

Universidad de las Ciencias Informáticas



Facultad 4

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

Título:

***Sistema para la Gestión de la
Información Estadística del Festival de
Artistas Aficionados***

Autor:

Lídice Martínez Cabrera

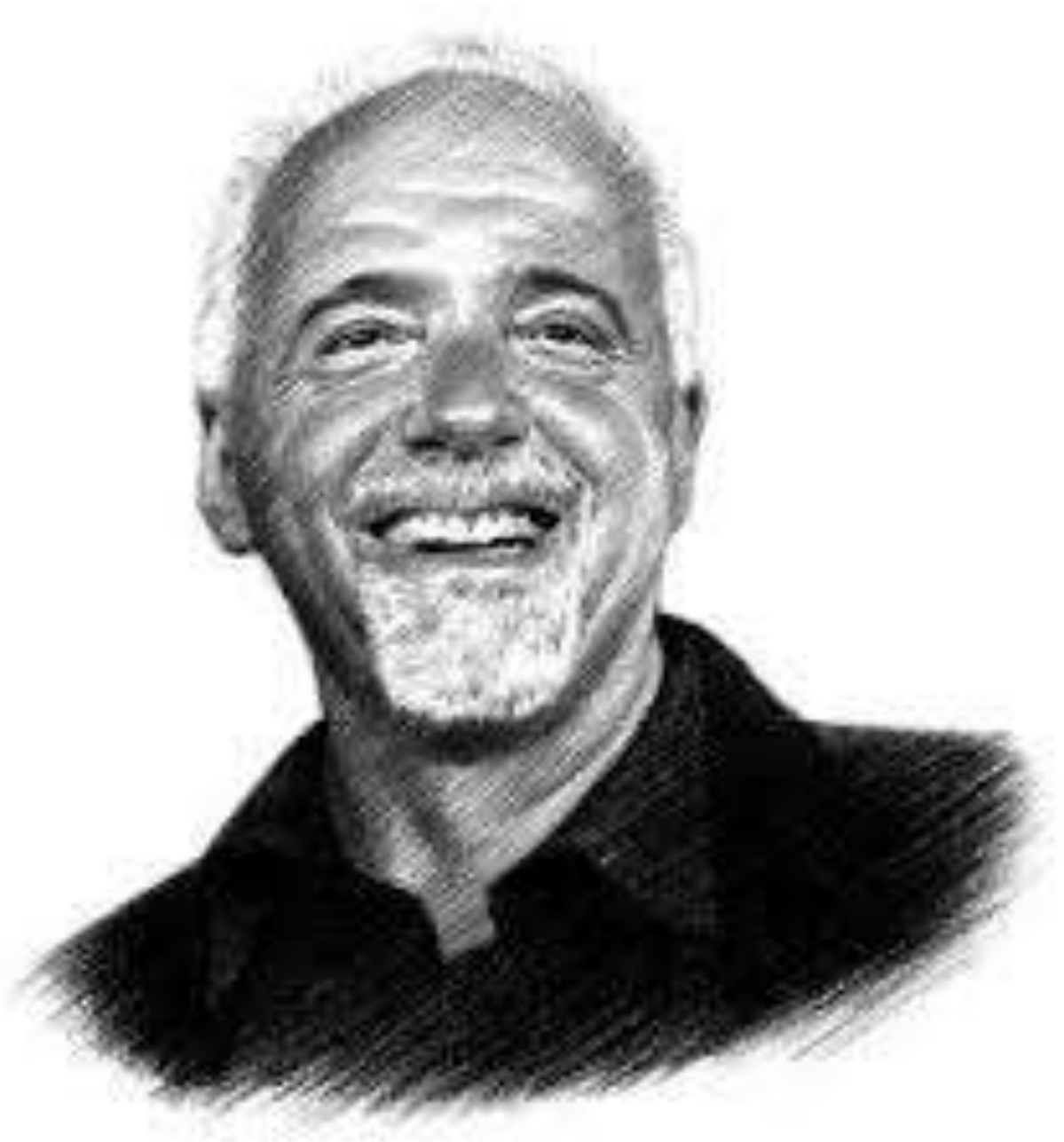
Tutores:

Ing. Julio César Espronceda Pérez
Ing. Carlos Amador Puentes Cruz

Cotutores:

Lic. Gladys Martínez Iglesia
Ing. Hernan Antonio Sánchez Guzmán

La Habana, 2019



*El camino es el que nos enseña la mejor forma de llegar
y nos enriquece mientras lo estamos cruzando.*

Pablo Coelho.

Dedicatoria

Dedico este trabajo a mis padres por haberme dado la vida y educarme para ser la mujer que hoy soy:

A mi mamá que durante estos años ha mostrado su preocupación y apoyo incondicional, por haber estado en todo momento ya fueran buenos o malos.

A mi papá, quien me quiso hasta el último momento, que a pesar de que la vida no le permitió estar a mi lado y me dejó cuando más lo necesitaba, estés donde estés debes estar orgulloso, soy la mujer que soñaste, gracias.

A mi abuela Melba que aunque no pueda comprender mucho siempre fue y será un ejemplo a seguir .

Agradecimientos

A mis padres Gonzalo y Lourdes por darme la luz de la vida y hacer de mí una mujer de bien, jamás podré pagarle lo mucho que me han dado.

A Melba, por ser la abuela más cariñosa y sacrificada del mundo, y brindarme su amor incondicional.

A Julio y Suleika, por ser mis segundos padres, gracias por el apoyo y la comprensión.

A todos y cada uno de mis amigos por soportar este carácter, comprenderme y ser mi paño de lágrimas.

A mis padrinos que a pesar que ya no los vea nunca olvidaré todos los momentos que pase a su lado, su amor y dedicación.

A mi novio Carlos por tratar de entenderme, por estar ahí siempre que lo necesito, por ser una persona tan maravillosa y por darme todo su amor y cariño.

A todos Gracias.

Declaración de Autoría

Declaro ser autor del presente Trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre este, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Lídice Martínez Cabrera

Firma de Autor

Ing. Julio César Espronceda Pérez

Ing. Carlos A Puentes Cruz

Firma de Tutor

Firma de Tutor

Lic. Gladys Martínez Iglesias.

Ing: Hernan Antonio Sánchez.

Firma de Cotutor

Firma de Cotutor

Resumen

Hoy en día no se puede hablar de eficiencia o novedad si no existe una aplicación y correcta utilización de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC). En relación al tema Cuba, ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las NTIC y lograr una cultura digital como una de las características imprescindibles del hombre nuevo. Tomando en consideración esa realidad mundial, se han desarrollado en el país múltiples programas encaminados a lograr la informatización de la sociedad. Como muestra de esto se encuentra la Universidad de las Ciencias Informáticas (UCI). La UCI cuenta con el proceso de extensión universitaria el cual se plantea desde diferentes ámbitos como son la actividad cultural, alcanzando su momento cumbre en la celebración de los festivales de artistas aficionados (FAA). Cabe destacar que su desarrollo trae consigo mucha información y en la actualidad no se gestiona de la mejor manera posible. En tal sentido, el presente trabajo de diploma tiene como objetivo desarrollar un sistema que contribuya a la gestión de la información estadística de los FAA en la UCI. Con tal motivo se realizó un estudio del framework destinado al desarrollo del sistema en cuestión. La propuesta de solución consiste en la construcción de un sistema que permita gestionar información sobre los FAA, además de generar reportes estadísticos y la exportar las planillas ya sea de las unidades artísticas, listado de los premios especiales o el listado de las unidades artísticas evaluadas de oro, plata, bronce o mención por facultades o a nivel de universidad.

Palabras clave: Extensión, Festival, Gestión, Información, Software.

Índice de contenido

Introducción	2
Capítulo 1: Fundamentación Teórica del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados	7
1.1. Introducción	7
1.2. Conceptos Asociados	7
1.2.1. La extensión universitaria como proceso	7
1.2.2. Dirección de extensión universitaria	8
1.2.3. Movimiento de Artistas Aficionados (MAA)	8
1.2.4. Festivales de Artistas Aficionados	8
1.2.5. Gestión de Información	9
1.2.6. Sistema de gestión de la información(SGI)	10
1.3. Estudio de Sistemas Homólogos	10
1.3.1. Ámbito Internacional	11
1.3.2. Ámbito Nacional	12
1.4. Metodologías, Herramientas, y Tecnologías	13
1.4.1. Metodología	13
1.4.2. AUP_UCI	15
1.4.3. Lenguaje Unificado de Modelación (UML)	15
1.4.4. Herramienta para el modelado	16
1.4.5. Lenguaje de Programación	16
1.4.6. Framework	18
1.4.7. Herramientas	19
1.4.8. Tecnologías	19
Capítulo 2: Características y diseño del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados	22
2.1. Introducción	22
2.2. Análisis	22
2.3. Características de la propuesta de solución	22
2.3. Modelo de dominio	23
2.3.1. Diagrama del modelo de dominio	24
2.4.1. Especificación de Requisitos del sistema	24
2.4.1. Requisito Funcional (RF):	25

2.4.2 Requisitos No Funcionales	26
2.5. Descripción de las historias de usuario	27
2.5.1 Historias de Usuarios	27
5.2.1 Estimación de esfuerzo por historia de usuarios	31
2.6. Plan de Iteraciones	32
2.6.1. Desarrollo del plan de iteraciones	33
2.7. Patrón Arquitectónico	34
2.8. Patrones de Diseño	35
2.8.1 Patrones	36
2.9. Modelo de datos.....	37
Capítulo 3: Implementación y prueba del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados	40
3.1 Introducción.....	40
3.2 Implementación.....	40
3.2.1 Estándares de codificación.	40
3.3 Diagrama de despliegue.....	41
3.4 Pruebas.....	42
3.4.1 Pruebas funcionales	43
3.4.2 Pruebas Unitarias	44
3.4.3 Pruebas Aceptación.....	48
Pruebas Alfa y Beta	48
3.5 Resultados de las pruebas aplicadas.....	49
3.6 Resultados de la solución implementada	50
3.7. Interfaces del Sistemas.....	52
Conclusiones generales.....	56
Recomendaciones	57
Bibliografía.....	58
Anexos	61
Anexo1(Historias de usuario)	61
Anexo 2 (Caso de pruebas (DCF))	69
Anexo3(Caso de pruebas (DCU)).....	74
Anexo 4. Modelo de encuesta aplicada a 10 especialistas del Centro cultural, para medir el nivel de satisfacción de la solución.	75

Índice de tablas

Tabla 1: Listado de requisitos funcionales.	26
Tabla 2: Historia de Usuario 1.	28
Tabla 3: Historia de usuario 2.	29
Tabla 4: Historia de usuario 3.	30
Tabla 5: Historia de usuario 4.	30
Tabla 6: Historia de usuario 5.	31
Tabla 7 : Estimación de esfuerzo por Historia de Usuario. "Elaboración Propia"	31
Tabla 8: Pan de iteraciones."Elaboracion Propia"	34
Tabla 9:Caso de prueba "Autenticar Usuario" (Fuente: Elaboración Propia).	43
Tabla 10: Variables de caso de prueba "Autenticar Usuario"(Fuente: Elaboración Propia).....	44
Tabla 11:Cálculo de la complejidad Ciclomática del método newAction() (Fuente: Elaboración propia).	45
Tabla 12 Caminos del grafo de flujo (Fuente: Elaboración propia).....	47
Tabla 13: Cuadro Lógico de Iadov Fuente(Elaboración Propia)	50
Tabla 14: Índices para el cálculo de satisfacción grupal (Fuente: (49)).	51
Tabla 15: Gestionar Unidad Artística	74

Índice de Ilustraciones

Ilustración 1: Patrón MVC.....	34
Ilustración 2:Modelo de dato.....	38
Ilustración 3:Diagrama de despliegue(Elaboración Propia).....	42
Ilustración 4: Resultados de Pruebas (Elaboración Propia).....	50
Ilustración 5:Pantallas principal.....	52
Ilustración 6: Pantalla Adicionar Jurado.....	53
Ilustración 7:Pantalla Adicionar Festival.....	53
Ilustración 8:Pantalla de la Búsqueda Avanzada.....	54
Ilustración 9:Pantalla Adicionar Unidad Artística.....	54

Introducción

El desarrollo de la informática y las telecomunicaciones han dado lugar a lo que de modo general se ha denominado las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC), las cuales en un proceso acelerado de convergencia penetran en diversos ámbitos de la vida humana. El trabajo, la escuela, el hogar, la distracción, son esferas en que las NTIC desempeñan cada día un mayor rol y por ello se aprecia su inserción en todos los procesos cotidianos, encontrándose fuertemente vinculadas a los cambios económicos, políticos y sociales de nuestra época, perfeccionando la automatización de la computación en la educación (Ledo, s. f.).

La informatización de procesos manuales o semi-informatizados, es una de las tareas que más ha evolucionado el entorno a finales del siglo XX y lo que va del XXI. Ha provisto al mundo de nuevas herramientas informáticas para la manipulación, transferencia, control y almacenamiento de la información. Esto permite a la sociedad interactuar mediante sistemas cada vez más simplificados y que logran incluir en ellos un amplio espectro de posibilidades y nuevas funcionalidades (Vázquez & Leyva, 2015).

Cuba es un país en desarrollo que no está exento de las necesidades del mundo actual, por consiguiente, tampoco de la nueva revolución de la tecnología digital. En los últimos tiempos ha estado sometida a profundos y considerables cambios en busca de un desarrollo paulatino de cada uno de sus sistemas incluyendo el educacional. La Batalla de Ideas ha jugado un papel preponderante en la creación de nuevos programas con vistas a incrementar la calidad de vida de cada uno de sus ciudadanos, al permitir la creación de varios proyectos educativos y sociales. Como parte de estos proyectos se tiene la fundación de la casa de altos estudios, Universidad de las Ciencias Informáticas (UCI). Esta constituye un pilar fundamental en la formación de nuevos profesionales comprometidos con la Revolución, el entorno social y con un alto desarrollo socio-político y humanista («Universidad de las Ciencias Informáticas», 2019).

La Universidad cuenta para el desarrollo de su estudiantes y profesores con varios procesos como son la formación, producción, investigación y como proceso transversal la extensión universitaria. Elemento integrador que facilita el flujo cultural continuo entre la Universidad y la Sociedad enriqueciéndolas mutuamente. («Programa Nacional de Extensión Universitaria.», 2004) La Extensión Universitaria tiene como objetivo promover la cultura en la comunidad intrauniversitaria y extrauniversitaria, para contribuir a su desarrollo cultural. («Programa Nacional de Extensión Universitaria.», 2004)

La Extensión Cultural como subproceso de la Extensión Universitaria, se plantea desde diferentes ámbitos o manifestaciones como el deporte, la cultura, la comunicación, la recreación sana y la formación de una cultura general e integral de la comunidad en interacción con la sociedad. En la universidad la actividad cultural se organiza desde la Vicerrectoría de Extensión Universitaria en coordinación con las Facultades docentes y las organizaciones estudiantiles. Para ello se cuenta con la labor de profesores e instructores de danza, música, teatro, artes plásticas y literatura quienes imparten talleres de creación y apreciación.

El Festival de Artistas Aficionados (FAA) de la FEU es el momento cumbre del Movimiento de Artistas Aficionados (MAA). Movimiento que se comenzó a organizar en Cuba a partir de la década del sesenta, con una amplia participación de los diferentes sectores estudiantiles: Pioneros, FEEM, FEU y laborales, a través de los sindicatos («Movimiento de Artistas Aficionados», 2019).

En el año 1963 se organizan en Cuba los primeros festivales con diferentes estructuras organizativas. En la UCI estos eventos comenzaron a realizarse en el 2003, cuando todas las facultades presentaron a sus talentos por primera vez, en marzo de ese año, en el antiguo teatro de Torrens.

Como parte de la infraestructura que garantiza el desarrollo de las artes, se encuentra el Centro Cultural Wifredo Lam que dispone de aulas especializadas para teatro, danza y música. Además, cuenta con una sala de video, galería de artes plásticas, librería, complejo recreativo, plazas y teatros, brindando así la calidad requerida para el desarrollo de los festivales.(«Universidad de las Ciencias Informáticas», 2019)

Los FAA en las diferentes manifestaciones artísticas presentan sus bases generales para el evento, que garantizan su ejecución y precisión interna, proporcionando una amplia documentación que se encuentra en estos momentos dispersa. La gestión de la información se hace de forma manual lo que trae consigo que se invierta mucho tiempo, recursos humanos y materiales, esta situación se ve agudizada al guardar la misma en archivos digitales, en una computadora provocando que en ocasiones no esté disponible, además se hace sumamente engorroso generar cualquier tipo de reporte, analizar y obtener algún tipo de estadística en un año, facultad, manifestación.

Esta información aumenta cada año con la celebración de los festivales, lo que ocasiona lentitud en el trabajo y dificultades en su control por lo que surge como **problema a resolver** ¿Cómo contribuir a la gestión de la Información estadística de los Festivales de

Artistas Aficionados en la Universidad de las Ciencias Informáticas? A partir del problema formulado se define como **objeto de estudio** los procesos de gestión de la información. Para darle solución el problema anterior se plantea como **objetivo** desarrollar un sistema que contribuya a la Gestión de la información estadística de los Festivales de Artistas Aficionados en la Universidad de las Ciencias Informáticas. Permitiendo identificar como **campo de acción** los procesos de gestión de la información estadísticas de los Festivales de Artistas Aficionados.

Para cumplir el objetivo se plantean las siguientes **tareas de la investigación**:

1. Análisis de los principales conceptos y trabajos relacionados con los sistemas de gestión de la información.
2. Análisis del entorno para el desarrollo del sistema.
3. Estudio de la metodología y las tecnologías a emplear en el proceso de desarrollo de software.
4. Análisis y diseño del sistema para la gestión de la información estadística.
5. Implementación del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados.
6. Validación y prueba del sistema informático para la gestión de la información estadística de los Festivales de Artistas Aficionados.

Una vez concluida la investigación se espera como posible resultado un sistema informático que contribuya a la gestión de la información estadística de los Festivales de Artistas Aficionados en la Universidad de las Ciencias Informáticas.

Empleándose diversos métodos investigativos para el cumplimiento de las tareas de investigación, clasificados en teóricos y empíricos. Los métodos teóricos permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación; de ellos se emplearon los siguientes:

- **Histórico-Lógico:** El método histórico analiza la trayectoria completa del fenómeno y el método lógico se basa en el estudio histórico del fenómeno (León & González, 2012). El empleo de este método posibilita realizar el estudio del estado del arte, investigar sobre los sistemas relacionados con la gestión de la información existentes, obteniendo así sus características y elementos fundamentales.

- **Análisis - Síntesis:** El análisis permite la división mental del fenómeno en sus múltiples relaciones y la síntesis establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas (León & González, 2012). La utilización de este método permite realizar el análisis teórico e identificar los principales conceptos a incluir en la fundamentación teórica y el análisis de la información, así como extraer los elementos importantes relacionados con los sistemas de gestión de la información.
- **Modelación:** La modelación es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad. El modelo es el eslabón entre el sujeto y el objeto intermedio. Este método se utiliza para reflejar la estructura, relaciones y características de la solución a través de diagramas, facilitando también el diseño y la comprensión de las clases necesarias para la implementación de la aplicación.

Los métodos empíricos representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a determinada elaboración racional; de ellos se emplearon los siguientes:

- **Análisis documental:** Para la revisión bibliográfica, la revisión de las fuentes básicas de información, el estudio de documentos clasificados, elementos teóricos que sustentan la concepción sobre la gestión de este proceso.
- **Entrevista:** La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información (León & González, 2012). Se emplea con asesores y miembros calificados del departamento de extensión universitaria para obtener ideas, referencias e información útil para elaborar los requisitos de la investigación.

El presente trabajo de diploma está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones generales, recomendaciones y referencias bibliográficas empleadas durante el desarrollo de la investigación, y por último para complementar la investigación se presentan una serie de anexos. A continuación, se muestra la descripción de los capítulos.

Capítulo 1. Fundamentos Teóricos. Se exponen los diferentes conceptos y términos necesarios para una mejor comprensión de la investigación. Realizando un análisis de las funcionalidades de los sistemas de gestión de la información y el estudio del *framework*

para el desarrollo del sistema. Además, se efectúa un estudio de las metodologías y herramientas de desarrollo que se utilizan para la propuesta de solución.

Capítulo 2. Características y diseño del sistema. Se explica el proceso de desarrollo de software que conlleva a la solución. Se exponen algunos de los elementos básicos a tener en cuenta para una solución exitosa como, requisitos funcionales y no funcionales, arquitectura del software y los distintos artefactos creados en el proceso, así como el modelo de datos.

Capítulo 3. Implementación y prueba. Se abordan los aspectos relacionados con la implementación del sistema a desarrollar. Se realiza la validación de la investigación desarrollada mediante la realización de pruebas de software, que garantizan que las funcionalidades cumplan las especificaciones requeridas.

Capítulo 1: Fundamentación Teórica del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados

1.1. Introducción

Durante una investigación es preciso obtener conocimiento teórico que facilite la realización de las actividades y así poder llevar a la práctica lo aprendido. Para ello se hace necesario realizar una profunda búsqueda bibliográfica con el fin de lograr una mayor comprensión del alcance de la investigación. En este capítulo se abordan los elementos teóricos que fundamentan la presente investigación. Así como análisis del estado del arte del tema propuesto, los conceptos asociados con los sistemas de gestión de la información. Se incluye además un análisis de la metodología, herramientas y tecnologías a emplear en el desarrollo de la propuesta de solución. Lo que favorece el desarrollo de la investigación y permite el cumplimiento del objetivo y la aplicación en sí, sean lo más óptimos posibles.

1.2. Conceptos Asociados

Para mejorar y ayudar a entender el desarrollo de la investigación es necesario definir y analizar ciertos conceptos de relevancia, que a su vez dan soporte teórico-metodológico y conceptual a la misma.

1.2.1. La extensión universitaria como proceso

La extensión universitaria es el proceso que tiene como objetivo promover la cultura en la comunidad intrauniversitaria y extrauniversitaria, para contribuir a su desarrollo cultural («Programa Nacional de Extensión Universitaria.», 2004)

Según Boaventura de Sousa Santos *“Inicialmente se le concebía como llevar el conocimiento universitario a la sociedad, extender la presencia de la universidad en la sociedad y relacionarla íntimamente con el pueblo. Sin embargo, su concepción ha cambiado a lo largo del tiempo hacia el establecimiento de un diálogo entre la universidad y la sociedad de manera que ambos actores se vean beneficiados”* (Santos, 2007) .

La Extensión Universitaria (EU) juega un papel importante dentro de las universidades, pues tiene como tarea principal el encargo social de salvaguardar, desarrollar y promover la cultura. Permite desarrollar un sistema de comunicación interna y externa que propicia el

diálogo, potencia la participación, facilita la difusión y divulgación de la cultura, contribuyendo así a la elevación del nivel cultural de la comunidad, su residencia y su entorno social. Dentro de la UCI la Extensión Universitaria (EU) posee varias aristas como son el deporte, la cultura, la comunicación, los proyectos comunitarios y socioculturales, para el desarrollo integral de los estudiantes y trabajadores.

1.2.2. Dirección de extensión universitaria

Es el primer órgano rector del proceso integrador y dinamizador del flujo cultural y la comunicación entre la universidad y la sociedad, así como también entre cada uno de sus miembros. En la UCI dentro de sus líneas de trabajo cuenta con la creación de proyectos socioculturales comunitarios con el fin de estrechar cada vez más los lazos de integración de la universidad con la sociedad. Este también cuenta con un movimiento de artistas aficionados (MAA) que permite estimular y cultivar la afición por el desarrollo de la extensión universitaria. Además, impulsa la creación y la promoción del arte como un medio en que el estudiante se sienta y comporte como universitario en el plano de su disfrute cultural e ideológico. Con un alto nivel de sensibilidad y de identidad con su centro («Universidad de las Ciencias Informáticas», 2019).

1.2.3. Movimiento de Artistas Aficionados (MAA)

La UCI posee un MAA integrado por estudiantes y profesores que trabajan en conjunto en talleres de creación y apreciación tanto de danza, música, teatro, artes plásticas y literatura. Este esfuerzo se evidencia en las cotas de calidad que se proponen alcanzar con su arte en los Festivales de Artistas Aficionados (FAA). Los festivales constituyen el momento cumbre del MAA, que se realiza de carácter anual en eventos a nivel de universidad y provinciales, así como nacional cada dos años. El MAA es reconocido por el Ministerio de Cultura, como uno de los de mayor nivel y calidad en el país, avalado por su sistematicidad y proyección social («Movimiento de Artistas Aficionados», 2019).

Para un artista aficionado poder participar en esta actividad, debe presentar una planilla de inscripción con un conjunto de datos del mismo, como: nombre y apellidos, carnet de identidad, número de solapín, facultad, género y año (en el caso de los graduados debe especificar el año de graduado).

1.2.4. Festivales de Artistas Aficionados

Dentro de los principales eventos que se desarrollan en el ámbito universitario para

promover el arte y la cultura se encuentran los Festivales de Artistas Aficionados de la FEU. Desarrollados en Cuba desde la década de los sesenta. En la UCI la celebración estos eventos comenzaron a realizarse en el 2003. Cuando todas las Facultades presentaron a sus talentos por primera vez, en marzo de ese año, en el antiguo teatro de Torrens, en las áreas que se encuentran después del Rectorado. Después la sede de los festivales se pasó para el área de Los Pinos, hasta que actualmente la mayoría se realizan en la Plaza Wifredo Lam. Hasta el momento, este centro de altos estudios ha celebrado 17 Festivales a nivel UCI, participado en 11 festivales provinciales (con 682 premios incluido 62 menciones y otros Premios Especiales) y en 8 nacionales, obteniendo 88 premios incluido 3 menciones y otros Premios Especiales como de Actuación, Vestuario, entre otros. Es importante destacar que, en la última edición del Festival provincial de la FEU, la UCI logró estar entre los tres primeros lugares en todas las manifestaciones de las artes en competencia y alcanzó el segundo lugar general. El desarrollo de estos festivales trae consigo una amplia gama de información ya sea de las unidades artísticas (talentos presentados en el festival), festivales o resultados del mismo que en la actualidad es necesario mejorar su gestión que proviene del latín gesto, hace referencia a la acción y a la consecuencia de administrar o gestionar algo.

1.2.5. Gestión de Información

Posiblemente la definición más mencionada sobre la Gestión de Información es la de (Lynda Woodman, 1985), en la cual se define que *“La gestión de la información es todo lo relacionado con: la obtención de la información adecuada, en la forma correcta, para la persona u organización indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta”* (Hernández, 1991).

Siguiendo la concepción de AJA *“La finalidad de la Gestión de la información es brindar elementos que permitan a la organización obtener, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, precisión y novedad suficientes para servir a los objetivos de la organización”* (Aja Quiroga, 2002).

Rojas Mesa, refiere que la gestión de información, *es el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve.* Tiene como elemento básico, la gestión del ciclo de vida de este recurso y se desarrolla en cualquier organización.

En particular, también se desarrolla en unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información (Torres Lebrato, 2015).

Se concluye que la gestión de la información es el proceso de organizar, evaluar, comparar, eliminar y divulgar la información existente en un contexto determinado, garantizando su integridad, confidencialidad y disponibilidad en el momento que sea necesario, propiciando los recursos precisos para una buena toma de decisiones. Posibilita que la información se encuentre más centralizada, organizada y disponible al personal a utilizar.

1.2.6. Sistema de gestión de la información(SGI)

Es un conjunto de sistemas y procedimientos que recopilan información de una variedad de fuentes, la compilan y la presentan en un formato legible. Los sistemas actuales de gestión de la información se basan en gran medida en la tecnología para recopilar y presentar datos, pero el concepto es más antiguo que las tecnologías informáticas modernas. El propósito principal de un sistema de gestión de la información es hacer que la toma de decisiones por parte de los especialistas sea más eficiente y productiva («Importancia del sistema de gestión de la información», s. f.).

Los sistemas de gestión de información constituyen hoy una alternativa de imprescindible presencia en cada organización al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización. («Gestión de la información para la mejora del proceso de toma de decisiones», s. f.)

Por lo tanto, un sistema de gestión de la información puede considerarse como un instrumento que permite la gestión de los recursos de información de una organización. Su finalidad es generar servicios que respondan a las necesidades de los usuarios, aprovechando al máximo sus recursos en función de la mejora continua y de la toma de decisiones.

1.3. Estudio de Sistemas Homólogos

Actualmente existen diversas soluciones para la gestión de la información. Pero en el ámbito de la presente investigación, se tomaron en cuenta solamente aquellas que permitieran la gestión de la información de procesos extensionistas (Festivales de Artistas Aficionados). A continuación, se relacionan sus características y principales

funcionalidades.

1.3.1. Ámbito Internacional

Sistema Nacional de Información de la Educación Superior (SNIES):

Es un sistema de información creado para responder a las necesidades de recopilar información de la Educación Superior en Colombia. Este sistema como fuente de información, en relación con las instituciones y programas académicos aprobados por el Ministerio de Educación Nacional, consolida y suministra datos, estadísticas e indicadores. Por la consolidación de información que hace el SNIES de las diferentes entidades del sector, es posible darle a conocer datos de interés a la comunidad en general sobre los avances en la Educación Superior colombiana. («SNIES - Sistemas información», 2012)

El sistema que se estudia no es factible para uso como solución de la presente investigación, ya que, se dedica a la gestión de la información generada en las instituciones de educación superior. No se dedica a gestionar específicamente los festivales de artistas aficionados de estas instituciones. Además, se gestiona la información de diferentes formas incluyendo en formato duro, siendo esto una de las problemáticas a resolver por la presente investigación.

Sistema para la Prevención de la Deserción en la Educación Superior

El Ministerio de Educación Nacional de Colombia propuso disminuir la deserción estudiantil en la educación superior como parte de la estrategia planteada para aumentar la cobertura, la calidad y la eficiencia educativas. Entre otras acciones, puso en marcha el Sistema para la Prevención de la Deserción en la Educación Superior —Spadies— ante la necesidad de contar con una visión sectorial e integrada de la problemática de la deserción, a partir de la cual se pudiera disponer de una conceptualización, una medición y una metodología de seguimiento del fenómeno aplicables a todas las instituciones de educación superior de ese país (CAN, 2012).

Este sistema le permite al Ministerio medir y monitorear los factores determinantes de la deserción, conocer su evolución en el tiempo y ver cómo se comportan diferentes instituciones y regiones. Igualmente, hace posible que cada institución cuente hoy con un perfil de sus estudiantes y con sistemas de alertas tempranas sobre los factores que los hacen vulnerables, lo que sirve para orientar de forma más eficiente apoyos y políticas. Además, permite hacer seguimiento estadístico a los niveles de deserción, seguimiento a

las condiciones de ingreso de los estudiantes, seguimiento al comportamiento de factores determinantes del fenómeno, estimación del riesgo de deserción para cada estudiante. Facilita la elección y evaluación de estrategias institucionales de apoyo a los estudiantes.(CAN, C. A., 2012).

1.3.2. Ámbito Nacional

Sistema automatizado para la gestión de la Extensión Universitaria

Sitio desarrollado en la Universidad de Pinar del Río por un colectivo de profesores de dicha institución. El sitio es una aplicación informática que potencia el proceso de gestión de la Extensión Universitaria de forma tal que favorece la promoción y el control de las actividades extensionistas. El sitio permite publicar información relacionada con los diferentes eventos que se van a estar efectuando, ya sean deportivos o culturales, mostrando en cada caso los deportes o categorías que los integran, la fecha en que se efectuarán, así como los participantes y los resultados que se obtienen. Posee un espacio dedicado a las noticias, lo cual permite a los usuarios mantenerse actualizados sobre los principales acontecimientos. Brinda la posibilidad, a cualquier usuario del sitio de lanzar diferentes convocatorias, las cuales van a ser publicadas con la finalidad de que estén disponibles para toda la comunidad estudiantil.(Pacheco Suárez, et al., 2012)

Este sitio no responde a algunas de las necesidades de la universidad, entre las cuales se encuentra la gestión de la información de los festivales de artista aficionados. Además, presenta una sección para publicar noticias lo cual no está entre los problemas a resolver de la presente investigación. También gestiona la información en la residencia estudiantil, no siendo útil debido a la existencia de un sistema en la universidad que gestiona dicha información.

Sistema informático de gestión universitaria (SIGUA)

Esta aplicación web fue diseñada con el objetivo final de facilitar el trabajo en los diferentes departamentos y secretarías docentes, al igual que a los estudiantes en el Instituto Superior de Arte. La interfaz Web que brinda esta aplicación permite agregar nuevos datos y realizar consultas a los mismos. La información introducida se almacena en la base de datos modelada para este fin(Quiñones Laffita, Laffita, Torres, & González, 2018)

Sistema de Gestión Universitaria.

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las

políticas, procedimientos y procesos de la organización. Para la realización del sistema se realiza una búsqueda y análisis de la bibliografía existente. A través del análisis y la síntesis, se identifican los principales elementos y rasgos que caracterizan a los sistemas de gestión académica, los diferentes procesos de formación, así como el papel preponderante del personal de secretaría de la Universidad. Se utiliza la observación directa del sistema de trabajo de los profesores, secretarías docentes, la secretaría general y de todo el personal implicado en el proceso de formación y el flujo de información en las diferentes áreas de la universidad para detectar las debilidades y fortalezas en esta área.(Bernal, Iñigo, & Llanes, 2010). Desarrollado por la dirección de Informatización perteneciente a la Vicerrectoría de Tecnología de la UCI, gestiona un grupo de información de las áreas docentes, productivas e investigativas de la universidad referente a Pregrado, Residencia, Postgrado, Cooperación, Teleformación, Biblioteca, Desarrollo, Tecnologías, Investigación, Extensión, Ingreso y Egreso.

Conclusiones parciales del estudio

Como se refleja en el estudio realizado a los sistemas anteriores, que de una forma u otra se relacionan con la gestión de información en el proceso extensionista. No satisfacen los problemas existentes en la Universidad. Se evidencia que en nuestros días no existe ningún sistema dedicado a la gestión de la información estadística de los festivales de artistas aficionados de la UCI. Por lo tanto, constituye una necesidad primordial para la Universidad desarrollar un sistema que permita gestionar este proceso y realizar todo tipo de reporte estadístico. También permitió analizar las tecnologías empleadas en el desarrollo de cada sistema

1.4. Metodologías, Herramientas, y Tecnologías

1.4.1. Metodología.

Desde el punto de vista informático *Méndez* define que una metodología de desarrollo de software es el conjunto de procedimientos, técnicas, herramientas y un soporte documental a la hora de desarrollar un producto de software, que indica quién, cuándo y cómo hacer algo (*Méndez, 2010*). La presencia de una metodología en el desarrollo de un proyecto garantiza un producto con más calidad y reduce el tiempo de entrega del producto ya que al tener una mejor planificación la producción permite cumplir con la fecha establecida. (Sommerville, 2005)

Existen dos tipos primordiales de metodologías, las Pesadas y las Ligeras:

Las pesadas se componen de numerosa documentación antes de empezar el proceso de implementación, teniendo mucho más peso del análisis y el diseño sobre el proyecto. Además, estas son más convenientes para proyectos grandes o cuyo rendimiento y nivel de calidad son críticos para el éxito de éste.

Las Ligeras permiten la comunicación cara a cara entre los miembros del proyecto y los clientes, además ofrecen una buena solución para proyectos de corta duración donde los requisitos no se conocen con exactitud son considerablemente prácticas, ya que no tienen en cuenta gran parte de la documentación y capacitadas para utilizarse en proyectos en los que los requisitos funcionales suelen variar con frecuencia durante el proceso de desarrollo del software .(Canós & Letelier, 2012)

Ejemplos de metodologías pesadas podrían ser:

- Rational Unified Process (RUP).
- ICONIX.
- Métrica 3.

Mientras que ejemplos de metodologías ligeras podrían ser:

- EXtreme Programming (XP).
- Crystal.
- SCRUM.
- Aup.

En la presente investigación se trabajará con la metodología Aup en su variante UCI la cual incorpora buenas prácticas de los grupos antes mencionado. Las metodologías para el desarrollo de software definen reglas a cumplir para lograr un proceso disciplinado sobre el desarrollo de software, con el fin de hacerlo más predecible y eficiente. La calidad del software se ve afectada directamente de manera positiva en cada una de sus fases de desarrollo, haciendo énfasis en la calidad y menor tiempo de construcción del software. Citando a Ph. D. Roger S. Pressman [Pressman (2010)] "(...) *producir lo esperado en el tiempo esperado y con el coste esperado (...)*". En la UCI, se utilizó la metodología Proceso Unificado Ágil (AUP, por sus siglas en inglés) para definir una variante (AUP-UCI) a ser utilizada por sus proyectos productivos. De esta manera, se adapta a los procesos

personalizados de la Universidad, proceso que se encuentra certificado con el nivel 2 de CMMI, respalda así la calidad del producto final.

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado, Gestión de Cambios, y Refactorización de Base de Datos para mejorar la productividad.

El proceso AUP establece un Modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP (RogerS. Pressman, 2010).

1.4.2. AUP_UCI

En el desarrollo de la aplicación se decidió utilizar la metodología de desarrollo de software variación de AUP para la UCI, una variante realizada por la Universidad de las Ciencias Informáticas a la metodología ágil AUP (Proceso Ágil Unificado) y está definida por la universidad como el documento rector de la actividad productiva. La metodología variación de AUP para la UCI está formada por tres fases, (inicio, ejecución y cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases (inicio, elaboración, construcción y transición) propuestas en AUP («Guía para el desarrollo de almacenes de datos.», 2016).

AUP-UCI propone 7 disciplinas: Modelado de negocio, requisitos, análisis y diseño, pruebas internas, pruebas de liberación. La metodología AUP-UCI será utilizada en su escenario: 4 Historia de usuario («Guía para el desarrollo de almacenes de datos.», 2016).

1.4.3 Lenguaje Unificado de Modelación (UML)

El lenguaje de modelado es cualquier lenguaje informático gráfico o textual que proporciona el diseño y la construcción de estructuras y modelos siguiendo un conjunto sistemático de reglas y marcos. El lenguaje de modelado es parte de y es similar al lenguaje artificial. («What is Unified Modeling Language (UML)?», s. f.) Por otra parte Lenguaje

Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje de modelado estandarizado que permite a los desarrolladores especificar, visualizar, construir y documentar artefactos de un sistema de software. Por lo tanto, UML hace que estos artefactos sean escalables, seguros y robustos en ejecución. UML es un aspecto importante involucrado en el desarrollo de software orientado a objetos. Utiliza la notación gráfica para crear modelos visuales de sistemas de software.(Barragan & Alejandro, 2016) Para nuestra investigación se empleará Lenguaje Unificado de Modelado en su versión 2.5.

1.4.4 Herramienta para el modelado

Visual Paradigm

Es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Es una potente herramienta multiplataforma de diseño y gestión fácil de usar. Proporciona a los desarrolladores de software la plataforma de desarrollo de vanguardia para crear aplicaciones de calidad. Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los IDE líderes, lo que supera a todo su proceso de desarrollo de código de modelo en esta solución integral. («Visual Paradigm Product Overview», 2016)

1.4.5. Lenguaje de Programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo.(«Lenguajes de programación», s. f.) El lenguaje de programación definido para el desarrollo de la propuesta de solución es PHP del lado del servidor y JavaScript, HTML, CSS, Twig del lado del cliente.

PHP (Hypertext Preprocessor):

PHP es uno de los lenguajes más utilizados actualmente en el desarrollo de aplicaciones web y viene experimentando un constante crecimiento en su nivel de utilización en Internet. PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad,

robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML.(Cobo, 2005)

JS (JavaScript):

JavaScript, fue diseñado para ser un lenguaje de elaboración script que pudieran incrustarse en archivos HTML. No es complicado, y fue creado para darle más dinamismo a las páginas. JavaScript es el lenguaje interpretado más utilizado, principalmente en la construcción de páginas Web, con una sintaxis muy semejante a Java y a C. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Es similar que PHP, aunque posee diferencias notables como, por ejemplo, los procesos de JavaScript se ejecutan del lado del cliente (en el navegador), y, por tanto, no implican intercambio de datos con el servidor.(Maza, 2012)

HTML (HyperText Markup Language):

Es un lenguaje de marcado de hipertexto (por sus siglas en inglés HyperText Markup Language), y se emplea para el desarrollo de aplicaciones Web. Cabe recalcar que no se trata de un lenguaje de programación puesto que no contiene funciones aritméticas, estructuras de control, entre otras características de un lenguaje de programación. HTML genera aplicaciones Web estáticas, aunque en conjunto con diferentes lenguajes de programación se puede crear aplicaciones dinámicas, HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje HTML. Esta nueva versión (aún en desarrollo), y en conjunto con CSS3, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándolo así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora muchas características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web (mediante los APIs). HTML5 está destinado a sustituir no sólo HTML 4, sino también XHTML 1 y DOM Nivel 2. Esta versión nos permite una mayor interacción entre nuestras páginas web y el contenido media (video, audio, entre otros) así como una mayor facilidad a la hora de codificar nuestro diseño básico.(«¿Qué es HTML5? | HTML5», s. f.)

CSS (Cascading Style Sheets):

Acrónimo recursivo en inglés de CSS Cascading Style Sheets (hojas de estilo en cascada)

es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Permite colocar colores de fondo, fuentes y marginando al contenido en múltiples columnas, entre otras funcionalidades relacionadas a la apariencia visual. («Capítulo 1. Introducción (Introducción a CSS)», s. f.) Para el desarrollo de la propuesta de solución es utilizado CSS3 que es su última versión.

Twig:

Es un moderno motor de plantillas para PHP, es rápido ya que compila las plantillas hasta el código PHP optimizado. La sobrecarga en comparación con el código PHP normal se reduce al mínimo. Tiene un modo de caja de arena para evaluar el código de la plantilla no confiable, esto permite que Twig se utilice como lenguaje de plantilla para aplicaciones donde los usuarios pueden modificar el diseño de la plantilla. Es impulsado por un lexer y un analizador flexible lo que permite al desarrollador definir sus propias etiquetas y filtros personalizados. («Home - Twig - The flexible, fast, and secure PHP template engine», s. f.)

1.4.6. Framework

Un framework simplifica el desarrollo de las aplicaciones, ya que automatiza muchos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. («1.1. Symfony en pocas palabras (Symfony 1.4, la guía definitiva)», s. f.)

Symfony

Es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. Symfony está desarrollado completamente con PHP y ha sido probado con éxito en sitios como Yahoo! Answers, delicious, DailyMotion y muchos otros sitios web de primer nivel. Symfony

es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux, etc.) como en plataformas Windows. («1.1. Symfony en pocas palabras (Symfony 1.4, la guía definitiva)», s. f.) Se decide utilizar Symfony en su versión 3.4.

1.4.7. Herramientas

Entorno de Desarrollo Integrado

NetBeans IDE es un entorno de desarrollo integrado (IDE por sus siglas en inglés), modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de aplicación. La programación mediante NetBeans se realiza a través de componentes de software modulares, también llamados módulos. Ayuda a los desarrolladores a escribir, depurar, refactorizar y probar su código. Constantemente valida la calidad del código y ofrece soluciones inmediatas para los problemas encontrados en todos los niveles, desde la instrucción individual para arquitectura global, utilizando las inspecciones de código avanzado y análisis de matriz de dependencia. Para la implementación de la propuesta solución se utiliza NetBeans IDE 8.2. («NetBeans IDE 8.2 Release Information», s. f.)

Gestor de Bases de Datos:

PgAdmin es la plataforma de desarrollo y administración de código abierto más popular y rica en funciones para PostgreSQL. Puede ejecutarse como una aplicación web o de escritorio. Es una herramienta multiplataforma capaz de ejecutarse en sistemas con Microsoft Windows, Linux y MacOS. Está diseñado para múltiples versiones de PostgreSQL y sus derivados. Presenta una extensiva documentación, además de una comunidad virtual. Dentro de sus herramientas destacan: («pgAdmin - PostgreSQL Tools», s. f.)

- Herramienta de consulta con resaltado de sintaxis en color.
- Visualización gráfica del plan de consulta.
- Depurador de lenguaje de procedimiento.

1.4.8 Tecnologías

Servidor de Bases de Datos

PostgreSQL es un potente sistema de base de datos objeto-relacional. Es una herramienta de código abierto con más de 30 años de desarrollo activo que le ha ganado una sólida reputación de confiabilidad, solidez de funciones y rendimiento. Se puede encontrar una gran cantidad de información que describe cómo instalar y usar PostgreSQL a través de la documentación oficial. La comunidad PostgreSQL proporciona muchos lugares útiles para familiarizarse con la tecnología, descubrir cómo funciona y encontrar oportunidades de carrera. («PostgreSQL: The world's most advanced open source database», s. f.)

Servidor Web

El servidor HTTP Apache es un servidor web HTTP de código abierto desarrollado por la Apache Software Foundation (ASF), para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache presenta entre otras características que es altamente configurable, bases de datos de autenticación y negociado de contenido, aunque es criticado por la falta de una interfaz gráfica que ayude en su configuración. Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Para el alojamiento de la propuesta solución se emplea el servidor web HTTP Apache en su versión 2.4.18. («About the Apache HTTP Server Project - The Apache HTTP Server Project», s. f.)

Doctrine

Doctrine es un mapeador de objetos-relacional (ORM) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un SGBD (sistema de gestión de bases de datos). Es uno de los más conocidos que interactúa con este lenguaje y su utilización brinda grandes facilidades a los desarrolladores a la hora de su trabajo con los datos, automatizando la generación de ficheros que responden a las tablas relacionales de la base de datos. («Doctrine, The Open-Source PHP ORM and Persistence Tools Project», s. f.)

Biblioteca

Una biblioteca es un conjunto de recursos (algoritmos) prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones. Las declaraciones de las funciones utilizadas en estas bibliotecas, junto con algunas macros y constantes predefinidas que facilitan su utilización, se agrupan en ficheros de nombres conocidos que suelen encontrarse con extensiones .lib, .js, .dll, entre otras. jQuery es una biblioteca de JavaScript que permite simplificar el proceso de desarrollo de una aplicación web.

jQuery: Es una biblioteca de JavaScript rápida y concisa que simplifica el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo web. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. («¿Qué son JavaScript y jQuery?», s. f.)

Control de versiones

Git es un sistema de control de versiones distribuido de código abierto y gratuito, diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia. Git es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido. Supera a las herramientas de SCM como Subversion, CVS, Perforce y ClearCase con características como bifurcaciones locales baratas, áreas de preparación convenientes y múltiples flujos de trabajo. («Git», s. f.)

Conclusiones del Capítulo

En el presente capítulo se abordaron elementos referentes al caso de estudio en cuestión, en tal sentido se puede arribar a las siguientes conclusiones:

- La definición de conceptos relacionados con el dominio del problema posibilitó una mayor comprensión del caso de estudio.
- El análisis de sistemas homólogos, permitió identificar las tendencias en el desarrollo de herramientas informáticas para la gestión de información; así como comprender la necesidad de desarrollar uno propio adaptado a las necesidades de la institución.
- El estudio de las principales tendencias tecnológicas permitió definir como metodología de desarrollo AupUci en su escenario 4; así como las herramientas principales, NetBeans, PgAdmin, Visual Paradigm, como lenguajes de programación HTML, Css, Php, JavaScript, Twig, Uml y tecnologías a utilizar para la implementación de la propuesta solución, PostgreSQL, Apache, Doctrine.

Capítulo 2: Características y diseño del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados

2.1 Introducción

En el presente capítulo se efectúa la propuesta de solución, luego de analizar el estudio del arte, las características de las herramientas y la metodología a ser utilizada en el desarrollo del sistema. La metodología AUP en su versión UCI define en cada uno de sus escenarios un grupo de artefactos que permiten una mejora en la comprensión del proceso de desarrollo del sistema.

Para el desarrollo de la propuesta se selecciona el escenario 4 donde se exponen los conceptos fundamentales del dominio y se presentarán los siguientes artefactos: Modelo de dominio, la especificación de requisitos funcionales y no funcionales que tendrá la propuesta de solución, se modelarán las historias de usuarios, así como una descripción detallada de las mismas.

2.2. Análisis

En esta fase se presentan las principales características asociadas a la propuesta de solución. Se realiza, además, un análisis de los principales conceptos asociados al dominio de la presente investigación evidenciando su relación por medio de un modelo de dominio.

2.3 Características de la propuesta de solución

Dadas las necesidades planteadas en la situación problemática de la presente investigación, la solución propuesta constituye un sistema para la gestión la información estadística de los Festivales de Artistas Aficionados.

Este sistema permite la organización, centralización y disponibilidad de la información estadística de los festivales de artistas aficionados en la Universidad de las Ciencias Informáticas, además de brindar información referente a festivales desarrollados en años anteriores y de los festivales que se estén celebrando en el momento.

Por otra parte, permitirá la creación de un festival para cada una de las facultades docentes en un año determinado.

- Adicionar las unidades artísticas pertenecientes al festival de una facultad. Asignándoles a cada unidad artística el listado de sus integrantes.

- Gestionar la información referente al jurado que evalúa las unidades artísticas. Jurado que emitirá los premios (Oro, Plata, Bronce, Mención o Premio Especial) alcanzados por cada unidad artística.
- Permite asignar un premio especial a los artistas destacados en el desarrollo del festival.

Una vez insertado el festival con las unidades artísticas correspondientes, el sistema permite realizar reportes estadísticos en dependencia de los filtros seleccionados por el usuario, brindando el listado de la información requerida por el mismo. Ejemplo (Listado de las unidades artísticas de la Facultad 1 con premio oro en el año 2019). Además, el sistema permitirá la exportación de las planillas de inscripción por unidades artísticas en formato PDF.

Además, el sistema exportara el listado de las unidades artísticas con premio (Oro, Plata Bronce o Mención), a nivel de facultad y universidad en formato PDF.

En otro sentido, se considera un sistema informático, porque a través de los dispositivos comunes de un sistema de cómputo, se podrá visualizar y manipular la información de los festivales de artistas aficionados. Para un mejor entendimiento en el desarrollo del capítulo se describen los artefactos que permiten conceptualizar el funcionamiento del sistema.

2.3 Modelo de dominio

Un modelo de dominio o modelo conceptual es una representación de conceptos del dominio de un problema determinado para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. El modelo de dominio es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir.

El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Cuando se realiza la programación orientada a objetos, el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio constituye una primera versión del sistema.(Peñalvo & Holgado, 2017)

2.3.1 Diagrama del modelo de dominio

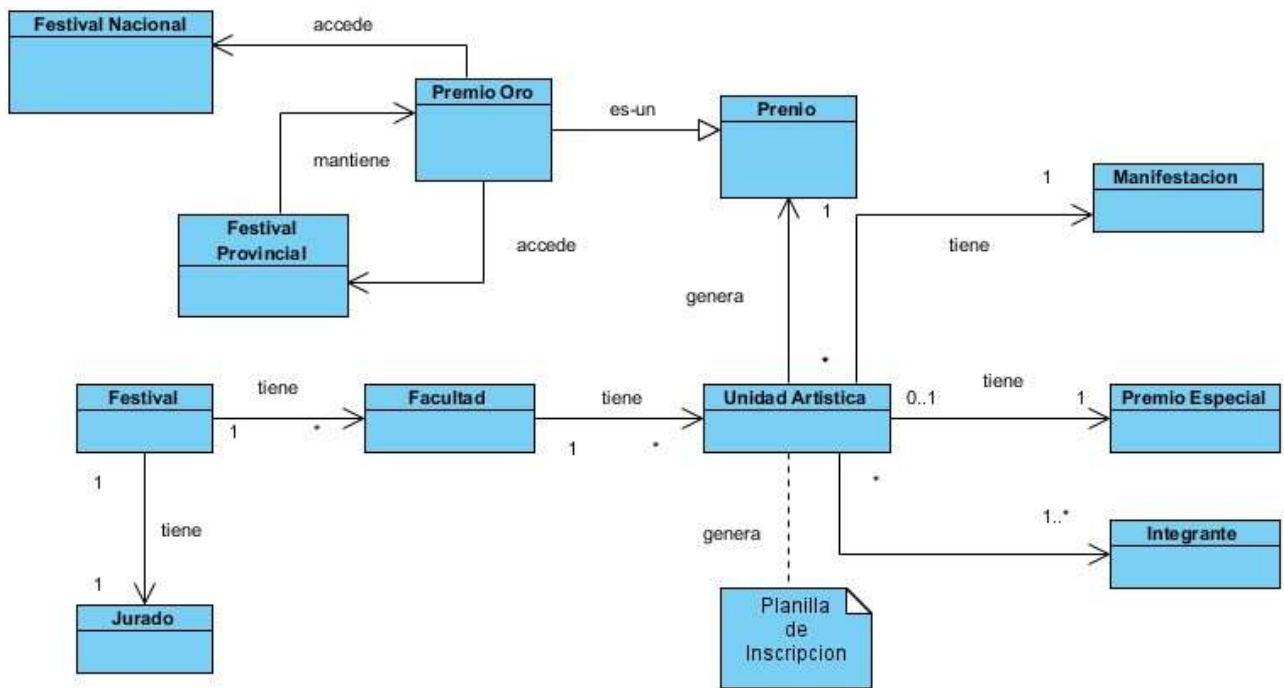


Figura 1: Modelo de dominio

2.4. Diseño

En esta fase se muestra los requisitos funcionales y no funcionales con lo que debe cumplir la propuesta de solución. Así como los artefactos generados a partir de la metodología seleccionada para guiar el proceso de desarrollo.

2.4.1 Especificación de Requisitos del sistema

Es el proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. La meta de esta es entregar una especificación de requerimientos de software correcta y completa. La misma apunta a mejorar la forma en que se comprenden y definen sistemas de software complejos, trata los principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadora de forma sistemática y repetible. («Vista de La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software», s. f.)-(Chaves, 2005)

2.4.1 Requisito Funcional (RF):

Característica requerida del sistema que expresa una capacidad de acción del mismo (una funcionalidad); generalmente expresada en una declaración en forma verbal. Un requisito funcional describe la interacción entre el sistema y su ambiente independientemente de su implementación. («Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix», s. f.) A continuación, los Requisitos Funcionales del Sistema.

REQUISITOS FUNCIONALES

No.	Nombre	Prioridad	Complejidad
RF1	Autenticar	Alta	Media
RF2	Crear Festival	Alta	Media
RF3	Modificar Festival	Alta	Alta
RF4	Listar Festival	Media	Media
RF5	Eliminar Festival	Media	Baja
RF6	Crear Unidad Artística	Alta	Alta
RF7	Modificar Unidad Artística	Alta	Alta
RF8	Listar Unidad Artística	Alta	Alta
RF9	Eliminar Unidad Artística	Media	Baja
RF10	Generar Reporte	Alta	Alta
RF11	Exportar Planilla	Alta	Alta
RF12	Crear Jurado	Alta	Media
RF13	Modificar Jurado	Media	Media

RF14	Listar Jurado	Media	Media
RF15	Eliminar Jurado	Media	Baja
RF16	Crear premio especial	Media	Media
RF17	Exportar listado premio especial	Alta	Alta
RF18	Exportar listado de UA con premio	Alta	Alta

Tabla 1: Listado de requisitos funcionales.

2.4.2 Requisitos No Funcionales.

Este tipo de requisito describe las restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos requisitos a menudo se aplican al sistema en su totalidad. («Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix», s. f.)

Usabilidad

RnF1: Tipo de Aplicación Informática.

La herramienta debe ser web, pero con características muy similares a las aplicaciones de escritorio en cuanto al diseño de las interfaces visuales y los tiempos de respuesta de la interacción del usuario con el sistema.

RnF2: El sistema debe poseer una interfaz fácil de utilizar para cualquier tipo de usuario con conocimientos básicos de computación en el manejo de ordenadores.

Finalidad.

RnF3: El objetivo que persigue la aplicación es que los usuarios puedan tener una herramienta que les permita interactuar de manera visual con los recursos que gestiona la información del festival.

RnF4. Hardware y software requerido para utilizar la aplicación.

- Sistema Operativo: Podrá utilizar Windows o Linux como Sistema operativo para el Sistema.
- Tener instalador de apache en su versión 2.4, en el caso de windows xamp en su

versión 7.2 o superior. Además, PHP en su versión 7.2 o superior.

- El sistema debe de ejecutarse sobre el navegador web Mozilla Firefox en su versión 59 o superior.
- La computadora que se utilice debe contar como mínimo con 2 GB de RAM y de almacenamiento 80 GB.

Seguridad:

RnF5.El acceso a la información debe estar restringido por usuario, contraseña y rol. Se podrá acceder a las funcionalidades del sistema solamente después de autenticarse.

RnF6.Cuando un usuario se autentique o registre en el sistema se le brindara la información correspondiente con su rol.

Restricciones de diseño e implementación:

RnF7.Marco de trabajo para el desarrollo del sistema del lado del cliente. El marco de trabajo que se utilizará en el desarrollo del Sistema es jQuery en su versión 3.3 ,Bootstrap en su versión 3.3,HTML5, CSS3, y JavaScript versión 1.8 ,del lado del servido tiwg2,php versión 7.2,,doctrine versión 2.5.

Interfaz

RnF8.El usuario deberá acceder a la aplicación a través del protocolo HTTP o HTTPS usando el navegador Firefox en sus versiones 59.0 o superior.

2.5. Descripción de las historias de usuario

Los requisitos funcionales describen lo que debe cumplir el sistema en un lenguaje técnico. Una historia de usuario (HU) es una representación de un requisito de software escrito en una o dos frases al utilizar el lenguaje común del usuario, son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, permiten responder rápidamente a los requisitos cambiantes, es una manera simple de describir una tarea concisa que aporta valor al usuario o al negocio. A continuación, se muestran algunas de las HU:

2.5.1 Historias de Usuarios

- **Autenticar**

Historia de usuario	
Número: 1	Nombre del requisito: Autenticar
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: alta	Puntos estimados: 1
Nivel de Complejidad: media	Puntos reales: 1
<p>Descripción: El sistema debe proporcionar que el usuario pueda autenticarse, insertando, su usuario y contraseña, validando que no deje campos vacíos y que los mismos sean correctos, luego de su autenticación el sistema debe re direccionar al usuario hacia la pantalla inicial.</p>	
<p>Observaciones: El sistema debe mostrar una notificación si el usuario y contraseña son incorrectos.</p>	
<p>Prototipo de Interfaz Gráfica:</p> 	

Tabla 2: Historia de Usuario 1.

- **Crear Festival**

Historia de usuario	
Número: 2	Nombre del requisito: Crear Festival
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: alta	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1
<p>Descripción: El sistema debe proporcionar que el cliente pueda crear un festival, mostrándole una</p>	

interfaz donde pueda llenar los campos de dicho festival. Debe validar que se encuentre todos llenos y correctos. Campos (Facultad, Año, Tipo de festival).

Observaciones: Los campos para crear un festival tienen que encontrarse llenos en su totalidad en el caso del año el cliente solo podrá insertar números de 4 cifras, correspondiente a un año.

Prototipo de Interfaz Gráfica:

El prototipo muestra un formulario con el título "Crear Festival". Hay tres campos de entrada:

- Facultad:** Un menú desplegable con el texto "-Seleccione-" y una flecha hacia abajo.
- Año:** Una línea de puntos "....." para ingresar un número de 4 cifras.
- Tipo de Festival:** Un menú desplegable con el texto "-Seleccione-" y una flecha hacia abajo.

Tabla 3: Historia de usuario 2.

- **Modificar Festival**

Historia de usuario	
Número: 3	Nombre del requisito: Modificar Festival
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: alta	Puntos estimados: 0.8
Nivel de Complejidad: alta	Puntos reales: 0.8
Descripción: El sistema debe proporcionar que el cliente pueda modificar un festival, que se encuentre creado, en caso de que algunos de sus campos estén incorrectos.	
Observaciones: Los campos para modificar un festival tienen que encontrarse llenos en su totalidad en el caso del año el cliente solo podrá insertar números de 4 cifras, correspondiente a un año.	
Prototipo de Interfaz Gráfica:	

Listado de festivales			
			<input type="button" value="Buscar"/> <input type="text" value="2014"/>
Año	Facultad	Tipo de Festival	Modifica
2014	Facultad 1	Universidad	
2014	Facultad4	Universidad	
2014	Facultad3	Universidad	
2014	Facultad 6	Universidad	

Tabla 4:Historia de usuario 3.

- **Listar Festival**

Historia de usuario																													
Número: 4	Nombre del requisito: Listar Festival																												
Usuario: Cliente	Iteración Asignada: 1																												
Prioridad en negocio: media	Puntos estimados: 0.6																												
Nivel de Complejidad: media	Puntos reales: 0.6																												
<p>Descripción: El sistema debe proporcionar que el cliente pueda obtener un listado del festival, que se encuentre creado a través de una interfaz gráfica.</p>																													
<p>Observaciones:</p>																													
<p>Prototipo de Interfaz Gráfica:</p>																													
<table border="1"> <thead> <tr> <th colspan="4">Listado de festivales</th> </tr> </thead> <tbody> <tr> <td colspan="3"></td> <td style="text-align: right;"> <input type="button" value="Buscar"/> <input type="text"/> </td> </tr> <tr> <th>Año</th> <th>Facultad</th> <th>Tipo de Festival</th> <th></th> </tr> <tr> <td>2014</td> <td>Facultad 1</td> <td>Universidad</td> <td></td> </tr> <tr> <td>2016</td> <td>Facultad4</td> <td>Universidad</td> <td></td> </tr> <tr> <td>2017</td> <td>Facultad3</td> <td>Universidad</td> <td></td> </tr> <tr> <td>2018</td> <td>Facultad 6</td> <td>Universidad</td> <td></td> </tr> </tbody> </table>		Listado de festivales							<input type="button" value="Buscar"/> <input type="text"/>	Año	Facultad	Tipo de Festival		2014	Facultad 1	Universidad		2016	Facultad4	Universidad		2017	Facultad3	Universidad		2018	Facultad 6	Universidad	
Listado de festivales																													
			<input type="button" value="Buscar"/> <input type="text"/>																										
Año	Facultad	Tipo de Festival																											
2014	Facultad 1	Universidad																											
2016	Facultad4	Universidad																											
2017	Facultad3	Universidad																											
2018	Facultad 6	Universidad																											

Tabla 5:Historia de usuario 4.

- **Eliminar Festival.**


Historia de usuario	
Número: 5	Nombre del requisito: Eliminar Festival
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: media	Puntos estimados: 0.6
Nivel de Complejidad: media	Puntos reales: 0.6
<p>Descripción: El sistema debe proporcionar que el cliente pueda obtener un listado del festival, que se encuentre creado a través de una interfaz gráfica., con la opción que se pueda eliminar algún festival.</p> <p>Observaciones: El sistema dará la opción de eliminar un festival si el usuario registrado tiene rol de administrador.</p>	
<p>Prototipo de Interfaz Gráfica:</p> 	

Tabla 6:Historia de usuario 5.

5.2.1 Estimación de esfuerzo por historia de usuarios

Su función principal es identificar problemas percibidos, proponer soluciones y estimar el esfuerzo que requieren implementar las ideas propuestas, consiste en la suma del tiempo de duración de las historias de usuarios, expresado en semanas. A continuación, se muestra la tabla de estimación de fuerza por las historias de usuario.

Tabla 7 : Estimación de esfuerzo por Historia de Usuario. "Elaboración Propia"

Historia de Usuario	Puntos estimados (semanas)

1	Crear festival	3
2	Editar festival	
3	Listar festival	
4	Eliminar festival	
5	Crear unidades artísticas	3
6	Editar unidad artística	
7	Listar Unidad Artística	
8	Eliminar Unida Artística	
9	Crear Jurado	3
10	Editar Jurado	
11	Listar Jurado	
12	Eliminar Jurado	
13	Generar reporte	1
14	Exportar Planillas	1
15	Crear Premio especial	1
16	Exportar Premio Especial	0.8
17	Exportar listado del jurado	1
18	Autenticación	1

Luego de realizar la estimación aproximada del tiempo de desarrollo se obtuvo un tiempo aproximado de 15 semanas equivalentes a 3 meses y 15 días.

2.6. Plan de Iteraciones

Una vez definidas las Historias de Usuario y realizada una previa estimación de los esfuerzos. Se procede a la planificación de la etapa de implementación del sistema. En el que se define un plan de iteraciones, donde se especifica la estimación aproximada con

que se implementarán las HU organizadas en cada una de las iteraciones. Teniendo en cuenta el esfuerzo asociado y las prioridades del cliente, se precisa una versión que sea de valor para este.

2.6.1. Desarrollo del plan de iteraciones

El plan tiene como objetivo mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU. Se definieron una primera iteración que consta de un total de 13 HU y una segunda de 5 HU para la implementación del Sistema.

Iteración	Historia de Usuario	Duración en semanas
1	Autenticar	1
	Crear Festival	1
	Modificar festival	0.8
	Listar Festival	0.6
	Eliminar Festival	0.6
	Crear Unidad Artística	1
	Modificar Unidad Artística	0.8
	Listar Unidad Artística	0.6
	Eliminar Unidad Artística	0.6
	Crear Jurado	1
	Modificar Jurado	0.8
	Listar Jurado	0.6
	Eliminar Jurado	0.6
	Generar Reporte	1

	Exportar Planilla	0.8
	Crear Premio especial	1
	Exportar Premio especial	1
	Exportar Listado Jurado	1
Total de	La duración en semanas	14.8

Tabla 8: Pan de iteraciones. "Elaboración Propia"

2.7. Patrón Arquitectónico

Symfony está basado en el patrón de MVC (modelo, vista y controlador), como se presenta en la Ilustración 1.

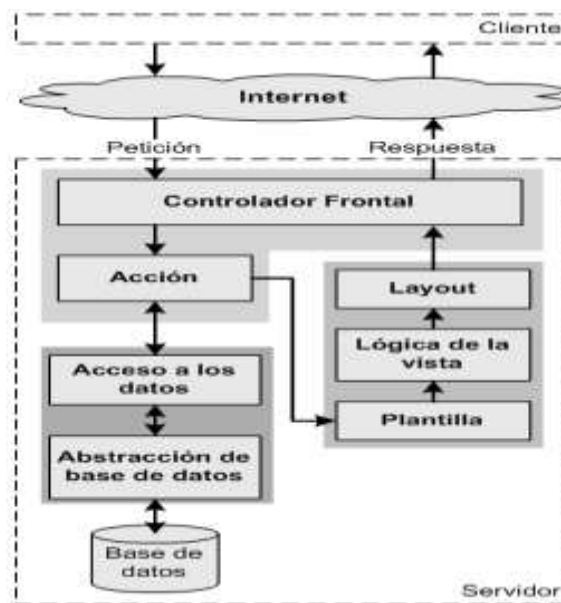


Ilustración 1: Patrón MVC.

El modelo es la lógica del negocio, que se encarga de buscar la información que requiere la aplicación, se comunica con la base de datos para obtener la información. En dicho Framework el almacenamiento de datos se realiza mediante la utilización de objetos por lo que no se accede de forma evidente a la base de datos por esta razón posee un alto nivel de abstracción. («MVC (Model, View, Controller) explicado.», s. f.)

Vista es la interfaz de la aplicación en donde el usuario puede manipular o interactuar con

el software, debido a que el modelo es transformado en una página web que sea amigable. Para la realización de la vista Symfony utiliza principalmente la plantilla twig que es un motor moderno de php. Esta plantilla tiene tres principales características las cuales son:

1. **Rápido:** Twig compila la plantilla de Php de forma optimizada y sin formato.
2. **Seguro:** Twig utiliza sandbox que permite evaluar el código por lo que el usuario puede modificar el diseño de la plantilla con confianza.
3. **Flexible:** Twig permite que el desarrollador defina sus propias etiquetas y filtros personalizados. («MVC (Model, View, Controller) explicado.», s. f.)

Controlador es el que se encarga de procesar la petición del usuario y realizar los cambios necesarios en el modelo o la vista, debido a que es el intermediador para la comunicación entre la vista y el modelo. El controlador está dividido en varios componentes los cuales son:

El controlador frontal es el que se encarga de aplicar la configuración y determinar las acciones que se pueden ejecutar.

Verifica las peticiones del usuario y prepara los datos en la capa de presentación.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

El uso del patrón arquitectónico en la propuesta de solución permite separar las clases encargadas del modelo de negocio como Festival, Unidad Artística, Premio especial, Usuario, Jurado, de las clases encargadas de controlar las peticiones del usuario como FestivalController, UnidadArtisticaController, PremioController entre otras y del código encargado de mostrar los datos como edit.html.twig, show.html.twig, new.html.twig. Además, facilita la creación de una estructura de directorios dentro de cada Bundle para separar los objetos de modelo, la vista y el controlador.

2.8. Patrones de Diseño

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el

desarrollo de software”(«Que es un patron de diseño», s. f.). En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: su nombre, cuándo aplicarlo, la solución y las consecuencias (costos y beneficios).

2.8.1 Patrones

Gof

Gang of Four (GoF, en español la pandilla de los cuatro) formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, son los autores del libro "Design Patterns: Elements of Reusable Object Oriented Software", según GoF los patrones de diseño "describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos". Presentan un conjunto de 23 patrones de diseño clasificados a partir de dos criterios: propósito y alcance, y definen además las siguientes categorías (Guerrero, Suárez, & Gutiérrez, 2013):

Patrones Creacionales: Inicialización y configuración de objetos.

Patrones Estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.

Patrones de Comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos. A continuación, se describen los patrones utilizados.

Observer (observador): Hace un uso sofisticado de la orientación a objetos. Define una dependencia de uno a muchos, de forma que cuando el objeto escuchado cambia su estado, el patrón se encarga de que todos los oyentes sean notificados de esos cambios. Está estrechamente vinculado a la invocación implícita. El patrón observador garantiza la comunicación entre los componentes. El presente patrón es utilizado en los mensajes flash, ejemplo de estos mensajes “Se ha creado correctamente un festival”.

Facade (fachada): Pertenece al grupo de los patrones estructurales, provee de una única interfaz para acceder a un sistema completo, que actúa como único punto de acceso al mismo, y hace que éste sea más fácil de utilizar, por lo que el cliente trabaja con la interfaz sencilla sin estar preocupado de las complejas operaciones que se realizan por el sistema(«Thesis | Conceptual Model | Communications Protocols», s. f.). Las clases controladoras que heredan de la clase Controller la cual implementa la interfaz ContainerAwareInterface, sencilla y cómoda para el usuario, mientras por detrás se ejecutan los procesos que garantizan el funcionamiento.

Peso Mosca (flyweight): Es un objeto compartido que puede ser usado en diferentes contextos simultáneamente de forma transparente e indistinguible. Existe una clase BaseController en el sistema que cuenta con todas las acciones, las controller al necesitar una acción hereda de la clase BaseController, es decir, pasa a ser hijo del mismo, por lo que se comporta como él y puede ejecutar la acción deseada, esto garantiza que toda la implementación este sobre un solo objeto, por lo que los modelos son más ligeros.

PatronesGrasp

Creador. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos (una tarea muy común). La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación. La ventaja de utilizar este patrón es eliminar la dependencia entre las clases logrando una mayor mantenibilidad y reutilización. Dentro del sistema se evidencia por sus acciones en la clase controladora “FestivalController.php”, la cual crea instancias de la entidad festival.

Alta Cohesión: Verifica que la información que almacena una clase debe ser coherente y debe estar, en la medida de lo posible, relacionada con la clase. En la solución propuesta las clases clientes y las servidoras, se encargan cada una de realizar una función específica, permitiendo que las mismas no se saturen y a través de la asignación de responsabilidades se garantiza la alta cohesión en las clases existentes.

Bajo Acoplamiento: Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Controlador: Este patrón se evidencia en las clases FestivalController, Unidad Artística Controller entre otras del ambiente.

2.9. Modelo de datos.

El modelo de datos determina la estructura lógica de una base de datos y el modo de almacenar, organizar y manipular los datos. Dentro de sus propósitos se encuentra definir los datos persistentes que serán almacenados(RogerS. Pressman, 2010)

Con el objetivo de definir las clases persistentes se identifican los conceptos, en el dominio del negocio, que persisten en el tiempo. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo, por lo que no todos los conceptos definidos son transformados en clases de este tipo.

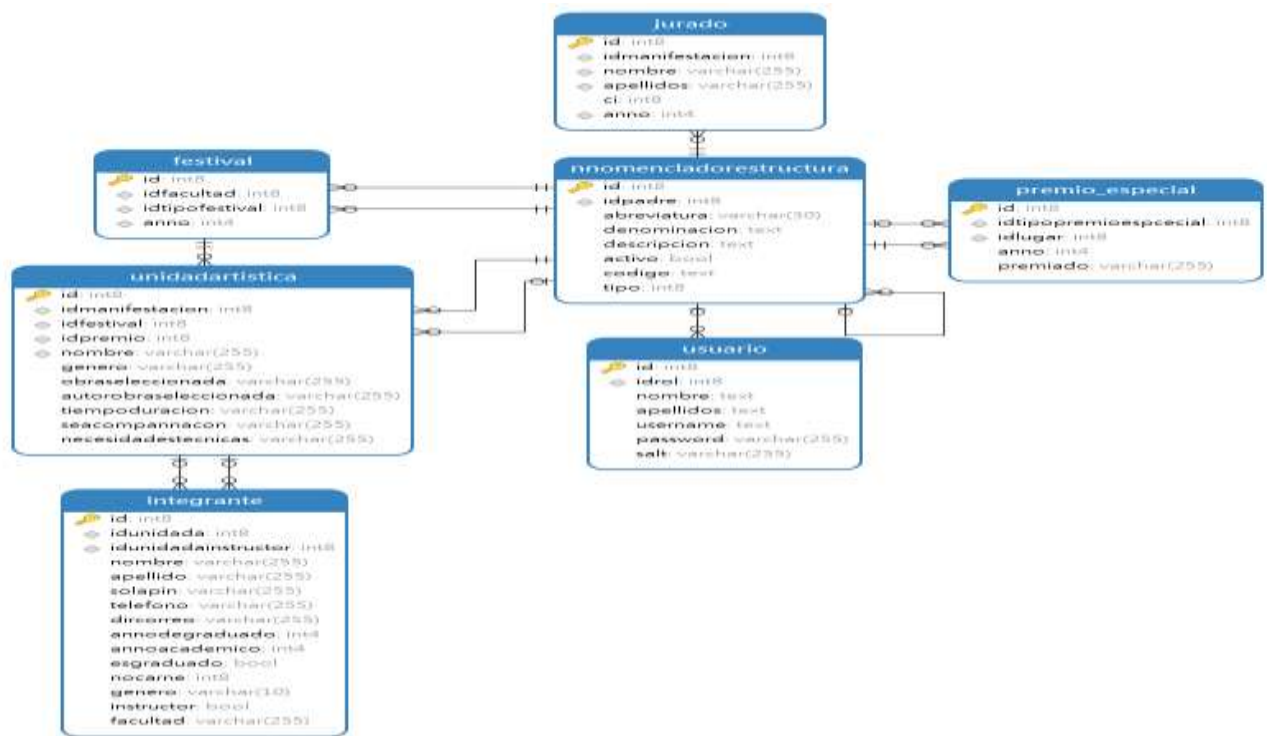


Ilustración 2: Modelo de dato

Unidad Artística: Almacena las características de una unidad (Nombre, Género, Autor etc.), así como las relaciones con, la facultad, la manifestación, el premio y los integrantes que compone la misma.

Integrante: Almacena las características de los integrantes (Nombre, Solapín, Correo electrónico etc.), así como la relación con la unidad artística a la que pertenece.

Festival: Almacena las características de un festival (Año), así como la relación con la facultad a la que pertenece y el tipo de festival. En caso de que el tipo de festival sea Provincial o Nacional, solo almacena en año y el tipo de festival.

Jurado: Almacena las características de un jurado (Nombre, Apellido, CI, Año), así como la relación con la manifestación a evaluar.

Usuario: Almacena las características de un usuario (Nombre, Apellido, Usuario, Contraseña), así como la relación con el rol que desempeña.

Premio especial: Almacena las características de los premios (Año, Premiado), así como la relación con el tipo de premio especial y el lugar alcanzado.

Nomenclador: Almacena las características de todos los nomencladores del sistema.

Conclusiones del Capítulo

En el presente capítulo se desarrolló el modelo de dominio donde se definieron los principales conceptos del entorno y sus relaciones, el diseño del modelo posibilitó la comprensión del contexto del sistema.

- La definición de los requisitos funcionales y no funcionales del sistema permitió tener una visión más clara sobre las características y restricciones que debe cumplir el mismo.
- El diseño de los prototipos de interfaz de usuario permite guiar el proceso de desarrollo de las interfaces y validar los requerimientos funcionales. Las características abordadas sobre el diseño del sistema permitieron establecer la comprensión de cada elemento (arquitectura, clases, modelo de datos) que componen al mismo.
- Se especificó el proceso de implementación del sistema a partir del desglose de las historias de usuario en tareas de ingeniería, lo que permitió especificar los procedimientos necesarios para dar cumplimiento a cada historia de usuario.
- Se definió la arquitectura MVC como arquitectura a aplicar por las facilidades que proporciona, así como se utilizaron varios patrones para el diseño que facilitan la reutilización y mejor organización de la propuesta de solución que contribuyen al perfeccionamiento de la implementación del código.

Capítulo 3: Implementación y prueba del sistema para la gestión de la información estadística de los Festivales de Artistas Aficionados

3.1 Introducción

En el presente capítulo se abordan los principales aspectos relacionados con la implementación de la solución propuesta. Los estándares de codificación se definen para lograr una mejor comprensión y estandarización del código resultante. La verificación del correcto funcionamiento del software se realiza mediante la aplicación de las pruebas correspondientes.

3.2 Implementación

La etapa de implementación del desarrollo de software es el proceso de convertir una especificación del sistema en un sistema ejecutable. Los procesos de diseño y programación de software se encuentran implicados en esta etapa; si se utiliza un enfoque evolutivo de desarrollo, también puede incluir un refinamiento de la especificación del software(Sommerville, 2005).

3.2.1 Estándares de codificación.

Los estándares de codificación son un conjunto de convenciones (denominaciones, formatos) establecidas para la escritura de código. Estos estándares varían en dependencia del lenguaje de programación y son necesarios para los programadores y miembros del equipo de trabajo. Generalmente el software no es mantenido por el/los autor(es) por ende debe ser revisado y/o manejado por otros. Los estándares posibilitan una mejor lectura e interpretación del software y su empleo permite aplicar un conjunto de lineamientos (Ampuero & López., 2010). Los siguientes estándares de codificación son empleados durante el desarrollo de los componentes:

- **Tamaño máximo de línea:** La longitud máxima recomendable para una línea de código es de 140 caracteres.
- **Clases:** El nombre de la clase debe estar en concordancia con el nombre del archivo y debe declararse en notación PascalCase.
- **Funciones y métodos:** Los nombres de funciones deben empezar siempre con una letra minúscula, utilizando el formato camelCase.

- **Variables:** Los nombres de variables pueden contener caracteres alfanuméricos. Los números están permitidos en los nombres de variable, pero no se aconseja en la mayoría de los casos. El formato definido para los nombres de variables debe ser camelCase.
- **Declaración de clases:** La llave "{" debe escribirse siempre a continuación del nombre de la clase.
- **Declaración de funciones y métodos:** La llave "{" debe escribirse siempre seguido del nombre.
- **Uso de Funciones y Métodos:** Los argumentos de la función tienen que estar separados por un único espacio posterior después del delimitador coma.
- **Sentencias de Control:** Las sentencias de control basadas en las construcciones if y elseif deben tener un solo espacio en blanco antes del paréntesis de apertura del condicional y un solo espacio en blanco después del paréntesis de cierre. La llave de apertura "{" se escribe en la misma línea que la sentencia condicional. La llave de cierre "}" se escribe siempre en su propia línea.
- **Operadores:** Se debe colocar un espacio antes y después de los operadores binarios (+, -, +=, !=, ==, etc.).
- **Comentarios:** Los comentarios deben ser añadidos de forma que resulten prácticos, para explicar el flujo del código y el propósito de las funciones y variables. Utilizar los caracteres "//" para una sola línea, en caso de tener más de una línea comenzar con "/*" y cerrar "*/".

3.3 Diagrama de despliegue

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar la disposición física de los componentes que integran al sistema y las relaciones entre sus componentes.

- Permiten modelar la disposición física o topología de un sistema.
- Muestra el hardware usado y los componentes instalados en el hardware.
- Muestra las conexiones físicas entre el hardware y las relaciones entre componentes.

A continuación, se presenta el diagrama de despliegue de la solución, donde se muestran componentes a utilizar por el desarrollador para desplegar el sistema en cuestión.

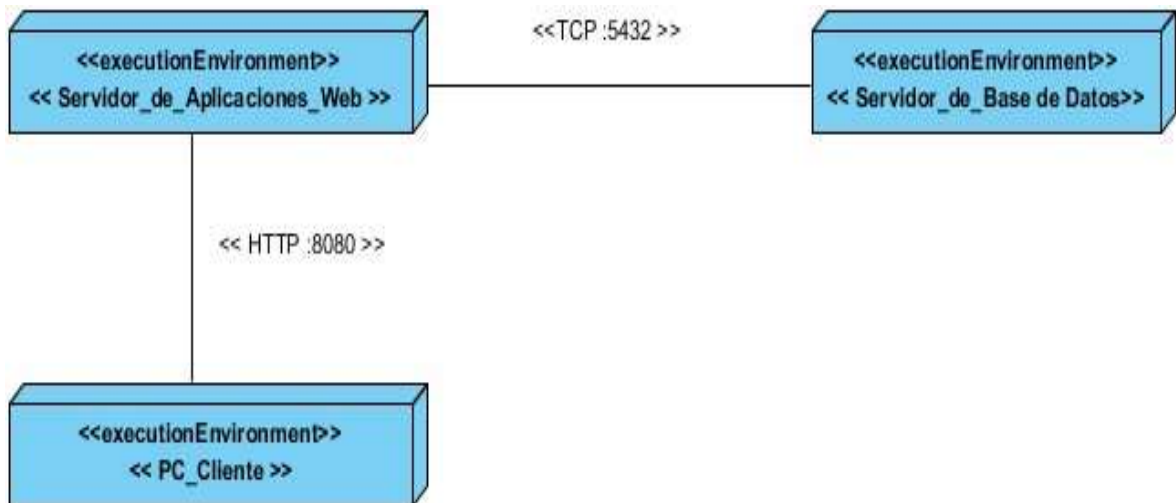


Ilustración 3: Diagrama de despliegue (Elaboración Propia).

Este diagrama se considera para lograr un despliegue exitoso de la aplicación. En este se definen:

Dispositivo Cliente: Se refiere a las estaciones de trabajo (Cliente: PC) que el usuario utilizará para conectarse, vía HTTP, con el servidor de aplicaciones web.

Servidor de Bases de Datos: Almacena toda la información que brinda el sistema. La información es obtenida o modificada en dependencia del nivel de privilegio del usuario que realiza la petición. La comunicación con el servidor de aplicaciones es a través del protocolo TCP empleando el puerto 5432.

Servidor Web: Es el encargado de brindar la interfaz del sistema para que los usuarios puedan hacer uso de este, almacena todo el código fuente del sistema y se comunica por medio de los protocolos TCP con el servidor de bases de datos.

3.4 Pruebas

Las pruebas de software son un elemento que forman parte del proceso de verificación y validación. La verificación se refiere al conjunto de tareas que garantizan que el software implementa correctamente una función específica. La validación es un conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos del cliente (Roger Pressman, 2002). Las pruebas dirigidas al sistema tienen como objetivo evaluar el grado de cumplimiento de los requerimientos del mismo, a través de la realización de los casos de pruebas permitiendo detectar el mayor número de no conformidades y solucionarlas antes de la entrega del software.

3.4.1 Pruebas funcionales

Las pruebas funcionales son diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que vamos a probar, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas. Las pruebas funcionales se basan en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. El objetivo principal de las pruebas funcionales es analizar el producto terminado y determinar si hace todo lo que debería hacer y si lo hace correctamente. El método utilizado para llevar a cabo este tipo de prueba es caja negra (Roger Pressman, 2002).

Método de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software. Las técnicas de prueba de caja negra permiten derivar conjuntos de condiciones de entrada que revisarán los requerimientos funcionales para un programa (Roger S. Pressman, 2010). El método de caja negra presenta varias técnicas de prueba como son: partición de equivalencia, análisis de valores límites y grafos de causa-efecto.

En la presente investigación se utilizará específicamente dentro del método de caja negra la técnica de partición de equivalencia generando los casos de pruebas de dicha técnica sobre las diferentes interfaces que responden a los requisitos funcionales.

A continuación, se presenta el diseño de caso de pruebas (DCP) del requisito "Autenticar" donde se analizarán las variables y condiciones que puedan determinar la respuesta del sistema. En el anexo 2 se pueden consultar los casos de pruebas correspondientes a los restantes requisitos funcionales.

Tabla 9: Caso de prueba "Autenticar Usuario" (Fuente: Elaboración Propia).

Escenario	Descripción	Variables		Respuesta esperada	Respuesta
		Usuario	Contraseña		
EC 1.1. Autenticar usuario correctamente	El usuario debe introducir los datos de los campos seleccionados y dar clic a en	V admin	V Administrador	Se accede al sistema / seleccionar el botón autenticarse / Inserta usuario y contraseña /Mostrar un	El sistema autentica correctamente y muestra un mensaje de confirmación

	botón autenticar			mensaje de confirmación de la autenticación / accede a la interfaz para poder gestionar la información del festival	
EC 1.2. Autenticar usuario Incorrectamente .		I Admin v admin I jbgkghklshfio	V administrador I Administrador I Fgfgfhturyfj	Se accede al sistema / Selecciona el botón de autenticarse / Insertar usuario y contraseña /Mostrar un mensaje con la notificación q esta incorrecto el usuario o la contraseña	. El sistema muestra un mensaje con la negación del usuario autenticado y especificando que el usuario o contraseña están incorrecto

Tabla 10: Variables de caso de prueba "Autenticar Usuario"(Fuente: Elaboración Propia)

No.	Variable	Valor Nulo	Descripción
1	Usuario	No	Es un campo que permite al usuario insertar su usuario correctamente. Acepta solo letras y no acepta espacio entre letras.
2	Contraseña	No	Es un campo que permite al usuario insertar su contraseña correctamente. Acepta cualquier carácter alfa numérico.

3.4.2 Pruebas Unitarias

Las pruebas unitarias son una forma de probar pequeñas e individuales porciones de código. A través de ellas se puede verificar que determinado módulo o funcionalidad se ejecuta dentro de los parámetros y especificaciones concretadas en los requisitos. El objetivo principal de una prueba unitaria es comprobar el correcto funcionamiento de una

unidad de código. El método utilizado para realizar este tipo de prueba se denomina caja blanca (Roger Pressman, 2002)

Método de caja blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, consiste en probar el código. Se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
2. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los métodos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. Entre las técnicas de pruebas existentes en caja blanca se encuentran: prueba del camino básico, prueba de condición, prueba de flujo de datos y prueba de bucles. En el presente trabajo se utilizará la técnica del camino básico. La aplicación de esta técnica permite: generar el grafo de flujo, calcular la Complejidad Ciclomática (CC), determinar los caminos linealmente independientes y diseñar los casos de prueba para forzar la ejecución de cada camino del conjunto básico (Roger S. Pressman, 2010).

La técnica del camino básico fue aplicada a los métodos del sistema. Luego de la determinación de los nodos y flujos de control del código se obtuvo el grafo de flujo y se calculó la complejidad Ciclomática del algoritmo según muestra la tabla 11.

Tabla 11: Cálculo de la complejidad Ciclomática del método newAction() (Fuente: Elaboración propia).

Método:

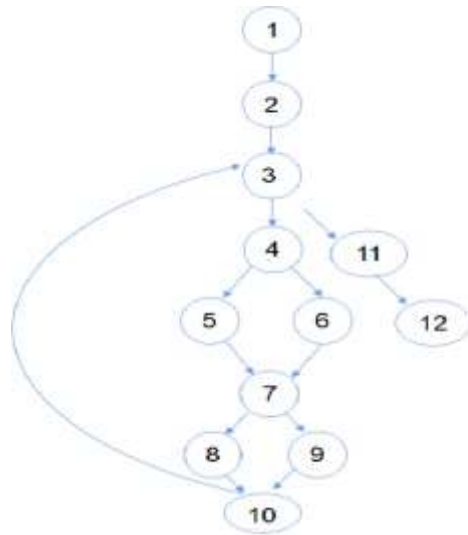
```
public function newAction(Request $request, TranslatorInterface $translator) {
    $unidadA = new UnidadA();
    $form = $this->createForm(UnidadAType::class, $unidadA);
    $form->handleRequest($request);
    if ($form->isSubmitted()) {
        $datos_integrante_request = $request->get('appbundle_unidadA')['integrante'];
        $em = $this->getDoctrine()->getManager();
        try {
            $em->persist($unidadA);
            foreach ($datos_integrante_request as $data) {
                $integrante = json_decode($data['integrante'], true);
            }
        } catch (Exception $e) {
            // ...
        }
    }
}
```

```

    $integranteBD = new Integrante ();
    $integranteBD->setNombre($integrante['nombre']);
    $integranteBD->setApellido($integrante['apellidos']);
    $integranteBD->setFacultad($integrante['facultad']);
    $integranteBD->setNocarne($integrante['ci']);
    $integranteBD->setTelefono($integrante['telefono']);
    $integranteBD->setDircorreo($integrante['correo']);
    $integranteBD->setGenero($integrante['genero']);
    $integranteBD->setSolapin($integrante['solapin']);
    if ($integrante['esgraduado']) {
        $integranteBD->setEsgraduado(true);
        $integranteBD->setAnnodegraduado($data['anno']);
    } else {
        $integranteBD->setEsgraduado(false);
        $integranteBD->setAnnoacademico($data['anno']);
    }
    if ($integrante['esinstructor']) {
        $integranteBD->setInstructor(true);
        $integranteBD->setUnidadainstructor($unidadA);
    } else {
        $integranteBD->setInstructor(false);
        $integranteBD->setUnidadA($unidadA);
    }
    $em->persist($integranteBD);
}
$em->flush();
} catch (\Exception $e) {
    if ($e instanceof UniqueConstraintViolationException) {
        $this->addFlash('danger', 'formularios.msgvalidacion.unidadA');
        return $this->render('@App/unidadA/new.html.twig', array(
            'unidadA' => $unidadA,
            'form' => $form->createView(),
        ));
    }
    $this->addFlash('danger', $e->getMessage());
    return $this->render('@App/unidadA/new.html.twig', array(
        'unidadA' => $unidadA,
        'form' => $form->createView(),
    ));
}
$this->addFlash('success', $translator->trans('formularios.unidadA.integrantestable.correcto', array(), 'AppBundle'));
return $this->redirectToRoute('unidadA_new');
}
return $this->render('@App/unidadA/new.html.twig', array(
    'unidadA' => $unidadA,
    'form' => $form->createView(), ));
}

```

Grafo resultante:



Complejidad Ciclomática:

$V(G) = \#$ de regiones
 $V(G) = 4$

$V(G) = A - N + 2$
 $V(G) = 14 - 12 + 2$
 $V(G) = 2 + 2$
 $V(G) = 4$

$V(G) = P + 1$
 $V(G) = 3 + 1$
 $V(G) = 4$

La complejidad Ciclomática es igual a 4, lo que significa que existen 4 posibles caminos linealmente independientes y hay que diseñar un mínimo de 4 casos de prueba para el algoritmo. La Tabla 12 muestra los caminos existentes.

Tabla 12 Caminos del grafo de flujo (Fuente: Elaboración propia).

No.	Camino
1	1,2,3,4,5,7,8,10,3,11,12
2	1,2,3,4,5,7,9,10,3,11,12
3	1,2,3,4,6,7,8,10,3,11,12
4	1,2,3,4,6,7,9,10,3,11,12

Los casos de prueba para las pruebas de caja blanca por la técnica de camino básico se ejecutan por cada camino independiente que se determine en un algoritmo específico. A continuación, se muestra el caso de prueba para el camino básico independiente 1 del algoritmo, el resto se encuentra en los anexos.

Proceso:

Adicionar unidad artística

Caso de Prueba:
Adicionar unidad artística
Camino independiente:
1,2,3,4,5,7,8,10,3,11,12
Entradas
Datos de la unidad artística
Resultados esperados:
<ul style="list-style-type: none"> • Adicionar unidad artística. • Adicionar los integrantes de la unidad artística sean graduados y no sean instructor. • Adicionar los datos de los integrantes de las unidades artística correctamente.
Condiciones de ejecución:
<ul style="list-style-type: none"> • Se debe haber adicionado una unidad artística con los integrantes.

Con la realización de los casos de prueba diseñados se probó la ejecución de cada sentencia del código al menos una vez, teniendo en cuenta todas las condiciones lógicas en sus variantes verdaderas y falsas. Los resultados del método de caja blanca fueron satisfactorios.

3.4.3 Pruebas Aceptación

Las pruebas del software, conocidas también como técnicas de evaluación dinámica, son un elemento crítico para la garantía de la calidad del sistema. Representan una revisión final de las especificaciones del diseño y de la implementación. Su principal objetivo es diseñar pruebas que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Pruebas Alfa y Beta

Cuando se construye software a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un periodo de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema.

Pruebas Alfa: La prueba alfa se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El software se usa en un escenario natural con el

desarrollador “mirando sobre el hombro” de los usuarios y registrando los errores y problemas de uso. Las pruebas alfa se realizan en un ambiente controlado.

Pruebas Beta: La prueba beta se realiza en uno o más sitios del usuario final. A diferencia de la prueba alfa, por lo general el desarrollador no está presente. Por tanto, la prueba beta es una aplicación “en vivo” del software en un ambiente que no puede controlar el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante las pruebas beta, es posible hacer modificaciones y luego preparar la liberación del producto de software a toda la base de clientes.

Para la aplicación de las pruebas de aceptación se seleccionaron un grupo de personas han desempeñado durante varios años diversos roles dentro del proceso de gestión del movimiento de artistas aficionados y con cargos dentro del departamento de extensión universitaria. A continuación, se relacionan los involucrados en las pruebas.

Ing. Julio César Espronceda Pérez (Jefe de departamento de actividades extras curriculares)

Lic. Gladys Martínez Iglesia (Especialista Superior de la dirección de extensión universitaria)

Lic. César Cuten Díaz (Profesor de teatro)

Lic. Rosa María Hernández (Profesora de literatura)

Lic. Antonio Gutiérrez Labonit (Director de extensión universitaria)

Ing. Marta Acosta Álvarez (Profesora de literatura)

Lic. Siry Frades Guzmán (Profesora de música)

Lic. Carmen Tamara Echeverría (Profesora de música)

Ing. Julio Jesús García Coste (Profesor de danza)

Ing. José Luis Guisao Jorge (Profesor de danza)

3.5 Resultados de las pruebas aplicadas

Durante la realización de las pruebas se detectaron un conjunto de no conformidades relacionadas con errores de validación y funcionalidad. Los resultados se muestran en la siguiente figura, donde se evidencia la cantidad de casos de prueba ejecutados, los casos de prueba con no conformidades, las cuales fueron corregidas al pasar de una iteración a otra. Se realizaron tres iteraciones, durante la primera iteración se analizaron 22 casos de prueba, de los cuales cuatro resultaron tener un total de 11 no conformidades. En la segunda iteración a través de las pruebas de regresión se verificó que las no conformidades anteriores estuviesen solucionadas sobre los casos de prueba que incidían; de estas

pruebas se obtuvo 5 no conformidad, quedando resuelta en la tercera iteración y cumpliéndose correctamente los requerimientos del sistema.

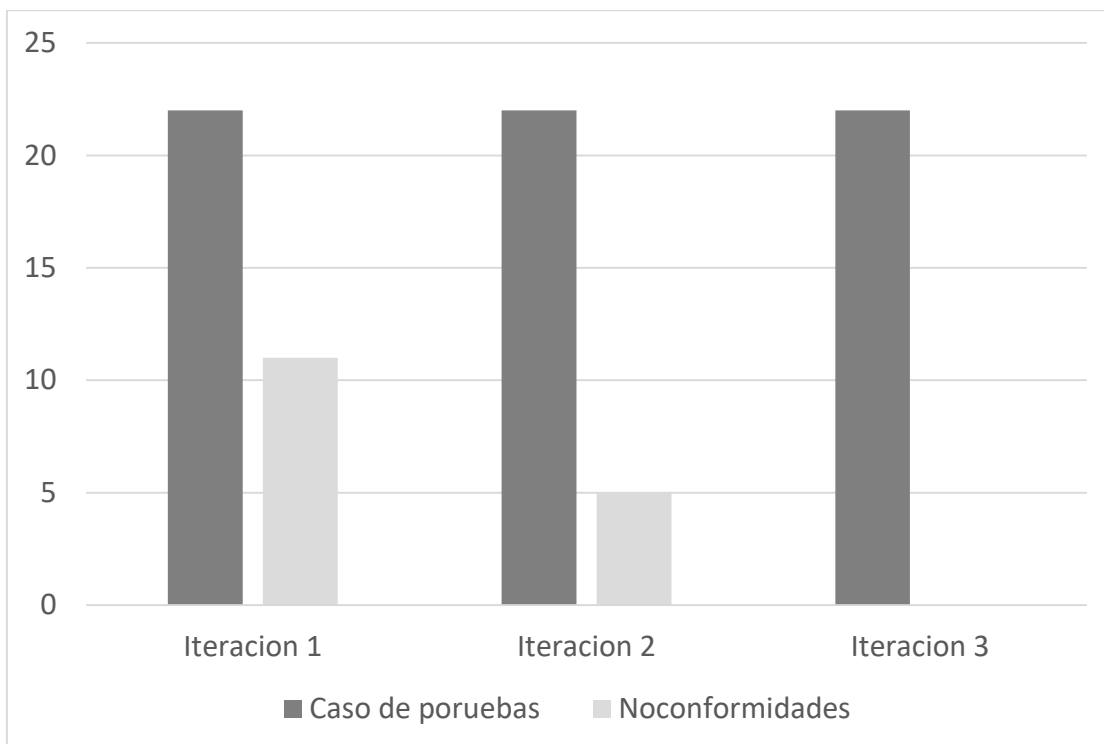


Ilustración 4: Resultados de Pruebas (Elaboración Propia)

3.6 Resultados de la solución implementada

Con el objetivo de evaluar la satisfacción del usuario final se utiliza la técnica de ladov. Esta técnica evalúa el nivel de satisfacción del usuario, de esta forma se conoce si los componentes realizados cumplen las expectativas de los clientes; está conformada por cinco preguntas: 3 cerradas y 2 abiertas («La técnica de ladov», s. f.).

La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, debido a que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las preguntas cerradas (preguntas 1, 2 y 3) que se intercalan dentro de un cuestionario (ver Anexo 4) y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov". («La técnica de ladov», s. f.)

Tabla 13: Cuadro Lógico de ladov Fuente(Elaboración Propia)

	1. ¿Considera usted que es posible participar en el festival de artistas aficionados sin la entrega de la documentación establecida?									
	No			No sé			Sí			
3. ¿Satisface sus necesidades, según el rol que desempeña en la gestión del movimiento de artistas aficionados, el sistema propuesto?	2. ¿Utilizaría usted la aplicación propuesta a la hora de generar la documentación referente al festival de artistas aficionados?									
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No	
	Me satisface mucho.	1	2	6	2	2	6	6	6	6
	No me satisface tanto.	2	2	3	2	3	3	6	3	6
	Me da lo mismo.	3	3	3	3	3	3	3	3	3
	Me disgusta más de lo que me satisface.	6	3	6	3	4	4	3	4	4
	No me satisface nada.	6	6	6	6	4	4	6	4	5
	No sé qué decir.	2	3	6	3	3	3	6	3	4

La escala de satisfacción es la siguiente:

(1). Clara satisfacción, (2). Más satisfecho que insatisfecho, (3). No definida, (4). Más insatisfecho que satisfecho, (5). Clara insatisfacción y (6). Contradictoria. Para medir el grado de satisfacción se tomó una muestra de 10 personas.

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 como se presenta en la siguiente Tabla.

Tabla 14: Índices para el cálculo de satisfacción grupal (Fuente: (49)).

Índice	Escala
+1	Máximo de satisfacción

0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula mediante la siguiente expresión:

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

Donde:

- A, B, C, D, E representan el número de sujetos con índice individual 1, 2, 3 o 6, 4, 5 respectivamente.
- N representa el número total de sujetos del grupo.

Esto permite reconocer las categorías grupales:

- Insatisfacción: desde (-1) hasta (-0,5).
- Contradictorio: desde (-0,49) hasta (+0,49).
- Satisfacción: desde (+0,5) hasta (1).

Luego se aplicó la técnica calculando el índice de satisfacción grupal (ISG) obteniendo como resultado 0.85 aproximadamente, lo que significa una clara satisfacción con el desarrollo del Sistema Informático para la gestión de la información estadística de los festivales de artistas aficionados.

$$ISG = \frac{7(+1) + 3(+0.5) + 0(0) + 0(-0.5) + 0(-1)}{10} = 0.85$$

3.7. Interfaces del Sistemas

Luego de culminado el desarrollo del sistema propuesto es posible observar las pantallas principales del Sistemas para la gestión de la información estadística de los FAA de la UCI. Exponiendo así, el resultado obtenido durante la implementación de las historias de usuario descritas anteriormente. Ver ilustraciones siguientes.

Ilustración 5: Pantallas principal

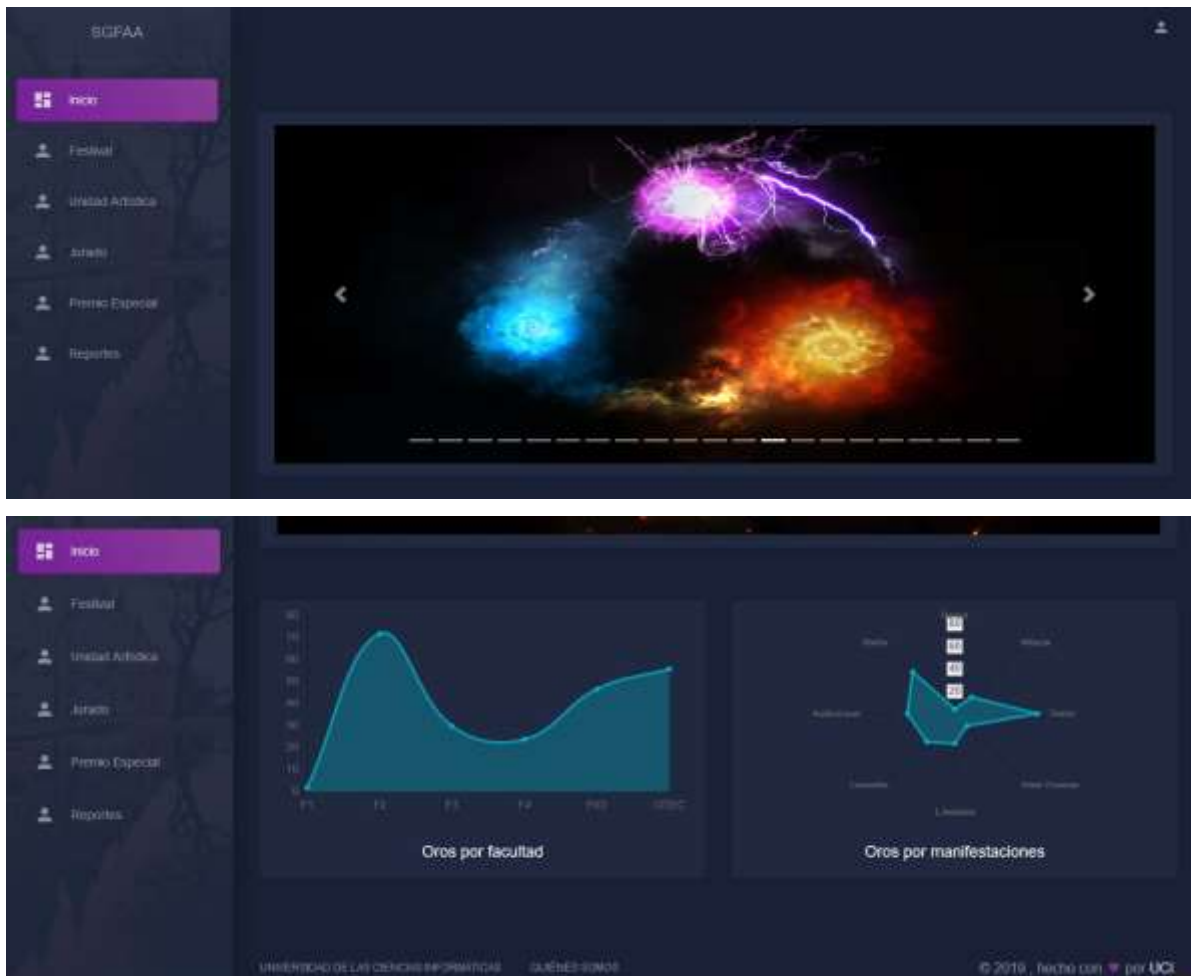


Ilustración 6: Pantalla Adicionar Jurado.

The screenshot shows the SGFAA dashboard with the 'Adicionar jurado' form open. The form has the following fields:

- Nombre ***: A text input field.
- CI ***: A text input field.
- Manifestación ***: A dropdown menu with the selected option being '-Selecciona-'.
- Apellido ***: A text input field.
- Año ***: A text input field with the value '2019' entered.
- Es presidente**: A checkbox.

At the bottom of the dashboard, there is a footer with the text: "LIBERTAD DE LAS DECISIONES INFORMÁTICAS" and "© 2019 - hecho con por UCI".

Ilustración 7: Pantalla Adicionar Festival

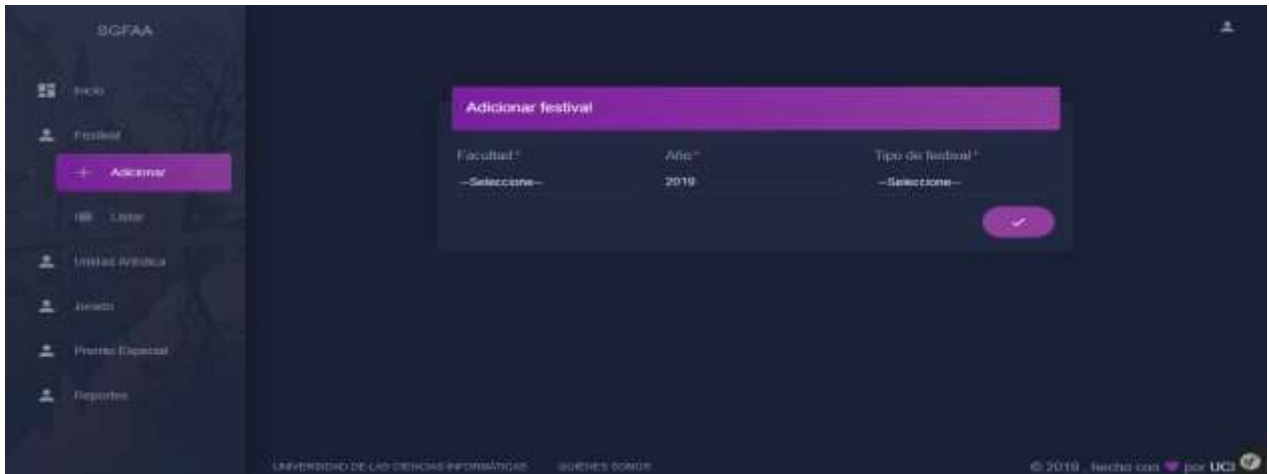


Ilustración 8: Pantalla de la Búsqueda Avanzada

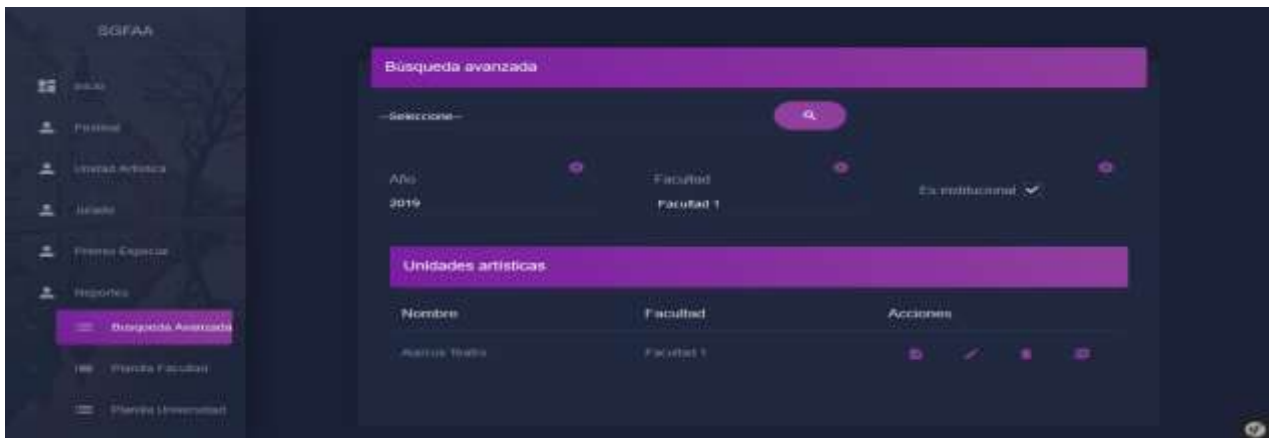
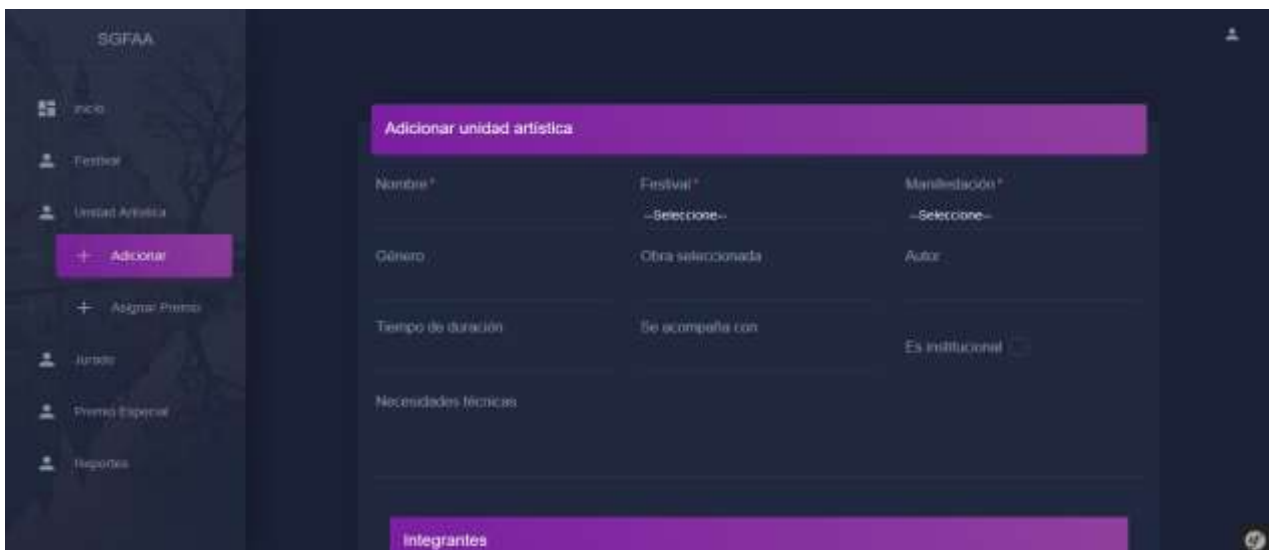


Ilustración 9: Pantalla Adicionar Unidad Artística



Conclusiones del Capítulo

Al finalizar el presente capítulo quedó desarrollada satisfactoriamente la propuesta de solución.

- Logrando así la implementación del Sistema Informático para la gestión de la información estadística de los festivales de artistas aficionados.
- La utilización de los estándares de codificación, permitió la obtención del código de forma organizada, facilitando su comprensión por los desarrolladores que requieran su uso.
- La ejecución de las pruebas de software posibilitó detectar las no conformidades existentes y solucionarlas.
- La aplicación de la técnica de ladov demostró la satisfacción del usuario con el desarrollo del Sistema Informático para la gestión de la información estadística de los festivales de artistas aficionados.

Conclusiones generales

Con el desarrollo de la presente investigación se obtuvo un Sistema informático para la gestión estadística de los Festivales de Artistas Aficionados en la Universidad de las Ciencias Informáticas. La realización de un análisis sobre el desarrollo del sistema informático para la gestión de la información estadística posibilitó la comprensión de sus características y funcionamiento.

- El análisis de sistemas homólogos, permitió identificar las tendencias en el desarrollo de herramientas informáticas para la gestión de información; así como comprender la necesidad de desarrollar uno propio adaptado a las necesidades de la institución.
- Las tecnologías y herramientas seleccionadas permitieron guiar el proceso de desarrollo del sistema informático para la gestión información.
- La ejecución de pruebas de software permitió detectar y corregir errores no identificados durante la implementación, posibilitando el cumplimiento de las especificaciones requeridas.

Recomendaciones

Para dar continuidad a la presente investigación se recomienda, la integración de la solución propuesta con el Sistema de Gestión Universitaria (SGU).

Bibliografía

- Symfony en pocas palabras (Symfony 1.4, la guía definitiva). (s. f.). Recuperado 23 de enero de 2019, de <https://uniwebsidad.com/libros/symfony-1-4/capitulo-1/symfony-en-pocas-palabras?from=librosweb>
- About the Apache HTTP Server Project - The Apache HTTP Server Project. (s. f.). Recuperado 23 de enero de 2019, de http://httpd.apache.org/ABOUT_APACHE.html
- Aja Quiroga, L. (2002). Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. *ACIMED*, 10(5), 7-8. Recuperado de http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1024-94352002000500004&lng=es&nrm=iso&tlng=es
- Ampuero, M. A., & López, Y. (2010). «Creando un profesional con disciplina en el proceso de desarrollo de software». *Revista de Ingeniería Industrial*, Vol. XXVII, No. 1, 27-30. Recuperado de https://www.ingeniare.cl/index.php?option=com_ingeniare&view=va&aid=249&vid=72&lang=es
- Barragan, A., & Alejandro, L. (2016). Lenguaje de modelamiento unificado (UML) para modelamiento de embotelladora. *Scientia Et Technica*, 21(1). Recuperado de <http://www.redalyc.org/resumen.oa?id=84950584006>
- Bernal, Y. P., Iñigo, D. C., & Llanes, G. Á. (2010). *Sistema de gestión académica para la universidad. En: Memorias XIV Congreso de Informática en la Educación*. Editorial Universitaria, 2010. Recuperado de <http://beduniv.reduniv.edu.cu/index.php?page=13&id=122&db=2>
- CAN, C. A. (2012). SPADIES - Sistema para la Prevención de la Deserción de la Educación Superior - Sistemas información. Recuperado 13 de junio de 2019, de https://www.mineduacion.gov.co/sistemasdeinformacion/1735/w3-article-212299.html?_noredirect=1
- Canós, J. H., & Letelier, M. C. P. P. (2012). Metodologías Ágiles en el Desarrollo de Software. Recuperado de <http://10.22.1.21:8080/jspui/handle/123456789/476>
- Capítulo 1. Introducción (Introducción a CSS). (s. f.). Recuperado 23 de enero de 2019, de <https://uniwebsidad.com/libros/css/capitulo-1>
- Chaves, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes*, 6(10).
- Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.
- Concepto de gestión — Definicion.de. (s. f.). Recuperado 23 de enero de 2019, de <https://definicion.de/gestion/>
- Doctrine, The Open-Source PHP ORM and Persistence Tools Project. (s. f.). Recuperado 23 de enero de 2019, de <https://www.doctrine-project.org/index.html>
- Gestión de la información para la mejora del proceso de toma de decisiones. (s. f.). Recuperado 6 de junio de 2019, de <https://www.gestipolis.com/gestion-de-la-informacion-para-la-mejora-del-proceso-de-toma-de-decisiones/>
- Git. (s. f.). Recuperado 23 de enero de 2019, de <https://git-scm.com/>
- Guerrero, C. A., Suárez, J. M., & Gutiérrez, L. E. (2013). Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Información tecnológica*, 24(3), 103-114. <https://doi.org/10.4067/S0718-07642013000300012>
- Guía para el desarrollo de almacenes de datos. (2016, junio). Recuperado 23 de enero de 2019, de

- https://www.researchgate.net/publication/319470221_GUIA_PARA_EL_DESARROLLO_DE_ALMACENES_DE_DATOS_GUIDE_FOR_THE_DEVELOPMENT_OF_DATA_WAREHOUSES
- Hernández, J. L. (1991). La gestión de la información en las organizaciones: Una disciplina emergente. *Revista General de Información y Documentación*, 1(2), 7. <https://doi.org/>
- Home - Twig - The flexible, fast, and secure PHP template engine. (s. f.). Recuperado 23 de enero de 2019, de <https://twig.symfony.com/>
- Importancia del sistema de gestión de la información. (s. f.). Recuperado 6 de junio de 2019, de <https://pyme.lavoztx.com/importancia-del-sistema-de-gestin-de-la-informacin-4665.html>
- La técnica de ladov. (s. f.). Recuperado 29 de abril de 2019, de <https://www.efdeportes.com/efd47/iadov.htm>
- Ledo, L. M. V. (s. f.). Revista Cubana de Informática Médica (RCIM). Recuperado 22 de enero de 2019, de http://www.rcim.sld.cu/revista_9/articulos_htm/alfabetizdigital.htm
- Lenguajes de programación. (s. f.). Recuperado 23 de enero de 2019, de <https://es.ccm.net/contents/304-lenguajes-de-programacion>
- León, A. R. H., & González, S. C. (2012). *El proceso de investigación científica (2 edición)* (2.^a ed.). La Habana: Universitaria. Recuperado de beduniv.reduniv.edu.cu/fetch.php?data=249&type=pdf&id=2862&db=0
- Maza, M. Á. S. (2012). *Javascript*. Innovación Y Cualificación.
- Mendez. (2010). *Itsa metodologias de desarrollo de software (alejandra virrueta mend... Tecnología*. Recuperado de <https://es.slideshare.net/virrueta/itsa-metodologias-de-desarrollo-de-software-alejandra-virrueta-mendez>
- Movimiento de Artistas Aficionados. (2019, junio 6). Recuperado 9 de junio de 2019, de <https://www.mes.gob.cu/es/movimiento-de-artistas-aficionados>
- MVC (Model, View, Controller) explicado. (s. f.). Recuperado 23 de enero de 2019, de <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- NetBeans IDE 8.2 Release Information. (s. f.). Recuperado 23 de enero de 2019, de <https://netbeans.org/community/releases/82/>
- Pacheco Suárez, Y, arlenis, , Jaime Infante, Ramón A, , (2012). *Sistema automatizado para la gestión de la extensión universitari a. Universidad de Pinar del Río*.
- Peñalvo, F. J. G., & Holgado, A. G. (2017). Modelo de dominio. Recuperado de https://Modelo_de_dominio
- pgAdmin - PostgreSQL Tools. (s. f.). Recuperado 23 de enero de 2019, de <https://www.pgadmin.org/>
- PostgreSQL: The world's most advanced open source database. (s. f.). Recuperado 23 de enero de 2019, de <https://www.postgresql.org/>
- Pressman, Roger. (2002). *Ingeniería del Software, Un Enfoque Practico*. (5ta Edicion). (Editorial McGraw-Hill) - Intercambios Virtuales. Recuperado de <https://www.intercambiosvirtuales.org/libros-manuales/ingenieria-del-software-un-enfoque>
- Pressman, RogerS. (2010). *Ingeniería de software . Un Enfoque Práctico* (7ma edición). México.
- Programa Nacional de Extensión Universitaria. (2004, 8).
- ¿Qué es HTML5? | HTML5. (s. f.). Recuperado 23 de enero de 2019, de <https://www.arkaitzgarro.com/html5/capitulo-1.html>
- Que es un potron de diseño. (s. f.).
- ¿Qué son JavaScript y jQuery? (s. f.). Recuperado 23 de enero de 2019, de <https://openclassrooms.com/en/courses/3693206-introduccion-a-jquery/3693213->

- que-son-javascript-y-jquery
- Quiñones Laffita, A., Laffita, A. Q., Torres, I. H., & González, F. C. (2018). Uso de la tecnología en la gestión extensionista, efectividad de la página web extensión universitaria. *Revista Cubana de Ciencias Informáticas*, 12(0), 176-189. Recuperado de [http://rcci.uci.cu/?journal=rcci&page=article&op=view&path\[\]=1785](http://rcci.uci.cu/?journal=rcci&page=article&op=view&path[]=1785)
- Santos, B. de S. (2007). *La Universidad en el siglo XXI Para una reforma democrática y emancipatoria de la universidad* (4ta ed.). Recuperado de www.boaventuradesousasantos.pt/media/universidad_siglo_xxi-.pdf
- SNIES - Sistemas información. (2012). Recuperado 9 de junio de 2019, de <https://www.mineducacion.gov.co/sistemasdeinformacion/1735/w3-propertyname-2672.html>
- Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.
- Thesis | Conceptual Model | Communications Protocols. (s. f.). Recuperado 10 de abril de 2019, de <https://es.scribd.com/document/86038103/Thesis>
- Tipos de requisitos: Funcional vs. No Funcional | Tecnología y Synergix. (s. f.). Recuperado 9 de abril de 2019, de <https://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>
- Torres Lebrato, L. (2015). La gestión de información y la gestión del conocimiento. *Revista Archivo Médico de Camagüey*, 19(2), 96-98. Recuperado de http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1025-02552015000200002&lng=es&nrm=iso&tlng=es
- Universidad de las Ciencias Informáticas. (2019). Recuperado 22 de enero de 2019, de <http://www.uci.cu>
- Vázquez, O. C., & Leyva, R. V. (2015). *Sistema de Gestión de Información para el Vicedecanato de Investigación y Postgrado de la Facultad 2* (Tesis de Grado). Universidad de las Ciencias Informática, La Habana, Cuba.
- Vista de La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. (s. f.). Recuperado 9 de abril de 2019, de <https://revistas.ucr.ac.cr/index.php/intersedes/article/view/790/851>
- Visual Paradigm Product Overview. (2016). Recuperado 23 de enero de 2019, de https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html
- What is Unified Modeling Language (UML)? - Definition from Techopedia. (s. f.). Recuperado 23 de enero de 2019, de <https://www.techopedia.com/definition/3243/unified-modeling-language-uml>

Anexas

Anexo1(Historias de usuario)


Crear Unidad Artística

Historia de usuario	
Número: 6	Nombre del requisito: Crear unidad Artística
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: alta	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1
Descripción: El sistema debe proporcionar que el cliente pueda crear unidad artística, mostrándole un interfaz donde pueda llenar los campos de dicha unidad artística. Debe validar que este se encuentre todos llenos y correctos. Campos (Nombre, Festival, Manifestación).	
Observaciones: Los campos para crear una unidad artística tienen que encontrarse llenos en su totalidad.	
Prototipo de Interfaz Gráfica:	
<p>El prototipo muestra un formulario con el título "Crear Unidad Artística". Contiene tres campos de entrada: "Nombre" (un campo de texto vacío), "Festival" (un menú desplegable con "-Seleccione-" y una flecha hacia abajo) y "Manifestacion" (un menú desplegable con "-Seleccione-" y una flecha hacia abajo). En la esquina inferior derecha del formulario hay un botón con un ícono de una marca de verificación.</p>	

Anexo 1:Historia de Usuario 6.

- **Modificar Unidad Artística.**

Historia de usuario	
Número: 7	Nombre del requisito: Modificar unidad Artística
Usuario: Cliente	Iteración Asignada: 1

Prioridad en negocio: alta	Puntos estimados: 0.8
Nivel de Complejidad: alta	Puntos reales: 0.8
Descripción: El sistema debe proporcionar que el cliente pueda modificar una unidad artística, que se encuentre creada, en caso de que algunos de sus campos estén incorrectos.	
Observaciones: Los campos para modificar una unidad artística tienen que encontrarse llenos en su totalidad.	
Prototipo de Interfaz Gráfica:	
	

Anexo 2: Historia de usuario 7.

- **Listar Unidad Artística.**

Historia de usuario	
Número: 8	Nombre del requisito: Listar Unidad Artística
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: media	Puntos estimados: 0.6
Nivel de Complejidad: media	Puntos reales: 0.6
Descripción: El sistema debe proporcionar que el cliente pueda obtener un listado de las Unidad Artística, que se encuentre creada a través de una interfaz gráfica.	
Observaciones:	
Prototipo de Interfaz Gráfica:	

Listado de Unidades Artísticas		
		<input type="button" value="Buscar"/> <input type="text"/>
Nombre	Festival	Manifestacion
Canto por Cuba	Facultad 1	Musica
Espacio Abierto	Facultad4	Danza
Conga Habanera	Facultad3	Danza


Anexo 3:Historia de usuario 8.

- **Eliminar Unidad Artística.**

Historia de usuario																								
Número: 9	Nombre del requisito: Eliminar Unidad Artística																							
Usuario: Cliente	Iteración Asignada: 1																							
Prioridad en negocio: media	Puntos estimados: 0.6																							
Nivel de Complejidad: media	Puntos reales: 0.6																							
<p>Descripción: El sistema debe proporcionar que el cliente pueda obtener un listado de las unidades artísticas, que se encuentre creada a través de una interfaz gráfica., con la opción que se pueda eliminar alguna unidad artística.</p>																								
<p>Observaciones: El sistema dará la opción de eliminar una unidad artística si el usuario registrado tiene rol de administrados</p>																								
<p>Prototipo de Interfaz Gráfica:</p>																								
<table border="1"> <thead> <tr> <th colspan="4">Eliminar Unidad Artística</th> </tr> </thead> <tbody> <tr> <td colspan="2"></td> <td style="text-align: right;"> <input type="button" value="Buscar"/> <input type="button" value="Facultad 2"/> </td> </tr> <tr> <th>Nombre</th> <th>Festival</th> <th>Manifestacion</th> <th>Eliminar</th> </tr> <tr> <td>Canto por Cuba</td> <td>Facultad 2</td> <td>Musica</td> <td></td> </tr> <tr> <td>Espacio Abierto</td> <td>Facultad 2</td> <td>Danza</td> <td></td> </tr> <tr> <td>Conga Habanera</td> <td>Facultad 2</td> <td>Danza</td> <td></td> </tr> </tbody> </table>		Eliminar Unidad Artística						<input type="button" value="Buscar"/> <input type="button" value="Facultad 2"/>	Nombre	Festival	Manifestacion	Eliminar	Canto por Cuba	Facultad 2	Musica		Espacio Abierto	Facultad 2	Danza		Conga Habanera	Facultad 2	Danza	
Eliminar Unidad Artística																								
		<input type="button" value="Buscar"/> <input type="button" value="Facultad 2"/>																						
Nombre	Festival	Manifestacion	Eliminar																					
Canto por Cuba	Facultad 2	Musica																						
Espacio Abierto	Facultad 2	Danza																						
Conga Habanera	Facultad 2	Danza																						

Anexo 4:Historia de usuario 9.

- **Crear Jurado**

Historia de usuario	
Número: 12	Nombre del requisito: Crear Jurado
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: alta	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1
<p>Descripción: El sistema debe proporcionar que el cliente pueda crear un jurado, mostrándole un interfaz donde pueda llenar los campos de dicho jurado. Debe validar que este se encuentre todos llenos y correctos. Campos (nombre, manifestación, festival).</p>	
<p>Observaciones: Los campos para crear un jurado tienen que encontrarse llenos en su totalidad</p>	
<p>Prototipo de Interfaz Gráfica:</p> 	

Anexo 5: Historia de usuario 10.

- **Modificar Jurado.**

Historia de usuario	
Número: 13	Nombre del requisito: Modificar Festival
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: medio	Puntos estimados: 0.8
Nivel de Complejidad: alta	Puntos reales: 0.8
<p>Descripción: El sistema debe proporcionar que el cliente pueda modificar un jurado, que se encuentre creado, en caso de que algunos de sus campos estén incorrecto.</p>	

Observaciones: Los capos para modificar un jurado tienen que encontrarse llenos en su totalidad

Prototipo de Interfaz Gráfica:

Modificar Jurado

Nombre	Manifestacion	Festival	Modificar
Juana	Danza	Facultad 4	
Carlos	Musica	Facultad 3	
Julio	Teatro	Facultad 1	

Anexo 6: Historia de usuario 11.

- **Listar Jurado**

Historia de usuario	
Número: 14	Nombre del requisito: Listar Jurado
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: media	Puntos estimados: 0.6
Nivel de Complejidad: media	Puntos reales: 0.6
<p>Descripción: El sistema debe proporcionar que el cliente pueda obtener un listado del jurado, que se encuentre creado a través de una interfaz gráfica.</p>	
<p>Observaciones:</p>	

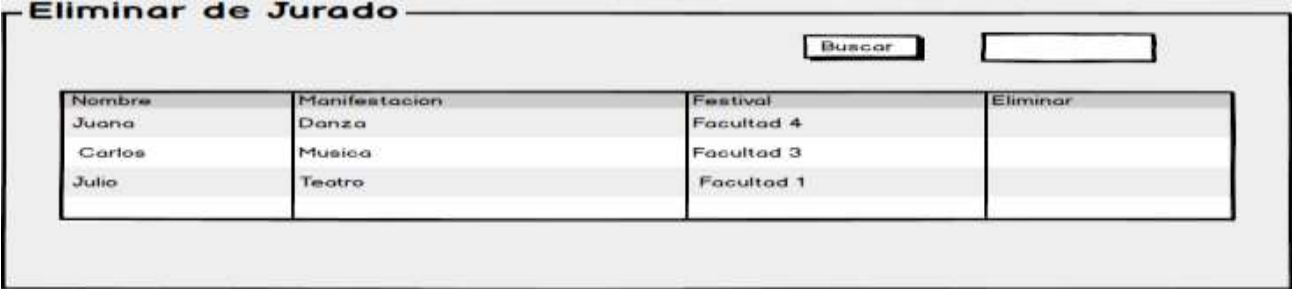
Prototipo de Interfaz Gráfica:

Listado de Jurado

Nombre	Manifestacion	Festival
Juana	Danza	Facultad 4
Carlos	Musica	Facultad 3
Julio	Teatro	Facultad 1

Anexo 7: historia de usuario 12.

- **Eliminar Jurado.**

Historia de usuario	
Número: 15	Nombre del requisito: Eliminar Jurado
Usuario: Cliente	Iteración Asignada: 1
Prioridad en negocio: media	Puntos estimados: 0.6
Nivel de Complejidad: media	Puntos reales: 0.6
<p>Descripción: El sistema debe proporcionar que el cliente pueda obtener un listado del jurado, que se encuentre creado a través de una interfaz gráfica., con la opción que se pueda eliminar algún jurado</p>	
<p>Observaciones: El sistema dará la opción de eliminar un jurado si el usuario registrado tiene rol de administrados</p>	
<p>Prototipo de Interfaz Gráfica:</p> 	

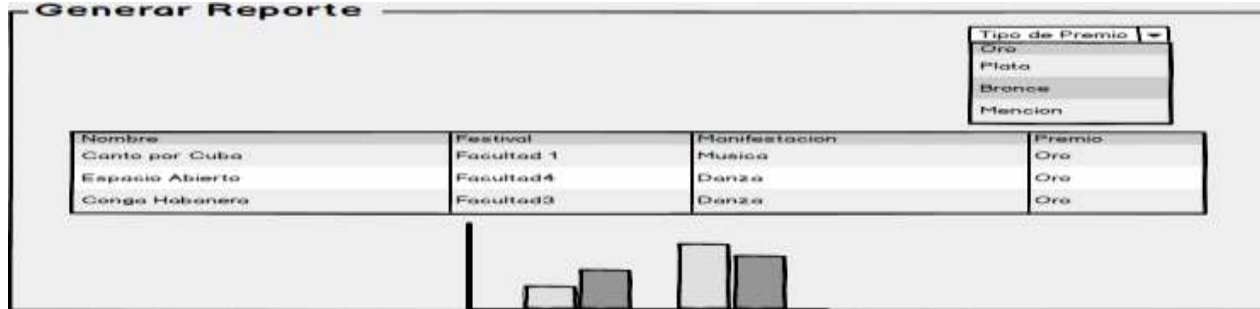
Anexo 8: Historia de usuario 13.

- **Generar Reporte.**

Historia de usuario	
Número: 10	Nombre del requisito: Generar Reporte
Usuario: Cliente	Iteración Asignada: 2
Prioridad en negocio: medio	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1

Descripción: El sistema debe proporcionar que el cliente pueda generar reporte referente a los festivales , unidades artísticas , premios , premios especiales, existente en el sistema.

Prototipo de Interfaz Gráfica:



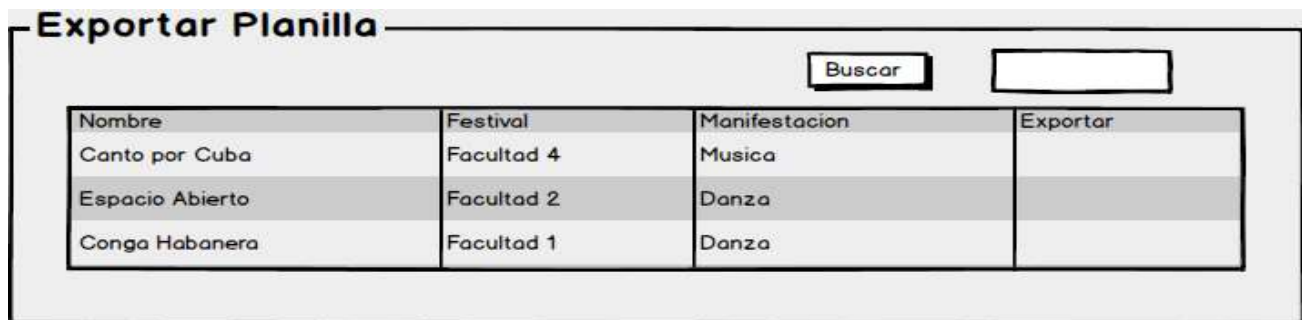
Anexo 9: Historia de usuario 14 .

- **Exportar Planilla.**

Historia de usuario	
Número: 11	Nombre del requisito: Exportar Planilla
Usuario: Cliente	Iteración Asignada: 2
Prioridad en negocio: alta	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1


Descripción: El sistema debe proporcionar que el cliente pueda exportar planillas referente a los unidades artísticas existente en el sistema.

Prototipo de Interfaz Gráfica:



Anexo 10: Historia de usuario 15 .

- **Crear Premio Especial.**

Historia de usuario	
Número: 16	Nombre del requisito: Crear Premio Especial
Usuario: Cliente	Iteración Asignada: 2
Prioridad en negocio: alta	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1
Descripción: El sistema debe proporcionar que el cliente pueda crear premios especiales referente a los unidades artísticas existente en el sistema	
Observación: El sistema debe validar que al crear un premio especial se encuentren los campos llenos en su totalidad	
Prototipo de Interfaz Gráfica:	
	

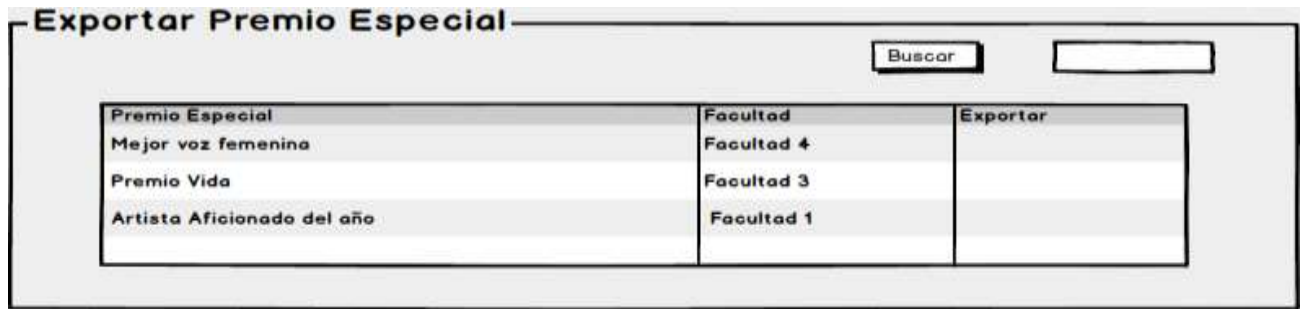
Anexo 11: Historia de usuario 16.

- **Exportar Listado de Premio Especial.**

Historia de usuario	
Número: 17	Nombre del requisito: Exportar Listado de Premio Especial.
Usuario: Cliente	Iteración Asignada: 2
Prioridad en negocio: media	Puntos estimados: 1
Nivel de Complejidad: alta	Puntos reales: 1
Descripción: El sistema debe proporcionar que el cliente pueda exportar listado de premios especiales referente	

a los unidades artísticas existente en el sistema.

Prototipo de Interfaz Gráfica:



Anexo 12: Historia de usuario 17.

- **Exportar Listado del Jurado.**

Historia de usuario																	
Número: 18	Nombre del requisito: Exportar Listado del Jurado																
Usuario: Cliente	Iteración Asignada: 2																
Prioridad en negocio: media	Puntos estimados: 1																
Nivel de Complejidad: alta	Puntos reales: 1																
<p>Descripción:El sistema debe proporcionar que el cliente pueda exportar listado del jurado referente a los unidades artísticas existente en el sistema.</p>																	
<p>Prototipo de Interfaz Gráfica:</p> <table border="1"> <thead> <tr> <th>Nombre</th> <th>Manifestacion</th> <th>Festival</th> <th>Exportar</th> </tr> </thead> <tbody> <tr> <td>Juana</td> <td>Danza</td> <td>Facultad 4</td> <td></td> </tr> <tr> <td>Carlos</td> <td>Musica</td> <td>Facultad 3</td> <td></td> </tr> <tr> <td>Julio</td> <td>Teatro</td> <td>Facultad 1</td> <td></td> </tr> </tbody> </table>		Nombre	Manifestacion	Festival	Exportar	Juana	Danza	Facultad 4		Carlos	Musica	Facultad 3		Julio	Teatro	Facultad 1	
Nombre	Manifestacion	Festival	Exportar														
Juana	Danza	Facultad 4															
Carlos	Musica	Facultad 3															
Julio	Teatro	Facultad 1															

Anexo 13: Historia de usuario 18 .

Anexo 2 (Caso de pruebas (DCF))

Gestionar festival

Escenario	Descripción	Variables			Respuesta esperada	Respuesta
		Facultad	Año	Tipo de festival		
EC 1.1. Crear festival satisfactoriamente	El sistema debe proporcionar que el usuario pueda crear un festival, mostrándole un interfaz donde pueda llenar los campos de dicho festival. Debe validar que este se encuentre todos llenos y correctos. campos (Facultad, Año, Tipo de festival).	V Facultad 1	V 2019	V Universidad	El sistema debe mostrarle los campos al usuario para crear un festival y una vez creado mostrarle un mensaje que ha sido creado satisfactoriamente o no un festival determinado	El sistema muestra los campos para crear un festival y una vez creado muestra un mensaje notificando al usuario que el festival ha sido creado satisfactoriamente
EC 1.2. Modificar festival satisfactoriamente	El sistema debe proporcionar que el usuario pueda modificar el festival anteriormente creado mostrándole un listado con los festivales existente donde pueda elegir el deseado, el listado debe tener habilitado el botón de modificar(editar)	Facultad 1	2019	Universidad	El sistema debe mostrarle al usuario el listado con los festivales existente, permitiéndole modificar los parámetros del festival seleccionado	El sistema muestra al usuario el listado con los festivales existente, permitiendo modificar los parámetros del festival seleccionado
Listar	El sistema debe	Facultad 1	2019	Universidad	El sistema debe	El sistema

festival	proporcionar que el usuario pueda acceder a la funcionalidad listar festival mostrando un listado con los festivales creados	Facultad 5 ...	2018....	ad Universid ad.....	mostrarle al usuario una vez seleccionada la funcionalidad listar festival , el listado de los festivales existente	muestra al usuario una vez seleccionada la funcionalidad listar festival , el listado de los festivales existente
Eliminar Festival	El sistema debe proporcionar que el usuario pueda acceder a la funcionalidad listar festival mostrando un listado con los festivales creados, y teniendo disponible el botón eliminar	Facultad1	2019	Universid ad	El sistema debe mostrarle al usuario una vez seleccionada la funcionalidad listar festival , el listado de los festivales existen y permitirle eliminar el festival seleccionado	El sistema muestra al usuario una vez seleccionada la funcionalidad listar festival , el listado de los festivales existen y permitirle eliminar el festival seleccionado
No.	Variable	Valor Nulo		Descripción		
1	Facultad	No		Es un campo que permite al usuario insertar la facultad a la que corresponde el festival correctamente. Acepta solo letras y no acepta espacio entre letras.		
2	Año	No		Es un campo que permite al usuario insertar el año correspondiente al festival. Acepta caracteres numérico.		
3	Tipo de Festival	No		Es un campo que permite al usuario seleccionar el tipo de festival que desea crear.		

Anexo 14: Gestionar Festival.

Gestionar Unidad Artística

Escenario	Descripción	Variables			Respuesta esperada	Respuesta
		Nombre	Festival	Manifestación		
EC 1.1. Crear Unidad Artística	El sistema debe proporcionar que el usuario pueda crear una Unidad Artística, mostrándole un interfaz donde pueda llenar los campos de dicha Unidad. Debe validar que este se encuentre llenos y correctos. campos (Nombre , Festival, Manifestación).	V Creo en ti	V Facultad1 _2019	V Teatro	El sistema debe mostrarle los campos al usuario para crear una Unidad Artística y una vez creado mostrarle un mensaje que ha sido creado satisfactoria mente o no	El sistema muestra los campos para crear una unidad artística y una vez creada muestra un mensaje notificando al usuario que la unidad artística ha sido creada satisfactoriam ente
EC 1.2. Modificar Unidad Artística	El sistema debe proporcionar que el usuario pueda modificar la unidad artística anteriormente creada mostrándole un listado con las existente donde pueda elegir el deseado, el listado debe tener habilitado el botón de modificar(editar)	Palcos	Facultad 2_2019	Literatura	El sistema debe mostrarle al usurario el listado con las unidades artísticas existente, permitiéndole modificar los parámetros de la misma	El sistema muestra al usurario el listado con las unidades artísticas existente, permitiendo modificar los parámetros que el usuario desee

Listar Unidad Artística	El sistema debe proporcionar que el usuario pueda acceder a la funcionalidad Búsqueda avanzada mostrando un listado con las unidades artísticas creadas mediante un filtro seleccionado	Palcos ...	Facultad 1_2019.	Literatura	El sistema debe mostrarle al usuario una vez seleccionada la funcionalidad de búsqueda avanzada , el listado de las unidades artísticas deseada	El sistema muestra al usuario una vez seleccionada la funcionalidad de búsqueda avanzada , el listado de las unidades artísticas deseada
Eliminar Festival	El sistema debe proporcionar que el usuario pueda acceder a la funcionalidad de Búsqueda avanzada mostrando un listado con las unidades artísticas creados , y teniendo disponible l botón eliminar	Facultad1	2019	Universid ad	El sistema debe mostrarle al usuario una vez seleccionada la funcionalidad de búsqueda avanzada val , el listado de las unidades artísticas existen y permitirle eliminar la unidad artística seleccionada	El sistema muestra al usuario una vez seleccionada la funcionalidad de búsqueda avanzada val , el listado de las unidades artísticas existen y permitirle eliminar la unidad artística seleccionada
No.	Variable	Valor Nulo			Descripción	
1	Nombre	No			Es un campo que permite al usuario insertar el nombre de la	

				unidad artística correctamente. Acepta solo letras y no acepta espacio entre letras.
2	Festival	No		Es un campo que permite al usuario seleccionar el festival al que corresponde la unidad artística. Acepta caracteres numérico.
3	Manifestación	No		Es un campo que permite al usuario seleccionar el tipo de manifestación correspondiente a la unidad artística que se esté creando.

Tabla 15: Gestionar Unidad Artística

Anexo3(Caso de pruebas (DCU))

Proceso:
Adicionar unidad artística
Caso de Prueba:
Adicionar unidad artística
Camino independiente:
1,2,3,4,5,7,9,10,3,11,12
Entradas
Datos de la unidad artística
Resultados esperados:
<ul style="list-style-type: none"> • Adicionar unidad artística • Adicionar los integrantes de la unidad artística sean no se graduados y no sean instructor. • Adicionar los datos de los integrantes de las unidades artística correctamente
Condiciones de ejecución:
<ul style="list-style-type: none"> • Se debe haber adicionado integrantes de la unidad artística

Anexo 15:Caso de prueba unitaria 2

Caso de prueba unitaria 3

Proceso:
Adicionar unidad artística

Caso de Prueba:
Adicionar unidad artística
Camino independiente:
1,2,3,4,6,7,8,10,3,11,12
Entradas
Datos de la unidad artística
Resultados esperados:
<ul style="list-style-type: none"> • Adicionar unidad artística • Adicionar los integrantes de la unidad artística sean no se graduados e instructor. • Adicionar los datos de los integrantes de las unidades artística correctamente
Condiciones de ejecución:
<ul style="list-style-type: none"> • Se debe haber adicionado integrantes de la unidad artística

Anexo 16:Caso de prueba unitaria 3

Caso de prueba unitaria 4

Proceso:
Adicionar unidad artística
Caso de Prueba:
Adicionar unidad artística
Camino independiente:
1,2,3,4,6,7,9,10,3,11,12
Entradas
Datos de la unidad artística
Resultados esperados:
<ul style="list-style-type: none"> • Adicionar unidad artística • Adicionar los integrantes de la unidad artística sean graduados e instructor. • Adicionar los datos de los integrantes de las unidades artística correctamente
Condiciones de ejecución:
<ul style="list-style-type: none"> • Se debe haber adicionado integrantes de la unidad artística

Anexo 4. Modelo de encuesta aplicada a 10 especialistas del Centro cultural, para medir el nivel de satisfacción de la solución.

Estimado especialista: Lee con cuidado cada pregunta antes de responder. Te agradecemos tu participación y franqueza al decirnos lo que piensas del sistema para la gestión de la información estadística de los festivales de artistas aficionados.

1. ¿Considera usted que es posible participar en el festival de artistas aficionados sin la entrega de la documentación establecida?

No No sé Sí

2. ¿Utilizaría usted la aplicación propuesta a la hora de generar la documentación referente al festival de artistas aficionados?

No No sé Sí

3. ¿Satisface sus necesidades, según el rol que desempeña en la gestión del movimiento de artistas aficionados, el sistema propuesto?

Me satisface mucho.
 No me satisface tanto.
 Me da lo mismo.
 Me disgusta más de lo que me satisface.
 No me satisface nada.
 No sé qué decir.

4. ¿Modificaría usted algún elemento del sistema de gestión de la información estadística del festival? Argumente.

5. ¿Considera que se debería extender el uso del sistema propuesto al resto de la Universidad? Argumente.