

Universidad de Matanzas
Facultad de Ciencias Técnicas
Departamento de Informática

Aplicación web para gestionar producciones musicales en la Radio Cubana

Trabajo de Diploma en Opción al Título de Ingeniero Informático

Autor: Omar Guerra Medina

Tutora: MSc. Mayli Estopiñan Lantigua

Matanzas, Cuba

Julio, 2020

Declaración de autoría

Yo Omar Guerra Medina, declaro que este documento es en su totalidad de mi autoría y autorizo a la Universidad de Matanzas Sede Camilo Cienfuegos a su uso en la forma que estime conveniente.

Para que así conste firmo la presente a los ____ días del mes de junio del año 2020.

Firma del Autor

Firma del Tutor

Resumen

La difusión de obras musicales por medios tradicionales como la radio es una de las mejores formas de realizar un producto comunicativo directo y preciso. El control del uso y el pago del derecho de autor son las dos razones que obligan a buscar una forma eficiente de realizar dichas funciones. La presente investigación está orientada al desarrollo de una aplicación web para gestionar las producciones musicales en la Radio Cubana con el objetivo de solucionar la falta de exactitud, seguridad y complejidad en la realización de los reportes, investigaciones, estadísticas y el pago a la Agencia Cubana de Derecho de Autor Musical (ACDAM). Al tratarse de una aplicación web es posible acceder desde cualquier estación de trabajo conectada a la Red Corporativa de la Radio, así como hacer uso de tecnologías alternativas como teléfonos inteligentes, tabletas, pantallas táctiles entre otras. Entre las principales herramientas que se utilizan destaca la metodología ágil de desarrollo de software **Scrum**, para el servidor se empleó el lenguaje de programación **Php** con el framework **Laravel**, de cara al cliente, **Vue** como principal marco de trabajo junto al lenguaje de programación **JavaScript**. Todas estas tecnologías posibilitan el desarrollo de una aplicación web intuitiva, rápida y segura, que facilita el trabajo de todos los participantes en ese proceso.

Palabras claves: Radio Cubana, Obra musical, Producción musical, Laravel, Vue, Scrum.

Abstract

The diffusion of musical pieces by traditional means such as the radio is one of the best ways to make a direct and precise communication product. Control of the use and payment of copyright are the two reasons why an efficient way of performing these functions must be found. The present research is oriented to the development of a web application to manage the musical productions in the Cuban Radio with the objective of solving the lack of accuracy, security and complexity in the accomplishment of the reports, investigations, statistics and the payment to the Cuban Agency of Musical Copyright (ACDAM). Among the main tools used is the agile methodology of software development Scrum, for the server was used the programming language **Php** combined with the **Laravel** framework, for the client was selected **Vue** as the main framework along with the programming language **JavaScript**. All these technologies allow the development of an intuitive, fast and secure web application, which facilitates the work of all those involved in the process.

Keywords: Radio Cubana, musical work, musical production, Laravel, Vue, Scrum.

Índice

Introducción	1
1. Capítulo I: Marco Teórico y Tendencias Tecnológicas	4
1.1. Conceptos básicos relacionados con el problema	4
1.2. Antecedentes del problema de investigación	4
1.3. Descripción del negocio.....	5
1.4. Justificación de la Investigación.....	7
1.6. Métodos de la investigación.....	8
1.7. Metodologías de desarrollo de software.....	8
1.7.1. Scrum.....	11
1.8. Tecnologías, frameworks y lenguajes de programación.....	12
1.8.1. Principales lenguajes de programación	12
1.8.2. Frameworks.....	15
1.8.3. BackEnd.....	16
1.8.4. FrontEnd.....	18
1.8.5. Gestor de base de datos.....	20
1.8.6. Herramientas	21
2. CAPÍTULO 2: Diseño y Desarrollo de la Solución Propuesta.....	22
2.1. Etapa de planificación.....	22
2.1.1. Definición del equipo	22
2.1.2. Historias de usuario	23
2.1.3. Pila del Producto (Product Backlog).....	34
2.1.4. Estimación de los Sprint	36
2.1.5. Plan de Entregas (Sprints Backlog)	38
2.2. Base de datos.....	43
2.3. Estudio de Costo	44
2.3.1. Método de puntos de función	44

3. CAPÍTULO 3: Validación del Trabajo Desarrollado	48
3.1. Descripción de la solución propuesta.....	48
3.2. Interfaces.....	51
Principales interfaces de la aplicación web.	51
3.3. Algoritmos.....	54
3.4. Pruebas de software	57
3.4.1. Pruebas de aceptación	57
Conclusiones	60
Recomendaciones	61
Referencias bibliográficas	62

Índice de Tablas

Tabla 1 Comparación de metodologías. Fuente: (Tinoco Gómez, 2010).....	9
Tabla 2: Equipo del proyecto. Fuente: Elaboración propia	22
Tabla 3: HU 1 - Diseño y creación de Base de Datos. Fuente: Elaboración propia	23
Tabla 4: HU 1 - Sistema de autenticación. Fuente: Elaboración propia	24
Tabla 5: HU 3 - Gestionar emisoras. Fuente: Elaboración propia.....	25
Tabla 6: HU 4 - Gestionar programas. Fuente: Elaboración propia	26
Tabla 7: HU 5 - Gestionar obras musicales. Fuente: Elaboración propia	26
Tabla 8: HU 6 - Gestionar producciones musicales. Fuente: Elaboración propia	27
Tabla 9: HU 7 - Gestionar listas organizativas. Fuente: Elaboración propia	29
Tabla 10: HU 8 - Gestionar permisos para aprobar reportes ACDAM. Fuente: Elaboración propia	29
Tabla 11: HU 9 - Gestionar reportes ACDAM. Fuente: Elaboración propia	30
Tabla 12: HU 10 – Realizar modelo clásico. Fuente: Elaboración propia	31
Tabla 13: HU 11 – Realizar estadística general. Fuente: Elaboración propia	32
Tabla 14: HU 12 – Realizar estadística por listas organizativas. Fuente: Elaboración propia	33
Tabla 15: HU 13 – Gestionar creadores. Fuente: Elaboración propia.....	33
Tabla 16: Pila de productos ordenada por prioridad e importancia. Fuente: Elaboración propia	35
Tabla 17: Tiempo para cada Sprint. Fuente: Elaboración propia.....	36
Tabla 18: Estimación del Sprint N° 1. Fuente: Elaboración propia.....	36
Tabla 19: Estimación del Sprint N° 2. Fuente: Elaboración propia.....	36
Tabla 20: Estimación del Sprint N° 3. Fuente: Elaboración propia.....	37
Tabla 21: Estimación del Sprint N° 4. Fuente: Elaboración propia.....	37
Tabla 22: Estimación del Sprint N° 5. Fuente: Elaboración propia.....	37
Tabla 23: Estimación del Sprint N° 6. Fuente: Elaboración propia.....	37
Tabla 24: Estimación del Sprint N° 7. Fuente: Elaboración propia.....	37
Tabla 25: Estimación del Sprint N° 8. Fuente: Elaboración propia.....	38
Tabla 26: Planificación de entregas Sprint 1. Fuente: Elaboración propia.....	38
Tabla 27: Planificación de entregas Sprint 2. Fuente: Elaboración propia.....	39
Tabla 28: Planificación de entregas Sprint 3. Fuente: Elaboración propia.....	39
Tabla 29: Planificación de entregas Sprint 4. Fuente: Elaboración propia.....	40
Tabla 30: Planificación de entregas Sprint 5. Fuente: Elaboración propia.....	40
Tabla 31: Planificación de entregas Sprint 6. Fuente: Elaboración propia.....	41
Tabla 32: Planificación de entregas Sprint 7. Fuente: Elaboración propia.....	41
Tabla 33: Planificación de entregas Sprint 8. Fuente: Elaboración propia.....	42
Tabla 34: Tabla de valores.....	44
Tabla 35: Tabla completa.....	44
Tabla 36: Factor de ajuste	45
Tabla 37. Prueba de aceptación 1. Fuente: Elaboración propia	58
Tabla 38. Prueba de aceptación 2. Fuente: Elaboración propia	58

Ilustración 1 Ciclo de vida de una producción musical. Fuente: Elaboración propia	6
Ilustración 2: Uso de lenguajes de programación. Fuente (Natalya Prokofyeva, 2017)	12
Ilustración 3 Frameworks para lado servidor más utilizados. Fuente: Google Trends	17
Ilustración 4: Comparativa frameworks para FrontEnd. Fuente: NPM Trends	18
Ilustración 5: Base de Datos	43

Introducción

El vertiginoso desarrollo de las tecnologías transforma el mundo de las comunicaciones. La difusión de obras musicales se convierte en uno de los principales canales para trasladar un mensaje preciso a los distintos destinatarios. La llamada autopista de la información constituye un reto para instituciones de la vida jurídica, entre ellas el Derecho de Autor, en el marco de las nuevas vías para socializar y divulgar contenidos.

El concepto, que lo integran normas jurídicas y principios, regula los derechos morales y patrimoniales que la ley concede a los autores, por el solo hecho de la creación de una obra literaria, artística, musical, científica o didáctica, esté publicada o inédita.

De acuerdo con el criterio de especialistas lo que más preocupa al mundo autoral, está en enfrentar y ofrecer soluciones a los problemas que se generan con el uso combinado de la tecnología digital y las telecomunicaciones.

Cuba contempla y protege en su legislación, la Ley 14 de 1977 (Calderio, 1977), este derecho especificando los deberes y obligaciones de los sujetos, así como el alcance y las limitantes en la reproducción y divulgación de las obras.

Igualmente funcionan entidades como la Agencia Cubana de Derecho de Autor Musical (ACDAM) (ACDAM, 2020) que tiene como misión proteger los intereses patrimoniales de los autores, compositores, editores y demás titulares en el campo de la música y las artes escénicas, a través de la recaudación y distribución de los derechos de autor que se generan a propósito de la explotación de las obras en todo el territorio nacional y en el extranjero por medio de las sociedades de autores con las cuales se han suscrito contratos de representación recíproca.

La ACDAM se interrelaciona con organismos como el Instituto Cubano de Radio y Televisión y controla la transmisión de la música en las 100 emisoras: 1 internacional, 6 nacionales, 19 provinciales y 74 municipales para el pago de los derechos a los correspondientes autores e intérpretes.

En Cuba se transmiten mensualmente cerca de un millón de canciones, por cada obra difundida se les retribuye a sus autores e intérpretes nacionales y extranjeros. Tal proceso implicaría responsabilidad, veracidad y exactitud porque se pueden lesionar derechos. Sin embargo, el procedimiento de control de la gestión musical se realiza analógicamente por las especialistas de música y las fonotecarias en cada una de las emisoras a través de un modelo ofrecido por la ACDAM.

No existe un sistema informático que supla todas las necesidades para la gestión de las producciones musicales en el país, que interrelacione todas las emisoras y que ofrezca una estadística detallada de la cantidad de obras difundidas por períodos de fechas, el balance entre música cubana y extranjera, segmentación por áreas geográficas y programas, y los autores, intérpretes y géneros divulgados, entre otros.

La necesidad de este tipo de programa responde a las propias prioridades del Instituto Cubano de Radio y Televisión (ICRT) (ICRT, 2020) y la agenda de la política nacional de automatizar diversos procesos como señaló el Presidente de los Consejos de Estado y de Ministros, Miguel Díaz Canel Bermúdez durante la clausura del Evento Nacional de la Unión de Informáticos de Cuba. (ACN, 2018)

Por todo lo antes descrito se plantea el **problema científico** a resolver:

¿Cómo contribuir a la eficiencia en la gestión de las producciones musicales en la Radio Cubana?

Como **Objeto de Estudio** se identifica la gestión de las producciones musicales en los medios de difusión y se define como **Campo de Acción** la informatización de las producciones musicales en la Radio Cubana.

El **Objetivo General**: Desarrollar una aplicación web que contribuya a la eficiencia en la gestión de producciones musicales en las emisoras de la Radio Cubana.

Como **Hipótesis** se plantea:

Al informatizar el proceso de gestión de producciones musicales en la Radio Cubana se ganará en exactitud y confiabilidad de los datos. Aumentará las facilidades para realizar estadísticas, reportes y mejorará el control de las obras radiadas.

Para dar respuesta se definen los **Objetivos Específicos**:

- Revisar los referentes teóricos y metodológicos del proceso actual de gestión de producciones musicales y los antecedentes informáticos.
- Seleccionar las herramientas y metodologías acordes para el diseño y desarrollo de la aplicación web.
- Desarrollar la aplicación web para la gestión de producciones musicales en la Radio Cubana
- Validar la solución mediante la realización de pruebas tecnológicas y que cumpla con las especificaciones requeridas por el cliente.

El **Resultado de esta investigación** se define en el desarrollo de una aplicación web donde se realice una unificación de las todas las fonotecas que componen el sistema de la Radio Cubana. Un sistema único que registre todas las producciones musicales, permitió realizar un control eficiente, que permita realizar investigaciones y estadísticas complejas, así como el correcto pago a la ACDAM.

Esta investigación se estructura de la siguiente manera:

- **Capítulo 1: Marco Teórico Referencial:** se plantean las definiciones fundamentales asociadas al campo de acción. Se hace un estudio sobre los antecedentes, las tendencias y tecnologías que serán usadas. Se exponen las características fundamentales de los lenguajes de programación, los gestores de bases de datos y las características fundamentales de las metodologías ágiles de desarrollo de software.
- **Capítulo 2: Diseño y Desarrollo de la Solución Propuesta:** se argumenta la solución que se propone al problema de investigación, se presenta una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software SCRUM. Se desarrolla la solución propuesta.
- **Capítulo 3: Validación de la Solución Propuesta:** se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio de los clientes y los propios del desarrollo de software.

1. Capítulo I: Marco Teórico y Tendencias Tecnológicas

Introducción

En este capítulo se realiza la fundamentación teórica del trabajo, se explican los conceptos básicos relacionados con las producciones musicales en la Radio Cubana, se analizan los antecedentes de esta investigación, así como la justificación de la misma. Además, se realiza un estudio de algunas de las tendencias tecnológicas actuales y un breve análisis de las herramientas utilizadas en el desarrollo de esta aplicación web para la gestión de las producciones musicales en la Radio Cubana.

1.1. Conceptos básicos relacionados con el problema

Obra musical completa:

Una canción es una composición musical para la voz humana, con letra y comúnmente acompañada por instrumentos musicales (Latham, 2001)

Se define como obra musical completa cuando se utiliza la totalidad de una canción tal y como está inscrita en los registros de derecho de autor.

Fragmento de obra:

El uso parcial de una obra musical al momento de ser emitida en un espacio determinado, un corte musical, una cortina. (Hernández, 2011)

Producción musical:

Es un documento escrito por el director de un programa de radio que recoge la lista de obras musicales y fragmentos de obras con el objetivo de ser utilizadas dentro de un espacio radial.

Derecho de autor:

Es el conjunto de derechos garantizados jurídicamente como estímulo al desarrollo y ampliación de la creación artística, literaria y científica. (Calderio, 1977)

1.2. Antecedentes del problema de investigación

Con el objetivo de investigar la existencia de aplicaciones informáticas o formas de gestión que respondiesen a la idea que se plantea en esta investigación, se consultaron distintas fuentes de información.

En el ámbito internacional no se encuentran aplicaciones informáticas destinadas a realizar específicamente el proceso que se aborda en este documento.

Por otra parte, en el ámbito nacional se han desarrollado las siguientes aplicaciones:

- **SIS2001** fue una solución informática para gestionar las obras musicales desarrollada en el año 2001 por la Radio Cubana, la misma funciona en consola, con muchas limitaciones y se utilizó hasta el año 2015 por la mayoría de las emisoras del país, aún existen algunas emisoras que lo utilizan.
- **Remus**, es el nombre de una aplicación de escritorio desarrollada por ACDAM en el año 2015 para uso de la Radio Cubana. Con este sistema se suplían solo las necesidades de la ACDAM y no permitía realizar investigaciones y estadísticas necesarias para la Radio, su mayor dificultad es la inserción de datos pues usaba un sistema basado en nomencladores que exige insertar por separado los elementos de la obra musical para después introducirla, requiere mucho tiempo y no es posible en la dinámica de una emisora.
- A raíz de los primeros intentos de implementación de la aplicación Remus en la emisora provincial de Matanzas Radio 26 y con la ayuda de los trabajadores que atienden el área de la fonoteca se realizó una captura de requisitos y problemáticas encontradas en dicha aplicación. En el febrero del año 2017 el autor de esta investigación desarrolla una solución informática titulada **SGDAM** que eliminó todos los inconvenientes que tenía su predecesor. Como elemento distintivo implementó un mecanismo para reconocer una producción musical en formato Word con solo arrastrarla desde el explorador hacia un panel destinado a ese fin. En su momento estaba definido dentro de las políticas de la Radio Cubana que la computadora de la fonoteca perteneciera a la red de trabajo y esta debía estar aislada por razones de seguridad informática. Con el tiempo se demostró que de esta forma resultaba poco exacta la información pues cada emisora interpreta los datos a su manera y en el caso de las emisoras provinciales, deben reportar la información teniendo en cuenta los reportes de las emisoras municipales, que, al no coincidir, nunca se logró un resultado real y exacto en la gestión de las producciones musicales. Tampoco fue posible tener servidor central de obras musicales ni realizar investigaciones y estadísticas que involucren múltiples emisoras.

1.3. Descripción del negocio

El ciclo de vida de una producción musical en una emisora de radio sigue los siguientes pasos de acuerdo al manual de calidad.

- El director de un programa de radio escribe de su propia inspiración una producción musical.
- La producción musical debe ser entregada y aprobada por una especialista de música con tres días de antelación
- De ser aprobada, es utilizada en el programa de radio para la que fue escrita.
- La producción musical resultante luego de ser transmitida es almacenada y procesada en la fonoteca de dicha emisora
- Se pagan los tributos por cada obra musical emitida a la ACDAM

Ilustración 1 Ciclo de vida de una producción musical. Fuente: Elaboración propia



La forma de ser procesada varía con respecto a la emisora. El procesamiento consiste en extraer información de las producciones musicales como, por ejemplo:

- Cantidad de obras cubanas y cantidad de obras extranjeras
- Cantidad de veces que se radia un autor específico o un intérprete
- Obras musicales más difundidas
- Cantidad de obras por zonas geográficas
- Cantidad de obras de un género musical determinado

Procesamiento 1: se realiza un conteo a mano de cada una de las obras musicales. Cada vez que se requiere una nueva información es necesario volver a realizar el proceso

Procesamiento 2: se utilizan hojas de cálculo en Microsoft Excel como apoyo.

Procesamiento 3: se vierten los datos en la aplicación de consola SIS2001 y con esta es posible realizar estadísticas muy limitadas, este proceso debe ser realizado cada cierto periodo de tiempo, máximo 6 meses pues el software presenta dificultades en la gestión de la base de datos por las limitaciones de la tecnología utilizada.

Procesamiento 4: se introducen diariamente las producciones musicales a la aplicación de escritorio SGDAM. Esta realiza todas las estadísticas e informaciones requeridas de forma local. Es la forma de procesamiento más utilizada en la Radio Cubana.

1.4. Justificación de la Investigación.

En una emisora de Radio Provincial se difunden cerca de 120 obras musicales diarias, cada una de ellas tiene un nombre, un conjunto de autores, intérpretes, géneros, voz predominante, entre otros elementos. Toda esta información se procesa para poder obtener estadísticas como ¿cuál es la obra más difundida en el mes? En el estado actual se complica realizar estadísticas cuando concurren, varios tipos de programas, distintas emisoras, así como la autoría del tema o el intérprete, ejemplo ¿cuál es la obra más difundida en el territorio occidental?

En la actualidad cada emisora tiene su propia base de datos de obras musicales, que en la mayoría de los casos no coinciden sintácticamente nombres, colaboración de intérpretes y en muchos casos los autores son desconocidos. El proceso actual de inserción de datos es muy engorroso, requiere teclear toda la información una a una lo que incurre en errores humanos y en el pago a la Agencia Cubana de Derecho de Autor.

Con esta investigación se persigue unificar todas las bases de datos y que pueda existir colaboración entre todas las emisoras para perfeccionar la información correspondiente a cada obra. Con un sistema informático que autocomplete, que use reconocimiento de patrones y verifique la información de los datos se espera contribuir en la eficiencia y seguridad de la información, lo que aumentará la velocidad con la que se introducen, así como la calidad de la misma. Entre los resultados también se espera realizar estadísticas e investigaciones más complejas y rápidas, tener un mayor control de las producciones musicales, así como evaluar la calidad de las mismas.

1.5. Resultados de la Investigación.

Con esta investigación se contribuye a disminuir las insuficiencias en la gestión de producciones musicales en la Radio Cubana. Aumenta la eficiencia de este proceso y la exactitud en la fiabilidad de los datos generados por el mismo, así como los reportes realizados a la ACDAM. Se espera desarrollar una solución informática capaz de realizar estadísticas profundas sobre el comportamiento de la música, verificar y evaluar la calidad de las producciones musicales e identificar aquellas de mala calidad para mejorar el producto radiofónico. Con este sistema informático se automatiza parte del trabajo de las fonotecarias del

país, de las especialistas de música y coordinadores musicales y se fomenta la colaboración entre emisoras. Se logra un servidor central de música, **un anhelo de la Radio Cubana para centralizar y enriquecer el patrimonio musical del medio de comunicación.**

1.6. Métodos de la investigación.

Los métodos de investigación son las herramientas que los investigadores utilizan para obtener y analizar los datos. Estas incluyen el muestreo, los cuestionarios, las entrevistas, los estudios de casos, el método experimental, los ensayos y grupos de enfoque. La elección del método de investigación es predeterminada por el problema a resolver y por los datos que se pueden obtener. Así, tenemos métodos de investigación cuantitativos, cualitativos o mixtos. (Halcomb, 2015)

Los métodos utilizados para esta investigación se definen a continuación:

Métodos Teóricos:

- Análisis histórico-lógico: permitió estudiar la trayectoria y desarrollo del proceso de gestión de las producciones musicales en la Radio Cubana
- Analítico-sintético: permitió el análisis de la bibliografía y los resultados.
- Hipotético-deductivo: facilitó la deducción de mejores formas para la gestión de las producciones musicales en la Radio Cubana.

Métodos Empíricos:

- Observación: facilitó el entendimiento del proceso de gestión de las producciones musicales en la Radio Cubana
- Entrevista: Permitted el levantamiento de requisitos y en la comprensión del objeto de estudio.
- Análisis de documentos: se realizó un estudio de todos los documentos estadísticos, modelos ACDAM, y normas definidas por la Radio Cubana para la gestión de las producciones musicales. Así como la documentación necesaria para realizar la investigación.

1.7. Metodologías de desarrollo de software

El desarrollo de software no es una tarea sencilla, por mucho tiempo esta labor se ha llevado adelante sin una metodología definida. Al respecto algunos autores definen una metodología como una colección de

procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información.

Avison y Fitzgerald (1995) presentan una definición de las metodologías de desarrollo clara que destaca sus principales componentes, fases, herramientas y técnicas. Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información.

Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. (Tinoco Gómez, Rosales López, & Bacalla, 2010)

Tabla 1 Comparación de metodologías. Fuente: (Tinoco Gómez, 2010)

Metodologías Agiles	Metodologías tradicionales
Se basan en heurísticas provenientes de prácticas de producción de código	Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo
Preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente por el equipo	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso muy controlado, numerosas normas
Contrato flexible e incluso inexistente	Contrato prefijado
El cliente es parte del desarrollo	Cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10)	Grupos grandes
Pocos artefactos	Más artefactos
Menor énfasis en la arquitectura del software	La arquitectura del software es esencial

Basado en estas definiciones se decide utilizar para desarrollar la solución propuesta las metodologías ágiles.

Manifiesto Ágil:

La Alianza Ágil elaboró un conjunto de doce principios comunes a las metodologías ágiles de desarrollo que se enuncian a continuación:

- Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
- Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos ágiles promueven el desarrollo sostenido. Los promotores, desarrolladores y usuarios debemos mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.

La utilización de todas las buenas prácticas enumeradas en el manifiesto ágil no implica ser ágil, sin embargo, el hecho de incumplir una de ellas te transforma en no ágil. (Pérez Pérez, 2012)

1.7.1. Scrum

En Scrum un proyecto se ejecuta en bloques temporales (iteraciones-sprints) de un mes natural (pueden ser de dos o tres semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite.

El Sprint es el ritmo de los ciclos de Scrum. Está delimitado por la reunión de planificación del sprint y la reunión retrospectiva. Una vez que se fija la duración del sprint es inamovible. La mayoría de los equipos eligen dos, tres o cuatro semanas de duración. Diariamente durante el sprint, el equipo realiza una reunión de seguimiento muy breve. Al final del sprint se entrega el producto al cliente en el que se incluye un incremento de la funcionalidad que tenía al inicio del sprint.

El proceso parte de la lista de requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente ha priorizado los requisitos balanceando el valor que le aportan respecto a su coste y han sido divididos en iteraciones y entregas. (Pérez Pérez, 2012)

Los pilares de la metodología Scrum más importantes son:

- Transparencia

Con el **método Scrum** todos los implicados tienen conocimiento de qué ocurre en el proyecto y cómo ocurre. Esto hace que haya un entendimiento “común” del proyecto, una visión global.

- Inspección

Los miembros del equipo Scrum frecuentemente inspeccionan el progreso para detectar posibles problemas. La inspección no es un examen diario, sino una forma de saber que el trabajo fluye y que el equipo funciona de manera auto-organizada.

- Adaptación

Cuando hay algo que cambiar, el equipo se ajusta para conseguir el objetivo del sprint. Esta es la clave para conseguir el éxito en proyectos complejos, donde los requisitos son cambiantes o poco definidos y en donde la adaptación, la innovación, la complejidad y flexibilidad son fundamentales. (Abellán, 2020)

Luego de investigar sobre las distintas metodologías ágiles dispuestas para el desarrollo de software se define Scrum como metodología principal para el desarrollo de la solución propuesta por esta investigación.

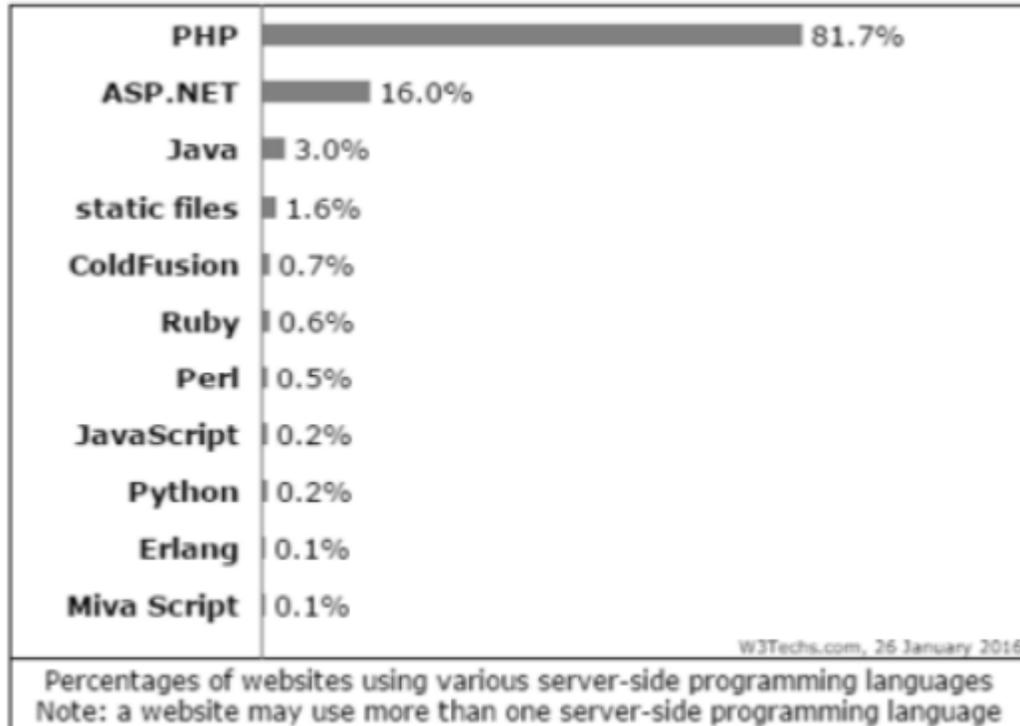
1.8. Tecnologías, frameworks y lenguajes de programación

1.8.1. Principales lenguajes de programación

JavaScript (JS): es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB and Adobe Acrobat. JavaScript es un lenguaje de programación basado en prototipos, multi-paradigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional). (MDN, 2020)

Antes de seleccionar cual lenguaje se utilizaría para el lado del servidor se consultaron distintas fuentes, entre ellas la principal fue el artículo escrito por (Natalya Prokofyeva, 2017) Donde se realiza la siguiente comparación:

Ilustración 2: Uso de lenguajes de programación. Fuente (Natalya Prokofyeva, 2017)



PHP: (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular. Es **interpretado**, se ejecuta en el **servidor** y se puede incrustar en **HTML**. Se utiliza principalmente en **desarrollo web**, para el que es especialmente adecuado, aunque tiene otros usos. (Costa, 2013)

“Más de 250 millones de sitios web se realizaron en PHP, entre ellos algunos tan famosos como: Facebook. com, Yahoo INC, Wikipedia. org, Friendster. com, Digg. com, Sourceforge. org, Flickr. com, Meneame. net, etc. Y no únicamente lo utilizan las grandes” (Muñoz, 2013)

Principales ventajas:

- Lenguaje totalmente libre y abierto.
- Posee una curva de aprendizaje muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil de instalar
- Fácil acceso e integración con la base de datos.
- Posee una comunidad muy grande.
- Es el lenguaje con mayor usabilidad en el mundo.
- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas y/o páginas web con acceso a una Base de Datos.
- El código escrito en PHP es invisible al navegador ya que se ejecuta al lado del servidor y los resultados en el navegador es HTML.

CSS: es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML. (Pérez Pérez, 2012)

SASS: (Syntactically Awesome Stylesheets) es un lenguaje de hoja de estilos inicialmente diseñado por Hampton Catlin y desarrollado por Natalie Weizenbaum. Después de sus versiones iniciales, Nathan Weizenbaum y Chris Eppstein han continuado extendiendo Sass con SassScript, un lenguaje de script simple, usado en los ficheros Sass. La extensión de un fichero sass es .scss

“Sass hace que CSS sea divertido nuevamente. Sass es una extensión de CSS, que agrega reglas anidadas, variables, mixins, herencia de selector y más. Se traduce a CSS estándar bien formateado con la herramienta de línea de comandos o un complemento para su sistema de compilación.” (Weizenbaum, 2020)

Principales ventajas:

- Reduce el tiempo para crear y mantener el CSS.
- Permite tener una organización modular de los estilos, lo cual es vital para proyectos grandes.
- Proporciona estructuras avanzadas propias de los lenguajes de programación, como variables, listas, funciones y estructuras de control.
- Permite generar distintos tipos de salida, comprimida, normal o minimizada, tanto en desarrollo como en producción, además se hace una forma muy fácil.
- Permite vigilar los ficheros, de tal manera que, si ha habido un cambio en la hoja de estilos, se regenera de automáticamente (modo watch).
- Tiene muy pocas dependencias, sobre todo la nueva versión, que es dart-sass. En las anteriores versiones se dependía de muchas librerías de Ruby y era un poco engorroso de instalar, pero con la nueva versión, la instalación es muy fácil.
- Existen muchas herramientas asociadas, muchas librerías hechas con Sass y una comunidad muy importante de usuarios.

Bootstrap-vue: Con BootstrapVue, se pueden crear proyectos receptivos, aptos para dispositivos móviles y accesibles para ARIA en la web utilizando Vue.js y la biblioteca de front-end CSS más popular del mundo: Bootstrap v4.

“Con más de 85 componentes, más de 45 complementos disponibles, varias directivas y más de 670 íconos, BootstrapVue proporciona una de las implementaciones más completas del sistema de cuadrícula y componentes Bootstrap, al momento de realizar esta investigación la versión disponible es la v4.5 para Vue.js v2.6, completo con marcado automatizado de accesibilidad WAI-ARIA.” (bootstrap-vue, 2020)

HTML: significa Lenguaje de Marcado de Hipertextos (HyperText Markup Language), es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web.

Otras tecnologías además de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).

"Hipertexto" se refiere a los enlaces que conectan las páginas web entre sí, ya sea dentro de un mismo sitio web o entre diferentes sitios web. los vínculos son un aspecto fundamental de la web. Al subir contenido a Internet y vincularlo a páginas creadas por otras personas, te haces participante activo en la red mundial (World Wide Web). (MDN, 2020)

1.8.2. Frameworks

El termino framework se define como un conjunto de herramientas, librerías, convenciones y buenas prácticas que pretenden unir las tareas repetitivas en módulos fácilmente reutilizables. Además se menciona que un framework es un conjunto de componentes que aporta mucho a los lenguajes de programación con el desarrollo de aplicaciones o sistemas web.

Los objetivos principales que se obtiene al usar un framework son los siguientes: acelerar el desarrollo, permitir reutilizar y difundir las buenas prácticas de desarrollo. (Gutierrez, 2017)

Objetivos de los Frameworks Web

Entre los principales objetivos de los frameworks web están (Saavedra, 2009)

- Tener un rápido desarrollo.
- Permitir reutilizar código.
- Difundir las buenas prácticas de desarrollo gracias al uso de patrones de diseño.
- Disminuir el esfuerzo en el desarrollo.
- Tener como aliado a las metodologías de desarrollo Ágiles como XP, Scrum, AD y otros

Características de los Frameworks Web.

Los frameworks presentan características que los identifica, entre las principales están: (Saavedra, 2009).

- Abstracción de urls y sesiones.
- Acceso a datos.
- Controladores.
- Autenticación y control de acceso.

- Separación entre diseño y estilo.

Ventajas de usar Frameworks Web.

Al usar un framework, la velocidad de desarrollo aumenta, así como la productividad del desarrollo web, todo esto también va en relación al conocimiento de los aspectos básicos de utilización que presenta un framework web. (Bach.Llatas Yovera, 2017)

La reducción de costos es otra ventaja con respecto al trabajo que se debe realizar, ya que un framework permitirá al desarrollador dar prioridad a los aspectos esenciales que se debe realizar en un desarrollo web.

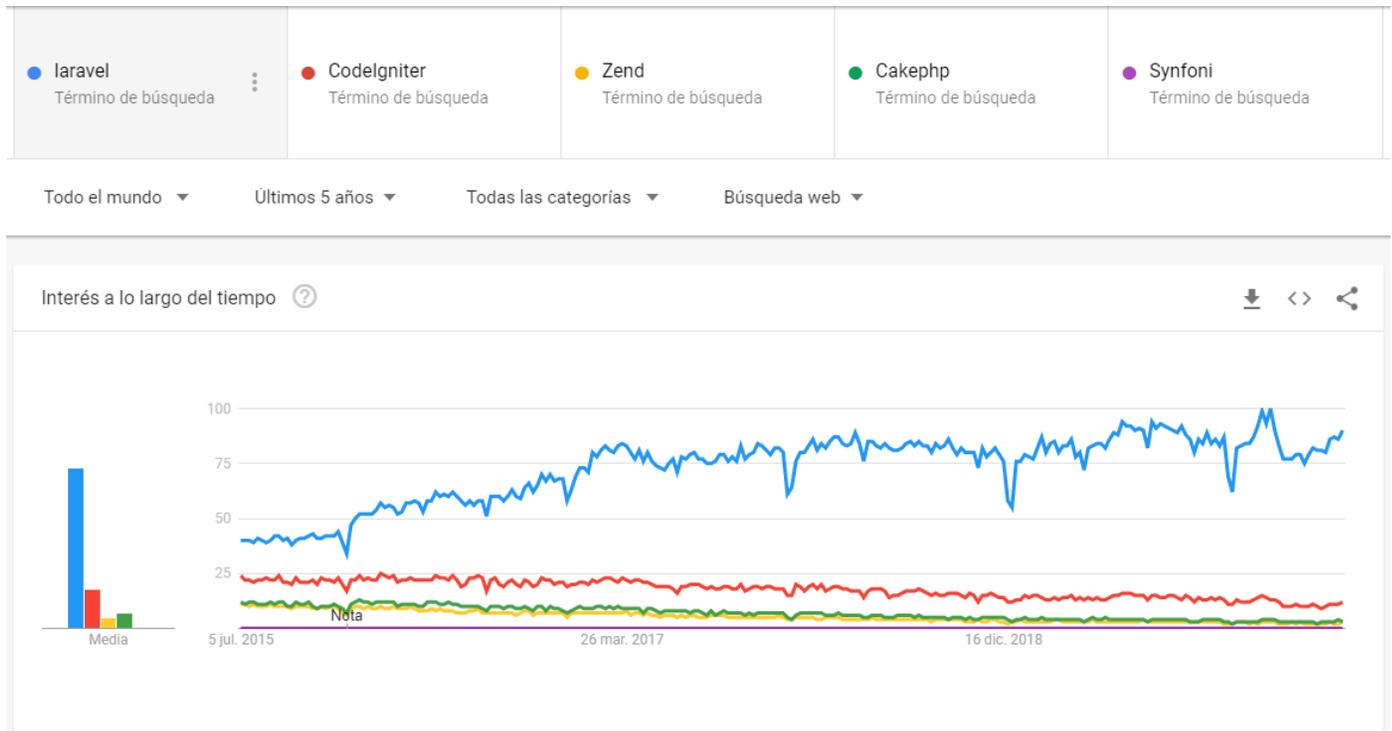
La utilización de estándares y convenciones de código de los frameworks, permite mejorarlas aplicaciones web al momento de realizar un trabajo en equipo, de esta manera permite una mejor comprensión del código y un trabajo más organizado. (Larrea, 2015)

1.8.3. BackEnd

El backend es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no se observan como, por ejemplo, la comunicación con el servidor.

Para decidir cuál framework utilizar en el desarrollo de la solución propuesta por esta investigación se utilizó la herramienta de tendencias por palabras del motor de búsqueda Google (Trends, 2020) y arrojó los siguientes datos:

Ilustración 3 Frameworks para lado servidor más utilizados. Fuente: Google Trends



1.8.3.1. Laravel

Laravel es un marco de trabajo (Framework) de aplicación web con sintaxis expresiva y elegante. Se cree que el desarrollo debe ser una experiencia agradable y creativa para ser verdaderamente gratificante. Laravel elimina el dolor del desarrollo al facilitar las tareas comunes utilizadas en muchos proyectos web, como:

- Motor de enrutamiento simple y rápido.
- Potente contenedor de inyección de dependencia.
- Múltiples back-end para sesión y almacenamiento en caché.
- Base de datos expresiva e intuitiva ORM.
- Migraciones de esquemas agnósticos de bases de datos.
- Robusto procesamiento de trabajos en segundo plano.
- Transmisión de eventos en tiempo real.

- Laravel es accesible, potente y proporciona las herramientas necesarias para aplicaciones grandes y robustas. (Otwell, 2020)

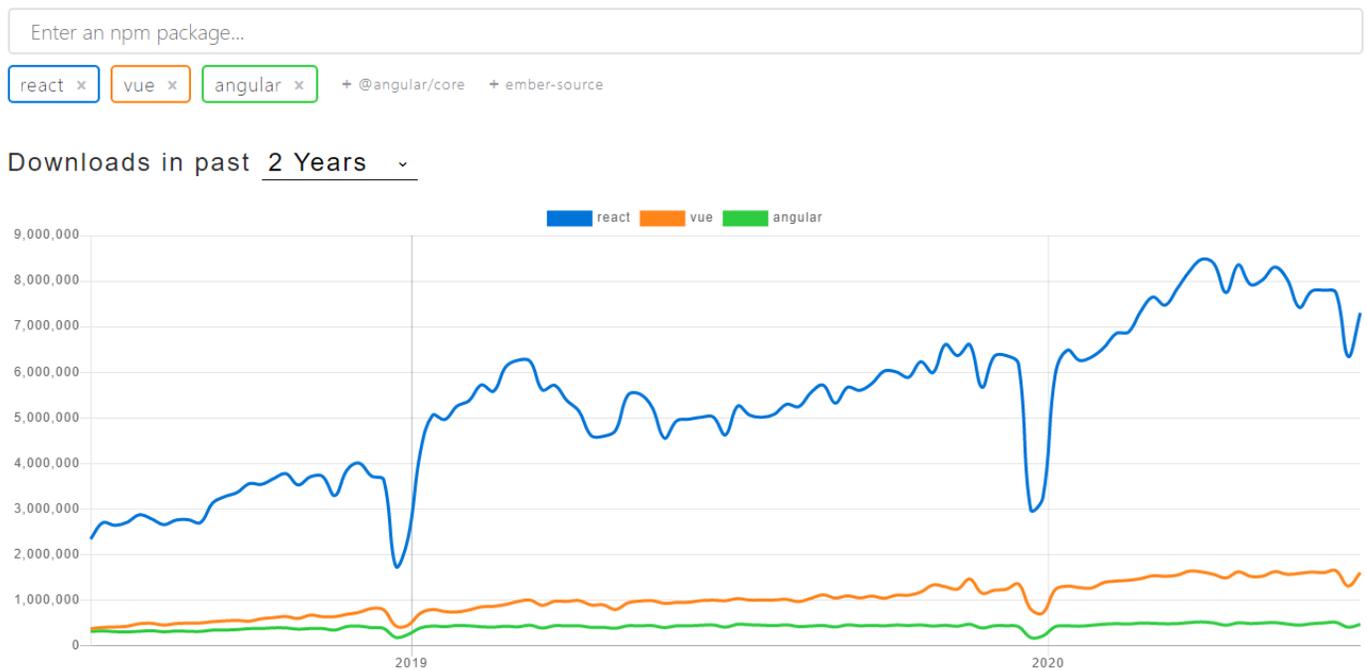
1.8.4. FrontEnd

El Frontend se enfoca en el usuario, en todo con lo que se puede interactuar y lo que se ve mientras se navega. Así como en una primera cita, la web busca causar una buena impresión y agradar al usuario, para lo cual utiliza HTML, CSS y JAVASCRIPT. Buena experiencia de usuario, inmersión y usabilidad, son algunos de los objetivos que busca un buen frontend y hoy en día existen una gran variedad de frameworks, preprocesadores y librerías que ayudarán en esta tarea. ¡Conocerlos y usarlos correctamente es el reto a cumplir! Para un frontend la creatividad es el recurso más valioso, ya que tendrá que tomar fuentes, colores, imágenes y todos los recursos de los cuales disponga para crear sitios agradables que se vean bien en todos los dispositivos y resoluciones. (Benites, 2017)

Para decidir cuál framework utilizar para el desarrollo del frontEnd en la solución propuesta por esta investigación se utilizó el sitio web NPM Trends (Potter, 2020) y se obtuvieron los siguientes datos:

Ilustración 4: Comparativa frameworks para FrontEnd. Fuente: NPM Trends

react vs vue vs angular



A partir de los datos reflejados en la ilustración anterior, se hizo necesario investigar entre estos frameworks cuál se ajusta mejor a laravel y cuáles son sus principales características.

React y Vue tienen mucho en común, ya que ambos son frameworks de JavaScript de UI enfocados únicamente a la parte front-end.

- **DOM virtual:** Una de las mayores similitudes entre Vue.js (a partir de la versión 2.0) y React es el uso de lo que se llama un 'DOM Virtual'. Un DOM virtual es una representación virtual del árbol DOM. Actualizar el DOM real frecuentemente es una tarea muy pesada. Sin embargo, actualizar objetos en JavaScript es relativamente ligero en comparación. Con un DOM virtual, se crea un objeto en JavaScript que representa el árbol DOM real. Las actualizaciones de cualquier elemento se realizan en el DOM virtual en lugar del DOM real. Cuando se cambia algo, se crea un nuevo objeto DOM virtual y se determinan los cambios entre lo antiguo y lo nuevo. Estos cambios se aplican entonces al DOM real.
- **Arquitectura basada en componentes:** Tanto React como Vue fomentan una arquitectura basada en componentes. Esto significa, la separación de la aplicación en distintos trozos de funcionalidad relacionada con una forma definida para cada trozo que forman la aplicación completa.

Principales diferencias:

- **Templating vs JSX:** La mayor diferencia entre React y Vue es cómo se hace la plantilla. En Vue, se usa el HTML tradicional para plantillas. Este enfoque aprovecha los atributos personalizados en elementos HTML estándar. Los atributos también se pueden utilizar en componentes de un solo archivo, aunque requiere un paso de compilación para convertir la sintaxis del componente en JavaScript y HTML válidos. Vue fomenta el uso de HTML para desarrollar, mientras que utiliza un método familiar parecido al de Angular para mostrar el contenido dinámico con la sintaxis de estilo "bigote" (`{ }`). Por esta razón, Vue es más fácil de integrar en las aplicaciones existentes. React por su parte recomienda que se escriba todo el código de plantilla en JavaScript usando una "extensión de sintaxis a JavaScript" llamada JSX. JSX es JavaScript con alguna sintaxis de XML. Sin embargo, una vez que te acostumbras a él, se hace mucho más flexible y robusto.
- **React Native:** Un área donde React tiene una ventaja es la capacidad de crear aplicaciones móviles nativas con React Native. La capacidad de usar JavaScript, CSS y HTML para crear aplicaciones

móviles nativas legítimas es un cambio de juego. La comunidad de Vue y el gigante de comercio electrónico Alibaba Group trabajan para crear la versión de Vue de React Native llamada Weex.

Cuál elegir.

Si se necesita un nuevo framework para la interfaz de usuario de JavaScript, cualquiera de los dos dará un rendimiento excepcional. React parece ser más popular, pero Vue está ganando adeptos rápidamente. Si lo que necesitas es integrar un nuevo framework de JavaScript en una base de código existente, elegiría Vue. (Delgado, 2017) Por otro lado, la curva de aprendizaje de vue es más pequeña. Vue es nativo compatible con laravel.

1.8.4.1. Vue

Vue (pronunciado /vju:/, como view) es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas cuando se utiliza en combinación con herramientas modernas y librerías de apoyo. (You, 2020)

- Accesible.
- Versátil, su núcleo es bastante pequeño y se escala a través de plugins.
- Escalable por el mismo tema de la versatilidad.
- Reactivo.
- Optimizado.
- Comunidad.
- Licencia MIT.

1.8.5. Gestor de base de datos

MySQL: se define como uno de los sistemas de bases de datos más populares en el desarrollo de aplicaciones web, la cual se caracterizan por su alto rendimiento para trabajar con bases de datos. Además, gran cantidad de aplicaciones web están desarrolladas con PHP y MySQL.

Es un sistema que representa a un Sistema Gestor de Base de Datos (SGBD), en donde se puede insertar las estructuras del catálogo, pero para la ejecución de sentencias de definición y manejo de datos difusos, se realizará en un módulo externo, el cual este será el que se ocupe de enviar las consultas a la base de datos en un lenguaje entendible, es decir en SQL para de esta manera obtener los resultados y dar formato a los datos de salida para que sean mostrados al usuario.

MySQL es la base de datos de código abierto más popular, ya que posee sus potentes funciones, además es fácil de configurar y usar. (Bach.Llatas Yovera, 2017)

1.8.6. Herramientas

Visual Studio Code: Es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity). (Microsoft, 2020)

Visual Paradigm: 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.

Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. (Comparacion de herramientas CASE, 2013)

Conclusiones Parciales

Con este Capítulo se obtuvieron los siguientes resultados:

Se definieron conceptos indispensables para la comprensión del objeto de estudio como obra musical completa. Se detalló el negocio tal y como funciona en la realidad, con esto se logró un correcto análisis entre el cliente y el equipo de trabajo. Se analizaron aplicaciones relacionadas al objeto de investigación determinándose que las mismas no se ajustan ni son de utilidad para resolver el problema planteado. Se definió Scrum como la metodología indicada para desarrollar la solución propuesta por esta investigación. Se definieron las tecnologías para el desarrollo de la aplicación web, laravel para el servidor y Vue para el cliente. Se selección MySql como sistema gestor de base datos. Se eligieron las principales herramientas a utilizar, entre ellas visual studio code.

En resumen, se obtiene una base teórica que permite continuar con el desarrollo de esta investigación.

2. CAPÍTULO 2: Diseño y Desarrollo de la Solución Propuesta.

Introducción

En el presente capítulo se realiza el proceso de análisis y diseño de la solución propuesta. Se realiza la descripción del sistema. Se definen los requisitos funcionales y no funcionales. Se desarrolla la etapa de planificación basada en la metodología ágil de desarrollo de software Scrum donde se recogen las Historias de usuario, se presenta el equipo de trabajo, se definen las entregas. Se muestra el diagrama de base de datos y se resumen los patrones utilizados en la solución propuesta. Se acomete un estudio de factibilidad para determinar si es factible el desarrollo de la aplicación.

2.1. Etapa de planificación

La etapa de planificación es la 1ra etapa del proyecto. Tiene como objetivo lograr una eficiente organización del prototipo inicial del problema y proporcionar un buen comienzo a una solución eficaz. Con este objetivo y según las ideas del cliente sobre el software se desarrollarán las Historias de Usuario, mediante la cual se obtendrá un punto de partida para el resto de la planificación del proyecto. De igual forma se realizará un estimado de cada una de las entregas del proyecto y del tiempo, que se basan en que la planificación inicial se podría afectar debido a cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto.

2.1.1. Definición del equipo

Los roles en la metodología Scrum son el Scrum Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, el Product Owner, que representa a los dueños o interesados por el producto final y el Scrum Team que incluye a los desarrolladores.

Tabla 2: Equipo del proyecto. Fuente: Elaboración propia

Rol	Miembro
Product Owner (Dueño del producto)	Directivos de la Radio Cubana
Scrum Team (Equipo de desarrollo)	Omar Guerra Medina
Scrum master (Lider de proyecto)	Omar Guerra Medina

2.1.2. Historias de usuario

Las historias de usuarios fueron desarrolladas por el equipo de trabajo durante el proceso de levantamiento de requisitos para la implementación del proyecto. Para la estimación de los datos se tomó los siguientes criterios:

- Prioridades en el Negocio (PN): Se medirá en función al rango de: Alta, Media y Baja, las cuales, serán asignadas por el Product Owner.
- Importancia del Desarrollo (ID): Se asignarán valores con ponderaciones del 1 al 100 entre el Product Owner y los miembros del equipo Scrum.
- Tiempo Estimado (TS): Se asignará con ponderaciones del 1 al 20 entre el Product Owner y los miembros del equipo Scrum.
- Programador responsable: Define dentro del equipo, quien o quienes son responsables de la historia
- Rol: Representa el o los usuarios responsables para realizar esa acción

Tabla 3: HU 1 - Diseño y creación de Base de Datos. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 1	Rol: Ninguno
Nombre Historia: Diseño y creación de Base de Datos	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 100
Tiempo Estimado: 10	Responsable: Omar Guerra Medina
Descripción: <ul style="list-style-type: none"> - Se creará el diseño de la base de datos y será definida por medio de migraciones con el ORM (mapeo objeto-relacional). Las tablas se definirán por orden de dependencias, la 1ra no tendrá dependencia con ninguna otra, las ultimas tablas están destinadas a ser las relaciones mucho a mucho del sistema, el nombre de estas será, por orden alfabético y en singular los nombres de las tablas que relaciona 	

Observaciones:

- Las tablas deben estar relacionadas en cascada por el Id, siempre que se cree una llave foránea debe tener un índice asignado. Todo el modelo de base de datos será escrito en el idioma inglés.

Tabla 4: HU 1 - Sistema de autenticación. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 2	Rol: Usuario autenticado
Nombre Historia: Sistema de autenticación	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 99
Tiempo Estimado: 8	Responsable: Omar Guerra Medina
Descripción: <ul style="list-style-type: none"> - El acceso al sistema se realizará por el nombre de usuario y no por un correo. Se utilizará una contraseña acorde con los elementos definidos en el apartado de seguridad definido en los requisitos no funcionales. - Al crear un nuevo usuario la contraseña por defecto será el carnet de identidad (CI) y siempre que se restaure la contraseña por un administrador esta volverá ser el CI - La primera vez que un usuario se autentique en el sistema se le sugerirá que cambie su contraseña y hasta que este no realice dicha acción se continuará notificando cada vez que ingrese al mismo. - Un administrador deberá ser capaz de crear, eliminar o modificar (CRUD) un usuario. - Los usuarios podrán ser filtrados por el nombre, la emisora a la que pertenecen y por los que están despedidos 	
Observaciones:	

- Si es eliminada una emisora, todos los usuarios que pertenezcan a ella perderán el derecho a la autenticación en el sistema.
- Los usuarios no se eliminan de la base de datos, se les despide y pueden ser recontractados.

Tabla 5: HU 3 - Gestionar emisoras. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 3	Rol: <ul style="list-style-type: none"> - Súper administrador - Administrador (Solo los administradores de emisoras marcadas como provincial)
Nombre Historia: Gestionar emisoras	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 95
Tiempo Estimado: 12	Responsable: Omar Guerra Medina
Descripción: <ul style="list-style-type: none"> - Realizará CRUD sobre las emisoras - Al crearse una nueva emisora automáticamente se deberá crear dos programas con ella, uno de continuidad y otro de transmisión especial - La lista de emisoras solo será posible visualizarla por usuarios que pertenezcan a una emisora provincial, por un supera administrador o por un director nacional de música. - Se deberá poder realizar CRUD para asociar fragmentos a la emisora, esta acción la podrán realizar solo los fonotecarios o especialistas de música y solo en su emisora. - Las emisoras podrán ser filtradas por el nombre y por las que fueron descontinuadas 	
Observaciones:	

- Las emisoras no se eliminan de la base de datos, se sacan del aire y pueden ser reintegradas
- Una emisora fuera del aire no permite que se creen nuevos programas ni usuarios para ella.

Tabla 6: HU 4 - Gestionar programas. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 4	Rol: <ul style="list-style-type: none"> - Jefe de programación - Director nacional de música
Nombre Historia: Gestionar programas	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 95
Tiempo Estimado: 14	Responsable: Omar Guerra Medina
Descripción: <ul style="list-style-type: none"> - Realizará CRUD sobre los programas radiales - Los programas podrán ser filtrados por el nombre, la emisora a la que pertenecen y por las que fueron descontinuados 	
Observaciones: <ul style="list-style-type: none"> - Los programas no se eliminan de la base de datos, son sacados de la parrilla de programación y pueden ser reintegrados 	

Tabla 7: HU 5 - Gestionar obras musicales. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 5	Rol:

	<ul style="list-style-type: none"> - Director nacional de música - Fonotecarios - Especialista de música - Usuarios marcados como colaboradores
Nombre Historia: Gestionar obras musicales	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 95
Tiempo Estimado: 20	Responsable: Omar Guerra Medina
Descripción: <ul style="list-style-type: none"> - Realizará CRUD sobre las obras musicales - Agregar obra permitirá agregar también todos elementos que componen una obra. - Agregar obra permitirá continuar agregando obras sin cerrar esa funcionalidad. - Agregar obra permitirá desde crear nuevo ACDAM, que sea creada la obra con todos sus elementos y se inserte al finalizar la creación de la misma. - Las obras musicales podrán ser combinadas con otras, esto mismo debe ocurrir con los autores y con los intérpretes. - Los obras musicales podrán ser filtradas por su nombre. 	
Observaciones: <ul style="list-style-type: none"> - Las obras musicales solo podrán ser eliminadas si no tienen ningún ACDAM asociado, de ser así la única opción será combinarlas. 	

Tabla 8: HU 6 - Gestionar producciones musicales. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 6	Rol:

	<ul style="list-style-type: none"> - Especialista de música - Fonotecario
Nombre Historia: Gestionar producciones musicales	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 95
Tiempo Estimado: 15	Responsable: Omar Guerra Medina
<p>Descripción:</p> <ul style="list-style-type: none"> - Se permitirá crear una nueva producción musical siempre y cuando no exista otra producción musical creada con el mismo programa y fecha de emisión asociado - Las producciones musicales podrán ser eliminadas o modificadas si aún no están asociadas a ningún reporte aprobado o no - Podrán ser creadas sin asociar obras musicales, pero no se tendrán en cuenta para realizar los reportes y serán eliminadas a partir de las próximas 24 horas a su creación. - Se podrán crear producciones musicales asociadas a otras producciones musicales, en estos casos no será editable la cantidad de obras musicales pues se toma de otra producción musical. Esto se conoce como reprís de programa. - Las producciones musicales podrán ser filtradas por el programa a que pertenecen, la emisora a que pertenecen, si están reportadas o no, y de estarlo por el identificador del reporte. - Obligatoriamente se deberá seleccionar un periodo de fecha para visualizar las producciones musicales 	
<p>Observaciones:</p> <ul style="list-style-type: none"> - Las obras musicales pueden tener productores (creadores) fuera de la plantilla de la emisora. - Una producción musical debe ser posible imprimir con el formato utilizado por la Radio Cubana 	

Tabla 9: HU 7 - Gestionar listas organizativas. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 7	Rol: Director nacional de música
Nombre Historia: Gestionar listas organizativas	
Prioridad en el Negocio: Baja	Importancia del Desarrollo: 70
Tiempo Estimado: 6	Responsable: Omar Guerra Medina
<p>Descripción:</p> <ul style="list-style-type: none"> - Realizará CRUD sobre las listas organizativas - Deberán estar compuestas por un arreglo de obras musicales, Autores e intérpretes. - Se podrá definir una prioridad dentro de las listas organizativas - Deberán ser reutilizables - Las listas organizativas podrán ser filtradas por su nombre, por su contenido, el nombre de un intérprete, de un autor o de una obra. - podrán visualizarse las eliminadas y ser reintegradas. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> - De las listas organizativas depende la estadística por listas. - Esta funcionalidad sustituye a lo que en la Radio Cubana se conoce como maqueta musical. - Podrán ser vistas o utilizadas por medio de las estadísticas por cualquier usuario autenticado. 	

Tabla 10: HU 8 - Gestionar permisos para aprobar reportes ACDAM. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 8	Rol: Director nacional de música

Nombre Historia: Gestionar permisos para aprobar reportes ACDAM	
Prioridad en el Negocio: Media	Importancia del Desarrollo: 80
Tiempo Estimado: 3	Responsable: Omar Guerra Medina
Descripción:	
<ul style="list-style-type: none"> - Se definirá qué roles podrán marcar un reporte como aprobado 	
Observaciones:	

Tabla 11: HU 9 - Gestionar reportes ACDAM. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 9	Rol:
	<ul style="list-style-type: none"> - Fonotecario - Especialista de música
Nombre Historia: Gestionar reportes ACDAM	
Prioridad en el Negocio: Alta	Importancia del Desarrollo: 95
Tiempo Estimado: 18	Responsable: Omar Guerra Medina
Descripción:	
<ul style="list-style-type: none"> - Se pre visualizará la cantidad de obras musicales, fragmentos por emisora antes de crear el reporte - Solo se permitirá seleccionar la fecha final del reporte, se reportarán todas las producciones musicales ante de esa fecha que no hayan sido reportadas aún. - Los reportes solo podrán ser realizados por emisoras provinciales o superiores - Si la emisora es provincial reportará obligatoriamente todas las emisoras municipales que tenga. 	

- Solo podrá existir un reporte a la vez por emisora sin ser aprobado.
- Los reportes podrán ser eliminados solo si aún no están aprobados.
- Si se desea modificar un reporte aun no aprobado, será necesario eliminarlo y volverlo a crear.
- Solo podrá ser aprobado por solo uno de los roles definidos para aprobar reportes.
- Los reportes generarán dos documentos, la factura y la cantidad de obras.
- La cantidad de obras solo será exportable a Excel cuando el reporte sea aprobado. El formato del documento Excel estará definido por la ACDAM y deberá estar cifrado con contraseña.

Observaciones:

- La creación y aprobación de un reporte solo será posible para usuarios que pertenezcan a esa emisora en particular

Tabla 12: HU 10 – Realizar modelo clásico. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 10	Rol: Usuario autenticado
Nombre Historia: Realizar modelo clásico	
Prioridad en el Negocio: Media	Importancia del Desarrollo: 80
Tiempo Estimado: 12	Responsable: Omar Guerra Medina
<p>Descripción:</p> <ul style="list-style-type: none"> - Se deberá poder realizar consultas que generen un modelo capaz de mostrar los datos acordes al modelo de estadísticas musicales utilizado por la Radio Cubana. - Se deberá definir un periodo a consultar. - Se definirá la cantidad de elementos que contendrá las secciones. 	

- Se podrá filtrar por emisora y si es una emisora provincial se podrá realizar la consulta para todas las emisoras municipales que la componen.

Observaciones:

- Este documento debe ser posible imprimir con el formato utilizado por la Radio Cubana

Tabla 13: HU 11 – Realizar estadística general. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 11	Rol: Usuario autenticado
Nombre Historia: Realizar estadística general	
Prioridad en el Negocio: Media	Importancia del Desarrollo: 80
Tiempo Estimado: 10	Responsable: Omar Guerra Medina
<p>Descripción:</p> <ul style="list-style-type: none"> - Se podrá realizar consultas dentro de un periodo definido - Se podrá seleccionar una emisora y si es una emisora provincial se podrá realizar la consulta para todas las emisoras municipales que la componen. - Se podrá seleccionar un programa - Se podrá seleccionar una forma de búsqueda: Obras musicales, autores, intérpretes, géneros. - Se podrá escribir un texto de búsqueda que responderá a la forma de búsqueda seleccionada. - Se podrá seleccionar la cantidad de elementos que se quiere devolver en la consulta - Se podrá seleccionar una región geográfica a la que pertenece la obra musical, las regiones podrán ser seleccionadas al inverso. Ejemplo, todas las extranjeras será seleccionar Cuba y marcarlo como excepto. 	
Observaciones:	

- Este reporte deberá ser posible imprimirlo.

Tabla 14: HU 12 – Realizar estadística por listas organizativas. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 12	Rol: Usuario autenticado
Nombre Historia: Realizar estadísticas por listas organizativas	
Prioridad en el Negocio: Media	Importancia del Desarrollo: 80
Tiempo Estimado: 12	Responsable: Omar Guerra Medina
<p>Descripción:</p> <ul style="list-style-type: none"> - Se deberá seleccionar las una de las listas organizativas activas, con los elementos definidos en esta se extraerá la consulta. - Deberá ser posible seleccionar una emisora y si es una emisora provincial se podrá realizar la consulta para todas las emisoras municipales que la componen. - Se podrá seleccionar un programa - Se deberá definir un periodo a consultar. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> - Este reporte deberá ser posible imprimirlo. - Esta funcionalidad es la culminación de lo que en la Radio Cubana se conoce como Maqueta musical, ahora permitirá más opciones. 	

Tabla 15: HU 13 – Gestionar creadores. Fuente: Elaboración propia

HISTORIA DE USUARIO	
ID: HU 13	Rol:

	<ul style="list-style-type: none"> - Director nacional de música - Jefe de programación - Súper administrador - Administrador
Nombre Historia: Gestionar creadores	
Prioridad en el Negocio: Media	Importancia del Desarrollo: 80
Tiempo Estimado: 5	Responsable: Omar Guerra Medina
Descripción: <ul style="list-style-type: none"> - Realizará CRUD sobre creadores - Los creadores podrán ser filtrados por el nombre, la emisora a la que pertenecen y por los que están despedidos - Los creadores deberán ser visibles para todas las emisoras a pesar de que no pertenezcan a ellas. - Solo serán editables por la emisora a la que pertenecen. 	
Observaciones: <ul style="list-style-type: none"> - Los creadores no se eliminarán de la base de datos, se les da de baja y pueden ser readmitidos. 	

2.1.3. Pila del Producto (Product Backlog)

En Scrum los requisitos se expresan como elementos de la pila de producto (Product Backlog). El Product Backlog es una lista viva de requisitos funcionales y no funcionales priorizados por su valor para el cliente y el desarrollo. Al decir que se trata de una lista viva, deja claro que los requisitos que en ella aparecen y el orden de los mismos es cambiante a lo largo de la vida del proyecto. Los requisitos se van abordando en Sprints en el orden en que aparecen en el Product Backlog (González, 2007)

Tabla 16: Pila de productos ordenada por prioridad e importancia. Fuente: Elaboración propia

#	ID	Historia	Prioridad	Importancia	Tiempo estimado (días)
1	1	Diseño y creación de Base de Datos	Alta	100	10
2	2	Sistema de autenticación	Alta	99	8
3	3	Gestionar emisoras	Alta	95	12
4	4	Gestionar programas	Alta	95	14
5	5	Gestionar obras musicales	Alta	95	20
6	6	Gestionar producciones musicales	Alta	95	15
7	9	Gestionar reportes ACDAM	Alta	95	18
8	10	Realizar modelo clásico	Media	80	12
9	8	Gestionar permisos para aprobar reportes ACDAM	Media	80	3
10	11	Realizar estadística general	Media	80	10
11	12	Realizar estadísticas por listas organizativas	Media	80	12
12	13	Gestionar creadores	Media	80	5
13	7	Gestionar listas organizativas	Baja	70	6

El tiempo del equipo de trabajo por sprint se define en la siguiente tabla:

Tabla 17: Tiempo para cada Sprint. Fuente: Elaboración propia

Team Scrum	Jornada Laborar (h)	Horas de trabajo al proyecto por día (h)	Horas de trabajo al proyecto por semana (h)	Semanas de Trabajo por mes (u)	Total de horas (h)	Total de días laborables para el proyecto (días)
Omar Guerra Medina	8	8	40	4	160	20

2.1.4. Estimación de los Sprint

Se define para un máximo de 30 días por cada Sprint y basado en la pila de productos las siguientes estimaciones.

Tabla 18: Estimación del Sprint N° 1. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Diseño y creación de Base de Datos	Alta	100	10
Sistema de autenticación	Alta	99	8
Total de días del Sprint	18 días		

Tabla 19: Estimación del Sprint N° 2. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Gestionar emisoras	Alta	95	12
Gestionar programas	Alta	95	14
Total de días del Sprint	26 días		

Tabla 20: Estimación del Sprint N° 3. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Gestionar obras musicales	Alta	95	20
Total de días del Sprint	20 días		

Tabla 21: Estimación del Sprint N° 4. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Gestionar producciones musicales	Alta	95	15
Total de días del Sprint	18 días		

Tabla 22: Estimación del Sprint N° 5. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Gestionar reportes ACDAM	Alta	95	18
Total de días del Sprint	18 días		

Tabla 23: Estimación del Sprint N° 6. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Realizar modelo clásico	Media	80	12
Gestionar permisos para aprobar reportes ACDAM	Media	80	5
Total de días del Sprint	18 días		

Tabla 24: Estimación del Sprint N° 7. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
----------	-----------	-------------	------------------------

Realizar estadística general	Media	80	10
Realizar estadísticas por listas organizativas	Media	80	12
Total de días del Sprint	22 días		

Tabla 25: Estimación del Sprint N° 8. Fuente: Elaboración propia

Historia	Prioridad	Importancia	Tiempo estimado (días)
Gestionar creadores	Media	80	5
Gestionar listas organizativas	Baja	80	6
Total de días del Sprint	11 días		

2.1.5. Plan de Entregas (Sprints Backlog)

Sprints Backlog es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento "Terminado" (Sutherland, 2013)

Tabla 26: Planificación de entregas Sprint 1. Fuente: Elaboración propia

SPRINT No.1	
Fecha Inicio	22-10-2019
Fecha fin	14-11-2019
Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: - 28-10-2019

	<ul style="list-style-type: none"> - 04-11-2019 - 11-11-2019
Tareas a Desarrollar	<ul style="list-style-type: none"> - Diseño y creación de base de datos - Sistema de autenticación

Tabla 27: Planificación de entregas Sprint 2. Fuente: Elaboración propia

SPRINT No.2	
Fecha Inicio	15-11-2019
Fecha fin	20-12-2019
Revisión de los avances	<p>Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes:</p> <ul style="list-style-type: none"> - 18-11-2019 - 25-11-2019 - 02-12-2019 - 09-12-2019 - 16-12-2019
Tareas a Desarrollar	<ul style="list-style-type: none"> - Gestionar emisoras - Gestionar programas

Tabla 28: Planificación de entregas Sprint 3. Fuente: Elaboración propia

SPRINT No.3	
Fecha Inicio	03-1-2020
Fecha fin	30-1-2020

Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: <ul style="list-style-type: none"> - 06-1-2020 - 13-1-2020 - 20-1-2020 - 27-1-2020
Tareas a Desarrollar	- Gestionar obras musicales

Tabla 29: Planificación de entregas Sprint 4. Fuente: Elaboración propia

SPRINT No.4	
Fecha Inicio	31-1-2020
Fecha fin	26-2-2020
Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: <ul style="list-style-type: none"> - 03-2-2020 - 10-2-2020 - 17-2-2020 - 24-2-2020
Tareas a Desarrollar	- Gestionar producciones musicales

Tabla 30: Planificación de entregas Sprint 5. Fuente: Elaboración propia

SPRINT No.5	
Fecha Inicio	27-2-2020

Fecha fin	24-3-2020
Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: <ul style="list-style-type: none"> - 02-3-2020 - 09-3-2020 - 16-3-2020 - 23-3-2020
Tareas a Desarrollar	- Gestionar reportes ACDAM

Tabla 31: Planificación de entregas Sprint 6. Fuente: Elaboración propia

SPRINT No.6	
Fecha Inicio	25-3-2020
Fecha fin	14-4-2020
Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: <ul style="list-style-type: none"> - 30-3-2020 - 06-4-2020 - 13-4-2020
Tareas a Desarrollar	- Realizar modelo clasico - Gestionar permisos para aprobar reportes ACDAM

Tabla 32: Planificación de entregas Sprint 7. Fuente: Elaboración propia

SPRINT No.7

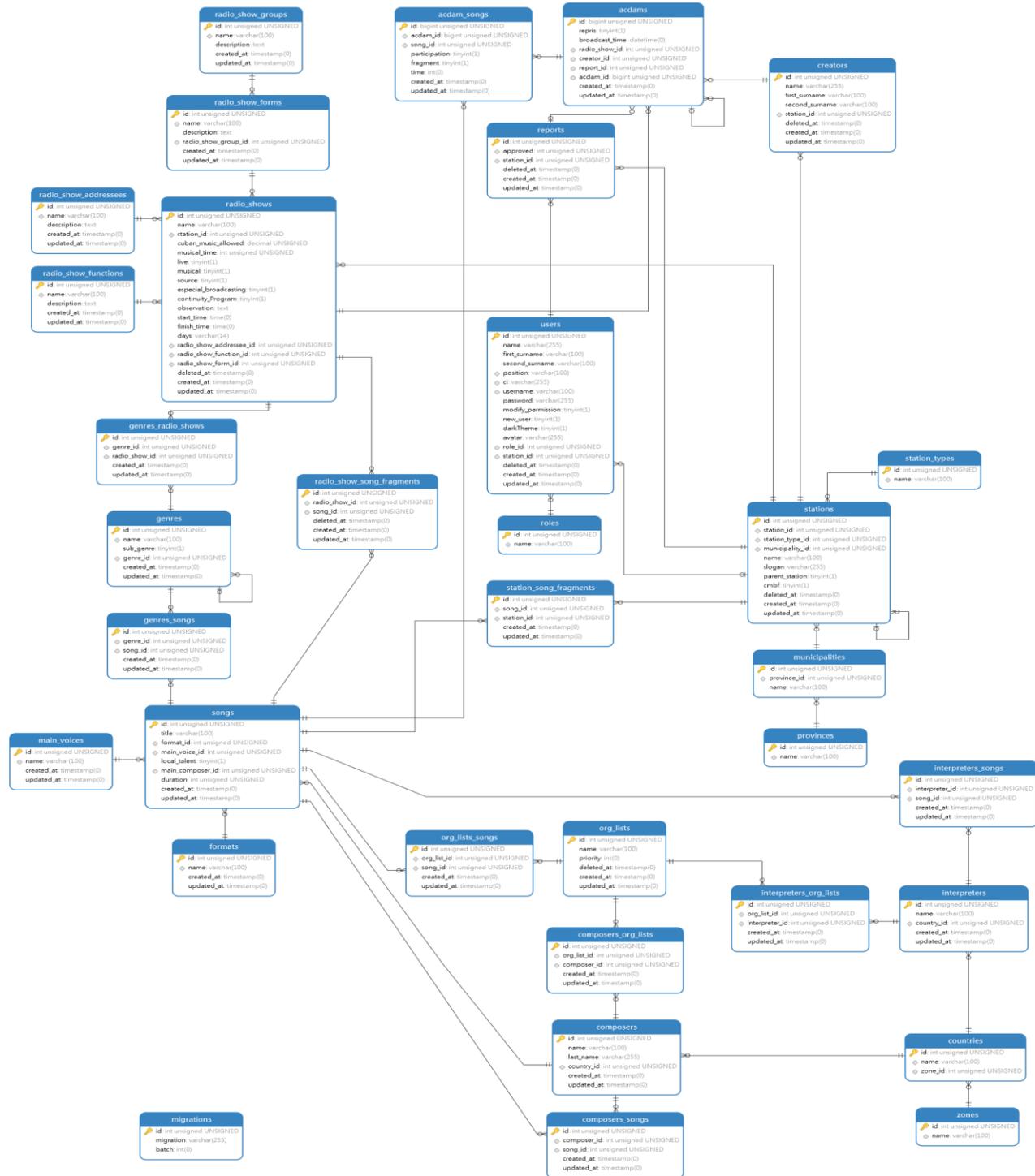
Fecha Inicio	19-3-2020
Fecha fin	15-4-2020
Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: <ul style="list-style-type: none"> - 23-3-2020 - 30-3-2020 - 06-4-2020 - 13-4-2020
Tareas a Desarrollar	<ul style="list-style-type: none"> - Realizar estadística general - Realizar estadísticas por listas organizativas

Tabla 33: Planificación de entregas Sprint 8. Fuente: Elaboración propia

SPRINT No.8	
Fecha Inicio	16-4-2020
Fecha fin	30-4-2020
Revisión de los avances	Las revisiones se realizarán semanalmente. Las fechas de revisión serán las siguientes: 20-4-2020 27-4-2020
Tareas a Desarrollar	<ul style="list-style-type: none"> - Gestionar creadores - Gestionar listas organizativas

2.2. Base de datos

Ilustración 5: Base de Datos



2.3. Estudio de Costo

No importa que tan grande o tan pequeño sea un proyecto de desarrollo de software, una buena estimación de su costo permitirá resolver problemas asociados al esfuerzo y tiempo invertido en su realización.

Las principales técnicas de estimación de proyectos de desarrollo de software asumen que el tamaño de software es un vector importante para la determinación del esfuerzo para su construcción. Por lo anterior, saber su tamaño es uno de los primeros pasos del proceso de estimación de esfuerzo, plazo y costo.

2.3.1. Método de puntos de función

El Análisis de Puntos de Función (FPA, por sus siglas en inglés) es una técnica de medición de las funcionalidades ofrecidas por un software desde el punto de vista de sus usuarios. Punto de función (FP, por sus siglas en inglés), que es su unidad de medida, tiene por objetivo tornar la medición independiente de la tecnología utilizada para su construcción (Albrecht, 1979). Es decir, el FPA busca medir lo que el software hace y no como es construido.

Por tanto, el proceso de medición (también llamado conteo de puntos de función) se basa en una evaluación estandarizada de los requerimientos funcionales del usuario. Este procedimiento de medición está descrito por el IFPUG en su Manual de Prácticas de Medición.

Tabla 34: Tabla de valores

Tipo/Complejidad	Baja	Media	Alta
Entrada Externa (EI)	3	4	6
Salida Externa (EO)	4	5	7
Consulta Externa (EQ)	3	4	6
Archivo Lógico Interno (ILF)	7	10	15
Archivo de Interfaz Externo (EIF)	5	7	10

Tabla 35: Tabla completa

Tipo/Complejidad	Baja	Media	Alta	Total
Entrada Externa (EI)	$3*6=18$	$4*3=12$	$6*7=42$	72

Salida Externa (EO)	4*4=16	5*5=25	7*11=77	118
Consulta Externa (EQ)	-	4*5=20	6*11=66	86
Archivo Lógico Interno (ILF)	-	-	15*1=15	15
Archivo de Interfaz Externo (EIF)	-	-	-	0
				PFSA=291

Luego de obtener los puntos de función sin ajustar, se debe calificar cada uno de los factores de valor de ajuste, en una escala del 0 al 5:

Tabla 36: Factor de ajuste

No.	Factores de valor de ajuste	Valor
1	Comunicación de datos	3
2	Proceso distribuido de datos	3
3	Desempeño	1
4	Configuración	1
5	Volumen de transacciones	3
6	Captura de datos en línea	5
7	Eficiencia del usuario final	5
8	Actualización de datos en línea	3
9	Complejidad	3
10	Reusabilidad	3

11	Facilidad de instalación	1
12	Facilidad de operación	2
13	Instalación múltiple	0
14	Facilidad de cambio	5
	FCT	38

Calculo de Puntos de Función Ajustados

$$PFA = PFSA * [0.65 + (0.01 * FCT)]$$

$$PFA = 291 * [0.65 + (0.01 * 38)]$$

$$PFA = 291 * [0.65 + 0.38]$$

$$PFA = 291 * 1.03$$

$$PFA = 299.73$$

Calculo Esfuerzo [hora/persona]

$$E = PFA / 1/8 \text{ personas/hora}$$

$$E = 299.73 / 1/8 \text{ personas/hora}$$

$$E = 2397.84 \text{ horas/personas}$$

Tomando 20 días laborables en el mes y 8 horas productivas al día, obtenemos 160 horas laborables al mes.

$$\text{Duración de proyecto en horas} = 2397.84 \text{ horas/persona} \times 1 \text{ persona} = 2397.84 \text{ horas}$$

$$\text{Duración en meses} = 2397.84 \text{ horas} / (160 \text{ horas/mes}) = \mathbf{14.9 \text{ meses}}$$

Cálculo del Presupuesto del Proyecto

Suponiendo un sueldo de 600.00 MN

$$\text{Costo Total} = \text{sueldo de 1 participante} * \text{cantidad de participantes} * \text{Tiempo de desarrollo}$$

Costo Total = $600 * 1 * 14.9$

Costo Total = 8991.9 MN

Conclusiones parciales

En este capítulo se elaboraron los elementos descriptivos y artefactos fundamentales correspondientes a las fases de planificación, diseño e implementación que propone la metodología de desarrollo SCRUM.

Se definió la estimación de costos por el método de puntos de función donde se determinó la factibilidad económica con un costo total de 8991.9 MN y un tiempo de desarrollo aproximado del software de 14 meses.

Se obtuvo un buen entendimiento de los procesos implicados en la solución propuesta fundamental para el desarrollo de la misma.

3. CAPÍTULO 3: Validación del Trabajo Desarrollado

Introducción

En el siguiente capítulo se describe la solución propuesta, se analizan las principales interfaces de usuario, los algoritmos más importantes en el software y un conjunto de pruebas para comprobar su correcto funcionamiento.

3.1. Descripción de la solución propuesta

Se propone desarrollar una aplicación web y desplegarla en los servidores del ICRT de manera tal que tenga se tenga acceso solo desde la VPN de dicha institución.

La solución contiene un sistema de autenticación de usuarios por roles:

Reglas de autenticación:

- La sesión de usuario debe expirar a partir de las 6 horas de su ultimo uso por el usuario.
- Si el usuario pertenece a una emisora que esta fuera del aire no podrá iniciar sección.
- Si el usuario está de baja no puede iniciar sección.

Roles de alto nivel:

Súper Administrador:

- Tiene permisos para realizar cualquier acción en el sistema, excepto realizar crear, editar o eliminar producciones musicales y gestionar fragmentos.
- Es el único que puede realizar CRUD sobre emisoras provinciales, nacionales e internacionales.
- Es el único que puede realizar CRUD sobre usuarios bajo los roles de Súper Administrador y Director Nacional de Música.
- Es el único que puede transferir un trabajador entre provincias o entre emisoras nacionales e internacionales.

Director nacional de música:

- Realizar CRUD sobre listas organizativas

- Realizar CRUD sobre las formas y grupos de programas
- Realizar CRUD sobre los destinatarios
- Realizar CRUD sobre las funciones de programas
- Realizar CRUD sobre programas de radio
- Realizar CRUD sobre obras musicales, autores, intérpretes, géneros, formatos
- Acceder a los registros del sistema
- Ver el listado de emisoras
- Realizar CRUD sobre creadores

Roles regionales:

Son los que pueden actuar solo sobre su emisora. Si pertenece a una emisora marcada como provincial pueden realizar las mismas acciones sobre estas, con excepción del Administrador que tiene más funciones en ese caso.

Administrador:

- Realizar CRUD sobre usuarios de la emisora a la que pertenece
- Si pertenece a una emisora marcada como provincial puede:
 1. Realizar CRUD sobre emisoras municipales donde el padre sea solo la emisora a la que pertenece.
 2. Realizar CRUD sobre usuarios en esas emisoras

Jefe de Programación:

- Realizar CRUD sobre programas de radio
- Realizar CRUD sobre creadores
- Ver el listado de usuarios

Especialista de música:

- Realizar CRUD sobre producciones musicales (Solo en la emisora a la que pertenece)

- Realizar CRUD sobre obras musicales, autores, intérpretes, géneros, formatos
- Realizar reportes ACDAM (factura y lista de obras a pagar)

Fonotecario:

- Realizar CRUD sobre producciones musicales (Solo en la emisora a la que pertenece)
- Realizar CRUD sobre obras musicales, autores, intérpretes, géneros, formatos
- Realizar reportes ACDAM

Otros roles:

Los siguientes roles existen porque pueden ser seleccionados para aprobar reportes ACDAM

Coordinador

Asesor

El siguiente rol solo puede consultar y solicitar estadísticas. Las acciones que tiene permitido son comunes con el conjunto de roles anteriores.

Director de programas

Las acciones comunes son

- Ver la lista de obras musicales, autores, intérpretes, géneros, formatos.
- Ver la lista de producciones musicales.
- Acceder a las estadísticas.
- Ver la lista de listas organizativas.
- Ver la lista de programas y sus elementos.
- Modificar la foto de perfil y el modo de su interfaz.

Junto a los roles existen un permiso agregado:

- Si el usuario se da de alta como colaborador podrá editar obras musicales y sus componentes.

La funcionalidad más utilizada será la de insertar producciones musicales y con esta insertar obras musicales. La aplicación web realiza estas funciones de la forma más sencilla, intuitiva y segura posible.

Para esto se destinó un acceso directo en el menú principal desde el cual se puede insertar todos los elementos necesarios para introducir una producción musical.

Todas las pantallas donde se listen elementos se puede realizar filtros excepto en las formas y grupos de programa, las funciones de programa y los destinatarios por su baja complejidad.

La aplicación cuenta con pantallas para gestionar emisoras, usuarios, creadores y programas. Estas funciones tienen en cuenta si la emisora a la que pertenece el usuario está marcada como provincial.

Cuenta también con pantallas destinadas a realizar reportes, estadísticas y listas organizativas. Los reportes deben ser aprobados por aquellos roles que estén dados de alta para esto.

3.2. Interfaces

La **interfaz gráfica de usuario**, conocida también como GUI (del inglés *graphical user interface*), utiliza imágenes, iconos y menús para mostrar las acciones disponibles en un dispositivo, entre las que el usuario puede escoger una o varias. (Villalobos, 2020)

La interfaz gráfica de usuario de escritorio de Windows o Mac OS son ejemplos muy populares de GUI, que reemplazaron la comunicación mediante comandos de texto y código binario por elementos gráficos, hoy generalizada en la mayoría de equipos.

Principales interfaces de la aplicación web.

Interfaz: Modelo clásico.

Ilustración 6. Reporte modelo clásico

Modelo Clasico Imprimir

Emisora: Radio 26 Provincial: Provincial Cantidad: 5 Periodo: 2020-07-01 - 2020-07-31

DIRECCIÓN NACIONAL DE MÚSICA
ESTADÍSTICAS MUSICALES

Emisora: Todo el país		Provincia: Cuba		Municipio:		Periodo: 2020-07-01	
Obras, autores e intérpretes por zonas geográficas.							
No	Zonas	Cantidad de obras	%	Cantidad autores	%	Cantidad intérpretes	%
1	Cuba	1	100.0	1	100.0	1	100.0
2	Extranjero	0	0.0	0	0.0	0	0.0
Total general		1	100.0	1	100.0	1	100.0
1	LA	0	0.0	0	0.0	0	0.0
2	NA	0	0.0	0	0.0	0	0.0
3	EUR	0	0.0	0	0.0	0	0.0
4	Asia	0	0.0	0	0.0	0	0.0
5	Africa	0	0.0	0	0.0	0	0.0
6	Others	0	0.0	0	0.0	0	0.0

Obras musicales más difundidas.					
No	Título	Intérpretes	Autores	País	Frecuencia
1	A donde vas	Laritza bacallao	Leni Torres	CUBA	1

Autores más difundidos.				
No	Autores	País	Cantidad	Frecuencia

El reporte modelo clásico es una de las principales funcionalidades de esta aplicación.

Esta interface está compuesta por:

- Barra de título, más abajo en gris, muestra el título (Modelo clásico) y un botón para imprimir.
- Barra de búsqueda:
 1. Emisora
 2. Si se va analizar como provincial o no.
 3. Cantidad
 4. Periodo de fechas.

El cuerpo del modelo clásico está compuesto por 6 partes:

- Cabecera: especifica la emisora, la región que se está consultando y el periodo.
- Zonas: Muestra la cantidad de obras, interpretes, autores y géneros difundidos en el periodo de tiempo seleccionado.
- Obras más difundidas
- Intérpretes más difundidos
- Autores mas difundidos
- Géneros más difundidos.

Puede ser impreso o exportado a PDF

Interfaz: Nueva producción musical

Ilustración 7. Pantalla nueva producción musical

Nueva producción

Reprisado Reutilizado

Programa:

Productor: Todos

Fecha de transmisión:

Seleccionar obra musical Participación Fragmento Cantidad: 0

Lista de obras musicales

Listado de registros asociados:

Título	Autores	Intérpretes	Géneros	Formato	País	Voz	Duración	P	TL	F
↑ Para habilitar esta sección cree un acdam en el panel superior No hay registros										

Autores más difundidos

La introducción de las producciones musicales es la tarea más recursiva del proyecto, es crucial que sea intuitiva, sencilla y asequible.

Solo pueden ser gestionadas por:

- Especialista de música
- Fonotecarios

Una producción musical luego de ser reportada no puede ser editada ni borrada.

Las producciones musicales se pueden imprimir con el formato original definido por la Radio Cubana junto al creador de este sistema.

Ilustración 8. Vista de impresión para una producción musical.

Título	Autores	Intérpretes	Géneros	Formato	Pais	Voz	Duración	P	TL	F
A donde vas	Leni Torres	Laritza bacallao	Afro-cubana, Ambient	Single	CUBA	M	00:00:50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Interfaz: Agregar obra musical.

Ilustración 9. Pantalla agregar obra musical

Título:

Formato:

Voz:

Duración:

Talento Local:

Autores:

Intérpretes:

Géneros:

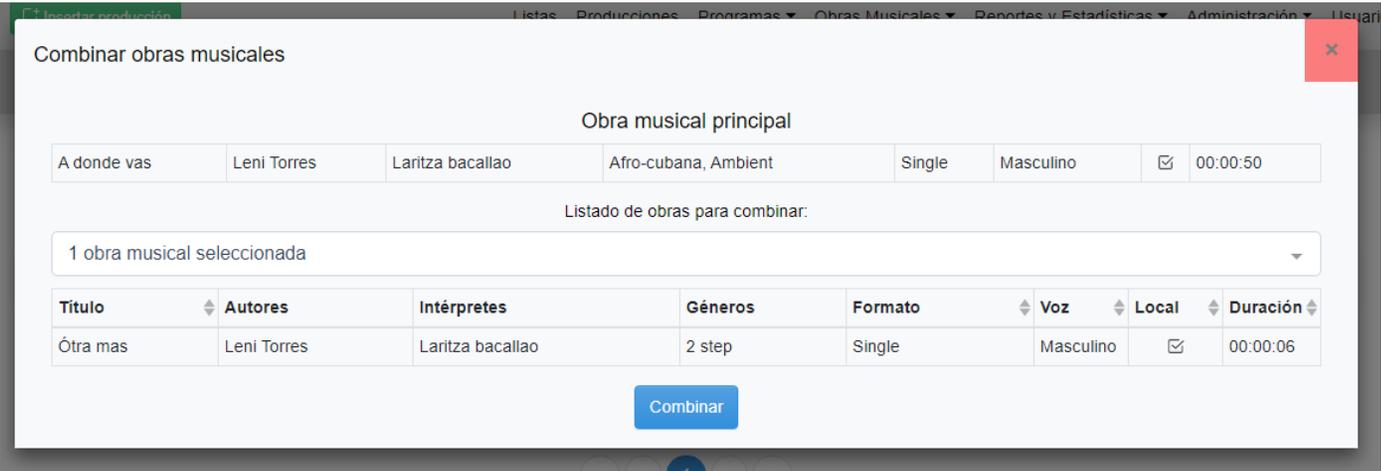
Dentro de las obras musicales destaca la posibilidad de insertar casi todos los elementos que la componen desde el botón “Agregar obra musical” a esto se le nombró como sistema todo en uno.

Esta funcionalidad funciona para autores, intérpretes y géneros.

La obra musical debe tener un tiempo mayor a 5 segundos.

Es obligatorio seleccionar un autor principal para definir el país y la zona de la obra musical.

Las obras musicales, así como los autores en sus acciones tienen una opción de combinar.



Combinar obras musicales

Obra musical principal

A donde vas	Leni Torres	Laritza bacallao	Afro-cubana, Ambient	Single	Masculino	<input checked="" type="checkbox"/>	00:00:50
-------------	-------------	------------------	----------------------	--------	-----------	-------------------------------------	----------

Listado de obras para combinar:

1 obra musical seleccionada

Título	Autores	Intérpretes	Géneros	Formato	Voz	Local	Duración
Ótra mas	Leni Torres	Laritza bacallao	2 step	Single	Masculino	<input checked="" type="checkbox"/>	00:00:06

Combinar

Combinar permite unificar varios elementos en uno. El principal objetivo es resolver las diferencias que puedan existir al insertar la información y evitar la redundancia en la base de datos.

Los compositores o autores siguen las mismas normas que las obras musicales, así como los intérpretes.

3.3. Algoritmos

A continuación, se presentan algunos fragmentos de códigos cruciales para la seguridad e integridad de los datos.

Función: Restringir acceso.

Ilustración 10. Función Restringir acceso

```
public static function restrictAccess($query, bool $allow_children)
{
    $User = Auth::user();
    /** @var User $user */
    if (!$User->isRole( role: MyRoles::SuperAdmin) & !$User->isRole( role: MyRoles::DirectionMusical)) {
        $stationId = $User->station_id;
        if ($allow_children & $User->station->parent_station) {
            $query->whereRaw("(stations.id = $stationId or stations.station_id = $stationId)");
        } else {
            $query->where('stations.id', $stationId);
        }
    }
    return $query;
}
```

Esta función es utilizada para devolver los programas y emisoras correspondientes a cada rol.

Descripción del algoritmo por líneas:

1. Obtener el usuario autenticado del sistema
2. Comprobar que no sea un rol de alto nivel.
3. Obtener el id de emisora a la que pertenece el usuario.
4. Si el usuario trabaja en una emisora marcada como provincial y se le permite ver a los municipios
5. Se modifica la consulta para que vea la emisora donde radica y los municipios
6. O solo se le permite ver su emisora
7. Devolver la consulta.

Función: Crear obra musical

Ilustración 11. Función Crear obra musical

```
public function create(array $data, $log = true)
{
    if ($this->itsExists($data))
        throw new \Exception(MyMessages::combination_exist());
    $song = parent::create($data);
    $song->composers()->attach($data['composers']);
    $song->interpreters()->attach($data['interpreters']);
    $song->genres()->attach($data['genres']);
    return $this->search(['songs.id' => $song['id']])->get();
}
```

Descripción del algoritmo por líneas:

1. Comprobar que no exista otra obra en la base de datos con esta combinación.
2. De ser así lanzar una excepción que será manejada en lado del cliente.
3. Crear la obra musical utilizando el **patrón repository**.
4. Agregar la lista de autores
5. Agregar la lista de intérpretes
6. Agregar la lista de géneros
7. Retornar con la búsqueda de la obra. Esto se realiza con el objetivo que devuelva también las relaciones dispuestas en el **ORM Eloquent** para las obras musicales.

Esta función obtiene un arreglo de datos del **Request** (Mecanismo de seguridad y validación de datos) definido para ella:

```
public function rules()
{
    return [
        'title' => 'required|string',
        'format_id' => 'required|integer|exists:formats,id',
        'main_voice_id' => 'required|integer|exists:main_voices,id',
        'duration' => 'required|integer|min:1',
        'local_talent' => 'required|boolean',
        'main_composer_id' => 'required|integer|in_array:composers.*',
        'composers.*' => 'required|integer|distinct|exists:composers,id',
        'interpreters.*' => 'required|integer|distinct|exists:interpreters,id',
        'genres.*' => 'required|integer|distinct|exists:genres,id',
    ];
}
```

En este se definen las reglas que el lado del usuario debe cumplir para que pueda llegar a la función Crear obra musical del servidor.

3.4. Pruebas de software

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta. (Jústiz NúñezI, Gómez SuárezII, & Delgado Dapenall, 2014)

3.4.1. Pruebas de aceptación

Las pruebas de aceptación son un conjunto de pruebas realizadas por el usuario del sistema y preparadas por el equipo de desarrollo. La ejecución y aprobación final corresponden al usuario.

Como resultado de las pruebas de aceptación se obtendrán un conjunto de tablas donde cada una de ellas estarán compuestas por los siguientes datos:

- **Número:** Identificador de la prueba realizada.
- **EPP:** Nombre del elemento de la pila de producto al que hace referencia la prueba a realizar.
- **Nombre:** Nombre que se le da a la prueba a realizar.
- **Descripción:** Se informa la funcionalidad que se desea probar.
- **Condiciones de Ejecución:** Mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- **Entradas:** Descripción de los pasos para realizar la prueba.
- **Resultado esperado:** Se hará una breve descripción del resultado al realizar la prueba.
- **Evaluación de la prueba:** Con respecto al resultado de la prueba se evaluará por medio de tres criterios:
 - **Satisfactoria:** Resultado de la prueba es exactamente el esperado por el usuario.
 - **Parcialmente satisfactoria:** El resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.

- **No satisfactoria:** El resultado de la prueba realizada genera un error de en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, la funcionalidad requerida por el cliente no tenga resultado, lo que invalida el EPP.

A continuación, las principales pruebas de aceptación realizadas.

Tabla 37. Prueba de aceptación 1. Fuente: Elaboración propia

PRUEBAS DE ACEPTACIÓN	
Número: 1	No. EPP: 1
Responsable: Omar Guerra Medina	
Nombre Caso de Prueba: Prueba de integridad a la base de datos	
Descripción: Comprobar integridad de la base de datos al realizar las siguientes acciones: <ul style="list-style-type: none"> - Crear - Editar - Eliminar Comprobar que se elimine en cascada correctamente.	
Condiciones de ejecución: MySQL Server 8 o superior corriendo	
Entradas: Datos de Pruebas	
Resultado esperado: La base de datos responde correctamente, todos los conjuntos de datos se comportaron como se esperaba.	
Evaluación: Prueba satisfactoria	

Tabla 38. Prueba de aceptación 2. Fuente: Elaboración propia

PRUEBAS DE ACEPTACIÓN	
Número: 2	No. EPP: 2
Responsable: Omar Guerra Medina	

Nombre Caso de Prueba: Prueba de autenticación en el sistema.
Descripción: Se realizan intentos de entrar al sistema con conjuntos de usuarios y contraseñas incorrectos, campos en blanco y al final con un usuario registrado y habilitado.
Condiciones de ejecución: Usuario valido en el sistema.
Entradas: Usuario y contraseña.
Resultado esperado: El sistema detecta de correctamente en la entrada de los datos incorrectos e informa al usuario. En otro caso se abre el sistema correctamente.
Evaluación: Prueba satisfactoria

Conclusiones parciales

En este capítulo se presentó la descripción de la solución ilustrando cada uno de los roles con detalle. Se diseñaron algunas interfaces y algoritmos del sistema. Se realizaron pruebas de aceptación con las cuales se obtuvieron los siguientes resultados:

- Se demostró el buen funcionamiento del software y el cumplimiento de los requerimientos del cliente.
- El cliente confirmó que la aplicación aumenta la eficiencia y eficacia del proceso de gestión de las producciones musicales en su emisora lo que se traduce en mayor eficiencia y eficacia en la Radio Cubana.

Conclusiones

Como resultado de esta investigación se cumplieron los objetivos definidos y se obtuvo las siguientes conclusiones:

- El estudio de los referentes teóricos metodológicos del proceso actual de gestión de producciones musicales y los antecedentes informáticos, permitió contar con los elementos necesarios para dar solución a la problemática planteada y llegar a la conclusión de que los sistemas informáticos encontrados en el estudio del estado del arte no dan solución problema de investigación.
- Se seleccionaron las herramientas y metodologías como laravel para el servidor web y Vue como framework JS para el cliente, MySql como sistema gestor de base datos y visual studio code como IDE de desarrollo, siendo estas las más idóneas para el desarrollo de la propuesta de solución.
- Se desarrolló la aplicación web para la gestión de producciones musicales en la Radio Cubana siguiendo la metodología de desarrollo de software Scrum, ajustado a los requerimientos del usuario y logrando un producto informático que cumple con el objetivo general trazado en la investigación.
- Se realizó la estimación de costo por medio del método de puntos de función, arrojando como resultado la factibilidad de la realización del sistema informático, con un costo de 8991.9 MN para un tiempo de desarrollo aproximado de 14.9 meses.
- La realización de las pruebas de aceptación al sistema informático permitió detectar errores en el sistema y la rápida corrección de los mismos.
- Luego de haber verificado los objetivos específicos del trabajo de diploma se puede concluir que el objetivo general se cumplió al obtener Una aplicación web para la gestión de producciones musicales en las emisoras de la Radio Cubana. Se obtuvo, una mayor eficiencia y eficacia en la exactitud y confiabilidad de los datos. Aumentó las facilidades para realizar estadísticas, reportes y mejoró el control de las obras radiadas. con esto se afirma que la hipótesis planteada en los inicios de la investigación es verdadera

Recomendaciones

- Desarrollar un módulo que permita aprobar las producciones musicales
- Generar reportes con gráficas para mejorar la comprensión de los datos
- Agregar más campos a los autores y los intérpretes como la biografía para que el sistema sirva como una herramienta didáctica.
- Desarrollar el conjunto de módulos que permitan que el sistema maneje archivos de audio, los compare por medio de la huella acústica de los mismo y pueda convertirse en el servidor central de musical de la Radio Cubana.

Referencias bibliográficas

- Abellán, E. (05 de 03 de 2020). *Metodología Scrum: qué es y cómo funciona*. Obtenido de <https://www.wearemarketing.com>: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- ACDAM. (22 de Enero de 2020). *Agencia Cubana de Derecho de Autor Cubana*. Obtenido de <http://www.acdam.cu>
- ACN. (03 de 10 de 2018). *Clausuró Díaz-Canel conferencia nacional de la Unión de Informáticos*. Obtenido de Vanguardia: <http://www.vanguardia.cu/de-cuba/12297-clausuro-diaz-canel-conferencia-nacional-de-la-union-de-informaticos>
- Aitana. (16 de Octubre de 2018). *Visual Studio Code: Funcionalidades y extensiones*. Obtenido de Aitana: <https://blog.aitana.es/2018/10/16/visual-studio-code/>
- Albrecht, A. (1979). *Measuring Application Development Productivity*. IBM.
- autores, C. d. (2012). *Proceso de Desarrollo y Gestión de Proyectos de Software*. La Habana, Cuba.
- Bach.Llatas Yovera, L. H. (2017). *ANÁLISIS COMPARATIVO DE FRAMEWORKS PHP PARA MEDIR EL RENDIMIENTO*. Perú: Pimentel – Perú.
- Badal, H. (28 de 03 de 2017). *linkedin*. Obtenido de <https://es.linkedin.com/pulse/ventajas-y-desventajas-de-una-web-app-hector-badal-mba>
- Benites, A. G. (2017). *Frontend y Backend*. Obtenido de devcode: <https://devcode.la/blog/frontend-y-backend/>
- bootstrap-vue. (22 de Enero de 2020). *bootstrap-vue*. Obtenido de <https://bootstrap-vue.js.org/>
- Calderio, B. R. (1977). *Ley No. 14 1977 Ley de derecho de autor*. La habana.
- clinicbox. (22 de 07 de 2018). *Aplicaciones Web vs Escritorio*. Obtenido de <https://clinicbox.mx>: <https://clinicbox.mx/2018/07/22/aplicaciones-web-vs-escritorio/>
- Comparacion de herramientas CASE*. (10 de 5 de 2013). Obtenido de Comparacion de herramientas CASE: <http://herramientascascomparaciones.blogspot.com/>

- Corrales, J. A. (2 de Agosto de 2019). *Interfaz de usuario o UI: ¿qué es y cuáles son sus características?* Obtenido de Rockcontent: <https://rockcontent.com/es/blog/interfaz-de-usuario/>
- Costa, C. (10 de Octubre de 2013). *PHP: ¿Qué es y para qué sirve?* Obtenido de Agencia La Nave: <https://blog.agencialanave.com/php-que-es-y-para-que-sirve/>
- Cubana, R. (22 de Enero de 2020). *Portal y sitio oficial de la Radio Cubana en Internet*. Obtenido de <http://www.radiocubana.cu/>
- Delgado, D. O. (19 de 10 de 2017). *Diferencias entre Vue.js y React*. Obtenido de Openwebinars: <https://openwebinars.net/blog/diferencias-entre-vuejs-y-react/>
- Flores Herrera, J. (25 de Agosto de 2015). *Qué Es HTML*. Obtenido de Código Facilito: <https://codigofacilito.com/articulos/que-es-html>
- Gonçalves, L. (03 de 01 de 2020). *QUÉ ES LA METODOLOGÍA ÁGIL*. Obtenido de <https://luis-goncalves.com>: <https://luis-goncalves.com/es/que-es-la-metodologia-agil/>
- González, R. C. (12 de 11 de 2007). *Expresando Scrum: Scrum y la gestión de requisitos*. Obtenido de geeks: <https://geeks.ms/>
- Gustavo. (1 de Abril de 2020). *¿Qué es HTML? Explicación de los fundamentos del Lenguaje de marcado de hipertexto*. Obtenido de Hostinger: <https://www.hostinger.es/tutoriales/que-es-html/>
- Halcomb, E. H. (2015). *Mixed Methods Research*.
- Haro, J. M. (14 de 01 de 2008). *Diseño e implementación de un marco de trabajo presentación para aplicaciones JEE*. *Diseño e implementación de un marco de trabajo presentación para aplicaciones JEE*.
- Hausman, C. (2012). *Producción moderna de radio: producción, programación y ejecución*. CENGAGE LEARNING.
- Hernández, S. J. (2011). *Manual online de Realización Radial*. La habana: Radio Cubana.
- ICRT. (20 de 05 de 2020). *Principal*. Obtenido de ICRT: <https://www.icrt.gob.cu/>
- Jústiz NúñezI, D., Gómez SuárezII, D., & Delgado Dapenall, M. D. (2014). *Proceso de pruebas para productos de software en un laboratorio de calidad*. *Ingeniería Industrial*, 35.

- Lara Sobrino, J. d. (2019). *Aplicación web para la gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivo Matanzas*. Universidad de Matanzas, Facultad de Ciencias Técnicas. Departamento de Informática. Matanzas: Universidad de Matanzas.
- Larrea, M. J. (2015). *Estudio comparativo de productividad de frameworks PHP orientados a objetos para desarrollar el Sistema de Seguimiento de Incidentes de la Infraestructura de Red en la ESPOCH. 10*.
- Latham, A. (2001). *DICCIONARIO OXFORD ENCICLOPÉDICO DE LA MÚSICA*. Pinkneys Green, Berkshire.
- Libres, R. (22 de Enero de 2020). *Radios Libres*. Obtenido de <https://radioslibres.net/que-es-una-radio-en-linea-y-ventajas/>
- MDN, c. d. (26 de 6 de 2020). *JavaScript*. Obtenido de MDN web docs: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Microsoft. (05 de 07 de 2020). *Code.visualstudio*. Obtenido de Code.visualstudio: <https://code.visualstudio.com/docs>
- Muñoz, V. J. (2013). *El nuevo PHP. Conceptos avanzados*. España: Bubok Publishing S. L.
- Natalya Prokofyeva, V. B. (2017). *Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems*. Latvia: Riga Technical University, 1 Kalku Street, Rīga, LV-1658, Latvia.
- Otwell, T. (07 de 05 de 2020). *Laravel*. Obtenido de Laravel: <https://laravel.com/>
- Pérez Pérez, M. J. (2012). *Guía Comparativa de Metodologías Ágiles . Guía Comparativa de Metodologías Ágiles .* Valladolid , Valladolid , España.
- Potter, J. (05 de 07 de 2020). *NPM Trends*. Obtenido de NPM Trends: <https://www.npmtrends.com/>
- Pressman, R. S. (1995). *Ingeniería de Software*. Mc Graw Hill.
- Rey Guevara, C. (2017). *Estudio de la efectividad de la aplicación de la metodología ágil de desarrollo Scrum*. UNIVERSIDAD TECNOLÓGICA ISRAEL. Universidad Tecnológica Israel.
- Rubenfa. (14 de Julio de 2014). *Patrones de diseño: qué son y por qué debes usarlos*. Obtenido de Genbeta: <https://www.genbeta.com/desarrollo/patrones-de-diseno-que-son-y-por-que-debes-usarlos>

- Saavedra, E. L. (2009). *Framework para el desarrollo de aplicaciones*. Catalyst.
- Sierra, C. D. (15 de Noviembre de 2007). *Visual Paradigm For Uml*. Obtenido de SlideShare:
<https://es.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>
- SUBRA, J.-P. (s.f.). *Scrum un método ágil para sus proyecto*. eni.
- Sutherland, K. S. (2013). *La Guía de Scrum*.
- Tinoco Gómez, O., Rosales López, P. P., & Bacalla, S. (julio de 2010). Criterios de selección de metodologías de desarrollo de software. *Criterios de selección de metodologías de desarrollo de software*. Lima, Lima, Perú: Universidad Nacional Mayor de San Marcos.
- Training, L. A. (23 de 8 de 2017). *que-son-las-pruebas-de-aceptacion/*. Obtenido de <https://losandestraining.com>:
<https://losandestraining.com/2017/08/23/que-son-las-pruebas-de-aceptacion/>
- Trends, G. (05 de 07 de 2020). *Google Trends*. Obtenido de Google Trends: <https://trends.google.es/trends/>
- Villalobos, C. (18 de Marzo de 2020). *MARKETING*. Obtenido de Qué es la interfaz de usuario y qué tipos existen:
<https://blog.hubspot.es/marketing/interfaz-usuario>
- vuejs. (22 de Enero de 2020). *vuejs.org*. Obtenido de <https://vuejs.org/>
- Weizenbaum, N. (3 de 9 de 2020). Sass makes CSS fun!
- You, E. (05 de 07 de 2020). *Vuejs*. Obtenido de Vuejs: <https://vuejs.org/>