



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 4

**MÓDULO DE INSTITUCIONES CULTURALES PARA EL SISTEMA GERENCIAL DE
LA EXTENSIÓN UNIVERSITARIA (SGEU).**

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

Autores: Arthur Javier Gargallo Fleitas

José Adrián Torres Peraza

Tutores: MSc. Julio César Espronceda Pérez

Ing. Alvaro Alejandro Acosta Ruiz

La Habana, 2023

*“Los sueños nos salvan, los sueños nos elevan y nos transforman en algo
mejor”.*
Joe Kelly

Arthur Javier Gargallo Fleitas

A mi madre y a mi padre, por en 2017 saber perdonar mis errores y mis muestras de inmadurez y por decidir darme una oportunidad para que hoy estuviese aquí. A mis abuelos por ser también excelentes seres humanos y saber cómo educarme. A Annie Mariay por ser mi compañera, sostén emocional y familia cuando más lo he necesitado. A Roly por siempre ver mi potencial y saber lo que doy. A Airely y a José Adrián por estar conmigo desde el principio de esta contienda y nunca dejarme tirado. A Lieter, por ofrecermme una amistad incondicional y verdadera y ver la persona que soy y no la que la gente cree. A mi tía Antonia, por siempre tratarme como un hijo propio. Al resto de mis familiares y amigos, de los que aún puedo disfrutar su compañía y los que ya no están.

José Adrián Torres Peraza

A mi madre por siempre apoyarme y motivarme en los momentos más difíciles, por demostrarme que yo sí puedo sin ti esto no sería posible. A mi padrastro por estar conmigo desde que tengo uso de razón, aconsejarme y guiarme siempre por el buen camino. A mis abuelos por todos los consejos y la manera en que me criaron por la cual siempre estaré muy agradecido. A mi hermana por siempre ser mi apoyo emocional cuando las cosas no están del todo bien. A mi compañero de tesis y amigo Arthur por darme la oportunidad de trabajar con él y de cargar conmigo todos estos años. A mis primos Jorge Luis y Daylet y mis padrinos por todas las cosas que han hecho por mí. A la segunda familia que viene conmigo desde el primer día, Airely, Jair, Leandro.

Agradecimientos

Arthur Javier Gargallo Fleitas

Quiero agradecerle en un primer momento a mis tutores Álvaro y Julio por su confianza y por decidir otorgarme esta oportunidad en un momento de necesidad extrema para mí y así poder graduarme como profesional de las Ciencias Informáticas. Quiero agradecerle a mis compañeros de estudio durante estos cinco años, a los que están y a los que desafortunadamente por motivos de fuerza mayor no pudieron continuar conmigo en este trayecto. De todos ellos, me llevo las mejores experiencias y los mejores momentos. Tengo que agradecerle en especial a Airely, Jose, Jair, Leandro y Randy quiénes desde el primer día me ayudaron a creecerme como persona y a madurar mediante el compañerismo y el poder curativo de la risa. Sin duda le agradezco a Annie Mariay por su infinito amor, por soportarme estos dos años y especialmente por decidir legarme esta tesis. Le agradezco a toda mi familia, a mi madre, sobre todo, por su inmensa dedicación y sacrificio por mí durante tanto tiempo; le agradezco a mi padre por su apoyo incondicional y por siempre responder con un rotundo sí cuando lo he necesitado, le agradezco a mis abuelos, a mi abuela por su amor y dedicación de madre y a mi abuelo por forjar en mí los valores que hoy tengo. Le agradezco a mi padrastro por la fe depositada en mí y por tanto apoyo en todos los aspectos. Le agradezco a mi tía Antonia, por todo el inmenso apoyo que ha mostrado desde que nací. Finalmente agradezco a todas las demás personas que de una forma u otra me ayudaron con esta ardua contienda estos cinco años.

José Adrián Torres Peraza

Quiero agradecer a toda mi familia en general a mis compañeros que me

apoyaron en momentos muy difíciles en los cuales cometí una serie de errores, a mis tutores Álvaro y Julio por su confianza y por decidir otorgarme la oportunidad en un momento de necesidad extrema para mí y mi compañero de tesis, gracias por el apoyo y la confianza en nosotros. A todos los compañeros que por un motivo u otro no pudieron continuar la carrera, gracias Aponte por todas las cosas que me enseñaste de la vida, gracias Arthur y Airely por ser mis padres en la universidad y terminar de criarme, gracias Arianna por ser mi guardaespaldas. Gracias a Darlenys, Maylin, Alejandro, Randy, Lekyam, Elic, Jair, Leandro por todos los momentos compartidos. Gracias a mi familia aquí en la Habana mis abuelos Jorge Y Ana por todos los consejos, a mis tíos Isa y Jorgito, a mi papá. Finalmente agradecer a todas las personas que durante estos cinco largos años me ayudaron para que esto hoy fuese posible.

Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 8 días del mes de diciembre del año 2023.



Arthur Javier Gargallo Fleitas

Autor



MSc. Julio César Espronceda Pérez

Tutor



José Adrián Torres Peraza

Autor



Ing. Alvaro Alejandro Acosta Ruiz

Tutor

En sincronía con los avances de la ciencia y la técnica a nivel mundial, Cuba ha trasladado el concepto de la informatización de la sociedad a nuevas dimensiones los años recientes. Como vanguardia en este aspecto tenemos a la Universidad de las Ciencias Informáticas (UCI). La UCI en alianza con el Ministerio de Educación Superior y la Universidad de la Habana ha concretado la idea de crear un sistema de gestión para el control de los subprocesos de extensión universitaria de las instituciones de la Educación Superior en Cuba. La extensión universitaria tiene como objetivo promover el desarrollo cultural y la transferencia de conocimiento y la cultura entre los distintos sectores de la comunidad universitaria. La extensión universitaria cuenta con siete subprocesos clave, entre ellos las instituciones culturales. Dicho subproceso posee diferentes problemas que inciden en la realización de las actividades comprendidas dentro de la gestión del mismo. La presente investigación tiene como principal objetivo desarrollar un módulo que contribuya a la gestión de información de las instituciones culturales. Para el desarrollo de la propuesta se realizó un estudio de sistemas homólogos. Se seleccionó AUP en su variante UCI en el escenario cuatro como metodología de desarrollo. Se utilizó Python como lenguaje de programación y Django como framework. Se realizó un análisis del negocio y se modelaron todos los procesos que realiza para obtener los requisitos. La aplicación permite gestionar la información de las instituciones culturales, garantizando el control de la información y el acceso a la misma. Para garantizar la calidad de la solución se realizó una estrategia de pruebas. Para comprobar el nivel de satisfacción de los usuarios se realizó la técnica de Iadov, arrojando resultados positivos.

Palabras clave: actividades, institución, extensión, gestión.

Introducción	1
1 Marco teórico referencial sobre el módulo Instituciones Culturales para el Sistema Gerencial de Extensión Universitaria (SGEU)	6
1.1 Conceptos asociados al tema	6
1.1.1 Extensión Universitaria	6
1.1.2 Gestión de la extensión universitaria	7
1.1.3 Actividades extensionistas	8
1.1.4 Gestión de información	9
1.1.5 Sistema de gestión de información	10
1.1.6 Instituciones Culturales	10
1.2 Estudio de sistemas homólogos	10
1.2.1 Ámbito internacional	10
1.2.2 Ámbito nacional	11
1.2.3 Consideraciones del estudio de los sistemas	12
1.3 Análisis de las metodologías, herramientas y tecnologías para el desarrollo del sistema . . .	13
1.3.1 Metodología	13
1.3.2 Herramientas y tecnologías	13
1.4 Consideraciones parciales	19
2 Características y diseño del módulo de Instituciones Culturales para el Sistema Gerencial de Extensión Universitaria	20
2.1 Descripción de la propuesta de solución	20
2.1.1 Actividades Culturales	20
2.1.2 Gestión de Comentarios	21
2.1.3 Gestión de asientos	21
2.1.4 Gestión de locales	21
2.1.5 Gestión de patrimonio	21
2.2 Especificación de requisitos del sistema	21
2.2.1 Requisitos funcionales	22

2.2.2	Requisitos no funcionales	23
2.3	Descripción de las historias de usuario	24
2.4	Estilo arquitectónico	27
2.4.1	Patrón arquitectónico	27
2.5	Diagrama de clases de diseño	28
2.6	Patrones de diseño	30
2.7	Modelo de datos	31
2.8	Consideraciones parciales	33
3	Implementación y pruebas del módulo de Instituciones Culturales para el Sistema Gerencial de Extensión Universitaria	35
3.1	Estándares de codificación	35
3.2	Diagrama de despliegue	38
3.3	Pruebas de software	39
3.3.1	Estrategia de pruebas	39
3.3.2	Pruebas unitarias	40
3.3.3	Pruebas de Integración	44
3.3.4	Pruebas funcionales	46
3.3.5	Prueba de seguridad	49
3.3.6	Prueba de rendimiento	50
3.3.7	Pruebas de aceptación	51
3.4	Satisfacción de los usuarios	52
3.5	Consideraciones parciales	54
	Conclusiones	56
	Recomendaciones	57
	Acrónimos	58
	Referencias bibliográficas	60
	Apéndices	64
A	Historias de Usuario	65
A.1	Historias de Usuario	65
B	Empleo de los métodos científicos	93
B.1	Modelo de encuesta aplicada a especialistas	93
B.1.1	Resultados de la encuesta aplicada a especialistas	93

B.2	Guía de observación para el análisis de las soluciones existentes	94
-----	---	----

Índice de figuras

1.1	Mapa de procesos de Extensión Universitaria	8
2.1	Representación de la arquitectura. (Fuente: Elaboración propia)	28
2.2	Diagrama de clases de gestión de Actividades	29
2.3	Modelo de datos	32
3.1	Representación de la arquitectura. (Fuente: Elaboración propia)	38
3.2	Representación de la arquitectura. (Fuente: Elaboración propia)	39
3.3	Prueba de seguridad. (Fuente: Elaboración propia)	44
3.4	Representación de la integración del directorio del Módulo en cuestión al proyecto. (Fuente: Elaboración propia)	45
3.5	Integración Descendente. (Fuente: Elaboración propia)	46
3.6	Representación del resultado de las pruebas. (Fuente: Elaboración propia)	47
3.7	Representación del resultado de las pruebas. (Fuente: Elaboración propia)	49
3.8	Prueba de seguridad. (Fuente: Elaboración propia)	50
3.9	Prueba de seguridad. (Fuente: Elaboración propia)	50
3.10	Prueba de seguridad. (Fuente: Elaboración propia)	50
3.11	Prueba de rendimiento. (Fuente: Elaboración propia)	51
3.12	Prueba de rendimiento. (Fuente: Elaboración propia)	51

Índice de tablas

2.1	Descripción de los Requisitos Funcionales. (Fuente: Elaboración propia).	22
2.2	Historia de usuario # 1	24
2.3	Historia de usuario # 2	25
2.4	Historia de usuario # 3	26
3.1	Estándares de codificación a utilizar en la implementación del sistema. (Fuente: Elaboración propia).	35
3.2	Estrategia de pruebas. (Fuente: Elaboración propia).	40
3.3	Cálculo de la complejidad ciclomática del método post de la clase aprobar solicitud	41
3.4	Caminos del grafo de flujo (Fuente: Elaboración propia).	43
3.5	Caso de Prueba para el camino básico 1 (Fuente: Elaboración propia).	43
3.6	Caso de prueba “Añadir participante” (Fuente: Elaboración Propia).	47
3.7	Variables de caso de prueba “Añadir participante”(Fuente: Elaboración Propia).	48
3.8	Cuadro Lógico de Iadov (Fuente: Elaboración propia).	53
A.1	Historia de usuario # 4	65
A.2	Historia de usuario # 5	66
A.3	Historia de usuario # 6	66
A.4	Historia de usuario # 7	67
A.5	Historia de usuario # 8	68
A.6	Historia de usuario # 9	69
A.7	Historia de usuario # 10	70
A.8	Historia de usuario # 11	70
A.9	Historia de usuario # 12	71
A.10	Historia de usuario # 13	71
A.11	Historia de usuario # 14	72
A.12	Historia de usuario # 15	73
A.13	Historia de usuario # 16	73
A.14	Historia de usuario # 17	74
A.15	Historia de usuario # 18	74
A.16	Historia de usuario # 19	75

A.17 Historia de usuario # 20	75
A.18 Historia de usuario # 21	76
A.19 Historia de usuario # 22	76
A.20 Historia de usuario # 23	77
A.21 Historia de usuario # 24	77
A.22 Historia de usuario # 25	78
A.23 Historia de usuario # 26	78
A.24 Historia de usuario # 27	79
A.25 Historia de usuario # 28	79
A.26 Historia de usuario # 29	79
A.27 Historia de usuario # 30	80
A.28 Historia de usuario # 31	80
A.29 Historia de usuario # 32	81
A.30 Historia de usuario # 33	81
A.31 Historia de usuario # 34	82
A.32 Historia de usuario # 35	82
A.33 Historia de usuario # 36	83
A.34 Historia de usuario # 37	83
A.35 Historia de usuario # 38	84
A.36 Historia de usuario # 39	85
A.37 Historia de usuario # 40	86
A.38 Historia de usuario # 41	86
A.39 Historia de usuario # 42	86
A.40 Historia de usuario # 43	87
A.41 Historia de usuario # 44	87
A.42 Historia de usuario # 45	88
A.43 Historia de usuario # 46	89
A.44 Historia de usuario # 47	89
A.45 Historia de usuario # 48	90
A.46 Historia de usuario # 49	90
A.47 Historia de usuario # 50	91
A.48 Historia de usuario # 51	91
B.1 Encuesta para aplicar Iadov	93
B.2 Guía de observación para estudio de soluciones (Fuente: Elaboración propia).	94

En las últimas décadas, las Tecnologías de la Información y las Comunicaciones (TICs) han transformado profundamente diversos aspectos de la sociedad a nivel mundial, desde las formas de comunicación hasta las formas de trabajar, aprender, consumir información y la interrelación del propio mundo en sí. Estas tecnologías engloban una amplia gama de herramientas, como computadoras, internet, dispositivos móviles, software y sistemas de información (Padín, 2023).

Cuba, como país, ha decidido desde hace unos años unirse al uso de las TICs de manera diversificada en lo que se ha denominado “Proceso de Informatización de la Sociedad Cubana” (Bodaño García, 2022). El gobierno y las instituciones cubanas han estado trabajando en los últimos años para promover el desarrollo y la aplicación de las TICs en diferentes ámbitos de la sociedad. El objetivo principal del proceso de informatización en Cuba es facilitar el acceso a las TICs y promover su uso en beneficio de todos los ciudadanos. Esto incluye mejorar la infraestructura de telecomunicaciones, ampliar el acceso a internet, fomentar la capacitación en el uso de las TICs y desarrollar aplicaciones y servicios digitales que sean útiles y pertinentes para la población y las propias instituciones cubanas. De acuerdo a (González Reyes y Febles Estrada, 2021) los principales avances han sido:

- **Infraestructura tecnológica:** Se han implementado iniciativas para mejorar la infraestructura tecnológica en las instituciones educativas, como la instalación de redes de internet y la adquisición de equipamiento tecnológico. Esto permite ofrecer conectividad y recursos digitales adecuados para el desarrollo de actividades académicas.
- **Plataformas y sistemas educativos en línea:** Se han desarrollado y implementado plataformas y sistemas educativos en línea para la gestión de contenidos, la comunicación entre docentes y estudiantes, y la realización de actividades académicas. Estas plataformas facilitan la organización de cursos, la entrega de materiales didácticos y la evaluación del aprendizaje.
- **Recursos digitales y contenidos educativos:** Se han elaborado recursos y contenidos digitales para enriquecer la experiencia educativa. Esto incluye materiales didácticos interactivos, libros electrónicos, simulaciones y herramientas de apoyo al estudio. Estos recursos digitales permiten a los estudiantes acceder a contenidos actualizados y diversificados, y favorecen un aprendizaje más interactivo y autónomo.
- **Capacitación en el uso de las TICs:** Se han llevado a cabo programas de capacitación y formación para docentes y estudiantes, con el objetivo de desarrollar habilidades digitales y fomentar un uso efectivo

de las TICs en el proceso educativo. Estos programas abarcan desde el aprendizaje de herramientas tecnológicas básicas hasta la integración de las TIC en la planificación y diseño de actividades de enseñanza.

- Investigación y desarrollo en educación digital: Se promueve la investigación y el desarrollo de nuevas metodologías y enfoques pedagógicos basados en el uso de las TICs. Esto implica la exploración de estrategias innovadoras, como el aprendizaje en línea, el uso de entornos virtuales de aprendizaje y la gamificación. Estas investigaciones buscan mejorar la calidad de la educación y adaptarla a las demandas de la sociedad digital.

Como baluarte y soldado inexpugnable en el proceso de informatización de la sociedad tenemos a la Universidad de las Ciencias Informáticas (UCI). A través de sus programas educativos, la UCI brinda una formación sólida y actualizada en áreas como la informática, la ingeniería de software, la ciberseguridad y otras disciplinas relacionadas con las TICs. Esto asegura que Cuba cuente con profesionales especializados en tecnología, capacitados para liderar proyectos de desarrollo y gestión de sistemas informáticos.

Como parte de esos proyectos la UCI desarrolla un Sistema de Gestión Gerencial de Extensión Universitaria el cual se encarga de generar la información pertinente sobre el proceso de Extensión Universitaria. Dicho proceso representa una de las principales funciones de la universidad, va dirigida a los estudiantes, a la comunidad, instituciones y organismos gubernamentales.

La Extensión Universitaria (EU) en las universidades cubanas constituye uno de los procesos más importantes en la educación superior, la misma se enfoca en la formación socio-humanista, en la formación de valores y al intercambio sociocultural en el entorno en el que se encuentre la Universidad. La UCI no constituye una excepción, en dicho campus, es considerada un proceso basado en la investigación, creación, promoción y enseñanza de eventos de índole cultural, deportivo, investigativo entre otros. Para el trabajo y la organización de este proceso se generan proyectos que abarcan desde información hasta productos en etapa de desarrollo o despliegue los cuales generan un aporte científico y productivo a la Educación Superior.

Según (MES, 2004) la EU está dividida en varios subprocesos:

- Promoción de la Cultura
- Formación y Capacitación
- Cátedras Honoríficas
- Movimiento de artistas aficionados
- Proyectos extensionistas
- Instituciones culturales
- Movimiento deportivo

Cada uno de estos subprocesos manejan una gama amplia de información derivada la realización de actividades y otros eventos. El subproceso de Instituciones culturales, por ejemplo, maneja la información relacionada con la realización de talleres, conferencias, exposiciones de arte, peñas, entre otros. Con un número disímil de temas, en colaboración con importantes invitados. Lo cual hace necesario la coordinación

de los locales a utilizar, los recursos necesarios, las fechas previstas para cada actividad y la promoción que debe realizarse tomando en cuenta el tipo de público a la que está destinada. Por su parte las propias instituciones en sí como organización también generan una amplia gama de información que implica ser controlada y supervisada. Cada institución cuenta con locales o espacios que poseen recursos materiales y patrimonio cultural propio de la institución. Teniendo en consideración lo antes expuesto se hace necesario la gestión de la información relacionada a estas actividades vinculadas con las Instituciones culturales.

Como parte del proceso de la gestión de las Instituciones culturales, tras el análisis en curso se pueden identificar un número de limitaciones que caracterizan la situación problemática:

- La información recopilada de las actividades realizadas o en proceso dentro de las instituciones culturales, se encuentra dispersa y en algunas ocasiones no informatizadas. Lo que genera un atraso en la gestión de estas.
- Las plataformas individuales de las instituciones culturales no poseen un espacio en el cual se registren las actividades.
- No existe un control general de la información del proceso de gestión de las instituciones culturales en base a las actividades realizadas. Lo que dificulta el trabajo estadístico y recopilativo de la documentación adjunta.
- No existe una plataforma interactiva que realice la solicitud de actividades de acuerdo a los intereses de promotores y el público. Esto crea esquemas de burocratismos y ralentiza la realización de estas.
- Falta de capacidad de la institución para realizar un seguimiento efectivo de los recursos materiales y el patrimonio cultural, así como su protección y conservación adecuada.
- Pérdida de documentación de suma importancia para la organización de actividades. En muchas ocasiones la información se encuentra en formato no digital, lo que puede generar extravío. Y por consiguiente esta situación determina comenzar el proceso desde cero.

A partir de la situación problemática anterior queda determinado como **problema científico**, ¿Como contribuir con la gestión de la información de las instituciones culturales en el Sistema Gerencial de Extensión Universitaria?

A partir del problema de investigación formulado, se define como **objeto de estudio**: la gestión de la información de las instituciones culturales, y dentro de este se precisa como **campo de acción**: la gestión de información asociada al proceso de Instituciones Culturales en la UCI.

Para darle solución al problema antes planteado, se define como **objetivo general**: Desarrollar un módulo que contribuya a la gestión de la información de las Instituciones Culturales para el Sistema Gerencial de Extensión Universitaria.

Del mismo se derivan las siguientes tareas investigativas:

1. Elaborar el marco teórico de la investigación a partir del estado del arte existente asociada al proceso de gestión de Instituciones Culturales en la UCI.
2. Diagnosticar el estado actual de la gestión de la información asociada al proceso de Instituciones Culturales en la UCI.

3. Diseñar el sistema informático para la gestión de información asociada al proceso de Instituciones Culturales en la UCI.
4. Implementar el módulo para la gestión de información asociada al proceso de Instituciones Culturales en la UCI.
5. Validar el sistema informático para la gestión de información asociada al proceso de Instituciones Culturales en la UCI.

Para darle solución a los objetivos trazados en la investigación se aplicaron los siguientes **Métodos Científicos**:

Métodos teóricos

- **Histórico-Lógico:** se emplea para identificar posibles funcionalidades que pueda tener el sistema a partir del análisis de la evolución de los sistemas de gestión de instituciones culturales.
- **Análisis-Síntesis:** empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sistema. Así como para sintetizar la información que se obtuvo mediante la entrevista con el cliente de manera que pudiera ser usada en el desarrollo del mismo, además, en la identificación de los elementos del marco teórico de la investigación.
- **Modelación:** para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del sistema, y mediante este, identificar las características y relaciones fundamentales.

Métodos empíricos

- **Encuesta:** empleado en los encuentros con el cliente para obtener la información necesaria que permita determinar las características, cualidades y requisitos con los que debe contar la propuesta de solución.
- **Observación:** empleado para obtener el conocimiento necesario del funcionamiento de las soluciones existentes para la gestión de actividades que realizan de las Instituciones Culturales de las Instituciones de Educación Superior (IES) cubanas.

El presente trabajo de diploma está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones generales, recomendaciones y las 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.** Marco teórico referencial sobre el sistema para la gestión del proceso de promoción de la cultura en la UCI: se realiza un estudio del estado del arte sobre los principales aspectos de la investigación. Se describen el proceso de desarrollo de software, así como las tendencias, técnicas, metodología y tecnologías usadas en la propuesta de solución.
- **Capítulo 2.** Características y diseño del sistema para la gestión de Instituciones Culturales en la UCI: se identifican y describen los conceptos asociados al dominio del problema y los procesos relacionados con el negocio teniendo en cuenta la metodología seleccionada en el capítulo anterior. Se definen cuáles son los requerimientos funcionales, no funcionales y el modelo de datos.

- **Capítulo 3.** Implementación y pruebas del sistema para la gestión del proceso de promoción de la cultura en la UCI: en este capítulo se define el estándar de codificación que sirve de guía para la implementación de la solución propuesta, así como la estrategia de pruebas a aplicar para lograr un correcto funcionamiento.

Marco teórico referencial sobre el módulo Instituciones Culturales para el SGEU

En este capítulo se definen los principales conceptos que sustentan el dominio de la investigación y que posibilitan un entendimiento profundo del tema. Se expone una valoración del estado del arte de la investigación realizada con el estudio de algunas soluciones parecidas y tendencias actuales a la hora de resolver problemas similares. Se describen además, las principales tecnologías, herramientas y la metodología que se pretende usar para el desarrollo del sistema.

1.1. Conceptos asociados al tema

1.1.1. Extensión Universitaria

La Extensión universitaria EU es un proceso generado por las Instituciones de la Educación Superior IES, la cual está basada en la investigación, creación, enseñanza y promoción de eventos culturales, deportivos, investigativos, etc, dentro y fuera de la comunidad universitaria (Vega Mederos, 2017). Este proceso en concordancia de intereses y expectativas comunes promueve la interacción dialéctica universidad-sociedad. Formando parte del desarrollo evolutivo de la cultura en general y los valores fomentados por esta.

La EU contribuye destacadamente en la ejecución y promoción de la cultura dentro de las universidades. Siendo estas la vía principal para elevar el nivel cultural y contribuir al desarrollo de la comunidad universitaria. Dígase estudiantes, trabajadores y profesores. En la UCI existe un amplio patrimonio cultural y un fuerte movimiento artístico, deportivo e investigativo. Entre los que se pueden desglosar según (Estrada Molina, 2015) tenemos:

- El Movimiento de Artistas Aficionados
- El Movimiento deportivo
- Cátedras Honoríficas
- Cursos de capacitación
- Proyectos comunitarios

Estos elementos forman parte de un proceso que define e influye en el comportamiento de la sociedad universitaria.

1.1.2. Gestión de la extensión universitaria

El proceso de gestión de la EU posee numerosas dificultades en cuanto a su conceptualización, debido a la diversidad de criterios, falta de formación de sus actores, determinados por la situación económica, política y social de la nación, criterio que comparte al plantear que existe una dispersión conceptual que se traduce en las diversas aristas y factores que intervienen en la EU, dificultando el estudio del tema. Una definición aceptada pudiera ser la que propone el Programa Nacional de Extensión Universitaria (PNEU) que lo concibe como un sistema abierto y flexible con objetivos bien marcados, que se encuentra en constante intercambio de información, que permite adquirir datos y recursos para garantizar el éxito del proceso.

Este sistema aterrizado por el Trabajo Sociocultural Universitario (TSU) el cuál está formado por capas que engloban una a las otras de forma descendente: programa, proyectos, actividades, acciones y tareas, donde cada uno de estos elementos tendrían a su vez una caracterización de proceso. En esta definición según (Aportela, 2016), a pesar de coincidir con el PNEU en cuanto a lo que plantea, le hace una crítica en lo relacionado con su puesta en práctica, pues considera que descuida los procesos a tener en cuenta durante la extensión, y no posibilita vías para instrumentar coherentemente el proceso, a la vez que hace énfasis en la importancia de conocer bien a las IES, para adaptar el proceso a sus condiciones reales.

El análisis y valoración del SGC por las autoridades de la Dirección de Extensión Universitaria (DEU) del MES ha conllevado a plantear su implementación a través de un sistema de información que permita la gestión eficiente y eficaz del proceso a nivel de país.

El SGC-EU actualizado declara siete subprocesos, a saber:

- Promoción de la Cultura.
- Proyectos extensionistas.
- Cátedras Honoríficas.
- Instituciones Culturales .
- Movimiento Deportivo.
- Movimiento de Artistas Aficionados.
- Formación y Capacitación.

La figura 1.1 muestra la relación entre todos los subprocesos:

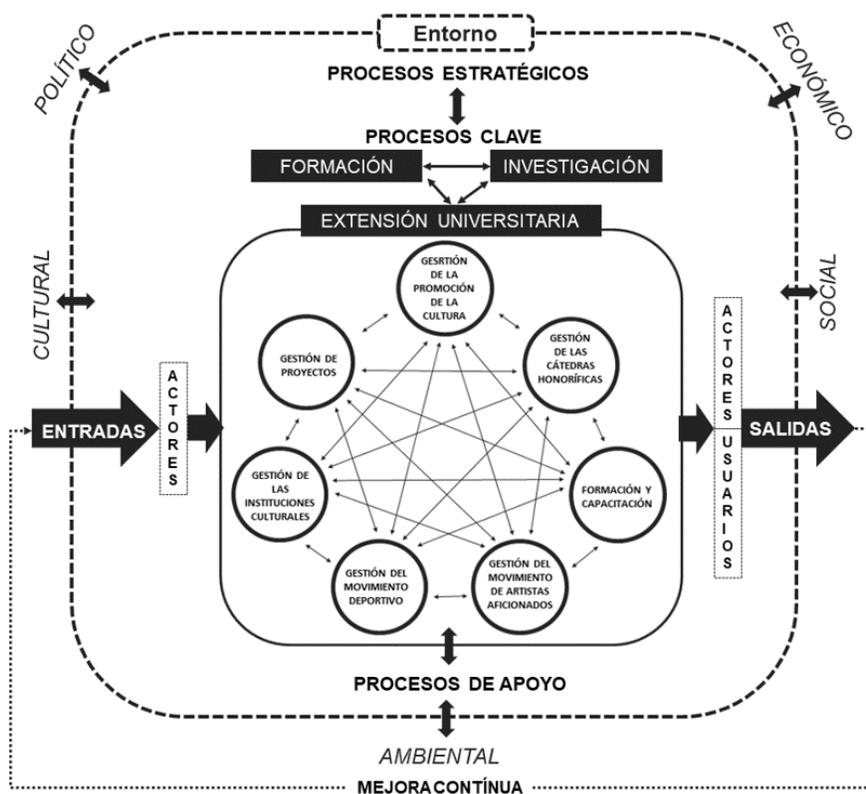


Figura 1.1. Mapa de procesos de Extensión Universitaria (Fuente: (Aportela; Fernández-Larrea; Loor; Caridad Balsin-de Herrera y Mainegra, 2022))

1.1.3. Actividades extensionistas

Son aquellas que como parte de la planeación del trabajo sociocultural universitario ofrecen tratamiento a una situación y problema específico. Se identifican como tales el conjunto de acciones y tareas que tienen un carácter similar o están estrechamente relacionadas y que se orientan al cumplimiento de un objetivo específico o a parte del mismo (Cedeño y Rivera, 2021). Las actividades pueden formar parte o no de un proyecto, su planificación puede ser independiente de estos, pero siempre estarán en el marco de la política que establece un programa. Surgen como resultado de un diagnóstico, o por el interés de un grupo en particular y mantienen las propiedades de la extensión como proceso, por lo que deben ser planificadas y poder brindar la capacidad de ver el progreso y cumplimiento de sus etapas, en pos de lograr la solución del problema o situación, por la cual fueron creadas.

La gestión de la información de las Actividades Extensionistas (AE) se encuentra relacionada con parte del proceso de gestión de la EU. Por lo que sigue una serie de pasos para la creación de actividades
Pasos para diseñar y definir una actividad según:

1. Diagnóstico de la situación actual del tema a promover.

2. Definir a partir de qué proceso va a resolver el problema.
3. Definir el objetivo de la actividad.
4. Definir el público meta.
5. Definir el tema (el que se va a tratar, debatir, proyectar)
6. Definir el tipo de actividad (taller, conferencia, curso, exposición de arte, peña literaria o artística, videodebate, entre otras).
7. Definir recursos disponibles y necesarios.
8. Realizar la convocatoria a la actividad.
 - Convocatoria al ponente, conferencista, artista u otro si es necesario.
 - Convocatoria al público. Debe contener la información completa de la actividad, hora, lugar, tema, en que consiste y definir vía de promoción (electrónica correo, intranet o web, impresa por carteles o medios impresos como periódicos o boletines, vía radial o tv).
9. Definir el local de la actividad (debe poseer las condiciones técnicas y de espacio para realizar la actividad para que el público y el exponente se sientan a gusto.
10. Ejecución de la actividad. Debe comenzar en tiempo y ocupar el tiempo promocionado, así como debe ocurrir lo que se anunció.
11. Evaluación de la actividad. Se debe anotar cantidad de participantes, cantidad de participantes que permanecieron hasta el final de la actividad, aplicar instrumentos prácticos para evaluar la satisfacción con la actividad (puede ser por observación o encuestas sencillas).
12. Análisis de la actividad realizada. Para evaluar lo que se debe mantener, modificar o determinar si ese tipo de actividad resuelve o no la situación inicial.

1.1.4. Gestión de información

Gestión de la información Gestión de la información (GI) es la denominación convencional de un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención (por creación o captura), hasta su disposición final (su archivo o eliminación). Tales procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados. El objetivo de la GI es garantizar la integridad, disponibilidad y confidencialidad de la información (Barzaga-Sablón; Pincay; Nevárez-Barberán y Cobeña, 2019). Según la revista informática-tecnológica cubana de Joven club la GI es el proceso de controlar, almacenar y recuperar la información adquirida por una entidad, a través de diferentes fuentes. De igual modo pone en uso los recursos de información de la organización, de origen interna como externa para operar, aprender y adaptarse a los cambios del ambiente (JovenClub, 2020). Se concluye que gestión de información es el control, almacenamiento y recuperación de información por parte de una entidad para garantizar la integridad, disponibilidad y confidencialidad de la misma.

1.1.5. Sistema de gestión de información

Un sistema es un conjunto de elementos dinámicamente relacionados, en interacción que desarrollan una actividad para lograr un objetivo o propósito. Un sistema de gestión de información son herramientas que se basadas en el diseño de gestión del conocimiento que propone la codificación del conocimiento explícito y la difusión y socialización del conocimiento tácito dentro de las organizaciones (M. D. Pérez, 2017). Se basan en la recopilación y presentación de datos. Su propósito es hacer que la toma de decisiones por parte de los especialistas sea más eficiente y productiva. 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.

1.1.6. Instituciones Culturales

Las instituciones culturales universitarias suelen ser entidades académicas o departamentos dentro de una universidad dedicados a promover actividades y programas culturales en el ámbito educativo. Estas instituciones tienen como objetivo fomentar la apreciación y difusión de la cultura entre la comunidad universitaria y la sociedad en general. Estas instituciones pueden adoptar diferentes formas y nombres de acuerdo a su categoría cultural: Casa de la Cultura, Biblioteca, Librería, Discoteca, Museo, Galería, entre otros. A pesar de las diferencias en la denominación, su finalidad es similar: desarrollar y promover actividades culturales dentro y fuera del ámbito universitario. Para llevar a cabo estas actividades, las instituciones culturales universitarias suelen contar con un equipo de profesionales que incluye gestores culturales, curadores, programadores, docentes y personal administrativo. Estos profesionales trabajan en colaboración con la comunidad universitaria, artistas, expertos culturales y organizaciones externas para desarrollar y ejecutar sus programas culturales. (González-Aportela; Batista-Maineira y Fernández-Larrea, 2020).

1.2. Estudio de sistemas homólogos

En esta sección se realizará un análisis de soluciones similares a la que se propone, a fin de obtener las características principales de estos sistemas, los elementos claves que no pueden faltar, las arquitecturas más utilizadas y en las condiciones en las que se usan, así como cuestiones específicas de cada solución dentro del campo de acción de la presente investigación.

1.2.1. Ámbito internacional

Portal de extensión universitaria de la Universidad de la Laguna (ULL):

La ULL se encuentra en San Cristóbal de La Laguna, Santa Cruz de Tenerife, España. La web permite una navegación intuitiva, interfaz clara, lo cual evita la sobrecarga de elementos con el fin de no confundir a los

usuarios. Brinda además la posibilidad de filtrar la información que necesita el cliente mediante un buscador; posee un módulo de cultura y extensión, y este a su vez incluye varios submódulos donde se publica información referente a cursos y talleres, actividades culturales, colectivos culturales, aulas y cátedras culturales y premios (Méndez Pérez; Santana Pérez; González de la Rosa; Albelo Jorge; Torres Padrón; Mendoza Jiménez; Dorta Afonso; Acosta Rubio et al., 2022).

Sistema Nacional de Información Cultural- SINIC (Colombia):

Herramienta Informática para el fortalecimiento del sector cultural de Colombia. Al ingresar al Sistema de Información Nacional de Cultura y registrarse (González-Hernández; Isaza-Gómez; Miranda-Calderón y Mosquera-Vente, 2022), se obtienen los siguientes beneficios:

- Dar a conocer nuestro patrimonio y acervo de expresiones culturales frente a las comunidades nacional e internacional.
- Incentivar y consolidar la comunicación y el intercambio de la información cultural.
- Fuente de consulta a través de bases de datos y redes informáticas.
- Impulso a la creación de grupos de investigación y de trabajo.
- Acceso a programas y convenios institucionales. (Posibilidad de participar en programas y proyectos, a través de convocatorias realizadas desde el Ministerio de Cultura y sus Organismos Adscritos)
- Búsqueda de proveedores y clientes (Visibilidad ante el Ministerio, Entidades y Agentes Culturales en general, tanto municipales, regionales, nacionales e internacionales) .

Sistema de Gestión de Inventario para galerías de artes. (España):

ITgallery es la manera más sencilla y elegante de gestionar, compartir y organizar tu inventario. Automatiza las tareas de tu colección y disfruta las ventajas de tener tu información bajo control en un mismo lugar. Según (Salazar, 2023) estas son las principales características del sistema:

- Tu inventario siempre accesible: La manera más sencilla de gestionar tu colección.
- Genera documentos PDF: Sin necesidad de maquetar puedes generar portafolios, fichas técnicas, certificados de autenticidad, facturas, listados de obras o formularios de envío.
- Búsquedas personalizadas: localiza fácilmente cualquier información que necesites. Aplica criterios de búsqueda para localizar la obra, exposición venta o contacto que necesitas en cada momento

1.2.2. Ámbito nacional

Sistema de gestión de la programación cultural en la dirección municipal de cultura en el municipio especial de la Isla de la Juventud

Sistema informático distribuido del tipo cliente-servidor, construido sobre la base de la utilización de software libre, con el uso de Preprocesador de Hipertexto (PHP, por sus siglas en inglés) como lenguaje de pro-

gramación. Surge en mayo de 2016 (C Martínez-Tena; Expósito-García; Ferrer-Balart y Aragón-Martínez, 2023) como parte de un Trabajo de diploma para optar por el título de Ingeniería en Informática. Cuenta con un proceso de automatización que elimina los problemas existentes en el flujo de datos del sistema:

- Insertar Actividades: El especialista en programación de cada institución será el único autorizado a insertar las actividades en el sistema.
- Publicación de la programación: El sistema mediante las actividades insertadas dará una panorámica del comportamiento de las mismas en el mes.
- Administración de programación cultural: Contará con la administración completa sobre las diferentes actividades a realizar.
- Alerta al programador general: mediante la administración de todos los datos de la programación, este alertará al programador el acercamiento de esta actividad y si se mantiene o no y si está planificada para la hora establecida.

Sistema para el área de Extensión Cultural de la Universidad de las Ciencias Informáticas.

Sistema para la gestión de actividades culturales dentro de la UCI desarrollado en el lenguaje de programación PHP, usa una arquitectura Modelo Vista Controlador (MVC, por sus siglas en inglés), permite un seguimiento de la información relacionada con la actividad, el tipo de actividad, las personas que participan y entre otras informaciones interesantes como el estado de la misma (Cabeza Matos; Anias Santos; Cruz Amarán; Llano Castro y Alonso Beatón, 2015). El sistema gestiona las actividades extensionistas pero solo las relacionadas con la cultura, no le da un enfoque más general a estas actividades imposibilitando tomar en consideración, la gestión de otras como son las de origen deportivas, por lo tanto no responde a las necesidades de crear un sistema para gestionar actividades extensionistas de forma general, que puedan ser usadas después por otros subprocesos, esto siguiendo lo planteado en la investigación en el marco de los procesos de extensión universitaria.

1.2.3. Consideraciones del estudio de los sistemas

Como resultado del estudio realizado a los sistemas anteriores, se concluye que aunque manejan aspectos vinculados a los procesos relacionados a la gestión de las instituciones culturales como el control de actividades, programación y manejo de patrimonio cultural, por sí solos estos sistemas no satisfacen la automatización completa de los procesos de una institución cultural ni se pueden ver desde una perspectiva directa que vincule a los sistemas con la propia institución por lo tanto 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 de las instituciones culturales aplicable a la UCI. Por lo tanto, constituye una necesidad primordial para la universidad desarrollar un sistema que permita gestionar dicho proceso. También el estudio posibilitó conocer las tecnologías empleadas en el desarrollo de este tipo de sistema.

1.3. Análisis de las metodologías, herramientas y tecnologías para el desarrollo del sistema

En esta sección se realizará una descripción de las principales herramientas y tecnologías que se usarán en el desarrollo del sistema para dar solución a la investigación, así como la metodología de desarrollo de software empleada.

Cabe aclarar que las herramientas y metodologías que se describen fueron establecidas por el cliente, y definidas como uno de los requisitos no funcionales (se precisa en el capítulo 2, epígrafe 2.2.2), pues la propuesta de solución es un módulo que se integra al SGEU.

1.3.1. Metodología

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo han evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo. Por lo es de gran importancia la realización de un análisis para elegir la que resulte mejor para el proyecto en cuestión (Maida y Pacienza, 2015).

Para el desarrollo del módulo de Instituciones Culturales, se decidió partir de un enfoque ágil pues se trata de un proyecto que posee un equipo de desarrollo pequeño, con poca experiencia, un alto dinamismo en cuanto a los cambios en los requisitos y poca cultura de trabajo en equipo. Por lo tanto se realizó un análisis de las metodologías ágiles más populares y se decidió hacer uso de la metodología Proceso Unificado Ágil versión UCI (AUP-UCI, por sus siglas en inglés), en su escenario número cuatro, pues es una metodología muy flexible, usada en los procesos productivos de la UCI, y fue elegida en su escenario cuatro pues, no existe la necesidad de modelar el negocio, pues el equipo cuenta con un cliente altamente comprometido y dispuesto a aclarar cualquier duda en tecnicismos del mismo, no obstante con el estudio y análisis del proceso de promoción de la cultura, se obtuvo una visión más que clara de cómo es dicho proceso.

1.3.2. Herramientas y tecnologías

Herramientas de modelado

- **Visual Paradigm v8.0:** es una herramienta Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés). 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 (Paradigm, 2022). Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de

diagramas. Posee un número de características entre las que se encuentran:

- Utiliza el lenguaje de Unificado de Modelado Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) que le permite la corrección de errores.
 - Permite la realización de diagramas de flujos de datos que pueden generarse a lo largo del desarrollo del sistema.
 - Licencia gratuita y comercial.
- **UML v2.5.1:** El Lenguaje Unificado de Modelado fue creado para forjar un lenguaje de modelado visual común, semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene. Ayudan a mostrar errores potenciales en las estructuras de aplicaciones, el comportamiento del sistema y otros procesos empresariales. Este lenguaje de modelado automatiza la producción de software y los procesos, simplifica las complejidades y ayuda a resolver los problemas arquitectónicos constantes (Koç; Erdoğan; Barjakly y Peker, 2021) . El UML será empleado en la elaboración de la mayoría de los diagramas de la presente investigación.

Lenguajes de programación

- **Python v3.10.5 :** Según (Srinath, 2017) es un lenguaje de programación diseñado por Guido van Rossum, que se caracteriza por ser de alto nivel, interpretado, de propósito general y sigue una filosofía que enfatiza la limpieza y la fácil comprensión, con el uso de indentación y la ausencia de elementos que están presentes en otros lenguajes que dificultan su entendimiento. Además de soportar múltiples paradigmas de programación entre los que están el procedural, el funcional y la Programación Orientada a Objeto (POO).

Está diseñado para resolver problemas reales del mundo de la programación, lo que se ve potenciado con su flexibilidad y el hecho de ser multiplataforma y de tipado dinámico, permitiendo a los programadores hacer su trabajo con mayor libertad, fomentando de esta manera su creatividad a la hora de encontrar soluciones.

Según (Balogun, 2022) el lenguaje de programación Python es menos complejo que el lenguaje de programación C++, luego de efectuar una comparativa de los dos, en cuanto a una métrica denominada complejidad de software, resaltando la importancia de que los desarrolladores de software tomen siempre en cuenta el hecho de que elegir un lenguaje de programación poco complejo, facilitará la administración y el mantenimiento del sistema informático que se desee crear (ibíd.).

Resumen de sus principales características según (Srinath, 2017):

- No es fácil de mantener, es necesaria experiencia para escribir código mantenible.
- Lento comparado con otros lenguajes, debido a que es muy flexible.

- Gran cantidad de librerías disponibles para resolver problemas.
- Código abierto.
- Fácil de combinar con otros lenguajes.
- Puede ser usado para realizar una gran variedad de programas.
- Posee una comunidad inmensa.
- Simple y fácil de entender.

Este lenguaje de programación es junto con las tecnologías de JavaScript uno de los más usados para el desarrollo web en el lado del servidor (Pérez, 2019), aunque en el campo del big data y aprendizaje automático se está volviendo cada vez más popular, no solo porque su extenso paquete de librerías para ello lo respalde, sino porque puede ser fácilmente integrado con las aplicaciones web.

- **JavaScript (ECMAScript 2019):** Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (ibíd.).

Este lenguaje fue creado con bajas expectativas, sin embargo en poco tiempo superó a Java como el lenguaje principal para crear páginas web dinámicas. Aunque los primeros 20 años de JavaScript se caracterizaron por fallidos intentos de mejorarlo, rediseñarlo o reemplazarlo, al final de ese período se convirtió en el lenguaje de programación más difundido en el mundo no solo para páginas web, adicionándose la posibilidad de construir aplicaciones web usando Node.js, así como aplicaciones de escritorio y para teléfonos móviles, con el uso de React Native, esto sin dejar de hablar de su uso en robots y sistemas embebidos (Wirfs-Brock y Eich, 2020).

Lenguaje de etiquetas de hipertexto

- **Lenguaje de marcado de hipertexto (HTML, por sus siglas en inglés) 5:** Es un lenguaje de marcado o etiquetas que se emplea para darle formato a los documentos o páginas web que se publican en Red Informática Mundial (WWW, por sus siglas en inglés). Fue desarrollado por Tim Berners-Lee basándose en el estándar El Lenguaje de marcado generalizado estándar (SGML, por sus siglas en inglés). La organización El Consorcio de la Red Informática Mundial (W3C, por sus siglas en inglés) se encarga de su estandarización, para así reducir los problemas de compatibilidad con los navegadores web, que son las aplicaciones que se usan para su interpretación y comprensión humana. En su versión más reciente Lenguaje de marcado de hipertexto Versión 5 (HTML5, por sus siglas en inglés), se le da mucha importancia a la semántica de las etiquetas, ayudando con esto a una mejor interpretación de los navegadores y los motores de búsqueda en donde se encuentren indexadas (Luján Mora, 2016),

(Consortium et al., 1999) y (Malqui, 2018).

Lenguaje de hojas de cascada

- **Hoja de estilo en Cascada (CSS, por sus siglas en inglés) 3:** El CSS es un lenguaje de estilos estandarizado por W3C y utilizado para definir la presentación, el formato y la apariencia de un documento de marcado como por ejemplo HTML. Nacieron con la necesidad de diseñar la presentación de la información de tal manera que resulte más comprensible, además de permitir facilitar dicho proceso y poderlo hacer para ofrecer distintas presentaciones de la información en dispositivos diferentes con características particulares a cada uno. En su última versión Hoja de estilo en Cascada Versión 3 (CSS3, por sus siglas en inglés), se ofrecen nuevas facilidades y herramientas que permiten al diseñador o programador presentar los contenidos de forma más vistosa, en menos tiempo y con menos esfuerzo, por lo que resulta una herramienta muy atractiva (Puig, 2013) y (Malqui, 2018).

Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés)

- **Microsoft Visual Studio Code v1.73.1:** Es un editor de código ligero y adaptable desarrollado por Microsoft, puede correr sobre Windows, macOS y Linux. Tiene soporte para JavaScript, TypeScript and Node.js, pero usando su rico ecosistema de extensiones se puede personalizar a niveles importantes, permitiendo construir un entorno de desarrollo ideal para cualquier programador, en casi cualquier lenguaje de programación, agregándole capacidades que rivalizan con cualquier IDE especializado (Microsoft, 2022).

Framework

- **Django v4,0.1:** Según (Gore; R. K. Singh; A. Singh; A. P. Singh; Shabaz; B. K. Singh y Jagota, 2021) es un entorno de programación en Python diseñado para el desarrollo de aplicaciones web del lado del servidor. Es un framework sencillo de instalar, con posee una documentación muy completa y una amplia comunidad. Esta herramienta está basada en la arquitectura MVC, lo que se denomina Modelo Vista Plantilla (MTV, por sus siglas en inglés), entre sus filosofías está la de la creación de aplicaciones complejas y robustas en poco tiempo, garantizando siempre la seguridad, y basada siempre en No te repitas (DRY, por sus siglas en inglés). Está diseñado para trabajar con bases de datos relacionales, aunque su flexibilidad le permite trabajar con las no relacionales también, además de ser utilizado para aplicaciones de comunicación en tiempo real por internet, así como integrada con las prácticas modernas de JavaScript.

El éxito de Django queda evidenciado en la gran cantidad de ejemplos de organizaciones importantes del mundo que adoptan esta tecnología, tales como Pinterest, Instagram, Public Broadcasting System (PBS, por sus siglas en inglés) y National Geographics, y principalmente se debe a su capacidad de optimizar el tiempo de desarrollo de sistemas complejos, pues con sus principios de diseño produce

uno de los procesos productivos más eficientes comparados con otros frameworks para el desarrollo web (**gore2021djangog**).

Cuenta con un lenguaje de plantillas, que permite separar la lógica de presentación de la lógica del negocio posibilitando numerosas ventajas para el proyecto:

- Los diseñadores no pueden afectar el código de programación y tienen total libertad a la hora del diseño sin necesidad de molestar al programador.
- Los programadores pueden efectuar cambios en el código de la aplicación sin afectar la parte visual.
- Se genera un código más limpio y legible.

Posee en su núcleo un Mapeador de Objeto Relacional (ORM, por sus siglas en inglés) que permite crear bases de datos y realizar consultas sin usar Lenguaje de Consultas Estructurado (SQL, por sus siglas en inglés), sino aplicando código Python usando las librerías de Django para ello. Esto posibilita que en caso de que sea necesario cambiar de tipo de base de datos o de gestor se hace extremadamente fácil, pues solo hay que cambiar algunas opciones en el archivo de configuración del framework.

Django posibilita definir proyectos y aplicaciones para sistemas de información web. Cuando se define un proyecto Django, automáticamente se genera una serie de archivos, y uno de los más relevantes es (`settings.py`). El archivo `setting.py` se utiliza para definir variables de configuración del sistema web tales como las aplicaciones que incluye, parámetros de conexión a base de datos, idioma, entre otros. Una vez que se define un proyecto Django, es posible definir aplicaciones para dicho proyecto. Para cada aplicación de proyecto Django, se pueden definir modelos que representan tablas en la base de datos establecida (archivo `models.py`) junto con parámetros de visualización de estos modelos, los cuales permiten mediante el ya mencionado ORM generar las tablas de la base de datos y las consultas. Además, en un proyecto Django la capa plantilla se define con archivos con extensión `.html` para la visualización de datos (archivos `.html` en carpeta `templates`). Los modelos que se usarán en una aplicación deben estar incluidos en el (archivo `admin.py`).

Sistema Gestor de Base de Datos (SGBD)

- **Postgresql v14.3:** Los SGBD son sistemas que generan independencia de datos, gestionan la concurrencia de acceso, garantiza la integridad de los datos, proveen la persistencia de los mismo y brindan la seguridad necesaria a toda la información que se encuentra almacenada en las bases de datos (Pili-cita Garrido; Borja López y Gutiérrez Constante, 2021). PostgreSQL es un SGBD de tipo relacional y código libre, que posee una comunidad activa y en constante desarrollo, que da la posibilidad de crear tipos de datos definidos por el programador y posee una alta compatibilidad con el framework Django . Se tiene buen criterio de este gestor en cuanto a su estabilidad, potencia, robustez y la facilidad de administración e implementación.

Bibliotecas

- **JQuery v3.6:** Es una biblioteca de JavaScript software libre y de código abierto, que permite simplificar el trabajo con JavaScript. Es rápida, sencilla, eficaz y poderosa, se usa para darle dinamismo y funcionalidades al frontend de las páginas web, mediante la manipulación del DOM y el manejo de eventos o acciones del usuario, facilita la aplicación de animaciones y la utilización de AJAX para simular el comportamiento de una aplicación de escritorio (Ventura Bautista, 2021). El hecho de simplificar las funcionalidades del lenguaje garantiza lograr mejores resultados con menor tiempo y esfuerzo.
- **Bootstrap v4.0:** Bootstrap es una biblioteca, dirigido al front-end de una aplicación web, por lo que está orientado a la parte que se ejecuta en el cliente, en este caso el navegador web. Utiliza HTML5, CSS y Javascript, los lenguajes por excelencia para contenido web, para hacer uso de sus librerías que permiten diseñar la web de una manera profesional creando al mismo tiempo un diseño adaptado para dispositivos móviles, con el uso de la filosofía "mobile first" del inglés móvil primero, la cual consiste en enfocarse primero en la vista de teléfonos móviles y luego ir escalando hasta llegar a las vistas de mayor resolución de pantalla (Jacob Thornton, 2022). Es código abierto y uno de los proyectos más populares de todos los tiempos en GitHub.

Control de versiones

- **Sistema de Control de versiones:** Un sistema de control de versiones guarda en el tiempo los cambios que se apliquen a un archivo o a un conjunto de ellos, permitiendo que se puedan tomar versiones específicas de ellos (Miller, 2022). Por lo tanto permite revertir los archivos a su estado previo, incluso revertir un proyecto completo, comparar cambios en el tiempo, comprobar quién fue el que modificó algo que puede haber ocasionado problemas.

Existen en la actualidad 3 tipos de sistemas de control de versiones:

- Sistema de control de versiones local
- Sistema de control de versiones Centralizado
- Sistema de control de versiones Distribuido

Para el desarrollo del módulo de instituciones culturales del para el SGEU se usará uno del tipo distribuido, muy popular en nuestros días llamado Git. El cual permite además de las ventajas anteriormente mencionadas, colaborar con diferentes grupos de personas de forma simultánea en un mismo proyecto, por lo que se ajusta perfectamente a las necesidades de trabajo del equipo de desarrollo (Chacon y Straub, 2014). Para garantizar el control de versiones en el sistema, se propone el uso del cliente Git para la plataforma Gitlab disponible en la UCI en la dirección <http://codecomunidades.prod.uci.cu>.

Herramientas para las pruebas

- **Locust v2.1** Es una herramienta de prueba de carga de usuario distribuida y fácil de usar. Está diseñado para realizar pruebas de carga de sitios web y determinar cuántos usuarios simultáneos puede manejar un sistema. La idea es que, durante una prueba, un enjambre de usuarios atacará el sitio web. El comportamiento de cada usuario, se define mediante el uso de código Python, y el proceso de enjambre se supervisa desde una interfaz de usuario web en tiempo real. Esto permite identificar cuellos de botella en el código antes de permitir el ingreso de usuarios reales. Locust se basa completamente en eventos y, por lo tanto, es posible admitir miles de usuarios simultáneos en una sola máquina. A diferencia de muchas otras aplicaciones basadas en eventos, no utiliza devoluciones de llamada. En su lugar, utiliza procesos ligeros (Pradeep y Sharma, 2019). Cada prueba dentro del sitio en realidad se ejecuta dentro de su propio proceso. Esto le permite escribir escenarios muy expresivos en Python sin complicar su código con devoluciones de llamada (ibíd.).
- **Selenium IDE v3.17** es un IDE para desarrollar pruebas basadas en web de Selenium. Se implementa como una extensión del navegador y proporciona una forma eficiente de desarrollar, registrar y depurar pruebas. El IDE de Selenium incluye todo el núcleo de Selenium, lo que permite grabar y ejecutar pruebas de manera fácil y rápida en el entorno real (Selenium; Debugging y Points, 2008). También contiene un menú contextual que permite seleccionar un elemento de la interfaz de usuario de la página que se muestra en el navegador y luego seleccionar un comando de Selenium de una lista con parámetros predefinidos. La herramienta simula clics y validación en una página web o una aplicación web. Lo hace interactuando con elementos previamente registrados en la página web .

1.4. Consideraciones parciales

Luego de haber realizado el estudio sobre el estado del arte , se puede concluir lo siguiente:

- En el desarrollo del capítulo se obtuvo una mejor dimensión acerca del problema planteado dando cumplimiento a las primeras tareas trazadas en la investigación a partir del análisis de los conceptos asociados a la solución.
- El uso de las herramientas y tecnologías seleccionadas a utilizar definieron las bases para el diseño e implementación del módulo a desarrollar.
- El estudio de los sistemas homólogos existentes posibilitó crear la base para el desarrollo del módulo a desarrollar.

Características y diseño del módulo de Instituciones Culturales para el Sistema Gerencial de Extensión Universitaria

El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, así como estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación del sistema. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

2.1. Descripción de la propuesta de solución

A partir del problema y el objetivo planteado en la presente investigación, se propone como solución un sistema informático para la gestión de la información asociada a los procesos vinculados a las Instituciones Culturales. El software se desarrollará sobre la base de una aplicación web posibilitándoles a los usuarios disfrutar de los beneficios que estas aplicaciones reportan, el acceso múltiple al sistema por medio de un servidor web, así como las actualizaciones y los mantenimientos del software sin necesidad de distribuir e instalar. La gestión de la información se realizará a través de una base de datos localizada en el servidor lo que permitirá, realizarla de forma simultánea por varios usuarios.

El sistema contribuye a la gestión de la información de las instituciones culturales, generada durante el proceso de EU en la UCI dirigido por el Dirección de Extensión Universitaria (DEU). El sistema permite la centralización de la información asociada a las AE almacenada de manera organizada, facilitando su búsqueda y visualización.

2.1.1. Actividades Culturales

A partir de los roles y permisos otorgados a los usuarios, el sistema permitirá la gestión de las actividades culturales de la institución. La entrada de nuevas actividades al sistema por parte del usuario posibilitará que dicha información se encuentre ordenada. Una actividad podrá ser solicitada por cualquier usuario

autenticado en el sistema, el cual deberá designar un responsable para la misma y llenar todos los campos del formulario que la caractericen, como son: el nombre, tema, tipo, área, lugar, fecha de inicio, fecha de fin, objetivo y obra a promover. El usuario designado como responsable tiene la opción de confirmar su responsabilidad o rechazarla. Además el responsable podrá subir evidencias tanto en formato de foto o video. Las actividades solo podrán ser modificadas por los usuarios con permiso para ello. La visualización de las actividades será en forma de lista ordenada por la fecha más reciente.

2.1.2. Gestión de Comentarios

Como parte de la retroalimentación del proceso el sistema contará con un espacio para la realización de comentarios, donde los interesados del proceso de gestión de las AE podrán dejar sus impresiones, opiniones y sugerencias del mismo, facilitando la mejora del proceso. El comentario insertado por el usuario pasará a una fase de revisión por el administrador o los usuarios con los permisos necesarios. Luego de ser revisado y comprobado el comentario pasa a la fase de publicado o puede ser eliminado si el comentario contiene ofensas o no cumple con las políticas necesarias para su publicación. El usuario que realiza el comentario puede eliminarlo si lo desea.

2.1.3. Gestión de asientos

El sistema permitirá a los usuarios autenticados reservar asientos destinados para una actividad específica en función de la cantidad de público admitido. También se podrán visualizar los detalles de los asientos en función de la actividad y el estado del mismo.

2.1.4. Gestión de locales

El sistema permitirá solo a los usuarios autenticados atender las solicitudes de local para las actividades y otorgar un espacio específico para la realización de las mismas.

2.1.5. Gestión de patrimonio

El sistema proceso permitirá registrar, guardar y preservar la información relacionada con el patrimonio cultural de la institución (obras, esculturas y pinturas) para los usuarios registrados. Los usuarios registrados podrán visualizar la información del patrimonio exclusivamente.

2.2. Especificación de requisitos del sistema

Con el objetivo de que el sistema propuesto se desarrolle con las necesidades dictadas por el cliente final a partir del uso de la tormenta de ideas, se obtuvieron los requisitos funcionales y no funcionales.

2.2.1. Requisitos funcionales

Teniendo en cuenta las características de la propuesta en cuanto a prestaciones y operatividad se identifican 53 requisitos funcionales. A continuación, la tabla 2.1 muestra los requisitos del sistema.

Tabla 2.1. Descripción de los Requisitos Funcionales. (Fuente: Elaboración propia).

No.	Nombre	Prioridad	Complejidad
RF1	Listar Instituciones	Alta	Media
RF2	Solicitar Actividad	Alta	Media
RF3	Listar Actividades	Alta	Baja
RF4	Modificar Actividad	Media	Baja
RF5	Mostrar Detalles de Actividad	Media	Baja
RF6	Buscar Actividad	Media	Baja
RF7	Listar solicitudes	Media	Baja
RF8	Buscar solicitud	Media	Baja
RF9	Aceptar solicitud con sugerencias	Alta	Baja
RF10	Aceptar solicitud sin sugerencias	Alta	Baja
RF11	Rechazar Solicitud	Alta	Baja
RF12	Reservar Asiento	Alta	Media
RF13	Listar Asientos disponibles	Alta	Media
RF14	Listar reservaciones de asientos	Alta	Media
RF15	Cancelar asiento reservado	Alta	Media
RF16	Adicionar invitado	Alta	Media
RF17	Añadir participantes	Media	Media
RF18	Listar participante por actividad	Alta	Media
RF19	Listar invitados por actividad	Alta	Baja
RF20	Eliminar participación	Alta	Baja
RF21	Ver detalles del invitado	Alta	Media
RF22	Ver detalles del participante	Media	Baja
RF23	Buscar invitado	Media	Baja
RF24	Buscar participante	Alta	Media
RF25	Insertar evidencia	Alta	Media
RF26	Insertar sugerencia	Alta	Media
RF27	Listar sugerencias	Media	Media
RF28	Valorar actividad	Alta	Media
RF39	Listar espacios disponibles por local	Alta	Media
RF30	Listar solicitudes de local	Alta	Media
RF31	Asignar espacio	Media	Media
RF32	Desvincular local	Alta	Media
RF33	Asignar recurso	Media	Media

Continúa en la siguiente página

No.	Nombre	Prioridad	Complejidad
RF34	Listar recursos asociados	Media	Media
RF35	Desvincular recurso	Alta	Media
RF36	Listar evidencia por actividad	Alta	Media
RF37	Crear comentario	Media	Media
RF38	Listar comentarios	Media	Media
RF39	Eliminar comentario	Media	Media
RF40	Mostrar detalles del comentario	Media	Media
RF41	Mostrar programación cultural semanal	Alta	Media
RF42	Mostrar programación cultural mensual	Alta	Media
RF43	Mostrar programación cultural anual	Alta	Media
RF44	Listar patrimonio cultural	Alta	Media
RF45	Insertar patrimonio cultural	Alta	Media
RF46	Eliminar patrimonio cultural	Alta	Media
RF47	Modificar patrimonio cultural	Alta	Media
RF48	Buscar patrimonio cultural	Alta	Media
RF49	Generar informe de actividades por área	Alta	Media
RF50	Generar informe de actividades por tipo	Alta	Media
RF51	Generar informe de actividades por tema	Alta	Media

2.2.2. Requisitos no funcionales

- Usabilidad.
 - RnF1. Tipo de aplicación informática: la herramienta debe ser web, que contemple estándares y patrones de diseño de Interfaz de Usuario (UI, por sus siglas en inglés) modernas.
 - RnF2. El sistema debe poseer una UI fácil de utilizar para cualquier tipo de usuario con conocimientos básicos de computación en el manejo de ordenadores.
- Confiabilidad.
 - RnF3. El sistema debe ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario.
- Hardware y software requerido para utilizar la aplicación.
 - RnF4. Sistema Operativo: Podrá utilizar cualquier sistema operativo.
 - RnF5. El sistema debe de ejecutarse sobre cualquier navegador web.
 - RnF6. La computadora que se utilice el cliente debe contar como mínimo con 2 GB de RAM y la que se utilice para el servidor debe contar como mínimo 500 GB de almacenamiento y como mínimo 4 GB de RAM.
- Eficiencia.
 - RnF7. El sistema debe permitir que los usuarios (100) interactúen con él de manera concurrente.

- RnF8. El tiempo de demora de una petición al servidor debe ser menor a cinco (5) segundos
- Seguridad.
 - RnF9. El acceso a la información debe estar restringido por usuario, contraseña y rol.
 - RnF10. Se podrá acceder a información sin necesidad de autenticarse.
 - RnF11. Cuando un usuario se autentique o registre en el sistema se le brindará la información correspondiente con su rol.
- Restricciones de diseño e implementación.
 - RnF12. El marco de trabajo que se utilizará está basado principalmente en el uso de herramientas no privativas:
 - Lenguaje de programación del lado del servidor Python v3.10.5.
 - Lenguaje de programación del lado del cliente JavaScript ECMASCRIPT2015.
 - Lenguaje para el estilo del documento web CSS3.
 - Lenguaje para la estructuración del documento web HTML5.
 - Lenguaje para el modelado del proceso UML v2.5.1.
 - Framework de desarrollo web Django v4.0.1
 - Framework para los estilos Bootstrap v5.1.
 - Herramienta para el modelado del proceso Visual Paradigm v8.0 .
 - Herramienta para la codificación Microsoft Visual Studio Code v1.73.
 - Herramienta para la realización de pruebas de rendimiento Locust v2.13.0 .
 - Herramienta para la realización de pruebas de seguridad Acunetix Web Vulnerability Scanner v9.5.
 - Herramienta para la automatización de pruebas funcionales Selenium IDE v3.17.2.
 - RnF13. Se utilizará como metodología de software la variación de AUP-UCI en su escenario No.4.

2.3. Descripción de las historias de usuario

Para el encapsulamiento de los requisitos del software, tal como lo plantea la metodología AUP-UCI, en su escenario 4, se generaron un total de 51 Historias de Usuario (HU), a continuación, se muestran las correspondientes a los requisitos funcionales registrar actividad, modificar actividad y listar actividad, el resto se encuentra en el anexo A.1:

Tabla 2.2. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Crear Actividad
Usuario: Usuario común	

Continúa en la próxima página

Tabla 2.2. Continuación de la página anterior

Prioridad en negocio: Alta	Riesgo en desarrollo: Media																								
Puntos estimados: 1	Iteración asignada: No. 1																								
Programador responsable: Arthur Javier Gargallo Fleitas																									
Descripción: El sistema debe permitir la solicitud de una actividad en determinada institución por parte de los usuarios, rellendo los campos de nombre de actividad, lugar en el que se realizará, tipo de actividad, hora de inicio y fin, fecha de inicio y fin, área, objetivo y cantidad de personas.																									
Observaciones: La UI solo aparecerá aunque el usuario no esté autenticado. No se pueden dejar campos vacíos y los campos nombre y obra a promover deben ser escritos con letra inicial mayúscula, en caso contrario el sistema debe mostrar un mensaje de error que avise al usuario el tipo de error y el lugar dónde se cometió.																									
Interfaz: <div style="text-align: center; border: 1px solid black; padding: 10px; background-color: #e0ffff; margin: 10px auto; width: 80%;"> <p>Registrar Actividad</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><input type="text"/></td> <td style="width: 25%;"><input type="text" value="▼"/></td> <td style="width: 25%;"><input type="text"/></td> <td style="width: 25%;"><input type="text" value="▼"/></td> </tr> <tr> <td style="text-align: center;">Nombre</td> <td style="text-align: center;">Tema</td> <td style="text-align: center;">Obra a promover</td> <td style="text-align: center;">Lugar</td> </tr> <tr> <td><input type="text" value="▼"/></td> <td><input type="text" value="▼"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td style="text-align: center;">Público Meta</td> <td style="text-align: center;">Área</td> <td style="text-align: center;">Fecha de inicio</td> <td style="text-align: center;">Fecha fin</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td style="text-align: center;">Hora de inicio</td> <td style="text-align: center;">Hora fin</td> <td style="text-align: center;">Objetivo</td> <td style="text-align: center;">Cantidad de personas</td> </tr> </table> <p style="text-align: center; margin-top: 10px;"><input type="button" value="Aceptar"/></p> </div>		<input type="text"/>	<input type="text" value="▼"/>	<input type="text"/>	<input type="text" value="▼"/>	Nombre	Tema	Obra a promover	Lugar	<input type="text" value="▼"/>	<input type="text" value="▼"/>	<input type="text"/>	<input type="text"/>	Público Meta	Área	Fecha de inicio	Fecha fin	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Hora de inicio	Hora fin	Objetivo	Cantidad de personas
<input type="text"/>	<input type="text" value="▼"/>	<input type="text"/>	<input type="text" value="▼"/>																						
Nombre	Tema	Obra a promover	Lugar																						
<input type="text" value="▼"/>	<input type="text" value="▼"/>	<input type="text"/>	<input type="text"/>																						
Público Meta	Área	Fecha de inicio	Fecha fin																						
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																						
Hora de inicio	Hora fin	Objetivo	Cantidad de personas																						

Tabla 2.3. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Modificar Actividad
Usuario: Usuario común	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: No. 2
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir modificar la información de una actividad en determinada institución por parte de los usuarios, rellendo los campos de nombre de actividad, lugar en el que se realizará, tipo de actividad, hora de inicio y fin, fecha de inicio y fin, área, objetivo y cantidad de personas.	
Observaciones: La UI solo aparecerá si el usuario está autenticado. No se pueden dejar campos vacíos y los campos nombre y obra a promover deben ser escritos con letra inicial mayúscula, en caso contrario el sistema debe mostrar un mensaje de error que avise al usuario el tipo de error y el lugar dónde se cometió.	

Continúa en la próxima página

Tabla 2.3. Continuación de la página anterior

Interfaz:

Tabla 2.4. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Listar Actividades
Usuario: Usuario común	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: No. 3
Programador responsable: José Adrián Torres Peraza	
Descripción: El sistema debe permitir mostrar la información de las actividades en una lista, permitiendo el análisis de los datos. Esta funcionalidad está disponible para todos los usuarios del sistema, contribuyendo a la visualización y divulgación de la información.	
Observaciones: No es necesario que el usuario esté autenticado.	

Continúa en la próxima página

Tabla 2.4. Continuación de la página anterior



2.4. Estilo arquitectónico

El diseño arquitectónico se interesa por entender cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de ese sistema. Es la primera etapa en el proceso de diseño del software. Es el enlace crucial entre el diseño y la ingeniería de requerimientos, ya que identifica los principales componentes estructurales en un sistema y la relación entre ellos. La salida del proceso de diseño arquitectónico consiste en un modelo arquitectónico que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación (Sommerville, 2011).

Para el diseño de la propuesta de solución se utilizará el estilo Llamada y Retorno. Este estilo permite que los datos sean pasados como parámetros y el manejador principal proporciona un ciclo de control sobre las subrutinas (Reynoso y Kicillof, 2019).

2.4.1. Patrón arquitectónico

Un patrón arquitectónico se puede considerar como una descripción abstracta estilizada de buena práctica, que se ensayó y puso a prueba en diferentes sistemas y entornos (Sommerville, 2011) . De este modo, un patrón arquitectónico debe describir una organización de sistema que ha tenido éxito en sistemas previos. Debe incluir información sobre cuándo es y cuándo no es adecuado usar dicho patrón, así como sobre las fortalezas y debilidades del patrón .

Como propuesta de solución para el desarrollo de la aplicación se determinó el patrón MTV que propone Django como una versión del MVC. La figura 2.1 muestra el diagrama de paquetes de la propuesta de solución siguiendo el patrón MTV.

El funcionamiento de este estilo comienza cuando el usuario a través del navegador envía la solicitud a la Vista. La vista se encarga de ejecutar la lógica empresarial e interactuar con un modelo para transportar datos y representar una plantilla. El modelo ayuda a manejar la base de datos. Es una capa de acceso a datos que maneja los datos. La plantilla es una capa de presentación que maneja completamente la parte de la interfaz de usuario. La vista se utiliza para ejecutar la lógica empresarial e interactuar con un modelo para transportar datos y representar una plantilla.

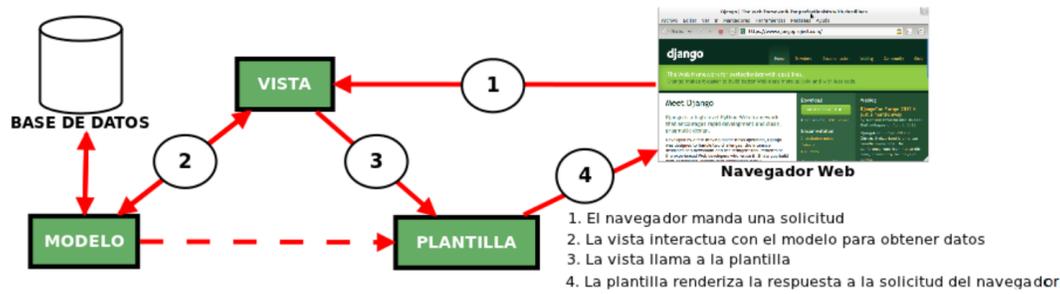


Figura 2.1. Representación de la arquitectura (Solórzano Ávila, 2018)

- **Modelo:** Es la parte del MTV que hace de medio de almacenamiento de datos, donde se puede indicar y controlar el comportamiento de los datos. Representa la información con la que trabaja la aplicación. Este no tiene conocimiento específico de las vistas o de las plantillas, ni siquiera contiene referencias a ellos.
- **Vistas:** Contiene la lógica de la aplicación, la que responde a las peticiones de los usuarios e interactúa con la template y hace uso del modelo de datos para poder acceder a la información almacenada. Es el equivalente del controlador en MVC. Es necesaria para devolver una respuesta hacia el cliente que la solicita, también procesa las peticiones o solicitudes que accederán al modelo para poder entregar u obtener los datos.
- **Plantillas:** Es la interfaz de usuario, es la parte del Framework que permite mostrar los datos almacenados y procesados. Da la posibilidad al usuario de interactuar con el sistema. Decide la forma en la que se presentarán los datos devueltos por la vista en el navegador web utilizando estilos CSS o brindando dinamismo a través de JavaScript.

2.5. Diagrama de clases de diseño

Un diagrama de clase es un tipo de diagrama Lenguaje Unificado de Modelado (UML) donde se describe el sistema a través de los diferentes tipos de objetos que lo componen, las relaciones entre ellos, sus métodos y atributos (Koç; Erdoğan; Barjakly y Peker, 2021). Al aplicar estereotipos web se describen los componentes que conforman el modelo, la vista y plantillas presentes en la propuesta solución. Fueron generados en total 10 diagramas de clase. A continuación se presenta el diagrama correspondiente a la gestión de las actividades de las instituciones culturales. Ver Figura 2.2

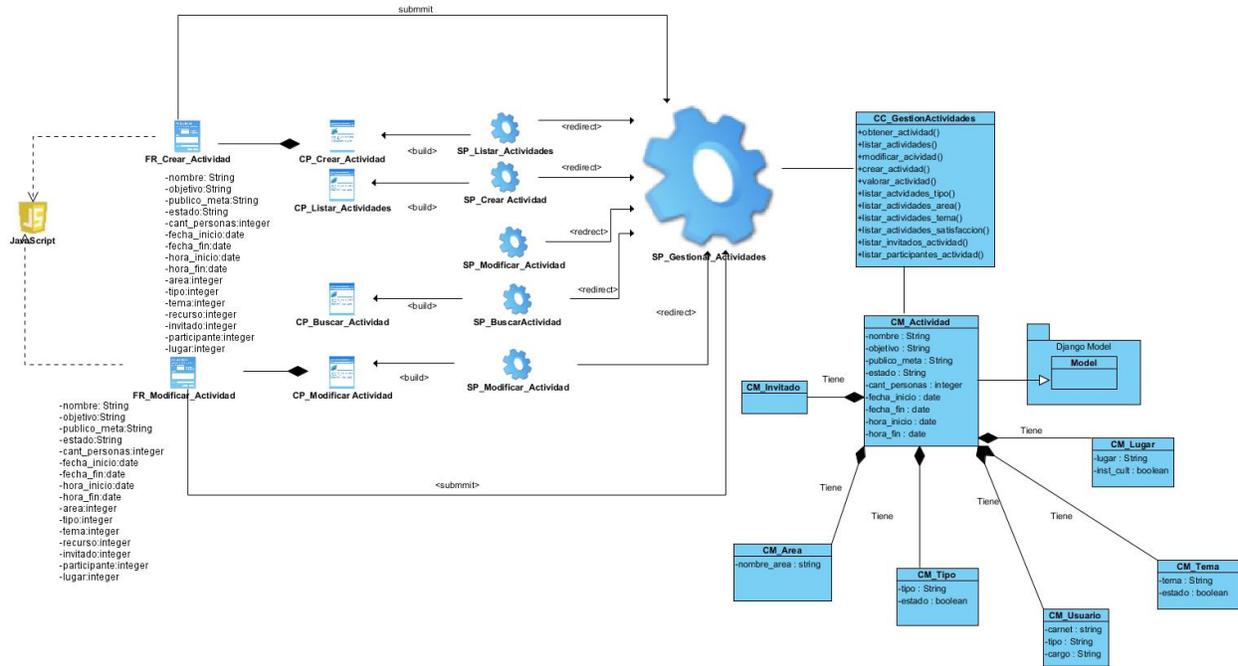


Figura 2.2. Diagrama de clases de gestión de Actividades (Fuente:Elaboración propia)

Descripción de las clases

El diagrama hace uso de estereotipos Web para facilitar su comprensión las <client page>enfocan en la parte visual del sitio, donde aquellas que cuentan con un formulario necesitan, para su funcionamiento del paquete JavaScript, para garantizar el envío de los datos al servidor y su validación en el cliente, mediante los mensajes <submit>las <server page>se refieren a las encargadas del envío de los datos al cliente y de los objetos html, css y javascript que serán renderizados en las client page, así como del enrutamiento y la conexión con las vistas, dentro de la arquitectura de Django quedan recogidas en las plantillas. Las clases CC se refieren a las clases controladoras que se reflejan en el funcionamiento de las vistas del framework, recogen la lógica del negocio en forma de tener la capacidad para responder a las acciones del cliente y en funcionalidades que se ejecuten sin este conocerlo, permitiendo el correcto funcionamiento de la aplicación, sirviendo de mediador entre el cliente y los datos. Por último las CM, son las clases modelos, las cuales definen la lógica de almacenamiento de los datos con los que funciona el sistema, dentro de la arquitectura que define el marco de desarrollo, forman parte del modelo, donde dichas clases, para usar todas las potencialidades que ofrece Django deben heredar de Model; clase que se encuentra en el módulo models de Django, que permite hacer uso de su ORM.

2.6. Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño que puede ser utilizada de forma repetida, cuya efectividad ya está probada. Puede ser usado en diferentes contextos, no es código de programación sino una descripción de cómo resolver un problema dado. Con el uso de ellos en la confección de una aplicación, se garantizarán diseños flexibles, reusables y modulares. Cada patrón resuelve un problema en específico, por lo que es importante conocerlos y aplicarlos en la situación en la que lo requiera. Son soluciones bien detalladas que promueven buenas prácticas, facilitando su comprensión y reusabilidad (Gamma; Helm; R. Johnson; R. E. Johnson; Vlissides et al., 1995; Pressman, 2005).

1. Patrones Generales de Software de Asignación de Responsabilidad (GRASP, por sus siglas en inglés): Los patrones GRASP, más que patrones propiamente dichos, son una serie de buenas prácticas de aplicación recomendable en el diseño de software. Entre ellos, los patrones Experto, Creador, Bajo acoplamiento y Alta cohesión guardan directa relación con la creación y asignación de responsabilidades a los objetos (Piñeiro Vidal y Roa Parets, 2020).

- **Experto:** Posibilita una adecuada asignación de responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con reutilización de componentes. Trae como beneficios que se mantiene el encapsulamiento de la información puesto que los objetos usan su propia información para llevar a cabo tareas, lo que permite crear sistemas más robustos y mantenibles, se distribuye el comportamiento de las clases que contienen la información requerida, por lo que las clases se vuelven más fáciles de entender y mantener. Este patrón está presente en las clases controladoras `instituciones.py`, `locales.py`, y `patrimonio.py`; las cuales tienen implementadas las funcionalidades de crear, obtener, modificar y eliminar de las entidades que manejan respectivamente. (ibíd.).
- **Creador:** Aporta un principio general para la creación de objetos, una de las actividades más frecuentes en programación. Asigna a una clase la responsabilidad de crear instancias de la otra cuando se cumplen determinadas condiciones en su relación, como son que la primera agrega a la segunda, o contiene los datos de inicialización de la segunda o utiliza estrechamente a esa clase para su funcionamiento (ibíd.). Se evidencia en las clases `localC` y `Patrimonio` al crear instancias de `locales` y `patrimonios`.
- **Bajo acoplamiento:** Es una medida de la fuerza con que una clase se relaciona con otras, porque las conoce y recurre a ellas; una clase con bajo acoplamiento no depende de muchas otras, mientras que otra con alto acoplamiento presenta varios inconvenientes; es difícil entender cuando está aislada, es ardua de reutilizar porque requiere la presencia de otras clases con las que esté conectada y es cambiante a nivel local cuando se modifican las clases afines (ibíd.). Se evidencia al separar las clases del modelo, de las vistas y de los controladores .
- **Alta cohesión:** Es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal; una clase con baja cohesión

realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar. Este patrón expresa que se debe asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad (Piñeiro Vidal y Roa Parets, 2020). Se evidencia en las clases clientes y las servidoras, se encargan cada una de realizar una función específica, permitiendo que las mismas no se saturen.

- **Controlador:** Le asigna a una clase la responsabilidad de manejar las respuestas del sistema a las acciones de los usuarios. El controlador define el método para la operación del sistema. Permite aumentar el potencial para la reutilización, asegura que la lógica de la aplicación no se maneje en la interfaz y permite validar las operaciones de manera eficiente y segura (ibíd.). Se evidencia en las clases `instituciones.py`, `locales.py`, entre otras
2. Pandilla de cuatro (GOF, por sus siglas en inglés): Son patrones usualmente aplicados en el trabajo orientado a objetos. Los mismos se clasifican de acuerdo a sus funciones en tres grupos: creacionales, estructurales y de comportamiento. De ellos se utiliza:
- **Decorador:** Este patrón permite añadir dinámicamente funcionalidad a un objeto. Permitiendo crear nuevas clases que hereden de la primera, pero añadiendo nuevas funcionalidades (ibíd.). Este patrón se evidencia en la herencia que existe en las plantillas HTML donde se hereda de `MenuPieDP.html` para reutilizar el menú lateral y el pie de página que se mantiene en toda la aplicación.
 - **Observer (Observador):** Es un patrón de comportamiento, tiene como función fundamental avisar cuando se realiza algún cambio sobre el objeto que estamos observando, garantizando la comunicación entre los componentes. Está compuesto por dos partes, el sujeto que es el objeto cuyo estado quiere vigilarse a largo plazo y los objetos observadores, que han de ser informados de cualquier cambio en el sujeto (ibíd.). Este patrón se aplica en el método `signal @receiver` el cual se encarga del monitorear o cuando se realiza un adición en la entidad `Actividad` para luego crear una cantidad de objetos de tipo `Asiento` igual al atributo `cant-personas` del objeto `Actividad`.

2.7. Modelo de datos

Para la determinación de una estructura lógica de una base de datos y el modo de almacenar, organizar y manipular los datos se emplea el modelo de datos (Pressman, 2005). Con el objetivo de definir las clases persistentes se identifican los conceptos, en el dominio del negocio, que persisten en el tiempo. A partir del diagrama de clases persistentes se generó el siguiente diagrama entidad-relación, el cual posee un conjunto de tablas correspondientes a cada componente:

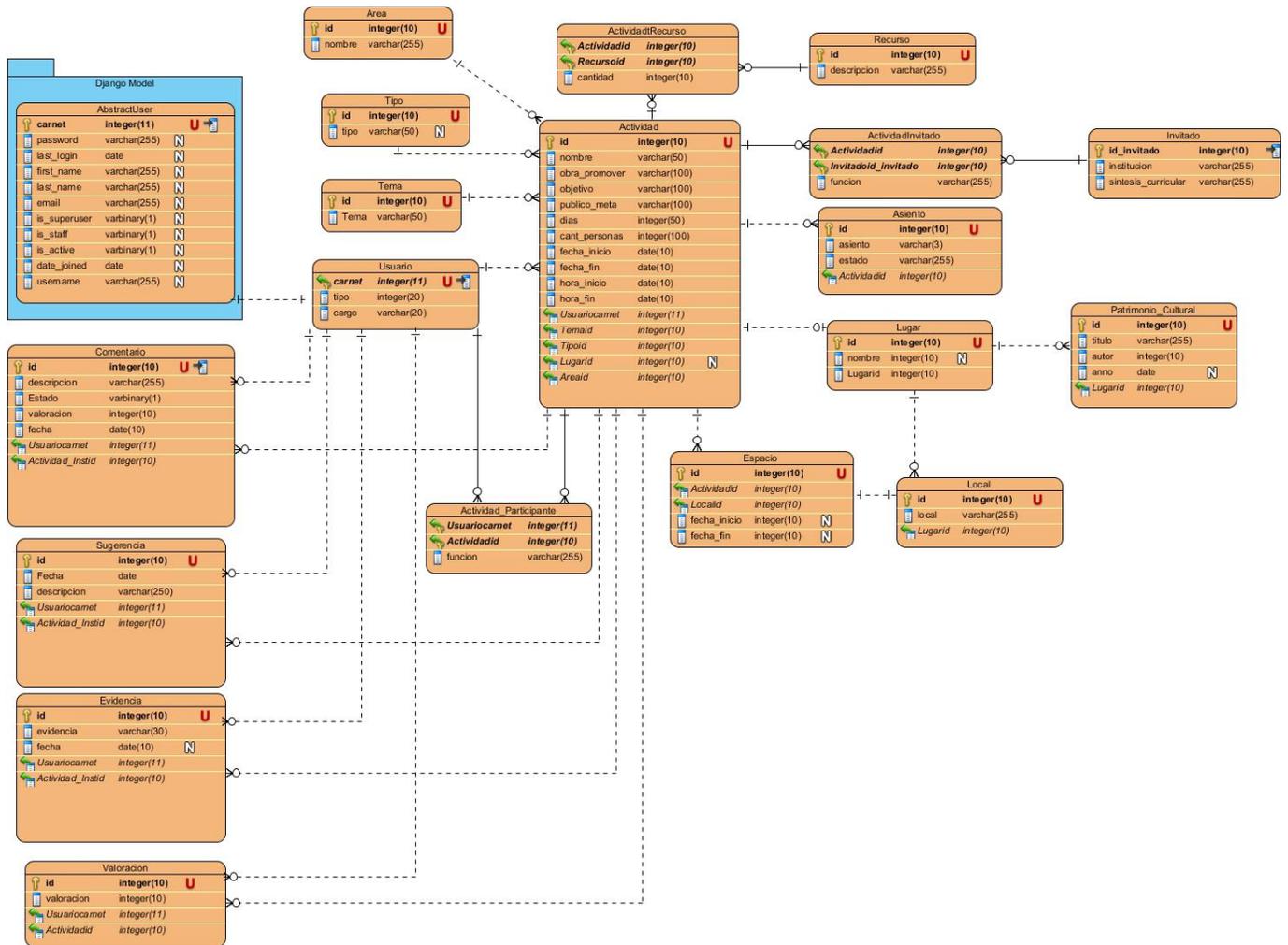


Figura 2.3. Modelo de datos (Fuente:Elaboración propia)

Descripción de las tablas

1. Actividad: Tabla principal del modelo de datos en la que se registran las actividades con sus características.
2. Lugar: Tabla nomencladora de las instituciones que se pueden usar para crear actividades.
3. Area: Tabla nomencladora de las áreas que se pueden usar para crear las actividades.
4. Tipo: Tabla nomencladora de los tipos de actividades que existen.
5. Tema: Tabla nomencladora de los temas de actividades que se pueden usar.
6. Evidencia: Tabla que recoge todas las evidencias de las actividades, donde una actividad puede tener muchas evidencias, pero una evidencia es de una sola actividad.
7. Recurso: Tabla nomencladora de los recursos que existen.
8. RecursoActividad: Tabla que recoge los recursos de cada actividad, donde una actividad puede tener

varios recursos y un recurso puede estar en varias actividades.

9. **ActividadParticipante:** Tabla que relaciona a las actividades con los usuarios, que serían los participantes, una actividad puede tener varios participantes, y un participante puede estar en varias actividades los invitados, la cual refleja los invitados de una actividad.
10. **Invitado:** Tabla que guarda los invitados dentro del sistema, posee una relación de mucho a mucho con la tabla Actividad.
11. **Usuario:** Tabla que guarda los usuarios del sistema, usa la tabla User proporcionada por Django en una relación de especialización generalización, lo que permite asignarle roles, permisos, así como la autenticación en el sistema, mediante las funcionalidades del framework.
12. **AbstractUser:** Clase modelo proporcionada por Django, que contiene todos los campos de la tabla User del framework, y permite que el modelo User sea redefinido usando herencia, para de esta forma personalizarlo.
13. **Valoracion:** Tabla que recoge las valoraciones realizadas por cada usuario por cada actividad. Se entiende por valoración, cada uno de los registros de esta tabla.
14. **Comentario:** Tabla que guarda los comentarios realizados a cada actividad. Responde a las relaciones de un usuario hace varios comentarios y un comentario es de un usuario, mientras que un comentario es de una actividad y una actividad tiene varios comentarios.
15. **Sugerencias:** Tabla que guarda todas las sugerencias de los usuarios para la mejora del proceso. Responde a la relación un usuario tiene muchas sugerencias y una sugerencia es de un usuario.
16. **PatrimonioCultural:** Tabla que guarda todas las obras que representan el patrimonio cultural de la institución: Posee una relación de uno a muchos con Lugar.
17. **localC:** Tabla que guarda todas los sitios disponibles para la realización de actividades en una institución. Posee una relación de uno a muchos con Lugar, debido a que una institución posee varios locales.
18. **Espacio:** Tabla que guarda todas las solicitudes de locales realizadas. Posee una relación de uno a muchos con Actividad, debido a que una actividad puede solicitar varios espacios en caso de que no se apruebe en uno y con localC dado para un local puede haber muchas solicitudes de espacio.

2.8. Consideraciones parciales

Luego de haber realizado el análisis y diseño del sistema propuesto y generar los artefactos que dispone la metodología seleccionada, se puede concluir lo siguiente:

- La descripción de la propuesta de solución a través del análisis de sus principales características, permitió identificar las funcionalidades del sistema propuesto, acorde a las necesidades del cliente para contribuir a la gestión las instituciones culturales en la UCI.
- La especificación de los requisitos del sistema permitió describir cada uno de los requisitos funcionales a través de las HU, así como identificar los requisitos no funcionales teniendo en cuenta un

conjunto de características generales y restricciones del mismo.

- El establecimiento de la arquitectura del sistema a partir del patrón arquitectónico, facilitó aislarse del problema en que se enmarca la presente investigación, asegurando un entendimiento común entre las partes interesadas.
- La identificación de los patrones de diseño permitió disminuir el impacto de los cambios futuros en el código fuente de la misma.
- Las características abordadas sobre el diseño del sistema a través de la comprensión de cada elemento que lo componen, facilitó el enfoque en cuanto a composición lógica y física de la propuesta de solución.

Implementación y pruebas del módulo de Instituciones Culturales para el Sistema Gerencial de Extensión Universitaria

En el presente capítulo se abordan los principales aspectos relacionados con la implementación de la solución propuesta. Se definen los estándares de codificación para lograr una mejor comprensión y estandarización del código resultante. Se realiza la verificación del correcto funcionamiento del software mediante la aplicación de las pruebas correspondientes y se presenta el nivel de satisfacción de los usuarios de la solución implementada aplicando la técnica de Iadov.

3.1. Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código (Llerena Izquierdo, 2020). Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento.

A continuación, se define el estándar de codificación empleado en el sistema.

Tabla 3.1. Estándares de codificación a utilizar en la implementación del sistema. (Fuente: Elaboración propia).

Tipo de estándar	Descripción
Organización del código	<ul style="list-style-type: none"> • El código en una página se organizará por bloques • La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios
Continúa en la siguiente página	

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL MÓDULO DE INSTITUCIONES CULTURALES PARA EL SISTEMA GERENCIAL DE EXTENSIÓN UNIVERSITARIA

Tipo de estándar	Descripción
Máxima longitud de las líneas	<ul style="list-style-type: none"> • Todas las líneas deben estar limitadas a un máximo de setenta y nueve caracteres • Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.
Líneas en blanco	<ul style="list-style-type: none"> • Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco • Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco • Se puede utilizar líneas en blanco escasamente para separar secciones lógicas
Importación	<ul style="list-style-type: none"> • Las importaciones deben estar en líneas separadas • Siempre deben colocarse al comienzo del archivo • Deben quedar agrupadas de la siguiente forma: <ul style="list-style-type: none"> ◦ Importaciones de la librería estándar ◦ Importaciones terceras relacionadas ◦ Importaciones locales de la aplicación / librerías • Cada grupo de importaciones debe estar separado por una línea en blanco
Continúa en la siguiente página	

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL MÓDULO DE INSTITUCIONES CULTURALES PARA EL SISTEMA GERENCIAL DE EXTENSIÓN UNIVERSITARIA

Tipo de estándar	Descripción
Espacios en blanco en expresiones y sentencias	<ul style="list-style-type: none"> • Evitar utilizar espacios en blanco en las siguientes situaciones: <ul style="list-style-type: none"> ◦ Inmediatamente dentro de paréntesis, corchetes y llaves ◦ Inmediatamente antes de una coma, un punto y coma o dos puntos ◦ Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función ◦ Inmediatamente antes de un corchete que empieza una indexación ◦ Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro • Deben rodearse con exactamente un espacio los siguientes operadores binarios: <ul style="list-style-type: none"> ◦ Asignación (=) ◦ Asignación de aumentación (+=, -=, *=, /=) ◦ Comparación (==, <, >, >=, <=, =, <>, in, not in, is, is not) ◦ Expresiones lógicas (and, or, not) • Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad • No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto
Comentarios	<ul style="list-style-type: none"> • Los comentarios deben ser oraciones completas • Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula • Si un comentario es corto el punto final puede omitirse • Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código JavaScript mientras que en Python por el carácter “#” y deben ubicarse en la misma línea que se desea comentar
Continúa en la siguiente página	

Tipo de estándar	Descripción
Convenciones de nombramiento	<ul style="list-style-type: none">• Nunca se deben utilizar como simples caracteres para nombres de variables los caracteres ele minúscula “l”, o mayúscula “O”, ele mayúscula “L” ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0)• Los módulos deben tener un nombre corto y en minúscula.• Los nombres de clases deben utilizar la convención “CapWords” (palabras que comienzan con mayúsculas)• Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guión bajo “_”• Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guión bajo (_)

A continuación, se muestra un fragmento de código donde se evidencia el uso de la codificación.

```
class Mis_Actividades(LoginRequiredMixin, ListView):
    model = Actividade
    login_url = 'actividades'
    template_name = "Actividades/listar/actividades/responsable/mis-actividades.html"
    context_object_name = 'lista_objetos'
    paginate_by = 4
```

Figura 3.1. Ejemplo de código en Python.

3.2. Diagrama de despliegue

Los diagramas de despliegue muestran cómo los componentes de software se despliegan físicamente en los procesadores; es decir, el diagrama de despliegue muestra el hardware y el software en el sistema, así como el middleware usado para conectar los diferentes componentes en el sistema. En esencia, los diagramas de despliegue se pueden considerar como una forma de definir y documentar el entorno objetivo (Sommerville, 2011).

A continuación, la figura 3.2 muestra el diagrama correspondiente al sistema propuesto.

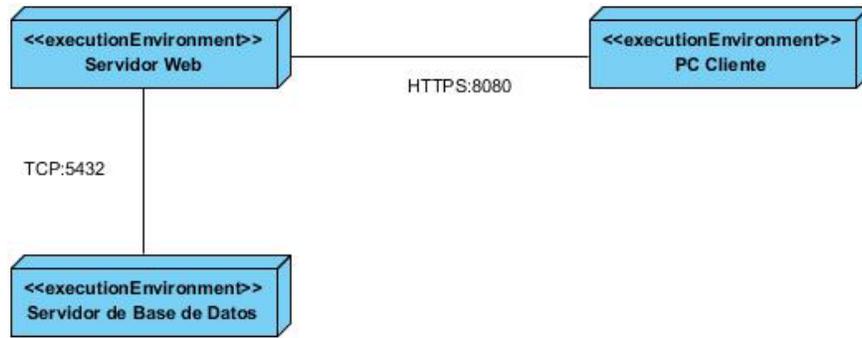


Figura 3.2. Representación del diagrama de despliegue.

PC Cliente: Se refiere a el conjunto de todos los clientes que consumirán el software desde sus computadoras.

Sevidor Web: Elemento de cómputo, dedicado al procesamiento y la lógica de la aplicación web, alimentada por la aplicación servidora.

Sevidor BD: Elemento de cómputo, dedicado a almacenar y proveer datos necesarios para el funcionamiento de la aplicación web.

3.3. Pruebas de software

Las pruebas intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para buscar errores, anomalías o información de atributos no funcionales del programa (Sommerville, 2011).

A fin de encontrar los errores del sistema y garantizar un nivel aceptable de calidad y confianza, se realizaron pruebas de software, de caja negra y caja blanca, tanto manuales como estáticas, haciendo uso de las principales técnicas existentes, y aprovechando los módulos unittest.py y TestCase.py, ofrecidos por Python y Django respectivamente, que facilitan las pruebas automatizadas, apoyándose en la estrategia de pruebas recomendada por el framework, Desarrollo dirigido por pruebas (TDD, por sus siglas en inglés).

3.3.1. Estrategia de pruebas

A continuación, se muestra la estrategia de prueba a seguir para validar la propuesta de solución.

Continúa en la siguiente página

Tabla 3.2. Estrategia de pruebas. (Fuente: Elaboración propia).

Pruebas	Método	Herramienta	Alcance
Funcional	Caja negra con particiones equivalentes	Selenium IDE	Se probará el funcionamiento del 100 % de los requisitos.
Unitaria	Caja blanca con la técnica del camino básico	TestCase de django para la realización de pruebas automatizadas	Se automatizarán pruebas para las unidades de código separadas por módulos.
Rendimiento	Pruebas de carga y estrés	Locust	Se aplicará sobre un entorno de pruebas con prestaciones menores que el de despliegue, debido a falta de recursos. Se probará la aplicación con 100 usuarios concurrentes buscando tiempos de respuesta menores a 5 segundos.
Seguridad		Acunetix web vulnerability scanner 9.5	Se aplicará para detectar vulnerabilidades: <ul style="list-style-type: none"> • Inyección SQL • Programación Cross-Site (o XSS) • Ataques de fuerza bruta a las credenciales • Redirecciones y reenvíos no validados
Aceptación	Pruebas de alpha y beta	Técnica de Iadov	Se le aplicará la técnica con un cuestionario a un grupo de especialistas y trabajadores con años de experiencia en el proceso de promoción de la cultura. El éxito de la prueba se medirá: <ul style="list-style-type: none"> • Insatisfacción: desde (-1) hasta (-0,5) • Contradictorio: desde (-0,49) hasta (+0,49) • Satisfacción: desde (+0,5) hasta (1)
Ejecutado por:	José Adrian Torres Peraza y Arthur Javier Gargallo Fleitas		

3.3.2. Pruebas unitarias

Se aplican a un componente del software. Podemos considerar como componente, a una función, una clase, una librería, etc. Estas pruebas las ejecuta el desarrollador, cada vez que va probando fragmentos de código o scripts para ver si todo funciona como se desea. El objetivo de las pruebas unitarias es aislar cada parte

del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito, que el fragmento de código debe satisfacer. El método utilizado para realizar este tipo de prueba se denomina caja blanca (Sommerville, 2011).

Método de caja blanca

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo
- Se utilizan las decisiones en su parte verdadera y en su parte falsa
- Se ejecuten todos los bucles en sus límites
- Se utilizan todas las estructuras de datos internas.

Para la realización de las pruebas unitarias, se le aplicó la técnica de prueba del camino básico a las unidades código que responden a funcionalidades críticas del software, lo cual permitió generar el grafo de flujo, calcular la Complejidad Ciclomática (CC) para determinar los caminos linealmente independientes y el número mínimo de escenarios de los casos de prueba para forzar la ejecución de cada camino del conjunto básico.

Luego en apoyo a las pruebas se usó el módulo TestCase que ofrece el framework, Django, para la automatización de las pruebas unitarias. Con él se probó cada módulo desarrollado, y gracias a la aplicación de la técnica de camino básico, en aquellas funcionalidades críticas, se pudieron automatizar pruebas para cada uno de los escenarios o caminos posibles, garantizando probar todo el código en cuestión.

Entre los elementos de código que fueron probadas se encuentra el referente a post de la clase , que se encarga de controlar la lógica de aprobar una solicitud.

Tabla 3.3. Cálculo de la complejidad ciclomática del método post de la clase aprobar solicitud

Método

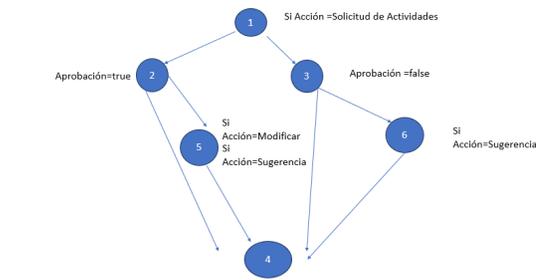
```
class Sugerencias_actividad(View):
    def get(self, request, pk, **kwargs):
        if kwargs['estado'] == '2':
            tipo = '1'
        elif kwargs['estado'] == '5':
            tipo = '2'
        elif kwargs['estado'] == '7':
            tipo = '3'

        my_form = SugerenciaForm(
            initial={"usuario": request.user.carnet, "actividad": pk, 'tipo': tipo})
        template_name = "Actividades/crear-update/sugerencia_form.html"
        actividad_x = Actividade.objects.get(pk=pk)
        estado = kwargs['estado']
        context = {'actividad': actividad_x, "form": my_form, 'estado': estado}
        return render(request, template_name, context)

    def post(self, request, pk, **kwargs):
        form = SugerenciaForm(request.POST)
        estado = kwargs['estado']
        if estado == '2':
            success = "propuestas-actividades"
        elif estado == '5':
            success = "solicitudes_area"
        elif estado == '7':
            success = "solicitudes_IES"
        actividad_x = Actividade.objects.get(pk=pk)
        template_name = "Actividades/crear-update/sugerencia_form.html"

        if form.is_valid():
            form.save()
            Metodo_Cambio_Estado(pk, estado)
            return redirect(success)
        else:
            return render(request, template_name, {"form": form, "errors": form.errors, "actividad": actividad_x})
```

Grafo resultante:



Complejidad Ciclomática:

$V(G) = \# \text{ de regiones}$
 $V(G) = 4$

$V(G) = A - N + 2$
 $V(G) = 8 - 6 + 2$
 $V(G) = 4$

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

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.

Como resultado se obtuvo que la CC es igual a 4, lo que significa que existen cuatro posibles caminos linealmente independientes y hay que diseñar un mínimo de cuatro casos de prueba para el algoritmo. La

tabla 3.4 muestra los caminos existentes.

Tabla 3.4. Caminos del grafo de flujo (Fuente: Elaboración propia).

No.	Camino
1	1,2,4
2	1,2,5,4
3	1,3,4
4	1,3,6,4

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.

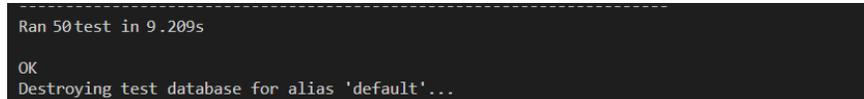
Tabla 3.5. Caso de Prueba para el camino básico 1 (Fuente: Elaboración propia).

Proceso:
Administrar solicitudes.
Casos de prueba:
Aprobar Actividad Cultural 1.1.
Camino independiente:
1,2,5,4
Entradas:
<ul style="list-style-type: none"> • Sugerencia:Comentario
Resultados esperados:
<ul style="list-style-type: none"> • El sistema aprueba una solicitud con algunas sugerencias y la solicitud se vuelve actividad.
Condiciones de ejecución:
<ul style="list-style-type: none"> • Usuario autenticado. • Usuario con rol de responsable de la institución.

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. La obtención de la CC de valor 4 del método post ejemplificado, permitió determinar que existen 4 caminos linealmente independientes, suficientes para probar el código al menos una vez.

Los resultados del método de caja blanca fueron satisfactorios. Se automatizaron un total de 53 casos de

prueba con el uso de la biblioteca TestCase, de los cuales a 7 se le aplicó la técnica del camino básico, permitiendo que su automatización garantice probar todos los caminos con un mínimo de escenarios diseñados, y obteniendo 0 errores.



```
Ran 50 test in 9.209s
OK
Destroying test database for alias 'default'...
```

Figura 3.3. Resultado de las pruebas unitarias.

3.3.3. Pruebas de Integración

Las pruebas de integración son una técnica para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz y el código. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño. Existen dos tipos de integración no incremental e incremental la integración no incremental combina todos los componentes y se prueba el programa en su conjunto. La integración incremental consiste en probar el programa en pequeños segmentos para facilitar el aislamiento y corrección de errores. En la propuesta de solución se escogió el enfoque incremental para la realización de las pruebas de integración mediante la estrategia integración descendente, aplicando la técnica de integración primero en anchura. Esta técnica incorpora todos los componentes directamente subordinados en cada nivel y se mueve horizontalmente a través de la estructura. Las pruebas de integración se desarrollaron luego de las pruebas unitarias y de forma incremental. Las pruebas funcionales de integración son similares a las pruebas de caja negra. En la disciplina de pruebas aplicada a la solución se integran los diferentes componentes a un proyecto que presenta una determinada estructura. En este proyecto se ubican los elementos de los componentes en los directorios correspondientes, garantizando la correcta integración al mismo:

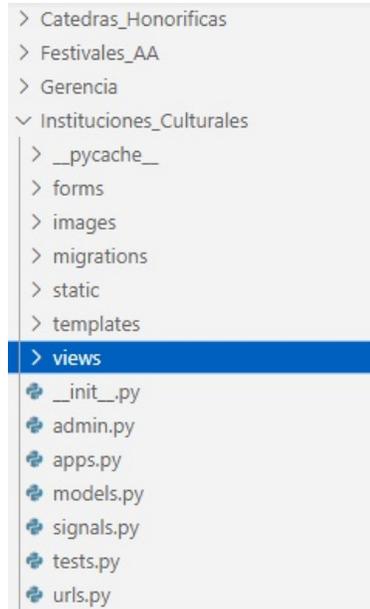


Figura 3.4. Representación de la integración del directorio del Módulo en cuestión al proyecto.

El directorio más importante es Gerencia, aquí es donde se cargan todos los estilos además de la gestión de los componentes del proyecto. Algunos de los directorios más importantes también son:

- Media: Gestiona la parte de los datos guardados correspondiente a la multimedia que se solicita a través de los diferentes módulos.
- Sistema-Gestion-EU: Administra las URLs de cada uno de los módulos del proyecto, así como la configuración y gestión de las conexiones con la Base de Datos.

La estructura del proyecto permite definir como se integran y se relacionan los componentes con los diferentes directorios. Mediante la aplicación de la técnica de integración primero en anchura se establece un diagrama de tres niveles con los principales componentes que se integran (ver figura 3.6).



Figura 3.5. Integración Descendente

Los componentes Inicio, Mov.Deportivo, Reportes, Cursos, C.Honoríficas, F.Artistas.A, Proyectos.EU y Promoción.C e Instituciones Culturales se integran primero al módulo Gerencia y este a su vez al nivel de control APP. El proceso de integración se realiza en una serie de cinco pasos:

- El módulo de control principal se usa como un controlador de prueba y los representantes se colocan con todos los componentes directamente subordinados .
- Los demás módulos se sustituyen uno a la vez con componentes reales atendiendo al enfoque primero en anchura.
- Las pruebas se llevan a cabo conforme se integra cada componente .
- Al completar cada conjunto de pruebas otro representante se sustituye con el componente real.
- Las pruebas de regresión pueden realizarse para asegurar que no se introdujeron nuevos errores.

3.3.4. Pruebas funcionales

Este tipo de prueba se realiza sobre el sistema funcionando, comprobando que cumpla con la especificación. Para estas pruebas, se utilizan las especificaciones de casos de prueba. Las pruebas basadas en requerimientos son pruebas de validación más que de defecto: se intenta demostrar que el sistema implementó adecuadamente sus requerimientos (Sommerville, 2011).

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 (Pressman, 2005). 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. Para la aplicación de pruebas de regresión sobre los casos de prueba definidos se usará la herramienta Selenium IDE, que permite grabar todas las interacciones de un usuario con el navegador y posibilita ejecutar de forma automática las mismas, reduciendo el tiempo y los costos de las pruebas funcionales.

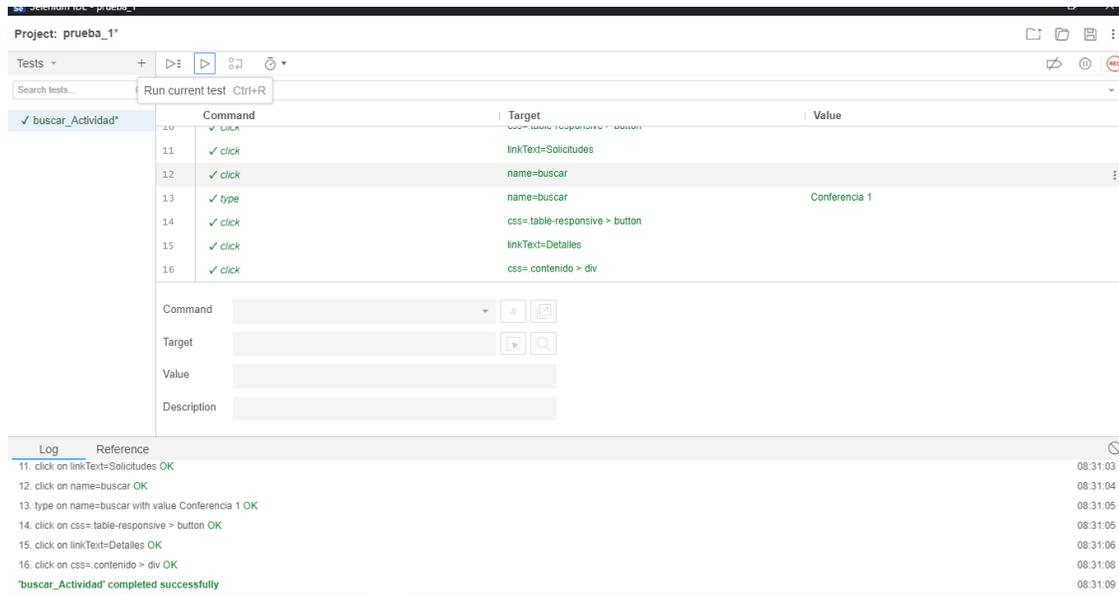


Figura 3.6. Representación del resultado la ejecución de una prueba usando Selenium IDE, Buscar Actividad.

A continuación, la tabla 3.6 muestra el diseño de caso de pruebas del requisito “Añadir participante” donde se analizarán las variables y condiciones que puedan determinar la respuesta del sistema.

Tabla 3.6. Caso de prueba “Añadir participante” (Fuente: Elaboración Propia).

Escenario	Descripción	Variables		Respuesta esperada	Respuesta
		Participante	Función		

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL MÓDULO DE INSTITUCIONES CULTURALES PARA EL SISTEMA GERENCIAL DE EXTENSIÓN UNIVERSITARIA

EC 1.1. Añadir participante correctamente	El usuario debe seleccionar los participantes y definir su función de ser necesario, para luego dar clic a en botón "Siguiente"	V art-hurjgf	V Activista	Se envían los datos al servidor, y se muestran debajo del subtítulo Participantes Vinculados aquellos participantes insertados en la actividad	El sistema añade al participante y muestra los participantes vinculados
EC 1.2. Añadir participante incorrectamente		I No se selecciona participante	V No se define función	Se presiona el botón para vincular un participante y el sistema no envía los datos	Al no seleccionar ningún participante el sistema no envía los datos al servidor

No.	Variable	Valor Nulo	Descripción
1	Participante	No	Es un campo checkbox que permite al usuario elegir de un listado de personas que cumplan ciertas características las más adecuadas para participar en la actividad cultural que se está definiendo o modificando
2	Función	Sí	Es un campo que permite al usuario insertar la función de un participante específico dentro de la actividad. Es un campo opcional y acepta cualquier carácter alfa numérico.

Tabla 3.7. Variables de caso de prueba "Añadir participante"(Fuente: Elaboración Propia).

Las pruebas de caja negra se aplicaron con el objetivo de evaluar las interfaces de comunicación con el usuario, las que demostraron coherencia y funcionalidad, así como probar todas aquellas funcionalidades directamente relacionadas con los requisitos funcionales del sistema. La técnica de partición de equivalencia es aplicada para evaluar los diferentes escenarios que pueden tener lugar ante la ejecución de una acción. Como resultado de la aplicación de estas pruebas se ejecutan las posibles variantes que posee una interfaz de comunicación con el usuario, resolviendo las no conformidades arrojadas y perfeccionando lo obtenido. 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 figura 3.7, donde se evidencia la cantidad de casos de prueba ejecutados, los casos de prueba con no conformidades. Se realizaron cuatro iteraciones, durante la primera iteración se analizaron 51 casos de prueba, de los cuales 35 resultaron no conformidades. En la segunda y tercera iteración a través de las pruebas de regresión, con el uso del software Selenium IDE, se verificó que las no conformidades anteriores estuviesen solucionadas, y de estas pruebas se obtuvieron 23 y 12 nuevas no conformidades respectivamente, quedando resuelta en la cuarta iteración y cumpliéndose correctamente los requisitos funcionales.

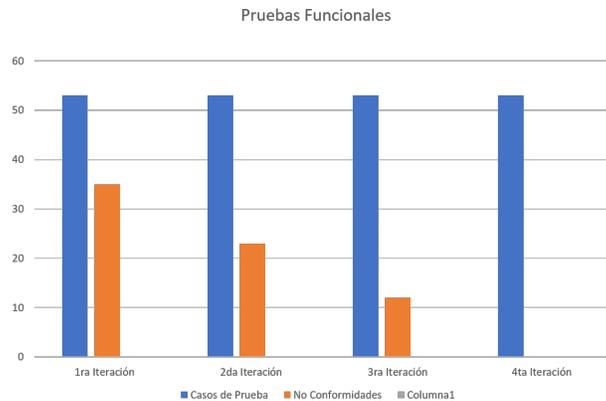


Figura 3.7. Representación del resultado de las pruebas.

3.3.5. Prueba de seguridad

Las pruebas de seguridad se diseñan para sondear las vulnerabilidades del entorno lado cliente, las comunicaciones de red que ocurren conforme los datos pasan de cliente a servidor y viceversa, y el entorno del lado servidor. Cada uno de estos dominios puede atacarse, y es tarea del examinador de seguridad descubrir las debilidades que puedan explotar quienes tengan intención de hacerlo (Pressman, 2005).

Las pruebas de seguridad se aplicaron con ayuda de la herramienta Acunetix Web Vulnerability Scanner 11.0.1 que establece alertas de tipo: alta, media, baja e informativa, realizándose en tres iteraciones durante el desarrollo de la propuesta solución.

En una primera iteración se obtuvo un total de 1 error de mensaje, con clasificación de nivel bajo.

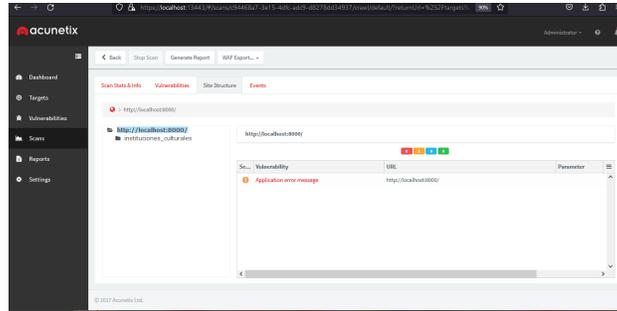


Figura 3.8. Prueba de seguridad 1ra iteración.

Después de aplicar refactorización del código y realizar las validaciones correspondientes, se aplicó la segunda iteración en búsqueda de vulnerabilidades al sistema, arrojando como resultado el mismo error, se procedió a realizar una tercera iteración donde persistió aún el error y luego del análisis correspondiente se identificó la relación del error con la forma de enviar los datos de otro módulo del sistema que está relacionado con el que se probaba.

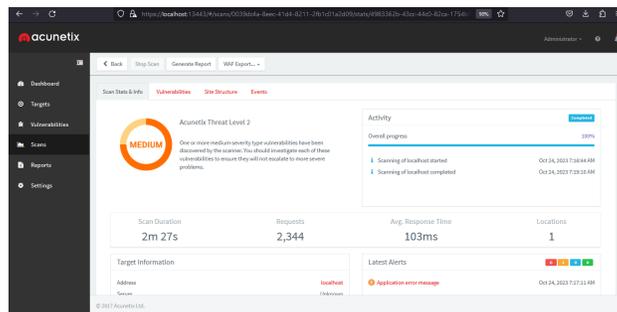


Figura 3.9. Prueba de seguridad 2da iteración.

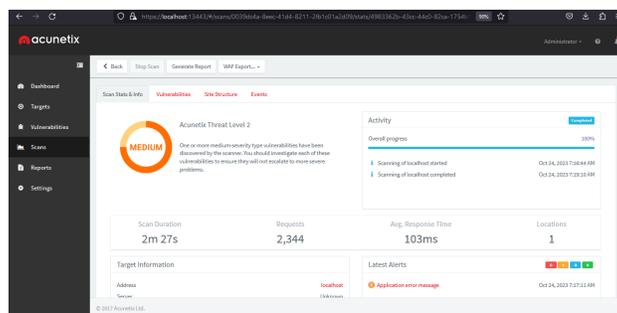


Figura 3.10. Prueba de seguridad 3ra iteración.

3.3.6. Prueba de rendimiento

Las pruebas de rendimiento deben diseñarse para garantizar que el sistema procese su carga pretendida. Esto implica efectuar una serie de pruebas donde se aumenta la carga, hasta que el rendimiento del sistema se

vuelve inaceptable. Las pruebas de rendimiento se preocupan tanto por demostrar que el sistema cumple con sus requerimientos, como por descubrir problemas y defectos en el sistema (Sommerville, 2011).

Para la realización de las pruebas de rendimiento del sistema se utilizó el paquete de python locust destinado para la ejecución de estas pruebas mediante código python. El cuál permitió probar la aplicación simulando un entorno similar al de producción o despliegue de la aplicación, donde actuaban de forma concurrente 100 usuarios, realizando alrededor de 5 peticiones por segundo, obteniendo un tiempo de respuesta máximo menor que cinco segundos, cumpliendo así con lo pactado con el cliente en los requisitos no funcionales del sistema.



Figura 3.11. Prueba de rendimiento con 100 usuarios concurrentes.

A fin de comprobar el comportamiento del sistema sobre una carga superior, hasta llevarlo al límite, se le aplicó la prueba de estrés, con la presencia de 200 usuarios concurrentes, conectándose de uno en uno por cada segundo, y se obtuvo como resultado que, el sistema superó la barrera de 5 segundos de respuesta a los 160 usuarios concurrentes y comenzó a presentar fallos en las respuestas a los clientes a partir de los 190 usuarios, llegando a un **2 por ciento** de fallos con respecto al total de peticiones con los 200 usuarios en el sistema.

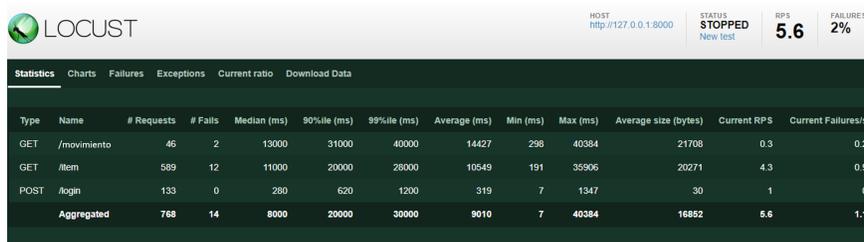


Figura 3.12. Prueba de rendimiento con 200 usuarios concurrentes.

3.3.7. Pruebas de aceptación

Para la revisión final de las especificaciones del diseño y de la implementación se realizaron las pruebas de aceptación con el objetivo de determinar diferentes tipos de errores y resolverlos (Pressman, 2005), para ello se emplean las pruebas Alfa y Beta.

- **Pruebas Alfa:** Se desarrollan en conjunto, el desarrollador y los usuarios finales. Con el objetivo de registrar los errores y problemas del uso del software.
- **Pruebas Beta:** El sistema fue desplegado en un servidor del centro cultural, con 10 clientes conectados en red local, para la realización de las pruebas. A diferencia de la prueba alfa, la prueba beta es una aplicación del software en su ambiente final. El objetivo es registrar todos los problemas que se encuentran durante la prueba.

Para la aplicación de las pruebas de aceptación se seleccionó un grupo de profesores y especialistas que han impartido durante varios años diversos cursos en DEU de la UCI. A continuación, se relacionan los involucrados en las pruebas:

- MsC. Julio César Espronceda Pérez (Profesor)
- Ing. Alvaro Alejandro Acosta Ruiz (Especialista B en Ciencias Informáticas)
- MsC. Mirta Beltrandez Sardiñas (Especialista B en Ciencias Informáticas)
- Lic. Yanaida Delgado Gonzáles (Especialista Superior)
- Ing. Madelaine Bárbara Labadié Martínez (Especialista Superior)
- MsC. Antonio Gutiérrez Lanorit (Vicerrector de Extensión)
- MsC. Yoisbel Tabares León (Profesor)
- Lic. Jesús Ramón Gonzales Crespo (Especialista Superior)
- MsC. Zuleika Remedio Frometa (Profesor)
- Ing. Allan Pierra Fuentes (Director de Soporte Tecnológico)

Como resultado después de haber aplicado las pruebas Alfa y Beta se identificaron nuevas no conformidades, las cuales fueron resueltas. Las pruebas Alfas aplicadas en un ambiente controlado, donde se probó por parte del equipo de desarrollo la funcionalidad del sistema, arrojaron errores en el cumplimiento de los requisitos no funcionales. Mientras que las pruebas Beta aplicadas en el ambiente final donde será desplegada la solución arrojaron errores en el proceso de búsqueda de una Actividad Cultural y el nivel de aprobación de estas.

3.4. Satisfacción de los usuarios

Con el objetivo de evaluar el sistema implementado se utiliza la técnica de Iadov, esta técnica evalúa el nivel de satisfacción del usuario, permitiendo conocer si la solución propuesta cumple con las expectativas esperadas. Esta técnica constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas (preguntas 1, 2 y 3) que se intercalan dentro de un cuestionario (Ver Anexo B.1) . Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de Iadov”, el cual se muestra a continuación en la tabla 3.8.

Tabla 3.8. Cuadro Lógico de Iadov (Fuente: Elaboración propia).

	1- ¿Consideras adecuado el uso de las Tecnologías de la Información y las Comunicaciones (TIC) en el proceso de informatización de la Gestión de Instituciones Culturales?								
	NO			NO SÉ			SÍ		
3- ¿Satisface sus necesidades como gestor Cultural, el sistema propuesto?	2- ¿Utilizaría usted la aplicación propuesta a la hora de gestionar las Instituciones Culturales UCI?								
	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	3	2	3	3	3	6	3	6
Me da lo mismo.	3	3	3	3	3	3	3	4	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

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción.

Escala de satisfacción:

- Clara satisfacción (1).
- Más satisfecho que insatisfecho (2).
- No definida (3).
- Más insatisfecho que satisfecho (4).
- Clara insatisfacción (5).
- Contradictoria (6).

Para medir el grado de satisfacción se tomó una muestra de 10 personas. Las personas que fueron seleccionadas tienen más de 5 años de experiencia en el trabajo de la extensión universitaria. De ellas 5 son Máster en Ciencias, 3 son Ingenieros y el resto son Licenciados.

La técnica de Iadov permite conocer el Índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y -1 de la siguiente forma:

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 utilizando la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

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 de haber aplicado la técnica, calculando el ISG se obtiene como resultado 0.8 lo que significa una buena satisfacción con el uso del módulo de Instituciones Culturales.

$$ISG = \frac{7(+1) + 2(+0,5) + 1(0) + 0(-0,5) + 0(-1)}{10} = 0.8$$

A partir de la realización de las pruebas y la aplicación de la técnica Iadov, se determinó que el sistema informático cumplió con los indicadores establecidos de funcionalidad, obteniendo el máximo calificativo en todos los aspectos, además se validó que el proceso de gestión de la información se garantiza en sus tres dimensiones, de organización de la información excelente, centralización, mediante un diseño de una base de datos robusta y normalizada, sin duplicación de la información, así como la disponibilidad de la misma, con los tiempos de respuesta esperados, respaldados por las pruebas de rendimiento. Corroborando la hipótesis que dio inicio a la investigación.

3.5. Consideraciones parciales

En este capítulo se abordaron los elementos del módulo de Instituciones Culturales para el SGEU, así como las pruebas realizadas al mismo y los resultados obtenidos; lo que permite concluir:

- La implementación de la propuesta de solución facilitó obtener una aplicación funcional capaz de contribuir con la gestión de las Instituciones Culturales UCI

- La validación de la propuesta de solución, mediante una estrategia de pruebas de software permitió corroborar la calidad de la misma, estableciendo así el cumplimiento de los requisitos definidos por el cliente.
- La aplicación de la técnica de Iadov arrojó como resultado un alto índice de satisfacción de la solución propuesta por parte de un grupo de profesores y especialistas de la DEU en la UCI, lo cual se traduce en una clara satisfacción con el uso del sistema informático propuesto.

El presente Trabajo de Diploma concluye con el desarrollo de un Sistema para la Gestión del Proceso de Instituciones Culturales de la UCI.

- El análisis y la fundamentación teórica de los principales conceptos asociados al subproceso de gestión de Instituciones Culturales, permitió comprender el alcance y relevancia de la investigación.
- El estudio y análisis de la gestión del subproceso de Instituciones Culturales dentro de la UCI, permitió conocer en que estado se encontraba, y corroborar la necesidad de la creación de una solución que contribuyera a la ejecución de dicho proceso.
- El diseño del módulo para la gestión del proceso de Instituciones Culturales, sentó las bases para la implementación del mismo, lo cual no hubiese sido posible sin el análisis que se realizó de soluciones similares que permitieron detectar más requisitos al sistema, en conjunto con los extraídos durante las entrevistas con los clientes.
- Con la implementación del módulo, se creó un producto de software, sobre la base de la tecnología web, que permitió contribuir a la gestión de la información asociada al proceso de Instituciones Culturales.
- La definición de una estrategia de pruebas, permitió comprobar el correcto funcionamiento del módulo para la Gestión del Proceso de Instituciones Culturales, a partir del cumplimiento de los requisitos pactados con el cliente.

Recomendaciones

Para dar continuidad a la presente investigación se recomienda:

- Corregir el método de envío de datos entre el módulo Gerencia y resto de módulos asociados, incluido el de Instituciones culturales
- Incorporar al módulo de instituciones culturales nuevas funcionalidades administrativas relacionadas con el control de presupuesto, control de capital humano y control de medios básicos.

- AE** Actividades Extensionistas. 8, 20
- AUP-UCI** Proceso Unificado Ágil versión UCI. 13, 24, 25
- CASE** Ingeniería de Software Asistida por Computadora. 13
- CC** Complejidad Ciclomática. 41, 42
- CSS** Hoja de estilo en Cascada. 15, 18, 28
- CSS3** Hoja de estilo en Cascada Versión 3. 16, 24
- DEU** Dirección de Extensión Universitaria. 20, 49, 52
- DRY** No te repitas. 16
- EU** Extensión Universitaria. 2, 6–8, 13, 20
- GI** Gestión de la información. 9
- GOF** Pandilla de cuatro. 31
- GRASP** Patrones Generales de Software de Asignación de Responsabilidad. 30
- HTML** Leguaje de marcado de hipertexto. 15, 16
- HTML5** Leguaje de marcado de hipertexto Versión 5. 15, 18, 24
- HU** Historias de Usuario. 25, 33
- IDE** Entorno de Desarrollo Integrado. 16, 19
- IES** Instituciones de Educación Superior. 4, 6, 7, 59
- ISG** Índice de satisfacción grupal. 50, 51
- MTV** Modelo Vista Plantilla. 16, 27, 28
- MVC** Modelo Vista Controlador. 12, 16, 27, 28
- ORM** Mapeador de Objeto Relacional. 17
- PBS** Public Broadcasting System. 16
- PHP** Preprocesador de Hipertexto. 11, 12
- PNEU** Programa Nacional de Extensión Universitaria. 7
- POO** Programación Orientada a Objeto. 14, 31

- SGBD** Sistema Gestor de Base de Datos. 17
- SGEU** Sistema Gerencial de Extensión Universitaria. 6–19
- SGML** El Lenguaje de marcado generalizado estándar. 15
- SQL** Lenguaje de Consultas Estructurado. 17
-
- TDD** Desarrollo dirigido por pruebas. 39
- TIC** Tecnologías de la Información y las Comunicaciones. 50, 74
- TICs** Tecnologías de la Información y las Comunicaciones. 1, 2
- TSU** Trabajo Sociocultural Universitario. 7
-
- UCI** Universidad de las Ciencias Informáticas. 2–6, 12, 13, 20, 33, 35–53, 74
- UI** Interfaz de Usuario. 23, 25, 26
- UML** Lenguaje de Modelado Unificado. 14, 24
-
- W3C** El Consorcio de la Red Informática Mundial. 15
- WWW** Red Informática Mundial. 15

Referencias bibliográficas

- APORTELA, Odette González, 2016. *SISTEMA DE GESTIÓN DE LA CALIDAD DEL PROCESO DE EXTENSIÓN UNIVERSITARIA EN LA UNIVERSIDAD DE LA HABANA*. Tesis doctoral. Universidad de La Habana (vid. pág. 7).
- APORTELA, Odette González; FERNÁNDEZ-LARREA, Mercedes González; LOOR, Tania Miladi Zambrano; CARIDAD BALSINDE HERRERA, José de la y MAINEGRA, Amado Batista, 2022. RETOS EN LA GESTIÓN DEL PROCESO EXTENSIONISTA CUBANO: SISTEMA DE INFORMACIÓN GERENCIAL RETOS EN LA GESTIÓN DEL PROCESO EXTENSIONISTA CUBANO: SISTEMA DE INFORMACIÓN GERENCIAL (vid. pág. 8).
- BALOGUN, MO, 2022. Comparative Analysis of Complexity of C++ and Python Programming Languages. *Asian J. Soc. Sci. Manag. Technol.* Vol. 4, págs. 1-12 (vid. pág. 14).
- BODAÑO GARCÍA, Jorge Antonio, 2022. La informatización de la sociedad como política del estado cubano. *Opuntia Brava*. Vol. 14, n.º 4 (vid. pág. 1).
- C MARTÍNEZ-TENA, Alicia de la; EXPÓSITO-GARCÍA, Elpidio; FERRER-BALART, Anisley y ARAGÓN-MARTÍNEZ, Nelson Amey, 2023. Revisitar a la gestión cultural desde los programas de desarrollo cultural. Una reflexión necesaria. *Santiago*, págs. 332-348 (vid. pág. 12).
- CABEZA MATOS, Kirenia; ANIAS SANTOS, Javier; CRUZ AMARÁN, Damaris; LLANO CASTRO, Eileén y ALONSO BEATÓN, Yurisleis, 2015. *Solución para la gestión de información de los procesos de Extensión Universitaria en el área de Extensión Cultural de la Universidad de las Ciencias Informáticas*. B.S. thesis. Universidad de las Ciencias Informáticas (vid. pág. 12).
- CEDEÑO, Raiza Peña y RIVERA, Eliezer Colón, 2021. Etapas de desarrollo de la dirección del proceso de extensión universitaria: El caso de la Universidad de la Isla de la Juventud en Cuba (vid. pág. 8).
- CHACON, Scott y STRAUB, Ben, 2014. *Pro git*. Springer Nature (vid. pág. 18).
- CONSORTIUM, World Wide Web et al., 1999. HTML 4.01 specification (vid. pág. 16).
- ESTRADA MOLINA, Odiel, 2015. *La Extensión Universitaria y su impacto desde la Carrera | Universidad de las Ciencias Informáticas* [online] [visitado 2022-04-12]. Disponible desde: <https://www.uci.cu/la-extension-universitaria-y-su-impacto-desde-la-carrera> (vid. pág. 6).

- GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; JOHNSON, Ralph E; VLISSIDES, John et al., 1995. *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH (vid. pág. 30).
- GONZÁLEZ REYES, Alena y FEBLES ESTRADA, Ailyn, 2021. Principales áreas de impacto y resultados de la informatización de la sociedad cubana (vid. pág. 1).
- GONZÁLEZ-APORTELA, Odette; BATISTA-MAINEGRA, Amado y FERNÁNDEZ-LARREA, Mercedes González, 2020. Sistema de gestión de la calidad del proceso de extensión universitaria, una experiencia en la Universidad de La Habana. *Revista Electrónica Calidad en la Educación Superior*. Vol. 11, n.º 2, págs. 105-134 (vid. pág. 10).
- GONZÁLEZ-HERNÁNDEZ, Esnel; ISAZA-GÓMEZ, Germán Darío; MIRANDA-CALDERÓN, Katherine y MOSQUERA-VENTE, Andrés Mauricio, 2022. Panorama de la formación de posgrado para el área educación física, deporte, actividad física, recreación y afines en Colombia. *Revista Digital: Actividad Física y Deporte*. Vol. 8, n.º 2 (vid. pág. 11).
- GORE, Himanshu; SINGH, Rakesh Kumar; SINGH, Ashutosh; SINGH, Arnav Pratap; SHABAZ, Mohammad; SINGH, Bhupesh Kumar y JAGOTA, Vishal, 2021. Django: Web Development Simple & Fast. *Annals of the Romanian Society for Cell Biology*. Vol. 25, n.º 6, págs. 4576-4585 (vid. pág. 16).
- JACOB THORNTON, Mark Otto, 2022. *Get started with Bootstrap* [online] [visitado 2022-06-27]. Disponible desde: <https://getbootstrap.com/docs/5.2/getting-started/introduction/> (vid. pág. 18).
- KOÇ, Hatice; ERDOĞAN, Ali Mert; BARJAKLY, Yousef y PEKER, Serhat, 2021. UML diagrams in software engineering research: a systematic literature review. *Multidisciplinary Digital Publishing Institute Proceedings*. Vol. 74, n.º 1, págs. 13 (vid. págs. 14, 28).
- LLERENA IZQUIERDO, Joe, 2020. *Codifica en Python* (vid. pág. 35).
- LUIJÁN MORA, Sergio, 2016. *Programación de aplicaciones web. Historia. Principios básicos y clientes web*. Luján Mora, Sergio (vid. pág. 15).
- MAIDA, Esteban Gabriel y PACIENZIA, Julián, 2015. Metodologías de desarrollo de software (vid. pág. 13).
- MALQUI, Diego David, 2018. Diseño y desarrollo Web accesible utilizando HTML5 y CSS3 con nivel de conformidad A, respecto a las pautas WCAG (vid. pág. 16).
- MÉNDEZ PÉREZ, Juan Albino; SANTANA PÉREZ, Alberto Javier; GONZÁLEZ DE LA ROSA, Manuel; ALBELO JORGE, Cynthia; TORRES PADRÓN, Laura; MENDOZA JIMÉNEZ, Javier; DORTA AFONSO, Daniel; ACOSTA RUBIO, Zamira del Rosario et al., 2022. Diagnóstico de Responsabilidad Social de la Universidad de La Laguna (vid. pág. 11).
- MES, 2004. *Programa Nacional de Extensión Universitaria* (vid. pág. 2).
- MICROSOFT, 2022. *Get Started with Visual Studio Code* [online] [visitado 2022-06-08]. Disponible desde: <https://code.visualstudio.com/learn/overview> (vid. pág. 16).

- MILLER, Curtis G, 2022. Introduction to Git (vid. pág. 18).
- PADÍN, Rocío Rodríguez, 2023. *Aprendizaje cooperativo a través de las TIC*. Aula Magna Proyecto clave McGraw Hill (vid. pág. 1).
- PARADIGM, Visual, 2022. *About Visual Paradigm* [online] [visitado 2022-06-24]. Disponible desde: <https://www.visual-paradigm.com/aboutus/> (vid. pág. 13).
- PÉREZ, Javier Eguíluz, 2019. *introduccion a JavaScript* (vid. pág. 15).
- PILICITA GARRIDO, Anabel; BORJA LÓPEZ, Yolanda y GUTIÉRREZ CONSTANTE, Gonzalo, 2021. Rendimiento de MariaDB y PostgreSQL (vid. pág. 17).
- PIÑEIRO VIDAL, Elizabeth y ROA PARETS, Reinier, 2020. Módulo para el control del personal civil del Sistema de gestión de la información y control estadístico del Ministerio del Interior (vid. págs. 30, 31).
- PRADEEP, S y SHARMA, Yogesh Kumar, 2019. A pragmatic evaluation of stress and performance testing technologies for web based applications. En: *A pragmatic evaluation of stress and performance testing technologies for web based applications. 2019 Amity International Conference on Artificial Intelligence (AICAI)*, págs. 399-403 (vid. pág. 19).
- PRESSMAN, Roger S, 2005. *Software engineering: a practitioner's approach*. Palgrave macmillan (vid. págs. 30, 31, 46, 49, 51).
- PUIG, Jordi Collell, 2013. CSS3 y Javascript avanzado. *Diponible en <https://openlibra.com/es/book/download/css3-y-javascript-avanzado>* (vid. pág. 16).
- REYNOSO, Carlos y KICILLOF, Nicolás, 2019. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft (vid. pág. 27).
- SALAZAR, Juan Diego Ramírez, 2023. Administración de proyectos y sistemas de gestión de la innovación, Raíz Capital/Naranja Republik (vid. pág. 11).
- SELENIUM, IDE; DEBUGGING, Selenium IDE Script y POINTS, Inserting Verification, 2008. *SELENIUM-IDE* (vid. pág. 19).
- SOLÓRZANO ÁVILA, Juan Alfredo, 2018. Desarrollo de una aplicación web multiplataforma usando el framework Django, para publicitar eventos sociales, aplicado en el municipio del cantón Morona. (Vid. pág. 28).
- SOMMERVILLE, Ian, 2011. *Ingeniería de software* [online] [visitado 2022-06-18]. ISBN 978-607-32-0604-4 978-607-32-0603-7. Disponible desde: http://www.ingebook.com/ib/NPcd/IB_Escritorio_Visualizar?cod_primaria=1000193&libro=1518. OCLC: 1257013649 (vid. págs. 27, 38, 39, 41, 46, 51).
- SRINATH, KR, 2017. Python—the fastest growing programming language. *International Research Journal of Engineering and Technology (IRJET)*. Vol. 4, n.º 12, págs. 354-357 (vid. pág. 14).

- VEGA MEDEROS, Juan Francisco, 2017. La extensión universitaria: función integradora del quehacer universitario en el presente siglo (vid. pág. 6).
- VENTURA BAUTISTA, Jesús Natividad, 2021. JQUERY, AJAX Definición de jQuery, programar JavaScript con jQuery. Selección de un elemento del documento mediante el id. Selección de elementos por el tipo de elementos. Método text (), text (valor). Métodos attr (nombre de propiedad), attr (nombre de propiedad, valor) y removeAttr (nombre de propiedad). Métodos addClass y removeClass. Método html () y html (valor). Administración de eventos con jQuery. Eventos mouseover y mouseout. Tipos de eventos, efectos, tipos efectos, iteración, Ajax, funciones anónimas, ejemplos y aplicaciones (vid. pág. 18).
- WIRFS-BROCK, Allen y EICH, Brendan, 2020. JavaScript: the first 20 years. *Proceedings of the ACM on Programming Languages*. Vol. 4, n.º HOPL, págs. 1-189 (vid. pág. 15).

Apéndice

Historias de Usuario

A.1. Historias de Usuario

Tabla A.1. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Adicionar Invitado
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe ser capaz de adicionar un invitado a una actividad.	
Observaciones: Para este proceso es necesario estar autenticado con el rol de responsable de actividad.	
Interfaz: 	

Tabla A.2. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Buscar Actividad
Usuario: Responsable de Institución	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe ser capaz de buscar una actividad determinada a partir de su nombre	
Observaciones: El proceso es visible para cualquier usuario autenticado con el rol de responsable de actividad.	
Interfaz:	
	

Tabla A.3. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Crear Comentario
Usuario: Usuario común	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir realizar un comentario sobre una actividad.	
Observaciones: El proceso será visible para cualquier usuario.	

Continúa en la próxima página

Tabla A.3. Continuación de la página anterior

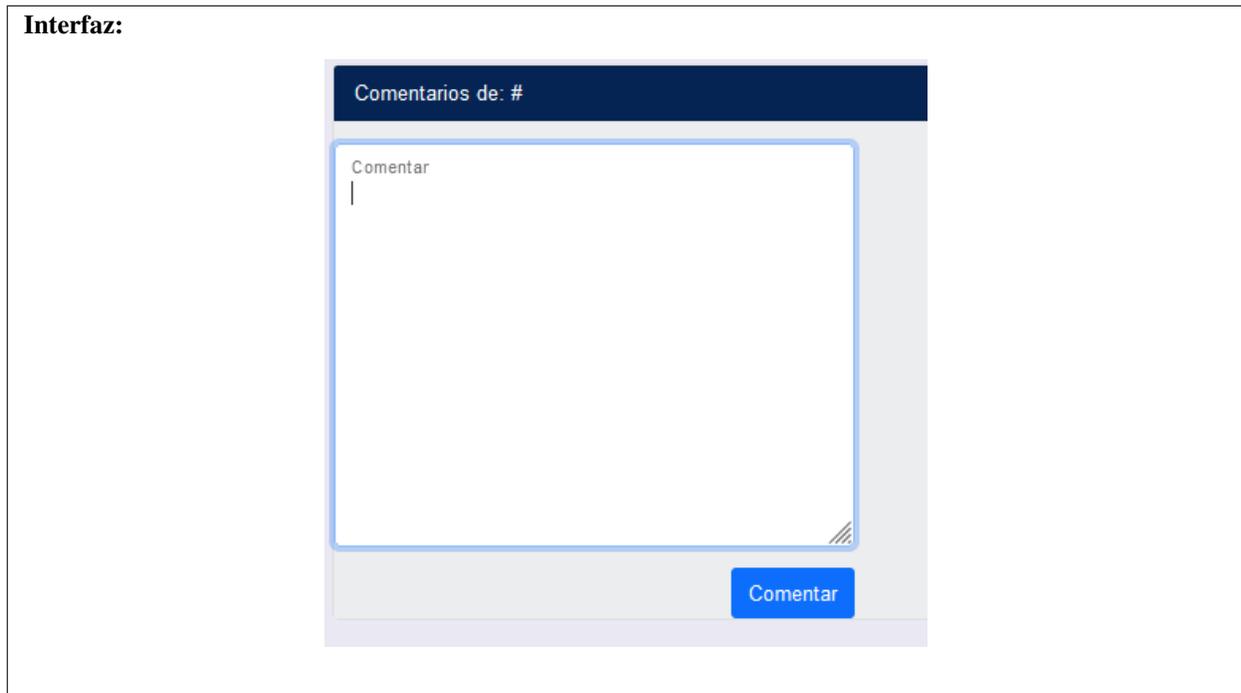


Tabla A.4. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Solicitar Actividad
Usuario: Usuario común	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir solicitar una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema.	

Continúa en la próxima página

Tabla A.4. Continuación de la página anterior

Interfaz:

The image shows a screenshot of a web form titled "Registrar Actividad". The form is set against a light blue background and contains the following fields and controls:

- Nombre:** A text input field.
- Tema:** A dropdown menu.
- Obra a promover:** A text input field.
- Lugar:** A dropdown menu.
- Público Meta:** A dropdown menu.
- Área:** A dropdown menu.
- Fecha de inicio:** A text input field.
- Fecha fin:** A text input field.
- Hora de inicio:** A text input field.
- Hora fin:** A text input field.
- Objetivo:** A text input field.
- Cantidad de personas:** A text input field.
- Aceptar:** A button located at the bottom center of the form.

Tabla A.5. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Mostrar Detalles de Actividad
Usuario: Usuario común	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe mostrar los detalles de una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema	

Continúa en la próxima página

Tabla A.5. Continuación de la página anterior

Interfaz:

Conferencia

Detalles de actividad: Conferencia

Responsable: (joseatp)
Solicitante: (joseatp)
Área: Area 3
Lugar: Centro Cultural Wifredo Lam
Tipo: Conferencia
Dias: Miércoles
Estado: Confirmado por el Responsable
Capacidades 50
Valoración: 5
Objetivo:
 Fomentar el deporte

Recursos Participantes

[Asientos](#) [Sugerencias](#) [Modificar](#) Valorar ☆☆☆☆☆

Activa desde el 6 de diciembre de 2023 a las 19:00 a 7 de diciembre de 2023 a las 20:00

Tabla A.6. Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Insertar Evidencia
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe insertar evidencia de una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución	
Interfaz:	
<p>Evidencias de actividad: Nueva actividad</p> <p>Evidencias</p>	

Tabla A.7. Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Recursos de Actividad
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe listar los recursos necesarios para la realización de una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de Institución.	
Interfaz:	
	

Tabla A.8. Historia de usuario # 11

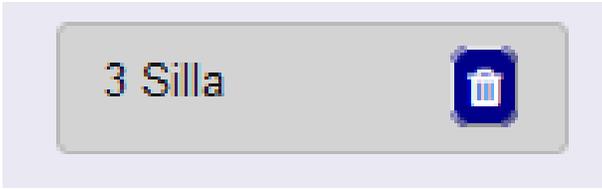
Historia de usuario	
Número: 11	Nombre: Desvincular Recurso
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe desvincular los recursos de una actividad seleccionada.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución	
Interfaz:	
	

Tabla A.9. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Añadir Participantes
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir insertar un participante a una actividad seleccionada.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de actividad.	
Interfaz:	
	

Tabla A.10. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Insertar Sugerencia
Usuario: Responsable de Institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir insertar una sugerencia a una actividad seleccionada.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución	

Continúa en la próxima página

Tabla A.10. Continuación de la página anterior

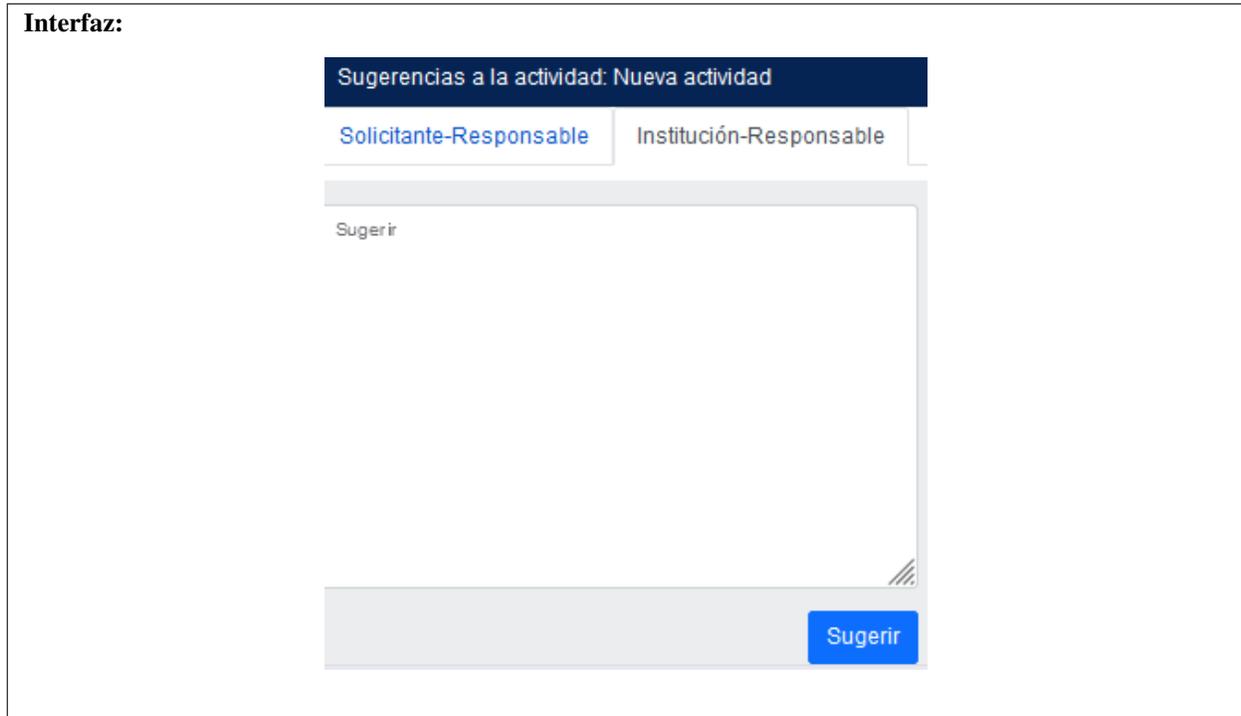


Tabla A.11. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Listar Invitado por actividad
Usuario: Responsable de institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir listar los invitados de una actividad seleccionada.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con el rol de responsable de institución.	

Continúa en la próxima página

Tabla A.11. Continuación de la página anterior

Interfaz:

Participantes

Nombre	Función	Estado	Entrada
Participantes			
r2d2		Invitado por responsable	28 de octubre de 2023
Invitados			
Jose Adrian	..	Aprobada	28 de octubre de 2023

Tabla A.12. Historia de usuario # 15

Historia de usuario	
Número: 15	Nombre: Listar espacios disponibles por local
Usuario: Responsable de local	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir listar los espacios disponibles por locales.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de local.	
Interfaz:	
	

Tabla A.13. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Listar participantes por actividad
Usuario: Responsable de institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1

Continúa en la próxima página

Tabla A.13. Continuación de la página anterior

Programador responsable: Arthur Javier Gargallo Fleitas
Descripción: El sistema debe permitir listar los Participantes de una actividad determinada.
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución.
<p>Interfaz:</p> 

Tabla A.14. Historia de usuario # 17

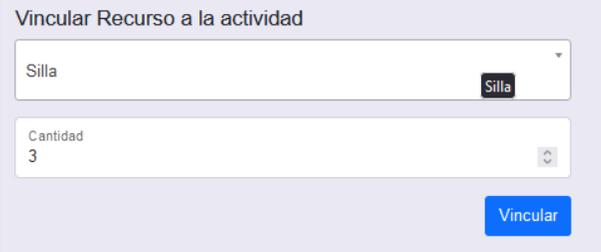
Historia de usuario	
Número: 17	Nombre: Asignar Recursos
Usuario: Responsable de Institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir vincular recursos a una actividad seleccionada.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución.	
<p>Interfaz:</p> 	

Tabla A.15. Historia de usuario # 18

Historia de usuario	
Número: 18	Nombre: Asignar Espacio
Usuario: Responsable de local.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media

Continúa en la próxima página

Tabla A.15. Continuación de la página anterior

Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir aprobar la solicitud de un local para la realización de determinada actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con el rol de responsable de local.	
Interfaz:	
	

Tabla A.16. Historia de usuario # 19

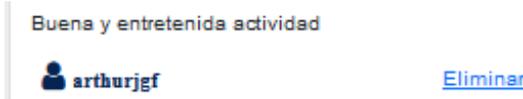
Historia de usuario	
Número: 19	Nombre: Eliminar comentario.
Usuario: Responsable de Institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir la eliminación de un comentario seleccionado.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución.	
Interfaz:	
	

Tabla A.17. Historia de usuario # 20

Historia de usuario	
Número: 20	Nombre: Desvincular local.
Usuario: Responsable de Local.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir liberar un local seleccionado.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de local.	

Continúa en la próxima página

Tabla A.17. Continuación de la página anterior

Interfaz:

Nombre de Local	Actividad	Fecha inicio	Fecha fin	Hora de inicio	Hora de finalización	Reservar	Liberar
Aula de Música	Act 1	2023-10-30	2023-10-31	02:00:00	02:00:00	Reservar	Liberar
Galería de Arte	Act2	2023-10-30	2023-10-31	03:01:00	03:01:00	Reservar	Liberar

Tabla A.18. Historia de usuario # 21

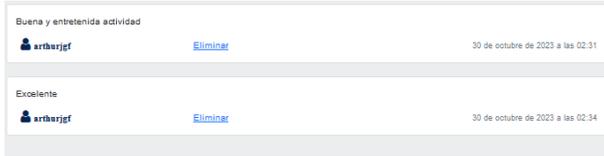
Historia de usuario	
Número: 21	Nombre: Listar comentarios.
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo Fleitas	
Descripción: El sistema debe permitir listar los comentarios de un actividad seleccionada.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con rol de responsable de institución.	
Interfaz:	
	

Tabla A.19. Historia de usuario # 22

Historia de usuario	
Número: 22	Nombre: Listar solicitud de local.
Usuario: Responsable de Local	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir listar las solicitudes de un local seleccionado.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con Rol de responsable de local.	
Interfaz:	
	

Tabla A.20. Historia de usuario # 23

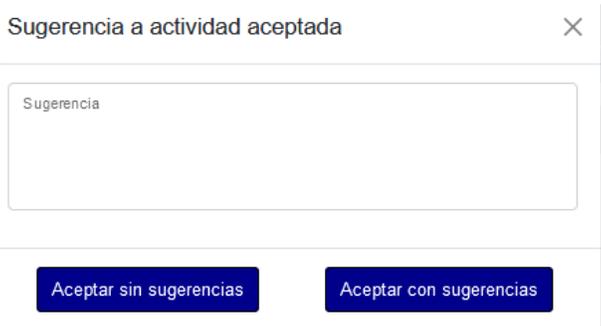
Historia de usuario	
Número: 23	Nombre: Aceptar solicitud con sugerencias.
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir aceptar la solicitudes de actividades y plantear sugerencias sobre estas.	
Observaciones: Este proceso será visible para cualquier usuario autenticado dentro del sistema con Rol de responsable de institución.	
Interfaz: 	

Tabla A.21. Historia de usuario # 24

Historia de usuario	
Número: 24	Nombre: Buscar participantes
Usuario: Responsable de actividad, Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo	
Descripción: El sistema debe permitir buscar un participante a partir del carnet de identidad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz: 	

Tabla A.22. Historia de usuario # 25

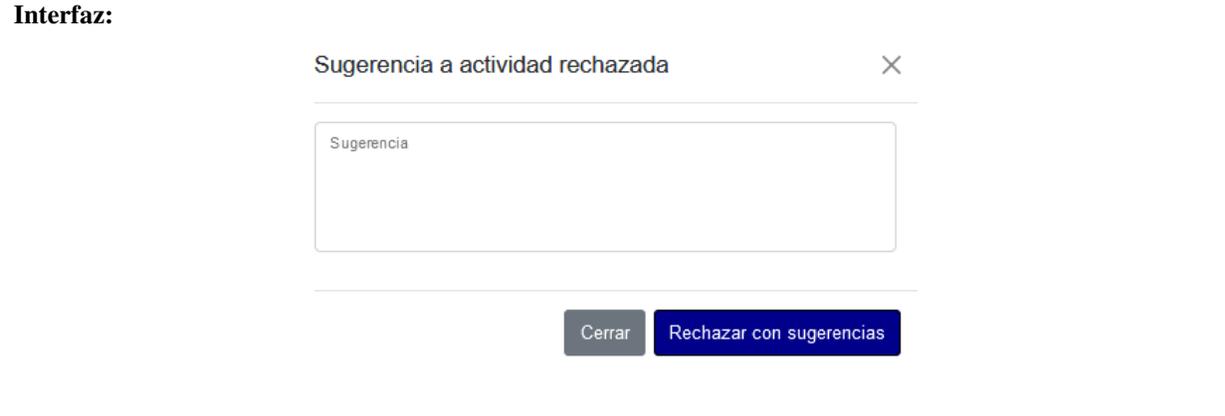
Historia de usuario	
Número: 25	Nombre: Rechazar solicitud.
Usuario: Responsable de institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo	
Descripción: El sistema debe permitir rechazar una solicitud para la realización de una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz: 	

Tabla A.23. Historia de usuario # 26

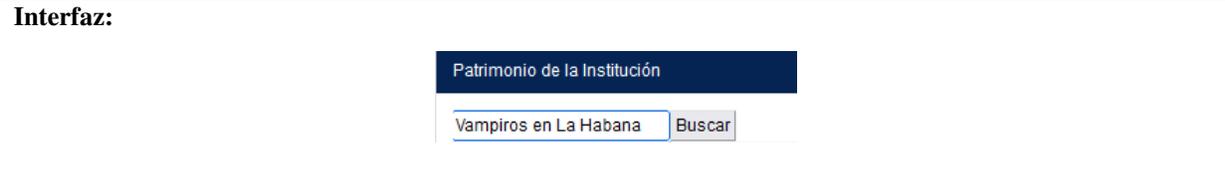
Historia de usuario	
Número: 26	Nombre: Buscar patrimonio cultural
Usuario: Responsable de institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir buscar el patrimonio cultural.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz: 	

Tabla A.24. Historia de usuario # 27

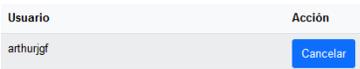
Historia de usuario	
Número: 27	Nombre: Cancelar asiento reservado
Usuario: Usuario común	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir cancelar un asiento reservado.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	
Interfaz:	
	

Tabla A.25. Historia de usuario # 28

Historia de usuario	
Número: 28	Nombre: Mostrar detalles del comentario
Usuario: Responsable de institución	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo	
Descripción: El sistema debe mostrar los detalles de un comentario.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz:	
	

Tabla A.26. Historia de usuario # 29

Historia de usuario	
Número: 29	Nombre: Eliminar patrimonio cultural
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1

Continúa en la próxima página

Tabla A.26. Continuación de la página anterior

Programador responsable: José Adrian Torres Peraza
Descripción: El sistema debe permitir eliminar el patrimonio cultural.
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.
Interfaz:

Nombre	Autor	Fecha de Creación	Fecha de Admisión	Acciones
Vampiros en La Habana	Juan Padrón	4 de diciembre de 2023 a las 00:00	12 de diciembre de 2023 a las 00:00	x 🔗

Tabla A.27. Historia de usuario # 30

Historia de usuario	
Número: 30	Nombre: Insertar patrimonio cultural
Usuario: Responsable de Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir insertar patrimonio cultural a una institución.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz:	

Tabla A.28. Historia de usuario # 31

Historia de usuario	
Número: 31	Nombre: Listar instituciones

Continúa en la próxima página

Tabla A.28. Continuación de la página anterior

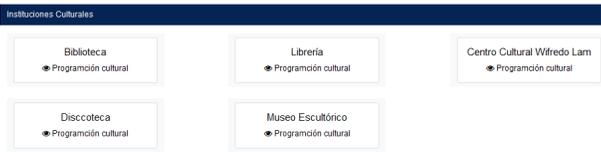
Usuario: Usuario común	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza	
Descripción: El sistema debe permitir listar las instituciones.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	
Interfaz: 	

Tabla A.29. Historia de usuario # 32

Historia de usuario	
Número: 32	Nombre: Mostrar programación cultural semanal.
Usuario: Usuario común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo	
Descripción: El sistema debe permitir listar la programación semanal.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	
Interfaz: 	

Tabla A.30. Historia de usuario # 33

Historia de usuario	
Número: 33	Nombre: Mostrar programación cultural mensual.
Usuario: Usuario de común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1

Continúa en la próxima página

Tabla A.30. Continuación de la página anterior

Programador responsable: Arthur Javier Gargallo
Descripción: El sistema debe permitir listar la programación mensual.
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.
Interfaz:



Tabla A.31. Historia de usuario # 34

Historia de usuario	
Número: 34	Nombre: Mostrar programación cultural anual.
Usuario: Usuario de común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo	
Descripción: El sistema debe permitir listar la programación anual.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	
Interfaz:	

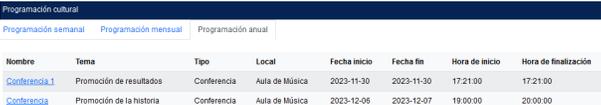


Tabla A.32. Historia de usuario # 35

Historia de usuario	
Número: 35	Nombre: Listar patrimonio cultural.
Usuario: Responsable de institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza.	
Descripción: El sistema debe permitir listar el patrimonio cultural.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	

Continúa en la próxima página

Tabla A.32. Continuación de la página anterior

Interfaz:



Tabla A.33. Historia de usuario # 36

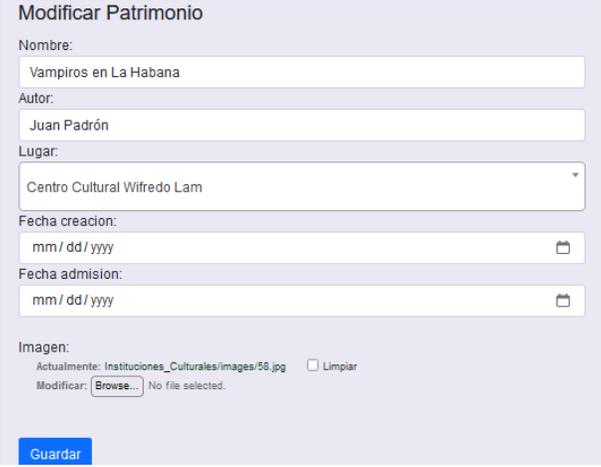
Historia de usuario	
Número: 36	Nombre: Modificar patrimonio cultural.
Usuario: Responsable de institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza.	
Descripción: El sistema debe permitir modificar el patrimonio cultural.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz:	
	

Tabla A.34. Historia de usuario # 37

Historia de usuario	
Número: 37	Nombre: Generar informe de actividad por área.
Usuario: Usuario común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1

Continúa en la próxima página

Tabla A.34. Continuación de la página anterior

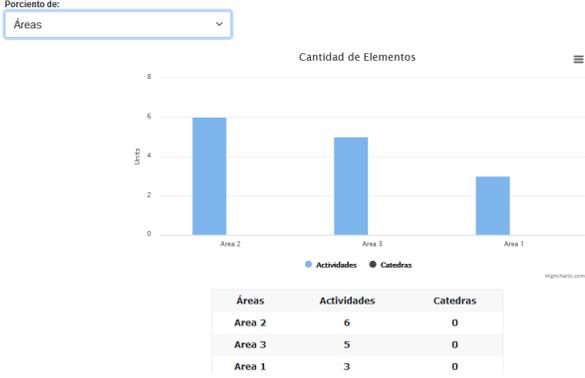
Programador responsable: Arthur Javier Gargallo.												
Descripción: El sistema debe permitir generar reportes de las actividades por áreas.												
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.												
<p>Interfaz:</p>  <p>The screenshot displays a web interface for generating reports. At the top, there is a dropdown menu labeled 'Por ciento de:' with 'Áreas' selected. Below it is a bar chart titled 'Cantidad de Elementos' showing the number of 'Actividades' (blue bars) and 'Catedras' (grey bars) for three areas: Area 2, Area 3, and Area 1. The y-axis is labeled 'Elementos' and ranges from 0 to 8. The data is as follows:</p> <table border="1"> <thead> <tr> <th>Áreas</th> <th>Actividades</th> <th>Catedras</th> </tr> </thead> <tbody> <tr> <td>Area 2</td> <td>6</td> <td>0</td> </tr> <tr> <td>Area 3</td> <td>5</td> <td>0</td> </tr> <tr> <td>Area 1</td> <td>3</td> <td>0</td> </tr> </tbody> </table>	Áreas	Actividades	Catedras	Area 2	6	0	Area 3	5	0	Area 1	3	0
Áreas	Actividades	Catedras										
Area 2	6	0										
Area 3	5	0										
Area 1	3	0										

Tabla A.35. Historia de usuario # 38

Historia de usuario	
Número: 38	Nombre: Generar informe de actividad por tipo.
Usuario: Usuario común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir generar reportes de las actividades por tipo.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	

Continúa en la próxima página

Tabla A.35. Continuación de la página anterior



Tabla A.36. Historia de usuario # 39

Historia de usuario	
Número: 39	Nombre: Generar informe de actividad por tema.
Usuario: Usuario común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir generar reportes de las actividades por tema.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	
Interfaz:	

Temas	Actividades
Medio Ambiente	5
Historia de la profesión	4
Promoción de la historia	3
Promoción de resultados	2
Salud	0

Tabla A.37. Historia de usuario # 40

Historia de usuario	
Número: 40	Nombre: Valorar actividad.
Usuario: Usuario común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir realizarle una valoración a una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de usuario común.	
Interfaz:	
	

Tabla A.38. Historia de usuario # 41

Historia de usuario	
Número: 41	Nombre: Listar solicitudes.
Usuario: Responsable de institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir listar las solicitudes para la realización de una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz:	
	

Tabla A.39. Historia de usuario # 42

Historia de usuario	
Número: 42	Nombre: Aceptar solicitud con sugerencia.
Usuario: Responsable de institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja

Continúa en la próxima página

Tabla A.39. Continuación de la página anterior

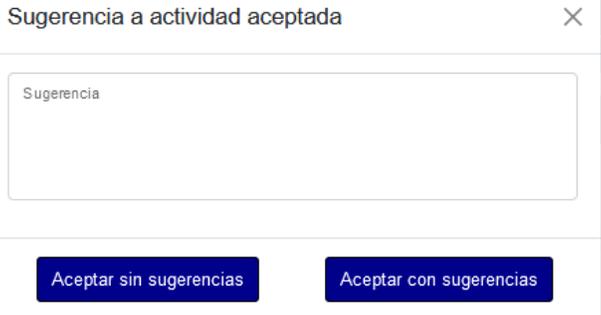
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir aceptar una solicitud con una sugerencia.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz: 	

Tabla A.40. Historia de usuario # 43

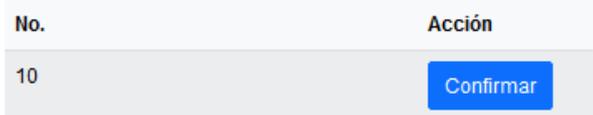
Historia de usuario	
Número: 43	Nombre: Reservar asiento.
Usuario: Usuario común.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir reservar un asiento para asistir a una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol usuario común.	
Interfaz: 	

Tabla A.41. Historia de usuario # 44

Historia de usuario	
Número: 44	Nombre: Listar asientos disponibles.
Usuario: Responsable de institución.	

Continúa en la próxima página

Tabla A.41. Continuación de la página anterior

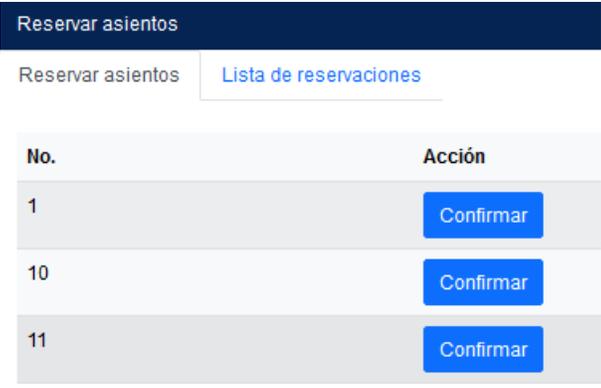
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza.	
Descripción: El sistema debe listar los asientos disponibles.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz: 	

Tabla A.42. Historia de usuario # 45

Historia de usuario	
Número: 45	Nombre: Eliminar participante.
Usuario: Responsable de institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza.	
Descripción: El sistema debe permitir la eliminación de un participante.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz: 	

Tabla A.43. Historia de usuario # 46

Historia de usuario	
Número: 46	Nombre: Ver detalle de invitado.
Usuario: Responsable de institución.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir ver los detalles de los invitados.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz:	
	

Tabla A.44. Historia de usuario # 47

Historia de usuario	
Número: 47	Nombre: Ver detalle de participante.
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir ver los detalles de los participantes.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	

Continúa en la próxima página

Tabla A.44. Continuación de la página anterior

Interfaz:

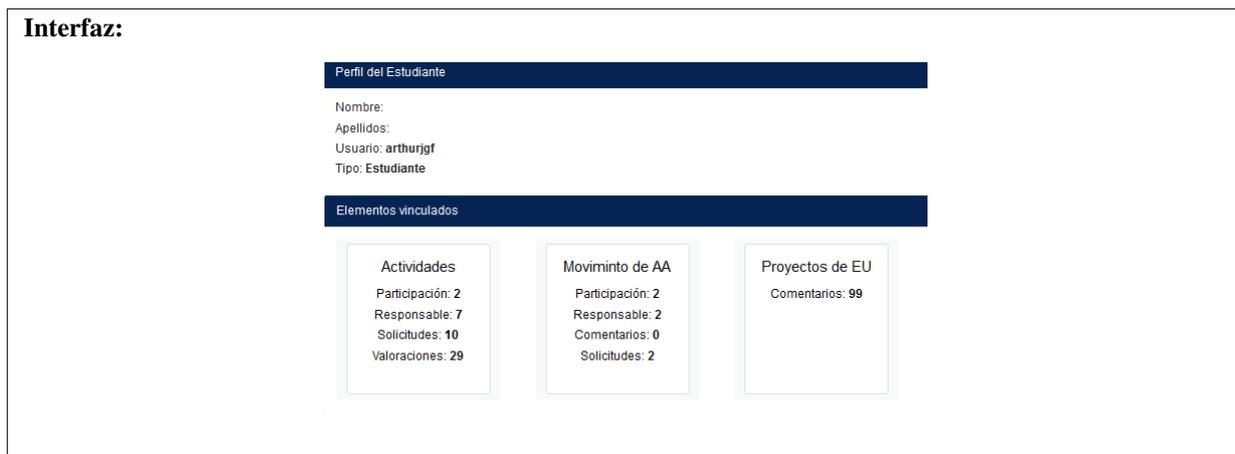


Tabla A.45. Historia de usuario # 48

Historia de usuario	
Número: 48	Nombre: Buscar invitado.
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir la búsqueda de los invitados.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
Interfaz:	
	

Tabla A.46. Historia de usuario # 49

Historia de usuario	
Número: 49	Nombre: Listar sugerencia.
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	
Descripción: El sistema debe permitir listar las sugerencias realizadas a una actividad.	

Continúa en la próxima página

Tabla A.46. Continuación de la página anterior

<p>Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.</p>
<p>Interfaz:</p> 

Tabla A.47. Historia de usuario # 50

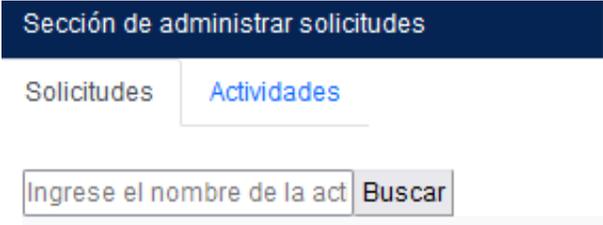
Historia de usuario	
Número: 50	Nombre: Listar recursos asociados.
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: José Adrian Torres Peraza.	
Descripción: El sistema debe listar los recursos asociados a una actividad.	
Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.	
<p>Interfaz:</p> 	

Tabla A.48. Historia de usuario # 51

Historia de usuario	
Número: 51	Nombre: Buscar solicitud.
Usuario: Responsable de institución.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Arthur Javier Gargallo.	

Continúa en la próxima página

Tabla A.48. Continuación de la página anterior

<p>Descripción: El sistema debe permitir la búsqueda de las solicitudes de actividades.</p>
<p>Observaciones: Este proceso será visible para cualquier usuario autenticado en el sistema con el rol de responsable de institución.</p>
<p>Interfaz:</p> 

Empleo de los métodos científicos

B.1. Modelo de encuesta aplicada a especialistas

Tabla B.1. Encuesta para aplicar Iadov

<p>Estimado(a): Lea con cuidado cada pregunta antes de responder. Le agradecemos su participación y franqueza al decirnos honestamente lo que piensa sobre el uso del módulo para la gestión del proceso de Instituciones Culturales en la UCI.</p>
<p>1- ¿Considera usted adecuado no usar las TIC para el proceso de gestión de Instituciones Culturales ? _ No _ No sé _ Sí</p>
<p>2- ¿Satisfacen sus necesidades como gestor de Instituciones Culturales, el módulo propuesto? _ No _ No sé _ Sí</p>
<p>3- ¿Utilizaría usted la aplicación propuesta a la hora de gestionar las Instituciones Culturales en la UCI? _ 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.</p>
<p>4- ¿Modificaría usted algún elemento del módulo propuesto para la gestión Instituciones Culturales? Argumente.</p>
<p>5- ¿Considera usted que el módulo propuesto permite una correcta gestión del proceso de promoción de la cultura? Argumente.</p>

B.1.1. Resultados de la encuesta aplicada a especialistas

- Pregunta 1. De los 10 encuestados el 100 % considera que No.
- Pregunta 2. De los 10 encuestados, 9 de ellos consideran que el sistema propuesto si satisfacen sus

necesidades.

- Pregunta 3. De los 10 encuestados, 9 de ellos consideran que le satisface mucho el uso del sistema propuesto y el resto considera que le da lo mismo.

Para las restantes preguntas de la encuesta el total de la muestra coincidió que las funcionalidades que ofrece el sistema permite una correcta gestión del proceso de Instituciones Culturales, los argumentos principalmente se basaron en:

- El sistema permite divulgar con éxito las actividades culturales.
- El sistema permite hacer el proceso auditable y comprobable.
- El sistema permite incorporar al proceso tanto a la comunidad universitaria como a la sociedad.
- El sistema ofrece un medio para la retroalimentación del proceso.

Sin embargo en referencia a la pregunta cuatro se observó, que a pesar de que las funcionalidades que cuenta el sistema, no son suficientes para la gestión del proceso, para algunos encuestados sería beneficioso incluirle otras como reportes estadísticos, aplicación de mensajería y sistema recomendador para actividades a los usuarios.

B.2. Guía de observación para el análisis de las soluciones existentes

Tabla B.2. Guía de observación para estudio de soluciones (Fuente: Elaboración propia).

No.	Indicadores	D	R	B	MB	E
1	Gestionan actividades culturales.					
2	Ven a las actividades culturales como núcleo del proceso de Instituciones Culturales.					
3	Gestionan medios y logística vinculada a las actividades culturales.					
4	Permiten intercambio entre los interesados para la mejora del proceso.					
5	Proveen un medio para medir el impacto de la realización del proceso.					
6	Da visibilidad al proceso de forma tal que facilite la participación de los interesados en el mismo.					
7	Permiten obtener las opiniones de los interesados sobre la realización del proceso.					

Leyenda	Deficiente	Regular	Bien	Muy bien	Excelente
	D	R	B	MB	E

- Portal de extensión universitaria de la Universidad de la Laguna (ULL)
- Sistema Nacional de Información Cultural- SINIC (Colombia)
- Sistema de Gestión de Inventario para galerías de artes (España)
- Sistema de gestión de la programación cultural en la dirección municipal de cultura en el municipio especial de la Isla de la Juventud
- Sistema para el área de Extensión Cultural de la Universidad de las Ciencias Informáticas