

**Universidad de las Ciencias Informáticas
Facultad 4**

**Sistema de gestión para el otorgamiento de
premios de la FEU para estudiantes en la UCI**

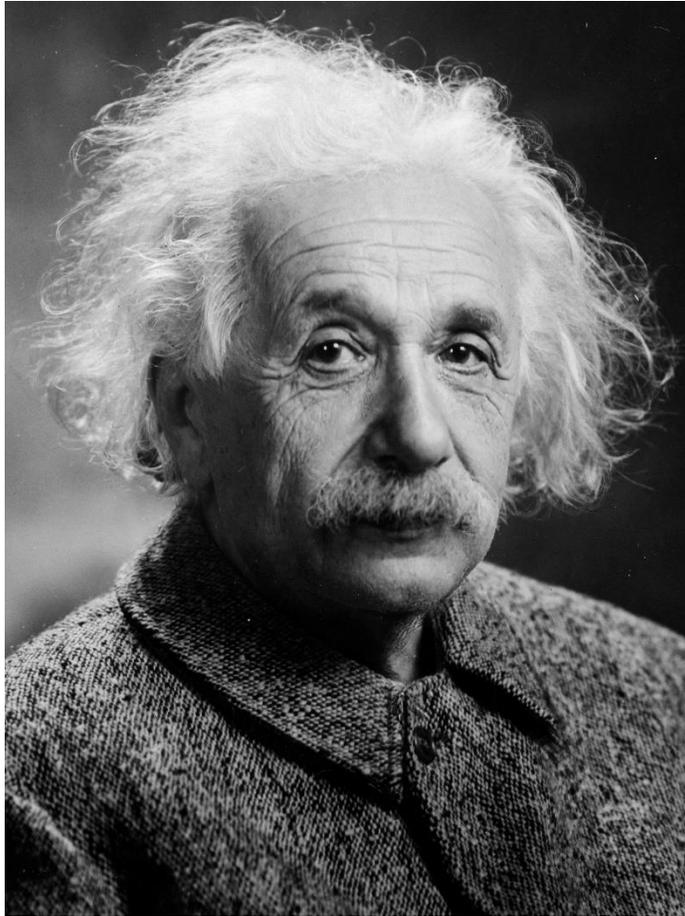
**Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas**

Autor(es): Jessica de la Caridad Martínez Hernández

Tutor(es): Ms.C. A Yordankis Matos López

Co-tutor: Ing. Javier González Vega

La Habana, diciembre 2022



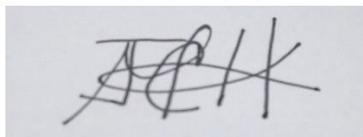
“Estoy agradecido por todos los que me dijeron NO. Es gracias a ellos que estoy siendo yo mismo.”

Albert Einstein

DECLARACIÓN DE AUTORÍA

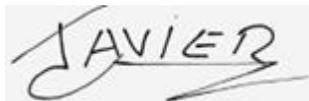
Declaro ser autor de la presente tesis que tiene por título: "Sistema de gestión para el otorgamiento de premios de la FEU en la UCI" y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 7 días del mes de diciembre del año 2022.

Jessica de la Caridad Martínez Hernández



Firma del Autor

Ing. Javier González Vega



Firma del Co-tutor

Ms.C. Yordankis Matos López



Firma del Tutor

RESUMEN

La Federación Estudiantil Universitaria es una organización cubana que acoge en su seno a todos los estudiantes universitarios, canalizando sus inquietudes y defendiendo sus intereses. Entre sus funciones se encuentra la entrega de premios a estudiantes que contribuyan al trabajo y perfeccionamiento de la organización. En la Universidad de las Ciencias Informáticas, a partir de una investigación realizada, se evidenció la falta de información referente a este proceso; no existe una adecuada comunicación entre los miembros de los secretariados de las facultades y sus homólogos de la universidad; la divulgación que se le hace a las distinciones no llega a la mayoría de la membresía, y el desconocimiento de las pautas para optar por ellos. El sistema de gestión de entrega de premios se concibe como una herramienta informática para garantizar una mayor eficiencia en la gestión de la información de todas las actividades desarrolladas en este proceso. Por tanto, el objetivo general del presente trabajo fue desarrollar un sistema que contribuya a la gestión de la información de la entrega de premios de la Federación Estudiantil Universitaria para estudiantes en la Universidad de las Ciencias Informáticas. Se presentan los resultados obtenidos de la selección de herramientas para el desarrollo del sistema, el levantamiento de requisitos, el análisis, diseño e implementación del software. Para el desarrollo de este sistema se empleó la metodología de software Programación Extrema. La solución implementada, luego de haber realizado las pruebas de software correspondientes, favorece la centralización de los datos asociados al proceso antes descrito.

PALABRAS CLAVE

Sistema de gestión, premios, estudiantes, Federación Estudiantil Universitaria, actividades

ABSTRACT

The University Student Federation is a Cuban organization that welcomes all university students, channeling their concerns and defending their interests. Among its functions is the delivery of prizes to students who contribute to the work and improvement of the organization. At the University of Informatics Sciences, from an investigation carried out, the lack of information regarding this process was evidenced; there is no adequate communication between the members of the secretariats of the faculties and their counterparts at the university; the disclosure that is made to the distinctions does not reach the majority of the membership, and the ignorance of the guidelines to opt for them. The prize-giving management system is conceived as a computer tool to guarantee greater efficiency in the management of the information of all the activities carried out in this process. Therefore, the general objective of this work was to develop a system that contributes to the information management of the awards ceremony of the University Student Federation for students at the University of Informatics Sciences. The results obtained from the selection of tools for the development of the system, the gathering of requirements, the analysis, design and implementation of the software are presented. For the development of this system, the Extreme Programming software methodology was used. The implemented solution, after having carried out the corresponding software tests, favors the centralization of the data associated with the process described above.

KEYWORDS

Management system, awards, students, University Student Federation, activities.

ÍNDICE

Introducción	- 1 -
Capítulo 1 Fundamentación teórica	- 6 -
1.1 Conceptos asociados	- 6 -
1.1.1 Federación Estudiantil Universitaria	- 6 -
1.1.2 Premios que otorga la FEU para estudiantes.	- 8 -
1.1.3 Gestión de la información	- 13 -
1.1.4 Sistemas de gestión	- 14 -
1.2 Análisis de sistemas homólogos existentes	- 15 -
1.2.1 Sitios web de organizaciones internacionales.	- 15 -
1.2.2 Sitio web de organizaciones nacionales.	- 17 -
1.2.3 Resultados del estudio de sistemas homólogos.....	- 18 -
1.3 Metodología de desarrollo de software	- 18 -
1.4 Frameworks de desarrollo web	- 21 -
1.5 Lenguaje de programación	- 22 -
1.6 Entorno de desarrollo integrado (IDE)	- 25 -
1.7 Gestor de base de datos.....	- 26 -
1.8 Herramienta para el modelado de la solución	- 27 -
1.9 Conclusiones parciales	- 28 -
Capítulo 2 Descripción de la propuesta	- 29 -
2.1 Propuesta de solución	- 29 -
2.2 Definición de los roles del sistema	- 30 -
2.3 Requisitos de software	- 31 -
2.3.1 Requisitos funcionales	- 31 -
2.3.2 Requisitos no funcionales.....	- 40 -

VII

2.4 Historias de usuario	- 42 -
2.5 Matriz de trazabilidad	- 44 -
2.6 Plan de iteraciones	- 45 -
2.7 Plan de entregas	- 47 -
2.8 Tarjetas CRC.....	- 48 -
2.9 Modelo de datos	- 49 -
2.10 Patrón arquitectónico	- 50 -
2.11 Patrones de diseño	- 51 -
2.11.1 Patrones GOF	- 52 -
2.11.2 Patrones GRASP	- 52 -
2.12 Conclusiones parciales	- 53 -
Capítulo 3 Implementación y prueba	- 55 -
3.1 Estándar de codificación de Django.....	- 55 -
3.2 Diagrama de despliegue	- 55 -
3.3 Pruebas de software.....	- 57 -
3.3.1 Estrategia de prueba	- 57 -
3.3.2 Pruebas unitarias	- 58 -
3.3.3 Pruebas funcionales	- 61 -
3.4 Conclusiones parciales	- 64 -
Conclusiones generales	- 65 -
Recomendaciones:.....	- 66 -
Referencias bibliográficas	- 67 -
Anexos.....	- 74 -
Anexo 1: Encuesta aplicada a estudiantes de la UCI	- 74 -
Anexo 2: Historias de usuario	- 76 -

VIII

Anexo 3: Tarjetas CRC.....	- 108 -
Anexo 4: Matriz de trazabilidad	- 109 -
Anexo 5: Modelo físico de datos	- 114 -
Anexo 6: Algunas de las no conformidades de las pruebas funcionales.....	- 116 -

ÍNDICE DE FIGURAS

Figura 1 Representación del patrón arquitectónico MTV. Fuente: Elaboración propia	- 51 -
Figura 2 Diagrama de despliegue	- 57 -
Figura 3. Respuesta de encuestados correspondiente a la pregunta 1. Fuente: Elaboración propia.....	- 74 -
Figura 4. Respuesta de encuestados correspondiente a la pregunta 2. Fuente: Elaboración propia.....	- 74 -
Figura 5. Respuesta de encuestados correspondiente a la pregunta 3. Fuente: Elaboración propia.....	- 75 -
Figura 6. Respuesta de encuestados correspondiente a la pregunta 4. Fuente: Elaboración propia.....	- 75 -
Figura 7. Respuesta de encuestados correspondiente a la pregunta 5. Fuente: Elaboración propia.....	- 76 -

ÍNDICE DE TABLAS

Tabla 1 Fases de la metodología XP.	- 20 -
Tabla 2 Roles del sistema.	- 30 -
Tabla 3 Requisitos funcionales	- 32 -
Tabla 4: Requisitos no funcionales identificados	- 40 -
Tabla 5 Historia de usuario Crear actividad Comisión Disciplinaria	- 42 -
Tabla 6 Plan de iteraciones.	- 45 -
Tabla 7 Plan de entregas.....	- 47 -
Tabla 8 Tarjeta CRC 1	- 48 -
Tabla 9 Tarjeta CRC 2	- 48 -
Tabla 10 Caso de prueba del método seleccionado	- 60 -
Tabla 11 Prueba funcional Crear propuesta	- 61 -
Tabla 12 No conformidades para una primera iteración de las pruebas funcionales - 63 -	
Tabla 13 Tipos de no conformidades detectadas	- 63 -
Tabla 14 Tarjeta CRC 3	- 108 -
Tabla 15 Tarjeta CRC 4	- 108 -
Tabla 16 Tarjeta CRC 5	- 109 -
Tabla 17 Matriz de trazabilidad.....	- 109 -
Tabla 18 Modelo físico de datos	- 115 -

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) pueden definirse como las tecnologías que utilizan la informática, la microelectrónica y las telecomunicaciones para crear nuevas formas de comunicación con el fin de facilitar la emisión, acceso y tratamiento de la información. Estas permiten una nueva forma de procesamiento de los datos que combina las tecnologías de la comunicación (TC) y las tecnologías de la información (TI). La suma de ambas al desarrollo de redes da como resultado un mayor acceso a la información. (1)

Para los expertos, no hay duda de que las TIC son una oportunidad para que el mundo educativo evolucione. Por lo tanto, los profesionales de la educación tienen múltiples razones para aprovechar las oportunidades que ofrecen estas tecnologías, siempre con la finalidad de orientarnos hacia un nuevo paradigma educativo que se centre en la actividad de los estudiantes.(2)

Los sistemas de gestión son el equivalente del siglo XXI a la revolución industrial. En aquel entonces, las empresas lograron aumentar la productividad, pasando de la fabricación artesanal a la masiva. Ahora, los programas informáticos son capaces de generar tareas automáticamente, aunque no de manera tan evidente a la vista como con una máquina industrial. La ventaja del software es que ocupa poco espacio y que además es asequible para cualquier tamaño de empresa.

Algunas pequeñas y medianas empresas, incluso los autónomos, pueden optar a implementar en su organización un sistema de gestión. La clave es introducir los datos en el sistema, para que se pueda acceder a la información de manera ágil. (3)

Cuba cuenta con más de 7,5 millones de usuarios de internet reveló la ministra de Comunicaciones, Mayra Arevich Marín, al intervenir en el panel Cuba hacia la transformación digital. El evento aconteció en la jornada de apertura de la 18va. Convención y Feria Internacional Informática 2022, que sesiona en el Palacio de las

Convenciones de la Habana (4). Lograr que la informatización de la sociedad cubana contribuya mejor al desarrollo económico del país y, por ende, a la calidad de vida de la población, es uno de los principales desafíos que existen en la actualidad.

La Universidad de las Ciencias Informáticas (UCI) es uno de los centros de la Educación Superior en Cuba que contribuye fervientemente con la informatización de la sociedad, para informatizar al país y desarrollar la industria cubana del software. Con el desarrollo de estas aplicaciones informáticas se ponen en manos de la sociedad más herramientas para el uso y acceso al conocimiento a través de plataformas informáticas seguras y efectivas. También se ayudarían a establecer puentes de comunicación directa entre instituciones, profesionales, centros estudiantiles, o ciudadanos, aprovechando el ciberespacio como una amplia red de transmisión de información e interacción social. (5)

En la UCI existe una estructura de la Federación Estudiantil Universitaria (FEU), la cual es una organización de masa que agrupa a los estudiantes universitarios, canalizando sus inquietudes y defendiendo sus intereses. Dentro de las actividades que realiza, presenta un sistema de reconocimiento, denominado distinciones, encargado de reconocer el accionar de los estudiantes en el transcurso del curso. Para ello se realiza la entrega de premios a los estudiantes más destacados en las actividades que se realizan en la universidad, que contribuyan al trabajo y perfeccionamiento de la organización. En el proceso de entrega de dichos premios se tomarán en cuenta determinadas pautas que el estudiante deberá cumplir. Para identificar posibles problemas en el desarrollo de dicho proceso, se decidió realizar un diagnóstico preliminar aplicando una encuesta. El objetivo era evaluar, desde la perspectiva de los estudiantes, que nivel de información poseía cada uno y qué importancia le daban a este proceso.

Con los resultados se evidenciaron varias deficiencias, tales como, la falta de información referente al proceso de entrega de premios de la FEU. No existe una adecuada comunicación entre los miembros de los secretariados de las facultades y sus homólogos de la universidad. La divulgación que se le hace a las distinciones

no llega a la mayoría de la membresía, y el desconocimiento de las pautas para optar por ellos. Estas problemáticas afectan el correcto desarrollo de este proceso desde la base, por lo que atentan contra la transparencia que lo caracteriza. La oportunidad de ser una universidad con alta disponibilidad tecnológica permite dar solución a los problemas mencionados.

Luego de lo expuesto anteriormente se plantea el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión de la información de la entrega de premios de la FEU para estudiantes en la UCI?

A partir del problema planteado se identifica como **objeto de estudio**: La gestión de la información de la entrega de premios.

Para dar solución a la problemática planteada se define como **objetivo general**: desarrollar un sistema que contribuya a la gestión de la información de la entrega de premios de la FEU para estudiantes en la UCI.

Para dar seguimiento al objetivo general se han trazado los siguientes **objetivos específicos**:

1. Establecer referentes teóricos de la investigación mediante el análisis de sistemas de gestión y herramientas para su informatización.
2. Analizar herramientas y tecnologías que se emplearán en el proceso de desarrollo del sistema de gestión de entrega de premios de la FEU para estudiantes en la UCI.
3. Diseñar un sistema que permita la gestión de la información asociada a la entrega de premios de la FEU en la UCI.
4. Implementar un sistema que permita la gestión asociada a la entrega de premios de la FEU en la UCI.
5. Validar el sistema propuesto a partir de las pruebas de software unitarias y funcionales.

Se delimita dentro del objeto de estudio de la investigación el **campo de acción**: La gestión de la información asociada a la entrega de premios de la FEU para

estudiantes en la UCI.

Para el desarrollo de la investigación se emplean los siguientes métodos de investigación:

Métodos Teóricos:

- **Histórico-lógico:** permitió realizar un estudio del arte e investigar sobre los sistemas relacionados con la gestión, obteniendo sus características y sus elementos a tomar en cuenta para seleccionar los patrones a implementar en la propuesta de solución.
- **Analítico-sintético:** Al emplear este método se realizó el análisis teórico y se identificaron los principales conceptos que se deben incluir en la documentación teórica y el análisis de la información. Facilitó la realización de un análisis de las herramientas similares que responden al objeto de estudio de la investigación con el objetivo de identificar características que tributen al desarrollo de la propuesta. Además, se pudo extraer elementos importantes relacionados con los sistemas de gestión.
- **Modelación:** permitió reflejar las características y relaciones de la solución a través de diagramas, permitiendo el diseño y una mejor comprensión de las clases que se utilizarán para la implementación de la propuesta de solución.

Métodos Empíricos:

- **Análisis documental:** Permitió la obtención de información relacionada con la temática abordada mediante el asesoramiento a través de diferentes artículos y libros.
- **Encuesta:** Permitió conocer opiniones y valoraciones de los encuestados sobre la temática que se aborda.

El presente trabajo cuenta con la siguiente estructura: introducción, tres capítulos, conclusiones generales y referencias bibliográficas además de una serie de anexos.

A continuación, se muestra la descripción de los capítulos:

Capítulo 1 Fundamentación teórica: En el presente capítulo se realiza un estudio de las tecnologías, herramientas y la metodología que se utilizará para el desarrollo del sistema de gestión. Se hace un estudio de los sistemas homólogos existentes y de las principales tendencias de la web que se tomará en cuenta para la implementación y desarrollo de la propuesta de solución.

Capítulo 2 Descripción de la Propuesta: En el presente capítulo se presenta la propuesta de solución y se definen los roles que estarán presentes en sistema. Se definen los requisitos funcionales identificados, con sus correspondientes historias de usuario, así como el plan de entrega, las tarjetas CRC y el modelo de datos, para un mejor entendimiento del código de la propuesta de solución. Se definirán el estilo arquitectónico y los patrones de diseño que se utilizarán.

Capítulo 3 Implementación y Prueba: En el presente capítulo se muestra el estándar de codificación utilizado en la implementación de la propuesta. También se realizan las pruebas de software que permiten verificar la calidad del sistema. Además, se presenta el diagrama de despliegue que muestra la configuración del funcionamiento del sistema.

Capítulo 1 Fundamentación teórica

En el presente capítulo se realiza un estudio de las tecnologías, herramientas y la metodología que se utilizará para el desarrollo del sistema de gestión. Se hace un estudio de los sistemas homólogos existentes y de las principales tendencias de la web que se tomará en cuenta para la implementación y desarrollo de la propuesta de solución.

1.1 Conceptos asociados

1.1.1 Federación Estudiantil Universitaria

La Federación Estudiantil Universitaria, más conocida por sus siglas FEU, es una organización cubana que acoge en su seno a todos los estudiantes universitarios cubanos. Sigue las orientaciones del Partido Comunista de Cuba y la Unión de Jóvenes Comunistas. Inicialmente sólo pertenecían a ella los alumnos de la Universidad de La Habana. Actualmente se han unido aquellos que estudian en la municipalización de la Universidad cubana. Fue fundada en 1922 por Julio Antonio Mella, quien fue un líder revolucionario de la década de los años 20 del siglo XX en Cuba. Esta organización surge al calor de las reformas universitarias desarrolladas en América. (7)

Defiende como esencia de su funcionamiento la participación activa de todos sus miembros, así como el principio del centralismo democrático, promoviendo la elegibilidad y el pensamiento colectivo como sustento básico para la dirección, la responsabilidad individual, así como el fomento de la crítica y autocrítica constructiva y constante; reconoce el peso decisivo del criterio de las masas que la integran; practica la rendición de cuentas de sus miembros, la discusión amplia y permanente de los asuntos de la Organización y la sociedad encaminando siempre su accionar a lograr la más estrecha vinculación de los estudiantes universitarios a los cuales se debe y representa. (8)

La FEU acepta en sus actividades políticas, investigativas, culturales, deportivas y recreativas la participación de los estudiantes universitarios que no son miembros

de sus filas y representa sus intereses en la vida universitaria. Y reconoce para el cumplimiento de su misión los siguientes objetivos generales:

- Representar al estudiantado universitario; canalizando sus inquietudes, defendiendo sus intereses y viabilizando sus iniciativas. Garantizar el ejercicio efectivo de los derechos de sus miembros y velar por el cumplimiento de sus deberes.
- Contribuir a la formación del Hombre Nuevo al que aspiró el Che, sobre la base del amor a la Patria, a la Revolución, a nuestros mejores valores y como defensor de las ideas que sustentan la ideología de la sociedad cubana.
- Reafirmar la vocación social de los universitarios a través de la actividad comunitaria, el cumplimiento de tareas de impacto social y la participación desde el estudio u otras actividades en la solución de problemas de la sociedad.

Reconoce como objetivos específicos:

- Fomentar la preparación académica del estudiantado a través del estudio consciente, la investigación científica y la necesidad de la superación constante.
- Promover el conocimiento de nuestra historia, tradiciones y el apego a la cultura e identidad nacional como fuente para la formación en valores de las nuevas generaciones.
- Potenciar el movimiento científico, deportivo y de artistas aficionados, como complementos imprescindibles en la formación integral de los estudiantes universitarios.
- Consolidar la solidaridad en las relaciones con el estudiantado revolucionario, progresista y democrático de todo el mundo, así como la atención y relaciones con los estudiantes de otras nacionalidades en Cuba.
- Integrar estos objetivos en la participación activa de los universitarios en la vida de la sociedad y su formación como graduados cada vez mejores, ética, profesional, cultural y políticamente, dispuestos a servir a la Revolución donde sea necesario.

Puede ingresar a la FEU todo estudiante universitario que curse estudios superiores en Cuba, sin distinción de sexo, raza, creencia religiosa, nacionalidad o procedencia social, que voluntariamente desee pertenecer a ella y no tenga estatus de trabajador; así como los dirigentes estudiantiles activos hasta dos años después de graduados.

1.1.2 Premios que otorga la FEU para estudiantes.

Según la Real Academia de la Lengua Española, la palabra premio es una distinción, un galardón o una recompensa que se otorga a alguien por algún mérito o servicio. Por lo general se trata de una compensación como reconocimiento a un esfuerzo o un logro. (9)

El premio puede ser obtenido de manera directa por ese esfuerzo, mérito que alguien ha hecho, o luego de participar en algún tipo de concurso mediante el cual el individuo que llega más lejos lo gana y lo recibe. Siempre es algo buscado por la persona que corresponde ya que siempre representa algún tipo de reconocimiento al esfuerzo, al trabajo, al conocimiento, o valor dependiendo de cada situación y también porque les aporta prestigio a su persona y a su carrera. (10)

En la Universidad de las Ciencias Informáticas, la FEU, entre los premios que otorga se encuentran: Premio A un futuro Maestro, Distinción Joven Aniversario de la FEU, Distinción 13 de Marzo y Distinción Orlando Ramos. A continuación, se analizarán las precisiones de cada uno de ellos.

Premio A un Futuro Maestro

Se entregará a maestros en formación, miembros del Movimiento de Alumnos Ayudantes “Frank País García”, y otros estudiantes vinculados a la impartición de docencia que se reconozcan entre sus compañeros y el claustro profesoral por su destacada participación y apoyo a la docencia.

a) Requisitos:

1. Haber participado y obtenido resultados en Festivales de Clases u otros eventos relacionados con la docencia.

2. Poseer una destacada participación en las actividades y procesos de la Organización.

3. Gozar del reconocimiento de sus compañeros de aula y los alumnos a los que da clases.

b) Precisiones:

1. Este proceso lo dirigirá el vicepresidente de la FEU de la universidad, las propuestas a reconocer se realizarán a todos los niveles, las mismas deben ser aprobadas por los Consejos de la FEU a cada nivel.
2. Se hará una entrega en el mes de diciembre en el marco de la Jornada por el día del Educador, después de realizada la evaluación del Proceso Docente Educativo; el reconocimiento se hará en lugares públicos, y quedará constancia del mismo en el local de la FEU y en su portal web.
3. El más destacado por facultad se le entregará el reconocimiento a nivel de Universidad en el acto central por el Aniversario de la FEU y la Jornada del Educador.

Distinción Joven Aniversario de la FEU

Con el objetivo de reconocer a los estudiantes de la organización que por sus resultados de trabajo constituyan ejemplo ante el estudiantado, el Secretariado de la FEU en la UCI, acordó establecer la distinción “Joven Aniversario de la FEU”.

Precisiones:

- Las propuestas deben salir del seno de la brigada, en este nivel no existe un número fijo por brigadas, sino todos aquellos que cumplan con los requisitos.
- El Secretariado debe revisar las propuestas, para llevar a aprobación del Consejo de la Facultad los estudiantes más destacados e integrales. El Consejo solo aprobará la propuesta que se mandará a la Universidad.

- El Secretariado UCI revisará cada propuesta y finalmente llevará al Consejo FEU UCI el grupo de estudiantes más integrales de los enviados por la facultad, para la aprobación final.
- El proceso de reconocimiento se realizará en el marco de los festejos por cada uno de los aniversarios de la emblemática y legendaria organización universitaria.

Requisitos para obtener la distinción:

1. Haber mantenido una trayectoria integral.
 - Actitud ante el cumplimiento de tareas de cualquier índole dadas por la organización.
2. Haber sido un estudiante destacado en las esferas de la vida universitaria.
 - Producción e Investigación.
 - Resultados Docentes.
 - Resultados en el Movimiento Alumnos.
 - Deporte y Cultura.
 - Concursos y Eventos.
 - Resultados obtenidos en la Residencia.
3. No haber sido objeto de análisis disciplinarios que trascendieran en la imagen del estudiante.
 - Comisiones disciplinarias donde resulte sancionado por estar involucrado en alguna indisciplina.

Nota: De la reunión de brigada saldrán todas las propuestas, las cuales revisará el

secretariado de la facultad para seleccionar la propuesta que se le hará a la Universidad, siendo el secretariado de esta quien apruebe finalmente.

Premio “Patria” para los corresponsales estudiantiles

Se otorga a jóvenes universitarios que hayan mantenido una trayectoria destacada dentro del Movimiento de Corresponsales de la organización. El reconocimiento es entregado por el Consejo de la FEU de la universidad y el Secretariado Nacional al nivel correspondiente.

a) Requisitos:

1. Mantener una trayectoria destacada dentro del Movimiento de Corresponsales de la FEU.
2. Participar activamente en todas las actividades y movimientos que promueve la FEU, así como contar con el prestigio y respeto de sus profesores y compañeros de brigada en el centro donde cursa estudios.

b) Precisiones:

1. Cada Facultad previa discusión y aprobación en su Consejo de la FEU a ese nivel, propone al Consejo de la FEU de la Universidad su candidato. El Consejo de la FEU de la Universidad analiza las propuestas y aprueba el estudiante que obtendrá el reconocimiento.
2. El premio se otorgará a todos los niveles cada 14 de marzo día de la prensa en Cuba, en homenaje a la fundación del periódico Patria por José Martí.

Distinción 13 de marzo

Se entrega a los dirigentes estudiantiles más destacados durante los 5 años de la carrera.

a) Precisiones:

1. Cada Secretariado de la facultad avalará sus propuestas, las cuales serán revisadas por la dirección del Secretariado UCI, y estos a su vez harán una propuesta al Consejo FEU UCI para su aprobación.

2. Antes de llevarse al Consejo, el presidente y vicepresidente de la FEU despachará con cada presidente de facultad la propuesta.
3. Para aprobar la selección deberán estar al menos el 85% de los miembros del Consejo de la FEU de la Universidad.
4. El proceso de reconocimiento se realizará en acto solemne, cada 13 de marzo, en conmemoración al aniversario de los asaltos al Palacio Presidencial y la toma de Radio Reloj.

Distinción “Orlando Ramos”

Se les otorga a los estudiantes que hayan tenido resultados relevantes en el campo de la ciencia y tengan una participación activa en las actividades de corte científico-investigativa convocado por la organización. Es el máximo reconocimiento que otorga la FEU de la Universidad de las Ciencias Informáticas, a aquellos estudiantes con una trayectoria relevante en la actividad investigativa.

a) Requisitos:

1. Haber obtenido al menos un premio como mínimo de relevante en alguna Jornada Científica Estudiantil a nivel de universidad.
2. Tener al menos una participación en los eventos científicos-investigativos externos a la UCI.
3. Tener resultados positivos en la producción.
4. Tener alto aprovechamiento académico.
5. Buena actitud ante las tareas convocadas por la FEU.

c) Precisiones:

1. Este proceso lo dirige el vicepresidente de la FEU de la universidad, cada Facultad previa discusión y aprobación en su Consejo de la FEU a ese nivel, propone al Consejo de la FEU de la Universidad su candidato. El Consejo de la FEU de la Universidad analiza las propuestas y aprueba el estudiante que obtendrá el reconocimiento.
2. El premio se otorgará a todos los niveles cada 15 de enero, en el marco del Acto Central por el Día de la Ciencia en Cuba.

3. No existirá un número total para entregar, pero sí es responsabilidad del Consejo entregar la distinción a los mejores. Debe ser un número ínfimo.

1.1.3 Gestión de la información

En el contexto de las organizaciones, la gestión de la información se puede identificar como la disciplina que se encargaría de todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, al coste adecuado, en el momento oportuno, en el lugar apropiado y articulando todas estas operaciones para el desarrollo de una acción correcta. En este contexto, los objetivos principales de la Gestión de la Información son: maximizar el valor y los beneficios derivados del uso de la información, disminuir el coste de adquisición, procesamiento y uso de la información, determinar responsabilidades para el uso efectivo, eficiente y económico de la información y asegurar un suministro continuo de la información. (11)

Woodman (1985) plantea: «la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa.» (12)

Páez Urdaneta considera como funciones de la gestión de información:

1. Determinar las necesidades internas de información (relativas a las funciones, actividades y procesos administrativos de la organización) y satisfacerlas competitivamente.
2. Determinar las necesidades de inteligencia externa de la organización y satisfacerlas competitivamente.
3. Desarrollar la base informacional de la organización y garantizar su accesibilidad.
4. Optimizar el flujo organizacional de la información y el nivel de las comunicaciones.

5. Desarrollar la estructura informacional de la organización y garantizar su operatividad.
6. Manejar eficientemente los recursos institucionales de información, mejorar las inversiones sucesivas en ellos, y optimizar su valor y su aprovechamiento organizacional.
7. Garantizar la integridad y accesibilidad a la memoria corporativa.
8. Evaluar periódicamente la calidad e impacto del soporte informacional para la gestión y el desarrollo de la organización.
9. Optimizar el aprovechamiento de la base y la estructura informacionales de la organización para incrementar su productividad o el rendimiento de la inversión.
10. Establecer, aplicar y supervisar los procedimientos relativos a la seguridad de la información organizacional.
11. Entrenar a los miembros de la organización en el manejo o la utilización, de los Recursos informacionales de la organización.
12. Contribuir a modernizar u optimizar las actividades organizacionales y los procesos administrativos, relacionados con ellas.
13. Garantizar la calidad de los productos informacionales de la organización, y asegurar su dimensión efectiva. (14)

Se concluye que la gestión de la información es ir en busca de nuevos significados, análisis; es producir un impacto en el medio ambiente de cualquier organización. Mediante la gestión se proporcionan los recursos necesarios para una buena toma de decisiones, se desarrollan nuevos conocimientos que posibilitan la calidad y eficiencia en los servicios.

1.1.4 Sistemas de gestión

Algunos autores como Davis y Olson (1985) conceptualizan los sistemas de gestión como un «sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización.» (15)

Moreiro González (1998) lo define como «el conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina. En su evolución el sistema puede manejar la función de inteligencia corporativa y generar productos de inteligencia.» (16)

Un sistema de gestión permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. Este aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa.

1.2 Análisis de sistemas homólogos existentes

Se realizó una búsqueda de sistemas de gestión y otras aplicaciones web de organizaciones nacionales y extranjeras con el objetivo de identificar funcionalidades que puedan tomarse en cuenta para la solución de la propuesta. Para ello se tomaron en cuenta varios criterios, entre ellos: si los sistemas presentaban mostraban la información de los premios, si mostraban un listado de los ganadores y si daban la posibilidad a los usuarios de proponer a una persona para recibir un determinado premio. A continuación, se muestran sus características y principales funcionalidades.

1.2.1 Sitios web de organizaciones internacionales.

Sistema de la Asociación Universitaria Iberoamericana de Postgrado

La Asociación Universitaria Iberoamericana de Postgrado -AUIP- es un organismo internacional no gubernamental reconocido por la UNESCO, dedicada al fomento

de los estudios de postgrado y doctorado en Iberoamérica. Actualmente está integrada por más de 247 prestigiosas instituciones de educación superior de España, Portugal, América Latina y el Caribe.

La AUIP presta servicios de información y divulgación sobre los postgrados que se ofrecen, colabora en procesos de evaluación interna y externa, acreditación y armonización curricular de esa oferta académica, facilita la movilidad e intercambio de profesores y estudiantes, incentiva el trabajo académico e investigador a través de redes de centros de excelencia en diversos campos del conocimiento, auspicia eventos académicos y científicos que están claramente relacionados con la formación avanzada y organiza cursos itinerantes internacionales en temas de interés para profesores y directores de programas de postgrado y doctorado.

Existen varias secciones dedicadas al otorgamiento de premios. Entre las condecoraciones que otorgan se encuentra el Premio AUIP a la Calidad del Postgrado en Iberoamérica, creado con el objetivo de promover los requisitos de calidad que aseguren un adecuado nivel de competitividad, reconocer públicamente los logros alcanzados por los programas, premiar a aquellas instituciones o programas que demuestren interés en permanecer activos aplicando una estrategia de mejoramiento continuo, facilitar los procesos de autoevaluación y de evaluación externa y divulgar las estrategias institucionales exitosas en pos de la calidad de sus programas de formación avanzada. Por tanto, son un reconocimiento a la calidad de los programas de formación avanzada que ofrecen las instituciones asociadas al sistema AUIP. Los premios pretenden potenciar los esfuerzos institucionales que se están haciendo para mejorar la oferta académica y de paso estimular su mejoramiento cualitativo. Para otorgar este premio la plataforma ofrece un apartado donde se muestra la descripción, los requisitos para su obtención y todo lo referente para la inscripción; además ofrece un listado con todos los ganadores de cada edición. Otras distinciones que forman parte de la plataforma son los Premios AUIP a la Sostenibilidad y el Premio Ignacio Ellacuría.

Sitio web de la Universidad de Guadalajara

En este sitio web se encuentra un apartado de convocatorias a diversos eventos, entre ellos se encuentra el Premio Nacional al Estudiante Universitario, en el cual se describe el premio en cuestión en el que podrán participar todos los estudiantes que cuenten con inscripción vigente en cualquiera de las carreras técnicas o de licenciatura de las universidades públicas o privadas del país. Otra distinción que otorga es el Premio ANUIES 2022, convocado por el Consejo Nacional de la Asociación Nacional de Universidades e Instituciones de Educación Superior, galardón que reconoce la labor destacada de quienes contribuyen al desarrollo de la educación superior del país.

Sitio web de la Universidad Autónoma de Santo Domingo

La Universidad Autónoma de Santo Domingo, primada de América, tiene como misión formar críticamente profesionales, investigadores y técnicos en las ciencias, las humanidades y las artes necesarias y eficientes para coadyuvar a las transformaciones que demanda el desarrollo nacional sostenible, así como difundir los ideales de la cultura de paz, progreso, justicia social, equidad de género y respeto a los derechos humanos, a fin de contribuir a la formación de una conciencia colectiva basada en valores. En el sitio web aparece una sección dedicada al Premio al Mérito Estudiantil, que se otorga a estudiantes de las diferentes facultades de la universidad; en ella aparece un listado con los ganadores del premio, con un filtrado que permite buscar a los estudiantes por facultad, además de un buscador.

1.2.2 Sitio web de organizaciones nacionales.

Sistema de Gestión Universitaria XAUCE AKADEMOS

Herramienta multiplataforma que contribuye al perfeccionamiento de los procesos académicos de una institución. Su uso permite el desarrollo coherente de una estrategia organizacional que articule todos los niveles de decisión presentes en los procesos universitarios. Todos los roles del proceso educativo están involucrados

en la solución, por lo que se permitirá el acceso a la información de forma segura a todos los niveles, facilitando la toma de decisiones. Desarrollado por la dirección de Informatización perteneciente a la Vicerrectoría de Tecnología de la UCI, gestiona un grupo de información de las áreas docentes, productivas e investigativas de la universidad referente a Pregrado, Residencia, Postgrado, Cooperación, Teleformación, Biblioteca, Desarrollo, Tecnologías, Investigación, Extensión, Ingreso y Egreso.

1.2.3 Resultados del estudio de sistemas homólogos

Después del estudio de los sistemas antes mencionados, se concluye que ninguno se adecua a las especificidades del negocio inherente a la investigación. Los sistemas homólogos encontrados no permiten ser utilizados para dar solución a la problemática planteada pues presentan características propias de las instituciones para las que fueron creadas, que no están acordes con los procesos que se desarrollan en la organización. Por lo tanto, se evidencia la necesidad de desarrollar una solución propia referente a la gestión de información asociada a la entrega de premios de la FEU para estudiantes en la UCI.

De los sistemas que se estudiaron existen funcionalidades que se pueden implementar en el sistema a desarrollar, como son:

1. Apartado donde se muestra la información referente al premio en cuestión
2. Listado con los ganadores de cada edición
3. Se identificaron requisitos no funcionales a tomar en cuenta

Luego de analizado el resultado anterior se puede apreciar que el estudio de dichas soluciones aportó elementos significativos a tomar en consideración para el desarrollo de la propuesta de solución. Durante la investigación no se encontró ninguna herramienta cubana que pudiera ser utilizada para dar solución a la problemática planteada. Por esta razón se propone incluir en la propuesta de solución las funcionalidades antes mencionadas.

1.3 Metodología de desarrollo de software

Las metodologías de desarrollo de software se utilizan en el ámbito de la programación, entre otros, con el objetivo de trabajar en equipo de manera organizada. Son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible.

El trabajo con una metodología de desarrollo de software permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar. En la actualidad se pueden diferenciar dos grandes grupos de metodologías de desarrollo de software: las ágiles y las tradicionales.

Las metodologías ágiles de desarrollo de software son las más utilizadas debido a su alta flexibilidad y agilidad. Los equipos de trabajo que las utilizan son mucho más productivos y eficientes, ya que saben lo que tienen que hacer en cada momento. Además, la metodología permite adaptar el software a las necesidades que van surgiendo por el camino, lo que facilita construir aplicaciones más funcionales.

Las metodologías ágiles se basan en la metodología incremental, en la que en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final. Sin embargo, los ciclos son mucho más cortos y rápidos, por lo que se van agregando pequeñas funcionalidades en lugar de grandes cambios.

Este tipo de metodologías permite construir equipos de trabajo autosuficientes e independientes que se reúnen cada poco tiempo para poner en común las novedades. Poco a poco, se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones, ya que puede comprobar cómo avanza el proyecto en tiempo real. (17)

Extreme Programming (XP) es una metodología ágil de desarrollo de software con bases en la comunicación constante y la retroalimentación y es idónea para proyectos con requisitos imprecisos. Uno de sus fines principales es el de construir un producto que vaya en línea con los requerimientos del cliente. En ese sentido es adaptable a los cambios, generando una rápida respuesta frente a cualquier inconveniente. Por otro lado, el equipo de trabajo tiene la ventaja de potenciar sus relaciones, ya que el proceso que de este se desprende es abierto, conjunto y

de aprendizaje continuo. (18)

Tabla 1 Fases de la metodología XP.

Fases XP	Objetivos de las fases (XP)
PLANIFICACIÓN	Según la identificación de las historias de usuario, se priorizan y se descomponen en mini-versiones. La planificación se va a ir revisando. Cada dos semanas aproximadamente de iteración, se debe obtener un software útil, funcional, listo para probar y lanzar.
DISEÑO	En este paso se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Se obtendrá el prototipo. Además, para el diseño del software orientado a objetos, se crearán tarjetas CRC (Clase-Responsabilidad-Colaboración).
CODIFICACIÓN «DE TODOS»	La programación aquí se hace «a dos manos», en parejas en frente del mismo ordenador. Incluso, a veces se intercambian las parejas. De esta forma, se asegura que se realice un código más universal, con el que cualquier otro programador podría trabajar y entender. Y es que deber parecer que ha sido realizado por una única persona. Así se conseguirá una programación organizada y planificada.
PRUEBAS	Se deben realizar pruebas automáticas continuamente. Al tratarse normalmente de proyectos a corto plazo, este testeo automatizado y constante es clave. Además, el propio cliente puede hacer pruebas, proponer nuevas pruebas e ir validando las mini-versiones. (19)

Teniendo en cuenta las características expuestas anteriormente, es necesario el uso de una metodología que responda con facilidad a cambios continuos, con una planificación flexible y abierta donde haya constante interacción entre el cliente y el equipo de desarrollo, por lo tanto, se propone usar la metodología XP en el desarrollo de la propuesta de solución.

1.4 Frameworks de desarrollo web

Un framework es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas. Es una estructura conceptual y tecnológica de asistencia definida con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. (20)

Django 4.0.1

Django es ya conocido desde hace tiempo como un framework de alto nivel para Python que combina diferentes bibliotecas. Generalmente, los desarrolladores lo utilizan para la creación de aplicaciones efectivas, ya que este framework es adecuado para casi cualquier proyecto.

Entre sus ventajas destaca que Django permite pasar del desarrollo de conceptos a la finalización del proyecto completo de una manera muy rápida y eficaz. Además, ayuda a los desarrolladores a evitar cometer errores de seguridad comunes, protegiendo automáticamente el sitio web. Para ello, Django se encarga de administrar la cuenta y la contraseña sin escribir información relevante en los archivos de cookies donde podría ser robada. Asimismo, este framework es muy rápido debido al hecho de que no está sobrecargado con detalles adicionales.

Django es capaz de reducir la cantidad de código, simplificando la creación de aplicaciones web y acelerando su desarrollo.

Debido a las características que presenta Django 4.0.1 se considera como una opción favorable para el desarrollo de la propuesta de solución, ya que, en comparación con otros frameworks, es mucho más veloz en cuanto al tiempo para desarrollar un proyecto y su interfaz para acceso a la base de datos y hacer consultas es sumamente buena. También es muy escalable y portable por lo que se puede ejecutar en muchas plataformas, dando mucha libertad al programador al momento de ejecutar las aplicaciones. (21)

1.5 Lenguaje de programación

Un lenguaje de programación es un lenguaje formal (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. (22)

Lenguaje de programación del lado del servidor:

Python 3.9

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina.

Python es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. También permite desarrollar webs complejas en menos líneas de código, lo

que permite que estas sean más ligeras y optimizadas. Django es uno de los frameworks de Python más populares de la actualidad, que puede ser utilizado para crear webs dinámicas y muy seguras. Python es también muy utilizado para hacer scraping, es decir, para obtener información de todo tipo de webs, tal y como lo hacen Netflix, Instagram o Pinterest (23). Debido a las tantas ventajas que brinda y al ser uno de los más usados en la actualidad se ha seleccionado como lenguaje de programación para implementar la propuesta de solución.

Lenguaje de programación del lado del cliente:

HTML5

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión del lenguaje HTML. Esta nueva versión (aún en desarrollo), y en conjunto con CSS3, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándolo así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora muchas características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web.

Algunas de las nuevas características de HTML5 serían:

- Nuevas etiquetas semánticas para estructurar los documentos HTML, destinadas a remplazar la necesidad de tener una etiqueta `<div>` que identifique cada bloque de la página.
- Los nuevos elementos multimedia como `<audio>` y `<video>`.
- La integración de gráficos vectoriales escalables (SVG) en sustitución de los genéricos `<object>`, y un nuevo elemento `<canvas>` que nos permite dibujar en él.
- El cambio, redefinición o estandarización de algunos elementos, como `<a>`, `<cite>` o `<menu>`.
- MathML para fórmulas matemáticas.
- Almacenamiento local en el lado del cliente. (24)

CSS3

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

El CSS sirve para definir la estética de un sitio web en un documento externo y eso mismo permite que modificando ese documento (la hoja CSS) se pueda cambiar la estética entera de un sitio web, el mismo sitio web puede variar totalmente de estética cambiando solo la CSS, sin tocar para nada los documentos HTML. CSS es un lenguaje utilizado para dar estética a un documento HTML (colores, tamaños de las fuentes, tamaños de elemento, con css se puede establecer diferentes reglas que indicarán como debe presentarse un documento. Se pueden indicar propiedades como el color, el tamaño de la letra, el tipo de letra, si es negrita, si es itálica, también se puede dar forma a otras cosas que no sean letras, como colores de fondo de una página, tamaños de un elemento. (25)

JavaScript (JS)

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Alguna de las principales características que presenta JS son: es simple, no hace falta tener conocimientos avanzados de programación para aprender a manejar JavaScript y es recomendado por muchos expertos a la hora de encontrar un lenguaje para comenzar a programar. Maneja objetos dentro de la página Web y sobre ese objeto se pueden definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos etc. Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto se puede cambiar totalmente el aspecto

de nuestra página al gusto del usuario, evitando tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario.

1.6 Entorno de desarrollo integrado (IDE)

Un IDE es el entorno digital empleado para desarrollar cualquier tipo de software cuyo objetivo es agilizar todo el proceso de diseño de software, ofreciendo un servicio integral al programador. La mayoría de los IDE permiten trabajar con diferentes lenguajes de programación y distintos sistemas operativos, aunque hayan sido diseñados para ser empleados específicamente en uno de ellos.

Además, los IDE mejoran su productividad, ya que pueden analizar códigos fácilmente, verificar la sintaxis simultáneamente y obtener comentarios instantáneos cuando surgen errores de sintaxis. También obtienen funciones y herramientas para organizar recursos, tomar atajos y evitar errores. Dado que el lenguaje de programación seleccionado para implementar la propuesta de solución es Python, se seleccionó uno los IDE más populares para programar en este lenguaje, a continuación, se describe dicho entorno. (26)

PyCharm 2021.1

Es desarrollado por JetBrains, una empresa de software checa, que proporciona todas las herramientas que necesita para Python en un solo lugar. Tiene muchas funciones de productividad para ahorrar tiempo y adoptar el enfoque centrado en el teclado. PyCharm conoce todos los aspectos del código. Por lo tanto, se puede confiar en esta solución para la verificación de errores sobre la marcha, la finalización de código inteligente, la navegación de proyectos sin esfuerzo y la rápida corrección de errores. Ayuda a los desarrolladores a escribir códigos limpios y mantenibles y a regular el control de calidad, refactorización inteligente, inspecciones y asistencia en las pruebas. Tiene un editor de código inteligente que ofrece el mejor soporte de su clase no solo para Python sino también para JavaScript, TypeScript, CoffeeScript, CSS, famosos lenguajes de plantillas, Node.js, AngularJS y más. Obtiene una búsqueda inteligente si desea saltar a una clase,

usos, implementación, prueba y más específicos. Admite la integración con varias herramientas de base de datos y VCS; intérpretes remotos, terminal SSH, Vagrant y Docker, Oracle, PostgreSQL, MySQL, SQL Server y más. También puede editar código SQL, examinar datos, ejecutar consultas y modificar esquemas con PyCharm. Es un IDE multiplataforma y funciona en Windows, Linux y Mac OS con una clave de licencia. Después de analizar las características del entorno se ha seleccionado PyCharm como el IDE a utilizar pues es uno de los IDE de Python más completos y populares, perfecto para todos los niveles de experiencia.

1.7 Gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD) o DataBase Management System (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible. En la actualidad, existen multitud de SGBD y pueden ser clasificados según la forma en que administran los datos en:

- Relacionales (SQL)
- No relacionales (NoSQL)

Desde que se comenzó a usar el modelo de bases de datos relacionales, en 1970, ha ido sufriendo una serie de transformaciones hasta convertirse, hoy en día, en el modelo más utilizado para administrar bases de datos. Este modelo se basa fundamentalmente en establecer relaciones o vínculos entre los datos, imaginando una tabla aparte por cada relación existente con sus propios registros y atributos.

PostgreSQL 13

Este sistema gestor de base de datos relacional está orientado a objetos y es libre, publicado bajo la licencia BSD. Sus principales características son:

- Control de Concurrencias multiversión (MVCC)

- Flexibilidad en cuanto a lenguajes de programación
- Multiplataforma
- Dispone de una herramienta muy fácil e intuitiva para la administración de las bases de datos.
- Robustez, eficiencia y estabilidad.

La principal desventaja es la lentitud para la administración de bases de datos pequeñas ya que está optimizado para gestionar grandes volúmenes de datos (27). Al ser una de las bases de datos más utilizadas en la actualidad, por ser gratuita, de código abierto y por ser altamente confiable, PostgreSQL, en su versión 13.0, ha sido seleccionado como gestor de bases de datos utilizado para la solución de la propuesta.

1.8 Herramienta para el modelado de la solución

Visual Paradigm v8.0 es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. (28)

Se caracteriza por:

1. Disponibilidad en múltiples plataformas (Windows, Linux).
2. Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
3. Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
4. Capacidades de ingeniería directa e inversa.
5. Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
6. Disponibilidad de múltiples versiones.
7. Licencia: gratuita y comercial.
8. Soporta aplicaciones Web.
9. Las imágenes y reportes generados, no son de muy buena calidad.

10. Varios idiomas.

Visual Paradigm 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. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos(29). Dadas las características antes mencionadas se concluye que Visual Paradigm es la herramienta idónea para modelar la propuesta de solución.

1.9 Conclusiones parciales

En el presente capítulo se abordaron elementos referentes a la problemática abordada, después de analizar cada aspecto se concluye que:

- Los sistemas homólogos estudiados responden a necesidades específicas de las instituciones para las que fueron creadas, lo que evidenció la necesidad de desarrollar un sistema propio y permitió la identificación de funcionalidades a incluir en la propuesta de solución.
- Se seleccionaron como metodología de desarrollo de software XP; Visual Paradigm v8.0 para el modelado, Pycharm 2021.1 como IDE; el lenguaje de programación para implementar el sistema será Python 3.9 con el framework Django 4.0.1 y como gestor de base de datos PostgreSQL 13.0; los cuales responden a las necesidades del proceso de desarrollo definido.

Capítulo 2 Descripción de la propuesta

En el presente capítulo se presenta la propuesta de solución y se definen los roles que estarán presentes en sistema. Se definen los requisitos funcionales identificados, con sus correspondientes historias de usuario, así como el plan de entrega, las tarjetas CRC y el modelo de datos, para un mejor entendimiento del código de la propuesta de solución. Se definirán el estilo arquitectónico y los patrones de diseño que se utilizarán.

2.1 Propuesta de solución

Dadas las necesidades planteadas en la situación problemática, la solución propuesta constituye un sistema de gestión para el otorgamiento de premios de la FEU en la UCI. Este sistema permite la organización, centralización y disponibilidad de la información acerca de los premios y sus respectivos estudiantes aprobados para recibir dichos premios. Estos procesos ayudarán a solucionar problemas que existen durante la realización de las propuestas para obtener dichas premiaciones. Dicho proceso tiene inmerso varios subprocesos, los cuales se describen a continuación:

Creación de un premio: El Consejo FEU crea un premio y le asigna determinados requisitos que deberá cumplir el estudiante que opte por obtenerlo, en correspondencia con las actividades que realiza el mismo en el curso.

Registrar actividades del estudiante: El Jefe de Grupo es el encargado de añadir a cada estudiante de su brigada, las actividades que realiza en el transcurso del curso.

Realizar propuesta: Los estudiantes de una brigada proponen a uno o varios estudiantes para que puedan obtener un premio. El sistema se encargará de comparar los requisitos del premio seleccionado y las actividades que haya realizado dicho estudiante en el curso y le ofrecerá a los que realicen la propuesta únicamente la opción de elegir a los estudiantes que cumplan con el premio.

Realizar premiación: El Consejo FEU del nivel correspondiente del premio se encarga de elegir entre los candidatos quién será aprobado para obtener

determinado premio otorgándole un lugar de premiación.

3.2 Definición de los roles del sistema

Dentro de la metodología del Desarrollo de Software se encuentra la fase del Análisis, donde se define claramente qué es lo que va a hacer el programa y para esto lo principal debe de ser la identificación de roles. Los roles son los usuarios y cualesquiera otros sistemas con los que se pueda comunicar con el sistema. Un actor especifica un rol jugado por un usuario o cualquier otro sistema que interactúa con el sujeto. En la tabla que se muestra a continuación se especifican los roles presentes en la propuesta de solución.

Tabla 2 Roles del sistema.

Roles	Justificación
Usuario	Representa el rol de los estudiantes de una brigada que se encargan de realizar las propuestas de los estudiantes para obtener un premio.
Jefe de Grupo	Usuario que representa a los directivos que se encargan de realizar las operaciones básicas de gestionar los datos de las diferentes actividades realizadas por los estudiantes de su brigada
Usuario de Sistema	Representa el rol encargado de gestionar los premios y elegir los aprobados para obtener dichos premios. El Usuario del Sistema es el rol que representa al Consejo FEU.
Administrador	Representa el rol encargado de gestionar los usuarios y los grupos donde se otorgan los permisos de acuerdo al rol

Dichos roles no necesariamente coinciden con los usuarios. Un usuario puede interpretar distintos roles. Los roles representan papeles que interpretan personas, periféricos u otros sistemas cuando el sistema está en uso. Estos podrían desempeñar distintos papeles dependiendo del caso de uso en que participe (30).

2.3 Requisitos de software

Los requisitos se han convertido en un punto clave en el desarrollo de las aplicaciones informáticas. Un gran número de proyectos de software naufragan debido a una mala definición, especificación o administración de requisitos. Factores tales como requisitos incompletos o mal manejo de los cambios de los requisitos llevan a proyectos completos al fracaso total.

El profesor Ian Sommerville presenta una definición acerca de lo que es un requisito: “Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema.” (31)

El objetivo principal de la *Especificación de Requisitos del Sistema* (ERS) es servir como *medio de comunicación* entre clientes, usuarios, ingenieros de requisitos y desarrolladores. En la ERS deben recogerse tanto las necesidades de clientes y usuarios (*necesidades del negocio*, también conocidas como *requisitos de usuario*, *requisitos de cliente*, *necesidades de usuario*, etc.) como los requisitos que debe cumplir el sistema software a desarrollar para satisfacer dichas necesidades (*requisitos del producto*, también conocidos como *requisitos de sistema* o *requisitos software*) (32).

2.3.1 Requisitos funcionales

Requerimiento funcional (FR) es una descripción del servicio que debe ofrecer el software. Describe un sistema de software o su componente. Solo se implementan entradas en el sistema de software, su transporte y salidas. Puede ser un cálculo, manipulación de datos, proceso comercial, interacción del usuario o cualquier otra funcionalidad específica que defina qué función probablemente realizará su sistema. En ingeniería de software e ingeniería de sistemas, los requisitos funcionales pueden abarcar desde la declaración abstracta de alto nivel de los requisitos del conductor hasta especificaciones detalladas de requisitos funcionales matemáticos. (33)

Tabla 3 Requisitos funcionales

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar usuario	El usuario introduce su usuario y contraseña, si los datos son válidos el usuario queda autenticado con el nivel de privilegios asignados de acuerdo a su rol; si los datos no son válidos el sistema muestra un mensaje de error.	Alta	Alta
RF2	Listar usuario	El sistema permite que los usuarios con rol administrador visualicen el listado con los usuarios existentes.	Baja	Baja
RF3	Crear usuario	El sistema debe permitir que los usuarios con rol administrador inserten nuevos usuarios al sistema a través de los siguientes datos: <ul style="list-style-type: none">• Nombre• Apellidos• Correo• Usuario• Contraseña	Alta	Media
RF4	Modificar usuario	El sistema debe permitir que los usuarios con rol administrador modifiquen los usuarios ya existentes a través de los siguientes datos: <ul style="list-style-type: none">• Nombre• Apellidos• Correo	Media	Media

		<ul style="list-style-type: none"> • Usuario • Contraseña 		
RF5	Eliminar usuario	El sistema debe permitir que los usuarios con rol administrador eliminen usuarios ya existentes	Media	Media
RF6	Listar estudiantes	El sistema debe permitir que los usuarios con rol jefe de grupo visualicen un listado de estudiantes	Baja	Baja
RF7	Crear estudiante	El sistema debe permitir que los usuarios con rol administrador inserten nuevos estudiantes al sistema a través de los siguientes datos: <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • Brigada • Facultad • Curso • Carrera • Imagen 	Alta	Alta
RF8	Modificar estudiante	El sistema debe permitir que los usuarios con rol administrador modifiquen los datos de los estudiantes a través de los siguientes datos: <ul style="list-style-type: none"> • Nombre • Primer Apellido • Segundo Apellido • Brigada • Facultad 	Media	Media

		<ul style="list-style-type: none"> • Curso • Carrera • Imagen 		
RF9	Eliminar estudiante	El sistema debe permitir que los usuarios con rol jefe de grupo eliminen a los estudiantes ya existentes	Media	Media
RF10	Importar listado de estudiantes	El sistema debe permitir a los usuarios con rol jefe de grupo importar un listado con los estudiantes a través de los siguientes datos: <ul style="list-style-type: none"> • Fichero a importar • Formato (csv, xls, xlsx, tsv, json, yaml) 	Media	Baja
RF11	Exportar listado de estudiantes	El sistema debe permitir a los usuarios con rol jefe de grupo exportar un listado con los estudiantes a través de los siguientes datos: <ul style="list-style-type: none"> • Formato (csv, xls, xlsx, tsv, json, yaml, ods, html) 	Media	Baja
RF12	Agregar actividad a estudiante	El sistema debe permitir que los usuarios con rol jefe de grupo agreguen actividades a un estudiante a través de los siguientes datos: <ul style="list-style-type: none"> • Tipo de Actividad (Cultural, Deportiva, Proyectos, Eventos, 	Alta	Alta

		<p>Ayudantía, Residencia, Comisión Disciplinaria)</p> <ul style="list-style-type: none"> • Fecha • Observaciones 		
RF13	Eliminar actividad del estudiante	El sistema debe permitir que los usuarios con rol jefe de grupo eliminen una actividad del estudiante	Media	Media
RF14	Listar actividades del estudiante	El sistema debe permitir que los usuarios con rol jefe de grupo visualicen un listado con las actividades del estudiante	Baja	Baja
RF15	Listar premio	El sistema permite que los usuarios con rol usuario de sistema visualicen el listado con los premios existentes.	Baja	Baja
RF16	Crear premio	<p>El sistema debe permitir que los usuarios con rol usuario de sistema creen nuevos premios a partir de los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre • Descripción • Fecha de premiación • Nivel • Lugar del nivel • Curso 	Alta	Media
RF17	Modificar premio	<p>El sistema debe permitir que los usuarios con rol usuario de sistema modifiquen los premios ya existentes a través de los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre 	Media	Media

		<ul style="list-style-type: none"> • Descripción • Fecha de premiación • Nivel • Lugar del nivel • Curso 		
RF18	Eliminar premio	El sistema debe permitir que los usuarios con rol usuario de sistema eliminen premios ya existentes	Media	Media
RF19	Mostrar datos del premio	El sistema debe permitir que los usuarios con rol usuario de sistema visualicen los datos del premio	Baja	Baja
RF20	Agregar requisitos a premio	<p>El sistema debe permitir que los usuarios con rol usuario del sistema agreguen requisitos al premio seleccionado introduciendo los siguientes datos:</p> <ul style="list-style-type: none"> • Tipo de Actividad (Cultural, Deportiva, Proyectos, Eventos, Ayudantía, Residencia, Comisión Disciplinaria) • Opción (Cantidad, Campo) • Atributo • Operación (igual que, mayor que, menor que) • Valor 	Alta	Alta
RF21	Listar requisitos de premio	El sistema debe permitir a los usuarios con rol usuario de	Baja	Baja

		sistema visualizar un listado de los requisitos del premio		
RF22	Eliminar requisito del premio	El sistema debe permitir que los usuarios con rol usuario de sistema eliminen un requisito del premio.	Media	Media
RF23	Crear actividad Proyectos	El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevos proyectos a partir de los siguientes datos: <ul style="list-style-type: none"> • Nombre de proyecto • Evaluación (B, R, M) 	Alta	Media
RF24	Crear actividad Ayudantía	El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevos alumnos ayudantes a partir de los siguientes datos: <ul style="list-style-type: none"> • Asignatura • Evaluación (B, R, M) 	Alta	Media
RF25	Crear actividad Deportiva	El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevos actividad deportiva a partir de los siguientes datos: <ul style="list-style-type: none"> • Tipo (Inter-años, Inter-facultad, universitarios, provinciales, nacionales) • Deporte • Resultado (Oro, Plata, Bronce, Ninguno) 	Alta	Media

RF26	Crear actividad Cultural	<p>El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades culturales a partir de los siguientes datos:</p> <ul style="list-style-type: none">• Nivel (Facultad, UCI, Nacional, Provincial)• Nombre de la Obra• Resultado (Oro, Plata, Bronce, Ninguno, Mención)• Manifestación (Literatura, Artes Plásticas, Audiovisual, Danza, Teatro)	Alta	Media
RF27	Crear actividad Residencia	<p>El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades en la residencia a partir de los siguientes datos:</p> <ul style="list-style-type: none">• Tipo (Cuartelería, Guardia estudiantil, TSU)• Evaluación (B, R, M)	Alta	Media
RF28	Crear actividad Comisión Disciplinaria	<p>El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas comisiones disciplinarias a partir de los siguientes datos:</p> <ul style="list-style-type: none">• Tipo (Muy Graves, graves, menos graves, leves)	Alta	Media

RF29	Crear actividad Eventos	<p>El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevos eventos a partir de los siguientes datos:</p> <ul style="list-style-type: none">• Resultado (Relevante, Destacado, Mención, Ninguno)• Nivel (Facultad, UCI, Provincial, Nacional, Internacional)• Nombre del evento	Alta	Media
RF30	Crear propuesta	<p>El sistema debe permitir que los usuarios con rol usuario inserten una propuesta a partir de los siguientes datos:</p> <ul style="list-style-type: none">• Premio• Estudiante• Observaciones	Alta	Media
RF31	Listar propuestas	<p>El sistema debe permitir que los usuarios con rol usuario visualicen el listado con las propuestas existentes</p>	Baja	Baja
RF32	Asignar lugar de premiación	<p>El sistema debe permitir que los usuarios con rol usuario de sistema otorguen el lugar de premiación a un estudiante a través de los siguientes datos:</p> <ul style="list-style-type: none">• Premio• Estudiante• Lugar	Media	Media

RF33	Listar aprobados para premios	El sistema debe permitir que se visualice un listado de los aprobados por premio	Baja	Baja
RF34	Exportar pdf de listado de aprobados por premio	El sistema debe permitir exportar un pdf del listado de los aprobados de acuerdo al premio que se seleccione	Baja	Baja

Los requisitos a tener en cuenta para la implementación de la propuesta de solución se muestran en la tabla mostrada anteriormente. Como resultado de la aplicación de las técnicas de captura de información se identificaron 34 requisitos funcionales.

2.3.2 Requisitos no funcionales

Son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad. Son propiedades o cualidades que el producto debe tener. Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del sistema. Alternativamente, definen restricciones tales como la capacidad de los dispositivos de entrada/salida y la representación de los datos utilizados en la interfaz del sistema (33).

Tabla 4: Requisitos no funcionales identificados

Requisitos	Descripción
Requisito de seguridad	<ul style="list-style-type: none"> • RnF 1 Los usuarios tendrán acceso a las funcionalidades e información del rol asignado. • RnF 2 La entrada al sistema se realizará a través de la autenticación. • RnF 3 El sistema tras 10 minutos de inactividad cerrará sesión automáticamente.

Requisito de soporte	<ul style="list-style-type: none">• RnF 4 Se podrá incorporar al sistema nuevas funcionalidades si son necesarias.• RnF 5 El sistema debe recibir mantenimiento anualmente.
Requisito de software.	<ul style="list-style-type: none">• RnF 6 Se requiere de la instalación del servidor web Apache para poder visualizar la interfaz web.• RnF 7 Se requiere de un gestor de base de datos PostgreSQL en su versión 13 o superior.• RnF 8 Se necesita instalado en la estación de trabajo un sistema operativo: Linux, Windows 7 o superior.• RnF 9 Se requiere en el servidor un sistema operativo de Linux Server y Windows Server 2012 R2 o superior.• RnF 10 Se requiere instalar en la estación de trabajo un navegador Chrome o Firefox en sus últimas versiones.• RnF 11 Se requiere instalar en la estación de trabajo un visor de PDF para mostrar los documentos generados.
Requisito de usabilidad.	<ul style="list-style-type: none">• RnF 12 El sistema deberá mostrar mensajes de notificación ante la ocurrencia de eventos.• RnF 13 El sistema deberá ser intuitivo con el fin de establecer una navegación sencilla para el cliente.• RnF 14 El sistema deberá mostrar mensajes de confirmación ante la ocurrencia de acciones críticas con el fin de prever a los usuarios de cometer errores.

Requisito de hardware.	<p>Para el servidor:</p> <ul style="list-style-type: none">• RnF 15 Requerimientos mínimos: Procesador Intel(R) Core (TM) i3-4160 CPU 3.60GHz de velocidad de procesamiento, 2GB de memoria RAM y una capacidad de 40GB de disco duro. <p>Para el cliente:</p> <ul style="list-style-type: none">• RnF 16 Requerimientos mínimos: Procesador Pentium IV a 1.40 GHZ con 1 GB de memoria RAM y una capacidad superior a 40GB de disco duro.
Requisito de integridad de datos.	<ul style="list-style-type: none">• RnF 17 El sistema debe garantizar la persistencia de los datos insertados por el usuario.

En la tabla que se muestra anteriormente se presentan los requisitos no funcionales identificados. Estos se conciben como las características que hacen al producto atractivo, usable, rápido o confiable. Describen la experiencia del usuario cuando trabaja con el producto. No son parte de la razón fundamental del producto, pero si son necesarios para hacer funcionar el producto de la manera deseada.

2.4 Historias de usuario

Las historias de usuario son parte de un enfoque ágil que ayuda a cambiar el enfoque de escribir sobre los requisitos a hablar sobre ellos. Son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema.

Tabla 5 Historia de usuario Crear actividad Comisión Disciplinaria

Historia de usuario

Número: RF28	Nombre del requisito: Crear actividad Comisión Disciplinaria
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo comisión disciplinaria.</p> <p>Objetivo: Permitir incluir nuevas actividades de tipo comisión disciplinaria a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para incluir una actividad comisión disciplinaria hay que: - Tener en cuenta los siguientes datos: Tipo - Estar autenticado en el sistema con el rol jefe de grupo</p> <p>Comportamientos válidos y no válidos (flujo central y alternos): El campo tipo no es obligatorio Tipo: Campo de selección con las opciones: Muy Graves, graves, menos graves, leves</p> <p>Flujo de la acción a realizar: - El sistema debe permitir seleccionar los datos para incluir una nueva actividad comisión disciplinaria. - Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad comisión disciplinaria y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información. - Si selecciona la opción Cancelar regresará a la vista previa.</p>	
<p>Observación: Las actividades de comisión disciplinaria son para tener un registro de las sanciones que posee cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de</p>	

determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.
--

Durante el diseño de la propuesta de solución se identificaron 34 historias de usuario que corresponden a cada requisito funcional y presentan una descripción que permita que el desarrollador conozca cómo funciona para su posterior implementación. En la tabla mostrada anteriormente se ilustra una de las historias de usuario asociada a un requisito correspondiente a la primera iteración. En el Anexo 2 se presentan las restantes historias de usuario.

2.5 Matriz de trazabilidad

La trazabilidad en la Ingeniería de Software es una práctica de control que ayuda a obtener el producto en el dominio de la solución lo más exacto y fiable posible a las necesidades expresadas por el cliente en el dominio del problema. La trazabilidad está condicionada por los cambios y las validaciones que los participantes del proyecto hagan al sistema durante el proceso de desarrollo (33). Según el estándar IEEE 830-1998, la trazabilidad es la habilidad para seguir la vida de un requerimiento en ambos sentidos, hacia sus orígenes o hacia su implementación a través de las especificaciones generadas durante el proceso de desarrollo.

Una de las técnicas más utilizadas para recoger la información bidireccional de trazas, son las matrices de trazabilidad. Éstas muestran diversos elementos en filas y columnas indicando en cada celda de la matriz si los elementos están o no trazados y en qué dirección. Este tipo de técnicas permite un análisis gráfico de la trazabilidad de requisitos y la gestión de su impacto ante posibles cambios. (34)

Se realiza una matriz de trazabilidad con el objetivo de comprobar que las historias de usuario obtenidas satisfacen los requisitos funcionales identificados en el sistema. En el Anexo 4 se muestra una matriz de trazabilidad donde todos los requisitos funcionales se muestran de forma vertical y las historias de usuario se muestran de forma horizontal, de manera que cada historia de usuario se corresponda con un requisito funcional, como plantea la metodología de software empleada.

2.6 Plan de iteraciones

En la metodología XP, la creación del sistema se divide en iteraciones. La duración ideal de una iteración está entre una y tres semanas. Para cada una de las iteraciones el cliente establece un conjunto de HU que serán implementadas en cada iteración del sistema. Al final de cada iteración se realizan las pruebas de software y la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos.

Tabla 6 Plan de iteraciones.

Iteración	Historias de Usuario	Duración total de las iteraciones (semanas)
1	Autenticar usuario	3 semanas
	Listar usuario	
	Crear usuario	
	Modificar usuario	
	Eliminar usuario	
	Listar estudiantes	
	Crear estudiante	
	Modificar estudiante	
	Eliminar estudiante	
	Mostrar datos del estudiante	
	Importar listado de estudiantes	
	Exportar listado de estudiantes	

2	Crear actividad Proyectos	4 semanas
	Crear actividad Ayudantía	
	Crear actividad Deportiva	
	Crear actividad Cultural	
	Crear actividad Residencia	
	Crear actividad Comisión Disciplinaria	
	Crear actividad Eventos	
	Agregar actividad a estudiante	
	Eliminar actividad a estudiante	
	Listar actividades del estudiante	
3	Listar premio	4 semanas
	Crear premio	
	Modificar premio	
	Eliminar premio	
	Mostrar datos del premio	
	Agregar requisitos a premio	
	Listar requisitos de premio	
	Eliminar requisito del premio	

	Crear propuesta	
	Listar propuesta	
	Mostrar propuesta	
	Asignar lugar de premiación	
	Listar aprobados para premio	
	Exportar pdf de listado de aprobados de premio	

En la tabla mostrada anteriormente se muestra la distribución de las historias de usuario(HU), en cada una de las tres iteraciones definidas. En la primera iteración se tomaron en cuenta 12 HU con una duración de 3 semanas; la segunda iteración presenta 10 HU con una duración de 4 semanas y la tercera iteración tiene 14 HU con 4 semanas, teniendo un total de 11 semanas para la realización de las mismas.

2.7 Plan de entregas

La entrega del software es el proceso completo de hacer llegar un producto a los clientes, desde la conceptualización hasta el desarrollo. Al concluir cada iteración se realiza una versión de la aplicación. En el plan de entregas se realiza un cronograma de entregas donde se tienen en cuenta los siete días de la semana.

Tabla 7 Plan de entregas

Entregable	1ra Iteración	2da Iteración	3ra Iteración
Versión	Versión 1.0	Versión 1.2	Versión 1.3
Fecha	(25/09/22-01/10/22)	(02/10/22-29/10/22)	(30/10/22-19/11/22)

La tabla mostrada anteriormente refleja las posibles versiones que tendrá el sistema con un plan de tres iteraciones empezando el 21 de agosto del 2022, y finalizando

el 31 de octubre del 2022.

2.8 Tarjetas CRC

Las Tarjetas CRC (Clase-Responsabilidades-Colaboradores), permiten conocer que clases componen el sistema y cuales interactúan entre sí. La técnica consiste en dibujar una tarjeta por cada clase u objeto, y dividirla en tres zonas:

- En la parte superior, el nombre de la clase;
- Debajo, en la parte izquierda, las responsabilidades de dicha clase que constituyen sus objetivos;
- A la derecha de las responsabilidades, los colaboradores, que son otras clases que ayudan a conseguir cumplir a esta con sus responsabilidades (35).

Tabla 8 Tarjeta CRC 1

Tarjeta CRC	
Nombre de la clase: RequisitoCreateView	
Responsabilidades: obtenerPremio(self, pk): Obtiene la información del premio al que se le quiere añadir requisitos.	Colaboradores:

Tabla 9 Tarjeta CRC 2

Tarjeta CRC	
Nombre de la clase: PremioCreateView	
Responsabilidades: obtenerCurso(self, pk): Obtiene la información del curso al que se le quiere añadir el premio.	Colaboradores:

Las tarjetas CRC permiten ver las clases como algo más que repositorio de datos, para conocer el comportamiento de cada una en un alto nivel. Fueron descritas un total de 5 tarjetas CRC, en este epígrafe se presentan dos de estas. En el Anexo 3 se muestran las restantes.

2.9 Modelo de datos

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten. La mayoría de los modelos de datos se pueden representar por medio de un diagrama de base de datos acompañante. (34)

Brinda un contexto más específico. De este modo, se muestra la estructura de las diferentes tablas y los componentes de las mismas, en donde se incluye el nombre de las columnas, el tipo de datos de cada una de ellas y sus restricciones, las claves primarias y claves externas, las distintas relaciones entre tablas, los índices y los procedimientos dentro de la base de datos. se puede aplicar a un solo proyecto; no obstante, también puede integrarse con otros modelos de datos físicos, dependiendo del alcance que tenga el proyecto. Requiere un gran conocimiento de los datos que se implementarán físicamente, es decir, de todos los datos físicos que se incluirán en la base de datos. Por ello, suele ser un modelo complicado para los implicados en el proyecto que no tienen mucha información acerca del mismo y que no conocen a fondo la estructura que se llevará a cabo en la base de datos. También es importante destacar que, si bien puede ser un modelo de gran utilidad, incluso los cambios más pequeños pueden llevar a la obligación de modificar todas las estructuras relacionadas con la estructura modificada. (35)

En el Anexo 5 se muestran las principales entidades del proceso del negocio con los pertinentes atributos asociados a las mismas, mediante el modelo de datos

físicos del sistema. Este diagrama ayuda a comprender mejor las entidades presentes, así como, la relación que se establece entre ellas.

2.10 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. 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 (35).

Django es conocido como un Framework MTV (Modelo-Vista-Plantilla). En el patrón de diseño MTV,

- M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos. Los Modelos son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (añadir, modificar y borrar) y consultar registros en la base de datos.
- T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento. Una plantilla (template) es un fichero de texto que define la estructura o diagrama de otro fichero (tal como una página HTML), con marcadores de posición que se utilizan para representar el contenido real. Una vista puede crear dinámicamente una página usando una plantilla, rellenándola con datos de un modelo. Una plantilla se puede usar para definir la estructura de cualquier tipo de fichero;
- V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre los modelos y las plantillas.

Una vista es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formato de la respuesta a las plantillas ("templates"). (34)

En la siguiente figura se puede observar una representación gráfica del funcionamiento del Patrón Modelo Vista Plantilla:



Figura 1 Representación del patrón arquitectónico MTV. Fuente: Elaboración propia

En la propuesta de solución esto se pone de manifiesto al interactuar el usuario con el sistema, este envía la petición a la clase de la vista correspondiente en el archivo views.py, el cual si debe interactuar con la base de datos, consulta los modelos necesarios en el archivo models.py, el cual se encarga de hacer consultas necesarias a la base de datos, y seguidamente muestra al usuario la plantilla que corresponde con la petición que este hizo, de lo contrario no consulta el modelo y solo retorna la plantilla correspondiente.

2.11 Patrones de diseño

En ingeniería de software se conoce como patrón de diseño a una solución estandarizada, probada y repetible a un problema común en el diseño de software. Un patrón de diseño es una descripción o plantilla sobre cómo resolver un

problema que se puede utilizar en muchas situaciones diferentes.

El uso de patrones de diseño permite acelerar el proceso de desarrollo al proporcionar paradigmas probados y comprobados en el diseño de software. Estandarizar las soluciones a problemas comunes permite también mejorar la comunicación entre desarrolladores ya que se emplean nombres bien conocidos a la hora de describir un problema de diseño y como abordar su solución. (35)

2.11.1 Patrones GOF

En la construcción de aplicaciones Web, los problemas a los que diseñadores y desarrolladores se enfrentan difieren de los que se presentan en otro tipo de procesos de desarrollo. El catálogo de referencia definido por GoF (Gamma et al, 1994) no muestra una tipificación que permita identificar qué patrón puede emplearse específicamente para el desarrollo de aplicaciones Web. En el análisis realizado a procesos de desarrollo se identifican los patrones de diseño que son empleados para el desarrollo de aplicaciones Web, aunque son 23 patrones de diseño los que se presentan en el catálogo GoF, no todos pueden ser usados para el desarrollo de aplicaciones orientadas a la Web (36). En la propuesta de solución se ve reflejado el patrón Decorador.

Decorador: Permite añadir funcionalidad extra a un objeto (de forma dinámica o estática) sin modificar el comportamiento del resto de objetos del mismo tipo. Django presenta un decorador que permite añadir funcionalidades dinámicas. Esto se ve reflejado en las vistas generadas que heredan el diseño de la plantilla "home.html", que contiene la estructura y el diseño básico de las demás plantillas.

2.11.2 Patrones GRASP

Patrones GRASP es la denominación común que reciben las buenas prácticas asociadas a la asignación de responsabilidades entre los objetos de un software, surgida de la frase en inglés *General Responsibility Assignment Software Patterns*. La existencia de estos patrones está estrechamente relacionada con el paradigma

de Programación Orientada a Objetos, donde un objeto representa dentro del código fuente a una entidad real.

Patrón Experto

El patrón experto se encarga de lograr que cada objeto dentro de un software sea creado por la clase que posee todos los elementos para hacerlo. Garantiza que, aunque otras clases utilicen el mismo objeto, solo la experta tenga la responsabilidad de construirlo. Esto favorece que cada objeto puede cambiar sus atributos o comportamiento, sin que las clases relacionadas deban cambiar drásticamente. La creación del objeto se realiza dentro de sí mismo, característica que se conoce como encapsulamiento. En la propuesta de solución, este patrón se refleja en la clase Premio, que es la clase que realiza instancias a Propuesta y Requisitos.

Patrón Creador

Este patrón es similar al anterior, pero su misión está enfocada en asignar la responsabilidad correcta para crear instancias de un objeto, o dicho claramente, realizar las llamadas a los objetos cuando van a ser utilizados. En la propuesta de solución este patrón se refleja, por ejemplo, en la clase Premio y Propuesta, permitiendo la creación de objetos. (38)

2.12 Conclusiones parciales

En el presente capítulo se abordaron elementos referentes a la solución de la problemática abordada, después de analizar cada aspecto se concluye que:

- Se identificaron los requisitos funcionales del sistema, que permitieron tener una visión más clara sobre sus características y, en correspondencia con estos, se describió detalladamente el funcionamiento del sistema propuesto, con el empleo de historias de usuario.
- Se identificaron las facilidades que proporciona la arquitectura MTV, previamente definida en correspondencia con el framework de desarrollo que

se empleará y se complementaron con la identificación de patrones de diseño de software que aportan buenas prácticas a la implementación.

Capítulo 3 Implementación y prueba

En el presente capítulo se muestra el estándar de codificación utilizado en la implementación de la propuesta. También se realizan las pruebas de software que permiten verificar la calidad del sistema. Además, se presenta el diagrama de despliegue que muestra la configuración del funcionamiento del sistema.

3.1 Estándar de codificación de Django

Los estándares de codificación definidos por los desarrolladores de Django tienen el objetivo de cuando se vaya a desarrollar un sistema, a los otros desarrolladores se les facilite entender el código. Alguno de los estándares que utiliza se muestran a continuación:

Trabajo con funciones

Se utilizará el estándar de indentación en el nombre de la función, el inicio con minúscula para las funciones creadas en los archivos en los archivos views.py asociados a Python, por ejemplo: `def obtenerPremio()`

Trabajo con clases

Se utilizará el estándar de indentación en el nombre de la función, el inicio con mayúscula para las clases creadas en los archivos en los archivos views.py asociados a Python, por ejemplo: `class PremioListView`

Trabajo con formularios

Para el trabajo con formularios en Django, la indentación del uso de forms en los archivos forms.py será el nombre en mayúscula asociado al formulario junto con la palabra Form, por ejemplo: `PremioForm(ModelForm)`

3.2 Diagrama de despliegue

Diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Describen la

topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP. Muestran la configuración en funcionamiento del sistema incluyendo su software y su hardware. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de la red, el tiempo de respuesta. (39)

El diagrama de despliegue se estructura con los siguientes elementos:

- **Nodo:** es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Un Nodo es un elemento de hardware o software.
- **Asociación de Comunicación:** Esto está representado por una línea sólida entre dos nodos. Muestra el camino de la comunicación entre los nodos.
- **Dispositivos:** Un dispositivo es un nodo que se utiliza para representar un recurso físico computacional en un sistema. (40)
- **Pc_cliente:** Es la estación de trabajo que representa un navegador web que se emplea para conectarse a la aplicación hospedada en el servidor de aplicaciones mediante un protocolo http.
- **Servidor_aplicación:** Estación de trabajo que hospeda el código fuente de la aplicación, y que brinda a los usuarios las interfaces de la misma para realizar los procesos definidos por cada rol. Esta estación se comunica con el servidor de la base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP/IP.
- **Base_datos:** Este servidor es el encargado de almacenar los datos del sistema y se comunica con el servidor de aplicaciones.

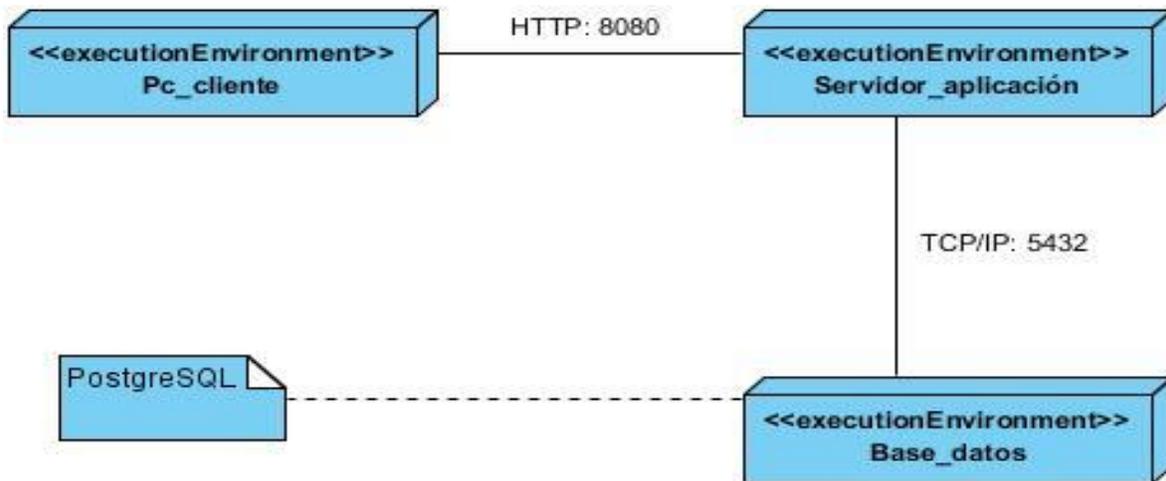


Figura 2 Diagrama de despliegue

Con la elaboración del diagrama de despliegue se puede modelar la disposición física del sistema, muestra el hardware usado; los componentes instalados en el hardware. Además, se conocen los enlaces de comunicación entre estos componentes.

3.3 Pruebas de software

La prueba de software es el proceso de evaluación y verificación de un producto o aplicación de software para saber si hace lo que se supone que debe hacer. Los beneficios de las pruebas incluyen la prevención de defectos, la reducción de los costos de desarrollo y la mejora del rendimiento. (41)

Cuando se le van a aplicar pruebas a un software, se realiza una estrategia de prueba teniendo en cuenta una serie de objetivos en diferentes escenarios y niveles de trabajo, debido a que las pruebas son agrupadas por niveles que se encuentran en distintas etapas del proceso de desarrollo.

3.3.1 Estrategia de prueba

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el

diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (45).

Teniendo en cuenta los niveles de prueba: unidad y sistema, se definen los diferentes tipos de prueba a realizar, en las que se tendrán en cuenta las tres iteraciones definidas en el proyecto. En el primer nivel de pruebas, por cada método que se cree en el sistema, que responden a los requisitos funcionales, se van a desarrollar pruebas unitarias para comprobar que el sistema funcione de forma correcta y que la salida de los métodos sean las esperadas. Una vez que el sistema ya se encuentre desarrollado se realizarán pruebas funcionales, para las cuales se utilizará el método de caja negra, empleando la técnica de partición de equivalencia. En caso de que se detecten no conformidades, al final de cada iteración se realizarán pruebas de regresión para verificar que las no conformidades encontradas hayan sido corregidas.

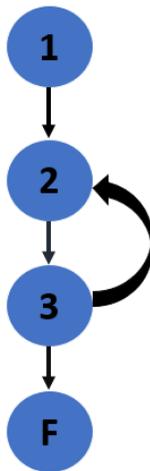
3.3.2 Pruebas unitarias

En programación, una prueba unitaria o test unitario es una forma efectiva de comprobar el correcto funcionamiento de las unidades individuales más pequeñas de los programas informáticos (42). Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños tests que validan el comportamiento de un objeto y la lógica. El unit testing suele realizarse durante la fase de desarrollo de aplicaciones de software. Normalmente las llevan a cabo los desarrolladores, aunque en la práctica, también pueden realizarlas los responsables de QA. (43)

Las pruebas unitarias se realizaron empleando la técnica del camino básico la cual permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de diseños de casos de prueba de caminos de ejecución que garantiza que durante esta se ejecuta por lo menos una vez cada sentencia del programa. Para obtener un conjunto de caminos independientes por los cuales puede circular el flujo de control se construye el Grafo de Flujo asociado y se calcula la complejidad ciclomática. (44)

A continuación, se muestra el proceso de pruebas realizado a la función limpiarSelect, que se muestra en la figura. Primeramente, se comienza por analizar el código y enumerar las instrucciones. Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir el grafo de flujo que se muestra a continuación:

```
69  
70 function limpiarSelect() {  
71   ❶ var $select = document.querySelector( selectors: '#SelAtributo');  
72   ❷ for (let i = $select.options.length; i >= 0; i--) {  
73     ❸ $select.remove(i);  
74   ❹ }  
75 }  
76
```



Seguidamente se calcula la complejidad en una de tres formas:

1. El número de regiones del gráfico de flujo corresponde a la complejidad ciclomática.

$$V(G) = \text{Regiones}$$

$$V(G) = 2$$

2. La complejidad ciclomática $V(G)$ para un gráfico de flujo G se define como $V(G) = E - N + 2$ donde “E” es el número de aristas del gráfico de flujo y “N” el número de nodos del gráfico de flujo.

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

$$V(G) = 4 - 4 + 2 = 2$$

3. La complejidad ciclomática $V(G)$ para un gráfico de flujo G también se define como $V(G) = P + 1$ donde "P" es el número de nodos predicados contenidos en el gráfico de flujo "G". (45)

$$V(G) = \text{Nodos predicados} + 1$$

$$V(G) = 1 + 1 = 2$$

Por lo que el camino básico sería:

Camino básico: 1-2-3-F

Luego se definen los casos de prueba para cada camino básico obtenido. A continuación, se muestra el resultado de las pruebas aplicadas en la siguiente tabla:

Tabla 10 Caso de prueba del método seleccionado

Descripción: se verificará que se limpie el contenido de los select quedando estos vacíos.		
Condición de ejecución	Entrada	Resultados Esperados
La variable \$select no debe estar vacía, para ellos se debe haber seleccionado previamente el campo tipo de Actividad que estará en correspondencia con los atributos que se muestren en el campo de selección.	\$select (campo de selección para la entrada de datos de los atributos)	Que se eliminen las opciones del campo de selección Atributo

Se calculó la complejidad ciclomática para los algoritmos del sistema. Los

resultados que se obtuvieron definen la cantidad de caminos independientes dentro del fragmento de código y la cantidad de casos de prueba diseñados. Las realizaciones de las pruebas unitarias permitieron obtener casos de prueba que garanticen que se ejerciten por lo menos una vez los caminos independientes de cada nivel de unidad de software y que ejerciten las estructuras internas de datos para asegurar su validez.

3.3.3 Pruebas funcionales

A través del método de caja negra se examinó si la herramienta funciona correctamente. La técnica empleada para realizar estas pruebas fue la partición de equivalencia, con el objetivo de detectar defectos en las funcionalidades y validar los requisitos de la propuesta de solución. A continuación, se muestra una tabla con el caso de prueba “Crear propuesta”.

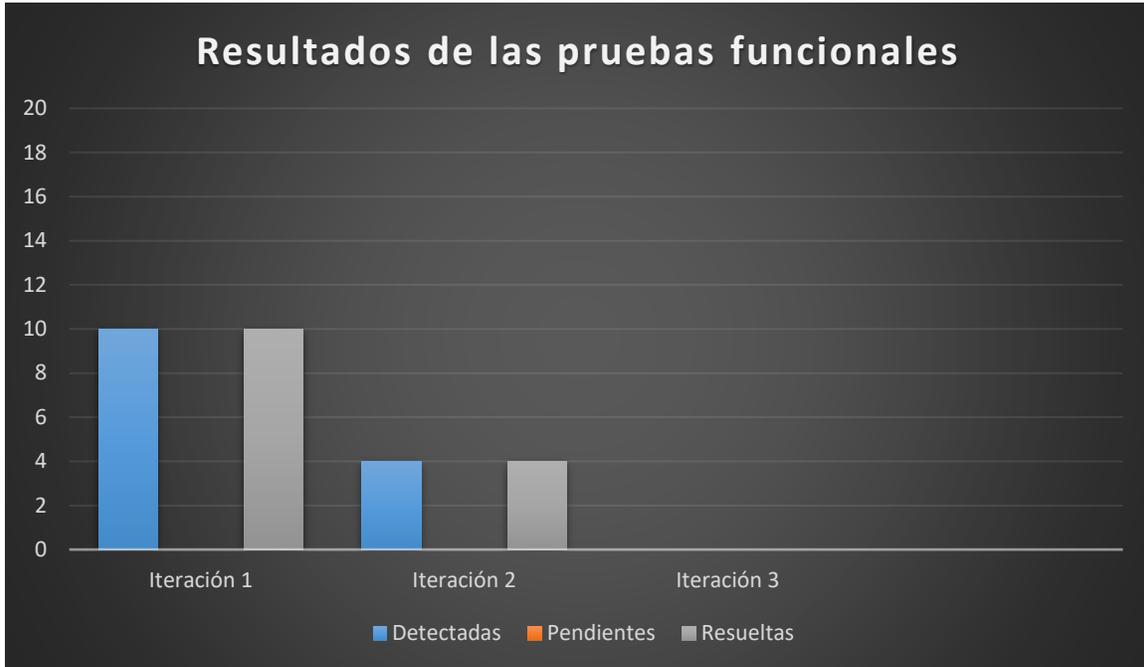
Tabla 11 Prueba funcional Crear propuesta

Escenario	Descripción	Requisito	Respuesta del sistema	Resultado de la prueba
EC 1.1 Crear propuesta	Al presionar el botón “Añadir propuesta” aparece un formulario donde se selecciona primeramente el premio y seguidamente se presiona el botón “Proponer Estudiante” que llena el campo de selección Estudiante con las	Válido El sistema crea una propuesta para el premio y estudiantes seleccionados con las observaciones.	El sistema guarda la información en la base de datos, muestra el mensaje de confirmación “Se ha añadido correctamente”, añade la propuesta al listado que se muestra debajo y oculta el	Satisfactoria

	personas que cumplen con los requisitos del premio, se llena el campo observaciones y se presiona el botón "Guardar".		formulario de crear propuesta.	
EC 1.2 Campos vacíos	Presiona el botón "Guardar", existiendo uno o más campos vacíos	Inválido	El sistema muestra un mensaje "Existen campos vacíos"	Satisfactorio. El sistema alertó al usuario de la existencia de campos vacíos.
EC 1.3 Existe estudiante ya propuesto	Presionar botón guardar con los campos previamente llenos	Inválido	El sistema muestra un mensaje "El estudiante ya ha sido propuesto para ese premio"	Satisfactorio. El sistema alertó al usuario de la existencia de un estudiante ya propuesto.

Se realizaron pruebas funcionales al sistema para comprobar la correcta implementación de los requisitos funcionales del mismo. Para asegurar que las no conformidades detectadas en cada iteración fueran corregidas, se aplicaron pruebas de regresión antes de pasar a la próxima iteración. Se llevaron a cabo tres iteraciones durante las pruebas del sistema. En las dos primeras iteraciones se detectaron diferentes no conformidades, las cuales se centraron en errores funcionales, ortográficos y de validaciones.

Tabla 12 No conformidades para una primera iteración de las pruebas funcionales



En la primera iteración estas pruebas se detectaron un total de 10 no conformidades realizadas a un total de 12 requisitos funcionales. En la segunda iteración se realizaron pruebas de regresión para verificar que las no conformidades de la iteración anterior estaban resueltas y se detectaron un total de 4 no conformidades más. En la tercera iteración se verificó nuevamente la existencia de nuevas no conformidades y los resultados fueron satisfactorios obteniéndose cero no conformidades. Las no conformidades encontradas se muestran en el Anexo 6.

Tabla 13 Tipos de no conformidades detectadas

Tipos de no conformidades	Primera iteración	Segunda iteración	Tercera iteración	Cantidad de no conformidades
Funcionales	3	2	0	5
Validaciones	2	0	0	2
Ortográficos	5	2	0	7
	10	4	0	Total = 14

Las pruebas funcionales fueron diseñadas con el objetivo que se estableció para la etapa de ejecución de estas pruebas. Fueron probados 34 requisitos funcionales, que representan un 100% del total definido, incluidos en la planificación para este nivel de pruebas.

3.4 Conclusiones parciales

En el presente capítulo, se abordaron elementos referentes a elementos de la implementación y prueba de la propuesta de solución y, después de analizar cada aspecto, se concluye que:

- La utilización de estándares de código para la implementación de la propuesta de solución posibilitó adoptar una estructura homogénea y la obtención de un código más comprensible y fácil de mantener.
- Las pruebas realizadas permitieron detectar los defectos presentes, corregirlos en el menor tiempo posible y entregar al cliente una aplicación con un adecuado nivel de calidad, obteniendo su aceptación.

Conclusiones generales

Al finalizar el desarrollo de la presente investigación, cumpliéndose los objetivos planteados se puede concluir que:

- El desarrollo del marco teórico de la investigación permitió identificar rasgos comunes entre sistemas informáticos con similitudes a la propuesta desarrollada con lo que se determinaron características funcionales a incluir en la solución.
- La metodología software XP permitió guiar el proceso de desarrollo de software generando artefactos que permitieron comprender el diseño del sistema.
- La utilización de las herramientas y tecnologías definidas, garantizó el éxito en el desarrollo del sistema, generando la documentación necesaria para ser utilizada como consulta en la realización de nuevas versiones del sistema.
- La realización de la estrategia de prueba, y la ejecución de las pruebas de unitarias y funcionales definidas en esta, permitió comprobar el correcto funcionamiento del sistema desarrollado evidenciando el cumplimiento del objetivo planteado

Recomendaciones:

Al término de esta investigación se recomienda:

Realizar el diseño del sistema de gestión de manera responsive, de manera que se pueda acceder al mismo desde cualquier dispositivo móvil.

Referencias bibliográficas

1. Tecnologías de la Información y las Comunicaciones (TIC). *Ministerio de Tecnologías de la Información y las Comunicaciones*. [En línea] [Citado el: 24 de Octubre de 2022.] <https://mintic.gov.co/portal/inicio/Glosario/T/5755:Tecnologias-de-la-Informacion-y-las-Comunicaciones-TIC>.
2. ¿Qué son las TICs y por qué son importantes? *Southern New Hampshire University*. [En línea] 10 de Noviembre de 2021. [Citado el: 19 de Mayo de 2022.] <https://es.snhu.edu/noticias/que-son-las-tics-y-por-que-son-importantes>.
3. Arianna. ¿Qué es un sistema de gestión y para qué sirve? *Proyectos Empezar*. [En línea] 24 de Mayo de 2021. <https://www.holded.com/es/blog/sistema-de-gestion#:~:text=En%20sentido%20estricto%2C%20un%20sistema,contabilidad%20y%20gesti%C3%B3n%20de%20proyectos..>
4. Guevara, Yurisander. Cuba cuenta con más de 7,5 millones de usuarios de internet. [En línea] 21 de Marzo de 2022. <https://www.juventudrebelde.cu/ciencia-tecnica/2022-03-21/cuba-cuenta-con-mas-de-7-5-millones-de-usuarios-de-internet>.
5. Márquez, Odalys Rosa Falcón. La UCI y la informatización de la sociedad cubana. [En línea] 21 de Junio de 2016. [Citado el: 09 de Mayo de 2022.] <https://www.uci.cu/la-uci-y-la-informatizacion-de-la-sociedad-cubana#:~:text=La%20Universidad%20de%20las%20Ciencias,la%20industria%20cubana%20del%20software..>
6. FEU. [En línea] Abril de 2016. [Citado el: 19 de Mayo de 2022.] <https://www.ujc.cu/es/feu>.
7. Federación Estudiantil Universitaria. *Facultad de Ciencias Médicas Manuel Fajardo*. [En línea] [Citado el: 21 de Mayo de 2022.] <https://instituciones.sld.cu/fcmfajardo/organizaciones-sindicato/organizaciones-feu/>.
8. Manicato. [En línea] [Citado el: 21 de Mayo de 2022.] <https://manicatocuba.site123.me/>.
9. Diccionario de la lengua española. [En línea] 2022. [Citado el: 21 de Mayo de 2022.] <https://dle.rae.es/premio>.
10. Bembibre, Cecilia. Premio. *Definición ABC*. [En línea] [Citado el: 21 de Mayo de

2022.] <https://www.definicionabc.com/general/premio.php>.

11. Pérez-Montoro, Mario. Gestión de la información. *glossariumBITri*. [En línea] 18 de 11 de 2009. [Citado el: 21 de Mayo de 2022.] <http://glossarium.bitrum.unileon.es/Home/gestion-de-la-informacion>.

12. L., Woodman. *Information management in large organizations*. London : s.n., 1985.

13. Paez, U. ¿Qué es la gestión de Información? *La Gestión de Información. Estudio del tema en los Congresos INFO. Tesis de Licenciatura. Cuba, Departamento de Bibliotecología y Ciencia de la Información, Universidad de La Habana*. 1990.

14. Davis, G. y Olsón. *Management Information Systems: Conceptual foundations, Structure and Development*. s.l. : 2a ed. Nueva York: McGrawhill, 1985.

15. Moreiro, G. *Introducción al estudio de la información y la documentación*. Medellín : Editorial de Antioquía., 1998.

16. Universidades, Santander. Metodologías de desarrollo de software: ¿qué son? [En línea] 21 de 12 de 2020. [Citado el: 21 de 04 de 2022.] <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.

17. ¿Qué es XP y cómo usarlo en el desarrollo de un proyecto? *Orientación Universia*. [En línea] 13 de Agosto de 2020. [Citado el: 22 de Mayo de 2022.] <https://orientacion.universia.edu.pe/infodetail/orientacion/consejos/que-es-xp-y-como-usarlo-en-el-desarrollo-de-un-proyecto-6157.html>.

18. METODOLOGÍA XP O PROGRAMACIÓN EXTREMA. *Sinnaps*. [En línea] 2020. [Citado el: 22 de Mayo de 2022.] <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>.

19. Muenta, Gabriela. Guía completa del Framework: qué es, cuáles tipos existen y por qué es importante en Internet. [En línea] Enero de 2020. [Citado el: 24 de Octubre de 2022.] <https://rockcontent.com/es/blog/framework/>.

20. BEDU, Staff. 3 razones para usar el framework Django. [En línea] 23 de 10 de 2020. [Citado el: 21 de 04 de 2022.] <https://bedu.org/blog/tecnologia/3-razones-para-usar-el-framework-django>.

21. Los 5 lenguajes de programación más utilizados en el mundo y para qué sirven. [En línea] 15 de Diciembre de 2021.

<https://blog.centrodelearning.com/2021/12/15/lenguajes-de-programacion-mas-utilizados/>.

22. Universidades, Santander. Python: qué es y por qué deberías aprender a utilizarlo. [En línea] 09 de Abril de 2021. <https://www.becas-santander.com/es/blog/python-que-es.html>.

23. ¿QUÉ ES HTML5? [En línea] [Citado el: 24 de Octubre de 2022.] <https://www.arkaitzgarro.com/html5/capitulo-1.html>.

24. CSS3. [En línea] 2022. [Citado el: 24 de Octubre de 2022.] <https://www.ecured.cu/CSS3>.

25. ¿Qué es un IDE en programación? *UNIR REVISTA*. [En línea] 06 de Julio de 2021. <https://www.unir.net/ingenieria/revista/ide-programacion/>.

26. Marín, Rafael. Los gestores de bases de datos más usados en la actualidad. [En línea] 16 de Abril de 2020. <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>.

27. Visual Paradigm. [En línea] 2022. https://www.ecured.cu/Visual_Paradigm.

28. VISUAL PARADIGM. [En línea] 2022. <https://prezi.com/j84ywyfdzvit/visual-paradigm/#:~:text=Visual%20Paradigm%20es%20una%20herramienta,desde%20diagramas%20y%20generar%20documentaci%C3%B3n..>

29. Vega, Miguel. ACTORES, ROLES E INTERACCIONES. *ACTORES DENTRO DEL DESARROLLO DE SOFTWARE*. [En línea] [Citado el: 25 de Octubre de 2022.] <http://fsi2013.blogspot.com/2013/12/actores-roles-e-interacciones.html>.

30. Requisitos de Software. *Ecured*. [En línea] 2022. [Citado el: 23 de Mayo de 2022.] https://www.ecured.cu/Requisitos_de_Software.

31. Especificación de Requisitos del Sistema. *Marco de Desarrollo de la Junta de Andalucía*. [En línea] [Citado el: 23 de Mayo de 2022.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>.

32. ¿Qué son los requisitos funcionales? Especificación, tipos, EJEMPLOS. *Ebooks Online*. [En línea] 2022. [Citado el: 23 de Mayo de 2022.] <https://ebooksonline.es/que-es-un-requisito-funcional-especificacion-tipos-ejemplos/>.

33. Sommerville, Ian. *Ingeniería de Software—Ian Sommerville 7ma Edicion.pdf*

(PDFy mirror). 2014.

34. ANAYA R., Tabares M. S., Arango F. *Una revisión de modelos y semánticas para la trazabilidad de requerimientos*. s.l. : Revista EIA.

35. Ferraro, Maria de los Angeles. *ESPECIFICACIÓN Y TRAZABILIDAD DE REQUERIMIENTOS EN EL DESARROLLO DE APLICACIONES WEB*.

36. Kent Beck, Ward Cunningham. *A Laboratory For Teaching Object-Oriented Thinking*.

37. Lucidchart. Qué es un modelo de base de datos. *¿Cuáles son tus necesidades de creación de diagramas de base de datos?* [En línea] [Citado el: 01 de Noviembre de 2022.] <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos>.

38. KeepCoding, Redacción. *¿Qué es un modelo de datos físicos?* [En línea] 27 de Septiembre de 2022. [Citado el: 01 de Noviembre de 2022.] <https://keepcoding.io/blog/modelo-de-datos-fisicos/>.

39. Somerville, Ian. *Ingeniería-de-software-ian-somerville-9-edicion*. [En línea] [Citado el: 1 de diciembre de 2022.] <https://baixardoc.com/documents/ingenieria-de-software-ian-somerville-9-edicion--5d191c3256fdf>.

40. El patrón de diseño MTV. *El libro de Django*. [En línea] 2022. [Citado el: 23 de Mayo de 2022.] <https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv>.

41. Quiroga, David. *PATRONES DE DISEÑO DE SOFTWARE*. [En línea] 18 de Septiembre de 2021. [Citado el: 23 de Mayo de 2022.] <https://clibre.io/blog/por-secciones/codigo/item/433-patrones-de-diseno-de-software>.

42. Carlos A. Guerrero, Johanna M. Suárez, Luz E. Gutiérrez. *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web*. *Información tecnológica*. [En línea] [Citado el: 27 de Mayo de 2022.] https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000300012.

43. Lago, Neybis. *Los patrones de diseño en el desarrollo de software*. *SAASRADAR*. [En línea] 24 de Febrero de 2022. [Citado el: 27 de Mayo de 2022.] <https://saasradar.net/patrones-diseno-de-software/>.

44. Diagrama de despliegue. [En línea] 2022. [Citado el: 4 de Octubre de 2022.] https://www.ecured.cu/Diagrama_de_despliegue.
45. La Guía Fácil de los Diagramas de Despliegue UML. [En línea] 22 de Septiembre de 2022. [Citado el: 4 de Octubre de 2022.] <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>.
46. ¿Cómo funcionan las pruebas de software? [En línea] [Citado el: 4 de Octubre de 2022.] <https://www.ibm.com/cl-es/topics/software-testing>.
47. Pressman, R. S. *Ingeniería de Software. Un Enfoque Práctico*. 2000. 810.
48. Unit testing: la prueba de calidad para software. [En línea] [Citado el: 3 de Octubre de 2022.] <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-papel-del-unit-test-en-el-desarrollo-de-software/>.
49. Motivos para realizar un test unitario. [En línea] [Citado el: 3 de Octubre de 2022.] <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>.
50. ¿Qué son las pruebas unitarias y para qué se emplean? . [En línea] 2022. [Citado el: 01 de Octubre de 2022.] <https://www.digite.com/es/agile/pruebas-unitarias//>.
51. Moreno, Oscar. La complejidad ciclomática y cómo simplificar tus desarrollos. [En línea] 10 de Septiembre de 2019. [Citado el: 01 de Octubre de 2022.] <http://oscarmoreno.com/la-complejidad-ciclomatica/>.
52. METODOLOGIAS AGILES “PROCESO UNIFICADO AGIL (AUP)”. *Ingeniería de Software*. [En línea] 2022. http://ingenieriadesoftware.mex.tl/63758_aup.html.
53. Agile Unified Process. *Ecured*. [En línea] [Citado el: 21 de 04 de 2022.] https://www.ecured.cu/Agile_Unified_Process.
54. 11 Mejores Editores de Código de Python o entornos de desarrollo IDE. [En línea] 01 de Enero de 2022. <https://ciberninjas.com/mejores-ide-python/>.
55. White. *Intelligence management en: Cronin,B. Information management: from strategies to action*. London : s.n., 1985.
56. Calvo, Diego. Metodología XP Programación Extrema (Metodología ágil). [En línea] 07 de Abril de 2018. [Citado el: 22 de Mayo de 2022.] <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>.
57. Modelo de Dominio. [En línea] 2022. [Citado el: 23 de Mayo de 2022.]

https://hmong.es/wiki/Domain_model.

58. Modelo de dominio. *Ecured*. [En línea] 2022. [Citado el: 23 de Mayo de 2022.]

https://www.ecured.cu/Modelo_de_dominio.

59. Requerimientos Funcionales y No Funcionales, ejemplos y tips. [En línea] 2018 de Abril de 2018. [Citado el: 23 de Mayo de 2022.]

<https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>.

60. El Modelo base de datos: Definición y tipos. [En línea] [Citado el: 23 de Mayo de 2022.] <https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/>.

61. Leiva, Antonio. Patrones de diseño de software. *Curso Arquitecturas Gratis Ingeniería de software*. [En línea] 2021. [Citado el: 27 de Mayo de 2022.]

<https://devexperto.com/patrones-de-diseno-software/>.

62. Tarancón, Yixander Yero. Xabal Excriba. Gestor de Documentos Administrativos. [En línea] 2015. [Citado el: 3 de Octubre de 2022.]

<http://excriba.prod.uci.cu/proxy/alfresco/api/node/content/workspace/SpacesStore/a622adab-eac5-4fb3-ba08->

63. *Metodología de desarrollo para la Actividad productiva de la UCI*. Sánchez, Tamara Rodríguez. 2015.

64. Niveles de prueba de software. [En línea] 2022. [Citado el: 3 de Octubre de 2022.]

https://www.ecured.cu/Niveles_de_prueba_de_software.

65. Calikli, Gul, Uzundag, Berna A. y Bener, Ayse. Confirmation Bias in Software Development and Testing: An Analysis of the Effects of Company Size, Experience and Reasoning Skills. [En línea] [Citado el: 3 de Octubre de 2022.]

https://www.researchgate.net/publication/235430372_Confirmation_Bias_in_Software_Development_and_Testing_An_Analysis_of_the_Effects_of_Company_Size_Experience_and_Reasoning_Skills.

66. Freepik. Pruebas de aceptación de software según el ISTQB. [En línea] [Citado el: 01 de Noviembre de 2022.] <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>.

Anexos

Anexo 1: Encuesta aplicada a estudiantes de la UCI

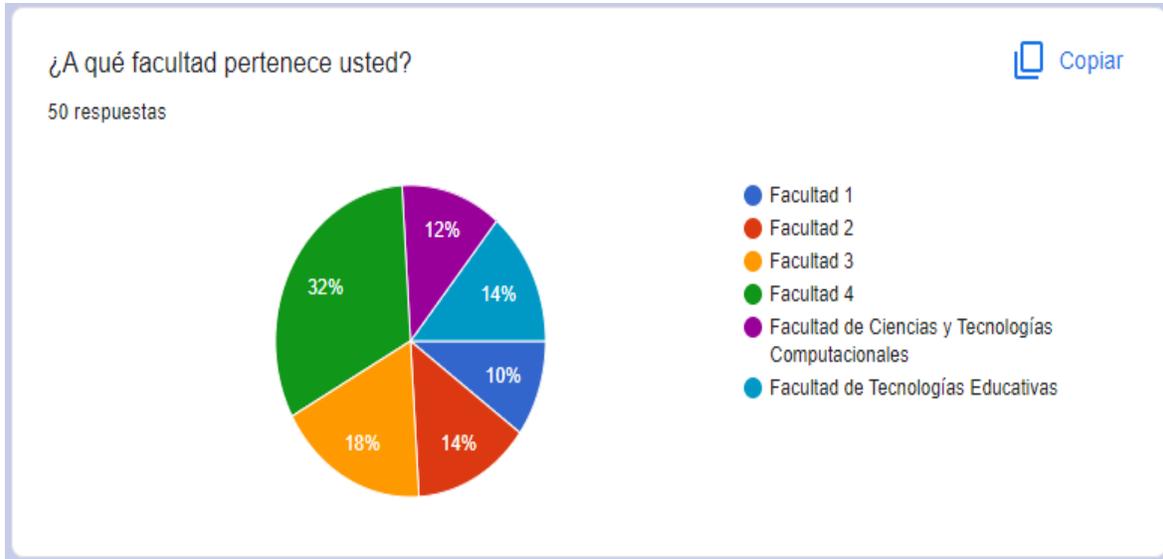


Figura 3. Respuesta de encuestados correspondiente a la pregunta 1. Fuente: Elaboración propia

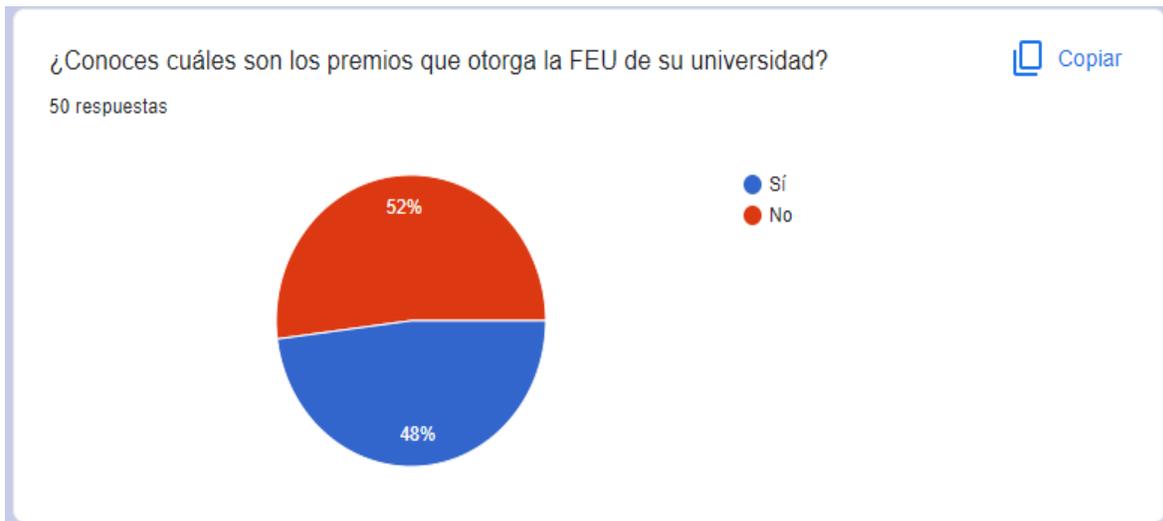


Figura 4. Respuesta de encuestados correspondiente a la pregunta 2. Fuente: Elaboración propia.

¿Conoces quiénes han sido ganadores de alguna condecoración en tu facultad?

[Copiar](#)

24 respuestas

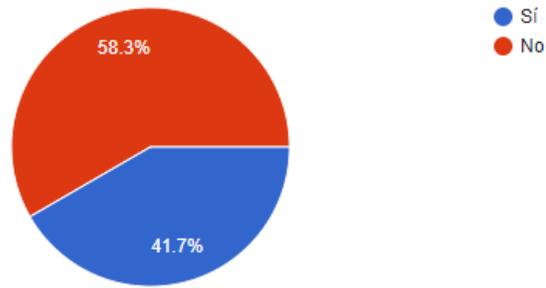


Figura 5. Respuesta de encuestados correspondiente a la pregunta 3. Fuente: Elaboración propia

¿Estás interesado en obtener algún premio que otorga la FEU?

[Copiar](#)

24 respuestas

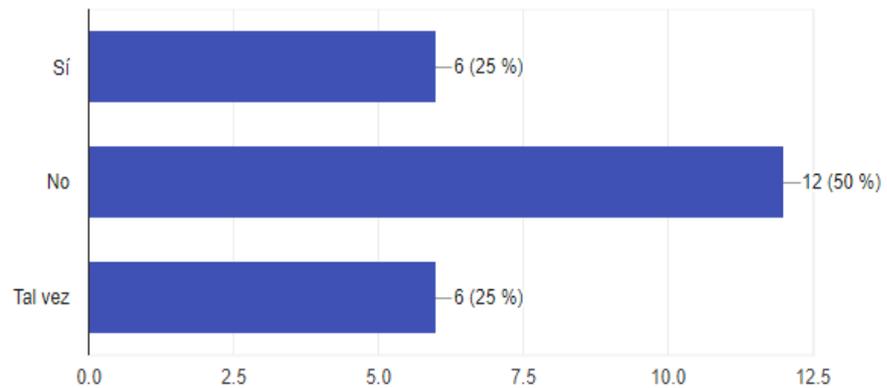


Figura 6. Respuesta de encuestados correspondiente a la pregunta 4. Fuente: Elaboración propia

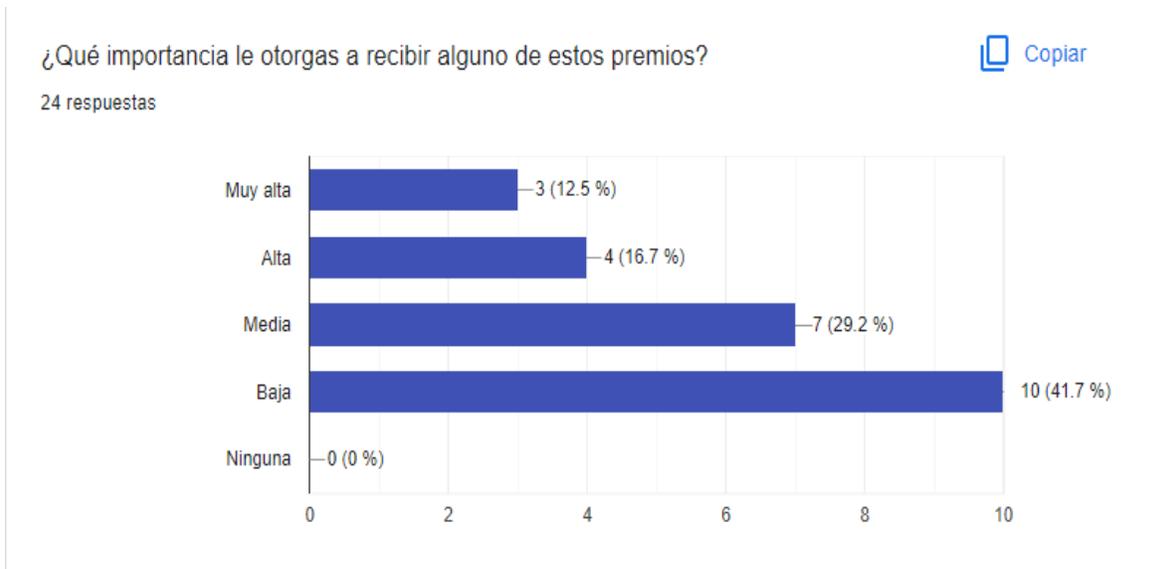


Figura 7. Respuesta de encuestados correspondiente a la pregunta 5. Fuente: Elaboración propia

Anexo 2: Historias de usuario

Historia de usuario	
Número: RF1	Nombre del requisito: Autenticar usuario
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días
<p>Descripción: El sistema debe permitir que los usuarios ingresen los datos de usuario y contraseña al sistema para poder acceder a las funcionalidades de acuerdo al rol que presenten.</p> <p>Objetivo: Permitir acceder a las funcionalidades que ofrece el sistema de acuerdo al rol del usuario.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para autenticarse al sistema el usuario deberá: - Tener en cuenta los siguientes datos: Usuario, Contraseña</p>	

Comportamientos válidos y no válidos (flujo central y alternos):

Los campos son obligatorios.

Usuario: Campo de texto

Contraseña: Campo de texto

Flujo de la acción a realizar:

- El sistema debe permitir ingresar el usuario y la contraseña para acceder al sistema.
- Cuando el usuario ingresa los datos correctamente para acceder al sistema y presiona el botón Iniciar sesión, se redirecciona a la página de Servicios.
- Si los datos introducidos son incorrectos el sistema muestra un mensaje de error notificando que se introduzcan los campos correctos.

Observación:

Historia de usuario

Número: RF2

Nombre del requisito: Listar usuario

Programador: Jessica C. Martínez

Iteración asignada: 1

Prioridad en negocio: Baja

Riesgo en desarrollo: N/A

Tiempo estimado: 1 día

Tiempo Real: 1 día

Descripción: El sistema debe permitir listar los usuarios registrados en el sistema.

Objetivo: Permitir listar los datos de los usuarios en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

Para añadir un nuevo usuario hay que:

- Estar autenticado en el sistema con el rol Administrador.
- Debe existir al menos un usuario en el sistema.

Flujo de la acción a realizar:

- Una vez seleccionado la opción Usuarios en el panel de Administrador, aparece un listado de los usuarios registrados en el sistema. En el listado se brinda la posibilidad de modificar los datos del usuario y de eliminar uno o varios usuarios.

Observación:

Historia de usuario

Número: RF3	Nombre del requisito: Crear usuario
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días

Descripción: El sistema debe permitir que los usuarios se registren en el sistema. También da la opción a los usuarios con rol administrador añadir usuarios.

Objetivo: Permitir añadir los datos de los usuarios en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

Para añadir un nuevo usuario hay que:

- Verificar que el usuario no exista en el sistema
- Tener en cuenta los siguientes datos: Nombre, Apellidos, Correo, Usuario, Contraseña.
- Estar autenticado en el sistema con el rol Administrador.
- Debe existir al menos un usuario en el sistema.

Comportamientos válidos y no válidos (flujo central y alternos):

Los campos son obligatorios

Nombre: Campo de texto

Apellidos: Campo de texto.

Correo: Campo de texto

<p>Usuario: Campo de texto</p> <p>Contraseña: Campo de texto</p> <p>Flujo de la acción a realizar:</p> <p>-Una vez seleccionada la opción Registrase, el sistema debe permitir introducir los datos para crear un nuevo usuario en el sistema.</p> <p>- Cuando el usuario introduce correctamente los datos necesarios para registrarse y selecciona la opción Registrar, se envía un código de verificación al correo, y una vez verifique que el correo es válido, se adiciona un nuevo elemento en el listado de usuarios y el sistema muestra un mensaje de información.</p> <p>- Si selecciona la opción Ya estoy registrado regresará a la vista previa.</p>
<p>Observación:</p>

Historia de usuario	
Número: RF4	Nombre del requisito: Modificar usuario
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días
<p>Descripción: El sistema debe permitir que los usuarios con rol administrador modifiquen usuarios.</p> <p>Objetivo: Permitir modificar los datos de los usuarios en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para modificar un usuario hay que:</p> <ul style="list-style-type: none"> - Tener en cuenta los siguientes datos: Nombre, Apellidos, Correo, Usuario, Contraseña. - Estar autenticado en el sistema con el rol Administrador. 	

- Debe existir al menos un usuario en el sistema.

Comportamientos válidos y no válidos (flujo central y alternos):

El campo usuario y contraseña son obligatorios

Nombre: Campo de texto

Apellidos: Campo de texto.

Correo: Campo de texto

Usuario: Campo de texto que debe ser único

Contraseña: Campo de texto

Flujo de la acción a realizar:

-Una vez seleccionado el usuario, el sistema debe permitir introducir los datos para modificar un usuario en el sistema.

- Cuando el usuario introduce correctamente los datos necesarios para modificar y selecciona la opción Guardar, se realiza el cambio en el listado de usuarios y el sistema muestra un mensaje de confirmación.

- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Historia de usuario	
Número: RF5	Nombre del requisito: Eliminar usuario
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días
Descripción: El sistema debe permitir que los usuarios con rol administrador eliminen uno o varios usuarios.	
Objetivo: Permitir eliminar usuarios en el sistema.	

Acciones para lograr el objetivo (precondiciones y datos):

Para modificar un usuario hay que:

- Estar autenticado en el sistema con el rol Administrador.
- Debe existir al menos un usuario en el sistema.

Flujo de la acción a realizar:

- El sistema permite eliminar uno o varios usuarios, para eliminar varios usuarios, el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción Eliminar de las acciones y se muestra un mensaje para completar la acción.

- Si el usuario selecciona la opción "Sí, estoy seguro", se elimina el o los usuarios seleccionados, se actualiza el listado de usuarios y el sistema muestra un mensaje de confirmación.

- Si selecciona la opción No regresará a la vista previa.

Observación:

Historia de usuario

Número: RF6

Nombre del requisito: Listar estudiantes.

Programador: Jessica C. Martínez

Iteración asignada: 1

Prioridad en negocio: Baja

Riesgo en desarrollo: N/A

Tiempo estimado: 1 día

Tiempo Real: 1 día

Descripción: El sistema debe permitir listar los estudiantes registrados en el sistema de la brigada en la que se encuentre el usuario.

Objetivo: Permitir listar los datos de los estudiantes de la brigada en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

Para añadir un nuevo usuario hay que:

- Estar autenticado en el sistema con el rol Jefe de Grupo.
- Debe existir al menos un usuario en el sistema.

Flujo de la acción a realizar:

- Una vez seleccionado la opción Estudiantes, aparece un listado de los estudiantes de la brigada registrados en el sistema. En el listado se brinda la posibilidad de añadir actividades a los estudiantes.

Observación:

Historia de usuario

Número: RF7	Nombre del requisito: Crear estudiante
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días

Descripción: El sistema debe permitir a los usuarios con rol administrador añadir estudiantes.

Objetivo: Permitir añadir los datos de los estudiantes en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

Para añadir un nuevo estudiante hay que:

- Tener en cuenta los siguientes datos: Nombre, Primer Apellido, Segundo Apellido, Brigada, Facultad, Curso, Carrera, Imagen.
- Estar autenticado en el sistema con el rol Administrador.
- Debe existir al menos un usuario en el sistema.

Comportamientos válidos y no válidos (flujo central y alternos):

Nombre: Campo de texto

Primer Apellido: Campo de texto

Segundo Apellido: Campo de texto

Brigada: Campo de selección

Facultad: Campo de selección

Curso: Campo de selección

Carrera: Campo de texto

Imagen: Campo de tipo archivo

Flujo de la acción a realizar:

-Una vez seleccionada la opción Estudiante, el sistema debe permitir introducir los datos para crear un nuevo estudiante en el sistema.

- Cuando el usuario introduce correctamente los datos necesarios para añadir un nuevo estudiante y selecciona la opción Guardar, se adiciona un nuevo elemento en el listado de estudiantes y el sistema muestra un mensaje de confirmación.

- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Historia de usuario	
Número: RF8	Nombre del requisito: Modificar estudiante
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días
Descripción: El sistema debe permitir que los usuarios con rol administrador modifiquen estudiantes.	
Objetivo: Permitir modificar los datos de los estudiantes en el sistema.	
Acciones para lograr el objetivo (precondiciones y datos):	

Para modificar un estudiante hay que:

- Tener en cuenta los siguientes datos: Nombre, Primer Apellido, Segundo Apellido, Brigada, Facultad, Curso, Carrera, Imagen.
- Estar autenticado en el sistema con el rol Administrador.

Comportamientos válidos y no válidos (flujo central y alternos):

Nombre: Campo de texto

Primer Apellido: Campo de texto

Segundo Apellido: Campo de texto

Brigada: Campo de selección

Facultad: Campo de selección

Curso: Campo de selección

Carrera: Campo de texto

Imagen: Campo de tipo archivo

Flujo de la acción a realizar:

-Una vez seleccionado el estudiante, el sistema debe permitir introducir los datos para modificar un estudiante en el sistema.

- Cuando el usuario introduce correctamente los datos necesarios para modificar y selecciona la opción Guardar, se realiza el cambio en el listado de estudiantes y el sistema muestra un mensaje de confirmación.

- Si selecciona la opción Cancelar regresará a la vista previa.

Observación:

Historia de usuario	
Número: RF9	Nombre del requisito: Eliminar estudiante
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días

<p>Descripción: El sistema debe permitir que los usuarios con rol administrador eliminen uno o varios estudiantes.</p> <p>Objetivo: Permitir eliminar estudiantes en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para eliminar un estudiante hay que:</p> <ul style="list-style-type: none">- Estar autenticado en el sistema con el rol Administrador.- Debe existir al menos un estudiante en el sistema. <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">- El sistema permite eliminar uno o varios estudiantes, para eliminar varios estudiantes, el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción Eliminar de las acciones y se muestra un mensaje para completar la acción.- Si el usuario selecciona la opción "Sí, estoy seguro", se elimina el o los estudiantes seleccionados, se actualiza el listado de estudiantes y el sistema muestra un mensaje de confirmación.- Si selecciona la opción No regresará a la vista previa.	
Observación:	

Historia de usuario	
Número: RF10	Nombre del requisito: Importar listado de estudiantes
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día

Descripción: El sistema debe permitir que los usuarios con rol administrador importen un listado de estudiantes.

Objetivo: Permitir importar un listado con los datos de los estudiantes en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

Para importar un listado de estudiantes hay que:

- Tener en cuenta los siguientes datos: Fichero a importar, Formato.
- Estar autenticado en el sistema con el rol Administrador.

Comportamientos válidos y no válidos (flujo central y alternos):

Formato: Campo de selección con las opciones csv, xls,xlsx, tsv, json, yaml

Fichero a importar: Campo de tipo archivo

Flujo de la acción a realizar:

-Una vez seleccionada la opción de Importar, el sistema debe permitir importar el listado con los datos de los estudiantes en el sistema.

- Cuando el usuario introduce correctamente los datos necesarios para importar y selecciona la opción Enviar, se realiza el cambio en el listado de estudiantes y el sistema muestra un mensaje de confirmación.

Observación:

Historia de usuario	
Número: RF11	Nombre del requisito: Exportar listado de estudiantes
Programador: Jessica C. Martínez	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día
Descripción: El sistema debe permitir que los usuarios con rol administrador exporten un listado de estudiantes.	

Objetivo: Permitir exportar un listado con los datos de los estudiantes.

Acciones para lograr el objetivo (precondiciones y datos):

Para exportar un listado de estudiantes hay que:

- Tener en cuenta los siguientes datos: Formato.
- Estar autenticado en el sistema con el rol Administrador.

Comportamientos válidos y no válidos (flujo central y alternos):

Formato: Campo de selección con las opciones csv, xls,xlsx, tsv, json, yaml, ods, html

Flujo de la acción a realizar:

-Una vez seleccionada la opción de Exportar, el sistema debe permitir exportar el listado con los datos de los estudiantes.

- Cuando el usuario introduce correctamente los datos necesarios para exportar y selecciona la opción Enviar, se exporta el listado de estudiantes en el formato especificado y el sistema muestra un mensaje de confirmación.

Observación:

Historia de usuario	
Número: RF12	Nombre del requisito: Agregar actividad a estudiante
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo puedan añadir actividades a un estudiante.</p> <p>Objetivo: Permitir añadir nuevas actividades a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos):</p>	

Para incluir una actividad hay que:

- Tener en cuenta los siguientes datos: Tipo de Actividad, Fecha, Observaciones.
- Estar autenticado en el sistema con el rol jefe de grupo

Comportamientos válidos y no válidos (flujo central y alternos):

El campo tipo de actividad y fecha son obligatorios

Tipo de Actividad: Campo de selección con las opciones: Cultural, Deportiva, Proyectos, Eventos, Ayudantía, Residencia, Comisión Disciplinaria.

Fecha: Campo de texto.

Observaciones: Campo de texto.

Flujo de la acción a realizar:

- El sistema debe permitir introducir los datos para incluir una nueva actividad.
- Cuando el usuario introduce correctamente los datos necesarios para incluir una actividad y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de confirmación.

Observación: Cuando se selecciona el tipo de actividad, en correspondencia con este se muestra un formulario con los datos de esa actividad determinada.

Historia de usuario	
Número: RF13	Nombre del requisito: Eliminar actividad del estudiante
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días
Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo eliminen una actividad de los estudiantes.	
Objetivo: Permitir eliminar actividades del estudiante en el sistema.	

Acciones para lograr el objetivo (precondiciones y datos):

Para eliminar una actividad del estudiante hay que:

- Estar autenticado en el sistema con el rol jefe de grupo.
- Debe existir al menos una actividad en el sistema.

Flujo de la acción a realizar:

- El sistema permite eliminar una actividad del estudiante si se selecciona la opción Eliminar y se muestra un mensaje para completar la acción.
- Si el usuario selecciona la opción "Sí, estoy seguro", se elimina la actividad seleccionada, se actualiza el listado de actividades y el sistema muestra un mensaje de confirmación.
- Si selecciona la opción Cancelar se regresará a la vista previa.

Observación:

Historia de usuario

Número: RF14

Nombre del requisito: Listar actividades del estudiante.

Programador: Jessica C. Martínez

Iteración asignada: 2

Prioridad en negocio: Baja

Riesgo en desarrollo: N/A

Tiempo estimado: 1 día

Tiempo Real: 1 día

Descripción: El sistema debe permitir listar las actividades del estudiante en el sistema de la brigada en la que se encuentre el usuario con rol jefe de grupo.

Objetivo: Permitir listar las actividades del estudiante en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

Para listar las actividades hay que:

- Estar autenticado en el sistema con el rol Jefe de Grupo.
- Debe existir al menos un estudiante en el sistema.

Flujo de la acción a realizar:

- Una vez seleccionado la opción Añadir Actividad en el listado de estudiantes, aparece un listado de las actividades del estudiante seleccionado. En el listado se brinda la posibilidad de eliminar actividades de los estudiantes.

Observación:

Historia de usuario	
Número: RF23	Nombre del requisito: Crear actividad Proyectos
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo proyectos.</p> <p>Objetivo: Permitir incluir nuevas actividades de tipo proyectos a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para incluir una actividad proyectos hay que:</p> <ul style="list-style-type: none">- Tener en cuenta los siguientes datos: Nombre, Evaluación.- Estar autenticado en el sistema con el rol jefe de grupo <p>Comportamientos válidos y no válidos (flujo central y alternos): El campo nombre y evaluación son obligatorios Evaluación: Campo de selección con las opciones: B, R, M. Nombre: Campo de texto.</p>	

Flujo de la acción a realizar:

- El sistema debe permitir seleccionar los datos para incluir una nueva actividad de tipo proyectos.
- Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad de tipo proyectos y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación: Las actividades de tipo proyectos son para tener un registro de las actividades en proyectos en las que participa cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.

Historia de usuario	
Número: RF24	Nombre del requisito: Crear actividad Ayudantía
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo ayudantía.</p> <p>Objetivo: Permitir incluir nuevas actividades de tipo ayudantía a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para incluir una actividad ayudantía hay que:</p> <ul style="list-style-type: none">- Tener en cuenta los siguientes datos: Asignatura, Evaluación.	

- Estar autenticado en el sistema con el rol jefe de grupo

Comportamientos válidos y no válidos (flujo central y alternos):

El campo asignatura y evaluación son obligatorios

Evaluación: Campo de selección con las opciones: B, R, M.

Asignatura: Campo de texto.

Flujo de la acción a realizar:

- El sistema debe permitir seleccionar los datos para incluir una nueva actividad de tipo ayudantía.

- Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad de tipo ayudantía y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información.

- Si selecciona la opción Cancelar regresará a la vista previa.

Observación: Las actividades de tipo ayudantía son para tener un registro de las actividades relacionadas con el movimiento de alumnos ayudantes en las que participa cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.

Historia de usuario	
Número: RF25	Nombre del requisito: Crear actividad Deportiva
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días

<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo deportiva.</p> <p>Objetivo: Permitir incluir nuevas actividades de tipo deportiva a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para incluir una actividad deportiva hay que:</p> <ul style="list-style-type: none">- Tener en cuenta los siguientes datos: Tipo, Deporte, Resultado- Estar autenticado en el sistema con el rol jefe de grupo <p>Comportamientos válidos y no válidos (flujo central y alternos): El campo resultado, tipo y deporte son obligatorios Resultado: Campo de selección con las opciones: Oro, Plata, Bronce, Ninguno. Tipo: Campo de selección con las opciones: Inter-años, Inter-facultad, universitarios, provinciales, provinciales, nacionales. Deporte: Campo de texto.</p> <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">- El sistema debe permitir seleccionar los datos para incluir una nueva actividad de tipo deportiva.- Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad de tipo deportiva y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información.- Si selecciona la opción Cancelar regresará a la vista previa.	
<p>Observación: Las actividades de tipo deportiva son para tener un registro de las actividades deportivas en las que participa cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.</p>	

Historia de usuario	
Número: RF26	Nombre del requisito: Crear actividad Cultural
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo cultural.</p> <p>Objetivo: Permitir incluir nuevas actividades de tipo cultural a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para incluir una actividad cultural hay que: - Tener en cuenta los siguientes datos: Nivel, Resultado, Nombre de la Obra, Manifestación - Estar autenticado en el sistema con el rol jefe de grupo</p> <p>Comportamientos válidos y no válidos (flujo central y alternos): El campo resultado, nivel, manifestación y nombre son obligatorios Resultado: Campo de selección con las opciones: Oro, Plata, Bronce, Mención, Ninguno. Nivel: Campo de selección con las opciones: Facultad, Universidad, Provincial, Nacional. Manifestación: Campo de selección con las opciones: Literatura, Artes Plásticas, Audiovisual, Danza, Teatro. Nombre de Obra: Campo de texto.</p> <p>Flujo de la acción a realizar:</p>	

- El sistema debe permitir seleccionar los datos para incluir una nueva actividad de tipo cultural.
- Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad de tipo cultural y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación: Las actividades de tipo cultural son para tener un registro de las actividades culturales en las que participa cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.

Historia de usuario	
Número: RF27	Nombre del requisito: Crear actividad Residencia
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo residencia.</p> <p>Objetivo: Permitir incluir nuevas actividades de tipo eventos a los estudiantes.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para incluir una actividad eventos hay que:</p> <ul style="list-style-type: none"> - Tener en cuenta los siguientes datos: Tipo, Evaluación. - Estar autenticado en el sistema con el rol jefe de grupo 	

Comportamientos válidos y no válidos (flujo central y alternos):

El campo tipo y evaluación son obligatorios

Tipo: Campo de selección con las opciones: Cuartelería, Guardia estudiantil, TSU.

Evaluación: Campo de selección con las opciones: B, R, M.

Flujo de la acción a realizar:

- El sistema debe permitir seleccionar los datos para incluir una nueva actividad de tipo residencia.
- Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad de tipo residencia y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación: Las actividades de tipo residencia son para tener un registro de las actividades de la residencia en los que participa cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.

Historia de usuario

Número: RF29	Nombre del requisito: Crear actividad Eventos
Programador: Jessica C. Martínez	Iteración asignada: 2
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días

Descripción: El sistema debe permitir que los usuarios con rol jefe de grupo creen nuevas actividades de tipo eventos.

Objetivo: Permitir incluir nuevas actividades de tipo eventos a los estudiantes.

Acciones para lograr el objetivo (precondiciones y datos):

Para incluir una actividad eventos hay que:

- Tener en cuenta los siguientes datos: Resultado, Nivel, Nombre
- Estar autenticado en el sistema con el rol jefe de grupo

Comportamientos válidos y no válidos (flujo central y alternos):

El campo resultado, nivel y nombre son obligatorios

Resultado: Campo de selección con las opciones: Relevante, Destacado, Mención, Ninguno.

Nivel: Campo de selección con las opciones: Facultad, Universidad, Provincial, Nacional, Internacional.

Nombre: Campo de texto.

Flujo de la acción a realizar:

- El sistema debe permitir seleccionar los datos para incluir una nueva actividad de tipo eventos.
- Cuando el usuario selecciona correctamente los datos necesarios para incluir una actividad de tipo eventos y selecciona la opción Guardar, se crea un nuevo elemento en el listado de actividades y el sistema muestra un mensaje de información.
- Si selecciona la opción Cancelar regresará a la vista previa.

Observación: Las actividades de tipo eventos son para tener un registro de los eventos en los que participa cada estudiante, información necesaria que se incluye dentro del registro de actividades y que se toma en cuenta en la creación de los requisitos de determinado premio con el fin de comprobar que el estudiante cumple con los requerimientos de un premio.

Historia de usuario

Número: RF15	Nombre del requisito: Listar premio.
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Baja	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día
Descripción: El sistema debe permitir listar premios a los usuarios con rol Usuario de Sistema. Objetivo: Permitir listar premios en el sistema. Acciones para lograr el objetivo (precondiciones y datos): Para listar premios hay que: <ul style="list-style-type: none">- Estar autenticado en el sistema con el rol Usuario de Sistema. Flujo de la acción a realizar: - Una vez seleccionado la opción Premios, aparece un listado de los premios. En el listado se brinda la posibilidad de añadir requisitos, de editar y de eliminar.	
Observación:	

Historia de usuario	
Número: RF16	Nombre del requisito: Crear premio
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
Descripción: El sistema debe permitir que los usuarios con rol usuario de sistema creen nuevos premios. Objetivo: Permitir incluir nuevos premios.	

Acciones para lograr el objetivo (precondiciones y datos):

Para incluir un premio hay que:

- Tener en cuenta los siguientes datos: Nombre, Descripción, Fecha de Premiación, Nivel, Lugar del Nivel, Curso.
- Estar autenticado en el sistema con el rol usuario de sistema.

Comportamientos válidos y no válidos (flujo central y alternos):

Los campos son obligatorios

Lugar del Nivel: Campo de selección con las opciones: Facultad 1, Facultad 2, Facultad 3, Facultad 4, FTE, CITED, Universidad.

Nivel: Campo de selección con las opciones: Facultad, Universidad.

Curso: Campo de selección.

Nombre: Campo de texto.

Descripción: Campo de texto.

Fecha de premiación: Campo de texto.

Flujo de la acción a realizar:

- El sistema debe permitir introducir los datos para incluir un nuevo premio en el sistema.
- Cuando el usuario selecciona la opción Nuevo premio se muestra un formulario, introduce correctamente los datos necesarios para incluir un premio y selecciona la opción Guardar, se crea un nuevo elemento en el listado de premios, se oculta el formulario y el sistema muestra un mensaje de confirmación.
- Si selecciona la opción Cancelar se oculta el formulario de premios.

Observación:

Historia de usuario	
Número: RF17	Nombre del requisito: Modificar premio
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Media	Riesgo en desarrollo: N/A

Tiempo estimado: 2 días	Tiempo Real: 2 días
<p>Descripción: El sistema debe permitir que los usuarios con rol usuario de sistema modifiquen premios.</p> <p>Objetivo: Permitir modificar los datos de los premios en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para modificar un premio hay que:</p> <ul style="list-style-type: none">- Tener en cuenta los siguientes datos: Nombre, Descripción, Fecha de Premiación, Nivel, Lugar del Nivel, Curso.- Estar autenticado en el sistema con el rol Usuario de Sistema. <p>Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>Los campos son obligatorios</p> <p>Lugar del Nivel: Campo de selección con las opciones: Facultad 1, Facultad 2, Facultad 3, Facultad 4, FTE, CITED, Universidad.</p> <p>Nivel: Campo de selección con las opciones: Facultad, Universidad.</p> <p>Curso: Campo de selección.</p> <p>Nombre: Campo de texto.</p> <p>Descripción: Campo de texto.</p> <p>Fecha de premiación: Campo de texto.</p> <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">-Una vez seleccionado el premio, el sistema debe permitir introducir los datos para modificar un premio en el sistema.- Cuando el usuario selecciona la opción Editar en el listado de premios, introduce correctamente los datos necesarios para modificar y selecciona la opción Guardar, se realiza el cambio en el listado de premios y el sistema muestra un mensaje de confirmación.- Si selecciona la opción Cancelar se oculta el formulario.	
Observación:	

Historia de usuario	
Número: RF18	Nombre del requisito: Eliminar premio
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días
<p>Descripción: El sistema debe permitir que los usuarios con rol usuario de sistema eliminen un premio.</p> <p>Objetivo: Permitir eliminar premios en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para eliminar un premio hay que:</p> <ul style="list-style-type: none">- Estar autenticado en el sistema con el rol usuario de sistema.- Debe existir al menos un premio en el sistema. <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">- El sistema permite eliminar un premio si se selecciona la opción Eliminar y se muestra un mensaje para completar la acción.- Si el usuario selecciona la opción "Sí, estoy seguro", se elimina el premio seleccionado, se actualiza el listado de premio y el sistema muestra un mensaje de confirmación.- Si selecciona la opción Cancelar regresará a la vista previa.	
Observación:	

Historia de usuario	
Número: RF19	Nombre del requisito: Mostrar datos de premio
Programador: Jessica C. Martínez	Iteración asignada: 3

Prioridad en negocio: Baja	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día
Descripción: El sistema debe permitir que los usuarios visualicen la información de un premio. Objetivo: Permitir mostrar la información de un premio. Acciones para lograr el objetivo (precondiciones y datos): Para mostrar la información de un premio hay que: <ul style="list-style-type: none">- Estar autenticado en el sistema con el rol usuario.- Debe existir al menos un premio en el sistema. Flujo de la acción a realizar: <ul style="list-style-type: none">- El sistema permite mostrar la información de un premio si se selecciona dicho premio en el apartado donde se realizan las propuestas.	
Observación:	

Historia de usuario	
Número: RF20	Nombre del requisito: Agregar requisito a premio.
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
Descripción: El sistema debe permitir que los usuarios con rol usuario de sistema puedan añadir requisitos a un premio. Objetivo: Permitir añadir nuevos requisitos a los premios.	

Acciones para lograr el objetivo (precondiciones y datos):

Para incluir un requisito hay que:

- Tener en cuenta los siguientes datos: Tipo de Actividad, Opción, Operación, Valor, Atributo.
- Estar autenticado en el sistema con el rol usuario de sistema.

Comportamientos válidos y no válidos (flujo central y alternos):

Los campos son obligatorios

Tipo de Actividad: Campo de selección con las opciones: Cultural, Deportiva, Proyectos, Eventos, Ayudantía, Residencia, Comisión Disciplinaria.

Opción: Campo de selección con las opciones: Cantidad, Campo.

Operación: Campo de selección con las opciones: igual que, mayor que, menor que.

Atributo: Campo de selección.

Valor: Campo de selección o de texto que se corresponde en función al atributo seleccionado.

Flujo de la acción a realizar:

- El sistema debe permitir introducir los datos para incluir un nuevo requisito.
- Cuando el usuario introduce correctamente los datos necesarios para incluir un premio y selecciona la opción Guardar, se crea un nuevo elemento en el listado de requisitos y el sistema muestra un mensaje de confirmación.

Observación: Primero se seleccionan los campos tipos de actividad y opción y se presiona el botón Añadir que aparece a continuación. En correspondencia con los campos previamente seleccionados se habilitarán los demás campos a llenar. En caso de que se seleccione la opción cantidad se habilitarán los campos operación y el campo valor será para introducir únicamente números. En caso de que se seleccione la opción campo se habilitarán los campos atributo, operación y valor. El campo de selección atributo será llenado en correspondencia con el tipo de actividad que se seleccione previamente. El campo valor aparecerá en correspondencia al atributo seleccionado.

Historia de usuario	
Número: RF21	Nombre del requisito: Listar requisitos para premio.
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Baja	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día
<p>Descripción: El sistema debe permitir listar los requisitos del premio a los usuarios con rol usuario de sistema.</p> <p>Objetivo: Permitir listar los requisitos del premio en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para listar los requisitos hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol Usuario de Sistema. - Debe existir al menos un premio en el sistema. <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Una vez seleccionado la opción Añadir Requisito en el listado de premios, aparece un listado de los requisitos del premio seleccionado. En el listado se brinda la posibilidad de eliminar requisitos de los premios. 	
Observación:	

Historia de usuario	
Número: RF22	Nombre del requisito: Eliminar requisito del premio
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Media	Riesgo en desarrollo: N/A
Tiempo estimado: 2 días	Tiempo Real: 2 días

<p>Descripción: El sistema debe permitir que los usuarios con rol usuario de sistema eliminen un requisito de los premios.</p> <p>Objetivo: Permitir eliminar requisitos del premio en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para eliminar un requisito del premio hay que:</p> <ul style="list-style-type: none">- Estar autenticado en el sistema con el rol usuario de sistema.- Debe existir al menos un premio en el sistema. <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">- El sistema permite eliminar un requisito del premio si se selecciona la opción Eliminar y se muestra un mensaje para completar la acción.- Si el usuario selecciona la opción "Sí, estoy seguro", se elimina el requisito seleccionado, se actualiza el listado de requisitos y el sistema muestra un mensaje de confirmación.- Si selecciona la opción Cancelar se regresará a la vista previa.	
Observación:	

Historia de usuario	
Número: RF30	Nombre del requisito: Crear propuesta
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: N/A
Tiempo estimado: 3 días	Tiempo Real: 3 días
<p>Descripción: El sistema debe permitir que los usuarios con rol usuario creen nuevas propuestas.</p> <p>Objetivo: Permitir incluir nuevas propuestas.</p>	

Acciones para lograr el objetivo (precondiciones y datos):

Para incluir una propuesta hay que:

- Tener en cuenta los siguientes datos: Premio, Estudiante, Observaciones.
- Estar autenticado en el sistema con el rol usuario.

Comportamientos válidos y no válidos (flujo central y alternos):

Los campos premio y estudiante son obligatorios

Premio: Campo de selección.

Estudiante: Campo de selección.

Observaciones: Campo de texto.

Flujo de la acción a realizar:

- El sistema debe permitir introducir los datos para incluir una nueva propuesta en el sistema.
- Cuando el usuario selecciona la opción Nueva propuesta se muestra un formulario, introduce correctamente los datos necesarios y selecciona la opción Guardar, se crea un nuevo elemento en el listado de propuestas, se oculta el formulario y el sistema muestra un mensaje de confirmación.
- Si selecciona la opción Cancelar se oculta el formulario de propuestas.

Observación: Cuando se seleccione el premio se presiona el botón Proponer estudiante, el cual hace que el campo de selección Estudiante se llene únicamente con los estudiantes de la brigada del usuario actual que cumplen con los requisitos del premio.

Historia de usuario	
Número: RF31	Nombre del requisito: Listar propuestas.
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Baja	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día

<p>Descripción: El sistema debe permitir listar propuestas a los usuarios con rol Usuario y Usuario de Sistema.</p> <p>Objetivo: Permitir listar propuestas en el sistema.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para listar propuestas hay que:</p> <ul style="list-style-type: none">- Estar autenticado en el sistema con el rol Usuario o Usuario de Sistema. <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">- Una vez seleccionado la opción Propuestas, aparece un listado de las propuestas. En el listado se brinda la posibilidad de eliminar propuestas en caso de que el usuario presente el rol Usuario de Sistema.	
<p>Observación: Una vez se cree una propuesta en el listado se mostrará, además de la información de dicha propuesta, el nombre de la persona que añadió la propuesta.</p>	

Historia de usuario	
Número: RF34	Nombre del requisito: Exportar pdf del listado de aprobados para premio.
Programador: Jessica C. Martínez	Iteración asignada: 3
Prioridad en negocio: Baja	Riesgo en desarrollo: N/A
Tiempo estimado: 1 día	Tiempo Real: 1 día
<p>Descripción: El sistema debe permitir que los usuarios de sistema puedan exportar un pdf con el listado de los aprobados para optar por el premio.</p> <p>Objetivo: Permitir exportar un pdf del listado de aprobados para un premio.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para mostrar la información de un premio hay que:</p>	

<ul style="list-style-type: none">- Estar autenticado en el sistema con el rol usuario del sistema. <p>Flujo de la acción a realizar:</p> <ul style="list-style-type: none">- El sistema permite exportar un pdf del listado de aprobados para un premio.
Observación:

Anexo 3: Tarjetas CRC

Tabla 14 Tarjeta CRC 3

Tarjetas CRC	
Nombre de la clase: ActividadCreateView	
Responsabilidades: obtenerEstudiante(self, pk): Obtiene la información del estudiante al que se le quiere añadir las actividades. obtenerValores(self, idActividad, campo): Obtiene los valores que va a tener determinado campo.	Colaboradores:

Tabla 15 Tarjeta CRC 4

Tarjetas CRC	
Nombre de la clase: GanadoresListView	
Responsabilidades: obtenerPremio(self): Obtiene la información del premio para el cual fue propuesto el estudiante para que se le otorgue el lugar de premiación. obtenerLugar(self): Obtiene los	Colaboradores:

lugares de premiación que serán asignados a los estudiantes propuestos que hayan sido aprobados para obtener el premio.	
---	--

Tabla 16 Tarjeta CRC 5

Tarjetas CRC	
Nombre de la clase: PropuestaListView	
Responsabilidades: obtenerPremio(self): Obtiene la información del premio al que se le quiere asignar un estudiante para que lo obtenga. obtenerEstudiante(self, pk): Obtiene la información del estudiante al que se quiere premiar	Colaboradores:

Anexo 4: Matriz de trazabilidad

Tabla 17 Matriz de trazabilidad

	HU1 Autenticar usuario	HU2 Listar usuario	HU3 Crear usuario	HU4 Modificar usuario
RF1 Autenticar usuario	X			

RF2 Listar usuario		X		
RF3 Crear usuario			X	
RF4 Modificar usuario				X
	HU5 Eliminar usuario	HU6 Listar estudiantes	HU7 Crear estudiante	HU8 Modificar estudiante
RF5 Eliminar usuario	X			
RF6 Listar estudiantes		X		
RF7 Crear estudiante			X	
RF8 Modificar estudiante				X
	HU9 Eliminar estudiante	HU10 Importar listado de estudiantes	HU11 Exportar listado de estudiantes	HU12 Agregar actividad a estudiante
RF9 Eliminar estudiante	X			
RF10 Importar listado de estudiantes		X		

RF11 Exportar listado de estudiantes			X	
RF12 Agregar actividad a estudiante				X
	HU13 Eliminar actividad del estudiante	HU14 Listar actividades del estudiante	HU15 Listar premio	HU16 Crear premio
RF13 Eliminar actividad del estudiante	X			
RF14 Listar actividades del estudiante		X		
RF15 Listar premio			X	
RF16 Crear premio				X
	HU17 Modificar premio	HU18 Eliminar premio	HU19 Mostrar datos del premio	HU20 Agregar requisitos a premio
RF17 Modificar premio	X			
RF18 Eliminar premio		X		

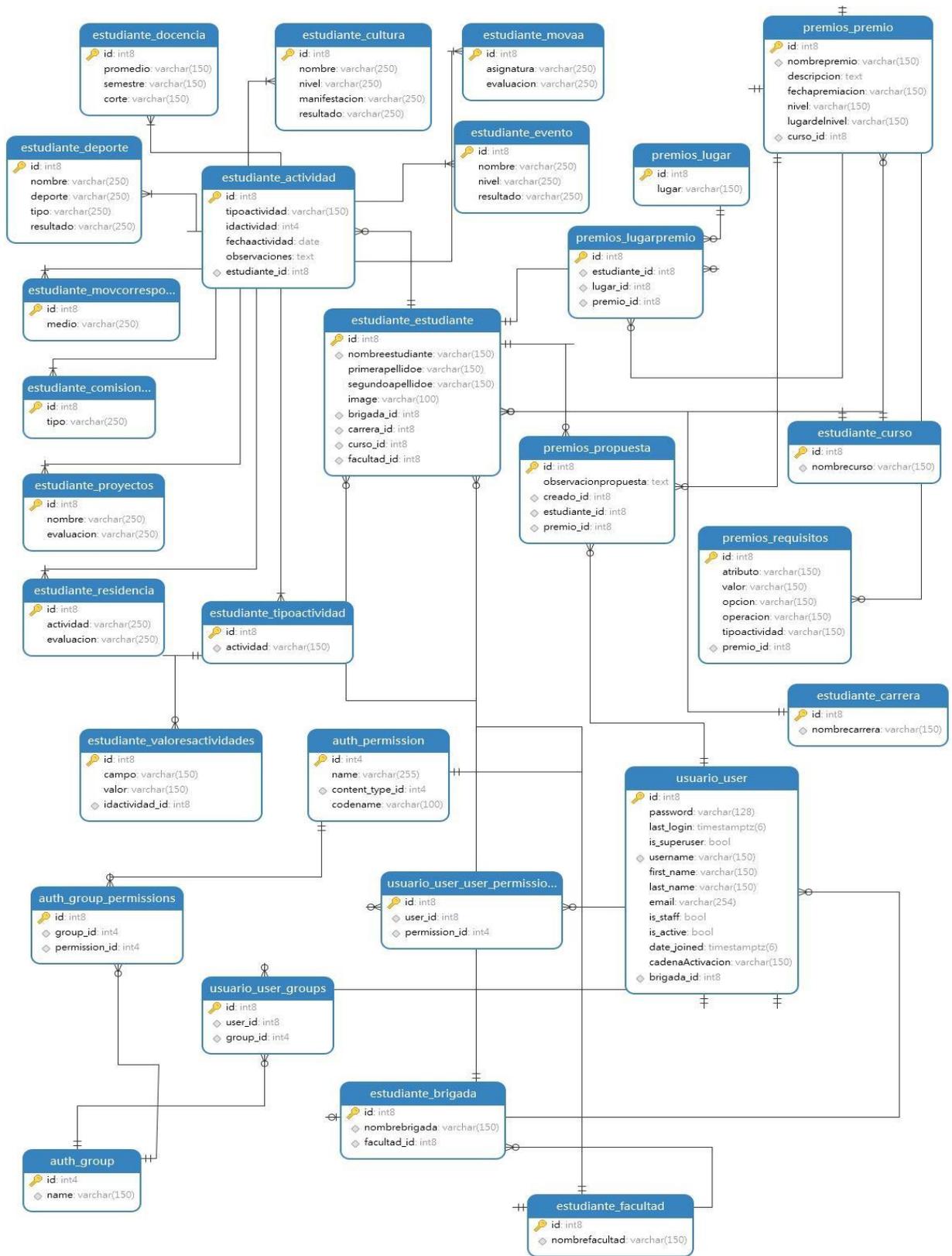
RF19 Mostrar datos del premio			X	
RF20 Agregar requisitos a premio				X
	HU21 Listar requisitos de premio	HU22 Eliminar requisito del premio	HU23 Crear actividad Proyectos	HU24 Crear actividad Ayudantía
RF21 Listar requisitos de premio	X			
RF22 Eliminar requisito del premio		X		
RF23 Crear actividad Proyectos			X	
RF24 Crear actividad Ayudantía				X
	HU25 Crear actividad Deportiva	HU26 Crear actividad Cultural	HU27 Crear actividad Residencia	HU28 Crear actividad Comisión Disciplinaria
RF25 Crear actividad Deportiva	X			

RF26 Crear actividad Cultural		X		
RF27 Crear actividad Residencia			X	
RF28 Crear actividad Comisión Disciplinaria				X
	HU29 Crear actividad Eventos	HU30 Crear propuesta	HU31 Listar propuestas	HU32 Asignar lugar de premiación
RF29 Crear actividad Eventos	X			
RF30 Crear propuesta		X		
RF31 Listar propuestas			X	
RF32 Asignar lugar de premiación				X
	HU33 Listar aprobados de premios	HU34 Exportar pdf de listado de aprobados de premio		

RF33 Listar aprobados de premios	X			
RF34 Exportar pdf de listado de aprobados de premio		X		

Anexo 5: Modelo físico de datos

Tabla 18 Modelo físico de datos



Anexo 6: Algunas de las no conformidades de las pruebas funcionales

No	No Conformidades	Tipo	Estado
1	No se valida la existencia de campos vacíos cuando se crea un estudiante	Validaciones	Resuelto
2	No se valida cuando se crea el usuario que este ya exista en el sistema	Validaciones	Resuelto
3	Cuando se añade un nuevo estudiante no se muestra un mensaje de confirmación de que la operación fue realizada.	Funcional	Resuelto
4	Cuando se elimina un estudiante no se muestra un mensaje de confirmación de que la operación fue realizada.	Funcional	Resuelto
5	En la plantilla de autenticación se encontró un error ortográfico: la palabra sesión estaba escrita con c	Ortográfico	Resuelto
6	En la plantilla de registro de usuario se encontró un error ortográfico: la palabra introduzca estaba escrita sin la z	Ortográfico	Resuelto
7	En la plantilla de registro de usuario se encontró un error ortográfico: la palabra usuario estaba escrita sin la a	Ortográfico	Resuelto
8	En la plantilla de listar estudiantes se encontró un error ortográfico: la palabra imagen estaba escrita como image	Ortográfico	Resuelto
9	Cuando se va a modificar un estudiante no se muestran en los campos los datos del estudiante	Funcional	Resuelto
10	En la plantilla donde se van a mostrar los datos de los estudiantes no se muestran	Funcional	Resuelto

	algunos datos como por ejemplo la facultad		
11	En el registro de usuarios no se adicionó correctamente un usuario determinado en la base de datos	Funcional	Resuelto