

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**ANÁLISIS Y DISEÑO DE LA GESTIÓN  
DE LOS EVENTOS CULTURALES  
DE LA FEU EN LA UCI**

Trabajo de diploma para optar por el título de:

**Ingeniero en  
Ciencias Informáticas**

**Autor: Arian Abel Couso Linares**

**Tutores: Lic. Yoemny González Almaguer**

**Ing. Osay Gonzalez Fuentes**

**Ciudad de la Habana, Junio 2007**

*A mis padres Abel y Bertha, por su confianza, amor y sabios  
consejos. Y porque este era su sueño.*

*A mi hermano Arniel, por seguir mis pasos.*

*A toda mi familia, porque están en todo lo que hago y se merecen  
más de lo que hoy le doy.*

*Arian Abel*

*A mis padres, por todo su esfuerzo, cariño y comprensión desde el inicio de esta agotadora carrera, sin ellos no estaría hoy aquí y no sería lo que soy.*

*A mi hermano, por entenderme y estar ahí cuando lo necesite...*

*A mis abuelos, mis primos, tíos y toda mi familia por preocuparse por mí en cada una de las etapas vividas y enseñarme el valor de la familia.*

*A la familia Bazán-Couso, por acogerme en su casa y convertirse en mis padres durante estos cinco años.*

*Al tío Alfredo, por convertirme en su hijo y enseñarme muchas cosas de la vida...*

*Al primo Amílcar, por haber cargado todos estos años conmigo...*

*A Wendy y Alberto, por ser más que amigos, hermanos...*

*A Olivia (Winnie), por llenarme de felicidad estos últimos días... te quiero.*

*A todos mis amigos, sólo por eso, por ser mis amigos y estar siempre...*

*A mis tutores, por la ayuda que me brindaron.*

*A la Revolución por darme la oportunidad de formar parte de este proyecto.*

*A todos los que de una forma u otra contribuyeron a mi formación como profesional.*

Declaro que soy el único autor de este trabajo y autorizo al de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Arian Abel Couso Linares

\_\_\_\_\_  
Firma del Autor

Lic. Yoemny González Almaguer

\_\_\_\_\_  
Firma del Tutor

Ing. Osay Gonzalez Fuentes

\_\_\_\_\_  
Firma del Tutor

Título: **Análisis y Diseño de la Gestión de los Eventos Culturales de la FEU en la UCI.**

Autor: Arian Abel Couso Linares

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado consideramos que el estudiante está apto para ejercer como Ingeniero en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_\_ puntos.

\_\_\_\_ días del mes de junio de 2007.

---

Lic. Yoemny González Almaguer

---

Ing. Osay Gonzalez Fuentes

### Resumen

Con el auge del uso de las TICS, se ha visto cómo las mismas han contribuido al desarrollo de la humanidad en todos los aspectos de su vida, sobre todo a la hora de organizar y gestionar las distintas tareas que le corresponden a diario. Igualmente la Federación Estudiantil Universitaria (FEU) en relación directa con el tamaño y la magnitud de sus actividades se ha incorporado a dicho auge, en el cual le ha resultado inevitable incluirse.

El trabajo que se presenta consiste en realizar una investigación exhaustiva de los sistemas que gestionan los Eventos Culturales en las organizaciones estudiantiles en la actualidad, un estudio del negocio y funcionamiento de estos en la Universidad de las Ciencias Informáticas (UCI), permitiendo a través del mismo una mejora de los procesos que actualmente se llevan a cabo.

Dentro de los aspectos interesantes están las principales herramientas, metodologías de desarrollo y la justificación de su uso para realizar el análisis del sistema. También se hace una propuesta de las herramientas a utilizar para la implementación del sistema.

Todo esto con el objetivo de analizar y diseñar un sistema para la gestión de los Eventos Culturales de FEU en la UCI.

---

**Índice**

Introducción .....	1
Capítulo 1. Fundamentación Teórica.....	4
Introducción.....	4
1.1. Sistemas de Gestión de los Eventos Culturales.....	4
1.2. Software Libre.....	5
1.3. Herramientas y Tecnologías.....	8
1.4. Herramientas que se proponen para el desarrollo.....	13
Conclusiones.....	17
Capítulo 2. Características del Sistema.....	18
Introducción.....	18
2.1. Situación Problemática.....	18
2.2. Objeto de Automatización.....	19
2.3. Información que se maneja.....	19
2.4. Propuesta del Sistema.....	21
2.5. Modelo de Negocio.....	21
2.6. Actores y Trabajadores del Negocio.....	22
2.7. Casos de Uso del Negocio.....	23
2.8. Especificaciones de los Casos de Uso del Negocio.....	24
2.9. Diagrama de Clases del Modelo de Objeto del Negocio.....	26
2.10. Especificación de los Requisitos de Software del Sistema.....	27
2.11. Requerimientos Funcionales.....	27
2.12. Requerimientos no Funcionales.....	28
2.13. Definición de Actores del Sistema.....	29
2.14. Listado de Casos de Uso.....	30
2.15. Diagrama de Casos de Uso del Sistema.....	32
2.16. Casos de Uso por Ciclos.....	33
2.17. Casos de Uso Expandidos.....	35
Conclusiones.....	44
Capítulo 3. Análisis y Diseño del Sistema.....	45

Introducción.....	45
3.1. Análisis.....	45
3.1.1. Modelo Conceptual de Clases de Análisis. ....	45
3.2. Diseño.....	45
3.2.1. La vista de Interacción.....	46
3.2.2. Interacción.....	46
3.2.3. Diagrama de Secuencia.....	47
3.2.4. Diagrama de Diseño Web del Sistema.....	47
3.2.5. Descripción de las Clases.....	47
3.2.5.1. Clases de Control.....	48
3.2.5.2. Clases de Entidad.....	52
3.2.5.3. Clases de Interfaz.....	55
3.2.6. Diseño de la Base de Datos del Sistema.....	60
3.2.6.1. Descripción de las tablas.....	61
3.3. Estimación de Costo-Beneficio.....	64
Conclusiones.....	70
Conclusiones Generales.....	71
Recomendaciones.....	72
Referencias Bibliográficas.....	73
Bibliografía.....	74
Anexos.....	76
Anexo 1. Diagrama de Casos de Uso del Negocio.....	76
Anexo 2. Diagrama de Actividades del caso de Uso del Negocio.....	77
Anexo 3. Diagrama de Clases del Modelo de Objeto del Negocio.....	78
Anexo 4. Diagrama de Casos de Uso del Sistema.....	79
Anexo 5. Diagramas del Modelo de Clases del Análisis.....	80
Anexo 6. Diagramas de Interacción (Secuencia).....	84
Anexo 7. Diagrama de Diseño Web del Sistema.....	88
Anexo 8. Diagrama de Entidad Relación de la Base de Datos.....	89
Glosario de Términos.....	90

### **Introducción**

La informática en Cuba ha alcanzado un gran desarrollo en los últimos años, se está llevando a todos los niveles de nuestra sociedad y economía. Con este avance se consigue que el desarrollo del trabajo sea superior tanto cuantitativamente como cualitativamente y además le da un valor agregado a cada servicio que se presta. Algo muy parecido ocurre en las universidades cubanas y especialmente en la Universidad de las Ciencias Informáticas, la que tiene como labor fundamental formar jóvenes profesionales integrales y con alto grado de desarrollo intelectual.

Los estudiantes y profesores de la UCI, desde el surgimiento de la misma, han procurado automatizar muchos de los procesos que garantizan su buen funcionamiento como centro universitario. En la UCI, específicamente la Federación Estudiantil Universitaria ha decidido desarrollar e implementar una aplicación que gestione sus actividades en la universidad, con el objetivo de ofrecer una mayor organización a la comunidad universitaria.

Así surge el proyecto Kainos, diseñado con el objetivo de crear un espacio digital para la FEU, donde se registran y promueven sus actividades y funcionamiento, a través de la red interna de la UCI. Kainos sigue incorporando funcionalidades a su espacio digital, el cual no es más que un Sitio Web con novedosas características, incorporando varios módulos, entre ellos: Gestión de los Eventos Científicos, Gestión del Plan de Trabajo y Emulación y Gestión de los Eventos Culturales. El siguiente trabajo se centrará en el análisis y diseño del Módulo Gestión de los Eventos Culturales de la FEU en la UCI.

La actividad cultural contribuye a elevar el desarrollo artístico-cultural de los estudiantes universitarios. En la UCI los eventos más significativos anualmente son los Festivales Culturales de Base, Festival UCI y la Gala de Premiación como conclusión al trabajo realizado por los artistas aficionados al Arte y la Literatura.

La problemática de estos eventos en la comunidad universitaria es la falta de organización, información y divulgación de los mismos, por tanto constituye como objetivo general de este trabajo realizar el análisis y diseño de una aplicación Web que sea capaz de gestionar los Eventos Culturales de un curso escolar en la UCI.

A partir de un análisis del objetivo general se deriva los siguientes objetivos específicos:

1. Revisar aspectos teóricos conceptuales sobre los sistemas de gestión de eventos culturales en las organizaciones estudiantiles.
2. Gestionar requerimientos y necesidades de los usuarios.
3. Analizar y diseñar una aplicación que permita el manejo de la información de los Eventos Culturales de la Federación Estudiantil Universitaria en la Universidad de las Ciencias Informáticas.

Dentro de las tareas que se proponen para dar solución a los objetivos planteado está:

1. Realizar un estudio preliminar del problema y de la situación actual.
2. Desarrollar fundamentación teórica del objeto de estudio.
3. Desarrollar el Modelo de la aplicación siguiendo la metodología de RUP.
4. Proponer un diseño preliminar.
5. Proponer el modelo de Base de Datos.

El presente documento se estructura en tres capítulos y varios anexos, que incluye todo lo relacionado con el trabajo investigativo, así como el análisis y diseño de la herramienta que se propone.

### Capítulo 1 Fundamentación Teórica.

En este capítulo se aborda el estado actual de desarrollo de los sistemas de gestión de la Información cultural en el ámbito mundial, nacional y en la Universidad, además de un estudio de las tecnologías en las que se apoya el desarrollo del sistema en función de un análisis de las tendencias actuales.

### Capítulo 2 Características del Sistema.

En este capítulo se realiza un estudio del problema dado a solucionar que está orientado a fundamentar la necesidad de la creación de un sistema que automatice el proceso de gestión de los Eventos Culturales de la FEU en la UCI.

### Capítulo 3 Análisis y Diseño del Sistema.

En este capítulo se abordará el análisis y diseño de la aplicación en los aspectos más importantes en función de satisfacer los requerimientos esperados, se confeccionan varios diagrama como son, los diagramas de Interacción, el diagrama de Entidad Relación, entre otros. Además se realiza un análisis del esfuerzo (Costo-Beneficio).

### Capitulo 1. Fundamentación Teórica.

#### Introducción

En este capítulo se aborda el estado actual de desarrollo de los sistemas de gestión de la Información cultural en el ámbito mundial, nacional y en la Universidad, además de un estudio de las tecnologías en las que se apoya el desarrollo del sistema en función de un análisis de las tendencias actuales.

#### 1.1. Sistemas de Gestión de los Eventos Culturales.

No se ha encontrado precedentes alguno de sistemas que gestione los eventos culturales de las organizaciones estudiantiles, solo existen sitios Web que promueven estos eventos, en forma de cartelera o simplemente publicando artículos sobre estos, donde se detalla de manera precisa las características generales de los eventos, así como cualquier información adicional de utilidad para el usuario. Es fácil darse cuenta del desarrollo que han alcanzado estos sitios de promoción en el mundo, y para citar algunos, podemos mencionar la Intranet de nuestra Universidad, en la cual se promueve las actividades culturales, los principales eventos que se desarrollan a lo largo del curso escolar, la visita de alguna agrupación musical y todas aquellas actividades que de cierta forma contribuyen al desarrollo de la cultura.

Existen a nivel nacional e internacional cientos de sitios que de cierta forma divulgan y promueven todo lo relacionado a los eventos culturales. En estos sitios podemos encontrar mucha información de eventos tanto internacionales como propios del país. Para poner algunos ejemplos: [www.cult.cu](http://www.cult.cu) (Sitio Web de la Cultura Cubana), [www.atenas.cult.cu](http://www.atenas.cult.cu) (Sitio Web de la Cultura matancera), [www.leedor.com](http://www.leedor.com) (Sitio Web Internacional de Cultura), [www.italica.rai.it](http://www.italica.rai.it) (Sitio Web de la Cultura Italiana).

### 1.2. Software Libre.

“El software libre, es un movimiento tecnológico que ha revolucionado la sociedad por sus implicaciones propias. Presenta características especiales que han permitido (y hecho necesaria) la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos, y nuevas normas legales.

El Software Libre es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión”.

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.” {1}

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el Software Libre.

Para que estas libertades sean reales, deben ser irrevocables mientras no se haga nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no se le haya dado motivos, el software no es libre.

Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir Software Libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, *copyleft* ["izquierdo de copia"]; expresado muy simplemente es la regla que implica que, cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

Esto da la medida de la viabilidad económica de este sistema libre, que por ser libre no es necesariamente gratuito, sino que nos da la posibilidad de comercializarlo, regalarlo, prestarlo con total libertad y protegido legalmente. Evidentemente, a pesar de las lagunas técnicas que pueda tener aun el Software Libre, es la alternativa para los países subdesarrollados.

### **Ventajas**

- ❖ **Costo:** el costo total de propiedad del sistema operativo libre Linux es menos de la mitad que el de Windows. Gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costos de administración.
- ❖ **Innovación tecnológica:** el desarrollo en comunidad de este sistema y el conocimiento del Código Fuente, propician que a cada instante, un desarrollador necesite nuevas actualizaciones y las realice él mismo, proponiendo nueva funcionalidad al programa

- ❖ Requisitos de hardware y durabilidad de las soluciones: un mejor soporte en general para las versiones antiguas de software y de plataformas de hardware o software minoritarias. No necesitas comprar nuevas versiones para estar actualizado.
- ❖ Escrutinio público: el proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Cada mejora es socializada libremente, la comunidad puede cambiar la realidad de las innovaciones.
- ❖ Independencia del proveedor: el Software Libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente.
- ❖ Desarrollo de la industria local: en el Software Libre no hay costo de licencia debido al derecho a copia y, al disponer del código fuente de la aplicación, es posible desarrollar internamente las mejoras o las modificaciones necesarias. De este modo, se contribuye a la formación de profesionales en nuevas tecnologías y al desarrollo local y de la industria nacional de software.
- ❖ Datos personales, privacidad y seguridad: el Software Libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el mundo del software libre, cualquier programador puede realizar una auditoria para comprobar que no se ha introducido ningún código malicioso, y, a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos.

- ❖ Adaptación del software: el Software Libre permite personalizar, gracias al hecho de que se dispone del código fuente, los programas tanto como sea necesario hasta que cubran exactamente cualquier necesidad.
- ❖ Lengua: los principales proyectos libres, tienen numerosas traducciones, gracias a que no precisan autorización de ningún propietario y cualquier persona o institución puede realizarlas.

### **Desventajas**

Una de las principales desventajas de los Sistemas Libres, (que a veces constituye una ventaja), radica en el hecho de que existe más de una imagen y cada desarrollador o usuario utiliza la que le conviene. Sin embargo, la comunidad nunca se pone de acuerdo hacia cual versión se debe migrar.

Para los especialistas en tecnologías de la informática, esto no representa un problema, pero si hablamos de usuarios finales (secretarias, profesionales de otras ramas que usan la computadora solo como medio auxiliar, estudiantes, población en general), si cada vez que ven una PC, tiene una versión diferente (Debian, Ubuntu, Nova, etcétera), entonces se les hace muy difícil su trabajo.

### **1.3. Herramientas y Tecnologías.**

Las herramientas utilizadas para la realización de este trabajo de diploma son las siguientes:

#### **Visual Paradigm**

Al escoger las herramientas para analizar y diseñar el software en cuestión nos basamos en las tendencias más actuales a nivel nacional e internacional, seleccione para el análisis y diseño la herramienta de modelado Visual Paradigm 5.3 Enterprise Edition pues aunque es Software

privativo gratuito para modelado en UML, la universidad adquirió la licencia. Su uso se ha extendido entre los analistas de todo el mundo, dadas sus facilidades de integración con distintos entornos de desarrollo, incluye UML 2.0, brindando la posibilidad de modelar el sistema orientado a procesos en el futuro.

### **Subversion (Tortoise)**

Para garantizar un correcto control de versiones y gestión de cambio se decidió utilizar el Subversion con el cliente Tortoise, ya que es una de las herramientas que más se ha venido utilizando en nuestra universidad ocupando el 98% de los proyectos que la utilizan actualmente, frente a un 2 % de los proyectos que usan el Visual Source Safe.

### **J2EE**

Java 2 Enterprise Edition, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Java con arquitectura de N niveles distribuidos, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Java EE es también considerada informalmente como un estándar. Es por esto que se decidió utilizar J2EE

La utilización de Servidores de Aplicaciones y la ejecución de aplicaciones basadas en tecnología J2EE impactan en mayor o menor medida en distintas disciplinas dentro de una organización. Algunas de las más relevantes:

1. Modelado de Datos.
2. Análisis de Requerimientos.
3. Arquitectura y Diseño.
4. Codificación.
5. Testing.
6. Entorno y Operación.

### 7. Administración de Proyectos.

#### **Arquitectura basada en Hibernate+Spring+JSF.**

##### ***Diseño de la Arquitectura de Alto Nivel.***

La arquitectura de alto nivel implica subdividir la aplicación en componentes funcionales y particionar estos componentes en capas. El diseño de la arquitectura de alto nivel es neutral a las tecnologías utilizadas.

##### ***Arquitectura Multi-Capa.***

Una arquitectura multicapa particiona todo el sistema en distintas unidades funcionales: cliente, presentación, lógica-de-negocio, acceso a datos, y bases de datos. Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica-de-negocios.

La capa del cliente es donde se consumen y presentan los modelos de datos. Para una aplicación Web, la capa cliente normalmente es un navegador Web. Los clientes pequeños basados-en-navegador no contienen lógica de presentación; se trata en la capa de presentación.

La capa de presentación expone los servicios de la capa de lógica-de-negocio a los usuarios. Sabe cómo procesar una petición de cliente, cómo interactuar con la capa de lógica-de-negocio, y cómo seleccionar la siguiente vista a mostrar.

La capa de la lógica-de-negocio contiene los objetos y servicios de negocio de la aplicación. Recibe peticiones de la capa de presentación, procesa la lógica de negocio basada en las peticiones, y media en los accesos a los recursos de la capa de datos. Los componentes de la

capa de lógica-de-negocio se benefician de la mayoría de los servicios a nivel de sistema como el control de seguridad, de transacciones y de recursos.

La capa de acceso a datos es el puente entre la capa de lógica-de-negocio y la capa de datos. Encapsula la lógica para interactuar con la capa de datos.

Los datos de la aplicación persisten en la capa de datos. Contiene bases de datos relacionales, bases de datos orientadas a objetos, y sistemas antiguos. {2}

### ***La Capa de Presentación y Java Server Faces.***

JSF es un marco de trabajo de componentes de interface de usuario del lado del servidor para aplicaciones Web basadas en Java. JSF contiene un API para representar componentes UI y manejar sus estados, manejar sus eventos, la validación del lado del servidor, y la conversión de datos, definir la navegación entre páginas, soportar internacionalización y accesibilidad; y proporcionar extensibilidad para todas estas características. También contiene dos librerías de etiquetas JSP (JavaServer Pages) personalizadas para expresar componentes UI dentro de una página JSP y para conectar componentes a objetos del lado del servidor.

La capa de presentación recoge la entrada del usuario, presenta los datos, controla la navegación por las páginas y delega la entrada del usuario a la capa de la lógica-de-negocio. La capa de presentación también puede validar la entrada del usuario y mantener el estado de sesión de la aplicación.

### ***La Capa de Lógica-de-Negocio y el Marco de Trabajo Spring.***

Los objetos y servicios de negocio existen en la capa de lógica-de-negocio. Un objeto de negocio no sólo contiene datos, también la lógica asociada con ese objeto específico.

Los servicios de negocio interactúan con objetos de negocio y proporcionan una lógica de negocio de más alto nivel. Se debería definir una capa de interface de negocio formal, que contenga los interfaces de servicio que el cliente utilizará directamente. Con la ayuda del marco de trabajo Spring, implementará la capa de lógica-de-negocio de la aplicación JCatalog.

### ***Manejo de Beans con contexto de aplicación.***

Spring puede organizar de forma efectiva nuestros objetos de la capa central y manejar las conexiones por nosotros. Spring puede eliminar la proliferación de solitarios y facilita unas buenas prácticas de programación orientada a objetos, por ejemplo utilizando interfaces.

### ***Integración con Hibernate.***

Spring no nos fuerza a utilizar su potente característica de abstracción JDBC. Se integra bien con marcos de trabajo de mapeo O/R, especialmente con Hibernate. Spring ofrece un manejo seguro y eficiente de sesiones Hibernate, maneja la configuración de la SessionFactory de Hibernate y las fuentes de datos JDBC en el contexto de la aplicación, y hace que la aplicación sea más fácil de testear.

### ***La Capa de Integración de Hibernate.***

Hibernate es un marco de trabajo de mapeo Open Source que evita la necesidad de utilizar el API JDBC. Hibernate soporta la mayoría de los sistemas de bases de datos SQL. El *Hibernate Query Language*, diseñado como una extensión mínima, orientada a objetos, de SQL, proporciona un puente elegante entre los mundos objeto y relacional. Hibernate ofrece facilidades para recuperación y actualización de datos, control de transacciones, repositorios de conexiones a bases de datos, consultas programáticas y declarativas, y un control de relaciones de entidades declarativas.

### **RUP**

Se utilizará la metodología RUP por las tendencias actuales que existen, que también hablan mucho de la utilización de XP o Extreme Programming que proponen un ciclo de desarrollo de software de manera ágil y novedosa, contando siempre con un cliente interno que tendrá un contacto directo con el equipo de desarrollo, razón por la cual no es apropiado para el sistema precisamente por las características del cliente. RUP es la que mejor se adapta a este caso de estudio y a las condiciones del equipo de desarrollo, teniendo en cuenta que el organigrama de equipo del mismo es descentralizado democrático y la comunicación es abierta. RUP es uno de los procesos más generales y abarcadores de los existentes actualmente, ya que en realidad está pensado englobar todo el desarrollo del mismo, además de ser una metodología probada y segura. RUP se basa en casos de uso para describir lo que se espera del software y está orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal. {3}

### **1.4. Herramientas que se proponen para el desarrollo.**

#### **Apache Tomcat**

Como servidor de aplicaciones la propuesta es Apache Tomcat, debido a que constituye uno de los más completos contenedores de Servlet gratuito, que programadores de Servlet de Java o Páginas de JavaServer (JSP) utilizan con frecuencia para probar su código. Tomcat persigue además la total compatibilidad con las versiones de Servlet y las especificaciones de API JSP que admite. Sin embargo, se trata de algo más que un servidor de pruebas, dado que muchas empresas lo emplean en la actualidad en entornos de producción debido a su contrastada estabilidad.

¿Por qué se necesita usar Tomcat para ejecutar JAVA en Apache?

El funcionamiento principal de Apache desde su creación fue la de aceptar y responder requests de páginas de internet, y como fue mencionado en servidores de páginas y JAVA Application Server. Estos requests correspondían a documentos estáticos (puro HTML), es por esto que cuando se necesita ejecutar algún tipo de contenido dinámico (programas como JAVA) es necesario coordinar los esfuerzos de Apache con otro ambiente. En el caso de JAVA es precisamente Tomcat. Quién ofrece facilidades para ejecutar los dos componentes más utilizados en ambientes JAVAS: JSP, (Java server Pages) y servlets.

### **IDE Eclipse.**

Como Entorno Integrado de Desarrollo (IDE) proponemos la utilización del Eclipse, el cual no es más que un software creado originalmente por IBM y actualmente, la Fundación Eclipse, una entidad sin ánimo de lucro, fundada por IBM se encarga de llevar a cabo el desarrollo de este proyecto. La definición que da la Fundación Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

Eclipse es un entorno integrado de desarrollo (IDE, Integrated Development Environment) multiplataforma englobado en el movimiento de software libre para crear aplicaciones clientes de cualquier tipo.

### **Beneficios.**

- Es open-source.
- Soporta la construcción de una variedad de herramientas para el desarrollo de aplicaciones.
- Soporta el desarrollo de aplicaciones basadas en GUI y non-GUI.
- Soporta herramientas que manipulan diferentes tipos de archivos como por ejemplo Java, C, C++, EJB, HTML, GIF, etc.
- Corre en una gran cantidad de sistemas operativos incluyendo Windows y Linux.
- Provee a los desarrolladores, herramientas (ej.- PDE) que facilitan la creación de plugins.

- Mediante JDT facilita la creación de aplicaciones programadas en Java.

### **Desventajas.**

Si bien Eclipse es multiplataforma, los plugins no tienen por qué serlo. Existen plugins que sólo corren en una plataforma, o que aún no han sido desarrollados para más de una. Al ser una herramienta open-source, se desarrollan plugins que no tienen todas las funcionalidades que tienen en otras herramientas comerciales, como ser IBM Websphere.

### **PostgreSQL.**

Proponemos el uso de PostgreSQL, por ser un motor de base de datos, servidor de base de datos relacional libre, liberado bajo una licencia de software libre.

Algunas de sus características fundamentales son:

- Alta concurrencia
- Amplia variedad de tipos nativos
- Los usuarios pueden crear sus propios tipos de datos.
- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*).
- Disparadores (*triggers*).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Funciones.

Inconvenientes:

- Consume recursos y carga el sistema.
- Límite del tamaño de cada fila de las tablas a 8k.

- Es de 2 a 3 veces más lenta que MySQL.
- Menos funciones en PHP.

### **Java.**

Se propone que la implementación sea en el lenguaje Java dadas sus características y potencialidades.

Java está totalmente basado en clases y objetos. Todo en Java (aparte de algunos tipos primitivos) es un objeto. Contrariamente a lenguajes híbridos como C++ o el popular Visual Basic, en Java no se permite programar fuera de un objeto o clase. No hay módulos ni funciones globales.

Una característica más distintiva de Java es su capacidad multiplataforma. Lenguajes como C o COBOL se han implementado en múltiples plataformas, pero siempre han necesitado recompilaciones o adaptaciones al pasar de una a otra. En cambio, Java desde el principio ha sido pensado para adaptarse a varios entornos. Esto lo consigue no sólo a nivel de código fuente, sino también a nivel de código compilado. El programa que se escribe puede compilar en Windows o en Linux, y funciona. Pero, además, hasta el programa compilado puede ejecutarse sin más preparación, en distintas máquinas. Eso lo consigue porque Java se compila y ejecuta, no en un procesador o entorno en particular, sino en lo que se llama una "virtual machine", una máquina virtual. Nuestro programa Java podrá ejecutarse en cualquier sistema operativo que tenga una máquina virtual Java compatible.

Java no es solamente un lenguaje, es una tecnología. Al estar basado en clases y objetos, viene acompañado de un conjunto de éstos, que sirve como base para la programación de aplicaciones, de texto, gráficas o que se ejecuten en una página Web como "applets".

Java maneja prácticamente todas las bases de datos relacionales, desde Microsoft SQL Server, Oracle, Sybase, Informix, como Access, MySQL y PostgreSQL.

### **Conclusiones**

En este capítulo se realizó un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, fundamentándose las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar aunque en esta etapa solo se habla de ideas, es correcto que estén bien fundamentadas, porque estas constituyen la base para el posterior desarrollo de este trabajo.

## Capítulo 2. Características del Sistema.

### Introducción

En este capítulo se realiza un estudio del problema dado a solucionar que está orientado a fundamentar la necesidad de la creación de un sistema que automatice el proceso de gestión de los Eventos Culturales de la FEU en la UCI. Contiene además un análisis del negocio funcional actual y de los requerimientos esperados para el desarrollo del sistema. Se desarrollan los casos de uso previamente definidos y se llega a la selección de los casos a implementar.

### 2.1. Situación Problemática.

En la UCI, como en todas las Universidades se realizan actividades culturales, y dentro de ellas están, los Festivales de Artistas Aficionados del Arte y la Literatura. Estos festivales están organizados de la siguiente forma: primero se realizan los festivales a nivel de base, los cuales son representados por cada una de las facultades. Seguido a esto se realiza el festival a nivel de Universidad; en este festival participan los números culturales seleccionados en los festivales de facultad, Y para dar conclusión al desarrollo artístico-cultural se realiza la Gala de Premiaciones, donde se da a conocer los resultados finales del Festival de Artistas Aficionados al Arte y la Literatura en la UCI, para ello, se ha creado un sistema de emulación que permitirá seleccionar las facultades más destacadas en la realización de dichos eventos.

En los festivales de facultades solo participan las siguientes manifestaciones de Teatro, Música y Danza. Estas manifestaciones están representadas por una serie de números culturales. En el festival de la Universidad se agregan dos manifestaciones más, que son Artes Plásticas y Literatura. Estas manifestaciones no son presentadas en los festivales de base debido a que se presentan en forma de exposición, entonces se convocan a todas las facultades para realizar una sola donde participarían todas las obras.

La ausencia de un sistema automatizado que gestione las actividades culturales de la FEU en la UCI afecta directamente a la comunidad universitaria. Con el objetivo de realizar todas estas acciones expresadas anteriormente, una solución exitosa pudiera ser, contribuir al funcionamiento eficiente de estas actividades y elevar la satisfacción de disfrute de la comunidad universitaria, con el pronto análisis y diseño de un sistema.

### **2.2. Objeto de Automatización.**

Se desea automatizar todos los procesos mencionados anteriormente donde las personas encargadas de gestionar estos eventos podrá crear dichos eventos y adicionarles las manifestaciones correspondientes a cada uno de estos, como también podrá agregar los números culturales a las manifestaciones y el sistema de emulación. Una vez que la persona encargada haya introducido todos lo relacionado con el evento creado, le da la posibilidad a la comunidad universitaria de acceder a dicha información.

### **2.3. Información que se maneja.**

La información que se manipula serán todos los datos del evento creado, como el nombre del evento, a quien va dedicado, las manifestaciones, los números culturales, dentro de estos el nombre del número cultural y los integrantes o autores.

La información que se maneja en el sistema de emulación de los Festivales de Artistas Aficionados al Arte y la Literatura en la UCI es la siguiente:

Distinciones Otorgadas.

- 1er Lugar: Reconocimiento de Oro.
- 2do Lugar: Reconocimiento de Plata.
- 3er Lugar: Reconocimiento de Bronce.

- Mejor Espectáculo.
- Premio de la Cubanía
- Y en caso de que fuese necesario se otorga alguna Mención.

Sistema de Puntuación.

### 1. Organización del festival (20ptos).

1.1. Sistema de Divulgación (10ptos).

1.2. Entrega del guión 48 hrs. antes del espectáculo (5ptos).

1.3. Comenzar el espectáculo a la hora fijada (5ptos).

### 2. Desarrollo del Festival (30ptos).

2.1. Organización y montaje de la presentación del espectáculo (10ptos).

2.2. Duración del espectáculo de 1:30 hrs. a 2:00 hrs. (5ptos)

2.3. Iniciativas durante todo el día y el movimiento cultural creado (15ptos).

### 3. Resultados Obtenidos en le Festival a nivel de Facultad y Universidad (25ptos).

3.1. En el Festival de Facultad por cada número cultural seleccionado para participar en el Festival de Universidad se otorgarán 5 puntos.

3.2. En el Festival de Universidad

3.2.1. Por cada Diploma de Oro se otorgarán 8 puntos.

3.2.2. Por cada diploma de Plata se otorgarán 6 puntos.

3.2.3. Por cada Diploma de Bronce se otorgarán 4 puntos.

3.2.4. Por cada Mención se otorgarán 2 puntos.

### **2.4. Propuesta del Sistema.**

Se propone un sistema que podrá almacenar todos los datos referentes a las actividades culturales y brindará la facilidad a los encargados de gestionar estas actividades, como permitirles crear eventos, adicionar manifestaciones artísticas y a estas números culturales, crear sistema de emulación o modificarlo, además le brindara a la comunidad universitaria la posibilidad de acceder a esta información de forma directa.

### **2.5. Modelo de Negocio.**

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de características, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo de negocio.

El modelado del negocio es una técnica para comprender los procesos del negocio de la organización. Los propósitos que se persiguen al realizarse el modelado del negocio, son:

- Entender la estructura y la dinámica de la organización.
- Entender los problemas actuales e identificar mejoras potenciales.
- Asegurarse de que los clientes, usuarios finales y desarrolladores tienen una idea común de la organización.
- Derivar los requerimientos del sistema a partir del modelo de negocio que se obtenga.

La UCI no cuenta en estos momentos con un sistema que gestione la organización y preparación de los Festivales de Artistas Aficionados al Arte y la Literatura, actualmente este trabajo se realiza de la siguiente forma: se realiza la convocatoria a participar en una edición más de los festivales, seguidamente se convoca a una reunión a los activistas de cultura de cada una de las facultades y el personal encargado de la preparación de estos eventos en la universidad, en esta reunión se discute todo lo relacionado con el evento, las bases y se define el orden de presentación de cada una de las facultades, luego prosigue la preparación de la facultades, el montaje de los números y el guión, pero esto no es conocido por la comunidad universitaria hasta el día de la presentación puesto que no existe un sitio donde puedan encontrar esta información detallada. Para esto damos como pronta solución, la informatización de este proceso de gestión de los eventos culturales.

### **2.6. Actores y Trabajadores del Negocio.**

Después de todas las entrevistas y estudios necesarios realizados (haciendo una simplificación del negocio) se obtiene un listado de actores y trabajadores del negocio como se muestra a continuación:

<b>Actores del negocio</b>	<b>Justificación</b>
Consejo FEU de la UCI.	Son los que solicitan la realización de los Eventos Culturales de la FEU en la Universidad.

<b>Trabajadores del negocio</b>	<b>Justificación</b>
Responsable de Cultura en la Universidad.	Es el encargado de la organización, preparación y realización de los eventos culturales en la universidad por la FEU.
Responsable de Cultura en las Facultad.	Es el encargado de la organización, preparación y realización de los eventos culturales en la facultad por la FEU.
Jurado.	Es el encargado de otorgar los premios.
Artistas Aficionados.	Son los participantes de los eventos, los que conforman los números culturales.

## 2.7. Casos de Uso del Negocio.

1-) Gestión de los Eventos Culturales: Es el proceso que se lleva a cabo en la Universidad para la realización de los Eventos Culturales de la FEU, siendo más específico los Festivales de Artistas

Aficionados al Arte y la Literatura; estos son: Festival de Base, Festival UCI y la Gala de Premiaciones como cierre al trabajo realizado por los artistas.

El diagrama de casos de uso del negocio puede verse en el anexo 1.

### 2.8. Especificaciones de los Casos de Uso del Negocio.

Este proceso incluye una fase de modelado del negocio, que describe los procesos del negocio de la organización bajo estudio de manera que se puedan construir, de forma sencilla y directa, versiones iniciales de los modelos conceptuales y de casos de uso. Cada proceso del negocio se describe haciendo uso de un diagrama de actividades UML con calles (swimlanes). Posteriormente, se identifican los casos de uso del sistema a partir de las actividades y los conceptos (clases del dominio) a partir de los datos (objetos de información que fluyen entre las actividades). {4}

<b>Nombre del Caso de Uso:</b>	Gestión de los Eventos Culturales.
<b>Actores:</b>	Consejo FEU de la UCI.
<b>Propósitos:</b>	El objetivo de este proceso es la realización de los Eventos Culturales de la FEU en la Universidad, así como la preparación y organización de los mismos.
<b>Casos de Uso Asociados:</b>	-
<b>Resumen:</b> El proceso comienza cuando el Consejo FEU de la UCI solicita la realización de los Eventos Culturales. Luego el Responsable de Cultura de la Universidad lanza la convocatoria de participación a una edición más de los Festivales de Artistas Aficionados al Arte y la Literatura. El Responsable de Cultura de cada Facultad comienza con la preparación del Festival de Base y los números culturales que fueron premiados con Oro en estos, participarían en el Festival UCI, y como cierre a esta jornada de los festivales se realiza la Gala de Premiaciones, siendo el Responsable de Cultura en la Universidad el encargado de la	

preparación de estos dos últimos eventos.

**Flujo de Trabajo**

**Acción del Actor**

1. El Consejo FEU de la UCI solicita la realización de los Eventos Culturales.

**Respuesta del Negocio**

2. El Responsable de Cultura en la Universidad recibe la solicitud y lanza la convocatoria, y toda la información necesaria para la preparación de los mismos.

3. El Responsable de Cultura de la Facultad recibe la convocatoria y envía las manifestaciones.

4. Los Artistas Aficionados deciden en que manifestaciones participarán y montan los números culturales.

5. Responsable de cultura de la Facultad con los números culturales ya creados confecciona el guión del Festival de Base.

6. Los Artistas Aficionados presentan los números culturales.

7. Si el número cultural es premiado el Jurado le concede su premio.

7.1. Si no son premiados los números culturales se acaba el caso de uso.

8. Los Artistas Aficionados reciben premio, si este premio otorgado es Oro pasa al Festival UCI.

8.1. Si no es Oro el premio recibido se acaba el caso de uso.

9. Responsable de cultura en la Universidad recibe los números culturales premiados con Oro y prepara

<p>15. El Consejo FEU de la UCI recibe resumen terminándose el caso de uso.</p>	<p>el Guión del Festival UCI.</p> <p>10. Los Artistas Aficionados presentan sus números culturales en le Festival UCI.</p> <p>11. El Jurado si este es premiado les otorga su premio.</p> <p>11.1. Si no es premiado se acaba el caso de uso.</p> <p>12. Responsable de cultura en la Universidad recibe los números culturales premiados y prepara el guión de la Gala de Premiaciones.</p> <p>13. Los Artistas Aficionados reciben premio y presentan los números culturales por el cual obtuvieron el premio.</p> <p>14. Responsable de cultura en la Universidad realiza un resumen de los Eventos Culturales y se lo envía al Consejo FEU de la UCI.</p>
<p><b>Mejoras:</b></p>	<p>-</p>
<p><b>Cursos Alternos:</b></p>	

El diagrama de actividades del caso de uso del negocio puede verse en el anexo 2.

## 2.9. Diagrama de Clases del Modelo de Objeto del Negocio.

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos de negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. Aunque se puede construir un único diagrama, se recomienda confeccionarlo para cada caso de uso del negocio para una mejor claridad.

Como todo diagrama de clases, se pueden representar, además de la asociación los distintos tipos de relaciones entre las entidades de negocio (agregación, composición y generalización / especialización), la cardinalidad y navegabilidad de las relaciones, entre los trabajadores y estos con las entidades.

El diagrama de clases del modelo de objeto de negocio puede verse en el anexo 3.

### **2.10. Especificación de los Requisitos de Software del Sistema.**

El análisis de requisitos permite al ingeniero de sistemas especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software. El análisis de requisitos permite al ingeniero analista refinar la definición del software y construir los modelos de los dominios de datos, funcional y de comportamiento que van a ser tratados por el software. El análisis de requisitos proporciona modelos al diseñador del software que pueden traducirse en el diseño de datos, arquitectónico y de interfaz. Finalmente, la especificación de requisitos proporciona al diseñador y al cliente los medios para valorar la calidad una vez que se ha construido el software.

### **2.11. Requerimientos Funcionales.**

Los requerimientos funcionales no son más que la determinación clara y concisa de qué debe ser capaz de hacer el sistema, éstas se corresponden con opciones que ejecutará el software, operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema.

1. Crear Eventos Culturales que serian: Festival de Base, Festival UCI y Gala de Premiaciones.
2. Modificar Eventos Culturales.
3. Eliminar Eventos Culturales.

4. Agregar Manifestaciones Artísticas a los eventos creados.
5. Agregar Números Culturales a las manifestaciones artísticas agregadas a los eventos creados.
6. Mostrar Eventos Culturales creados.
7. Agregar Sistema de Emulación.
8. Modificar Sistema de Emulación.
9. Eliminar Sistema de Emulación.
10. Mostrar Sistema de Emulación.
11. Otorgar Premios.
12. Quitar Premios.
13. Mostrar Premios Otorgados.

### **2.12. Requerimientos no Funcionales.**

#### **Requisitos de Interfaz o Apariencia.**

- ❖ Interfaz con un diseño sencillo que contenga pocos gráficos, con vista a acelerar la velocidad de respuesta hacia el usuario.
  
- ❖ La interfaz debe limitarse a presentar las funcionalidades netamente del sistema logrando la concentración del usuario en las tareas que esté realizando, además de algunos otros contenidos netamente informativos.

### Requisitos de Rendimiento.

- ❖ El tiempo de respuesta de cada uno de las páginas debe ser menor que un minuto, excepto en aquellas que por las actividades que realizan, requieran más tiempo, tal es el caso de la generación de los distintos reportes que se necesiten generar.
- ❖ Estará implementado sobre una tecnología Web, facilitando su uso a través de la red.

### Requisitos de Soporte.

- ❖ La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos PostgreSQL, permitiéndole interactuar con otros sistemas estableciendo vías de compatibilidad.
- ❖ La capa que contiene las Reglas del Negocio debe estar sobre Apache Tomcat en Windows.
- ❖ Lograr la solidez de los datos realizando mantenimientos automatizados en la base de datos, orientados a la actualización y corrección de la información, a horas del día donde haya la menor cantidad de usuarios conectados.

### 2.13. Definición de Actores del Sistema.

Actores	Justificación
Responsable de Cultura en la Universidad.	Es la persona que interactúa con el sistema, la encargada de la realización y creación de estos eventos a nivel central en la universidad.

Aficionados.	Son los que accederán a la información brindada por el sistema, pueden ser los usuarios del dominio UCI.
--------------	--

#### 2.14. Listado de Casos de Uso.

CU-1	Gestionar Eventos Culturales.
<b>Actor</b>	Responsable de Cultura en la Universidad.
<b>Descripción</b>	Este caso de uso le permite al Responsable de Cultura en la Universidad crear, modificar y eliminar un evento cultural, que estos pueden ser: Festival de Base (Facultad), Festival UCI (Universidad) y la Gala de Premiaciones.
<b>Referencia</b>	Requerimiento Funcional: 1, 2, 3.

CU-2	Agregar Manifestaciones.
<b>Actor</b>	Responsable de Cultura en la Universidad.
<b>Descripción</b>	Este caso de uso le permite al Responsable de Cultura en la Universidad agregarle a los eventos creados las manifestaciones artísticas.
<b>Referencia</b>	Requerimiento Funcional: 4

CU-3	Agregar Números Culturales
<b>Actor</b>	Responsable de Cultura en la Universidad.

<b>Descripción</b>	Este caso de uso le permite al Responsable de Cultura en la Universidad agregarles a las manifestaciones agregadas a los eventos creados los números culturales que les corresponden.
<b>Referencia</b>	Requerimiento Funcional: 5

CU-4	Gestionar Sistema de Emulación.
<b>Actor</b>	Responsable de Cultura en la Universidad.
<b>Descripción</b>	Este caso de uso le permite al Responsable de Cultura en la Universidad agregar, modificar y eliminar el Sistema de Emulación por el cual se registrarán los Eventos Culturales.
<b>Referencia</b>	Requerimiento Funcional: 7, 8, 9.

CU-5	Gestionar Premios.
<b>Actor</b>	Responsable de Cultura en la Universidad.
<b>Descripción</b>	Este caso de uso le permite al Responsable de Cultura en la Universidad otorgar y quitar premios a los números culturales de los Eventos Culturales.
<b>Referencia</b>	Requerimiento Funcional: 11, 12.

CU-6	Mostrar Eventos Culturales.
<b>Actor</b>	Aficionados, Responsable de Cultura en la Universidad.
<b>Descripción</b>	El caso de uso se inicia cuando uno de los actores solicita al sistema que le muestre un evento cultural determinado, con todos sus datos.

<b>Referencia</b>	Requerimiento Funcional: 6
-------------------	----------------------------

CU-7	Mostrar Premios Otorgados.
<b>Actor</b>	Aficionados, Responsable de Cultura en la Universidad.
<b>Descripción</b>	El caso de uso se inicia cuando uno de los actores solicita al sistema que le muestre los números culturales que le fue asignado un premio.
<b>Referencia</b>	Requerimiento Funcional: 13

CU-8	Mostrar Sistema de Emulación.
<b>Actor</b>	Aficionados, Responsable de Cultura en la Universidad.
<b>Descripción</b>	El caso de uso se inicia cuando uno de los actores solicita al sistema que le muestre el sistema de emulación por el cual se registrarán los eventos culturales.
<b>Referencia</b>	Requerimiento Funcional: 10

### 2.15. Diagrama de Casos de Uso del Sistema.

Un diagrama de casos de uso (Use Case Diagram) es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un diagrama Main Use Case, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

Entre los elementos de un diagrama de casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas o no entre ellos:

- "comunica" (<<communicates>>): Relación (asociación) entre un actor y un caso de uso que denota la participación del actor en dicho caso de uso.
- "usa" (<<uses>>) (o <<include>> en la nueva versión de UML): Relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro.
- "extiende" (<< extends>>): Relación de dependencia entre dos casos de uso que denota que un caso de uso es una especialización de otro.

El diagrama de casos de uso del sistema puede verse en el anexo 4.

### 2.16. Casos de Uso por Ciclos.

Cód.	Nombre de caso de uso	Paquete	Justificación de la selección.
	Gestionar Eventos Culturales.		Este caso de uso le permite al Responsable de Cultura en la Universidad crear un Evento Cultural (Festival de Base, Festival UCI y Gala de Premiaciones.), modificarlo y eliminarlo.

<p>1</p>	<p>Gestionar Premios.</p> <p>Gestionar Sistema de Emulación.</p> <p>Agregar Manifestaciones Artísticas.</p> <p>Agregar Números Culturales.</p> <p>Mostrar Eventos Culturales.</p>	<p>Este caso de uso le permite al Responsable de Cultura en la Universidad otorgar premios y quitar premios.</p> <p>Este caso de uso le permite al Responsable de Cultura en la Universidad agregar, modificar y eliminar el sistema de emulación.</p> <p>Este caso de uso le permite al Responsable de Cultura en la Universidad agregar manifestaciones artísticas a los eventos creados.</p> <p>Este caso de uso le permite al Responsable de Cultura en la Universidad agregar los números culturales a las manifestaciones agregadas a los eventos creados.</p> <p>Este caso de uso les permite a los aficionados visualizar todos los eventos culturales que se han creado, así como todos sus datos.</p> <p>Este caso de uso les permite a los</p>
----------	---	---

	Mostrar Premios Otorgados.		aficionados visualizar todos los números culturales a los cuales le fue asignado un premio.
	Mostrar Sistema de Emulación.		Este caso de uso les permite a los aficionados visualizar el sistema de emulación por el cual se registrará el evento cultural.

### 2.17. Casos de Uso Expandidos.

<b>Caso de uso</b>	
CU-1	Gestionar Eventos Culturales.
<b>Propósito</b>	Crear, modificar y eliminar un evento cultural, que puede ser: Festival de Base (Facultad), Festival UCI (Universidad) y la Gala de Premiaciones.
<b>Actor:</b> Responsable de Cultura en la Universidad.	
<b>Resumen:</b> El caso de uso se inicia cuando el Responsable de Cultura en la Universidad desea crear, modificar o eliminar un evento cultural.	
<b>Referencias</b>	RF-1, 2, 3.
<b>Sección " Crear Evento Cultural "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor solicita crear un Evento Cultural.  2. El actor introduce todos lo datos necesarios en el formulario.	1.1. El sistema visualiza la página donde está el formulario para crear un evento cultural.  2.1. El sistema verifica que los datos introducidos son correctos y los almacena en la base de datos y muestra un mensaje que se cumplió satisfactoriamente el proceso.

<b>Sección " Modificar Evento Cultural "</b>	
<p>1. El actor solicita modificar un Evento Cultural.</p> <p>2. El actor selecciona el Evento Cultural que desea modificar.</p> <p>3. El actor modifica los datos que desea y pulsa el botón guardar.</p>	<p>1.1. El sistema visualiza la página donde están los Eventos Culturales creados.</p> <p>2.1. El sistema visualiza todo los datos relacionados con el Evento Cultural seleccionado.</p> <p>3.1. El sistema verifica que los datos modificados son correctos y los guarda en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente.</p>
<b>Sección " Eliminar Evento Cultural "</b>	
<p>1. El actor solicita eliminar un Evento Cultural.</p> <p>2. El actor selecciona el Evento Cultural que desea eliminar y pulsa el botón eliminar.</p>	<p>1.1. El sistema visualiza la página donde están los Eventos Culturales creados.</p> <p>2.1. El sistema elimina todos los datos relacionados con el Evento Cultural seleccionado y muestra un mensaje que se cumplió el proceso satisfactoriamente.</p>
<b>Flujos Alternos</b>	
<b>Sección " Crear Evento Cultural "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>Acción 2</p> <p>3. El actor corrige el error en los datos señalados.</p>	<p>2.2. En caso de no ser correctos los datos introducidos el sistema muestra un mensaje de error al lado de dicho campo.</p> <p>3.1. El sistema verifica que los datos introducidos son correctos y los almacena en la base de datos y muestra un mensaje que se cumplió satisfactoriamente el proceso.</p>

Sección " Modificar Evento Cultural "	
Acción del Actor	Respuesta del Sistema
Acción 3	3.2. En caso de no ser correctos los datos introducidos, el sistema muestra un mensaje de error al lado de dicho campo.
4. El actor corrige el error en los datos modificados y pulsa el botón guardar.	4.2. El sistema almacena en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente.
<b>Puntos de extensión.</b>	

Caso de uso	
CU-2	Agregar Manifestaciones.
<b>Propósito</b>	Agregar las manifestaciones artísticas a los eventos creados.
<b>Actor:</b> Responsable de Cultura en la Universidad.	
<b>Resumen:</b> El caso de uso se inicia cuando el Responsable de Cultura en la Universidad desea agregarle a un evento cultural ya creado las manifestaciones artísticas que lleva.	
<b>Referencias</b>	RF-4
Acción del Actor	Respuesta del Sistema
1. El actor solicita agregar manifestaciones artísticas.	1.1. El sistema muestra la página donde están todos los eventos creados.
2. El actor selecciona el evento al cual le desea agregar las manifestaciones.	2.1. El sistema le muestra los datos del evento solicitado y le posibilita agregar las manifestaciones.
3. El actor agrega las manifestaciones.	3.1. El sistema verifica si los datos son correctos y los guarda en la base de datos y muestra un mensaje que se cumplió

	satisfactoriamente el proceso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 3	3.2. En caso de no ser correctos los datos introducidos, el sistema muestra un mensaje de error al lado de dicho campo.
4. El actor corrige el error en los datos.	4.1. El sistema almacena en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente
<b>Puntos de extensión.</b>	

Caso de uso	
CU-3	Agregar Números Culturales.
<b>Propósito</b>	Agregar números culturales a las manifestaciones artísticas a un determinado evento.
<b>Actor:</b> Responsable de Cultura en la Universidad.	
<b>Resumen:</b> El caso de uso se inicia cuando el Responsable de Cultura en la Universidad desea agregarles los números culturales a las manifestaciones artísticas de un determinado evento creado.	
<b>Referencias</b>	RF-5
Acción del Actor	Respuesta del Sistema
1. El actor solicita agregar números culturales.	1.1. El sistema muestra la página donde están todos los eventos creados.
2. El actor selecciona el evento al cual le desea agregar los números culturales.	2.1. El sistema le muestra los datos del evento solicitado incluyendo las manifestaciones agregadas.

<p>3. El actor selecciona a que manifestación le desea agregarle los números culturales.</p> <p>4. El actor introduce todos los datos solicitados por el sistema.</p>	<p>3.1. El sistema le muestra el formulario con los datos necesarios para ingresar un número cultural.</p> <p>4.1. El sistema verifica los datos introducidos y los guarda en la base de datos mostrando un mensaje que se cumplió satisfactoriamente el proceso.</p>
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>Acción 4</p> <p>5. El actor corrige el error en los datos.</p>	<p>4.2. En caso de no ser correctos los datos introducidos, el sistema muestra un mensaje de error al lado de dicho campo.</p> <p>5.1. El sistema almacena en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente</p>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU-4	Gestionar Sistema de Emulación.
<b>Propósito</b>	Agregar, modificar y eliminar el Sistema de Emulación.
<b>Actor:</b> Responsable de Cultura en la Universidad.	
<b>Resumen:</b> El caso de uso se inicia cuando el Responsable de Cultura en la Universidad desea agregar, modificar o eliminar el Sistema de Emulación.	
<b>Referencias</b>	RF-7, 8, 9.
<b>Sección " Agregar Sistema de Emulación "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El actor solicita agregar el Sistema de Emulación.</p> <p>2. El actor introduce el Sistema de Emulación.</p>	<p>1.1. El sistema muestra la página con el formulario para agregar el Sistema de Emulación.</p> <p>2.1. El Sistema guarda la información en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente.</p>
<b>Sección " Modificar Sistema de Emulación "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El actor solicita modificar el Sistema de Emulación.</p> <p>2. El actor modifica la información que necesita y pulsa el botón guardar.</p>	<p>1.1. El sistema visualiza el Sistema de Emulación y le posibilita al actor modificarlo.</p> <p>2.1. El Sistema guarda la información modificada en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente.</p>
<b>Sección " Eliminar Sistema de Emulación "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El actor solicita eliminar el Sistema de Emulación.</p> <p>2. El actor pulsa el botón eliminar.</p>	<p>1.1. El sistema visualiza el Sistema de Emulación y le posibilita al actor eliminarlo.</p> <p>2.1. El sistema muestra un mensaje que se cumplió el proceso satisfactoriamente.</p>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU-5	Gestionar Premios.
<b>Propósito</b>	Otorgar y quitar premios a los números culturales.
<b>Actor:</b> Responsable de Cultura en la Universidad	

<b>Resumen:</b> El caso de uso se inicia cuando el Responsable de Cultura en la Universidad desea otorgar o quitar un premio.	
<b>Referencias</b>	RF-11, 12.
<b>Sección " Otorgar Premio "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El actor solicita la opción de de otorgar premios.</li> <li>2. El actor selecciona el evento al cual desea otorgarle los premios.</li> <li>3. El actor selecciona el número cultural al cual le asignará un premio y pulsa el botón guardar.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema visualiza la página donde están todos los Eventos Culturales creados.</li> <li>2.1. El sistema le muestra todos los números culturales del evento seleccionado, posibilitando otorgar premios.</li> <li>3.1. El sistema guarda la información y muestra un mensaje que se cumplió el proceso satisfactoriamente y da la posibilidad de asignar otro premio.</li> </ol>
<b>Sección " Quitar Premio "</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El actor solicita la opción de de quitar premios.</li> <li>2. El actor selecciona el evento al cual desea quitarle los premios.</li> <li>3. El actor selecciona el número cultural al cual le desea quitar el premio y pulsa el botón quitar.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema visualiza la página donde están todos los Eventos Culturales creados.</li> <li>2.1. El sistema le muestra todos los números culturales del evento seleccionado que fueron premiados, posibilitando quitar premios.</li> <li>3.1. El sistema le quita el premio al número cultural seleccionado y salva en la base de datos mostrando un mensaje que se cumplió el proceso satisfactoriamente.</li> </ol>
<b>Puntos de extensión.</b>	

Caso de uso	
CU-6	Mostrar Eventos Culturales.
<b>Propósito</b>	Mostrar los eventos culturales con todos sus datos.
<b>Actores:</b> Responsable de Cultura en la Universidad, Aficionados.	
<b>Resumen:</b> El caso de uso se inicia cuando uno de los actores solicita al sistema que le muestre los eventos culturales creados, este selecciona un evento determinado y este le muestra todos los datos que este tiene.	
<b>Referencias</b>	RF-6
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando cualquiera el actor solicita ver los eventos culturales.	1.1. El sistema muestra todos los eventos culturales creados.
2. El usuario selecciona el evento que desea visualizar sus datos.	2.1. El sistema realiza la búsqueda adecuada en la base de datos y visualiza la información solicitada.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<b>Puntos de extensión.</b>	

Caso de uso	
CU-6	Mostrar Premios Otorgados.
<b>Propósito</b>	Mostrar los números culturales a los cuales les fue otorgado un premio.
<b>Actores:</b> Responsable de Cultura en la Universidad, Aficionados.	

<b>Resumen:</b> El caso de uso se inicia cuando uno de los actores solicita al sistema que le muestre los números culturales que le fue asignado un premio.	
<b>Referencias</b>	RF-13
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el actor solicita que le muestre los números culturales que les fue asignado un premio. 2. El usuario selecciona el evento del cual desea obtener los números culturales premiados.	1.1. El sistema le muestra los eventos culturales que han sido creados. 2.1. El sistema realiza la búsqueda adecuada en la base de datos y visualiza la información solicitada.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Puntos de extensión.</b>	

<b>Caso de uso</b>	
CU-6	Mostrar Sistema de Emulación.
<b>Propósito</b>	Mostrar el Sistema de Emulación por el cual se registrarán los eventos culturales.
<b>Actores:</b> Responsable de Cultura en la Universidad, Aficionados.	
<b>Resumen:</b> El caso de uso se inicia cuando uno de los actores solicita al sistema que le muestre el sistema de emulación por el cual se registrarán los eventos culturales.	
<b>Referencias</b>	RF-10
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el actor solicita visualizar el sistema de emulación.	1.1. El sistema le muestra los eventos culturales.

2. El usuario selecciona el evento cultural de cual desea visualizar el sistema de emulación.	2.1. El sistema realiza la búsqueda adecuada en la base de datos y visualiza la información solicitada.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Puntos de extensión.</b>	

### Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución, y a través de los procesos de negocio y el levantamiento de requisitos, se pudo obtener un listado con las funciones que debe tener el sistema y se describieron paso a paso todas las acciones del actor del sistema con los casos de uso con los que interactúa. A partir de esto se comenzará a desarrollar la etapa posterior que sería el análisis y diseño.

### Capítulo 3. Análisis y Diseño del Sistema.

#### Introducción

En este capítulo se abordará el análisis y diseño de la aplicación en los aspectos más importantes en función de satisfacer los requerimientos esperados, se confecciona el diagrama de clases de análisis y los diagramas de interacción, se completa el modelo de clases de diseño y se aplica el diseño Web para las especificaciones de la aplicación. Se realiza además el diseño del modelo de datos para el sistema y un análisis del esfuerzo.

#### 3.1. Análisis.

##### *3.1.1. Modelo Conceptual de Clases de Análisis.*

El modelo conceptual se realizó utilizando el diagrama de clases de la notación UML de forma simplificada. Se utilizan las clases preliminares, las asociaciones preliminares en estas, la multiplicidad o cardinalidad para cada asociación y los nombres para las clases y las asociaciones.

El modelo conceptual de clases del análisis se muestra en el anexo 5.

#### 3.2. Diseño.

El modelo de diseño intenta preservar la estructura definida por el modelo de análisis. Es un modelo de objetos que describe la realización física de los casos de uso, además de tener en cuenta los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, entre otras. Otro de los aspectos a tener en cuenta del diseño es que crea una entrada apropiada y un punto de partida para actividades de implementación.

### **3.2.1. La vista de Interacción.**

Los objetos obran recíprocamente para implementar comportamiento. Esta interacción se puede describir de dos maneras complementarias, una de ellas se centra en los objetos individuales y la otra en una colección de objetos cooperantes. Una máquina de estados es una vista estrecha y profunda del comportamiento, una vista reduccionista que mira a cada objeto individualmente.

Una especificación de la máquina de estados es exacta y conduce inmediatamente al código. Sin embargo, puede ser difícil entender el funcionamiento total de un sistema, debido a que una máquina de estados se centra en un solo objeto a la vez, y se deben combinar los efectos de muchas máquinas de estado para determinar el comportamiento de todo el sistema. La vista de interacción proporciona una visión más integral del comportamiento de un sistema de objetos. Esta vista es modelada por colaboraciones.

### **3.2.2. Interacción.**

Una interacción es un conjunto de mensajes dentro de una colaboración que son intercambiados por roles de clasificador a través de roles de asociación. Cuando una colaboración existe en tiempo de ejecución, los objetos ligados a roles de clasificador intercambian instancias de mensajes a través de los enlaces ligados a los roles de asociación. Una interacción modela la ejecución de una operación, caso de uso, u otra entidad de comportamiento. Un mensaje es una comunicación unidireccional entre dos objetos, un flujo de objeto con la información de un remitente a un receptor. Un mensaje puede tener parámetros que transporten valores entre los objetos. Un mensaje puede ser una señal (una comunicación explícita entre objetos, con nombre, y asíncrona) o una llamada (la invocación síncrona de una operación con un mecanismo para el control, que retorna posteriormente al remitente). La creación de un nuevo objeto se modela como un evento causado por el objeto creador y recibido por la propia clase. El evento de creación está disponible para la nueva instancia como evento actual en la transición desde el estado inicial del objeto. Los mensajes se pueden ordenar en hilos secuenciales de control. Los

hilos separados representan conjuntos de mensajes concurrentes. La sincronización entre los hilos se modela mediante restricciones entre mensajes en diversos hilos. Una construcción de sincronización puede modelar divisiones del control, uniones del control, y bifurcaciones. El orden de los mensajes se puede mostrar en dos tipos de diagramas: un diagrama de secuencia (que se centra en las secuencias de tiempo de los mensajes) y un diagrama de colaboración (que se centra en las relaciones entre los objetos que intercambian los mensajes). {4}

### ***3.2.3. Diagrama de Secuencia.***

El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de secuencia, debido a que muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.

Los diagramas de interacción (secuencia) definidos se muestran en el anexo 6.

### ***3.2.4. Diagrama de Diseño Web del Sistema.***

El diagrama de diseño Web del sistema (diagrama de clases para diseño orientado a objetos) se obtiene como resultado del refinamiento del modelo conceptual y se basa fundamentalmente en los diagramas de interacción.

El diagrama de diseño Web del sistema se puede ver en el anexo 7.

### ***3.2.5. Descripción de las Clases.***

Para facilitar el análisis, se clasifican las clases en:

### 3.2.5.1. Clases de Control.

- ❖ Representan coordinación, secuencia, transacciones, y control de otros objetivos.
- ❖ Se usan mucho para encapsular el control de un caso de uso concreto.
- ❖ También derivaciones y cálculos complejos.
- ❖ Manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos (de interfaz y de entidad).

Las descripciones de las clases de Control pueden verse a continuación:

<b>Nombre:</b> Servicios_CulturalesImpl	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
eventos_culturalesDao	Eventos_CulturalesDao
sistema_emulacionDao	Sistema_EmulacionDao
manifestaciones_artisticasDao	Manifestaciones_ArtisticasDao
numeros_culturalesDao	Numeros_CulturalesDao
premiosDao	PremiosDao
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
crearEvento_Cultural	Permite crear un nuevo evento cultural.
modificarEvento_Cultural	Permite modificar un evento cultural creado.
eliminarEvento_Cultural	Elimina un evento cultural.
guardarEvento_Cultural	Permite guardar después de modificados los eventos culturales.

devolverEvento_Cultural	Devuelve un evento creado.
devolverTodos_Eventos_Culturales	Devuelve una lista con todos los eventos creados.
insertarManifesatciones_Artisticas	Permite insertar una manifestación artística.
devolverManifestacion_Artistica	Devuelve una manifestación artística.
devolverTodas_Manifestaciones_Artisticas	Devuelve una lista con todas las manifestaciones artísticas.
insertarNumero_Cultural	Permite insertar un número cultural.
devolverNumero_Cultural	Devuelve un número cultural.
devolverTodos_Numeros_Culturales	Devuelve una lista de números culturales.
otorgarPremio	Permite otorgar un premio.
quitarPremio	Permite quitar un premio.
devolverPremio_Otorgado	Devuelve un premio.
devolverTodos_Premios_Otorgados	Devuelve una lista de premios.
insertarSistema_Emulacion	Permite insertar un sistema de emulación.
modificarSistema_Emulacion	Permite modificar el sistema de emulación.
eliminarSistema_Emulacion	Elimina el sistema de emulación.
guardarSistema_Emulacion	Permite guardar el sistema de emulación después de modificado.
devolverSistema_Emulacion	Devuelve el sistema de emulación.

<b>Nombre:</b> Sistema_EmulacionDaolmpl	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
sistema_emulacion	Sistema_Emulacion
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>

insertarSistema_Emulacion	Permite insertar un sistema de emulación.
modificarSistema_Emulacion	Permite modificar el sistema de emulación.
eliminarSistema_Emulacion	Elimina el sistema de emulación.
guardarSistema_Emulacion	Permite guardar el sistema de emulación después de modificado.
devolverSistema_Emulacion	Devuelve el sistema de emulación.

<b>Nombre:</b> Eventos_CulturalesDaoImpl	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
eventos_culturales	Eventos_Culturales
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
crearEvento_Cultural	Permite crear un nuevo evento cultural.
modificarEvento_Cultural	Permite modificar un evento cultural creado.
eliminarEvento_Cultural	Elimina un evento cultural.
guardarEvento_Cultural	Permite guardar después de modificados los eventos culturales.
devolverEvento_Cultural	Devuelve un evento creado.
devolverTodos_Eventos_Culturales	Devuelve una lista con todos los eventos creados.

<b>Nombre:</b> Manifestaciones_ArtisticasDaoImpl	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
manifestaciones_artisticas	Manifestaciones_Artisticas
<b>Para cada responsabilidad:</b>	

Nombre	Descripción
insertarManifesatciones_Artisticas	Permite insertar una manifestación artística.
devolverManifestacion_Artistica	Devuelve una manifestación artística.
devolverTodas_Manifestaciones_Artisticas	Devuelve una lista con todas las manifestaciones artísticas.

<b>Nombre:</b> Numeros_CulturalesDaolmpl	
<b>Tipo de clase:</b> Control	
Atributo	Tipo
numeros_culturales	Numeros_Culturales
<b>Para cada responsabilidad:</b>	
Nombre	Descripción
insertarNumero_Cultural	Permite insertar un número cultural.
devolverNumero_Cultural	Devuelve un número cultural.
devolverTodos_Numeros_Culturales	Devuelve una lista de números culturales.

<b>Nombre:</b> PremiosDaolmpl	
<b>Tipo de clase:</b> Control	
Atributo	Tipo
premios	Premios
<b>Para cada responsabilidad:</b>	
Nombre	Descripción
otorgarPremio	Permite otorgar un premio.
quitarPremio	Permite quitar un premio.
devolverPremio_Otorgado	Devuelve un premio.
devolverTodos_Premios_Otorgados	Devuelve una lista de premios.

### 3.2.5.2. Clases de Entidad.

- ❖ Modelan información que poseen una larga vida.
- ❖ Modelan la información y el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.
- ❖ Derivan normalmente de una clase de entidad del negocio.
- ❖ Puede tener comportamiento complejo.
- ❖ Aísla los cambios en la información que represente.

Las descripciones de las Clases de Entidad pueden verse a continuación:

<b>Nombre:</b> Sistema_Emulacion	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idSistema_Emulacion	int
Descripcion	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setDescripcion	Se agrega el sistema de emulación.
getDescripcion	Devuelve el sistema de emulación.
setidSistema_Emulacion	Se le agrega un id al sistema de emulación
getDescripcion	Devuelve el id del sistema de emulación.

<b>Nombre:</b> Eventos_Culturales
<b>Tipo de clase:</b> Entidad

<b>Atributo</b>	<b>Tipo</b>
idEventos_Culturales	int
Nombre_Evento_Cultural	String
Lugar	String
Fecha	date
Hora	time
Tipo_Evento	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setIdEventos_Culturales	Se le agrega un id al evento cultural.
getIdEventos_Culturales	Devuelve el id del evento cultural.
setNombre_Evento_Cultural	Se le agrega un nombre al evento cultural.
getNombre_Evento_Cultural	Devuelve el nombre del evento cultural.
setLugar	Se agrega el lugar donde se realizará el evento cultural.
getLugar	Devuelve el lugar donde se realizará el evento cultural.
setFecha	Se agrega la fecha de realización del evento cultural.
getFecha	Devuelve la fecha de realización del evento cultural.
setHora	Se agrega la hora de comienzo del evento cultural.
getHora	Devuelve la hora de comienzo del evento cultural.
setTipo_Evento	Agrega el tipo de evento cultural.
getTipo_Evento	Devuelve el tipo de evento cultural.

<b>Nombre:</b> Manifestaciones_Artisticas	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idManifestaciones_Artisticas	int
Nombre_Manifestacion_Artistica	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setidManifestaciones_Artisticas	Se agrega un id a la manifestación artística.
getNombre_Manifestacion_Artistica	Devuelve el id de la manifestación artística.
setNombre_Manifestacion_Artistica	Se le agrega el nombre de la manifestación artística.
getNombre_Manifestacion_Artistica	Devuelve el nombre de la manifestación artística.

<b>Nombre:</b> Premios	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idPremios	int
Tipo_Premio	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setidPremios	Se agrega un id al premio.
getidPremios	Devuelve el id del premio.
setTipo_Premio	Se le agrega un tipo de premio.
getTipo_Premio	Devuelve el tipo de premio.

<b>Nombre:</b> Numeros_Culturales	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
idNumeros_Culturales	int
Nombre_Numero_Cultural	String
Integrantes	String
Tema	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setidNumeros_Culturales	Se agrega un id al número cultural.
getidNumeros_Culturales	Devuelve el id del número cultural.
setNombre_Numero_Cultural	Se le agrega el nombre del número cultural.
getNombre_Numero_Cultural	Devuelve el nombre del número cultural.
setIntegrantes	Se agregan los integrantes.
getIntegrantes	Devuelven los integrantes.
setTema	Se le agrega el tema musical.
getTema	Devuelve el tema musical.

### 3.2.5.3. Clases de Interfaz.

- ❖ Modelan la interacción entre el sistema y sus actores (usuarios u sistemas externos).
- ❖ Reciben y presentan información y peticiones de y hacia los actores.
- ❖ Reúnen los requisitos en los límites del sistema.

- ❖ Suelen ser abstracciones de ventanas, formularios, interfaces de impresoras, sensores, terminales...
- ❖ Describan lo que se obtiene con la interacción, no el proceso físico de cómo se ejecuta.
- ❖ Una clase de interfaz por cada actor, y viceversa.

Las descripciones de las Clases de Interfaz pueden verse a continuación:

<b>Nombre:</b> Sistema_EmulacionBean	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
idSistema_Emulacion	int
Descripcion	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setDescripcion	Se agrega el sistema de emulación.
getDescripcion	Devuelve el sistema de emulación.
setidSistema_Emulacion	Se le agrega un id al sistema de emulación
getDescripcion	Devuelve el id del sistema de emulación.
insertarSistema_Emulacion	Permite insertar un sistema de emulación.
modificarSistema_Emulacion	Permite modificar el sistema de emulación.
eliminarSistema_Emulacion	Elimina el sistema de emulación.
guardarSistema_Emulacion	Permite guardar el sistema de emulación después de modificado.
devolverSistema_Emulacion	Devuelve el sistema de emulación.
<b>Nombre:</b> Eventos_CulturalesBean	

<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
idEventos_Culturales	int
Nombre_Evento_Cultural	String
Lugar	String
Fecha	date
Hora	time
Tipo_Evento	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setIdEventos_Culturales	Se le agrega un id al evento cultural.
getIdEventos_Culturales	Devuelve el id del evento cultural.
setNombre_Evento_Cultural	Se le agrega un nombre al evento cultural.
getNombre_Evento_Cultural	Devuelve el nombre del evento cultural.
setLugar	Se agrega el lugar donde se realizará el evento cultural.
getLugar	Devuelve el lugar donde se realizará el evento cultural.
setFecha	Se agrega la fecha de realización del evento cultural.
getFecha	Devuelve la fecha de realización del evento cultural.
setHora	Se agrega la hora de comienzo del evento cultural.
getHora	Devuelve la hora de comienzo del evento cultural.
setTipo_Evento	Agrega el tipo de evento cultural.

getTipo_Evento	Devuelve el tipo de evento cultural.
crearEvento_Cultural	Permite crear un nuevo evento cultural.
modificarEvento_Cultural	Permite modificar un evento cultural creado.
eliminarEvento_Cultural	Elimina un evento cultural.
guardarEvento_Cultural	Permite guardar después de modificados los eventos culturales.
devolverEvento_Cultural	Devuelve un evento creado.
devolverTodos_Eventos_Culturales	Devuelve una lista con todos los eventos creados.

<b>Nombre:</b> Manifestaciones_ArtisticasBean	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
idManifestaciones_Artisticas	int
Nombre_Manifestacion_Artistica	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setidManifestaciones_Artisticas	Se agrega un id a la manifestación artística.
getIdManifestaciones_Artisticas	Devuelve el id de la manifestación artística.
setNombre_Manifestacion_Artistica	Se le agrega el nombre de la manifestación artística.
getNombre_Manifestacion_Artistica	Devuelve el nombre de la manifestación artística.
insertarManifesatciones_Artisticas	Permite insertar una manifestación artística.
devolverManifestacion_Artistica	Devuelve una manifestación artística.
devolverTodas_Manifestaciones_Artisticas	Devuelve una lista con todas las manifestaciones artísticas.

<b>Nombre:</b> Numeros_CulturalesBean	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
idNumeros_Culturales	int
Nombre_Numero_Cultural	String
Integrantes	String
Tema	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setidNumeros_Culturales	Se agrega un id al número cultural.
getidNumeros_Culturales	Devuelve el id del número cultural.
setNombre_Numero_Cultural	Se le agrega el nombre del número cultural.
getNombre_Numero_Cultural	Devuelve el nombre del número cultural.
setIntegrantes	Se agregan los integrantes.
getIntegrantes	Devuelven los integrantes.
setTema	Se le agrega el tema musical.
getTema	Devuelve el tema musical.
insertarNumero_Cultural	Permite insertar un número cultural.
devolverNumero_Cultural	Devuelve un número cultural.
devolverTodos_Numeros_Culturales	Devuelve una lista de números culturales.

<b>Nombre:</b> PremiosBean	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
idPremios	int

Tipo_Premio	String
<b>Para cada responsabilidad:</b>	
<b>Nombre</b>	<b>Descripción</b>
setidPremios	Se agrega un id al premio.
getidPremios	Devuelve el id del premio.
setTipo_Premio	Se le agrega un tipo de premio.
getTipo_Premio	Devuelve el tipo de premio.
otorgarPremio	Permite otorgar un premio.
quitarPremio	Permite quitar un premio.
devolverPremio_Otorgado	Devuelve un premio.
devolverTodos_Premios_Otorgados	Devuelve una lista de premios.

### ***3.2.6. Diseño de la Base de Datos del Sistema.***

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. La estructura tiene que ser desarrollada para la necesidad de las aplicaciones que la usaran, esto nos puede ayudar a realizar un proceso del negocio para alcanzar un valor agregado para el cliente.

La puesta en práctica de la base de datos es el paso final en el desarrollo de aplicaciones de soporte del negocio. Tiene que conformarse con los requisitos del proceso del negocio, que es la primera abstracción de la vista de la base de datos.

El diagrama de entidad relación de la base de datos puede verse en el anexo 8.

### 3.2.6.1. Descripción de las tablas.

El gestor de la base de datos a utilizar es estructurado por tanto es necesario llevar las clases a tablas, se partió para esto de definir las clases persistentes. Las clases simples se representaron como tablas y en las relaciones de 1...m se añadió una llave extranjera al extremo m.

A continuación pueden verse las descripciones de las tablas.

<b>Nombre: Eventos_Culturales</b>		
<b>Descripción:</b> Esta tabla contiene el listado de los Eventos Culturales.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idEventos_Culturales	INTEGER(10)	Identificador del Evento Cultural.
Nombre_Evento_Cultural	VARCHAR(255)	Nombre del Evento Cultural.
Lugar	VARCHAR(255)	Lugar donde se realizará el Evento Cultural.
Fecha	DATE	Día, mes y año del comienzo del Evento.
Hora	TIME(7)	Hora de comienzo del Evento.
Tipo_Evento	VARCHAR(255)	Tipo del Evento creado, que puede ser(Festival de Base, Festival UCI y Gala de Premiaciones)

<b>Nombre: Manifestaciones_Artísticas</b>		
<b>Descripción:</b> Esta tabla contiene el listado de las manifestaciones artísticas que contendrán los Eventos Culturales.		

Atributo	Tipo	Descripción
idManifestaciones _Artísticas	INTEGER(10)	Identificador de la Manifestación Artística.
Nombre_Manifestación _Artística	VARCHAR(255)	Nombre de la Manifestación Artística.

Nombre: Números _ Culturales		
<b>Descripción:</b> Esta tabla contiene el listado de los números culturales de los Eventos Culturales.		
Atributo	Tipo	Descripción
idNúmeros_Culturales	INTEGER(10)	Identificador del Número Cultural.
Nombre_Número	VARCHAR(255)	Nombre que lleva el número cultural.
Integrantes	VARCHAR(255)	Nombre de los artistas que componen el número cultural.
Tema	VARCHAR(255)	Tema musical que acompaña la presentación del número cultural.

Nombre: Premios		
<b>Descripción:</b> Esta tabla contiene el listado de los premios que se pueden otorgar.		
Atributo	Tipo	Descripción
idPremio	INTEGER(10)	Identificador del Premio.
Tipo_Premio	VARCHAR(255)	Nombre del Premio.

<b>Nombre: Sistema_Emulación</b>		
<b>Descripción:</b> Esta tabla contiene el listado de los premios que se pueden otorgar.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idSistema_Emulación	INTEGER(10)	Identificador del sistema de emulación.
Descripción	VARCHAR(255)	Descripción del sistema de emulación por el cual se registrará el Evento.

<b>Nombre: Integrante</b>		
<b>Descripción:</b> Esta tabla contiene el listado de los premios que se pueden otorgar.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idIntegrante	INTEGER(10)	Identificador del integrante.
Nombre	VARCHAR(255)	Nombre del integrante.
1er_Apellido	VARCHAR(255)	Primer apellido del integrante.
2do_Apellido	VARCHAR(255)	Segundo apellido del integrante.
Facultad	NUMERIC	El numero de la facultad a la que pertenece.
Año_Cursa	VARCHAR(255)	El año que cursa en la universidad.

### 3.3. Estimación de Costo-Beneficio.

La Estimación por Puntos de Caso de Uso es un método de estimación de esfuerzo de un proyecto de desarrollo de software a partir de los casos de uso.

#### Puntos de Casos de Uso

El primer paso es el cálculo de puntos de casos de uso sin ajustar. Lo cual se hace a partir de la siguiente ecuación:

**UUCP = UAW + UUCW** donde:

**UUCP:** Puntos de casos de uso sin ajustar.

**UAW:** Factor de peso del los actores sin ajustar.

**UUCW:** Factor de peso de los casos de uso sin ajustar.

#### Cálculo de UAW

Actor del Sistema:	Tipo de Actor	Factor de Peso
Responsable de Cultura en la Universidad	Complejo	3

Cantidad de actores de tipo complejo: 1

**UAW=** Sumatoria de la multiplicación de la cantidad de actores de un tipo específico con su factor de peso.

$$\mathbf{UAW = 1 \times 3}$$

$$\mathbf{UAW = 3}$$

#### Cálculo de UUCW

Éste se calcula por los casos de uso del sistema y su complejidad. La complejidad esta dada por la cantidad de secuencias atómicas que tenga el caso de uso.

Casos de Uso	Tipo de Caso de Uso	Factor de Peso
Gestionar Eventos Culturales	Medio	10
Agregar Manifestaciones Artísticas	Simple	5
Agregar Números Culturales	Simple	5
Gestionar Premios	Medio	10
Gestionar Sistema de Emulación	Medio	10
Mostrar Eventos Culturales	Simple	5
Mostrar Premios Otorgados	Simple	5
Mostrar Sistema de Emulación	Simple	5

Cantidad de casos de uso simples: 5

Cantidad de casos de uso medio: 3

**UUCW** = Sumatoria de la multiplicación de la cantidad de Casos de Uso de un tipo específico con su factor de peso.

$$\mathbf{UUCW} = ((5 \times 5) + (3 \times 10))$$

$$\mathbf{UUCW} = 55$$

Al ser **UAW = 3** y **UUCW = 55**

**UUCP = UAW + UUCW** sustituyendo tenemos:

$$\mathbf{UUCP} = 3 + 55$$

$$\mathbf{UUCP} = 58$$

Después de haber calculado los puntos de Casos de Uso sin ajustar se calculan los puntos de Casos de Uso ajustados con la siguiente ecuación:

**UCP = UUCP x TCF x EF** donde:

**UCP:** Puntos de Casos de Uso ajustados.

**UUCP:** Puntos de Casos de Uso sin ajustar.

**TCF:** Factor de complejidad técnica.

**EF:** Factor de ambiente.

### **Cálculo del Factor de Complejidad Técnica (TCF).**

Se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor asignado</b>
T1	Sistema distribuido	2	0
T2	Objetivos de performance o tiempo de respuesta	1	2
T3	Eficiencia del usuario final	1	1
T4	Procesamiento interno complejo	1	1
T5	El código debe ser reutilizable	1	0
T6	Facilidad de instalación	0.5	2
T7	Facilidad de uso	0.5	3
T8	Portabilidad	2	0
T9	Facilidad de cambio	1	3
T10	Concurrencia	1	0

T11	Incluye objetivos especiales de seguridad	1	2
T12	Provee acceso directo a terceras partes	1	0
T13	Se requieren facilidades especiales de entrenamiento	1	0

El Factor de Complejidad Técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso } i \times \text{Valor asignado } i)$$

$$TCF = 0.6 + 0.01 \times \Sigma ((2 \times 0) + (1 \times 2) + (1 \times 1) + (1 \times 1) + (1 \times 0) + (0.5 \times 2) + (0.5 \times 3) + (2 \times 0) + (1 \times 3) + (1 \times 0) + (1 \times 2) + (1 \times 0) + (1 \times 0))$$

$$TCF = 0.6 + (0.01 \times 11.5)$$

$$TCF = 0.6 + 0.115$$

$$TCF = 0.715$$

### Cálculo del Factor Ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos actores son los que se contemplan en el cálculo del Factor de ambiente.

Factor	Descripción	Peso	Valor asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5
E2	Experiencia en la aplicación	0.5	4
E3	Experiencia en trabajo orientado a objetos	1	3
E4	Capacidad del analista líder	0.5	3
E5	Motivación	1	5

E6	Estabilidad de los requerimientos	2	5
E7	Personal a tiempo completo	-1	3
E8	Dificultad del lenguaje de programación	-1	3

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso } i \times \text{Valor asignado } i)$$

$$EF = 1.4 - 0.03 \times \Sigma ((1.5 \times 5) + (0.5 \times 4) + (1 \times 3) + (0.5 \times 3) + (1 \times 5) + (2 \times 5) + (-1 \times 3) + (-1 \times 3))$$

$$EF = 1.4 - (0.03 \times 23)$$

$$EF = 0.71$$

Al ser **UUCP = 58**, **TCF = 0.715** y **EF = 0.71**

**UCP = UUCP x TCF x EF** sustituyendo tenemos:

$$UCP = 58 \times 0.715 \times 0.71$$

$$UCP = 29.4437$$

El esfuerzo en horas-hombre viene dado por:

**E = UCP x CF** donde:

**E:** Esfuerzo

**UCP:** Puntos de Casos de Uso ajustados (calculado anteriormente).

**CF:** Factor de conversión (para este tipo de proyecto 20 horas-hombre/Punto de Casos de Uso).

Al tener **UCP = 29.4437** y **CF = 20 horas-hombre/Punto de Casos de Uso**

**E = UCP x CF** sustituyendo tenemos:

$$E = 29.4437 \times 20 \text{ horas-hombre/Punto de Casos de Uso}$$

$$E = 588.874 \text{ horas-hombre}$$

Actividad	Porcentaje	Esfuerzo
Análisis	10%	147
Diseño	20%	294
Implementación	40%	589
Prueba	15%	221
Sobrecarga	15%	221
Total	100%	1472

El proyecto requerirá de 1472 horas-hombre a desarrollar por una persona en un tiempo de 184 días; tomando que se trabajará en el proyecto como promedio 8 horas diarias. Por lo cual se puede decir que se terminará el proyecto en aproximadamente 6 meses ajustándose a los requerimientos del cliente.

### **Beneficios:**

El presente trabajo, como parte de las aplicaciones que se desarrollan con el objetivo de ser usadas dentro de la Universidad de las Ciencias Informáticas, no reporta un beneficio monetario directo. Sin embargo; desde el punto de vista de los beneficios intangibles que reporta se puede decir, que con el sistema para la Gestión de los Eventos Culturales de la FEU, se mejorará el manejo de la información relacionada con estas actividades.

### **Conclusiones**

En este capítulo se realizó el análisis y diseño donde se relaciona los requisitos funcionales y los no funcionales, llevándolo a una expresión de la programación entre las diferentes herramientas mencionadas en capítulos anteriores y algunos aspectos importantes para la necesidad del usuario, a su vez la preparación para la etapa de implementación del sistema. Además se realizó el análisis del esfuerzo del proyecto, donde pudimos ver la duración y costo del proyecto.

### Conclusiones Generales

Una vez culminada la realización de este trabajo, se llegó a la conclusión que el objetivo fundamental de éste fue logrado. El análisis y diseño de un sistema Web que gestione los Eventos Culturales de la FEU en la UCI permitirá un eficiente manejo de la información relacionada con estas actividades culturales por parte de los usuarios.

Se considera además que el presente trabajo ha cumplido con todos los objetivos propuestos:

- ❖ Se identificaron todas las necesidades de los clientes.
- ❖ Los procesos y especificaciones del sistema han sido descritos.
- ❖ Se definió el comportamiento del sistema en su interacción con el medio, esto se realizó mediante los Diagramas de Interacción.
- ❖ Se obtuvieron los Diagramas de Clases del Análisis y el Diagrama de Clases del Diseño.
- ❖ Se construyó el Diagrama de Entidad Relación, modelo que facilita la generación y creación de la Base de Datos del Sistema.

### Recomendaciones

Hechas las conclusiones del trabajo, se recomienda:

1. La completa terminación del sistema para la gestión de los Eventos Culturales de la FEU en la UCI, con la finalidad de explotar todas las funcionalidades que se diseñaron, y de esta manera se pueda hacer un óptimo uso de sus capacidades de funcionamiento y se puedan aprovechar al máximo las ventajas que este ofrece al personal que trabajará con el sistema.
2. Utilizar las herramientas propuestas para el desarrollo del sistema por parte del grupo de programadores.

### Referencias Bibliográficas

1. STALLMAN, R. M. “*El movimiento del Software Libre y el sistema operativo GNU/Linux*” En 2007.
2. SHEN, D. Y. “*Integración de JSF, Spring e Hibernate para crear una Aplicación Web del Mundo Real*”. 2006, Disponible en: [http://programacion.com/tutorial/jap\\_jsfwork/](http://programacion.com/tutorial/jap_jsfwork/).
3. SANCHEZ, M. A. M. “*Metodologías De Desarrollo De Software*”. 2004, Disponible en: [www.informatizate.net/articulos/pdfs/](http://www.informatizate.net/articulos/pdfs/).
4. BOOCH, G. “*The Unified Modeling Language User Guide*”. Addison-Wesley 1999. 482 p.

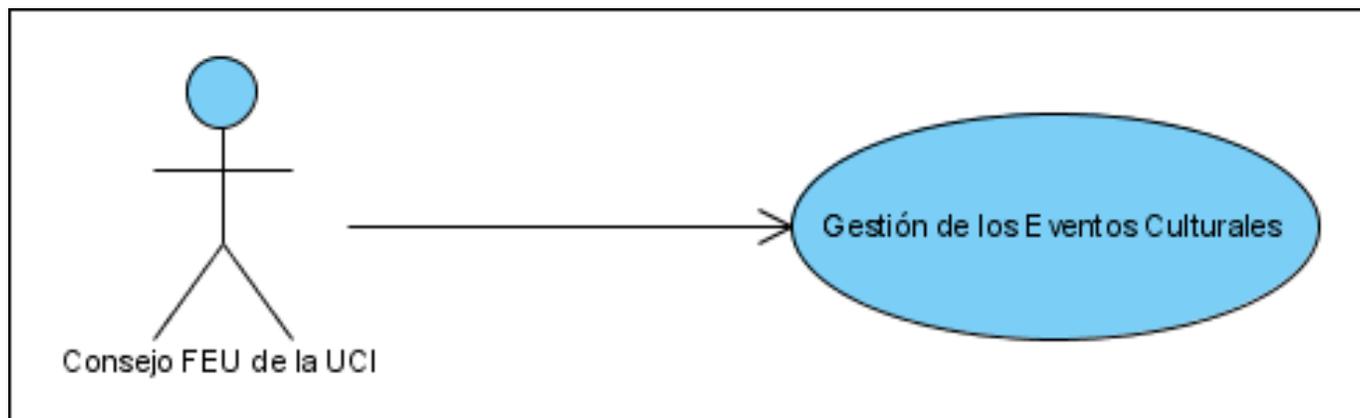
## Bibliografía

1. GARCIA, R. M. M. *Diseño de Bases de Datos*. 1999.
2. GONZÁLEZ, H. S. *Manual Hibernate*. 2003, Disponible en: <http://www.javahispano.org/articles.article.action?id=82>.
3. GRADY BOOCH, I. J., JAMES RUMBAUGH. *El proceso unificado del desarrollo de software* 1999, Disponible en: <http://www.monografias.com/trabajos22/desarrollo-software/desarrollo-software.shtml>.
4. LARMAN, C. *La Captura de Requerimientos*. Venezuela: Universidad Simón Bolívar, Departamento de Computación y Tecnología de la Información, [Consultado el: 05/04 de 2007]. Disponible en: <http://www ldc.usb.ve/~teruel/ci4712/clases/clase2.html>.
5. ---. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Segunda ed. 2002. 499 p.
6. MOLPECERES, A. *Procesos de desarrollo: RUP, XP y FDD*. 2002, Disponible en: <http://www.javahispano.org/articles.article.action?id=76>.
7. PULIDO, A. S. *Introducción al UML*. 2001, Disponible en: <http://www.yoprogramo.com/articulo4.php>.
8. SMALDONE, J. *Software Libre versus Software Propietario* [Consultado el: 08/12 de 2006]. Disponible en: <http://www.smaldone.com.ar/opinion/slvssp.shtml>.

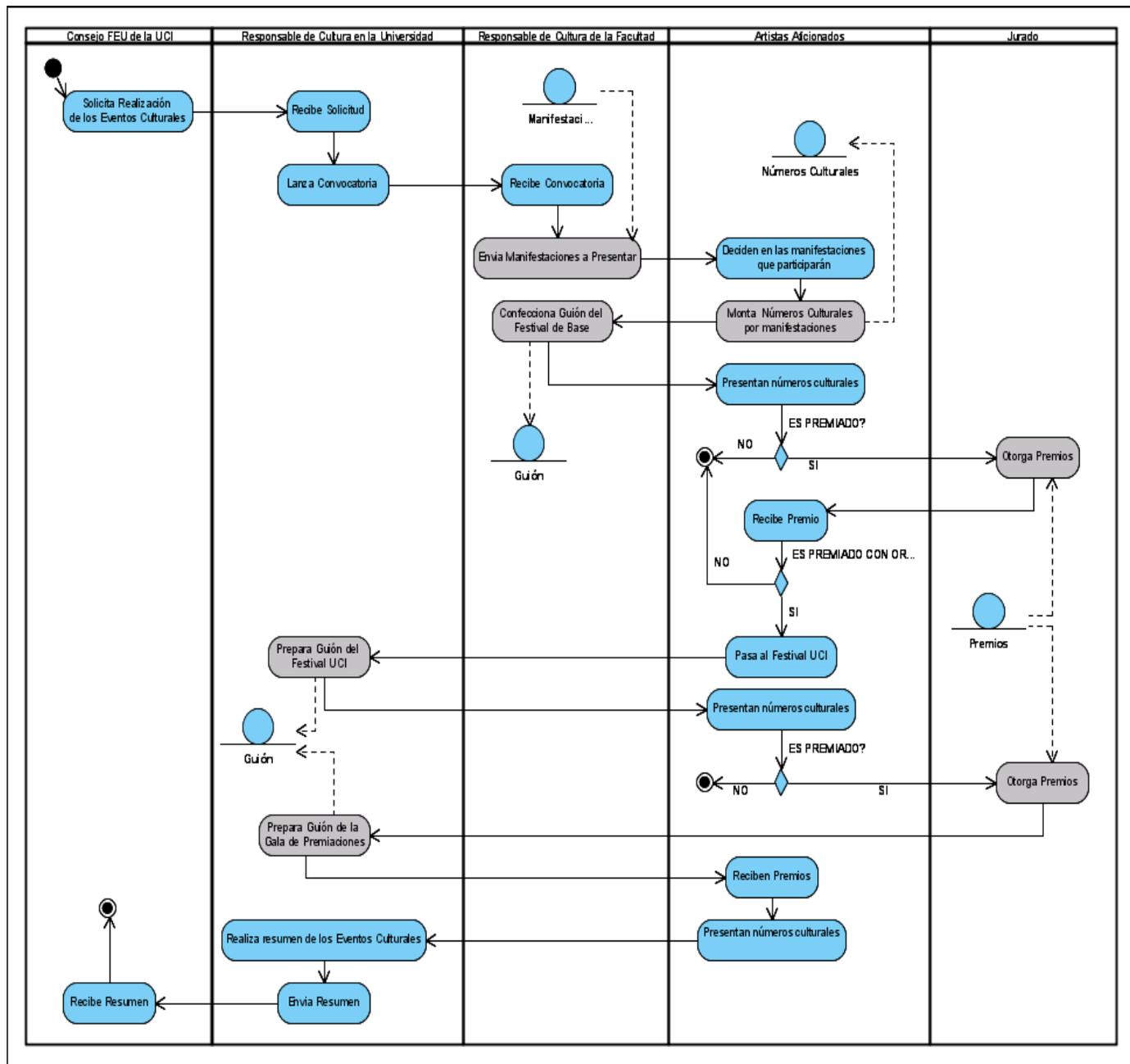
9. VILAS, A. F. *Diagramas de Casos de Uso* [Consultado el: 24/02 de 2007]. Disponible en: <http://www.javahispano.org/articles.article.action?id=82>.
10. YOEMNY GONZALEZ, A. Y. P. *Sistema para la Informatización de los Servicios Médicos de la UCI*. Computación. Universidad de la Habana, 2004.

**Anexos.**

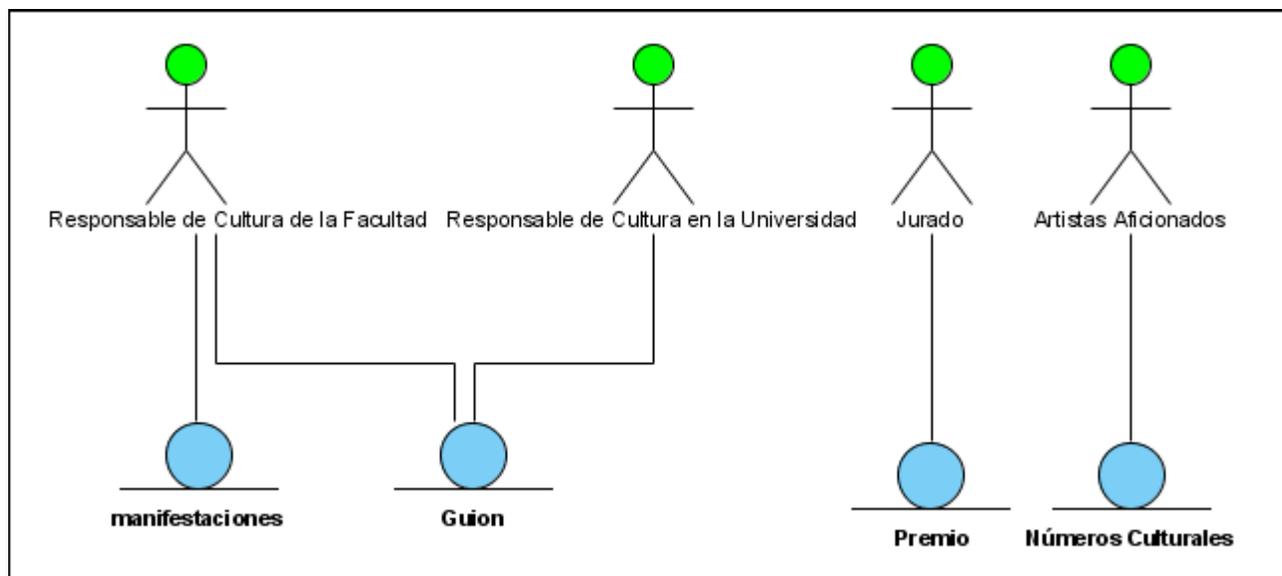
**Anexo 1. Diagrama de Casos de Uso del Negocio.**



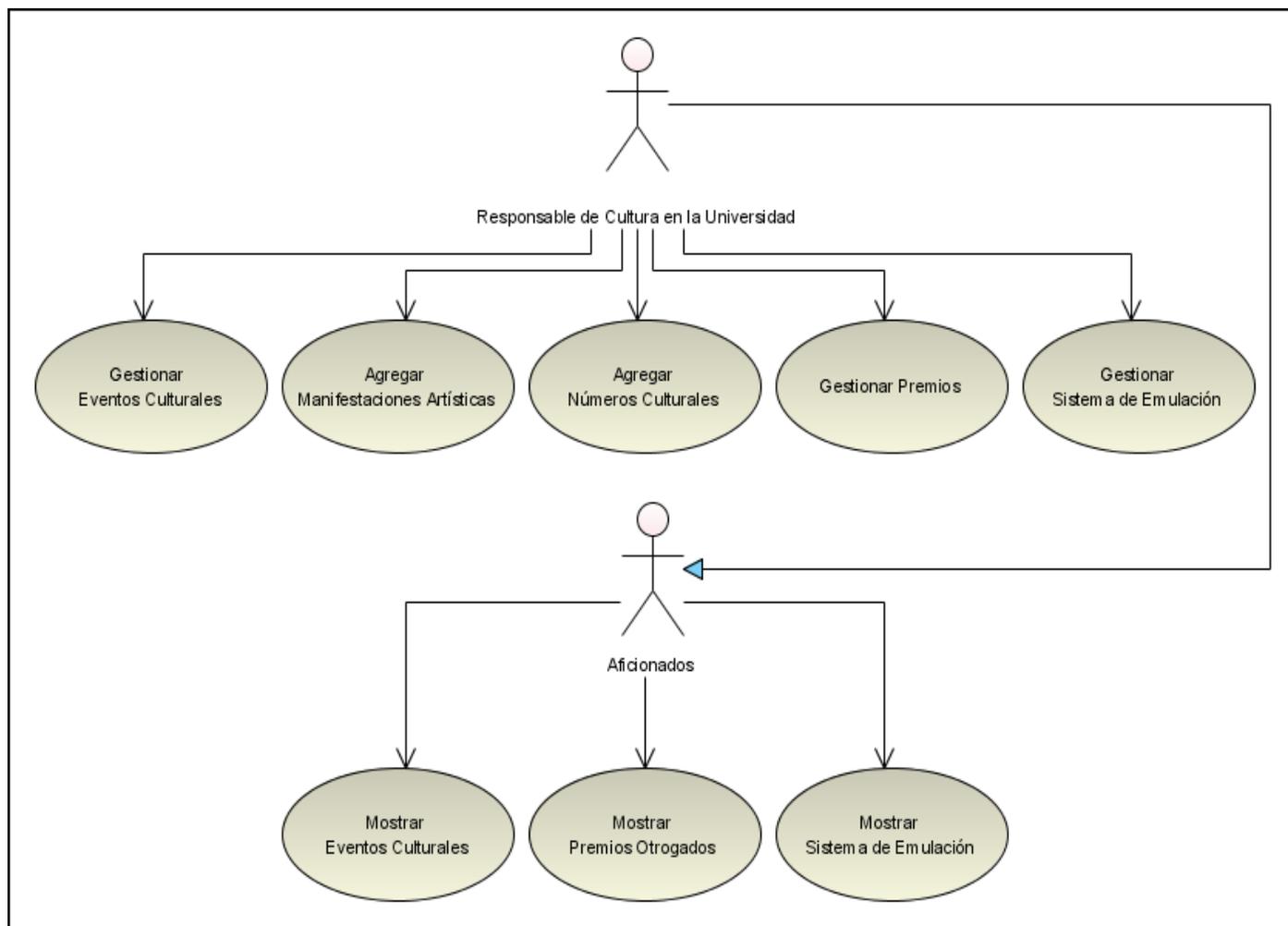
## Anexo 2. Diagrama de Actividades del caso de Uso del Negocio.



**Anexo 3. Diagrama de Clases del Modelo de Objeto del Negocio.**



**Anexo 4. Diagrama de Casos de Uso del Sistema.**



### Anexo 5. Diagramas del Modelo de Clases del Análisis.

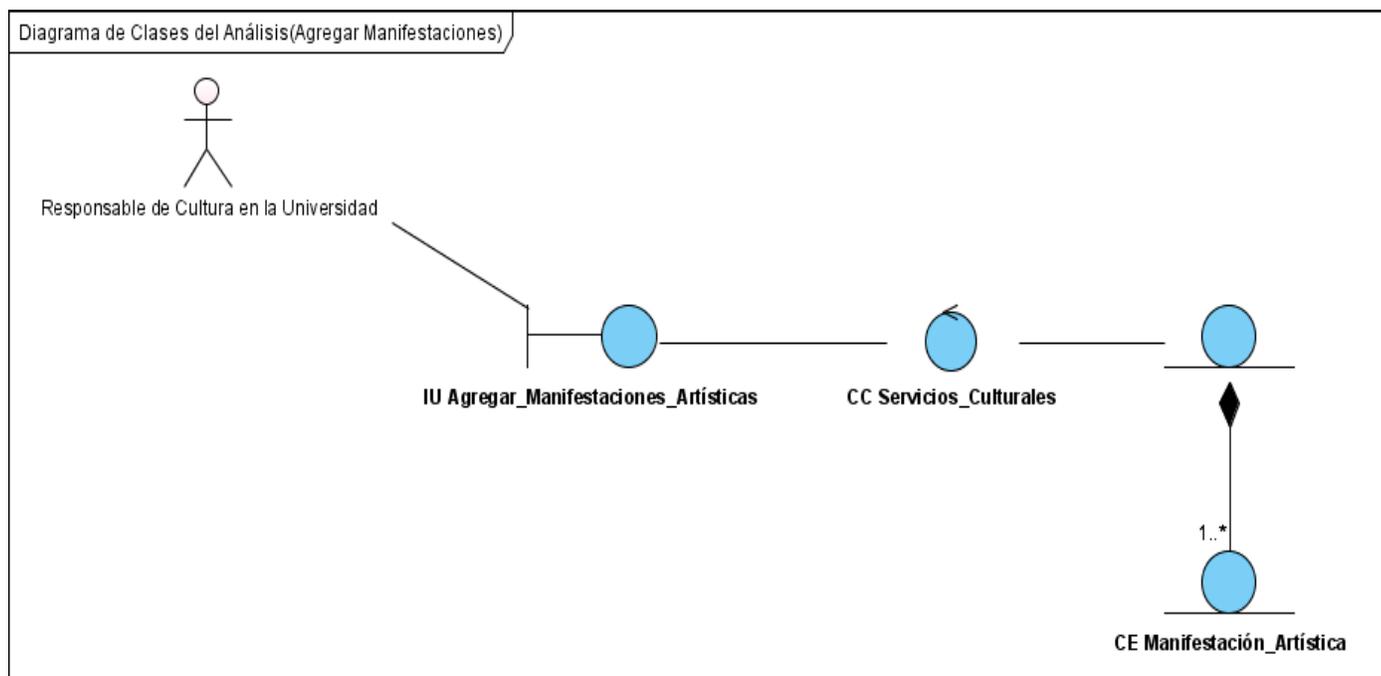
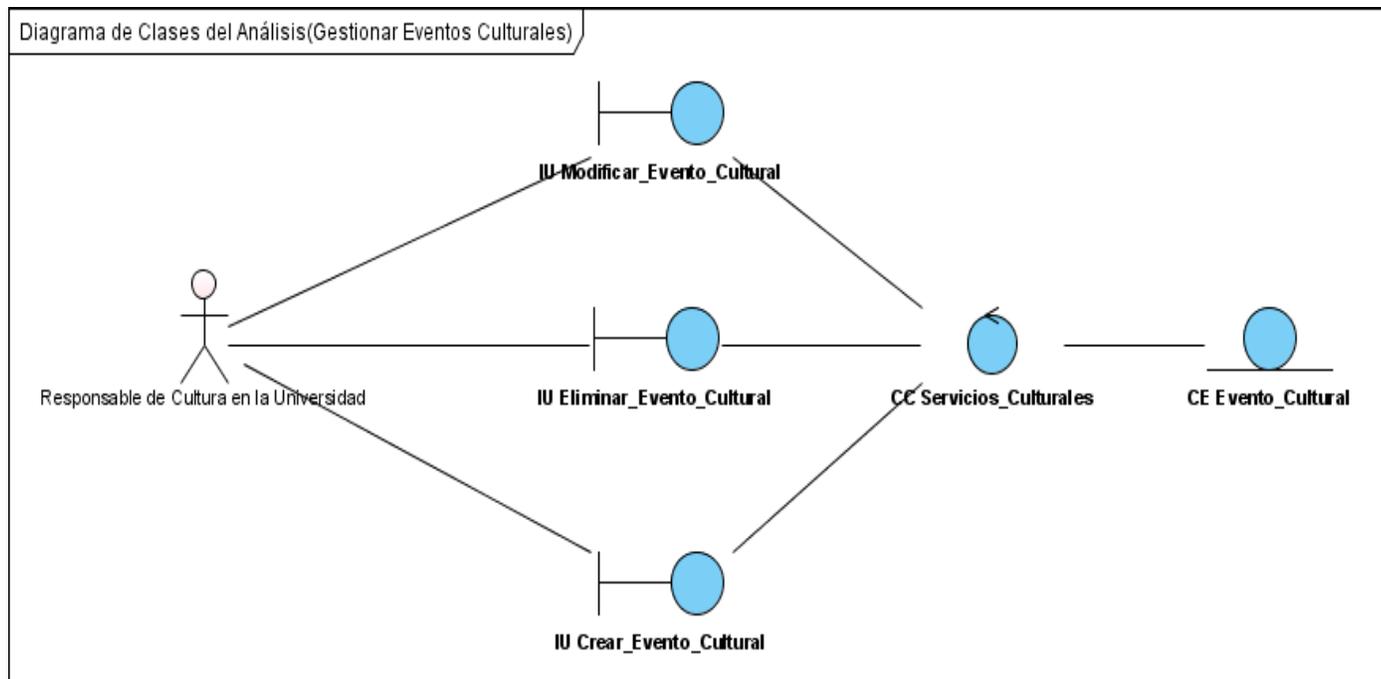


Diagrama de Clases del Análisis(Agregar Números Culturales)

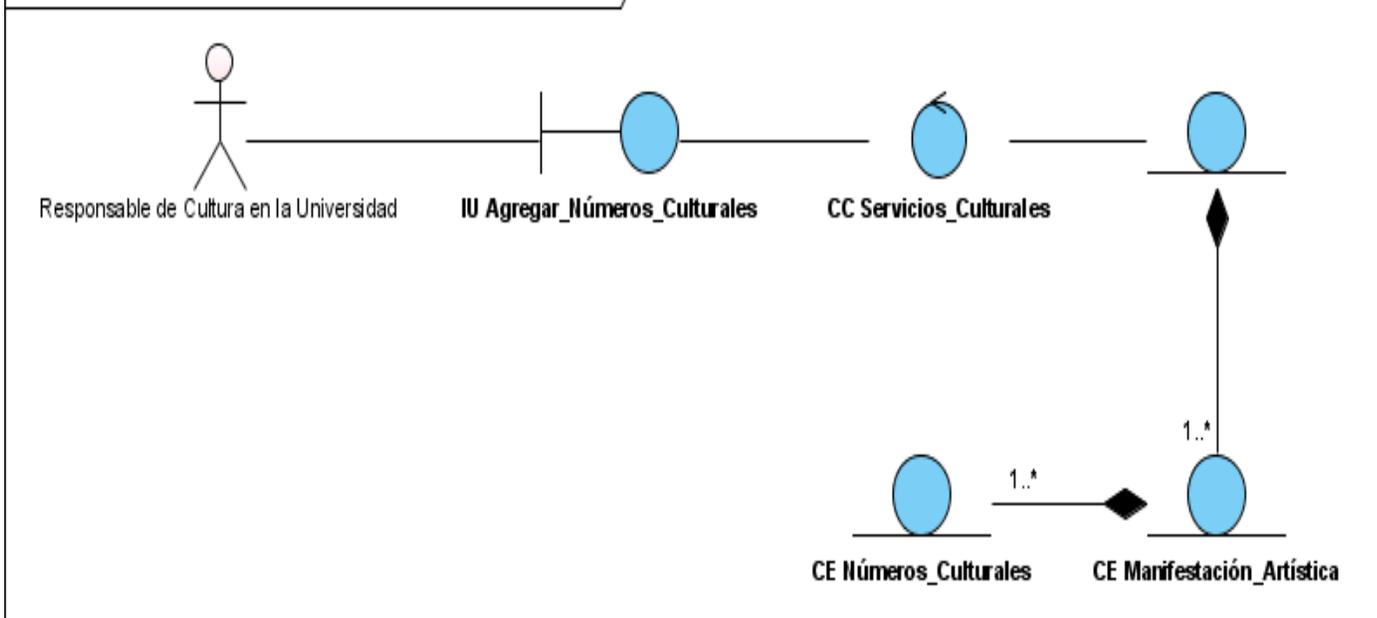


Diagrama de Clases del Análisis(Gestionar Premios)

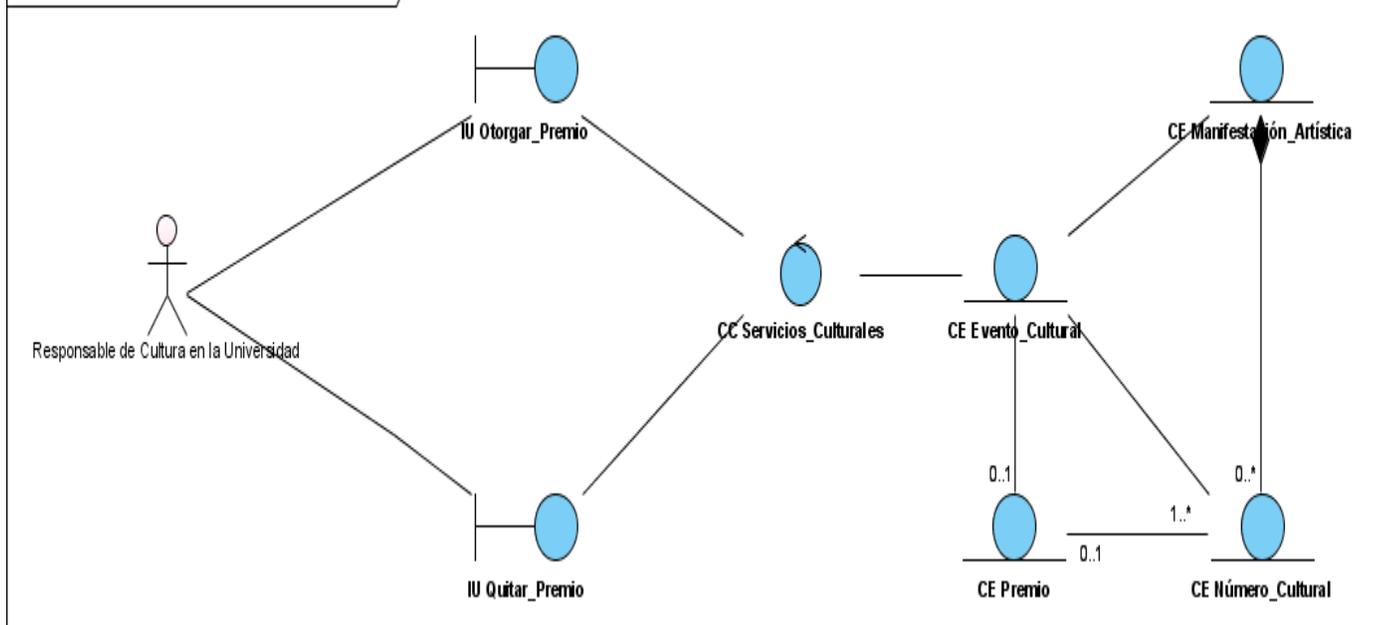


Diagrama de Clases del Análisis(Gestionar Sistema de Emulación)

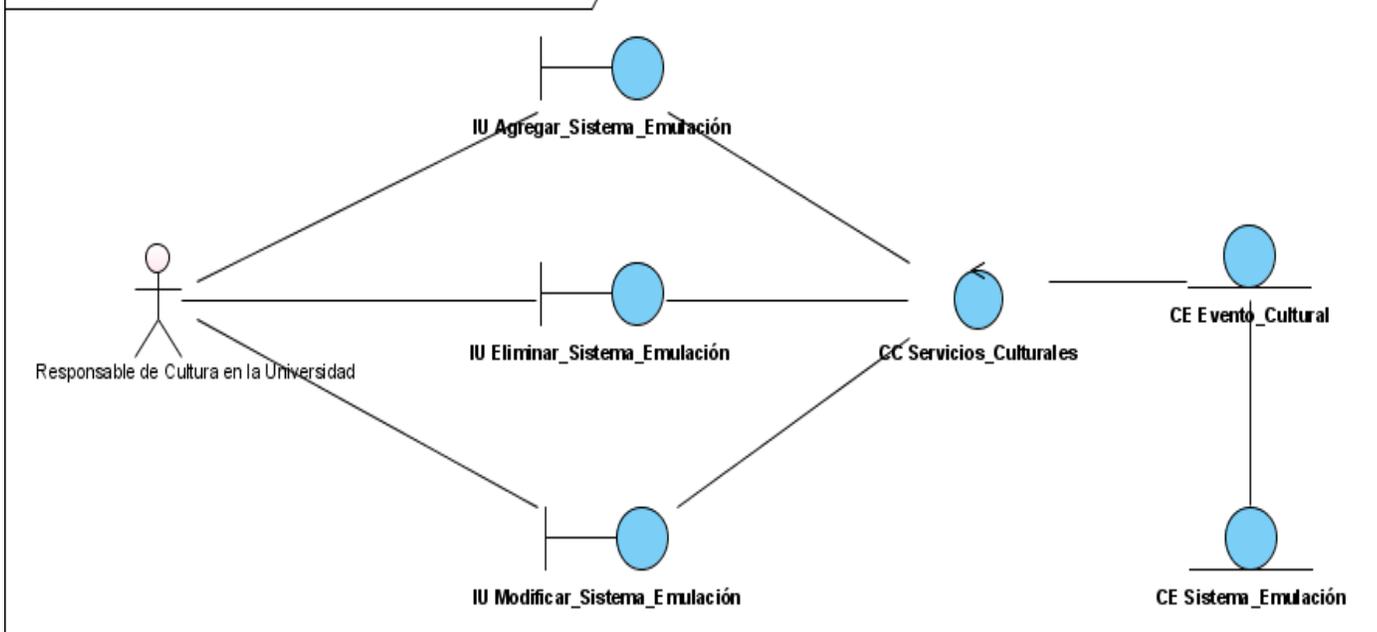


Diagrama de Clases del Analisis(Mostrar Eventos)

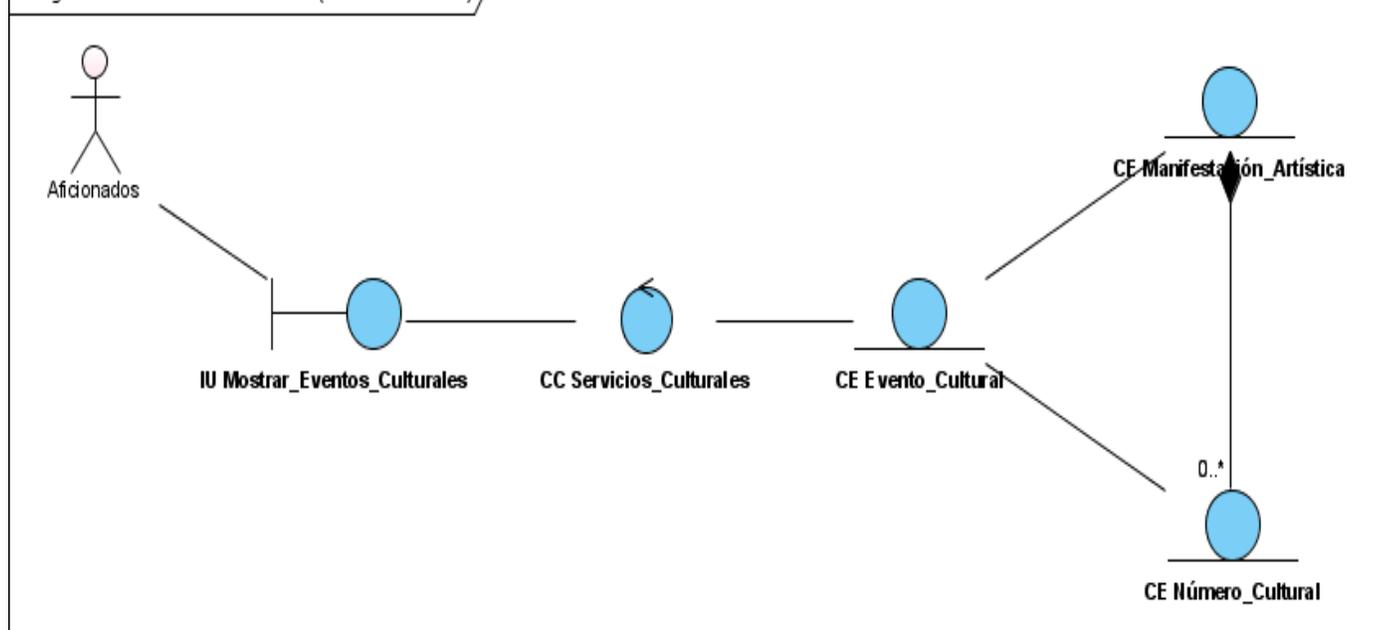


Diagrama de Clases del Analisis(Mostrar Premios Otorgados)

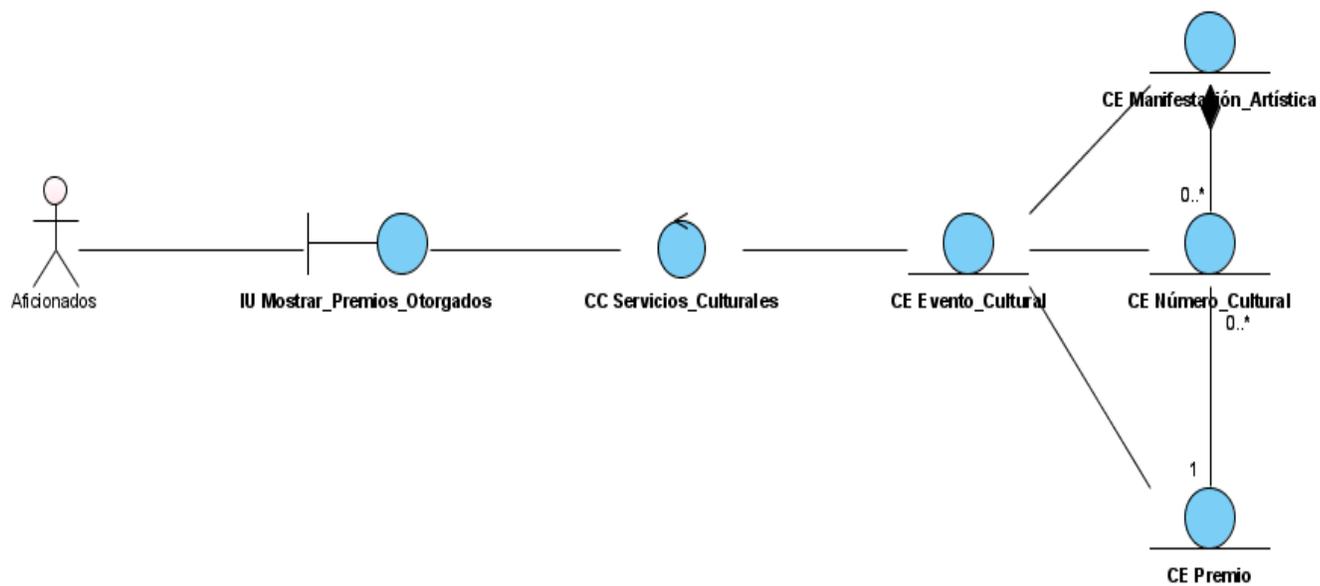
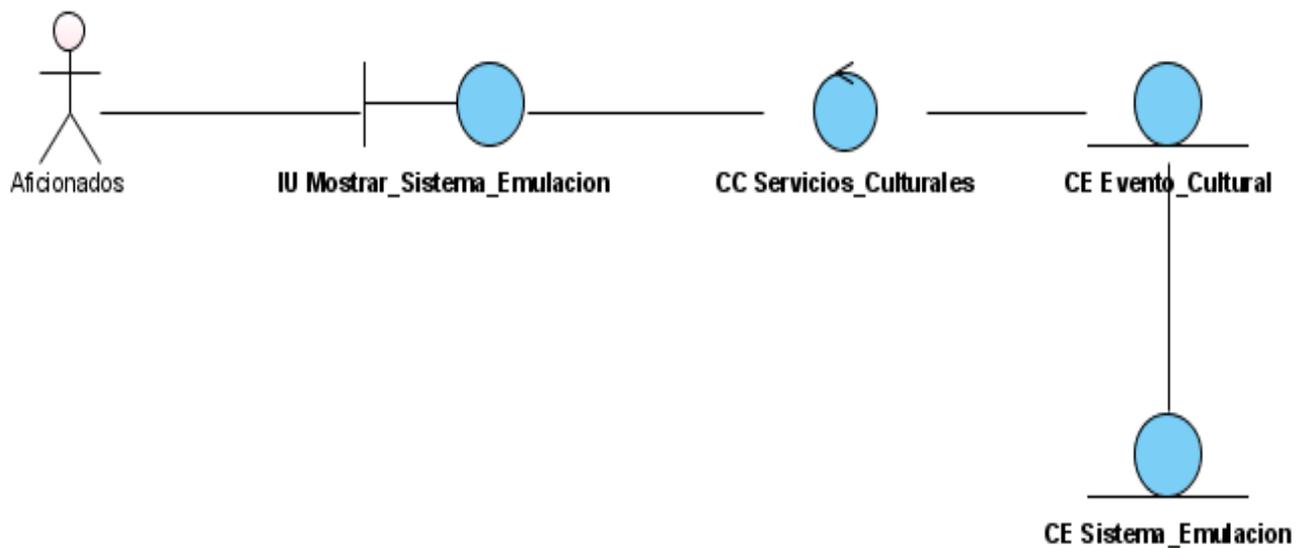
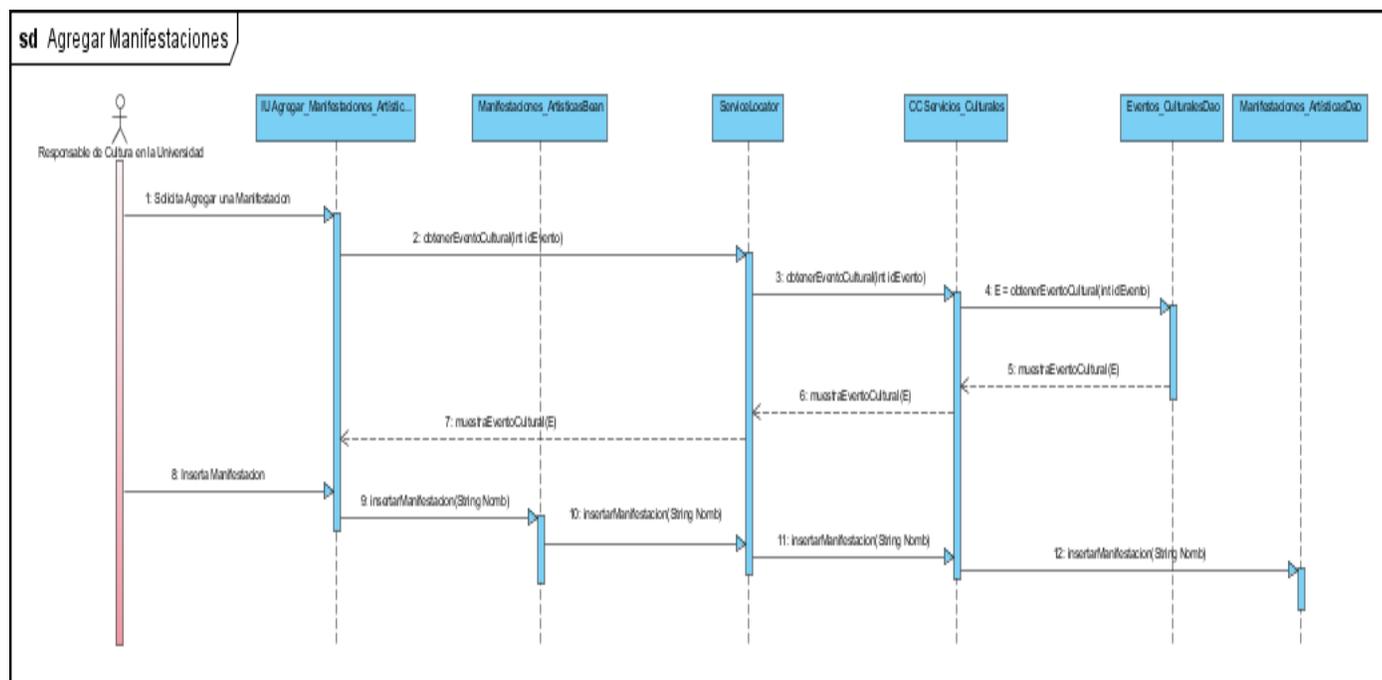
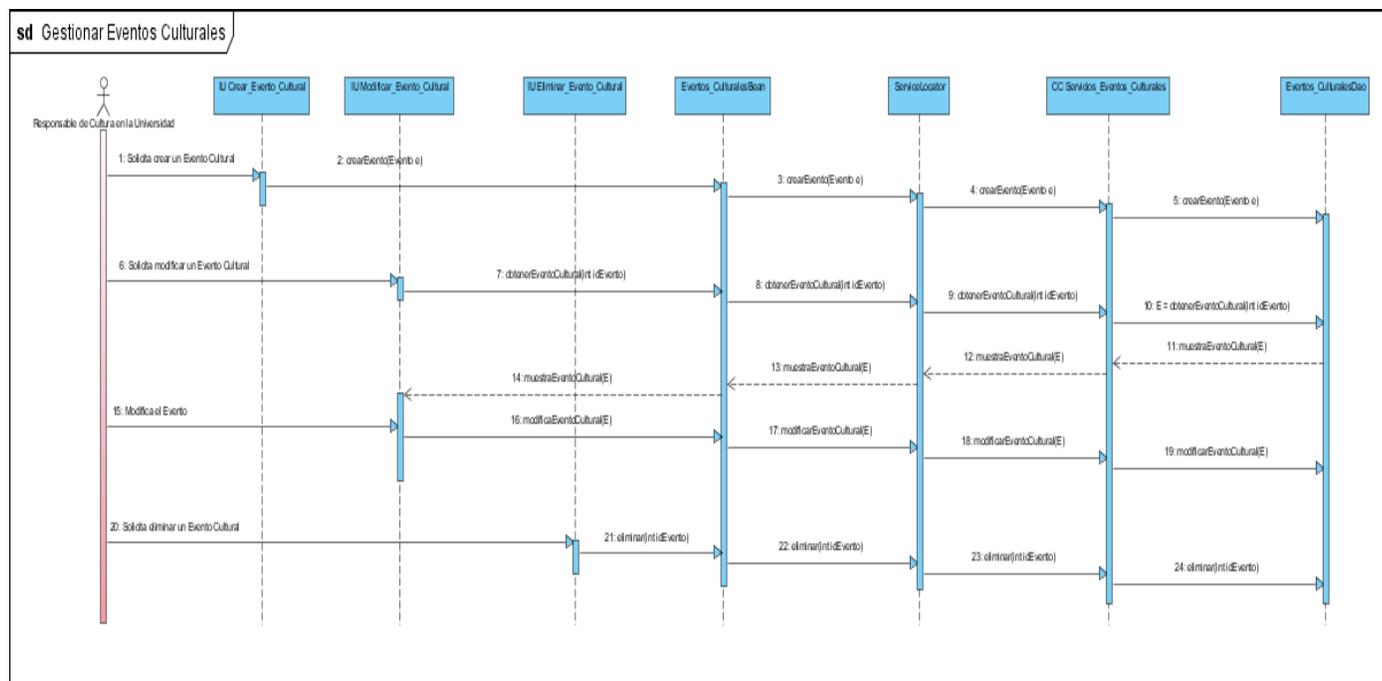
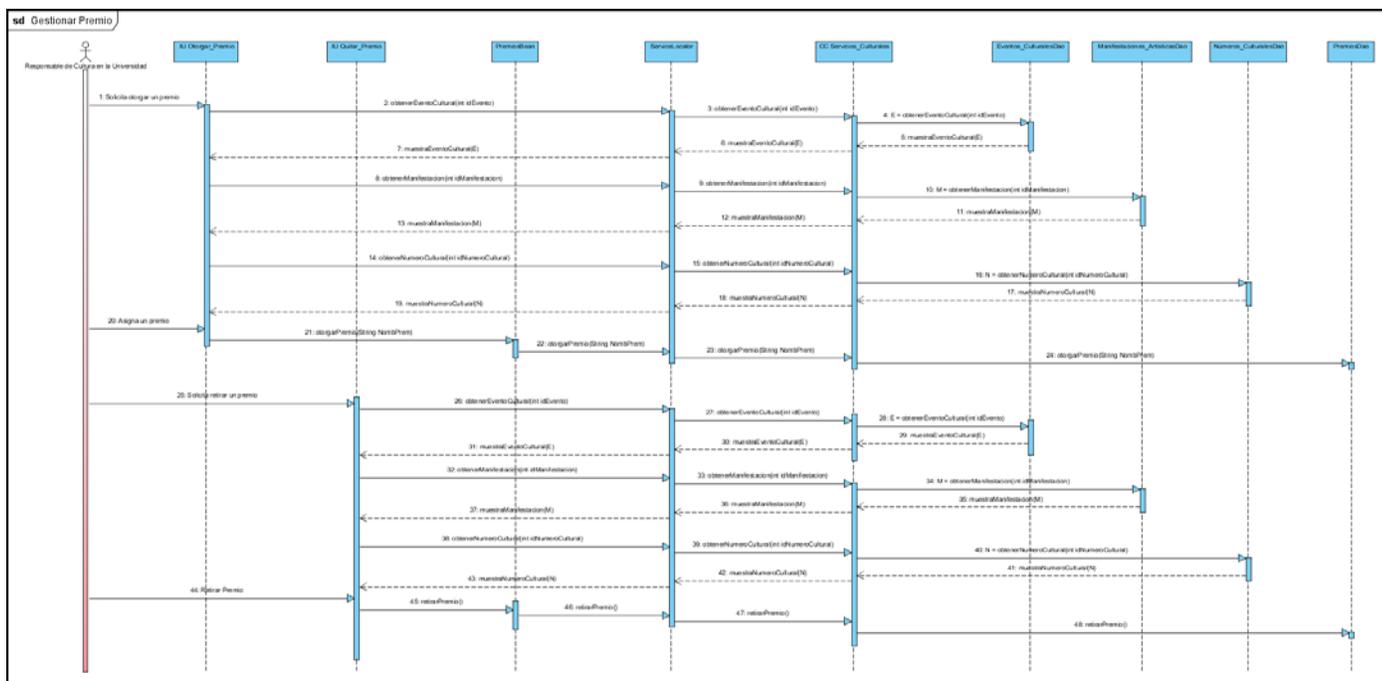
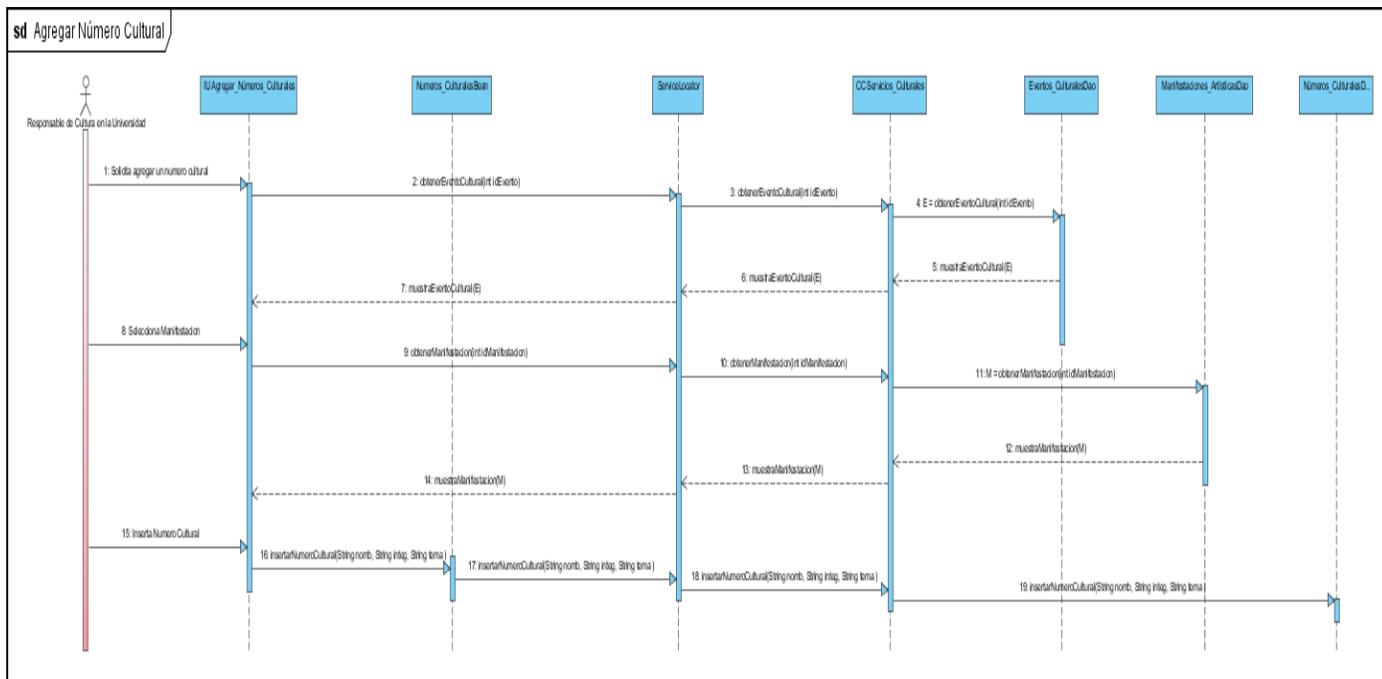


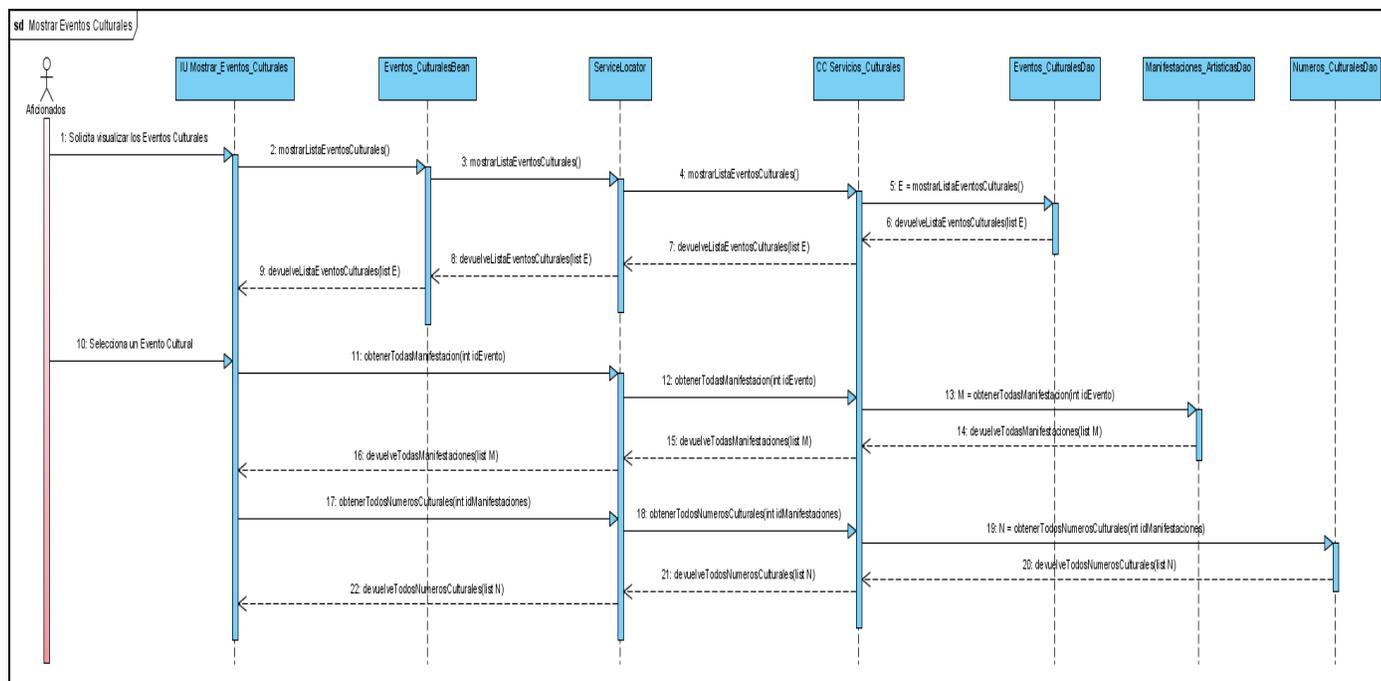
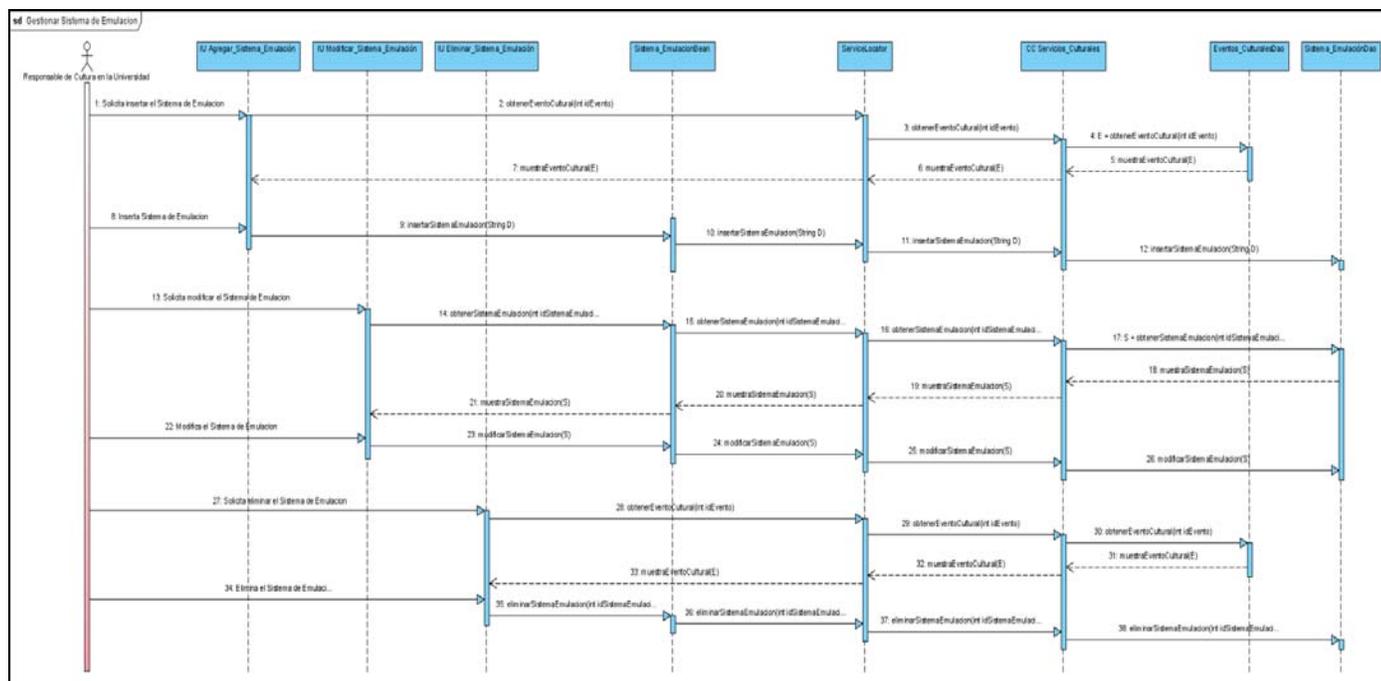
Diagrama de Clases del Analisis(Mostrar Sistema de Emulacion)

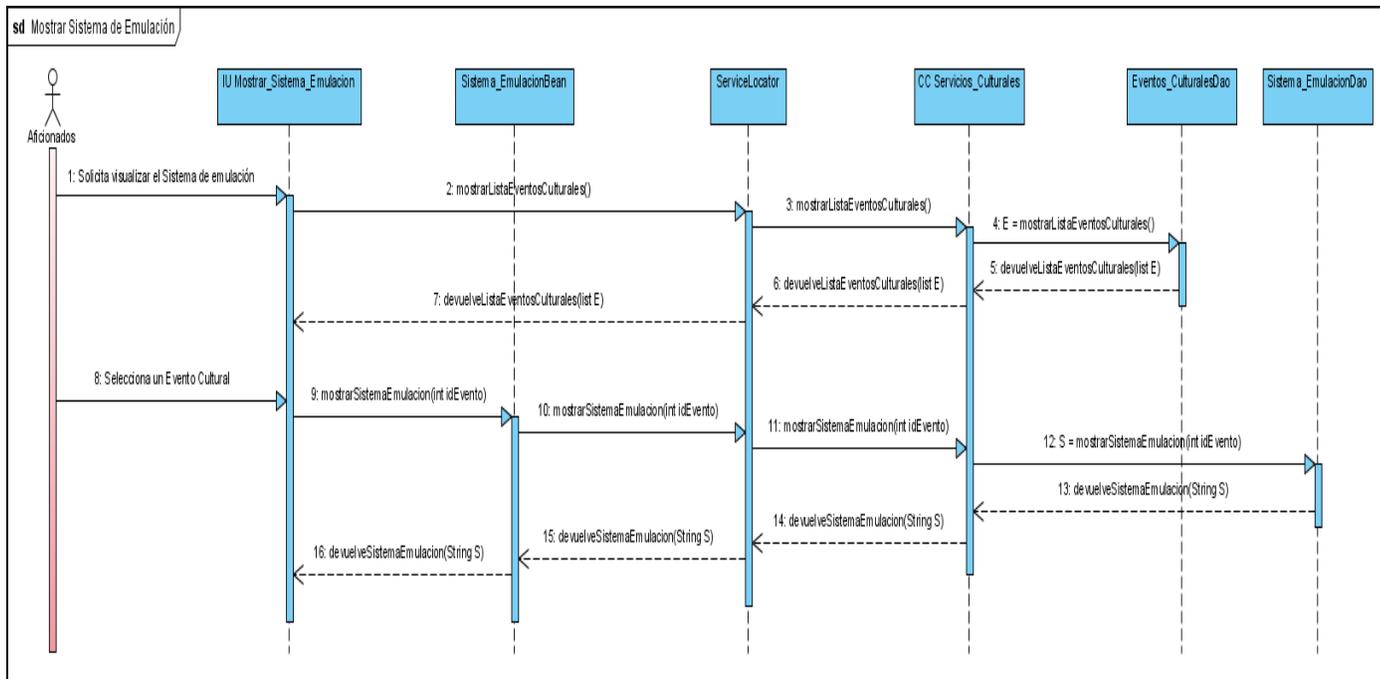
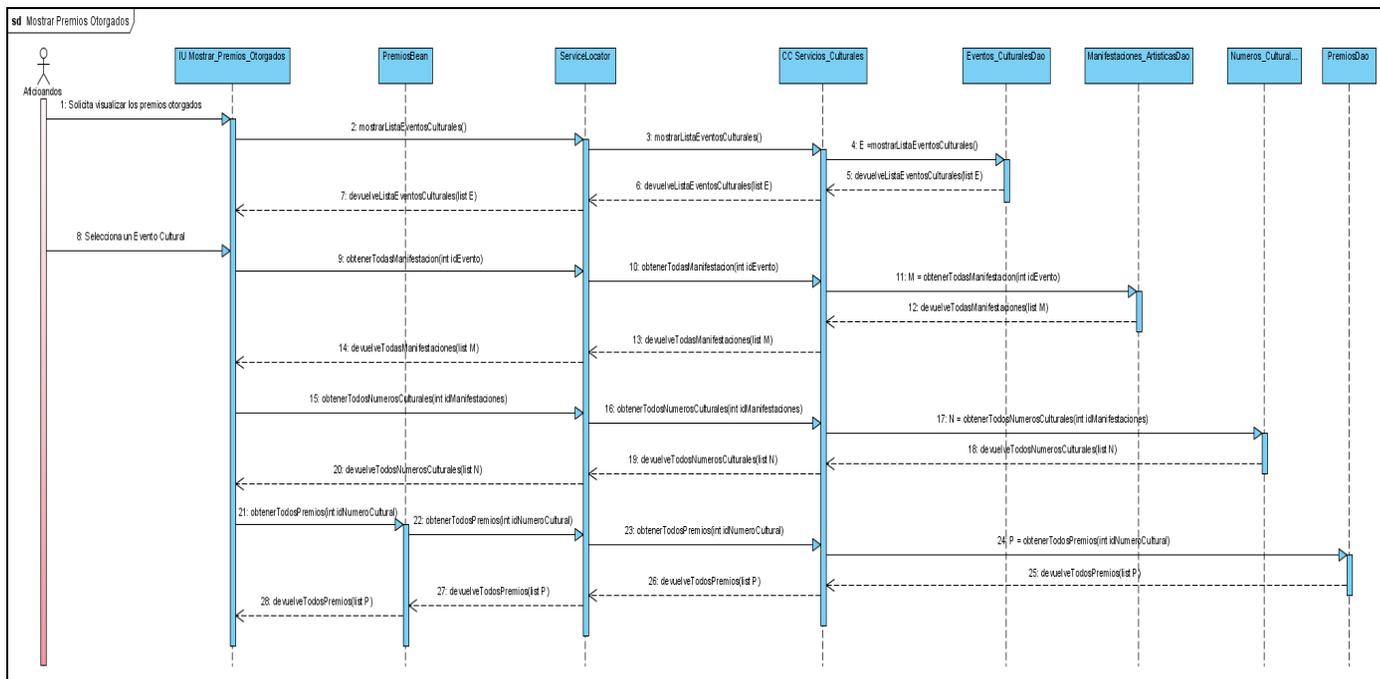


## Anexo 6. Diagramas de Interacción (Secuencia).

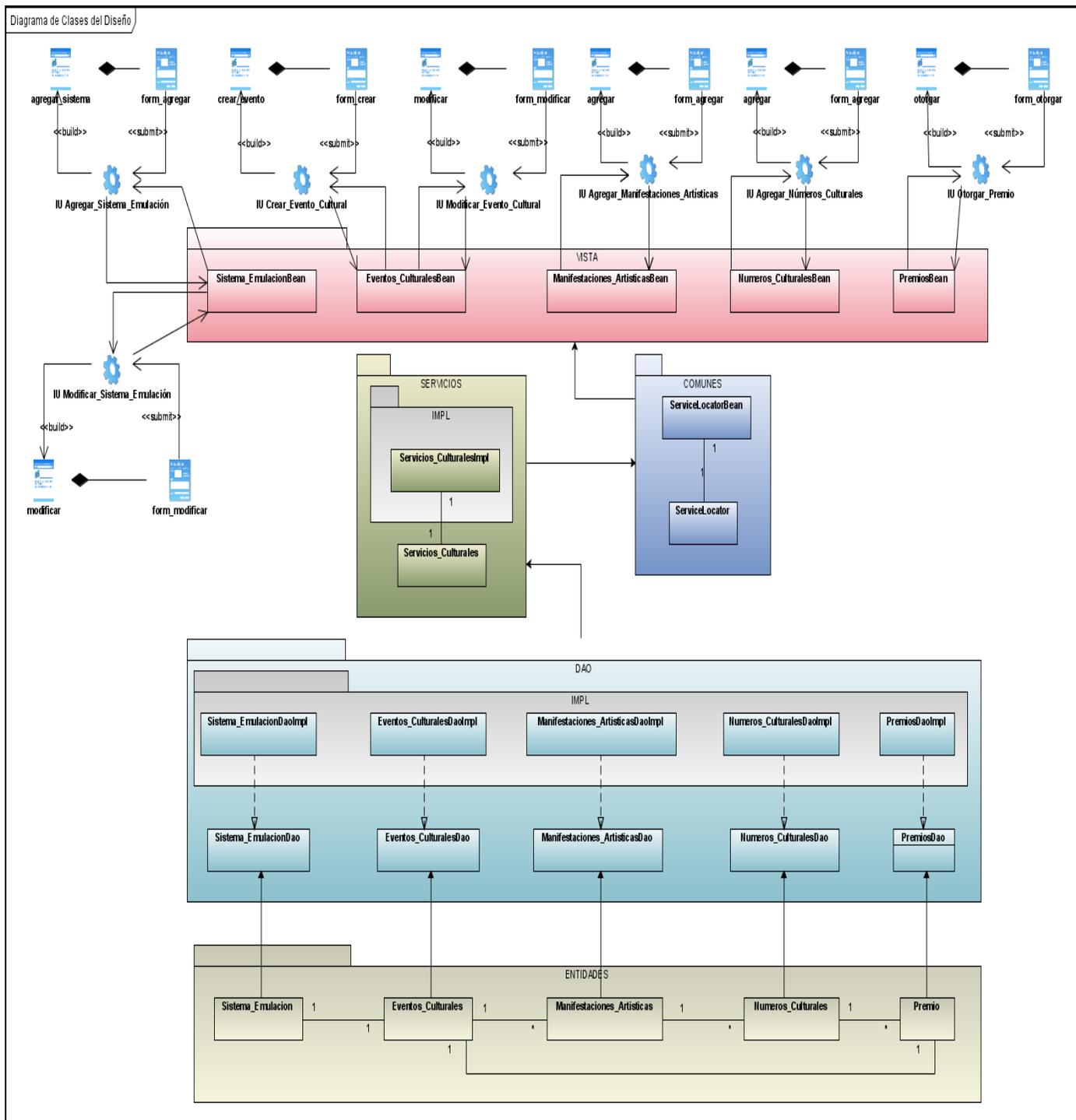




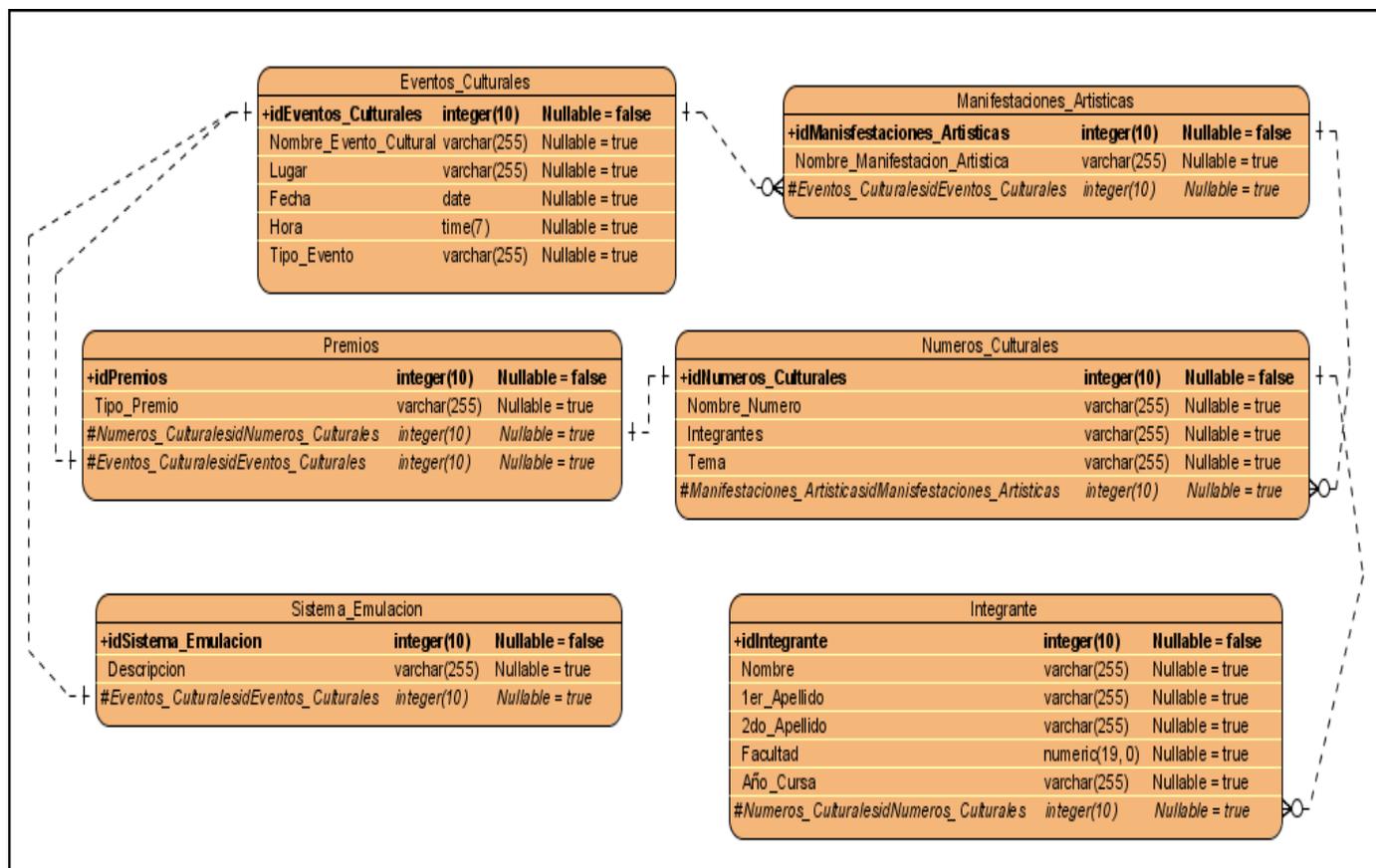




## Anexo 7. Diagrama de Diseño Web del Sistema.



### Anexo 8. Diagrama de Entidad Relación de la Base de Datos.



### Glosario de Términos

#### Siglas

- TICS (Tecnologías de la Informática y las Comunicaciones)
- FEU (Federación Estudiantil Universitaria)
- UCI (Universidad de las Ciencias Informáticas)
- RUP (Rational Unified Process)
- UML (Unified Modeling Language)
- J2EE (Java 2 Enterprise Edition)
- JSF (Java Server Faces)
- JSP (Java Server Pages)
- API (Application Programming Interface)
- DAO (Data Access Object)
- HTML (Hypertext Markup Language)
- IDE (Integrated Development Environment)

#### Términos

##### API

Una API (Del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones, interfaz de programación de la aplicación) es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación.

##### Base de Datos

Una base de datos de consta de una colección de tablas que contienen datos y otros objetos, como vistas, índices, procedimientos almacenados y desencadenadores, que se definen para poder llevar a cabo distintas operaciones con datos.

### **Cliente-Servidor**

Cuando se menciona este término no se piensa en otra cosa más que en bases de datos, dado que generalmente (Y de manera incorrecta) este término se usa como sinónimo de esto. Este término, en su más amplia definición, se usa para describir una aplicación en la cual dos o más procesos separados trabajan juntos para completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular. Esta operación se conoce como Proceso Cooperativo, dado que dos procesos separados cooperan para completar la tarea en particular.

### **Componentes UI**

Componentes de interfaz de usuario.

### **CU**

Caso de uso.

### **DAO**

Clases de acceso a datos.

### **Hibernate**

Hibernate es el puente entre nuestra aplicación y la BD, sus funciones van desde la ejecución de sentencias SQL hasta la creación, modificación y eliminación de objetos persistentes.

### **HTML**

Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986.

### **Interface**

Conexión entre dos dispositivos de hardware, entre dos aplicaciones o entre un usuario y una aplicación que facilita el intercambio de datos.

### **Internet**

Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

### **Intranet**

Una red interna, conectando a múltiples usuarios a través de la tecnología de Internet, usualmente dentro de una organización. Una Intranet es típicamente usada para incrementar el acceso a la información, comunicación, y trabajo entre usuarios individuales y la organización.

### **Información:**

Agregación de datos que tiene un significado específico más allá de cada uno de éstos. Un ejemplo: 1, 9, 8 y 7 son datos; 1987 es una información. La información ha sido siempre un recurso muy valioso, revalorizado hoy más aun por el desarrollo y la expansión de las Tecnologías de la Información y de las Comunicaciones.

### **JSF**

Es un Framework de desarrollo basado en el patrón Modelo Vista Controlador.

### **JSP**

Es una tecnología Java que permite a los programadores generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento web.

En las JSP se escribe el texto que va a ser devuelto en la salida (normalmente, código HTML) incluyendo código java dentro de él, para poder modificar o generar contenido dinámicamente.

### **Microsoft**

Compañía que manufactura los sistemas de operación DOS y Windows. Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

### **PHP**

Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes. Se caracterizan por ser tecnología (open source), y por su fácil desarrollo y mantenimiento. Existe mucha documentación en Internet.

### **Proceso**

Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.

### **Red**

Una red es dos o más computadoras que están físicamente conectadas con las otras y capaces de compartir información.

### **Software**

Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

### **Sistema**

Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a determinado objeto.

### **Sitio Web**

Sistema de computación que corre un servidor Web y que se ha establecido para editar documentos en Web.

### **Spring**

Spring es un marco de trabajo para aplicaciones J2EE que pretende reducir el esfuerzo de desarrollo para la construcción de aplicaciones.

### **UML y UML2.0**

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

La diferencia mas notable entre UML y UML 2.0 es que el primero es orientado a objetos y el segundo es orientado a procesos, lo cual muestra su principal ventaja en que en el mundo real la mayoría de las cosas son procesos.

### **Web (WWW)**

Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.