizar las pruebas.

**Tareas**

1. Comprender el entorno de trabajo en la Agencia Cubana y estudiar normas y regulaciones sobre el correcto uso del ISMN en publicaciones musicales.
2. Entrevistar a directivos y especialistas que dominan el funcionamiento y los procesos de la Agencia Cubana de la Música.
3. Estudiar la documentación existente y el Manual de Usuario sobre el correcto uso de los ISMN.
4. Analizar los documentos legislativos pertinentes a los procesos de la Agencia aplicándolos correctamente en los procesos que lo requieran.
5. Sostener entrevistas con el equipo de especialistas del centro para identificar los principales requisitos funcionales y no funcionales del proyecto.
6. Realizar seguimiento en las solicitudes de los editores, la confección, gestión, asignación y entrega de las músicas notadas para detectar los principales problemas durante el proceso y comprender las diferentes etapas.
7. Realizar estimaciones de tiempo y costo del proyecto.
8. Determinar los beneficios tangibles e intangibles que aporta el sistema.
9. Realizar la modelación, análisis y diseño necesarios para la gestión y el procesamiento automatizado de la información.
10. Elegir las herramientas de diseño y desarrollo adecuadas para el proyecto, así como también el patrón de diseño a utilizar.
11. Diseñar los prototipos de interfaces visuales para la interacción de los usuarios con la aplicación.
12. Definir entidades del negocio y sus relaciones. Identificar sus atributos.
13. Confeccionar el modelo lógico y físico del sistema.
14. Implementar un sistema que garantice la realización de los procesos relacionados con la aplicación del ISMN a nivel nacional.
    1. Escribir el código fuente que permita la gestión de las entidades o clases modeladas.
    2. Elaborar manual de usuario con videos explicativos del funcionamiento de la plataforma.
    3. Establecer permisos y seguridad en la base de datos.
    4. Crear y generar el modelo físico del sistema.
    5. Implementar las interfaces de usuario.
    6. Implementar servicio Web.
    7. Realizar despliegue del sistema.
15. Realizar las pruebas
    1. Seleccionar los métodos y los tipos de prueba a realizar.
    2. Corregir los errores detectados en cada etapa.
    3. Comprobar la correcta validación de los datos ingresados, los permisos establecidos y la veracidad de la información emitida por el sistema.

## Valor práctico

El sistema permitirá a los especialistas de la Agencia poder administrar y gestionar lo referente a las solicitudes y asignaciones de ISMN a piezas de producción musical, podrán realizar registros de editores y autores en el sistema, insertar nuevas piezas serializadas, consultar resúmenes estadísticos, entre otras funcionalidades para un correcto procesamiento de la información que maneja la Agencia. A través del sistema se generarán y asignarán los ISMN, con su correspondiente código de barras, reportes estadísticos de solicitudes y Editoriales requeridos por las Editoriales y Editores independientes con gran rapidez. Se ofrecerá una interfaz agradable y dinámica para garantizar el acceso de los clientes a la información y que puedan realizar sus solicitudes de una manera rápida y sencilla. Todo lo relacionado con las nuevas inscripciones e información estadística del sistema editorial de música cubana estará en el formato adecuado para su salida y almacenada en una base de datos permitiendo organizarla y actualizarla de manera que esté siempre accesible a los especialistas y los editores en línea. El sistema aportará seguridad y confiabilidad en la gestión de los rangos y prefijos asignados a Cuba y la validez de los ISMN. Cada editorial tendrá su perfil en el sistema a través del cual podrá editar su propia información, enviar sus solicitudes ISMN y ver sus publicaciones en el apartado de ´Colecciones Musicales´.

## Estructuración del contenido.

El documento se encuentra estructurado en cinco capítulos. A continuación, se describen brevemente:

En el **Capítulo 1, Fundamentación teórica**, se presentan los temas referentes al objeto de estudio, comparando los sistemas existentes vinculados al campo de acción. Se definen los objetivos estratégicos de la organización y el flujo actual de los procesos. Se estudian las técnicas y tecnologías actuales y se realiza un análisis crítico de las fuentes y bibliografías consultadas.

En el **Capítulo 2, Modelo del negocio**, se hace referencia a la modelación del negocio actual, con la descripción de las reglas del negocio, los actores y trabajadores del negocio, los casos de uso y el modelo de objetos.

En el **Capítulo 3, Requisitos**, se aborda la temática sobre los requisitos funcionales y no funcionales del sistema, los paquetes definidos y los actores de cada caso de uso del sistema.

En el **Capítulo 4, Descripción de la solución propuesta**, se expone todo lo relacionado con la construcción del software mediante los diagramas de clase de diseño, el modelo físico de datos, la seguridad del sistema, el tratamiento de errores y el diagrama de despliegue.

En el **Capítulo 5 Validación y factibilidad de la solución propuesta**: Se expone una comparativa entre el tiempo y costo real que conllevó el desarrollo y el modelo de estimación basado en casos de uso, estudiado en clases. Además, se muestran los casos de uso de prueba realizados.

# **Capítulo 1 Fundamentos teóricos.**

## Introducción

En este capítulo se puntualizan los objetivos estratégicos de la Cámara Cubana del Libro, así como una descripción del flujo actual para la asignación de los ISBN en el país (Método similar al ISMN que ya está implantado en Cuba). Partiendo de lo anterior se identifican los procesos que serán objetos de automatización. También se abordan los fundamentos teóricos y conceptuales que guían la investigación y la revisión de software existentes asociados al campo de acción. Se fundamentan las tecnologías propuestas para el desarrollo del proyecto y se realiza un análisis crítico de la bibliografía empleada en la confección del trabajo.

## Objetivos estratégicos de la organización.

La Cámara Cubana del Libro se constituye oficialmente en 1997. Desde esa fecha, es la institución encargada de representar a las entidades editoriales, así como a las personas jurídicas y naturales, que conforman el sector del libro y de las publicaciones en la República de Cuba. En conjunto con la Agencia Cubana del ISBN también administra y gestiona el número ISBN, el código de barra a las publicaciones cubanas, reasignación de los rangos de prefijos que se le asignan al país y será la encargada de gestionar los rangos y prefijos ISMN asignados a Cuba. Su misión es profundizar la relación entre editores, editoriales y bibliotecas. [4]

### Flujo actual de los procesos.

Proceso de asignación actual para el ISMN.

El Grupo de Gestión de la Agencia se encarga de velar por la correcta asignación de códigos ISBN a las diferentes editoriales y editores independientes que lo solicitan para sus publicaciones y actualmente de asignar ISMN de manera manual a las principales obras musicales. Este grupo está formado por dos especialistas en bibliotecología que se encargan de recibir las solicitudes ISMN por parte de los editores, luego proceden a la revisión de la solicitud, verificar si es necesario la asignación de un nuevo prefijo al editor solicitante o reasignación de rangos, luego pasan a la confección y asignación del ISMN a la nueva publicación y almacenamiento de estas en las tablas correspondientes. Cuando el especialista determina la finalización de los procesos notifica al editor para la entrega de su ISMN y código de barra adjunto a su publicación.

Proceso de inscripción de un nuevo editor.

El Grupo de Gestión de la Agencia procede a la revisión de la solicitud, en caso de ser correcta se realiza una nueva tabla en Excel con los datos del nuevo Editor y se confecciona un acta de nuevas producciones que se le envía al nuevo Editor para la posterior asignación de los ISMN correspondientes y su identificación como parte de la Agencia. En caso que la solicitud presente errores se le envía por correo un documento de requisitos con los requerimientos y fallas en su solicitud.

### Análisis crítico de la ejecución de los procesos.

Actualmente el método de trabajo empleado para la asignación de ISMN y asociación de editores a la Agencia, no cumple a cabalidad los objetivos del centro, existiendo la necesidad de mejorarlo en algunos aspectos.

La utilización de tablas de Excel para la captura y el procesamiento de datos, constituye una tarea engorrosa y provoca errores de diversa índole.

Cada especialista durante la asociación de editores, asignación ISMN y registro de nuevas publicaciones musicales realiza una nueva tabla y esta se almacena en su PC personal, impidiendo unificar toda la información en un lugar específico; ésta queda dispersa en las PC de cada especialista.

No se cuenta con validación en la entrada de datos a la hora de realizar los ISMN, corriendo el riesgo de cometer errores.

Actualmente no se puede realizar los códigos de barra que representen los ISMN ya que no se cuenta con las herramientas que permitan llevar a cabo esta tarea.

No se cuenta con respaldo de la información, al no quedar ésta almacenada en un servidor dedicado, donde se proteja contra usos indebidos, ya que tampoco existe un control de acceso a los datos.

A todo esto, se le suma la falta de entrenamiento de los especialistas en la propia herramienta Excel, lo que demanda en ocasiones más tiempo del requerido para realizar alguna acción, por ejemplo, utilización de fórmulas matemáticas, reestructuración del contenido de las tablas, confección de gráficos de diversos tipos, etc.

## Procesos objeto de automatización.

Para dar solución a la problemática existente durante el proceso de gestión de información se propone la construcción de un sistema informático que facilite la ejecución de las labores de recopilación, procesamiento, creación y respaldo de la información.

Recopilación, creación y gestión de la información.

Desarrollo de un módulo para la recopilación, creación y gestión de la información, que permita la captura automatizada de los datos fundamentales de los editores asociados y las publicaciones musicales registradas, como son: título, fecha, autor, ISMN. Además, tenga la posibilidad de agregar otros datos como temática, tipo de obra, colección digital, entre otros. Esto simplifica el trabajo de los especialistas ya que muchos datos no tendrán que llenarlos manualmente y les proporciona una herramienta de trabajo para generar de forma sencilla los códigos de barras que necesiten.

Procesamiento de la información

Este módulo permitirá el procesamiento automatizado de la información, por tanto, en lo adelante, los especialistas no tendrán que pasar por la engorrosa tarea de tener que confeccionar las tablas de forma manual, ni tener que definir las fórmulas matemáticas que realicen los cálculos pertinentes. Lo mismo sucederá con los gráficos estadísticos. Éstos se crearán automáticamente. Dispondrá también de la opción de filtrar la información, por fecha, editorial, entre otros términos, actualizándose automáticamente, tanto las tablas como los gráficos, además podrá exportarse la información a diversos formatos (jpg, pdf, json). Se emitirán alertas automáticas en el caso, por ejemplo, de haberse agotarse los prefijos asignados a una editorial.

Respaldo e integridad de los datos.

Este módulo se encargará de las tareas de validación durante la entrada de datos, control de acceso, dígitos de control de los ISMN y también será el responsable de la realización de salvas periódicas de la información.

### Sistemas automatizados existentes vinculados al campo de acción.

Se realizó una investigación de las herramientas automatizadas existentes para las asignaciones y gestión de información de las Agencias ISMN internacionales.

U.S. ISMN Agency

Es un sitio web creado en Estados Unidos en 2018 que ofrece a cualquier persona que lo visite información relevante acerca del funcionamiento del ISMN y sus ventajas, muestra las colecciones digitales de origen estadounidense registradas en el sitio, contactos de la U.S. Agency y permite hacer búsquedas generales y avanzadas en sus catálogos. Desde su creación el sitio ha registrado casi 10000 obras musicales que recogen gran parte del patrimonio y la historia de los Estados Unidos de América. Particularmente el sitio les permite a sus usuarios registrar nuevas publicaciones, hacer solicitudes, enviar cuestionarios y asociarse a la Agencia, pero para la inscripción se necesita pagar una cuota.

Camlibro ISMN

Este sitio pertenece a la Agencia ISMN de Colombia y al igual que el sitio web anterior les permite a sus usuarios hacer solicitudes y recibir los ISMN por correo, pero como punto negativo no muestra información en la página principal acerca de la utilización del código ISMN y sus ventajas.

Bookchamber

Este sitio es de origen ruso y le permite a la Cámara Rusa del Libro poder gestionar de manera remota las solicitudes de ISMN e ISBN que los usuarios envían. Este sitio muestra información general acerca del procedimiento de solicitudes, así como también el correcto uso de los ISMN, pero este sitio no muestra colecciones digitales registradas ni otras ediciones de obras musicales lo que no lo hace un sitio enteramente informativo para quien lo visite.

**Consideraciones Finales**

Todos los países asociados a la Agencia Internacional ISMN deben tener un sitio web que los represente de manera global y muestre las obras musicales que forman parte del patrimonio cultural de cada país, además esto les facilita a las agencias la asignación de ISMN, gestión de solicitudes y asociaciones de editores. Por lo que se decide desarrollar un sitio web propio, que se ajuste a las características del entorno nacional y que brinde un servicio gratuito y de calidad a la Agencia Cubana del Libro mediante el empleo de tecnologías libres (open source).

## Fundamentos de los objetivos.

La solución que se propone en este trabajo está encaminada a la satisfacción de las necesidades generadas por la situación problemática existente. Para alcanzar el objetivo general planteado es necesario realizar un estudio detallado del modo actual en que se realizan los procesos implicados en el flujo de trabajo de análisis de información. Además de diseñar una base de datos que admita, de forma segura, el almacenamiento de la información necesaria para el desarrollo de las actividades involucradas y satisfacer las expectativas de los usuarios finales; proporcionándoles comodidad, confiabilidad y seguridad en el momento de interactuar con el sistema.

## 1.6 Tendencias y tecnologías actuales para el desarrollo web.

En la actualidad se ven avances notables en el área tecnológica, lo que ha permitido que las personas interactúen con las computadoras hasta el punto de incorporar las herramientas informáticas a su quehacer diario. Para el desarrollo de software existen muchas tendencias y tecnologías concebidas para facilitar el trabajo en todas las etapas del ciclo de vida de un proyecto hasta lograr convertirlo en un producto de calidad, lo que se ve reflejado en un mejor servicio al usuario. Ya sea para desarrollar aplicaciones web o para mantener un servidor web, es crucial conocer las tendencias y tecnologías actuales en el desarrollo web. A continuación, se realiza un análisis comparativo entre algunas de las tecnologías existentes candidatas a ser empleadas, fundamentalmente aquellas bajo licencias de software libre (free software, en inglés) y código abierto (open source, en inglés).

### 1.6.1 Metodologías de desarrollo de software

En el área de la Ingeniería de Software, el término «metodología» hace referencia a un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de software. Éstas han evolucionado a la par de los paradigmas de programación, tratando de ajustarse a las nuevas necesidades, proporcionando herramientas y prácticas que se acoplan mejor a los desarrollos actuales. Las metodologías se clasifican en tradicionales y ágiles. Las metodologías tradicionales proveen una estructura bien definida y un orden riguroso en el proceso de desarrollo. Por lo general tienen abundantes artefactos y requieren de disímiles roles. Invierten un tiempo considerable en la comprensión del negocio y el modelado del sistema. Perciben la captura de requisitos del proyecto como una fase previa al desarrollo del mismo que, una vez completada, debe proporcionar una «fotografía exacta» de lo que desea el cliente [5].

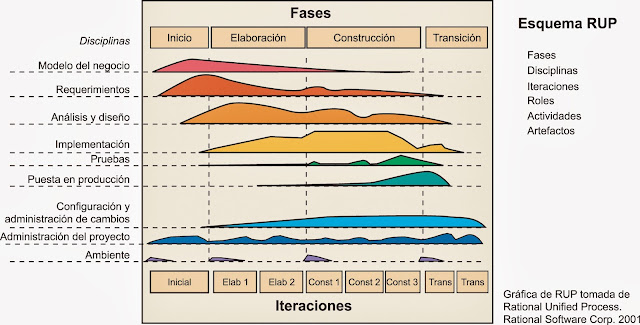
Entre las metodologías tradicionales destaca **RUP**, que se encuentra entre las más utilizadas para el análisis, implementación y documentación de sistemas.

Esta metodología se enfoca en el concepto de iteraciones cortas que deben finalizar con una entrega de software funcional, de la que pueden surgir modificaciones u otros requerimientos. No se someten a principios rígidos ni excesivos controles, proponen pocos artefactos y roles, y generan poca documentación. Entre las metodologías ágiles que han tenido gran acogida en el desarrollo de software destacan: **Scrum, Programación Extrema** (**XP**, por sus siglas en inglés), así como combinaciones de éstas [6].

Proceso Unificado de Desarrollo de Software (RUP)

Consiste en un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. Utiliza el lenguaje UML para preparar todos los esquemas de un sistema de software. RUP tiene tres características fundamentales, es dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que son: Inicio (Inception, en inglés), Elaboración (Elaboration en inglés), Construcción (Construction, en inglés) y Transición (Transition, en inglés). RUP a través de sus diferentes disciplinas, contribuye a una evolución favorable del desarrollo del proyecto y ayuda a una mejor comprensión, organización y estructuración del mismo. En este trabajo se utilizarán las disciplinas desde Requerimientos hasta Prueba, en sus diversas fases [7].

Las fases del ciclo de vida se pueden ver en la siguiente figura: [8]



1. Fases del ciclo de vida de RUP.

Scrum

Es un marco de trabajo para el desarrollo y mantenimiento de proyectos complejos. Tiene su origen en estudios llevados a cabo por Hirotaka Takeuchi e Ikujiro Nonaka en 1986 acerca de la eficiencia lograda por empresas tecnológicas que realizaban productos de buena calidad en menos tiempo y con menor costo que otras de su tipo. Para orquestar el proceso de desarrollo, Scrum define distintos roles: Dueño del producto (Product Owner), Equipo de desarrollo (Development Team) y Scrum Master. Las Historias de Usuario son el elemento base de Scrum. Describen las características que el cliente espera que tenga el software que se va a desarrollar [9], o sea, los objetivos o requisitos del sistema. Se presentan desde la perspectiva del usuario, utilizando un lenguaje cercano al dominio de la aplicación que sea comprensible para clientes y desarrolladores. Al inicio no es necesario detallar exhaustivamente todas las Historias de Usuario sino sólo las que tienen mayor nivel de prioridad para el cliente. Esto permite que el proceso de desarrollo pueda adaptarse a posteriores modificaciones, una demostración de la flexibilidad del modelo. Scrum es un proceso para la gestión y control de proyectos que «no dice nada sobre cómo hay que hacer las cosas», por tanto, se aplica o combina fácilmente con otras metodologías de desarrollo que están más orientadas a la Ingeniería de Software y la programación [9].

### 1.6.2Herramientas CASE (Computer Aided Software Engineering)

Visual Paradigm Enterprise Edition v17.0

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Algunas de sus características principales son:

* Ingeniería inversa: Código a modelo, código a diagrama.
* Generación de código: Modelo a código, diagrama a código
* Soporte ORM: Generación de objetos Java desde la BD.
* Generación de BD: Transformación de diagramas de Entidad-Relación en tablas de BD.
* Ingeniería inversa de BD: Desde SGBDs existentes a diagramas de Entidad-Relación.
* Generador de informes para generación de documentación.
* Distribución automática de diagramas: Reorganización de las figuras y conectores de los diagramas UML.

Rational Rose

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, que permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación. Proporciona mecanismos para realizar la ingeniería inversa, es decir, que a partir del código se pueda obtener información sobre su diseño; adicionalmente permite generar código en diferentes lenguajes a partir de un diseño en UML, brinda la posibilidad de que varias personas trabajen a la vez, permitiendo que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y permite que tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. El desarrollo es un proceso iterativo, que comienza con una aproximación del análisis, diseño e implementación para identificar los riesgos y probar el sistema, cuando la implementación pasa todas las pruebas que se determinan, se añaden los elementos modificados al modelo y una vez modificado se realiza la siguiente iteración. Rational, además, soporta los diagramas de UML, excepto los Diagramas de Implementación [10].

### 1.6.3 Tendencias tecnológicas

Inteligencia artificial (IA) y aprendizaje automático (ML).

La IA y el ML están revolucionando los métodos de desarrollo web en la actualidad. Básicamente, pueden mejorar una página web para ofrecer mejores experiencias de usuario, como chatbots que son realmente útiles, o la herramienta ChatGPT que es tan utilizada hoy en día. En comparación con el desarrollo de una aplicación web tradicional, una aplicación web de IA y ML necesita más ayuda en el desarrollo del back-end.

Aplicación web progresiva (PWA).

Las PWA son tendencia en 2023 debido al crecimiento de las redes 5G y a la disponibilidad de dispositivos de bajo coste y alto rendimiento que facilitan su acceso y uso. Los usuarios también se benefician de tener acceso a estas aplicaciones web incluso cuando sus dispositivos están desconectados o tienen una conectividad limitada.

Aplicación de una sola página (SPA).

Las SPA son una popular tendencia de desarrollo web que ofrece experiencias de usuario rápidas y con capacidad de respuesta. Cargan todo el HTML, CSS y JavaScript necesario en una sola página y actualizan dinámicamente el contenido sin necesidad de recargar la página.

### Lenguajes de programación.

**Lenguajes de programación del lado del servidor**

PHP

Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy. Lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por PHP como lenguaje de programación. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server y otros. Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. [Datos de W3Techs](https://w3techs.com/technologies/details/pl-php) muestra que un asombroso 78,9% de todos los sitios web con un lenguaje de programación del lado del servidor conocido utilizan PHP. Esto significa que casi 8 de cada 10 sitios web de Internet utilizan PHP [11].

Ventajas de PHP:

* Es un lenguaje multiplataforma
* Capacidad de conexión con la mayoría de los gestores de base de datos que se utilizan en la actualidad
* Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones)
* Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
* Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
* Permite las técnicas de programación orientada a objetos.
* Biblioteca nativa de funciones sumamente amplia.

Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. [12]  
  
En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

* La cantidad de biblioteca que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
* La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
* La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
* Además, Python es gratuito, incluso para propósitos empresariales

Java

Es un lenguaje de Programación Orientado a Objeto (POO) que incluye una combinación de características que lo hacen único, es válido para realizar todo tipo de aplicaciones empresariales profesionales y cumple con el requisito no funcional de portabilidad. Este lenguaje presenta características que lo diferencian de otros [13]:

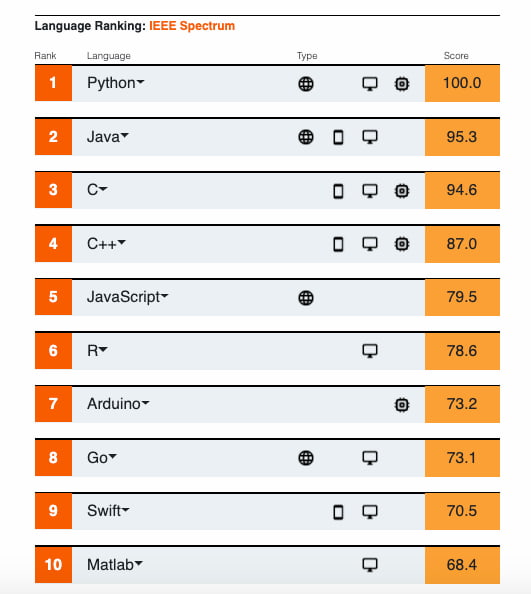
* Actualmente es público, puede conseguirse el Kit de Desarrollo de Java (Java Developer’s Kit, JDK por sus siglas en inglés) gratis.
* Se pueden escribir aplicaciones para intranet, aplicaciones clientes/servidor, aplicaciones distribuidas en redes locales y en Internet.
* Las aplicaciones creadas con Java son independientes de la arquitectura, se ejecutan indistintamente en gran variedad de equipo con diferente microprocesador y sistema operativo.
* La sintaxis de Java es similar a la de C++. Por tanto, no representa obstáculo alguno para los desarrolladores en C/C++ iniciarse en Java.
* Es robusto, cuenta con un fuerte mecanismo de manejador de excepciones, siendo estas un camino más para manejar los errores.
* Java es multihilo (Multithreading, en inglés), es decir que tiene la capacidad de ejecutar varios hilos de ejecución en paralelo.

**Lenguajes de programación del lado del cliente.**

JavaScript (JS).

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.[14]

**Los lenguajes de programación más populares de 2020, según Spectrum/IEEE, analizados con 11 métricas de 8 fuentes distintas**[15] **:**



1. Lenguajes de programación más populares en el año 2020.

### Frameworks de desarrollo web más utilizados en la actualidad.

Django

Django es un framework de Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Con Django, puede llevar aplicaciones web desde el concepto hasta el lanzamiento en cuestión de horas. Django se ocupa de gran parte de las molestias del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto. Puntos fuertes del framework[16] :

* Rapidez. Django fue diseñado para ayudar a los desarrolladores a llevar las aplicaciones desde el concepto hasta su finalización lo más rápido posible.
* Completamente cargado. Django incluye docenas de extras que puede usar para manejar tareas comunes de desarrollo web. Django se encarga de la autenticación del usuario, la administración de contenido, los mapas del sitio, las fuentes RSS y muchas más tareas, desde el primer momento
* Seguro. Django se toma la seguridad muy en serio y ayuda a los desarrolladores a evitar muchos errores de seguridad comunes, como la inyección de SQL, las secuencias de comandos entre sitios, la falsificación de solicitudes entre sitios y el secuestro de clics. Su sistema de autenticación de usuarios proporciona una forma segura de administrar las cuentas y contraseñas de los usuarios.
* Escalable. Algunos de los sitios más concurridos del planeta utilizan la capacidad de Django para escalar de forma rápida y flexible para satisfacer las demandas de mayor tráfico.
* Increíblemente versátil. Empresas, organizaciones y gobiernos han utilizado Django para crear todo tipo de cosas, desde sistemas de gestión de contenido hasta redes sociales y plataformas informáticas científicas.

Laravel

Laravel es un framework de aplicación web con una sintaxis expresiva y elegante. Creemos que el desarrollo debe ser una experiencia agradable y creativa para que sea realmente satisfactorio. Laravel intenta eliminar el dolor del desarrollo al facilitar las tareas comunes que se usan en la mayoría de los proyectos web, como[17] :

* Motor de enrutamiento simple y rápido.
* Potente contenedor de inyección de dependencia.
* Múltiples back-ends para sesiones y almacenamiento en caché.
* Migraciones de esquemas independientes de la base de datos.
* Robusto procesamiento de trabajos en segundo plano.
* Transmisión de eventos en tiempo real.

Laravel es accesible, pero potente, y proporciona las herramientas necesarias para aplicaciones grandes y sólidas. Una excelente combinación de simplicidad, elegancia e innovación le brinda un conjunto de herramientas completo necesario para crear cualquier aplicación que se le encomiende.

Express y Node.js

Node.js es una herramienta versátil para desarrolladores de JavaScript que amplía las capacidades de JavaScript añadiendo funcionalidad backend. Node.js es un motor V8 de Chrome que se ejecuta con JavaScript basado en eventos y de código abierto. Los desarrolladores prefieren Node.js por su capacidad para desarrollar aplicaciones web escalables con facilidad.[18]

Angular

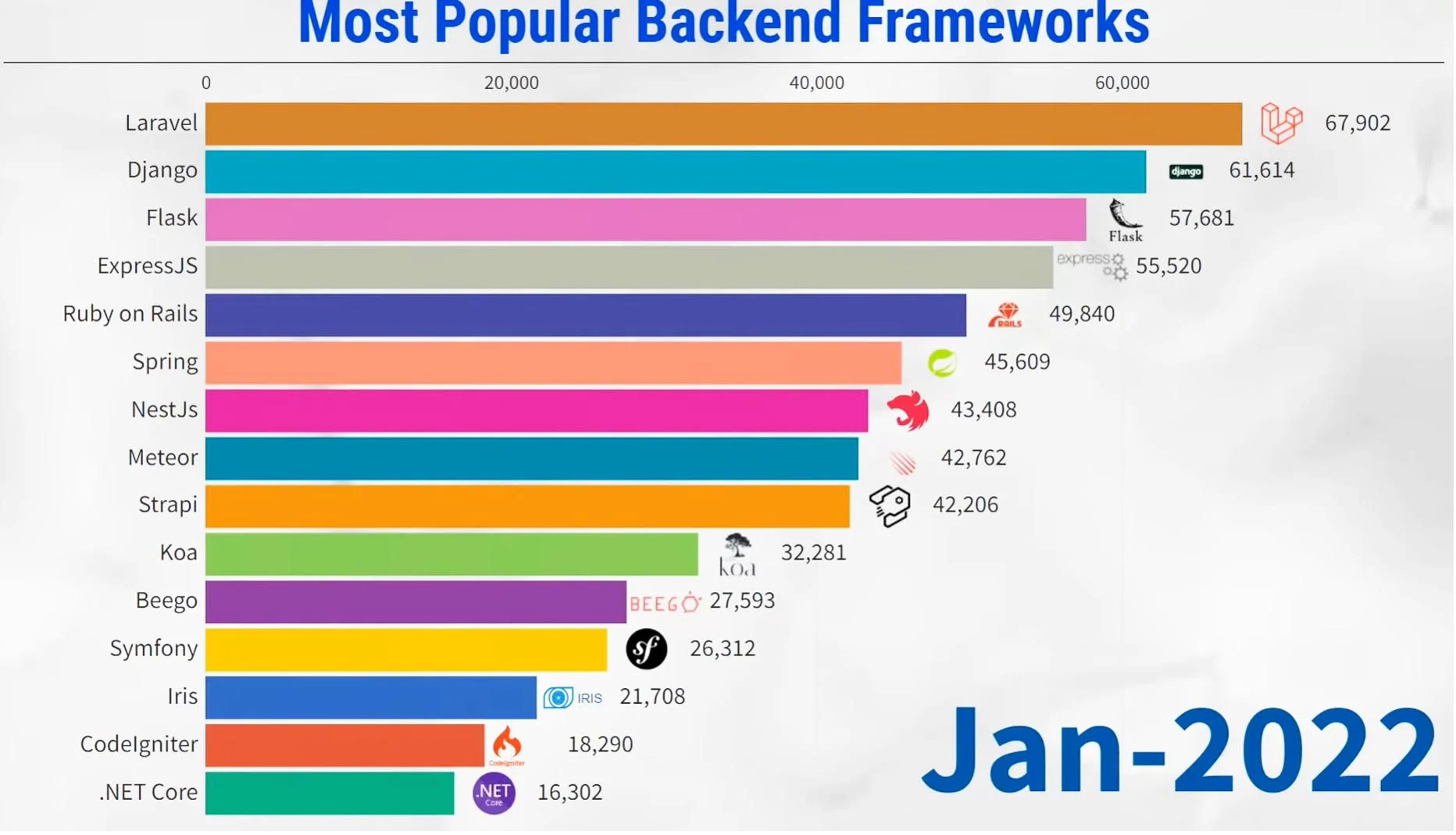
Angular es un framework de diseño de aplicaciones y plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas. Permite crear funciones rápidamente con plantillas simples y declarativas. Extiende el lenguaje de plantilla con componentes propios y usa una amplia gama de componentes existentes. Desde el prototipo hasta la implementación global, Angular ofrece la productividad y la infraestructura escalable que soportan a las aplicaciones más grandes de Google. [19]

React JS

React JS es esencialmente una biblioteca de JavaScript, aunque por sus innumerables funcionalidades y su fuerza en el mercado front-end muchos lo consideran un framework de desarrollo que le permite crear interfaces de usuario interactivas de forma rápida y sencilla. React le permite crear aplicaciones web dinámicas donde los datos en las UI se actualizan constantemente. Algunas de sus características son[20] :

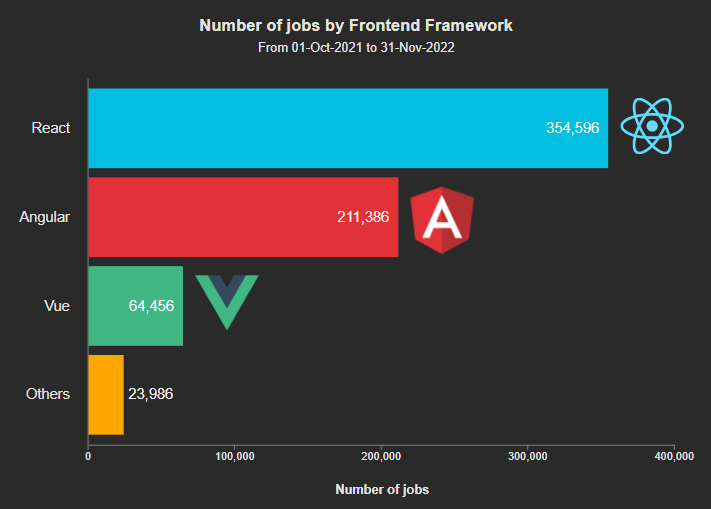
* Composición de componentes. En ReactJS podremos pasar funciones como parámetros, algo también conocido con el nombre de composición funcional.
* Desarrollo declarativo. La primera de las ventajas de React es que es más declarativo, porque contamos con un estado de la aplicación y sus componentes responden ante la variación de ese estado.
* Flujo de datos unidireccional. Ese patrón de funcionamiento que posee React hace que los componentes superiores propaguen los datos a los que están en un orden inferior, estos trabajarán con esos datos y cuando cambian su estado envían los eventos hacia los de orden superior para actualizarse.
* Isomorfismo. También conocido como Javascript Universal, es la capacidad de ejecutar el código tanto en el cliente como el servidor, solucionando problemas de posicionamiento tradicionales.
* Elementos y JSX. El código que embebemos con ReactJS dentro de Javascript, es JSX, aunque tenga apariencia HTML, es decir, utilizamos una sintaxis propia y el resultado son elementos en memoria y no DOM tradicional.

**Frameworks backend más populares en el año 2022** [18]**.**

****

1. Frameworks backend más populares en el año 2022.

**Frameworks frontend más populares en el año 2022** [21]**.**

****

1. Número de empleos disponibles en 2022 para los Frameworks Frontend más utilizados.

### 1.6.6 Entorno de desarrollo integrado (IDE)

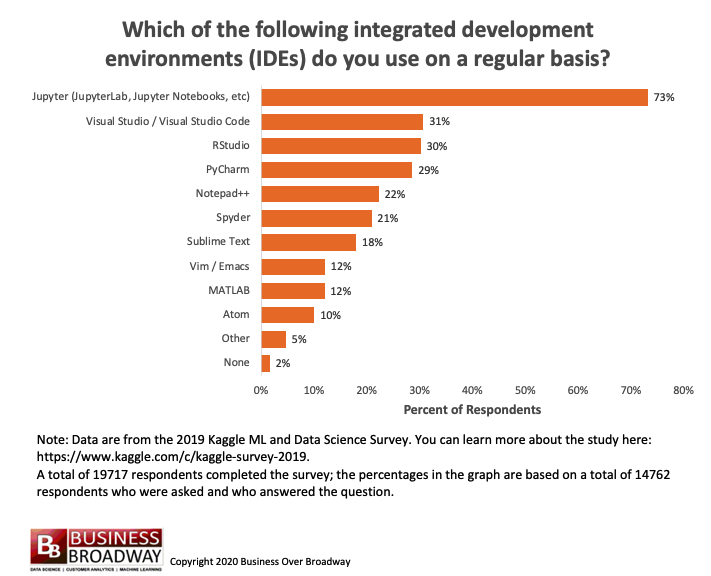
JetBrains PyCharm

PyCharm es un entorno de desarrollo integrado (IDE) de Python que proporciona una amplia gama de herramientas esenciales para los desarrolladores de este lenguaje, estrechamente integradas para crear un entorno conveniente para el desarrollo productivo de ciencia de datos, web y Python. PyCharm permite una mayor productividad debido a los muchos atajos que ofrece, posee herramientas de finalización inteligente de código, autocompletado y autoimportación, comprobaciones PEP8, refactorizaciones, asistencia para pruebas y todo lo que necesita un programador para ahorrar mucho tiempo y realizar un desarrollo productivo de Python.[22]

Visual Studio

IDE de Visual Studio es una plataforma de lanzamiento creativa que puede utilizar para editar, depurar y compilar código y, finalmente, publicar una aplicación. Además del editor y depurador estándar que ofrecen la mayoría de IDE, Visual Studio incluye compiladores, herramientas de completado de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software. Visual Studio es el IDE más rápido para la productividad. Tenga como destino cualquier plataforma o dispositivo. Compile cualquier tipo de aplicación. Trabaje en equipo y en tiempo real. Diagnostique y detenga problemas antes de que ocurran. Posee un gran número de extensiones que ofrecen un mayor dinamismo y adaptación del editor a los diferentes lenguajes y aplicaciones.[23]

**Entornos de desarrollo integrado más utilizados.**[24]

****

1. Entorno de desarrollo integrado (IDE) más usados en el año 2019.

### 1.6.7 Sistemas Gestores de Bases de Datos (SGBD / DBMS)

MySQL

MySQL es un sistema de base de datos relacional simple. Es muy eficiente y fácil de usar, lo que la convierte en una de las tecnologías más reconocibles. Con SQL, puede comprender rápidamente varios conceptos de lenguaje de consulta estructurado (SQL) para crear sistemas de almacenamiento de datos potentes. Está disponible de forma gratuita y es de código abierto, aunque también está disponible bajo varias licencias propietarias. El servidor MySQL es multiproceso, multitarea y está diseñado para funcionar en sistemas de producción de carga pesada. Tiene motores transaccionales y no transaccionales y es uno de los sistemas de bases de datos más fáciles de instalar. MySQL es muy querido entre los usuarios porque es fácil de usar, confiable y rápido.[25]

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de objetos de código abierto. Es totalmente compatible con SQL y fue creado para ser rico en funciones. También es extensible, lo que lo hace útil para cualquier persona que necesite herramientas empresariales. Fue diseñado específicamente para la eficiencia y se puede integrar en casi cualquier software.

PostgreSQL está orientado a objetos, lo que permite ampliar los tipos de datos para crear sus tipos personalizados y es compatible con casi cualquier base de datos. Esta sección detallará su historia, características y casos de uso.

PostgreSQL tiene mucho que ofrecer como sistema de gestión de bases de datos. Se ha ganado su reputación por su solidez de funciones, alta confiabilidad, rendimiento, flexibilidad y facilidad de replicación.[25]

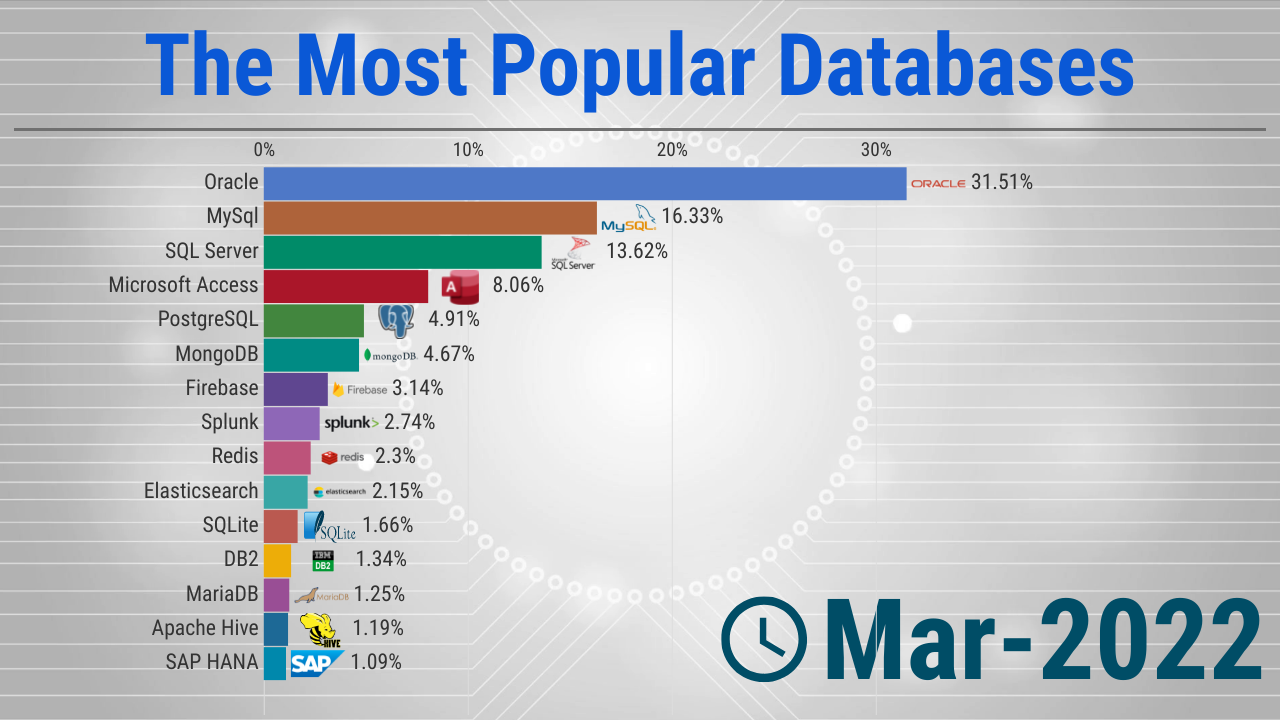
SQLite

SQLite es un sistema de gestión de bases de datos relacionales integrado y sin servidor. Es una biblioteca de código abierto en memoria sin configuración y no requiere ninguna instalación. Además, es muy conveniente ya que tiene un tamaño inferior a 500 kb, lo que es significativamente menor que otros sistemas de gestión de bases de datos.

¿Por qué utilizar SQLite?

* SQLite es un software de código abierto. El software no requiere ninguna licencia después de la instalación.
* SQLite no tiene servidor, ya que no necesita un proceso o sistema de servidor diferente para funcionar.
* SQLite le facilita trabajar en múltiples bases de datos en la misma sesión simultáneamente, lo que lo hace flexible.
* SQLite es un DBMS multiplataforma que puede ejecutarse en todas las plataformas, incluidas macOS, Windows, etc.
* SQLite no requiere ninguna configuración. No necesita configuración ni administración.[26]

**Sistemas Gestores de Bases de Datos más populares.**

****

1. Sistemas Gestores de Bases de Datos más populares en marzo de 2022.

### 1.6.8 Servidor Web

Apache

El servidor Apache es uno de los servidores web gratuitos más potentes del mercado. Fue creado en 1995 por Rob McCool. Según las estadísticas [de W3techs](https://w3techs.com/technologies/details/ws-apache), es utilizado por el**31,**5% de todos los sitios web por los que se conoce el servidor.

Las principales características de Apache son las siguientes[27] :

* Es gratuito y de código abierto.
* Fácil de instalar y configurar.
* Altamente extensible y adaptable con módulos.
* Funciones incorporadas para la autenticación y validación de usuarios.
* Soporte para lenguajes como Perl, PHP y Python.

Algunas de las ventajas que podemos encontrar en un servidor Apache son las siguientes:

* **Soporte:**Es uno de los servidores más utilizados desde hace más de dos décadas, por lo que detrás hay una comunidad muy amplia acompañada de una extensa documentación.
* **Multiplataforma**: Funciona tanto en **Microsoft Windows como en Unix/Linux**y**Mac**, lo que lo convierte en un servicio excelente tanto para un servidor como para un entorno de desarrollo en cualquier estación de trabajo.
* **Características**: Contiene un gran número de módulos oficiales y de terceros que le permiten ampliar la funcionalidad del servidor.
* **Simplicidad**: Es uno de los servidores más fáciles de instalar y configurar, especialmente en las distribuciones de Linux donde forma parte de los repositorios de cada distribución. En el caso de Windows, se puede encontrar empaquetado con el servidor MySQL y PHP en un único instalador que facilita mucho la instalación.
* **Seguridad**: Apache lleva incorporadas funciones para hacer seguro el software con módulos de autorización y autentificación y funciones de control de acceso, así como soporte para la encriptación con **certificados SSL/TLS**.

Las dos desventajas de Apache son:

* El rendimiento puede verse afectado en páginas de alto tráfico, especialmente después de 10.000 conexiones;
* Si los módulos no se utilizan correctamente, se pueden crear agujeros de seguridad.

La única desventaja notable es el bajo rendimiento en comparación con otras alternativas del mercado que son más rápidas y capaces de manejar mucho más tráfico con los mismos recursos.

Nginx

Nginx en simples palabras es un servidor web, proxy inverso y balanceador de carga open source que hoy por hoy soporta a millones de páginas web en todo el mundo.

Su nombre se pronuncia «engine-ex», como motor en inglés, y es hoy por hoy el rival más cercano al clásico servidor Apache.

Surgió en el año 2004 de la mano de su creador Igor Sysoev, y desde ese entonces no paró de crecer. Fue creado originalmente al problema C10K, es decir, que el servidor web lograra despachar 10 mil conexiones concurrentes de forma estable y segura.

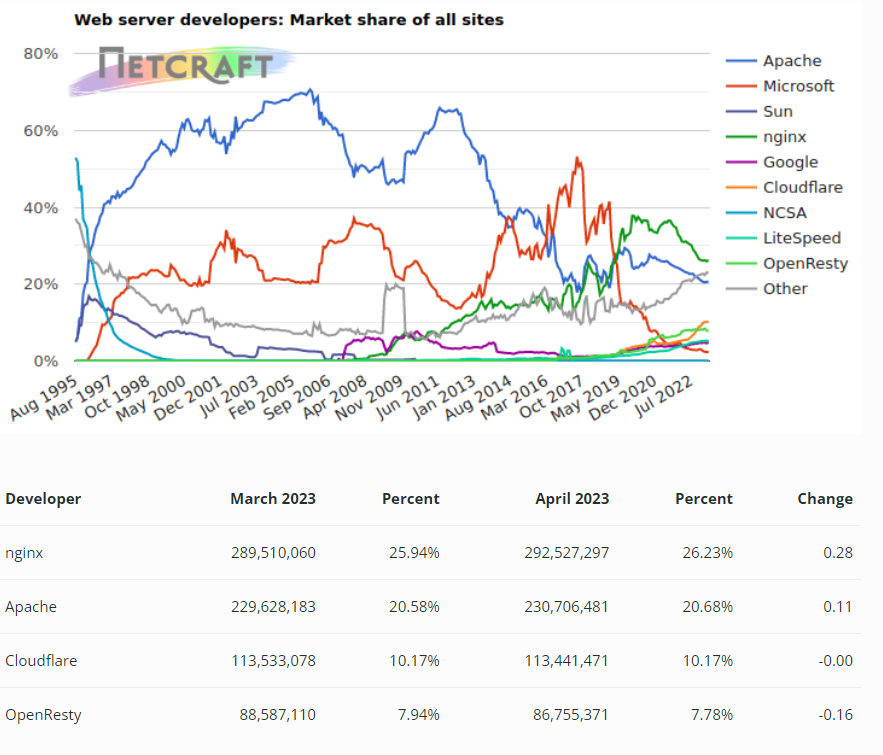
Nginx fue concebido desde un principio para ser un servidor web escalable, rápido y eficiente, por eso hoy por hoy los sitios más populares del mundo lo usan para potenciar sus infraestructuras de servidores HTTP, de mailing, balanceadores de carga, servidores proxy, etc.

Entre sus usuarios más populares se encuentran Yahoo, Netflix, Cloudflare, WordPress.com, Adobe, Apple y muchos más.

Alguna de las características de Nginx[28] :

* Servidor de archivos estáticos y dinámicos
* Sirve como proxy inverso + servidor de cache
* Soporte de autenticación HTTP
* Soporta índices y autoindexado
* Balanceo de carga inteligente
* Tolerante ante fallos
* Ofrece alta disponibilidad
* Escalabilidad asegurada
* Soporte para FastCGI y FastCGI Cache
* Compatible con IPv4 e IPv6
* Soporte HTTP, SPDY, QUICK y HTTP2
* Soporte para HTTPS con certificados SSL
* Permite hosts virtuales, así como basados en IPs
* Streaming nativo con módulo RTMP
* Streaming de MP4 y FLV
* Acepta compresión Gzip y Brotli

Comparativa de los servidores web más utilizados, tomada del sitio netcraft.com:



1. Comparativa entre los servidores web.

### 1.6.9 Tecnologías seleccionadas

Todas las herramientas y tecnologías expuestas con anterioridad son de probada calidad, utilizadas mundialmente por empresas de primer nivel. Se hace difícil en la actualidad, con el elevado nivel de competitividad imperante, determinar las ventajas de unas sobre otras, quedando a elección del desarrollador, la utilización de una u otra. En cualquier caso, habrá un gran número de posibilidades correctas para la elección. Luego de describir cada una de ellas, de exponer sus características y teniendo en cuenta la política de desarrollo de aplicaciones de la Agencia Cubana del Libro, basada en el empleo de herramientas de software libre, se decidió utilizar para la construcción del proyecto las siguientes herramientas:

* Proceso Unificado de Rational (RUP) como metodología de desarrollo de software.
* Visual Paradigm v17.0 como herramienta CASE.
* Python v3.12 como lenguaje de programación.
* Apache2 como servidor Web.
* SQLite 3.45.3 como sistema de gestión de bases de datos durante el desarrollo de la aplicación y PostgreSQL v17.0 para la etapa de producción.
* JetBrains PyCharm2021 como IDE de desarrollo.
* Django v5.0 como framework backend de desarrollo.

## Análisis crítico de las fuentes y bibliografías utilizadas (Estado del arte).

Para la confección del presente documento, fueron consultadas diversas fuentes bibliográficas, que sustentan los aspectos tratados, y enriquecen el nivel técnico y científico de la investigación. Gran parte de estas referencias corresponden a libros y artículos virtuales de sitios confiables en Internet. La bibliografía es actualizada, pues la mayoría de los títulos y referencias consultadas han sido publicados en los últimos cinco años. La selección y revisión bibliográfica ha sido muy rigurosa para garantizar su veracidad. Las fuentes consultadas son de un alto nivel científico-técnico en el campo de la informática.

## Conclusiones

El análisis crítico de la forma de trabajo en el Grupo de Gestión de la Agencia Cubana del Libro en relación a los procesos de asignación de ISMN y gestión de solicitudes, permitió identificar los principales procesos objetos de automatización y las ventajas que supondría para el centro contar con una herramienta informática que facilitara la labor de los especialistas. Fue de gran importancia el estudio realizado sobre las tendencias y tecnologías actuales para poder determinar cuáles eran las más idóneas para el desarrollo del proyecto, tomando siempre en consideración las características de la agencia para la selección de dichas tecnologías. El análisis crítico de las fuentes bibliográficas permitió apoyar la veracidad de la información empleada en la investigación realizada para este trabajo.

# **Capítulo 2. Modelo de negocio.**

## 2.1 Introducción

En este capítulo se hace referencia al estudio de una de las fases más importantes en la confección de un sistema informático, la modelación del negocio, para comprender los procesos de la organización, los cuales se describen mediante los artefactos más importantes: actores y trabajadores del negocio y sus casos de uso, así como los diagramas de actividades de cada uno. Se plasman las reglas del negocio que debe cumplir la aplicación y se genera el modelo de objetos.

## 2.2 Modelo de Negocio Actual

El negocio comienza cuando las editoriales, editores independientes u oficinas de autoría solicitan a la Agencia Cubana la identificación única de sus producciones y músicas notadas o el registro de nuevos editores. Esta solicitud se debe realizar por un agente del cliente que presencialmente deberá llenar una solicitud impresa por un especialista de la Agencia, dicha solicitud contiene datos importantes acerca de la música notada o el nuevo editor que se desea registrar. Una vez realizada la solicitud se le entrega al especialista de la Agencia, que será el encargado de revisar la solicitud, generar, asignar y notificar al cliente en un plazo de 3 días. Para la asignación ISMN el especialista deberá revisar el rango de prefijos asignados a Cuba por la Agencia Internacional del ISMN, corroborar la validez y evitar la duplicidad del ISMN generado. Si la respuesta del especialista es positiva entonces procede a almacenar en una tabla de Excel todos los datos de la solicitud junto al ISMN asignado, esta tabla corresponde solo a la editorial o editor de la música notada que se está registrando. Si la respuesta es negativa se le envía por correo la solicitud al cliente con los errores señalados y un documento de requisitos, se repite el proceso hasta que sea correcta para un nuevo registro.

## 2.3 Reglas del negocio

Las reglas de negocio son restricciones acerca de cómo deben operar los procesos de una entidad. Estas pueden ser leyes y regulaciones impuestas, pero también expresan la arquitectura y estilo del negocio seleccionado.[7]

A continuación, se presentan las reglas establecidas durante las solicitudes de registro y asignación:

* Las solicitudes de ISMN deben realizarse por editoriales o editores asociados que estén registrados previamente en la Agencia Cubana del Libro. Debe contener firma y cita previa con el agente.
* En la confección del ISMN y la revisión de la solicitud debe estar presente un especialista de la Agencia que esté capacitado en la información de la música notada.
* El especialista de la Agencia Cubana debe enviar anualmente a la Agencia Internacional del ISMN las actualizaciones respecto al número de las músicas notadas publicadas en el país.
* Se debe pedir a las editoriales y editores independientes sus direcciones y metadatos relacionados con sus publicaciones para que el especialista sepa siempre como comunicarse con el editor.
* Cada publicación quedará archivada en la Agencia Cubana del Libro para posteriores consultas.
* Las Solicitudes enviadas a la Agencia Cubana del Libro no debe exceder un plazo de respuesta de tres días.
* Las Solicitudes rechazadas no deben volver a ser recibidas hasta que no se cumpla un plazo de 3 días.
* Las Solicitudes rechazadas en más de tres ocasiones no deben volver a ser consideradas por la Agencia.
* Los especialistas podrán asociar a nuevos editores siempre y cuando presenten la aprobación de sus editoriales o el Instituto de la Música.
* Los ISMN no son reutilizables, ni se pueden repetir entre diferentes publicaciones, debe ser único.
* Las Solicitudes ISMN o de Inscripción no deben repetirse, ni tener fallos ortográficos, archivos que no contengan la información requerida, esto es motivo de rechazo por parte de la agencia.

## 2.4 Actores del negocio

Un candidato a actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio y se beneficia de él.[29]

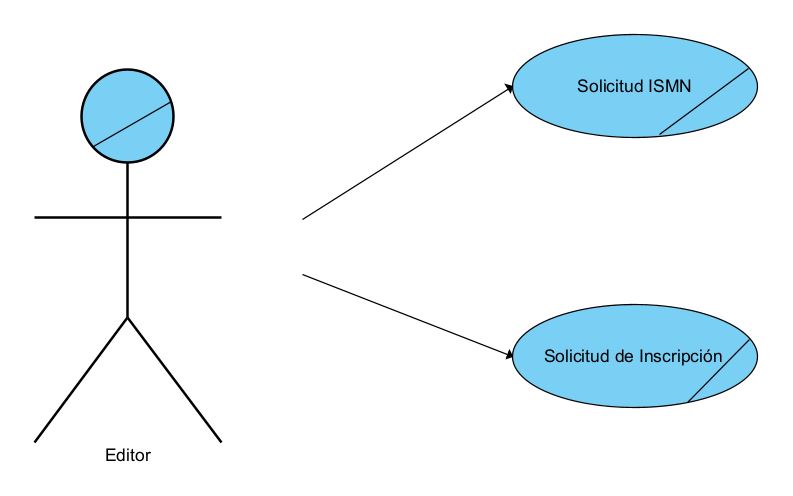
A continuación, quedan definidos los actores del negocio**.**

|  |  |
| --- | --- |
| Nombre de Actor | Descripción |
| Editor | Son quienes reciben los ISMN para identificar de manera única, comercializar, distribuir o ejercer el derecho de autor en sus producciones musicales. Son quienes solicitan la asociación de nuevos editores a la Agencia Cubana del Libro. Pueden ser Editoriales, artistas, editores y organizaciones musicales independientes. |

1. Descripción de los actores del negocio.

## 2.5 Diagrama de casos de uso del negocio

Un caso de uso del negocio define qué debe ocurrir en el negocio cuando aquel se realiza, describe el comportamiento de una sucesión de acciones que produce un resultado de valor para un actor particular del negocio. Es decir, un caso de uso del negocio describe una secuencia de acciones realizadas en el negocio que produce un resultado de valor observable para un actor del negocio.[29]



1. Diagrama de casos de uso.

## 2.6 Trabajadores del negocio

Un trabajador del negocio es una abstracción de un humano o de un sistema de software que representa un rol dentro del desarrollo de una realización de un caso de uso del negocio.[[1]](#footnote-1)

|  |  |
| --- | --- |
| **Nombre del trabajador** | **Descripción** |
| Especialista de la Agencia | Es quien recibe la solicitud de ISMN o de inscripción. Se encarga de revisar las solicitudes de los editores, elaborar y asignar los ISMN, almacenar solicitudes y nuevas publicaciones que se van generando en sus tablas correspondientes. |

1. Descripción de los trabajadores del negocio

## 2.7 Casos de uso del negocio

Los casos de uso del negocio definen la ejecución de una secuencia de acciones que producen un resultado de valor para un actor del negocio en particular.[[2]](#footnote-2)

### 2.7.1 Caso de Uso del Negocio “Solicitar ISMN”

|  |  |
| --- | --- |
| **Nombre del Caso de Uso del Negocio:** | **Solicitar ISMN** |
| **Propósito**:  Confeccionar, validar, gestionar y asignar los ISMN a las nuevas publicaciones musicales que presentan las editoriales y editores independientes asociados. | |
| **Resumen:**  El caso de uso inicia cuando el especialista de la Agencia recibe la solicitud de ISMN por parte de un Editor asociado. Se procede a la revisión de la solicitud verificando los criterios en el documento de requisitos de la Agencia para su aceptación, en caso de ser correcta se procede a la confección, se selecciona el número prefijo que identifica al Editor solicitante y el número prefijo que identifica a la publicación para la cual se generará el ISMN, se verifica la disponibilidad de espacio en el prefijo del Editor y de existir se procede a la confección y asignación del ISMN y almacenamiento de la publicación musical en las tablas correspondientes, de no existir disponibilidad de espacio en el prefijo del Editor se procede a generar un nuevo Prefijo para el Editor solicitante y luego se sigue el curso anteriormente descrito. En caso de no existir disponibilidad en los prefijos de las publicaciones ni en los prefijos de los Editores se notificaría al solicitante a la espera de los nuevos prefijos para Cuba. Cuando el especialista determina la finalización de los procesos notifica al editor para la entrega de su nueva publicación adjunto a su ISMN correspondiente. En caso que la solicitud presente errores se le envía por correo el documento de requisitos de la Agencia junto a los fallos concretos en su solicitud. | |

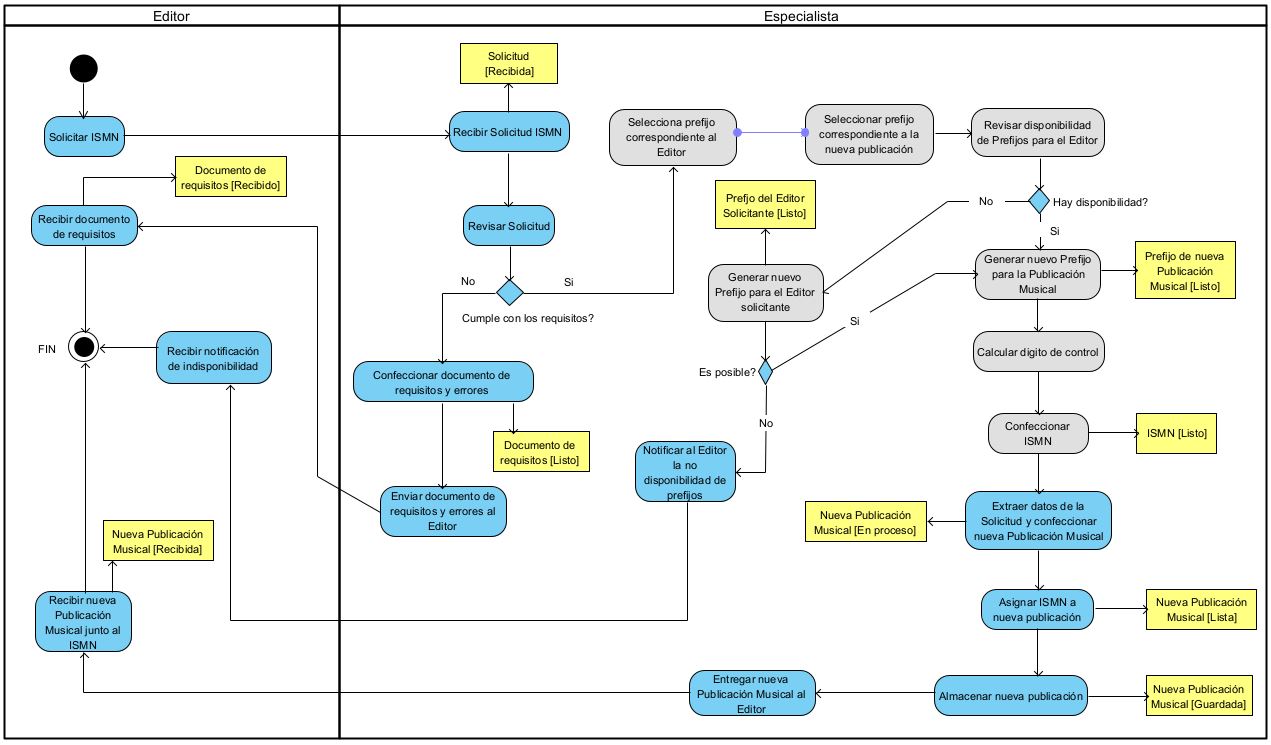
1. Caso de Uso del Negocio "Solicitar ISMN"

### 2.7.2 Caso de Uso del Negocio “Solicitar registro de Inscripción”

|  |  |
| --- | --- |
| **Nombre del Caso de Uso del Negocio:** | **Solicitar registro de Inscripción** |
| **Propósito:**  Insertar y registrar una nueva Editorial o Editor independiente como nuevo asociado de la Agencia Cubana del ISMN. | |
| **Resumen:**  El caso de uso inicia cuando el especialista de la Agencia recibe la solicitud de inscripción por parte de un nuevo editor independiente o una nueva editorial. Se procede a la revisión de la solicitud, verificando los criterios en el documento de requisitos de la Agencia para su aceptación, en caso de ser correcta se selecciona un nuevo prefijo para el Editor solicitante de acuerdo con la información recogida en la solicitud recibida, se revisa la disponibilidad de prefijos para el rango seleccionado y en caso de existir disponibilidad se procede a generar una nueva tabla en Excel con los datos del nuevo Editor y su Prefijo correspondiente. En caso de no existir disponibilidad para asignar un prefijo se le notifica al Editor solicitante en espera de las nuevas asignaciones de prefijos para Cuba. En caso que la solicitud presente errores se le envía por correo el documento de requisitos de la Agencia junto a los fallos concretos en su solicitud. | |

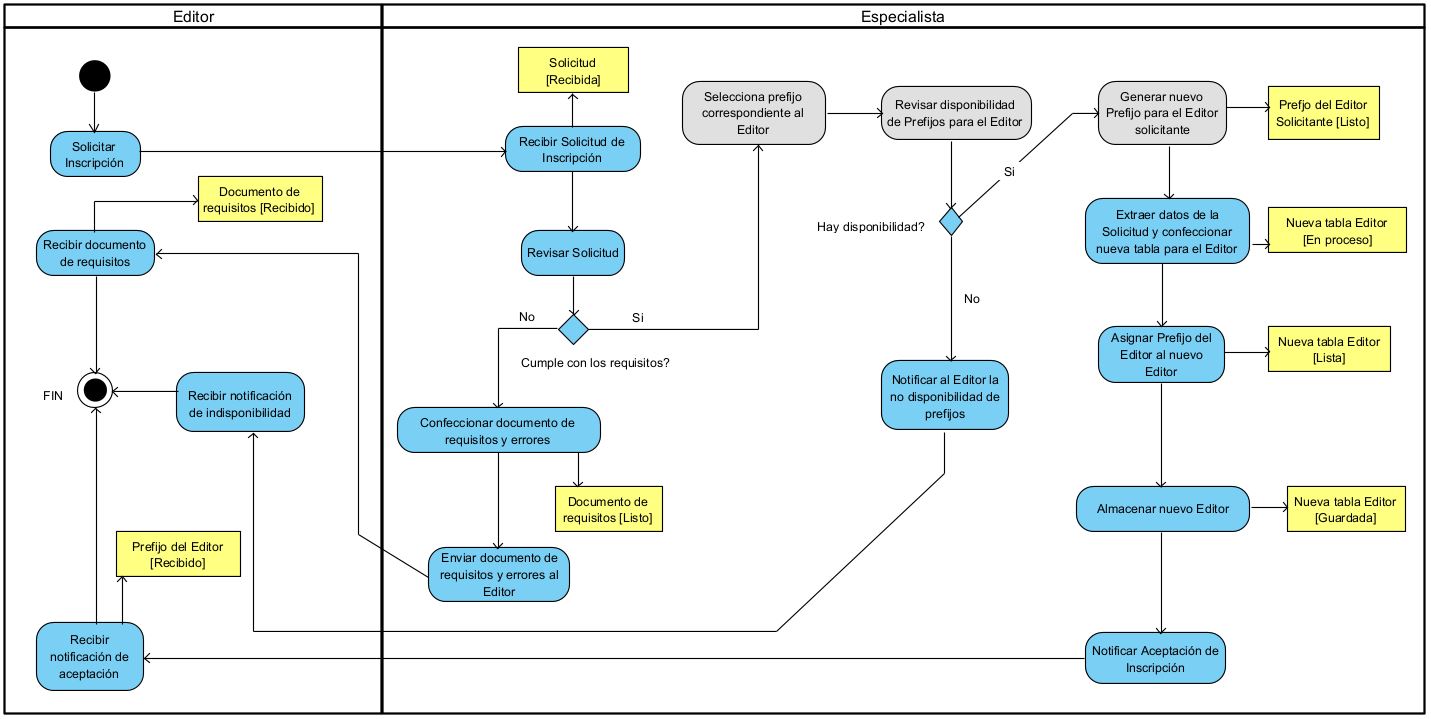
1. Caso de Uso del Negocio "Solicitar registro de nuevo Editor"

### 2.7.3 Diagrama de actividades del caso de uso “Solicitar ISMN”



1. Diagrama de actividades del caso de uso "Solicitar ISMN".

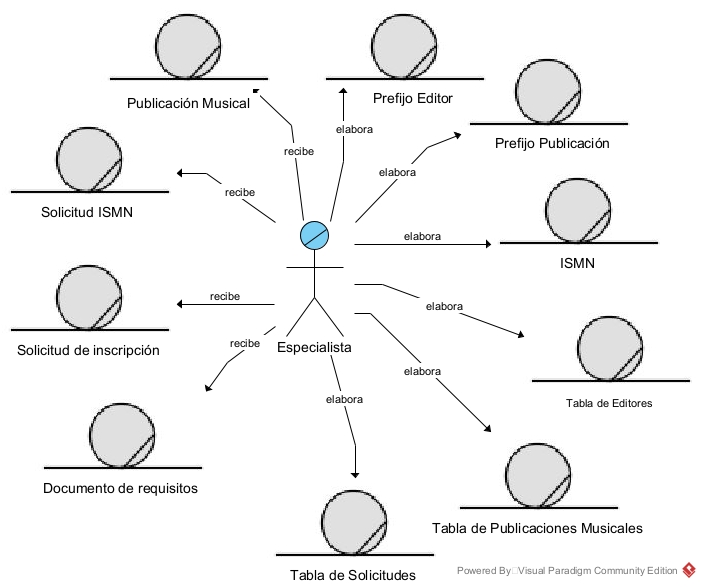
2.7.4 Diagrama de actividades del caso de uso “Solicitar registro de Inscripción”.



1. Diagrama de actividades del caso de uso "Solicitar registro de nuevo Editor.

## 2.8 Modelo de objetos

El modelo de objetos del negocio identifica todos los “roles” y “cosas” en el negocio. ¿Qué cosas se manejan en el negocio?, ¿Quiénes lo hacen?, ¿Cuáles son sus responsabilidades?, ¿Qué es lo que persiste en el tiempo?, son preguntas que se responden con este modelo. Describe las relaciones entre las entidades y los trabajadores del negocio. Además, combina diagramas y descripciones literales que propician un mejor análisis de la información que se maneja y transforma dentro de los procesos.[7]



1. Modelo de objetos.

### 2.8.1 Descripción de las Entidades del Negocio

|  |  |
| --- | --- |
| **Nombre de la Entidad** | **Descripción** |
| Solicitud de ISMN | Documento mediante el cual, las Editoriales o editores independientes reflejan la necesidad de obtener un ISMN. Este documento recoge los datos de la publicación que será representada por el ISMN solicitado, contiene la firma del Editor. |
| Solicitud de inscripción | Documento mediante el cual los editores o editores independientes reflejan la necesidad de asociarse como nuevos editores de la Agencia Cubana. Este documento recoge los datos del nuevo editor independiente o editorial como su nombre, promedio de publicaciones por año y otros datos. |
| ISMN | Es el documento que recoge todo lo relacionado al ISMN que genera el especialista y que fue solicitado por el editor asociado. Contiene el ISMN en cuestión con información de la publicación musical que representa, el Editor que publica, un código de verificación y otro que difiere de otros estándares de codificación. |
| Documento de requisitos | Es elaborado por el especialista de la Agencia Cubana cuando las solicitudes presentadas por las editoriales o editores independientes presentan algún problema. Este documento es enviado a los editores, contiene una explicación por parte del especialista para la rectificación de la solicitud o motivos del rechazo. |
| Publicación Musical | Se trata de las publicaciones de música notada que se almacenan en la Cámara Cubana del Libro, puede ser una partitura, antología, libros de canciones o cualquier otra obra musical que presente el editor para ser registrada mediante el ISMN. |
| Prefijo del Editor | Es el documento que recoge todo lo relacionado al número prefijo que representa a los Editores. Contiene el número de prefijo, el rango y los datos del Editor al que pertenece y el número de lote. |
| Prefijo Publicación | Es el documento que recoge todo lo relacionado al número prefijo que representa a las diferentes Publicaciones Musicales almacenadas en la Cámara Cubana del Libro. Contiene el número de prefijo, el rango y los datos de la Publicación a la que pertenece y el número de lote. |
| Tabla de Editores | Es la tabla Excel que contiene la lista de todos los Editores inscriptos en la Cámara Cubana del Libro. Contiene datos extraídos de las solicitudes de inscripción de los diferentes editores, sus prefijos asociados, fecha de inscripción o de baja, entre otros datos. |
| Tabla de Publicaciones Musicales | Es la tabla Excel que contiene la lista de todas las Publicaciones Musicales almacenadas en la Cámara Cubana del Libro. Contiene datos extraídos de las solicitudes ISMN enviadas por los diferentes Editores, sus ISMN asignados, fecha de realización, rango de prefijo, entre otros datos. |

1. Descripción de las Entidades del Negocio

## 2.9 Conclusiones

Después de realizado el capítulo se logró una mayor comprensión de la estructura y funcionamiento de la entidad, específicamente de la actividad de análisis de información. Como resultado de este estudio se determinaron y describieron los actores, trabajadores y casos de uso del negocio. Se ilustró la interacción entre actores y casos de uso a través del diagrama de casos de uso del negocio y los diagramas de actividades correspondientes. La relación entre trabajadores y entidades del negocio se reflejó mediante la representación gráfica del modelo de objetos del negocio.

# 

# **Capítulo 3 Requisitos**

## 3.1 Introducción

En este capítulo se describen los actores, casos de uso del sistema y los requisitos funcionales y no funcionales, cuyo propósito es guiar el desarrollo hacia el sistema correcto. Se muestran los principales paquetes de la estructura del producto de software a obtener y sus relaciones, con la intención de alcanzar una visión general de la arquitectura definida.

## 3.2 Definición de los requisitos funcionales

**Seguridad**

* Gestionar información de los usuarios.
* Autenticar usuarios.
* Gestionar autenticación de correo.
* Mostrar información de usuario conectado.
* Modificar contraseña de usuario.
* Restablecer contraseña de usuario.
* Realizar salvas y restauración de la base de datos.
* Visualizar las trazas de las operaciones de los usuarios en el sistema.
* Gestionar Grupos de Autenticación y Autorización.
* Gestionar Asignación de usuarios a Grupos de Autorización.

**Procesos de la Agencia**

* Gestionar rangos y prefijos de los Editores.
* Gestionar rangos y prefijos de las Publicaciones Musicales.
* Registrar Editoriales y Editores independientes.
* Asignar prefijos.
* Gestionar Editoriales y Editores independientes.
* Registrar Publicaciones Musicales.
* Gestionar Publicaciones Musicales.
  + Asignar prefijos.
* Solicitar ISMN.
* Gestionar Solicitudes de ISMN.
  + Solicitudes en Proceso.
  + Solicitudes Aprobadas.
  + Solicitudes Pendientes.
  + Solicitudes Rechazadas.
* Gestionar Tipo de publicación.
* Gestionar Tipo de obra.
* Crear nuevo autor/colaborador.
* Seleccionar portada de catálogo.
* Gestionar Reseña.
* Gestionar tamaño de código de barra.
* Gestionar formato de salida de código de barra.
* Asignar ISMN.
* Reasignar rangos de prefijo.
* Agregar rango de prefijo.
* Listar Editores.
* Listar Publicaciones Musicales.
* Listar Solicitudes.
* Gestionar Catálogo.

**Gestionar Nomencladores**

* Gestionar nomencladores del Editor.
  + Gestionar tipo de Editor
  + Gestionar rango de prefijo (rango superior, rango medio, rango medio-inferior)
* Gestionar nomencladores de Publicación Musical.
  + Gestionar género musical.
  + Gestionar idioma.
  + Gestionar tipo (independiente, volumen, partitura, obra completa).
  + Gestionar descripción física.
  + Gestionar rango de prefijo (rango superior, rango medio, rango medio-inferior)
* Reportes en formato PDF (Todos los reportes en PDF y Gráficos, brindarán a los usuarios la información que se describe a continuación por rangos de fechas según la necesidad que cada caso requiera, y se podrá imprimir o guardar).
* Listado de Editores independientes y Editoriales registradas en la Agencia.
* Publicación Musical independiente.
* Reporte anual de listado de Publicaciones Musicales e ISMN asignados, solicitado por la Agencia Internacional.

**Realizar Búsqueda general.**

**Organizar tabla de Editores, Publicaciones y Solicitudes.**

**Sistema de Alertas o Notificaciones.**

* El sistema deberá emitir alertas automáticas, por ejemplo, cuando el rango de prefijos de una editorial esté cerca de agotarse, cuando existan solicitudes pendientes, la entrada de datos sea incorrecta en los formularios, entre otras situaciones que requieran la notificación del sistema al especialista de la Agencia.

**Generar Reportes.**

Condiciones:

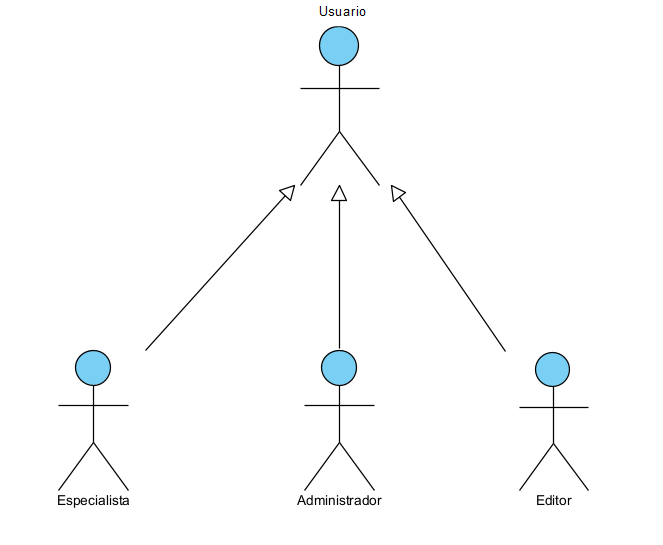
* Se debe poder filtrar los reportes de las publicaciones por su fecha de inserción, título, autor, editor, género y otras opciones disponibles de exportación.
* Se debe poder filtrar los reportes de las solicitudes por su fecha de inserción, nombre del editor solicitante, tipo (ISMN, Inscripción), estado y otras opciones de exportación.
* Se debe poder filtrar los reportes de los Editores por su tipo (Editorial, Independiente, Ambos), fecha de inserción, nombre, prefijo, estado y otras opciones de exportación.
* Se deben poder ordenar los valores de las tablas, ascendente o descendentemente, al hacer *click* en alguna de sus columnas.
* Se debe poder exportar la información de las tablas y gráficos a diversos formatos PDF y también a formato JSON para conectar con otros sistemas.
* Generar reportes gráficos
* Editores independientes o Editoriales con mayor cantidad de publicaciones.
* Editores independientes o Editoriales con mayor actividad en el último año.
* Porciento de solicitudes rechazadas.
* Solicitudes recibidas en el último mes.
* Solicitudes aceptadas en el último mes.
* Notificar estado de las solicitudes.
* Notificar indisponibilidad de prefijo.

## 3.3 Actores del sistema a automatizar

Los actores representan los elementos externos al sistema que colaboran con él y por lo general suelen corresponderse con trabajadores del negocio.[[3]](#footnote-3)

|  |  |
| --- | --- |
| **Nombre del Actor del sistema** | **Descripción** |
| Usuario | Es el actor genérico que representa al Editor (Cliente), al Especialista y al Administrador. Se autentica en la aplicación y accede a las funcionalidades de cambiar contraseña, ver su información de perfil de usuario, restablecer contraseña, autenticarse y cerrar sesión. Puede realizar búsqueda generales en la colección musical que ofrece la Agencia Cubana del Libro, visualizar publicaciones musicales y sus ISMN correspondientes. |
| Administrador | Tiene todos los permisos en el sistema. Gestiona la información de los usuarios y las funcionalidades de la aplicación. Accede al registro de las actividades de los usuarios (trazas). Realiza las salvas y restaura de la Base de Datos, gestiona los grupos de autorización y los rangos de prefijos asignados al país. |
| Especialista | Es el encargado de la introducción de datos al sistema. Tiene el permiso de crear, listar y mostrar ISMN, editores y estadísticas de las solicitudes. Solo podrá eliminar aquellos objetos que él mismo haya creado. También puede generar alertas y notificaciones a los editores (clientes). Este actor hereda las funcionalidades y los requerimientos del actor Usuario. |
| Editor | Este actor representa a las Editoriales o editores independientes que acceden al sistema principalmente para realizar las solicitudes de ISMN o solicitudes de inscripción. Heredan las funcionalidades y los requerimientos del actor Usuario y una vez autenticados en el sistema pueden acceder a su lista personal de producciones musicales e ISMN registrados en la Agencia. También pueden comunicarse con el especialista a través de envió de preguntas, acceder a catálogos de la biblioteca musical, colecciones digitales y recibir notificaciones del sistema. |
| 1. Actores del sistema a automatizar | |

### 3.4 Jerarquía de actores del sistema

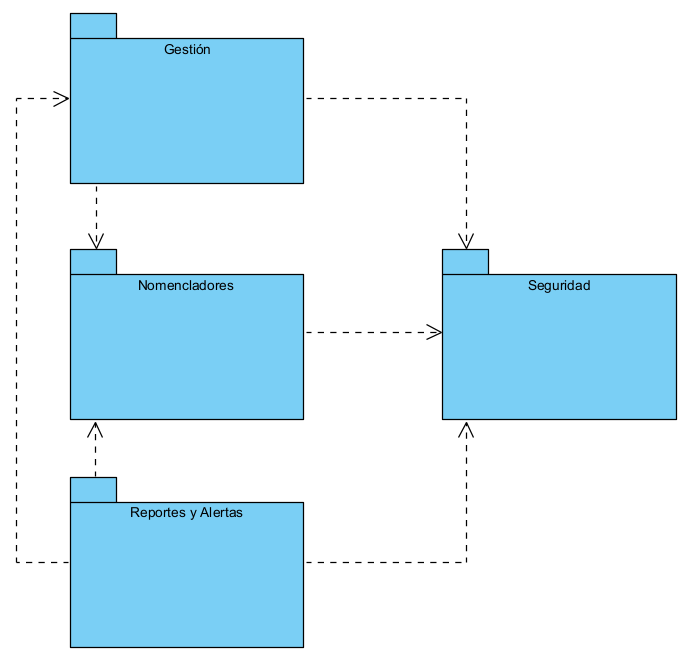


1. Jerarquía de actores del sistema.

## 3.5 Paquetes y sus relaciones

Dividiremos el sistema en cuatro grandes grupos como un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí.[[4]](#footnote-4)

### 3.5.1 Diagrama de Paquetes del Sistema



1. Diagrama de Paquetes del Sistema.

**Seguridad:** Se encuentran las funcionalidades necesarias para el acceso de los usuarios al sistema. Este paquete hace uso del componente de seguridad de Django cuyo funcionamiento se basa en identificar primero al usuario (autenticación) y en comprobar después si ese usuario tiene acceso al recurso solicitado (autorización), se usarán los tokens de seguridad para limitar los intentos de autorización. La autenticación, que siempre se aplica en primer lugar, se basa en el uso de un cortafuego (firewall, en inglés) cuyo trabajo consiste en determinar si el usuario necesita estar autenticado, y si lo necesita, enviar una respuesta al usuario para iniciar el proceso de autenticación. Una vez que un usuario se autentica, la etapa de autorización determina si el usuario tiene acceso al recurso que solicita. Además, contiene las funcionalidades para el cambio de contraseña, restablecer contraseña, salva y restaura de la base de datos y visualización de trazas.

**Reportes y Alertas:** Contiene todas las informaciones de reportes estadísticos relacionados con las producciones que se insertan al sistema, así como los reportes de estado de los ISMN en proceso de entrega, notificaciones al usuario, muestra de información y alertas que muestra el sistema.

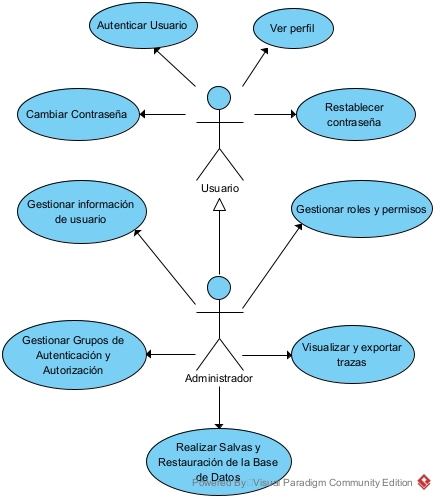
**Nomencladores:** Contiene todas las funcionalidades que permiten la gestión de todos los nomencladores del sistema, ya que éstos son características importantes para describir a las producciones musicales, solicitudes y editores asociados.

**Gestión:** Este paquete agrupa las funcionalidades que permiten realizar toda la gestión de los ISMN, las solicitudes, las publicaciones musicales y los editores en la Agencia Cubana. Cuenta con posibilidades de filtros, generación de código de barras, opciones de exportarción, entre otras opciones.

## 3.6 Diagrama de casos de uso del sistema a automatizar

El diagrama de casos de uso del sistema es un artefacto gráfico que documenta el comportamiento de un sistema desde el punto de vista del usuario. En él se representan los actores, los casos de uso del sistema y las relaciones que existen entre actores y casos de uso.[[5]](#footnote-5)

3.6.1 Diagrama de Casos de Uso del Sistema del paquete “Seguridad”



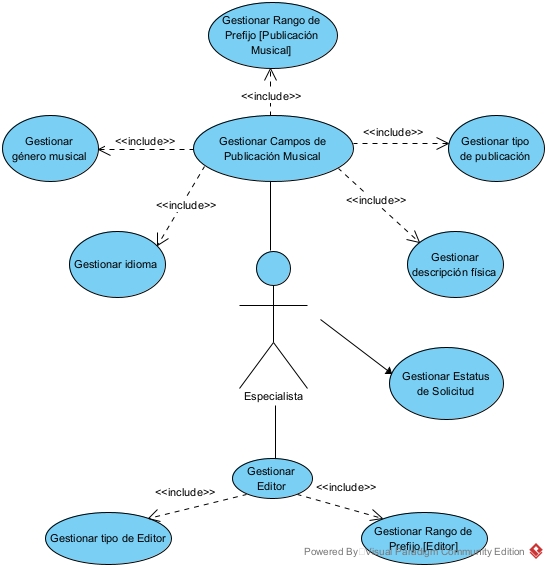
1. Diagrama de Casos de Uso del Sistema del paquete "Seguridad".

### 3.6.2 Diagrama de Casos de Uso del Sistema del paquete “Reportes y Alertas”



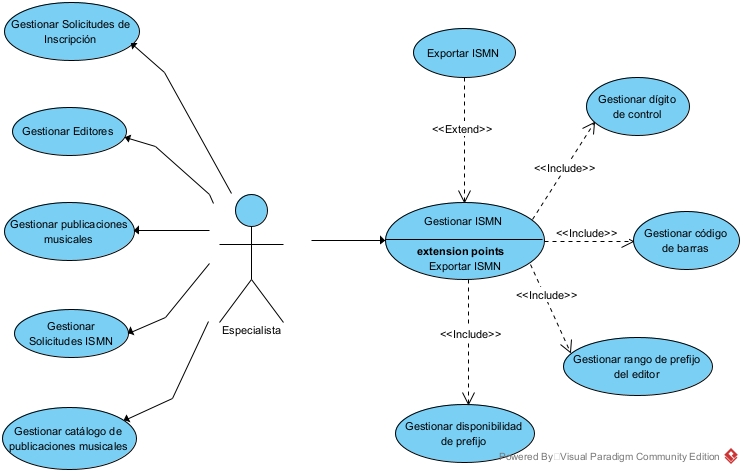
1. Diagrama de Casos de Uso del Sistema del paquete "Reportes y Alertas".

### 3.6.3 Diagrama de Casos de Uso del Sistema del paquete “Nomencladores”



1. Diagrama de Casos de Uso del Sistema del paquete "Nomencladores".

### 3.6.4 Diagrama de Casos de Uso del Sistema del paquete “Gestión”



1. Diagrama de Casos de Uso del Sistema del paquete "Gestión"

## 3.7 Descripción de los casos de uso

A continuación se presentan algunos de los casos de usos más utilizados en el sistema:

|  |  |
| --- | --- |
| **Nombre del Caso de Uso** | **Gestionar ISMN** |
| Actor | Especialista |
| Propósito | Permite la creación, asignación, generación de códigos de barra y visualización de los ISMN procesados. |
| Resumen | El caso de uso inicia cuando el actor termina la revisión de una Solicitud ISMN, cumple con los requisitos de la Agencia y decide aceptar la solicitud. Se procede a revisar la disponibilidad del prefijo del Editor, determinar el valor de prefijo que debe tener la nueva publicación musical teniendo en cuenta el rango de prefijo del Editor y el número de publicaciones asociadas hasta el momento, luego se gestiona el dígito de control y finalmente se gestiona la asignación y almacenamiento del ISMN. En caso de no existir disponibilidad en el prefijo del Editor se notifica al Especialista de la Agencia. |
| Precondiciones | El usuario autenticado debe pertenecer al grupo de Especialistas. |
| Postcondiciones | Se notifica al correo del Editor que se ha terminado la gestión del ISMN de su publicación musical, se actualizan las tablas y las trazas luego de las acciones del usuario. |

1. Descripción del caso de uso “Gestionar ISMN”.

|  |  |
| --- | --- |
| **Nombre del Caso de Uso** | **Ver Colección de Publicaciones Musicales.** |
| Actor | Usuario |
| Propósito | Permite visualizar todas las publicaciones insertadas en el sistema y que están registradas en la Agencia. |
| Resumen | El caso de uso inicia cuando el actor accede a la sección “Colecciones Musicales”. Aquí, el actor podrá consultar las publicaciones musicales, títulos, ISMN, autores y editores asociados a cada publicación musical y sus códigos de barra. Podrá insertar términos de búsqueda para filtrar el catálogo. El caso de uso termina cuando el usuario abandona la sección. |
| Precondiciones | No requiere estar autenticado en la aplicación. |
| Postcondiciones | Ninguna. |

1. Descripción del caso de uso "Visualizar registro de Editores"

|  |  |
| --- | --- |
| **Nombre del Caso de Uso** | **Gestionar Editores** |
| Actor | Especialista |
| Propósito | Permite la inserción, modificación, eliminación y visualización de los editores en la aplicación. |
| Resumen | El caso de uso inicia cuando el actor realiza alguna de las acciones, antes mencionadas en el propósito, sobre la lista de editores de la aplicación. Cuando efectúa la operación deseada y esta se registra en la base de datos, se da por terminado el caso de uso. |
| Precondiciones | El usuario autenticado debe pertenecer al grupo de Especialistas. |
| Postcondiciones | Se actualizan las trazas luego de las acciones del usuario. |

1. Descripción del caso de uso "Gestionar Editores"

|  |  |
| --- | --- |
| **Nombre del Caso de Uso** | **Enviar Solicitud de ISMN** |
| Actor | Editor |
| Propósito | Permite al Editor enviar la solicitud de ISMN al Especialista de la Agencia. |
| Resumen | El caso de uso inicia cuando el actor accede a la sección “Solicitud ISMN”. Aquí, el actor podrá llenar un formulario con los datos de la publicación musical para la cual desea el ISMN y hacer la solicitud a la Agencia. El caso de uso termina cuando el usuario envía la solicitud o abandona la página. |
| Precondiciones | El usuario autenticado debe pertenecer al grupo de Editores. |
| Postcondiciones | Se notifica al Especialista la existencia de una nueva solicitud. Se actualizan las trazas luego de las acciones del usuario. |

1. Descripción del caso de uso "Enviar Solicitud de ISMN"

|  |  |
| --- | --- |
| **Nombre del Caso de Uso** | **Exportar Publicación Musical** |
| Actor | Editor, Especialista |
| Propósito | Permite realizar un reporte PDF con los datos de una publicación junto su ISMN y código de barras correspondiente. |
| Resumen | El caso de uso inicia cuando el actor selecciona la opción de ´Exportar´ dentro de la visualización de una publicación musical almacenada en el sistema. Se descargará a su navegador un documento .pdf con, el título, nombre del Editor, autor, link de portada, código de barras y demás datos de la publicación musical. El caso de uso termina cuando el actor abandona esta sección. |
| Precondiciones | El usuario autenticado debe pertenecer al Grupo de Editores o de Especialistas. |
| Postcondiciones | Se actualizan las trazas luego de las acciones del usuario. |

1. Descripción del caso de uso "Realizar búsqueda general"

## 3.8 Definición de los requisitos no funcionales.

**RNF1. Apariencia o interfaz externa:** El entorno de la aplicación debe ser amigable y de fácil navegación. Se utilizarán colores y logos que identifican a la Agencia Cubana del ISMN y a la Cámara Cubana del Libro según sus identidades corporativas. Contará con menús desplegables y botones de acción directa para acortar los pasos a seguir cuando se quiere alcanzar un resultado determinado.

**RNF2. Usabilidad:** Cualquier tipo de usuario podrá navegar en la aplicación, ya que los enlaces sugerirán la información que se maneja en cada página y las acciones de los formularios serán claras. El vocabulario empleado será acorde son la actividad que realizan los usuarios.

**RNF3. Escalabilidad:** El nuevo modelo de datos, los módulos a implementar y las herramientas a utilizar no deben comprometer futuros desarrollos. El sistema debe tener la capacidad de agregar nuevas funcionalidades o interactuar con otros modelos de datos sin que ello requiera un rediseño completo.

**RNF4. Rendimiento:** El tiempo de respuesta no debe ser mayor de diez segundos. Los analistas podrán trabajar concurrentemente sin el peligro de que colapse el servidor. Para ello debe disponerse de los requerimientos mínimos de hardware, realizar una gestión adecuada de la caché y optimizar las peticiones al servidor para que consuman la menor cantidad de recursos.

**RNF5. Soporte:** La aplicación será compatible con los lenguajes HTML5 y CSS3, soportada en los navegadores más utilizados, como Chrome, Firefox y Microsoft Edge. Gracias a los mecanismos de acceso a datos, la aplicación podrá migrar a cualquier Sistema Gestor de Base de Datos. Se realizarán comentarios al código a modo de documentación, con vista a futuras actualizaciones o incorporación de nuevas funcionalidades. Se deberán poder realizar salvas de la base de datos.

**RNF6. Portabilidad:** La aplicación deberá ser multiplataforma, o sea, podrá configurarse tanto en servidores Windows como en servidores Linux. De igual forma será accesible desde cualquier estación de trabajo conectada a la red local de la Agencia, sin importar el SO que tenga instalado, siempre y cuando tenga configurado un navegador web.

**RNF7. Legales:** La aplicación será propiedad legal de la Agencia Cubana del Libro y contará con los permisos legales de derechos del autor otorgados por la Agencia Cubana de Derechos de Autores Musicales (ACDAM).

**RNF8. Confiabilidad:** Con el objetivo de minimizar la ocurrencia de errores, se validarán los formularios de entrada de datos generando alertas a los usuarios cuando sus acciones produzcan algún error o un comportamiento inadecuado de la aplicación.

**RFN9. Seguridad:** Para acceder a la aplicación, el usuario necesitará autenticarse, y según su nivel de acceso, dado por su rol, podrá ejecutar acciones específicas en el sistema. Las contraseñas serán cifradas antes de ser almacenadas. Se registrarán las operaciones realizadas por cada usuario mediante las trazas que proporcionará la aplicación. Se pedirá confirmación cada vez que se quiera eliminar alguna información, evitando así accidentes por equivocaciones del usuario. Si el usuario no tiene el privilegio de ejecutar alguna acción (insert, update, delete, etc) se le notificará en el acto.

**RNF10. Software:** Se recomienda para el despliegue de la aplicación un servidor con sistema operativo Linux e instalados Apache v2.4 o superior, PostgreSQL v9.3 o superior y Django v3.0 o superior. Para acceder a la aplicación, el usuario deberá contar con alguno de estos navegadores web, al menos en estas versiones: Mozilla Firefox v36, Google Chromev37, Opera v23 o Internet Explorer v11.

**RNF11. Hardware:** Se recomienda para las estaciones de trabajo, memoria RAM de 1 GB, microprocesador Pentium IV a 2.2 GHz o Celeron a 3.0 GHz e interfaces de red a 100 Mbps. Para el servidor de aplicación, se recomienda memoria RAM de 4 GB, microprocesador de 2 núcleos a 3.0 GHz y espacio disponible en disco duro de 10 GB.

## 3.9 Conclusiones

La captura de requisitos facilita comprender la necesidad del cliente y permite el intercambio mutuo para buscar y proponer soluciones, hasta lograr convertirlos en un sistema operacional. La definición de paquetes permite una mayor organización y comprensión del proyecto. Con los diagramas de casos de uso por paquetes quedan bien distribuidos los requisitos funcionales y permite una mejor comprensión de las acciones a realizar por cada actor del sistema. Es importante la descripción de los casos de uso para que se tengan en cuenta el cumplimiento de precondiciones y postcondiciones una vez implementado el sistema.

# **Capítulo 4 Descripción de la solución propuesta.**

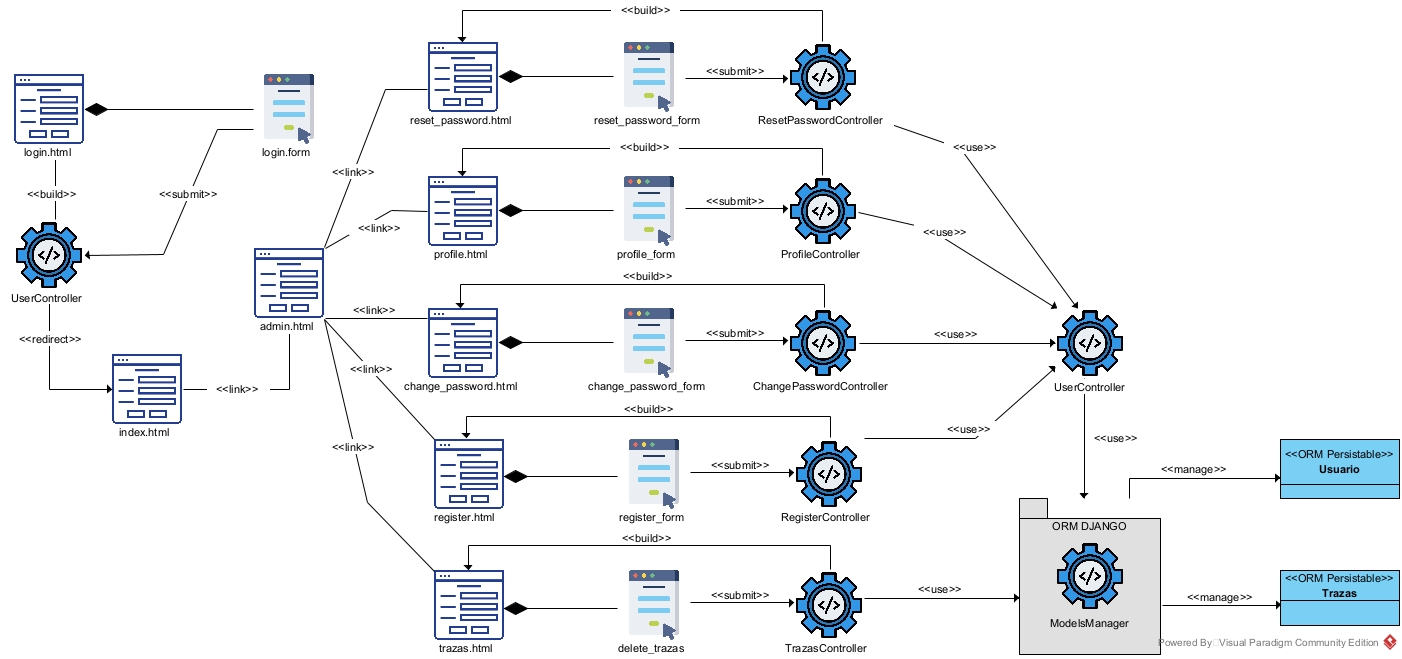
## 4.1 Introducción.

En este capítulo se abordarán los principios y tecnologías utilizadas durante el desarrollo del sistema, para ello se mostrarán las clases de diseño por paquetes. También se muestran elementos generales como el tratamiento de errores, los modelos físico y lógico de la aplicación y se expone la distribución de la aplicación mediante el diagrama de despliegue.

## 4.2 Diagrama de clases del diseño.

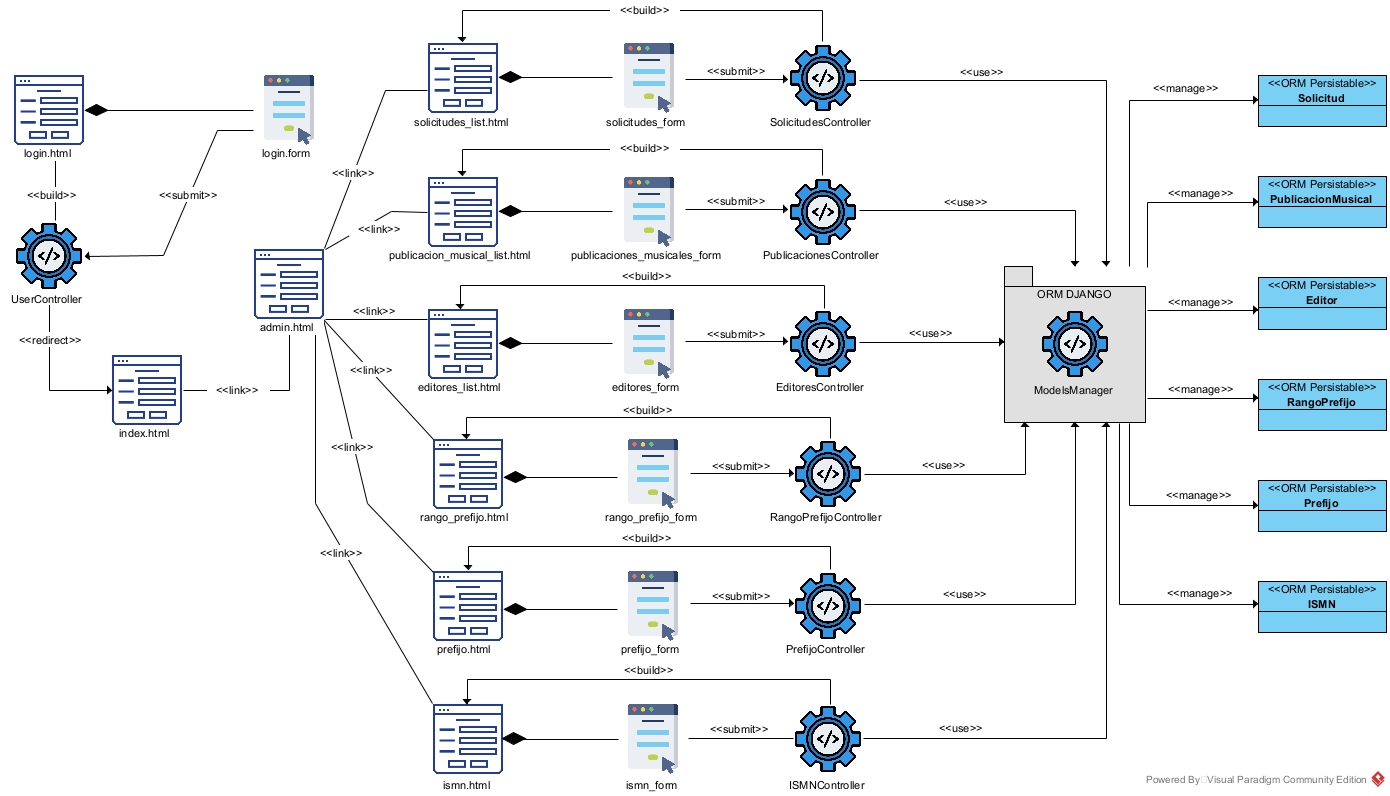
Las clases creadas durante el diseño del sistema ayudan a una mejor comprensión al ser concebidas para satisfacer los requisitos funcionales y no funcionales del sistema desarrollado.

### 4.2.1 Diagrama de diseño. Paquete “Seguridad”



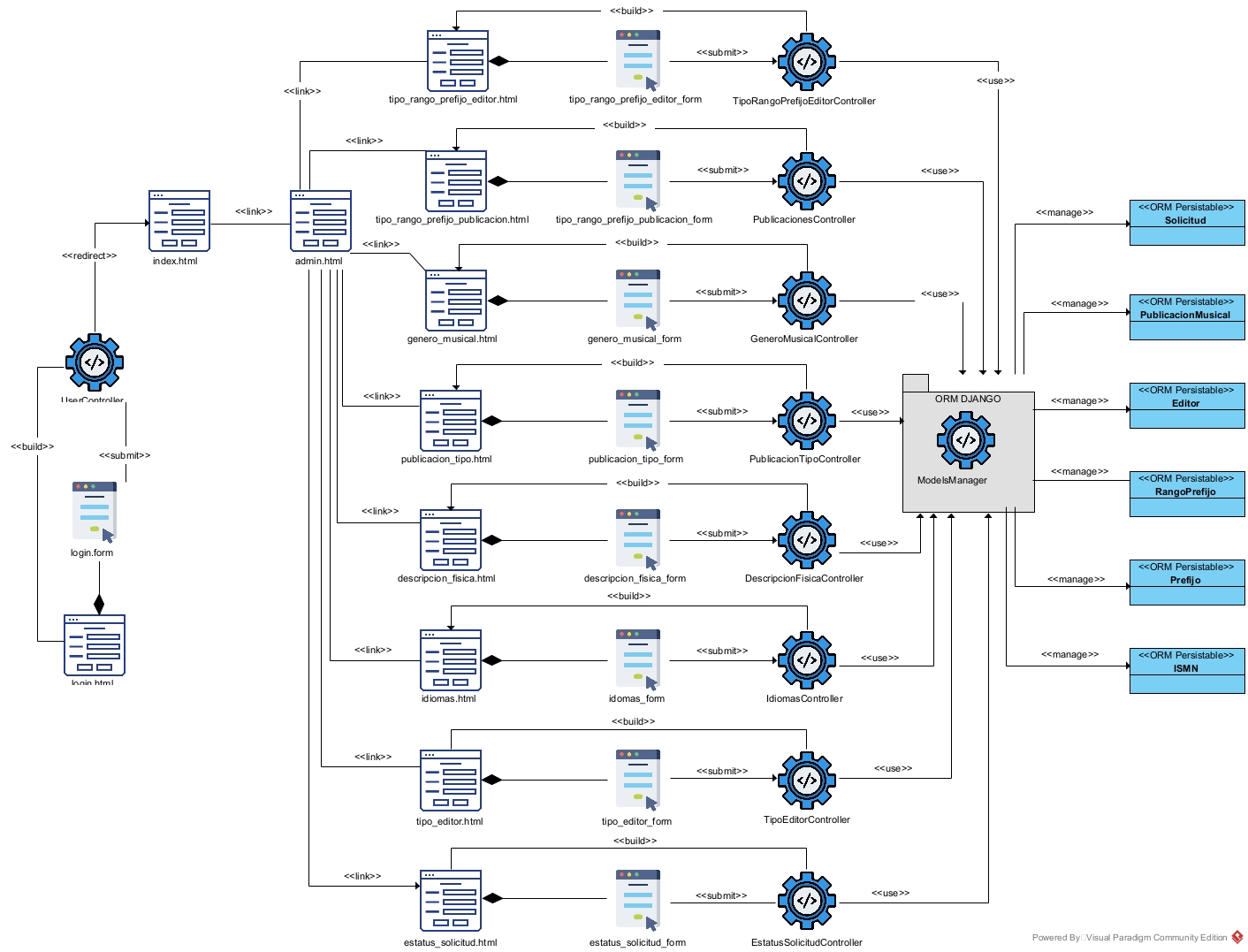
1. Diagrama de diseño. Paquete "Seguridad".

### 4.2.2 Diagrama de diseño. Paquete “Gestión”



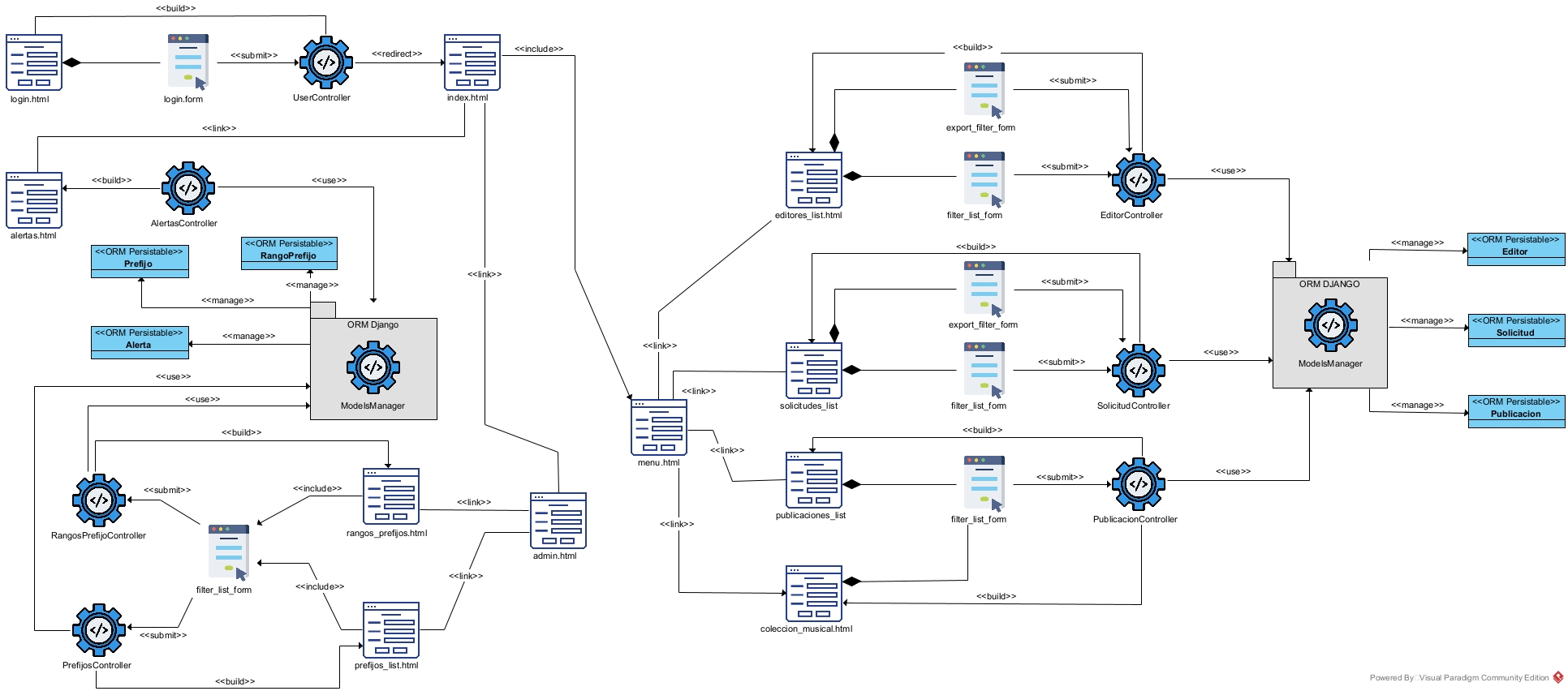
1. Diagrama de diseño. Paquete "Gestión.

### 4.2.3 Diagrama de diseño. Paquete “Nomencladores”



1. Diagrama de diseño. Paquete "Nomencladores".

### 4.2.3 Diagrama de diseño. Paquete “Reportes y Alertas”



1. Diagrama de diseño. Paquete "Reportes y Alertas".

## 4.3 Principios de diseño.

Los patrones de diseño brindan una solución reusable, probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Es la manera de documentar la experiencia acumulada en la estrategia empleada para solucionar un problema. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles.[31]

**Patrones GRASP**: Los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos.[32]

Los patrones GRASP utilizados en la solución propuesta fueron:

**Bajo acoplamiento**: Plantea que se debe poder reutilizar las funcionalidades de las distintas clases, con un nivel de dependencia mínima. Este patrón se evidencia en el sistema, pues cada módulo de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas. Por ejemplo, se puede cambiar el valor “name” de la clase UrlPatterns que representa las rutas URL de las distintas páginas del proyecto sin tener que afectar la implementación subyacente o se puede modificar el HTML de una página sin tener que tocar el código Python que la renderiza en el módulo views.py.

**Alta cohesión**: Asigna responsabilidades de manera tal que la cohesión siga siendo alta, o sea que las funcionalidades de las clases estén altamente relacionadas de forma tal que exista una colaboración entre ellas para compartir el esfuerzo y no caiga todo el peso sobre una única clase. Usar este patrón simplifica el mantenimiento y favorece el bajo acoplamiento. Este patrón se evidencia en la utilización de vistas de Django basadas en clases para poder realizar herencias y una mejor descomposición de los componentes que la integran.

**Creador**: Este patrón es quien guía el proceso de asignación de responsabilidades relacionadas con la creación de objetos. Tiene el objetivo de asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este patrón se pone de manifiesto en la acción de asignación de prefijos donde las clases controladoras “InsertEditor” e “InsertPublicación\_Musical” son encargadas de tomar las entradas del usuario y generar una clase “Prefijo”.

**Patrones GOF:** Patrones publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la banda de los cuatro (del inglés, Gang of Four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además introducir más notación UML. Existen 23 patrones GoF de los cuales 15 se utilizan con frecuencia.

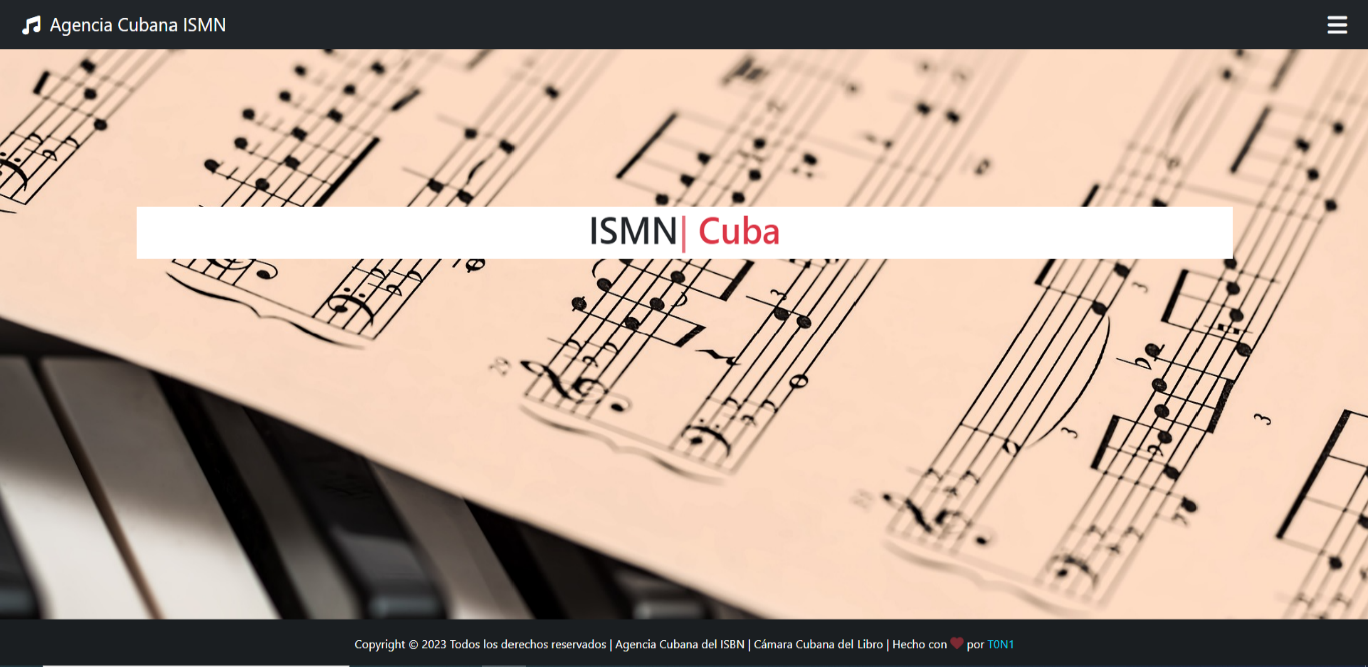
Los patrones de diseño del grupo GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.[33] A continuación, se describe el patrón GoF utilizado en el sistema:

**Decorador:** Patrón estructural que extiende la funcionalidad de un objeto dinámicamente de manera tal que es transparente a sus clientes, utiliza una instancia de una subclase de la clase original que delega las operaciones al objeto original. Este patrón se evidencia en la clase “Loginrequire” y “Cachecontrol” (decoradores que trae Django implementados, para limitar el acceso a una clase) que delegan algunas operaciones del sistema a las clases “Login” y “Register” en caso de que el usuario quiera acceder a rutas protegidas por sus decoradores.

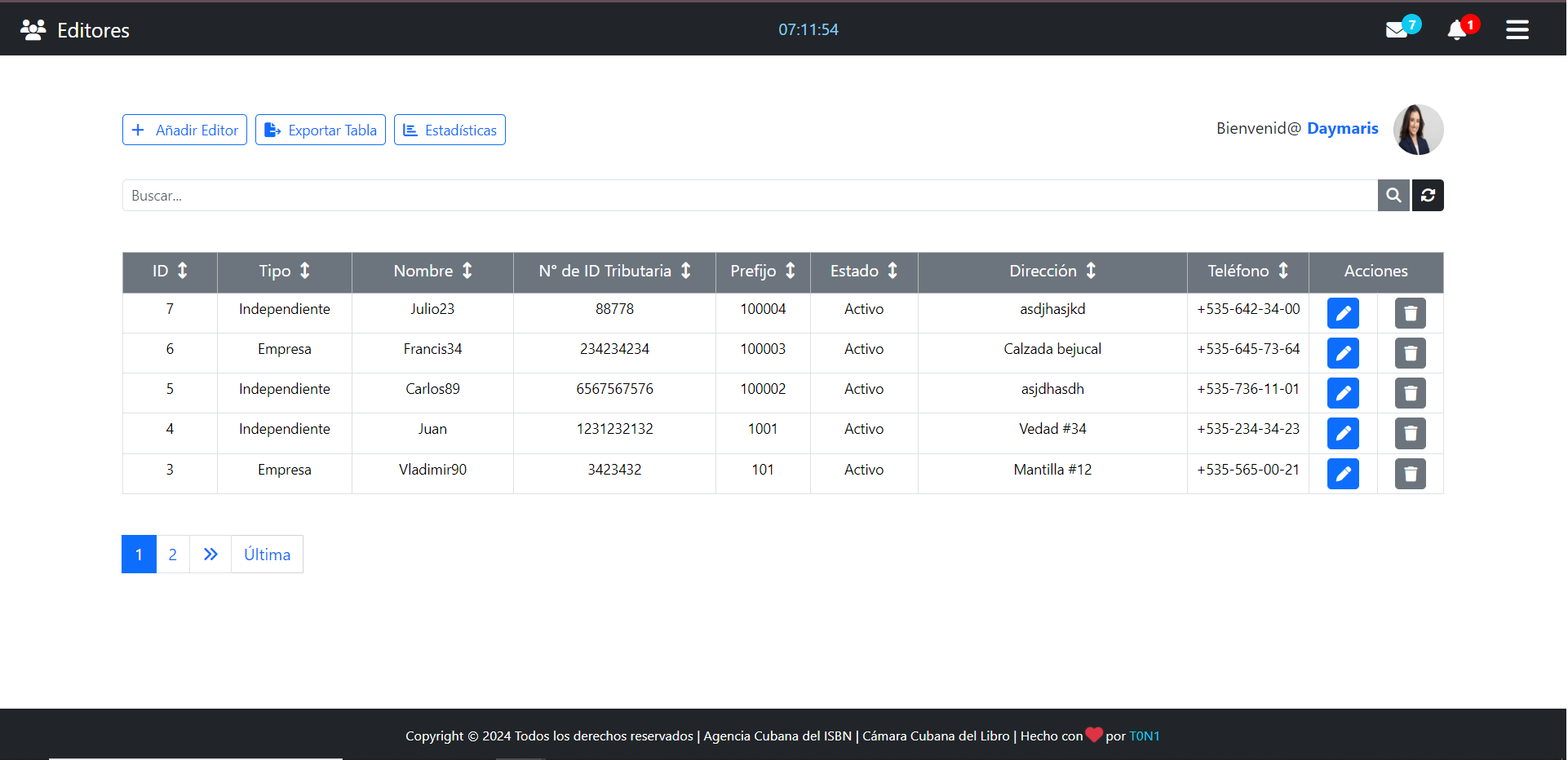
### 4.3.1 Interfaz de usuario

Para el diseño de la interfaz de usuario se tuvieron en cuenta varios aspectos:

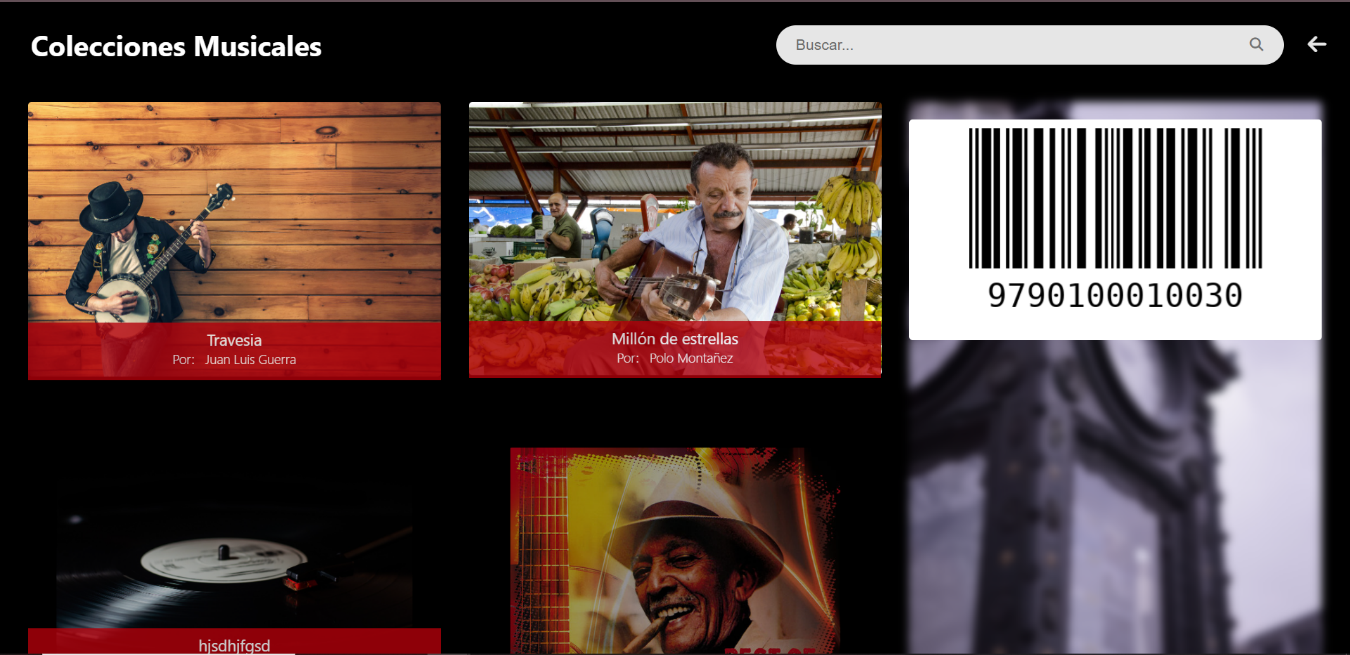
* Operatividad: La interfaz no puede exigir al usuario una interacción que no pueda realizar.
* Comprensibilidad: Tanto la información presentada como el funcionamiento de la interfaz deben ser comprensibles para el usuario.
* Estructura: Proporcionar estructura a todos los elementos de la interfaz. Se tienen una cabecera, un footer y un sidebar (barra lateral), cada uno con una función diferente y además se emplea un espacio para la información relevante, es decir, el contenido principal.
* Forma y tamaño: Para diferenciar o resaltar elementos, por ejemplo: encabezados.
* Color: También utilizado para resaltar elementos, ya sean botones, enlaces, etc. Además del empleo de colores claros para facilitar al usuario la lectura de la información y evitar distracciones.
* Usabilidad: Debe ser intuitiva para el usuario de manera que le sea fácil la navegación y la recuperación de información.



1. Página de inicio de la aplicación.



1. Página de Lista de Editores.



1. Página de Colecciones Musicales.

### 4.3.2 Formato de salida de los reportes.

Las salidas de los reportes que presenta la aplicación, muestran la información organizada en forma de modelos, estructurados por la Agencia Cubana del ISMN y de manera digital para poder ser enviados por correo o impresos.



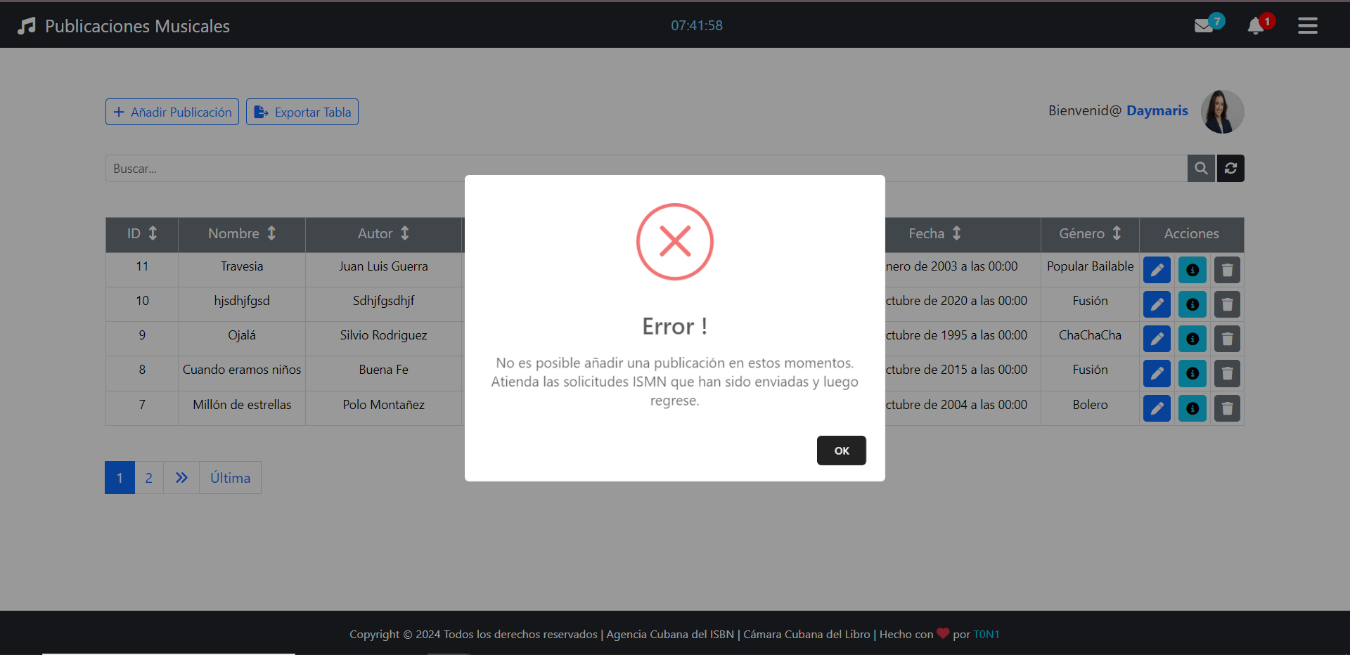
1. Reporte Estadístico de Solicitudes Página 1.



1. Reporte Estadístico de Solicitudes Página 2.

## 4.4 Tratamiento de errores

El sistema está diseñado para mostrar mensajes de error cuando no se inserta un dato correctamente en un formulario como pudiera ser una fecha en futuro, un nombre de usuario que no está registrado, un archivo que no corresponde a la extensión requerida, entre otros. También muestra mensajes de confirmación cuando se solicita eliminar un dato, para prevenir borrados accidentales. En caso que existan solicitudes ISMN pendientes el sistema no debe permitir al especialista de la Agencia insertar manualmente una Publicación Musical en la Base de Datos dado que esto puede llevar al sistema a desperdiciar prefijos e incurrir en errores. Los errores HTTP del lado del servidor como pudieran ser los “Page not found 404” o “Internal server error 500” se capturan mediante sentencias de código en el modelo ‘Middleware’ que actúa como capa intermedia entre las peticiones del cliente y las respuestas del servidor, se redirecciona a una página de error elaborada y decoradas para que el sistema no colapse y no muestre errores poco vistosos al usuario.



1. Error que muestra el sistema al intentar insertar una Publicación Musical mientras aún hay solicitudes ISMN pendientes.

Vista de un error desde el servidor, de forma amigable al cliente y mediante la plantilla 404.html:



1. Error que muestra el sistema al intentar acceder a una URL que no existe en el sistema.

## 4.5 Diseño de la base de datos.

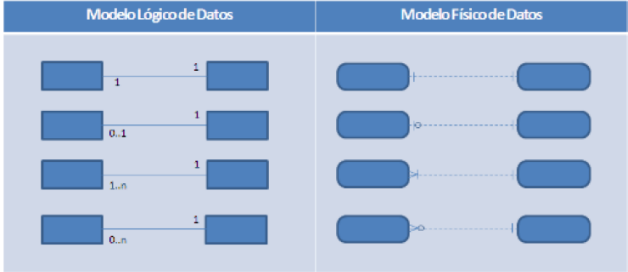
En este punto del proyecto se presenta la forma en que serán almacenados los datos, utilizando para ello una base de datos relacional. Esta representación responde al análisis previamente realizado, donde se identificó la información que por su importancia se debe almacenar, o sea, los datos que deben persistir y las relaciones que entre ellos existan.

Con la identificación de las clases se definen aquellas que son persistentes, teniendo en cuenta que la persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Seguidamente se muestran los modelos lógico y físico de datos obtenidos del diseño.

El **Modelo Lógico** de datos tiene como objetivo proveer una vista de las entidades claves que persisten, así como sus relaciones.

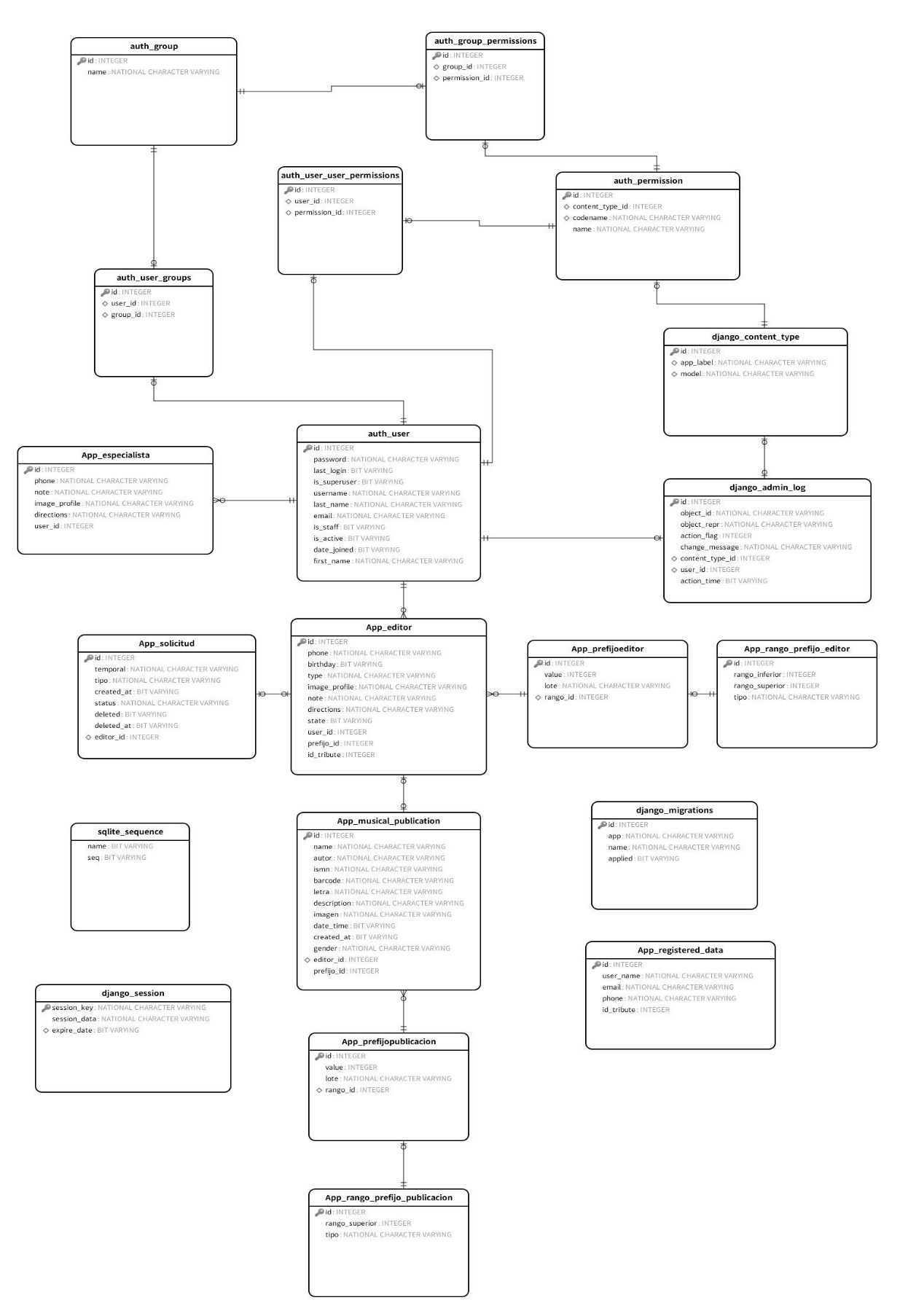
El **Modelo Físico** de la base de datos está dirigido a representar detalladamente la estructura física de la misma, como: tablas, vistas y procedimientos almacenados. Este modelo representa, además, las características de almacenamiento de los datos, como pueden ser, el tipo de dato, longitud y acceso de estos, aplicados a las columnas de una tabla y que son de gran importancia, pues constituyen reglas de validación en sí mismo que ya le van confiriendo un carácter dinámico a la base de datos. Los datos son transformados según los perfiles específicos del lenguaje y el SGBD. En el caso de este sistema se utilizó SQLite3 como SGBD, donde los tipos de datos string quedaría como varchar, los datos int quedan como integer y los boolean como bool, ya que utilizaremos la versión 3.0.1 que trae implementado este tipo de datos.

**Transición del Modelo lógico al modelo Físico.**

****

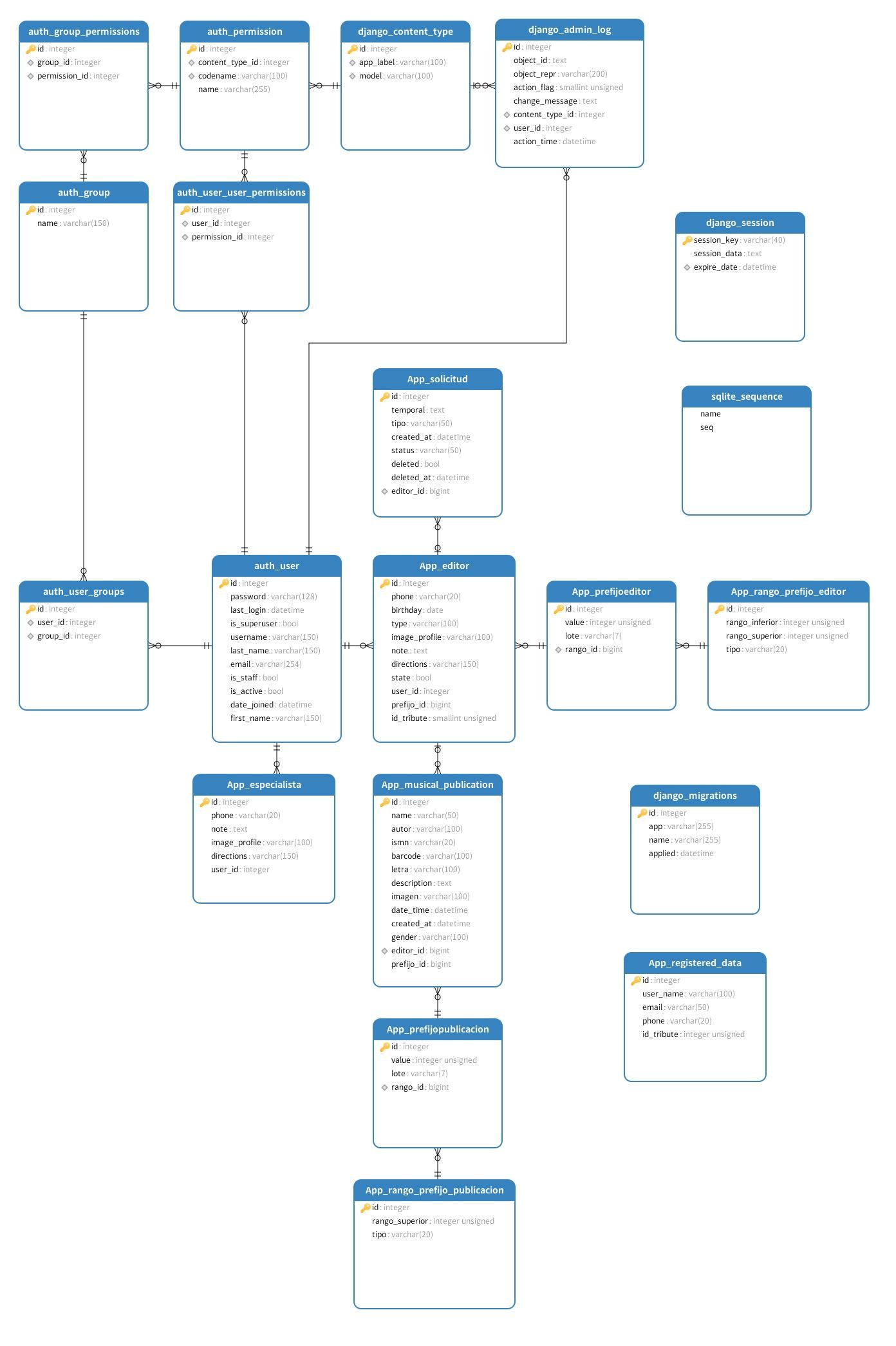
1. Transición del modelo lógico de datos al modelo físico.

### 4.5.1 Modelo lógico de datos.



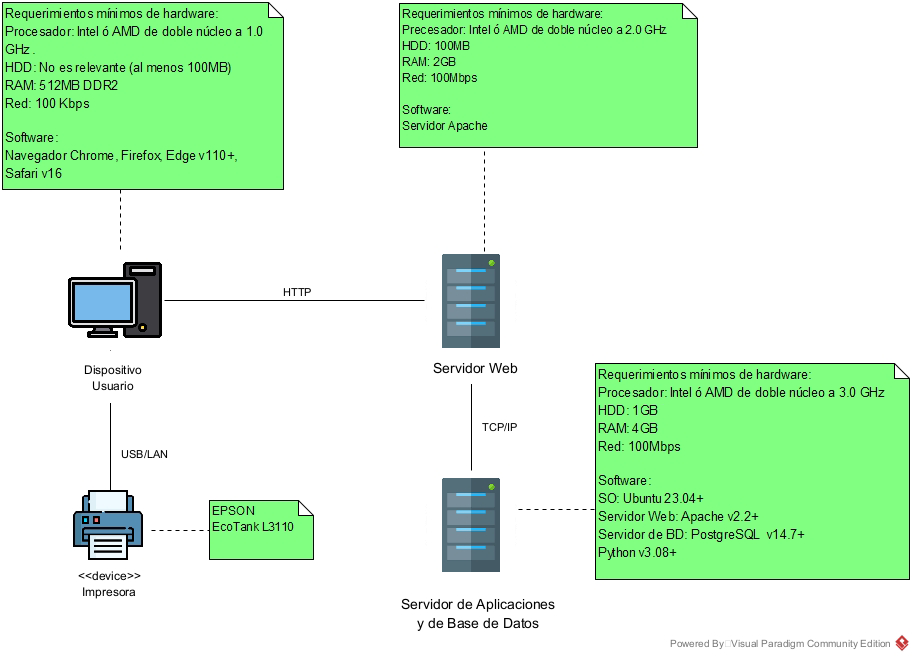
1. Modelo lógico de datos.

### 4.5.2 Modelo físico de datos.



1. Modelo físico de datos.

## 4.6 Diagrama de despliegue.



1. Diagrama de despliegue.

**Dispositivo Cliente**: Estación de trabajo desde la cual el especialista de la Agencia accede al sistema, introduce los datos y consulta todos los reportes devueltos por la aplicación, así como también representa los dispositivos desde los cual los editores realizan consultas y envían información a la base de datos. Debe contar con un navegador web instalado.

**Servidor Web:** Servidor encargado de recibir las peticiones de los usuarios, realizar las consultas correspondientes, almacenar y transmitir el contenido solicitadoal navegador del usuario.

**Servidor de Aplicaciones y de Base de Datos:** En este, se encuentran alojados todos los archivos del framework, recursos web como imágenes, scripts, hojas de estilo, etc. También se encuentra instalada la base de datos del sistema.

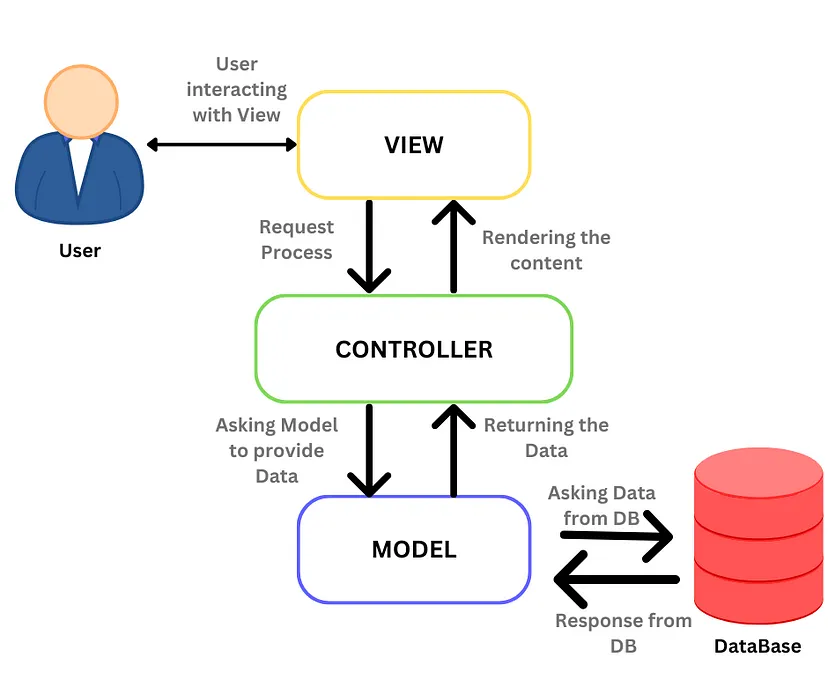
**Impresora**: Las PC de los especialistas estarán conectadas al servicio de impresión en red. Esta impresora es utilizada para imprimir las publicaciones musicales e ISMN una vez terminados, modelos, listados y otros documentos.

**Arquitectura empleada**

Se emplea una arquitectura cliente-servidor, utilizando el patrón de diseño Modelo Vista Controlador (MVC). Este es un patrón de arquitectura de software que plantea la construcción de aplicaciones informáticas agrupando las partes que las componen en tres capas o paquetes:

* 1. **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Es la abstracción mediante clases, de los datos almacenados en la base de datos.
  2. **Vista**: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.
  3. **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información. Hace de intermediario entre la 'vista' y el 'modelo'.

Este patrón permite separar la lógica de presentación, de la lógica de los datos y lógica del negocio, delegando cierta autonomía en caso de que sea necesario realizar cambios en uno u otro.

****

1. Funcionamiento de un Modelo Vista Controlador.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

* 1. **Interacción del usuario a través de la Vista:** Comienza cuando un usuario interactúa con la aplicación, generalmente a través de la interfaz de usuario presentada por la Vista. Esta interacción podría ser simple como enviar un formulario o hacer clic en un botón, o podría implicar acciones más complejas. La Vista captura estas acciones del usuario y envía una solicitud al Controlador.
  2. **Controlador:** A menudo considerado como el cerebro del patrón MVC, captura estas solicitudes y determina cómo deben manejarse. Tiene en cuenta la lógica y las reglas del negocio.
  3. **Petición de datos al Modelo:** El Controlador solicita datos al Modelo, especificando los que necesita. El Modelo, responsable de la integridad de los datos, formula consultas precisas a la base de datos y se comunica con el servidor de la base de datos enviándolas.
  4. **Respuesta de la Base de Datos:** El servidor de la base de datos entra en acción, procesa consultas y recupera los datos. Estos datos, cuidadosamente empaquetados como un conjunto de resultados, se envían de vuelta al modelo.
  5. **El Modelo prepara los datos:** El Modelo procesa datos, asegurando la alineación con la estructura de la aplicación. Luego entrega los datos al Controlador.
  6. **El Controlador decide:** El Controlador, armado con datos, toma decisiones basadas en las acciones del usuario y la lógica de la aplicación.
  7. **Presentación de datos:** Finalmente, cuando los datos deben mostrarse o presentarse al usuario, el Controlador colabora con la Vista. La Vista utiliza estos datos para representar la interfaz de usuario en consecuencia.

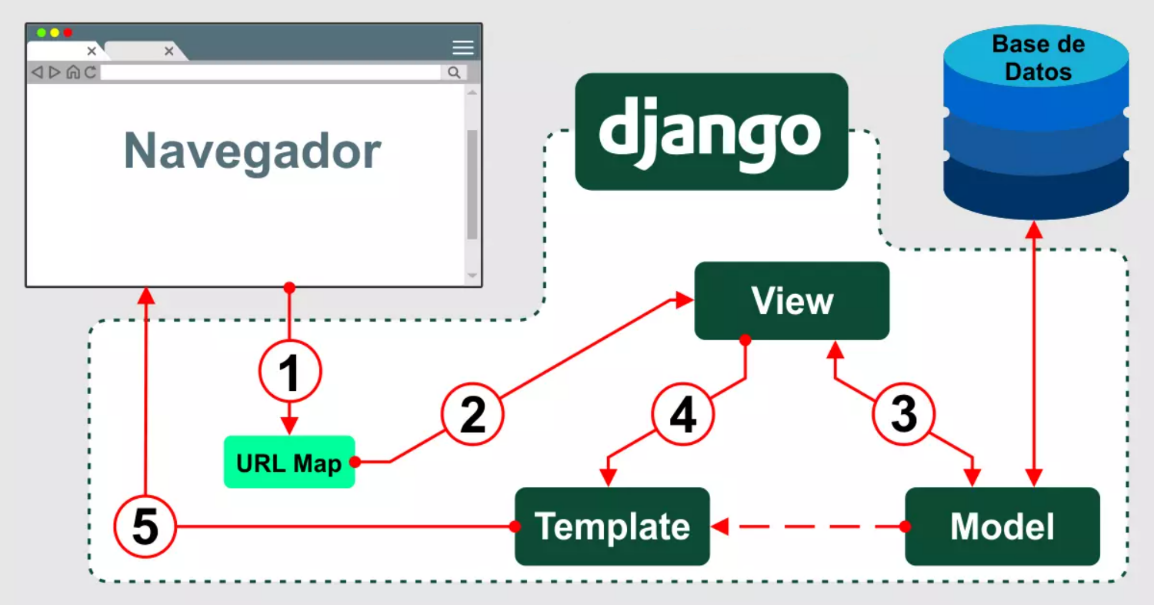
### 4.6.1 El flujo de las aplicaciones Django y el Modelo-Vista-Plantilla (MVT).

Django utiliza el patrón de diseño MVT, las siglas en ingles MTV corresponden a Model (Modelo), Template (Plantilla) y View (Vista).

En la práctica el patrón MTV es muy similar al MVC. Realmente este no se desvía demasiado, simplemente lo implementa de una manera distinta y para evitar confusiones es llamado MTV, en su composición se puede ver la similitud:

* **Modelo:** Maneja todo lo relacionado con la información, esto incluye como acceder a esta, la validación, relación entre los datos y su comportamiento.
* **Vista:** Es un enlace entre el modelo y el template. Decide qué información será mostrada y por cual template.
* **Template:** Decide como será mostrada la información.

**Funcionamiento del MTV de Django**



1. Django Patrón MTV.

Cuando el usuario se encuentra en su navegador, al momento de hacer click en un enlace o escribir una dirección **(1)** se accede al mapa de URLs (también conocido como URL map o URL conf), en este archivo cada ruta está asociado con una view **(2)**, si se necesita algún dato se solicitará este a model **(3)**, el cual a través del ORM generará la consulta a la base de datos, luego los datos son enviados al template **(4)** que contiene la lógica de presentación para estos. Luego de renderizar y procesar la página, esta se envía al navegar que hizo la solicitud **(5)**.

**Mecanismo de Acceso a Datos**

ORM significa Mapeador relacional de objetos (Object Relational Mapping). El objetivo principal de ORM es enviar datos entre una base de datos y modelos en una aplicación. Mapea una relación entre la base de datos y un modelo. Entonces, ORM asigna atributos de objetos a campos de una tabla. La principal ventaja de utilizar ORM es que hace que todo el proceso de desarrollo sea rápido y sin errores. Básicamente, elimina la necesidad de escribir código SQL. Django está equipado con un ORM que es uno de los mejores en la industria en la actualidad. Está estrechamente acoplado con el framework y es muy eficiente. El ORM de Django es mejor conocido por su capacidad para manejar consultas de complejidad media a baja y conjuntos de datos de medianos a grandes. Las migraciones son otra característica útil de Django ORM.[34]

El uso principal de una base de datos es almacenar y recuperar datos cuando sea necesario. Django ORM utiliza Querysets para lograr esta funcionalidad. Queryset es una lista de objetos presentes en un modelo. Se puede utilizar Querysets para filtrar, organizar y gestionar los datos. El objetivo principal de estos es hacer que la recuperación de consultas sea más rápida y sencilla.

### 4.6.2 Estructuración en Capas.

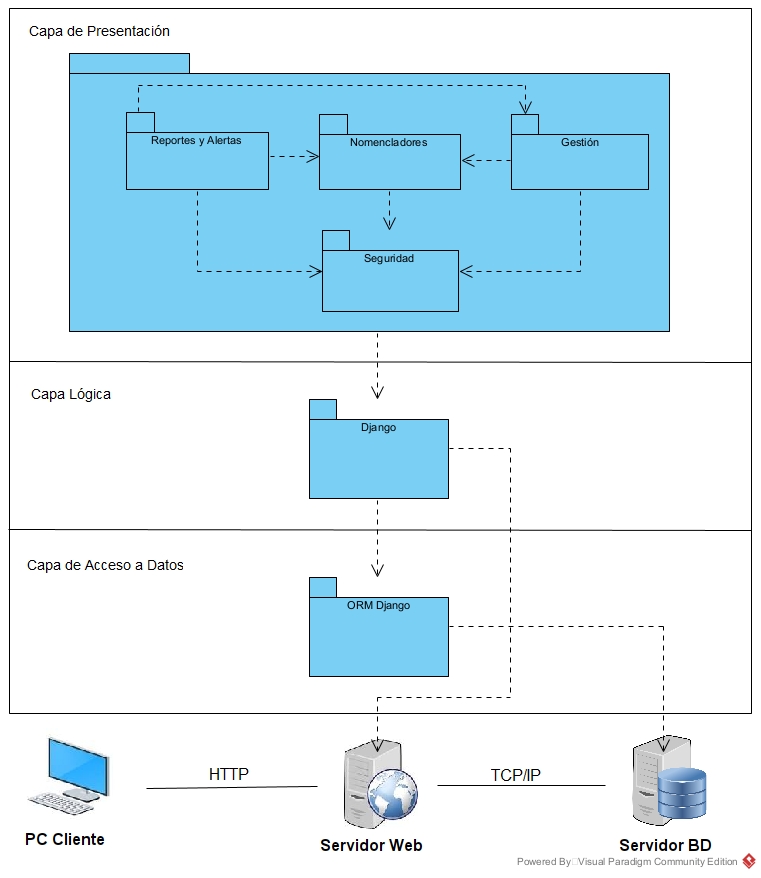
**Descripción del diagrama de estructuración en capas.**

En el enfoque de responsabilidad las capas se encuentran separadas unas de otras en: capa de presentación, capa de la lógica del negocio y capa de acceso a datos, donde los elementos están explícitamente agrupados según su nivel de responsabilidad dentro del sistema, donde cada una de estas capas brinda servicios a la capa superior y hace uso de las funcionalidades de la capa inferior.

**Capa de Presentación:** Empleada para interactuar con el usuario. En esta capa de abstracción se implementan los módulos a los que acceden los especialistas y editores, como son: Reportes, Gestión, Nomencladores y Seguridad.

**Capa de Lógica de Negocio:** Se encuentran los archivos del framework Django. Aquí se encuentran todas las clases, paquetes y bibliotecas que ofrecen diversas funcionalidades y servicios a la Capa de Presentación, necesarios para el funcionamiento de la aplicación.

**Capa de Acceso a Datos:** Capa puente encargada de mediar entre los controladores y la base de datos. Se ubican las clases, subsistemas y paquetes que contienen la lógica para convertir el código Python en sentencias SQL para acceder al medio de almacenamiento persistente, es la única que puede tener acceso directo a la información almacenada en la base de datos.



1. Estructuración en capas.

## 4.7 Conclusiones

En este capítulo se describieron de forma breve los principios de diseño del sistema, como el tratamiento de errores. Además, se modelaron las clases de diseño de la aplicación, se mostró el modelo lógico del cual se generó el modelo físico de datos y el diagrama de despliegue con la distribución de los nodos y las características de cada uno, concluyendo así con la propuesta de solución que satisface los requerimientos del sistema.

# **Capítulo 5. Validación y factibilidad de la solución propuesta.**

## 5.1 Introducción.

Para emprender cualquier proyecto empresarial, que involucre recursos humanos y materiales, es necesario conocer con antelación los beneficios que éste aportará a la empresa, cuál sería su costo aproximado, entre otros aspectos, todo en aras de poder determinar la factibilidad del mismo. En este capítulo se definen los tipos de prueba ejecutadas en el sistema y se diseñan los casos de pruebas para obtener resultados de estos.

## 5.2 Tipos de pruebas ejecutadas.

El desarrollo de software de alta calidad está determinado por factores muy importantes, que gracias a eso el producto final sale con la eficacia requerida. Para dichos factores se tiene que tener en cuenta que es en realidad lo que necesita el cliente, que se puede modificar, o probar en caso que haga falta, y además de que pueda ser adaptable a cualquier entorno de trabajo.

Para la obtención de un buen producto se hace necesario poner en práctica el aseguramiento de la calidad, que no es más que aquellas acciones que hacen que un producto o servicio cumpla con determinados requisitos de calidad. Para esto, en el desarrollo de software se utiliza la prueba de software, que es una de las tareas más importantes, pero también es compleja y no siempre adoptada correctamente.

La prueba de software es la verificación dinámica del comportamiento de un programa contra el resultado esperado, a partir de un conjunto finito de casos de prueba, específicamente seleccionados dentro de un dominio usualmente infinito de ejecuciones posibles.[35]

Para lograr la entrega de un producto con la mayor calidad posible garantizando una correcta funcionalidad del sistema se realizan diferentes tipos de prueba. Las pruebas que se ejecutaron en este proyecto fue la prueba de funcionalidad con el método de prueba de caja negra y la prueba de validación.

**Pruebas de funcionalidad:**

Las pruebas Funcionales deben enfocarse en los requisitos funcionales, las pruebas pueden estar basadas directamente en los Casos de Uso (o funciones de negocio), y las reglas del negocio. Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Las metas de estas pruebas son:

• Verificar la apropiada aceptación de datos.

• Verificar el procesamiento y recuperación y la implementación adecuada de las reglas del negocio.

**Pruebas de Validación:**

Son el proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su cometido. Es normalmente una parte del proceso de pruebas de software de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones y tutoriales. La validación es el proceso de comprobar que lo que se ha especificado es lo que el usuario realmente quería. Facilita la depuración de errores durante la entrada de datos al sistema.

Este tipo de prueba está basada en técnicas de caja negra, que es, aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea, entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace.[36]

Un caso de prueba es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación, un sistema software, o una característica de éstos son parcial o completamente satisfactorias.

Para preparar los casos de pruebas, se necesitan un número de datos que ayuden a la ejecución de estos casos y permitan que el sistema se ejecute en todas sus variantes. Se pueden introducir datos válidos o no válidos para el programa según lo que se desea: hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa funcione correctamente.

## 5.3 Diseño de los casos de prueba.

### 5.3.1 Pruebas de validación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nombre del CU** | **Escenario** | **Descripción** | **Juego de Datos** | **Resultado esperado** | **Resultado real** |
| 1 | Autenticar usuario | Usuario y contraseña correctos. | Se introduce un usuario que se encuentra registrado y activo en la base de datos. | Usuario: Daymaris  Contraseña: \*\*\*\*\*\*\*\* | El sistema carga el perfil del usuario autenticado. | Se permite la entrada al sistema y se redirecciona a la página correspondiente al usuario que ha inicado sesión. |
| 2 | Autenticar usuario | Campos vacíos. | Se deja al menos un campo vacío o ambos campos. | Usuario:  Daymaris  Contraseña:  Null | El sistema verifica los datos del usuario y muestra el mensaje de error: “¡Acceso Denegado! Inserte sus datos correctamente” | Muestra el mensaje de error y no permite la entrada al sistema. |
| 3 | Autenticar usuario | Usuario correcto, pero contraseña incorrecta. | Se introduce un usuario que se encuentra registrado en la base de datos, pero la contraseña es incorrecta | Usuario: Daymaris  Contraseña:  \* | El sistema verifica los datos del usuario y muestra el mensaje de error: “¡Acceso Denegado! Inserte sus datos correctamente” | Muestra el mensaje de error y no permite la entrada al sistema. |
| 4 | Autenticar Usuario | Usuario incorrecto. | Se introduce un nombre de usuario no registrado en la base de datos. | Usuario:  Toni  Contraseña:  \*\*\*\*\*\*\* | El sistema verifica los datos del usuario y muestra el mensaje de error: “¡Acceso Denegado! Inserte sus datos correctamente” | Muestra el mensaje de error y no permite la entrada al sistema. |
| 5 | Autenticar usuario | Usuario y contraseña correctos, pero inactivo. | Se introduce un usuario que se encuentra registrado pero inactivo en la base de datos. | Usuario: Carmen  Contraseña: \*\*\*\*\*\*\*\* | El sistema verifica los datos del usuario y muestra un mensaje de error: “Su usuario ha sido desactivado. Contacte a la Agencia.” | Muestra el mensaje de error, no permite la entrada al sistema y limpia los campos del formulario. |
| 6 | Registro de usuario. | Usuario existente en el sistema. | Se introduce un nombre de usuario que se encuentra registrado en la BD. | Usuario:  Daymaris  Email:  daymarisisbn@gmail.com  Contraseña:  \*\*\*\*\*\*\*\* | El sistema valida los datos del usuario a registrar y muestra un mensaje de error: “Este nombre de usuario ya existe en nuestra Base de Datos. Inserte otro nombre de usuario”. | Se cancela el registro, se limpian los campos del formulario de registro y se muestra el mensaje de error. |
| 7 | Registro de usuario. | Contraseña que no cumple con los parámetros de seguridad. | Se introduce una contraseña de menos de 8 caracteres. | Usuario:  Toni  Email:  toni@gmail.com  Contraseña:  \*\*\*\* | El sistema valida los datos del usuario a registrar y muestra un mensaje de error: “¡Contraseña no valida! Su contraseña debe tener al menos 8 caracteres”. | Se cancela el registro, se limpian los campos del formulario de registro y se muestra el mensaje de error. |
| 8 | Registro de usuario | Número de confirmación de registro incorrecto. | Se introduce un número de confirmación de registro incorrecto. | Número de confirmación:  \*\*\*\* | El sistema valida el número de confirmación y muestra un mensaje de error: “Confirmación incorrecta! Revise su correo ”. | Se cancela el registro, se limpia el campo del formulario de confirmación de correo y se muestra el mensaje de error. |

1. Descripción caso de uso de prueba <Autenticar, Registrar Usuario>

### 5.3.2 Pruebas de Funcionalidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sección** | **Escenario** | **Juego de Datos** | **Resultado esperado** | **Resultado real** |
| **SC1: Crear Solicitud ISMN** | **EC1.1:** Solicitud Creada y enviada satisfactoriamente. | titulo: “Cómo fue?”  autor = “Benny Moré”  genero\_musical: “Bolero”  fecha\_creacion: 10/04/1956  descripcion: “Emblema de Cuba”  letra: “media/publications/letters/Como\_Fue.pdf”  portada: “media/publications/cover/Como\_Fue.jpg” | El sistema crea la solicitud y la almacena en la BD. | El sistema valida los datos, crea una solicitud y la almacena en la BD. |
| **EC1.2:** Solicitud no creada, campo obligatorio vacío. | titulo: “Cómo fue?”  autor = “Benny Moré”  genero\_musical: “Bolero”  fecha\_creacion: 10/04/1956  descripcion: “Emblema de Cuba”  letra: “”  portada: “media/publications/cover/Como\_Fue.jpg” | El sistema valida los datos, pero no crea la solicitud y notifica el error. | El sistema valida los datos y no crea una solicitud, ya que detecta campos requeridos vacíos e informa del error y el campo en cuestión. |
| **EC1.3:** Solicitud no creada, ya está en proceso. | titulo: “Cómo fue?”  autor = “Benny Moré”  genero\_musical: “Bolero”  fecha\_creacion: 10/04/1956  descripcion: “Emblema de Cuba”  letra: “media/publications/letters/Como\_Fue.pdf”  portada: “media/publications/cover/Como\_Fue.jpg” | El sistema valida los datos y no crea la solicitud. | El sistema no crea la solicitud, valida que ya existe una solicitud con los mismos campos ‘titulo’ y ‘autor’ y notifica al usuario que su solicitud ya está siendo procesada. |
| **SC2: Rechazar Solicitud.** | **EC2.1:** Solicitud rechazada. | status = “Pendiente”  reject = “False”  reject\_at = “”  motivo = “El documento que contiene la letra de la publicación o la partitura no se abre correctamente” | El sistema actualiza las estadísticas, captura el campo: “motivos” y notifica al correo del editor. | El sistema actualiza las estadísticas, captura el campo: “motivos” y tarda en responder para finalmente mostrar un error 500:“TimeOutError” |
| **EC2.2:** Error de conexión. | status = “Pendiente”  reject = “False”  reject\_at = “”  motivo = “El documento que contiene la letra de la publicación o la partitura no se abre correctamente” | El sistema actualiza las estadísticas, captura los motivos y notifica al correo del editor. | El sistema actualiza las estadísticas, captura el valor del campo: “motivos”, renderiza la plantilla de rechazo y la envía al correo del editor correctamente. |
| **EC2.3:** Error al enviar el rechazo. | status = “Pendiente”  reject = “False”  reject\_at = “”  motivo = “” | El sistema actualiza las estadísticas, captura los motivos y notifica el error al especialista. | El sistema actualiza las estadísticas, captura el valor del campo: “motivos”, y muestra un mensaje de error al especialista indicando que no ha especificado un motivo. |

1. Descripción caso de uso de prueba <Gestionar Solicitud>

## 5.4 Resultados obtenidos en las pruebas.

**Resultado**

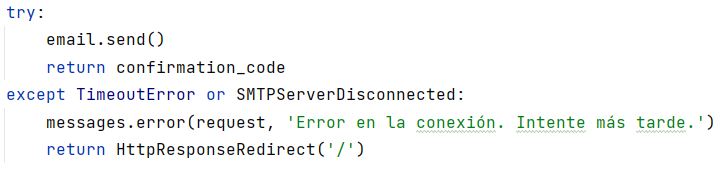
Con las pruebas funcionales realizadas al caso de uso “Gestionar Solicitud ISMN”, se evaluaron un total de 6 escenarios.

En 1 de ellos no se obtuvo el resultado esperado, lo que significa que el 16,6% de las veces la aplicación no funcionó correctamente.

**Problemas identificados y cambios realizados en el caso de uso Gestionar Solicitud ISMN:**

En el escenario 2.1 se encontró que la aplicación presenta un error al no establecer conexión con el servidor SMTP.

Para dar solución a esta deficiencia funcional se modificó el código encargado de notificar al editor del rechazo en su solicitud:



1. Solución al error del escenario de prueba 2.1

La solución aplicada fue declarar una excepción que notifica al usuario y evita que el servidor se caiga tras ocurrir un error de conexión con el SMTP.

Durante el proceso de prueba se obtuvieron los resultados esperados, se lograron detectar algunos errores de validación y de funcionamiento los cuales fueron corregidos satisfactoriamente. Se obtuvo la aprobación de los usuarios lo cual permitió al desarrollador corregir requisitos a la hora de interactuar con el sistema, lográndose así la realización de un software funcional y seguro, que cumple con las necesidades del cliente.

## 5.5 Beneficios tangibles e intangibles.

### 5.5.1 Introducción

Para emprender cualquier proyecto empresarial, que involucre recursos humanos y materiales, es necesario conocer con antelación los beneficios que éste aportará a la empresa, cuál sería su costo aproximado, entre otros aspectos, todo en aras de poder determinar la factibilidad del mismo.

### 5.5.2 Planificación basada en casos de uso

La estimación mediante el análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. Este método se va refinando con la experiencia de cada ingeniero de software ya que lleva incorporado muchos elementos de apreciación y valoración por parte de este.[37]

El primer paso a realizar en la estimación es calcular los Puntos de Caso de Uso sin Ajustar, calculándose de la siguiente forma:

**PCU = FPA + FPCU donde:**

- **PCU**: Puntos de casos de uso sin ajustar.

- **FPA**: Factor de peso de los actores sin ajustar.

- **FPCU**: Factor de peso de los casos de uso sin ajustar

**El Factor de Peso de los Actores sin ajustar (FPA)** se calcula teniendo en cuenta la cantidad de actores del sistema y la complejidad de cada uno de ellos. Teniendo en cuenta que los actores del sistema pueden ser personas u otros sistemas, a continuación se muestra una tabla con la descripción de la complejidad de cada uno.

|  |  |  |
| --- | --- | --- |
| **Tipo de Actor** | **Descripción** | **Factor de Peso** |
| Simple | Un sistema que interactúa con el sistema desarrollado mediante una interfaz de programación (API, por sus siglas en inglés) | 1 |
| Medio | Un sistema que interactúa con el sistema desarrollado mediante un protocolo o una interfaz basada en texto. | 2 |
| Complejo | Una persona que interactúa con el sistema mediante una interfaz gráfica. | 3 |

1. Factores de Peso de los actores sin ajustar

El sistema está integrado por 3 actores complejos.

|  |  |  |
| --- | --- | --- |
| **Actor** | **Tipo** | **Peso** |
| Usuario | Complejo | 3 |
| Especialista | Complejo | 3 |
| Administrador | Complejo | 3 |

1. Complejidad de los actores del sistema.

**FPA:** FPA: ∑ (cantidad de actores de tipo i) \* (factor de peso en i)

**FPA** = 3 \* 3 = 9

**El Factor de peso de los casos de uso sin ajustar (FPCU)** se calcula teniendo en cuenta la cantidad de casos de uso del sistema y la complejidad de cada uno de ellos. La complejidad se determina a partir de la cantidad de transacciones que se realizan. La transacción es una secuencia atómica de actividades, las cuales se realizan completamente o no se realiza ninguna.

|  |  |  |
| --- | --- | --- |
| **Tipo de CU** | **Descripción** | **Factor de Peso** |
| Simple | De 1 a 3 transacciones | 5 |
| Medio | De 4 a 7 transacciones | 10 |
| Complejo | De 8 o más transacciones | 15 |

1. Factor de peso de los casos de uso sin ajustar.

El FPCU se calcula teniendo en cuenta el total de casos de uso, multiplicado por el factor de peso asignado a cada uno.

**FPCU** = ∑ (cantidad de CU de tipo i) \* (factor de peso en i)

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Tipo de CU** | **Peso** |
| Autenticar Usuario | Simple | 5 |
| Cambiar Contraseña | Simple | 5 |
| Ver información de usuario conectado | Simple | 5 |
| Restablecer Contraseña | Medio | 10 |
| Gestionar información de usuario | Simple | 5 |
| Gestionar grupos de autenticación y autorización | Simple | 5 |
| Gestionar roles de usuario | Simple | 5 |
| Visualizar y exportar trazas | Simple | 5 |
| Realizar salvas y restaura de la BD | Simple | 5 |
| Gestionar Solicitudes de Inscripción | Simple | 5 |
| Gestionar Editores | Simple | 5 |
| Gestionar publicaciones musicales | Simple | 5 |
| Gestionar Solicitudes ISMN | Simple | 5 |
| Gestionar catálogo de publicaciones musicales | Simple | 5 |
| Exportar ISMN | Simple | 5 |
| Gestionar ISMN | Medio | 10 |
| Gestionar Campos de Publicación Musical | Medio | 10 |
| Gestionar Editor | Simple | 5 |
| Ver Colección de publicaciones musicales | Simple | 5 |
| Realizar búsqueda general | Simple | 5 |
| Visualizar ISMN y código de barras | Simple | 5 |
| Visualizar listado de editores | Simple | 5 |
| Visualizar listado de publicaciones musicales | Simple | 5 |
| Ver estadísticas | Simple | 5 |
| Exportar tablas estadísticas | Simple | 5 |
| Exportar publicación musical | Simple | 5 |
| Visualizar motivos de rechazo | Simple | 5 |
| Visualizar aceptación de inscripción | Simple | 5 |
| Visualizar solicitud ISMN aceptada | Simple | 5 |
| Ver solicitudes pendientes | Simple | 5 |
| Visualizar Alertas | Simple | 5 |
| Generar Alertas | Simple | 5 |

1. Pesos de los casos de uso sin ajustar

El sistema tiene 29 casos de uso de tipo simple y 3 casos de uso de tipo medio. **FPCU** = (29 \* 5) + (3 \* 10) = 175

Por lo tanto, utilizando la fórmula **PCU = FPA + FPCU** se tiene que:

**PCU** = 9 + 175 = 184

Para el cálculo de los Puntos de CU Ajustados se utiliza la ecuación:

**PCUA = PCU \* FCT \* FA** donde:

**- PCUA:** Puntos de CU Ajustados

**- PCU:** Puntos de CU sin Ajustar

**- FCT:** Factor de Complejidad Técnica

**- FA:** Factor de Ambiente

**El Factor de complejidad técnica (FCT)**

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor** | **Descripción** | **Peso** | **Valor Asignado** | **Total** |
| T1 | Sistema distribuido | 2 | 0 | 0 |
| T2 | Tiempo de respuesta | 1 | 3 | 3 |
| T3 | Eficiencia del usuario final | 1 | 3 | 3 |
| T4 | Procesamiento interno complejo | 1 | 1 | 1 |
| T5 | El código debe ser reutilizable | 1 | 2 | 2 |
| T6 | Facilidad de instalación | 0.5 | 4 | 2 |
| T7 | Facilidad de uso | 1 | 4 | 4 |
| T8 | Portabilidad | 2 | 4 | 8 |
| T9 | Facilidad de cambio | 1 | 4 | 4 |
| T10 | Concurrencia | 1 | 3 | 3 |
| T11 | Objetivos especiales de seguridad | 1 | 1 | 1 |
| T12 | Acceso a terceras partes | 1 | 0 | 0 |
| T13 | Entrenamiento a terceros usuarios | 1 | 0 | 0 |
| **Total** | | | | **29** |

1. Factores de complejidad técnica.

Para calcular el FCT se utiliza la ecuación:

**FCT** = 0.6 + 0.01 \* ∑ (Peso en i \* Valor asignado en i)

**FCT** = 0.6 + 0.01 \* 29 = 0.89

**El Factor de Ambiente (FA)** se calcula empleando el mismo criterio del caso anterior. Se tienen en cuenta el entrenamiento y las habilidades del grupo involucrado en el desarrollo del proyecto. En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor** | **Descripción** | **Peso** | **Valor Asignado** | **Total** |
| E1 | Familiaridad con el modelo de proyecto utilizado. | 1.5 | 4 | 6 |
| E2 | Experiencia en la aplicación | 0.5 | 4 | 2 |
| E3 | Experiencia en orientación a objetos. | 1 | 5 | 5 |
| E4 | Capacidad del analista. | 0.5 | 5 | 2.5 |
| E5 | Motivación | 1 | 5 | 5 |
| E6 | Estabilidad de los requerimientos | 2 | 5 | 10 |
| E7 | Personal media jornada | -1 | 0 | 0 |
| E8 | Dificultad del lenguaje de programación | -1 | 2 | -2 |
| **Total** | | | | **28.5** |

1. Factores ambientes del sistema

**FA** = 1.4 – 0.03 \* ∑ (Peso en i \* Valor asignado en i)

**FA** = 1.4 – 0.03 \* 28.5 = 0.545

Una vez que se tienen todos los valores se procede a realizar el cálculo de los puntos de Casos de Uso ajustados.

**PCUA = PCU \* FCT \* FA**

**PCUA** = 184\* 0.89 \* 0.545 = 89.2492

**El Esfuerzo (E)** se calcula mediante la siguiente ecuación:

**E = PCUA x FC** donde:

**- E:** Esfuerzo estimado en horas-hombre

**- PCUA:** Puntos de CU ajustados

**- FC:** Factor de conversión

Para decidir cuántas horas-hombre requiere cada Punto de Casos de Uso se valora el siguiente criterio: si al sumar el número de factores de ambiente que están por debajo de la media (3) de E1 a E6 y los que están por encima de E7 y E8 el resultado es:

- Menor o igual que 2, se usa como factor de conversión 20 horas-hombre.

- 3 o 4, se usa como factor de conversión 28 horas-hombre.

- Mayor o igual que 5, debe ajustarse, pues se considera de alto riesgo con posibilidad de fracaso.

El número de factores con valor por debajo de la media de E1 a E6 es 2 y por encima de la media de E7 a E8 es 0; por tanto se usará como factor de conversión (FC) 20 horas-hombre.

**E = PCUA \* FC**

E = 89.2492 \* 20 = 1796.98

Para determinar las estimaciones del esfuerzo para las demás actividades relacionadas con el desarrollo del software se utiliza el siguiente criterio:

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Porcentaje** | **Horas/Hombre** |
| Análisis | 10 | 365.64 |
| Diseño | 20 | 731.29 |
| Implementación | 40 | **1662.58** |
| Pruebas | 15 | 398.46 |
| Otras Actividades | 15 | 298.46 |
| Total | 100 | **3456.43** |

1. Estimación de esfuerzo en horas.

El **esfuerzo** total **E (Total)** es de **3456.43** horas/hombre

**La Estimación del tiempo de desarrollo del proyecto (TDES)** se calcula:

**TDES** (total) = **E (Total)** / **CH**, donde

- **TDES:** Tiempo de desarrollo del proyecto

- **E (Total):** Esfuerzo total

- **CH:** Cantidad de hombres involucrados en el proyecto

**TDES =** 3456.43 / 1 = **3456.43** horas/hombre

Analizando que un mes tiene aproximadamente 24 días laborables, cada día tiene 8 horas de trabajo, entonces el tiempo total se calcularía:

**T** = 24 días \* 8 horas

**T** = 192 horas

**TT** = 3456.43 / 192

**TT** = 18.002 ≈ 18 meses de desarrollo (1 año y medio).

**El Costo de desarrollo del proyecto (C)** se calcula:

**C** (total) = **E** (total) \* **CHH**

- **C** (total): costo total.

- **CHH**: costo horas hombre.

**CHH = K \* THP**

- **K**: coeficiente que tiene en cuenta los costos indirectos (1.5 y 2.0)

- **THP**: tarifa horaria promedio

Para realizar los siguientes cálculos se considera como salario mínimo mensual de un desarrollador $2375.00 CUP.

**THP** = salario básico mensual / horas laborales en el mes

**THP** = $2375.00 /192 horas

**THP** = $12.37 horas

Entonces:

**CHH** = 1.5 \* 12.37

**CHH**= 18.56

**C (total)** = E (total) \* CHH

**C (total)** = 3456.43 \* 18.56

**C (total)** = 64151,34 ≈ **$ 64150 CUP**

### 5.5.1 Beneficios tangibles:

Esta aplicación web proveerá a la Agencia una herramienta para la gestión y control de las producciones musicales generadas en el país, así como también promover la utilización del ISMN y facilitar su gestión por parte de los editores, artistas y compositores.

### 5.5.2 Beneficios intangibles:

* Disminución del tiempo y el esfuerzo de los especialistas para realizar la labor de asignación y gestión de los ISMN y las publicaciones musicales.
* Se hace posible la obtención de reportes de una manera rápida y sencilla.
* Aumenta la velocidad en la búsqueda de información.
* Disminución de errores humanos.
* Conservación de la memoria histórica de la música en Cuba.
* Apoyo en la gestión y mantenimiento de derechos de autor.
* Asignación remota de los ISMN a los editores asociados.

## 5.6 Análisis de costos y beneficios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tareas desarrolladas** | **Cantidad de trabajadores** | **Salario/Horas** | **Tiempo en Horas Consumidas** | **Costo por Horas** |
| **Estudio del negocio** | 1 | 12.37cup/h | 200h (4h \* 50d) | $ 2474 |
| **Captura de requisitos** | 200h (5h \* 40d) | $2474 |
| **Estudio y selección de las tecnologías** | 140h (4h \* 35d) | $1731.80 |
| **Diseño de la solución propuesta** | 500h (5h \* 100d) | $6185 |
| **Implementación de la aplicación** | 1600h (8h \* 200d) | $19792 |
| **Documentación** | 210h (3h \* 70d) | $2597.70 |
| **Pruebas de software** | 260h (4h \* 65d) | $3216.20 |
| **Presentación del software al usuario** | 40h (4h \* 10d) | $494.80 |
| **Total** | | | **2550 horas** | **$38965.50cup** |
| **Gastos en consumo de electricidad, medios de cómputo, materiales de oficina y acceso a Internet** | 1 | 5.70cup/h | 4320h (8h \* 540d) | $24624 |
| **Total** | | | **6870 horas** | **$63589.50cup** |

1. Análisis de costo y beneficio.

Luego de analizar y calcular el tiempo dedicado al desarrollo de la aplicación se estima que el tiempo real es de 1 año y medio aproximadamente con un costo de $63589CUP por conceptos de gastos de salario, consumo eléctrico, internet y depreciación de equipos de trabajo. Es posible llegar a la conclusión de que la realización del mismo resulta totalmente factible, teniendo en cuenta para la emisión del criterio final los siguientes aspectos:

• En el desarrollo de la aplicación no se incurrió en gastos adicionales de equipamiento o compra de otros sistemas necesarios ni de herramientas de desarrollo requeridas para el mismo.

• La misma fue desarrollada por un estudiante y no por trabajadores profesionales de experiencia.

## 5.7 Conclusiones

En este capítulo se muestran los resultados de las pruebas realizadas a la aplicación, llamadas a garantizar su buen desempeño y estabilidad, validando estos análisis para diferentes escenarios y casos de uso. Se realizó el análisis de la factibilidad correspondiente al desarrollo del proyecto, considerando los beneficios que aporta su implementación, el tiempo estimado de realización y los gastos en que incurre la empresa por un desarrollo de este tipo importado, con los correspondientes gastos por concepto de mantenimiento. Se concluye que es factible este desarrollo.

# **Conclusiones**

Luego del desarrollo del presente trabajo, se concluye que se logró cumplir con los objetivos propuestos, ya que se diseñó e implementó una aplicación informática que permite la inserción de los códigos ISMN en las publicaciones musicales del país. Para ello:

* Se analizó el flujo actual del proceso de asignación ISBN.
* Se estudiaron las tendencias actuales en las tecnologías de desarrollo web, acordándose la utilización de herramientas de código libre.
* El sistema visualiza las publicaciones musicales junto a sus editores, el ISMN generado por el especialista y su código de barra correspondiente.
* Se implementaron los mecanismos de seguridad requeridos para garantizar el acceso a la información gestionada y garantizar su integridad.
* Se incrementó la productividad y efectividad del equipo de especialistas de la Agencia, permitiendo asignar mayor cantidad de ISMN y gestionar un mayor número de publicaciones musicales en un tiempo considerablemente menor.
* Para el estudio de factibilidad se utilizó el análisis de Puntos de Casos de Uso que resulta un método obsoleto respecto al uso de las tecnologías actuales. No se encontró otro que se pudiera adecuar. No obstante, el desarrollo de la aplicación no tenía presiones de tipo restrictivas en cuanto al tiempo o los recursos a emplear por lo que solo se ha incorporado este análisis con un enfoque docente.
* La experiencia de trabajo con un marco de trabajo de tantas facilidades como Django ha demostrado en la práctica lo potenciado que puede resultar el desarrollo de una aplicación de complejidad media para un desarrollador principiante como el autor.

# **Recomendaciones**

En aras de ampliar y mejorar las funcionalidades de la aplicación se proponen las siguientes recomendaciones:

* Mantener seguimiento de las futuras actualizaciones o nuevas versiones de los frameworks y tratar de incorporarlos al sistema actual.
* Crear una cuenta en GitHub para la Agencia Cubana de la Música que permita mantener la base de datos y el código fuente de la aplicación en un lugar seguro, para poder restaurar el sistema en caso de corromperse la información.

**Trabajo futuro**

* Rediseñar la interfaz gráfica de la aplicación.
* Al momento de presentar este proyecto, el Catálogo de Música Tradicional Cubana, aún no está listo, por tanto, una vez listo e insertado el catálogo por los especialistas en el sistema, deberá realizarse una fase de prueba para los módulos que intervienen con esta nueva información.
* Incorporar una funcionalidad al sistema para ofrecer los códigos ISMN en formato QR.

# **Glosario de siglas y términos**

**ISMN:** International Standard Music Number, o en español, Número Internacional Normalizado para la Música (Música escrita o Música impresa) identifica a todas las partituras musicales impresas publicadas ya estén disponible para la venta, su alquiler o gratuitamente.

**ISBN:** Es un identificador internacional único para publicaciones monográficas (libros, e-books, mapas, audiolibros); la asignación de un número sustituye a la creación de largos registros de descripción bibliográfica y, por tanto, se ahorran tiempo y gastos de personal y se reducen los errores de copia.

**Frameworks:** Son paquetes de clases y funcionalidades que ayudan en la construcción de aplicaciones informáticas. Intentan aliviar el exceso de carga asociado con actividades comunes empleadas para el desarrollo de aplicaciones; por ejemplo, accesos a bases de datos, manejo de sesiones, construcción de plantillas. Permiten la construcción de sistemas escalables mediante la reutilización de código y emplean patrones de diseño consistentes y modernos.

**CCL:** La Cámara Cubana del Libro es la institución encargada de representar a las entidades editoriales, así como a las personas jurídicas y naturales, que conforman el sector del libro y de las publicaciones en la República de Cuba, que voluntariamente se adhieran a ella y cumplan su Reglamento ante otras Cámaras Nacionales del Libro, organismos regionales e internacionales del libro y los Comités Organizadores de Ferias Internacionales. Convoca, organiza y gestiona la Feria Internacional del Libro de la Habana. Forma parte del sistema de instituciones adscriptas al Instituto Cubano del Libro.

**MINCULT: Ministerio de Cultura**. Órgano de la Administración Central del Estado de la República de Cuba, encargado de dirigir, orientar, controlar y ejecutar en el ámbito de su competencia la aplicación de la política cultural del Estado y del Gobierno cubano, así como garantizar la defensa, preservación y enriquecimiento del patrimonio cultural de la nación cubana.

**Prefijo (Editor):** Parte numérica del código ISMN que asigna un número único a cada editor o editorial dentro de un área específica. Este número permite identificar qué editor ha hecho la publicación musical. Varía en longitud dependiendo del número de publicaciones musicales que tenga o necesite publicar el editor, para editores que generan un gran número de publicaciones se les asignan números más cortos. Su función principal es facilitar la catalogación y distribución de los ISMN.

**Prefijo (Publicación):** El prefijo de una publicación en un ISMN es una parte del código que identifica de manera única cada título o edición específica publicada por una editorial o editor independiente. Este número es asignado internamente por el software en relación al número de publicaciones que tiene hasta el momento el editor y varía en longitud, si el prefijo identificador del editor es pequeño entonces el prefijo identificador de la publicación será de mayor longitud. Sirve para distinguir entre diferentes músicas notadas o ediciones dentro del catálogo del mismo editor. Su principal función es asegurar que cada publicación musical tenga un identificador exclusivo.

**Rango de un prefijo:** Los rangos de un prefijo son rangos numéricos del "prefijo identificador de un editor" en un ISMN, son conjuntos de números asignados a diferentes áreas geográficas, países o regiones lingüísticas, determinados por la Agencia Internacional del ISMN. Cada rango define el espacio numérico dentro del cual se pueden asignar prefijos identificadores a los editores en esa área específica. Estos rangos garantizan que los prefijos identificadores de editores no se repitan entre diferentes regiones y que cada editor dentro de una región tenga un identificador único.

# **Referencias**

[1] «International ISMN Agency». Accedido: 15 de mayo de 2024. [En línea]. Disponible en: https://www.ismn-international.org/agency

[2] J. Coscia, H. González, y E. Escalada, «Manual del Usuario Edición en español 2013».

[3] «Agencia Cubana del ISBN». Accedido: 27 de febrero de 2023. [En línea]. Disponible en: https://www.claustrofobias.com/catalogo/agencia-cubana-del-isbn/, https://www.claustrofobias.com/catalogo/agencia-cubana-del-isbn/

[4] «Misión y visión | Feria del Libro de la Habana», Cámara Cubana del Libro. Accedido: 15 de mayo de 2024. [En línea]. Disponible en: http://www.filhcuba.cu/mision-y-vision

[5] P. R. González y J. G. Sopeña, «ESTUDIO DE LA APLICACIÓN DE METODOLOGÍAS ÁGILES PARA LA EVOLUCIÓN DE PRODUCTOS SOFTWARE».

[6] R. S. Pressman, «Ingenieria del Software. Un Enfoque Practico».

[7] I. Jacobson, *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. 1999.

[8] «Metodologia Rup». Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://metodolorup.blogspot.com/2017/10/principios-o-funcionalidad-de.html

[9] «scrum\_master.pdf». Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://www.scrummanager.com/files/scrum\_master.pdf

[10] R. R. E. E. A. para crear los diagramas que se van generando durante el proceso de I. en el D. del S. D. por R. Corporation, «Rational Rose Enterprise Edition - EcuRed». Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://www.ecured.cu/Rational\_Rose\_Enterprise\_Edition

[11] P. Hudson, *PHP is a Nutshell*. O’Reilly Media, 2006.

[12] «Qué es Python». Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://desarrolloweb.com/articulos/1325.php

[13] U. Verdejo, «El lenguaje de Programación Java», Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://www.academia.edu/25663204/El\_lenguaje\_de\_Programaci%C3%B3n\_Java

[14] S. Suehring, *Step by step Javascript*, Third. O’Reilly Media, 2013.

[15] @Alvy, «Los lenguajes de programación más populares de 2020, según Spectrum/IEEE, analizados con 11 métricas de 8 fuentes distintas», Microsiervos. Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://www.microsiervos.com/archivo/ordenadores/lenguajes-programacion-mas-popularesx-2020.html

[16] «Django overview», Django Project. Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://www.djangoproject.com/start/overview/

[17] «laravel/framework». The Laravel Framework, 2 de mayo de 2023. Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://github.com/laravel/framework

[18] A. Jacinto, «Tendencias de desarrollo de la tecnología back-end a seguir en 2023», Startechup Inc. Accedido: 30 de abril de 2023. [En línea]. Disponible en: https://www.startechup.com/es/blog/back-end-technology-trends-2023/

[19] «Angular». Accedido: 30 de abril de 2023. [En línea]. Disponible en: https://angular.io/

[20] F. G. de Zúñiga, «Ventajas de React: las principales y más prácticas | Arsys», Blog de arsys.es. Accedido: 2 de mayo de 2023. [En línea]. Disponible en: https://www.arsys.es/blog/caracteristicas-reactjs

[21] «The Most Demanded Frontend Frameworks in 2022», Devjobsscanner. Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://www.devjobsscanner.com/blog/the-most-demanded-frontend-frameworks-in-2022/

[22] «PyCharm: the Python IDE for Professional Developers by JetBrains», JetBrains. Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://www.jetbrains.com/pycharm/

[23] TerryGLee, «Características avanzadas - Visual Studio (Windows)». Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://learn.microsoft.com/es-es/visualstudio/ide/advanced-feature-overview

[24] B. Hayes, «Most Popular Integrated Development Environments (IDEs) Used by Data Scientists |». Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://businessoverbroadway.com/2020/07/14/most-popular-integrated-development-environments-ides-used-by-data-scientists/

[25] «PostgreSQL vs MySQL: Explore Their 12 Critical Differences», Kinsta®. Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://kinsta.com/blog/postgresql-vs-mysql/

[26] «What is SQLite? And When to Use It?», Simplilearn.com. Accedido: 15 de mayo de 2024. [En línea]. Disponible en: https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite

[27] https://www.facebook.com/Twaino-565027893984018, «Servidor Apache: una guía completa para principiantes (2023)», Twaino. Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://www.twaino.com/es/blog/crear-un-sitio-web/servidor-apache-una-guia-completa-para-principiantes/

[28] S. Borges, «¿Qué es Nginx? Características, Ventajas e Instalación», Infranetworking. Accedido: 3 de mayo de 2023. [En línea]. Disponible en: https://blog.infranetworking.com/que-es-nginx/

[29] CUJAE, «Leccion 16-ANP-Modelo de objetos del negocio de Ingeniería de Software 1». 7 de enero de 2022.

[30] CUJAE, «Lección 22 ANP - Artefactos del Flujo Requisitos de Ingeniería de Software 1.»

[31] «Qué son los Patrones de Diseño de software / Design Patterns». Accedido: 12 de abril de 2023. [En línea]. Disponible en: https://profile.es/blog/patrones-de-diseno-de-software/

[32] «GRASP», *Wikipedia, la enciclopedia libre*. 3 de diciembre de 2020. Accedido: 12 de abril de 2023. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=GRASP&oldid=131422527

[33] «Patrones Gof - EcuRed». Accedido: 12 de abril de 2023. [En línea]. Disponible en: https://www.ecured.cu/Patrones\_Gof

[34] «What is django ORM?» Accedido: 8 de octubre de 2023. [En línea]. Disponible en: https://www.tutorialspoint.com/what-is-django-orm

[35] «What is Software Testing and How Does it Work? | IBM». Accedido: 8 de octubre de 2023. [En línea]. Disponible en: https://www.ibm.com/topics/software-testing

[36] «Black Box Vs White Box Testing». Accedido: 8 de octubre de 2023. [En línea]. Disponible en: https://www.practitest.com/resource-center/article/black-box-vs-white-box-testing/

[37] «Estimacion del esfuerzo basada en casos de usos (1) by Rocio Arevalo - Issuu». Accedido: 21 de mayo de 2024. [En línea]. Disponible en: https://issuu.com/rocio-arevalo/docs/estimacion-del-esfuerzo-basada-en-c

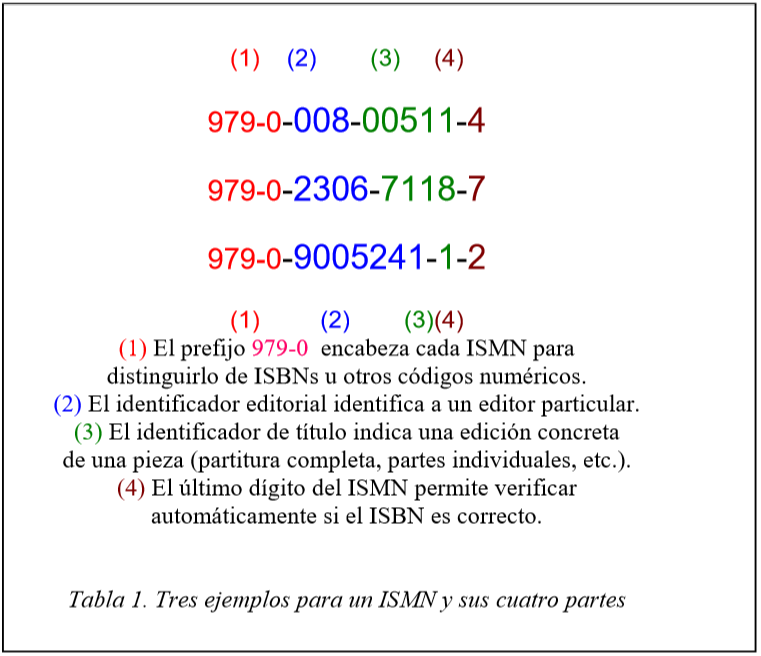
# **Anexos**

## Anexo1. ISMN listo



1. 1 Código de barras de un ISMN listo.

## Anexo2. Estructuración de los ISMN



1. Ejemplos de ISMN y su composición.

1. 7 [7] [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)
4. [7] [↑](#footnote-ref-4)
5. [30] [↑](#footnote-ref-5)